

## Treball final de grau

**Estudi: Grau en Enginyeria Electrònica Industrial i Automàtica**

**Títol: Panell informatiu per esdeveniments esportius**

**Document: 1. Memòria**

**Alumne: Roger Gasull Vilagran**

**Tutor: Sr. Albert Figueras Coma**

**Departament: Enginyeria Elèctrica, Electrònica i Automàtica**

**Àrea: Enginyeria de Sistemes i Automàtica**

**Convocatòria (mes/any): juny/2020**

**ÍNDEX**

1 INTRODUCCIÓ .....	3
1.1 Antecedents .....	3
1.2 Objecte.....	3
1.3 Abast.....	3
2 CARACTERÍSTIQUES BÀSIQUES DEL PANELL .....	4
3 ELECCIÓ DE L'ELECTRÒNICA .....	9
3.1 Leds .....	9
3.2 Drivers.....	11
3.2.1 Selecció R <sub>SET</sub> .....	13
3.2.1 Velocitat de refresc .....	14
3.3 Microcontrolador.....	15
3.4 Comunicacions.....	18
3.5 Font d'alimentació .....	18
3.6 Cablejat i connectors.....	20
4 DESCRIPCIÓ DEL HARDWARE .....	23
5 DESCRIPCIÓ DEL SOFTWARE.....	26
5.1 Lliberies.....	27
5.2 Estructura del programa .....	27
5.2.1 Interrupcions .....	29
5.3 COMUNICACIONS .....	30
6 INTERFÍCIE D'USUARI .....	31
7 RESUM PRESSUPOST .....	33
8 CONCLUSIONS .....	34
9 RELACIÓ DE DOCUMENTS .....	35
10 BIBLIOGRAFIA.....	36
11 GLOSSARI .....	37
A CODI PROGRAMA.....	38
B MANUAL D'INSTRUCCIONS .....	75

C CÀLCULS ..... 77

## **1 INTRODUCCIÓ**

### **1.1 Antecedents**

Actualment cronoexagon.com cronometra curses esportives amb dos sistemes independents. Per una banda trobem el sistema de recopilació i gestió de dades de cada participant, que està format per: el hardware TimingSense, que recull les lectures dels xips; el servidor Sportmaniacs, que emmagatzema aquestes lectures; i el software RaceTec Timing, que combina les lectures amb les dades dels participants per tal d'obtenir la classificació final. Per altra banda, el rellotge digital marca el temps durant l'esdeveniment. Aquest és controlat per un comandament de radiofreqüència amb un botó Start/Stop.

### **1.2 Objecte**

Per modernitzar el cronometratge, em proposo combinar els dos sistemes de tal manera que, en una sola pantalla, es mostri el temps de cursa, el temps del primer classificat i el temps i nom del participant que entra a l'arribada, així cada atleta podrà veure la diferència respecte al primer. Altrament, amb un software, es podrà gestionar altra informació com el pas d'atletes pels diferents punts de control del circuit, els patrocinadors, els premis per categories o altra informació rellevant que també es podrà mostrar per pantalla. D'aquesta manera la informació de la cursa serà accessible a tot el públic.

### **1.3 Abast**

A partir d'aquesta idea, mitjançant una matriu de LEDs, tipus DIP d'alta lluminositat de color vermell, i un microcontrolador realitzaré una pantalla, d'aproximadament 100x60 cm i a doble cara, que visualitzi totes les dades esmentades. A part, caldrà implementar un sistema de comunicació per poder comunicar-ho amb l'ordinador.

## 2 CARACTERÍSTIQUES BÀSIQUES DEL PANELL

El panell informatiu per esdeveniments esportius inclou la pantalla, la placa de control, la font d'alimentació, cables, estructura i els elements necessaris per al muntatge. El panell pot mostrar el temps de cursa, el temps de cada participant pel pas de meta o algun dels punts de control, o també es pot deixar fixe el primer participant, d'aquesta manera la resta d'atletes poden veure la diferència. A més a més inclou un quadre on introduir text i d'aquesta manera l'organitzador pot informar dels premis, patrocinadors o qualsevol informació que consideri rellevant.

El panell consta d'un total de 5760 píxels (LEDs), distribuïts en 144 LEDs en l'eix horitzontal i 40 LEDs en l'eix vertical. Aquest eix vertical està dividit en 5 files d'igual mida, és a dir, de 8 LEDs cada una. Les dues de dalt estan reservades pel rellotge que pot ser ascendent o descendent segons desitgi l'usuari. I les tres de sota serveixen per mostrar la resta d'informació. Aquestes tres de sota tenen dues distribucions diferents, segons si agafem la informació del software de cronometratge o si estem mostrant text personalitzat. En les dues següents taules podem veure les dues distribucions esmentades.

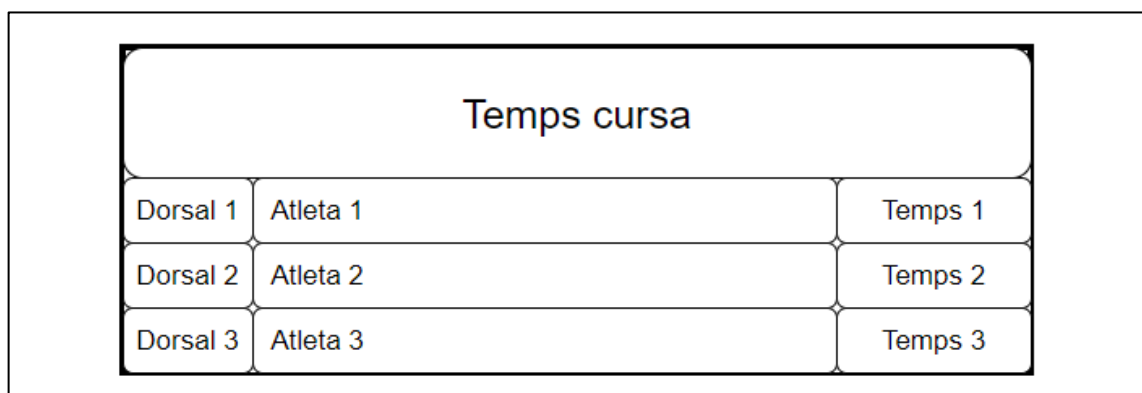


Figura 1. Distribució del panell amb informació del software.

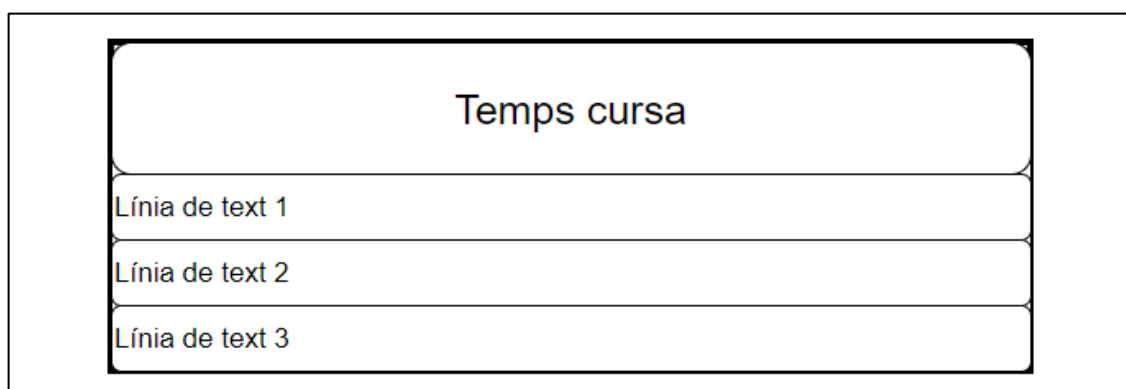
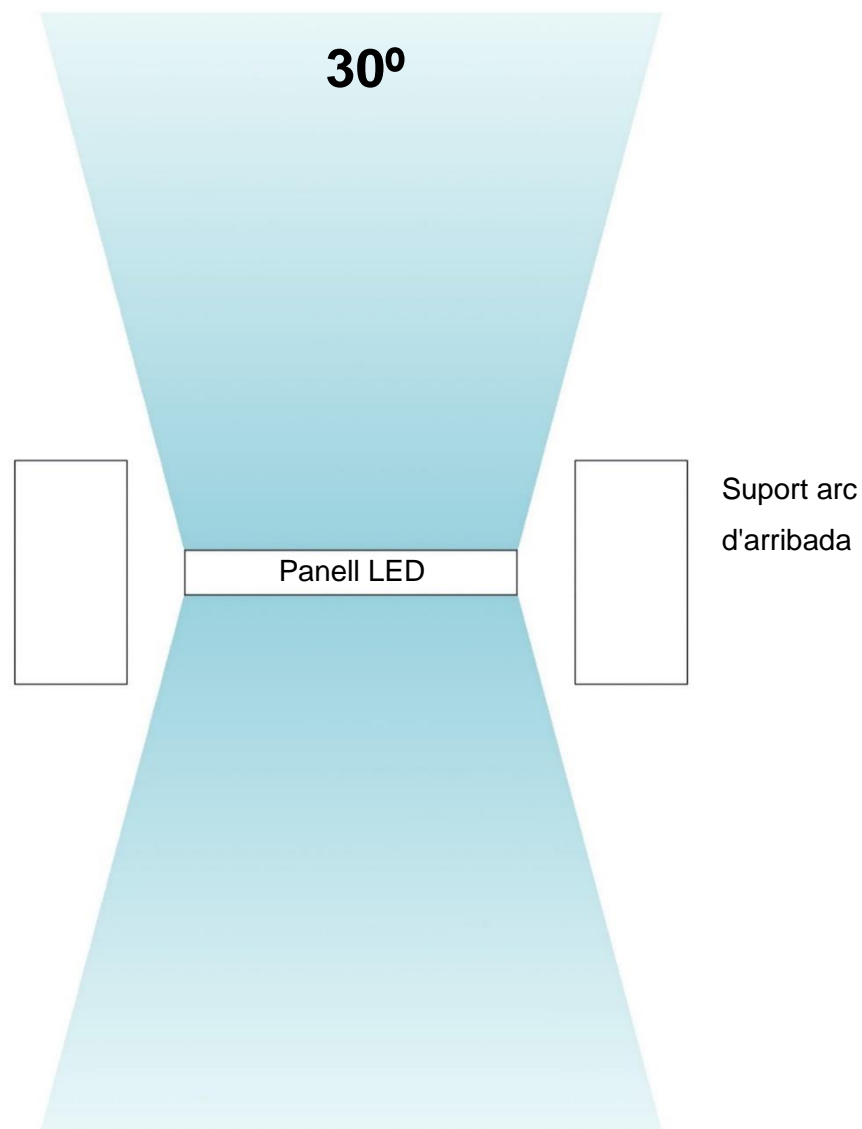


Figura 2. Distribució del panell amb informació personalitzada.

A més a més, incorpora LEDs de color vermell d'alta brillantor que permeten visualitzacions perfectes inclús a l'exterior i contra la llum directa del sol. La interfície d'usuari permet al panell l'ajust de la intensitat de la brillantor garantint una visibilitat perfecta en qualsevol condició ambiental. Els LEDs elegits i descrits en l'apartat corresponent, s'han buscat amb un angle d'obertura de 30 graus, això és degut a les especificacions tècniques del projecte. Aquest panell està pensat per anar penjat a l'arc d'arribada i aquest, incorpora dos peus de suport, els quals redueixen el camp de visió. Per tant, s'han buscat uns LEDs que focalitzin la seva llum a 30 graus i no es malgasta cap a direccions on no es pot accedir físicament per veure el panell. A la següent figura ho podem veure, on els dos rectangles dels costats representen els suports, el rectangle central la posició del panell i el polígon de color blau el camp de visió.



*Figura 3. Col·locació del panell amb l'arc d'arribada.*

La distància de visió serveix per definir l'interval entre dos píxels. Així doncs s'ha optat per un pitch p5 (5mm) que significa que la seva visió serà òptima a partir de dos metres de distància, la qual és molt difícil que sigui menor en una cursa. En la següent figura podem veure una representació de la distància de visió en funció del pitch.

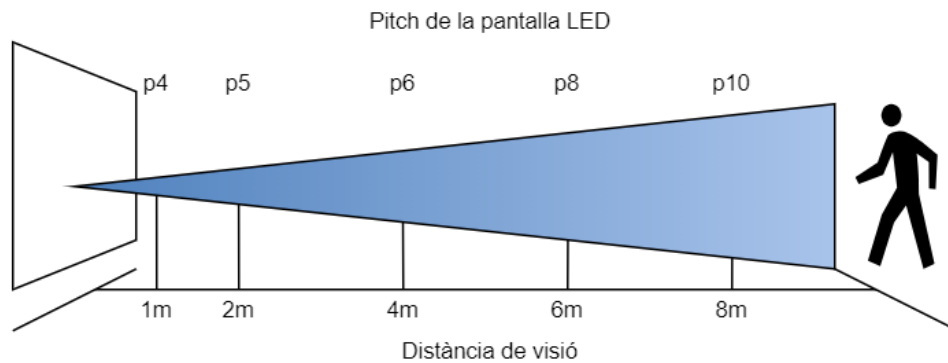


Figura 4. Pitch en funció de la distància.

Pel disseny de les lletres de les 3 files de baix, s'ha utilitzat una mida de 5x7 píxels, aquest és una mida llegible des d'una distància considerable, també s'ha intentat fer lletres simples i majúscules per tal que no es puguin confondre. A continuació les podem veure, on cada quadre groc representa un dels LEDs.

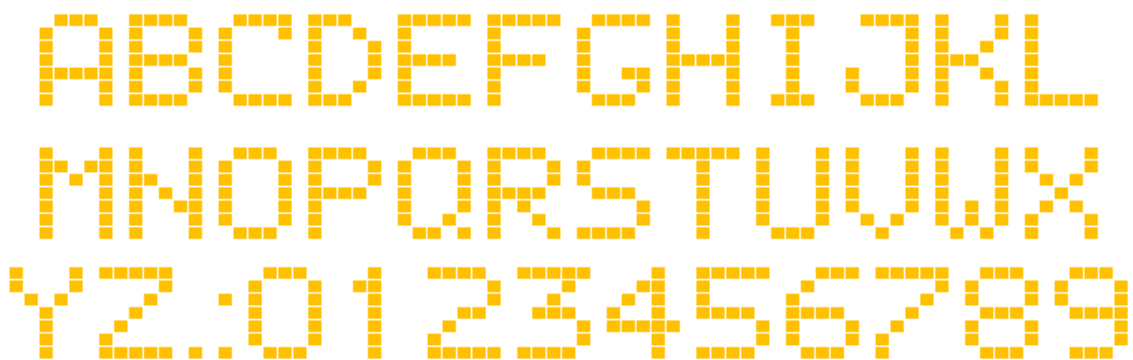


Figura 5. Disseny caràcters.

Cal no oblidar que la part principal del panell és mostrar el temps de cursa, per aquest motiu s'ha decidit que els números del rellotge siguin el doble de grans que els altres, de tal manera que tenen una mida de 8x16 píxels. Ho podem veure a la següent figura.



Figura 6. Disseny números del rellotge.

Com a característica extra s'ha afegit un control de lluminositat, en experiències cronometrants s'ha detectat la necessitat d'aquest, ja que en les curses nocturnes el rellotge de meta entre en conflicte amb el servei de fotografia. Això és degut al fet que els rellotges actuals tenen la lluminositat fixa i definida al màxim perquè es puguin veure amb la llum solar, però no tenen en compte les curses nocturnes on aquests rellotges enlluernen a la càmera fent il·legible el temps i fent malbé la foto. A continuació podem veure una fotografia de l'arribada d'una cursa on començava a fer-se de nit.



Figura 7. Enlluernament rellotge actual.

També té una fiabilitat molt alta, ja que els components que s'han escollit són de qualitat, pensats i dissenyats per aquest propòsit i amb un ampli rang de temperatura de funcionament de  $-30^{\circ}\text{C}$  a  $70^{\circ}\text{C}$ . S'ha agafat aquest rang perquè és el pitjor de tots els components. A la següent taula podem veure tots els rangs de temperatures de tots els components descrits en els pròxims apartats i com la font d'alimentació ens marca el nostre rang.

Element	Rang de funcionament	Element	Rang de funcionament
Font d'alimentació	$-30^{\circ}\text{C}$ a $70^{\circ}\text{C}$	Mòdul Wi-Fi	$-40^{\circ}\text{C}$ a $85^{\circ}\text{C}$
Cable pla IDC	$-40^{\circ}\text{C}$ a $105^{\circ}\text{C}$	Driver	$-40^{\circ}\text{C}$ a $85^{\circ}\text{C}$
T.B. alimentació	$-40^{\circ}\text{C}$ a $110^{\circ}\text{C}$	LED	$-40^{\circ}\text{C}$ a $100^{\circ}\text{C}$
Connector pins	$-40^{\circ}\text{C}$ a $150^{\circ}\text{C}$	Mòdul RTC	$-40^{\circ}\text{C}$ a $85^{\circ}\text{C}$
Teensy 3.2	$-40^{\circ}\text{C}$ a $85^{\circ}\text{C}$	Condensadors	$-40^{\circ}\text{C}$ a $85^{\circ}\text{C}$
Transceptor de bus	$-40^{\circ}\text{C}$ a $85^{\circ}\text{C}$	Resistències	$-40^{\circ}\text{C}$ a $85^{\circ}\text{C}$

Taula 1. Rang de funcionament dels diferents elements.



Pel que fa a la connectivitat, aquest incorpora un mòdul Wi-Fi per accedir a ell i poder-lo controlar, més endavant hi ha un apartat dedicat a la connectivitat on està més detallat.

Finalment, el panell té unes dimensions de 148,2cm per 58,2cm i incorpora un cable d'alimentació schuko de 3m de longitud. A continuació trobem una taula amb el resum de les característiques.

