

## **Treball final de grau**

**Estudi: Doble titulació GEEIA - GEE**

**Títol: Sistema de control del consum de potència d'un mini-aerogenerador d'eix horitzontal de 2kW.**

**Document:** 1. Memòria

**Alumne:** Adrià Quintana Tarradas

**Tutor:** Lluís Pacheco Valls

**Departament:** ATC

**Àrea:** ATC

**Convocatòria (mes/any):** juny/2023

## ÍNDEX

1. INTRODUCCIÓ .....	8
1.1. Antecedents.....	8
1.2. Objecte .....	9
1.3. Especificacions i abast .....	9
2. FUNCIONAMENT DEL SISTEMA.....	11
2.1. Generació.....	12
2.1.1. Millores del mecanisme de frenat .....	16
2.1.2. Sistema de mesura de potència generada.....	17
2.2. Sistema d'emmagatzematge d'energia .....	19
2.2.1. Gestió del baix consum .....	22
2.3. Frenat elèctric.....	23
2.3.1. Introducció i justificacions .....	24
2.3.2. Maquinari del frenat elèctric.....	26
2.3.3. Estructura de control del frenat elèctric.....	27
2.3.4. Sistema d'acumulació de potència del frenat.....	28
2.4. Carregues del sistema.....	31

2.5.	Sistema de control de potència.....	33
2.5.1.	Variables de control i condicions.....	33
2.5.2.	Estats i accions.....	36
2.5.3.	Organigrama lògic .....	37
2.5.4.	Estratègia i justificacions .....	47
2.6.	Maniobra d'emergència .....	53
2.6.1.	Causes de l'aturada d'emergència.....	53
2.6.2.	Estats i accions.....	54
2.6.3.	Variables de control i condicions.....	54
2.6.4.	Organigrama lògic .....	57
2.6.5.	Estratègia i justificacions .....	59
2.7.	Maniobra d'error .....	62
2.7.1.	Error del sistema.....	63
2.7.2.	Estats i accions.....	64
2.7.3.	Variables de control i condicions.....	65
2.7.4.	Organigrama lògic .....	67
2.7.5.	Estratègia i justificacions .....	68

3.	INSTAL·LACIÓ ELÈCTRICA .....	72
3.1.	Cables conductors .....	72
3.2.	Proteccions.....	75
3.3.	Compensació de la potència reactiva .....	76
3.4.	Posada a Terra.....	76
3.5.	Quadre elèctric .....	77
4.	CIRCUIT DE CONTROL.....	78
4.1.	Criteris d'elecció dels components.....	78
4.1.1.	Elecció del dispositiu de control.....	79
4.1.2.	Elecció del elements de mesura .....	79
4.1.3.	Elecció dels elements de comunicació.....	79
4.2.	Dispositiu de control .....	80
4.3.	Sensors .....	81
4.4.	Relés .....	83
4.5.	Comunicacions .....	83
4.6.	Maquinari del sistema de mesura de potència generada .....	85
4.7.	PCB del sistema .....	86

5.	PROGRAMARI DEL SISTEMA DE CONTROL.....	89
5.1.	Introducció.....	89
5.2.	Sistema multitasca.....	89
5.3.	Funció principal del programa.....	91
5.4.	Mecanisme de control d'execució del codi.....	93
5.5.	Programació del control de potència consumida.....	94
5.5.1.	Programació del control de la bomba .....	95
5.6.	Programació del frenat elèctric .....	96
5.7.	Programació de la maniobra d'emergència.....	102
5.8.	Programació de la maniobra d'error.....	106
5.8.1.	Diagnòstics del sistema .....	106
5.8.2.	Aturada d'error.....	109
5.9.	Adquisició de dades.....	111
5.9.1.	Lectura de la potència generada.....	111
5.9.2.	Lectura dels sensors connectats al convertidor analògic digital .....	112
5.10.	Programació del dispositiu de càlcul de potència generada. ....	118
5.11.	Programació del baix consum .....	124

6.	COMUNICACIONS.....	127
6.1.	Xarxa I2C Multimestre .....	127
6.1.1.	Introducció i justificacions de la xarxa d'I2C.....	128
6.1.2.	Mode mestre.....	131
6.1.3.	Mode esclau .....	135
6.1.4.	Test de comunicacions I2C.....	140
6.1.5.	Comunicació I2C amb el sistema de mesura de potència generada.....	141
6.2.	Xarxa del Modbus.....	143
6.2.1.	Integració de la rutina d'esclau de Modbus.....	144
6.2.2.	Funcions del Modbus.....	144
6.2.3.	Variables d'interacció amb el PLC .....	148
7.	EXPERIMENTS DEL PROJECTE .....	153
8.	RESUM DEL PRESSUPOST.....	154
9.	CONCLUSIONS .....	155
10.	RELACIÓ DE DOCUMENTS.....	157
11.	BIBLIOGRAFIA .....	158
12.	GLOSSARI.....	159

A.	CODI .....	162
A.1.	Codi principal del PIC.....	162
A.2.	Llibreria del Modbus.....	192
A.2.1.	Capçalera del Modbus.....	192
A.2.2.	Codi del Modbus.....	194
A.3.	Codi del sistema de mesura de potència generada.....	202
B.	CÀLCULS DE LA INSTAL·LACIÓ.....	208
B.1.	Càlculs de dimensionament dels conductors .....	208
B.2.	Càlculs de dimensionament del condensador de reactiva.....	209
B.3.	Càlcul de posada a terra .....	210
C.	EXPERIMENTS EN EL LABORATORI.....	211
C.1.	Experiments de la xarxa I2C .....	211
C.2.	Experiments amb el Modbus.....	214
C.3.	Proves amb el dispositiu de potència generada .....	217
C.4.	Proves del convertidor analògic a digital amb el mode mitjana de ràfega .....	218
C.5.	Proves del frenat elèctric .....	221
C.6.	Lectura dels sensors de corrent.....	222

---

C.7. Proves de control dels relés.....	224
D. POSADA EN FUNCIONAMENT.....	226



## 1. INTRODUCCIÓ

Per dur a terme l'explicació detallada del projecte descrit a la present memòria, que consisteix en control del consum de potència d'un aerogenerador, i amb la finalitat de comprendre'l millor, primer de tot cal detallar els antecedents del projecte, seguidament s'ha de descriure l'objecte i finalment l'abast. Aquests tres punts anteriors queden descrits en els següents apartats.

### 1.1. Antecedents

El sistema de control del consum de potència, que queda detallat en aquesta memòria, forma part de l'aerogenerador de dos kilowatts que s'està desenvolupant en el projecte llavor de la Universitat de Girona. Aquest projecte de mini-eòlica queda ubicat l'ECOgranja Vilà del veïnat de Pols, situat al municipi alt-empordanès d'Ordis, en la Figura 1, es pot veure on queda emplaçat el molí de vent. Aquest projecte es va iniciar a partir d'un ajut de la Generalitat de Catalunya, per tal de poder comprar material i desenvolupar una turbina eòlica de dos kilowatts adaptada a diferents règims de vent. Inicialment, no existia cap element elèctric i electrònic, en l'aerogenerador, exceptuant l'alternador i el regulador, és a dir, s'ha hagut de desenvolupar pràcticament tot el sistema. El projecte llavor consta de diferents parts: l'estructura mecànica, el mecanisme de Pitch, la part elèctrica i electrònica.

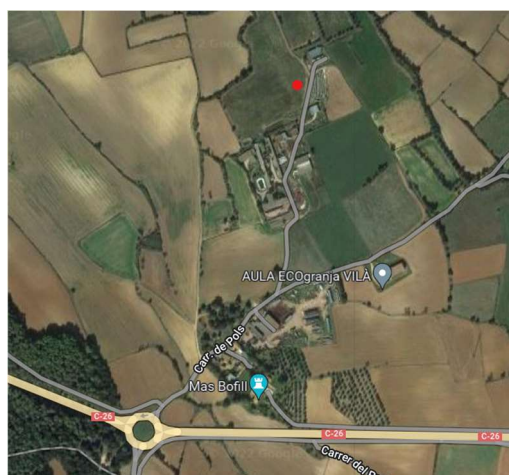


Figura 1. Emplaçament del molí de vent.

Altrament, en el conjunt del projecte llavor, aquest treball final de grau està centrat en el sistema de control del consum de potència, que es duu a terme amb un microcontrolador. Aquest queda interconnectat amb els altres microcontroladors, que controlen i regulen altres processos del molí, en una xarxa multi mestre amb I2C autònoma. A més, el dispositiu queda

interconnectat, actuant com a esclau, en una xarxa Modbus amb un PLC mestre que permet adquirir dades i tenir un control total del consum de potència.

## 1.2. Objecte

L'objecte d'aquest treball consisteix a desenvolupar un sistema de control del consum de potència fiable, i que treballi a la perfecció de manera coordinada amb els altres dispositius del molí, perquè l'aerogenerador sigui el més eficient possible, robust a pertorbacions i segur per l'entorn i les persones.

## 1.3. Especificacions i abast

El sistema de control del consum de potència es duu a terme amb un microcontrolador PIC. Aquest queda interconnectat amb dos microcontroladors més, que controlen i regulen altres processos del molí, en una xarxa multimestre amb I2C autònoma. A més, els tres dispositius queden interconnectats, actuant com a esclaus, en una xarxa Modbus amb un PLC mestre que permet adquirir dades i tenir un control total de l'aerogenerador, que es pot observar en la Figura 2.



Figura 2. Estructura mecànica de l'aerogenerador a l'Ecogranja Vilà.

L'aerogenerador va connectat de forma aïllada a la xarxa, i la seva potència es destina: en il·luminar uns focus de 600 W a la granja, en accionar una bomba de 800 W que omple dos

dipòsits d'aigua per fins ramaders, i en carregar una bateria de 24 V que permeti el funcionament dels focus i la bomba en cas de no fer vent, i que també es pugui destinar l'energia emmagatzemada per a altres aplicacions, en cas d'excedents. En conseqüència, caldrà dissenyar un sistema perquè el microcontrolador permeti dur a terme la rutina de transferència de l'energia dels diferents elements. En el control de la transferència de potència, també s'ha implementat una aturada d'emergència, perquè pari de forma immediata la generació per evitar danys irreparables, i una aturada d'error, perquè davant d'un error en les comunicacions o en els sensors, la turbina es pari per evitar entrar en situacions crítiques.

Per tal d'aconseguir l'objectiu principal d'aquest sistema, s'han de dur a terme diferents tasques: la programació del microcontrolador pel consum de potència de l'aerogenerador, amb les aturades en casos d'error i emergència, tenint en compte que hi ha d'haver una part de comunicacions, i que en situacions de generació nul·la, el dispositiu de control tingui el menor consum possible; una altra tasca que s'ha de tenir en compte és l'estudi i l'elecció dels diferents components elèctrics que interactuen amb el microcontrolador; altrament, es confeccionarà un esquema elèctric del molí. Finalment, es dissenyarà una PCB per poder interconnectar el microcontrolador amb els sistemes de comunicacions i els circuits de potència.

## 2. FUNCIONAMENT DEL SISTEMA

En el present capítol s'explica com funciona el molí, tenint en compte tots els elements de la instal·lació, i les diferents situacions que es poden donar. D'aquesta manera, s'ha dividit el capítol en set apartats: en el primer apartat es detalla la generació del sistema i com el sistema percep la potència generada, tenint en compte la interacció dels components que intervenen en la generació, en el segon apartat, s'explica el mecanisme d'emmagatzematge d'energia i com el sistema de control gestiona l'energia de l'element de control, per allargar al màxim possible la bateria en cas d'haver un llarg període sense generació; en el tercer apartat s'exposa el funcionament del frenat elèctric, que és un element de seguretat a la instal·lació i per garantir el bon funcionament del sistema; en el quart es donen detalls de com són les càrregues que el sistema hi destina energia; en el cinquè es mostra de forma acurada com funciona la transferència de potència; en el sisè com actua el sistema davant d'una emergència; i finalment, en l'últim apartat s'informa com el sistema entra en error i es para donada la situació. Per facilitar la comprensió de com està dissenyada la instal·lació, i que siguin més entenedores les explicacions, en la Figura 3 hi ha un esquema de com el sistema està implementat.

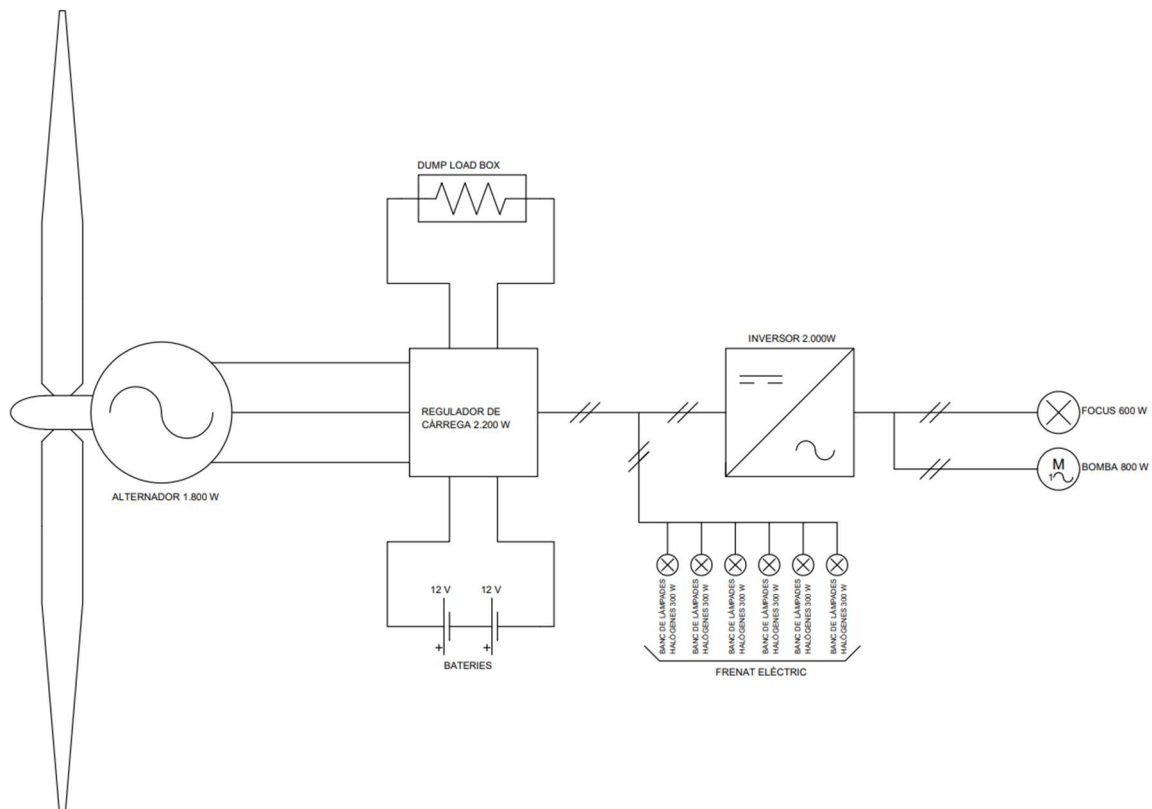


Figura 3. Instal·lació elèctrica de potència.

## 2.1. Generació

A nivell de maquinari, en la generació només actuen dos elements, el generador que transforma l'energia cinètica del vent en energia elèctrica, i el regulador de càrrega, que transforma l'energia elèctrica generada de corrent altern en corrent continu, buscant en tot moment la màxima transferència de potència generada, i a la vegada controla la càrrega de les bateries.

El generador és de la marca Ginlong Technologies. El seu rotor va acoblat directament a l'eix de gir de l'aerogenerador, que és horitzontal, i es mou per l'efecte del vent sobre tres pales, l'angle de les quals és modificable respecte la direcció del vent. En conseqüència, s'elimina la caixa d'engranatges, i es minimitzen les pèrdues mecàniques.

La màquina elèctrica escollida, segons les dades subministrades pel fabricant, es tracta d'un alternador síncron trifàsic d'imants permanents, i està format per vuit parells de pols, generant una potència nominal de 1.800 W, i una tensió de línia nominal de 400 V. El generador té la seva sortida connectada en estrella, i en condicions nominals el motor té una velocitat de gir de 480 rpm i un parell de 44,5 N·m. Les Figures 4 i 5, subministrades pel fabricant, representen, la corba de la potència generada en funció de la velocitat de rotació, i la tensió de sortida en funció de la velocitat de rotació, respectivament. Com es pot observar en les gràfiques, en tenir una velocitat baixa d'arrencada degut al seu parell resistiu baix, el sistema permet iniciar la generació amb vents poc intensos.

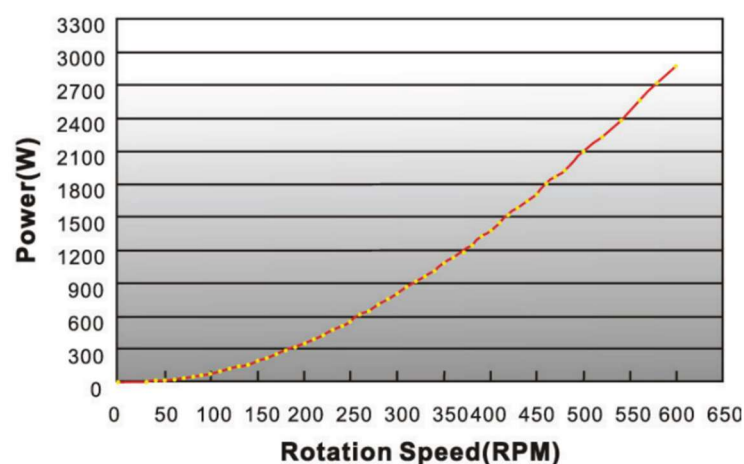


Figura 4. Potència generada en funció de la velocitat de rotació.

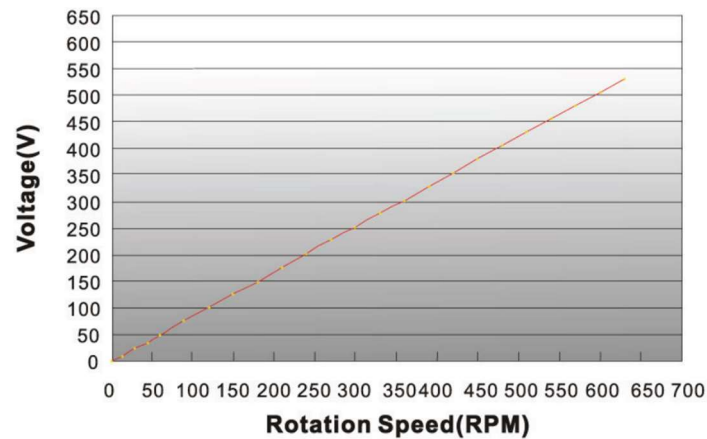


Figura 5. Tensió de sortida en funció de la velocitat de rotació.

A més a més, el generador està capacitat per treballar la intempèrie a l'emplaçament on està situat, suportant la climatologia i la pols del terreny, sense risc de danyar-se, ja que la carcassa té una IP66, i les propietats mecàniques dels materials, com pot ser la carcassa i l'eix que estan fabricats d'alumini amb tractament tèrmic i pintura amb impressió. En la Figura 6, es mostra com és físicament l'alternador.



Figura 6. Alternador de la turbina eòlica.

L'altre component que és clau per la gestió de la generació és el regulador de càrrega. El regulador escollit és de la marca TechnoSpin Inc, concretament, és el model PowerSpin TSW 2200. Segons el fabricant, aquest regulador pot funcionar tant en una instal·lació connectada a la xarxa, anomenada configuració grid-tie installation, com en una instal·lació en illa com és el cas de del molí que s'està dimensionant, que és una configuració off-grid. La configuració en illa és la mateixa que la indicada a la Figura 7, però en lloc de tenir connectades quatre bateries de dotze volts, formant un conjunt d'acumulació de 48 V, està format per dues bateries creant un sistema de 24 V. En aquest cas, a la configuració del BMS del regulador se li haurà d'especificar el sistema de bateries que té connectat a través d'uns interruptors dip. Aquests interruptors dip poden modificar de manera total el treball del regulador. En el cas del control

de càrrega de la bateria, el procés ve programat de fàbrica, però a través d'aquests switch de parametrització que incorpora el regulador, es pot modificar el procés de càrrega. D'aquesta manera, se li indica el tipus de bateria segons la tecnologia: si és de liti, GEL, AGM, etc. I també se li marca el nivell de tensió de les bateries del sistema, és a dir, el número de bateries de 12 V que incorpora. Altrament, hi ha un dip que configura si el sistema treballa en illa o connectat a la xarxa.

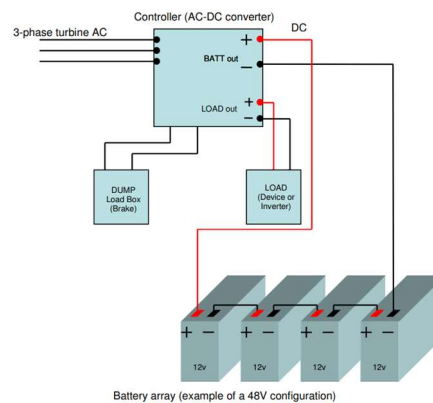


Figura 7. Configuració del sistema en illa.

El regulador rep l'energia del generador a partir d'una entrada formada per tres connectors, on a cada un d'ells se li connecta una fase del sistema. El regulador passa de corrent altern a corrent continu, i posteriorment, en funció de la demanda energètica transmet l'energia als dispositius connectats a ell. Aquest controlador de càrrega té connectats: una sortida de bateries, un inversor, que és amb el que es connecten les càrregues de la instal·lació, i un frenat elèctric, anomenat dump load box.

Mecànicament, aquest controlador de turbina eòlica incorpora un indicador LED, que senyalitza l'estat de funcionament del regulador de manera simple, intuïtiva i clara. I es pot identificar automàticament si hi ha producció de vent, amb un LED verd encès, o si el frenat està accionat, amb un LED vermell encès, o si no hi ha generació, amb les dues làmpades apagades.

Una de les avantatges que incorpora el regulador seleccionat és els seus mecanismes de protecció: pot actuar contra curtcircuits i sobrecàrregues, presenta una protecció antireversa, en altres paraules, evita que el corrent de la bateria quan està carregada vagi cap al generador, si aquest no produeix electricitat; pot dur a terme l'apagat total del generador; evita la descàrrega automàtica de les bateries per fugues de corrent o un excedent molt gran de corrent; i pot la recuperar la tensió en cas d'un sot. Per tant, és un element d'alta fiabilitat i que

pot proporcionar un efecte d'ús superior. De totes maneres, cal destacar que el generador té una potència nominal de 1.800 W, i el regulador de càrrega pot treballar fins a 2.200 W. Per tant, en condicions normals de generació, mai no s'arribarà al límit de funcionament del regulador, en cas de produir-se actuaran els mecanismes de protecció, sigui el control del Pitch, el frenat, etc. I si aquests no actuen, poden saltar les proteccions. A més a més, en poder treballar amb potències fins a 2.200 Watts, fa que el regulador sigui idoni per accionar les càrregues de l'inversor com la bomba i els focus, i a la vegada permet la càrrega de les bateries.

Aquest regulador de càrrega té un nivell de protecció IP67, per tant, tot i estar instal·lat a recer, el sistema pot funcionar sense danyar-se per les hostilitats de l'ambient.

La tecnologia del regulador està basat en el mètode de seguiment del punt de màxima potència o MPPT. La manera com es duu a terme el seguiment del punt màxim de potència, es basa a partir de les lectures de la potència instantània generada. Hi ha un mecanisme de control de càrrega que segueix la corba de potència màxima, que està emmagatzemada en una Look-up table en el regulador. L'avantatge d'aquest tipus de regulador és que està desenvolupat per superar les incerteses de les corbes de potència. A més a més, es controla la tensió de sortida cap a l'inversor perquè sigui constant, per evitar danys en el sistema per sobretensions.

A l'hora d'instal·lar el regulador, com es mostra en la Figura 8, cal tenir en compte la connexió dels terminals d'entrada i els de sortida, en cas de confusió, el regulador no quedarà danyat gràcies a que les proteccions haurien actuat.

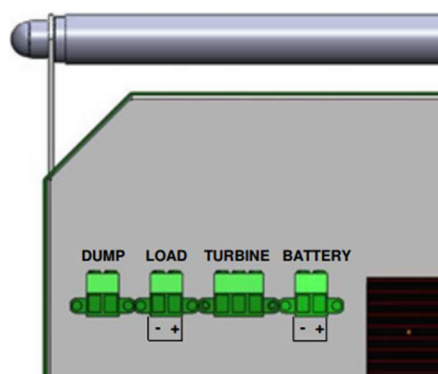


Figura 8. Regulador de càrrega de l'aerogenerador.

Quan el generador es curtcircuita, el rotor per efectes electromagnètics queda bloquejat i és molt complicat que giri. De tal manera que sempre que es vulgui evitar que el rotor giri: sigui



perquè no hi ha generació i es vol fer manteniment, o en cas d'emergència que hi hagi el mínim de generació possible, per assegurar la integritat de l'estructura, convé parar la generació clavant el rotor. Aquest curtcircuit es duu a terme a través de dos relés en el quadre elèctric, connectats a través de borns.

A continuació, en els següents subapartats, es detalla els motius pels quals s'ha decidit implementar millores en el sistema de frenat, i com el sistema de control mesura la potència generada.

### **2.1.1. Millores del mecanisme de frenat**

Un altre punt per on el regulador transfereix energia, és en el frenat elèctric que disposa de forma intrínseca. Aquest frenat només actua quan la bateria està carregada, i el molí segueix en el règim de generació. Quan la bateria està descarregada, tots els excedents energètics entre la potència demandada i al consumida es destinen a carregar la bateria, equilibrant la potència generada amb la demandada. Si la bateria està carregada, automàticament el regulador desconnecta la bateria de l'alimentació, i connecta el frenat, de tal manera que els excedents passen a transferir-se en el frenat, per tal d'equiparar la potència generada i la demandada. De la manera com està configurat el sistema de bateries, com que són dues bateries de 12 V, quan es detecten 28,8 V, cada bateria té 14,4 V, el sistema considera que la bateria està carregada, i connecta el frenat de fàbrica. Aquest sistema de frenat presenta varis inconvenients: el primer és que aquest frenat elèctric s'hauria d'activar quan la potència generada és molt superior a la demandada. Si durant aquesta situació la bateria està en procés de càrrega, i la demanda de les càrregues que connecta l'inversor és nul·la o no és la suficient per cobrir la potència generada, si no es duu a terme cap acció, aquest excedent va cap a la bateria de manera que es pot originar un sobrecorrent que pot danyar-la, en conseqüència, per evitar un corrent perjudicial, el frenat hauria d'accionar-se. El segon inconvenient que presenta aquest sistema és que l'energia que dissipa no és regulable, és a dir, si la bateria està carregada i encara segueix el règim de generació, el sistema dissipa tota l'energia que està generant, siguin 300 W, o 1.200 W, a través un trossetat PWM del senyal. Aquest fet no és desitjable perquè no existeix cap rampa de càrrega, si tota l'energia generada es consumeix de cop, el sistema frena d'una forma molt brusca arribant a afectar l'estructura mecànica del molí. Per solucionar aquest problema el sistema hauria d'anar connectant càrrega de forma progressiva fins a arribar a frenar el sistema de forma més suau. Finalment, el pitjor inconvenient que presenta aquest sistema i que està relacionat amb el primer problema esmentat, és la manca de mesures que allarguin la vida útil de la bateria. Com que

les bateries poden arribar a absorbir tota la generació, l'energia que poden acumular en poc temps pot arribar a ser molt gran. Conseqüentment, el sistema no està dissenyat tenint en compte la profunditat de càrrega i de descàrrega de la bateria, o els valors de corrent, que tot i ser assumibles per la bateria, la seva circulació li pot acabar minvant la vida útil. És a dir, si el regulador de forma habitual carrega la bateria a un corrent pròxim al seu nominal, la vida útil de la bateria es veu escurçada. Altrament, com que l'energia generada es destina a carregar la bateria, implica que la bateria té una del càrrega al 100%, això significa que el sistema transfereix tota l'energia per carregar la bateria fins al seu màxim, i un cop adquirit tal nivell de càrrega, l'energia es dissipa en el frenat. Pel que fa a la profunditat de descàrrega, aquesta és aleatòria, és a dir, en cas de no haver vent, quan la bateria està carregada alimenta les càrregues fins que la bateria estigui descarregada, o fins que no es torni a carregar per efecte del vent, demostrant que la profunditat de descàrrega és aleatòria, i no és gens favorable per la vida útil de la bateria. Per tant, caldria aplicar un control a la descàrrega de la bateria, perquè aquesta tingui una llarga vida útil i a la vegada emmagatzemi la suficient energia. Aquest frenat de fàbrica tampoc no permet parar la generació en cas que hi hagi algun defecte mecànic, com el trencament d'una pala, o vibracions excessives.

Per tal de millorar el frenat de fàbrica, s'ha dissenyat un sistema de frenat, que s'explica el seu funcionament en l'apartat 2.3.

### 2.1.2. Sistema de mesura de potència generada

En el present subapartat es descriu com s'adquireixen les dades de la potència generada de l'aerogenerador.

Per conèixer la potència que s'està generant en tot moment, cal determinar els valors eficaços de tensió i de corrent. La mesura de potència generada, s'adquireix a través d'un sensor de corrent, que els valors de sortida que proporciona són instantanis i no eficaços. Conseqüentment, a partir dels valors adquirits, cal obtenir el seu valor eficaç, que es calcula a partir de l'Equació 1.

$$X_{rms} = \sqrt{\frac{1}{T} \cdot \int_{t_0}^{t_0+T} x^2(t) \cdot dt} \quad (\text{Eq.1})$$

Referent al senyal de tensió, els valors del voltatge generat poden arribar a pics instantanis de 480 Volts, tal com es mostra en la gràfica de la Figura 5, conseqüentment, caldria dissenyar

un sistema de sensorització o de mesura, perquè els nivells de voltatge de sortida de l'alternador s'adeqüin als permesos pel convertidor analògic/digital del microcontrolador. La solució més adequada i amb la que s'ha optat per mesurar la tensió, es basa en els principis de funcionament del generador síncron trifàsic, d'aquesta manera s'evita dissenyar un circuit de sensorització i condicionament per obtenir les dades de tensió. Tenint en compte que la força electromotriu generada i la freqüència elèctrica depenen de la velocitat angular del rotor. Així doncs, si s'aconsegueix el valor de la freqüència elèctrica, es pot obtenir el valor de la tensió que s'està generant, amb la velocitat angular rotòrica com a enllaç. Per tant, en el mateix microprocessador que calcula la potència generada, també es calcula el valor eficaç de tensió, que permet determinar la potència aparent produïda. Es pot conèixer la freqüència del senyal de la intensitat, i a partir de l'Equació 2, es pot assolir la velocitat de sincronisme, que equival a la de gir del rotor.

$$n_{mec} = \frac{60 \cdot f_{elèctrica}}{Pp} \quad (\text{Eq.2})$$

Atès que la màquina té vuit parell de pols, per convertir de freqüència elèctrica a velocitat mecànica s'ha de multiplicar per 7,5.

$$\varepsilon_a = K \cdot n \quad (\text{Eq.3})$$

Tan bon punt es coneix la freqüència, a partir de la corba de la Figura 5, on el fabricant es basa en l'Equació 3, s'obté el voltatge de sortida. En l'alternador escollit, a partir de la gràfica, queda demostrat com K és igual a 0,8333.

En un generador síncron, la potència activa i reactiva depèn de la càrrega connectada. Com que en els receptors de càrrega s'ha connectat un condensador per compensar la potència reactiva, i el factor de potència és molt proper a 1, afegit en la capacitat de l'inversor de reduir la potència reactiva, la potència aparent s'ha considerat com a potència activa. Per tant, a partir d'un factor aproximat i el rendiment del generador, el dispositiu troba la potència activa generada. S'ha de tenir en compte que els factors en el programa, s'han preestablert amb uns valors inicials, però en la posada en funcionament, aquests coeficients poden variar.

La mesura de potència generada es descentralitza del dispositiu principal de control, és a dir, el microcontrolador PIC no calcula la potència generada, aquesta operació es transfereix en un altre microprocessador. El microprocessador que adquireix els valors instantanis de corrent

i en calcula la potència generada és una ESP32. I queda interconnectat amb el PIC a través de la xarxa I2C multimestre, actuant com a esclau. El PIC controla de manera total aquest dispositiu, decidint quan adquirir les dades de potència, i el pot posar en mode dormir, és a dir, en baix consum, i despertar-lo. Les justificacions d'aquesta descentralització i de l'elecció d'aquest microprocessador, es detallen en l'Annex E.

## 2.2. Sistema d'emmagatzematge d'energia

El sistema d'emmagatzematge d'energia està format per dues bateries de 12 V de GEL, connectades en sèrie. El motiu d'haver optat per aquesta tecnologia, enlloc d'altres com poden ser les de ió liti, és perquè el seu preu és molt baix i el seu rendiment és alt. A més, és idònia per aplicacions de baixa potència, i permeten llargs temps de descàrrega gràcies a la seva alta resistència a descàrregues profundes i repetits cicles de descàrrega. A la vegada, ofereix una combinació immillorable de rendiment en cicles de càrrega i descàrrega, és a dir, es perd poca potència en l'acció de carregar i descarregar la bateria. Un altre factor molt favorable a l'aplicació del molí és que permet un alt pic de corrent d'arrencada, ideal per accionar la bomba. A més a més, no cal realitzar accions de manteniment, com afegir aigua destil·lada, com és el cas de les bateries de plom àcid. En l'entorn on s'ha instal·lat la turbina, s'ha de tenir en compte que hi ha períodes de forta generació, provocant que l'energia apareix de forma sobtada, i implicaria que el sistema es carregués de cop. Aquest punt que s'ha de tenir en compte en el disseny del frenat elèctric, per tal de limitar el corrent de càrrega.

Les bateries són del model Monobloc de Gel 12v/160 de cicle profund de la marca Blackbull. En la Taula 1 hi figuren les propietats més importants de la bateria, proporcionades pel fabricant, i que s'han tingut ne compte al llarg del projecte.

Paràmetres	Valors
Corrent màxim de càrrega en aplicació Stand-by (Ampers)	100
Corrent màxim recomanable en aplicació cíclica (Ampers)	45
Tensió de la bateria (Volts)	12
Capacitat a C100 (Ampers-hora)	160
Capacitat a C20 (Ampers-hora)	135
Tensió màxima de càrrega (Volts)	14,4

Taula 1. Propietats de la bateria Blackbull Monobloc de Gel 12v/160.

El regulador de càrrega carrega la bateria amb un procediment format per quatre etapes. Els dip-switchs del número u al tres es disposen de tal manera que indiquen del sistema està format per dues bateries de 12 V, mentre que els que van del quatre al sis permeten escollir que les bateries són de Gel.

La primera etapa, anomenada Bulk, la bateria accepta tot el corrent proporcionat pel regulador de càrrega eòlic, amb la finalitat de carregar la bateria el més ràpid possible. Durant aquesta etapa, la tensió de la bateria augmenta de forma progressiva fins a assolir un valor configurat, que en el cas del regulador és de 14,3 V per bateries de gel. La segona etapa és l'absorció, que s'hi arriba quan la tensió de càrrega obté el valor que s'ha establert en el regulador. Durant aquesta etapa, la tensió de la bateria es manté constant, a través d'una variació amb PWM, mentre el corrent de càrrega poc a poc va disminuint a mesura que la bateria es carrega. Aquesta etapa assegura que la bateria estigui totalment carregada i a punt per subministrar energia. D'aquesta manera es protegeix la bateria, evitant sobreescalfament. En altres tipus de bateria, la següent etapa és una acció de manteniment, anomenada flotació. En les bateries de gel no s'incorpora perquè les pot danyar. Durant aquest període, el voltatge de la bateria minva una mica per evitar la sobrecàrrega, i es manté en un valor constant menor que el de l'etapa anterior. En el cas de la bateria de gel, l'etapa que procedeix a l'absorció és l'equalització. Aquesta etapa també és de manteniment de la bateria, i no es realitza sempre, sinó que també es duu a terme de forma periòdica. Bàsicament, consisteix en una càrrega de reforç per activar l'electròlit, anivellar les tensions de les cel·les i completar les reaccions químiques. En la Figura 9, es mostra com varia la tensió en el temps en les diferents etapes de de càrrega de la bateria.

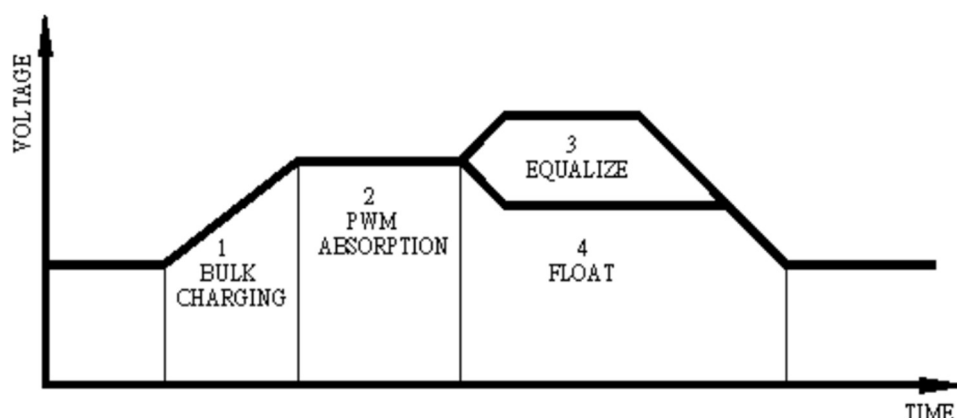


Figura 9. Tensió de la bateria al llarg del procés de càrrega.

El sistema electrònic de control queda dotat amb una sensorització adient, que a partir de la tensió de les bateria, es coneix l'estat de càrrega, que segons el nivell de càrrega que tingui, el sistema de control de consum accionarà les càrregues, o aplicarà el frenat elèctric.

A la corba de la Figura 10, proporcionada pel fabricant indica com varia la tensió en funció de l'estat de càrrega de la bateria, i el temps de càrrega. Juntament amb el corrent de càrrega de la bateria en funció del temps total de càrrega sense danyar-se. A pesar del que indica el fabricant, que la bateria aguanta descàrregues profundes i repetits cicles de descàrrega, el sistema de control de consum evitarà que la descàrrega sigui total, amb la finalitat d'allargar la vida útil a la bateria.

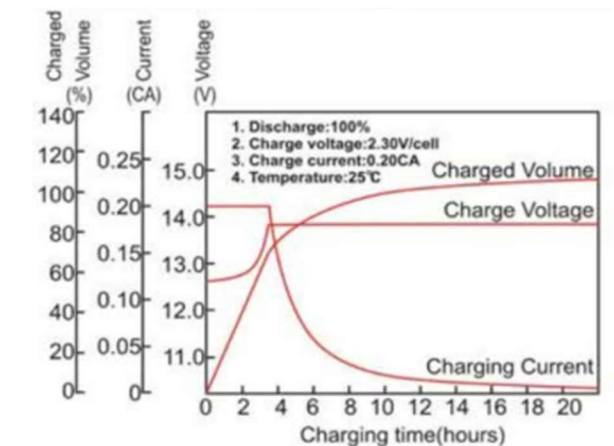


Figura 10. Característiques de càrrega d'una cel·la de la bateria

A la gràfica de la Figura 11, indica el temps d'autodescàrrega de la bateria en funció de la temperatura, és a dir, el temps en mesos en que es descarregaria la bateria, sense tenir cap càrrega connectada. La informació que aporta aquesta corba és clau per comprendre el comportament de les bateries en períodes on la generació de vent és nul·la, en altres paraules, quin percentatge es descarregaria en període on el vent és escàs.

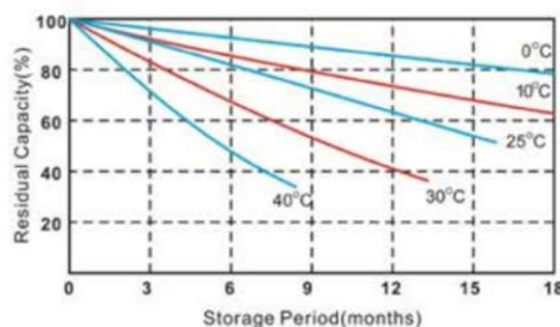


Figura 11. Característica d'autodescàrrega de les cel·les de la bateria.

De la manera com treballa la bateria en aquest sistema, es considera un ús cíclic, perquè contínuament s'està carregant i descarregant. En aquest cas, tal com especifiquen altres fabricants com: Ultracell, Exide, o Trojan, el corrent de càrrega per a bateries de gel de 12 V, no es recomana un corrent de càrrega superior a 45 A, tot i que el corrent nominal sigui superior. Per tant, a través del mecanisme de frenat, el corrent de càrrega es limita per allargar la vida útil.

Una altra aplicació de la bateria és alimentar altres càrregues a partir de pinces de cocodrill. Un exemple seria subministrar energia a la bateria d'un cotxe o un tractor per fer l'arrencada. Una dels avantatges que disposa aquesta bateria és que per la naturalesa del seu electròlit, que és gelificat, es pot utilitzar en posició horitzontal i vertical, fet que permet transportar-la de manera segura, i sense vessaments. A més a més, mecànicament disposa d'unes nanses. L'únic inconvenient d'utilitzar la bateria per a aquesta finalitat és que no es controla la profunditat de descàrrega mentre s'està fent servir, perquè la bateria queda desendollada i el sensor no pot monitorar l'estat de càrrega.

### **2.2.1. Gestió del baix consum**

El sistema de control, format pel PIC i els sensors connectats a ell, queda alimentat gràcies a l'energia generada per l'aerogenerador. En períodes de generació, el PIC pot funcionar amb un consum nominal. L'energia que rep mentre hi hagi generació, no és gens crítica, i per tant, amb tots els perifèrics funcionant, el rellotge intern amb la freqüència de treball al màxim i el mode de programa que executa, segons el fabricant, el PIC té un consum màxim de 800 mW, demostrant que el consum energètic dels diferents dispositius no és crític, perquè sempre poden rebre energia, i inclòs, pot haver situacions on l'energia generada s'hagi de perdre en el frenat elèctric. Però, quan la generació és nul·la, el consum energètic ha de quedar de la manera més optimitzada possible, per tal d'allargar al màxim el funcionament dels equips. Ja que un consum de 800 mW, connectat a la bateria de manera constant, pot provocar una pèrdua energètica, que en casos que la climatologia sigui molt extrema, i que no faci gens de vent, en un llarg període de temps, pot provocar que l'energia es perdi en alimentar el PIC, escurçant el temps amb el que al bomba pot funcionar sense generació. Aquest criteri d'optimitzar el consum quan no hi ha generació, perquè els dispositius de la turbina consumeixin el mínim, s'aplica en tots els dispositius de la turbina.

El PIC té diferents modes de baix consum, però en aquest cas s'escull el mode sleep o dormir. En aquest mode, el processador s'atura temporalment, i s'apaguen els perifèrics que no

s'utilitzen. Només el rellotge de baix consum i alguns mòduls perifèrics específics romanen actius. D'aquesta manera, el PIC es pot despertar, en qualsevol moment, mitjançant una interrupció, i es pot posar a dormir per programa. En el mode de baix consum, el sistema funciona a 25  $\mu\text{W}$ . Per tant, tenint en compte que tant el sistema de control de l'angle d'inclinació de l'angle de les pales, com el sistema de monitoratge, optimitzen el seus respectius funcionaments per consumir al mínim, i que el sistema de control de potència només es despertarà en les interrupcions que intervenen quan no hi ha generació, i un cop realitzades es tornarà a posar a dormir, els elements de control de la turbina passen a tenir un consum molt menor. Així doncs, en cas que no hi hagi generació, l'energia que surt de la bateria per alimentar els dispositius de control és molt menor, allargant la durada de la bateria.

Tant el sistema de control de l'angle d'inclinació de les pales, com el de potència, i el sistema de monitoratge de la turbina detecten que no hi ha generació a causa del vent flux, la velocitat de rotació i la potència generada són pràcticament nul·les. Quan els tres dispositius confirmen aquesta situació, els dispositius electrònics de control es posen a dormir. En canvi, quan es detecta generació per part dels tres dispositius de control del molí sempre estan actius i no es posen a dormir.

Quan el PIC que controla la potència transferida està dormint, pot ser despertat per qualsevol interrupció del programa causada pels perifèrics o les xarxes de comunicació. A partir d'aquest punt, el sistema segueix amb el que estava fent. Tal com es mostrarà més endavant, la condició de dormir està al final de bucle, per tant, un cop es desperta el sistema, es farà una volta al bucle, que li permet determinar l'estat dels sensors, desenvolupar els tests i canviar l'estat dels relés.

### **2.3. Frenat elèctric**

Per tal d'explicar el frenat elèctric de la manera més clara possible, s'ha dividit aquest apartat en quatre subapartats: en el primer es comenten totes les justificacions sobre el funcionament del frenat, en el segon subapartat es descriu el maquinari que el compona, en el tercer s'explica el control del frenat, i en l'últim subapartat, que està relacionat amb el tercer, es descriu com el frenat afegeix i treu potència, segons el seu funcionament.



### 2.3.1. Introducció i justificacions

En la gestió de la generació, el frenat elèctric és un element primordial. Tot i que, el regulador escollit incorpora un frenat elèctric de fàbrica, com s'ha descrit anteriorment, el frenat elèctric dissenyat en el marc d'aquest treball, millora el frenat del sistema. Val a dir, que el frenat de fàbrica no s'ha eliminat de la instal·lació elèctrica de l'aerogenerador, s'ha deixat com a unitat auxiliar, perquè en cas que quedi danyat el frenat que s'ha dissenyat, no es perdi la protecció bàsica de la bateria que ofereix el frenat de fàbrica.

Cal destacar, que el frenat elèctric funciona en un llaç de control separat al control de consum de càrregues, el motiu es deu a que el frenat ha de funcionar de forma robusta, per tant, no pot dependre de cap estat del sistema.

El funcionament del frenat elèctric es basa en la diferència entre la potència generada i la potència elèctrica demandada. Quan la potència elèctrica demanda és la mateixa que la potència generada, el generador gira a una velocitat constant. Si de cop la potència mecànica per efecte del vent augmenta, i la potència elèctrica demandada es manté constant, o si la potència elèctrica de cop disminueix i la potència mecànica es manté constant, l'alternador comença a accelerar. Si aquesta acceleració no es controla, es podria donar que el generador síncron excedís els seus límits mecànics o els límits elèctrics, arribant-se a donar sobretensions, o inclòs freqüències elèctriques excessives per la instal·lació connectades aigües avall. En conclusió, el frenat elèctric a partir d'aquests fenòmens mencionats, intrínsecs del generador síncron trifàsic, permet que els paràmetres elèctrics i mecànics del generador no excedeixin els seu límits, i en conseqüència no es pateixin danys ni en la instal·lació elèctrica ni en el propi generador. Per evitar l'augment de velocitat i equiparar el balanç de potències, el sistema de frenat connecta la càrrega. En una emergència, la turbina pot ser aturada bruscament per protegir l'estructura mecànica del generador de les vibracions i velocitats excessives. En aquest cas, es connecten sis bancs de resistències per disminuir la velocitat i la potència generada fins a fer-la negligible. Això permet als borns de l'alternador ser curtcircuitats perquè el rotor quedi clavat, gràcies al bloqueig per efectes electromagnètics, i no pugui girar.

El frenat elèctric implementat pot actuar per tres causes diferents: la primera i més important és la protecció mecànica del molí, sigui perquè es percep una velocitat de rotació molt alta, o vibracions excessives pel trencament d'una pala o d'un rodament, la segona és la mencionada en el paràgraf anterior, i és que la bateria està carregada, i l'excedent energètic es perd en el

frenat, i la connexió de càrregues depèn de l'increment de potència; i la tercera es basa en la protecció de la bateria. Aquesta última quan la potència de càrrega de la bateria supera un límit prefixat, que aquest és inferior al nominal, per tal d'allargar la vida útil de la bateria. La diferència entre la potència que es destina en el carregador de la bateria amb el límit prefixat, es transfereix en el frenat elèctric. D'aquesta manera, en limitar la potència a un valor inferior al nominal, i donant una càrrega més lenta, la bateria té una vida útil més extensa.

Tal com s'ha esmentat en la introducció d'aquest apartat, el banc de dissipació d'energia per dur a terme el frenat elèctric, s'executa afegint càrrega resistiva. Una de les maneres seria que aquestes resistències dissipin l'energia en forma de calor, com seria escalfant aigua, o simplement deixant perdre l'energia. La primera opció no s'ha plantejat, i en la casuística que passa en afegir una resistència, amb la qual l'energia de la frenada es perd, apareixen dues opcions: la primera és afegir la resistència a la base del molí on es troba el quadre elèctric, sent aquesta nefasta perquè pot provocar un augment de temperatura a l'espai on es guarda l'electrònica, donant peu a mals funcionaments, i a sobreescalfaments que poden ocasionar pèrdues. I una altra alternativa, que seria afegir la resistència a l'exterior del molí. En aquesta possibilitat, no seria adequada perquè per la configuració del frenat, hi haurien sis resistències de 300 W connectades a l'exterior, i podria ser perillós per les persones i animals de l'exterior. La solució optada passa per instal·lar les resistències a l'exterior de la base del molí, però en forma de làmpada halògena. D'aquesta manera part de l'energia es transmet en forma de llum, i l'escalfament no és tant alt, i de forma empírica l'usuari pot saber que el frenat elèctric actua, fet que implicaria que s'està perdent de manera total o parcial energia de la generació. A més a més, a mesura que l'energia a dissipar va en increment, la llum de les làmpades per l'efecte del PWM és més intensa, que és quan la calor és més elevada.

De la manera com es pot observar en el document dos d'aquest projecte, corresponent als plànols, els sis bancs estan connectats en paral·lel entre ells i amb l'inversor. El motiu de la disposició del frenat dissenyat en aquest bus de contínua, es deu a la capacitat que es pot tenir d'ajustar la càrrega. És a dir, en aquest bus el senyal és de 24 V continu, i com s'ha mencionat anteriorment, l'energia que es transfereix en aquest bus, depèn de la demanda que hi hagi, sempre i quan, la potència generada sigui superior a la consumida. Per tant, gràcies a la capacitat d'ajustar la demanda del frenat i del bus, és idònia aquesta disposició. Si es col·loqués en el terminal on hi ha el frenat de fàbrica, com que el corrent que flueix en aquest bus el gestiona internament el regulador, no es permetria regular el frenat com es voldria, a més a més, cal tenir en compte el fet que la tensió que s'aplica depèn d'un trossejat de tensió del regulador, que provocaria un funcionament anòmal en la commutació dels bancs.

Connectar el frenat dissenyat en el bus de la bateria tampoc no és una bona idea, perquè pertorbaria el funcionament del regulador carregar la bateria. A més a més, només funcionaria pel mode del frenat destinat a la protecció d'una potència de càrrega de la bateria superior a la del llindar. Per la dissipació d'energia en cas de la bateria carregada i una potència generada superior a la consumida, en commutar el frenat es consumiria energia de la bateria, ja que el regulador per si sol, de forma momentània talla l'entrada corrent procedent de la generació, i només es consumiria energia de la bateria sense protegir el sistema, fins que el nivell de tensió de la bateria, baixés per sota del llindar que té configurat el regulador. En aquest moment, el regulador transferiria l'excedent en el frenat de fàbrica, i en el frenat dissenyat. En aquest escenari es perdria el control total del frenat, i el sistema seria molt més ineficient, ja que una part de l'energia de la bateria es perdria de forma absurda.

### **2.3.2. Maquinari del frenat elèctric**

Principalment, el frenat està compost per sis mòduls interruptors MOSFET, que connecten sis bancs de làmpades halògenes funcionant com a resistències.

Les bombetes són de la marca Philips, i com detalla el fabricant i s'ha comprovat de forma empírica en el laboratori, amb una tensió de 24 Volts, consumeixen una potència de 50W. Les làmpades estan connectades amb una configuració de dues branques en paral·lel, i amb tres bombetes connectades en sèrie per ramal. Addicionalment, segons el fabricant, aquestes làmpades són ideals per funcionar amb dimming, que seria un cas similar al frenat elèctric que s'està desenvolupant.

El mòdul commutador escollit està format per dos transistors MOSFET de potència. Un transistor commuta a un altre que habilita el pas de corrent. Per tant, gràcies a aquest sistema d'electrònica de potència, commutant els transistors a alta freqüència, amb un senyal PWM, es pot ajustar la potència que es dissipa en cada banc de càrrega. El resultat d'aquesta tècnica és que es permet una connexió de la càrrega de manera progressiva.

Les propietats destacades a la Taula 2 han sigut fonamentals per escollir el commutador MOSFET que activa el frenat, ja que les seves propietats elèctriques estan dins dels rangs de treball de cada banc del frenat.

Propietat	Característica del component	Valor de treball del frenat
Potència màxima (W)	400	300
Tensió màxima de la càrrega (V)	36	24
Freqüència màxima de treball del PWM (kHz)	20	16,67
Tensió de treball del PWM (V)	3,3 a 20	5

Taula 2. Propietats elèctriques del frenat i dels commutadors MOSFET.

### 2.3.3. Estructura de control del frenat elèctric

L'estructura de control s'ha desenvolupat tenint en compte els tres modes de funcionament del frenat elèctric, que es detallen en els següents paràgrafs.

El primer mode, que és l'emergència, és el més important i és el que domina els altres dos modes. El sistema arriba a aquest mode a través de dues vies. La primera és que en el sistema s'ha donat una situació que es necessita una parada urgent, que pot causar efectes irreversibles en la integritat del molí i/o a la instal·lació elèctrica, com el cas d'una potència generada superior a la nominal de l'alternador, o vibracions extremes. La segona és que en el sistema s'ha detectat un error, com podria ser una fallada de les comunicacions, o una lectura d'una valor incoherent d'un sensor. Sigui per estat d'error o d'emergència, cal aturar el sistema per raons de seguretat, en conseqüència, el frenat ha de tenir una demanda estimada de 1.800 Watts per parar l'alternador, per tant, si s'entra en aquest mode, s'han de connectar tots els bancs, i el PWM dels MOSFET ha d'estar amb un cicle de treball al 100%.

El segon mode és el que realitzaria el frenat de fàbrica. És a dir, que la bateria està carregada, i en cas que les potència demandada sigui inferior a la generada, aquest excedent es perd en el frenat, mantenint sempre el nivell de càrrega. D'aquesta manera s'evita que la bateria es sobrecarregui i quedi danyada, o que la velocitat de rotació de la turbina excedeixi els límits mecànics, mantenint-se constant.

El tercer mode és una mesura de seguretat addicional a la bateria que té com a objectiu allargar la seva vida útil. Com ja s'ha esmentat, el sistema d'emmagatzematge d'energia està format per dues bateries de 12 V. El regulador de càrrega pot transferir una potència màxima de 2.200 W, però com que el generador té una potència de 1.800 W, la potència de càrrega queda limitada a aquest valor. Aquest fet es tradueix, que en un màxim de generació la bateria

pot arribar a tenir un corrent màxim d'entrada de 75 A. Segons el fabricant, per un ús en stand-by, és a dir que la bateria s'utilitza de forma puntual, el corrent de càrrega en la primera etapa del cicle pot arribar a 100 A. En aquest cas, el corrent s'ha limitat a 40 A, fixant una potència màxima de càrrega de 960 W. Aquest límit pot ser canviat a través del PLC amb la comunicació Modbus. Però s'ha inicialitzat a aquest valor perquè és inferior al límit donat pel fabricant i és un valor recomanat per diferents companyies. D'aquesta manera quan es supera el límit de funcionament de la bateria, l'excedent de càrrega es transfereix en el frenat.

Tal com s'ha descrit anteriorment, el primer mode és d'emergència o error, i predomina sobre els altres dos modes, que es donen en condicions normals de funcionament, d'aquesta manera, si el frenat està funcionant en el segon o tercer mode, i de cop es diagnostica un error o s'arriba a l'estat d'emergència, automàticament, el frenat passa a treballar en el primer mode. Per exemple, si la bateria està descarregada, el frenat funciona en el mode tres, per protegir la bateria, però és independent dels estats del llaç de control de consum de potència que es puguin donar en aquesta casuística.

#### **2.3.4. Sistema d'acumulació de potència del frenat**

El dispositiu de control, segons el mode de treball del frenat que funcioni, afegeix o treu potència del frenat partint de criteris diferents.

Exceptuant el mode de funcionament d'emergència, que consisteix en connectar de forma instantània tota la càrrega del frenat al màxim, sempre que es doni un excedent energètic no desitjat, el sistema el transfereix potència en el frenat de forma progressiva per tal d'equilibrar la potència generada amb la demandada. Quan el sistema treballa amb la bateria descarregada, i en el bus de l'inversor hi ha càrregues connectades, la diferència entre la potència generada i la requerida per les càrregues s'emmagatzema a la bateria. Si la bateria està carregada, i es dona un excedent, aquest s'ha de consumir en algun punt de la instal·lació per evitar la turbina acceleri, tenint en compte que no es pot destinar a la bateria perquè es pot sobrecarregar i pot danyar-se. El regulador, de forma automàtica, quan la bateria està carregada, desconnecta la bateria, i transfereix l'excedent energètic al frenat. Quan això succeeix, l'excedent energètic en lloc de transferir-se en el frenat de fàbrica, ho fa en el frenat dissenyat. En aquest cas, a partir de les mesures dels sensors de voltatge i intensitat de la bateria, el frenat es posa en funcionament en el mode dos. Llavors, el frenat de forma continuada resta el valor de la potència generada i la potència demandada. Si la diferència és superior a zero, l'excedent es transfereix en el frenat. Un cop transferida la potència al

frenat, passat un període transitori, la potència generada passa a ser igual a la demandada, que inclou la potència del frenat que s'està dissipant. En aquest cas, s'ha de tenir en compte que donada la situació, com que la diferència entre potències és zero, la potència que es transfereix en el frenat es manté. És a dir, si no es mantingués, i la potència generada superés la demandada, i l'excedent s'enviés al frenat, passat aquest temps transitori, com que l'error entre potències tornaria a ser zero, es desconnectaria el frenat, i la potència generada tornaria a superar la demandada. Per tant, la potència del frenat el que fa és acumular les diferències entre potències. Si la potència generada supera la consumida, la potència del frenat augmenta amb un valor igual a l'excedent. Quan la diferència torna a ser zero, la potència del frenat es manté. Si passat un temps torna a augmentar la diferència entre potències, el frenat torna a augmentar la potència, i així successivament fins que el frenat arriba al seu màxim de 1.800 W. En cas que la potència demandada superés la potència generada, i la diferència fos negativa, es reduiria la potència que es transfereix en el frenat, per tal de tornar a igualar les potències. Així doncs, el que s'està duent a terme és un acumulador d'error entre potències. Si la diferència entre la generació i la demanda és positiva augmenta la transferència de potència del frenat, i si es negativa, es treu potència del frenat. D'aquesta manera es segueix l'Equació 4.

$$P_{frenat_i} = P_{frenat_{i-1}} + (P_{generada_i} - P_{demandada_i}) \quad (\text{Eq.4})$$

Quan la bateria està carregada el frenat funciona en el mode tres. En aquest mode el funcionament de la transferència d'energia del frenat funciona exactament igual que el mode dos, però en lloc de d'afegir la diferència entre la potència generada i l'acumulada, en el frenat es transfereix la diferència entre la potència de la bateria i el límit prefixat, tal com marca l'Equació 5.

$$P_{frenat_i} = P_{frenat_{i-1}} + (P_{bateria_i} - P_{límit}) \quad (\text{Eq.5})$$

Quan la diferència entre la potència de la bateria i el límit és negativa, també es resta potència en el frenat, i l'energia es destina a la bateria de nou. Com que és un límit no perillós, i el corrent de la bateria funciona dins del límit, tampoc no interessa que es perdi energia, la protecció de la bateria i l'allargament de la vida útil és molt important, però també és fonamental aprofitar al màxim l'energia generada, per tant, com que el corrent no danya la bateria es prioritza que l'energia es destini a carregar-la. Si de cop el corrent de càrrega sobrepassés el del límit, es tornaria a transferir l'energia excedentària al frenat, augmentant la seva potència.

La potència del frenat es divideix en sis intervals de 300 W. Quan la potència està entre zero i tres-cents, en augmentar la potència del frenat, en el commutador MOSFET se li incrementa el cicle de treball del senyal PWM fins a arribar a una potència de 300 W. Quan arriba aquest valor, el PWM que controla el commutador es manté a nivell alt, i es commuta el PWM del següent banc, ajustant la potència del frenat, fins que el sistema assoleix els 600 W. Conseqüentment, de forma successiva es connecten els bancs de càrrega fins a assolir el màxim de 1.800 W. La manera com es connecten i desconnecten les càrregues segueix la gràfica de la Taula 3.

Potència del frenat (W)	Estat del commutador MOSFET de cada banc de frenat					
	Banc 1	Banc 2	Banc 3	Banc 4	Banc 5	Banc 6
0	Nivell baix	Nivell baix	Nivell baix	Nivell baix	Nivell baix	Nivell baix
(0,300)	PWM	Nivell baix	Nivell baix	Nivell baix	Nivell baix	Nivell baix
[300-600)	Nivell alt	PWM	Nivell baix	Nivell baix	Nivell baix	Nivell baix
[600-900)	Nivell alt	Nivell alt	PWM	Nivell baix	Nivell baix	Nivell baix
[900-1.200)	Nivell alt	Nivell alt	Nivell alt	PWM	Nivell baix	Nivell baix
[1.200-1.500)	Nivell alt	Nivell alt	Nivell alt	Nivell alt	PWM	Nivell baix
[1.500-1.800)	Nivell alt	Nivell alt	Nivell alt	Nivell alt	Nivell alt	PWM
1.800	Nivell alt	Nivell alt	Nivell alt	Nivell alt	Nivell alt	Nivell alt

Taula 3. Estat dels MOSFET dels bancs de resistències de frenat en funció de la potència.

Quan la potència del frenat es troba entre el límit inferior i superior d'un banc, el cicle de treball del senyal de PWM que commuta el commutador MOSFET d'aquest banc de frenada varia seguint l'Equació 6.

$$DutyPWM(\%) = \frac{\sqrt{R_{banc} \cdot P_{banc}}}{V_{bus}} \cdot 100 \quad (\text{Eq.6})$$

De forma experimental, tal com es mostra en l'annex corresponent, s'ha obtingut que la resistència del banc de frenada de cada làmpada és de 1,94 Ohms. Com que és un valor que ha canviat en cada banc, s'ha deixat com a paràmetre de càlcul el valor teòric que és de 1,92 Ohms. A més, la potència de cada banc s'obté amb la diferència de la potència del frenat amb el límit inferior de potència del banc, i es té en compte que la tensió de bus és de 24 V. Per

tant, en funció de la potència que s'ha de dissipar de forma progressiva en cada banc, es calcula el cicle de treball del PWM.

