

Treball final de grau

Estudi: Grau en Enginyeria Electrònica Industrial i Automàtica

Títol: Monitoratge d'una impressora 3D d'extrusió de fil per determinar la influència de l'estratègia de fabricació en les peces fabricades.

Document: 1. Memòria

Alumne: Lluc Vicens Linares

Tutor: Ines Ferrer Real

Departament: Enginyeria Mecànica i de la Construcció Industrial

Àrea: Enginyeria dels Processos de Fabricació

Convocatòria (mes/any): setembre/2023

ÍNDEX

1. INTRODUCCIÓ.....	3
1.1 Antecedents.....	3
1.2 Objecte.....	3
1.3 Especificacions i abast.....	4
2. TECNOLOGIA DE FABRICACIÓ ADDITIVA.....	5
2.1 Classes de tecnologia additiva en la indústria.....	5
2.2 Problemàtica impressores FDM.....	6
3. DISSENY PROVA DE CONCEPTE.....	8
3.1 Muntatge maquinari.....	9
3.2 Programari utilitzat.....	10
3.2.1 Marlin.....	11
3.2.2 Pronter face.....	12
3.2.3 Cura.....	12
4. COMPONENTS ELECTRÒNICS DEL MONITORATGE.....	13
4.1 Impressora FDM.....	14
4.2 Acceleròmetre.....	15
4.3 Sensor ambiental.....	16
4.4 Sensor temperatura.....	17
4.5 Sensor tèrmic d'infraroig.....	19
4.5.1 MLX90614.....	20
4.5.2 MLX90640 32x24.....	20
4.6 Microcontroladors.....	21
4.6.1 Arduino DUE.....	22
4.6.2 Adafruit ESP32-Feather.....	22
5. PROGRAMACIÓ.....	24
5.1 Programa d'adquisició d'acceleracions i paràmetres ambientals.....	24
5.2 Programa càmera.....	26
5.3 Programa interfície de monitoratge.....	28
5.4 Programa tractament d'imatge.....	31
6. MUNTATGE I VALIDACIÓ.....	32

7. DISSENY PLA EXPERIMENTAL.....	37
7.1 Mesura de paràmetres del sistema de monitoratge.....	38
7.2 Mesura de precisió dimensional.....	41
7.3 Mesura de rugositat superficial.....	41
8. RESUM DEL PRESSUPOST.....	45
9. CONCLUSIONS.....	46
10. RELACIÓ DE DOCUMENTS.....	47
11. BIBLIOGRAFIA.....	48
12. GLOSSARI.....	49
A. RESULTAT MESURA DE PARÀMETRES DEL SISTEMA DE MONITORATGE.....	51
A.1 Peça A.....	51
A.2 Peça B.....	53
A.3 Peça C.....	54
A.4 Peça D.....	56
A.5 Peça E.....	57
A.6 Peça F.....	59
B. Format JSON.....	61
C. CÀLCULS.....	63
D. PROGRAMA.....	66
D.1 Codi programa d'adquisició acceleracions i paràmetres ambientals.....	66
D.2 Codi programa càmera.....	70
D.3 Codi interfície de monitoratge.....	72
D.4 Codi programa tractament d'imatge.....	85

1. INTRODUCCIÓ

Amb la realització d'aquest projecte es pretén exposar de forma clara i precisa tota la informació i detalls necessaris per al disseny, programació i implementació d'un sistema de monitoratge d'una impressora 3D d'extrusió de fil.

1.1 Antecedents

Actualment, un dels termes més rellevants en el sector industrial és la Indústria 4.0. Aquest és un terme per anomenar a la quarta revolució industrial i a l'ús de les tecnologies avançades per transformar els processos de fabricació i producció de la indústria. Aquesta nova revolució es caracteritza per la inclusió de IoT en la indústria, la digitalització dels processos i la posterior anàlisi de les dades obtingudes. En el cas de les tecnologies d'impressió 3D, és essencial el monitoratge per tal de garantir la qualitat, fiabilitat i eficiència del procés. Fer un seguiment exhaustiu dels diferents paràmetres durant el procés d'impressió, permet que els operadors puguin detectar problemes més aviat i fer ajustos necessaris per assolir una impressió de gran qualitat.

Per això, el grup de recerca GREP (Enginyeria de Producte, Procés i Producció) de la UdG i Eurecat volen començar una línia de recerca conjunta sobre el monitoratge de processos de fabricació additiva, concretament els d'extrusió de filament, per tal d'adquirir un coneixement més ampli de com diferents factors: les vibracions, la temperatura o la humitat; poden afectar a la qualitat de les peces fabricades, des del punt de vista d'acabat superficial, precisió dimensional o propietats mecàniques.

1.2 Objecte

Dissenyar i implementar un sistema de sensorització en una impressora d'extrusió de fil (conegut com la tecnologia Fused Deposition Modelling (FDM) o Fused Filament Fabrication (FFF)) i, posteriorment, dissenyar un pla experimental, amb diferents estratègies de fabricació i geometries, que permeti analitzar la implementació realitzada des de diferents punts de vista.

1.3 Especificacions i abast

L'abast del present projecte comportarà l'estudi dels paràmetres essencials que determinen la qualitat del procés de fabricació i estat de la impressora. L'estudi i selecció dels sistemes de sensorització més adients que garanteixin la mesura correcta dels paràmetres seleccionats. El disseny del sistema de monitoratge, que inclou part de mecànica, electrònica i desenvolupament de codi de control i la posterior implementació del sistema en una màquina d'impressió 3D FDM comercial i posada a punt. El desenvolupament d'un pla experimental de validació del sistema de monitoratge implementat. I finalment l'anàlisi i valoració de les peces fabricades (acabat superficial i precisió dimensional).

2. TECNOLOGIA DE FABRICACIÓ ADDITIVA

Per tal de posar en context el projecte, primerament s'exposaran alguns conceptes fonamentals de la impressió 3D.

La impressió 3D, altrament coneguda com a fabricació additiva, és una tecnologia basada en la superposició de capes de material per tal de crear objectes tridimensionals. Aquesta tecnologia forma part dels mètodes de fabricació Solid free-form Fabrication (SFF), que engloba totes les tecnologies que permeten formar peces directament d'un disseny creat per ordinador sense la necessitat d'utilitzar motlles. La fabricació additiva en concret, ofereix grans avantatges, respecte als mètodes de fabricació tradicionals, com la reducció de temps i costos de producció, la possibilitat de crear prototips propis de forma ràpida i l'optimització de l'ús de materials reciclats. Al contrari de la creença general, la fabricació additiva no és una tecnologia moderna, el seu naixement es remunta l'any 1986 amb la patentació de la tecnologia SLA per part de Charles Hull. Tot i això, no és fins a l'aparició de la primera impressora de sobretaula l'any 2012 que la fabricació additiva experimenta un creixement exponencial fins a convertir-se en una eina de treball indispensable en diferents camps com l'arquitectura, la indústria automobilística i la moda entre molts altres.