<b>Paràmetre</b>	<b>Valor</b>
Angle de visió	120° (90°+30°)
Color del LEDs	Vermell
Píxels per costat (LEDs)	5760 (144x40)
Pitch	5mm
LEDs total	11.520
Numero de drivers	180
Numero de drivers en cascada	18
Velocitat de refresc	100Hz
Connectivitat	Mòdul Wi-Fi
Rang de funcionament	-30°C a 70°C
Alimentació	90 ~ 264VAC
Dimensions	148,2x58,2cm

*Taula 2. Resum característiques.*

### 3 ELECCIÓ DE L'ELECTRÒNICA

A continuació es realitza l'estudi i selecció dels diferents components necessaris per a la realització del projecte.

#### 3.1 Leds

Com a element principal del projecte, s'han buscat diferents models de LED per tal d'obtenir el millor resultat, agafant com a paràmetres la lluminositat, durabilitat, color i angle de visió.

Pel que fa al muntatge del LED tenim dos tipus els SMD i els DIP. Els SMD ofereixen una gran amplitud en l'angle de visió i són utilitzats en pantalles d'alta resolució. Pel que fa a aquest projecte, els elements a mostrar en pantalla són números i lletres, pel que fa que els avantatges que presenten els SMD no siguin tan necessàries. En canvi, els tipus DIP ofereixen una millor dissipació de la calor, resistència a impactes i un major brillo, el que els fa l'opció idònia per pantalles de localització exterior com en aquest projecte.

Un altre dels factors clau del LED és la seva lluminositat, ja que si no és l'adequada, la pantalla no es veuria bé amb la presència de la llum solar. En la següent figura podem veure la lluminositat recomanada segons el tipus d'aplicació.

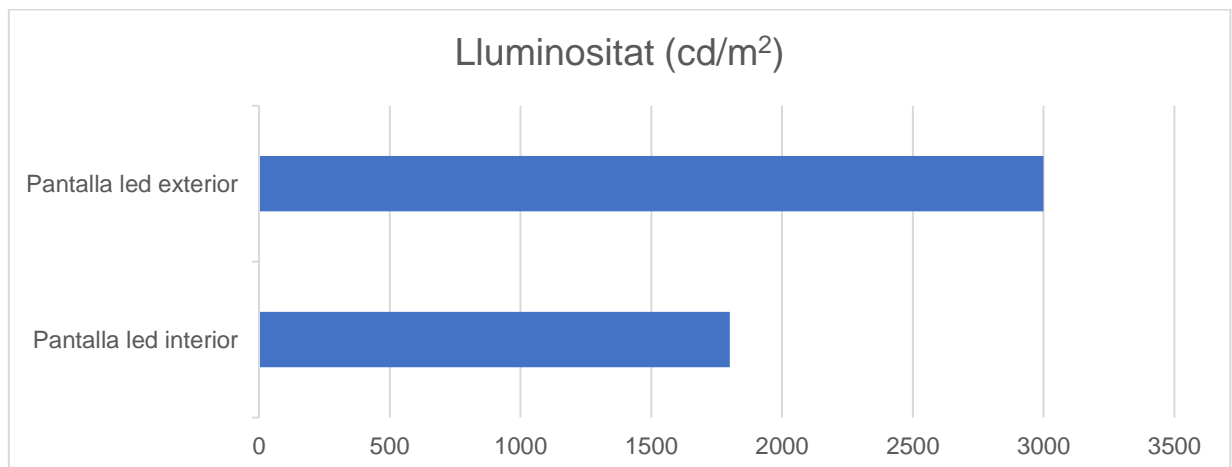


Figura 8. Lluminositat.

Segons les especificacions, la pantalla LED mostra els elements de color vermell, mirant la longitud d'ona estàndard de la figura 9 veiem que el LED escollit hauria de tenir un valor al voltant de 625nm.

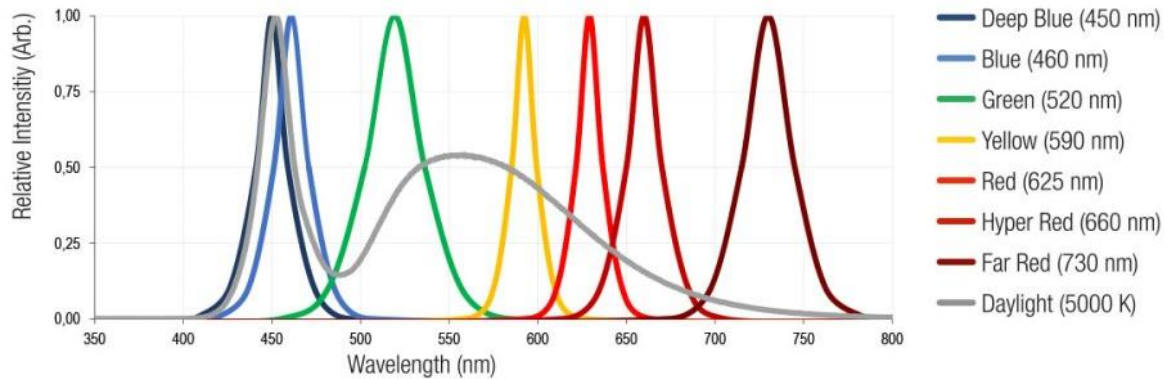


Figura 9. Longitud d'ona estàndard.

Amb les característiques necessàries esmentades, buscant en les diferents marques que hi ha al mercat, com Nichia, Cree, Osram, Epistar, Silan o Lumileds, s'ha trobat i escollit el model C503B de la marca Cree. Aquest LED ha estat dissenyat per ser utilitzat en senyalització viària, el qual porta incorporat epoxy per oferir un millor rendiment a temperatures altes i una millor protecció contra la humitat. A més, porta un tractament de resina per protegir-lo dels rajos UV de la llum solar. A la següent taula podem veure algunes de les característiques, extretes del seu datasheet, que han servit per escollir-lo.

Model	C503B-RCN-CW0Z0AA2		
Mida	5mm		
Longitud d'ona	Mínima	Típica	Màxima
	618nm	624nm	630nm
Lluminositat ( $I_f=20\text{mA}$ )	Mínima		Màxima
	3.000mcd		12.000mcd
Angle de visió	30°		
Intensitat	Mínima	Típica	Màxima
	10mA	20mA	50mA
Intensitat de pic (ample de pols $\leq 0.1\text{ms}$ )	200mA		
Voltatge	5V		
Potència	130mW		
Temperatura funcionament	-40°C → 100°C		

Taula 3. Característiques LED C503B.

A part de les característiques esmentades també presenta una lent transparent, el qual farà la pantalla més agradable i entenedora quan aquest estigui apagat. Finalment, segons un estudi de VisualLed, la marca Cree és una de les que té major fiabilitat, tal com podem veure en el gràfic extret del seu estudi.

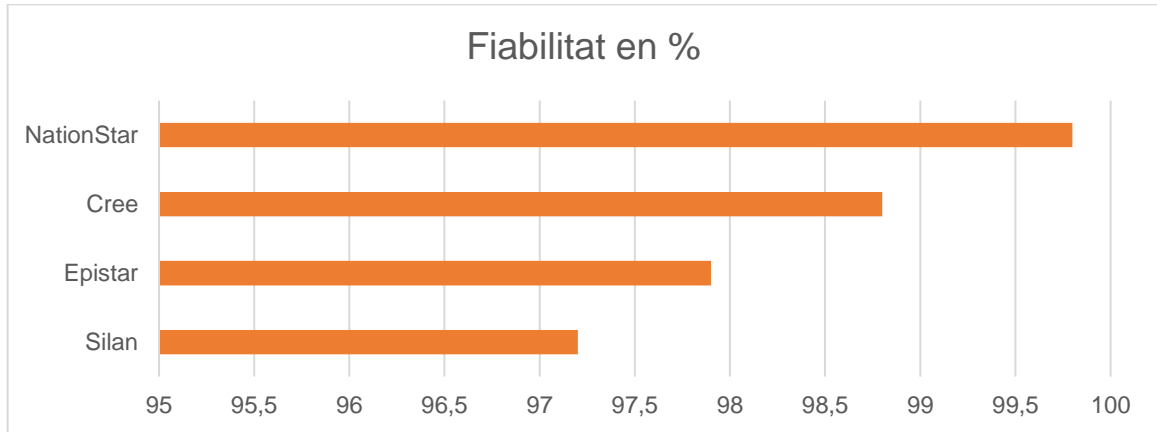


Figura 10. Estudi fiabilitat dels LEDs.

### 3.2 Drivers

Per controlar els LEDs s'ha escollit el MAX7219 de la marca Maxim Integrated. És un controlador multiplexat per pantalles LED de fins a 64 píxels. Admet una tensió d'alimentació de 4V a 5,5V i condueix un corrent de segment de fins a 50mA. El controlador implementa una interfície d'esclaus compatible amb SPI que és controlada des del microcontrolador utilitzant 3 pins de sortida digital.

El SPI és un protocol síncron que treballa en mode full dúplex per rebre i transmetre informació, permetent que dos dispositius puguin comunicar-se entre ells al mateix temps utilitzant canals diferents o línies diferents en el mateix cable. A l'ésser un protocol síncron el sistema compta amb una línia addicional a la de dades que s'encarrega de portar el procés de sincronisme. Tal com es pot veure a la següent figura.

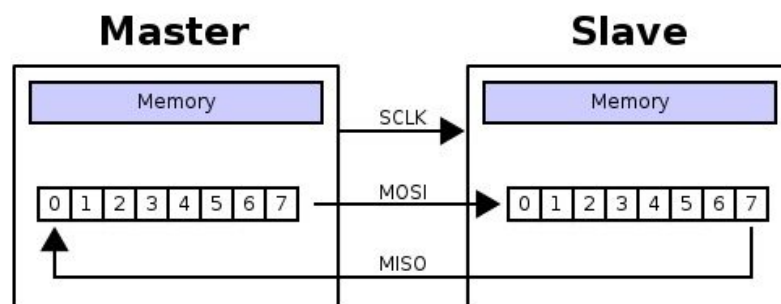


Figura 11. Protocol SPI.

Dins d'aquest protocol el microcontrolador fa de mestre i és l'encarregat de transmetre la informació als diferents MAX7219 que fan d'esclau. Per realitzar aquest procés hi ha dos registres de desplaçament, un pel mestre i un altre per l'esclau. Aquests registres s'encarreguen de guardar els bits per a la transmissió de la informació.

Aquestes quatre línies per dur a terme la comunicació són:

**MOSI (Master Out Slave In):** Línia utilitzada per transportar els bits que provenen del mestre cap a l'esclau.

**MISO (Master In Slave Out):** Línia utilitzada per transportar els bits que provenen de l'esclau cap al mestre.

**CLK (Clock):** Línia que prové del mestre i s'encarrega d'enviar el senyal del rellotge per sincronitzar els dispositius.

**SS (Slave Select):** Línia encarregada de seleccionar i a la vegada, habilitar un esclau.

Per connectar els diferents esclaus hi ha dues maneres, encadenats o en paral·lel. En el mode encadenat l'entrada del MOSI va connectada en el primer esclau i la resta va al MOSI de l'esclau anterior. A més, s'utilitza un únic canal de selecció, provinent del mestre. Per altra banda en el tipus paral·lel utilitza un únic MOSI provinent del mestre que va a tots els esclaus en paral·lel. A més, cada esclau té un canal de selecció individual. A continuació podem veure els dos tipus en la figura.

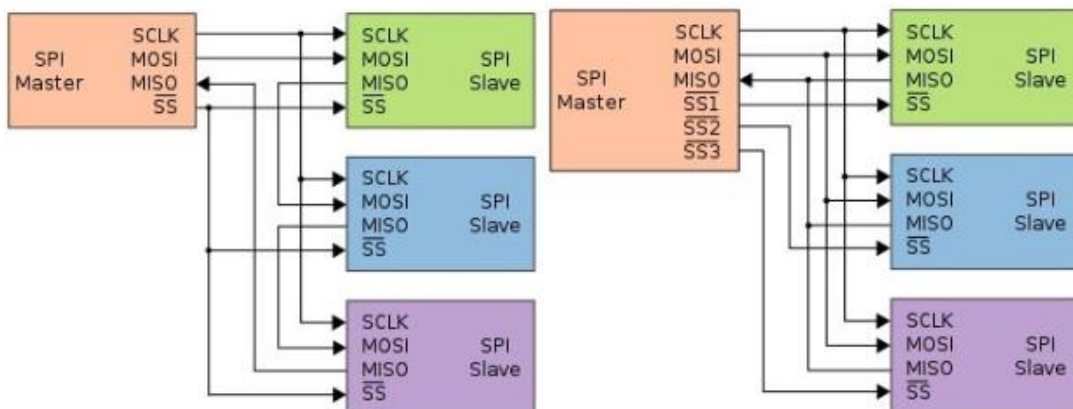


Figura 12. Tipo encadenat i tipo paral·lel.

En el nostre cas, el MAX7219 utilitza el tipus encadenat. Tenim 5 files de LEDs on cada una té 18 MAX7219, enviem la seqüència des del microcontrolador en el primer i aquest es va encadenant en el següent fins ha arribat a l'últim. Una altra particularitat és que el canal MISO que va de l'esclau (MAX7219) al mestre (microcontrolador) no existeix, ja que els drivers només reben informació i no n'envien al mestre. I així amb les 5 files, de tal manera que s'utilitzen 15 cables (3 cables per MAX7219 \* 5 files) per controlar la pantalla. A la següent figura podem veure amb més detall com és amb una fila, ja que totes són iguals.

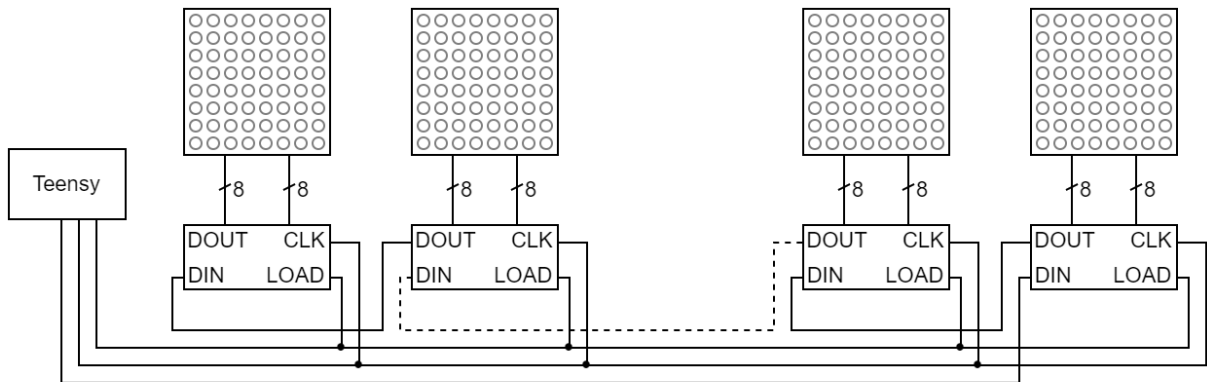


Figura 13. Encadenat dels drivers.

Per cada driver, tal com indica el seu datasheet, també hem afegit dos condensadors entre  $V+$  i GND. Aquests dos tenen un valor de  $0,1\mu\text{F}$  i  $10\mu\text{F}$ , respectivament i serveixen per suprimir/filtrar els senyals de soroll introduïts a través de les línies d'alimentació.

### 3.2.1 Selecció $R_{\text{SET}}$

La resistència  $R_{\text{SET}}$  és l'únic component que limita el corrent per a tots els LEDs individuals. Aquesta resistència no serveix per controlar la brillantor de la pantalla, sinó que la seva funció és protegir el driver MAX7219 i els LEDs de corrents excessius. Per establir la brillantor s'ha de fer, i es fa, a través de software.

Per calcular el valor de la resistència necessitem les següents dades extretes dels datasheets corresponents adjunts al projecte.

Datasheet LED C503B:

DC forward current (corrent màxim a llarg termini): 50mA; Forward voltatge (voltatge del LED): 2,1V

En el datasheet del driver trobem aquesta equació (Eq. 1):

$$\frac{V_{IN}}{R_{SET}} * 100 = I_{LED \text{ forward current}} \quad (\text{Eq. 1})$$

$$\frac{5V}{R_{SET}} * 100 = 50mA \quad (\text{Eq. 2})$$

$$R_{SET} = 10k\Omega \quad (\text{Eq. 3})$$

Segons el datasheet el valor mínim  $R_{SET}$  ha de ser de  $9,53k\Omega$ , en el nostre cas, veiem que el valor calculat és superior, per tant, és un valor admissible.

El corrent màxim que exigirà una sola matriu és 8 vegades el corrent que hem establert amb la resistència  $R_{SET}$  (+10mA per al mateix MAX7219). A l'annex C Càlculs podem veure les equacions aplicades amb més detall així com el consum.

### 3.2.1 Velocitat de refresc

Internament el MAX7219 multiplexa les files de la matriu. Això significa que els controladors canvien entre les files de la matriu molt ràpidament (unes 800 vegades per segon). Això fa la impressió que tots els LEDs estan encesos, mentre que en realitat només parpellegen molt ràpidament. El gran avantatge d'aquest truc és que en cada moment no hi ha més de 8 LEDs (una fila) encesos.

En els rellotges LED antics un dels problemes que presenten és que la velocitat de refresc és molt baixa. La següent figura representa un panell amb una velocitat molt baixa, i tal com es pot veure, en fer la fotografia és impossible saber el temps de cursa.



Figura 14. Rellotge LED antic.