A la vegada que es connecta càrrega, hi ha un sensor de corrent connectat aigües amunt dels sis bancs de càrrega, que mesura la potència real que s'està transferint, fent una realimentació de si el frenat està funcionant de forma correcta. Si la diferència entre el valor de referència de la potència del frenat, i la potència mesurada del frenat és molt gran, es tradueix que hi ha hagut un error o una avaria en el frenat, sigui perquè s'ha fos una làmpada, s'ha desconnectat un cable, o s'ha fos algun commutador MOSFET. En cas de detectar-se algun error d'aquest tipus, el sistema ha de parar la generació immediatament ja que pot ser una situació crítica per la integritat dels sistema. Ja que si el frenat treballa en mode u, l'aturada sobtada no es pot realitzar a través del frenat, si treballa en el mode dos, l'alternador es pot accelerar, i si treballa en el mode tres no es pot garantir la protecció de la bateria. Per tant, l'aturada completa s'ha de fer a través de connectar tot el frenat disponible, i amb el dispositiu de control de moviment de l'angle d'orientació de les pales, que les ha de disposar en el seu valor màxim.

#### **2.4. Carregues del sistema**

Les càrregues del sistema queden alimentades a través d'un inversor de potència. Sobre l'inversor es tindrà un control total, d'aquesta manera el sistema podrà connectar les càrregues seguint les diferents situacions de generació que es poden donar de manera automàtica. L'ondulador que casa amb les propietats seleccionades és el model 179-3338 del fabricant RS-PRO. Aquest convertidor de potència, segons les dades del fabricant, rep una tensió d'entrada de 24 Volts de corrent continu i té una sortida de 230 Volts en alterna i de configuració monofàsica, amb una freqüència de 50 hertz. La seva potència de sortida és de 2.000 Watts, d'aquesta manera, l'energia per alimentar les càrregues monofàsiques queda coberta. Altrament, aquest inversor incorpora diferents proteccions, sobretot per sobreescalfaments, sobretensions, sobrecàrregues, i mecanismes per regular els sots de tensió. En cas de tensions molt baixes, l'inversor s'apaga sol. Després d'una apagada, sigui per un sot de tensió, una baixada de potència o per dispars de proteccions, es restableix el funcionament de forma automàtica, i sempre que s'engega la seva arrencada és suau. Aquest element incorpora un filtratge de la tensió, evitant que aigües amunt passin harmònics de corrent i de tensió, que poden arribar a ser nefastos pels diferents elements de la instal·lació.

Per tal d'assegurar que l'inversor pot tenir un temps de vida útil llarg, s'ha comparat el corrent d'entrada màxim que és de 166,7 Ampers, implicant que la potència és de 4.000 Watts, valor



del qual no s'hi arribarà mai, per tant, en aquest sentit, els fusibles incorporats que protegeixen l'ondulador seria molt estrany que es fonguessin. A més a més, el corrent de sortida màxim és de 17,39 Ampers, per tant, a no sé que es doni un curtcircuit no es sobrepassarà mai. Inclòs pot suportar el corrent transitori de la bomba a l'engegada que és de 10,5 A. En la Figura 12 es mostra l'inversor.



Figura 12. Visió de la carcassa de l'inversor.

La il·luminació de la granja es duu a terme a través de sis focus, que són llums de descàrrega Philips de 100 Watts. Aquests funcionen per a tensions monofàsiques de 230 Volts, i a 50 Hz. En la Figura 13, hi ha una imatge de com és la bomba.



Figura 13. Disposició de la bomba que omple d'aigua els dipòsits.

La bomba és el model Acuaría 07 4N del fabricant ESPA. Per funcionar ha d'estar connectada a una instal·lació monofàsica de 230 V, amb una freqüència de 50 Hz, i necessita una potència

elèctrica de 1,10 CV, que són 800 W. Aquesta bomba té la capacitat hidràulica suficient per omplir els dos dipòsits, que queden interconnectats a través de la tècnica de vasos comunicants. En altres paraules, gràcies al sistema de bombeig, l'energia elèctrica produïda per la turbina queda emmagatzemada en els dipòsits en forma d'energia potencial hidràulica.

Un últim dispositiu que consumeix potència, i va a part dels receptors mencionats anteriorment, perquè és intrínsec del sistema de control, és la font d'alimentació que alimenta el conjunt de sensors i els tres microcontroladors que controlen les diferents tasques del molí. La font és el model LRS-100-5 del fabricant Mean Well. Es tracta d'una font commutada de tipus convertidor de DC-DC buck que va de cinc a 36 V d'entrada a cinc volts de sortida, i pot entregar una potència de fins a 80 Watts. La potència que aporta és suficient per alimentar els tres microcontroladors i tot el conjunt de sensors.

## **2.5. Sistema de control de potència**

El sistema de control de potència és la rutina principal del microcontrolador. Aquesta queda completada pel frenat d'emergència, i les aturades d'error i d'emergència. Sense aquests sistemes, la turbina no pot funcionar amb seguretat. El microcontrolador en funció de la potència generada i l'estat de la bateria transfereix energia. L'explicació de les consideracions que es tenen per transferir energia es detalla en el present apartat, i es divideix en quatre subapartats: en el primer s'expliquen les diferents variables de control, en el segon subapartat s'esmenten els diferents estats i les accions associades a cada estat, en el tercer s'explica l'organigrama lògic, que degut a la dificultat que presenta, es mostra amb taules de transicions, i en l'últim subapartat, es comenten les diferents estratègies i justificacions amb les que s'ha dissenyat l'algoritme de control.

### **2.5.1. Variables de control i condicions**

Els canvis d'estat del sistema depenen de les condicions determinades pels valors de les variables de control.

El sistema transfereix la càrrega en funció del valor de les següents variables de control: la primera, que és la que determina l'estat de la generació i és la que monitoritza la potència generada; també es mesura la potència que pot entregar o absorbir la bateria, a partir del nivell de voltatge i el corrent entregat. Aquestes dues variables, de manera independent, també es tenen en compte com a variables de control que determinen l'estat del sistema. Tant

el valor de voltatge, com la intensitat permeten conèixer l'estat de càrrega de la bateria, ja que amb el nivell de voltatge es pot conèixer l'estat de càrrega de la bateria, gràcies a la seva correlació; i per saber si la bateria està carregada d'una forma ràpida, es monitoritza el sentint del corrent de càrrega de la bateria. De la manera com està connectat el sensor, quan la bateria s'està carregant, el valor de corrent és positiu, i quan la bateria aporta energia, el valor de corrent és negatiu. Finalment, l'última variable de control del sistema és la que determina si es pot accionar la bomba a partir del nivell dels dipòsits.

Per posar el sistema en funcionament, la potència generada ha de superar un mínim per vèncer les pèrdues. A causa de la documentació dels fabricants del regulador de càrrega i de l'aerogenerador, s'ha estimat que la potència mínima de generació és de 240 W. Per tant, quan es detecta una generació superior a aquest valor es passa d'un estat de no generació.

Un cop el sistema està produint energia, el molí pot treballar amb dos escenaris: amb la bateria carregada o amb la bateria descarregada. En cada estat, el que primer s'analitza és l'estat de la bateria a partir del sentit del corrent, i en el cas de la bateria carregada, també s'analitza el nivell de tensió. Si el sistema està treballant amb la bateria carregada, si de cop el corrent de la bateria passa a ser positiu, vol dir que està rebent energia, en lloc d'aportar-la al sistema, en conseqüència, la bateria no està carregada. En el sistema, també se li ha limitat la profunditat de descàrrega de la bateria per allargar-ne la vida útil, en conseqüència, quan la tensió de la bateria baixa per sota d'un llindar, el sistema passa de tenir la bateria carregada, a tenir-la descarregada. La tensió llindar que s'ha fixat és de 23 V, que seria amb un 30 % de la càrrega acumulada, però aquest valor és estimat, i és susceptible a ser modificat, sigui perquè és incorrecte, o perquè es vol variar la profunditat de descàrrega. Si la bateria està descarregada i el valor de corrent passa a ser negatiu o zero, vol dir que la bateria està carregada, i es canvia d'escenari.

El següent anàlisi que es duu a terme en cada estat, és l'estat dels dipòsits. La monitorització dels dipòsits funciona amb un comparador amb histèresis que posa a nivell alt o a nivell baix una variable booleana, a partir de les dades aportades per dos sensors capacitius a diferents nivells. Si el nivell d'aigua dels dipòsits baixa per sota del sensor capacitiu de nivell baix, aquesta variable es posa a un nivell lògic alt, informant al sistema que la bomba es pot activar. Aquesta variable mantindrà l'estat fins que el sensor capacitiu de nivell superior no es posa a nivell alt. Arribat aquest punt, la variable booleana es posa a nivell baix, i no es torna a canviar l'estat fins que el sensor capacitatiu del nivell inferior es posa a zero.

Aquest algoritme que s'ha explicat i es mostra la Figura 14, es pot monitoritzar de manera simultània a altres tasques.

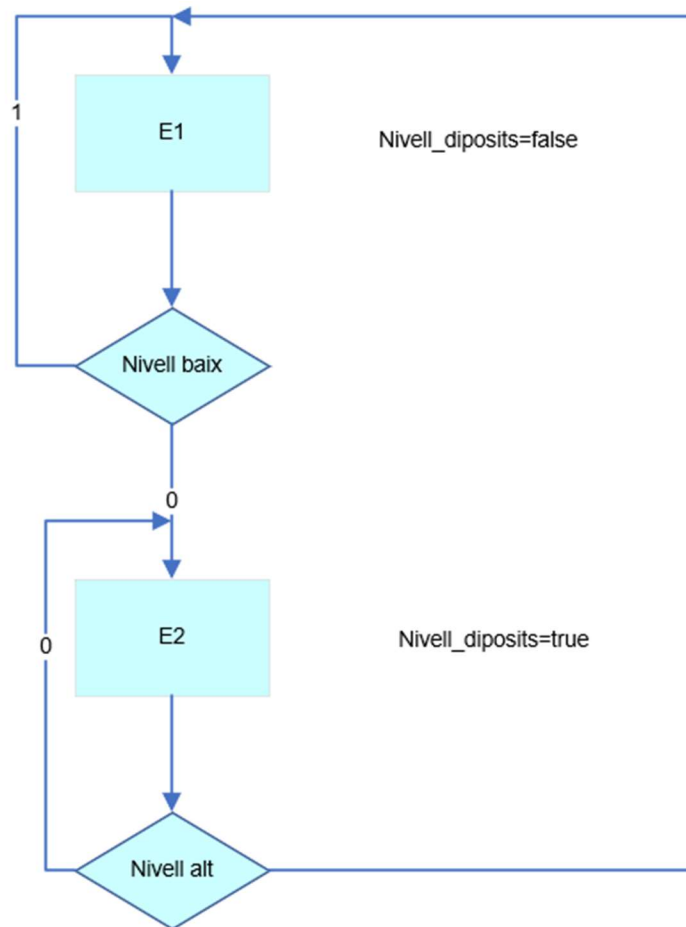


Figura 14. Diagrama de flux de la variable per accionar la bomba.

La següent condició que determina el pas d'un estat del sistema a un altre, és el nivell de potència generada del sistema. En aquest cas, es mira si la potència generada és superior a la suma de la potència de la bomba i els focus, si la potència generada és superior a la potència de la bomba, o si és superior a la potencia dels focus. En aquest últim cas, per poder connectar els focus s'afegeixen més condicions.

En l'escenari de la bateria descarregada, per poder connectar els focus la tensió de la bateria ha de superar un nivell de tensió, que més endavant s'explica el motiu. En la casuística de la bateria carregada, si la potència generada supera la potència dels focus ja es permet la connexió. Quan la bateria està carregada, de forma addicional no només s'avalua la generació sinó que també es té en compte la potència emmagatzemada. En el cas de la connexió bomba, la restricció perquè la bateria complementi la generació és pràcticament inexistent, ja

que es prioritza la transferència de consum en aquest receptor. Pel que fa els focus, només es permet que la bateria aporti energia pels focus, si la bomba no es pot fer servir perquè els dipòsits estan plens, i la potència generada supera un llindar equivalent a la meitat de la potència que consumeixen els focus.

Per tal de relacionar el nom de les variables que intervenen en el procés, com que en el programa i en l'organigrama els noms poden diferir, s'ha plantejat la Taula 4.

Nom en el programa de MPLabX	Nom en l'organigrama	Descripció
PotenciaGenerada	Sgen	Potència generada.
Smin	Smin	Potència mínima de generació.
Nivell_diposits	Nivell_diposits	Nivell dels dipòsits per commutar la bomba.
VBateria	Vbat	Tensió de la bateria.
Vbateria_carregada	Vmàx	Tensió màxima de la bateria, indica si està carregada.
Sllums	Sllums	Potència que consumeixen els llums.
SB	SB	Potència que consumeix la bateria.
Intensitat_bateria	Ibat	Intensitat de la bateria.
Vllums	Vllums	Tensió mínima de la bateria per connectar els llums.
Vbaix	Vbaix	Tensió mínima de la bateria, indica si està descarregada.
Sllindar	Sllindar	Potència llindar generada per connectar els llums.
Pbateria_W	Pbat	Potència que entrega la bateria.

Taula 4. Nom de les variables del sistema.

### 2.5.2. Estats i accions

El sistema de control es divideix en setze estats. En cada estat varien les accions de connexió de càrrega del sistema, i les condicions d'on s'obté l'energia per alimentar aquestes càrregues. És a dir, si l'energia prové tota de la generació, o una part de la generació i una part de la bateria, o tota de la bateria. A la Taula 5, es comenta: cada estat, les seves accions i d'on prové l'energia.

<b>Estat</b>	<b>Bomba</b>	<b>Focus</b>	<b>Frenat</b>	<b>Bateria</b>	<b>Generació</b>
INICI	OFF	OFF	OFF	Descarregada	No
ESTAT 1	OFF	OFF	OFF	Descarregada	Sgen>Smin
ESTAT 2	ON	ON	Pot actuar	Descarregada	Sgen>Sb+Sllums
ESTAT 3	OFF	ON	Actua	Descarregada	Sgen>Sb+Sllums
ESTAT 4	ON	OFF	Pot actuar	Descarregada	Sgen>Sb
ESTAT 5	OFF	OFF	Pot actuar	Descarregada	Sgen>Sb
ESTAT 6	OFF	ON	Pot actuar	Descarregada Vbat>Vllums	Sgen>Sllums
ESTAT 7	ON	ON	Pot actuar	Carregada	Sgen>Sb+Sllums
ESTAT 8	ON	OFF	Pot actuar	Carregada	Sgen>Sb
ESTAT 9	OFF	ON	Actua	Carregada	Sgen>Sb+Sllums
ESTAT 10	OFF	OFF	Actua	Carregada	Sgen>Smin
ESTAT 11	ON	ON	OFF	Carregada i aportant energia	Sgen>Sb
ESTAT 12	ON	OFF	OFF	Carregada i aportant energia	Sgen+Pbat>Sb
ESTAT 13	OFF	ON	Pot actuar	Carregada i pot aportar energia	Sgen>Sllindar
ESTAT 14	OFF	OFF	OFF	Carregada	No
ESTAT 15	ON	OFF	OFF	Carregada	No

Taula 5. Estats i accions del sistema.

### 2.5.3. Organigrama lògic

Degut a la complexitat de l'organigrama lògic, a la present memòria aquest organigrama queda reflectit en una taula de transicions per tal de fer una lectura més aclaridora dels canvis d'estat del sistema de potència del molí.

En les taules de transicions escrites a continuació, amb el caràcter de l'exclamació '!', es fa la negació lògica.

En la Taula 6, es determinen totes les transicions que es duen a terme partint de l'estat INICI, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
INICI	Sgen>Smin	Estat 1
	!(Sgen>Smin)	INICI

Taula 6. Taula de transicions de l'estat INICI.

En la Taula 7, es determinen totes les transicions que es duen a terme partint de l'Estat 1, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 1	$V_{bat} > V_{m\grave{a}x}$	Estat 10
	$\neg(V_{bat} > V_{m\grave{a}x}) \& (S_{gen} > SB + S_{llums}) \& \text{Nivell\_diposits}$	Estat 2
	$\neg(V_{bat} > V_{m\grave{a}x}) \& (S_{gen} > SB + S_{llums}) \& \neg \text{Nivell\_diposits}$	Estat 3
	$\neg(V_{bat} > V_{m\grave{a}x}) \& \neg(S_{gen} > SB + S_{llums}) \& (S_{gen} > SB) \& \text{Nivell\_diposits}$	Estat 4
	$\neg(V_{bat} > V_{m\grave{a}x}) \& \neg(S_{gen} > SB + S_{llums}) \& (S_{gen} > SB) \& \neg \text{Nivell\_diposits}$	Estat 5
	$\neg(V_{bat} > V_{m\grave{a}x}) \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB)$	Estat 5

Taula 7. Taula de transicions de l'Estat 1.

En la Taula 8, es determinen totes les transicions que es duen a terme partint de l'Estat 2, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 2	$\neg(I_{bat} > 0)$	Estat 7
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& (S_{gen} > SB + S_{llums})$	Estat 2
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& (S_{gen} > SB)$	Estat 4
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& ((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums}))$	Estat 6
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& (S_{gen} > S_{min})$	Estat 1
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& \neg(S_{gen} > S_{min}) \& (V_{bat} > V_{baix})$	Estat 14
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& \neg(S_{gen} > S_{min}) \& \neg(V_{bat} > V_{baix})$	INICI
	$(I_{bat} > 0) \& \neg \text{Nivell\_diposits} \& (S_{gen} > SB + S_{llums})$	Estat 3
	$(I_{bat} > 0) \& \neg \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& (S_{gen} > SB)$	Estat 5
	$(I_{bat} > 0) \& \neg \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& ((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums}))$	Estat 6
	$(I_{bat} > 0) \& \neg \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& (S_{gen} > S_{min})$	Estat 1
	$(I_{bat} > 0) \& \neg \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& \neg(S_{gen} > S_{min}) \& (V_{bat} > V_{baix})$	Estat 14
	$(I_{bat} > 0) \& \neg \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& \neg(S_{gen} > S_{min}) \& \neg(V_{bat} > V_{baix})$	INICI

Taula 8. Taula de transicions de l'Estat 2.

En la Taula 9, es determinen totes les transicions que es duen a terme partint de l'Estat 3, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 3	$\neg(I_{bat} > 0)$	Estat 9
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& (S_{gen} > SB + S_{llums})$	Estat 2
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& (S_{gen} > SB)$	Estat 4
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& ((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums}))$	Estat 6
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& (S_{gen} > S_{min})$	Estat 1
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& \neg(S_{gen} > S_{min}) \& (V_{bat} > V_{baix})$	Estat 14
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& \neg(S_{gen} > S_{min}) \& \neg(V_{bat} > V_{baix})$	INICI
	$(I_{bat} > 0) \& \neg \text{Nivell\_diposits} \& (S_{gen} > SB + S_{llums})$	Estat 3
	$(I_{bat} > 0) \& \neg \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& (S_{gen} > SB)$	Estat 5
	$(I_{bat} > 0) \& \neg \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& ((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums}))$	Estat 6
	$(I_{bat} > 0) \& \neg \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& (S_{gen} > S_{min})$	Estat 1
	$(I_{bat} > 0) \& \neg \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& \neg(S_{gen} > S_{min}) \& (V_{bat} > V_{baix})$	Estat 14
	$(I_{bat} > 0) \& \neg \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& \neg(S_{gen} > S_{min}) \& \neg(V_{bat} > V_{baix})$	INICI

Taula 9. Taula de transicions de l'Estat 3.

En la Taula 10, es determinen totes les transicions que es duen a terme partint de l'Estat 4, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 4	$\neg(I_{bat} > 0)$	Estat 8
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& (S_{gen} > SB + S_{llums})$	Estat 2
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& (S_{gen} > SB)$	Estat 4
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& ((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums}))$	Estat 6
	$(I_{bat} > 0) \& \text{Nivell\_diposits} \& \neg(S_{gen} > SB + S_{llums}) \& \neg(S_{gen} > SB) \& \neg((S_{gen} > S_{llums}) \& (V_{bat} > V_{llums})) \& (S_{gen} > S_{min})$	Estat 1

Taula 10.1. Taula de transicions de l'Estat 4.



<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 4	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}>S_{llums}) \& (V_{bat}>V_{llums}) \& !(S_{gen}>S_{min}) \& (V_{bat}>V_{baix})$	Estat14
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}>S_{llums}) \& (V_{bat}>V_{llums}) \& !(S_{gen}>S_{min}) \& !(V_{bat}>V_{baix})$	INICI
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& (S_{gen}>SB+S_{llums})$	Estat 3
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>SB)$	Estat 5
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& ((S_{gen}>S_{llums}) \& (V_{bat}>V_{llums}))$	Estat 6
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}>S_{llums}) \& (V_{bat}>V_{llums}) \& (S_{gen}>S_{min})$	Estat 1
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}>S_{llums}) \& (V_{bat}>V_{llums}) \& !(S_{gen}>S_{min}) \& (V_{bat}>V_{baix})$	Estat14
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}>S_{llums}) \& (V_{bat}>V_{llums}) \& !(S_{gen}>S_{min}) \& !(V_{bat}>V_{baix})$	INICI

Taula 10.2. Taula de transicions de l'Estat 4.

En la Taula 11, es determinen totes les transicions que es duen a terme partint de l'Estat 5, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 5	$!(I_{bat}>0)$	Estat 10
	$(I_{bat}>0) \& (\text{Nivell\_diposits} \& (S_{gen}>SB))$	Estat 4
	$(I_{bat}>0) \& !( \text{Nivell\_diposits} \& (S_{gen}>SB)) \& ((S_{gen}>S_{llums}) \& (V_{bat}>V_{llums}))$	Estat 6
	$(I_{bat}>0) \& !( \text{Nivell\_diposits} \& (S_{gen}>SB)) \& !(S_{gen}>S_{llums}) \& (V_{bat}>V_{llums})$	Estat 1

Taula 11. Taula de transicions de l'Estat 5.

En la Taula 12, es determinen totes les transicions que es duen a terme partint de l'Estat 6, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 6	$!(I_{bat}>0)$	Estat 9
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& (S_{gen}>SB+S_{llums})$	Estat 2
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>SB)$	Estat 4
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& ((S_{gen}>S_{llums}) \& (V_{bat}>V_{llums}))$	Estat 6

Taula 12.1. Taula de transicions de l'Estat 6.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 6	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}>S_{llums}) \& (V_{bat}>V_{llums}) \& (S_{gen}>S_{min})$	Estat 1
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}>S_{llums}) \& (V_{bat}>V_{llums}) \& !(S_{gen}>S_{min}) \& (V_{bat}>V_{baix})$	Estat 14
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}>S_{llums}) \& (V_{bat}>V_{llums}) \& !(S_{gen}>S_{min}) \& !(V_{bat}>V_{baix})$	INICI
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& (S_{gen}>SB+S_{llums})$	Estat 3
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>SB)$	Estat 5
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& ((S_{gen}>S_{llums}) \& (V_{bat}>V_{llums}))$	Estat 6
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& ((S_{gen}>S_{llums}) \& (V_{bat}>V_{llums})) \& (S_{gen}>S_{min})$	Estat 1
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& ((S_{gen}>S_{llums}) \& (V_{bat}>V_{llums})) \& !(S_{gen}>S_{min}) \& (V_{bat}>V_{baix})$	Estat 14
	$(I_{bat}>0) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& ((S_{gen}>S_{llums}) \& (V_{bat}>V_{llums})) \& !(S_{gen}>S_{min}) \& !(V_{bat}>V_{baix})$	INICI

Taula 12.2 Taula de transicions de l'Estat 6.

En la Taula 13, es determinen totes les transicions que es duen a terme partint de l'Estat 7, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 7	$I_{bat}>0$	Estat 2
	$!(I_{bat}>0) \& !(V_{bat}>V_{baix})$	Estat 1
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& (S_{gen}>SB+S_{llums})$	Estat 7
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>SB) \& (S_{gen}+P_{bat}>SB+S_{llums})$	Estat 11
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB+S_{llums})$	Estat 8
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& (S_{gen}+P_{bat}>SB)$	Estat 12
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& (S_{gen}>S_{llums})$	Estat 9
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& !(S_{gen}>S_{llums}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& !(S_{gen}>S_{llums}) \& !(S_{gen}>S_{min})$	Estat 14
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& (S_{gen}>SB+S_{llums})$	Estat 9
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>S_{llindar}) \& (S_{gen}+P_{bat}>SB+S_{llums})$	Estat 13

Taula 13.1. Taula de transicions de l'Estat 7.

Estat	Condicció	Estat següent
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>S_{llindar}) \& !(S_{gen}+P_{bat}>SB+S_{llums}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>S_{llindar}) \& !(S_{gen}+P_{bat}>SB+S_{llums}) \& !(S_{gen}>S_{min})$	Estat 14
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>S_{llindar}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>S_{llindar}) \& !(S_{gen}>S_{min})$	Estat 14

Taula 13.2. Taula de transicions de l'Estat 7.

En la Taula 14, es determinen totes les transicions que es duen a terme partint de l'Estat 8, en funció de les condicions de canvi d'estat.

Estat	Condicció	Estat següent
Estat 8	$I_{bat}>0$	Estat 4
	$!(I_{bat}>0) \& !(V_{bat}>V_{baix})$	Estat 1
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& Nivell\_diposits \& (S_{gen}>SB+S_{llums})$	Estat 7
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>SB) \& (S_{gen}+P_{bat}>SB+S_{llums})$	Estat 11
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB+S_{llums})$	Estat 8
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& (S_{gen}+P_{bat}>SB)$	Estat 12
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& (S_{gen}>S_{llums})$	Estat 9
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& !(S_{gen}>S_{llums}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& !(S_{gen}>S_{llums}) \& !(S_{gen}>S_{min})$	Estat 14
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !Nivell\_diposits \& (S_{gen}>SB+S_{llums})$	Estat 9
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>S_{llindar}) \& (S_{gen}+P_{bat}>SB+S_{llums})$	Estat 13
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>S_{llindar}) \& !(S_{gen}+P_{bat}>SB+S_{llums}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>S_{llindar}) \& !(S_{gen}+P_{bat}>SB+S_{llums}) \& !(S_{gen}>S_{min})$	Estat 14
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>S_{llindar}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !Nivell\_diposits \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>S_{llindar}) \& !(S_{gen}>S_{min})$	Estat 14

Taula 14. Taula de transicions de l'Estat 8.

En la Taula 15, es determinen totes les transicions que es duen a terme partint de l'Estat 9, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 9	$I_{bat} > 0$	Estat 6
	$\neg(I_{bat} > 0) \wedge \neg(V_{bat} > V_{baix})$	Estat 1
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge (S_{gen} > SB + S_{llums})$	Estat 7
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge (S_{gen} > SB) \wedge (S_{gen} + P_{bat} > SB + S_{llums})$	Estat 11
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge (S_{gen} > SB) \wedge \neg(S_{gen} + P_{bat} > SB + S_{llums})$	Estat 8
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge \neg(S_{gen} > SB) \wedge (S_{gen} + P_{bat} > SB)$	Estat 12
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge \neg(S_{gen} > SB) \wedge \neg(S_{gen} + P_{bat} > SB) \wedge (S_{gen} > S_{llums})$	Estat 9
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge \neg(S_{gen} > SB) \wedge \neg(S_{gen} + P_{bat} > SB) \wedge \neg(S_{gen} > S_{llums}) \wedge (S_{gen} > S_{min})$	Estat 10
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge \neg(S_{gen} > SB) \wedge \neg(S_{gen} + P_{bat} > SB) \wedge \neg(S_{gen} > S_{llums}) \wedge \neg(S_{gen} > S_{min})$	Estat 14
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits} \wedge (S_{gen} > SB + S_{llums})$	Estat 9
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge (S_{gen} > I_{lindar}) \wedge (S_{gen} + P_{bat} > SB + S_{llums})$	Estat 13
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge (S_{gen} > I_{lindar}) \wedge \neg(S_{gen} + P_{bat} > SB + S_{llums}) \wedge (S_{gen} > S_{min})$	Estat 10
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge (S_{gen} > I_{lindar}) \wedge \neg(S_{gen} + P_{bat} > SB + S_{llums}) \wedge \neg(S_{gen} > S_{min})$	Estat 14
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge \neg(S_{gen} > I_{lindar}) \wedge (S_{gen} > S_{min})$	Estat 10
$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge \neg(S_{gen} > I_{lindar}) \wedge \neg(S_{gen} > S_{min})$	Estat 14	

Taula 15. Taula de transicions de l'Estat 9.

En la Taula 16, es determinen totes les transicions que es duen a terme partint de l'Estat 10, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 10	$I_{bat} > 0$	Estat 5
	$\neg(I_{bat} > 0) \wedge \neg(V_{bat} > V_{baix})$	Estat 1
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge (S_{gen} > SB + S_{llums})$	Estat 7
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge (S_{gen} > SB) \wedge (S_{gen} + P_{bat} > SB + S_{llums})$	Estat 11
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge (S_{gen} > SB) \wedge \neg(S_{gen} + P_{bat} > SB + S_{llums})$	Estat 8

Taula 16.1 Taula de transicions de l'Estat 10.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& (S_{gen}+P_{bat}>SB)$	Estat 12
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& (S_{gen}>S_{llums})$	Estat 9
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& !(S_{gen}>S_{llums}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& !(S_{gen}>S_{llums}) \& !(S_{gen}>S_{min})$	Estat 14
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& (S_{gen}>SB+S_{llums})$	Estat 9
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>I_{lindar}) \& (S_{gen}+P_{bat}>SB+S_{llums})$	Estat 13
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>I_{lindar}) \& !(S_{gen}+P_{bat}>SB+S_{llums}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>I_{lindar}) \& !(S_{gen}+P_{bat}>SB+S_{llums}) \& !(S_{gen}>S_{min})$	Estat 14
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>I_{lindar}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>I_{lindar}) \& !(S_{gen}>S_{min})$	Estat 14

Taula 16.2. Taula de transicions de l'Estat 10.

En la Taula 17, es determinen totes les transicions que es duen a terme partint de l'Estat 11, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 11	$I_{bat}>0$	Estat 2
	$!(I_{bat}>0) \& !(V_{bat}>V_{baix})$	Estat 1
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& (S_{gen}>SB+S_{llums})$	Estat 7
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>SB) \& (S_{gen}+P_{bat}>SB+S_{llums})$	Estat 11
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB+S_{llums})$	Estat 8
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& (S_{gen}+P_{bat}>SB)$	Estat 12
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& (S_{gen}>S_{llums})$	Estat 9
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& !(S_{gen}>S_{llums}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& !(S_{gen}>S_{llums}) \& !(S_{gen}>S_{min})$	Estat 14
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& (S_{gen}>SB+S_{llums})$	Estat 9

Taula 17.1. Taula de transicions de l'Estat 11.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !\text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>I_{lindar}) \& (S_{gen}+P_{bat}>SB+S_{llums})$	Estat 13
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !\text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>I_{lindar}) \& !(S_{gen}+P_{bat}>SB+S_{llums}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !\text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>I_{lindar}) \& !(S_{gen}+P_{bat}>SB+S_{llums}) \& !(S_{gen}>S_{min})$	Estat 14
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !\text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>I_{lindar}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !\text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>I_{lindar}) \& !(S_{gen}>S_{min})$	Estat 14

Taula 17.2. Taula de transicions de l'Estat 11.

En la Taula 18, es determinen totes les transicions que es duen a terme partint de l'Estat 12, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 12	$I_{bat}>0$	Estat 4
	$!(I_{bat}>0) \& !(V_{bat}>V_{baix})$	Estat 1
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& (S_{gen}>SB+S_{llums})$	Estat 7
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>SB) \& (S_{gen}+P_{bat}>SB+S_{llums})$	Estat 11
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB+S_{llums})$	Estat 8
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& (S_{gen}+P_{bat}>SB)$	Estat 12
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& (S_{gen}>S_{llums})$	Estat 9
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& !(S_{gen}>S_{llums}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& \text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>SB) \& !(S_{gen}+P_{bat}>SB) \& !(S_{gen}>S_{llums}) \& !(S_{gen}>S_{min})$	Estat 14
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !\text{Nivell\_diposits} \& (S_{gen}>SB+S_{llums})$	Estat 9
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !\text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>I_{lindar}) \& (S_{gen}+P_{bat}>SB+S_{llums})$	Estat 13
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !\text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>I_{lindar}) \& !(S_{gen}+P_{bat}>SB+S_{llums}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !\text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& (S_{gen}>I_{lindar}) \& !(S_{gen}+P_{bat}>SB+S_{llums}) \& !(S_{gen}>S_{min})$	Estat 14
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !\text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>I_{lindar}) \& (S_{gen}>S_{min})$	Estat 10
	$!(I_{bat}>0) \& (V_{bat}>V_{baix}) \& !\text{Nivell\_diposits} \& !(S_{gen}>SB+S_{llums}) \& !(S_{gen}>I_{lindar}) \& !(S_{gen}>S_{min})$	Estat 14

Taula 18. Taula de transicions de l'Estat 12.

En la Taula 19, es determinen totes les transicions que es duen a terme partint de l'Estat 13, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 13	$I_{bat} > 0$	Estat 6
	$\neg(I_{bat} > 0) \wedge \neg(V_{bat} > V_{baix})$	Estat 1
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge (S_{gen} > SB + S_{llums})$	Estat 7
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge (S_{gen} > SB) \wedge (S_{gen} + P_{bat} > SB + S_{llums})$	Estat 11
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge (S_{gen} > SB) \wedge \neg(S_{gen} + P_{bat} > SB + S_{llums})$	Estat 8
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge \neg(S_{gen} > SB) \wedge (S_{gen} + P_{bat} > SB)$	Estat 12
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge \neg(S_{gen} > SB) \wedge \neg(S_{gen} + P_{bat} > SB) \wedge (S_{gen} > S_{llums})$	Estat 9
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge \neg(S_{gen} > SB) \wedge \neg(S_{gen} + P_{bat} > SB) \wedge \neg(S_{gen} > S_{llums}) \wedge (S_{gen} > S_{min})$	Estat 10
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge \neg(S_{gen} > SB) \wedge \neg(S_{gen} + P_{bat} > SB) \wedge \neg(S_{gen} > S_{llums}) \wedge \neg(S_{gen} > S_{min})$	Estat 14
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits} \wedge (S_{gen} > SB + S_{llums})$	Estat 9
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge (S_{gen} > I_{lindar}) \wedge (S_{gen} + P_{bat} > SB + S_{llums})$	Estat 13
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge (S_{gen} > I_{lindar}) \wedge \neg(S_{gen} + P_{bat} > SB + S_{llums}) \wedge (S_{gen} > S_{min})$	Estat 10
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge (S_{gen} > I_{lindar}) \wedge \neg(S_{gen} + P_{bat} > SB + S_{llums}) \wedge \neg(S_{gen} > S_{min})$	Estat 14
	$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge \neg(S_{gen} > I_{lindar}) \wedge (S_{gen} > S_{min})$	Estat 10
$\neg(I_{bat} > 0) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits} \wedge \neg(S_{gen} > SB + S_{llums}) \wedge \neg(S_{gen} > I_{lindar}) \wedge \neg(S_{gen} > S_{min})$	Estat 14	

Taula 19. Taula de transicions de l'Estat 13.

En la Taula 20, es determinen totes les transicions que es duen a terme partint de l'Estat 14, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condicció</b>	<b>Estat següent</b>
Estat 14	$(S_{gen} > S_{min}) \wedge \neg(V_{bat} > V_{baix})$	Estat 1
	$(S_{gen} > S_{min}) \wedge (V_{bat} > V_{baix})$	Estat 10
	$\neg(S_{gen} > S_{min}) \wedge \neg(V_{bat} > V_{baix})$	INICI
	$\neg(S_{gen} > S_{min}) \wedge (V_{bat} > V_{baix}) \wedge \text{Nivell\_diposits}$	Estat 15
	$\neg(S_{gen} > S_{min}) \wedge (V_{bat} > V_{baix}) \wedge \neg \text{Nivell\_diposits}$	Estat 14

Taula 20. Taula de transicions de l'Estat 14.

En la Taula 21, es determinen totes les transicions que es duen a terme partint de l'Estat 15, en funció de les condicions de canvi d'estat.

<b>Estat</b>	<b>Condició</b>	<b>Estat següent</b>
Estat 15	(Sgen>Smin)&!(Vbat>Vbaix)	Estat 1
	(Sgen>Smin)&(Vbat>Vbaix)	Estat 10
	!(Sgen>Smin)&!(Vbat>Vbaix)	INICI
	!(Sgen>Smin)&(Vbat>Vbaix)&Nivell_diposits	Estat 15
	!(Sgen>Smin)&(Vbat>Vbaix)&!Nivell_diposits	Estat 14

Taula 21. Taula de transicions de l'Estat 15.

#### 2.5.4. Estratègia i justificacions

Abans d'esmentar els successos que s'han considerat que es poden produir en cada estat, cal entrar en detall amb com s'ha decidit gestionar l'energia de la bateria en funció de la seva capacitat, i de com el regulador transfereix l'energia.

És conegut que la bateria té una capacitat de 160 Ah segons el criteri C100, i segons el criteri C20 és de 135 Ah. A partir d'aquestes dades aportades pel fabricant, s'ha aplicat la llei de Peukert, que és el model matemàtic que relaciona la capacitat variable de la bateria en funció del corrent de descàrrega. Aquest model queda descrit l'Equació 7.

$$Capacitat\ ideal = I^k \cdot temps \quad (Eq.7)$$

Un cop aplicat el model matemàtic, s'han trobat els valors de la constant de Peukert i de la capacitat ideal. Aquests valors permeten conèixer qualsevol valor de capacitat, en funció del corrent de descàrrega. La constant de Peukert per les bateries analitzades és de 1,118, i la capacitat ideal és de 169,13 Ampers-hora. Com que les bateries estan connectades en sèrie, la capacitat del conjunt de les dues bateries equival a la d'una, però el voltatge total queda duplicat. Amb el sistema de control del consum, d'entrada s'ha limitat la profunditat de descàrrega al 70% de la seva capacitat. Aquesta profunditat a partir del PLC i al xarxa Modbus pot ser modificable.

Si el molí es troba en un període de generació nul·la, i es necessita bombar aigua cap als dipòsits, i es coneix que la bomba necessita 800 W, si altrament es considera el rendiment de l'inversor, la bateria ha de subministrar 842 W amb un corrent de descàrrega de 35 A. Donada



la situació, si s'aïlla el terme del "temps" a l'Equació 7, aplicant el corrent de 35 Ampers, el temps que suportarà la descàrrega és de 3,176 hores, que amb la profunditat de descàrrega de la bateria limitada al 70 %, el temps passa a ser de 2,22 hores. Durant aquest temps, segons els cabal especificat pel fabricant, la bateria té l'energia suficient per impulsar 8,67 metres cúbics. La demanda d'aigua és variable segons l'època de l'any, però per tenir una idea de quants dies es pot abastir la granja sense que hi hagi generació eòlica, s'ha partit del cabal mitjà diari que consumeix cada vedell d'engreix segons l'ICAEN, que és de 20 litres al dia. Si la granja té una capacitat màxima de 300 vedells, en total al dia es poden arribar a consumir 6.000 litres de mitjana. Amb aquest consum, les bateries garanteixen un subministrament d'aigua a partir d'energia 100% renovable durant un dia i dotze hores, sense que hi hagi hagut generació eòlica. Realment, com ja s'ha comentat aquest nombre és orientatiu i depèn de la climatologia, i de la quantitat de vedells que hi hagi en cada moment a la granja. Val a dir, que en ubicar-se a la regió de l'Alt Empordà, on bufa la Tramuntana i la Marinada, és bastant improbable que hi hagin llargs períodes sense vent, per tant, en un període amb poca freqüència de vent, amb els vents suaus que es puguin donar en aquestes èpoques, la bateria es podrà anar carregant fins a tenir la suficient energia per poder connectar càrregues.

Un cop s'ha exposat el temps que pot funcionar la bomba sense que hi hagi generació eòlica, cal considerar que la bateria incorpora uns focus de 600 Watts que il·luminen la granja. Amb els paràmetres de la bateria escollida i la profunditat de descàrrega aplicada, en cas de no haver generació, els focus només funcionen durant 3,06 hores. Aquest temps és molt petit i no és desitjable, ja que tota l'energia es transfereix als focus, quan el propietari de la instal·lació vol prioritzar que el consum es destini a omplir els dipòsits. El motiu d'haver implementat els focus com a càrrega es deu a l'aprofitament d'energia durant la generació, és a dir, que si el molí està produint electricitat i els dipòsits estan omplerts, o si la bomba està funcionant i encara la generació supera la demanda, per no perdre energia en el frenat, el sistema, a través de la connexió d'un relé, es permet encendre uns focus. Conseqüentment, el relé que permet connectar els focus, només s'activa quan hi ha producció d'energia, quan no n'hi ha, només s'alimenta la bomba. Tot i que, en cas de generació, si l'energia produïda no és la suficient per alimentar les càrregues, i l'energia emmagatzemada a la bateria complementa l'energia generada, es permet connectar els focus, mentre duri aquest règim de generació, i l'estat de càrrega de la bateria estigui per sobre d'un llindar de connexió. Així doncs, fins que la bateria no tingui un nivell de càrrega suficient per poder connectar la bomba en cas que la producció pari durant un marge temporal extens, els focus no es connecten, i a la vegada, tampoc no es connectaran si la potència generada pel molí no és la suficient per cobrir la demanda energètica dels focus. Així mateix, si la potència generada és inferior a un

llindar i el nivell de bateria és l'adequat per poder connectar els focus, es segueix prioritant que es carregui la bateria.

Altrament, en cas de no haver generació, com que la bomba s'acciona per una comparació amb histèresis dels nivells dels dipòsits, aquest element no estarà funcionant de manera ininterrompuda, en conseqüència, com que s'anirà parant i engegant, el temps en que pot funcionar la bomba amb l'energia de la bateria s'allarga.

Una propietat del regulador, que també ha condicionat l'algoritme del control de càrregues, és la configuració de fàbrica que té establerta. De la manera com està programat, el regulador de càrrega només controla el corrent de càrrega de les bateries en funció l'energia del vent. L'energia que es transfereix cap a l'inversor depèn de les càrregues que aquest tingui connectades. L'inversor que connecta les càrregues està interconnectat al regulador amb un bus de corrent continu. Així doncs, quan augmenta la demanda en la interconnexió de l'inversor, part de l'energia que es destina a carregar bateries alimenta l'inversor. Per exemple, si es suposa que la bateria està descarregada i es generen 1.000 W, i a l'inversor hi ha una demanda de 800 W, la bateria només rep 200 W. Si la demanda de l'inversor baixés a 400 W, la bateria passaria rebre una potència de 600 W. Per tant, el regulador en sí, en funció de l'estat de càrrega de la bateria, controla el corrent de càrrega, i a la vegada, controla la derivació, és a dir, si la bateria està carregada, com ja s'ha detallat, transfereix l'energia d'excedent al dump load box que incorpora de fàbrica. Amb el frenat elèctric que s'ha dissenyat per aquest projecte, l'excedent energètic es transferirà a aquesta implementació, i no al frenat que ve de fàbrica. Cal dir que la intensitat de càrrega pot fluctuar en funció de la velocitat del vent, no obstant mai superarà el valor màxim de intensitat de càrrega per no fer malbé les bateries, tot i que, amb els modes del frenat dissenyat es buscarà aplicar un límit addicional inferior al nominal fixat, per allargar-ne la vida útil. Un altre dels problemes, que soluciona aquest treball, és el fet que el regulador no té cap control sobre la connexió de l'alimentació de les càrregues de l'inversor, és a dir, el sistema de control de consum recau sobre l'inversor per si mateix, i si aquest control no és prou robust, pot provocar mal funcionaments en el sistema, com poden ser les desconexions de càrregues degut a manques d'energia. És per aquest motiu, que en funció de les lectures dels sensors, que informen sobre la potència generada i la potència emmagatzemada a la bateria, s'activen uns relés que controlen les càrregues a on es transfereix la potència generada, i com ja s'ha fet esment, en funció d'on prové l'energia i les càrregues que té connectades el sistema, s'han definit setze estats.

A l'estat d'Inici, la bateria està descarregada i el sistema roman a l'espera de la producció energètica, per tant, no es poden connectar càrregues.

En l'Estat 1, hi ha la producció energètica necessària per superar les pèrdues pròpies del sistema, però la generació és bastant pobre i l'energia s'emmagatzema tota a la bateria. Com passa en tots els estats, com que de manera periòdica es llegeixen les dades dels sensors, i l'estat dels dipòsits, en funció de les dades es canvia a un altre estat.

En l'Estat 2, degut al nivell dels dipòsits i la potència que s'està generant, es permet connectar els focus i la bomba. Si el molí genera una potència superior a la suma de la potència consumida per la bomba i els focus, el sistema de control actua sobre el cicle de treball del PWM del frenat elèctric i/o en connecta els bancs de resistències, amb la finalitat d'augmentar la potència consumida, per evitar que el corrent de càrrega de la bateria sigui molt gran, i pugui danyar-la. Si la potència generada és molt gran, i es superen els límits de la instal·lació, s'haurà de parar la producció.

En l'Estat 3, la potència generada també és igual o superior a la suma de potència de la bomba i la dels focus, però no es permet connectar la bomba perquè els dipòsits estan plens. Per tant, el frenat ha d'actuar per evitar que el corrent de càrrega de la bateria sigui molt gran. I si la potència generada és molt alta, o el frenat es troba en el seu màxim, s'haurà de parar la producció per evitar danys.

Quan el sistema es troba en l'Estat 4, la potència generada és superior a la de la bomba, però no la suficient per connectar els focus de manera simultània. En aquest cas, l'excedent es destina a la bateria, i és poc probable que el corrent de càrrega sigui perjudicial per la bateria, ja que si 800 W van a la bomba, a la bateria hi va una potència inferior a 400 W. Però en cas d'algun error a la instal·lació, el frenat pot actuar.

Quan el molí es situa a l'Estat 5, la potència generada pot ser la suficient per connectar la bomba, però els dipòsits estan plens, o la potència generada no permet connectar la bomba. En aquest cas, no s'actua cap càrrega, i tot es destina a carregar la bateria. En cas de corrents perjudicials de càrrega, el frenat s'activa. Si la bateria té un nivell de càrrega superior a un llindar per connectat els focus, el sistema va a l'Estat 6, en cas contrari, el sistema roman inalterat a l'Estat 5.

Quan el sistema treballa en l'Estat 6, la bateria té el nivell de càrrega just per poder connectar els focus, i la potència generada és igual o superior a la consumida per aquestes càrregues. En aquest cas, com que l'excedent de potència és molt petit, és assumible per la bateria, per tant, el frenat pot actuar només en casos anòmals.

En l'Estat 7, la bateria està carregada i la potència generada pot connectar la bomba i els focus. En cas d'excedent, per evitar danys a la bateria, l'energia es perd en el frenat elèctric. Si la potència generada és superior a la màxima, o el frenat està al màxim, s'ha de parar la producció per evitar danys.

En la situació que es presenta en l'Estat 8, la bateria també està carregada, i la potència generada és superior a la de la bomba, però no és la suficient per connectar els focus. Donada la casuística, s'analitza si la potència generada es pot complementar amb la dels focus per alimentar les dues càrregues, com seria l'Estat 11. En l'escenari que la potència generada sigui lleugerament superior a la de la bomba, no s'utilitza energia de la bateria, per tant, el sistema roman a l'Estat 8.

En cas de trobar-se a l'Estat 9, com que la bateria està carregada, si l'energia és la suficient per connectar els focus, es permet el seu funcionament. Si hi ha excedent energètic, l'energia es dissipa en el frenat.

En la casuística de l'Estat 10, la potència generada és inferior a la que permet connectar els llums, i pel fet que la generació és molt escassa, no es permet connectar la bomba perquè els dipòsits estan plens, i com que les bateries estan carregades, l'energia es perd tota en el frenat. Aquest estat és una espera per si augmenta la potència generada a un valor que permeti connectar càrregues.

Quan el sistema funciona en l'Estat 11, la potència generada és superior a la que permet connectar la bomba, i com que la bateria està carregada, mentre la bomba omple els dipòsits, hi ha una petita aportació energètica de la bateria, que permet també connectar els focus. S'ha decidit que aquesta casuística sigui puntual, i que de forma momentània, durant el temps en que funciona la bomba, la bateria aporti l'energia fins a un màxim equivalent al consum total dels focus, és a dir, 600 Watts. Realment, en omplir els dipòsits, la bomba no tarda gaire, en conseqüència, no es gastarà gaire energia provinent de bateria en els focus. Per tant, en aquesta situació la bateria complementa la generació, i el frenat no actua.

En la situació plantejada a l'Estat 12, en la forma com s'aporta energia a la càrrega té un símil amb l'Estat 11. La potència generada és inferior a la de la bomba, i els dipòsits estan amb un nivell baix d'aigua. Però en tenir la bateria carregada, si la potència de la bateria complementa la generada i la suma de les dues permet connectar la bomba, s'habilita.

L'Estat 13 es dona quan la potència generada és superior a un llindar, i la bateria està totalment carregada. El llindar fixat té un valor inferior a la potència per encendre els llums, perquè part de l'energia que es transfereixi de la bateria complementi a la generació per connectar aquest tipus de càrregues, i a la vegada que la potència aportada per encendre els llums, no sigui tant gran com per consumir la bateria de forma tant ràpida com en el cas de no haver aportació de la generació. Aquesta situació es podria donar: quan l'energia generada fos superior a la de la bomba però els dipòsits estan plens, o que simplement la potència generada és superior a aquest llindar. Com que s'ha descartat que en cas de generació zero, es connectin els focus, l'energia només es destina a la bomba. En conseqüència, perquè la generació quedi complementada amb la bateria per encendre els focus, s'ha de garantir que només una petita part de la bateria vagi destinada a alimentar els focus. Aquest estat contempla: tant que l'energia dels focus vingui tota de la generació, com que l'energia dels focus vingui de la generació i la bateria. El valor del llindar s'inicialitza perquè l'energia provinent de la bateria només sigui un 50% del total que consumeixen els focus, és a dir, 300 Watts, però igual que tots els llindars pot ser modificable pel Modbus.

Quan no hi ha generació i la bateria està carregada, si la bomba no es pot connectar perquè els dipòsits estan plens, el sistema es troba a l'Estat 14.

Si no hi ha producció i l'estat de la bateria i dels dipòsits permet connectar la bomba, el sistema passa a treballar a l'Estat 15.

Dels Estats 14 i 15, si no hi ha producció i el voltatge de la bateria decau per sota d'un llindar, el sistema retorna a l'estat inicial.

Quan el sistema es troba en els Estats 14 i 15, el dispositiu de control es troba en el mode de baix consum o sleep. Com es pot observar, les úniques accions que s'executen és activa la bomba segons si la bateria disposa de l'energia suficient, i llegir els sensors, per si es pot posar en un estat de generació.

## **2.6. Maniobra d'emergència**

Per garantir un funcionament segur de l'aerogenerador, s'ha implementat una maniobra d'emergència que permet parar la generació de forma ràpida i segura, amb l'objectiu de causar el mínim dany possible a l'estructura i a la instal·lació elèctrica, i a la vegada que eviti posar en risc a les persones i a l'entorn on està instal·lat l'aerogenerador.

En el present apartat es divideix en cinc subapartats: en el primer es dona una explicació de com el sistema arriba a posar-se en emergència; seguidament, en el segon subapartat, s'esmenten els diferents estats de la parada d'emergència i les seves accions associades, en el tercer es fa esment de les variables de control i condicions de l'aturada d'emergència, seguidament, s'esmenta el procés sencer amb un organigrama, i en l'últim subapartat es comenten les estratègies i justificacions implementades en l'aturada d'emergència.

### **2.6.1. Causes de l'aturada d'emergència**

A partir de les especificacions mecàniques de l'aerogenerador i les propietats elèctriques de l'alternador, s'han definit les circumstàncies amb les quals l'aerogenerador s'ha d'aturar. En el procés per aturar la generació eòlica davant d'una emergència, hi intervenen tots els sistemes de control i de monitorització de la turbina, és a dir, el control de l'angle d'inclinació de la turbina, el sistema de control de potència i el sistema de monitorització de l'estructura mecànica.

A partir de les variables de mencionades a continuació, quan es superen valors que poden ser perillosos per la turbina, el sistema d'adquisició de dades envia el byte corresponent al codi d'emergència de forma simultània, a través de la xarxa de comunicacions, al sistema de control de consum de potència i al sistema de control de l'angle d'inclinació de les pales. Així doncs, la comanda d'emergència s'envia: quan el vent té una velocitat excessiva, quan la tensió sobre l'estructura està per sobre o molt proper al límit mecànic, quan es capten vibracions excessives que poden arribar a la freqüència de ressonància de l'estructura i pot cedir, o simplement es reben vibracions anòmales perquè s'ha trencat una pala o un rodament mecànic. El sistema de control d'angle d'inclinació de les pales evita que la velocitat de rotació superi un límit perillós pel conjunt format per la turbina i el generador. Utilitza un llaç de control que mesura la velocitat de rotació amb un encoder, i mou l'angle d'inclinació de les pales per controlar la velocitat. Si hi ha un error de control, les pales poden accelerar massa i arribar a una velocitat perjudicial. En aquest cas, el sistema gira les pales per frenar la turbina, i envia

per la xarxa de comunicacions una comanda d'emergència al sistema de control de potència. El sistema de control de potència es pot posar en estat d'emergència quan la potència generada és massa alta. Això pot passar si la velocitat és excessiva, i el sistema de control d'angle d'inclinació de les pales no ho detecta. Quan es percep una potència que supera el límit, el sistema de control de potència envia la comanda d'emergència al sistema de control d'angle d'inclinació de les pales, per aturar la generació i evitar danys. Encara que l'excés de potència pot ser detectat simultàniament amb la velocitat excessiva, les redundàncies fan el sistema més segur en casos d'emergència.

### **2.6.2. Estats i accions**

Un cop el sistema entra en estat d'emergència, el procés queda dividit en cinc etapes diferenciades. En la primera etapa, el cicle de treball del PWM del frenat elèctric es posa al màxim, es comunica a l'Esclau 1 que s'ha connectat el frenat al màxim, i s'indica al PLC que el sistema està en emergència. En la segona etapa, es fa una espera de dos segons perquè el sistema de control de l'angle d'inclinació de les pales, posicioni les pales per evitar el gir. Aquesta etapa d'espera es duu a terme tres vegades. Per a cada volta d'aquest bucle, el programa passa per l'etapa quatre, que la seva acció consisteix en augmentar el comptador de les voltes del llaç. Finalment, en les etapes tres i cinc, el sistema actua sobre els relés de curtcircuit del generador perquè el rotor quedi bloquejat fins que no es rearmi. La diferència entre aquestes dues etapes és que en l'etapa tres el sistema ha curtcircuitat el generador sota condicions de seguretat, i en l'etapa cinc l'ha curtcircuitat mentre encara està produint energia, i és possible que s'hagi danyat o el generador o els relés, i calgui una revisió, per aquest motiu es posen a nivell alt uns bits d'avís al PLC.

Finalment, el PLC per fer un test de la maniobra d'emergència pot provocar aquesta aturada. En aquest cas, el PLC es comunica amb els esclaus i provoca que el sistema de control de potència pari el sistema.

### **2.6.3. Variables de control i condicions**

Les variables de control que intervenen en l'emergència queden dividides en dos tipus: les variables per canviar d'etapa, i les variables per informar al PLC sobre l'estat d'emergència. Les variables de control per canviar d'etapa poden ser escrites en el PLC, únicament quan l'aerogenerador funciona amb condicions de treball segures. En cas d'emergència, el PLC no pot modificar aquestes variables. El motiu de distinció d'aquest dos tipus de variables es deu

a tenir el menor grau de dependència del PLC, que en cas d'emergència, aquest passa a ser un element únicament de supervisió. Per tant, el control de l'aturada recau de manera total sobre el esclaus dels sistema microcontrolador. El PLC sí que pot escriure la condició de rearmament, que només pot ser escrita quan l'aturada d'emergència s'ha fet de manera total, i el sistema roman a l'espera del rearmament. Per tant, en les etapes tres i cinc.

Per tal de relacionar el nom de les variables que intervenen en el canvi d'etapa, s'ha fet la Taula 22, ja que en el programa i en l'organigrama els noms poden diferir.

Nom en el programa de MPlabX	Nom en l'organigrama	Descripció
Emergencia	Emergencia	Variable booleana que inicia l'aturada d'emergència.
Espera_emergencia	Espera_emergencia	Variable integer que compta quantes les esperes de rebre el valor d'angle màxim, per part del sistema de control de l'angle d'inclinació de les pales.
Rebut_angle_Maxim	Angle Rebut	Variable booleana que indica que el sistema ha rebut l'angle màxim.
Potencia_limit_reles	Prelés	Variable sense signe integer que indica el límit de potència que suporten els relés per no ser danyats.
PotenciaGenerada	PGen	Potència generada.
Error_WATCHDOG	Error WATCHDOG	Variable booleana que indica si hi ha hagut un error d'execució del programa, on ha quedat encallat, durant l'emergència.
TMR2_ReadTimer()	Temps	Valor del temporitzador d'espera de rebre l'angle al màxim per part del sistema de control de l'angle d'inclinació de les pales.

Taula 22. Nom de les variables de canvi d'etapa.

En la Taula 23, hi ha les variables que són únicament de supervisió de l'emergència que rep el PLC.



Nom en el programa de MPlabX	Descripció
AvisEmergenciaPLC	Variable booleana que avisa al PLC que el sistema està en emergència.
MANIOBRA_FETA	Variable booleana que indica que l'aturada s'ha realitzat.

Taula 23. Nom de les variables de supervisió.

Finalment, hi ha variables que poden ser escrites des del PLC, però només quan la maniobra d'emergència s'ha aturat. I a la vegada serveixen per informar com es troba el sistema després del procediment d'emergència. Aquestes es mostren a la Taula 24.

Nom en el programa de MPlabX	Descripció
REVISIO_SISTEMA	Variable booleana que avisa al PLC que el sistema pot haver quedat danyat de forma severa, i que cal una revisió més exhaustiva.
POSSIBLES_DANYS_RELES_GENERADOR	Variable booleana que indica que els relés han curtcircuitat l'alternador, mentre generava una potència superior a la de treball.
REARMAMENT	Variable booleana que permet rearmar el sistema després de l'aturada d'emergència.

Taula 24. Nom de les variables de supervisió.

També existeix un bit de possibles danys dels relés que es posa a nivell alt quan el generador supera la potència límit de treball dels relés, i es curtcircuita l'alternador. Aquesta potència pot ser escrita a través del PLC, quan el sistema no està en emergència, però inicialment és de 600 W.

En l'organigrama lògic adjuntat en el següent subapartat, hi ha una condició que s'anomena condició de rearmament, i que tal com el seu nom indica permet rearmar el sistema. Aquesta condició de rearmament consisteix en que l'equació lògica mostrada en l'Equació 8.

$$\text{Condició de rearmament} = \text{REARMAMENT} \cdot \text{POSSIBLES\_DANYS\_RELES\_GENERADOR} \cdot \text{REVISIO\_SISTEMA} \quad (\text{Eq.8})$$

Com ja s'ha descrit anteriorment, des del PLC es poden escriure les variables de possibles danys als relés, i de revisió del sistema. Per tant, si aquestes no estan a nivell baix, no es pot rearmar el sistema, per molt que s'hagi posat a nivell alt la variable booleana de rearmament. La finalitat d'aquesta condició, per rearmar el sistema, és obligar a l'encarregat del

manteniment del molí que realitzi la tasca de revisar els possibles danys al sistema. Així doncs, és una mesura per assegurar un bon funcionament i ús de la turbina.

En cas que es donin condicions normals de funcionament, el PLC pot escriure sobre els bit d'emergència, per fer un test d'aquesta aturada.

#### **2.6.4. Organigrama lògic**

Una aturada d'emergència ha de ser ràpida i eficaç, en conseqüència, l'organigrama lògic ha de ser el més senzill possible, però a la vegada ha de garantir que el sistema funcioni d'una manera robusta. En la Figura 15, es mostra l'organigrama lògic.

El sistema entra en l'aturada d'emergència quan la variable Booleana anomenada "Emergencia", es posa a nivell alt. A partir d'aquest moment, de manera instantània s'entra a la primera etapa, i es duen a terme les accions pertinents, que un cop hagin finalitzat, es comprova si hi ha hagut un desbordament del Watchdog, més endavant es comenta el motiu d'introduir aquesta condició al principi del diagrama de flux. El Watchdog és un temporitzador intern del PIC, que al llarg de l'execució del programa s'ha d'anar rearmant, si aquest no s'ha rearmat, significa que el sistema ha quedat encallat en un llaç infinit, i es poden donar situacions crítiques si no s'actua. Així doncs, en cas de no rearmar-se aquest temporitzador, el sistema reinicia el codi, i es tracta aquest error de forma pertinent. En l'aturada d'emergència, si no ha succeït un desbordament del temporitzador del Watchdog, s'entra a un bucle, que com a màxim dura tres iteracions. En aquest bucle, el sistema de control de potència entra en l'etapa dos, on envia un missatge al sistema de control d'angle d'inclinació de les pales que el frenat elèctric està al màxim, i activa un temporitzador que provoca una espera durant dos segons. Mentre el PIC està en aquesta espera, el sistema no està parat, ja que està pendent de rebre el missatge de confirmació que l'angle de les pales està al màxim, i a la vegada queda comunicat amb el PLC que va enviant dades sobre l'estat de l'aturada d'emergència del sistema. Si durant l'etapa dos el sistema de control de potència rep a través del sistema de comunicacions, el datagrama corresponent a l'angle de les pales al màxim, el sistema passa a l'etapa tres. En cas d'exhaurir-se el temporitzador sense haver rebut el byte de l'angle d'inclinació de les pales està al màxim, el sistema passa a l'etapa quatre, on s'incrementa el comptador de voltes d'aquest bucle, amb la variable "Espera\_emergencia". Mentre aquest comptador no arribi a tres, el sistema retorna a l'etapa dos, on es realitza el mateix procediment. Si el comptador adquireix un valor de tres, el sistema comprova que la generació sigui inferior a la que suporten els relés com a màxim. A aquesta condició també

s'hi arriba si s'hagués donat un desbordament del Watchdog. Si la potència de generació és inferior, el sistema passa a l'etapa tres, és a dir, a l'espera del sistema a de ser rearmat sense que calgui una revisió del sistema i dels relés. En la situació contrària, el sistema entra a l'etapa cinc, és a dir, a l'espera de ser rearmat, amb els bits de danys dels relés i de revisió del sistema a nivell alt, perquè l'usuari faci la revisió corresponent.