2.1 Classes de tecnologia additiva en la indústria

Existeixen diverses tecnologies d'impressió 3D, cadascuna amb els seus avantatges i inconvenients. En l'actualitat els tres tipus de tecnologies més usades en la indústria són l'estereolitografia (Stereo Lithography Apparatus, SLA), la sinterització làser selectiva (Selective Laser Sintering, SLS) i la deposició de material fos (Fused Deposition Modeling, FDM o Fused Filament Fabrication, FFF).

La tecnologia SLA es basa en la construcció de l'objecte capa a capa mitjançant la curació de resines fotosensibles a la llum ultraviolada. Les impressores SLA permeten la creació d'objectes amb un gran nivell de detall i superfícies suaus. El fet d'utilitzar resines ofereix un ventall de materials disponibles propietats mecàniques i òptiques molt variades. Un dels inconvenients d'aquesta tecnologia és que les resines poden ser tòxiques i requereixen una gran precaució a l'hora de tractar amb elles.

Les impressores més emprades per la gran indústria són les SLS, ja que són molt precises, però també tenen un elevat cost. Aquesta tecnologia permet imprimir en diferents materials, inclosos metalls. El seu funcionament es basa en la dipositació de fines capes de material sobre una plataforma. Un cop dipositada la capa, un làser d'alta potència incideix sobre les partícules per fusionar-les i solidificar la capa. Permet imprimir objectes funcionals i duradors amb diferents tipus de materials. També aconsegueix una alta qualitat d'impressió que permet la producció d'objectes amb geometries complexes. Per contra, cal tenir en compte que aquesta és una tecnologia a l'abast de pocs, ja que tant la impressora com els materials que usa són cars.

Finalment, les impressores més populars i econòmiques són les FDM o FFF. La impressora que se sensoritzarà en aquest projecte fa ús d'aquesta tecnologia, per això és important comprendre el funcionament d'aquesta. Per a la construcció de la peça es diposita filament fos capa a capa sobre un llit calent. La dipositació de capes parteix de la part inferior de la peça i arriba a la part superior en completar la impressió. El material plàstic que s'utilitza s'introdueix en un fusor i aquest escalfa el filament fins a arribar a un estat semilíquid. Un cop arribat aquest estat, el fusor diposita el material seguint les trajectòries indicades sobre el llit calent per crear la primera capa. Seguidament, l'extrusor es mou verticalment una distància concreta, anomenada alçada de capa, per començar a crear la segona capa, que queda adherida a la primera. Aquest procés es repeteix de manera successiva fins a acabar la impressió.

En quant el desplaçament pels tres eixos que formen l'espai se segueix el següent criteri: eix X, pel desplaçament horitzontal, eix Y pel desplaçament frontal i eix Z pel desplaçament vertical. Tot i que sempre es manté aquest criteri, no totes les impressores 3D compten amb la mateixa configuració de maquinari. Per aquest motiu, el moviment Y en algunes impressores, el fa el llit calent i el fusor només es mou verticalment i horitzontalment. I en altres, el llit és fix, i és el fusor l'encarregat de desplaçar-se pels tres eixos.

2.2 Problemàtica impressores FDM

Tot i els grans avantatges d'aquesta tecnologia com són la seva accessibilitat, facilita d'ús i la possibilitat de treballar amb varietat de materials (PLA, PETG, ABS, TPU), en ser la més econòmica de totes les esmentades compte amb un seguit de problemes que es poden donar durant la impressió i que redueixen la qualitat final de la peça impresa. Cal tenir presents aquests errors (Figura 1), ja que amb el present projecte es vol monitorar diferents

variables del procés d'impressió per detectar-los i reduir-los. Un dels més comuns és la desadhesió de les cantonades de la peça o de la peça completa, a causa d'una temperatura de llit menor a l'adient. També són errors habituals l'aparició d'esquerdes en la base de les peces a causa d'una temperatura massa baixa de l'extrusor o la influència de paràmetres ambientals, el desplaçament de capes a causa dels moviments fets en els eixos X i Y, o l'aparició de vibracions en les parets exteriors i emplenament inconsistent o incorrecte degut a una elevada velocitat del capçal extrusor. Mitjançant la implementació d'un sistema de monitoratge robust, es poden aconseguir impressions 3D consistents i d'alta qualitat, reduir els residus de material i millorar l'eficiència general del procés d'impressió.

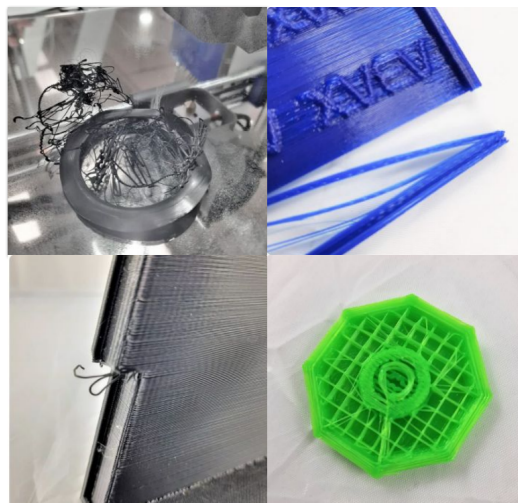


Figura 1: Errors comuns en impressores FDM.

En aquest projecte, es farà un seguiment de la temperatura de l'extrusor i el llit, ja que les desviacions d'aquestes temperatures del seu rang òptim poden provocar problemes com deformacions, mala adhesió de les capes o obstrucció del filament. També se sensoritzarà la broqueta de l'extrusor per controlar el procés d'impressió en temps real, assegurant que cada capa s'adhereix correctament a l'anterior i no hi ha buits. Addicionalment, es farà una supervisió de la velocitat del capçal d'impressió i el moviment del llit per ajudar a garantir que la impressió progressa al ritme esperat, ja que els canvis sobtats de velocitat o moviment poden indicar problemes mecànics. Finalment, l'últim paràmetre que es monitorarà serà la temperatura i la humitat ambiental per veure la seva influència en el procés d'impressió.