Com hem dit, el nostre driver té una velocitat d'escaneig típica de 800Hz, i cada un d'aquests refresca 8 files de LEDs, per tant, aplicant la següent equació ens surt que cada LED es refresca 100 vegades per segon.

$$\text{Refresc LED} = \frac{\text{Display Scan Rate}}{n^{\circ} \text{ files}} = \frac{800\text{Hz}}{8 \text{ files}} = 100\text{Hz} \quad (\text{Eq. 4})$$

Aquest valor és suficient per veure'ls tots encesos alhora, ja que els nostres ulls són capaços de veure fins a una freqüència de 40Hz. I també queda resolt el problema mencionat anteriorment, ja que per exemple, la càmera d'un mòbil d'última generació com l'iPhone 11 té una freqüència de 60Hz.

### 3.3 Microcontrolador

Per dur a terme el projecte s'han buscat diferents plaques de desenvolupament basades en un microcontrolador. En primer lloc es va pensar en un Arduino, una placa senzilla d'utilitzar i amb moltes funcions que permeten dur a terme diferents tipus de projectes, el problema que presentava aquesta placa de desenvolupament, era que potser no tindria suficients recursos, i a més està enfocada a un ús educatiu.

Buscant alternatives, es va pensar que una Teensy de la marca PJRC podria servir. Aquesta placa, a diferència de l'Arduino, està basada en un processador ARM de 32bits, augmentant significativament la velocitat i memòria a un preu molt similar. Així doncs, es va decidir utilitzar el model Teensy 3.2. A continuació podem veure algunes de les característiques i mides de la placa.

	Teensy 3.2		Teensy 3.2
<b>Processador:</b>	Cortex-M4	<b>Interrupcions:</b>	34 pins
<b>Velocitat:</b>	72 MHz	<b>Entrades analògiques:</b>	21 pins
<b>Overclock:</b>	96 MHz	<b>Resolució:</b>	13 bits
<b>Memòria Flash:</b>	256 kbytes	<b>Sortides analògiques:</b>	1 pin
<b>RAM:</b>	64 kbytes	<b>Resolució:</b>	12 bits
<b>EEPROM:</b>	2048 bytes	<b>Temporitzadors:</b>	12
<b>Digital I/O:</b>	34 pins	<b>Serial:</b>	3
<b>Voltatge sortida:</b>	3,3 V	<b>SPI:</b>	1
<b>Corrent sortida:</b>	10 mA	<b>I2C:</b>	2

Taula 4. Especificacions tècniques de la Teensy 3.2.



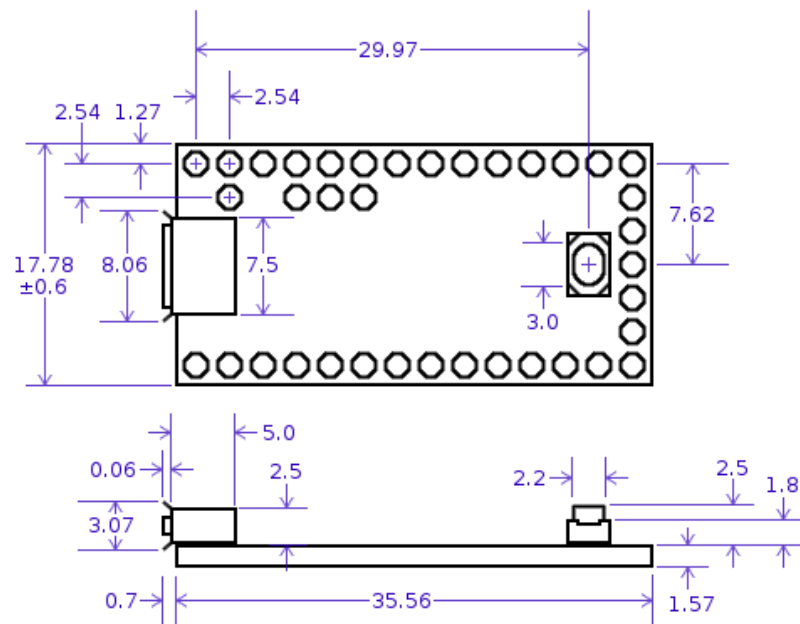


Figura 15. Mides Teensy 3.2 en mm.

Un cop vistes les característiques, la següent figura mostra la placa.



Figura 16. Placa Teensy 3.2.

En l'apartat del driver hem vist que eren necessaris 15 pins digitals per controlar els LEDs, més 3 pel mòdul Wi-Fi i 3 pel mòdul RTC que veurem més endavant a l'apartat d'interrupcions. A continuació podem veure com ha quedat l'assignació de pins.

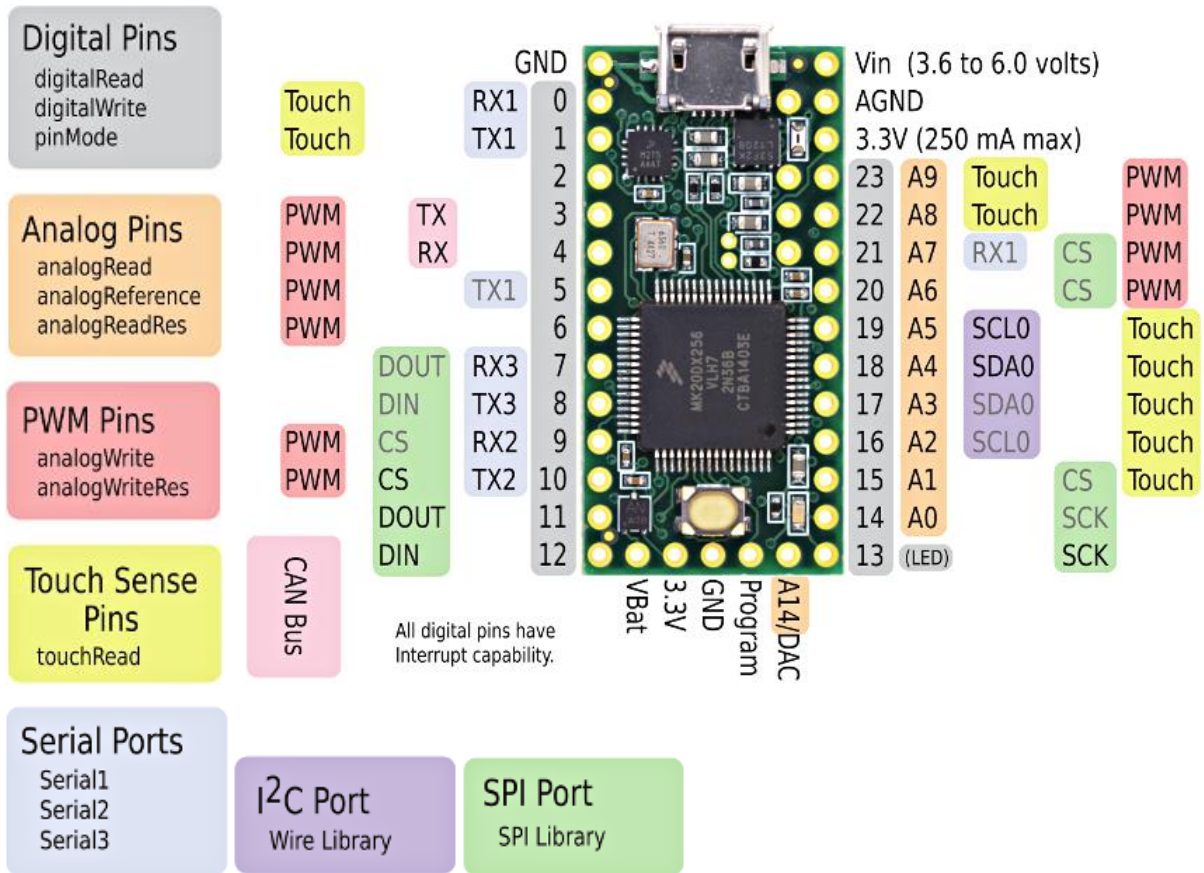


Figura 17. Pins Teensy 3.2.

Connexió	Pin	Pin	Connexió
GND	1	33	Vin (5V)
DIN1 (Fila 1)	2	32	N.C.
CLK1 (Fila 1)	3	31	3,3V (Mòdul RTC)
LOAD1 (Fila 1)	4	30	SQW (Mòdul RTC)
IRQ (Mòdul Wi-Fi)	5	29	LOAD5 (Fila 5)
DIN2 (Fila 2)	6	28	CLK5 (Fila 5)
RST (Mòdul Wi-Fi)	7	27	DIN5 (Fila 5)
CLK2 (Fila 2)	8	26	SCL (Mòdul RTC)
LOAD2 (Fila 2)	9	25	SDA (Mòdul RTC)
DIN3 (Fila 3)	10	24	LOAD4 (Fila 4)
CLK3 (Fila 3)	11	23	CLK4 (Fila 4)
CS (Mòdul Wi-Fi)	12	22	DIN4 (Fila 4)
MOSI (Mòdul Wi-Fi)	13	21	LOAD3 (Fila 3)
MISO (Mòdul Wi-Fi)	14	20	SCK (Mòdul Wi-Fi)

Taula 5. Assignació dels pins.

### 3.4 Comunicacions

Per poder controlar i interactuar amb el panell s'ha fet afegint un mòdul Wi-Fi, aquest permet accedir al programa i modificar les variables del panell. En la web del fabricant del microcontrolador recomanava el mòdul CC3000 d'un altre fabricant, així doncs aquest va ser l'elegit. Un cop ja s'havia començat a treballar amb aquest mòdul vam veure que aquest estava descatalogat pel fabricant i havia de ser substituït pel mòdul ATWINC1500, així doncs, es va fer el canvi i es va utilitzar l'últim mòdul mencionat.

Entre les seves característiques destaquen la comunicació SPI, la banda del Wi-Fi en 802.11bgn i admet el xifratge WEP, WPA i WPA2. També pot treballar des de 3V, ja que porta incorporat un canvi de nivell en els pins d'entrada de tal manera que la lògica positiva queda establerta entre 3V i 5V, en el nostre cas era necessari, recordem que la placa Teensy 3.2 treballa a 3,3V. Finalment, porta l'antena impresa a la PCB. A continuació podem veure'l:

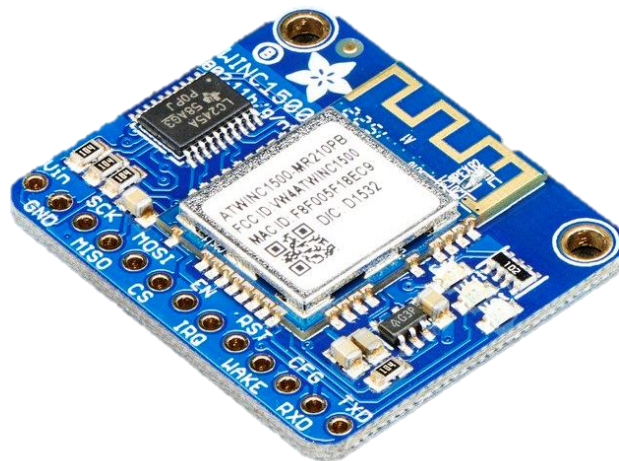


Figura 18. Mòdul Wi-Fi ATWINC1500.

### 3.5 Font d'alimentació

Per alimentar el panell s'ha optat per utilitzar dues fonts, una per cada costat. D'aquesta manera es pot garantir que tots els LEDs d'un mateix costat estan alimentats al mateix voltatge, ja que utilitzen la mateixa font, i per tant, tenen la mateixa lluminositat. Pel que fa a l'electrònica que intervé en el projecte requereix una alimentació contínua de 5V, per tant, s'ha buscat una font d'aquestes característiques amb una potència de 200W. A la següent figura podem veure la potència de cada element i en l'annex C Càlculs ho podem veure al detall.

Element	Nombre de panells/plaques	Potència	Potència total
Panell 24x16	6	12,30W	73,80W
Panell 24x24	6	18,45W	110,70W
<b>Suma potències:</b>			<b>184,50W</b>

Figura 19. Potència a una cara del panell.

Element	Nombre de panells/plaques	Potència	Potència total
Placa de control	1	2,48W	2,48W
<b>Suma potències:</b>			<b>2,48W</b>

Figura 20. Potència placa de control.

Aquests consums s'han calculat tenint en compte que tots els LEDs estan encesos, per tant, el consum real segurament serà bastant menor. Com que el consum de la placa de potència és menyspreable en comparació als panells s'ha decidit alimentar-lo a una de les dues fonts, de tal manera que l'alimentació queda distribuïda de la següent manera:

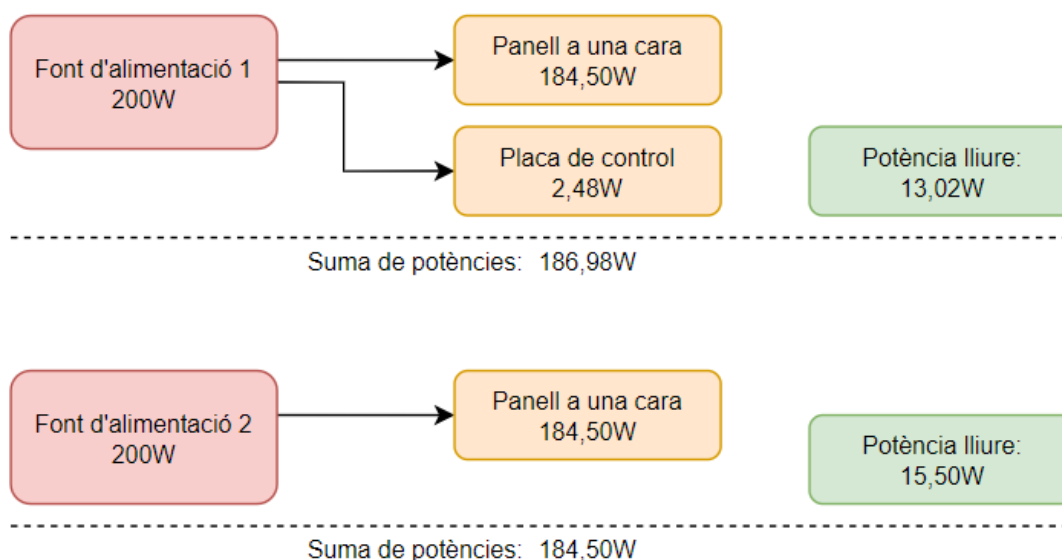


Figura 21. Distribució potència.

Buscant al mercat una font d'aquestes característiques s'ha trobat la font de la marca Mean Well i model UHP-200A-5 d'una potència de 200W, amb un voltatge de sortida de 5V. Entre algunes de les característiques fonamentals per escollir-la ha sigut que està dissenyada per

utilitzar-se en pantalles de senyalització led, com en el nostre projecte. Té un disseny fanless, és a dir, no necessita ventilador. Té protecció contra curtcircuits, sobrecarrega, sobretensió i excés de temperatura. Presenta un format molt reduït i allargat, ideal per determinar el gruix del nostre panell, i a més, té dos terminals block de sortida, que ens permet alimentar el panell i la placa de control sense elements addicionals, en un dels dos casos. En la següent figura podem observar les mides i les entrades/sortides, per veure la fitxa completa del producte hi ha adjunt el seu datasheet.

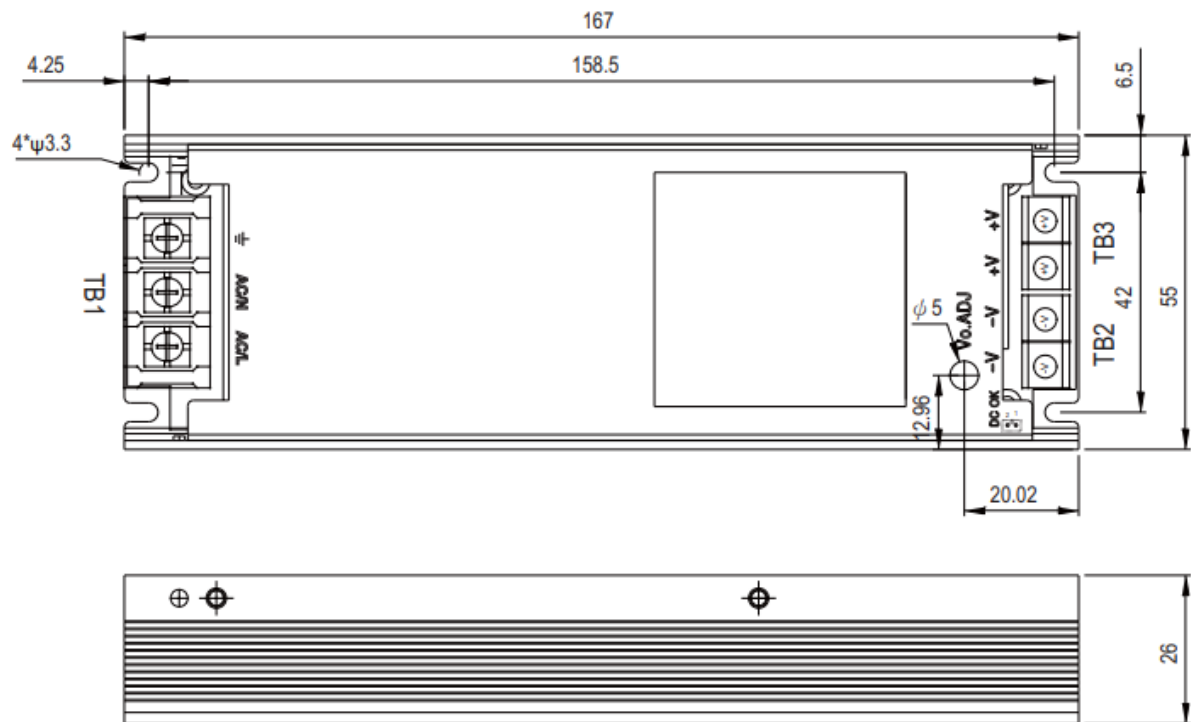


Figura 22. Dimensions font d'alimentació.