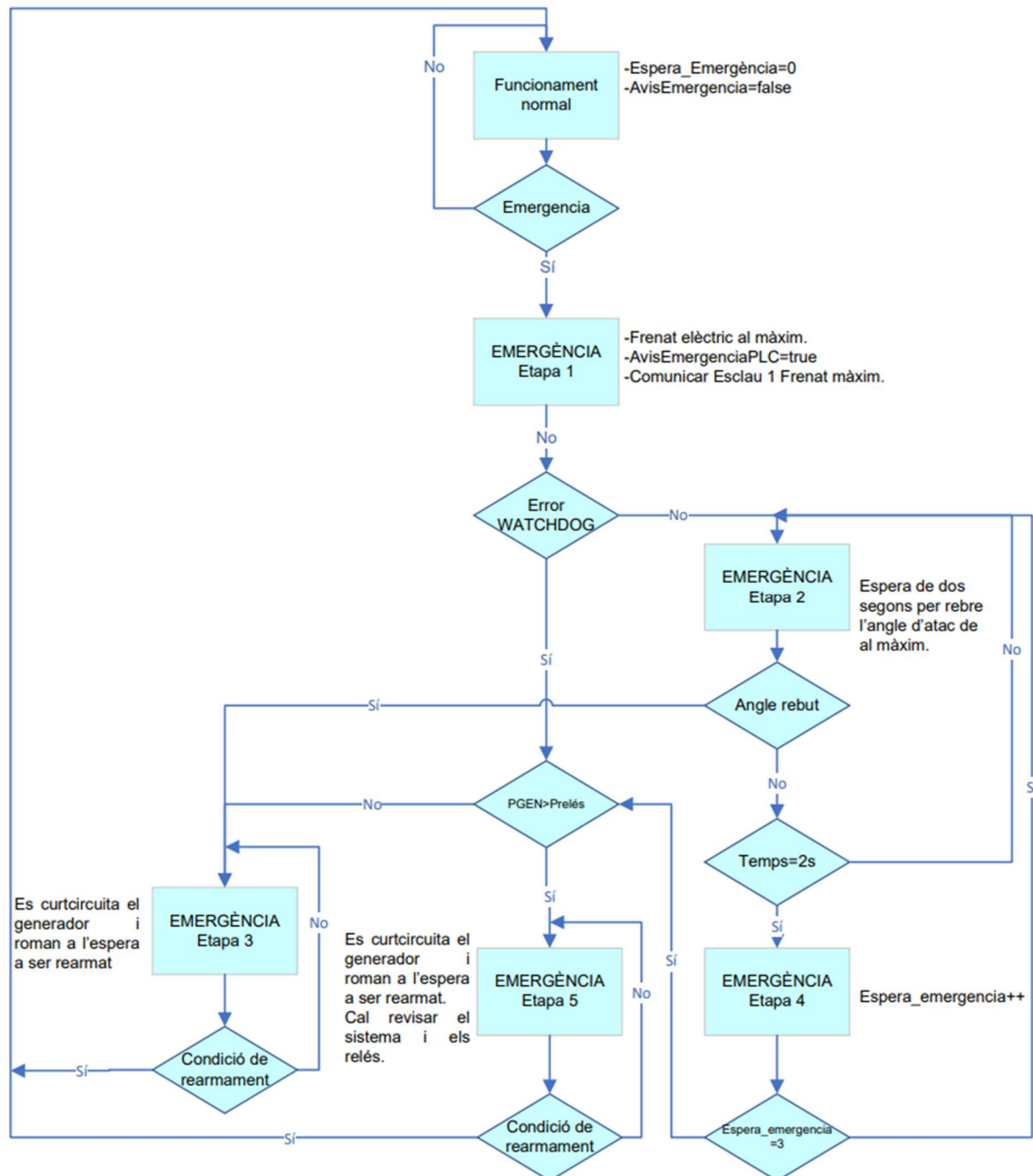


Figura 15. Organigrama lògic de l'aturada d'emergència.

En l'aturada d'emergència, el sistema de control de potència queda comunicat amb els altres dispositius de la turbina, i amb el PLC. En les etapes tres i cinc, el sistema queda en l'espera de ser rearmat. El temps en que està pendent a ser rearmat, depèn del PLC, per tant, és indeterminat.

Quan el sistema està duent a terme la parada d'emergència, es va reiniciant el mecanisme de verificació de l'execució correcta del programa. Si aquest no es rearma, significa que s'ha encallat en un bucle infinit, i en aquest cas, el sistema retorna a la primera etapa. Aquest retorn a l'etapa u no queda plasmat en el diagrama de flux, perquè visualment seria poc entenedor, però s'ha de tenir present que en qualsevol punt, si hi ha un error de desbordament del comptador Watchdog, el sistema retorna a aquesta etapa. És per aquest motiu, que un cop dutes a terme les accions de l'etapa u, es consulta si el comptador del Watchdog s'ha desbordat. Per tant, quan aquesta condició de desbordament de Watchdog és certa, significa que s'ha desbordat aquest comptador, a causa d'un bucle infinit, mentre el sistema ja estava prèviament en aturada d'emergència. Donada la situació, el sistema deixa de comunicar-se amb els altres dispositius, altrament, com que el frenat ja s'ha posat al màxim, el sistema curtcircuita el generador a través dels relés, donant-se la possibilitat que el sistema ja estigui parat, o que la generació sigui pràcticament nul·la, o que la generació superi la potència límit dels relés.

### **2.6.5. Estratègia i justificacions**

L'objectiu principal de la maniobra d'emergència consisteix en parar la generació de la manera més ràpida i segura possible. Per tant, a l'hora de dissenyar aquesta aturada s'han tingut en compte els mètodes de frenada de la turbina, i els errors que es poden donar en aquest procés, encara que aquests siguin poc provables, per tal de solucionar-los, i que el molí freni amb seguretat.

L'aturada de la turbina es pot fer modificant l'angle d'inclinació de les pales, i afegint càrrega a través del frenat elèctric dissenyat. Per tant, en diagnosticar-se una emergència per qualsevol dels dispositius de la turbina, a la vegada que l'angle d'inclinació de les pales es posa al màxim, és a dir, amb paral·lel a la direcció del vent, el frenat elèctric es posa al màxim, augmentant la demanda d'energia, per tal de superar la generació, i que el gir de la turbina freni fins a romandre pràcticament immòbil. A més, aprofitant l'efecte electromagnètic del bloqueig del rotor quan el generador queda curtcircuitat, s'evita que el generador mogui les pales, fins que no es repari la situació d'emergència, i es rearmi el sistema. En conseqüència,

un cop han actuat els dos mecanismes de fre, per evitar que la turbina es mogui, i quedi frenada del tot, es curtcircuiten els pols de l'alternador. D'aquesta manera, quan s'ha produït la frenada d'emergència, les pales de la turbina queden en paral·lel a la direcció del vent, i l'eix de la turbina bloquejat pels efectes electromagnètics del curtcircuit de l'alternador.

Quan es detecta una situació d'emergència, el més habitual seria l'operació sense incidències en les comunicacions i en els diferents mecanismes de frenada durant l'aturada d'emergència. D'aquesta manera a la vegada que actua el frenat elèctric, també ho fa el sistema de control de l'angle d'inclinació. Així doncs, en el bucle que comprèn entre l'etapa dos i l'etapa quatre, es donen dos segons d'espera com a mínim, perquè possiblement el sistema de control de l'angle d'inclinació no hagi tingut un marge suficient de reacció, ja que el moviment de l'angle de les pales, tot i ser molt ràpid no és immediat. Aquest temps és més que suficient perquè es freni la turbina mitjançant els dos mecanismes. Conseqüentment, en una situació normal el frenat es posa al màxim, a la vegada que actua l'angle d'inclinació de les pales. El missatge de confirmació que s'ha mogut al màxim l'angle d'orientació de les pales pot ser redundant, si el sistema ha frenat de manera correcta, però assegura que el curtcircuit del generador a través del relés es pot realitzar amb total seguretat, sense que es danyi cap element elèctric. De la mateixa manera que també pot semblar redundant que el sistema de control de potència comuniqui al sistema de control de l'angle d'inclinació de les pales que el frenat està al màxim, perquè mogui les pales al màxim, i esperi la resposta que les pales estan al màxim. Aquest cas és el mateix que quan el sistema de control de l'angle d'inclinació de les pales detecta l'emergència i li comunica al sistema de control de potència. Aquestes dues maniobres són fonamentals perquè és possible que un dels dos dispositius no hagi percebut l'emergència, implicant que un dels dos mecanismes de frenat no hagi actuat, i que el sistema no sigui prou robust. Així doncs, s'assegura que l'emergència és percebuda pels dos dispositius que actuen sobre la turbina per frenar la generació.

De la manera com s'ha plantejat l'emergència, els errors de l'aturada que es poden donar són de comunicació. En la comunicació entre dispositius de la turbina es poden donar dos tipus de fallades. La primera és que entre els dispositius no s'enviïn de manera correcta les comandes de frenat al màxim, i de l'angle d'inclinació de les pales al màxim. Quan s'ha produït aquest error, i es curtcircuita el sistema sense rebre la confirmació que l'angle està al màxim, cau la probabilitat que no s'hagi rebut la confirmació perquè el sistema de control de l'angle estigui tenint un error de control greu i que no hagi actuat, en conseqüència, quan es dona aquest escenari és probable que només hagi actuat el frenat elèctric, i que la turbina encara tingui certa velocitat de rotació, i en conseqüència encara generi energia. Com que en una

emergència el sistema s'ha de parar de la manera més ràpida possible, el sistema de control de potència curtcircuita el generador a través dels relés, mentre el generador estigui sota condicions de generació. Si es dona aquest fet, en produir-se el curtcircuit, el generador frena de manera sobtada, i para la generació de cop. En aquest cas, el sistema en curtcircuita el generador, no indica en el PLC que calgui cap revisió al sistema de control de potència, ni que s'hagin danyat els relés. Però caldria revisar el sistema de comunicacions de la turbina, el sistema de monitorització de l'estructura i el mecanisme d'orientació de les pales. L'altra situació que es pot donar és que a l'hora de curtcircuita el generador, hi hagi una generació superior a la que els relés poden suportar. En aquesta casuística, els bits de revisió del sistema i dels relés, estaran a nivell alt. I l'usuari ha d'assegurar una revisió del sistema i en concret dels relés, que poden haver-se danyat severament.

Cal destacar un fet important, el temps de dos segons d'espera, i que es facin tres iteracions no és fruit de l'atzar. Aquest temps s'ha escollit seguint criteris mecànics, ja que equival al temps que pot frenar la turbina, només amb l'actuació del frenat elèctric, sense el control de l'angle d'inclinació de les pales, partint del seu règim màxim de revolucions fins a quedar pràcticament parat. Val a dir, que aquests criteris mecànics són teòrics, i que a la posada en funcionament són susceptibles a ser modificats.

També és important destacar, que el curtcircuit amb una generació superior a la que poden suportar els relés seria una situació molt inusual, ja que implicaria una cadena d'errors. Per exemple, que no hagués funcionat el sistema de control de l'angle d'inclinació de les pales, sumat al fet que el frenat elèctric no funcionés a 1.800 W, i sense que el molí hi hagués cap càrrega connectada, o que s'estigués generant una potència molt elevada. És cert que un error d'aquest tipus, gràcies a les mesures de seguretat aplicades, com la pròpia monitorització del PLC, o la maniobra d'error, és poc probable, però cal considerar-lo.

El segon error de comunicació, que es pot donar entre els dispositius de la turbina, és el mateix que es pot donar en la xarxa de comunicació del PLC. I consisteix en el fet de caure en un bucle infinit, és a dir, que no s'executa el codi corresponent. El fet de caure en un llaç infinit, implica que s'ha produït un error de maquinari a la xarxa de comunicacions durant l'aturada d'emergència. L'aparició d'un bucle infinit en el programari, pot desencallar a una situació crítica en l'aturada d'emergència, que pot arribar a afectar a la integritat del molí de vent. La mesura de seguretat, per evitar aquest tipus de casuístiques, és emprar el temporitzador del Watchdog. En el cas de donar-se un desbordament del Watchdog en l'aturada d'emergència, la maniobra s'aplica de manera diferent que en els altres punts del programa, és a dir, el

sistema no entra en l'etapa d'error, sinó que continua estant en emergència, i es deixa de comunicar amb els altres dispositius de la turbina, i amb el PLC. Com que l'emergència ha sigut captada, el sistema s'assegura que el frenat estigui al màxim, i llavors actua sobre els relés de curtcircuit, avaluant també si la potència generada és superior o propera al límit dels relés. En aquest cas, és molt poc probable que en una emergència fallin les comunicacions perquè com s'explica en altres punts de la present memòria, es van fent tests de manera contínua, i en cas de diagnosticar-se un error en aquests tests, es para la generació per poder arreglar aquests errors. Però aquesta casuística s'ha de contemplar. També s'avalua si el curtcircuit del generador s'ha produït amb una potència generada per sobre del límit dels relés, posant-se a nivell alt els bits de revisió.

La maniobra d'emergència predomina sobre qualsevol situació de treball del molí. És a dir, si el sistema treballa amb condicions normals, o si el sistema està realitzant la maniobra d'error, i es percep la situació d'emergència, el sistema de forma immediata passa a dur a terme la maniobra d'emergència.

## **2.7. Maniobra d'error**

La transferència de potència de l'aerogenerador i la seva aturada d'emergència venen condicionades per la lectura dels sensors, i els missatges enviats per les dues xarxes de comunicacions. En conseqüència, si els sensors adquireixen valors incoherents de les dades que mesuren, o si la xarxa de comunicacions fallen, sigui per avaries en el maquinari, desconexions, o que els missatges es perceben de manera errònia, el dispositiu de control de potència ha de poder captar aquestes incidències i parar la producció per tal de fer una revisió dels diferents elements de la turbina, i tractar aquests errors. En cas de no tractar els diferents errors que es puguin donar, el sistema de control de potència pot funcionar de manera errònia podent desencadenar una situació crítica per la integritat de la turbina. Així doncs, per garantir un funcionament segur de l'aerogenerador, s'ha implementat una aturada del sistema quan es detecta un error i que no es restableixi el funcionament fins que no es revisi aquest error.

En el present apartat es divideix en cinc subapartats: en el primer es dona una explicació dels diferents errors que pot llegir el sistema; seguidament, en el segon subapartat, s'esmenten els diferents estats de la maniobra d'aturada d'error i les seves accions associades, en el tercer es fa esment de les variables de control i condicions de canvi d'estat d'aquesta aturada,

seguidament, s'esmenta el procés sencer amb un organigrama, i en l'últim subapartat es comenten les estratègies i justificacions implementades en aquesta maniobra.

### **2.7.1. Errors del sistema**

L'avaluació dels errors que es poden donar, es duu a terme mentre el sistema de control de potència està funcionant en condicions normals, és a dir, que el sistema no està en l'aturada d'emergència, ni amb una aturada perquè prèviament ja s'ha diagnosticat un error.

L'avaluació del sistema es divideix en dues parts: la primera és un anàlisi de les lectures de potències i el segona un test de la xarxa de comunicacions.

Les dues parts estan controlades, de manera que cíclicament mentre el sistema està condicions normals de funcionament es va avaluant si hi ha errors.

El funcionament de l'anàlisi d'errors de les potències mesurades consisteix en comparar el valor de potència llegit per part del sensor corresponent, amb el valor de potència teòric que hauria de tenir, que depèn de l'estat on es trobi l'algoritme de control de consum de potència, inclou un marge d'error permès. Per tant, es resta el valor de potència adquirit a partir de les lectures dels sensors amb el valor teòric, i seguidament es compara el resultat d'aquesta resta amb valor absolut amb el marge d'error. Si el resultat de la resta supera el marge d'error, el sistema es posa en estat d'error, ja que significa que un sensor està donant valors incoherents. Existeix un temporitzador per tal d'evitar que durant l'anàlisi degut a un període transitori, una de les magnituds respecte el seu valor teòric, pugui superar el marge d'error. Per tant, de la manera com s'ha plantejat els valors que es comparen ja són en estat estable, essent molt improbable que el sistema es posi en error a causa del període transitori. Dins d'aquesta avaluació, el sistema compara amb el marge d'error, que s'ha fixat a 12 W, la potència del frenat elèctric, la potència de la bateria, que es considera tant si entrega com si rep energia, i la potència de l'inversor. Una altra avaluació que es produeix en l'anàlisi d'error dels sensors, és que la potència generada s'hagi adquirit com a zero i es mesuri que la potència de l'inversor, de la bateria o del frenat sigui superior a zero, ja que significa que hi ha hagut algun error en el sistema de lectura de potència. Dins d'aquest anàlisi, també es mesura que la tensió de la bateria sigui coherent. El valor de la tensió de la bateria, de la manera com s'ha plantejat el funcionament de la turbina, mentre estigui actiu el dispositiu de control de potència, sempre serà superior a la meitat de la tensió, en conseqüència, sempre es llegiran més de 12



V. Si el sistema llegeix que la bateria té una tensió inferior a aquest nivell de tensió, vol dir que un sensor no està llegint bé la tensió de la bateria, per tant, el sistema s'ha de posar en error.

El següent test que s'examina és el bon funcionament de les comunicacions. En haver dues xarxes de comunicacions, cal tenir en compte els anàlisi que es poden dur a terme. Cada cert temps es comprova que la comunicació entre dispositius de la turbina funciona. Aquesta maniobra de test de la xarxa multimestre es comentarà en el capítol sis de manera més detallada.

Una altra casuística com el sistema entra en estat d'error és a partir del mecanisme de control d'execució del programa. Aquest mètode per posar-se en error, no és pròpiament un anàlisi, sinó que es tracta d'un mecanisme que es fa servir per detectar si el sistema s'ha bloquejat o ha deixat de funcionar correctament. Aquest mecanisme, permet detectar bucles infinits, així doncs, garanteix que el sistema estigui sempre funcionant correctament, i evita que es produeixin errors crítics causats per aquests punts on s'ha encallat el programa.

Una altra manera en que el sistema percep un error és a través de la recepció d'una comanda d'error per la xarxa de comunicació entre dispositius, a causa de la detecció d'un error crític en el sistema de control de l'angle d'inclinació de les pales, o en el sistema de monitoratge de la turbina.

Finalment, l'última manera com el sistema es pot posar en error és pel PLC.

### 2.7.2. Estats i accions

Quan el sistema entra en estat d'error, el procés d'aturada passa per vuit etapes diferenciades. Els seus estats i les accions associades es mostren en la Taula 25.

<b>Etapa</b>	<b>Accions</b>
Etapa 1	Comprovar que l'error no és de Watchdog, si anteriorment hi ha hagut producció, i avisar el PLC d'error a través d'una variable booleana.
Etapa 2	Activar el frenat elèctric al màxim.

Taula 25.1. Etapes i accions de l'aturada de l'error.

<b>Etapa</b>	<b>Accions</b>
Etapa 3	Desactivar el Watchdog i fer una espera abans de curtcircuitar el generador.
Etapa 4	Curtcircuitar el generador i posar a nivell alt els bits de supervisió.
Etapa 5	Comunicar al sistema de control de l'angle d'orientació de la turbina, que el frenat està al màxim, i es rearma el Watchdog.
Etapa 6	Fer una espera per rebre l'angle al màxim.
Etapa 7	Controlar el bucle que s'han rebut malament les dades de l'angle d'inclinació de les pales al màxim, i comunicar al sistema de control de l'angle d'orientació que el frenat està al màxim.
Etapa 8	Forçar un bucle infinit per desbordar expressament el Watchdog.

Taula 25.2. Etapes i accions de l'aturada de l'error.

### 2.7.3. Variables de control i condicions

De la mateixa manera com s'ha plantejat l'aturada d'emergència, les variables de l'aturada d'error es distingeixen segons les variables de canvi d'etapa i les d'informació del PLC. En cas d'aturada per un error, el PLC tampoc no pot canviar els valors llindars. Així doncs, en cas d'error també es vol deixar de dependre a la major mesura possible del PLC. A la Taula 26, es mostren les variables de canvi d'estat.

<b>Nom en el programa de MPLabX</b>	<b>Nom en l'organigrama</b>	<b>Descripció</b>
Error	Error	Variable booleana que inicia l'aturada d'error.
ESTAT_AMB_GENERACIO	Estat de generació	Variable booleana que informa si el sistema prèviament estava amb estat de generació.
Error_WATCHDOG	ErrorWatchdog	Variable booleana que indica si el sistema ha petit un error d'overflow del Watchdog.
Potencia_limit_reles	Prelés	Variable sense signe integer que indica el límit de potència que suporten els relés per no ser danyats.
Intents_Comunicacio	Comptador	Variable integer que controla les voltes del bucle de rebre el missatge de l'angle al màxim de forma correcta.

Taula 26. Nom de les variables de canvi d'etapa.

En el cas de la maniobra d'error, la condició per saber si s'ha rebut l'angle d'inclinació de les pales al màxim es farà mitjançant la comunicació amb el sistema responsable del posicionament de l'angle d'orientació de les pales. Existeix un temps d'espera per tal de curtcircuitar el generador quan existeix un error en l'execució del codi del programa.

La condició de rearmament és exactament la mateixa que la de rearmament provinent d'una emergència, també s'han d'avaluar els bits de revisió de sistema i de revisió dels relés.

El PLC a través de la xarxa de comunicació pot monitoritzar l'aturada deguda a un error del sistema, així com el tipus d'error que l'ha provocat. En la Taula 27, es mostren les dades que permeten supervisar el sistema en cas d'error, aquestes variables no queden plasmades en l'organigrama lògic.

<b>Nom en el programa de MPlabX</b>	<b>Descripció</b>
ERROR_Inversor	Booleà que avisa que hi ha un error de coherència en la lectura de la potència de l'inversor vers la real que hauria de tenir.
ERROR_Bateria	Booleà que avisa que hi ha un error de coherència en la lectura de la potència de la bateria vers la real que hauria de tenir.
ERROR_Frenat	Booleà que avisa que hi ha un error de coherència en la lectura de la potència del frenat vers la real que hauria de tenir.
ERROR_coherencia	Booleà que avisa que hi ha un error de coherència del sistema. És a dir, la potència generada és captada com zero, en un període de generació, i s'està detectant una potència superior a zero en alguns dels receptors d'energia. O que la tensió de la bateria té un nivell de tensió incorrecte.
Error_WATCHDOG	Booleà que avisa que hi ha un error de desbordament del Watchdog.
AvisErrorPLC	Booleà que avisa al PLC que el sistema està en estat d'error.
REVISIO_SISTEMA	Booleà que avisa si cal revisar el sistema després de l'emergència, s'ha d'escriure des del PLC si s'ha revisat.
POSSIBLES_DANYS_RELES_GENERADOR	Booleà que avisa de possibles danys als relés i/o al generador, s'ha d'escriure des del PLC si s'ha revisat.
Error_TEST_I2C	Booleà que indica que s'ha diagnosticat un error en el test de comunicacions.

Taula 27. Nom de les variables que es poden supervisar en el PLC.

Cal destacar que el PLC, quan treballa amb condicions normals de funcionament, pot realitzar un test, per comprovar que en cas d'error el sistema pari de forma correcte.

### 2.7.4. Organigrama lògic

L'organigrama lògic es mostra en la Figura 16.

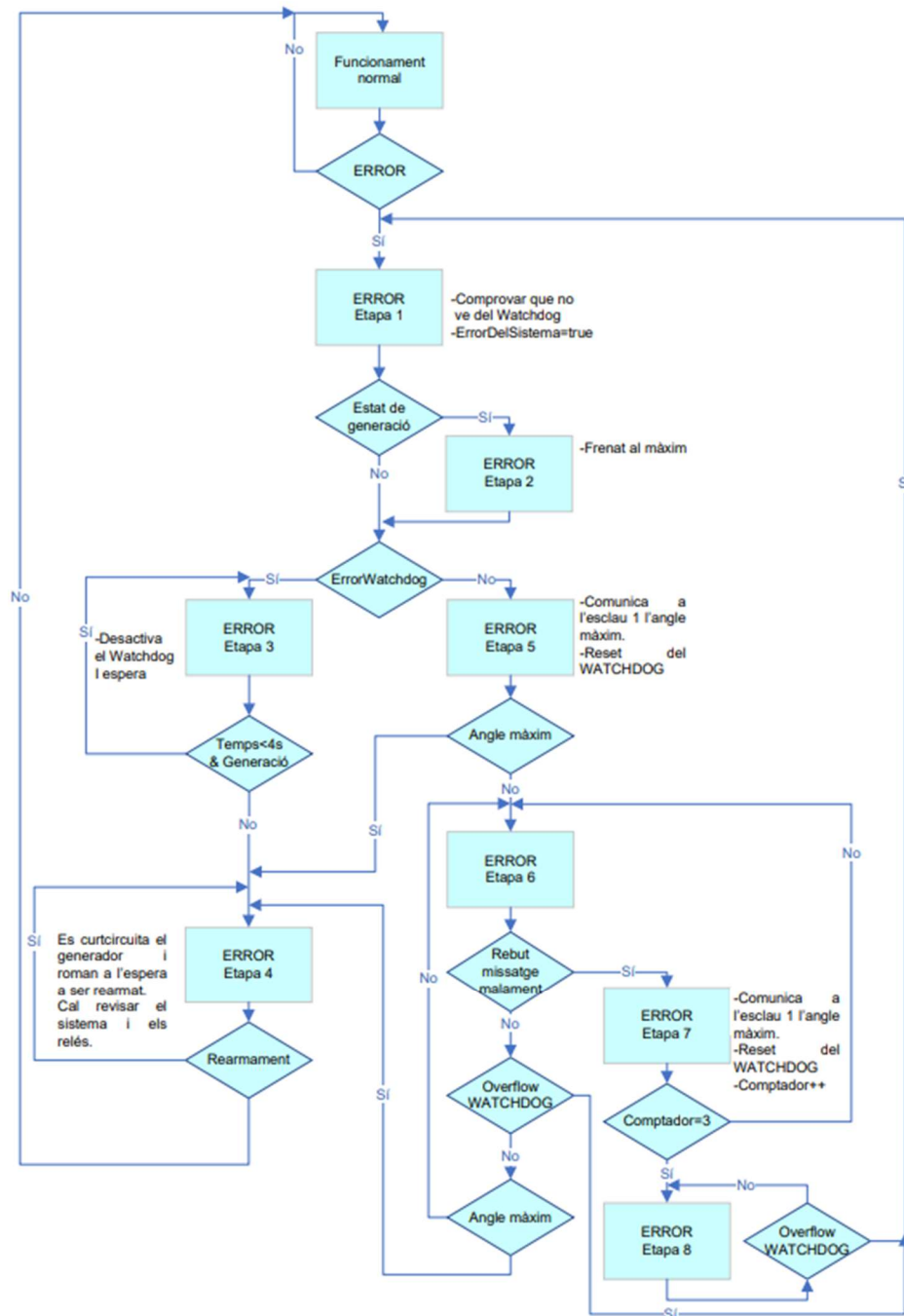


Figura 16. Organigrama lògic de l'aturada d'error.

Quan es diagnostica un error, el bit d'error es posa a nivell alt. Arribat aquest punt, el sistema entra en l'etapa u de l'aturada d'error, on es comprova que l'error no prové de la fallada en l'execució del codi, i s'activa la bandera booleana, que envia al PLC que el sistema es troba en la maniobra d'error. En cas que prèviament el sistema estigués en una etapa de generació, el sistema passa en l'etapa dos on s'activa el frenat elèctric al màxim, en cas de no haver generació, el sistema salta aquesta etapa, i passa directament a l'etapa tres o l'etapa cinc, en funció de si hi ha hagut un error causat per una execució deficient del codi. En l'etapa tres, es fa una espera de quatre segons abans de passar a l'etapa següent, només en la situació on el sistema prèviament estava amb generació, en cas de no haver hagut generació, l'espera no es realitza i es passa directament a l'etapa quatre, on el sistema curtcircuita el generador a través dels relés, posa a nivell alt els bits de revisió del sistema, i roman a l'espera a ser rearmat. En cas que l'error hagués sigut diferent a l'execució del codi deficient, el sistema va a l'etapa cinc. Donat l'escenari, el sistema comunica a l'Esclau 1 que el frenat elèctric està al màxim, significat que hi ha hagut un error, i que el mecanisme de frenat ha d'actuar. Si no es rep l'angle d'orientació de les pales al màxim, en l'etapa sis es fa una espera d'on no en sortirà fins que no es rebi correctament. En cas d'haver rebut el missatge d'angle d'inclinació de les pales al màxim, però que el missatge no coincidís amb el byte que s'espera rebre es torna a comunicar al sistema de control de l'angle, que el frenat està al màxim, per rebre la dada d'angle al màxim correctament. Aquest procés es repeteix fins a tres vegades, cada vegada passant per l'etapa set, on es duu a terme el control d'aquest bucle, sumant el comptador de voltes, i enviant la comanda de fre al màxim al sistema de control de l'angle d'inclinació de les pales. Quan el comptador supera el nombre màxim de dades errònies rebudes, el sistema va a l'etapa vuit on es produeix voluntàriament un bucle infinit a l'espera de forçar una errada en l'execució del codi del programa, i que es torni a tractar l'error. Si el sistema es troba en l'etapa sis, i no es rep cap dada en el temps que es pot desbordar el sistema de control d'execució del codi del programa, equival a que el sistema es troba en un bucle infinit, i es reinicia de nou l'aturada d'error. Si el sistema rep l'angle de manera correcta, el sistema va a l'etapa quatre.

Cal destacar que l'error en l'execució del codi de programa, reinicia el sistema complet, i com a conseqüència, també l'aturada d'error.

### **2.7.5. Estratègia i justificacions**

L'aturada d'error té un grau de domini del sistema menor que l'aturada d'emergència, és a dir, si en condicions normals de funcionament, el sistema capta un error, es deixa el que s'estava

fent, i s'atura el sistema seguint els passos de l'organigrama. En cas d'una aturada d'error, si el sistema es percep l'emergència, es passa de l'aturada d'error a l'aturada d'emergència. En conseqüència, l'operació prioritària és l'emergència, i la segona més prioritària és l'error. Davant d'un error, per seguretat s'atura el sistema, per evitar que a causa d'una lectura errònia d'un sensor, o un bucle infinit, s'arribi a una situació crítica. Quan es detecta un error, el sistema de control de consum de potència, inicia l'aturada, informant al PLC. A més, gràcies als tests, es pot conèixer el tipus d'error que s'ha donat.

Les situacions d'emergència, sempre es donen quan l'aerogenerador està en règim de generació energètica. Quan no hi ha generació, com que la turbina no està funcionant, l'alternador no genera electricitat, provocant que la instal·lació elèctrica aigües avall de la turbina no es danyi, ni es causen pertorbacions mecàniques que puguin danyar l'estructura del molí de vent. Però quan el sistema de control de potència transfereix càrrega sense generació, és a dir, les càrregues s'alimenten únicament amb l'energia provinent de la bateria, el sistema pot captar errors, tant en els sensors, com en les comunicacions. Les situacions que es poden captar com anòmales, quan no hi ha generació, es tracten com a errors, però a efectes pràctics es podria considerar com una emergència, ja que el sistema deixa el que estava duent a terme, atura la producció, i es reclama una revisió. Llavors el sistema no es pot rearmar fins que no es revisa tot. Pels motius esmentats, en la maniobra d'error s'ha d'analitzar segons si es troba en generació o no. Si el sistema diagnostica un error, mentre està en generació, el sistema s'ha de parar, i per tant han d'actuar els dos mecanismes de frenat de la turbina, tant el frenat elèctric com el frenat per l'angle d'inclinació de les pales. En cas de no haver-hi generació, el sistema no cal que pari la turbina, ja que si s'apliqués el frenat elèctric, en no haver-hi generació, es gastaria energia de la bateria per activar el frenat, sent una pèrdua d'eficiència energètica. En aquest segon cas, encara que el sistema funcioni a mode baix consum els tests es poden realitzar igualment, el motiu es comentarà en el capítol referent a la programació.

Com queda detallat, l'aturada d'error, igual que la d'emergència, es realitza de dues maneres diferents. Quan l'error és d'execució del codi del programa, el motiu principal és que alguna de les xarxes de comunicacions ha desencadenat un llaç infinit. Així doncs, quan es produeixen errors d'aquest tipus, i el sistema no es troba en generació, el generador queda bloquejat per evitar que entri en generació, es desactiven els relés que hi havia actius, i el sistema roman a l'espera de ser rearmat i revisat. Si l'error es percep quan hi ha generació, el sistema fa una espera de quatre segons, perquè el frenat elèctric per si sol freni la turbina, sense considerar el control de l'angle d'orientació, el control sobre l'execució del codi del

programa es desactiva i el sistema entra a l'etapa quatre. En aquest cas, si l'error és degut a la comunicació amb el PLC ho detecta, i no li envia res més en el sistema de control de consum de potència, per evitar que es torni a recaure a un error d'aquest tipus. Per tant, l'encarregat de manteniment de l'aerogenerador, si es produeix un error d'aquest tipus, com que el PLC ha detectat que la comunicació no s'ha pogut produir, primer haurà de reparar la xarxa, i després haurà de rearmar el sistema, en el cas que no es tregui l'energia en el microcontrolador, considerant l'estat de les bateries. Si l'error s'ha produït a causa d'una errada de comunicacions entre esclaus, el sistema no quedarà encallat de nou, i per aquest motiu s'opta per fer l'espera de quatre segons, prèvia a curtcircuitar el generador. Així doncs, donada la situació, com que la comunicació amb el PLC continua funcionant, es detecta l'error a causa de la comunicació entre esclaus.

Si la turbina està produint electricitat, i es produeix un error diferent d'una errada en l'execució en el codi de programa, el frenat elèctric es connecta al màxim, i es comunica que el frenat està al màxim al sistema de control de l'angle d'orientació de les pales. En aquest escenari, l'error pot provenir o d'una lectura errònia d'un dels sensors, o d'un error del test del sistema de comunicacions causat perquè les dades s'han llegit malament. Si la incidència es deu a un error de comunicació, el sistema comunica igualment que el frenat està al màxim, i es donen fins a tres intents de rebre l'angle de forma correcta, d'aquesta manera, si es rep un missatge, però no coincideix amb la comanda d'angle d'inclinació de les pales al màxim, el sistema espera a rebre el missatge de forma correcta durant tres intents. Per a cada dada rebuda, el sistema de control de potència comunica de nou al sistema de control de l'angle d'orientació de les pales, que el frenat està al màxim, i suma un comptador d'intents. Passats tres intents, és a dir, que s'han rebut tres dades errònies, en el sistema se li força un bucle infinit per provocar una errada en el control de l'execució del codi del programa. El motiu de continuar la comunicació amb el sistema de control de l'angle d'inclinació de les pales, en cas de diagnosticar-se una lectura de la informació del canal de comunicacions errònia, és la probabilitat que cessi la font d'emissions electromagnètiques que provoqui que les dades es rebin de forma errònia, o que davant d'un entorn de pertorbacions que facin que els missatges no es rebin de forma correcta, en un intent de comunicació les dades es rebin de la forma apropiada i el sistema pari la producció a través dels dos frenats. Si s'exhaureixen els intents, com que el frenat ja s'ha posat el màxim, en aquest temps el frenat haurà actuat reduint la velocitat, i com que addicionalment, es força un llaç infinit per garantir una espera perquè el frenat actuï, la velocitat de rotació serà tan petita, que es podria curtcircuitar el generador, sense perill a danyar severament el sistema. Quan el sistema no es troba en generació, també s'envia el missatge que el frenat està al màxim, encara que no estigui connectat. Aquest fet succeeix perquè el sistema de control de l'angle d'inclinació de les pales mogui les pales en

paral·lel al vent, per evitar que el sistema es posi en generació, i posteriorment es curtcircuita l'alternador, per evitar el gir de la turbina.



### 3. INSTAL·LACIÓ ELÈCTRICA

En el present capítol, queda explicat tots els aspectes que s'han tingut en compte ple disseny elèctric de la instal·lació. D'aquesta manera, el capítol queda desglossat en cinc apartats: en el primer apartat es detalla el dimensionament dels conductors, tant de potència com de control, i les canalitzacions; en el següent es fa esment dels dispositius de protecció elèctrica implementats, en el tercer apartat s'explica com s'ha compensat la reactiva, en el quart apartat es descriu la posada a terra de la instal·lació, i en l'últim s'indiquen els criteris de disseny del quadre elèctric.

Tot el que queda comentat en aquest capítol queda reflectit en els documents: de Plànols, de l'Estat d'amidaments i del Pressupost. I els càlculs justificatius dels diferents apartats d'aquest plànol es descriuen en l'apartat B.1 de l'Annex B de càlculs.

#### 3.1. Cables conductors

Pel dimensionament dels conductors de l'aerogenerador, com que és una instal·lació de baix consum treballant en illa, i hi ha circuits que funcionen amb corrent continu, i circuits amb corrent altern, en el dimensionament dels cables pel que fa el criteri d'intensitat màxima admissible del conductor, s'ha fet seguint la ITC-BT-19, que es mostra en la Figura 17, i per les línies enterrades la norma UNE 20460-5-523, que es mostra en la Figura 18. Pel criteri de caiguda de tensió, s'ha seguit els patrons descrits a l'IDAE, que són els mateixos dimensionar una instal·lació fotovoltaica.

Segons l'IDAE, en la línia que connecta el conjunt de panells fotovoltaics amb el regulador la caiguda de tensió màxima permesa és d'un u per cent, així com la línia que va del regulador de càrrega a la bateria, la que va del regulador de càrrega a l'inversor, la del regulador de càrrega al frenat dissenyat, i la del regulador de càrrega al frenat que incorpora el regulador per defecte. Pel circuit que connecta l'inversor amb les càrregues la caiguda de tensió no pot superar el tres per cent. En l'aerogenerador que s'està dimensionant s'ha seguit exactament el mateix criteri, però l'energia provés d'un alternador en comptes d'un conjunt de panells fotovoltaics.

A		Conductores aislados en tubos empotrados en paredes aislantes	3x PVC	2x PVC	2x XLPE o EPR	2x XLPE o EPR									
A2		Cables multiconductores en tubos empotrados en paredes aislantes	3x PVC	2x PVC	3x XLPE o EPR	2x XLPE o EPR									
B		Conductores aislados en tubos en montaje superficial o empotrados en obra			3x PVC	2x PVC			3x XLPE o EPR	2x XLPE o EPR					
B2		Cables multiconductores en tubos en montaje superficial o empotrados en obra		3x PVC	2x PVC		3x XLPE o EPR		2x XLPE o EPR						
C		Cables multiconductores directamente sobre la pared <sup>1)</sup>				3x PVC	2x PVC		3x XLPE o EPR	2x XLPE o EPR					
E		Cables multiconductores al aire libre <sup>2)</sup> Distancia a la pared no inferior a 0,3D <sup>3)</sup>					3x PVC		2x PVC	3x XLPE o EPR	2x XLPE o EPR				
F		Cables unipolares en contacto mutuo <sup>4)</sup> Distancia a la pared no inferior a D <sup>5)</sup>						3x PVC			3x XLPE o EPR <sup>6)</sup>				
G		Cables unipolares separados mínimo D <sup>5)</sup>								3x PVC <sup>7)</sup>		3x XLPE o EPR			
			mm <sup>2</sup>	1	2	3	4	5	6	7	8	9	10	11	
Cobre			1,5	11	11,5	13	13,5	15	16	-	18	21	24	-	
			2,5	15	16	17,5	18,5	21	22	-	25	29	33	-	-
			4	20	21	23	24	27	30	-	34	38	45	-	-
			6	25	27	30	32	36	37	-	44	49	57	-	-
			10	34	37	40	44	50	52	-	60	68	76	-	-
			16	45	49	54	59	66	70	-	80	91	105	-	-
			25	59	64	70	77	84	88	-	106	116	123	-	-
			35	77	86	96	104	110	119	-	131	144	154	-	-
			50	94	103	117	125	133	145	-	159	175	188	-	-
			70			149	160	171	188	-	202	224	244	-	-
			95			180	194	207	230	-	245	271	296	-	-
			120			208	225	240	267	-	284	314	348	-	-
			150			236	260	278	310	-	338	363	404	-	-
			185			268	297	317	354	-	386	415	464	-	-
240			315	350	374	419	-	455	490	552	-	-			
300			360	404	423	484	-	524	565	640	-	-			

Figura 17. Taula de la ITC-BT-19 pel dimensionament dels conductors.

Método de instalación	Sección mm <sup>2</sup>	Número de conductores cargados y tipo de aislamiento			
		XLPE3	XLPE2	PVC3	PVC2
	Cobre	21	24,5	17	20,5
	1,5	27,5	32,5	22,5	27,5
	2,5	35	42	29	36
	4	44	53	37	44
	6	58	70	49	59
	10	75	91	63	76
	16	96	116	81	98
	25	117	140	97	118
	35	138	166	115	140
	50	170	204	143	173
	70	202	241	170	205
	95	230	275	192	233
	120	260	311	218	264
	150	291	348	245	296
	185	336	402	282	342
	240	380	455	319	387

Figura 18. Taula 52-2 de la UNE 20460-5-523 pel dimensionament dels conductors.

Tots els aïllants dels conductors de les línies de potència, tenen un nivell d'aïllament de 450/750 V, i estan fabricats de XLPE, que és un material lliure d'halògens. Així doncs, tots els cables unipolars usats són del tipus H07Z1-K, i les deus mànegues que s'usen que són les que connecten: l'inversor amb la bomba i l'inversor amb els terminals d'entrada dels focus, són de tipus RZ1-K. En les línies on la tensió és contínua, és important utilitzar cables de secció prou gran per evitar caigudes de tensió excessives. A més, en circuits de frenat es poden arribar a obtenir corrents molt alts en situacions d'emergència, i per tant cal preveure aquestes situacions en el càlcul de les seccions dels conductors.

En la Taula 28, es mostra com queden distribuïdes les línies de l'aerogenerador, amb les seves característiques.

Línia	Tipus de tensió	Potència màxima (W)	Longitud (m)	Secció (mm <sup>2</sup> )	Tipus d'instal·lació REBT i UNE	Caiguda de tensió (%)
Generador	3X230VAC	1800,00	15,00	10,00	B	0,04
Regulador-Bateria	24V CC	1800,00	2,50	35,00	B	0,98
Regulador-Frenat emergència	24V CC	1800,00	2,50	35,00	B	0,98
Regulador-Frenat dissenyat	24V CC	1800,00	2,00	35,00	B	0,79
Banc frenat 1	24V CC	300,00	10,00	25,00	B	0,92
Banc frenat 2	24V CC	300,00	10,00	25,00	B	0,92
Banc frenat 3	24V CC	300,00	10,00	25,00	B	0,92
Banc frenat 4	24V CC	300,00	10,00	25,00	B	0,92
Banc frenat 5	24V CC	300,00	10,00	25,00	B	0,92
Banc frenat 6	24V CC	300,00	10,00	25,00	B	0,92
Regulador-Inversor dc	24V CC	1800,00	1,50	25,00	B	0,82
Línia de sortida de l'inversor	230VAC+N	1800,00	1,00	2,50	B	0,06
Inversor-bomba (quadre)	230VAC+N	800,00	1,00	2,50	B	0,03
Inversor-focus (quadre)	230VAC+N	600,00	1,00	2,50	B	0,02
Inversor-bomba (exterior)	230VAC+N	800,00	40,00	2,50	D	1,06
Inversor-focus mànega (exterior)	230VAC+N	600,00	30,00	2,50	D	0,60

Taula 28. Característiques de les línies de l'aerogenerador.

Els cables que alimenten els elements de control i els sensors, també estan aïllats amb material lliure d'halògens, i la secció del conductor és de 2,50 mm<sup>2</sup>.

El color dels cables s'han seguit els criteris de la norma UNE-EN 60204-1, pertinent a l'equip elèctric i a la seguretat de les màquines, ja que en haver elements de potència i de control,

l'instal·lador i l'encarregat de mantenir el molí pot treballar de forma segura i seguint una metodologia òptima. Aquesta norma també s'aplica en els cables de control.

Els cables de control amb senyals digitals, com que el corrent que pot circular a través d'ell és molt petit, l'aïllament és de PVC i la secció és de 1,5 mm<sup>2</sup>, aquests criteris també venen estipulats per la norma UNE-EN 60204-1.

En les canalitzacions de l'exterior del quadre elèctric, s'ha aprofitat l'estructura del molí. El tub buit que comunica la base del molí, on hi ha el quadre, amb la gòndola, on hi ha l'alternador, hi circulen diferents conductors, tant de control com de potència. En els bancs de frenat i el frenat que ve de fàbrica, com que es disposen de manera contigua al molí, s'ha emprat un tub corrugat de 32 mm arran de terra. I en les línies que alimenten les càrregues del molí, s'utilitza una canalització enterrada amb tub corrugat però de 20 mm. Les connexions del regulador amb l'inversor i la bateria, en estar tots els elements a la base del molí, no cal canalització, es pot fer embridant i grapant els cables a la paret.

### **3.2. Proteccions**

Per les protegir els diferents elements de la instal·lació, no només s'aprofiten els elements de protecció elèctrica que disposen el regulador de càrrega i la turbina. Per protegir de curtcircuits i sobrecàrregues, el circuit aigües avall de la instal·lació, s'ha implementat un magnetotèrmic.

Concretament, el dispositiu escollit és el model de la marca Schneider Acti9 iC60L. Aquest té una potència nominal de 1,45 kW, per tant, com que les càrregues consumeixen 1,4 kW, s'assegura que la línia no es sobrecarregui. Altrament, té una poder de tall de 20 kA.

Aquest interruptor automàtic no es disposa a la placa principal del quadre elèctric, sinó que es situa de manera contigua a l'inversor amb una caixa de protecció amb carril DIN d'IP66.

En cas de disparar-se el magnetotèrmic, l'element de control percep que la potència de l'inversor és zero, posant-se en estat d'error, ja que la potència de referència de l'inversor i la real difereixen. També es pot posar en emergència, ja que la demanda de potència passa a ser nul·la, i la turbina pot accelerar, actuant qualsevol dels dispositius de control.

### 3.3. Compensació de la potència reactiva

Amb la finalitat d'augmentar l'eficiència del sistema, s'ha decidit corregir el factor de potència de les càrregues. A més a més, com que el factor de potència de l'alternador depèn de les càrregues connectades, com més proper a la unitat sigui la potència de les càrregues, més fiable serà el sistema de mesura de potència, ja que es mesura la potència aparent, i es considera que el factor de potència és la unitat.

L'inversor, a partir de l'electrònica de potència, permet corregir el factor de potència de manera parcial. Així doncs, s'ha decidit incorporar el condensador de compensació de 10  $\mu\text{F}$  de la marca Ducati Energia, per corregir amb la major mesura possible el factor de potència. El seu càlcul es pot veure en l'Annex B.

### 3.4. Posada a Terra

Totes les masses metàl·liques del sistema, tant l'estructura de tot el molí, com les carcasses receptors o l'alternador, estan connectades a terra. La posada a terra s'ha fet a partir d'una elèctrode de posada a terra, dispostat de la manera com indica a la Figura 19.

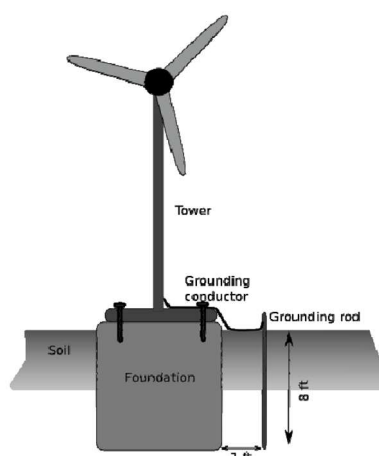


Figura 19. Posada a terra de la instal·lació.

A diferència de la imatge, el conductor de posada a terra surt de la platina de coure del quadre elèctric, i la línia principal de posada a terra és un conductor de 10  $\text{mm}^2$ , que passa per un born desendollable de posada a terra. Per tant, la línia principal surt de la base del molí cap a l'elèctrode gràcies a una canalització de tub corrugat, que es connecta amb l'elèctrode, que s'ha determinat de 2,5 m, valor superior als 8 ft que recomana el fabricant. El valor de

resistència de posada a terra és de 200  $\Omega$ , i que per 24 V, el corrent de fuga és de 12 mA, valor inferior a 30 mA, complint els valors marcat per la ITC-BT-18. Els càlculs de la posada a terra queden demostrats a l'Annex B.

### **3.5. Quadre elèctric**

El quadre elèctric es disposa a la cavitat de la base de la turbina, que està formada de formigó. La placa del quadre mesura 800 mm d'alçada per 800 mm d'amplada, i queda acoblada en una de les parets d'aquesta cavitat. Els components elèctrics queden disposats a través del carril DIN de 35 mm, que està reblat a la placa, i tots els conductors circulen per una canaleta de 40 mm d'amplada.

Les connexions de potència amb els elements externs es fan a través de borns. Pel dimensionament d'aquests borns, s'ha contemplat que la secció d'aquests borns s'ajusti a la secció del conductor, que l'acoblament entre borns i conductors es duu a terme a través de punteres. L'altre criteri que s'ha seguit és que el corrent nominal dels borns ha de ser superior al corrent nominal que circula pel circuit que connecta.

Tots els conductors de terra, es connecten a una platina de coure a través de connectors de rosca, i aquesta platina connecta tots els elements amb la línia principal de posada a terra, que passa pel born desendollable, anteriorment mencionat.

## 4. CIRCUIT DE CONTROL

En el present capítol es detallaran els diferents elements del circuit de control. El circuit de control es pot desglossar per diferents parts: sensors, relés, elements que intervenen a les comunicacions i el dispositiu de control. En cada apartat del present capítol es comenta el seu funcionament. En la Figura 20, hi ha un esquema de com queden distribuïts els elements del maquinari de control.

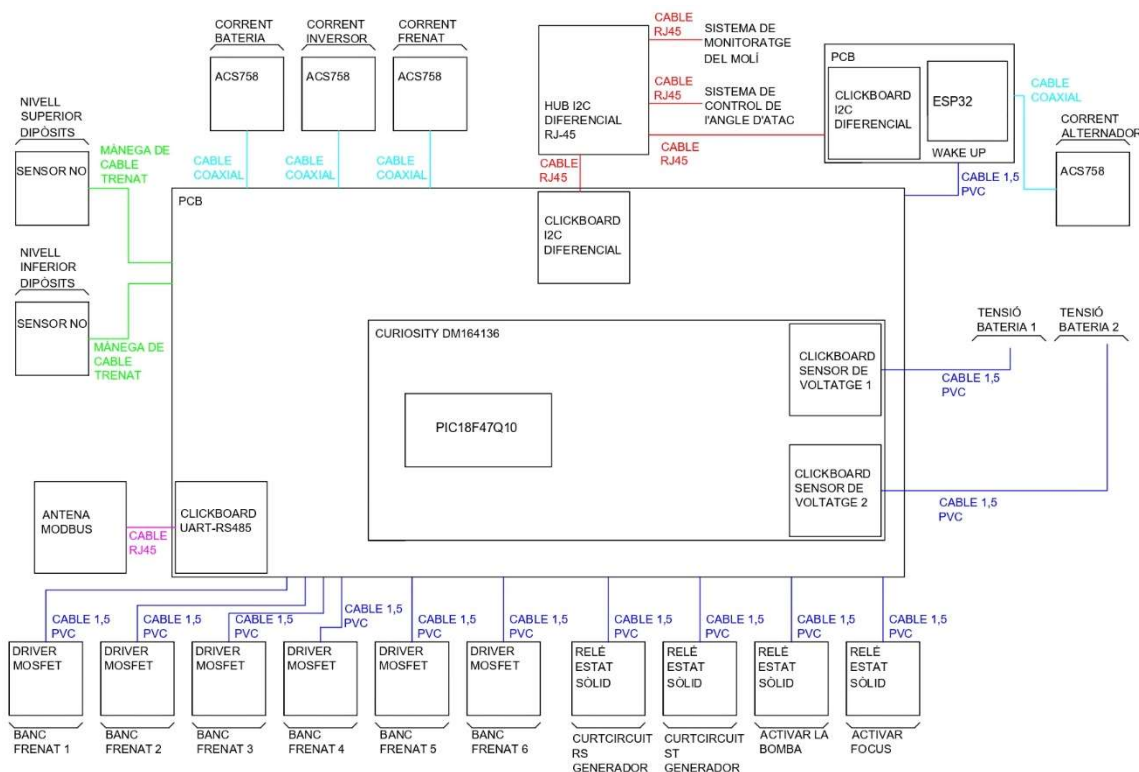


Figura 20. Disposició dels elements del maquinari de control.

Cal destacar que els senyals d'alimentació de cinc volts, i el de zero volts, gràcies als distribuïdors de tensió del quadre, tots els elements de control tenen el mateix senyal d'alimentació i la mateixa massa. Així doncs, tant les comunicacions, com les lectures dels sensors poden funcionar sense errors causats per les diferències de potencial entre masses.

### 4.1. Criteris d'elecció dels components

En el present apartat, s'esmenta com s'han elegit els diferents elements en funció de les necessitats del sistema, explicades en el capítol dos.

#### **4.1.1. Elecció del dispositiu de control**

De la manera com s'ha plantejat en el capítol dos, el sistema de control ha de ser multitasca: ha de poder llegir les dades dels sensors, ha de comunicar-se amb els altres dispositius, i ha de fer el seguiment de tots els estats i accions de control que es desenvolupen durant la transferència de potència. Per dur a terme les tasques que intervenen en el sistema, s'ha escollit el microcontrolador PIC18F47Q10. Amb aquest microcontrolador PIC, pot desenvolupar un sistema multitasca, a partir dels perifèrics que incorpora que li permeten desenvolupar interrupcions. Molts processos es poden sincronitzar per interrupcions temporitzades, amb qualsevol dels sis temporitzadors que incorpora; la lectura de dades amb l'ADCC també es pot dur a terme a través d'interrupcions associades a aquest perifèric, i les comunicacions tant de la EUSART i l'I2C, també es poden sincronitzar gràcies a les interrupcions que es produeixen quan es reben les dades. Conseqüentment, és un dispositiu ideal per implementar un sistema multitasca i precís en quant a temporització.

#### **4.1.2. Elecció del elements de mesura**

Els sensors són els elements que permeten conèixer l'estat del sistema de control de consum de potència. Segons es pot analitzar en el capítol dos, per transferir potència a les càrregues o parar el sistema per error o emergència, es comparen les mesures elèctriques amb diferents llinars. Així doncs, per calcular les potències dels diferents elements de generació i recepció de potència, es mesura: el corrent del generador, que es processa amb el microcontrolador ESP32 per obtenir la potència generada via I2C; el corrent que circula cap a la bateria, el corrent que es destina al frenat, la intensitat que estan consumint les càrregues connectades a l'inversor; la tensió a les bateries, per conèixer l'estat de càrrega; i el sensor de nivell dels dipòsits per saber si estan plens per accionar la bomba. Les diferents variables elèctriques internament per programa es processen per obtenir valors de potència.

#### **4.1.3. Elecció dels elements de comunicació**

El circuit de control queda comunicat amb els altres microcontroladors a través d'una xarxa I2C. Aquest protocol de comunicació permet crear una xarxa multimestre entre els diferents elements de control de la turbina, gràcies als dos MSSP que disposen els tres PIC que estan al molí, que els permeten funcionar com a mestre i com a esclau a la vegada. Per tal de donar fiabilitat i robustesa al bus, cal implantar diferents components, com convertidors d'I2C diferencial.



Com que a través del PLC es pot monitorar i actuar sobre variables del sistema de control de potència, s'ha implementat una xarxa de Modbus RTU, per comunicar el PLC, treballant com a mestre, amb el PIC, que funciona com a esclau. Com que els PIC disposen de mòduls EUSART, es permet que a través d'aquests perifèrics desenvolupar el protocol del Modbus, gràcies a la implementació de les passarel·les RS485 adequades.

## 4.2. Dispositiu de control

El dispositiu de control està format per una Placa Curiosity DM164136 de la companyia Microchip amb el microcontrolador PIC18F47Q10, juntament amb una PCB per facilitar les connexions amb els altres elements del sistema. El seu nucli és de vuit bits, i gràcies als perifèrics integrats i les elevades freqüències que pot tenir el senyal de rellotge de processament, està optimitzat per aplicacions on s'involucren nodes de sensor i control robusts en temps real

El microcontrolador escollit és de la gamma PIC de l'empresa Microchip. Els PIC són microcontroladors d'arquitectura Harvard, que és el tipus d'arquitectura que separa físicament la memòria de dades i la de instruccions. La memòria d'instruccions és on hi ha el conjunt de registres de control, que dominen recursos del maquinari del xip, que queden assignats a través de la memòria de dades, funcionant com una memòria RAM. A l'hora de programar el dispositiu, depenent del valor que se li assigni a cada registre, es configura el dispositiu, i els recursos del maquinari poden quedar habilitats o inhabilitats.

Els microcontroladors PIC, o microcontroladors de connexió perifèric, permeten descentralitzar tasques de la CPU, reduint el consum total d'energia, punt favorable per poder treballar en el molí de vent amb alimentació per bateries, i alhora ofereix respostes deterministes, en altres paraules permet sincronitzar i seqüenciar tasques seguint un fil lògic, en el que un acció és conseqüència de l'altre. Altrament, en treure dependència de la CPU es redueix la mida del codi, estalviant temps de desenvolupament, i es proporciona més espai de memòria per a l'aplicació. El PIC18F47Q10 té 40 pins, i els perifèrics independents del nucli que s'han utilitzat són: el CLC, que és el dispositiu que s'emprarà pel funcionament del frenat elèctric, el WWDT, l'ADCC, un generador de senyal PWM, dues EUSART, dos MSSP, sis temporitzadors i el Peripheral Pin Select. Addicionalment, pot funcionar a cinc volts, augmentant la immunitat al soroll. A més a més, és un microcontrolador de tipus ULP, o molt baix consum, propietat que el fa molt adequat pel sistema de control del consum de potència que s'està dissenyant. Quan està en mode sleep, els nivells de consum energètic

disminueixen dràsticament, permetent que en situacions que no hi ha generació optimitzar el consum de la bateria.

### 4.3. Sensors

En total el sistema de control de potència llegeix tres tipus de sensors: sensors de corrent, sensors de tensió i sensors de nivell.

Tots els sensors de corrent són del model ACS758LCB 100B, del fabricant Allegro MicroSystems. Aquests dispositiu consisteixen en sensors d'efecte Hall de precisió, linealitzats de baix offset. La mesura s'obté a partir d'una pista de coure a prop de la matriu per on circula el propi corrent, que es vol mesurar. La intensitat del circuit de potència flueix per aquesta ruta de coure que genera un camp magnètic, que el sensor d'efecte Hall converteix el valor de intensitat en un valor de tensió proporcional. Un element a destacar d'aquest sensor és l'escut integrat patentat per Allegro, que aporta immunitat d'alt nivell al conductor de corrent i errades de camps elèctrics, és a dir, el soroll i les interferències electromagnètiques.

Una de les grans avantatges d'aquest dispositiu és que està completament calibrat abans de l'enviament de fàbrica. A més a més, el model escollit mesura des de -100 A a 100 A. Quan el corrent és de -100 A, la tensió de sortida és de zero volts, i quan la tensió és de +100 A, la tensió de sortida és de cinc volts. En cas de no circular corrent, la tensió de sortida és de 2,5 V. En la Figura 21 es pot veure com l'aspecte del sensor amb la carcassa del carril DIN.



Figura 21. Sensor ACS758 en el carril DIN.

Cal destacar que el cable amb el que es connecta el sensor amb les entrades analògiques del microcontrolador, és coaxial, per evitar l'acoblament d'interferències en el senyal que poden donar valors erronis. Per tant, s'ha hagut de comprar una mànega de cable d'aquest tipus,

juntament amb els connectors pertinents, tant mascles com femelles per dur a terme el connexionat.

El sensor de tensió és un voltímetre de la marca Mikroelektronika, del model de la gamma ClickBoard. Per a cada bateria, hi ha un sensor de voltatge. Els cables que porten el senyal de la bateria al sensor són de PVC de color blau, i es connecten amb ell a través de la regleta que incorpora. A la Figura 22, es pot veure d'una manera més clara aquest concepte de sensor.

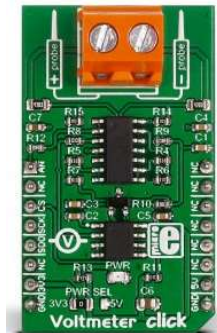


Figura 22. Sensor de voltatge ClickBoard.

Aquest transductor adquireix les dades de tensió a partir d'una entrada diferencial, que a través d'un circuit de condicionament format per seguidors de tensió, i un amplificador diferencial, condiona una entrada de 0 a 24 V, als valors que accepten les entrades analògiques del microcontrolador.

Finalment, pel control de l'accionament de la bomba, s'utilitzen dos interruptors de nivell de tipus flotador de la marca RS Pro, com es pot veure a la Figura 23, i van connectats amb una mànega de cables trenats de 1,5 mm<sup>2</sup> de PVC, que connecten l'alimentació de cinc volts del microcontrolador amb les entrades digitals. Aquests sensors estan dissenyats per treballa en entorns submergits.



Figura 23. Sensor de nivell RS PRO.

#### 4.4. Relés

El sistema de control per tal de controlar certs elements del sistema de potència, disposa de quatre relés d'estats sòlid, instal·lats en diferents parts del molí. Dos dels relés es dediquen a curtcircuitar el generador per aturar-lo. Els altres dos relés, un es dedica a connectar o desconnectar la bomba, i l'altre els focus.

Els relés d'estat sòlid de sortida alterna monofàsica són de la sèrie KSIM de i-Autoc. Aquests tenen una tensió de control de 4 V a 32 V de tensió contínua, i poden commutar càrregues de tensió alterna amb pics de corrents màxims de 25 A. Els relés disposen d'una sortida triac, i es poden fer servir per a càrregues resistives, inductives o capacitives adequades per utilitzar en aplicacions industrials. A més a més, tenen un indicador LED que marca si el relé està activat.