3. DISSENY PROVA DE CONCEPTE

Abans d'iniciar el projecte, s'ha cregut necessari dissenyar una prova de concepte per comprendre a fons el funcionament tant del maquinari com del programari de les impressores 3D FDM. Treballar directament sobre una impressora 3D comercial impedeix entendre la tecnologia FDM a més baix nivell, ja que no es veu la programació de la placa base. El fet d'iniciar el projecte sobre un prototip permet manipular el microprogramari de la placa base comprnent així cada element del muntatge i fer proves sense el perill de malmetre la impressora que posteriorment s'utilitzarà en el projecte.

Primerament, s'han estudiat el conjunt d'elements que conformen les impressores FDM de baix cost per tal de crear una maqueta amb les mateixes prestacions. Aquest muntatge pretén ser un primer contacte amb el maquinari, programari i comunicació a més baix nivell utilitzada en la tecnologia de fabricació additiva.

A continuació s'adjunta una imatge (Figura 2) del muntatge final de la maqueta emprada en aquest projecte.

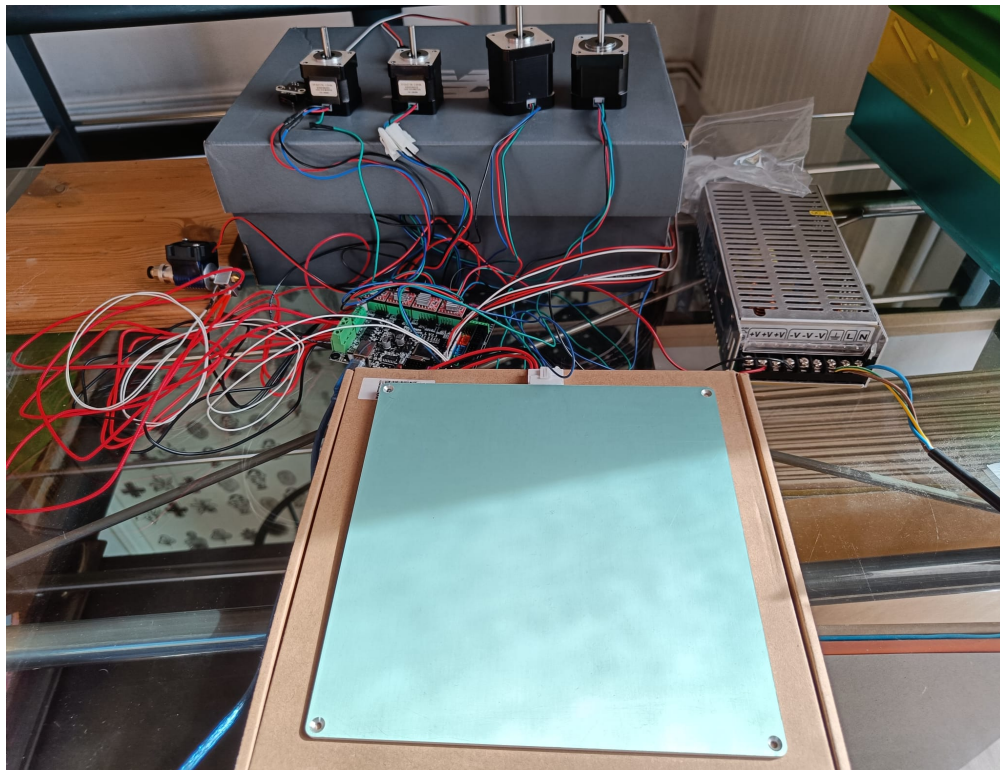


Figura 2: Prova de concepte en funcionament.

3.1 Muntatge maquinari

En aquest apartat s'explicaran tots els elements principals que formen part de qualsevol impressora 3D FDM i es concretaran aquells que seran incorporats en la maqueta que s'ha desenvolupat, per tal d'entendre el seu funcionament, ja que és fonamental per poder desenvolupar el present projecte, veure Figura 3. Aquests elements principals són: la placa base, els motors pas a pas, els drivers de control dels motors, els finals de cursa, la font d'alimentació, el llit calent i l'extrusor.

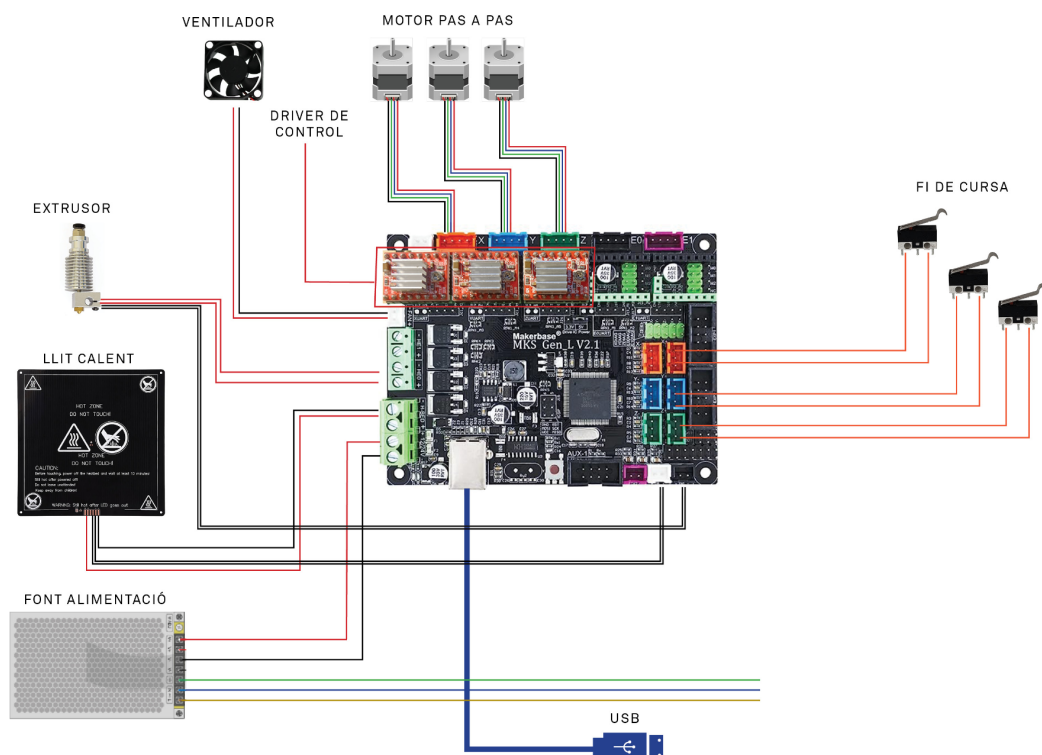


Figura 3: Esquema connexions dels elements de la maqueta.