### 3.6 Cablejat i connectors

Per escollir el gruix del cablejat i els connectors per les plaques, s'ha tingut en compte el corrent màxim que pot circular pel panell i la caiguda de voltatge admissible de cada component.

Pel que fa als connectors de senyal dels panells, s'ha decidit utilitzar una fila de pins de 6 i 9 contactes segons la mida del panell. Aquests connectors, en concret el model Molex 22-28-4066 i 22-28-4096, van soldats amb orifici passant a la PCB i admeten fins a 4A i 500V. També compleixen amb la norma UL-94 V0 que no era un requisit del projecte, però sí un factor a l'hora d'escollir aquests i no uns altres de preu similar. Aquesta norma s'utilitza per classificar

els materials segons la seva inflamabilitat, i el valor V0 és un dels més alts a l'escala, en altres paraules, significa que en cas d'incendi el foc s'extingeix en menys de 10 segons i sense que gotegi, impedit que pugui malmetre o incendiar altres objectes. Finalment, tenen un pas estàndard de 2,54mm entre pins que representa un avantatge considerable a l'hora de buscar recanvis o adaptadors.

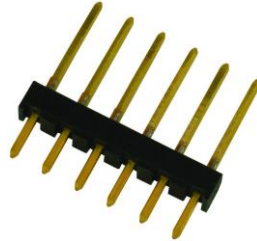


Figura 23. Connector de 6 pins.

Per connectar aquests connectors entre la placa de control i els panells s'ha utilitzat un cable pla IDC de la marca Samtec i model IDSD-0X-S-YY.00, on X indica el nombre de contactes i YY la longitud. S'ha escollit aquest, perquè amb el cable pla ja venen incorporats els connectors de 2,54mm en els dos extrems, sense que sigui necessari modificar-los o adaptar-los al projecte. Admeten fins a 3,4A i 600V, complint sobradament els requisits necessaris. Es recomana utilitzar cola tèrmica entre els pins i el cable, per tal de prevenir possibles desconexions en cas de vibracions o cops.

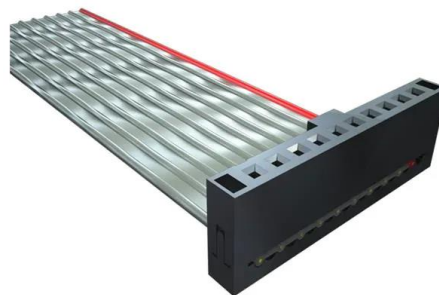


Figura 24. Cable IDC de 9 pins.

Per alimentar els panells i la placa de controls s'ha utilitzat una regleta de terminals PCB amb un pas de 5mm i 2 contactes (positiu i negatiu). Amb un valor nominal de corrent de 13,5A i una tensió de 300V. Si mirem l'annex C càlculs, podem veure que el corrent màxim del panell de 24x16 és de 2,46A, el panell de 24x24 és de 3,69A i la placa de control 0,50A. Partint del valor màxim, 3,69A i el voltatge de treball de 5V veiem que ens encaixen correctament al projecte. A la següent figura podem veure el connector.



Figura 25. Regleta d'alimentació.

Per alimentar les diferents plaques des de la font d'alimentació o el distribuïdor de potència s'ha utilitzat cable d'un gruix de 20AWG. Del paràgraf anterior veiem que la intensitat màxima serà de 3,69A, per tant, en el següent extracte de la taula normalitzada trobem aquest valor de 20AWG.

AWG	Secció en mm <sup>2</sup>	Intensitat
22	0,33	3,00A
20	0,52	5,00A
18	0,82	7,00A

Taula 6. Extracte taula normalitzada mida del cable i guia de classificació.

Per assegurar-nos que ens serveix, també s'ha calculat la pèrdua màxima de voltatge agafant de referència el panell més allunyat. En els plànols podem veure que el més lluny està a 1,1m, per tant, s'ha utilitzat aquest valor en la següent equació:

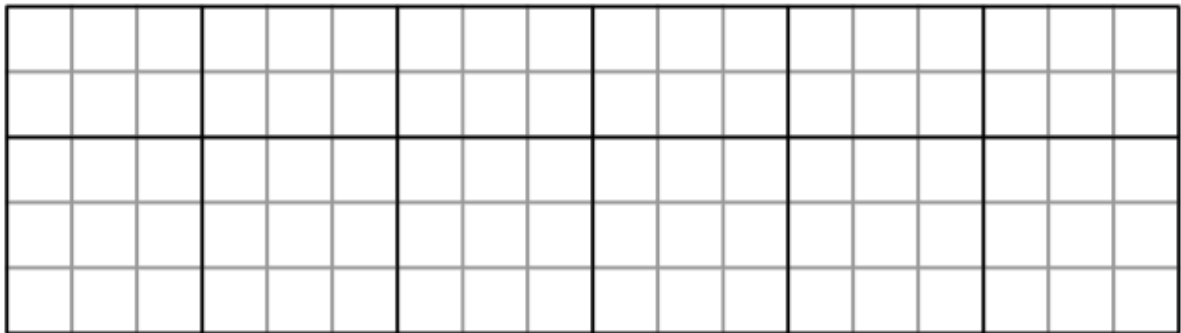
$$AV = \frac{I \cdot \rho \cdot 2l}{A} = \frac{3,69 \cdot 0,0178 \cdot (2 \cdot 1,1)}{0,52} = 0,28V \quad (\text{Eq. 5})$$

On la intensitat (I) és 3,69A, la resistivitat del coure ( $\rho$ ) és 0,0178 $\Omega$ mm<sup>2</sup>/m, la longitud (l) és 1,1m i l'àrea (A) de 0,52mm<sup>2</sup>.

Si restem la pèrdua de voltatge obtinguda, de la tensió d'alimentació veiem que en el pitjor cas tenim un voltatge de 4,72V, suficient per alimentar els drivers que treballen a una tensió d'entre 4V i 5,5V.

#### 4 DESCRIPCIÓ DEL HARDWARE

Com que el panell és de grans dimensions, aquest s'ha dividit en mòduls. D'aquesta manera hem aconseguit reduir els costos sense perdre qualitat. En concret, per cada costat, hi ha 12 mòduls, 6 d'una mida de 24x16 LEDs i 6 més de 24x24 LEDs. Cada un dels mòduls a dins té matrius de 8x8 que és la capacitat màxima de control del driver. La figura que ve a continuació representa les matrius que controla cada driver en color gris i amb negre tenim les diferents mides dels panells.



*Figura 26. Distribució panells.*

Com s'ha dit anteriorment, s'ha decidit utilitzar un pitch de 5mm, això significa que cada LEDs ocupa 5mm més 5mm de separació, per tant, la referència de 24x16 i 24x24 es pot utilitzar tant per definir la mida en LEDs o en centímetres. Si és fa en centímetres cal tenir en compte que ens referim a l'àrea on hi ha els LEDs, ja que, la part de control dels drivers s'ha col·locat a dalt en el cas dels panells de 24x16 i a sota en el cas de 24x24, de tal manera que a la part central només hi ha els LEDs i no hi ha cap element que pugui dificultar la visió.

La part de connexions queda detallada en els plànols, però sí que donarem alguns detalls perquè no hi hagi dubte a l'hora d'entendre'ls. Cada matriu de 8x8 necessita un driver, una resistència limitadora i dos condensadors. Les dades sempre s'envien a la primera matriu de la dreta, i aquesta amb un circuit de la PCB ho envia a la segona cap a l'esquerra, i així successivament fins a la tercera, on cop ha arribat a la tercera s'ha col·locat un connector per poder enviar-ho al següent panell, i així en cascada fins al final. Per tant, nosaltres enviem la informació a les 5 matrius que hi ha a la columna de l'esquerra i aquestes van transmetent la informació a la matriu de la seva esquerra fins al final. A més, cada panell incorpora un terminal bloc per alimentar-lo.



Aquest panell s'ha dissenyat de color negre, d'aquesta manera s'aconsegueix una visió agradable i que els LEDs ressaltin més quan es troben encesos. Per subjectar cada panell a l'estructura s'han utilitzat cargols de color negre per tal de mantenir el color de la PCB. En la part posterior s'ha col·locat tota l'electrònica per qüestions de disseny estètic, ja que així el panell queda més net per si no es vol tapar la part de dalt o de baix segons si és un panell o altre. A la següent figura podem veure un panell de 24x24 d'exemple.

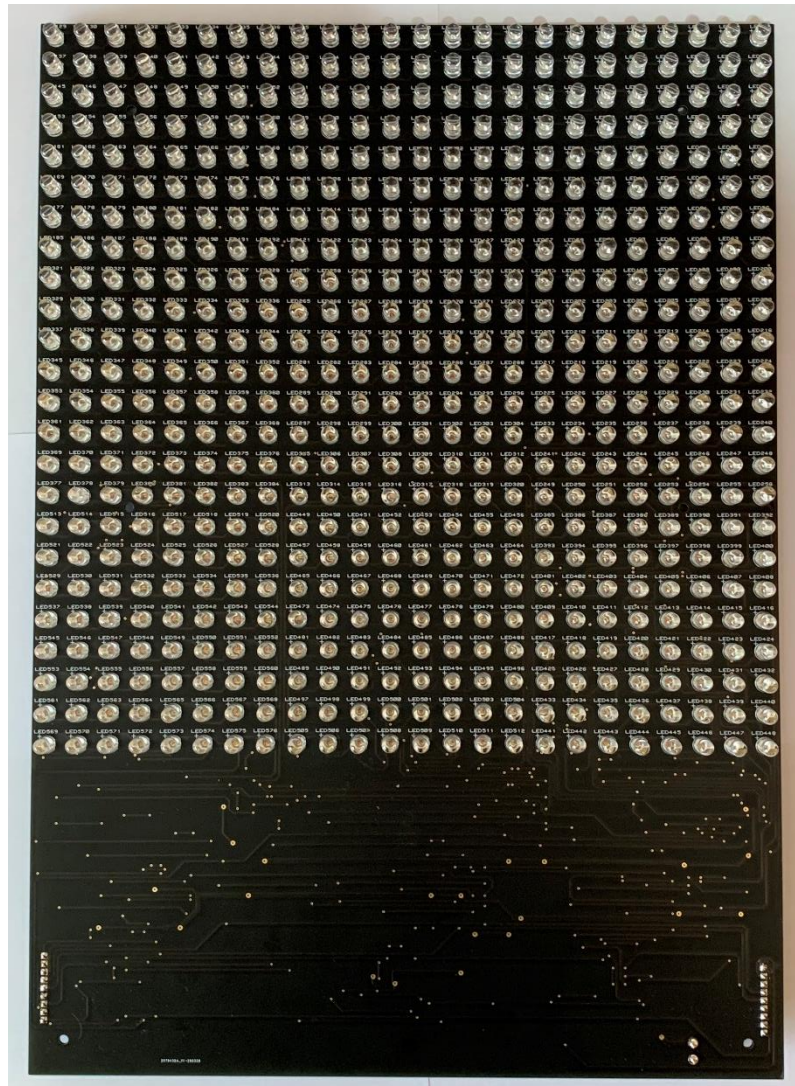


Figura 27. Fotografia panell 24x24.

Un dels problemes que presentaven els drivers amb la Teensy 3.2 era el voltatge de treball, els drivers treballen a un voltatge de 5V mentre que la Teensy 3.2 o fa a un voltatge de 3,3V. Per solucionar-ho es va buscar un transceptor de bus, concretament el model SN74HCT245 de la marca Texas Instruments. Aquest, alimentat a 5V, transforma el voltatge de 3,3V que li arriba del senyal de la Teensy i el passa a 5V perquè el drivers el considerin com a nivell lògic alt. Com que això només és necessari en el primer driver de cada fila, ja que aquest mateix driver passa la informació al següent i treballen al mateix voltatge, es va decidir col·locar els intercanviadors de voltatge a la placa de control. A la següent figura podem veure el funcionament.

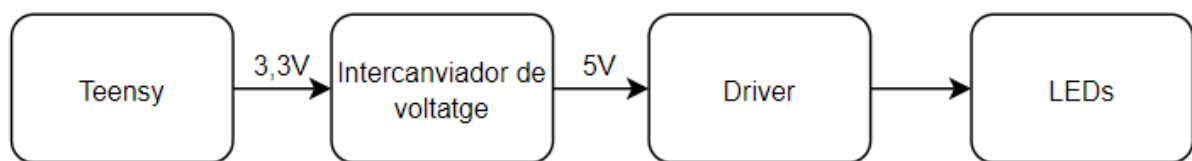


Figura 28. Intercanviador de voltatge.

Pel disseny de la placa de control, s'ha buscat reduir el màxim la mida, però deixant uns marges, per si en algun cas s'hagués de substituir algun component, es pogués dessoldar i soldar fàcilment. Aquestes mides són de 100mm per 60mm. Pel que fa al color de la PCB, com que aquesta placa va a l'interior i no es veu, s'ha decidit fer-la de color verd i així reduir-ne els costos. Com que el panell mostra el mateix a les dues cares, s'ha doblat el connector que hi ha després de l'intercanviador de voltatge que envia el senyal cap als panells.

Finalment, per alimentar totes les plaques, s'ha utilitzat un distribuïdor de potència, aquest ens permet entrar-li el corrent des de la font d'alimentació i la distribueix en 12 canals un per cada panell de 24x16 i 24x24, com que el panell complet és a doble cara, s'utilitzen dos distribuïdors, un per cada font d'alimentació i costat. A continuació podem veure una figura d'aquesta distribució.

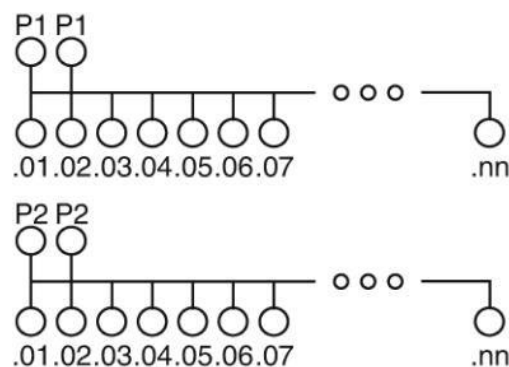


Figura 29. Distribuïdor de potència.

## 5 DESCRIPCIÓ DEL SOFTWARE

Aquest panell s'ha pensat per mostrar sempre un rellotge en la part superior i al centre, aquest rellotge pot ser ascendent o descendent, per si es vol mostrar el temps de curs, hora del dia o el temps que falta per començar la cursa. Per posar en marxa aquest rellotge tenim un botó d'inici que veurem més endavant i per ajustar-lo compte amb un offset que pot sumar o restar hores, minuts i segons.

En la part inferior tenim 3 línies de text, que ve poden mostrar les dades que agafem del software de cronometratge com, el dorsal, el nom i el temps. O també es poden personalitzar amb el text que desitgem. Si seleccionem el text personalitzat, l'usuari pot decidir que escriu a cada línia, en canvi, si se selecciona el mode amb el software de cronometratge, aquest va mostrant la informació a mesura que van entrant els atletes a l'arribada. De les tres línies disponibles, col·loca les dades en la primera línia, si entra un altre atleta, baixa les dades de l'atleta anterior a la línia 2 i a la línia 1 mostra les dades del més recent. Seguint aquest criteri l'atleta més recent sempre queda a dalt i els altres es van baixant fins a desaparèixer.

A més a més, a la part de selecció del rellotge ascendent o descendent també podem seleccionar la brillantor de la pantalla que va de 0 a 15, sent 0 el mínim i 15 al màxim.

Aquest panell s'ha dividit en cinc files, perquè cada driver només pot controlar una fila de fins a 8 LEDs d'altura. Les tres de baix corresponent a tres línies independents i les dues de dalt formen una sola línia amb caràcters més grans. En la propera figura podem veure aquesta divisió dels caràcters, la línia negra representa la separació entre la fila 1 i 2.

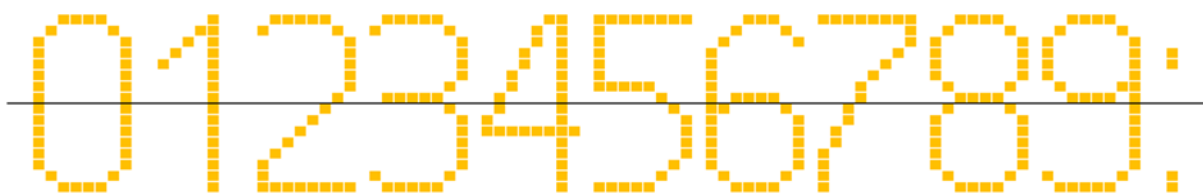


Figura 30. Divisió dels caràcters.

Per explicar el funcionament dels drivers ens centrarem a explicar el d'una fila, ja que la resta funciona igual. El MAX7219 disposa d'un registre no operatiu que s'utilitza quan es connecten varis MAX7219 en cascada. Tots els drivers han d'estar connectats a les línies LOAS/CS però les dades DIN només s'envien al primer, i aquest mitjançant DOUT o envia al DIN del següent driver. DOUT és una sortida de nivell lògic CMOS que maneja fàcilment els dispositius

connectats. Per exemple, si a una línia nostra de 18 drivers, volem escriure a l'últim driver, s'envia la paraula de 16 bits desitjada seguida de 17 codis no operatius 0xXX0X, segons el datasheet. Quan LOAD/CS puja, les dades es guarden en tots els dispositius. Els primers 17 xips reben instruccions sense operació, i l'últim rep les dades.

## 5.1 Llibreries

Per a fer el control del programa s'han utilitzat vaires llibreries ja definides i d'accés públic. Per guardar els caràcters, s'utilitza la llibreria Sprite, aquesta en permet guardar totes les lletres per després treballar amb elles més còmodament. A continuació tenim una figura d'exemple de declaració.

```
Sprite lletra_A = Sprite(5, 8,  
    B00000,  
    B01110,  
    B10001,  
    B10001,  
    B10001,  
    B11111,  
    B10001,  
    B10001  
);
```

Figura 31. Exemple llibreria sprite.