#### 4.5. Comunicacions

La xarxa I2C per evitar pertorbacions a les comunicacions és de tipus diferencial, concretament, en lloc de ser dos cables, un per l'SDA o canal de dades i l'altre per l'SCL, o senyal de rellotge; el protocol està format per 4 canals, dos per les dades: l'SDA+, i l'SDA-, i dos pel rellotge: l'SCL+ i l'SCL-. Per fer la conversió de dos a quatre canals, s'utilitza el dispositiu ClickBoard de la marca Mikroelektronika. El convertidor de I2C, els canals de sortida queden agrupats en un connector RJ45. Com que la mànega de RJ45 està formada per cables trenats i disposa d'una pantalla conductora connectada a terra, les pertorbacions electromagnètiques afecten molt poc el canal de comunicació. Com a conclusió, no només la comunicació és diferencial, sinó que també en estar agrupats en un cable RJ45, es garanteix que la comunicació sigui més robusta. Una altra propietat ideal d'aquest convertidor és que l'alimentació de la interfície diferencial, i la de la interfície de canal únic estan separades. D'aquesta manera, tal com es pot veure en l'esquema de la Figura 24, una part del circuit pot funcionar a cinc volts, i l'altra pot funcionar a 3,3 V, fent de convertidor de nivell.

En cas de voler canviar els nivells de tensió, s'ha de dessoldar i tornar a soldar uns jumpers, escollint de nou la tensió que pot ser de 3,3 V o de cinc volts. A través dels jumpers, es pot escollir la tensió de treball del circuit d'entrada i el de sortida.

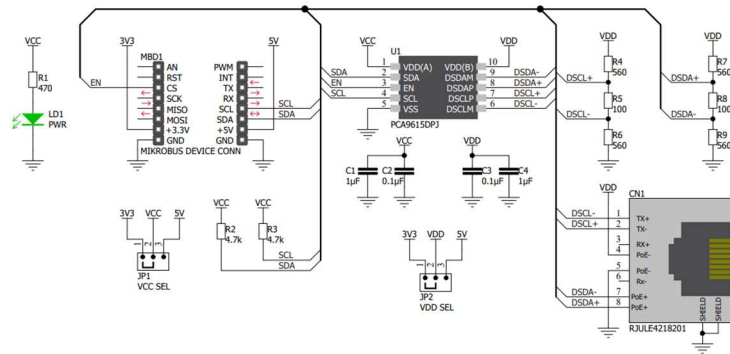


Figura 24. Esquema del convertidor de I2C diferencial.

Aquests jumpers s'han de soldar a la PCB del convertidor, tal com es mostra en la Figura 25. El senyal de l'habilitació, sempre estarà a nivell alt.

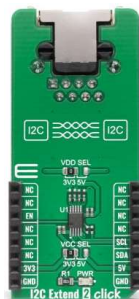


Figura 25. PCB del convertidor de I2C diferencial.

En tractar-se d'una xarxa I2C multimestre, implica que hi ha d'haver un punt per interconnectar tots els dispositius involucrats en la xarxa. Com que el canal de comunicació es tracta d'un cable RJ45, per interconnectar els dispositius es necessita un dispositiu centralitzador en la connexió. El centralitzador elegit és de la marca Campbell, i està format per 8 ports RJ45. Com que el centralitzador de connexió és d'I2C no li cal una alimentació externa, simplement, acobla els senyals a partir de rutes en una PCB, en conseqüència, no és un centralitzador d'Ethernet. En la Figura 26, hi ha la imatge de com és el centralitzador usat.



Figura 26. Centralitzador d'I2C de 8 ports de RJ45.

La xarxa de Modbus interconnecta el PLC mestre amb els diferents microcontroladors de l'aerogenerador. Per al funcionament del dispositiu en la xarxa de Modbus, és necessari un convertidor de UART a RS485, que és el canal de comunicació del Modbus. Així, s'ha seleccionat una ClickBoard de la marca Mikroelektronika com a convertidor. Tal com es mostra en la Figura 27.



Figura 27. Convertidor de UART a RS485.

#### 4.6. Maquinari del sistema de mesura de potència generada

Per dur a terme aquesta aplicació, s'ha escollit un microprocessador ESP32. La gran avantatge d'aquest dispositiu és que consisteix en un dispositiu de la sèrie "System on a chip" o sistema en un xip. Aquesta tecnologia es basa en un xip integrat per múltiples components que, en conjunt, conformen un sistema complet. Així doncs es pot formar un sistema encastat que pot obtenir les mesures de potència necessàries. El dispositiu en concret és el mòdul ESP32-WROOM-32D, i els seus diferents pins es poden observar en la Figura 28. Aquest dispositiu té un nucli de 32 bits, i pot funcionar a 240 MHz.

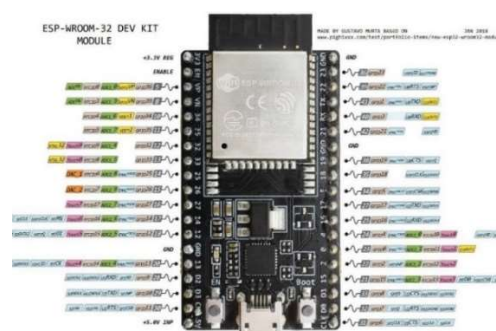


Figura 28. Assignació de pins del microprocessador ESP32-WROOM-32D.

Entre el microprocessador ESP32 i el PIC, hi ha d'haver quatre punts de connexió: el senyal de dades de l'I2C, el del rellotge, el senyal de retorn i el trigger per despertar la ESP 32, que prové d'un pin de sortida del PIC cap a una entrada digital de l'ESP32, que permet despertar

el sensor del deep-sleep. Com que el senyal del PIC és de cinc volts, i el de la ESP32 és de 3,3 V, pel senyal de despertar el microprocessador s'ha implementat un convertidor de nivell. En aquest dispositiu, en el convertidor d'I2C de canal únic a diferencial, el circuit de canal únic està alimentat a 3,3 V, i el canal diferencial està alimentat a cinc volts.

El sistema que permet adquirir les dades de potència generada gràcies a un sensor ACS758, que tracta les dades segons els criteris descrits anteriorment. A fi d'interconnectar el microcontrolador PIC, amb el sensor i amb el microprocessador s'ha dissenyat una placa de circuit imprès.

El sistema de mesura de potència, queda alimentat de la mateixa forma que el PIC, és a dir, des del mateix connector de l'USB, mitjançant un cable adaptador que es connecta directament de la sortida de la font de cinc volts. En la Figura 29, es visualitza com queda interconnectat el maquinari del sistema amb el PIC.

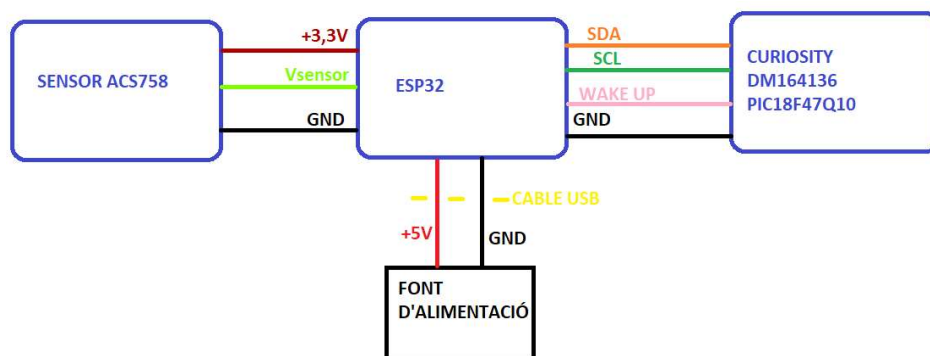


Figura 29. Diagrama de connexions del sistema de mesura de potència generada.

#### 4.7. PCB del sistema

La PCB està connectada amb la Curiosity amb un connector de tipus Sandwich. Com es pot observar en la imatge de la Figura 30, la placa Curiosity té dos jumpers que interconnecten els pins del microcontrolador amb l'exterior. Aquest són de tipus femella, mentre que els de la PCB són de tipus mascle. En la PCB els connectors tenen exactament les mateixes dimensions, i els jumpers estan connectats respectant les distàncies dels dos costats perquè puguin encaixar i interconnectar les dues parts. Addicionalment, la PCB dissenyada és de doble cara. I conté la part d'alimentació del microcontrolador, que el senyal li arriba provinent del concentrador de tensions del quadre elèctric, que està alimentat per la font d'alimentació, a través d'un cable que adapta els cinc volts i els zero volts amb un USB.



Figura 30. Jumpers femella de la placa Curiosity DM164136.

En el disseny de la PCB, els senyals d'entrada i sortida, s'han separat en diferents mòduls, a partir de regletes distribuïdes, per tal de facilitar la connexió amb els elements exteriors, i evitar perturbacions electromagnètiques. D'aquesta manera, es separen: les entrades digitals, les entrades analògiques, les sortides de PWM i les sortides digitals que activen relés.

En el disseny de la PCB, s'han aprofitat els jumpers de la placa Curiosity, destinats a la connexió de les ClickBoards. Tots els components ClickBoard són molt útils per treballar amb les plaques Curiosity, perquè poden anar connectades a elles directament. Aquests jumpers es poden veure en la Figura 30 que són els que estan a la dreta del PIC, i on s'han interconnectat els dos sensors de voltatge que mesuren les bateries. Les ClickBoards destinades a les comunicacions: el convertidor d'I2C de canal únic a diferencial, i el convertidor de UART a RS485, queden connectades a la PCB, fora de la placa Curiosity.

També cal tenir en compte que en la part de sota de la PCB hi ha el pla de massa. Aquesta superfície de coure permet que els retorns siguin el més directes possibles, i reduir els efectes d'antena que poden captar interferències i sorolls, que podrien originar mal funcionaments en el dispositiu de control. Les pistes d'alimentació i de senyals analògics s'han fet més gruixudes que les de senyal digital. S'ha defugit de crear circuits tancats que formin polígons que provoquin efectes d'antena, així com canvis de direcció amb angles de 90 graus. A més, es preveu recobrir la placa amb una pel·lícula d'esmail, cosa que disminuiria encara més el risc d'arc elèctric, i permetria desenvolupar la connexió de tipus Sandwich, sense perill de curtcircuits entre plaques. En al Figura 31, es mostra com es representa el PCB de connexions.



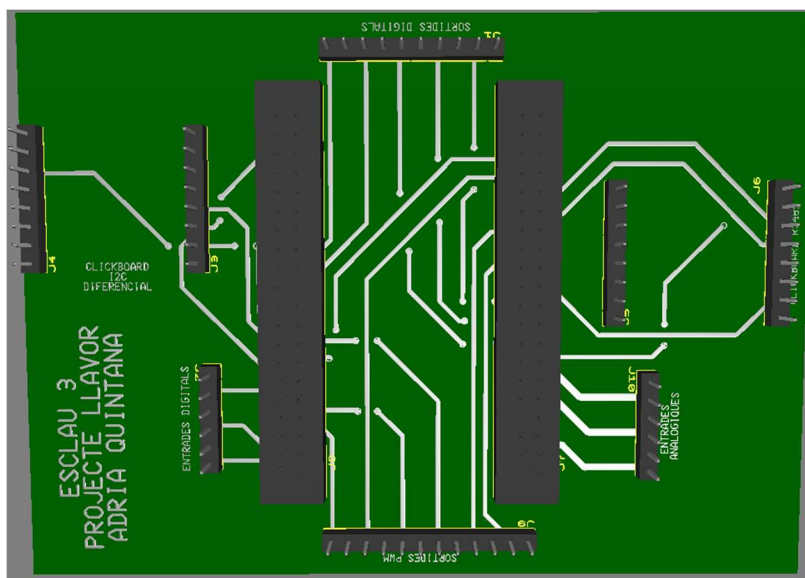


Figura 31. Visió en tres dimensions de la PCB.

## 5. PROGRAMARI DEL SISTEMA DE CONTROL

En aquest capítol, es detalla la metodologia i les consideracions que s'han seguit per programar el PIC, seguint el funcionament detallat en el capítol dos. El present capítol es divideix en onze apartats: en el primer apartat, s'introdueixen les eines emprades al llarg de la programació; en el segon apartat, es detalla com s'ha implementat el sistema multitasca, en el tercer, es comenten aspectes rellevants de la funció principal del programa, en el quart s'informa com funciona el mecanisme de control d'execució del codi, en el cinquè s'explica com s'ha programat l'algorisme de transferència de potència, en el sisè com s'ha programat el frenat elèctric, fent especial esment als perifèrics que s'utilitzen, en el setè i vuitè apartat s'explica com s'han programat les aturades d'emergència i d'error, respectivament; en el novè apartat es detalla com s'adquireixen les dades dels sensors; en el desè apartat fa esment de com s'ha programat el sistema de mesura de potència generada; i en l'últim apartat es descriu com funciona el mode de baix consum.

En aquest capítol, es comenten els punts més importants, per acompanyar les explicacions hi pot haver algun fragment de codi, però el codi sencer es pot consultar a l'Annex A.

### 5.1. Introducció

La programació del PIC s'ha realitzat a través del programa MPLabX IDE, i amb el compilador XC8. El llenguatge emprat és el C. Altrament, moltes tasques que es poden dur a terme en el PIC, sigui a través dels perifèrics o la configuració interna del sistema, s'han de configurar a través de l'escriptura de registres interns del microcontrolador. En la configuració del sistema principal, com l'assignació de pins, la definició de les interrupcions i la parametrització d'alguns perifèrics, exceptuant les comunicacions, els registres s'han escrit a través d'una eina que facilita la configuració i que incorpora l'entorn MPLabX, anomenada MCC.

### 5.2. Sistema multitasca

Per dur a terme les diferents accions de control dels algorismes comentats en el capítol dos, el sistema ha de processar tasques en paral·lel. D'aquesta manera, les principals operacions de control del sistema: que són les tasques realitzades en el mode de funcionament normal, com la transferència de potència i la comprovació que no hi ha generació per posar-se en mode sleep(), per poder treballar amb baix consum; i les deteccions d'error i d'emergència per aturar el sistema, es duen a terme en un llaç infinit de la funció del main(). La resta d'operacions

que són necessàries pel control de potència del sistema, i que condicionen el mode de treball de sistema de control de consum de potència, es realitzen a través de les diferents interrupcions. Les interrupcions són esdeveniments que requereixen atenció immediata del sistema operatiu, per exemple: el sistema es pot interrompre quan li arriba una dada en els sistemes de comunicacions, o a través d'un temporitzador. Donada l'ocasió, el processador suspèn temporalment la tasca actual, i executa un servei d'interrupció que l'atén. D'aquesta manera es poden atendre diferents tasques alhora, i seguir duent a terme el control del sistema. En la Figura 32, hi ha un organigrama que distingeix les diferents operacions del programa.

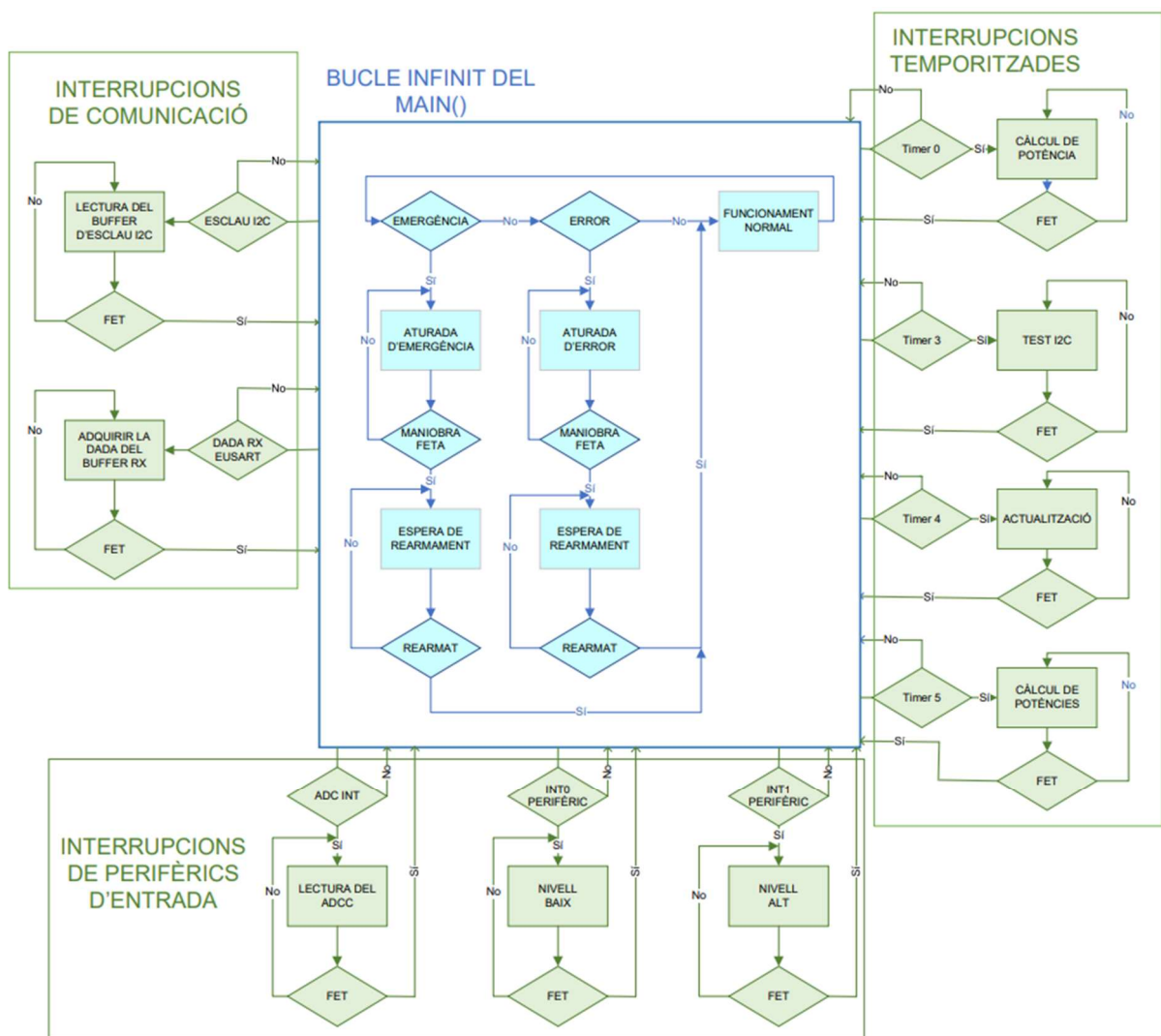


Figura 32. Organigrama del funcionament del programa.

Les interrupcions més prioritàries són les causades pels mòduls de comunicacions, sent la més prioritària la interrupció causada per la EUSART que controla el Modbus, i la segona és

la que provoca una recepció de dades al mòdul d'I2C treballant com a Esclau. Seguidament, les següents interrupcions més prioritàries són les que permeten conèixer l'estat del sistema, seguint l'ordre de més a menys prioritat: la interrupció causada per l'ADCC, que és la que permet adquirir dades, la interrupció del temporitzador TMR5, que és la que executa la comunicació amb el dispositiu que adquireix la potència generada, la interrupció del temporitzador TMR0 que permet calcular les potències, i la interrupció dels perifèrics que permet conèixer l'estat dels dipòsits. Finalment, hi ha la interrupció d'assajos del sistema, que es deixen com a última prioritat, perquè primer s'han d'haver adquirit les dades per avaluar si el sistema està amb error.

### 5.3. Funció principal del programa

La funció principal del codi, el main(), es divideix en dues parts: la primera part permet inicialitzar les diferents rutines que intervenen en el sistema, i la segona part consisteix en un llaç infinit, que executa les principals rutines del codi. Així doncs, en les primeres línies d'execució, el microcontrolador executa la funció de SYSTEM\_Initialize(). Aquesta funció escriu els registres de configuració dels pins d'entrades, de sortides, i dels associats als perifèrics que s'utilitzen en el PIC, que es mostren en la Taula 29; defineix el rellotge intern del microcontrolador, que en aquest cas, s'ha declarat a 8 MHz, per tal de sincronitzar les comunicacions i el sistema multitasca de manera correcte; habilita el Watchdog amb el temps configurat, i finalment, assigna els temporitzadors que s'empraran en el programa.

Nom	Funció	Pin
CLCIN0	Entrada física del CLC 1	RC6
CLCIN1	Entrada física del CLC 2	RC5
CLCIN2	Entrada física del CLC 3	RD2
CLCI0	Sortida de control de l'entrada del CLCIN0	RD3
CLCI1	Sortida de control de l'entrada del CLCIN1	RD4
CLCI2	Sortida de control de l'entrada del CLCIN2	RD5
CLC1OUT	Sortida de control de potència 1	RA6
CLC2OUT	Sortida de control de potència 2	RA7

Taula 29.1. Entrades i sortides del microcontrolador.

Nom	Funció	Pin
CLC3OUT	Sortida de control de potència 3	RB0
CLC4OUT	Sortida de control de potència 4	RB4
CLC5OUT	Sortida de control de potència 5	RC2
CLC7OUT	Sortida de control de potència 6	RB5
SDA1	Canal de dades 1	RC4
SCL1	Canal de rellotge 1	RC3
SDA2	Canal de dades 2	RB2
SCL2	Canal de rellotge 2	RB1
RX1	Rebre EUSART1	RC0
TX1	Transmetre EUSART1	RC1
RX2	Rebre EUSART2 (Reservat)	RD1
TX2	Transmetre EUSART1 (Reservat)	RD0
CANAL_AN0	Voltímetre	RA0
CANAL_AN1	Corrent del generador	RA1
CANAL_AN2	Corrent del regulador	RA2
CANAL_AN3	Corrent de l'inversor	RA3
channel_ANA4	Corrent de la bateria	RA4
R_T_RS485	Canviar del mode rebre a transmetre RS485	RA5
INT0	Nivell alt dipòsit	RB7
INT1	Nivell baix dipòsit	RB6
INT2	(Reservat)	RB3
K_GEN_RS	Relé RS	RD6
K_GEN_ST	Relé ST	RD7
DESPERTAR_ESP32	Despertar ESP32	RE0
K_BOMBA	Relé focus	RE1
K_FOCUS	Relé bomba	RE2

Taula 29.2. Entrades i sortides del microcontrolador.

Un cop duta a terme la funció `SYSTEM_Initialize()`, el sistema habilita les interrupcions globals i les interrupcions causades per perifèrics, a través de les funcions `INTERRUPT_GlobalInterruptEnable()`, i `INTERRUPT_PeripheralInterruptEnable()` que defineix la prioritat de les interrupcions del sistema. Seguidament, en cada interrupció se li inclou l'opció d'introduir-la al manipulador d'interrupcions.

Finalment, prèviament abans d'entrar al bucle infinit, s'analitza s'hi hi ha hagut un error de control d'execució del codi en algun moment que a reiniciat la funció principal.

#### 5.4. Mecanisme de control d'execució del codi

El mecanisme que permet detectar una error de control d'execució de codi es duu a terme a través del temporitzador del Watchdog. El Watchdog és un temporitzador intern que s'ha d'anar rearmant de manera constant, en cas de no rearmar-se i que el temporitzador es desbordi, significa que el sistema ha quedat encallat en un bucle infinit, i el sistema reinicialitza la funció del `main()`, sense esborrar les dades de les memòries Flash i RAM. Quan el temporitzador queda desbordat, hi ha dos bits dels registres de control d'aquest mecanisme de detecció d'error d'execució del codi, que es posen a nivell baix, que són: el setè bit menys significatiu del registre `STATUS`, que informa que el temporitzador s'ha desbordat, i el cinquè bit menys significatiu del registre `PCON0`, que indica que el Watchdog no s'ha rearmat dins del marge de temps establert. Quan es reinicia la funció `main()`, si aquests bits es troben a nivell baix, s'assigna a un valor cert la variable booleana `Error_WATCHDOG`, que el temporitzador Watchdog s'ha desbordat, i s'analitza si la variable booleana d'emergència estava a nivell alt, en el moment de desbordar-se el temporitzador. Si aquesta condició és certa, es manté a nivell alt el bit d'Emergència, i es posa a nivell baix el bit d'error, per evitar errors de coherència, i quan el sistema entri en bucle infinit, realitzarà les accions d'aturada d'emergència corresponents. Si la condició és falsa, el sistema posa a nivell alt el bit d'error, i en la primera iteració del llaç infinit el sistema realitzarà l'aturada d'error. Un cop realitzat el condicional per si s'ha produït un error de control d'execució del codi, el sistema habilita el temporitzador del Watchdog, escrivint sobre el registre de configuració del Watchdog.

```
if(!STATUSbits.TO&&!PCON0bits.RWDT){
    Error_WATCHDOG=true;
    if(Emergencia){
        Error=false;
        Emergencia=true;
    }else{
```

```
        Error=true;
    }}
    WDTCON0bits.SEN=1;
```

El temporitzador del Watchdog s'ha fixat en un temps de treball de 8,45 segons. El motiu és perquè aquest mecanisme de control d'execució del codi actua en totes les maniobres del sistema, així doncs, quan es troba en error i emergència, com que encara que es trobi en un error de control d'execució del codi, els pins de sortida dels perifèrics es mantenen com estaven prèviament a recaure en un llaç infinit, i el frenat elèctric estarà funcionant a 1.800 W. En cas que es doni un error d'aquest tipus, durant aquest temps, només en l'acció de frenada d'aquest frenat elèctric, la velocitat de la turbina s'haurà pogut reduir a un valor segur per poder curtcircuitar l'alternador.

## 5.5. Programació del control de potència consumida

L'algoritme de control de càrrega s'ha programat de la manera més entenedora possible, seguint la filosofia de programar amb funcions, que s'ha dut a terme al llarg del codi.

El sistema segueix l'organigrama de l'apartat cinc del capítol dos, i canvia d'estat a través d'una estructura if-then-else, que manté certa similitud al llenguatge estructurat que s'utilitza en la programació PLC. La programació de l'activació dels relés que connecten càrregues s'ha fet a través d'una estructura switch-case. En conseqüència, ha quedat una rutina similar a un menú de configuració, que seria l'elecció de l'estat, seguit per una altra rutina que canvia la sortida del sistema a través de l'opció seleccionada pel menú.

En el bucle infinit, s'ha insertat la funció anomenada Transferencia\_Consum(). Aquesta funció incorpora dins seu dues funcions. La primera que s'executa és la funció de canvi d'estat, anomenada Control\_Reles(), que segons la lectura dels sensors es canvia l'estat, i la segona és la que connecta els relés, anomenada Accionament\_Carregues().

```
void Transferencia_Consum(void) {
    Control_Reles();
    Accionament_Carregues();
}
```

Per l'accionament de càrregues, s'ha fet una llista d'enumeració en les variables globals, on s'afegeixen tots els possibles estats del sistema, incloent l'emergència i l'error que més

endavant es comentaran. I en funció de la lectura dels sensors, la variable d'enumeració anomenada "Estat", se li assigna un nou estat, i llavors, es commuten els relés pertinents a l'Estat que s'ha assignat. Altrament, en funció de l'Estat, s'assigna el valor de potència de referència de l'inversor. Aquesta variable determina la potència que hauria de tenir l'inversor amb les càrregues connectades, per tal de fer tests d'error, i determinar la potència demandada per part de les càrregues del sistema.

```
//Estats del sistema de control del consum:  
enum ESTATS{INICI,ESTAT1,ESTAT2,ESTAT3,ESTAT4,ESTAT5,ESTAT6,ESTAT7,ESTAT8,  
ESTAT9,ESTAT10,ESTAT11,ESTAT12,ESTAT13,ESTAT14,ESTAT15,EMERGENCIA};  
enum ESTATS Estat=INICI;
```

De la manera com s'ha comprovat en l'apartat cinc del capítol dos, corresponent a la taula de transicions, hi ha estats on es repeteixen les condicions de canvi d'estat i que porten als mateixos estats. Per aquestes condicions que són recurrents, també s'han agrupat en funcions per tal de simplificar el codi i fer-lo més entenedor. Un exemple és la funció de Control\_bateria\_descarregada() o Control\_bateria\_carregada().

Un cop executada la funció Transferencia\_Consum(), s'entra a la funció de Frenat\_Electric() que realitza les accions de protecció del sistema corresponents al frenat, seguidament es fa un rearmament del temporitzador del Watchdog, i finalment, s'estudia la condició d'sleep(), per treballar amb baix consum.

Cal destacar que aquesta rutina de control de consum de càrrega ve condicionada de les rutines d'emergència i d'error que dominen el sistema. És a dir, en cas de donar-se una emergència, el sistema de control de càrrega actua per assegurar la integritat del molí executant les maniobres corresponents. I en cas de trobar-se en error, el sistema queda parat fins que no s'arregli l'avaría.

### 5.5.1. Programació del control de la bomba

La connexió i desconexió de la bomba queda controlada per una comparació amb histèresis. Aquesta maniobra es duu a terme a través de les interrupcions per perifèrics. D'aquesta manera, l'estat dels dipòsits es pot monitoritzar de manera simultània a altres tasques. A través de la configuració física del sistema, s'ha assignat en dues entrades digitals a dues interrupcions per perifèrics, i cada vegada que hi ha un canvi de nivell lògic, es produeix una



interrupció. En la Figura 33, es mostra com s'han configurat les interrupcions per perifèrics, per monitoritzar els dipòsits i controlar la bomba.

External Interrupt	Edge Detect	
INT1	rising edge ▼	<b>Nivell alt dels dipòsits</b>
INT2	rising edge ▼	
INT0	falling edge ▼	<b>Nivell baix dels dipòsits</b>

Figura 33. Configuració del MCC de les interrupcions per canviar la variable per connectar la bomba.

El sensor de nivell baix s'ha connectat al pin RB6, i quan passa de nivell alt a nivell baix, s'ha configurat una interrupció al PIC, que dins d'aquesta rutina, es posa a nivell alt la variable de nivell dels dipòsits. El sensor de nivell alt s'ha connectat al pin RB7 i quan passa de nivell baix a nivell alt provoca una interrupció al PIC que posa a valor zero la variable booleana de nivell.

Quan la variable que indica que el nivell dels dipòsits permet connectar la bomba està en valor cert, si la potència produïda i l'estat de les bateries ho permeten, seguint l'algoritme de transferència de potència es connecta la bomba.

```
//NIVELL DELS DIPÒSITS
void Nivell_baix(void){
    Nivell_diposits=true;
}
void Nivell_alt(void){
    Nivell_diposits=false;
}
```

## 5.6. Programació del frenat elèctric

Les diferents rutines, que es duen a terme en el frenat elèctric, queden englobades en una funció que s'anomena `Frenat_Electric()`. Aquesta funció s'utilitza: en l'aturada d'error, en l'aturada d'emergència i en l'algoritme de control de transferència de potència. Dins de la funció de frenat hi ha dues funcions: la primera que s'executa és la funció del cas de frenat, anomenada `seleccio_cas_frenat()`; i ve seguida de la funció `maniobraFrenat()`.

En la selecció del cas de frenat, l'algoritme a partir d'una rutina de condicional, entra en funcionament en un dels tres modes de treball citats en l'apartat tres del capítol dos: el mode de protecció de la bateria, el mode de dissipar l'excedent de generació i el mode de frenar la turbina de cop en cas d'emergència; sumat al mode addicional de test del frenat per part del PLC. Immediatament, calcula la potència que s'ha de transferir en el frenat. En el codi queda plasmat el que s'ha plantejat en l'apartat tres del capítol dos, ja que, la primera condició que s'executa és analitzar si el sistema es troba en emergència o en error, a partir de la comprovació que el bit d'error o emergència estigui a nivell alt. En l'estructura de condicional, aquesta condició és la que domina els altres modes de treball. I quan s'hi entra es transfereix tota la potència al frenat.

El següent mode de treball, consisteix en dur a terme un test del frenat, escrivint sobre la variable de control del frenat. El mode de test, predomina sobre el segon i el tercer mode de treball del frenat.

```
}else if(!(Emergencia||Error)&&Test_frenat_PLC){

    //La potència del frenat depèn del valor escrit pel PLC
    Pfrenat_W=FRENAT_PLC;
```

En el mode de protecció de la bateria o mode quatre, s'hi entra quan la bateria no està carregada, i part de l'energia generada es transfereix a la bateria. Per aquest motiu, la condició és que el corrent de la bateria sigui inferior a zero. En cas que la bateria estigui carregada, el sistema treballa en el mode de derivació d'energia generada cap al frenat, per equilibrar la generació i la demanda, que és el mode tres.

En els modes tres i quatre, la potència que es transfereix a l'acumulador de potència del frenat prové d'una resta matemàtica. Quan el resultat de la resta és positiu, la potència s'afegeix, i quan és negatiu es treu. A més a més, en cada mode, com a mesura de prevenció d'errors de control, s'ha afegit un condicional, que limita el valor mínim de la variable de control del frenat a zero, per evitar tenir nombre negatius. En el cas del mode de treball tres, la diferència és la resta entre el límit de potència de càrrega de la bateria, i la potència de càrrega actual.

```
if(Intensitat_bateria>0){
    Pexcedent_bat=Pbateria_W-Plimit_bateria;
    Pfrenat_W=Pfrenat_W+Pexcedent_bat;
    if(Pfrenat_W<0)Pfrenat_W=0;
```

En el mode dos, la diferència és la potència generada menys la demanada. En la demanda s'hi considera: les pèrdues de potència, que es calcula a través del rendiment, la potència que es consumeix l'inversor, i la potència del frenat.

```

}else if(Intensitat_bateria==0&&VBateria>=Vbateria_carregada) {
Pexcedent_generacio=PotenciaGenerada-Pinversor_W-PFRENAT_MESURADA-
(PotenciaGenerada-PotenciaGenerada*rendiment/10);
if(Pfrenat_W<0) Pfrenat_W=0;

```

Finalment, quan les càrregues reben energia de les bateries, en cas que no es produeixi en una situació d'error o emergència, s'ha d'evitar que el frenat s'activi.

```

else if(Intensitat_bateria<0){ /*El frenat no ha d'actuar, l'energia de
* la bateria s'ha de destinar a alimentar càrregues.*/
Pfrenat_W=0; }

```

Un cop s'ha transferit potència al sistema d'acumulació de potència de frenat, es passa a la funció de maniobraFrenat(), on s'executa la rutina de connexió dels bancs de càrrega en funció del valor de l'acumulador. En tenir sis bancs de dissipació, hagués sigut ideal tenir sis senyals de PWM diferents, però el model de PIC escollit, només incorpora un senyal de PWM. Per solucionar aquest fet, s'ha emprat els perifèrics de CLC que incorpora el PIC. El CLC és una cel·la que incorpora diferents portes lògiques, que permeten configurar fins a vuit circuits lògics diferents, que poden ser controlats amb un total de vuit entrades lògiques. Aquests circuits treballen de manera independent a la CPU del PIC, però es poden interconnectar amb altres perifèrics. El model de PIC escollit permet parametritzar sis CLC, d'aquesta manera, amb un únic mòdul de PWM, que s'ha connectat en cada entrada dels sis CLC, aplicant lògica combinacional, s'han pogut tenir sis sortides de PWM diferents. En la Taula 30, es mostra la taula de la veritat que controla les sortides de PWM de commutació dels bancs.

CLCIN2	CLCIN1	CLCIN0	BANC1	BANC2	BANC3	BANC4	BANC5	BANC6
0	0	0	0	0	0	0	0	0
0	0	1	PWM	0	0	0	0	0
0	1	0	1	PWM	0	0	0	0
0	1	1	1	1	PWM	0	0	0
1	0	0	1	1	1	PWM	0	0
1	0	1	1	1	1	1	PWM	0
1	1	0	1	1	1	1	1	PWM

Taula 30. Taula de la veritat del control dels CLCs.

Una de les limitacions del CLC és que en funcionar de manera independent a la CPU, les seves entrades de control no poden ser controlades per la CPU. Aquest fet s'ha solucionat a través de les connexions de la PCB, on es connecta una sortida digital del PIC amb una entrada de control del CLC. En la Taula 31, es mostra com queden connectades les entrades de control del CLC amb les sortides digitals del PIC.

Pin de la sortida digital	Nom de la variable de la sortida digital	Pin de l'entrada de control del CLC	Entrada de control del CLC
CLCI2_PORT	RD5	CLCIN2	RD2
CLCI1_PORT	RD4	CLCIN1	RC5
CLCI0_PORT	RD3	CLCIN0	RC6

Taula 31. Descripció de les interconnexions del CLC.

El CLC es configura per registres de control del PIC, però per simplificar la tasca s'ha emprat l'eina gràfica de l'MCC. El banc de càrrega u, que funciona quan s'han de dissipar entre zero i 300 W, funciona com la Figura 34.

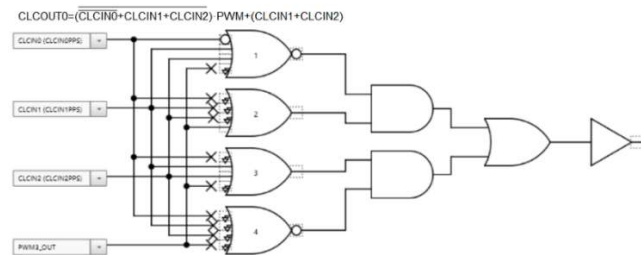


Figura 34. CLC de control del banc de càrrega u.

El banc de càrrega dos, que funciona quan s'han de dissipar entre 300 i 600 W, funciona com la Figura 35.

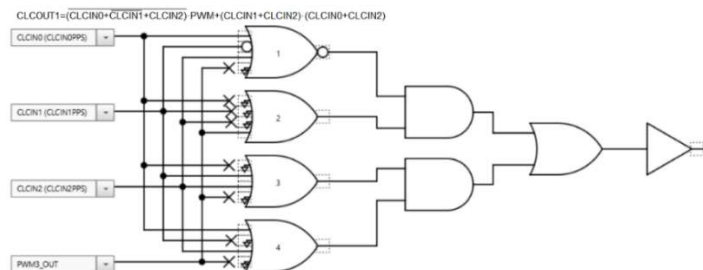


Figura 35. CLC de control del banc de càrrega dos.

El banc de càrrega tres, que funciona quan s'han de dissipar entre 600 i 900 W, funciona com la Figura 36.

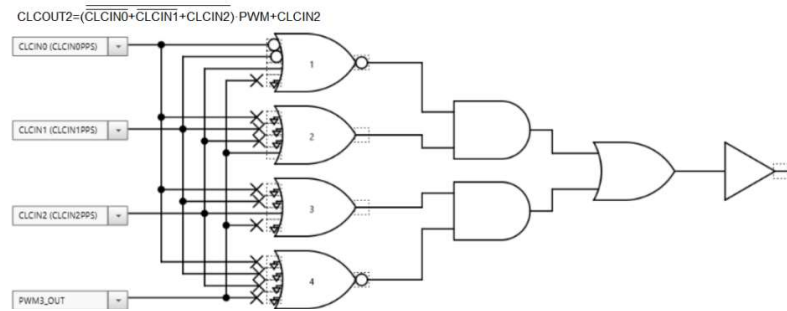


Figura 36. CLC de control del banc de càrrega tres.

El banc de càrrega quatre, que funciona quan s'han de dissipar entre 900 i 1.200 W, funciona com la Figura 37.

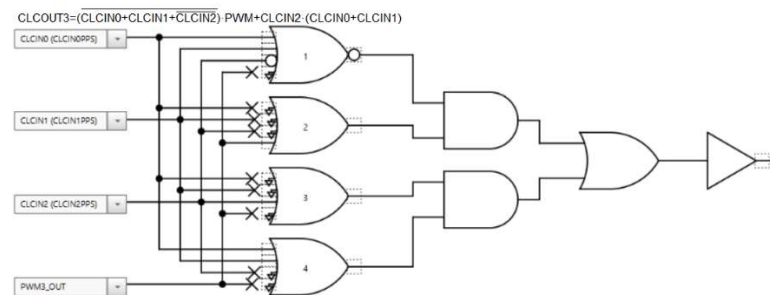


Figura 37. CLC de control del banc de càrrega quatre.

El banc de càrrega cinc, que funciona quan s'han de dissipar entre 1.200 i 1.500 W, funciona com la Figura 38.

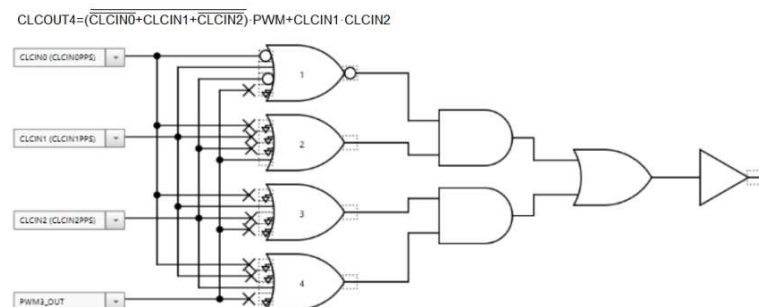


Figura 38. CLC de control del banc de càrrega cinc.

El banc de càrrega sis, que funciona quan s'han de dissipar entre 1.500 i 1.800 W, funciona com la Figura 39.

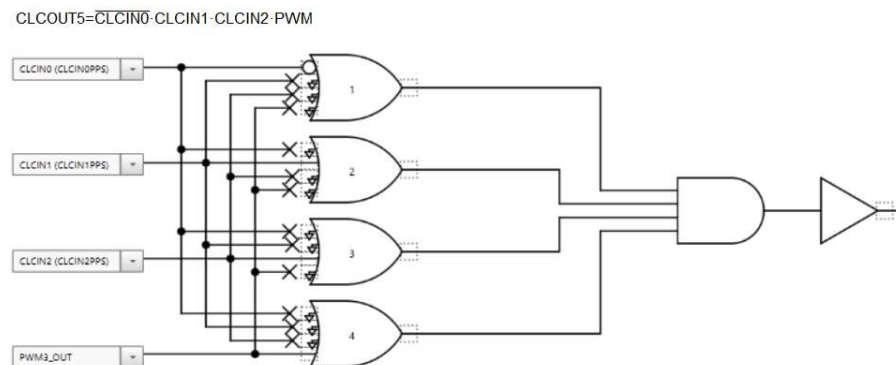


Figura 39. CLC de control del banc de càrrega sis.

El mòdul PWM s'ha configurat amb el temporitzador sis. En aquest temporitzador no se li pot associar cap interrupció, ja que afectaria al funcionament del PWM, perquè podria provocar parades del senyal. Aquest rellotge s'ha configurat de tal manera que el senyal de PWM sigui de 16,67 kHz, entrant dins del marge de freqüència dels commutadors MOSFET. A la Figura 40, es mostra com el temporitzador queda configurat.

Timer Clock	
Clock Source	FOSC/4 <input type="checkbox"/> Enable Clock Sync
Clock Frequency	32.768 kHz
Polarity	Rising Edge
Prescaler	1:1 <input type="checkbox"/> Enable Prescaler O/P Sync
Postscaler	1:1
Timer Period	
Timer Period	1 us ≤ 60 us ≤ 256 us
Actual Period	60 us (Period calculated via Timer Period)

Figura 40. Configuració del temporitzador sis.

Les sortides digitals, que connecten els senyals de control del CLC, es commuten a partir d'una acció de condicional. En funció de la potència que s'hagi calculat en la rutina anterior, es commuten els senyals de control segons la Taula 30. A part de commutar el control del CLC, també es calcula el cicle de treball del PWM. A continuació, es mostra un fragment de la funció de la maniobraFrenat(), per tal de clarificar com funciona aquesta acció.

```
if(Pfrenat_W>0 & Pfrenat_W<=300){ //Llindar de potències
    CLCI2_PORT=0; //Control de la taula de la veritat
```

```

CLCI1_PORT=0;
CLCI0_PORT=1;
dutyPWM=Calcul_duty(Pfrenat_W); //Càlcul del cicle de treball.
PWM3_LoadDutyValue(dutyPWM); } //Control del PWM.

```

Pel càlcul del cicle de treball del PWM, s'utilitza l'Equació 9, d'on s'obté el valor del cicle de treball de cada banc en un marge de potència de 300 W.

$$DutyPWM = \frac{1}{V_{inversor}} \cdot \sqrt{P_{banc} \cdot R_{frenat}} \quad (\text{Eq.9})$$

Cal considerar que el resultat de l'Equació 9 és una variable en coma flotant, que té un valor entre zero i u, i cal expressar-lo mitjançant una variable de 10 bits, en un valor que va de 0 a 1023.

```

uint16_t Calcul_duty(int potencia_banc){
    uint16_t Dcycle;
    float duty;
    //Aplica la fórmula de càlcul del duty cycle.
    duty=((float)sqrt((potencia_banc)*R_frenat)/24);
    duty=duty*1024;
    return Dcycle=(uint16_t)(duty); } //Es passa el valor en enter en el programa.

```

## 5.7. Programació de la maniobra d'emergència

El sistema de control de potència atura la generació cada vegada que detecta una emergència.

Com es pot observar en el codi adjuntat a sota, quan el booleà d'emergència està a nivell alt, el sistema passa en l'Etapa 1 d'emergència, i es realitzen totes les accions associades: el booleà d'avís del PLC es posa a nivell alt, perquè en la lectura d'estat del sistema que fa el PLC en el PIC, percebi que el sistema està en emergència; seguidament, crida la funció del frenat, perquè es posi en mode d'emergència i connecti el sis bancs, de la manera com s'ha comentat en l'apartat anterior. També es mostra com s'entra en la funció MANIOBRA\_EMERGENCIA(), on es duen a terme la resta d'etapes. A més a més, el booleà que posa el sistema en Error s'assigna a zero, perquè l'emergència predomina sobre l'error.

```

if(Emergencia){
    /* _____EMERGENCIA_ESTAT_1_____ */
    //Funció d'emergència:
    AvisEmergenciaPLC=true; //Avisa el PLC que està en emergència
    Error=false; //QUAN ES DETECTA UNA EMERGÈNCIA L'ERROR PASSA A SER ZERO.
    Frenat_Electric(); //Es posa el frenat al màxim.
    /* _____ */
    MANIOBRA_EMERGENCIA(); //Es fa la maniobra d'emergència
}

```

Dins de la funció MANIOBRA\_EMERGENCIA(), per tal de saber si prèviament el sistema estava en aturada d'emergència i hi ha hagut una fallada de comunicacions que ha provocat un llaç infinit, s'avalua el booleà que informa d'un desbordament de Watchdog. En cas de no haver hagut aquesta fallada, significat que és la primera vegada que s'entra en l'aturada d'emergència, el sistema entra en l'etapa dos, on a través de la funció Comunicar\_Esclau1(), es comunica al sistema de control de l'angle d'inclinació de les pales, a través de l'I2C, que s'ha connectat el frenat elèctric al màxim. Seguidament, es rearma el Watchdog, i s'entra a la maniobra de rebre la comanda que s'han mogut les pales a l'angle d'inclinació màxim, per part de dispositiu de control de l'angle d'orientació de les pales, que es troba dins de la funció Rebre\_Angle\_Maxim(). Dins d'aquest codi, es realitza l'espera de l'etapa dos. Primer de tot, s'inicialitza el temporitzador dos, amb la funció TMR2\_StartTimer(), que escriu els registre pertinents del PIC. Seguidament, s'espera a rebre la comanda de l'angle d'inclinació de les pales al màxim, que s'ha de rebre a través de la interrupció del PIC de lectura de l'I2C, que es comenta en el capítol sis. Dins d'aquesta rutina d'interrupció de l'I2C, quan es rep la comanda, es posa a nivell alt un booleà que informa que s'ha rebut la comanda de l'angle d'orientació de les pales al màxim. Aquesta espera per evitar causar errors d'execució de codi deficients, funciona amb un màxim de dos segons, que venen determinats pel temporitzador dos. En cas d'haver rebut el missatge de l'angle d'inclinació de les pales al màxim, es para el temporitzador, i seguidament es reinicialitza el temporitzador. En cas de no rebre l'angle durant dos segons, es surt del bucle a través d'un break, i es reinicialitza el temporitzador.

```

void Rebre_Angle_Maxim(void){
    TMR2_StartTimer();//S'inicia el temporitzador d'espera per l'angle màxim.
    while(!Rebut_Angle_Maxim){//Espera a rebre el missatge
        if(TMR2_ReadTimer()==0xFF){//El temporitzador ha arribat al seu màxim.
            TMR2_Stop();//Es para el temporitzador.
            break;//Es surt del bucle d'espera. } }
    if(Rebut_Angle_Maxim){//Rebut angle màxim
        TMR2_Stop();//Es para el temporitzador. }
    TMR2_WriteTimer(0x00);//Es tornar a posar a zero el timer }

```



Per dur a terme el llaç que es produeix entre les etapes dos i quatre, quan no es rep l'angle de les pales el màxim, s'ha emprat el mecanisme de programació del goto amb una etiqueta, tal com es mostra en el codi adjuntat. Si el comptador arriba a un màxim de tres voltes, s'analitza la potència que genera el generador i es curtcircuita. En cas que la potència generada sigui superior a la que suporten els relés s'indica que s'ha d'examinar el sistema.

```

/* _____EMERGENCIA_ESTAT_4_____ */
  Espera_emergencia++;
/* _____ */
  if(Espera_emergencia==3){ /*Passades les tres vegades es comprova que la
potència generada sigui zero.*/
    if(PotenciaGenerada<Potencia_limit_reles){
/* _____EMERGENCIA_ESTAT_3_____ */
      CurtcircuitarGenerador();//Els relés i el generador aguanten el curtcircuit.
      REVISIO_SISTEMA=false; //Sistema parat i amb emergència.
/* _____ */
    }else{
/* _____EMERGENCIA_ESTAT_5_____ */
      CurtcircuitarGenerador();//Els relés i el generador poden haver quedat danyats.
      REVISIO_SISTEMA=true; //Sistema parat i amb emergència i pendent de revisió.
      POSSIBLES_DANYS_RELES_GENERADOR=true;
/* _____ */
    }}else{
      goto NOCOMUNICACIO; //Es torna a enviar el missatge. Fins a tres vegades.}}
  MANIOBRA_FETA=true; //Serveix per indicar que s'ha fet la maniobra, al PLC.
}else{ //En aquest punt, el frenat està al màxim i s'ha curtcircuitat el generador.
  Emergencia=false;
  Espera_emergencia=0;
  Estat=EMERGENCIA;}

```

Si s'ha rebut el missatge de l'angle de les pales al màxim, el sistema curtcircuita el generador, i com que s'ha produït l'aturada d'emergència amb èxit, el bit de revisió del sistema es deixa a nivell baix, perquè no cal revisar el sistema de control de potència, es borra el booleà que s'ha rebut l'angle al màxim i el que posa el sistema amb emergència, i la variable d'enumeració d'estat s'assigna l'estat d'Emergència, perquè en el Switch-case de control de potència, no es dugui a terme cap de les accions, fins que el sistema no estigui rearmat.

```

if(Rebut_Angle_Maxim){
/* _____EMERGENCIA_ESTAT_3_____ */
  CurtcircuitarGenerador();
  REVISIO_SISTEMA=false;
  Rebut_Angle_Maxim=false;//Es borra el bit de control de l'angle màxim

```

Si el booleà que indica que hi ha hagut un error de desbordament del Watchdog es troba nivell alt, el sistema ha de curtcircuitar el generador per parar la producció de manera total. Un error de desbordament en aquesta maniobra, significa que hi ha hagut un error de comunicacions mentre prèviament estava aturant el sistema per una emergència. D'aquesta manera, com que el frenat ja estava connectat a 1.800 W, cal avaluar la generació abans de curtcircuitar el generador, per si s'han de posar a nivell alt el bits de revisió, i s'ha de curtcircuitar el generador.

Un cop realitzades les accions de frenada d'emergència, es borra el bit que posa el sistema en emergència, es reinicia el temporitzador, i la variable d'Estat se li assigna l'Emergència. Altrament, es desactiva el comptador del Watchdog, a través de l'escriptura del bit menys significatiu del registre de control WDTCON0, per evitar que constantment el programa es vagi reiniciant.

En la següent iteració del llaç de la funció main(), el sistema ha finalitzat la maniobra d'aturada, i es passa en la rutina de transferència de potència, però com que espera a ser rearmat, el switch-case que controla els relés es troba en el cas d'estat en emergència, posant els relés de control de la bomba i dels focus a zero. I el sistema no realitza cap acció, fins que no es rearma.

La condició de rearmament està inclosa dins de la funció de Control\_Reles(), que es troba dins de la funció de Transferencia\_Consum(). Per tant, si no es rearma el sistema, la variable d'enumeració d'estat seguirà en emergència. Així doncs, per reiniciar el sistema, cal complir la condició de rearmament. Quan es dona la condició, s'esborren tots els bits de control de l'aturada d'emergència, i es rehabilita el Watchdog. Seguidament, s'avalua en quin estat es troba el sistema, per posar-se a produir de nou.

```

if (Estat==EMERGENCIA&&REARMAMENT&&!REVISIO_SISTEMA
&&!POSSIBLES_DANYS_RELES_GENERADOR){ //El rearmament el fa el PLC.
    REARMAMENT=false;//la varibale de reamemaent es torna a posar a false.
    AvisEmergenciaPLC=false;
    WDTCON0bits.SEN=1;//Es torna a habilitar el WDT.
    Error_WATCHDOG=false;
    MANIOBRA_FETA=false;
    if (VBateria>Vbaix){
        Estat=ESTAT14; //Si la bateria està carregada s'envia a l'estat 14.
    }else{
        Estat=INICI; //Si la bateria està descarregada s'envia a inici.
    }
}

```

## **5.8. Programació de la maniobra d'error**

En la maniobra d'error hi intervenen dues parts diferenciades: la primera part són els autodiagnòstics que realitza el microcontrolador, que permeten determinar si s'ha donat un error en algun dels dispositius de l'aerogenerador, i la segona és l'aturada d'error, que realitza les diferents maniobres per aturar la producció.

En aquest apartat, es diferencien dos subapartats: en el primer s'expliquen els diferents diagnòstics que determinen si s'ha produït un error en el sistema elèctric, i en el segon, l'aturada d'error.

### **5.8.1. Diagnòstics del sistema**

Els assajos només es realitzen quan el sistema està funcionant en condicions normals d'operació, és a dir, que no es troba ni en una aturada d'emergència, ni d'error. Els diagnòstics poden portar el sistema a aturar-se per un error, però durant l'aturada d'error ja no es realitzen més assajos.

En el sistema es duen a terme dos assajos: un test que comprova la validesa de les mesures de potència, que avalua si el sistema està funcionant de manera correcta, i un test que comprova que el sistema de comunicacions de l'I2C multimestre funciona. Aquest test s'explica en el capítol pertinent a les comunicacions.

L'assaig de les mesures del sistema consisteix en comparar els valors de potència calculats a partir de les mesures obtingudes pels sensors, amb els respectius valors de referència. Aquesta comparació es realitza de manera regular a partir d'una interrupció temporitzada.

Els valors que es comparen en els assajos són: la potència de l'inversor, la potència del frenat, i la potència de la bateria. La comparació entre els valors reals i els valors de referència, es duu a terme a través d'un marge d'error que s'ha inicialitzat a 12 W, i que pot ser canviat pel PLC.

Quan hi ha la interrupció temporitzada que calcula les potències a partir dels valors obtinguts pels sensors, es posa a nivell alt una variable booleana, anomenada Test\_sensors, perquè en cas de trobar-se en condicions fora d'aturades d'error i emergència, executi els assajos en el

bucle de la funció main(). Així doncs, en aquesta part del codi, el primer que es realitza és el càlcul de la potència de la bateria, a través de la funció calcul\_Potencia\_teorica\_bateria(), que considera tots els escenaris en que es pot trobar la bateria.

El primer cas que es considera és que la bateria està descarregada i rep energia. Aquest fet es dona perquè la tensió està per sota del valor de voltatge de la bateria carregada, la potència generada és superior a zero i el corrent és superior a zero. Donada la circumstància, la potència destinada a la bateria és l'excedent d'energia, considerant el balanç de la potència generada, menys la consumida, que és la destinada a l'inversor i al frenat, i les pèrdues.

```
if(PotenciaGenerada>0&&(VBateria<=Vbateria_carregada||Intensitat_bateria>0)){
    //La bateria està descarregada i hi ha generació:
    Pbateria_teorica=PotenciaGenerada-Pinversor_teorica-Pfrenat_W
        -(PotenciaGenerada-PotenciaGenerada*rendiment/10);    }
```

El següent cas de treball de la bateria és que hi ha generació, la bateria està carregada i aporta energia per complementar la potència generada per alimentar les càrregues. En aquest cas, la potència de la bateria teòrica, depèn de si la diferència entre la potència de l'inversor i la potència generada menys les pèrdues és superior a zero. Si la diferència és superior a zero, vol dir que l'inversor consumeix més potència que la que genera la bateria, i significa que la bateria aporta potència, ergo, la diferència calculada és la potència que aporta la bateria.

```
if(Pinversor_teorica-(PotenciaGenerada*rendiment/10)>0)
    Pbateria_teorica=(Pinversor_teorica- PotenciaGenerada*rendiment/10);
else Pbateria_teorica=0;
```

El següent escenari que es pot trobar la bateria, és que no hi ha generació i les càrregues queden alimentades totes per la bateria. D'aquesta manera, la potència de l'inversor és la mateixa que la de les bateries. Finalment, es considera que no hi ha generació i la tensió de la bateria és inferior o igual al nivell de voltatge de bateria descarregada, i no hi ha aportació d'energia.

La potència de l'inversor de referència s'assigna en el switch-cas que controla els relés, que està dins de la funció Accionament\_Carregues(). En aquesta situació, en funció dels relés que hi hagin commutatats, la potència de l'inversor de referència és igual a la potència lliurar mínima, que ha portat al sistema a funcionar a un estat de transferència de potència determinat. En el codi adjuntat es descriu el funcionament. En aquesta funció també s'assigna

el valor de la variable booleana `ESTAT_AMB_GENERACIO`, que determina que el sistema hauria d'estar produint energia.

```
case ESTAT2:
    K_BOMBA_PORT=1;
    K_FOCUS_PORT=1;
    K_GEN_RS_PORT=0;
    K_GEN_ST_PORT=0;
    ESTAT_AMB_GENERACIO=true;
    Pinversor_teorica=SB_Sllums;
    break;
```

Finalment, la potència del frenat compara el valor mesurat amb el valor de referència que equival al sistema d'acumulació de potència del frenat, que es calcula en la funció `seleccio_cas_frenat()`. Aquesta comparació es realitzarà en el mateix punt del codi on es comparen la resta de les variables.

El següent pas que realitza el sistema consisteix en calcular l'error entre variables, que és la resta entre el valor de referència i el valor teòric. I seguidament, es compara aquest error amb el marge d'error del sistema. Aquesta comparació es realitza a través de la funció `comprovar_error_sensors()`, que retorna un booleà, que s'assigna a cada una de les variables booleanes que determinen d'on prové l'error del sistema. En aquesta funció si li passen dos paràmetres: el marge d'error, i l'error de la variable que s'estigui analitzant, i llavors en la rutina interna, les compara amb valor absolut. En cas de superar el marge d'error es retorna un booleà assignat en valor cert, i si no es supera, es retorna un valor fals.

```
bool comprovar_error_sensors(int error, int marge){ //L'error està entre -12 i 12 W
    if(error<marge&&error>(marge*-1)) return false;
    else return true;}

```

Seguidament, es comprova que no hi hagi un error de coherència, és a dir, que el sistema detecti una potència generada equivalent a zero, i que es mesurin potències a l'inversor o al frenat superiors a zero, en una estat on hi ha generació en el sistema. També s'avalua que la tensió del sistema bateries estigui per sota de la meitat.

```
bool comprovar_coherencia(unsigned int PGEN, unsigned int VBAT,
    unsigned int PFre, unsigned int Pinv, bool hi_ha_gen){
    if((PGEN==0&&(PFre>0||Pinv>0))&&hi_ha_gen) return true;
    else if(VBAT<13) return true;
```

```
return false; }
```

Un cop comparats els valors dels sensors amb les variables de referència, s'analitza si algun dels booleans que informen d'algun error està a nivell alt. Si només un dels valors s'hi troba, el sistema es posa en estat d'error. I s'avisar de l'error als altres dispositius de la turbina.

```
if(ERROR_Inversor||ERROR_Bateria||ERROR_Bateria||ERROR_coherencia){  
Error=true;  
AVIS_ERROR(); }
```

El sistema també es pot posar en error quan s'ha reinicialitzat a causa d'un desbordament de Watchdog, tal com s'ha esmentat anteriorment. Finalment, l'altre mecanisme que posa el sistema en aturada d'error és perquè un dels altres dispositius de la turbina ha detectat un error i ho capta a través de la xarxa multimestre d'I2C.

### 5.8.2. Aturada d'error

L'avaluació de si s'ha produït un error es realitza dins del bucle de la funció main(), quan no es produeix una emergència, a través d'un condicional que examina la variable booleana Error.

Si la condició d'error és certa, el sistema posa en valor cert la variable booleana AvisErrorPLC, per avisar al PLC. L'error pot ser percebut si hi ha generació o si no n'hi ha, per aquest motiu, per connectar el frenat s'avalua la variable ESTAT\_AMB\_GENERACIO. Si està en valor cert, es connecta el frenat, i si no, significa que no ho està, i per evitar perdre energia de la bateria en el frenat, no es connecta. Seguidament, el sistema realitza les diferents etapes en la maniobra d'aturada per error, en la rutina de la funció MANIOBRA\_ERROR(). Dins d'aquesta funció, l'aturada es diferencia segons si s'ha donat un error de desbordament del Watchdog, o per la resta d'errors. Si l'error és de control d'execució del codi, el sistema inhabilita el Watchdog, i analitza si es troba en un estat de generació o no. Si no està amb generació, es curtcircuita el generador, i roman a l'espera de ser rearmat. En cas d'estar en generació, el sistema entra en un bucle d'on no en surt fins que el comptador de Watchdog adquireix un valor de deu. Aquest comptador s'incrementa amb la interrupció temporitzada anomenada TMR4\_ACTUALITZACIO(), que es produeix cada 400 mil·lisegons. Per tant, en arribar el comptador a deu, significa que realitza una espera de quatre segons, perquè el frenat elèctric redueixi la velocitat de la turbina, i pugui curtcircuitar l'alternador. Seguidament, es posen els bits de revisió del sistema a nivell alt, i el sistema roman a l'espera de ser rearmat.

```

if(Error_WATCHDOG) {
    WDTCON0bits.SEN=0; //Borra el Watchdog
    while(Comptador_watchdog<10&&ESTAT_AMB_GENERACIO);
    CurtcircuitarGenerador(); //Es curtcircuita el generador.
    REVISIO_SISTEMA=true;//Cal revisar el sistema.
    POSSIBLES_DANYS_RELES_GENERADOR=true;//No es té certesa si es llegeix bé.
    Estat=ERROR; }

```

Si l'error no és d'execució deficient de codi, el sistema comunica al dispositiu de control de l'angle d'inclinació, que el frenat està al màxim, encara que no hi hagi generació, perquè mogui les pales per evitar posar-se en generació. Seguidament, es rearma el Watchdog, per comprovar que el codi funciona correctament, i s'entra a un bucle que no en surt fins que no es rep l'angle al màxim. La comanda de l'angle al màxim es controla a través del booleà I2C\_ESCLAU\_MISSATGE i la variable lectural2C. Per tant, si no es rep el missatge correctament, es torna a comunicar que el frenat està al màxim en un màxim de tres ocasions. D'aquesta manera, si es reben tres dades errònies, vol dir que hi ha un error en el canal de comunicació de l'I2C, i es força el sistema en un llaç infinit, perquè es tracti l'error com si fos de Watchdog, i freni el sistema a través d'un altre procediment. Si directament no es rep cap dada per l'I2C, el sistema haurà quedat encallat, significat que també hi ha un error de comunicacions, es produeix un llaç infinit i es força el Watchdog.

```

while(!Rebut_Angle_Maxim){
if(I2C_ESCLAU_MISSATGE&&lecturaI2C!=0x1F){ //Dada rebuda incorrecta.
    I2C_ESCLAU_MISSATGE=false;
    Comunicar_Esclau();
    CLRWDT();
    Intents_Comunicacio++;
    if(Intents_Comunicacio==3){
        Intents_Comunicacio=0;
        while(1); }}}

```

Si el sistema surt d'aquest llaç, significa que ha actuat el frenat elèctric i el frenat per inclinació de l'angle de les pales, per tant, el frenat de la generació ha sigut òptim, només es posa a nivell alt el bit de REVISIO\_SISTEMA, i s'inhabilita el Watchdog perquè el sistema roman a l'Espera de ser rearmat.