L'element cabdal en una impressora 3D és la placa base, ja que és l'encarregada de llegir les entrades, escriure a les sortides, realitzar el control segons comandes i executar el codi. Per a la maqueta s'ha escollit la placa MKS Gen L V2.1, aquesta és una placa de control d'impressores 3D que integra el microcontrolador ATMEGA2560. En ser una placa especialitzada en el control d'impressores, ja compta amb tot el seguit d'entrades i sortides necessàries per connectar-hi el conjunt d'elements que intervenen en el procés d'impressió.

Pel que fa a la cinemàtica, els motors més utilitzats en la tecnologia de la fabricació additiva són els pas a pas. S'utilitzen aquest tipus de motors perquè permeten fer moviments molt precisos sense la necessitat d'un control PID amb llaç tancat. En aquest muntatge s'usen tres motors, un per a cada eix de moviment amb què compta la impressora (X, Y, Z). Per controlar l'avanç del filament a través de l'extrusor també s'usa un motor (E). Pel que fa a la prova de concepte, com la impressió serà simulada, no s'ha tingut en compte aquest últim motor.

Per tal de traduir els senyals generats per la placa en moviments dels motors pas a pas és necessari emprar drivers de control al muntatge. En aquest cas s'han escollit tres controladors A4988, ja que són econòmics i compatibles amb la MKS Gen L V2.1. També s'han utilitzat tres finals de cursa, un per a cada eix, per simular la posició de repòs de la maqueta.

Finalment, també s'ha afegit un llit calent al muntatge. Aquest és la plataforma sobre la qual s'imprimeixen els objectes. Compte amb una resistència integrada la qual és l'encarregada de generar temperatura perquè les peces es puguin adherir amb facilitat. El llit que s'ha escollit pel muntatge compte amb unes dimensions estàndard de 220x220 mm. D'aquesta manera s'aconsegueix que tant el temps d'escalfament com de refredament de la superfície en la simulació s'aproximin al del projecte real. Addicionalment, s'ha inclòs un kit d'extrusió que disposa de dissipador, bloc calefactor i broqueta juntament amb ventilador. Aquest s'encarrega d'escalfar el filament i dipositar-lo per capes sobre la superfície d'impressió. Tot i que no s'utilitzarà la maqueta per imprimir, la seva inclusió en el muntatge servirà per aprendre el funcionament de la tecnologia, els codis de control CNC, i a monitorar la temperatura de l'extrusor i del llit.

Per últim, per tal d'abastir a tots els components de tensió s'ha incorporat una font d'alimentació de 12 V.

3.2 Programari utilitzat

En aquest apartat s'explicaran tot el seguit de programes que s'han utilitzat un cop muntada la maqueta tant per configurar la placa com per dur a terme les diferents proves.

3.2.1 Marlin

En comptar amb una placa de control és necessari carregar-li el programari amb la configuració del muntatge. Actualment, existeixen dos programaris lliures, Klipper i Marlin. Aquests, permeten a la impressora llegir i executar les comandes CNC del codi que es carrega a la impressora, i controlar el moviment dels motors, base, extrusor i ventiladors.

El Marlin és un microprogramari senzill i flexible que es carrega a la mateixa placa de la impressora. Al ser més senzill compte amb molta informació fàcil de trobar, permet una gran customització i destaca per la seva gran estabilitat. Per una altra banda, aquest microprogramari, és limitat per la potència de la placa de la impressora, fet que comporta a obtenir un procés més lent i amb acabats de menys qualitat. A diferència del Marlin, Klipper processa totes les comandes a través d'una placa externa, Raspberry Pi, el que suposa passar d'un processament de 8 bits a un de 32, i permet aconseguir una impressió més ràpida i de més qualitat. Per contra, el fet d'utilitzar una placa externa fa que el procés sigui més complex d'entendre i programar, i apareix la necessitat de comptar amb maquinari extern.

La impressora que finalment s'ha escollit per sensoritzar, l'Anycubic Kobra Neo, disposa d'una placa base de 8 bits que executa Marlin 2.0, per aquest motiu s'ha escollit programar la placa de la maqueta amb aquest mateix microprogramari. Aquest fet deixa la porta oberta per pròxims projectes on es vulgui millor la velocitat d'impressió implementant Klipper en lloc de Marlin.

Pel que fa a la programació amb Marlin és senzilla. Primerament, s'ha de descarregar l'última versió de Marlin de la seva web i obrir-la en un editor de codi com el Visual Studio Code. Amb el codi obert, s'ha de canviar els paràmetres segons les especificacions del muntatge del projecte, com són el número d'extrusors, temperatura màxima del llit i de l'extrusor, tipus de sensors de temperatura i drivers que s'utilitzen.

Un cop carregat el microprogramari a la placa base amb totes les especificacions del muntatge, s'han utilitzat dos programes més per poder simular una impressió i estudiar el comportament d'una màquina FDM.

3.2.2 Pronter face

Per tal de simular el funcionament en temps real d'una impressora 3D s'ha fet servir el programa Pronter Face. Aquest és un programari de codi obert que permet controlar i interactuar amb la impressora. Proporciona una interfície gràfica on es pot observar el monitoratge de l'estat de la impressora, visualitzar informació important durant la impressió (temperatures de llit i extrusor) i permet enviar comandes a la màquina.

Amb aquest programa s'ha pogut analitzar la resposta d'una impressora en rebre comandes amb G-Code. D'aquesta manera s'ha pogut entendre el funcionament de la comunicació en control numèric.

3.2.3 Cura

Ultimaker Cura també és un programari de codi obert que permet carregar, escalar i ajustar paràmetres dels models 3D que es vol imprimir. En el projecte el Cura només s'ha utilitzat per transformar els objectes en format STL (objecte 3D) en l'arxiu G-Code que es carregarà a la impressora per ser executat.

4. COMPONENTS ELECTRÒNICS DEL MONITORATGE

En aquest apartat es mostraran i es justificaran les eleccions dels diferents sensors i components electrònics que s'han escollit per tal de dur a terme el monitoratge.