Pel control dels drivers s'ha utilitzat la llibreria Matrix, aquesta ens permet declarar els pins on es troben connectats els primers drivers de cada fila amb la funció: Matrix linia1 = Matrix(DIN, CLK, LOAD, numChips);. I a la vegada escriure en els drivers amb la funció: linia1.write(p, 0, lletra\_A);. On linia1 és la fila, p la posició en l'eix X, 0 la posició en l'eix Y i lletra\_A és l'sprite declarat anteriorment.

Per comunicar-se amb el mòdul RTC s'utilitza la llibreria Wire que ens permet una comunicació I2C. I finalment, pel mòdul Wi-Fi s'han utilitzat les llibreries SPI per les comunicacions i la WiFi101 per utilitzar funcions ja definides.

## 5.2 Estructura del programa

Inicialment el programa intenta connectar-se a una xarxa Wi-Fi coneguda, si no en troba cap, aquesta crea un Acces Point sense contrasenya. Perquè l'usuari si pugui connectar i introduir les dades de la xarxa Wi-Fi que es vol connectar. Un cop la Teensy rep aquestes dades,

intenta connectar-se a la xarxa Wi-Fi, si es connecta vol dir que les dades són correctes, i per tant, les guarda per a futures ocasions. En el cas que no s'hagi pogut connectar, tornaria a crear l'Acces Point. A continuació podem veure una figura on queda detallat.

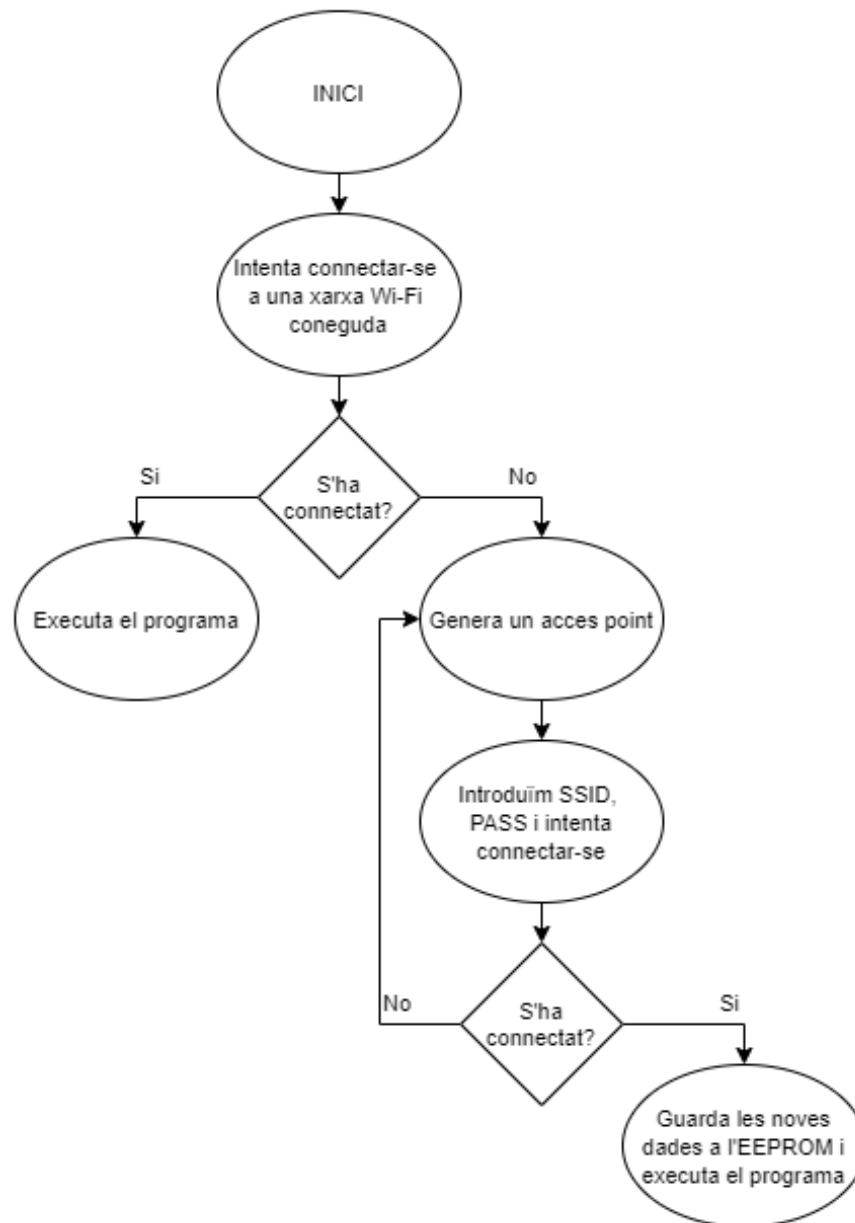


Figura 32. Diagrama de flux inicial.

Un cop ja tenim la connexió a internet, el programa executa el programa principal, on mostra el temps actualitzat, atletes, etc. I crea el web server per interactuar, el qual està descrit en els apartats d'interfície d'usuari i a l'annex B Manual.

### 5.2.1 Interrupcions

La funció principal del panell és mostrar el temps, per això s'ha prioritzat. Per fer-ho s'utilitza la funció d'interrupcions del microcontrolador. D'aquesta manera en produir-se un canvi en el temps, aquest s'actualitzarà i després seguirà amb la tasca que estigués fent.

El rellotge mostra hores, minuts i segon, per tant, les interrupcions han de ser cada 1 segon. Per fer-ho s'ha buscat hardware independent de la Teensy, és a dir, sense utilitzar les interrupcions internes. D'aquesta manera, el temps emprat dins la interrupció no influirà en la precisió del rellotge fent que es desviï.

Una de les opcions va ser utilitzar un cristall de quars, però es va descartar per la complexitat en dividir la freqüència exacta. Una altra opció va ser utilitzar un circuit integrat 555, però no té gaire precisió i també depèn de components externs.

Finalment, es va optar per utilitzar un mòdul DS1307 o DS3231, la diferència entre un i l'altre és que el DS1307 l'afecten les altes temperatures, provocant un desfasament de fins a 5 minuts al mes. Per altra banda, el DS3231 té un oscil·lador intern al qual no l'afecten tant els canvis de temperatura, pot tenir un desfasament d'uns minuts a l'any, fent-lo més precís. Per tant, s'ha utilitzat el DS3231 que podem veure a la següent figura.

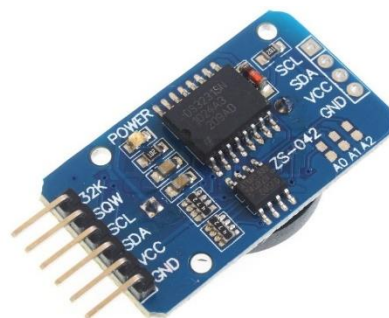


Figura 33. Mòdul RTC DS3231.

Tot i que aquest mòdul permet llegir i escriure l'hora actual, s'ha decidit no utilitzar aquestes funcions, i utilitzar el pin SQW configurat a 1Hz per generar pics d'interrupció i utilitzar-los per actualitzar el temps.

Per comprovar la fiabilitat d'aquest mòdul, s'ha fet un estudi experimental, aquesta consisteix a deixar el mòdul fent interrupcions de 1Hz durant 4 hores. El temps que sol durar una cursa normal. En el transcurs d'aquestes 4 hores es van registrar tots els valors del següent gràfic. En l'eix Y veiem el nombre de lectures i en l'eix X els mil·lisegons. Podem veure que la gran part de lectures estan al centre, i entre el valor més baix i el més alt hi ha 89 mil·lisegons (528-439). És a dir, que en els pitjors dels casos, entra una interrupció i l'altra tindriem un error de 0,089 segons.

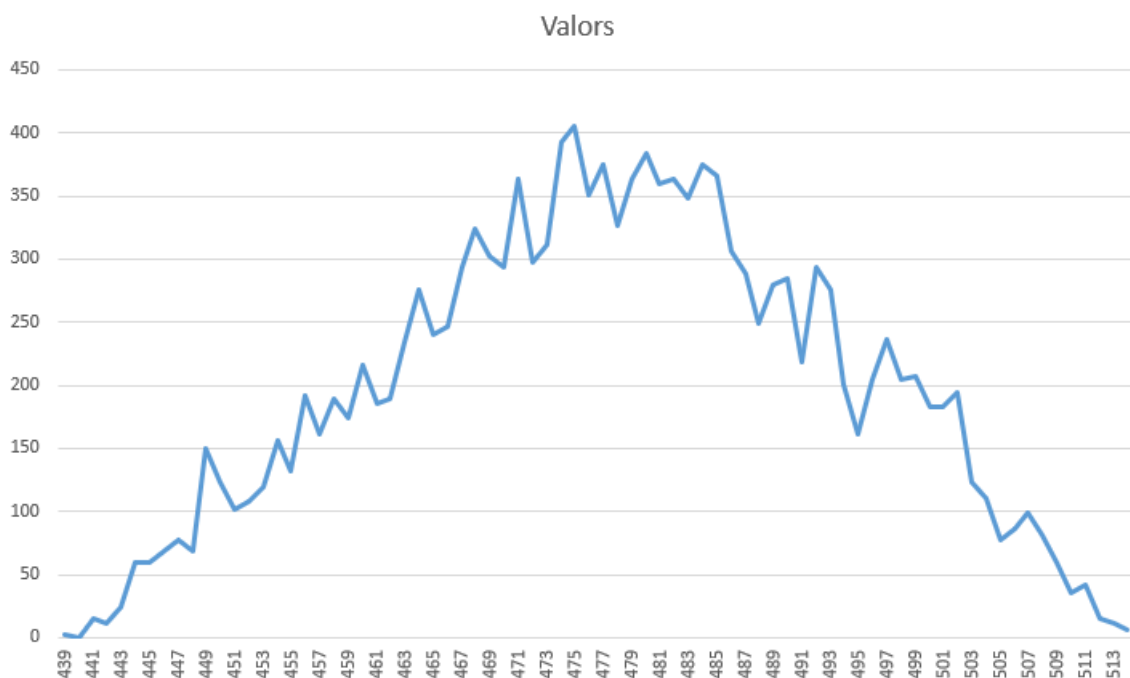


Figura 34. Precisió experimental DS3231.

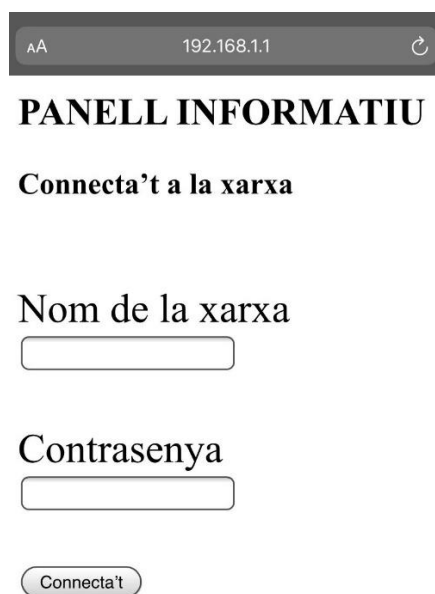
### 5.3 COMUNICACIONS

Per comunicar-se amb el panell es dona per suposat que a l'arribada hi haurà un router. El motiu és que la màquina de cronometratge necessita internet per enviar la lectura dels xips, i el cronometrador per executar el software de cronometratge. Per això el panell pretén aprofitar aquesta comunicació present en totes les curses, i de fàcil accés a través de qualsevol dispositiu.

Un cop tenim el panell connectat a la xarxa Wi-Fi, podem trobar els passos a l'annex B Manual d'usuari, el panell en mostra una IP en pantalla, en accedir al panell trobem el menú. Un cop l'hem configurat el microcontrolador rep les dades mitjançant el mètode GET. És a dir, en enviar les dades es genera un URL que aquesta és analitzada i s'extreuen les dades.

## 6 INTERFÍCIE D'USUARI

En panell consta de dues interfícies d'usuari. Una primera per registrar la connexió Wi-Fi i una segona pel control. En primer lloc, el panell crea una xarxa Wi-Fi, un cop connectats a ella, introduïm la IP 192.168.1.1 des de qualsevol navegador. I tindrem el panell de la següent figura.



AA 192.168.1.1 ↻

### PANELL INFORMATIU

#### Connecta't a la xarxa

Nom de la xarxa

Contrasenya

Connecta't

*Figura 35. Interfície per registrar el Wi-Fi.*

Com podem veure és una interfície simple i agradable per ser utilitzada per a qualsevol usuari. Un cop s'ha pogut connectar al router, el panell ens mostra la IP que li han assignat perquè ens hi puguem connectar.

Una vegada entrem en la IP, en trobarem dos blocs. El primer bloc està dividit en dues parts, una per seleccionar el mode de funcionament del rellotge en ascendent o descendent i també la brillantor, que si no en canvia de valor, per defecte està en ascendent i brillantor al màxim. En la segona part, hi ha un botó d'inici, aquest serveix per iniciar el rellotge i a sota ens indica si el rellotge està aturat o en marxa. Més avall es pot configurar un offset per acabar d'ajustar el temps de cursa o qualsevol que es vulgui mostrar. Aquest offset pot ser positiu o negatiu i es pot ajustar en hores, minuts i segons. Per defecte està l'offset positiu i amb un valor de 0 en tots els caps, és a dir, desactivat.

El segon bloc permet configurar les línies de text, podem seleccionar entre mostrar el temps dels atletes o bé text personalitzat. Si se selecciona el temps dels atletes, cal introduir l'URL



que genera el programa de cronometratge i també seleccionar si es vol o no, mostrar sempre el primer classificat. Si pel contrari se selecciona el text, cal introduir el text en les 3 línies per separat. En la següent figura podem veure aquesta interfície i en l'annex B manual les instruccions pas a pas perquè qualsevol usuari sense coneixements previs el pugui fer servir.

AA 192.168.0.105 ↻

### PANEL·L INFORMATIU

#### RELLOTGE

Mode funcionament

Ascendent:

Descendent:

Brillantor (0-15):

Offset: +  -

Hores (0-23):

Minuts (0-59):

Segons (0-59):

Actualitza

INICI

El rellotge està aturat.

#### TEXT A MOSTRAR

Primer sempre es mostra

URL:

Temps dels atletes:

Text:

Línia 1:

Línia 2:

Línia 3:

Envia

Figura 36. Interfície principal.

## **7 RESUM PRESSUPOST**

El pressupost per a la realització del panell informatiu per esdeveniments esportius format per diferents mòduls de perifèrics, així com la realització del circuit imprès, del muntatge de la placa, de la mecanització, de la verificació pel seu correcte funcionament i amb les hores de disseny té un import de: catorze mil quatre-cents vint-i-sis amb dinou cèntims, sense IVA.

## 8 CONCLUSIONS

Aquest projecte s'ha realitzat amb la finalitat de trobar una solució òptima a la necessitat de mostrar informació en temps real als atletes, que fins al dia d'avui només es mostrava el temps de cursa. I en algunes ocasions, sobretot en les curses grans, també es posava a disposició una app de consulta, però cal tenir en compte que no tots els atletes porten el mòbil a la cursa.

Gràcies a aquest panell s'ofereix la possibilitat de mostrar el temps de cursa, el temps de cada atleta, patrocinadors o inclús els premis, fent que la cursa sigui més interactiva, agradable i transparent de cara als participants, ja que ara tenen la informació a l'instant tan bon punt arriben. Com a requisit extra, s'ha afegit un control de la lluminositat, per controlar l'enlluernament i afavorir l'ús del panell en curses nocturnes.

A més a més, s'ha dissenyat la interfície d'usuari de manera clara i entenedora, per agilitzar i facilitar la seva utilització entre les persones a les quals va adreçades aquesta aplicació.

Les simulacions de la placa de control s'han efectuat en condicions favorables i sense tenir en compte tots els factors que poden afectar el dispositiu en l'aplicació real, ja que no s'ha pogut construir i veure el funcionament complet en elements reals. Però si s'ha construït un panell per a fer les proves i veure la seva lluminositat amb l'efecte de la llum solar.

Per tant, finalment es pot afirmar que s'han assolit els objectius inicials correctament i s'ha aconseguit realitzar el panell informatiu per esdeveniments esportius amb la implementació d'un sistema de control basant-se en una aplicació web.

Roger Gasull Vilagran

Graduat en Enginyeria Electrònica Industrial i Automàtica

Viladamat, 5 de juny de 2020

## **9 RELACIÓ DE DOCUMENTS**

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

## 10 BIBLIOGRAFIA

Adafruit ATWINC1500 WiFi Breakout (<https://learn.adafruit.com/adafruit-atwinc1500-wifi-module-breakout/overview>, 18 de març de 2020).

endsWith() (<https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/endswith/>, 26 de març de 2020).

HTML – Elementos (<https://tutorialehtml.com/es/tutoriales-html-elementos>, 29 de març de 2020).

Libraries ([https://www.pjrc.com/teensy/td\\_libs.html](https://www.pjrc.com/teensy/td_libs.html), 13 de març de 2020).

Matrix (<http://wiring.org.co/reference/libraries/Matrix/index.html>, 18 d'abril de 2020).

String replace Function (<https://www.arduino.cc/en/Tutorial/StringReplace>, 26 de març de 2020).

Teensy Technical Specifications (<https://www.pjrc.com/teensy/techspecs.html>, 19 de març de 2020).

WiFi101 library (<https://www.arduino.cc/en/Reference/WiFi101>, 21 de març de 2020).

Wire Size & Current Rating Guide ([https://www.jst.fr/doc/jst/pdf/current\\_rating.pdf](https://www.jst.fr/doc/jst/pdf/current_rating.pdf), 13 d'abril de 2020).