Dins de la funció Control\_Reles(), l'acció d'error predomina sobre altres estats de commutació del sistema, exceptuant l'emergència. D'aquesta manera, quan la variable d'enumeració d'Estat està en error, cal aplicar la condició de rearmament per posar en funcionament el

sistema de nou. Quan es rearma el sistema, reinicialitza totes les variables de control de l'error, siguin les que informen dels errors en els sensors i els errors de comunicació, es restableix el Watchdog, i en funció de l'estat de càrrega de la bateria, s'envia el sistema a l'Estat d'inici o a l'Estat 14, que es comenta en l'apartat cinc del capítol dos.

## 5.9. Adquisició de dades

En el present apartat es detalla el mètode d'adquisició de dades del sistema de control del consum de potència. Exceptuant la lectura de la potència generada, que s'adquireix pel sistema de mesura amb una lectura a través del bus I2C, la resta de magnituds elèctriques es calculen directament del convertidor analògic-digital.

### 5.9.1. Lectura de la potència generada

En el cas de la lectura de la potència generada, el mètode de control de l'adquisició de les dades es duu a terme a través d'una interrupció temporitzada. En el fragment de codi corresponent a la rutina de la interrupció, el PIC es comunica amb el dispositiu esclau que calcula la potència generada a través d'una petició de lectura d'I2C. En aquest cas es reben dos bytes, rebent primer els bits més significatius, i després els menys significatius. El temporitzador realitza la interrupció cada 200 mil·lisegons. Aquest temps permet captar variacions de generació, adquirint valors estables pràcticament instantanis, arribant a captar canvis de condicions de vent que impliquen fluctuacions de generació, o si hi ha hagut algun error.

```
void TMR0_AdquisicioPotencia(void) {  
  
    I2C_Master_Start();//Start condition  
    I2C_Master_Write(0x31);//Adressa de 7 bits 0x18 + Read  
    uint8_t Lectura2 =I2C_Master_Read(1); //Read + Ack a 1 per rebre una dada  
    uint8_t Lectural =I2C_Master_Read(0); //Read + Ack  
    I2C_Master_Stop();//Stop condition  
    PotenciaGenerada=(unsigned int)((Lectura2<<8)|Lectural);  
}
```



### 5.9.2. Lectura dels sensors connectats al convertidor analògic digital

El convertidor analògic-digital del PIC, es troba dins del grup dels anomenats convertidors analògic-digitals d'aproximacions successives, que el seu funcionament es mostra en la Figura 41.

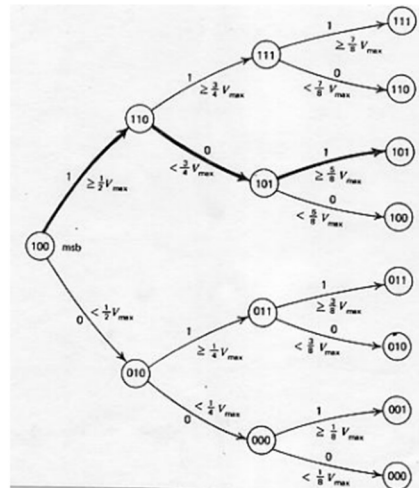


Figura 41. Algorisme d'adquisició del codi.

Un cop finalitza la conversió del codi, el convertidor analògic-digital emmagatzema el resultat de deu bits en dos registres de vuit bits, on els bits més significats es guardaren en un registre anomenat ADRESH i els menys significatius en un registre ADRESL. En la Figura 42 queda descrit de forma esquemàtica el hardware d'aquest convertidor.

El multiplexor es pot controlar a través del hardware, escrivint en un registre d'un byte anomenat ADPCH, un número que va del zero al 34. D'aquesta manera per programa es pot multiplexar cap al circuit de mostreig i retenció, l'entrada que es vol llegir.

En aquest convertidor, també se li pot configurar el senyal de rellotge. En aquest cas, el valor del rellotge és configurable a través del registre ADCS, on es pot seleccionar el rellotge de referència per defecte FRC o un oscil·lador perifèric FOSC amb un dividend postescalar.

El segon mòdul que forma aquest conjunt és un sistema computacional. Aquest mòdul permet després de la conversió del codi digital, realitzar un càlcul posterior. La funció de càlcul es pot seleccionar a través del mode d'operació ADC del registre de control ADCON2. Els modes de computació que permet aquest perifèric són: el mode basic, el mode d'acumulador, el mode

mitjana, el mode mitjana explosiva i el mode de filtre pas baix. En el sistema que s'està dissenyant s'ha escollit que el mòdul computacional treballi en el mode de mitjana de ràfega. En el mode de mitjana de ràfega o "Burst Average Mode", el mòdul computacional comença a adquirir mostres a partir d'un flanc de pujada extern al mòdul. Cada valor obtingut es suma en un registre acumulador de setze bits anomenat ADACC. I no fa la mitjana fins que no s'hagin adquirit el nombre de mostres configurades en el registre ADRPT.

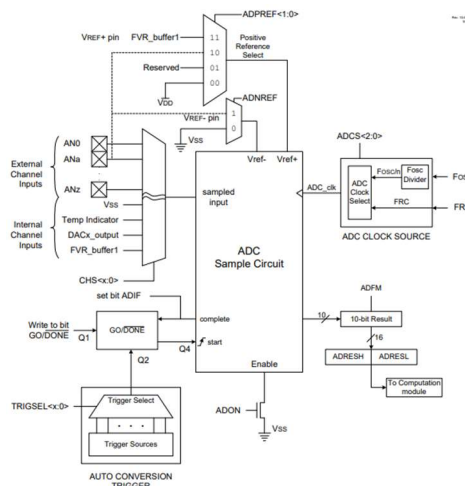


Figura 42. Esquema del mòdul convertidor analògic-digital.

Els senyals d'intensitat elèctrica, que es llegeixen en el convertidor analògic-digital, presenten un cert arrissat com el de la Figura 43, a causa que provenen de circuits electrònics de potència que regulen el senyal a partir de trossets de PWM. Per tant, per obtenir un valor potència cal calcular el valor mitjà de tensió i de corrent.

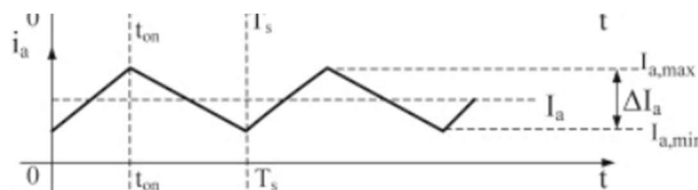


Figura 43. Forma d'ona de corrent d'un senyal PWM.

Per tant, aprofitant el mòdul computacional, es poden calcular mitjanes de senyals per hardware sense haver d'ocupar recursos de programari, optimitzant temps, i permetent que aquest temps sigui controlable, i a més, no es perden mostres.

Un cop realitzada la mitjana, el convertidor queda parat fins que no rep un altre flanc de pujada. Això vol dir que en una sola conversió, totes les mostres de dades s'acumulen fins que l'ADCNT, que és el comptador de mostres, coincideix amb el valor ADPRT establert, i llavors

es divideix el registre de l'acumulador, desplaçant a la dreta els bits que equivalguin al número fixat en el registre ADCRS. Aquest número només és potència de dos. En aquest cas, per tenir la mitjana més acurada, s'ha fixat el valor màxim que permet que és sis, equivalent a que es recullen 64 mostres de cada senyal. Per tant, el registre ADPRT ha de coincidir amb aquest valor, així doncs, s'ha entrat un valor de 64.

En la Taula 32, hi ha un recull de les cinc entrades analògiques del sistema. En cada entrada se li configura un pin, i a partir de les dades proporcionades pel full de característiques del producte, variant el registre ADPCH, es canvia el canal que es vol llegir.

Canal del convertidor	Valor del registre ADPCH	Pin del PIC	Variable de lectura
ANA0	0	RA0	Corrent de les bateries
ANA1	1	RA1	Tensió de la bateria u
ANA2	2	RA2	Tensió de la bateria dos
ANA3	3	RA3	Corrent de l'inversor
ANA4	4	RA4	Corrent del frenat

Taula 32. Senyals analògics del microcontrolador.

El procediment de lectura de les entrades analògiques és el següent: primer de tot es selecciona el multiplexor que llegeixi el canal ANA0, escrivint el registre ADPCH a zero. El mòdul ADCC no realitza res fins que no es rep el senyal de dispar. Un cop es rep el dispar, el sistema comença a adquirir mostres amb un període de mostreig de 19,55 microsegons, que ve determinat per la freqüència configurada a l'ADC. D'aquesta manera, en recollir les 64 mostres el sistema tardarà 1,25 mil·lisegons. Un cop recollides, el sistema calcula la mitjana i posa a nivell alt el bit d'interruptió ADTI, que indica que l'ADCC ja té un resultat, interrompent el sistema. Durant la interrupció, es llegeix la mitjana de les mostres, i es canvia el canal del multiplexor. Per tant, per tal de sincronitzar el sistema correctament, el senyal de dispar s'ha de realitzar en un temps superior a 1,25 mil·lisegons, en cas que es donés un dispar per sota d'aquest temps, el sistema l'ignoraria. Per evitar que el sistema tingui interrupcions degudes a l'ADCC en marges de temps molt consecutius, el senyal de dispar de l'ADCC s'ha configurat amb el temporitzador TMR1, i s'ha configurat que amb un temps de deu mil·lisegons. D'aquesta manera, el sistema té marge per canviar l'entrada del multiplexor, i realitzar altres funcions. Altrament, durant la interrupció el sistema llegeix el buffer, i canvia el canal del multiplexor, com que aquesta operació és molt ràpida, la durada d'aquesta interrupció és molt breu, i es pot procedir ràpidament a realitzar altres tasques.

La funció de la interrupció llegeix el buffer del resultat, considerant que està dividit en dos registres, un pels bits més significatius i un pels menys significatius. Seguidament, es realitza la funció `Lectura_Analogica()`.

```
void Lectura_de_valorMitja_AD(void){
MitjanaMostres=((uint16_t)((ADFLTRH << 8) + ADFLTRL));
Lectura_Analogica();
}
```

La funció `Lectura_Analogica()`, a través d'un switch-case, assigna les mitjana de mostres a la variable assignada al pin que ha llegit, i canvia el canal del multiplexor, incrementant un comptador de control. A continuació hi ha un exemple del codi on es mostra el que s'ha explicat.

```
case 0x00:
//PIN RA0
IBateria=(int)(MitjanaMostres-512);//Se li resta l'OFFSET de 2,5 Volts.
CanalLectura++;
ADPCH = CanalLectura;
break;
```

El mòdul ADCC retorna un valor de setze bits sense signe, per aquest motiu, com que el corrent de la bateria, pot ser tant positiu com negatiu, es canvia el valor a una variable entera amb signe. Per la resta de variables, es tracten com a valors enters sense signe.

Les variables que mesuren corrent, com que el sensor de corrent ACS758, presenta un offset de 2,5 V quan no circula corrent, cal restar aquest valor, per poder tractar les dades de forma correcta. Com que el convertidor és de 10 bits, i el senyal que mesura el pin de lectura va de zero a cinc volts, l'offset que es resta és 512.

El senyal de dispar de l'ADCC es realitza cada 10 mil·lisegons, per tant, cada 10 mil·lisegons es llegeix un canal diferent, considerant que hi ha cinc canals, cada 50 mil·lisegons, s'actualitzaran tots els valors de les variables.

Un cop obtingut el valor de la variable, cal efectuar el càlcul de potències. Aquest càlcul es duu a terme cada 60 mil·lisegons, a través d'una interrupció temporitzada, definida pel temporitzador zero, per donar marge al convertidor analògic digital a adquirir totes les dades dels diferents canals.

Dins d'aquesta interrupció es realitza la funció de càlcul de potència, i es posa a nivell alt el bit de test en cas que els sistema es trobi fora de l'aturada d'error o emergència.

En la funció de `Calcul_de_Potencies()`, s'han considerat les diferents potències mesurades, incloent la potència generada. En el càlcul de potències, s'ha tingut compte amb el tipus de variable que treballa el sistema. En la major mesura del possible, s'ha evitat treballar amb variables amb coma flotant. El motiu és que aquestes variables tenen un mínim de 32 bits, i el microcontrolador és de vuit bits, per tant, per optimitzar el codi s'ha decidit utilitzar en la mesura del possible variables de com a màxim setze bits.

Aquesta funció, el primer que realitza és comparar que la potència generada, que ja arriba amb un format enter sense signe, i es pot tractar com un valor decimal amb watts com a unitat, no superi un límit de potència perillós pel sistema. En cas de superar-lo, posa el sistema en emergència i es comunica al sistema de control de l'angle d'inclinació. Si la potència és inferior al mínim de commutació, es posa a zero, per donar una zona morta i que el sistema romangui amb la maniobra de baix consum, i es comunica en els altres dispositius de la turbina que no es detecta generació.

Seguidament, es calcula la potència mesurada en l'inversor i en el frenat elèctric dissenyat. Aquests dos elements queden connectats en els terminals del bus de continua del regulador de càrrega. En aquest punt, la tensió és de 24 V i només varia el corrent. Així doncs com que la tensió és constant, només cal tractar el corrent d'aquestes variables. El corrent a través de l'ADCC està en valor enter sense signe, així doncs, cal convertir aquesta variable en un valor enter decimal amb ampers. Com ja s'ha mencionat, s'ha intentat treballar amb variables de setze bits, així doncs, cal adaptar la fórmula de l'Equació 10 per complir el criteris de disseny i no causar cap error en el procediment.

$$P(W) = \text{ValorADCC}(\text{enter}) \cdot K_{\text{conversio}} \quad (\text{Eq.10})$$

La constant  $K_{\text{conversió}}$  és la relació entre la resolució del ADC del PIC i la sensibilitat del sensor, multiplicada per la tensió del bus que és de 24 V. Aquesta constant té un valor de 5,865, així doncs, s'ha arrodonit a 5,9, d'aquesta manera, per digitalitzar el sistema i no sobrepassar els 65.536, que provocaria un desbordament de la variable, el càlcul d'aquestes dues potències, es realitzen com el fragment de codi adjuntat.

```
Pinversor_W=(59*Iinversor)/10;
```

```
PFRENAT_MESURADA=(59*IFrenat)/10;//IGUAL QUE L'INVERSOR.
```

Per la potència del sistema de bateries es tenen en compte les tensions de les dues bateries i el corrent que circula cap a elles. Pel que fa a la tensió, com que l'ADC també retorna un enter, tenint en compte la resolució i la sensibilitat del sensor, el valor de tensió amb volts enters decimals, segueix l'Equació 11.

$$V_{bateria}(W) = \text{ValorADCC}(\text{enter}) \cdot 0,0235 \quad (\text{Eq.11})$$

Com que es mesuren dues bateries en sèrie, les dues tensions s'han de sumar, i per tal de compensar l'error, la constant que mesura el valor obtingut per l'ADCC, en un s'ha deixat a 0,23 i a l'altre a 0,24. Tenint en compte que els tipus de variables són enters, s'ha calculat la tensió de les bateries com el fragment de codi adjuntat.

```
VBateria=(23*VBateria1)/1000+(24*VBateria2)/1000; //Tensió de les bateries.
```

El corrent pot ser positiu o negatiu, així doncs s'han considerat els casos on aquest entrega o consumeix potència, però la potència s'ha volgut tractar com un valor positiu. Segons el conveni establert, quan la bateria entrega potència el corrent és negatiu, i quan l'absorbeix és positiu. Com que a causa dels defecte tèrmics del sistema la bateria es pot autodescarregar, s'ha establert una zona morta on no hi circula corrent, d'aquesta manera el sistema pot considerar el nivell de potència zero.

```
if(IBateria>Ibat_inferior&&IBateria<=Ibat_superior) IBateria=0;
```

Per passar d'un valor enter de intensitat a un valor enter amb magnitud d'ampers, s'ha aplicat l'Equació 12, tenint en compte la resolució i la sensibilitat del sensor.

$$I_{bateria}(A) = \text{ValorADCC}(\text{enter}) \cdot 0,0242 \quad (\text{Eq.12})$$

La variable del corrent de la bateria, en el programa també es tracta com la resta de variables d'intensitat mesurades, però a l'hora de calcular la potència, com que es deixa sempre com a valor positiu, es canvia el signe en el càlcul de potència. El corrent de la bateries es deixa amb valor enter sense signe, perquè en l'algoritme de transferència de càrrega i en la monitorització feta pel PLC es considera si el valor és positiu o negatiu.

```
Intensitat_bateria=(IBateria*24)/100;
  if(Intensitat_bateria<0){
    Pbateria_W=((unsigned int)(Intensitat_bateria*-1))*VBateria;
  }else{
    Pbateria_W=(unsigned int)(Intensitat_bateria)*VBateria;
  }
}
```

### 5.10. Programació del dispositiu de càlcul de potència generada.

El microprocessador ESP32 s'ha programat a través del paquet del Platform.io del programa VScode. Aquest s'ha programat amb la versió de C de l'Arduino. A més, l'entorn de programació del VScode, presenta els drivers de la placa, i les diferents lliberies que s'han utilitzat per l'aplicació que s'està desenvolupant.

Per programar amb llenguatge Arduino s'ha inclòs la llibreria Arduino.h, i per treballar amb I2C, s'ha afegit la llibreria Wire.h.

En el codi també es defineixen variables globals i els diferents constants. Com les constants per tractar les dades provinents de l'ACS758. En aquest cas, com que el convertidor analògic digital és de 12 bits, es defineix la resolució per passar del valor enter a valor de voltatge adquirit, i la constant del sensor que canvia de valor de voltatge a intensitat.

```
const float V_D_factor = 3.3/(4096-1); //Resolució ADC
const float A_V_factor = 50.0; //S=20 mV/A ->1/S=50V/A
```

En aquest punt del codi, també es defineixen les constants que passen del valor de freqüència elèctrica al valor de voltatge.

```
float VRMS;
const float const_freq_rpm=7.5;// Càlcul del la velocitat de rotació del generador,
const float Vrms_Nvoltes= 400.0/480.0;
```

El microprocessador només llegeix un canal del seu convertidor analògic digital. El mètode d'adquisició de les dades d'intensitat es fa mitjançant una interrupció temporitzada, ja que

d'aquesta manera es pot obtenir amb precisió el valor de cada mostra en un temps de mostreig constant, obtenint de forma controlada punts homogenis del senyal de mesura.

Així doncs, el microprocessador llegeix una variable en enter. Aquesta ha de poder estar disponible fora de la rutina d'interrupció, per tant, la directiva del compilador ha de ser de tipus volàtil. Específicament, el compilador carrega la variable de la RAM i no d'un registre d'emmagatzematge, que és una ubicació de memòria temporal on s'emmagatzemen i manipulen les variables del programa. Les altres dos variables que es tracten en la rutina d'interrupció temporitzada són: la variable de mostres, que compta cada mostra que es va adquirint, i serà útil pel càlcul de la freqüència elèctrica; i la variable interrupció, que ve a ser una bandera que marca que hi ha hagut la interrupció, i permet controlar l'emmagatzematge i el tractament de dades de corrent en un vector.

```
volatile int analog_in; //Variable del resultat de la conversió AD.
hw_timer_t * timer = NULL; //Declaració del timer d'adquisició.
volatile uint16_t mostres=0; //Mostres que s'adquireixen per cada període del senyal.
volatile bool interrupcio=false; //Flag que indicia si hi ha hagut interrupció.
```

La lectura d'intensitat es realitza a través del sensor ACS758 que està connectat en el seu canal analògic. La tensió de sortida d'aquest sensor, quan no circular corrent, té un òfset que equival a la meitat de la tensió d'alimentació. Com que el sensor s'alimenta a 3,3 V, l'òfset és de 1,65 V. Per compensar aquest òfset per programa, com que el convertidor és de dotze bits, cal restar 2.047, que equival a la conversió a digital de la tensió de 1,65 V, en el valor adquirit per tenir valors positius i negatius. Addicionalment, dins de la rutina d'interrupció, com que s'ha aconseguit una mostra, s'incrementa el comptador de mostres, i la variable booleana que indica una interrupció es posa a 1.

```
void IRAM_ATTR onTimer() {
    analog_in =(analogRead(A6)-2047); //Lectura de l'entrada analògica i restar l'òfset.
    mostres++; //Com que s'ha adquirit una mostra es suma el comptador.
    interrupcio=true; //Es fica el flag d'interrupció a true.
}
```

Finalment, en aquest punt del codi també es defineixen les variables globals, i el vector per poder calcular valors eficaços de corrent i la potència aparent. Un aspecte a comentar en aquest punt és que s'ha definit el vector "corrent\_i[]", de 1400 valors en coma flotant, que permet mesurar freqüències elèctriques fins a tres hertz. Aquest vector ocupa 5,6 kB de memòria.



```
float SRMS; //Valors de Saparent
const int Mostres_TOTALS=1400; //Vector de càlcul de corrent
float Corrent_i[Mostres_TOTALS]; //Variables de control pel càlcul RMS.
int comptador_mostres=0;
int pas_PerZero=0;
float Valor_zero;
float SumaQuadrats=0;
float IRMS;//VARIABLES RMS
float frequencia;
```

El següent punt de codi a comentar és el `setup()`. Dins d'aquesta funció s'inicialitzen tots els paràmetres necessaris per dur a terme l'algoritme del programa.

Dins del `setup()` es defineix el temporitzador que durà a terme la interrupció per llegir les dades. El temps que se li dona equival al temps de mostreig, que en aquest cas és de 500 microsegons. El motiu de la configuració d'aquest temps de mostreig és el següent: partint que segons el fabricant de l'alternador la velocitat màxima de rotació és de 630 revolucions per minut, i que la màquina està formada per vuit parells de pols. Com que la freqüència elèctrica màxima és de 84 Hz, que equival a un període d'11,90 mil·lisegons. Per consegüent, per obtenir el senyal de manera precisa, s'ha fixat un període de mostreig 20 vegades inferior al del senyal de mesura. D'aquesta manera, segons el criteri de Nyquist–Shannon, superem la freqüència mínima de mostreig, que equival a dos vegades la freqüència del senyal, per obtenir la freqüència de forma correcta. Per tant, segons el que s'ha comentat anteriorment, cada 595,20 microsegons s'hauria d'adquirir una dada, però com que aquest temps no pot ser parametrizat pel maquinari del dispositiu, s'ha buscat un valor inferior que permeti el dispositiu, així doncs, el temps de mostreig es farà cada 500 microsegons.

El temporitzador associat té una freqüència de 80 MHz. El que primer es duu a terme és inicialitzar-lo, a partir de la funció `timerBegin()`, que aplica un divisor de 4.000, perquè el període del senyal de rellotge del temporitzador sigui de 500 microsegons. Seguidament, la rutina on s'adquireixen les dades, es defineix en el vector d'interrupció de temporització amb la funció `timerAttachInterrupt()`. El següent pas tracta de definir els períodes de rellotge que tardarà a fer la interrupció el sistema, amb la funció `timerAlarmWrite()`, que en aquest cas, per a cada període de rellotge es farà la interrupció, i finalment amb la funció `timerAlarmEnable()` s'habilita la interrupció.

```
timer = timerBegin(0, 40000, true); //Divideix 80MHz/40000 = 2 kHz --> 1/2kHz = 0.5 ms
timerAttachInterrupt(timer, &onTimer, true); //Associem interrupció a que executi ISR
```

```
timerAlarmWrite(timer, 1, true); // El trigger de la interrupció serà cada 0.5 ms.  
timerAlarmEnable(timer); //habilita la interrupció
```

Finalment, com s'ha anat detallant, el microprocessador té la capacitat de posar-se en mode de deep-sleep gràcies a una comunicació I2C. Per tal de despertar-se i funcionar de nou, iniciant tot el procés del setup() i reiniciar el loop(), el microprocessador ho executa a partir d'un flanc de pujada detectat al pin 14. Aquest pin està inclòs dins d'una sèrie d'entrades i sortides digitals, associades un processador de molt baix consum, que funcionen mentre el microprocessador està dormint. D'aquesta manera, quan el pin 14 passa de nivell baix a alt, s'emet un senyal que desperta el microprocessador, i es posa en funcionament. Aquest pin associat a l'impuls de despertar-se es defineix en la funció `esp_sleep_enable_ext0_wakeup()`.

```
esp_sleep_enable_ext0_wakeup(GPIO_NUM_14,1); //Wake-up en el pint 14 de low a high.
```

La funció del loop() consisteix en el bucle infinit, on el microprocessador executa les operacions de càlcul per poder obtenir els valors de potència eficaç, i l'acció de preparar les dades per ser enviades via I2C en cas de petició. Val a dir, que les accions d'aquest bucle venen condicionades per les interrupcions del sistema. Pel que fa al càlcul de la potència eficaç, involucrant les operacions de la tensió i el corrent eficaç, el bucle segueix el diagrama de flux de la Figura 44.

Totes les variables que intervenen en els càlculs de valor eficaç estan inicialitzades a zero, per tant, en l'Estat 1, es calcula la potència eficaç, i aquest valor s'actualitza per a cada període del senyal de mesura, que en aquest cas és la intensitat de sortida del generador. En conseqüència, en la primera iteració la potència eficaç val zero, i en les següents iteracions, la potència eficaç val el valor previ del càlcul posterior, en altres paraules.

Quan es dona la interrupció i es torna al loop(), com que hi ha hagut una bandera d'interrupció, el sistema va de l'Estat 1 a l'Estat 2. En aquest estat, es calcula el valor adquirit en un float, amb les constants que passen del valor d'integer d'intensitat a Ampers, que s'han comentat anteriorment. El valor en float calculat es guarda en el vector "corrent\_i[]". Per a cada mostra adquirida analitza si s'ha fet un pas per zero. Per poder contemplar aquest anàlisi i que el programa no faci incoherències, el pas per zero es comença a avaluar a partir de la segona mostra, que equivaldria a passar a l'Estat 3. El pas per zero del senyal es determina: quan el valor instantani és més gran que zero, i el valor anterior és més petit que zero, o al cas invers. Per cada pas per zero s'incrementa un comptador que està predefinit a zero. Un cop calculat aquest procés, s'incrementa el comptador de posició del vector vector i borra la variable

booleana que indica que hi ha hagut interrupció d'interrupció, per adquirir les següents mostres.

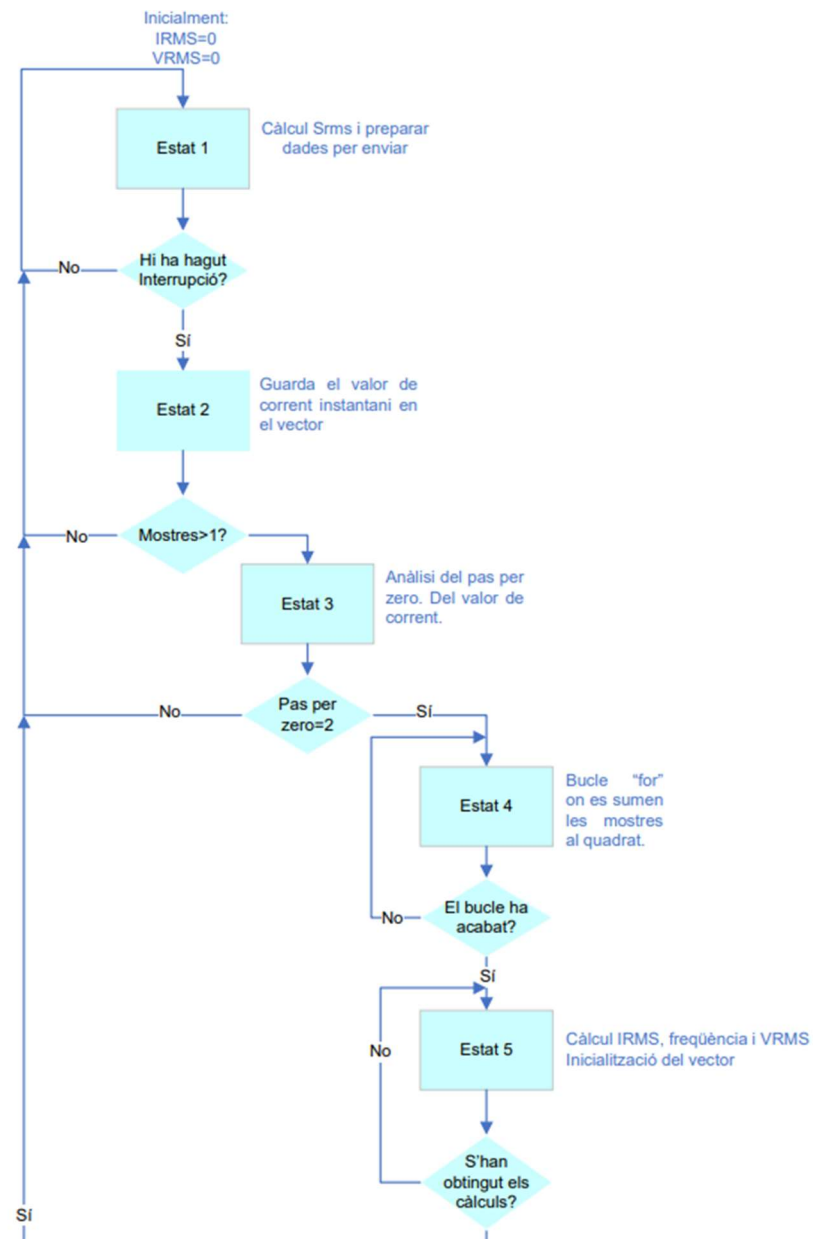


Figura 44. Diagrama de flux del càlcul de la potència eficaç

```

if(interruptio){
    Corrent_i[comptador_mostres]=analog_in*V_D_factor;
    if(comptador_mostres>1){
if((Corrent_i[comptador_mostres]>0.0&&Corrent_i[comptador_mostres-
1]<0.0)||Corrent_i[comptador_mostres]<0.0&&Corrent_i[comptador_mostres-
1]>0.0)||Corrent_i[comptador_mostres]==0.0){

```

```

    pas_PerZero++;
}}
comptador_mostres++; //Compte les dades del vector
interrupcio=false;//Es borra el flag de software d'interrupció.
}

```

Quan el comptador de pas per zero arriba a dos, es fa un bucle "for" per fer un sumatori de cada mostra al quadrat, seguint l'Equació 1, i per a cada iteració se suma un comptador, anomenat "i". Quan el comptador arriba al valor equivalent de les mostres guardades en el vector, es para el bucle i es fa l'arrel quadrada de la divisió del sumatori de les mostres al quadrat entre el nombre de mostres. D'aquesta manera es té el valor eficaç d'intensitat.

```

if(pas_PerZero==2){//Quan es detecten dos passos per zero es té un període d'ona.
    for (size_t i = 0; i <(comptador_mostres) ; i++)
        SumaQuadrats=SumaQuadrats+Corrent_i[i]*Corrent_i[i]; }
IRMS=sqrt(SumaQuadrats/(mostres));

```

Com que es coneix el temps de mostreig exacte per la interrupció temporitzada, amb el producte del valor total de mostres adquirides pel període de mostreig, es pot tenir el període del senyal. I si es fa l'invers d'aquesta multiplicació, es té la freqüència del senyal elèctric. Immediatament després del càlcul de freqüència i intensitat eficaç, es reinicialitzen a zero les variables que s'utilitzen per als càlculs de valor eficaç, com: la variable "mostres", que és amb la que es té el valor de freqüència, o la variable "comptador\_mostres", que és el comptador que posiciona el vector.

```

frequencia=1.0/((mostres)*0.0005);
comptador_mostres=0;
pas_PerZero=0;
mostres=0;
SumaQuadrats=0;

```

Un cop realitzats els càlculs elèctrics, s'arrodoneix el valor de la potència aparent, passant les variables de tipus coma flotant a tipus enter. I amb la finalitat d'enviar el valor via I2C, aquesta variable se separa en dos bytes, per enviar les dades en cas de petició del mestre.

```

char *dadesEnviar = (char *) &potencia;
LSBPotencia=dadesEnviar[0];
MSBPotencia=dadesEnviar[1];

```

Perquè el microprocessador ESP32 es posi en deep-sleep, el mestre via I2C ha d'enviar un datagrama d'un byte amb el valor hexadecimal 0x3C. En aquest cas, es trenca el bucle de la funció loop(), posant el dispositiu en mode de molt baix consum energètic. Quan es desperta del deep-sleep, el microprocessador torna a iniciar el setup(), i seguidament es reinicia el loop().

```
if(Produccio==0x3C){
  esp_deep_sleep_start();//Funció de dormir.
}
```

### 5.11. Programació del baix consum

En l'apartat dos del capítol dos, es planteja com es pretén allargar al màxim possible l'energia de la bateria, a través del mode baix consum que incorporen els PIC.

Quan no hi ha generació, per tal d'allargar al màxim possible l'energia acumulada a la bateria, s'estalvia energia posant en mode baix consum els elements de control. El mode de baix consum o mode sleep(), inhabilita la majoria dels perifèrics, i redueix la freqüència de rellotge del dispositiu. En els casos dels perifèrics de comunicació, tant els perifèrics d'I2C com el de la EUSART, estan dormits però permeten rebre adreces i dades provinents d'un dispositiu extern. Pel que fa el CLC i el PWM, com que el frenat no està activat, i no estan habilitats per treballar durant el mode sleep, romanen apagats. Durant el mode sleep, només funcionen dos temporitzadors, el TMR0 i el TMR1. Gràcies el temporitzador TMR1 i la interrupció de l'ADCC, el sistema pot seguir adquirint les dades del l'ADCC. A més, com que el TMR0 també funciona, el sistema també pot calcular les potències adquirides per l'ADCC i testejar el sistema. En aquest cas, com que el sistema es troba en estats de no generació, l'error de coherència degut a una potència generada zero i la potència del frenat i l'inversor diferent de zero no es detectarà. Així, els errors que es poden detectar són els errors de potències de les bateries o l'inversor, i l'error de coherència a causa de la lectura errònia dels sensors de les bateries.

El fet de no haver generació és percebut per tots els sistemes de control i monitorització de la turbina: el sistema de monitorització detecta que el vent és molt fluix, el sistema de control de l'angle d'atac detecta a través de l'encoder que la velocitat de rotació és pràcticament nul·la, i el sistema de control de potència considera la potència generada com a nul·la. Quan els tres

dispositius detecten que no hi ha generació, es permet que el sistema de control es posi a dormir. Quan es detecta generació per part dels tres dispositius, el sistema treballarà sempre despert, la condició que es posi a dormir serà sempre falsa. A través de l'I2C, els altres sistemes envien si detecten generació. De la manera com s'ha configurat el sistema, els tres sistemes de la turbina detecten o deixen de detectar la generació a la vegada. És a dir, el sistema de monitoratge detecta un règim de vent concret, que provoca que el sistema de control de l'angle d'inclinació de les pales detecti un règim de revolucions concretes, i que el sistema de control de potència detecti una potència específica.

Quan la condició de dormir és certa, s'envia al dispositiu de mesura de potència que es posi a dormir, s'inhabilita el Watchdog, per evitar que vagi despertant el sistema, i es posa a dormir el PIC.

Com que el temporitzador cinc, que és el que obté la potència generada a través de l'I2C està inhabilitat, després de despertar-se el sistema, es canvia de nivell baix a nivell alt el senyal de control que desperta el microprocessador ESP32, es fa un retard de 100 mil·lisegons, perquè el sistema de mesura de potència generada obtingui el valor de potència. I seguidament, es connecta amb el sistema de mesura de potència generada, per obtenir el valor de potència per tal de determinar si el sistema es pot posar en generació.

Cada vegada que es desperti el dispositiu, rehabilitarà el Watchdog, despertarà l'ESP32, farà l'espera de 100 mil·lisegons per donar temps al sistema de mesura de potència a obtenir valors, es comunicarà amb aquest dispositiu per tenir les dades de potència generada, i procedirà a fer una volta al llaç infinit. Si en els testos es detecta un error el sistema, el sistema passarà a estar amb error, en cas contrari, el sistema aplicarà l'algoritme de transferència de potència i si el sistema no detecta generació, seguirà en mode sleep, després de transferir la potència.

```
if(!GENERACIO_E1&&!GENERACIO_E2&&!ESTAT_AMB_GENERACIO) {
    dormirESP32();//DORMIM LA ESP32
    WDTC0N0bits.SEN=0;//EL DESHABILITEM PER EVITAR QUE DESPERTI EL SISTEMA
    SLEEP();//Dorm el PIC
    //S'HA DESPERTAT AQUÍ
    WDTC0N0bits.SEN=1;//REAHABILITA EL WATCHDOG.
    despertarESP32();//ES DESPERTA L'ESP32 I ES DONA MARGE A CALCULAR POTÈNCIES
    __delay_ms(100);
    I2C_Master_Start();           //Start condition
    I2C_Master_Write(0x31);       //7 bit address + Read
    uint8_t Lectura2 = (uint8_t)I2C_Master_Read(1); //Read + Acknowledge
```

```
uint8_t Lectura1 = (uint8_t)I2C_Master_Read(0); //Read + Acknowledge
I2C_Master_Stop(); //Stop condition
PotenciaGenerada=(unsigned int)((Lectura2<<8)|Lectura1);
//FA UNA VOLTA AL LOOP(), I TOTES LES INTERRUPCIIONS FUNCIONEN
}
```

Altrament, les interrupcions per perifèrics permeten despertar el sistema per determinar l'estat dels dipòsits, i a través de l'algoritme de transferència de potència, es determina si es pot connectar la bomba.

## 6. COMUNICACIONS

En el present capítol es comenta tot el que està relacionat amb les comunicacions. La comunicació entre els diferents dispositius de control i monitorització de la turbina s'ha dut a terme a través d'una xarxa I2C multimestre, així el sistema pot treballar de manera aïllada sense comunicació amb el PLC. D'aquesta manera, el PLC, a través de la xarxa Modbus, pot fer lectures de diferents variables de la turbina encarregades de monitoritzar i controlar els sistemes de cada dispositiu, i a més pot escriure variables de control, per adequar els llinars a la posada amb funcionament, i escriure directament sobre les sortides dels microcontroladors per testejar-los. En la Figura 45 es mostra un esquema de com queden les dues xarxes de comunicacions.

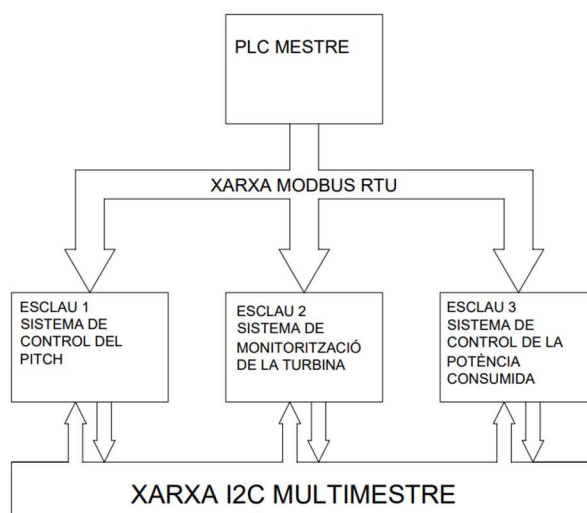


Figura 45. Xarxes de comunicacions dels dispositius de la turbina.

Així doncs, el present capítol es divideix en dos apartats, un que es comenta tot el referent a la comunicació I2C i l'altre a la comunicació Modbus.

### 6.1. Xarxa I2C Multimestre

L'apartat de l'I2C està format per una introducció on s'expliquen justificacions i consideracions de la xarxa multimestre de l'I2C, seguidament anirà un subapartat explicant la implementació del dispositiu treballant com a mestre, un altre subapartat on es comentarà la implementació del dispositiu treballant com a esclau, un quart apartat que detalla el funcionament del test de



comunicacions de la xarxa multimestre, i l'últim apartat que explica com s'ha dut a terme la comunicació entre el dispositiu de mesura de potència i el PIC.

### **6.1.1. Introducció i justificacions de la xarxa d'I2C**

El protocol de comunicació I2C és un estàndard de comunicació sèrie, utilitzat en sistemes electrònics per a la transferència de dades entre dispositius integrats.

En aquesta comunicació en sèrie, s'utilitzen dues línies, una destinada a transferir de dades (SDA), i l'altra és el senyal de rellotge (SCL) que permet la sincronització de la transmissió entre els dispositius connectats al bus. A més, el protocol I2C permet la transmissió de dades en ambdues adreces, cosa que permet que els dispositius actuïn com a mestres o esclaus al bus, dotant al bus de comunicacions d'una gran flexibilitat i robustesa en la connexió de múltiples dispositius formant una xarxa multimestre.

Els diferents sistemes de control de la turbina estan connectats a la xarxa I2C multimestre, perquè aquests dispositius treballin de manera independent al PLC, descentralitzant-ne el control total. Aquesta xarxa de comunicacions ha de ser molt robusta perquè ha de poder parar la turbina en cas de diagnosticar-se una situació d'emergència. L'altre ús de la xarxa multimestre és detectar que no hi ha generació, per posar a dormir els tres dispositius de control. De la mateixa manera que si es detecta generació, el sistema de control de potència aprofita la xarxa multimestre perquè els altres dispositius deixin de treballar en mode baix consum.

Per dotar el sistema d'una major robustesa, s'utilitzarà l'I2C diferencial, passant de dos a quatre conductors. D'aquesta manera, la xarxa de comunicacions adquireix: una major immunitat al soroll, una major distància i velocitat de transmissió, menor emissió d'interferències electromagnètiques i més seguretat en la transmissió de dades. Gràcies a la utilització del mòdul diferencial, la xarxa I2C pot funcionar a 400 kHz, permetent en els PIC executar les comandes de comunicació d'una manera més ràpida, actuant de forma més immediata en casos d'emergència, i sent més improbable que es produeixin col·lisions.

El PIC18F47Q10 disposa de dos interfícies MSSP, que permet està interconnectat en una mateixa xarxa I2C, actuant com a mestre i com esclau. Per tal de parametritzar la xarxa multimestre, el MSSP1 s'ha configurat per poder treballar com a mestre, i el MSSP2 com a esclau amb l'adreça 0x30.

Per programar l'I2C s'ha hagut d'anar a un nivell molt baix: primer s'ha determinat cada bit que s'envia per l'I2C segons el protocol, posteriorment s'ha hagut d'estudiar com queden sincronitzats els dos dispositius amb els diferents modes de transmissió, i finalment seguint el full de dades del fabricant s'han de configurat els registres de control del MSSP, perquè es segueixi el protocol I2C de manera correcta i que els dispositius quedin ben sincronitzats.

L'I2C té dos modes de transmissió. El primer és el mode d'escriptura o de transmissió de dades: en aquest mode, les dades únicament només es transmeten des d'un dispositiu mestre a un dispositiu esclau. El mestre comença la comunicació enviant un senyal d'inici a l'esclau. A continuació, el mestre envia el primer byte de transmissió, on set bits formen la direcció de l'esclau al qual es vol comunicar, i l'últim bit que marca si el mestre està en mode escriptura o lectura, que com que està en mode escriptura està a nivell baix. Quan l'esclau rep el datagrama, respon amb un senyal de reconeixement, que és un bit a nivell baix. Després de rebre el justificat de recepció, el mestre envia els següents paquets de dades a l'esclau. L'esclau tramet un senyal de reconeixement després de rebre cada byte de dades. Després que es remeten totes les dades, el mestre tramet un senyal de parada per finalitzar la comunicació. El segon mode és el de lectura, en aquest mode el dispositiu mestre sol·licita dades al dispositiu esclau i espera que l'esclau transmeti les dades sol·licitades en resposta a la sol·licitud del mestre. En aquest cas, el mestre envia un senyal d'inici i després l'adreça del dispositiu esclau amb què vol comunicar. Després de rebre un senyal de reconeixement de recepció de l'esclau, el mestre espera que l'esclau transmeti les dades sol·licitades. Llavors, el dispositiu mestre funciona com a receptor i el dispositiu esclau com a transmissor. El mestre envia un senyal reconeixement de recepció després de rebre cada byte de dades que li envia l'esclau. Quan l'esclau envia l'última dada envia un senyal de reconeixement a nivell alt, indicant al mestre que ha rebut totes les dades, i finalment, el mestre envia un senyal de parada per finalitzar la comunicació. En la Figura 46, es mostra com funciona la transmissió de l'I2C.

Cada vegada que es produeix un flanc descendent en el senyal de rellotge, es transmet un bit en el bus. La condició d'inici es dona quan el canal de dades passa de nivell alt a nivell baix, seguidament s'habilita el senyal de rellotge, i es detecta un bit amb nivell lògic zero, que és el d'inici. En la condició de parada, el rellotge es posa a nivell alt, i en el canal de dades es detecta un nivell baix de senyal, sent el bit de parada, en aquest punt el canal de dades i el del rellotge queden a nivell alt de manera indeterminada.

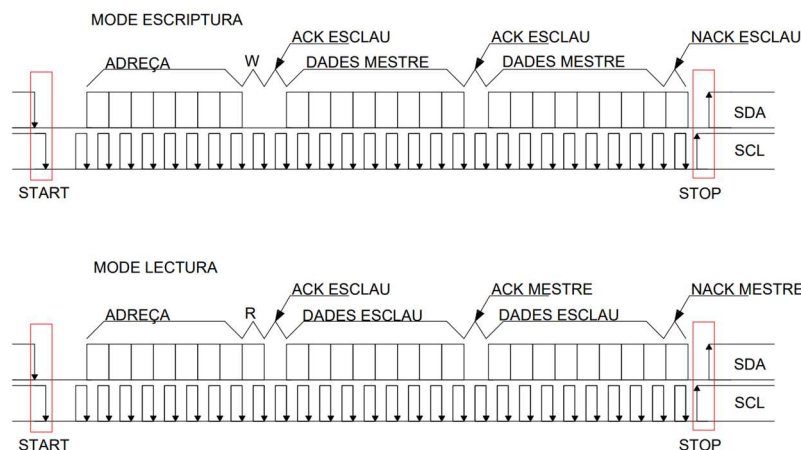


Figura 46. Seqüència de bits de l'I2C.

Cal tenir clar els codis que s'utilitzen en la xarxa multimestre. Aquests s'adjunten en la Taula 33. Els diferents codis s'han escollit de manera arbitrària i sota consens, al llarg del desenvolupament del projecte.

Codi hexadecimal	Funció
0x30	Adreça del microprocessador de càlcul de la potència generada. Amb el bit menys significatiu amb mode escriptura.
0x31	Adreça del microprocessador de càlcul de la potència generada. Amb el bit menys significatiu amb mode lectura.
0x20	Adreça del dispositiu de monitorització de la turbina. Amb el bit menys significatiu amb mode escriptura.
0x21	Adreça del dispositiu de monitorització de la turbina. Amb el bit menys significatiu amb mode lectura.
0x10	Adreça del dispositiu de control de l'angle d'atac. Amb el bit menys significatiu amb mode escriptura.
0x11	Adreça del dispositiu de control de l'angle d'inclinació. Amb el bit menys significatiu amb mode lectura.
0xA3	Byte de dades de frenat elèctric al màxim.
0xBB	Byte que indica que el sistema se li vol fer un test.
0x1F	Byte de dades de l'angle d'atac al màxim.
0xE1	Byte de dades de test del dispositiu de control de l'angle d'atac.

Taula 33.1. Codis de la xarxa multimestre d'I2C.

Codi hexadecimal	Funció
0xE2	Byte de dades de test del dispositiu de monitorització de la turbina.
0xE3	Byte de dades de test del sistema de control de potència
0x1E	Byte d'error del dispositiu de control de l'angle d'atac.
0x2E	Byte d'error del dispositiu de monitorització de la turbina.
0x3E	Byte d'error del sistema de control de potència
0x50	Byte d'adreça del dispositiu de control de la potència consumida. Amb el bit menys significatiu amb mode escriptura.
0x51	Byte d'adreça del dispositiu de control de la potència consumida. Amb el bit menys significatiu amb mode lectura.
0x12	Byte de detecció de no generació per part del dispositiu de control de l'angle d'inclinació.
0x13	Byte de detecció de generació per part del dispositiu de control de l'angle d'inclinació.
0x22	Byte de detecció de no generació per part del dispositiu de monitorització de la turbina.
0x23	Byte de detecció de generació per part del dispositiu de monitorització de la turbina.
0xF0	Byte de detecció de no generació per part del dispositiu de control de la potència consumida.
0xF1	Byte de detecció de generació per part del dispositiu de control de la potència consumida.

Taula 33.2. Codis de la xarxa multimestre d'I2C.

### 6.1.2. Mode mestre

En el mode Mestre, els registres de control del MSSP s'han de configurar perquè es puguin habilitar les opcions de transmetre i rebre dades, i a la vegada ha de controlar la seqüència del senyal de rellotge.

La primera funció que executa el PIC, i que es duu a terme en la funció del main(), en la part d'inicialitzacions de paràmetres per sobre del bucle del main(), és la inicialització del MSSP1 perquè treballi com a mestre. En aquesta funció, se li passa un paràmetre que és una variable de tipus long, per tal de definir la freqüència de treball del rellotge, i en conseqüència del bus.

```
void I2C_Master_Init(const unsigned long c){
```

```

SSP1CON1 = 0b00101000; //Configuració del MSSP1 actuant com a mestre de I2C
SSP1CON2 = 0;
SSP1ADD = (_XTAL_FREQ/(4*c))-1; //Definició de la freqüència del clock a 400 kHz.
SSP1STAT = 0;
    SSP1CLKPPS = 0x13;    //RC3->MSSP1:SCL1;
    RC3PPS = 0x0F;    //RC3->MSSP1:SCL1;
    RC4PPS = 0x10;    //RC4->MSSP1:SDA1;
    SSP1DATPPS = 0x14;    //RC4->MSSP1:SDA1;
    SSP1CON1bits.SSPEN = 0; //Es deshabilita el port sèrie per transmetre dades.
    SSP1CON3bits.SDAHT = 1; //Es prefixa el temps que detecten el nivell lògic.
}

```

El primer registre de control que s'escriu en la inicialització de l'MSSP1 és el SSP1CON1. En aquest cas, els quatre bits menys significatius se'ls configura perquè el dispositiu treballi com a mestre a una freqüència de treball que depèn del registre SSP1ADD. La freqüència en aquest registre queda definida a partir d'una fórmula en nombre enter decimal. En el codi es pot observar que a la funció se li passa un paràmetre anomenat 'c', que equival als 400 kHz. Quan treballa en mode mestre, el valor d'aquest registre aplica un divisor de la freqüència del rellotge principal del PIC, perquè doni la freqüència de rellotge de l'I2C. Seguidament, en el registre de control el bit que s'escriu és el sisè menys significatiu, que habilita el port sèrie per transmetre les dades. Els bits restants són bits de control, no s'utilitzen en el mode mestre, exceptuant el més significatiu, que es fa servir quan hi ha hagut una col·lisió, que automàticament s'escriu, i es borra per programa. El següent registre que se li atorga un valor és el SSP1CON2, en aquest cas, aquest registre de control manipula la sincronització del bus, activant la condició de parada, la condició d'inici, els bits de reconeixement, etc. Per tant, d'inici aquests bits estan tots a zero per deixar inactives les funcions de control de sincronització de l'I2C.

Seguidament, es declaren els pins del microcontrolador que seran el canal de dades i el senyal de rellotge. En el cas del mestre, el canal de dades és el pin RC3 i el canal per on hi haurà el rellotge és el pin RC4. Finalment, un cop assignats els pins, s'inhabilita el canal de transmissió, i en el registre de control tres es prefixa el temps mínim per detectar si un bit és un nivell lògic alt o baix, que en donar-se un u al tercer bit menys significatiu, el temps mínim de lectura lògica és de 300 nanosegons després del flanc de baixada del senyal de rellotge.

Un cop inicialitzat el PIC com a mestre amb l'MSSP1, perquè aquest dugui a terme el funcionament de la lectura i de l'escriptura, cal configurar els registres de configuració de forma adequada, així com l'escriptura i lectura del buffer de l'MSSP1. Tant si es vol llegir com

escriure, cal fer la condició d'inici i la d'aturada, així com les esperes per interactuar amb el buffer de forma adequada.

Durant la condició d'espera, el PIC queda encallat en un bucle, que no en surt fins que el registre SSP1STAT no detecta que s'ha transmès una dada del mestre en el bus, tenint en compte que ha de rebre el senyal de reconeixement per part de l'esclau; o que en el registre de configuració dos, ha de detectar que el mestre, que està en mode lectura, ha d'haver rebut les dades de l'esclau i que ha enviat el senyal de reconeixement. En aquest cas, amb aquest registre com que automàticament s'escriuen els seus bits, supervisa les accions anteriors analitzant si els bits corresponents a cada acció s'han escrit. Aquesta funció s'utilitza en qualsevol punt on es dugi a terme una acció de l'I2C, d'aquesta manera es té un funcionament més robust del bus.

```
void I2C_Master_Wait(){
    while ((SSP1STAT & 0x04) || (SSP1CON2 & 0x1F));}
```

La funció d'iniciar el bus, i el de parar-lo són molt similars, en aquest cas, consisteix en escriure el registre de configuració dos, sobre els bits corresponents a l'inici, que és el bit menys significatiu, i l'aturada, que és el tercer bit menys significatiu. Fins que no es borra el bit pertinent a l'operació que executa en el registre, el programa queda en bucle. Tant bon punt s'ha dut a terme l'acció desitjada, sigui inici, inici repetit o parada, per hardware es borra el registre. En el cas de la funció d'inici també s'ha de posar a nivell alt el bit que habilita el bus.

Les últimes dues funcions són la d'escriure un missatge i la de lectura de dades del bus. La d'escriptura tant si es vol llegir com escriure sempre s'inclou, ja que amb la funció d'escriptura s'escriu l'adreça, i després el mestre amb les funcions de lectura adquireix les dades i emet el senyal de reconeixement.

Per escriure un missatge en el bus, el primer que s'executa en la funció és una espera per assegurar que la maniobra anterior s'ha produït, sigui perquè s'ha rebut el reconeixement de l'esclau o s'ha donat el pols del senyal d'inici del bus per part del mestre. Un cop feta l'espera, el mestre escriu en el buffer del MSSP1 i la funció queda encallada fins que el mòdul MSSP1 no ha transmès el missatge, a partir del control del segon bit menys significatiu del registre SSP1ADD. En aquesta funció es passa un paràmetre de vuit bits, per tant, si es vol transmetre una variable entera, cal separar la variable en dos missatges, i llavors el dispositiu esclau ha de tenir en compte el que està rebent per tractar les dades, per aquest motiu la programació

de l'I2C és de molt baix nivell, perquè s'ha de saber exactament el que s'està enviant per tractar l'enviament i la recepció dels missatges.

```
static inline void I2C_Master_Write(unsigned short d){
    I2C_Master_Wait();//Acció d'espera
    SSP1BUF = d; //Escriure en el buffer.
    while(SSP1STATbits.R_nW); //Esperar que es buidi el buffer.
}
```

En la funció de lectura, el dispositiu ha de rebre dades i ha d'enviar el reconeixement cap a l'esclau en cas d'haver rebut el missatge de manera correcte. Cada vegada que s'utilitza la funció de lectura, el mestre ha de saber quantes dades s'han de rebre per executar la seqüència del senyal de reconeixement. Per realitzar aquest control, en la funció li passa un paràmetre de vuit bits que pot tenir un valor de zero o u. Amb aquesta variable executa el control del senyal de reconeixement. Si el paràmetre té valor u, el mestre un cop rep un byte, posa el bit de reconeixement a zero, significat que ha rebut el byte i que està a punt per rebre el següent.

```
unsigned short I2C_Master_Read(unsigned short a){
    unsigned short temp; //Variable de retorn
    I2C_Master_Wait(); //Espera del bus
    SSP1CON2bits.RCEN = 1; //habilita el bit de recepció
    I2C_Master_Wait();//Espera del bus que en aquest cas està rebent dades.
    temp = SSP1BUF; //El valor del registre SSP1BUF queda bolcat a la variable.
    I2C_Master_Wait();//Espera del bus
    SSP1CON2bits.ACKDT = (a)?0:1;//Seqüència ACKDT per tenir paquets de dades.
    SSP1CON2bits.ACKEN = 1;//Habilita l'Acknowledge.
    return temp; //Retorna la variable llegida.
}
```

Cada cop que s'executa la funció, el primer pas consisteix en fer l'espera de l'I2C, seguidament, s'habilita el mestre en mode recepció de dades, posant a nivell alt el quart bit menys significatiu del registre de control dos. Llavors, s'espera a rebre les dades del bus en el buffer, i quan aquest queda ple es fa la lectura del buffer i s'emmagatzemen les dades en una variable de vuit bits que és el retorn de la funció. Llavors, en funció del paràmetre que s'ha passat, s'escriu un zero o un u al bit de control del senyal de reconeixement que és el cinquè bit menys significatiu del registre de control dos, i després s'escriu un u sobre el quart bit més significatiu que és l'habilitació de de la seqüència. Per tant, amb el cinquè bit del registre SSP1CON2, s'escriu si el reconeixement és u o zero, i amb el quart s'habilita

l'operació. Com s'ha descrit anteriorment, quan no es volen rebre més dades de l'esclau, el bit de reconeixement ha d'estar a nivell alt, i si es volen continuar rebent dades, el bit de reconeixement ha de ser zero.

Per tant, quan el mestre treballa en mode escriptura, primer s'aplica la funció del bit d'inici, seguidament s'envien els bytes que es volen escriure a l'esclau, i finalment es realitza la funció el bit de parada. En el codi adjuntat es mostra un exemple.

```
I2C_Master_Start();
//S'envia l'adreça.
I2C_Master_Write(0x20);//Adreça 0x10 + LSB=0
//7 bit address + Write
//Indica que s'ha fet la maniobra de l'Esclau 3, és a dir, frenat al màxim:
I2C_Master_Write(0xE3);
I2C_Master_Stop();
```

En el cas de la lectura, després d'enviar la funció d'inici, s'executa una funció d'escriptura per enviar l'adreça de l'esclau amb el bit menys significatiu a nivell alt. Seguidament, amb la funció de recepció de dades cada vegada que es vulguin llegir paquets de dades, la variable de control del senyal de reconeixement es deixa a u, i un cop es sàpiga que es rep l'última dada, es passa el paràmetre de la funció a zero, i s'envia el bit de parada.

```
I2C_Master_Start();           //Start condition
I2C_Master_Write(0x31);      //7 bit address + Read
//7 bit address + Write
uint8_t Lectura2 = I2C_Master_Read(1); //Read + Acknowledge
uint8_t Lectura1 = I2C_Master_Read(0); //Read + Acknowledge
I2C_Master_Stop();          //Stop condition
PotenciaGenerada=(Lectura2<<8) |Lectura1;
```

En els casos que el sistema té un error de comunicació, degut a que el sistema està pendent de rebre el senyal de reconeixement o a buidar el buffer, a través d'un bucle, el sistema queda encallat i el Watchdog reinicia el sistema.

### 6.1.3. Mode esclau

El mòdul MSSP2 s'ha configurat que treballi com a esclau d'I2C. Per tant, per treballar amb aquest mode, se li ha de configurar una adreça i ha d'estar preparat per rebre dades. En el



codi referent al control del mode esclau, només s'han preparat dues funcions: la funció d'inicialització i la funció que permet llegir i escriure el buffer del MSSP2. Aquesta segona funció es realitza en una rutina d'interrupció.

Per inicialitzar el mode esclau, primer de tot, a través del registre SSP2STAT s'habilita el bit més significatiu perquè pugui rebre dades a 400 kHz. A continuació, en el registre SSP2ADD se li escriu l'adreça de l'esclau, a diferència del mode mestre que en aquest registre se li configura el divisor de freqüència per tenir el senyal de rellotge. L'adreça assignada és el codi 0x28. Seguidament, en el registre de configuració SSP2CON1, s'habilita perquè funcioni com a esclau, donant un valor de sis en decimal en els quatre bits menys significatius perquè l'MSSP2 treballi com a esclau amb una adreça de set bits, i en el quart i cinquè bit menys significatiu se'ls assigna un nivell lògic alt perquè el PIC permeti el funcionament del rellotge del mestre, i habiliti el port del MSSP2 per funcionar. Finalment, en el registre de configuració dos, es deixen tots els bits de control del bus inactius. Seguidament, s'habiliten els pins del canal de dades i el de rellotge, que en aquest cas el pin RB2 se li assigna el canal de dades i el pin RB1 el senyal de rellotge. I finalment, s'habilita la interrupció del MSSP2. El motiu d'habilitar les interrupcions del MSSP2 és que cada vegada que arribi un byte en el buffer, el sistema fa una interrupció. La bandera d'interrupció utilitzada és el SSP2IF i és la que s'utilitza per fer la funció de l'esclau.