Abans d'escollir els elements que formaran part del muntatge cal tenir clars els paràmetres que es volen sensoritzar. La temperatura i humitat relativa de l'ambient, són dues de les variables que interessa monitorar per veure com afecten els factors ambientals a l'acabat final de les peces. També es vol mantenir un control de temperatura tant del llit calent com de l'extrusor, per comprovar que els paràmetres entrats per programa es mantenen al llarg de la impressió. Juntament amb les temperatures, interessa fer un seguiment de les acceleracions en els tres eixos de moviment per estudiar el comportament de la impressora a l'hora de fer diferents geometries i emplenament de les peces. Finalment, es vol mantenir un control de la temperatura del material recent extrudit i de la punta de l'extrusor, per detectar possibles problemes d'adhesió entre les capes impreses.

Per tal d'entendre el muntatge i la relació de tots els components s'ha creat un esquema com el que es mostra en la Figura 4.

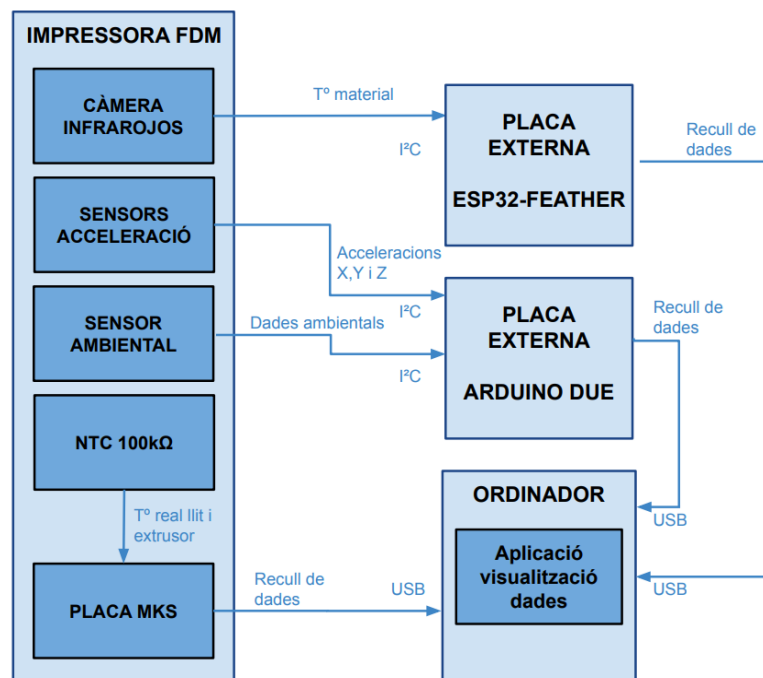


Figura 4: Esquema de les relacions i comunicacions entre els elements del sistema.

Com es pot veure s'utilitzaran dos microcontroladors externs que permetran recollir les dades dels sensors a través de protocol I²C, un cop obtingudes les dades les processaran per poder-les passar a l'ordinador a través de bus serial. Un cop arribin les dades a l'ordinador, aquestes seran tractades a l'aplicació que també es desenvoluparà en aquest projecte, per tal d'estudiar-ne els resultats. Pel que fa a la impressora, tal com es mostra en l'esquema, tots els sensors aniran muntats a ella i posteriorment es cablejaran per connectar-los amb els microcontroladors corresponents. La placa base de la impressora també es comunicarà amb l'ordinador a través de bus serial, per poder depurar la comunicació i recollir les dades de temperatura del llit i de l'extrusor.

A continuació s'exposa l'estudi dels diferents elements electrònics que s'ha elegit per conformar el sistema de monitoratge.

4.1 Impressora FDM

Per tal implementar el sistema de monitoratge el primer que es necessita és una màquina sobre la qual treballar. La impressora 3D que s'ha escollit és l'Anycubic Kobra Neo (Figura 5), aquesta compleix amb tots els requisits que es buscaven. Una impressora econòmica, amb configuració cartesiana, extrusió directa, bon volum d'impressió i una estructura on poder col·locar els nous elements del sistema a implementar. L'Anycubic també compte amb una gran capacitat per treballar amb diferents materials i rangs elevats de temperatura tant de l'extrusor com del llit per poder testejar-los a diferents temperatures.



Figura 5: Anycubic Kobra Neo.

El seguit de característiques que s'han valorat de la impressora escollida venen donades en la Taula 1.

Pes màquina	7,4 kg
Dimensions d'impressió	250x220x220 mm
Dimensió de l'embocadura	∅ 0,4 mm
Velocitat d'impressió	≤100mm/s (Max.) ≤60mm/s (Typ.)
Panell de control	Pantalla tàctil
Material d'impressió	PLA/ABS/PETG/TPU
Tipus d'extrusor	Extrusora directe
Ranura per a targeta de memòria	Targeta SD
Dimensions de la màquina	490x445x443 mm
Temperatura de l'embocadura	≤500 F/260 °C
Temperatura del llit calent	≤230 F/110 °C
Eix Z	Vareta roscada
Anivellament	Anycubic LeviQ anivellació precisa de 25 punts
Plataforma d'impressió	Fulla de metall recoberta de PEI 220x220 mm

Taula 1: Taula de característiques Anycubic Kobra Neo

4.2 Acceleròmetre

Per tal de captar les possibles vibracions que es generen durant la impressió, s'ha optat per utilitzar acceleròmetres. Aquests sensors detecten l'acceleració lineal del moviment en qualsevol dels tres eixos i ajudaran a comprendre l'estabilitat de la màquina i detectar vibracions no desitjades per veure com afecten el producte final.

En aquest cas s'ha escollit el sensor ADXL345 de la casa Analog Devices. Aquest acceleròmetre s'ha estès arreu del camp de la impressió 3D gràcies a la seva versatilitat, fiabilitat i bona relació qualitat preu. Tant és així que microprogramaris lliures com Klipper, ja compten amb suport integrat per aquest sensor. En aquest cas, l'utilitzen per mesurar les freqüències de ressonància de la impressora en els diferents eixos i poder-les compensar de manera automàtica per obtenir un millor acabat final.

Pel que fa a les seves característiques, el sensor ADXL345 és un acceleròmetre de tres eixos, baix consum i alta resolució de 13 bits, amb un rang seleccionable de ±2g, ±4g, ±8g, i

$\pm 16g$. Aquest es pot alimentar a 5 V i compta amb una interfície de comunicació que permet protocols com I²C i SPI. D'aquesta manera es podrà connectar i comunicar el sensor amb el microcontrolador extern sense problemes.