## **11 GLOSSARI**

AP: Access Point

AWG: American Wire Gauge

DIP: Dual In-line Package

IDC: Insulation-Displacement Technology/Termination

LED: Light Emitting Diode

PCB: Printed Circuit Board

SCK: Serial Clock

SMD: Surface-Mounted Device

SPI: Serial Peripheral Interface

USB: Universal Serial Bus

## A CODI PROGRAMA

```
#include <Sprite.h> // Sprite before Matrix
#include <Matrix.h>
#include <Wire.h>
#define RTCAddress 0x68 //0x68
#include <SPI.h> // Comunicació amb el modul WiFi
#include <WiFi101.h> // Llibreria interectuar amb el modul wifi
char APssid[] = "Panell_informatiu"; // Nom de la xarxa (SSID)

bool tenim_ssid = false;
String ssid, pass;

bool rellotge_iniciat = false;
int interrupcions = 0;
int hores = 0;
int minuts = 0;
int segons = 0;

int status = WL_IDLE_STATUS;
WiFiServer server(80);

//const int lletres_lliures = 14; //Lletres que caben entre el punt i el temps, ex:
R._ _ lletres _ _ 88:88:88
const int leds_lletra = 5; //Leds utilitzats per cada lletra + 1 espai
const int numChips = 18; //Numero de xips per fila

//          DIN, CLK, LOAD, #chips
Matrix linia1 = Matrix(0, 1, 2, numChips); //Linia 1 (primera de dalt)
Matrix linia2 = Matrix(4, 6, 7, numChips); //Linia 2
Matrix linia3 = Matrix(8, 9, 14, numChips); //Linia 3
Matrix linia4 = Matrix(15, 16, 17, numChips); //Linia 4
Matrix linia5 = Matrix(20, 21, 22, numChips); //Linia 5

int p = 0; //Posicio de cada lletra segons els leds

Sprite lletra_A = Sprite(5, 8,
```

```
B00000,  
B01110,  
B10001,  
B10001,  
B10001,  
B11111,  
B10001,  
B10001  
);
```

```
Sprite lletra_B = Sprite(5, 8,  
B00000,  
B11110,  
B10001,  
B10001,  
B11110,  
B10001,  
B10001,  
B11110  
);
```

```
Sprite lletra_C = Sprite(5, 8,  
B00000,  
B01111,  
B10001,  
B10000,  
B10000,  
B10000,  
B10000,  
B10000,  
B01111  
);
```

```
Sprite lletra_D = Sprite(5, 8,  
B00000,  
B11100,  
B10010,  
B10001,
```



```
B10001,  
B10001,  
B10010,  
B11100  
);
```

```
Sprite lletra_E = Sprite(5, 8,  
B00000,  
B11111,  
B10000,  
B10000,  
B11110,  
B10000,  
B10000,  
B11111  
);
```

```
Sprite lletra_F = Sprite(5, 8,  
B00000,  
B11111,  
B10000,  
B10000,  
B11110,  
B10000,  
B10000,  
B10000  
);
```

```
Sprite lletra_G = Sprite(5, 8,  
B00000,  
B01111,  
B10000,  
B10000,  
B10000,  
B10011,  
B10001,  
B01110
```

```
);
```

```
Sprite lletra_H = Sprite(5, 8,  
                        B00000,  
                        B10001,  
                        B10001,  
                        B10001,  
                        B11111,  
                        B10001,  
                        B10001,  
                        B10001  
                        );
```

```
Sprite lletra_I = Sprite(5, 8,  
                        B00000,  
                        B01110,  
                        B00100,  
                        B00100,  
                        B00100,  
                        B00100,  
                        B00100,  
                        B01110  
                        );
```

```
Sprite lletra_J = Sprite(5, 8,  
                        B00000,  
                        B01111,  
                        B00001,  
                        B00001,  
                        B00001,  
                        B10001,  
                        B10001,  
                        B01110  
                        );
```

```
Sprite lletra_K = Sprite(5, 8,  
                        B00000,
```

```
B10001,  
B10001,  
B10010,  
B11100,  
B10010,  
B10001,  
B10001  
);
```

```
Sprite lletra_L = Sprite(5, 8,  
B00000,  
B10000,  
B10000,  
B10000,  
B10000,  
B10000,  
B10000,  
B10000,  
B11111  
);
```

```
Sprite lletra_M = Sprite(5, 8,  
B00000,  
B10001,  
B11011,  
B10101,  
B10001,  
B10001,  
B10001,  
B10001  
);
```

```
Sprite lletra_N = Sprite(5, 8,  
B00000,  
B10001,  
B10001,  
B11001,  
B10101,
```

```
        B10011,  
        B10001,  
        B10001  
    );
```

```
Sprite lletra_O = Sprite(5, 8,  
        B00000,  
        B01110,  
        B10001,  
        B10001,  
        B10001,  
        B10001,  
        B10001,  
        B10001,  
        B01110  
    );
```

```
Sprite lletra_P = Sprite(5, 8,  
        B00000,  
        B11110,  
        B10001,  
        B10001,  
        B11110,  
        B10000,  
        B10000,  
        B10000  
    );
```

```
Sprite lletra_Q = Sprite(5, 8,  
        B00000,  
        B01110,  
        B10001,  
        B10001,  
        B10001,  
        B10001,  
        B10010,  
        B01101  
    );
```

```
Sprite lletra_R = Sprite(5, 8,  
                        B00000,  
                        B11110,  
                        B10001,  
                        B10001,  
                        B11110,  
                        B10100,  
                        B10010,  
                        B10001  
                        );
```

```
Sprite lletra_S = Sprite(5, 8,  
                        B00000,  
                        B01111,  
                        B10000,  
                        B10000,  
                        B01110,  
                        B00001,  
                        B00001,  
                        B11110  
                        );
```

```
Sprite lletra_T = Sprite(5, 8,  
                        B00000,  
                        B11111,  
                        B00100,  
                        B00100,  
                        B00100,  
                        B00100,  
                        B00100,  
                        B00100,  
                        B00100  
                        );
```

```
Sprite lletra_U = Sprite(5, 8,  
                        B00000,  
                        B10001,
```

```
B10001,  
B10001,  
B10001,  
B10001,  
B10001,  
B01110  
);
```

```
Sprite lletra_V = Sprite(5, 8,  
B00000,  
B10001,  
B10001,  
B10001,  
B10001,  
B10001,  
B10001,  
B01010,  
B00100  
);
```

```
Sprite lletra_W = Sprite(5, 8,  
B00000,  
B10001,  
B10001,  
B10001,  
B10001,  
B10101,  
B10101,  
B01010  
);
```

```
Sprite lletra_X = Sprite(5, 8,  
B00000,  
B10001,  
B10001,  
B01010,  
B00100,  
B01010,
```

```
B10001,  
B10001  
);
```

```
Sprite lletra_Y = Sprite(5, 8,  
B00000,  
B10001,  
B10001,  
B01010,  
B00100,  
B00100,  
B00100,  
B00100,  
B00100  
);
```

```
Sprite lletra_Z = Sprite(5, 8,  
B00000,  
B11111,  
B00001,  
B00010,  
B00100,  
B01000,  
B10000,  
B11111  
);
```

```
Sprite numero_zero = Sprite(5, 8,  
B00000,  
B01110,  
B10001,  
B10001,  
B10001,  
B10001,  
B10001,  
B01110  
);
```

```
Sprite numero_un = Sprite(5, 8,  
    B00000,  
    B00100,  
    B01100,  
    B00100,  
    B00100,  
    B00100,  
    B00100,  
    B00100,  
    B00100  
);
```

```
Sprite numero_dos = Sprite(5, 8,  
    B00000,  
    B11110,  
    B00001,  
    B00001,  
    B00110,  
    B01000,  
    B10000,  
    B11111  
);
```

```
Sprite numero_tres = Sprite(5, 8,  
    B00000,  
    B11111,  
    B00010,  
    B00100,  
    B01110,  
    B00001,  
    B00001,  
    B11110  
);
```

```
Sprite numero_quatre = Sprite(5, 8,  
    B00000,  
    B00010,  
    B00110,
```



```
        B01010,  
        B10010,  
        B11111,  
        B00010,  
        B00010  
    );
```

```
Sprite numero_cinc = Sprite(5, 8,  
        B00000,  
        B11111,  
        B10000,  
        B10000,  
        B11110,  
        B00001,  
        B00001,  
        B11110  
    );
```

```
Sprite numero_sis = Sprite(5, 8,  
        B00000,  
        B00111,  
        B01000,  
        B10000,  
        B11110,  
        B10001,  
        B10001,  
        B01110  
    );
```

```
Sprite numero_set = Sprite(5, 8,  
        B00000,  
        B11111,  
        B00001,  
        B00010,  
        B00100,  
        B01000,  
        B10000,
```

```
        B10000
    );

Sprite numero_vuit = Sprite(5, 8,
        B00000,
        B01110,
        B10001,
        B10001,
        B01110,
        B10001,
        B10001,
        B01110
    );

Sprite numero_nou = Sprite(5, 8,
        B00000,
        B01110,
        B10001,
        B10001,
        B01111,
        B00001,
        B00001,
        B01110
    );

Sprite numero_zero_gran_dalt = Sprite(8, 8,
        B00111100,
        B01000010,
        B10000001,
        B10000001,
        B10000001,
        B10000001,
        B10000001,
        B10000001,
        B10000001
    );

Sprite numero_zero_gran_baix = Sprite(8, 8,
```

```
B10000001,  
B10000001,  
B10000001,  
B10000001,  
B10000001,  
B10000001,  
B01000010,  
B00111100  
);
```

```
Sprite numero_un_gran_dalt = Sprite(8, 8,  
B00000100,  
B00001100,  
B00010100,  
B00100100,  
B01000100,  
B00000100,  
B00000100,  
B00000100  
);
```

```
Sprite numero_un_gran_baix = Sprite(8, 8,  
B00000100,  
B00000100,  
B00000100,  
B00000100,  
B00000100,  
B00000100,  
B00000100,  
B00000100  
);
```

```
Sprite numero_dos_gran_dalt = Sprite(8, 8,  
B01111100,  
B10000010,  
B00000001,  
B00000001,
```

```
B00000001,  
B00000001,  
B00000001,  
B00000010  
);
```

```
Sprite numero_dos_gran_baix = Sprite(8, 8,  
B00000100,  
B00001000,  
B00010000,  
B00100000,  
B01000000,  
B10000000,  
B10000000,  
B11111111  
);
```

```
Sprite numero_tres_gran_dalt = Sprite(8, 8,  
B01111100,  
B10000010,  
B00000001,  
B00000001,  
B00000001,  
B00000001,  
B00000010,  
B01111100  
);
```

```
Sprite numero_tres_gran_baix = Sprite(8, 8,  
B00000010,  
B00000001,  
B00000001,  
B00000001,  
B00000001,  
B00000001,  
B10000010,  
B01111100
```

```
);
```

```
Sprite numero_quatre_gran_dalt = Sprite(8, 8,  
    B00000110,  
    B00001010,  
    B00001010,  
    B00010010,  
    B00010010,  
    B00100010,  
    B00100010,  
    B01000010  
);
```

```
Sprite numero_quatre_gran_baix = Sprite(8, 8,  
    B01000010,  
    B10000010,  
    B11111111,  
    B00000010,  
    B00000010,  
    B00000010,  
    B00000010,  
    B00000010  
);
```

```
Sprite numero_cinc_gran_dalt = Sprite(8, 8,  
    B11111111,  
    B10000000,  
    B10000000,  
    B10000000,  
    B10000000,  
    B10000000,  
    B11111000,  
    B00000100  
);
```

```
Sprite numero_cinc_gran_baix = Sprite(8, 8,  
    B00000010,
```

```
B00000001,  
B00000001,  
B00000001,  
B00000001,  
B00000010,  
B00000100,  
B11111000  
);
```

```
Sprite numero_sis_gran_dalt = Sprite(8, 8,  
B00011100,  
B00100010,  
B01000001,  
B10000000,  
B10000000,  
B10000000,  
B10000000,  
B10111100  
);
```

```
Sprite numero_sis_gran_baix = Sprite(8, 8,  
B11000010,  
B10000001,  
B10000001,  
B10000001,  
B10000001,  
B10000001,  
B01000010,  
B00111100  
);
```

```
Sprite numero_set_gran_dalt = Sprite(8, 8,  
B11111111,  
B00000001,  
B00000001,  
B00000010,  
B00000100,
```

```
B00001000,  
B00010000,  
B00010000  
);
```

```
Sprite numero_set_gran_baix = Sprite(8, 8,  
B00100000,  
B00100000,  
B01000000,  
B01000000,  
B10000000,  
B10000000,  
B10000000,  
B10000000  
);
```

```
Sprite numero_vuit_gran_dalt = Sprite(8, 8,  
B00111100,  
B01000010,  
B10000001,  
B10000001,  
B10000001,  
B10000001,  
B01000010,  
B00111100  
);
```

```
Sprite numero_vuit_gran_baix = Sprite(8, 8,  
B01000010,  
B10000001,  
B10000001,  
B10000001,  
B10000001,  
B10000001,  
B01000010,  
B00111100  
);
```

```
Sprite numero_nou_gran_dalt = Sprite(8, 8,  
                                     B00111100,  
                                     B01000010,  
                                     B10000001,  
                                     B10000001,  
                                     B10000001,  
                                     B10000001,  
                                     B01000010,  
                                     B00111101  
                                     );
```

```
Sprite numero_nou_gran_baix = Sprite(8, 8,  
                                     B00000001,  
                                     B00000001,  
                                     B00000001,  
                                     B00000001,  
                                     B00000001,  
                                     B00000001,  
                                     B10000010,  
                                     B01111100  
                                     );
```

```
Sprite simbol_punt = Sprite(1, 8,  
                             B0,  
                             B0,  
                             B0,  
                             B0,  
                             B0,  
                             B0,  
                             B0,  
                             B1  
                             );
```

```
Sprite simbol_dospunts = Sprite(1, 8,  
                                 B0,  
                                 B0,
```



```
B0,  
B1,  
B0,  
B0,  
B0,  
B1  
);
```

```
Sprite simbol_dospunts_gran_dalt = Sprite(1, 8,  
B0,  
B0,  
B0,  
B1,  
B1,  
B0,  
B0,  
B0  
);
```

```
Sprite simbol_dospunts_gran_baix = Sprite(1, 8,  
B0,  
B0,  
B0,  
B0,  
B0,  
B0,  
B1,  
B1  
);
```

```
void setup() {  
  attachInterrupt(digitalPinToInterrupt(23), Actualitzar_rellotge, RISING);  
  Wire.begin(); //TWI Bus is formed  
  Wire.beginTransmission(RTCAddress); //device address and STSRT command are queued  
  Wire.write(0x0E); //Control Register Address is queued  
  Wire.write(0x00); //Data for Control Register is queued  
  Wire.endTransmission(); //queued information are transferred under ACK; STOP
```

```
linia1.clear();
linia2.clear();
linia3.clear();
linia4.clear();
linia5.clear();
WiFi.setPins(10, 3, 5); //pins on esta el modul connectat
Serial.begin(9600);

pinMode(14, OUTPUT); //comprovacions

// check for the presence of the shield:
if (WiFi.status() == WL_NO_SHIELD) {
  Serial.println("WiFi shield not present");
  // don't continue
  while (true);
}

// print the network name (SSID);
Serial.print("Creating access point named: ");
Serial.println(APssid);

// Create open network. Change this line if you want to create an WEP network:
status = WiFi.beginAP(APssid);
if (status != WL_AP_LISTENING) {
  Serial.println("Creating access point failed");
  // don't continue
  while (true);
}

// wait 10 seconds for connection:
delay(10000);

// start the web server on port 80
server.begin();

// you're connected now, so print out the status
printWiFiStatus();
```

```
}

void loop() {
  if (tenim_ssid == false) {
    Connexio_inicial();
  }
  if (tenim_ssid == true) {
    //server.begin();
    Servidor_web();
  }
  /*if (xxxxxx == ) {
    Servidor_web();
  }*/

  String dades = "Roger Gasull Vilagran"; //Variable que agafem del commentator,
  ex: Roger Gasull Vilagran

  String nom = dades[0]; //Agafem la primera lletra de dades, ex: R
  String punt = "."; //Definim el punt
  unsigned int longitud_string = dades.length(); //Longitud (caracters) de dades,
  ex: 20
  unsigned int posicio_espai = dades.indexOf(' '); //Posicio primer espai, ex: 5
  unsigned int posicio_cognom = dades.indexOf(' ', posicio_espai + 1); //Posicio
  segon espai, ex: 12
  unsigned int inici_cognom = posicio_espai + 1;
  unsigned int final_cognom = posicio_cognom - 1;

  String cognom = dades.substring(inici_cognom, final_cognom); //Agafem els
  caracters d'espai a espai (1r cognom)

  String nom_mostrar = nom + punt + cognom; //Nom que es veura per pantalla, ex:
  R.GASULL

  byte x; //x serveix pel for
  unsigned int longitud_nom_mostrar = nom_mostrar.length();
  for (x = 0; x <= longitud_nom_mostrar; x++) {
    if (nom_mostrar.substring(x) == ('A' or 'a')) {
      linia3.write(p, 0, lletra_A);
    }
    else if (nom_mostrar.substring(x) == ('B' or 'b')) {
      linia3.write(p, 0, lletra_B);
    }
  }
}
```

```
else if (nom_mostrar.substring(x) == ('C' or 'c')) {
    linia3.write(p, 0, lletra_C);
}
else if (nom_mostrar.substring(x) == ('D' or 'd')) {
    linia3.write(p, 0, lletra_D);
}
else if (nom_mostrar.substring(x) == ('E' or 'e')) {
    linia3.write(p, 0, lletra_E);
}
else if (nom_mostrar.substring(x) == ('F' or 'f')) {
    linia3.write(p, 0, lletra_F);
}
else if (nom_mostrar.substring(x) == ('G' or 'g')) {
    linia3.write(p, 0, lletra_G);
}
else if (nom_mostrar.substring(x) == ('H' or 'h')) {
    linia3.write(p, 0, lletra_H);
}
else if (nom_mostrar.substring(x) == ('I' or 'i')) {
    linia3.write(p, 0, lletra_I);
}
else if (nom_mostrar.substring(x) == ('J' or 'j')) {
    linia3.write(p, 0, lletra_J);
}
else if (nom_mostrar.substring(x) == ('K' or 'k')) {
    linia3.write(p, 0, lletra_K);
}
else if (nom_mostrar.substring(x) == ('L' or 'l')) {
    linia3.write(p, 0, lletra_L);
}
else if (nom_mostrar.substring(x) == ('M' or 'm')) {
    linia3.write(p, 0, lletra_M);
}
else if (nom_mostrar.substring(x) == ('N' or 'n')) {
    linia3.write(p, 0, lletra_N);
}
else if (nom_mostrar.substring(x) == ('O' or 'o')) {
```

```
        linia3.write(p, 0, lletra_O);
    }
    else if (nom_mostrar.substring(x) == ('P' or 'p')) {
        linia3.write(p, 0, lletra_P);
    }
    else if (nom_mostrar.substring(x) == ('Q' or 'q')) {
        linia3.write(p, 0, lletra_Q);
    }
    else if (nom_mostrar.substring(x) == ('R' or 'r')) {
        linia3.write(p, 0, lletra_R);
    }
    else if (nom_mostrar.substring(x) == ('S' or 's')) {
        linia3.write(p, 0, lletra_S);
    }
    else if (nom_mostrar.substring(x) == ('T' or 't')) {
        linia3.write(p, 0, lletra_T);
    }
    else if (nom_mostrar.substring(x) == ('U' or 'u')) {
        linia3.write(p, 0, lletra_U);
    }
    else if (nom_mostrar.substring(x) == ('V' or 'v')) {
        linia3.write(p, 0, lletra_V);
    }
    else if (nom_mostrar.substring(x) == ('W' or 'w')) {
        linia3.write(p, 0, lletra_W);
    }
    else if (nom_mostrar.substring(x) == ('X' or 'x')) {
        linia3.write(p, 0, lletra_X);
    }
    else if (nom_mostrar.substring(x) == ('Y' or 'y')) {
        linia3.write(p, 0, lletra_Y);
    }
    else if (nom_mostrar.substring(x) == ('Z' or 'z')) {
        linia3.write(p, 0, lletra_Z);
    }
    else if (nom_mostrar.substring(x) == ('.')) {
        linia3.write(p, 0, simbol_punt);
    }
```

```
    p = p - 4; //Les lletres tenen 5+1 leds i el punt 1+1 leds
  }
  p = p + leds_lletra;
}

}

/*FUNCIONS*/

void Servidor_web() {
  status = WiFi.status();

  if (status != WL_CONNECTED) { // Comparar l'estat anterior amb l'estat actual
    WiFi.begin(ssid, pass); // Inicialitza la configuració de xarxa de la
    biblioteca WiFi101 i proporciona l'estat actual.
    delay(10000); // Esperar 10 segons per la connexió
    server.begin(); // iniciar el servidor web al port 80
    printWiFiStatus();
    status = WiFi.status(); // Actualitzar l'estat de la variable
  }

  //server.begin();
  // listen for incoming clients
  WiFiClient client = server.available();
  if (client) {
    Serial.println("new client");
    // an http request ends with a blank line
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);

        // if you've gotten to the end of the line (received a newline
        // character) and the line is blank, the http request has ended,
        // so you can send a reply
        if (c == '\n' && currentLineIsBlank) {
```

```

// send a standard http response header
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");

//client.println("Connection: close"); // the connection will be closed
after completion of the response

//client.println("Refresh: 5"); // refresh the page automatically every
5 sec

client.println();
//client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<body>");

client.print("<h2>PANELL INFORMATIU</h2>");
client.print("<table style='background-color: #ccffff;'>");
client.print("<tbody>");
client.print("<tr>");
client.print("<td>");
client.print("<h3>RELLOTGE <hr/></h3>");
client.print("<p>Mode funcionament </p>");
client.print("<p>");

client.print("<form>");

client.print("Ascendent: <input id='mode' type='radio' name='mode'
checked='yes' /><a href='asce'></a><br />");
client.print("Descendent: <input type='radio' name='mode' /><br />");
client.print("Brillantor (0-15): <input type=text name='brillantor'
value=15><br>");
client.print("</p>");
client.print("</td>");

client.print("<td>");

client.print("<p><span style='color:
#999999;'>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124</span></p>");
client.print("</td>");

client.print("<td>");
client.print("<br>");