A nivell de maquinari, quan en el dispositiu d'esclau li arriba un byte, aquest queda guardat en un shift register d'entrada. Aquest registre per programa no és accessible, i s'anomena SSPSR. Després del bit d'inici, el primer byte passa per aquest shift register i es compara amb l'adreça emmagatzemada en el registre SSP2ADD, a través del registre SSP2MSK. Aquest últim registre mencionat, copia l'adreça assignada en els set bits més significatius, i deixa el bit menys significatiu per comparar l'últim bit del primer byte que ha arribat per indicar si el mestre vol fer l'operació de lectura o escriptura. Si l'adreça que ha arribat en l'esclau coincideix amb el registre SSP2MSK, es produeix la concordança d'adreces, i l'adreça rebuda es transfereix en el buffer del MSSP2. En cas de ser una petició d'escriptura, quan es dona aquesta concordança d'adreces, tots els bytes que arribin fins que no es detecti el bit de parada del mestre, es transfereixen del shift register d'entrada cap al buffer. Així doncs, tal com s'ha mencionat anteriorment, per a cada byte rebut en el buffer, la bandera d'interrupció SSP2IF es posa a nivell alt, permetent una rutina de software per tractar les dades emmagatzemades. Quan es troba en el mode lectura, el byte d'adreça que arriba, també es guarda en el buffer i s'executa la interrupció, però com que llavors els següents bytes els transmet l'esclau, l'operació passa a ser una mica diferent, ja que els bytes que omplen el

buffer no provenen del bus, sinó dels registres interns del pic. D'aquesta manera, cada vegada que el sistema rep una adreça que no correspon a la de l'esclau, l'ignora i no es duu a terme cap interrupció, l'esclau no se n'adona dels missatges rebuts en el shift register, quedant el sistema optimitzat, ja que no s'ha d'estar pendent del bus de manera constant, i els missatges dirigits cap a un altre esclau en la xarxa multimestre són ignorats.

El codi adjuntat és la funció de l'esclau d'I2C, que s'executa quan es dona la interrupció que una dada ha entrat en el buffer. Dins d'aquesta funció, s'assegura que la bandera estigui a nivell alt i que no s'hagi donat un error. El primer que es duu a terme és parar el rellotge, i posar-lo a nivell baix. En posar el rellotge a nivell baix, mentre l'esclau llegeix o escriu el buffer, permet que es doni temps a configurar les dades idòniament, evitant que es produeixi un desbordament en el buffer. Un cop parat el rellotge, aquesta interrupció tracta les dades de tres maneres diferents.

El primer cas que es dona és que un dels dos bits més significatius del registre de control dos té un valor lògic alt, significat que hi ha un desbordament en el buffer, és a dir, que l'esclau rep un byte del mestre mentre el buffer encara manté el byte anterior; o que hi ha una col·lisió del bus, que succeeix quan el buffer s'està escrivint mentre s'està transmetent el valor previ. Donada una de les dues situacions, l'esclau llegeix el buffer, posa a zero els bits de control del desbordament i la col·lisió, i reactiva el senyal de rellotge.

```
if ((SSP2CON1bits.SSPOV) || (SSP2CON1bits.WCOL)){
    lecturaI2C = SSP2BUF; //llegir buffer
    SSP2CON1bits.SSPOV = 0; // Es borra el bit d'OVERFLOW
    SSP2CON1bits.WCOL = 0; // Es borra el bit de col·lisió.
    SSP2CON1bits.CKP = 1; //Es reinicialitza el rellotge.
```

La segona situació, que es tracta en aquesta rutina, és que s'ha rebut l'adreça de l'esclau correctament, i el bit menys significatiu està a zero, significat que el sistema està en mode escriptura. En aquest cas, per a cada byte rebut s'habilita la seqüència del senyal de reconeixement, es llegeix el buffer i es reprèn el senyal de rellotge. De la manera com s'ha plantejat la xarxa multimestre, els codis d'emergència i de test de la xarxa que s'envien són d'un byte. A efectes de programació, la funció està preparada per rebre dos bytes, el primer és l'adreça i el segon el codi de la funció que ha de realitzar l'esclau.

```
if(!SSP2STATbits.D_nA && !SSP2STATbits.R_nW){
    SSP2CON2bits.ACKEN = 1; //Tant bon punt fa l'acknowledge es llegeix el buffer.
```

```
lecturaI2C = SSP2BUF; //Cada vegada que es llegeix el buffer, es borra.
SSP2CON1bits.CKP = 1; //Es reinicialitza el rellotge.
```

Altrament, dins d'aquesta funció, el valor del buffer es guarda en la variable `lecturaI2C`, i en funció d'aquest valor s'assignen variables de control, que condicionen el funcionament del programa. L'assignació de les variables de control s'assignen a través d'un switch-case. En aquesta acció del programa, si es rep la comanda d'emergència o error, el programa fa l'aturada d'emergència o error respectivament.

```
case 0xEE: //COMANDA D'EMERGÈNCIA
    Emergencia=true;
    break;
case 0x1E://Error detectat per l'Esclau 1.
    Error=true;
    break;
case 0x2E://Error detectat per l'Esclau 1.
    Error=true;
    break;
```

De la mateixa, si rep la comanda de test, a través de la variable `escripturaI2C`, que més endavant es comentarà, permet que en cas de petició de lectura, retornar la comanda de test. El byte amb valor `0xBB` és la comanda de test, i és la mateixa que s'usen tots els dispositius per a testejar la xarxa.

```
case 0xBB://COMANDA DE TEST
    escripturaI2C=0xE3;
    break;
```

També pot percebre la detecció de generació o no generació dels altres dispositius de la turbina per posar-se en mode baix consum.

```
case 0xC0:
    GENERACIO_E1=false;
    break;
case 0xC1:
    GENERACIO_E1=true;
    break;
```

Finalment, en aquest switch-case també es rep la comanda d'angle màxim, que s'utilitza en les maniobres d'error i d'emergència.

```
case 0x1F:
    Rebut_Angle_Maxim=true;
break;
```

La tercera situació és que l'adreça s'ha rebut correctament, i el bit menys significatiu està nivell alt. Igual que s'ha comentat en el paràgraf anterior, quan el mestre vol fer una lectura, es retorna només un byte. Quan s'entra en aquesta situació, en rebre l'adreça, s'habilita el senyal de reconeixement, es llegeix el buffer, que només hi ha escrita l'adreça, es posa a zero el bit de buffer ple, per evitar que el programa interpreti situacions errònies, s'omple el buffer amb el byte que es vulgui enviar al mestre, es reprèn el rellotge, i el programa queda encallat en un bucle, que no en surt fins que no s'ha transmès la dada del byte. Seguidament, es borra la bandera d'interrupció, i es posa en valor cert un booleà que indica que el perifèric MSSP2 està funcionant com a esclau.

```
SSP2CON2bits.ACKEN = 1; //Fa la seqüència de l'acknowledge
lecturaI2C = SSP2BUF;
SSP2STATbits.BF = 0; //Es borra el buffer per ser escrit.
SSP2BUF = escripturaI2C; //S'escriu el byte
SSP2CON1bits.CKP = 1; //Es reinicialitza el rellotge.
while(SSP2STATbits.BF); //Fins que no s'hagi enviat el byte
```

La distinció de lectura i escriptura es realitza a través del tercer bit menys significatiu del registre SSP2STAT. Aquest bit s'escriu per maquinari, quan el byte d'entrada coincideix amb l'adreça, amb el registre SSP2MSK i es separa el bit menys significatiu per interpretar si s'està realitzant una lectura o una escriptura. D'aquesta manera, quan aquest bit està a nivell alt, el segon bit menys significatiu del registre SSP2STAT, també es posa nivell alt, així doncs, aquest bit és accessible per programa i permet diferenciar les dues situacions que presenta l'esclau.

Com que només s'ha de rebre un byte de la comunicació entre els dispositius de control de la xarxa multimestre, es fa la lectura del sisè bit menys significatiu del registre SSP2STAT. Si aquest bit està a nivell baix, indica que l'últim byte rebut és l'adreça de l'esclau, i si està nivell alt és una dada. Inicialment, aquest bit està a nivell baix, significant que ha de rebre una adreça. Quan l'esclau rep l'adreça, aquest bit segueix a nivell baix, perquè indica que s'ha rebut una adreça i no una dada, llavors si en la funció entra en mode escriptura es llegeix el

buffer, que hi ha l'adreça, i el bit segueix a nivell baix, posteriorment, quan entra una dada, en trobar-se a nivell baix aquest bit, es llegeix la dada en el buffer i es posa el bit a nivell alt, indicant que en el buffer hi ha arribat una dada. Com que el mestre ja ha previst que només s'envia una dada, s'envia el no reconeixement, i el mestre envia el bit d'aturada. Si s'entra en el mode lectura, es llegeix l'adreça en el buffer, s'escriu la dada que es vol retornar, i en escriure la dada, el bit del registre SSP2STAT, es posa a nivell alt. Per tant, amb aquests dos bits d'estat mencionats es controla la rutina d'esclau d'I2C.

#### 6.1.4. Test de comunicacions I2C

En la Figura 47, es mostra l'organigrama del procediment del test.

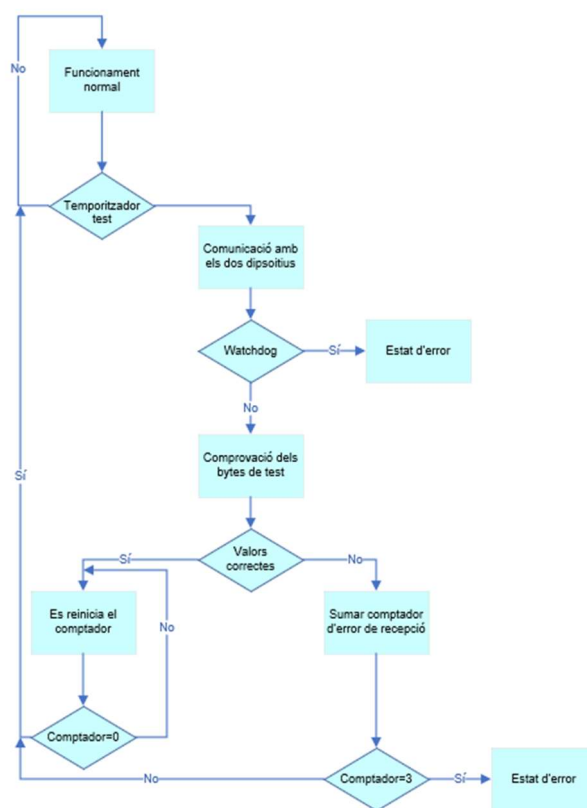


Figura 47. Organigrama del test de la xarxa d'I2C multimestre.

La xarxa multimestre ha de poder funcionar de forma robusta davant d'una situació d'emergència en l'aerogenerador. D'aquesta manera, si en el mode test es detecta una fallada de comunicació dels dispositius de control, el sistema es posa en estat d'error, parant la generació, i roman a l'espera fins que la xarxa queda arreglada. El test es duu a terme en una interrupció temporitzada de quatre segons, on el sistema es comunica amb els altres dos

dispositius de control de la turbina. Si el sistema no rep la comunicació, queda encallat en un bucle infinit, que en donar-se el desbordament del Watchdog, es sap que hi ha un error de maquinari, i per tant, el sistema es fica en estat d'error, indicant en el PLC que s'ha de revisar el sistema. Si els codis es reben, però els seus valors no es reben de manera correcta durant tres vegades seguides, es tradueix que hi ha algun error que pot ser degut al soroll o a interferències, i que un tècnic ho ha de tractar. Donada la casuística, el sistema també es posa en estat d'error, i no es rearma fins que no s'ha revisat.

Com que la interrupció temporitzada es duu a terme cada quatre segons, i la xarxa d'I2C funciona a 400 kHz, el marge temporal és més que suficient per fer les comunicacions dins de la rutina d'interrupció. El sistema de control de potència, primer es comunica amb cada dispositiu amb el codi de test, perquè els altres dispositius preparin el buffer amb el seu respectiu byte de test. I seguidament, es fa la lectura dels buffers d'esclau de cada dispositiu per rebre el byte de test, i així comprovar que la xarxa funciona. Per tal de donar temps a omplir els buffers de cada dispositiu, les accions d'escriptura i lectura es duen a terme de manera intercalada.

La funció que comprova que els missatges s'han rebut de forma incorrecte durant tres vegades seguides retorna un booleà. Aquesta funció es crida des de bucle de la funció main(), en la part del codi de funcionament normal, on no hi ha error ni emergència, i s'escriu directament en la bandera d'error del programa, perquè en la següent volta del bucle posi el sistema en estat d'error. En aquest cas, es passen dos paràmetres, que són els dos bytes rebuts per part dels esclaus, i es comprova si s'han rebut de forma incorrecte durant tres vegades consecutives.

```
if(HiHaTestComunicacioI2C){
    //En cas d'error es posa en true, l'error i el bit d'error del test.
    Error_TEST_I2C=errorTESTI2C(Test_ESCLAU1,Test_ESCLAU2);
    Error=Error_TEST_I2C;
    HiHaTestComunicacioI2C=false;}

```

Un cop comprovat si hi ha hagut un error, la variable booleana que indica que s'ha realitzat el test de comunicació se li assigna de nou un valor zero.

### 6.1.5. Comunicació I2C amb el sistema de mesura de potència generada

Primer de tot, en el codi del microprocessador ESP32 se li adjunten directives que li indiquen en el preprocessador la definició dels pins de l'I2C, que en aquest cas el pin 21 és pel canal

de dades, i el pin 22 pel senyal de rellotge. A més a més, es defineix l'adreça, que en el cas de l'I2C són els set bits més significatius d'un byte, ja que el bit menys significatiu es deixa per definir el mode escriptura o lectura. L'adreça definida és el byte de 0x18, que en el cas d'escriptura a causa del desplaçament d'un bit a la dreta, el byte que rep el dispositiu del mestre el seu valor és 0x30, i en el cas de lectura rep el 0x31.

```
#define SDA_PIN 21
#define SCL_PIN 22
#define I2C_SLAVE_ADDR 0x18//L'adreça de l'ESP32 treballant com a Esclau
```

La interacció de l'I2C en el microprocessador ESP32 és molt senzilla. Simplement, quan es rep l'adreça amb el mode d'escriptura, l'esclau rep un byte, que en el seu tractament, el guarda en una variable anomenada "Producció", que para l'adquisició de dades, posant el dispositiu en mode deep-sleep, en cas que aquest valgui el valor 0x3C, donant un control total de l'adquisició al mestre. En el mode lectura, l'esclau retorna el valor de potència aparent obtingut, que com que s'ha compensat la reactiva, la potència aparent equival a la potència activa. Com que els datagrames de l'I2C són d'un byte, i el valor de la potència és una variable entera sense signe de setze bits, el dispositiu primer envia els vuit bits més significatius, i després els menys significatius.

```
uint32_t P=0; /*Per inicialitzar l'I2C a la ESP32, s'ha d'omplir el buffer de
recepció/transmissió.*/
//S'envia el valor de potència
uint16_t potencia=0;
byte MSBPotencia=0;
byte LSBPotencia=0;
void onRequest(){ //Funció de la petició de lectura.
/*L'integer que guarda la potència es divideix entre dos i s'envia per separat.*/
  Wire.write(MSBPotencia);
  Wire.write(LSBPotencia);}
char Produccio;
void onReceive(int len){//Funció en mode escriptura.
  while(Wire.available()){
    Produccio=Wire.read(); //Byte de parar la proucció.}}
```

Dins de la funció setup(), s'inicialitza l'I2C: definint el pin del canal de dades, el del rellotge, i l'adreça; i s'assignen les rutines de lectura i escriptura en el vector d'interrupció del protocol de comunicació, a partir de les funcions incloses en la biblioteca Wire.h: Wire.onReceive() i Wire.onRequest(). En conseqüència, s'enllaça la funció que s'ha de dur a terme en cas d'una

lectura, o una escriptura de l'I2C. En el setup(), també es fa la inicialització del buffer de l'I2C, ja que en utilitzar-se un dispositiu ESP32 s'ha d'aplicar la funció slaveWrite() per escriure prèviament al buffer de resposta de l'esclau, donant-li un valor. Aquesta acció afegeix la capacitat d'esclau al microprocessador, i permet mantenir la compatibilitat amb Arduino. Sense aquesta maniobra el programa de l'I2C no funciona.

```
bool comprovar= Wire.begin(I2C_SLAVE_ADDR,SDA_PIN,SCL_PIN,400000);
Wire.onReceive(onReceive);//Mode escriptura
Wire.onRequest(onRequest);//Mode lectura
#if CONFIG_IDF_TARGET_ESP32//S'emplena el buffer per transmetre en el cas de la ESP32.
  char message[64];
  snprintf(message, 64, "%u Packets.", P++);
  Wire.slaveWrite((uint8_t *)message, strlen(message));
#endif
```

## 6.2. Xarxa del Modbus

A través de la xarxa Modbus, el PLC es pot adreçar als dispositius de control i monitorització de la turbina per testejar el seu funcionament, i llegir o escriure els valors de les seves variables de control i de sensorització.

El Modbus és un protocol de comunicació serial asíncrona simple i robust, que el fa ideal per a tota mena d'aplicacions industrials. Com que és un protocol estàndard, es poden establir comunicacions entre diferents dispositius electrònics, com és el cas d'aquest aerogenerador, que la comunicació és entre microcontroladors PIC, treballant com a esclaus, i un PLC actuant com a mestre. Concretament, el protocol de Modbus que s'utilitza és el Modbus RTU, que en la Figura 48, es mostra com són les trames de Modbus.

### **TRAMA RTU**

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	8 Bits	8 Bits	n*8 Bits	16 Bits	T1-T2-T3-T4

Figura 48. Trama de Modbus RTU.

Per aquest projecte s'ha adequat la funció de l'esclau de Modbus amb el programa del sistema de control de la turbina, d'aquesta manera, el sistema pot respondre en cas de petició del PLC, a la vegada que permet que el PLC llegeixi i escrigui totes les variables del sistema:



siguin variables de control, els valors dels sensors que monitoritzen el sistema, i l'estat de les sortides del PIC.

El present apartat es divideix en tres subapartats: el primer explica com s'ha integrat la rutina d'esclau en el sistema multitasca, en el segon s'expliquen les funcions de Modbus que intervenen en el PIC, i l'últim subapartat es detallen les variables que poden ser llegides i escrites. En el cas de l'escriptura de variables cal distingir el mode test i el mode monitorització.

### **6.2.1. Integració de la rutina d'esclau de Modbus**

El Modbus és un protocol asíncron, per tant, s'ha de configurar a través de la UART del microcontrolador PIC. Aquest fet no és una limitació perquè el PIC incorpora dos mòduls EUSART, que es poden configurar que funcionin de manera asíncrona, d'aquesta manera es parametriza una UART per poder-se comunicar amb els diferents dispositius.

Cada vegada que s'omple el buffer, el sistema s'interromp, així doncs, aprofitant la rutina d'interrupció, s'espera rebre una seqüència de bytes. Si el primer byte de la seqüència no coincideix amb l'adreça de l'esclau, cada vegada que s'interrompi el sistema es llegirà la dada per buidar el buffer, i s'ignorarà al final de la trama corresponent. Com que PLC farà peticions de lectura en temps de l'ordre de segons, quan es vol llegir l'estat dels tres dispositius de la turbina, el buffer de recepció tardarà com a mínim 833 microsegons en omplir-se, ja que la velocitat de funcionament és de 9.600 bps. Durant aquest temps previ a omplir-se el buffer, com que l'I2C treballa a 400 kHz i la freqüència de processament del dispositiu és molt ràpida, el sistema podrà executar diferents tasques. Gràcies a les funcions de la llibreria Modbus i les pròpies de la EUSART, si les dades que envia el PLC pel bus van dirigides al dispositiu de control de potència consumida, es pot sincronitzar la maniobra d'esclau de Modbus, de la mateixa manera que si la dada va dirigida a un altre esclau, es pot ignorar la petició.

### **6.2.2. Funcions del Modbus**

L'EUSART s'ha parametrizat perquè cada vegada que entra un byte en el buffer de recepció, executa una interrupció. Per tant, cal habilitar les interrupcions de la EUSART, i dins del codi del main(), s'ha de declarar la interrupció del buffer de recepció en el vector d'interrupcions. El Modbus funciona amb el perifèric EUSART 1, per motius deguts al hardware intern al PIC, per funcionar amb la EUSART1, s'ha d'habilitar també la EUSART 2.

Cada vegada que es produeix aquesta interrupció, el mòdul convertidor de UART a RS485 resta configurat en mode recepció, i es buida el buffer a través de la funció EUSART\_Read(). Per configurar la recepció de dades en l'adaptador UART a RS485, el pin de sortida RA5 es posa a nivell baix.

Dins de la interrupció, en funció de la dada rebuda en el buffer i el valor del comptador de control de dades rebudes, es fa un control del Modbus a través d'un switch-case. Quan en el buffer li entra una dada i el comptador està a zero, per programa es compara aquesta dada amb l'adreça de l'esclau, que en aquest cas, el sistema de control de potència és l'esclau tres. Si la dada que ha entrat coincideix amb la seqüència, es guarda aquest byte en un vector anomenat RxBuffer[], s'inicia l'emmagatzematge de dades que provenen del bus, i s'incrementa el comptador. A la següent interrupció, el comptador té un valor de u i s'analitza si les dades rebudes coincideixen amb la funció del Modbus. Les funcions de Modbus són estàndards i venen definides pel protocol. En el cas del sistema de control de potència, només s'executen dues funcions, la de llegir registres que té el valor 0x04 i la d'escriure múltiples registres que és el 0x10. Si la segona paraula rebuda no coincideix amb cap dels valors anteriors, el comptador es torna a posar a zero i s'ignoraran els següents bytes rebuts, si no coincideixen amb l'adreça. Si es rep la funció de llegir registres s'adquireixen sis bytes més, i s'entra dins de la funció modbus\_function(), on es tracten els bytes rebuts i es dona la resposta al PLC. En aquest cas, després d'haver rebut el byte de la funció, el sistema rep el d'adreça del registre que es vol llegir. Aquesta adreça és de setze bits, per tant es divideix en el tercer byte rebut i en el quart. Seguidament, el següent byte que li arriba és la quantitat de bytes que es volen llegir de cada registre, i finalment en el sisè i setè bytes que rep el CRC, que els la comprovació d'Errors. Un cop rebut el setè byte, el PIC entra al funció del modbus, i selecciona la funció de llegir registres.

```
if(RxBuffer[0] == SlaveAddress){//COMPARA ADREÇA
switch (RxBuffer[1]){//COMPARA FUNCIO
case READ_INPUT_REGISTERS:
modbus_read_input_registers(RxBuffer[4]);//Dada a llegir pel PLC
break;
case WRITE_MULTIPLE_REGISTERS://Escriu en el PIC
modbus_write_multiple_registers();
break;
default:

error = ILLEGAL_FUNCTION;
frame = 0;
break;
}}
c_control=0;
```

En aquesta situació el programa entra a una funció anomenada `modbus_read_input_registers()`, on se li passen el nombre de bytes que es volen llegir de cada registre com a paràmetre. Dins d'aquesta funció es comprova el CRC que no hi hagi cap error. Si el missatge és correcte, el sistema retorna la resposta amb un trama que consta: en el primer byte és l'adreça de l'esclau, el segon és la funció de Modbus, el tercer amb l'adreça del registre que s'ha llegit, el quart i cinquè byte hi ha el valor de les dades que guarda el registre dividit amb els bits més i menys significatius respectivament, i en el sisè i setè hi ha el CRC. La majoria de registres emmagatzemen dades formades per dos bytes, per tant el byte que indica el nombre de dades sempre serà dos, inclòs quan només hi ha un byte emmagatzemat, ja que en aquesta situació els bits més significatius seran zero i els menys significatius, el valor del byte que es llegeixi. El control de les adreces que es volen llegir, es duu a terme a través d'un switch-case de la variable de l'adreça del registre. Aquesta variable d'adreça s'ha creat unint els dos bytes d'adreça de registre.

La trama que es vol enviar cap al PLC, s'ha format a través d'un vector anomenat `response[]`. En cada posició d'aquest vector se li ha emmagatzemat cada byte que es vol enviar. En el cas de les dades que es volen llegir de cada registre, les variables de setze bits, queden separades en dos bytes diferents. Seguidament, a l'hora de transmetre aquesta resposta, s'ha d'habilitar la passarel·la perquè transmeti, posant a nivell alt el pin RA5, i a través de la funció `EUSART_Write()` en un bucle for, s'escriu el buffer de transmissió. Un cop enviades les dades, es posa a zero el pin de transmissió de la passarel·la, es retorna a la funció `modbus_function()`, i es reinicia el comptador de control de les dades rebudes a zero.

```
response[0] = SlaveAddress;
response[1] = READ_INPUT_REGISTERS;
response[2] = 2 * reg_num;
response[3] = inputregisterpart1;
response[4] = inputregisterpart2;
crc16 = MODBUS_CRC16_TX(5);
response[6] = CRC8_Hi(crc16);
response[5] = CRC8_Lo(crc16);
R_T_RS485_PORT = 1;
for (uint8_t jk = 0; jk <= 6; jk++){
EUSART1_Write(response[jk]);}
R_T_RS485_PORT=0; //Desactivem el bit de transmissió de dades
```

Si la funció rebuda és la d'escriure múltiples registres, el sistema es prepara per llegir 14 dades més. El fet de llegir més o menys dades, queda controlat en el cas set del switch-case, a través d'un condicional que compara la funció a partir del segon byte rebut. Si aquest byte és el de la funció d'escriure múltiples registres, en el cas set del switch-case només s'adquireix la dada i es guarda en el vector `RxBuffer`. Si la funció és la de llegir registres, s'entra a la funció `modbus` i es fa el tractament de dades. Així doncs, en el cas d'escriptura, un cop s'han llegit

totes les dades, s'entra a la funció de `modbus_function()`, que gràcies als paràmetres assignats a través de condicionals, executa la funció de `modbus_write_multiple_registers()`.

En la funció d'escriptura es distingeixen els dos modes d'escriptura de dades del PLC: el mode test i el mode monitorització. La trama en els dos casos és la mateixa, però només varia en quins bytes de la seqüència s'escriu el valor del registre que es vol modificar des del PLC. La trama de Modbus en el cas d'escriptura està formada per 15 bytes: el primer correspon a l'adreça de l'esclau, en el segon byte hi ha la funció de Modbus, en el tercer i quart hi ha l'adreça que es vol escriure, que és de setze bits; en el cinquè el nombre de bytes que es vol escriure en cada registre, que en aquest cas, sempre serà dos. Seguidament es reben els valors dels modes test, i els modes de monitorització. Si el sistema està en mode monitorització, els dos bytes pertinents als valors de test estaran a zero, que són el desè i l'onzè byte, i es llegiran els bytes corresponents als valors de monitorització, que són el vuitè i el novè. En el mode test serà el cas contrari. Altrament, en el mode test es poden escriure fins a quatre bytes, per aquest motiu s'envien el dotzè i tretzè byte. Finalment, en els dos últims registres hi ha el CRC.

Seguidament, si la funció que comprova el CRC retorna un valor correcte, el sistema retorna una trama al PLC que ha rebut de forma correcta les dades. En aquest cas, la trama de resposta també es retorna seguint el mateix procés que en el cas de lectura, però en aquesta situació només s'envien nou bytes que s'escriuen en el vector `response[]`. Aquest bytes seran: un byte per l'adreça de l'esclau, un byte per la funció de Modbus, els dos bytes d'adreça del registre, el dos bytes del valor que se li vol escriure al registre, i els dos bytes del CRC.

```
response[0] = SlaveAddress;
response[1] = WRITE_MULTIPLE_REGISTERS;
response[2] = starting_address_Hi;
response[3] = starting_address_Lo;
response[4] = reg_num_Hi;
response[5] = reg_num_Lo;
crc16 = MODBUS_CRC16_TX(6);
response[7] = CRC8_Hi(crc16);
response[6] = CRC8_Lo(crc16);
```

Un cop enviada la trama, s'escriu el valor del registre que ha enviat el PLC. Per seleccionar el registre a escriure, s'han ajuntat els dos bytes de l'adreça de registre, i a través d'un `switch-case`, es dona un el nou valor a la variable que es vol modificar des del PLC. En el sistema de control de potència, el mode monitorització només es dona quan es vol escriure el registre amb valor tres, la resta de registres són de test, per testejar les sortides del microcontrolador

o canviar els llindars de les variables de control en el cas d'una posada en funcionament o accions de manteniment.

Quan el sistema acaba la funció `modbus_function()`, o al principi de la trama es rep una adreça o una funció errònia, el comptador de control de Modbus de la funció de la interrupció del buffer de recepció es torna a posar a zero, i les variables que es van rebent de nou en cas que coincideixi l'adreça i la funció es van escrivint en el vector `RxBuffer[]`.

Cada vegada que s'entra a una de les dues funcions del Modbus es comproven els bytes del CRC, a través d'una funció de `checkCRC()`. Aquesta compara el CRC rebut amb el valor de CRC calculat a través de la funció `MODBUS_CRC16_RX()`. En aquesta funció se li passen els bytes que s'han llegit, i calcula el CRC en funció del vector `RxBuffer[]`. Aquesta funció també calcula el CRC per la trama de resposta en el vector `response[]`.

Quan en el Modbus es produeix un error, sigui perquè s'ha enviat una adreça de registre que no existeix, o que el CRC no coincideix, el sistema entra a la funció `error_function()`. En aquest cas, l'esclau respon al PLC mestre que ha rebut dades incoherents, a través d'una trama d'error que s'escriu a través del vector `response[]`.

```
response[0] = SlaveAddress;
response[1] = 0x80 + READ_INPUT_REGISTERS;
response[2] = error;
response[4] = CRC8_Hi(crc16);
response[3] = CRC8_Lo(crc16);
```

Les funcions pròpies del Modbus s'han dut a terme en una llibreria anomenada `modbus.c`. Aquesta llibreria combina funcions de l'EUSART creades a l'MCC, i variables que s'han declarat en l'arxiu `main.c`. Per poder interaccionar amb les variables de control i les dels valors dels sensors, cal declarar les variables com a externes, a l'arxiu de capçalera de la llibreria de Modbus. És a dir, en l'arxiu `modbus.h`. El fet de declarar variables externes significa que la variable es declara fora de la llibreria, però que s'utilitza en aquesta.

### 6.2.3. Variables d'interacció amb el PLC

El PLC pot monitorar el sistema de control de potència a través de la lectura de registres, i a la vegada pot escriure sobre diferents variables, segons si està en mode test o en mode monitorització.

Moltes variables, com les variables de control i els llindars de commutació, que es poden escriure en el mode test, també poden ser llegides, perquè d'aquesta manera l'encarregat de manteniment o de posada en funcionament del molí sap el seu valor. En els següents paràgrafs s'inclouen taules on es comenten les variables que es poden llegir o escriure des del PLC.

Algunes variables que llegeix el PLC i que permeten conèixer l'estat del sistema són booleans. Aquestes variables booleans, s'han unit en un únic registre. I llavors el PLC per programa tractarà cada bit del byte adquirit per separat. El registre u uneix els diferents bytes d'estat del sistema, de la mateixa manera que en el registre nou, es llegeixen els bits que informen del tipus d'error que s'ha donat. En el codi adjuntat es mostra com s'uneixen els bits d'estat.

```
ESTAT_SISTEMA_MODBUS=(10<<K_GEN_ST_PORT) | (9<<K_GEN_RS_PORT) |
    (8<<K_FOCUS_PORT) | (7<<K_BOMBA_PORT) | (6<<Nivell_diposits)
    | (5<<ESTAT_AMB_GENERACIÓ) | (4<<MANIOBRA_FETA) |
    (3<<POSSIBLES_DANYS_RELES_GENERADOR) | (2<<REVISIO_SISTEMA)
    | (1<<AvisErrorPLC) | AvisEmergenciaPLC;
inputregister = ESTAT_SISTEMA_MODBUS;
```

En la Taula 34, hi ha les variables que són únicament de lectura des del PLC.

Variable	Registre	Descripció
Pinversor_W	3	Potència mesurada de l'inversor
VBateria	6	Tensió de la bateria
Pbateria_W	4	Potència de la bateria (pot ser d'entrada o sortida i depèn del valor del corrent)
Intensitat_bateria	5	Corrent de la bateria. Si el corrent és positiu, entra a la bateria, i si és negatiu la bateria l'aporta.
PotenciaGenerada	2	Potència generada
PFRENAT_MESURADA	7	Potència que s'està dissipant en el frenat.
Pfrenat_W	8	Potència de l'acumulador del frenat.

Taula 34. Variables únicament de lectura.

En la Taula 35, hi han els valors que poden ser llegits i escrits des del PLC. Aquests són llindars de control o sortides del PIC directament, per fer tests del sistema o adequar els llindars per una posada en funcionament, o per accions de manteniment.

Variable	Valor inicial	Registre	Descripció
Smin (W)	240	6	Potència mínima de commutació.
SB_Sllums (W)	1.200	7	Suma de les potències dels llums i la bomba.
Sllums (W)	400	8	Potència dels llums.
SB (W)	800	9	Potència bomba.
Vllums (V)	25	10	Tensió de la bateria per poder connectar els focus.
Vbaix (V)	23	11	Tensió mínima de càrrega de la bateria. És la limitació de la profunditat de descàrrega al 70% de la seva capacitat.
Sllindar (W)	200	12	Potència de generació llindar pels focus.
Plimit_bateria (W)	960	13	Potència de càrrega límit de la bateria.
Vbateria_carregada (V)	27	14	Tensió de la bateria carregada.
FRENAT_PLC (W)	0	5	Unsigned integer per escriure el valor de la potència de frenat que es vol.
MARGE_ERROR (W)	12	15	Marge d'error dels sensors.
Potencia_limit_reles (W)	600	16	Variable que indica la potència límit dels relés.
Potencia_limit (W)	2.200	17	Variable que indica la potència límit generada.
ZONA_MORTA_SENSORS_DE_CORRENT (enter)	6	18	Zona morta dels sensors de corrent de la bateria.
Rendiment (Sobre deu)	9	19	Rendiment del sistema amb una escala de 0 a 10.

Taula 35. Variables de lectura i escriptura.

De la mateixa manera que s'ha plantejat en la lectura de dades, alguns registres incorporen variables que consisteixen en la unió de diferents variables booleanes, com pot ser el registre tres, corresponent en el mode automàtic, que escriu sobre les variables que poden rearmar el sistema. Aquestes variables són les de la Taula 36.

Variable	Descripció
REARMAMENT	Booleà que permet el rearmament del PIC.
REVISIO_SISTEMA	Booleà que avisa si cal revisar el sistema, i que s'escriu a nivell baix si s'ha revisat.
POSSIBLES_DANYS_RELES_GENERADOR	Booleà que avisa de possibles danys als relés i/o al generador. En aquest cas s'ha d'escriure a nivell baix per indicar que s'han revisat els relés.

Taula 36. Variables de rearmament del sistema.

Les variables de rearmament, que s'utilitzen en les aturades d'error i d'emergència i que es comenten en els apartats sis i set del capítol dos, queden unides en una variable d'un byte. Com que aquestes són variables booleanes, s'han de separar. Per aquest motiu, s'ha implementant el codi adjuntat a continuació, on a través d'un bucle for, per a cada iteració s'aplica una màscara en l'últim bit de la variable, i s'emmagatzema el booleà resultant en un vector, i seguidament, la variable original es desplaça un bit a la dreta.

```
VARIABLES_REARMAMENT = auto_data;

    for(int ia=0; ia<3; ia++){

        LECTURA_DEL_uint16_auto[ia]=VARIABLES_REARMAMENT&&0x0001;

        VARIABLES_REARMAMENT=VARIABLES_REARMAMENT>>1;
    }

//Un cop separats els bits en diferents posicions del vector, per separat es tracten:

    REARMAMENT=LECTURA_DEL_uint16_auto[2];

    REVISIO_SISTEMA=LECTURA_DEL_uint16_auto[1];

    POSSIBLES_DANYS_RELES_GENERADOR=LECTURA_DEL_uint16_auto[0];
```

En el registre quatre, que correspon a un registre de test, es duu a terme la mateixa operació, però amb les variables de la Taula 37. Aquestes variables també poden ser llegides. I algunes condicionen el funcionament d'altres variables, com el test del frenat.



<b>Variable</b>	<b>Descripció</b>
Emergencia	Variable que inicia la maniobra d'emergència. Aquest booleà pot ser escrit des del PLC per fer proves de l'emergència.
Error	Variable que inicia la maniobra d'error. Aquest booleà pot ser escrit des del PLC per fer proves de l'error.
Error_WATCHDOG	Aquesta variable també pot ser escrita per fer proves en la maniobra d'error.
K_BOMBA_PORT	Estat del relé de la bomba.
K_FOCUS_PORT	Estat del relé dels focus.
K_GEN_RS_PORT	Estat del relé de curtcircuit del generador (RS).
K_GEN_ST_PORT	Estat del relé de curtcircuit del generador (ST).
Test_frenat_PLC	Si es vol testejar el frenat.

Taula 37. Variables d'estat que pot escriure el PLC en mode test.

Per assegurar que el PLC realment només escriu sobre les variables quan el sistema no es troba ni en emergència ni en error, s'aplica una condició sobre l'escriptura. Si el sistema està en emergència o en error, la variable manté el valor, en cas contrari, es permet l'assignació que doni el PLC.

```
Smin = (Emergencia || Error) ? Smin : test_data;
```

## 7. EXPERIMENTS DEL PROJECTE

Al llarg del desenvolupament del sistema de control de potència consumida, de manera parcial s'han anat provant les diferents rutines del codi. Concretament, s'han fet experiments parcials en les dues xarxa de comunicacions, en la lectura dels sensors, en el frenat elèctric, en el sistema de mesura de potència generada, i en l'algoritme de transferència de potència.

Pel desenvolupament de la PCB, un cop s'ha programat el codi sencer, coneixent que les entrades i les sortides programades funcionen correctament, sense que hi hagin problemes deguts a les connexions internes del PIC, s'ha ajustat la PCB perquè quan arribi no se li hagi de fer cap modificació sobre les pistes.

Gràcies aquests experiments, es sap que el sistema és robust, però en la posada en funcionament, només s'hauran d'acabar d'ajustar certs paràmetres que es poden dur a terme a través del PLC, o si es necessari en un cas extrem, s'hagi de tornar a codificar el codi.

En l'Annex C hi ha el procediments i com s'han desenvolupat els experiments del sistema.

## **8. RESUM DEL PRESSUPOST**

El cinquè document d'aquest projecte és el pressupost que ve seguit de l'Estat d'amidaments. Aquests dos documents estan totalment lligats, ja que el pressupost tot i ser informatiu, s'indiquen els preus de les hores de treball dels diferents treballadors involucrats, i el cost de tot el material implicat.

El cost pressupostat del projecte és de set mil dos-cents vint-i-set euros amb cinquanta-nou cèntims, sense IVA.

## 9. CONCLUSIONS

Inicialment, exceptuant l'alternador i el regulador de càrrega, no hi havia cap element elèctric ni electrònic. A partir de les necessitats de l'ECOgarnja Vilà, s'ha decidit com buscar la millor solució per tal d'aprofitar al màxim possible l'energia del molí, garantint un funcionament robust i a la vegada el més eficient possible. Un cop establert un punt de partida, que determina: que l'aerogenerador treballa amb illa, que l'energia produïda es dedica a omplir dos dipòsits, i addicionalment a encendre uns focus per la granja; s'ha fixat la tecnologia utilitzada per al control del sistema i la gestió de l'energia.

El sistema té una producció de 1.800 W. El sistema controla el consum de potència segons els estats de l'algoritme de control, que depenen dels valors de potència generada, l'estat de les bateries i l'estat dels dipòsits. Aquestes paràmetres es poden determinar a partir de la lectura dels diferents sensors. D'aquesta manera, segons l'estat del sistema es pot: acumular l'energia en un sistema format per dues bateries, accionar la bomba que omple els dipòsits, que és de 800 Watts; davant d'un excedent o si la bomba no es pot connectar perquè els dipòsits estan plens, l'energia es transfereix en uns focus de la granja que consumeixen 600 Watts, i en cas de producció pic, permet inclòs tenir ambdues càrregues funcionant a la vegada i carregar la bateria. Altrament, el sistema dissenyant, en cas de diagnosticar-se una emergència, el sistema atura la producció de manera immediata. També, gràcies als autodiagnòstics que realitza el sistema, quan apareix un error degut a la mesura dels sensors o en les comunicacions entre els diferents dispositius, el sistema s'atura per evitar danys irreparables. Quan s'ha d'aturar la turbina, sigui per un error o una emergència, s'ha implementat un frenat elèctric, que connectant bancs de làmpades halògenes, actuant com a resistències de frenat, a través de commutadors MOSFET controlats per PWM, permet frenar la turbina. Aquest frenat, en situacions on el sistema transfereix energia sense haver-hi un error o una emergència, actua protegint la bateria, limitant la potència màxima de càrrega, i també, en cas que la bateria estigui carregada, protegeix el sistema igualant la potència generada a la demandada. Per reduir el consum de la turbina, i allargar al màxim l'energia acumulada a la bateria, quan no hi ha generació, el sistema de control de potència es posa en mode baix consum per optimitzar l'energia que utilitza.

El sistema de control de potència es comunica amb els altres dispositius de la turbina a través d'una xarxa multimestre amb bus I2C diferencial, que permet als dispositius treballar de manera independent del dispositiu mestre que és un PLC. El PLC es comunica amb el sistema de control de potència a través d'una xarxa Modbus RTU, i pot fer lectures i escriptures sobre

diferents variables del sistema. En la petició d'escriptura cal distingir el mode automàtic, que es dona quan s'ha de rearmar el sistema després d'una aturada d'error o emergència, i el mode test, que permet fer assajos sobre sortides del dispositiu de control, o canviar els llindars de control dels algoritmes.

Paral·lelament al control dels diferents modes de treball de la turbina, el sistema pot realitzar: la lectura dels sensors juntament amb el càlcul de les diferents potències, la determinació de si hi ha generació per posar-se en baix consum, autodiagnòstics, i el treball del sistema de forma sincronitzada amb les dues xarxes de comunicacions. Així doncs, s'ha implementat un sistema de control multitasca. Per aquest motiu s'ha escollit la tecnologia de control que ofereixen els microcontroladors PIC, que són versàtils, i tenen capacitat de treballar amb interrupcions de forma robusta i precisa. El model escollit és el PIC18F47Q10, que és un microcontrolador de molt baix consum, que incorpora els diferents perifèrics per controlar el sistema: dos MSSP per configurar l'I2C, una EUSART per configurar el Modbus, sis temporitzadors, un convertidor ADCC que se li ha configurat el maquinari perquè obtingui un valor mitjà de ràfega dels senyals llegits, sis CLC amb un generador de PWM per dissenyar el frenat elèctric, l'opció d'una interrupció per flancs de pujada o baixada per monitorar l'estat dels dipòsits, un comptador de Watchdog per controlar l'execució del codi, i diferents pins d'entrades i sortides digitals. A més a més, té la capacitat de posar-se en `sleep()` que permet treballar amb baix consum quan no hi ha generació. Gràcies a la sincronització de les diferents interrupcions, siguin: interrupcions causades per arribades de dades en els buffers de recepció dels perifèrics de comunicació, interrupcions causades pel convertidor analògic-digital, interrupcions temporitzades o interrupcions per pins externs, s'ha obtingut un sistema multitasca.

El dimensionament dels cables i l'elecció colors s'ha fet segons normativa. El quadre elèctric de la instal·lació queda a la base del molí, i s'ha dissenyat perquè sigui el més senzill possible d'instal·lar, i de fer el manteniment en cas d'avaría.

Adrià Quintana Tarradas

Graduat en Enginyeria Electrònica Industrial i Automàtica i en Enginyeria elèctrica

Girona, 22 de maig de 2023

## **10. RELACIÓ DE DOCUMENTS**

El projecte consta de cinc documents, que són: la Memòria, Plànols, Plec de condicions, l'Estat d'amidaments i Pressupost.

## 11. BIBLIOGRAFIA

ABB, Cuaderno Técnico número 12 Plantas eólicas ([https://library.e.abb.com/public/ac764cb1be081128c1257a30003c70d7/Cuaderno%20Tecnico\\_num%2012\\_Plantas%20eolicas.pdf](https://library.e.abb.com/public/ac764cb1be081128c1257a30003c70d7/Cuaderno%20Tecnico_num%2012_Plantas%20eolicas.pdf), 7 de febrer de 2023)

ALLEGRO Microsystems, ACS758 datasheet (<https://pdf1.alldatasheet.es/datasheet-pdf/view/533459/ALLEGRO/ACS758LCB-100B-PFF-T.html>, 12 de gener de 2023)

ALTERNATIVE ENERGY TUTORIALS, Dump and Diversion Loads, (<https://www.alternative-energy-tutorials.com/wind-energy/dump-load.html>, 04 de gener de 2023)

FERNANDEZ SALGADO JOSE MARÍA. Guía completa de la energía eólica 2ª Edición. Sense col·lecció. Paraninfo. Madrid. 2019

FRAILE MORA JESUS. Máquinas eléctricas 8ª Edición. Sense col·lecció. Garceta grupo editorial. Madrid. 2016

INSTRUMENTACION I CONTROL, Sistemas de Frenado eléctrico (<https://instrumentacionycontrol.net/sistemas-de-frenado-motores-trifasicos-asincronos-con-rotor-en-cortocircuito-por-electrofreno/>, 16 de març de 2023)

I2C-bus, I2C – What's That?, (<https://www.i2c-bus.org/>, 10 de gener de 2022)

MICROCHIP, MPLABX software User's guide (<http://ww1.microchip.com/downloads/en/devicedoc/50002027d.pdf>, 17 de novembre de 2022)

MIRCOCHIP, PIC18F27/47Q10 datasheet, (<https://www.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/PIC18F27-47Q10-Data-Sheet-40002043E>, 10 de novembre de 2022)

MODBUS TOOLS, Protocol Description, (<https://www.modbustools.com/modbus.html>, 20 d'abril de 2023)

## 12. GLOSSARI

ADC: Analog to Digital Converter.

ADCC: Analog-to-Digital Converter with Computation.

AGM: Absorbed Glass Mat.

BMS: Battery Management System.

CLC: Cel·la lògica configurable.

CPU: Central Processing Unit.

CRC: Cyclic Redundancy Check.

DIN: Institut Alemany de Normalització.

EUSART: Enhanced Universal Synchronous Receiver Transmitter.

GEL: Gel Electolyte.

ICAEN: Institut Català d'Energia.

IDAE: Institut per la Diversificació i Estalvi de l'Energia.

IP: Índex de protecció.

ITC: Instrucció Tècnica Complementària.

I2C: Inter-Integrated Circuit.

LED: Light Emitting Diode.



MCC: MPLABX Code configurator.

Modbus: Modular Digital Bus.

MOSFET: Metal-oxide-semiconductor Field-effect Transistor.

MPPT: Maximum Power Point Tracking.

MSSP: Master Synchronous Serial Port.

PCB: Printed Circuit Board.

PIC: Programmable Interface Controllers.

PLC: Programmable Logic Controller.

PPS: Peripheral Pin Select.

PVC: Clorur de Polivinil.

PWM: Pulse Width Modulation.

RAM: Random Access Memory.

REBT: Reglament Electrotècnic de Baixa Tensió.

RTU: Remote Terminal Unit.

SCL: Serial Clock.

SDA: Serial Data.

TMR0: Timer 0.

TMR5: Timer 5.

UART: Universal Asynchronous Receiver-Transmitter.

ULP: Ultra Low Power.

UNE: Una Norma Espanyola.

USB: Universal Serial Bus.

WWDT: Windowed Watchdog Timer.

XLPE: Polietilè Reticulat.

## A. CODI

En aquest annex es mostren els codis que s'han desenvolupat al llarg del projecte: el codi principal del PIC, el codi de la llibreria de Modbus i el codi de l'ESP32 que adquireix la potència mesurada.

### A.1. Codi principal del PIC

El següent codi realitza la funció de control principal del PIC.

```
//ADRIÀ QUINTANA TARRADAS
//PROJECTE LLAVOR
//UNIVERSITAT DE GIRONA
//
//CONTROL DEL CONSUM DE POTÈNCIA
//
//10/05/2023

#include <math.h> //Per calcular el dutty del PWM serà important.
#include "mcc_generated_files/mcc.h"
//#include "mcc_generated_files/i2c1_master.h"
#include "mcc_generated_files/pin_manager.h"

// PIC16F877A Configuration Bit Settings
// 'C' source line config statements
#include <xc.h>
#include <pic18f47q10.h>
#include "modbus.h"
#include <stdio.h>

#define _XTAL_FREQ 8000000

/*-----I2C MESTRE-----
*
* FUNCIONS DE L'I2C I VARIABLES DECLARADES: */

//INICIALITZACIÓ.
/*-----*/
void I2C_Master_Init(const unsigned long c)
{
    //Configuració del MSSP1 actuant com a mestre de I2C
    SSP1CON1 = 0b00101000;
    SSP1CON2 = 0;
    //Definició de la freqüència del clock
    SSP1ADD = (_XTAL_FREQ/(4*c))-1;
    //Es posa es mode alta freqüència 400 kHz
    SSP1STAT = 0;

    //Definició dels pins del canals de dades i de clock.
    SSP1CLKPPS = 0x13; //RC3->MSSP1:SCL1;
    RC3PPS = 0x0F; //RC3->MSSP1:SCL1;
    RC4PPS = 0x10; //RC4->MSSP1:SDA1;
```

```

    SSP1DATPPS = 0x14; //RC4->MSSP1:SDA1;
//Es deshabilita el pot sèrie per transmetre dades.
    SSP1CON1bits.SSPEN = 0;
//Es prefixa el temps que determina si el bit està a 1 o a 0.
    SSP1CON3bits.SDAHT = 1;
}
/*-----*/

//ESPERA
/*-----*/
void I2C_Master_Wait()
{
    while ((SSP1STAT & 0x04) || (SSP1CON2 & 0x1F));
}
/*-----*/

//INICI DE LA COMUNICACIÓ
/*-----*/
void I2C_Master_Start()
{
    /*Rutina que analitza si en la comunicació es transmeten dades i
    * l'Acknowledge*/
    I2C_Master_Wait();
//S'habiliten els pins.
    SSP1CON1bits.SSPEN = 1;
//S'incina la seqüència del bit start.
    SSP1CON2bits.SEN = 1;
//Quan acaba la seqüència aquest bit es posa a zero.
    while(SSP1CON2bits.SEN);
//Fins que no es posi a zero la funció no acaba.
}
/*-----*/

//BIT PER OBTENIR MÉS LETURES
/*-----*/
//En cas de fer una petició de lectura i voler rebre diferents datagrames.
void I2C_Master_RepeatedStart()
{
    I2C_Master_Wait();
    SSP1CON2bits.RSEN = 1;
}
/*-----*/

//PARAR LA COMUNIACIÓ
/*-----*/
//En cas de voler parar la comunicació
void I2C_Master_Stop()
{
    /*Rutina que analitza si en la comunicació es transmeten dades i
    * l'Acknowledge*/
    I2C_Master_Wait();
//S'incina la seqüència del bit stop.
    SSP1CON2bits.PEN = 1;
//Quan acaba la seqüència aquest bit es posa a zero.
    while(SSP1CON2bits.PEN);
//Fins que no es posi a zero la funció no acaba.
}

//ESCRITURA DEL MESTRE A L'ESCLAU
/*-----*/
static inline void I2C_Master_Write(unsigned d)
{
    //Acció d'espera

```

```

I2C_Master_Wait();
//Escriure en el buffer.
SSP1BUF = d;
//Esperar que es buidi el buffer i de rebre l'Acknowledge.
//while(SSP1STATbits.BF);
while(SSP1STATbits.R_nW);
}
/*-----*/

//LECTURA DEL MESTRE A L'ESCLAU
/*-----*/
unsigned short I2C_Master_Read(unsigned short a)
{
    unsigned short temp; //Variable de rtorn

    I2C_Master_Wait(); //Espera del bus
    SSP1CON2bits.RCEN = 1; //habilita el bit de recepció
    I2C_Master_Wait(); //Espera del bus que en aquest cas està rebent dades.
    temp = SSP1BUF; //El valor del registre SSP1BUF queda volcat a la variable.
    I2C_Master_Wait(); //Espera del bus
    SSP1CON2bits.ACKDT = (a)?0:1; /*Depenent del valor de la variable a, inicia
                                   la seqüència ACKDT que es per adquirir
                                   * diferents datagrames de l'esclau.*/
    SSP1CON2bits.ACKEN = 1; //Habilita l'Acknowledge.
    return temp; //Retorna la variable.
}
/*-----*/

//NO ACKNOWLEDGE
/*-----*/
static inline bool I2C1_MasterIsNack(void)
{
    return SSP1CON2bits.ACKSTAT;
}
/*-----*/

/*-----I2C ESCLAU-----*/
/*-----*/

//Variables de tractament:

bool I2C_ESCLAU_MISSATGE=false; //Hi ha hagut una petició.
uint8_t lecturaI2C=0; //Lectura de les dades rebudes.
uint8_t escripturaI2C=0; //Escriptura cap al mestre.

bool Emergencia=false; //Variable booleana que posa el sistema en Emergència.
bool Error=false; //Variable booleana que posa el sistema en Error.

bool Rebut_Angle_Maxim=false;

//Hi ha generació captada per l'Esclau 1:
bool GENERACIO_E1=false;

//Hi ha generació captada per l'Esclau 2:
bool GENERACIO_E2=false;

//Funció de lectura.

/*Aquesta funció es dona quan el byte que rep el buffer coincideix amb l'adreça
del dispositiu funcionant com a esclau. En aquest cas, es fa una interrupció,
aquesta ja s'ha posat en el handler, perquè es produeixi. */

void I2C_Slave_Read(void){

    /*Flag d'interrupció SSP2IF: Aquest flag es posa a nivell alt cada vegada

```

que el buffer rep un byte, a partir que el primer byte casa amb l'adreça. És a dir, si es reben bytes que van destinats a un altre esclau, el dispositiu no fa res, però a la que es rep un byte que casa amb l'adreça, aquest flag es posa a nivell alt i fa la interrupció. I a partir d'aquest moment, per a cada byte rebut, es fa la interrupció fins que no es detecta la condició STOP.

\*

Per a cada byte rebut, fins que el master no envia ell el seu Acknowledge i el bit STOP. L'Esclau envia l'Acknowledge per l'SDA, posant-lo a nivell baix, i també posa el rellotge a nivell baix per l'SCL. I un cop ha llegit les dades del buffer, o ha escrit el buffer, s'habilita el rellotge, es permet la transmissió de dades per l'SDA i es borra el flag d'interrupció.

\*/

```

if(SSP2IF==1){

    //De la manera com marca el protocol l'esclau posa a zero el clock.
    SSP2CON1bits.CKP = 0;

    //El motiu és per garantir el temps de configuració de les dades.

if ((SSP2CON1bits.SSPOV) || (SSP2CON1bits.WCOL))
{
    /*Aquesta situació pot ser deguda: o perquè l'Esclau rep un byte del
    * mestre mentre el buffer de l'esclau encara manté el byte anterior,
    * seria el cas d'OVERFLOW, o que el buffer s'està escrivint mentre
    * s'està transmetent el valor previ, aquest segon cas és
    * una col·lisió.*/

    /*En aquest cas, es llegeix el valor anterior guardat,
    per borrar el buffer.*/
    lecturaI2C = SSP2BUF;

    SSP2CON1bits.SSPOV = 0; // Es borra el bit d'OVERFLOW
    SSP2CON1bits.WCOL = 0; // Es borra el bit de col·lisió.
    SSP2CON1bits.CKP = 1; //Es reinicialitza el rellotge.
}

if(!SSP2STATbits.D_nA && !SSP2STATbits.R_nW)
{
    //El primer byte rebut coincideix amb l'adreça i està en mode escriptura.
    /*Tan bon punt es rep l'adreça, l'esclau genera l'Acknowledge,
    * al mateix temps que la interrupció. Això s'ha configurat en el registre
    SSP2CON2 en el bit 4, ACKEN. Quan es fa l'Acknowledge, tot és per
    * hardware, tant l'acció de posar a zero el canal SDA com la de borrar
    * aquest bit. */
    SSP2CON2bits.ACKEN = 1;

    //Tan bon punt fa l'acknowledge es llegeix el buffer.
    //Cada vegada que es llegeix el buffer, es borra.
    lecturaI2C = SSP2BUF;

    //En la primera dada rebuda, es llegeix l'adreça.
    //En les següents, es llegeixen els bytes posteriors.

    SSP2CON1bits.CKP = 1; //Es reinicialitza el rellotge.

    switch(lecturaI2C){

        case 0xEE: //COMANDA D'EMERGÈNCIA

            Emergencia=true;

            break;
            case 0xBB://COMANDA DE TEST

                escripturaI2C=0xE3;
    }
}

```

```

        break;
    case 0x1F:

        Rebut_Angle_Maxim=true;

        break;
    case 0x1E://Error detectat per l'Esclau 1.

        Error=true;

        break;
    case 0x2E://Error detectat per l'Esclau 1.
        Error=true;
        break;

    case 0xC0:

        GENERACIO_E1=false;

        break;
    case 0xC1:

        GENERACIO_E1=true;

        break;
    case 0xD0:

        GENERACIO_E2=false;

        break;
    case 0xD1:

        GENERACIO_E2=true;

        break;

    }

}

else if(!SSP2STATbits.D_nA && SSP2STATbits.R_nW)
{//El primer byte rebut coincideix amb l'adreça i està en mode lectura.
    SSP2CON2bits.ACKEN = 1;
    //Fa la seqüència de l'acknowledge, i llegeix el buffer.
    //En aquest cas es llegeix l'adreça més el LSB en mode escriptura.
    //Es borra el buffer per ser escrit.
    lecturaI2C = SSP2BUF;

    /*Per evitar que es doni l'overflow, es posa a zero LSB del registre
    SSP2STAT. Aquest bit indica que el buffer està preparat per rebre o
    enviar una altra dada.*/
    SSP2STATbits.BF = 0;

    //S'escriu el byte que es vulgui enviar cap al mestre en el buffer.
    SSP2BUF = escripturaI2C;

    //Es reinicialitza el rellotge.
    SSP2CON1bits.CKP = 1;
    //Fins que no s'hagi enviat el byte cap al mestre queda encallat.
    while(SSP2STATbits.BF);

}

SSP2IF = 0; //Es borra el flag d'interruptió.
I2C_ESCLAU_MISSATGE=true;

}
}

```

```

void I2C_Slave_Init(short address)
{
    SSP2STAT = 0x80;
    SSP2ADD = address; //Setting address
    SSP2CON1 = 0x36;    //As a slave device
    SSP2CON2 = 0x01;
    SSP2DATPPS = 0x0A; //RB2->MSSP2:SDA2;
    RB1PPS = 0x11;    //RB1->MSSP2:SCL2;
    RB2PPS = 0x12;    //RB2->MSSP2:SDA2;
    SSP2CLKPPS = 0x09; //RB1->MSSP2:SCL2;

    //TRISC3 = 1;      //Setting as input as given in datasheet
    //TRISC4 = 1;      //Setting as input as given in datasheet
    INTCONbits.GIE = 1; //Global interrupt enable
    INTCONbits.PEIE = 1; //Peripheral interrupt enable
    SSP2IF = 0;        //Clear interrupt flag
    SSP2IE = 1;        //Synchronous serial port interrupt enable
}

/* _____ */
/*-----LIMITS DE COMUTACIO RELES I FRENAT-----*/

//Amb el Modbus es poden canviar;

int Plimit_bateria=960;//Es preinicialitza el valor a 960 W que són 40 A
unsigned int Vbateria_carregada=27;
int Ibat_inferior=-6;
int Ibat_superior=6;
unsigned int rendiment=9; //Rendiment del sistema.

/* _____ */
/* _____ */
/* _____MODBUS_____ */

//ARRAY per convertir bool a uint16_t i viceversa
bool array_ESTAT_SISTEMA_MODBUS[11];
bool array_ERROR_SISTEMA_MODBUS[7];
bool array_TEST_SISTEMA_MODBUS[9];
uint16_t ESTAT_SISTEMA_MODBUS=0x0000;
uint16_t ERROR_SISTEMA_MODBUS=0x0000;;
uint16_t VARIABLES_REARMAMENT;
uint16_t TEST_SISTEMA_MODBUS;

uint8_t i = 0;
uint8_t c_control=0;
uint8_t RxBuffer[20];
uint8_t rcv;

/*-----EMERGENCIA-----*/

bool AvisEmergenciaPLC=false;
unsigned int Espera_emergencia=0;
unsigned int Potencia_limit_reles=600;
unsigned int Potencia_limit=2200;
bool REVISIO_SISTEMA=false;
bool POSSIBLES_DANYS_RELES_GENERADOR=false;
bool MANIOBRA_FETA=false;
bool REARMAMENT=false;

/* _____ */
/*-----ERROR-----*/
volatile unsigned int Pinversor_teorica=0;

```



```

volatile unsigned int Pbateria_teorica=0;
volatile bool Test_sensors=false;
bool AvisErrorPLC=false;
int ERROR_Pinversor=0;
int ERROR_Pfrenat=0;
int ERROR_Pbateria=0;
bool ERROR_Inversor=false;
bool ERROR_Bateria=false;
bool ERROR_Frenat=false;
bool ERROR_coherencia=false;

//Els valors de Test de l'I2C s'incialitzen com a correctes
uint8_t Test_ESCLAU1=0xE1;
uint8_t Test_ESCLAU2=0xE2;

uint8_t ComptadorErrorTest;
volatile bool HiHaTestComunicacioI2C=false;
bool Error_TEST_I2C=false;
bool Error_WATCHDOG=false;
volatile uint16_t Comptador_watchdog=0;
int MARGE_ERROR=12; //Incincialment el marge és de 12 W, ja que equival a I=0,5 A
uint16_t Intents_Comunicacio=0;

/*_____*/
/*-----ADQUISICIÓ DE DADES-----*/

//Sensors de mesura de magnituds elèctriques.

unsigned int PotenciaGenerada; //Potència obtinguda via I2C a la ESP32.
volatile unsigned int VBateria1;//Valor de tensió de la bateria.
volatile unsigned int VBateria2;//Valor de tensió de la bateria.
volatile int IBateria;//Valor de corrent de la bateria.
volatile unsigned int IFrenat;//Valor de corrent del regulador a l'inversor.
volatile unsigned int Iinversor;//Valor de corrent de l'inversor.

//CONTROL DEL ADC

uint16_t MitjanaMostres;//Valor de corrent de l'inversor.
uint8_t CanalLectura=0x00;

/*-----CÀLCUL DE POTÈNCIES-----*/
//INVERSOR
unsigned int Pinversor_W;
//BATERIA
volatile unsigned int VBateria;
int Intensitat_bateria;
int Pbateria_W;
unsigned int PFRENAT_MESURADA;

/*-----FUNCIONS DEL FRENAT-----*/
int Pfrenat_W=0;//Potència del frenat, inicialment val 0, permet connectar-lo.

//EXCEDENTS:

/*Els excedents afageixen o treuen potència del frenat, poden ser positius o
negatius.*/

//Avalua si la potència de càrrega passa del límit de protecció.
int Pexcedent_bat=0;
//Avalua si la potència de la generació passa del límit de protecció.
int Pexcedent_generacio=0;

//PWM:
uint16_t dutyPWM;

float R_frenat=1.92; //Resistència en ohms, valor testejat amb experiments.