Per a la present aplicació s'empraran dos acceleròmetres ADXL345. Un se situarà a l'extrusor per monitorar els eixos X i Z, i l'altre s'instal·larà a la base del llit per captar les vibracions de l'eix Y. Els dos sensors es comunicaran amb el microprocessador mitjançant I²C. En utilitzar el mateix model de sensor, s'ha de tenir en compte que de sèrie tots dos compten amb la mateixa adreça I²C (0x53). Per tal de modificar l'adreça, es pot fer a través del pin SDO connectant-lo a VCC o GND. Així doncs, les adreces i connexions queden de la següent manera, tal com s'indica a la Taula 2.

Situació ADXL345	Connexió pin SDO	Adreça
Extrusor	GND	0x53
Llit	VCC	0x1D

Taula 2: Connexions pin SDO dels diferents acceleròmetres.

4.3 Sensor ambiental

També es vol veure com afecten els paràmetres ambientals a la impressió, per això cal seleccionar un sensor que capti la temperatura i humitat de l'espai on es trobi la impressora. Per aquesta aplicació s'han considerat dues opcions, el sensor DHT 22 d'Aosong Electronics i l'SHT35 de Sensirion. Tots dos són sensors econòmics, de baix consum i fàcil instal·lació que s'adapten a les necessitats del projecte d'adquirir mesures precises de manera ràpida i fiable.

Els dos sensors poden ser alimentats tant a 3,3 V com a 5 V, això permet la possibilitat d'alimentar-los a través de la mateixa placa externa que recollirà les dades ambientals. Quant a la precisió, l'SHT35 compta amb un $\pm 0,2$ °C per a la temperatura i $\pm 1,5\%$ per la humitat, mentre que el DHT22 té una precisió de $\pm 0,5$ °C i $\pm 2-5\%$. Pel que fa al rang de mesura, ambdós tenen un rang per la humitat del 0 al 100%. On es diferencien és en el rang de temperatura, ja que el DHT22 té un rang de medició de -40 °C a 80 °C en comparació dels -40°C a 125°C del SHT35. Tot i la gran diferència, aquest no és un factor determinant pel fet que en l'aplicació del projecte no se superaran mai els 40 °C ambientals. Un altre factor on destaquen els dos sensors és en la velocitat de resposta. Segons els fulls de

característiques proporcionats pels fabricants, el temps de resposta d'ambdós sensors és de 2 s. Tot i això, en el seguit de proves que s'han dut a terme al laboratori, s'ha vist com el temps de resposta de l'SHT35 és lleugerament menor. Malgrat el gran temps de resposta dels dos sensors, en aquesta aplicació no serà necessari actualitzar els paràmetres ambientals cada segon, ja que tant la temperatura com la humitat ambiental no canvien dràsticament en períodes de temps curts. L'última característica que s'ha valorat per escollir el sensor ambiental ha estat la comunicació. Tenim que l'SHT35 compta amb una interfície digital que permet tant I²C com SPI, mentre que el DHT22 utilitza un senyal digital d'un sol cable.

Finalment, tot i que sobre el paper l'SHT35 és un sensor amb millors prestacions, a través de les proves que s'han dut a terme al laboratori, s'ha vist que DHT22 és una opció més fiable. Ja que és un sensor que porta molt de temps al mercat i amb el qual s'han desenvolupat molts projectes amb èxit. Per aquest motiu per desenvolupar el present projecte s'ha escollit el sensor DHT22.

4.4 Sensor temperatura

A part de captar la temperatura ambiental, també interessa monitorar la temperatura de l'extrusor i del llit calent. Per poder observar el marge d'error entre la temperatura real d'aquests components i la temperatura que s'indica per programa. A diferència dels altres sensors que s'utilitzaran en aquest projecte, aquest són necessaris per al funcionament de la impressora i ja venen inclosos. Totes les impressores porten un sensor de temperatura en contacte amb l'extrusor i un altre sota del llit. La informació d'aquests sensors s'utilitza per monitorar i controlar la temperatura del procés d'impressió. A través del microprogramari la impressora controla la temperatura dels dos actuadors per mitjà de dos PID, assegurant així que la temperatura es mantingui sempre dins d'uns límits permesos.

Pel present projecte es vol analitzar el marge d'error entre la temperatura real de l'extrusor i el llit, i la temperatura entrada per programa. Per fer-ho, s'aprofitaran les sortides dels sensors de temperatura i s'extraurà la informació directament de la mateixa placa de la impressora sense necessitat de passar per cap microcontrolador extern.

En aquest cas, l'Anycubic Kobra Neo usa termistors NTC de 100 k Ω de la casa EPCOS. Aquests són els més populars en el món de la impressió 3D gràcies a la seva precisió, disponibilitat i bona relació qualitat preu. Tant és així que moltes plaques d'impressores 3D

de codi obert venen preparades per connectar-hi aquests sensors. Compten amb una resistència nominal de 100 k Ω , per tant, tal com documenta el fabricant a 25 °C el sensor donarà una resistència de 100 k Ω . Una de les raons per la gran popularitat d'aquest sensor, és pel seu ampli rang de temperatura que permet mesurar amb precisió temperatures compreses entre -55 °C i 300 °C. Pel projecte, aquest rang és necessari, ja que a vegades la temperatura de l'extrusor pot arribar a superar els 260 °C.

Els termistors són sensors de temperatura per resistència, és a dir, varien la seva resistència en funció de la temperatura en què es troben. En el cas dels NTC, el valor de la resistència disminueix a mesura que augmenta la temperatura.