```

```

        // client.print("<form style='text-align: center; '> ");
        client.print("<input style='text-align: center; ' type='button'
value='INICI'>");
        // client.print("</form>");

        client.print("<p style='text-align: center; '><em>El rellotge est&agrave
aturat.</em></p>");

        client.print("<p> ");
        client.print("Offset:&nbsp;");

        client.print("+ <input type='radio' name='offset'
checked='yes' />&nbsp;&nbsp;&nbsp;");
        client.print("- <input type='radio' name='mode' /><br />");
        client.print("</p>");
        client.print("<form>");
        client.print("<p>");
        client.print("Hores (0-23): <input type=text name='hores' value=0><br>");
        client.print("Minuts (0-59): <input type=text name='minuts'
value=0><br>");
        client.print("Segons (0-59): <input type=text name='segons'
value=0><br>");

        client.print("<input type='submit' value='Actualitza' ><br>");
        client.print("</p>");
        client.print("</form>");
        client.print("</td>");
        client.print("</tr>");
        client.print("</tbody>");
        client.print("</table>");

        client.print("<br>");
        client.print("<table style='background-color: #ccffff; '>");
        client.print("<tbody>");
        client.print("<tr>");
        client.print("<td>");

        client.print("<h3>TEXT A MOSTRAR<hr/></h3>");
        client.print("<p>");

        client.print("Temps dels atletes <input type='radio' name='text'
checked='yes' /><br />");
        client.print("Text: <input type='radio' name='text' /><br />");
        client.print("</p>");
        client.print("</td>");

```



```

        client.print("<td>");

        client.print("<p><span style='color:
#999999;'>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124<br>&#124</span></p>");

        client.print("</td>");

        client.print("<td>");

        client.print("<br>");

        client.print("<input type='checkbox' name='primer' value='si' />Primer
sempre es mostra<br/>");

        client.print("<form>");

        client.print("<p>");

        client.print("URL: <input type='text' name='url'><br><br>");

        client.print("L&iacutenia 1: <input type='text' name='linia1'
value=0><br>");

        client.print("L&iacutenia 2: <input type='text' name='linia2'
value=0><br>");

        client.print("L&iacutenia 3: <input type='text' name='linia3'
value=0><br>");

        client.print("<input type='submit' value='Envia' ><br>");

        client.print("</p>");

        client.print("</form>");

        client.print("</td>");

        client.print("</tr>");

        client.print("</tbody>");

        client.print("</table>");

        client.println("</body>");

        client.println("</html>");

        break;
    }

    if (c == '\n') {
        // you're starting a new line
        currentLineIsBlank = true;
    }

    else if (c != '\r') {
        // you've gotten a character on the current line
        currentLineIsBlank = false;
    }
}

```

```
    }
}
// give the web browser time to receive the data
delay(1);

// close the connection:
client.stop();
Serial.println("client disconnected");
}
}

void Connexio_inicial() {
    if (status != WiFi.status()) { // Comparar l'estat anterior amb l'estat actual
        status = WiFi.status(); // Actualitzar l'estat de la variable
        if (status == WL_AP_CONNECTED) { // Un dispositiu s'ha connectat a l'AP
            Serial.print("Device connected to AP");
        } else { // El dispositiu s'ha desconnectat de l'AP i tornem a estar en mode
d'escolta
            Serial.println("Device disconnected from AP");
        }
    }
}

WiFiClient client = server.available(); // listen for incoming clients

if (client) { // if you get a client,
    Serial.println("new client"); // print a message out the serial port
    String currentLine = ""; // make a String to hold incoming data
from the client
    while (client.connected()) { // loop while the client's connected
        if (client.available()) { // if there's bytes to read from the
client,
            char c = client.read(); // read a byte, then
            //Serial.print(c); // print it out the serial monitor
            //String ssid_pass(c);
            //Serial.print(ssid_pass);

            if (c == '\n') { // if the byte is a newline character
                // if the current line is blank, you got two newline characters in a row.
```

```
// that's the end of the client HTTP request, so send a response:
if (currentLine.length() == 0) {
    client.println("HTTP/1.1 200 OK"); // HTTP headers always start with a
response code (e.g. HTTP/1.1 200 OK)

    client.println("Content-type:text/html"); // and a content-type so the
client knows what's coming, and

    client.println(); //then a blank line

// the content of the HTTP response follows the header:
//client.print("Click <a href=\"/H\">here</a> turn the LED on<br>");
//client.print("Click <a href=\"/L\">here</a> turn the LED off<br>");

client.print("<h2>PANELL INFORMATIU</h2>");
client.print("<h3>Connecta&rsquo;t a la xarxa</h3>");
client.print("<form>");
client.print("<p><br>");
client.print("Nom de la xarxa<br> <input type=text name='ssid'><br>");
client.print("<br>");
client.print("Contrasenya<br> <input type=text name='pass'><br>");
client.print("<br>");
client.print("<input type='submit' value='Connecta&rsquo;t'><br>");
client.print("</p><br>");
client.print("</form><br>");

client.println(); // The HTTP response ends with another blank line
break; // break out of the while loop:
}
else { // if you got a newline, then clear currentLine:
    currentLine = "";
}
}

else if (c != '\r') { // if you got anything else but a carriage return
character,
    currentLine += c; // add it to the end of the currentLine
}

if ((currentLine.startsWith("GET /?ssid=")) &&
(currentLine.endsWith("HTTP"))) {
```

```
String ssid_pass = currentLine;
ssid_pass.replace("+", " ");
//Serial.println(ssid_pass);
byte inici_ssid = ssid_pass.indexOf("ssid=");
byte separador = ssid_pass.indexOf("&pass=");
byte fi_pass = ssid_pass.indexOf(" HTTP");

if ((inici_ssid != 255) && (separador != 255) && (fi_pass != 255) &&
(tenim_ssid == 0)) {

    //Serial.println(inici_ssid);
    //Serial.println(separador);
    //Serial.println(fi_pass);

    ssid = ssid_pass.substring(inici_ssid + 5, separador);
    pass = ssid_pass.substring(separador + 6, fi_pass);

    Serial.println(ssid);
    Serial.println(pass);

    WiFi.disconnect(); // Desconnecta Teensy de la xarxa actual
    WiFi.end(); // Desactiva el mòdul WiFi

    WiFi.begin(ssid, pass); // Inicialitza la configuració de xarxa de la
biblioteca WiFi101 i proporciona l'estat actual.

    delay(10000); // Esperar 10 segons per la connexió

    server.begin(); // iniciar el servidor web al port 80

    printWiFiStatus();

    WiFi.disconnect(); //prova!!
    WiFi.end();//prova!!

    tenir_ssid = true;

}

}

}

}

client.stop(); // Tanca la connexió

Serial.println("client disconnected");

}

}

void Recuperar_connexio() {

    //en cas que perdi la connexio, aqui va el programa per recuperar-la

}
```

```
void printWiFiStatus() {
    Serial.print("SSID: ");

    Serial.println(WiFi.SSID()); // Imprimeix el SSID de la xarxa que estem
connectats

    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip); // Imprimeix la nostra IP
}

void Actualitzar_rellotge() {
    if (rellotge_iniciat == true) {
        interrupcions = interrupcions + 1;
        if (interrupcions >= 60) {
            minuts = minuts + 1;
            interrupcions = interrupcions - 60;
        }

        if (interrupcions == 0 || interrupcions == 10 || interrupcions == 20 ||
interrupcions == 30 || interrupcions == 40 || interrupcions == 50) {
            linia1.write(92, 0, numero_zero_gran_dalt);
            linia2.write(92, 0, numero_zero_gran_baix);
        }

        else if (interrupcions == 1 || interrupcions == 11 || interrupcions == 21 ||
interrupcions == 31 || interrupcions == 41 || interrupcions == 51) {
            linia1.write(92, 0, numero_un_gran_dalt);
            linia2.write(92, 0, numero_un_gran_baix);
        }

        else if (interrupcions == 2 || interrupcions == 12 || interrupcions == 22 ||
interrupcions == 32 || interrupcions == 42 || interrupcions == 52) {
            linia1.write(92, 0, numero_dos_gran_dalt);
            linia2.write(92, 0, numero_dos_gran_baix);
        }

        else if (interrupcions == 3 || interrupcions == 13 || interrupcions == 23 ||
interrupcions == 33 || interrupcions == 43 || interrupcions == 53) {
            linia1.write(92, 0, numero_tres_gran_dalt);
            linia2.write(92, 0, numero_tres_gran_baix);
        }
    }
}
```

```
    else if (interrupcions == 4 || interrupcions == 14 || interrupcions == 24 ||
interrupcions == 34 || interrupcions == 44 || interrupcions == 54) {
        linia1.write(92, 0, numero_quatre_gran_dalt);
        linia2.write(92, 0, numero_quatre_gran_baix);
    }
    else if (interrupcions == 5 || interrupcions == 15 || interrupcions == 25 ||
interrupcions == 35 || interrupcions == 45 || interrupcions == 55) {
        linia1.write(92, 0, numero_cinc_gran_dalt);
        linia2.write(92, 0, numero_cinc_gran_baix);
    }
    else if (interrupcions == 6 || interrupcions == 16 || interrupcions == 16 ||
interrupcions == 36 || interrupcions == 46 || interrupcions == 56) {
        linia1.write(92, 0, numero_sis_gran_dalt);
        linia2.write(92, 0, numero_sis_gran_baix);
    }
    else if (interrupcions == 7 || interrupcions == 17 || interrupcions == 27 ||
interrupcions == 37 || interrupcions == 47 || interrupcions == 57) {
        linia1.write(92, 0, numero_set_gran_dalt);
        linia2.write(92, 0, numero_set_gran_baix);
    }
    else if (interrupcions == 8 || interrupcions == 18 || interrupcions == 28 ||
interrupcions == 38 || interrupcions == 48 || interrupcions == 58) {
        linia1.write(92, 0, numero_vuit_gran_dalt);
        linia2.write(92, 0, numero_vuit_gran_baix);
    }
    else if (interrupcions == 9 || interrupcions == 19 || interrupcions == 29 ||
interrupcions == 39 || interrupcions == 49 || interrupcions == 59) {
        linia1.write(92, 0, numero_nou_gran_dalt);
        linia2.write(92, 0, numero_nou_gran_baix);
    }
    if (interrupcions < 10) {
        linia1.write(83, 0, numero_zero_gran_dalt);
        linia2.write(83, 0, numero_zero_gran_baix);
    }
    else if (interrupcions < 10) {
        linia1.write(83, 0, numero_un_gran_dalt);
        linia2.write(83, 0, numero_un_gran_baix);
    }
    else if ((interrupcions >= 20) && (interrupcions < 30)) {
```

```
        linia1.write(83, 0, numero_dos_gran_dalt);
        linia2.write(83, 0, numero_dos_gran_baix);
    }
else if ((interrupcions >= 30) && (interrupcions < 40)) {
    linia1.write(83, 0, numero_tres_gran_dalt);
    linia2.write(83, 0, numero_tres_gran_baix);
}
else if ((interrupcions >= 40) && (interrupcions < 50)) {
    linia1.write(83, 0, numero_quatre_gran_dalt);
    linia2.write(83, 0, numero_quatre_gran_baix);
}
else if ((interrupcions >= 50) && (interrupcions < 60)) {
    linia1.write(83, 0, numero_cinc_gran_dalt);
    linia2.write(83, 0, numero_cinc_gran_baix);
}

linia1.write(81, 0, simbol_dospunts_gran_dalt);
linia2.write(81, 0, simbol_dospunts_gran_baix);
if (minuts >= 60) {
    hores = hores + 1;
    minuts = minuts - 60;
}

if (minuts == 0 || minuts == 10 || minuts == 20 || minuts == 30 || minuts == 40
|| minuts == 50) {
    linia1.write(72, 0, numero_zero_gran_dalt);
    linia2.write(72, 0, numero_zero_gran_baix);
}
else if (minuts == 1 || minuts == 11 || minuts == 21 || minuts == 31 || minuts
== 41 || minuts == 51) {
    linia1.write(72, 0, numero_un_gran_dalt);
    linia2.write(72, 0, numero_un_gran_baix);
}
else if (minuts == 2 || minuts == 12 || minuts == 22 || minuts == 32 || minuts
== 42 || minuts == 52) {
    linia1.write(72, 0, numero_dos_gran_dalt);
    linia2.write(72, 0, numero_dos_gran_baix);
}
```

```
    else if (minuts == 3 || minuts == 13 || minuts == 23 || minuts == 33 || minuts
== 43 || minuts == 53) {
        linia1.write(72, 0, numero_tres_gran_dalt);
        linia2.write(72, 0, numero_tres_gran_baix);
    }
    else if (minuts == 4 || minuts == 14 || minuts == 24 || minuts == 34 || minuts
== 44 || minuts == 54) {
        linia1.write(72, 0, numero_quatre_gran_dalt);
        linia2.write(72, 0, numero_quatre_gran_baix);
    }
    else if (minuts == 5 || minuts == 15 || minuts == 25 || minuts == 35 || minuts
== 45 || minuts == 55) {
        linia1.write(72, 0, numero_cinc_gran_dalt);
        linia2.write(72, 0, numero_cinc_gran_baix);
    }
    else if (minuts == 6 || minuts == 16 || minuts == 16 || minuts == 36 || minuts
== 46 || minuts == 56) {
        linia1.write(72, 0, numero_sis_gran_dalt);
        linia2.write(72, 0, numero_sis_gran_baix);
    }
    else if (minuts == 7 || minuts == 17 || minuts == 27 || minuts == 37 || minuts
== 47 || minuts == 57) {
        linia1.write(72, 0, numero_set_gran_dalt);
        linia2.write(72, 0, numero_set_gran_baix);
    }
    else if (minuts == 8 || minuts == 18 || minuts == 28 || minuts == 38 || minuts
== 48 || minuts == 58) {
        linia1.write(72, 0, numero_vuit_gran_dalt);
        linia2.write(72, 0, numero_vuit_gran_baix);
    }
    else if (minuts == 9 || minuts == 19 || minuts == 29 || minuts == 39 || minuts
== 49 || minuts == 59) {
        linia1.write(72, 0, numero_nou_gran_dalt);
        linia2.write(72, 0, numero_nou_gran_baix);
    }
    if (minuts < 10) {
        linia1.write(63, 0, numero_zero_gran_dalt);
        linia2.write(63, 0, numero_zero_gran_baix);
    }
    else if (minuts < 10) {
```



```
        linia1.write(63, 0, numero_un_gran_dalt);
        linia2.write(63, 0, numero_un_gran_baix);
    }
    else if ((minuts >= 20) && (minuts < 30)) {
        linia1.write(63, 0, numero_dos_gran_dalt);
        linia2.write(63, 0, numero_dos_gran_baix);
    }
    else if ((minuts >= 30) && (minuts < 40)) {
        linia1.write(63, 0, numero_tres_gran_dalt);
        linia2.write(63, 0, numero_tres_gran_baix);
    }
    else if ((minuts >= 40) && (minuts < 50)) {
        linia1.write(63, 0, numero_quatre_gran_dalt);
        linia2.write(63, 0, numero_quatre_gran_baix);
    }
    else if ((minuts >= 50) && (minuts < 60)) {
        linia1.write(63, 0, numero_cinc_gran_dalt);
        linia2.write(63, 0, numero_cinc_gran_baix);
    }

    linia1.write(61, 0, simbol_dospunts_gran_dalt);
    linia2.write(61, 0, simbol_dospunts_gran_baix);

    if (hores == 0 || hores == 10 || hores == 20 || hores == 30 || hores == 40 ||
hores == 50) {
        linia1.write(52, 0, numero_zero_gran_dalt);
        linia2.write(52, 0, numero_zero_gran_baix);
    }
    else if (hores == 1 || hores == 11 || hores == 21 || hores == 31 || hores == 41
|| hores == 51) {
        linia1.write(52, 0, numero_un_gran_dalt);
        linia2.write(52, 0, numero_un_gran_baix);
    }
    else if (hores == 2 || hores == 12 || hores == 22 || hores == 32 || hores == 42
|| hores == 52) {
        linia1.write(52, 0, numero_dos_gran_dalt);
        linia2.write(52, 0, numero_dos_gran_baix);
    }
}
```

```
    else if (hores == 3 || hores == 13 || hores == 23 || hores == 33 || hores == 43
|| hores == 53) {
        linia1.write(52, 0, numero_tres_gran_dalt);
        linia2.write(52, 0, numero_tres_gran_baix);
    }
    else if (hores == 4 || hores == 14 || hores == 24 || hores == 34 || hores == 44
|| hores == 54) {
        linia1.write(52, 0, numero_quatre_gran_dalt);
        linia2.write(52, 0, numero_quatre_gran_baix);
    }
    else if (hores == 5 || hores == 15 || hores == 25 || hores == 35 || hores == 45
|| hores == 55) {
        linia1.write(52, 0, numero_cinc_gran_dalt);
        linia2.write(52, 0, numero_cinc_gran_baix);
    }
    else if (hores == 6 || hores == 16 || hores == 26 || hores == 36 || hores == 46
|| hores == 56) {
        linia1.write(52, 0, numero_sis_gran_dalt);
        linia2.write(52, 0, numero_sis_gran_baix);
    }
    else if (hores == 7 || hores == 17 || hores == 27 || hores == 37 || hores == 47
|| hores == 57) {
        linia1.write(52, 0, numero_set_gran_dalt);
        linia2.write(52, 0, numero_set_gran_baix);
    }
    else if (hores == 8 || hores == 18 || hores == 28 || hores == 38 || hores == 48
|| hores == 58) {
        linia1.write(52, 0, numero_vuit_gran_dalt);
        linia2.write(52, 0, numero_vuit_gran_baix);
    }
    else if (hores == 9 || hores == 19 || hores == 29 || hores == 39 || hores == 49
|| hores == 59) {
        linia1.write(52, 0, numero_nou_gran_dalt);
        linia2.write(52, 0, numero_nou_gran_baix);
    }
    if (hores < 10) {
        linia1.write(43, 0, numero_zero_gran_dalt);
        linia2.write(43, 0, numero_zero_gran_baix);
    }
    else if (hores < 10) {
```

```
        linia1.write(43, 0, numero_un_gran_dalt);
        linia2.write(43, 0, numero_un_gran_baix);
    }
    else if ((hores >= 20) && (hores < 30)) {
        linia1.write(43, 0, numero_dos_gran_dalt);
        linia2.write(43, 0, numero_dos_gran_baix);
    }
    else if ((hores >= 30) && (hores < 40)) {
        linia1.write(43, 0, numero_tres_gran_dalt);
        linia2.write(43, 0, numero_tres_gran_baix);
    }
    else if ((hores >= 40) && (hores < 50)) {
        linia1.write(43, 0, numero_quatre_gran_dalt);
        linia2.write(43, 0, numero_quatre_gran_baix);
    }
    else if ((hores >= 50) && (hores < 60)) {
        linia1.write(43, 0, numero_cinc_gran_dalt);
        linia2.write(43, 0, numero_cinc_gran_baix);
    }
}
}
```