```

```

//Variable que el frenat està al màxim:
bool FRENAT_MAXIM=false;

//TEST DES DEL PLC
bool Test_frenat_PLC=false;
unsigned int FRENAT_PLC=0;

/*-----CONTROL DEL CONSUM DE CÀRREGA-----*/

//Estats del sistema de control del consum:
enum ESTATS{INICI,ESTAT1,ESTAT2,ESTAT3,ESTAT4,ESTAT5,ESTAT6,ESTAT7,
            ESTAT8,ESTAT9,ESTAT10,ESTAT11,ESTAT12,ESTAT13,ESTAT14,ESTAT15,EMERGENCIA,
            ERROR};
enum ESTATS Estat=INICI;

//Variable booleana que indica si a l'estat del generador hi ha generació.
bool ESTAT_AMB_GENERACIO=false;
bool Despertat=false;

//Llindars de commutació dels estats dels relés:

unsigned int Smin=240; //Potència mínima de commutació
unsigned int SB_Sllums=1200;//Suma de les potències dels llums i la bomba
unsigned int Sllums=400; //Potències llums
unsigned int SB=800; //Potència bomba
unsigned int Vllums=25; //Tensió mínima de la bateria per connectar els llums
unsigned int Vbaix=23; //Tensió mínima de càrrega de la bateria
unsigned int Sllindar=200;//Potència de generació llindar pels focus.
bool Nivell_diposits=false;
/*-----CONTROL DEL CONSUM DE CÀRREGA-----*/

//-----DECLARACIÓ DE FUNCIONS-----

//Funció d'emergència:

void MANIOBRA_EMERGENCIA(void);
void Comunicar_Esclaul(void);
void Rebre_Angle_Maxim(void);
void CurtcircuitarGenerador(void);

//Funció d'error
void MANIOBRA_ERROR(void);
void ANALISI_ERRORS_SENSORS(void);
void ANALISI_ERROR_WATCHDOG(void);
void calcul_Potencia_teorica_bateria(void);
bool comprovar_error_sensors(int error, int marge);
bool comprovar_coherencia(unsigned int PGEN, unsigned int VBAT,
                          unsigned int PFre, unsigned int Pinv,bool hi_ha_gen);
void AVIS_ERROR(void);

//Funció de lectura de l'ADC:
void Lectura_Analogica(void);

//Càlcul de la potència de l'inversor i la bateria
void Calcul_de_Potencies(void);
void Comunicar_GENERACIO(uint8_t gen);

//Funcions del frenat:

//Frenat elèctric
void Frenat_Electric(void);
//Mode del frenat:
void seleccio_cas_frenat(void);
//Càlcul del duty cycle
uint16_t Calcul_duty(int potencia_banc);

```

```

//Comutació del CLC amb el duty del PWM
void maniobraFrenat(void);

//Funció comunicació dispositiu càlcul de potència generada:

void dormirESP32(void);
void despertarESP32(void);

//Control del consum de càrrega:
void Transferencia_Consum(void);
void Control_Reles(void);
void Control_bateria_descarregada(void);
void Control_bateria_carregada(void);
void Control_sense_Produccio(void);
void Accionament_Carregues(void);

//FUNCIONS DEL MODBUS

uint16_t ARRAY_uint16_t (bool *cadena, uint16_t longitud){
    uint16_t ENVIAR=0x0000;
    for(int i=0;i<longitud;i++){
        ENVIAR=(ENVIAR<<1);
        ENVIAR=ENVIAR+(uint16_t)*cadena;
        cadena++;
    }
    return ENVIAR;
}

void DADES_REBUDES_EUSART(void){
    R_T_RS485_PORT = 0; //Deshabilita la direcció de transmissió
    rcv = EUSART1_Read();//Escriu la variable de receive.
    //Emmagatzema les dades en el buffer.

    switch (c_control){
        case 0:
            if(rcv==0x03) {
                //ADREÇA
                RxBuffer[c_control] = rcv;
                c_control++;
            }
            break;
        case 1:
            if(rcv==0x04||rcv==0x10){
                //FUNCIO
                RxBuffer[c_control] = rcv;
                c_control++;
            }else c_control=0;
            break;
        case 2:
            RxBuffer[c_control] = rcv;
            c_control++;
            break;
        case 3:
            RxBuffer[c_control] = rcv;
            c_control++;
            break;
        case 4:
            RxBuffer[c_control] = rcv;
            c_control++;
            break;
        case 5:
            RxBuffer[c_control] = rcv;
            c_control++;
            break;
        case 6:
            RxBuffer[c_control] = rcv;

```

```

        c_control++;
        break;
    case 7:
        RxBuffer[c_control] = rcv;
        c_control++;
        if(RxBuffer[1]==0x04){
            modbus_function();
        }
        break;
    case 8:
        RxBuffer[c_control] = rcv;
        c_control++;
        break;
    case 9:
        RxBuffer[c_control] = rcv;
        c_control++;
        break;
    case 10:
        RxBuffer[c_control] = rcv;
        c_control++;
        break;
    case 11:
        RxBuffer[c_control] = rcv;
        c_control++;
        break;
    case 12:
        RxBuffer[c_control] = rcv;
        c_control++;
        break;
    case 13:
        RxBuffer[c_control] = rcv;
        c_control++;
        break;
    case 14:
        RxBuffer[c_control] = rcv;
        c_control++;
        break;
    case 15:
        RxBuffer[c_control] = rcv;
        modbus_function();
        break;
    }
}

//TEST DE COMUNICACIÓ DE L'I2C
bool errorTESTI2C(uint8_t byteTest1,uint8_t byteTest2);

//Rutines d'interrupció per adquirir dades

/*1) Interrupció per temps de cada 200 ms que obté la potència mesurada a
 * la sortida del generador*/

void TMR5_AdquisicioPotencia(void){

    I2C_Master_Start();           //Start condition
    I2C_Master_Write(0x31);       //7 bit address + Read
    //7 bit address + Write
    //I2C_Master_Write(0x8A);
    //Parar la producció
    //I2C_Master_Stop(); //Bit de parada
    uint8_t Lectura2 = (uint8_t)I2C_Master_Read(1); //Read + Acknowledge
    uint8_t Lectura1 = (uint8_t)I2C_Master_Read(0); //Read + Acknowledge
    I2C_Master_Stop();           //Stop condition
    PotenciaGenerada=(unsigned int)((Lectura2<<8)|Lectura1);
}

```

```

}
void Lectura_de_valorMitja_AD(void){

MitjanaMostres=((uint16_t)((ADFLTRH << 8) + ADFLTRL));
Lectura_Analogica();

}

//TEST DE L'I2C

void TMR3_TEST_I2C(void){
    if(!(Emergència||Error)){
        //ESCRIPTURA DEL CODI DE TEST AL DISPOSITIU DE CONTROL DEL PITCH
        I2C_Master_Start();           //Start condition
        I2C_Master_Write(0x10);       //7 bit address + Read
        I2C_Master_Write(0xBB);      //Read + Acknowledge
        I2C_Master_Stop();

        //ESCRIPTURA DEL CODI DE TEST AL DISPOSITIU DE MONITORITZACIÓ DE LA TURBINA.
        I2C_Master_Start();           //Start condition
        I2C_Master_Write(0x20);       //7 bit address + Read
        I2C_Master_Write(0xBB);      //Read + Acknowledge
        I2C_Master_Stop();

        //LECTURA DEL DISPOSITIU DE CONTROL DEL PITCH.
        I2C_Master_Start();           //Start condition
        I2C_Master_Write(0x11);       //7 bit address + Read
        Test_ESCLAU1 = I2C_Master_Read(0); //Read + Acknowledge
        I2C_Master_Stop();

        //LECTURA DEL DISPOSITIU DE MONITORITZACIÓ DE LA TURBINA.
        I2C_Master_Start();           //Start condition
        I2C_Master_Write(0x21);       //7 bit address + Read
        Test_ESCLAU2 = I2C_Master_Read(0); //Read + Acknowledge
        I2C_Master_Stop();

        HiHaTestComunicacioI2C=true;
    }else{
        //Si hi ha emergència no es fa el test de comprovació
    }
}

void TMR4_ACTUALITZACIO(void){
    //AQUÍ VA EL MODBUS

    if(!Emergència&&Error){
        //Aquest timer és de 400 ms
        //S'aprofita per fer una espera en cas d'error de watchdog
        Comptador_watchdog++;
    }else{
        //Quan hi ha emergència no fa res
    }
}

void TMR0_CALCUL_POTENCIES(void){

    Calcul_de_Potencies();
    if(!(Emergència||Error)){

        Test_sensors=true;

    }

}

```

```

//NIVELL DELS DIPÒSITS

void Nivell_baix(void){
    Nivell_diposits=true;
}
void Nivell_alt(void){
    Nivell_diposits=false;
}

void main()
{
    SYSTEM_Initialize();

    INTERRUPT_GlobalInterruptEnable();
    //
    //    // Disable the Global Interrupts
    //    //INTERRUPT_GlobalInterruptDisable();
    //
    //    // Enable the Peripheral Interrupts
    INTERRUPT_PeripheralInterruptEnable();

    //DECLARACIÓ DEL VECTOR D'INTERRUPCIIONS

    //TMR0 cada 200 ms.
    TMR0_SetInterruptHandler(TMR0_CALCUL_POTENCIES);
    TMR3_SetInterruptHandler(TMR3_TEST_I2C);
    TMR4_SetInterruptHandler(TMR4_ACTUALITZACIO);
    TMR5_SetInterruptHandler(TMR5_AdquisicioPotencia);
    ADCC_SetADTIIInterruptHandler(Lectura_de_valorMitja_AD);
    I2C2_SlaveSetIsrHandler(I2C_Slave_Read);//Interrupció I2C Esclau.
    EUSART1_SetRxInterruptHandler(DADES_REBUDES_EUSART);//Interrupció Modbus
    INT0_SetInterruptHandler(Nivell_baix);
    INT1_SetInterruptHandler(Nivell_alt);

    I2C_Master_Init(400000);        //Initialize I2C Master with 400KHz clock
    I2C_Slave_Init(0x50); //Inicicialització de l'Esclau a l'Adreça 0x28;
    SSP2BUF=0;//Es borra tot el que pugui haver al buffer de l'esclau.
    // Enable the Global Interrupts

    //FUNCIO QUE S'HA DETECTAT UN OVERFLOW DEL WDT
    /*Si hi ha desbordament del temporitzador del Watchdog, */
    if(!STATUSbits.TO&&!PCON0bits.RWDT){

        Error_WATCHDOG=true;

    /*En aquest cas, es pot donar un error de WATCHDOG quan hi ha una emergència,
    per tant, totes les variables volàtils es perden els valors emmagatzemats.*/

        //ERROR DE WTAHCDOG I Emergència:

        if(Emergencia){
            Error=false;
            Emergencia=true;

        }else{//TOTS ELS ALTRES CASOS
            Error=true;
        }

    }

    //HABILITA EL WDT
    WDTCON0bits.SEN=1;

```

```

while(1)
{

    //El primer pas del bucle és llegir si hi ha emergència.

    if(Emergencia){
        /*_____EMERGENCIA_ESTAT_1_____*/
        //Funció d'emergencia:
        AvisEmergenciaPLC=true; //Avisa el PLC que està en emergència
        Error=false;
        Frenat_Electric(); //Es posa el frenat al màxim.
        /*_____*/
        MANIOBRA_EMERGENCIA(); //Es fa la maniobra d'emergència
        //QUAN ES DETECTA UNA EMERGENCIA L'ERROR PASSA A SER ZERO.

    }else if(Error){
        /*_____ERROR_ESTAT_1_____*/
        //Funció d'error:
        AvisErrorPLC=true; //Avisa el PLC que està en emergència
        //EN CAS QUE HI HAGI GENERACIÓ
        /*_____*/
        if(ESTAT_AMB_GENERACIO){
            /*_____ERROR_ESTAT_2_____*/
            Frenat_Electric(); //Es posa el frenat al màxim.
            /*_____*/
        }

        //Si no hi ha generació, es procedeix a tractar l'error.

        //EN CAS QUE NO HI HAGI GENERACIÓ
        MANIOBRA_ERROR();

    }else{

        //Es comprova si s'ha fet el test d'error de comunicació d'I2C.
        if(HiHaTestComunicacioI2C){
            /*Si hi ha test, es mira si hi ha hagut un error de recepció de dades,
            és a dir, si s'han rebut les dades malament durant tres vegades
            * implicant que el soroll l'afecta molt, i s'han de prendre mesures.*/

            //En cas d'error es posa en true, l'error i el bit d'error del test.
            Error_TEST_I2C=errorTESTI2C(Test_ESCLAU1,Test_ESCLAU2);
            Error=Error_TEST_I2C;

            HiHaTestComunicacioI2C=false;
        }

        //Error dels sensors:

        //Es mira si hi ha hagut l'interrupció per calcular els valors tèorics.
        //S'avalua si hi ha algun error dels sensors.

        if(Test_sensors){//S'ha donat el flag posat en la interrupció.

            calcul_Potencia_teorica_bateria();

            //Es calcula l'error de lectura.
            ERROR_Pinversor=Pinversor_teorica-Pinversor_W;
            ERROR_Pfrenat=PFRENAT_MESURADA-Pfrenat_W;
        }
    }
}

```

```

ERROR_Pbateria=Pbateria_W-Pbateria_teorica;

//ES COMPROVA SI HI HA ALGUN ERROR EN ELS SENSORS:
ERROR_Inversor=comprovar_error_sensors(ERROR_Pinversor,MARGE_ERROR);
ERROR_Frenat=comprovar_error_sensors(ERROR_Pfrenat,MARGE_ERROR);
ERROR_Bateria=comprovar_error_sensors(ERROR_Pbateria,MARGE_ERROR);
ERROR_coherencia=comprovar_coherencia(Pinversor_W, VBateria,
PFRENAT_MESURADA, Pinversor_W,ESTAT_AMB_GENERACIO);

//En cas d'error, el sistema es posa en estat d'error.
if(ERROR_Inversor||ERROR_Frenat||ERROR_Bateria||ERROR_coherencia){
    Error=true;
    AVIS_ERROR();
}

Test_sensors=false;
//Es borra el flag que hi ha hagut la interrupció.

}

//RELÉS
Transferencia_Consum();

//Finalment s'avalua el frenat elèctric.

Frenat_Electric();

//CADA VEGADA QUE S'ARRIBA A AQUEST PUNT ES RESETEJA EL WATCHDOG
CLRWDI();

}

//No hi ha generació hi s'ha de posar en baix consum:
if(!GENERACIO_E1&&!GENERACIO_E2&&!ESTAT_AMB_GENERACIO) {
    //DORMIM LA ESP32
    dormirESP32();
    //DESHABILITEM EL WATCHDOG PER EVITAR QUE DESPERTI EL SISTEMA
    WDTCON0bits.SEN=0;
    //DORMIM LA ESP32
    SLEEP();
    //NOMÉS SERÀ DESPERTAT PER LES INTERRUPCIONS DE PERIFÈRICS
    //PARA TOTS ELS TMR MENYS EL 0 I 1.

    //S'HA DESPERTAT AQUÍ
    WDTCON0bits.SEN=1;//REAHABILITA EL WATCHDOG.

    //DESPERTA ESP32
    despertarESP32();
    //ES DESPERTA L'ESP32 I ES DONA MARGE A CALCULAR POTÈNCIES
    _delay_ms(100);
    /*Es comunica amb el càlcul de potència generada, per saber
    * l'estat de generació.*/
    I2C_Master_Start(); //Start condition
    I2C_Master_Write(0x31); //7 bit address + Read
    uint8_t lectura_2 = (uint8_t)I2C_Master_Read(1); //Read + Acknowledge
    uint8_t lectura_1 = (uint8_t)I2C_Master_Read(0); //Read + Acknowledge
    I2C_Master_Stop(); //Stop condition
    PotenciaGenerada=(unsigned int)((lectura_2<<8)|lectura_1);
    Despertat=true;
}
//FA UNA VOLTA AL LOOP(), I TOTES LES INTERRUPCIONS FUNCIONEN
}

```



```

//ENVIA QUE HI HA GENERACIÓ.
if(Despertat&&PotenciaGenerada>0){
    Comunicar_GENERACIO(0xF1);//hi ha generació
    Despertat=true;
}
}

}

void MANIOBRA_EMERGENCIA(void){

    //Es revisa que durant l'emergència no hi hagi hagut un error de Watchdog.
    if(!Error_WATCHDOG){

//L'emergència es tracta en dos casos:
/*1)Via I2C sigui per velocitats excessives o vibracions excessives per pala
* trencada, l'Emergencia s'ativa.
* 2) La potència generada és excessiva i l'Esclau 3, es posa en emergència.

* Sempre que s'arribi a l'Estat d'emergència, el frenat es posa al màxim, i
* ho comunica a l'Escalu 1, per indicar emergència. I mogui l'angle d'atac de
* les pales. Si l'emegència li prové de l'Esclau 2, que es qui detecta les
* vibracions, ja ho hurà fet. Però en seguretat val més ser redundat.
*/
//Amb la maniobra d'emergència, el frenat s'envia en el seu valor màxim.

    /*FER UNA ESPERA PERQUÈ ES CONNECTI EL FRENAT.*/

    if(!MANIOBRA_FETA){
    NOCOMUNICACIÓ:
    /*_____EMERGENCIA_ESTAT_2_____*/
    Comunicar_Esclau1(); //se li comunica a l'Esclau 1, que està al màxim.
    CLRWDT();//Es reseteja el WATCHDOG per donar temps a rebre l'angle.
    /*Temps de l'angle són dos segons i el watchdog 4. En aquest punt no hi
    * hauria d'haver problema de Watchdog. Les tres vegades que pot passar
    el programa per aquí es reseteja.*/
    Rebre_Angle_Maxim(); //Espera a que l'Esclau 1 posi l'angle al màxim.
    /*_____*/
    if(Rebut_Angle_Maxim){
    /*_____EMERGENCIA_ESTAT_3_____*/
        CurtcircuitarGenerador();
        REVISIO_SISTEMA=false;
        Rebut_Angle_Maxim=false;//Es borra el bit de control de l'angle màxim

        //Sistema parat i amb emergència.
        /*_____*/
    }if(Rebut_Angle_Maxim){
    /*_____EMERGENCIA_ESTAT_3_____*/
        CurtcircuitarGenerador();
        REVISIO_SISTEMA=false;
        Rebut_Angle_Maxim=false;//Es borra el bit de control de l'angle màxim

        //Sistema parat i amb emergència.
        /*_____else{
        //No s'ha rebut el frenat màxim.
        //No s'ha pogut comunicar amb l'Esclau 1.
        /*_____*/
        /*_____EMERGENCIA_ESTAT_4_____*/
        Espera_emergencia++;
        /*_____*/
        if(Espera_emergencia==3){
        /*Passades les tres vegades es comprova que la potència
        generada sigui zero.*/
            if(PotenciaGenerada<Potencia_limit_reles){
            /*_____EMERGENCIA_ESTAT_3_____*/
            CurtcircuitarGenerador();

```

```

        //Els relés i el generador aguanten el curtcircuit.
        REVISIO_SISTEMA=false;
        //Sistema parat i amb emergència.

/* _____ */
    }else{
/* _____ EMERGENCIA_ESTAT_5 _____ */
        CurtcircularGenerador();
        //Els relés i el generador poden haver quedat danyats.
        REVISIO_SISTEMA=true;
        //Sistema parat i amb emergència i pendent de revisió.
        POSSIBLES_DANYS_RELES_GENERADOR=true;
/* _____ */
    }

    }else{
        goto NOCOMUNICACIO; //Es torna a enviar el missatge.
        //la rutina es repeteix fins a tres vegades.
    }

}

MANIOBRA_FETA=true;
//Serveix per indicar que s'ha fet la maniobra, i avisa el PLC.
}else{
//En aquest punt, el frenat està al màxim i s'ha curtcircuitat el generador.
    Emergencia=false;
    Espera_emergencia=0;
    Estat=EMERGENCIA;
    //El rearmament el fa el PLC, de mentre el sistema roman en emergència.
}

}else{
    if(PotenciaGenerada<Potencia_limit_reles){
/* _____ EMERGENCIA_ESTAT_3 _____ */
        CurtcircularGenerador();
        //Els relés i el generador aguanten el curtcircuit.
        REVISIO_SISTEMA=false;
        //Sistema parat i amb emergència.
/* _____ */
    }else{
/* _____ EMERGENCIA_ESTAT_5 _____ */
        CurtcircularGenerador();
        //Els relés i el generador poden haver quedat danyats.
        REVISIO_SISTEMA=true;
        //Sistema parat i amb emergència i pendent de revisió.
        POSSIBLES_DANYS_RELES_GENERADOR=true;
/* _____ */
    }
    //En aquest punt, el frenat està al màxim i s'ha curtcircuitat el generador.
    Emergencia=false;
    Espera_emergencia=0;
    Estat=EMERGENCIA;
    //El rearmament el fa el PLC, de mentre el sistema roman en emergència.

//En aquesta situació es para el Watchdog perquè el sistema roman a l'espera.
    WDTCON0bits.SEN=0;
//Només queda comunicat amb el PLC.

}
}

void Comunicar_Esclau1(void){
    I2C_Master_Start();
    //S'envia l'adreça.
    I2C_Master_Write(0x10);//Adreça 0x10 + LSB=0
    //7 bit address + Write
    //Indica que s'ha fet la maniobra de l'Esclau 3, és a dir, frenat al màxim:
    I2C_Master_Write(0xA3);
}

```

```

    I2C_Master_Stop();
}
void Rebre_Angle_Maxim(void) {
    //Espera de dos segons per rebre la confirmació de l'angle al màxim
    TMR2_StartTimer();//S'inicia el temporitzador d'espera
    while(!Rebut_Angle_Maxim){//Espera a rebre el missatge
        if(TMR2_ReadTimer()==0xFF){//El temporitzador ha arribat al seu màxim.
            TMR2_Stop();//Es para el temporitzador.
            break;//Es surt del bucle d'espera.
        }
    }
    //Ha rebut el missatge de manera correcta.
    if(Rebut_Angle_Maxim){//Rebut angle màxim
        TMR2_Stop();//Es para el temporitzador.
    }

    TMR2_WriteTimer(0x00);//Es tornar a posar a zero el timer

}
void CurtcircuitarGenerador(void) {
    K_GEN_RS_PORT=1;
    K_GEN_ST_PORT=1;
}

void Lectura_Analogica(void) {
    switch(CanalLectura) {
        case 0x00:

            //PIN RA0

            IBateria=(int) (MitjanaMostres-512);
            //Se li resta l'OFFSET de 2,5 Volts.
            CanalLectura++;
            ADPCH = CanalLectura;

            break;
        case 0x01:

            //PIN RA1

            VBateria1=MitjanaMostres;
            CanalLectura++;
            ADPCH = CanalLectura;

            break;
        case 0x02:
            //PIN RA2

            VBateria2=MitjanaMostres;
            CanalLectura++;
            ADPCH = CanalLectura;

            break;
        case 0x03:
            //PIN RA3

            Iinversor=MitjanaMostres-512; //Se li resta l'OFFSET de 2,5 Volts.
            CanalLectura++;
    }
}

```

```

        ADPCH = CanalLectura;

        break;
    case 0x04:
        //PIN RA4

        IFrenat=MitjanaMostres-512; //Se li resta l'OFFSET de 2,5 Volts.
        CanalLectura=0;
        ADPCH = CanalLectura;

        break;
    }
}
}
void Calcul_de_Potencies(void){

    if(PotenciaGenerada>Potencia_limit){
        //Si es supera els límit de generació, s'actua l'ermegència.
        Emergencia=true;
    }else if(Smin>PotenciaGenerada){
        PotenciaGenerada=0;
        Comunicar_GENERACIO(0xF0);//Comunica que no hi ha generació.
    }

    /*La conversió de corrent en integer a Voltatge_Regulador de potència en W:
    P(W)=(I_integer*RESOLUCIÓ_AD)/Sensibilitat
    -> RESOLUCIÓ_AD=5/(2^10-1)=5/1023
    * -> Sensibilitat=20mV/A ->1/Sensibilitat=50A/V
    * -> Voltatge_Regulador=24 V -> Sempre surt a 24 V.
    * P(W)=I_integer*5/1023*50A/V*24=5,865*I_integer
    * Aproximació-> P(W)=5,9*I_integer
    * P(W)=(59*I_integer)/10
    */
    Pinversor_W=(59*Iinversor)/10;
    PFRENAT_MESURADA=(59*IFrenat)/10;//IGUAL QUE L'INVERSOR.

    //Primer cal saber la tensió de la bateria
    /*Com que el voltatge es troba de la manera següent:
    Vbat=(Vbat(integer)*RESOLUCIÓ_AD)/Sensibilitat
    ->Sensibilitat=5V/24V->1/Sensibilitat=24/5
    ->RESOLUCIÓ_AD=5/(2^10-1)=5/1023
    Vbat=Vbat(integer)*5/1023*24/5
    Vbat=Vbat(integer)*0.02346

    * Per compensar errors:
    * Vbat1=(23*Vbat1(integer))/1000
    * Vbat2=(24*Vbat2(integer))/1000
    * Vbat=Vbat1+Vbat2
    */
    VBateria=(23*VBateria1)/1000+(24*VBateria2)/1000;

    //El corrent de la bateria:
    /*Intensitat=(Intensitat(integer)*RESOLUCIÓ_AD)/Sensibilitat
    ->Sensibilitat=50A/V
    ->RESOLUCIÓ_AD=5/1023
    Intensitat=Intensitat(int)*50*5/1023=Intensitat(int)*0.2443
    Intensitat=(Intensitat(int)*24)/100;
    */

    //S'analitza si rep o entrega energia a partir dels corrents mínims.
    /*Per exemple:
    * El corrent mínim de càrrega és de 1,6 A,
    * que en integer és de 6.
    Aquesta situació es dona: quan no hi ha generació, quan hi ha generació i
    * l'energia generada cobreix el consum de l'inversor i res va a la bateria,

```

```

* quan la bateria està carregada i l'energia que es transfereix va tota
* a les càrregues.
*/

if(IBateria>Ibat_inferior&&IBateria<=Ibat_superior){
    IBateria=0;
}

Intensitat_bateria=(IBateria*24)/100;

if(Intensitat_bateria<0){
    Pbateria_W=(unsigned int)((Intensitat_bateria*-1)*VBateria);
}
else{
    Pbateria_W=(unsigned int)((Intensitat_bateria)*VBateria);
}

}

void Frenat_Electric(void){

    seleccio_cas_frenat();
    maniobraFrenat();

}

void seleccio_cas_frenat(void){

    if(Emergencia||Error){

        //Es posa el frenat al màxim;
        Pfrenat_W=1800;

    }
    else if(!(Emergencia||Error)&&Test_frenat_PLC){

        //La potència del frenat depèn del valor escrit pel PLC
        Pfrenat_W=FRENAT_PLC;

    }
    else{

        if(Intensitat_bateria>0){
            //Cas 1: Protegir la bateria, per allarga la vida útil.
            //Bateria carregant-se i potència molt alta, que pugui ser danyada.
            //El frenat s'activa per protegir la abteria

            Pexcedent_bat=Pbateria_W-Plimit_bateria;

            //En cas que la potència excedeixi el límit, la diferència es suma.
            //En cas que la potència baixi del límit, la diferència es resta.
            //La potència del frenat un cop baixa de 0,
            //la variable Pfrenat_W es reinicialitza a 0.
            Pfrenat_W=Pfrenat_W+Pexcedent_bat;
            if(Pfrenat_W<0){
                Pfrenat_W=0;
            }

        }
        else if(Intensitat_bateria==0&&VBateria>=Vbateria_carregada){

            //Cas 2 del frenat: bateria carregada, i l'excedent es perd al frenat.
            //Bateria està carregada i no aporta energia.
            /*En cas que la generació superi la potència de l'inversor,
            * i les pèrdues, el frenat compensa l'excedent i frena el sistema.
            */
            //El frenat s'activa per evitar sobrecàrregues. Frena la generació

```

```

Pexcedent_generacio=PotenciaGenerada-
    Pinversor_W-PFRENAT_MESURADA-
    (PotenciaGenerada-PotenciaGenerada*rendiment/10);

//Si l'excedent és positiu, s'afageix càrrega, i si és negatiu es treu.
Pfrenat_W=Pfrenat_W+Pexcedent_generacio;

if(Pfrenat_W<0){
    Pfrenat_W=0;
}
}
else if(Intensitat_bateria<0){
    /*Aquest estat pot semblar redundat, però es posa per un cas, ja
    que quan la bateria aporta energia, és a dir, el corrent és
    * més petit que zero, el frenat no ha d'actuar, l'energia de
    * la bateria s'ha de destinar a alimentar càrregues.*/
    Pfrenat_W=0;
}
}

}

uint16_t Calcul_duty(int potencia_banc){
    uint16_t Dcycle;
    float duty;
    //Aplica la fórmula de càlcul del duty cycle.
    duty=((float)sqrt((potencia_banc)*R_frenat)/24);

    //Com que el PWM és de 10 bits, es multiplica per 1024.

    //0 -> 0% Duty cycle
    //1023 -> 100% Duty cycle

    duty=duty*1024;

    return Dcycle=(uint16_t)(duty); //Es passa el valor en enter en el programa.
}

void maniobraFrenat(void){
    if(Pfrenat_W>0 & Pfrenat_W<=300){
        CLCI2_PORT=0;
        CLCI1_PORT=0;
        CLCI0_PORT=1;
        dutyPWM=Calcul_duty(Pfrenat_W);
        PWM3_LoadDutyValue(dutyPWM);

    }else if(Pfrenat_W>300 & Pfrenat_W<=600){
        CLCI2_PORT=0;
        CLCI1_PORT=1;
        CLCI0_PORT=0;
        dutyPWM=Calcul_duty(Pfrenat_W-300);
        PWM3_LoadDutyValue(dutyPWM);

    }else if(Pfrenat_W>600 & Pfrenat_W<=900){
        CLCI2_PORT=0;
        CLCI1_PORT=1;
        CLCI0_PORT=1;
        dutyPWM=Calcul_duty(Pfrenat_W-600);
        PWM3_LoadDutyValue(dutyPWM);

    }else if(Pfrenat_W>900 & Pfrenat_W<=1200){
        CLCI2_PORT=1;
        CLCI1_PORT=0;
    }
}

```

```

        CLCIO_PORT=0;
        dutyPWM=Calcul_duty(Pfrenat_W-900);
        PWM3_LoadDutyValue(dutyPWM);

    }else if(Pfrenat_W>1200 & Pfrenat_W<=1500){
        CLCI2_PORT=1;
        CLCI1_PORT=0;
        CLCIO_PORT=1;
        dutyPWM=Calcul_duty(Pfrenat_W-1200);
        PWM3_LoadDutyValue(dutyPWM);

    }else if(Pfrenat_W>1500 & Pfrenat_W<=1800){
        CLCI2_PORT=1;
        CLCI1_PORT=1;
        CLCIO_PORT=0;
        dutyPWM=Calcul_duty(Pfrenat_W-1500);
        PWM3_LoadDutyValue(dutyPWM);

    }else {
        CLCI2_PORT=0;
        CLCI1_PORT=0;
        CLCIO_PORT=0;

    }
}
//Frenat màxim:

/*Com que es pot donar un cert marge d'error a efecte pràctic el duty a 1023
 * o a 1000 és el mateix, per tant, el màxim ja es considera amb les sis bancs
 * connectats i el duty a 1000.*/

}
void Transferencia_Consum(void){
    Control_Reles();
    Accionament_Carregues();
}
void Control_Reles(void){

    //L'emergència preodimnia sobre tots els estats.
    if(Estat==EMERGENCIA&&REARMAMENT&&!REVISIO_SISTEMA
    &&!POSSIBLES_DANYS_RELES_GENERADOR){
        //Des del PLC s'ha d'indicar que el sistema s'ha revisat.
        //El rearmament el fa el PLC.
        REARMAMENT=false;//la varibale de reamemaent es torna a posar a false.
        AvisEmergenciaPLC=false;
        WDTCON0bits.SEN=1;//Es torna a habilitar el WDT.
        Error_WATCHDOG=false;
        MANIOBRA_FETA=false;
        /*Si mentre hi havia una emergència,
        hi ha hagut overflow del WDT*/
        if(VBateria>Vbaix){
            //Si la bateria està carregada s'envia a l'estat 14.
            Estat=ESTAT14;
        }else{
            //Si la bateria està descarregada s'envia a inici.
            Estat=INICI;
        }
    }

}
}else if(Estat==ERROR&&REARMAMENT&&!REVISIO_SISTEMA
    &&!POSSIBLES_DANYS_RELES_GENERADOR){
    //Des del PLC s'ha d'indicar que el sistema s'ha revisat.
    //El rearmament el fa el PLC.
    REARMAMENT=false;//la varibale de reamemaent es torna a posar a false.
    WDTCON0bits.SEN=1;//Es torna a habilitar el WDT.

    //Les variables d'indicació d'error es tornen a posar totes a zero.

```

```

ERROR_Inversor=false;
ERROR_Bateria=false;
ERROR_Frenat=false;
Error_WATCHDOG=false;
Error_TEST_I2C=false;
ERROR_coherencia=false;

//Les variables de càlcul d'error també es torenen a posar a zero.
ERROR_Pinversor=0;
ERROR_Pfrenat=0;
ERROR_Pbateria=0;

if(VBateria>Vbaix){
    //Si la bateria està carregada s'envia a l'estat 14.
    Estat=ESTAT14;
}else{
    //Si la bateria està descarregada s'envia a inici.
    Estat=INICI;
}

}else{
    //Es segueix l'arbre binàri dissenyat.
if(Estat==INICI){
    if(PotenciaGenerada>Smin){
        Estat=ESTAT1;
    }else{
        Estat=INICI;
    }
}
if(Estat==ESTAT1){
    if(VBateria>Vbateria_carregada){
        Estat=ESTAT10;
    }else{
        if(PotenciaGenerada>SB_Sllums){
            if(Nivell_diposits){
                Estat=ESTAT2;
            }else{
                Estat=ESTAT3;
            }
        }else{
            if(PotenciaGenerada>SB){
                if(Nivell_diposits){
                    Estat=ESTAT4;
                }else{
                    Estat=ESTAT5;
                }
            }else{
                Estat=ESTAT5;
            }
        }
    }
}
}
if(Estat==ESTAT2){
    if(Intensitat_bateria<=0){
        Estat=ESTAT7;
    }else{
//Avaluació dels estat amb la bateria descarregada. Sempre és el mateix.
        Control_bateria_descarregada();
    }
}
if(Estat==ESTAT3){
    if(Intensitat_bateria<=0){
        Estat=ESTAT9;
    }else{
        Control_bateria_descarregada();
    }
}

```



```

}
if(Estat==ESTAT4){
    if(Intensitat_bateria<=0){
        Estat=ESTAT8;
    }else{
        Control_bateria_descarregada();
    }
}
if(Estat==ESTAT5){
    if(Intensitat_bateria<=0){
        Estat=ESTAT10;
    }else{
        if(Nivell_diposits&&PotenciaGenerada>SB){
            Estat=ESTAT4;
        }else{
            if(PotenciaGenerada>Sllums&&VBateria>Vllums){
                Estat=ESTAT6;
            }else{
                Estat=ESTAT1;
            }
        }
    }
}
if(Estat==ESTAT6){
    if(Intensitat_bateria<=0){
        Estat=ESTAT9;
    }else{
        Control_bateria_descarregada();
    }
}
if(Estat==ESTAT7){
    if(Intensitat_bateria>0){
        Estat=ESTAT2;
    }else{
        //Avaluació dels estat amb la bateria carregada. Sempre és el mateix.
        Control_bateria_carregada();
    }
}
if(Estat==ESTAT8){
    if(Intensitat_bateria>0){
        Estat=ESTAT4;
    }else{
        Control_bateria_carregada();
    }
}
if(Estat==ESTAT9){
    if(Intensitat_bateria>0){
        Estat=ESTAT6;
    }else{
        Control_bateria_carregada();
    }
}
if(Estat==ESTAT10){
    if(Intensitat_bateria>0){
        Estat=ESTAT5;
    }else{
        Control_bateria_carregada();
    }
}
if(Estat==ESTAT11){
    if(Intensitat_bateria>0){
        Estat=ESTAT2;
    }else{
        Control_bateria_carregada();
    }
}
if(Estat==ESTAT12){
    if(Intensitat_bateria>0){

```

```

        Estat=ESTAT4;
    }else{
        Control_bateria_carregada();
    }
}
if(Estat==ESTAT13){
    if(Intensitat_bateria>0){
        Estat=ESTAT6;
    }else{
        Control_bateria_carregada();
    }
}
if(Estat==ESTAT14){
    if(PotenciaGenerada>Smin){
        if(VBateria>Vbaix){
            Estat=ESTAT10;
        }else{
            Estat=ESTAT1;
        }
    }else{
//Avaluació dels estat amb el molí sense producció. Sempre és el mateix.
        Control_sense_Produccio();
    }
}
if(Estat==ESTAT15){
    if(PotenciaGenerada>Smin){
        if(VBateria>Vbaix){
            Estat=ESTAT10;
        }else{
            Estat=ESTAT1;
        }
    }else{
        Control_sense_Produccio();
    }
}
}
}

void Control_bateria_descarregada(void){

    if(Nivell_diposits){
        if(PotenciaGenerada>SB_Sllums){
            Estat=ESTAT2;
        }else{
            if(PotenciaGenerada>SB){
                Estat=ESTAT4;
            }else{
                if(PotenciaGenerada>Sllums&&VBateria>Vllums){
                    Estat=ESTAT6;
                }else{
                    if(PotenciaGenerada>Smin){
                        Estat=ESTAT1;
                    }else{
                        if(VBateria>Vbaix){
                            Estat=ESTAT14;
                        }else{
                            Estat=INICI;
                        }
                    }
                }
            }
        }
    }else{
        if(PotenciaGenerada>SB_Sllums){
            Estat=ESTAT3;
        }else{

```

```

        if(PotenciaGenerada>SB){
            Estat=ESTAT5;
        }else{
            if(PotenciaGenerada>Sllums&&VBateria>Vllums){
                Estat=ESTAT6;
            }else{
                if(PotenciaGenerada>Smin){
                    Estat=ESTAT1;
                }else{
                    if(VBateria>Vbaix){
                        Estat=ESTAT14;
                    }else{
                        Estat=INICI;
                    }
                }
            }
        }
    }
}

void Control_bateria_carregada(void){

    if(VBateria<=Vbaix){
        Estat=ESTAT1;
    }else{
        if(Nivell_diposits){
            if(PotenciaGenerada>SB_Sllums){
                Estat=ESTAT7;
            }else{
                if(PotenciaGenerada>SB){
                    if(PotenciaGenerada+(unsigned int)Pbateria_W>SB_Sllums){
                        Estat=ESTAT11;
                    }else{
                        Estat=ESTAT8;
                    }
                }else{
                    if(PotenciaGenerada+(unsigned int)Pbateria_W>SB){
                        Estat=ESTAT12;
                    }else{
                        if(PotenciaGenerada>Sllums){
                            Estat=ESTAT9;
                        }else{
                            if(PotenciaGenerada>Smin){
                                Estat=ESTAT10;
                            }else{
                                Estat=ESTAT14;
                            }
                        }
                    }
                }
            }
        }
    }else{
        if(PotenciaGenerada>SB_Sllums){
            Estat=ESTAT9;
        }else{
            if(PotenciaGenerada>Sllindar){
                if(PotenciaGenerada+(unsigned int)Pbateria_W>Sllums){
                    Estat=ESTAT13;
                }else{
                    if(PotenciaGenerada>Smin){
                        Estat=ESTAT10;
                    }else{
                        Estat=ESTAT14;
                    }
                }
            }
        }else{
    }
}

```

```

        if(PotenciaGenerada>Smin){
            Estat=ESTAT10;
        }else{
            Estat=ESTAT14;
        }
    }
}
}

}
}
}

void Control_sense_Produccio(void){
    if(VBateria>Vbaix){
        if(Nivell_diposits){
            Estat=ESTAT15;
        }else{
            Estat=ESTAT14;
        }
    }else{
        Estat=INICI;
    }
}

void Accionament_Carregues(void){

    switch(Estat){
        case INICI:
            K_BOMBA_PORT=0;
            K_FOCUS_PORT=0;
            K_GEN_RS_PORT=0;
            K_GEN_ST_PORT=0;
            ESTAT_AMB_GENERACIO=false;
            Pinversor_teorica=0;
            break;
        case ESTAT1:
            K_BOMBA_PORT=0;
            K_FOCUS_PORT=0;
            K_GEN_RS_PORT=0;
            K_GEN_ST_PORT=0;
            ESTAT_AMB_GENERACIO=true;
            Pinversor_teorica=0;
            break;
        case ESTAT2:
            K_BOMBA_PORT=1;
            K_FOCUS_PORT=1;
            K_GEN_RS_PORT=0;
            K_GEN_ST_PORT=0;
            ESTAT_AMB_GENERACIO=true;
            Pinversor_teorica=SB_Sllums;
            break;
        case ESTAT3:
            K_BOMBA_PORT=0;
            K_FOCUS_PORT=1;
            K_GEN_RS_PORT=0;
            K_GEN_ST_PORT=0;
            ESTAT_AMB_GENERACIO=true;
            Pinversor_teorica=Sllums;
            break;
        case ESTAT4:
            K_BOMBA_PORT=1;
            K_FOCUS_PORT=0;
            K_GEN_RS_PORT=0;
            K_GEN_ST_PORT=0;
            ESTAT_AMB_GENERACIO=true;
            Pinversor_teorica=SB;
            break;
        case ESTAT5:
            K_BOMBA_PORT=0;

```

```
    K_FOCUS_PORT=0;
    K_GEN_RS_PORT=0;
    K_GEN_ST_PORT=0;
    ESTAT_AMB_GENERACIO=true;
    Pinversor_teorica=0;
break;
case ESTAT6:
    K_BOMBA_PORT=0;
    K_FOCUS_PORT=1;
    K_GEN_RS_PORT=0;
    K_GEN_ST_PORT=0;
    ESTAT_AMB_GENERACIO=true;
    Pinversor_teorica=S1lums;
break;
case ESTAT7:
    K_BOMBA_PORT=1;
    K_FOCUS_PORT=1;
    K_GEN_RS_PORT=0;
    K_GEN_ST_PORT=0;
    ESTAT_AMB_GENERACIO=true;
    Pinversor_teorica=SB_S1lums;
break;
case ESTAT8:
    K_BOMBA_PORT=1;
    K_FOCUS_PORT=0;
    K_GEN_RS_PORT=0;
    K_GEN_ST_PORT=0;
    ESTAT_AMB_GENERACIO=true;
    Pinversor_teorica=SB;
break;
case ESTAT9:
    K_BOMBA_PORT=0;
    K_FOCUS_PORT=1;
    K_GEN_RS_PORT=0;
    K_GEN_ST_PORT=0;
    ESTAT_AMB_GENERACIO=true;
    Pinversor_teorica=S1lums;
break;
case ESTAT10:
    K_BOMBA_PORT=0;
    K_FOCUS_PORT=0;
    K_GEN_RS_PORT=0;
    K_GEN_ST_PORT=0;
    ESTAT_AMB_GENERACIO=true;
    Pinversor_teorica=0;
break;
case ESTAT11:
    K_BOMBA_PORT=1;
    K_FOCUS_PORT=1;
    K_GEN_RS_PORT=0;
    K_GEN_ST_PORT=0;
    ESTAT_AMB_GENERACIO=true;
    Pinversor_teorica=SB_S1lums;
break;
case ESTAT12:
    K_BOMBA_PORT=1;
    K_FOCUS_PORT=0;
    K_GEN_RS_PORT=0;
    K_GEN_ST_PORT=0;
    ESTAT_AMB_GENERACIO=true;
    Pinversor_teorica=SB;
break;
case ESTAT13:
    K_BOMBA_PORT=0;
    K_FOCUS_PORT=1;
    K_GEN_RS_PORT=0;
    K_GEN_ST_PORT=0;
    ESTAT_AMB_GENERACIO=true;
```

```

        Pinversor_teorica=S1lums;
break;
case ESTAT14:
    K_BOMBA_PORT=0;
    K_FOCUS_PORT=0;
    K_GEN_RS_PORT=0;
    K_GEN_ST_PORT=0;
    ESTAT_AMB_GENERACIO=false;
    Pinversor_teorica=0;
break;
case ESTAT15:
    K_BOMBA_PORT=1;
    K_FOCUS_PORT=0;
    K_GEN_RS_PORT=0;
    K_GEN_ST_PORT=0;
    ESTAT_AMB_GENERACIO=false;
    Pinversor_teorica=SB;
break;
case EMERGENCIA:
    K_BOMBA_PORT=0;
    K_FOCUS_PORT=0;
    ESTAT_AMB_GENERACIO=false;

break;
case ERROR:
    K_BOMBA_PORT=0;
    K_FOCUS_PORT=0;
    ESTAT_AMB_GENERACIO=false;

break;
}

}

void dormirESP32(void){
    /*En el dispositiu de càlcul de la potència generada,
    * el microprocessador també es pot posar a dormir.
    * Quan en l'aerogenerador no hi ha producció d'energia,
    * i l'estat de la bateria està per sota d'un cert llindar,
    * on es volen minimitzar al màxim possible els elements
    * receptors d'energia amb l'objectiu que el sistema pugui
    * estar en un estat de menor consum possible. El PIC en
    * aquesta casuística, fa una petició d'escriptura al
    * microprocessador ESP32 via I2C, enviant un byte amb valor 0x3C,
    * que el posarà el dispositiu a dormir.
    */
    I2C_Master_Start();
    //S'envia l'adreça.
    I2C_Master_Write(0x30);
    //7 bit address + Write
    I2C_Master_Write(0x3C);
    I2C_Master_Stop();
    DESPERTAR_ESP32_PORT=0;
}

void despertarESP32(void){
    DESPERTAR_ESP32_PORT=1;
}

//Comprovació que les dades rebudes estan bé:
bool errorTESTI2C(uint8_t byteTest1,uint8_t byteTest2){

    if(byteTest1!=0xE1||byteTest1!=0xE2){
        ComptadorErrorTest++;
    }else{
        ComptadorErrorTest=0;
    }
    if(ComptadorErrorTest==3){

```

```

    return true;
}
else{
    return false;
}
}

void calcul_Potencia_teorica_bateria(void) {
    if(PotenciaGenerada>0&&
        (VBateria<=Vbateria_carregada||Intensitat_bateria>0)){
        //La bateria està descarregada i hi ha generació:
        Pbateria_teorica=PotenciaGenerada-Pinversor_teorica-(int)Pfrenat_W
            -(PotenciaGenerada-PotenciaGenerada*rendiment/10);

    }else if(PotenciaGenerada>0&&
        !(VBateria<=Vbateria_carregada||Intensitat_bateria>0)){
        //la bateria carregada. En aquest cas, la bateria pot aportar energia.
        //Per tant, l'energia mai no pot ser inferior a zero.
        Pbateria_teorica=(Pinversor_teorica-(PotenciaGenerada*rendiment/10)>0)?
            (Pinversor_teorica-PotenciaGenerada):0;

        /*En aquesta situació s'avalua quan la bateria complementa la generació.
        * És a dir, si per accionar les càrregues s'empra energia provinent de
        * la bateria. Per tant, aquesta situació
        * només es dona per la bateria carregada i hi ha generació.*/

    }else if(PotenciaGenerada==0&&(VBateria>Vbaix)){

        //Si no hi ha producció, i la bateria permet connectar càrrega.

        Pbateria_teorica=Pinversor_teorica;

    }else{
        //No hi ha producció i la bateria està descarregada.
        Pbateria_teorica=0;
    }
}

bool comprovar_error_sensors(int error, int marge){
    /* Si l'error està per sobre el marge inferior, o per sota
    * que és de -12 W inicialment, o per sobre el superior +12 W*/
    if(error<marge&&error>(marge*-1)) return false;
    else return true;}

bool comprovar_coherencia(unsigned int PGEN, unsigned int VBAT,
    unsigned int PFre, unsigned int Pinv, bool hi_ha_gen){

    /*Si la potència generada és zero i el corrent de fre i inversor
    * és superior a zero. Hi ha un error de coherència, causat per
    * un dels sensors.*/

    if((PGEN==0&&(PFre>0||Pinv>0))&&hi_ha_gen) return true;
    else if(VBAT<13) return true; /*Si la tensió de la bateria és
    * inferior a la meitat, vol dir que
    * un sensor no llegeix de forma
    * correcte una de les bateries.*/

    return false;
}

void MANIOBRA_ERROR(void) {
    if(Error_WATCHDOG) {

        //En aquest cas es a deshabilita el WDT:
        WDTCN0bits.SEN=0;

        /*Es fa una espera de 4 segons.
        * Aprofitant la interrupció del timer 4.

```

```

    * Per assegurar que el frenat esigui al màxim i
    * curtcircuitar el generador.
    */
//Si no hi ha producció, es curtcircuita l'alternador directament.
while(Comptador_watchdog<10&&ESTAT_AMB_GENERACIO);
CurtcircuitarGenerador(); //Es curtcircuita el generador.
REVISIO_SISTEMA=true;//Cal revisar el sistema.
POSSIBLES_DANYS_RELES_GENERADOR=true;
Estat=ERROR;

}else{

Comunicar_Esclau1();
/*DINS DE L'ERROR PER A CADA COP QUE ES FA LA COMUNICACIÓ
 * ES RESETEJA EL WATCHDOG.*/
CLRWDI();

//Es fa una comunicació inicial
while(!Rebut_Angle_Maxim){
//Si s'ha rebut una dada que no és correcta
if(I2C_ESCLAU_MISSATGE&&lecturaI2C!=0x1F){
I2C_ESCLAU_MISSATGE=false;
//Es torna a fer una petició fins a tres intents.
/* Si amb tres intents no es rep bé la dada, es considera una
 * fallada de comunicació i es deixa en bucle infint perquè
 * salti el watchdog.
 */
Comunicar_Esclau1();
/*DINS DE L'ERROR PER A CADA COP QUE ES FA LA COMUNICACIÓ
 * ES RESETEJA EL WATCHDOG.*/
CLRWDI();
Intents_Comunicacio++;
if(Intents_Comunicacio==3){
Intents_Comunicacio=0;
while(1);
}
}
}
/*Si no es rep cap dada, queda en bucle infinit i salta
 * el watchdog, i es tracta l'error com si hi hagués un error de
 * hardware amb una aturada més sobtada.*/

}

CurtcircuitarGenerador();
REVISIO_SISTEMA=true;//Cal revisar el sistema.
Estat=ERROR;
//En aquesta situació es para el Watchdog perquè el sistema roman a l'espera.
WDTCN0bits.SEN=0;
//Només queda comunicat amb el PLC.

}
Intents_Comunicacio=0;
Error=false;
}
void AVIS_ERROR(void){
I2C_Master_Start();
//S'envia l'adreça.
I2C_Master_Write(0x10);//Adreça Esclau 1 + LSB=0
//7 bit address + Write
//Indica que s'ha diagnosticat un error:
I2C_Master_Write(0x3E);//Codi d'error

```



```

    I2C_Master_Stop();
    I2C_Master_Start();
    //S'envia l'adreça.
    I2C_Master_Write(0x20);//Adreça Esclau 2 + LSB=0
    //7 bit address + Write
    //Indica que s'ha diagnosticat un error:
    I2C_Master_Write(0x3E);//Codi d'error
    I2C_Master_Stop();
}
void Comunicar_GENERACIO(uint8_t gen){
    I2C_Master_Start();
    //S'envia l'adreça.
    I2C_Master_Write(0x10);//Adreça Esclau 1 + LSB=0
    //7 bit address + Write
    //Indica que s'ha diagnosticat un error:
    I2C_Master_Write(gen);//Codi d'error
    I2C_Master_Stop();
    I2C_Master_Start();
    //S'envia l'adreça.
    I2C_Master_Write(0x20);//Adreça Esclau 2 + LSB=0
    //7 bit address + Write
    //Indica que s'ha diagnosticat un error:
    I2C_Master_Write(gen);//Codi d'error
    I2C_Master_Stop();
}

```

## A.2. Llibreria del Modbus

En el present apartat hi ha els arxius de la llibreria de Modbus que s'ha programat, tant l'arxiu del codi com la capçalera.

### A.2.1. Capçalera del Modbus

El codi adjuntat consisteix en la capçalera de la llibreria de Modbus.

```

//FUNCIO MODBUS
//ADRIÀ QUINTANA TARRADAS
//PROJECTE LLAVOR
//UNIVERSITAT DE GIRONA
//
//CONTROL DEL CONSUM DE POTÈNCIA
//
//10/05/2023

//#ifndef XC_HEADER_TEMPLATE_H
#define XC_HEADER_TEMPLATE_H
#include <xc.h>
#ifndef XC_HEADER_TEMPLATE_H
#define XC_HEADER_TEMPLATE_H
#include <xc.h>
#ifdef __cplusplus
extern "C" {
#endif
#ifdef __cplusplus
}
#endif
#endif

```

```

static uint16_t MODBUS_CRC16_RX(uint8_t len);
static uint16_t MODBUS_CRC16_TX(uint8_t len);
uint8_t CRC8_Lo (uint16_t crc16);
uint8_t CRC8_Hi(uint16_t crc16);
bool checkCRC(void);
void initial_leds_state(void);
void error_function(void);
void modbus_read_input_registers(uint16_t reg_num);
void modbus_write_multiple_registers(void);
void modbus_function (void);

extern uint16_t ESTAT_SISTEMA_MODBUS;
extern uint16_t ERROR_SISTEMA_MODBUS;
extern uint16_t VARIABLES_REARMAMENT;
extern uint16_t TEST_SISTEMA_MODBUS;
extern uint8_t c_control;

//Llindars de commutació dels estats dels relés:
extern unsigned int Smin; //Potència mínima de commutació
extern unsigned int SB_Sllums; //Suma de les potències dels llums i la bomba
extern unsigned int Sllums; //Potències llums
extern unsigned int SB; //Potència bomba
extern unsigned int Vllums; //Tensió mínima de la bateria per connectar els llums
extern unsigned int Vbaix; //Tensió mínima de càrrega de la bateria
extern unsigned int Sllindar; //Potència de generació llindar pels focus.
extern bool Nivell_diposits;
extern int Plimit_bateria; //Es preinicialitza el valor a 960 W que són 40 A
extern unsigned int Vbateria_carregada;
extern unsigned int Potencia_limit_reles;
extern unsigned int Potencia_limit;
extern int Ibat_inferior;
extern int Ibat_superior;
extern int MARGE_ERROR;

//CONTROL DE CàLCULS DE PROGRAMA

extern unsigned int rendiment; //Rendiment del sistema.

//EMERGÈNCIA
extern bool Emergencia;
extern bool AvisEmergenciaPLC;
extern bool REVISIO_SISTEMA;
extern bool POSSIBLES_DANYS_RELES_GENERADOR;
extern bool MANIOBRA_FETA;
extern bool REARMAMENT;

//ERROR
extern bool Error;
extern bool AvisErrorPLC;
extern bool ERROR_Inversor;
extern bool ERROR_Bateria;
extern bool ERROR_Frenat;
extern bool ERROR_coherencia;
extern bool Error_TEST_I2C;
extern bool Error_WATCHDOG;

//MESURES DELS SENSORS:
extern unsigned int PotenciaGenerada;
extern unsigned int Pinversor_W;
extern volatile unsigned int VBateria;
extern int Intensitat_bateria;
extern int Pbateria_W;
extern unsigned int PFRENAT_MESURADA;

```

```

extern int Pfrenat_W;
extern bool ESTAT_AMB_GENERACIO;

//TEST FRENAT
extern bool Test_frenat_PLC;
extern unsigned int FRENAT_PLC;

//Gobal variables
extern uint8_t i;
extern uint8_t RxBuffer[];

```

## A.2.2. Codi del Modbus

En el codi adjuntat hi ha la llibreria que s'ha programat pel Modbus.

```

//FUNCIO MODBUS
//ADRIÀ QUINTANA TARRADAS
//PROJECTE LLAVOR
//UNIVERSITAT DE GIRONA
//
//CONTROL DEL CONSUM DE POTÈNCIA
//
//10/05/2023

#include "mcc_generated_files/mcc.h"
#include "modbus.h"
#include <stdio.h>
#include <pic18f47q10.h>
#include <xc.h>
#define SlaveAddress 3 //DEFINIR ADREÇA DEL ESCLAU

bool LECTURA_DEL_uint16_auto[16];
bool LECTURA_DEL_uint16_test[16];

//Gobal variables
uint8_t response[20];
uint8_t error;
bool frame;
uint16_t comprovacio=0;
uint16_t ZONA_MORTA_SENSORS_DE_CORRENT=6;
//Funcions Modbus
typedef enum functions{READ_COILS = 0x01, //mode d'accés a bit
READ_INPUT_REGISTERS = 0x04,
WRITE_MULTIPLE_REGISTERS = 0x10,
}function;
//Definim les diferents excepcions que hi ha en el protocol
typedef enum exceptions{ILLEGAL_FUNCTION = 0x01,
ILLEGAL_DATA_ADDRESS = 0x02,
ILLEGAL_DATA_VALUE = 0x03,
}exception;
//LUT (Look Up Table) CRC sol·licitud
static uint16_t MODBUS_CRC16_RX(uint8_t len)
{
static const uint16_t table[256] = {
0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0x1BC1, 0xDA81, 0x1A40,

```

```

0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x55C1, 0x9481, 0x5440,
0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x5FC1, 0x9E81, 0x5E40,
0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x4BC1, 0x8A81, 0x4A40,
0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040 };
uint8_t index = 0;
uint16_t crc = 0xFFFF;
for(uint8_t j = 0; j < len; j++)
{
index = (RxBuffer[j]) ^ crc;
crc >>= 8;
crc ^= table[index];
}
return crc;
}
//LUT (Look Up Table) CRC resposta
static uint16_t MODBUS_CRC16_TX(uint8_t len)
{
static const uint16_t table[256] = {
0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCF41, 0xCE81, 0x0E40,
0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x55C1, 0x9481, 0x5440,
0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x5FC1, 0x9E81, 0x5E40,

```

```

0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040 };
uint8_t index = 0;
uint16_t crc = 0xFFFF;
for(uint8_t j = 0; j < len; j++)
{
index = (response[j]) ^ crc;
crc >>= 8;
crc ^= table[index];
}
return crc;
}
//CRC 8 bits low byte
uint8_t CRC8_Lo (uint16_t crcl6){
uint8_t crc_low = crcl6 | 0;
return crc_low;
}
//CRC 8 bits high byte
uint8_t CRC8_Hi(uint16_t crcl6){
uint8_t crc_high = (crcl6 - CRC8_Lo(crcl6)) >> 8;
return crc_high;
}
//Comprovació que la trama llegida i calculada de les dades que rebem són iguals
bool checkCRC(void){
if(RxBuffer[1] == READ_COILS || RxBuffer[1] == READ_INPUT_REGISTERS){
uint16_t crcl6;
uint8_t crc_Hi, crc_Lo;
crcl6 = MODBUS_CRC16_RX(6);
crc_Hi = CRC8_Hi(crcl6);
crc_Lo = CRC8_Lo(crcl6);
if(RxBuffer[7] == crc_Hi && RxBuffer[6] == crc_Lo){
comprovacio=1;
return true;
}
else{
return false;
}
}
else if(RxBuffer[1] == WRITE_MULTIPLE_REGISTERS){
uint16_t crcl6;
uint8_t crc_Hi, crc_Lo;
crcl6 = MODBUS_CRC16_RX(13);
crc_Hi = CRC8_Hi(crcl6);
crc_Lo = CRC8_Lo(crcl6);
if(RxBuffer[14] == crc_Hi && RxBuffer[13] == crc_Lo){
return true;
}
else{
return false;
}
}
}
void error_function(void){
if(RxBuffer[1] == READ_COILS){
uint16_t crcl6;
crcl6 = MODBUS_CRC16_TX(3);
response[0] = SlaveAddress;
response[1] = 0x80 + READ_COILS;
response[2] = error;
response[4] = CRC8_Hi(crcl6);
response[3] = CRC8_Lo(crcl6);
for(uint8_t j = 0; j < 5; j++){
EUSART1_Write(response[j]);
}
}
}

```

```

else if(RxBuffer[1] == READ_INPUT_REGISTERS){
uint16_t crc16;
crc16 = MODBUS_CRC16_TX(3);
response[0] = SlaveAddress;
response[1] = 0x80 + READ_INPUT_REGISTERS;
response[2] = error;
response[4] = CRC8_Hi(crc16);
response[3] = CRC8_Lo(crc16);
for(uint8_t j = 0; j < 5; j++){
EUSART1_Write(response[j]);
}
}
else if(RxBuffer == WRITE_MULTIPLE_REGISTERS){
uint16_t crc16;
crc16 = MODBUS_CRC16_TX(3);
response[0] = SlaveAddress;
response[1] = 0x80 + WRITE_MULTIPLE_REGISTERS;
response[2] = error;
response[4] = CRC8_Hi(crc16);
response[3] = CRC8_Lo(crc16);
for(uint8_t j = 0; j < 5; j++){
EUSART1_Write(response[j]);
}
}
}
void modbus_read_input_registers(uint16_t reg_num){
uint8_t starting_address_Lo, starting_address_Hi, reg_num_Hi, reg_num_Lo,
CRC_read_Hi, CRC_read_Lo,
j, k = 0;
//LA VARIABLE INPUTREGISTER ES LA QUE ES TRANSMET
uint16_t crc16, adc_result, starting_address;
uint16_t inputregister = 0;
frame = 1;
starting_address_Hi = RxBuffer[2];
starting_address_Lo = RxBuffer[3];
reg_num_Hi = RxBuffer[4];
reg_num_Lo = RxBuffer[5];
CRC_read_Hi = RxBuffer[6];
CRC_read_Lo = RxBuffer[7];
if(checkCRC() == true){
//Passem els 2 bytes de 8 bits en un numero de 16 bits
starting_address = starting_address_Hi;
starting_address = starting_address << 8;
starting_address = starting_address | starting_address_Lo;
reg_num = reg_num_Hi;
reg_num = reg_num << 8;
reg_num = reg_num | reg_num_Lo;
//Tot això diria que no cal
/*if(reg_num != 1){
error = ILLEGAL_DATA_VALUE;
frame = 0;
}
if(starting_address != 0){
error = ILLEGAL_DATA_ADDRESS;
frame = 0;
} */
//processament de la trama quan hi ha un error
if(frame == 0){
error_function();
}
//processament de la trama quan és correcta
if(frame == 1){
switch (reg_num)
{
// Declara aquí tots els registres

case 1:
// inputregister = la teva variable