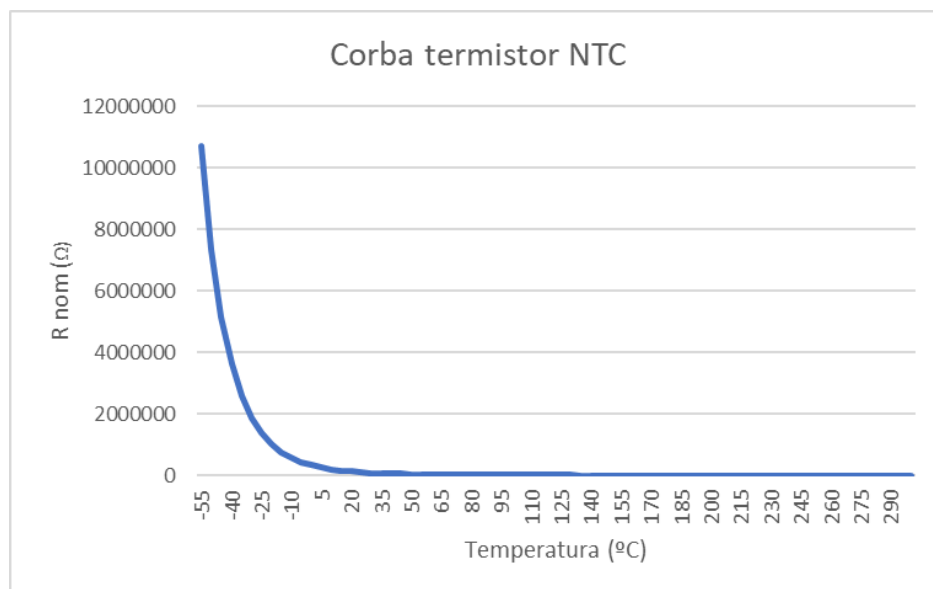


Figura 6: Corba de resistència nominal del termistor respecte a la temperatura.

Per veure la relació resistència nominal amb la temperatura del sensor, s'ha creat un gràfic amb les dades que el fabricant proporciona pel termistor de 100 k Ω . Com es pot veure, en la Figura 6, a diferència de sensors RTD com els PT100, els termistors NTC no tenen una relació resistència-temperatura lineal.

Per aconseguir un senyal de sortida lineal cal tractar-lo. Per aquest motiu les plaques de les impressores ja compten amb tècniques de compensació que incorporen linealitat a la sortida del sensor. En el cas de la impressora que s'utilitzarà en aquest projecte, l'Anycubic Kobra Neo, la mateixa placa compta d'un circuit de condicionament i un convertidor ADC d'alta velocitat per a termistors de 100 k Ω que permeten una ràpida lectura i resposta del sensor de temperatura.

4.5 Sensor tèrmic d'infraroig

Un altre punt del procés d'impressió que es vol monitorar és la temperatura del filament acabat d'extrudir. És important que la temperatura de l'última capa no disminueixi-hi dràsticament per tal d'assegurar una bona adherència amb la següent capa. Aquest fet cobra més importància quan es treballa amb dos materials diferents. Ja de per si, treballar amb dos materials en una mateixa peça comporta una disminució d'adherència entre les capes de diferent material, si s'afegeix el refredament de la capa inferior, encara s'agreuja més el problema. Per aquest motiu es vol mantenir un control de la temperatura de l'última capa per observar fins a quin punt influeixen els canvis de temperatura en el resultat final de la peça i en un futur utilitzar aquesta informació per millorar el procés d'impressió.

A l'hora d'escollir un sensor adequat s'han tingut en compte un seguit de punts claus. El primer de tots és l'absència de contacte, ja que es vol mesurar la temperatura del filament recent extrudit, és important que el sensor pugui mesurar la temperatura a distància. Un altre fet rellevant a valorar és el temps de resposta. Degut a la rapidesa en què es refreda el filament és necessari un sensor que compti amb un temps de resposta molt ràpid. I finalment el sensor ha de ser capaç de captar la temperatura tenint en compte que l'objecte a mesurar estarà en constant moviment.

De tots els sensors de temperatura que hi ha al mercat els que compleixen tots aquests punts són els sensors de temperatura per infrarojos. Un dels grans desavantatges dels sensors infrarojos és que a diferència dels sensors de contacte com els termistors o els termoparells, el seu preu és elevat. Per aquest motiu, en aquest projecte, en tractar-se d'un primer contacte amb la sensorització d'una impressora 3D, també s'ha buscat que el sensor no eleves excessivament el pressupost final.

Un cop feta la recerca de sensors que complien amb els requisits anteriors s'han acabat valorant dues opcions, l'MLX90614 i l'MLX90640 de la casa Melexis. Tots dos són sensors compactes, i de baix consum, 2 mA i 23 mA respectivament en el mode de funcionament. També compten amb interfície I²C i es poden alimentar a 3,3 V fet que possibilita la comunicació i alimentació amb el microcontrolador extern.

4.5.1 MLX90614

És el més econòmic dels dos i compta amb set versions diferents. Aquest sensor proporciona una lectura de temperatura puntual, és a dir, només ofereix un mesurament de temperatura per a un punt específic al seu camp de visió. Si en el camp de visió hi ha més d'un objecte a diferents temperatures, fa la mitjana. Per aquest motiu és important estudiar totes les versions i escollir-ne una on en el camp de visió càpiga una àrea suficientment petita per captar únicament la temperatura del filament. Segons l'estudi que s'ha desenvolupat, vegeu Annex C, la versió que més s'adapta a les necessitats del projecte és el xCI que a una distància de 10 cm cobreix una àrea de 0,6 cm².

Pel que fa a les característiques del sensor, aquest compta amb un ampli rang de temperatura d'operació de -40 °C a 125 °C i de -70 °C a 380 °C per la temperatura de l'objecte a mesurar. També disposa d'una gran resolució de 0,02 °C i un bon temps de resposta de 100 ms.

Un fet a tenir en compte d'aquest sensor, és que la seva precisió varia segons la temperatura ambient i la temperatura de l'objecte a mesurar. Per tant, si suposem que la temperatura del laboratori és de 25 °C la precisió del sensor serà de $\pm 0,5$ °C per a temperatures de fins a 60 °C i arribarà a ± 4 °C per temperatures superiors a 240 °C.

Per tant, tenim que l'MXL90614 és més econòmic i obté un sol valor de temperatura. Això pot ser un avantatge, ja que en obtenir tan sols un valor facilita el treball de processament de la informació i no serà necessària molta memòria RAM per part del microcontrolador. Per contra, en capturar només la temperatura d'un sol punt la col·locació del sensor s'haurà de fer amb molta precisió. Tot i això, hi ha la possibilitat que dins del camp de visió del sensor hi entrin valors no desitjats que puguin alterar el valor de la temperatura del filament.