## B MANUAL D'INSTRUCCIONS

PAS 1: Endollem el panell.

PAS 2: Ens connectem a la xarxa Wi-Fi del panell.

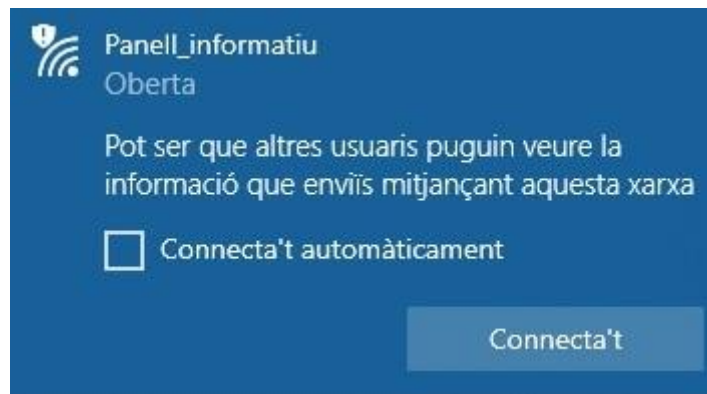


Figura 37. Xarxa WiFi del panell.

PAS 3. Introduïm la direcció IP 192.168.1.1 a qualsevol navegador del dispositiu que hem utilitzat per connectar-nos. Introduïm nom de la xarxa, contrasenya i li donem a connectar.

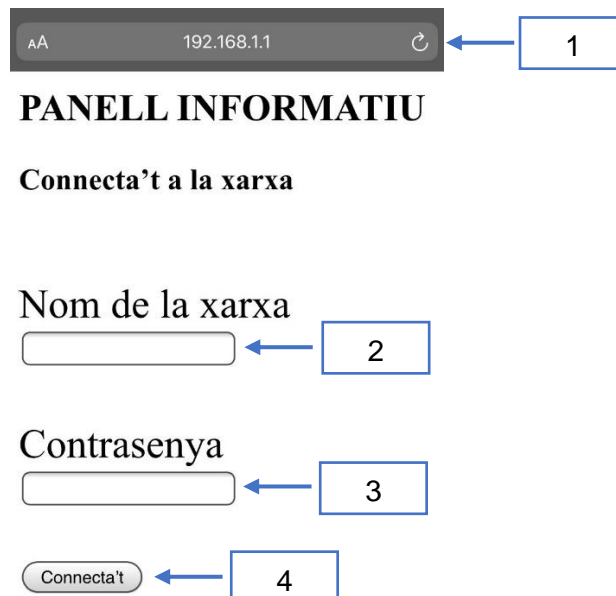


Figura 38. Interfície de connexió.

PAS 4: Ens connectem a la mateixa xarxa Wi-Fi que el panell. I anem a la direcció IP que ens mostri.

PAS 5: Seleccionem les opcions desitjades.

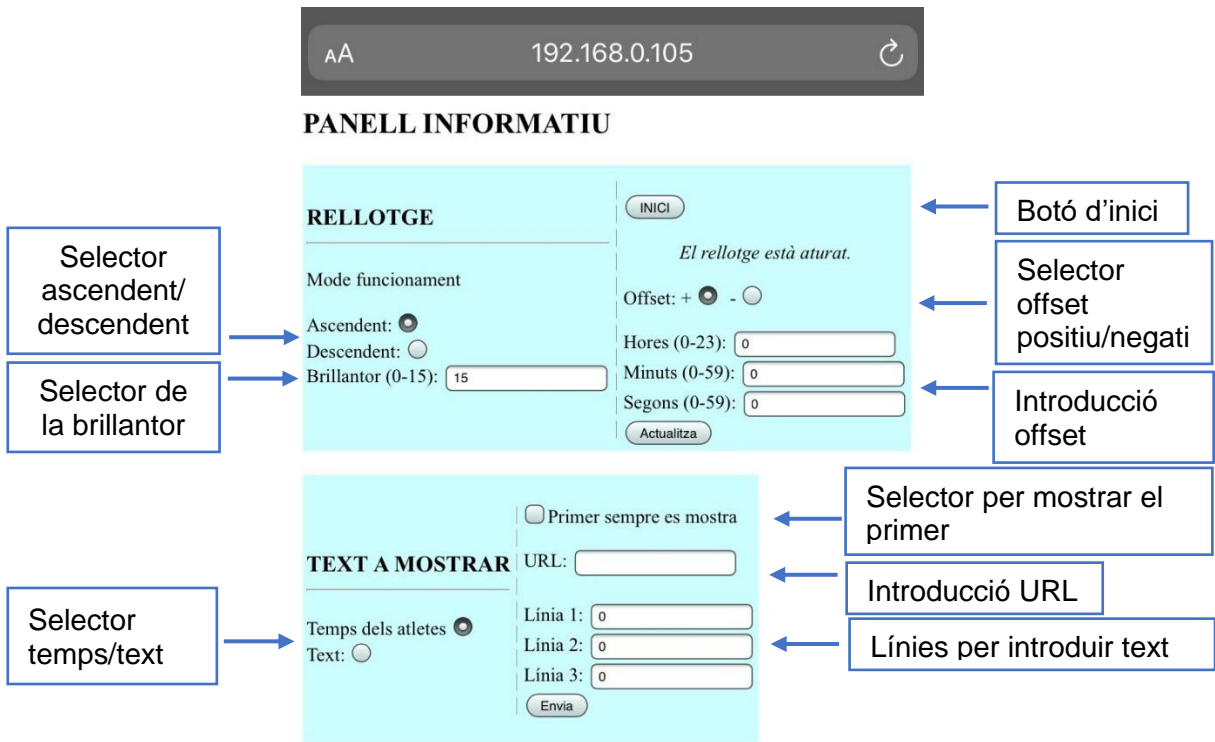


Figura 39. Interfície d'usuari.

La URL a introduir és la generada en el menú Commentator.

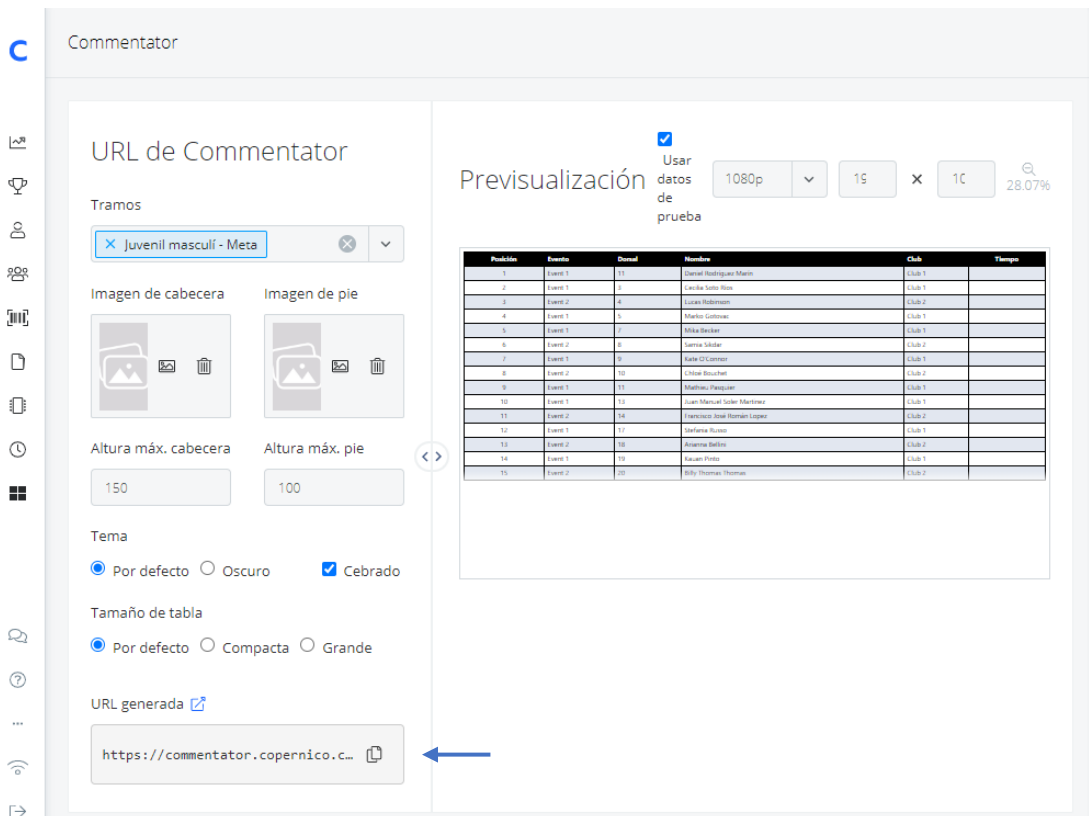


Figura 40. Menú Commentator.

## C CÀLCULS

Amb una resistència de 10kΩ al pin I<sub>SET</sub> cada LED utilitza 50mA amb un cicle de treball de 1/8, amb un corrent mitjà de 6,25mA

### Consum màxim del panell 24x16:

$$\text{Corrent mitjà LED} = I_{LED} * \text{duty cycle} = 50mA * \frac{1}{8} = 6,25mA \quad (\text{Eq. 6})$$

$$\text{Total de LEDs} = 24 * 16 = 384 \text{ leds} \quad (\text{Eq. 7})$$

$$\text{Total de drivers MAX7219} = \frac{384 \text{ leds}}{64 \text{ leds per driver}} = 6 \text{ drivers} \quad (\text{Eq. 8})$$

$$\text{Corrent panell 24x16} = \text{Corrent mitjà LED} * \text{LEDs} + \text{corrent drivers} * \text{drivers} \quad (\text{Eq. 9})$$

$$\text{Corrent panell 24x16} = 6,25mA * 384 + 10mA * 6 = 2.460mA = 2,46A \quad (\text{Eq. 10})$$

El corrent del driver s'ha extret del seu datasheet.

$$\text{Potència panell 24x16} = 2,46A * 5V = 12,30W \quad (\text{Eq. 12})$$

### Consum màxim del panell 24x24:

$$\text{Corrent mitjà LED} = I_{LED} * \text{duty cycle} = 50mA * \frac{1}{8} = 6,25mA \quad (\text{Eq. 13})$$

$$\text{Total de LEDs} = 24 * 24 = 576 \text{ leds} \quad (\text{Eq. 14})$$

$$\text{Total de drivers MAX7219} = \frac{576 \text{ leds}}{64 \text{ leds per driver}} = 9 \text{ drivers} \quad (\text{Eq. 15})$$

$$\text{Corrent panell 24x24} = \text{Corrent mitjà LED} * \text{LEDs} + \text{corrent drivers} * \text{drivers} \quad (\text{Eq. 16})$$

$$\text{Corrent panell 24x24} = 6,25mA * 576 + 10mA * 9 = 3.690mA = 3,69A \quad (\text{Eq. 17})$$

$$Potència panell 24x24 = 3,69A * 5V = 18,45W \quad (\text{Eq. 18})$$

### Consum màxim de la placa de control:

Dels seus datasheets podem extreure els següents consums:

Microcontrolador Teensy 3.2: 100,00mA

Mòdul RTC DS3231: 0,30mA

Transceptor de bus SN74HCT245: 48,08mA

Mòdul Wi-Fi ATWINC1500: 300,00mA

La suma de tots ells fa un total de:

$$Corrent = Teensy + Mòdul RTC + Transceptor * n^{\circ} transceptors + Mòdul WiFi \quad (\text{Eq. 19})$$

$$Corrent = 100,00mA + 0,30mA + 48,08mA * 2 + 300,00mA = 496,46mA = 0,50A \quad (\text{Eq. 20})$$

$$Potència placa de control = 0,50A * 5V = 2,48W \quad (\text{Eq. 21})$$

### Consum panell a una cara

$$P = P_{panell24*16} * n^{\circ} panells + P_{panell24*24} * n^{\circ} panells \quad (\text{Eq. 22})$$

$$P = 12,30 * 6 + 18,45 * 6 = 184,50W \quad (\text{Eq. 23})$$