```

```

    ESTAT_SISTEMA_MODBUS=(10<<K_GEN_ST_PORT) | (9<<K_GEN_RS_PORT) |
    (8<<K_FOCUS_PORT) | (7<<K_BOMBA_PORT) | (6<<Nivell_diposits)
    | (5<<ESTAT_ÀMB_GENERACIÓ) | (4<<MANIOBRA_FETA) |
    (3<<POSSIBLES_DANYS_RELES_GENERADOR) | (2<<REVISIO_SISTEMA)
    | (1<<AvisErrorPLC) | AvisEmergenciaPLC;
    inputregister = ESTAT_SISTEMA_MODBUS;
    break;
case 2:
    // inputregister = la teva variable
    inputregister = PotenciaGenerada;
    break;
case 3:
    inputregister = Pinversor_W;
    break;
case 4:
    inputregister = Pbateria_W;
    break;
case 5:
    inputregister = Intensitat_bateria;
    break;
case 6:
    inputregister = VBateria;
    break;
case 7:
    inputregister=PFRENAT_MESURADA;
    break;
case 8:
    inputregister=Pfrenat_W;
    break;
case 9:
    ERROR_SISTEMA_MODBUS=(5<<ERROR_coherencia) | (4<<Error_WATCHDOG) |
    (3<<Error_TEST_I2C) | (2<<ERROR_Frenat)
    | (1<<ERROR_Bateria) | ERROR_Inversor;
    inputregister=ERROR_SISTEMA_MODBUS;
    break;
//LECTURA VARIABLES DE CONTROL
case 10:
    inputregister=Smin;
    break;
case 11:
    inputregister=SB_Sllums;
    break;
case 12:
    inputregister=Sllums;
    break;
case 13:
    inputregister=SB;
    break;
case 14:
    inputregister=Vllums;
    break;
case 15:
    inputregister=Vbaix;
    break;
case 16:
    inputregister=Sllindar;
    break;
case 17:
    inputregister=Plimit_bateria;
    break;
case 18:
    inputregister=Vbateria_carregada;
    break;
case 19:
    inputregister=Test_frenat_PLC;
    break;
case 20:
    inputregister=FRENAT_PLC;

```

```

        break;
    case 21:
        inputregister=MARGE_ERROR;
        break;
    case 22:
        inputregister=Potencia_limit_reles;
        break;
    case 23:
        inputregister=Potencia_limit;
        break;
    case 24:
        inputregister=ZONA_MORTA_SENSORS_DE_CORRENT;
        break;
    case 25:
        inputregister=rendiment;
        break;
    default:
        inputregister = 0;
}

uint8_t inputregisterpart1 = (inputregister >> 8) & 0xFF; // Agafa els 8 bits més
significatius
uint8_t inputregisterpart2 = inputregister & 0xFF; // Agafa els 8 bits menys
significatius
R_T_RS485_PORT=1; //Activem el bit de transmissió de dades
response[0] = SlaveAddress;
response[1] = READ_INPUT_REGISTERS;
response[2] = 2 * reg_num; //2*reg_num_Lo
//response[3] = 0;
response[3] = inputregisterpart1; //canviat de notació big endian a mid-little endian
response[4] = inputregisterpart2; //ull
crc16 = MODBUS_CRC16_TX(5);
response[6] = CRC8_Hi(crc16); //crc em surt 0
response[5] = CRC8_Lo(crc16);
R_T_RS485_PORT = 1;
for (uint8_t jk = 0; jk <= 6; jk++){
EUSART1_Write(response[jk]);
}

R_T_RS485_PORT=0; //Desactivem el bit de transmissió de dades
}
}
}
// NOMES LI PODEM ESCRIURE DOS REGISTRES
void modbus_write_multiple_registers(void){
uint8_t starting_address_Hi, starting_address_Lo, reg_num_Hi, reg_num_Lo,
byte_count, auto_data_Hi, auto_data_Lo, test_data_Hi, test_data_Lo,
test_value_data_Hi, test_value_data_Lo, CRC_read_Hi, CRC_read_Lo;
uint16_t crc16, starting_address, reg_num, auto_data, test_data;
frame = 1; //inicialitzem com a trama correcta
starting_address_Hi= RxBuffer[2];
starting_address_Lo = RxBuffer[3];
reg_num_Hi = RxBuffer[4];
reg_num_Lo = RxBuffer[5];
byte_count = RxBuffer[6];
auto_data_Hi= RxBuffer[7];
auto_data_Lo= RxBuffer[8];
test_data_Hi = RxBuffer[9];
test_data_Lo = RxBuffer[10];
test_value_data_Hi = RxBuffer[11];
test_value_data_Lo = RxBuffer[12];
CRC_read_Hi = RxBuffer[13];
CRC_read_Lo = RxBuffer[14];
if(checkCRC() == true){
//passem a 16 bits
starting_address = starting_address_Hi;
starting_address = starting_address << 8;

```



```

starting_address = starting_address | starting_address_Lo;
reg_num = reg_num_Hi;
reg_num = reg_num << 8;
reg_num = reg_num | reg_num_Lo;
auto_data = auto_data_Hi;
auto_data = auto_data << 8;
auto_data = auto_data | auto_data_Lo;
test_data = test_data_Hi;
test_data = test_data << 8;
test_data = test_data | test_data_Lo;
if(starting_address < 0){ //sempre adreça inici de zero
frame = 0;
error = ILLEGAL_DATA_ADDRESS;
}
/*
if(reg_num != 3 && reg_num * 2 != byte_count){
frame = 0;
error = ILLEGAL_DATA_VALUE;
} */
//processament de la trama quan hi ha un error
if (frame == 0){
error_function();
}
//processament de la trama quan és correcta
if(frame == 1){//trama correcta
R_T_RS485_PORT=1;
response[0] = SlaveAddress;
response[1] = WRITE_MULTIPLE_REGISTERS;
response[2] = starting_address_Hi;
response[3] = starting_address_Lo;
response[4] = reg_num_Hi;
response[5] = reg_num_Lo;
crc16 = MODBUS_CRC16_TX(6);
response[7] = CRC8_Hi(crc16);
response[6] = CRC8_Lo(crc16);
R_T_RS485_PORT = 1;
for(uint8_t j = 0; j <= 7; j++){
EUSART1_Write(response[j]);
}

R_T_RS485_PORT=0;
}
}
//Aquest switch és a emplenar segons cada PIC
switch (reg_num)
{
    //MODE AUTO
    case 3:

        VARIABLES_REARMAMENT = auto_data;
        for(int ia=0; ia<3; ia++){
LECTURA_DEL_uint16_auto[ia]=VARIABLES_REARMAMENT&&0x0001;
VARIABLES_REARMAMENT=VARIABLES_REARMAMENT>>1;
        }

        REARMAMENT=LECTURA_DEL_uint16_auto[2];
        REVISIO_SISTEMA=LECTURA_DEL_uint16_auto[1];
        POSSIBLES_DANYS_RELES_GENERADOR=LECTURA_DEL_uint16_auto[0];

        break;

/* _____MODE__TEST_____ */
    case 4:

        TEST_SISTEMA_MODBUS = (Emergencia||Error)?TEST_SISTEMA_MODBUS:test_data;
        for(int it=0; it<9; it++){
LECTURA_DEL_uint16_test[it]=TEST_SISTEMA_MODBUS&0x0001;
TEST_SISTEMA_MODBUS=TEST_SISTEMA_MODBUS>>1;
        }
}

```

```
K_GEN_RS_PORT=LECTURA_DEL_uint16_test[0];
K_GEN_ST_PORT=LECTURA_DEL_uint16_test[1];
K_BOMBA_PORT=LECTURA_DEL_uint16_test[2];
K_FOCUS_PORT=LECTURA_DEL_uint16_test[3];
Emergencia=LECTURA_DEL_uint16_test[4];
//AQUEST DOS BITS S'ENVIEN A LA VEGADA:
Error=LECTURA_DEL_uint16_test[5];
Error_WATCHDOG=LECTURA_DEL_uint16_test[6];
//
Test_frenat_PLC=LECTURA_DEL_uint16_test[7];

break;
case 5:

    FRENAT_PLC =(Emergencia||Error)?FRENAT_PLC:test_data;

    break;
case 6:

    Smin =(Emergencia||Error)?Smin:test_data;

    break;
case 7:

    SB_Sllums =(Emergencia||Error)?SB_Sllums:test_data;

    break;
case 8:

    Sllums =(Emergencia||Error)?Sllums:test_data;

    break;
case 9:

    SB =(Emergencia||Error)?SB:test_data;

    break;
case 10:

    Vllums =(Emergencia||Error)?Vllums:test_data;

    break;
case 11:

    Vbaix =(Emergencia||Error)?Vbaix:test_data;

    break;
case 12:

    Sllindar =(Emergencia||Error)?Sllindar:test_data;

    break;
case 13:

    Plimit_bateria =(Emergencia||Error)?
        Plimit_bateria:test_data;

    break;
case 14:

    Vbateria_carregada =(Emergencia||Error)?
        Vbateria_carregada:test_data;

    break;
case 15:

    MARGE_ERROR =(Emergencia||Error)?MARGE_ERROR:test_data;
```

```

        break;
    case 16:

        Potencia_limit_reles =(Emergencia||Error)?
            Potencia_limit_reles:test_data;

        break;
    case 17:

        Potencia_limit =(Emergencia||Error)?
            Potencia_limit:test_data;

        break;
    case 18:

        ZONA_MORTA_SENSORS_DE_CORRENT=(Emergencia||Error)?
            ZONA_MORTA_SENSORS_DE_CORRENT:test_data;
        Ibat_inferior=(int)ZONA_MORTA_SENSORS_DE_CORRENT;
        Ibat_superior=(int)ZONA_MORTA_SENSORS_DE_CORRENT*-1;

        break;
    case 19:

        rendiment =(Emergencia||Error)?rendiment:test_data;

        break;

}
}

//Funcions modbus programades
void modbus_function (void){
if(RxBuffer[0] == SlaveAddress){//COMPARA ADREÇA

switch (RxBuffer[1]){//COMPARA FUNCIO
case READ_INPUT_REGISTERS:
modbus_read_input_registers(RxBuffer[4]);//Dada a llegir pel PLC
break;
case WRITE_MULTIPLE_REGISTERS://Escriu en el PIC
modbus_write_multiple_registers();
break;
default:

error = ILLEGAL_FUNCTION;
frame = 0;
break;
}

}

c_control=0;
}

```

### A.3. Codi del sistema de mesura de potència generada

En el present apartat hi ha el codi que s'ha programat pel sistema de mesura de potència generada, a través del Platform.io.

```

//
//SENSOR DE POTÈNCIA GENERADA
//
//Adrià Quintana Tarradas

```

```

//26/01/2023
//Projecte llavor UdG

//llibreries per poder desenvolupar el codi.

/*Com que es treballa amb Platform.io cal incloure
Arduino.h per adequar l'entorn de programació, i poder
treballar amb la ESP32.*/
#include <Arduino.h>

//El Wire.h inclou les funcions de l'I2C.
#include <Wire.h>

//*****CONFIGURACIÓ I2C*****
//Els pins que s'empren de l'I2C per canal de dades i clock
#define SDA_PIN 21
#define SCL_PIN 22
//L'adreça de l'ESP32 treballant com a Esclau.
#define I2C_SLAVE_ADDR 0x18

//*****CONFIGURACIÓ ADC AMB ELS SENSOR ACS758*****
const float V_D_factor = 3.3/(4096-1); //Resolució ADC
const float A_V_factor = 50.0; //Com que la sensibilitat del sensor són 20mV/A
/*que serien 0,05 A/mV, que resulta ser 50 A/V*/

/*PROPIETATS DE L'ALTERNADOR:
-La velocitat màxima de rotació és de 630 rpm.
-Com que hi ha un total de 16 pols -> 8 parells de pols.
-Si ngen=(60*f)/Pp -> fmàx=(Pp*ngen)/60 -> fmàx=84 Hz
*equivalent a un període de 11,90 ms.*/

/*El període de mostreig ha de ser 20 vegades inferior al del
corrent per obtenir un valor correcte del senyal segons els criteri
de Nyquist-Shannon. Cada 595,2 us s'hauria d'adquirir una dada, però
es farà cada 500 us, per tenir millor resolució del senyal en el cas extrem,
que per règims de tramuntana és probable.*/

//Cada 500 us s'adquireix una dada.

volatile int analog_in; //Variable on s'emmagatzema el resultat de la conversió AD.

hw_timer_t * timer = NULL;//Declaració del timer, per fer la interrupció d'adquisició.

volatile uint16_t mostres=0;//Mostres que s'adquireixen per cada període del senyal
de mesura.

volatile bool interrupcio=false;//Flag que indicia si hi ha hagut interrupció.

/* INTERRUPTIÓ DE TEMPS
-----*/

//Funció d'interruptió temportitzada per adquirir dades.
void IRAM_ATTR onTimer() {
  analog_in =(analogRead(A6)-2047); // Llegeix entrada analògica G34, i resta l'offset
del sensor.
  //També es pot posar analog_in = analogRead(34)
  mostres++;//Com que s'ha adquirit una mostra es suma el comptador.
  interrupcio=true;//Es fica el flag d'interruptió a true.
}

/*FUNCIONS DE L'I2C:*/

uint32_t P=0; /*Per inicialitzar l'I2C a la ESP32,
```

```

s'ha d'omplir el buffer de recepció/transmissió.*/

/*Quan li arriba el primer byte amb l'adreça junt amb LSB mode lectura,
retorna el valor de potència generada.*/

//S'envia el valor de potència
uint16_t potencia=0;
byte MSBPotencia=0;
byte LSBPotencia=0;

void onRequest(){ //Funció de la petició de lectura.

/*Amb la petició de lectura el mestre pot habilitar
la lectura de dades del sensor, o posar-lo a dormir.*/

/*L'integer que guarda la potència es divideix entre
dos i s'envia per separat.*/
  Wire.write(MSBPotencia);
  Wire.write(LSBPotencia);
}

char Produccio;
/*Amb la funció de recepció el mestre pot parar l'adquisició de
dades o reanduar-la.*/

void onReceive(int len){//Funció en mode escriptura.
  //Descomentar en cas de proves.
  //Serial.printf("onReceive[%d]: ", len);

  while(Wire.available()){
    /*Mentre el mestre estigui transmetent dades, el bus està disponible
i es reben els bytes. El mestre decideix quan parar, com que es sap al 100%
que es rep un byte es prepara la funció per rebre només un byte.*/
    Produccio=Wire.read(); //Byte de parar la proucció.
    //Descomentar en cas de proves.
    //Serial.printf("%d",Produccio);
  }
  //Descomentar en cas de proves.
  //Serial.println();
}

//VARIABLE RMS PEL VOLTATGE
float VRMS;
/*Càlcul del la velocitat de rotació del generador, a partir de la fórmula:
ngen=(60*f)/Pp -> ngen=const_freq_rpm*frequència.*/

const float const_freq_rpm=7.5;

//Factor de RPM a Vrms per tenir la tensió instantània.
//D'aquesta manera Vrms=Vrms_Nvoltes*const_freq_rpm*frequència.
const float Vrms_Nvoltes= 400.0/480.0;

//Valors de Saparent
float SRMS;

//Vector de càlcul de corrent
const int Mostres_TOTALS=1400;
float Corrent_i[Mostres_TOTALS];

//Variables de control pel càlcul RMS.
int comptador_mostres=0;
int pas_PerZero=0;
float Valor_zero;

```

```

float SumaQuadrats=0;
//VARIABLES RMS
float IRMS;
float frecuencia;

void setup(){

  //Descomentar en cas de fer proves
  Serial.begin(115200);

  /*Inicialització de l'I2C com a esclau, assignant els pins SDA i SCL,
  l'adreça com a esclau, i la freqüència del rellotge per transmetre les dades.*/
  bool comprovar= Wire.begin(I2C_SLAVE_ADDR,SDA_PIN,SCL_PIN,400000);

  //EN cas de proves descomentar:

  /*if (comprovar){

    //Serial.printf("OK a l'adreça: %#x\n",I2C_SLAVE_ADDR);
    }else {Serial.printf("error");}*/

  /*Es declaren les interrupciuous per quan l'adreça enviada amb el bus pel mestre,
  casa amb la de l'esclau*/
  Wire.onReceive(onReceive);//Mode escriptura
  Wire.onRequest(onRequest);//Mode lectura

  //S'emplena el buffer per transmetre en el cas de la ESP32.
  #if CONFIG_IDF_TARGET_ESP32
    //Serial.printf("És una ESP32");
    char message[64];
    snprintf(message, 64, "%u Packets.", P++);
    Wire.slaveWrite((uint8_t *)message, strlen(message));
  #endif

  // Aquestes instruccions que venen, serveixen per inicialitzar el temps d'interrupció,
  i lligar-ho amb el
  //codi que s'ha d'executar cada vegada que se'n produeixi una.
  timer = timerBegin(0, 40000, true); // units to 80MHz/40000 = 2 kHz --> 1/2kHz = 0.5
ms interval de
  //temps múltiple del qual es produirà una interrupció
  timerAttachInterrupt(timer, &onTimer, true); //associem interrupció a que executi ISR
onTimer
  timerAlarmWrite(timer, 1, true);// trigger the interrupt every 1 x 0.5 ms = 0.5 ms
  timerAlarmEnable(timer); //habilita la interrupció

  //S'habilita el wake up, perquè partir del pin14 a nivell alt, el mestre despretja el
  sensor.
  esp_sleep_enable_ext0_wakeup(GPIO_NUM_14,1);

}

void loop(){
  //Detecta interrupció.
  if(interruptio){

    Corrent_i[comptador_mostres]=analog_in*V_D_factor; //Càlcul del valor
    instantani.*A_V_factor

    /*Guardo les dades de corrent en un vector, que en funció del pas per zero es mouen
    les posicions.*/

    if(comptador_mostres>1){ //L'algoritme de càlcul del valor RMS comença a partir de
    la segona mostra.

    /*Es mira que es complixin les condicinions de pas per zero:
    1) Que una mostra sigui més gran que zero i l'anterior menor de zero.
    2) El cas invers, a l'anterior.
    3) O que la mostra sigui zero.*/

```

```

if((Corrent_i[comptador_mostres]>0.0&&Corrent_i[comptador_mostres-1]<0.0)||
(Corrent_i[comptador_mostres]<0.0&&Corrent_i[comptador_mostres-1]>0.0)||
Corrent_i[comptador_mostres]==0.0){

    /*Cada vegada que es passa pel zero es suma la variable, que aquesta permet
determinar el període.*/

    pas_PerZero++;

}

}

comptador_mostres++; //Compte les dades del vector

interrupcio=false;//Es borra el flag de software d'interrupció.
}

//Quan es detecten dos passos per zero, es sap que es té un període d'ona.
if(pas_PerZero==2){

    //A partir del vector resultant del pas per zero, es calcula el valor eficaç.
for (size_t i = 0; i <(comptador_mostres) ; i++)
{

    //El valor eficaç es calcula a partir de la suma dels quadrats de les mostres.

    SumaQuadrats=SumaQuadrats+Corrent_i[i]*Corrent_i[i];

}

    /*Es realitza l'arrel de la suma dels quadrats dividida per les msotres totals.
És a dir el
valor RMS.*/
    IRMS=sqrt(SumaQuadrats/(mostres));

    frecuencia=1.0/((mostres)*0.0005);/*Coneixent les mostres i el temps de
mostreig, es calcula
la freqüència elèctrica en Hz. El temps de mostreig queda determinat per la
tortina d'interrupció
temporitzada.*/

    //Descomentar en cas de fer proves:
    //Serial.printf("IRMS=%f      A      \n      f=%f      Hz\n      Ns=%d      \n",IRMS,
frecuencia,comptador_mostres);

    /*S'inicilaitzen tots els valors de les variables de control de l'operació, per
calular el valor
RMS del següent període*/

    comptador_mostres=0;
    pas_PerZero=0;
    mostres=0;
    SumaQuadrats=0;

}

/*Amb la freqüència del senyal, es pot conèixer el nombre de voltes que proporciona
l'esclau 1, i a
partir del nombre de voltes es coneix la FEM generada per l'alternador, és a dir, el
voltatge. La tensió
generada es calcula a paritr de la corba donada pel fabricant.*/

//Tensió eficaç del molí a partir de la constant donada pel fabricant de l'alternador.

```

```
VRMS=Vrms_Nvoltes*const_freq_rpm*frequencia;

//Càlcul de potència aparent.
SRMS=sqrt(3)*VRMS*IRMS;

/*S'arrodoneix per tal de poder enviar via I2C un integer, perquè el PIC faci les
respectives operacions.*/
potencia=(uint16_t)round(SRMS);

//DESCOMENTAR EN cas de fer proves:
/*Serial.printf("%d W, %x int",potencia,potencia);
Serial.println();*/

//Es separa la potència aparent en dos bytes les dades a enviar.
char *dadesEnviar = (char *) &potencia;
LSBPotencia=dadesEnviar[0];
MSBPotencia=dadesEnviar[1];

//DESCOMENTAR EN cas de fer proves:
//delay(1000);

//En cas de rebre aquest codi es posa a dormir la ESP32 consumint molt poc
if(Produccio==0x3C){
Serial.println("BONA NIT");
esp_deep_sleep_start();//Funció de dormir.
}

}
```



## B. CÀLCULS DE LA INSTAL·LACIÓ

En aquest annex es mostren els diferents càlculs que s'han emprat pel càlcul de la instal·lació elèctrica del molí, és a dir: el dimensionament dels conductors, el dimensionament del condensador que compensa la reactiva i els càlculs de la posada de terra.

### B.1. Càlculs de dimensionament dels conductors

En el dimensionament dels conductors s'han utilitzat dos criteris: el criteri que el corrent màxim del conductor sigui major al corrent nominal del circuit, seguint les Figures 17 i 18, de l'apartat u del capítol tres; i la caiguda de tensió que s'han seguit els criteris de l'IDAE.

Pels càlculs de la intensitat nominal del circuit, en cas de les línies de corrent altern trifàsic, s'ha seguit l'Equació 13.

$$I_{línia} = \frac{P_{línia}}{\sqrt{3} \cdot V_{línia} \cdot \cos(\varphi)} \quad (\text{Eq.13})$$

En aquest cas, s'ha considerat la tensió de línia de com la tensió nominal de l'alternador que és de 400 V, la potència de la línia que és de 1.800 W i el factor de potència que s'ha deixat com a la unitat.

Pels càlculs del corrent altern monofàsic s'ha seguit l'Equació 14.

$$I_{línia} = \frac{P_{línia}}{V_{línia} \cdot \cos(\varphi)} \quad (\text{Eq.14})$$

La tensió de la línia és de 230 V, la potència depèn de la línia que connecten, que en aquest cas es mostra en la Taula 28, i el factor de potència també s'ha deixat com a la unitat.

Pels càlculs del corrent continu s'ha seguit l'Equació 15.

$$I_{línia} = \frac{P_{línia}}{V_{línia}} \quad (\text{Eq.15})$$

La tensió de la línia és de 24 V, i la potència depèn de la línia que connecten, que en aquest cas es també mostra en la Taula 28.

Per la caiguda de tensió, en el cas dels circuits alterns trifàsics, s'ha seguit l'Equació 16.

$$e(\%) = \frac{P \cdot Long.}{Secció \cdot V_{línia} \cdot \gamma_{\theta}} \cdot \frac{100}{V_{línia}} \quad (\text{Eq.16})$$

Per la caiguda de tensió, en el cas dels circuits monofàsics i de corrent continu, s'ha seguit l'Equació 17.

$$e(\%) = \frac{2 \cdot P \cdot Long.}{Secció \cdot V_{línia} \cdot \gamma_{\theta}} \cdot \frac{100}{V_{línia}} \quad (\text{Eq.17})$$

Per les dues equacions, la conductivitat com que el conductor és de material aïllant lliure d'halògens, la conductivitat s'ha donat a un valor de 45,49. En el cas de la potència, la longitud i la secció, també es mostren en la Taula 28.

## B.2. Càlculs de dimensionament del condensador de reactiva.

En el cas del condensador de compensació, s'han tingut en compte la potència activa i reactiva que consumeixen les càrregues, i que es mostren en la Taula 38. En el cas de la potència de la bomba, internament incorpora un condensador que compensa la reactiva, i en el cas dels focus, com que el factor de potència és de 0,95 segons el fabricant, aplicant l'Equació 18, s'han obtingut els i sumant la reactiva de tots els focus, s'han obtingut els valors adjuntats a la taula.

$$Q(\text{VAR}) = 6 \cdot P_{\text{Focus}} \cdot tg(\arccos(\varphi)) \quad (\text{Eq.18})$$

<b>Potència activa de les càrregues (W)</b>	1.400,00
<b>Potència reactiva de les càrregues (VAR)</b>	197,21

Taula 38. Potències activa i reactiva de les càrregues.

Un cop establerta la potència reactiva, s'ha aplicat l'Equació 19, per calcular el condensador, d'on s'ha obtingut un resultat de 11,87  $\mu\text{F}$ . Com que comercialment aquest valor no existeix,

s'ha agafat un condensador amb un valor just per sota d'aquest, per tant s'ha agafat de 10  $\mu\text{F}$ . Altrament, la potència reactiva s'acaba de compensar del tot amb l'inversor.

$$C(\mu\text{F}) = \frac{Q_{\text{càrrega}} \cdot 1 \cdot 10^6}{V_{\text{línia}}^2 \cdot 2 \cdot \pi \cdot \text{freqüència}} \quad (\text{Eq.19})$$

### B.3. Càlcul de posada a terra

Tenint en compte que la posada a terra s'ha dut a terme amb un elèctrode de pica vertical de 2,5 m, i que l'ha resistivitat del terreny s'ha estimat de 500  $\Omega \cdot \text{m}$ , perquè en l'entorn on s'instal·larà el molí, el sòl està compost per sorra argilosa, s'ha aplicat l'Equació 20, d'on s'ha obtingut el resultat de 200  $\Omega$ .

$$R(\Omega) = \frac{\rho}{L} \quad (\text{Eq.20})$$

## C. EXPERIMENTS EN EL LABORATORI

En el present annex es detallen els procediments que s'han seguit en els diferents experiments. Alguns d'aquests experiments s'han anat desenvolupant paral·lelament a la programació del codi, i d'altres, s'ha adaptat la rutina principal d'estudi en un programa per estudiar el seu funcionament. En el primer apartat es mostren els experiments de la xarxa I2C, en el segon apartat s'explica com s'ha experimentat la xarxa de Modbus, en el tercer es comenten les proves en el dispositiu de mesura de potència generada, en el quart s'expliquen els experiments del convertidor analògic a digital computacional amb el mode de mitjana de ràfega, en el cinquè hi ha un experiment en el sistema d'acumulació de potència de frenat, en el sisè es fa un assaig del calibratge dels sensors de corrent connectant un banc de dissipació del frenat, i en l'últim apartat s'ha fet un test del control dels relés.

### C.1. Experiments de la xarxa I2C

Els experiments de la xarxa d'I2C s'han dut a terme seguint diferents etapes. L'objectiu final d'aquest experiment ha consistit en desenvolupar una xarxa I2C robusta.

Al llarg de les experiències, s'ha programat un microcontrolador PIC18f47q10 amb la placa Curiosity. Per analitzar les xarxes de comunicació, ha calgut anar a baix nivell, d'aquesta manera, cal analitzar els polsos del senyals de comunicació en el bus entre dispositius. En el cas de l'I2C, s'ha emprat un oscil·loscopi d'ordinador connectat per USB. Concretament, s'ha utilitzat el model 2204A de la marca PICOSCOPE.

La primera part de l'experiment ha consistit en programar el dispositiu mestre. En aquesta part, s'ha connectat la placa del PIC directament sobre l'oscil·loscopi, i s'ha enviat només l'adreça. Prèviament, es tenia una versió de codi, que s'ha anat modificant al llarg de l'experiment perquè aquesta comunicació funcioni. En aquest experiment, s'ha descobert que hi ha pins del microcontrolador que per funcionar per l'I2C s'han d'habilitar els dos mòduls MSSP, degut a connexions internes del PIC. Originalment, es desitjava que el mestre tingués assignat el canal de dades al pin RC3 i el rellotge al RC4, donada la circumstància el senyal de rellotge funcionava, i el de dades no. A partir d'un mètode d'assaig i error, mantenint els pins, s'ha configurat l'MSSP2, i llavors la comunicació ha funcionat. En la Figura 49, es mostra el muntatge, mentre que en la Figura 50 es mostra el resultat de l'experiment. Com es pot observar en la Figura 50, el sistema envia: la condició d'inici, l'adreça 0xAA, el senyal de reconeixement, indicat per una lletra 'S' al final de la trama, i finalment envia la condició de

parada, que s'indica amb una lletra P. Cal destacar que el senyal vermell és el rellotge i el senyal blau són les dades.

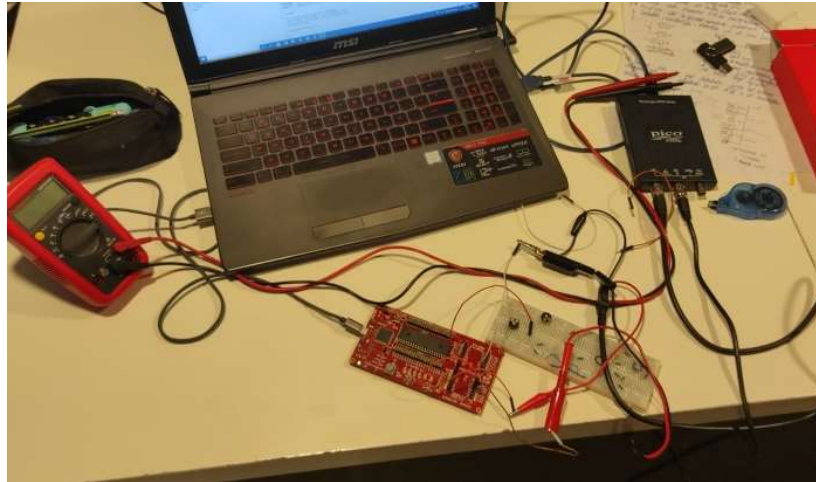


Figura 49. Muntatge de l'experiment amb el PIC funcionant com a mestre.

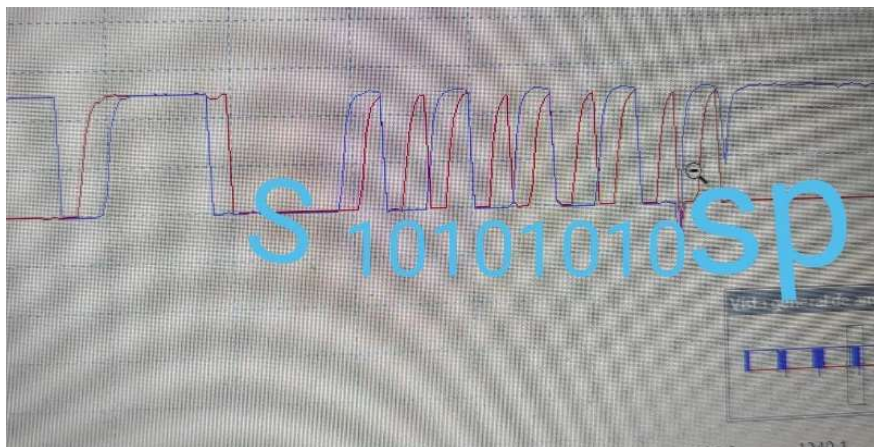


Figura 50. Byte d'adreça I2C.

Un cop s'ha configurat el mestre, el següent pas ha sigut fer una comunicació entre mestre i esclau. Per veure que realment funcionen els dos dispositius, s'han necessitat dos ordinadors amb l'entorn de MPLabX, un amb el programa mestre, i l'altre amb l'esclau, amb els dos programes funcionant com a mode depurador d'errors.

La primera part del segon experiment ha consistit en establir una comunicació mestre-esclau amb l'operació d'escriptura, enviant una comanda, perquè s'encengui un LED, i un cop s'ha enviat aquesta comanda, a la següent s'envia que s'apagui. Aquestes comandes s'han dut a

terme a amb una interrupció temporitzada de quatre segons. En el codi adjuntat a continuació es mostra la rutina de la interrupció.

```
void TMR0_CONTROL_LED(void){  
  
if(!controlled){  
    I2C_Master_Start();  
    I2C_Master_Write(0x30); //7 bit address + Write  
    I2C_Master_Write(0x8A); //ENCENDRE LED  
    I2C_Master_Stop(); //Bit de parada  
    controlled=true;  
}else{  
    I2C_Master_Start();          //Start condition  
    I2C_Master_Write(0x30);      //7 bit address + Read  
    I2C_Master_Write(0xA8);     //APAGAR LED  
    I2C_Master_Stop(); //Bit de parada  
    controlled=false;  
}}  
}}
```

En el codi de l'esclau, en el switch-case de les dades de lectura del buffer, es controlen els booleans que permeten encendre els LEDs.

En la següent experiència s'ha configurat el mode de petició de lectura del mestre de l'I2C. En aquest cas, en l'esclau es varia un potenciòmetre, i el mestre llegeix la variable del potenciòmetre. Quan la variable del potenciòmetre té un valor superior de 1.000 en valor enter, el mestre encén un LED. En la Figura 51 es mostra el muntatge de l'experiment.

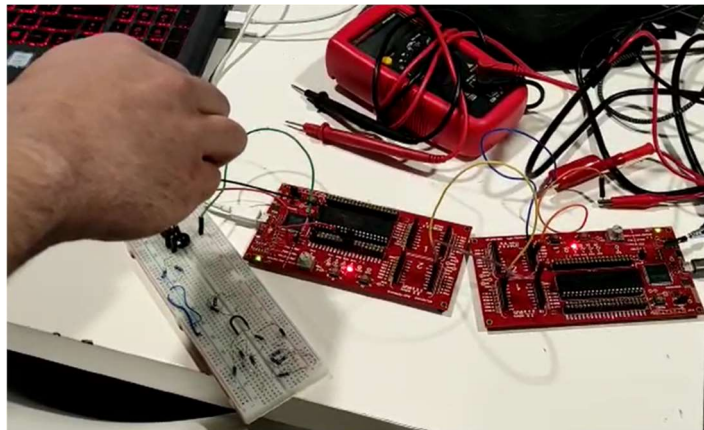


Figura 51. Muntatge de la petició de lectura del PIC.

En la Figura 52, es mostra la trama d'I2C de lectura. Com es pot observar, després de la condició d'inici, el primer byte que s'envia és l'adreça de l'esclau, amb el bit menys significatiu en mode lectura, també es pot observar la seqüència del senyal de reconeixement, i la condició de parada després de rebre el missatge.

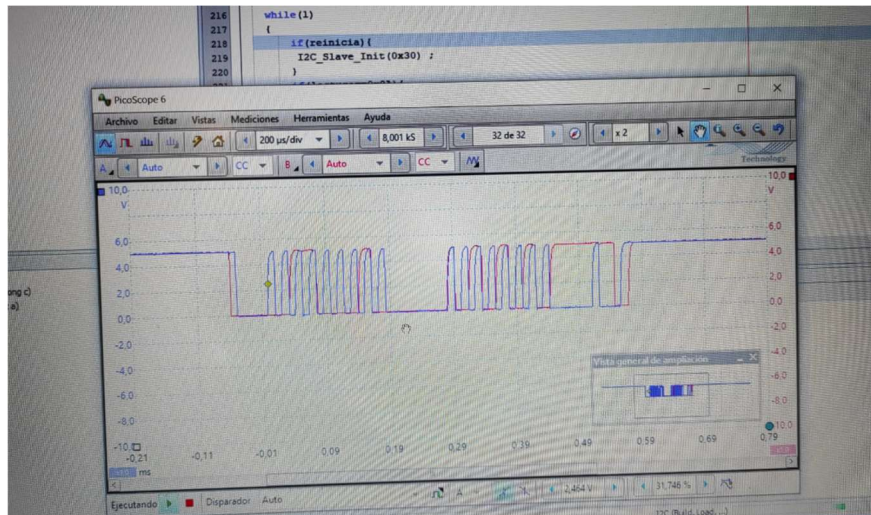


Figura 52. Trama d'I2C de lectura.

## C.2. Experiments amb el Modbus

En el Modbus s'han provat les rutines de lectura i escriptura, amb els modes de monitoratge i test. En aquest cas, els experiments han servit per ajustar el codi de la rutina de Modbus.

Primer de tot, es va fer una rutina sense interrupcions, a partir d'un codi simple on la funció estava en el bucle infinit adaptant les funcions de la EUSART en el protocol de Modbus.

```
while (1)
{
LATICbits.LATIC2 = 0; //Deshabilita la direcció de transmissió del PIC
for(i = 0; i < 2; i++){
rcv = EUSART1_Read();
RxBuffer[i] = rcv;
}
modbus_function();
}
```

Seguidament, en cada registre s'ha anat ajustant perquè llegeixi l'entrada analògica connectada amb un potenciòmetre, i pugui escriure directament sobre sortides digitals del PIC. Per comprovar que el sistema funciona, s'ha emprat el PLC amb les funcions de Modbus programades, un oscil·loscopi digital, i el PIC18F47Q10 amb el convertidor RS485 a UART. Durant aquest experiment, s'ha descobert que el convertidor per rebre i transmetre dades pel PIC s'ha d'habilitar un bit que pugui dur a terme aquesta funció de half-duplex.

En la Figura 53, es mostra el muntatge de l'experiment.



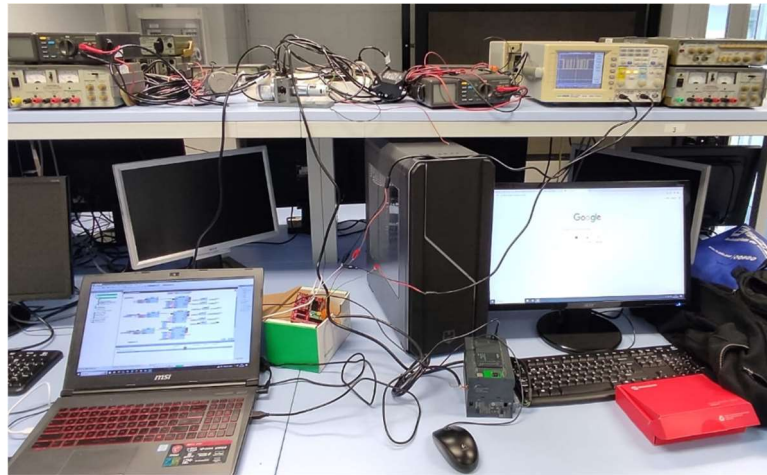


Figura 53. Muntatge de l'experiment de Modbus.

Degut a l'èxit de la lectura i escriptura de dades per aquest experiment, a partir de les interrupcions de la EUSART, s'ha fet la interacció de la rutina de l'esclau a través d'interrupcions, adaptant-la al funcionament del PIC. Per tant, s'ha escrit sobre llindars de control de les variables del PIC, i s'han fet lectures en registres.

En el canal u de l'oscil·loscopi, que es mostra en la Figura 54, es mostra la trama que envia el PLC cap al PIC, mentre que en la Figura 55, es mostra el canal dos de l'oscil·loscopi que consisteix en la resposta del PIC cap al PLC.

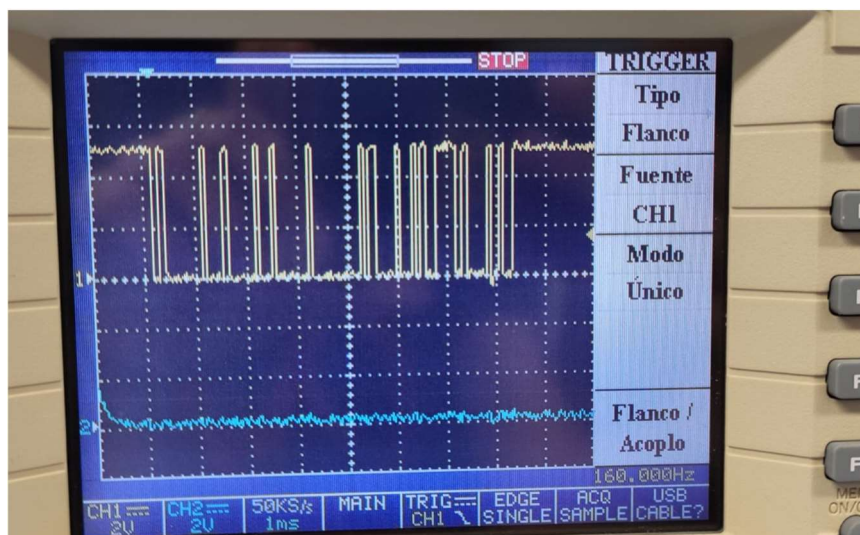


Figura 54. Trama de Modbus del PLC al PIC.





### C.3. Proves amb el dispositiu de potència generada

Pel dispositiu de potència generada, també s'han anat provant les diferents rutines del codi, cada vegada que s'han anat desenvolupant.

El primer que s'ha fet és la comunicació I2C, per aquest motiu, també s'ha utilitzat l'oscil·loscopi PICOSCOPE, el microcontrolador PIC i la ESP32. Tal com es mostra a la fotografia de la Figura 58.

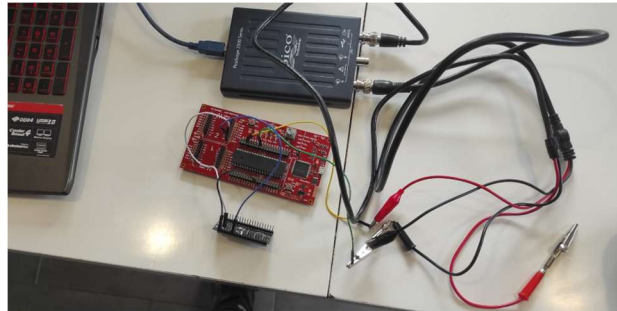


Figura 58. Muntatge de proves de I2C amb la ESP32.

Seguidament, un cop l'I2C ha funcionat, s'ha configurat la targeta PICOSCOPE funcionant com a generador de funcions, i s'ha connectat la seva sortida a l'entrada 34 de la ESP32. En el generador de funcions, s'ha parametritzat un senyal sinusoidal que va de zero a 2,65 V, amb un òfset de 1,65 V. D'aquesta manera, aquest senyal simula la resposta del sensor ACS758, que com s'ha comentat, aquest sensor té un òfset de 1,65 V si s'alimenta a 3,3 V. Bàsicament, en la primera iteració, el sistema ha calculat la freqüència i el valor eficaç de corrent, i a través del port sèrie del Platform.io, s'han llegit els valors calculats.

En la Figura 59, es mostra la resposta que s'ha llegit d'aquest senyal, que restant l'òfset, s'obté un senyal altern de dos volts de pic a pic amb una freqüència de 50 Hz. Aquest senyal té un valor eficaç teòric de 0,707 V i el valor que s'ha obtingut és de 0,694 V. Per tant, l'error és només de 1,84 %, i s'ha donat el resultat com a vàlid.

En la Figura 60, es mostra la resposta del port sèrie amb un idèntic a l'anterior però amb una freqüència de 200 Hz. En aquest cas el senyal obtingut és de 0,704 V, que encara s'aproxima més al valor teòric.

```

I2C > src > main.cpp > onTime
PROBLEMS OUTPUT DEBUG CONSOLE

Ns=40
IRMS=0.695163 A
f=50.000000 Hz
Ns=40
IRMS=0.695429 A
f=50.000000 Hz
Ns=40
IRMS=0.694785 A
f=50.000000 Hz
Ns=40
IRMS=0.694380 A
f=50.000000 Hz
Ns=40

```

Figura 59. Senyal calculat d'un volt de pic i 50 Hz.

```

Ns=10
IRMS=0.704090 A
f=200.000000 Hz
Ns=10
IRMS=0.703903 A
f=200.000000 Hz
Ns=10
IRMS=0.704683 A
f=200.000000 Hz
Ns=10
IRMS=0.703718 A
f=200.000000 Hz
Ns=10

```

Figura 60. Senyal calculat d'un volt de pic i 200 Hz.

Seguidament, s'ha implementat el càlcul de la potència juntament amb la comunicació I2C amb el PIC, i s'ha comprovat com l'enviament de dades, i el mètode per despertar i posar-se a dormir ha funcionat amb èxit.

#### C.4. Proves del convertidor analògic a digital amb el mode mitjana de ràfega

Per saber de forma exacta si el mode de mitjana de ràfega funciona, s'ha fet un experiment seguint la mateixa metodologia que en l'experiència anterior. En aquest cas, s'ha connectat la sortida del generador de funcions de la tarja oscil·loscopi PICOSCOPE en l'entrada RA1 configurada com a analògica en el PIC. El senyal de sortida s'ha fet a través d'una ona sinusoidal amb òfset. Si el mode de mitjana de ràfega funciona de manera correcte, s'hauria de veure a través del depurador d'errors, que el senyal llegit té el valor de l'òfset, que és el valor mitjà del senyal.

Per aquest experiment, el codi utilitzat és una versió reduïda a l'explicat en l'apartat nou del capítol cinc. Només es llegeix una entrada, i no es canvia el canal del multiplexor.

En la primera experiència, s'ha configurat el generador de funcions amb un senyal totalment continu amb un valor de 1,5 V. El PIC com a resultat de la conversió, ha obtingut un valor en hexadecimal 0x0133, que aplicant l'Equació 21, que converteix aquest valor en decimal a un valor de voltatge del senyal del generador de funcions. El resultat ha sigut de 1,5 V, per tant, per un senyal continu ideal funcionaria.

$$V_{mitjà}(V) = Valor\ enter(decimal) \cdot \frac{5\ V}{1023} \quad (Eq.21)$$

El següent pas, ha consistit en sumar en aquest mateix senyal, un senyal altern sinusoidal amb una forta desviació del valor mitjà, en aquest cas, com es mostra en la Figura 61 s'ha aplicat una tensió de un volt de pic a pic, amb una freqüència de 50 Hz.

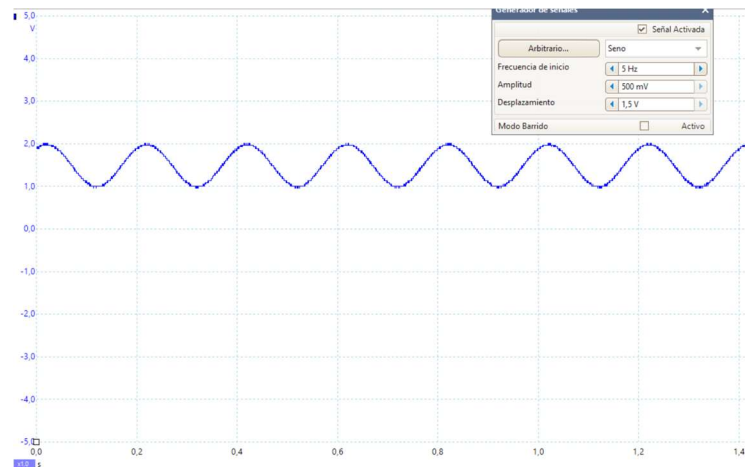


Figura 61. Senyal del generador de funcions de 1 V de pic a pic, 1,5 V d'òfset i 5 Hz.

En aquest cas, els valors obtingut no equivalen al valor exacte de l'òfset, però han sigut molt propers a aquest. Com que aquest senyal és un cas bastant extrem, perquè la variació respecte la mitjana és de 0,5 V, en tenir un error de desviació petit, s'ha deixat com a vàlid. En la Figura 62, es mostra un exemple d'un valor que s'ha obtingut amb el depurador d'errors. Aplicant l'Equació 21, el valor mesurat és de 1,57 V.

```

void Lectura_de_valorMitja_AD(void) {
    MitjanaMostres=((uint16_t)((ADFLTRH << 8) + ADFLTRL));
    ValorAdquirit=true;
}

```

Name	Type	Address	Value
MitjanaMostres	unsigned short	0x3D	0x0142

Figura 62. Valor enter en hexadecimal del senyal de 1 V de pic a pic, 1,5 V d'òfset i 5 Hz.

Aquest experiment s'ha repetit per un cas no gaire extrem, en aquest cas el senyal és de 0,5 V de pic a pic, i la freqüència és de 100 Hz, com es mostra en la Figura 63.

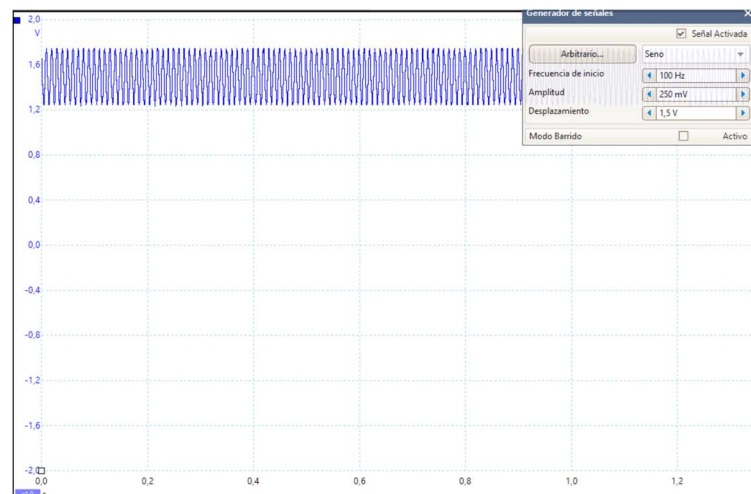


Figura 63. Senyal del generador de funcions de 0,5 V de pic a pic, 1,5 V d'òfset i 100 Hz.

En aquesta experiència, els valors encara han sigut més exactes que en el cas anterior. S'han obtingut valors que ronden al de l'exemple que es mostra en la Figura 64. El valor que es mostra és de 0x13C, que aplicant l'Equació 21, el valor és de 1,54 V.

```

240 void Lectura_de_valorMitja_AD(void) {
241     MitjanaMostres=((uint16_t)((ADFLTRH << 8) + ADFLTRL));
242     ValorAdquirit=true;
243 }
244
245
246
247
248
249
250 void main()
251 {
252     SYSTEM_Initialize();
253     INTERRUPT_GlobalInterruptEnable();
254 }

```

Name	Type	Address	Value
ValorAdquirit	bool	0x9	0x0
VBatena	unsigned int	0x6C	0x013C
SPR_20UF	SPR	0xF1	0x0
PotenciaGenerada	unsigned int	0x45	0x0000
MitjanaMostres	unsigned short	0x3D	0x013C
MitjanaMostres	unsigned short	0x3D	0x013C

Figura 64. Valor enter en hexadecimal del senyal de 0,5 V de pic a pic, 1,5 V d'òfset i 100 Hz.

### C.5. Proves del frenat elèctric

En aquest prova s'ha connectat un potenciòmetre en l'entrada analògica RA0, per tal de simular el sistema d'acumulació del frenat. Com que l'entrada analògica va de 0 a 1.023, s'ha dividit el rang de valors que pot tenir el convertidor en sis intervals de 170. Seguidament, s'ha ajustat el PWM amb un factor de càlcul, perquè el seu valor sigui de 1.023, degut a que aquesta variable és de 10 bits. Per tant, en el càlcul es resta el valor mínim de connectar cada banc amb el valor de l'entrada analògica, perquè la variable de control tingui un valor màxim de 170 i posteriorment aquest valor se li aplica un factor perquè arribi a 1.023. Per evitar tenir errors d'arrodoniments, a causa de les variables enteres, s'han dut a terme un seguit d'ajustaments en el factor. Cal destacar, que el valor màxim real és de 1.022. El fragment de codi adjuntat és el que s'ha utilitzat durant les proves.

```
else if(consum>170 & consum<=240){
    CLCI2_PORT=0;
    CLCI1_PORT=1;
    CLCI0_PORT=1;
    dutyPWM=(((uint16_t)consum-170))/2)*146)/5;
    PWM3_LoadDutyValue(dutyPWM);
}
```

La placa Curiosity incorpora un potenciòmetre que es connecta en l'entrada RA0. Per simular cada sortida s'han connectat els sis CLC en un set segments, on es pot veure com a mesura que augmenta el potenciòmetre es connecta cada segment, i varia la intensitat lumínica per efecte del PWM. En la Figura 65 es pot veure aquest muntatge.

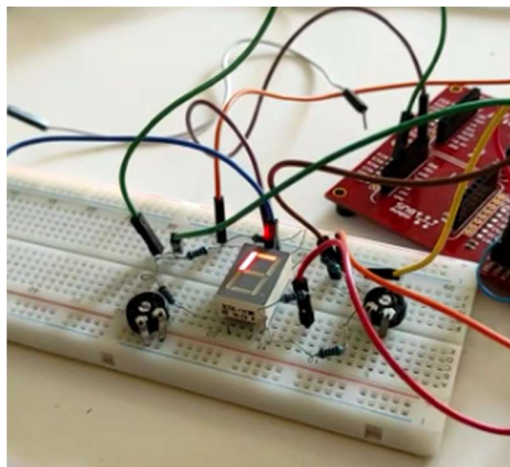


Figura 65. Muntatge de les proves del frenat.



## C.6. Lectura dels sensors de corrent

Tant bon punt han arribat els sensors de corrent ACS758, s'han fet proves per tal de conèixer la sensibilitat real d'aquests de sensors i com el PIC llegeix els senyals que proporcionen. Aquestes experiències s'han desenvolupat: amb el microcontrolador PIC, amb una bateria de 24 V, amb una passarel·la de UART a USB, amb el sensor de corrent mesurant la intensitat que circula entre la bateria i un banc de frenat format per bombetes halògenes, i amb un amperímetre que permet conèixer el corrent que alimenta el banc de dissipació. A més, aprofitant aquest muntatge s'ha pogut saber la potència real que consumeix un banc de frenat per diferents valors del PWM. En la Figura 66, hi ha una foto del muntatge de l'experiment.

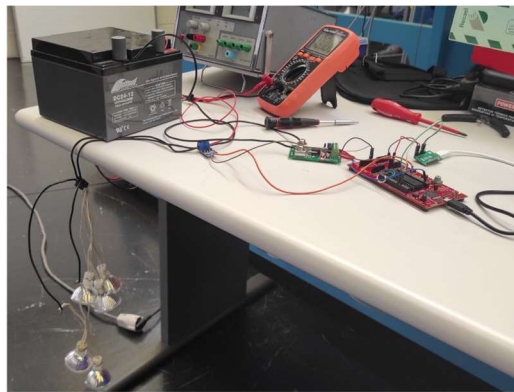


Figura 66. Muntatge de l'experiment.

Per estudiar els valors del sensor en funció del corrent que circula, s'ha fet ús del programa que s'ha utilitzat en l'experiment de l'apartat anterior, que estudia la rutina de frenat, però en aquest cas només s'ha emprat un banc de dissipació de frenat. Els valors en enter s'han adquirit a través de les lectures del port sèrie, visualitzades gràcies al Putty en un ordinador. En Figura 67 hi ha els resultats de la recta de calibratge que s'ha obtingut.

La corba de calibratge s'ha dibuixat a partir de les lectures de l'amperímetre, i valors de corrent en enter subministrats pel PIC, que s'han enviat pel port sèrie i s'han llegit pel putty per uns mateixos valors de corrent. Cal destacar que el convertidor ADC del microcontrolador és de deu bits, per tant, els números en enter van des de 0 a 1.023. Per tant, en cas de donar-se 100 Ampers, pel port sèrie s'hauria de visualitzar el número 0, i per valors de 100 Ampers 1.023. Per tant, per programa s'haurà de restar l'òfset per tenir el valor instantani correcte.

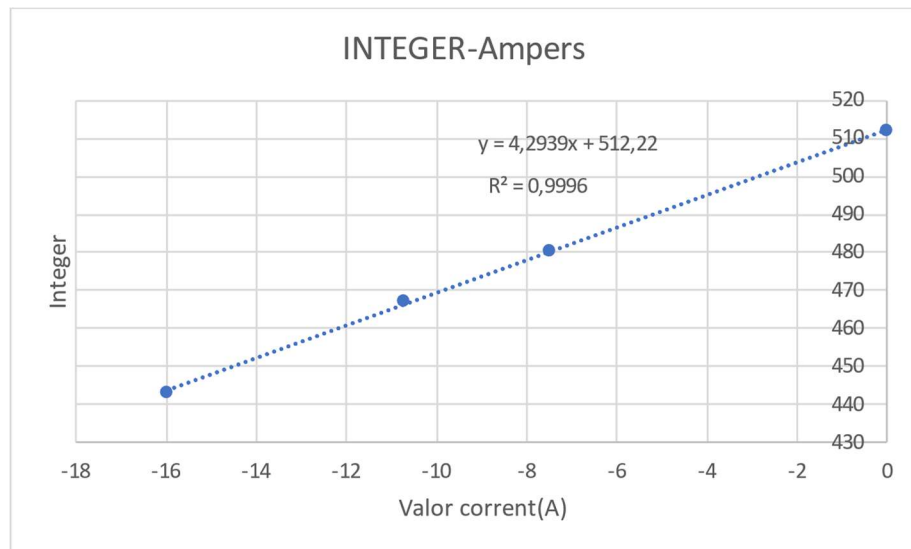


Figura 67. Recta de calibratge del sensor.

En la mesura del ADC, s'ha emprat el mode de mitjana de ràfega, i els augments del potenciómetre s'han fet de forma molt poc pronunciada, per poder captar de forma correcta els valors de corrent. Com que només es llegeix una entrada, el codi emparat per la lectura analògica, és el mateix que hi ha adjuntat en l'apartat nou del capítol cinc, però sense canviar l'entrada del multiplexor, ni s'ha restat l'òfset.

Altrament, quan el banc està funcionant amb el PWM al màxim, la intensitat adquirida és de 13,11 A, que si es multiplica per la tensió de la bateria resulten 314 W. Per tant, els bancs de frenada consumeixen 300 W.

En la Figura 68, es mostra com s'ha pres la temperatura del commutador MOSFET.



Figura 68. Mesura de temperatura del MOSFET.



Com que els bancs es connecten a partir de commutadors MOSFET, s'ha estudiat amb un mesurador de temperatura per infrarojos, la temperatura del MOSFET quan el frenat està funcionant al màxim. En aquest cas, passat un temps de cinc minuts, s'ha vist com la temperatura es mantenia constant a un valor de 28 °C. Per tant, com que està dins del rang de funcionament, s'ha comprovat com els MOSFET poden funcionar en aquest sistema.

### C.7. Proves de control dels relés

Pel control dels relés s'ha fet una simulació provant de manera parcial els diferents sensors a través de potenciòmetres. En aquest experiment, es monitoritza la potència generada i l'estat de les bateries, amb els valors de corrent i de tensió, a partir de simulacions parcials. És a dir, fixant els valors, i només variant un paràmetre, com pot ser la potència generada. A partir de l'encesa de llums LED, s'ha estudiat si l'algoritme funciona de manera correcte. En aquest cas, només s'han utilitzat tres entrades analògiques. Una que simula la potència generada, l'altre que simula la tensió de la bateria, i una última que simula el corrent. Els valors també s'obtenen a partir d'una mitjana de ràfega, seguint el codi adjuntat. En el cas de la potència generada, com que els valors van de zero a 1.800 W, en situacions ideals, el valor obtingut pel convertidor analògic-digital s'ha multiplicat per 2, i així els valors van de zero a 2.046 W. En el cas del corrent de la bateria, els resultats van de -75 A a 75 A, d'aquesta manera, en el resultat de la conversió se li resta 512, i llavors, se li multiplica per 14 i es divideix entre 100. Així el rang de mesura simulada va de -71 A a 71 A. Finalment, per la tensió del conjunt de bateries, com que es poden obtenir valors superiors a 24 V, s'ha fixat un màxim de 30 V. Per tant, en el resultat de la conversió s'ha multiplicat per tres, i s'ha dividit per cent.

```
case 0x00:
    //PIN RA0
    PotenciaGenerada =MitjanaMostres*2;
    CanalLectura++;
    ADPCH = CanalLectura;
    break;
case 0x01:
    //PIN RA1
    Intensitat_bateria =(int) ((MitjanaMostres-512)*14)/100;
    CanalLectura++;
    ADPCH = CanalLectura;
    break;
case 0x02:
    //PIN RA2
    VBateria=MitjanaMostres*3/100;
    CanalLectura++;
    ADPCH = CanalLectura;
    break;
```

Altrament, s'ha testejat el nivell dels dipòsits per encendre la bomba amb les interrupcions per perifèrics, tal com es mostra en el muntatge de la Figura 69.

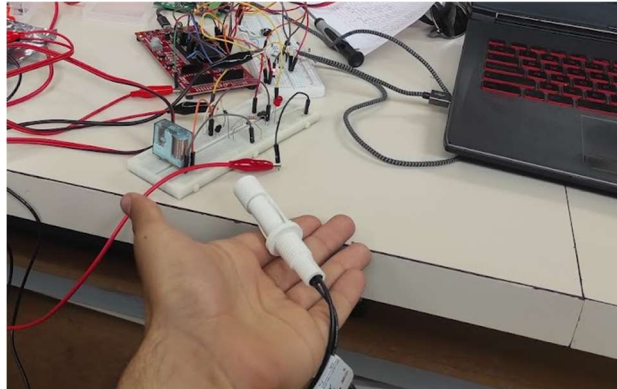


Figura 69. Muntatge de l'experiment del control dels relés.

Al llarg d'aquest experiment s'ha anat modificant el codi i retocant. Aquesta experiència ha servit per provar que el mètode de les lectures analògiques per la mitjana de ràfega, juntament amb les interrupcions per perifèrics funcionen, però és cert que els valors llindars del codi s'han adaptat a les situacions del molí, per aquest motiu s'han de realitzar experiments amb els elements reals de potència del molí i amb un senyal elèctric variant, com és el corrent de l'alternador del molí, a partir de les lectures dels sensors que s'apliquen. Per aquest motiu, en la posada en funcionament, s'han d'acabar d'ajustar els llindars d'aquest algoritme.

## D. POSADA EN FUNCIONAMENT

El sistema de control del consum de potència s'ha d'instal·lar i acabar de parametritzar en la posada en funcionament. Aquesta posada en funcionament es durà a terme quan: els diferents sistemes de control de la turbina estiguin programats i a punt d'instal·lar-se, el PLC estigui programat, es tingui tot el material incloent: el material elèctric, amb les PCB que s'han demanat el dia 25 de maig de 2.023; i estigui muntada l'estructura mecànica, amb el generador instal·lat i amb el sistema mecànic del moviment de l'angle d'inclinació de les pales acoblat a l'eix del generador.

Actualment, el sistema de control de potència està programat. Només falta soldar els components a les PCB de la placa Curiosity i de la ESP32.

Degut a que el muntatge de l'estructura mecànica no està acabat, no es pot donar una data límit per començar la posada en funcionament.