4.5.2 MLX90640 32x24

Pel que fa al segon sensor, l'MLX90640 32x24, el seu funcionament és completament diferent. Aquest sensor proporciona una imatge tèrmica completa de la matriu de píxels, permetent obtenir un mapa tèrmic detallat. Divideix el seu camp de visió en 768 píxels (32x24) i a cada un li associa una temperatura individual. Posteriorment, guarda aquesta informació en una matriu i l'envia al microcontrolador per ser processada. En estar creant constantment matrius de 32x24 es necessita un microcontrolador amb una certa capacitat de

memòria RAM, mínim 20 kB. Per això un Arduino Mega és insuficient, ja que tan sols compta amb 8 kB de memòria RAM. Per aquest motiu, per tal d'utilitzar aquest sensor és necessari un microcontrolador amb una memòria RAM superior a 20 kB.

El sensor pot treballar en ambients on la temperatura estigui compresa entre $-40\text{ }^{\circ}\text{C}$ i $85\text{ }^{\circ}\text{C}$. Pot mesurar temperatures d'objectes entre $-40\text{ }^{\circ}\text{C}$ i $300\text{ }^{\circ}\text{C}$ mantenint alts nivells de precisió de $\pm 1\text{ }^{\circ}\text{C}$ a tota la seva escala de mesurament. Amb una velocitat de fotogrames de 16 Hz, que pot arribar a 64 Hz segons el fabricant.

Amb aquest sensor s'obté una imatge més àmplia i, per tant, pot captar la temperatura de més d'un punt del procés de manera exacta. D'aquesta manera, es podria observar més d'un punt i acabar seleccionant el que més ens interessa per estudiar sense la necessitat d'apuntar amb gran exactitud el sensor en el punt desitjat. El punt negatiu d'utilitzar l'MXL90640 32x24 és que necessita un microcontrolador dedicat amb una memòria RAM superior a 20 kB, fet que encareix més el projecte.

Un cop estudiades les dues opcions s'ha decidit optar per l'MLX90640. Aquest sensor proporciona un mapa tèrmic, una opció molt interessant que permetrà no només estudiar un punt exacte sinó també altres parts que intervenen en el procés com la temperatura de la peça, del capçal i la de la zona d'impressió. Permetent així, trobar relacions entre temperatures i parts del procés. En obtenir una imatge general serà més fàcil estudiar el comportament del filament un cop extrudit i extreure'n conclusions. El fet d'incloure una càmera d'infrarojos també facilitarà la implementació de millores posteriors en el projecte. Com pot ser un mecanisme de retroalimentació sensorial en un control tancat per processos de fabricació additiva.

4.6 Microcontroladors

Finalment, es necessita un element que reculli totes les dades dels sensors i les envii a l'ordinador. Per tal de desenvolupar aquesta funció es farà ús de microcontroladors, en aquest cas s'ha escollit treballar amb plaques de desenvolupament que integren els microcontroladors per tal d'accedir a les seves entrades i sortides de manera més còmoda.

La idea inicial del projecte era implementar la solució en una sola placa, però finalment s'ha acabat executant amb dos microcontroladors. Després de realitzar diverses proves s'ha

arribat a la conclusió de la necessitat d'inclusió d'un segon microcontrolador per evitar la saturació del bus sèrie del dispositiu. Aquesta saturació ve donada en gran part pel fet que el sensor d'infrarojos envia una matriu de 768 valors per cada lectura. Això provoca una gran demanda de memòria RAM del microcontrolador, per aquest motiu es necessita un aparell amb gran capacitat de memòria. Tot i tenir un dispositiu prou potent, en el moment que coincideixen diverses lectures de diferents sensors al mateix moment la matriu de valors satura el bus de comunicacions i impedeix l'enviament de dades.

A continuació es presenten els dos microcontroladors que s'han escollit finalment per a la resolució del projecte.

4.6.1 Arduino DUE

Les plaques de desenvolupament amb què es treballa des d'EURECAT per fer prototipatge i sistemes DAQ econòmics són les de la casa Arduino. Per aquest motiu, inicialment s'ha escollit implementar el sistema de sensorització amb l'Arduino DUE un dels Arduinos més potents gràcies al xip Atmel SAM3X8E ARM Cortex-M3 i als seus 512 kB de memòria RAM. S'ha escollit aquest microcontrolador per intentar integrar tot el sistema en una sola placa. Però després de diverses proves s'ha conclòs que no era possible i que caldria la inclusió d'una segona placa degut a la saturació del port serial.

Tot i que per la nova funció a desenvolupar, que consisteix a fer la lectura de les acceleracions, temperatura i humitat ambiental, no es necessita una placa tan potent, com ja s'havia adquirit s'ha aprofitat l'Arduino Due. Però cal remarcar, que aquesta tasca la podria resoldre una placa més senzilla i econòmica com l'Arduino UNO. En aquest cas les característiques importants amb les quals ha de comptar el microcontrolador escollit són senzilles. Un mínim de quatre GPIO, una sortida de 3,3 V i una interfície de comunicació I²C.

4.6.2 Adafruit ESP32-Feather

Com a segona placa de desenvolupament s'ha escollit l'Adafruit ESP32-Feather, per tal de desenvolupar les funcions de lectura de dades de la càmera tèrmica. S'ha optat per aquesta placa perquè compte amb una memòria RAM elevada, 520 kB, necessària per treballar amb MLX90640 32x24. Malgrat el seu preu inferior, continua sent una placa molt completa la qual permet adaptar-se a diversos llenguatges de programació, amb un baix consum 500 mA i

que disposa d'un seguit d'entrades i sortides analògiques i digitals suficients per a la funció que desenvolupa en aquest projecte. La placa també disposa de diferents interfícies de comunicació que permeten protocols com SPI, I²C, I2S i UART. Per aquest projecte s'utilitzarà el protocol I²C, ja que és el que utilitza el sensor escollit.

Cal esmentar que les dues plaques compten amb pins de 3,3 i 5 V de sortida que permeten alimentar els sensors sense necessitat d'incloure cap regulador extern. En aquest projecte tant l'Arduino com l'ESP32 s'alimentaran a 5 V a través del port micro USB.

5. PROGRAMACIÓ

Per la realització d'aquest projecte s'han dissenyat dos programes d'adquisició de dades, una interfície gràfica i un programa per generar les imatges de la càmera. Els primers dos programes s'executaran en els microcontroladors externs i seran els encarregats de llegir les dades dels sensors i enviar-les per port serial. Per tal de recollir tota la informació del sistema, s'ha desenvolupat una aplicació en Visual Studio que farà d'interfície i programa de tractament de dades. Serà l'encarregada de rebre les cadenes dels microcontroladors, fer el tractament corresponent i mostrar el resultat per pantalla de forma entenedora per l'usuari. Addicionalment, aquest programa també permetrà la comunicació amb la impressora 3D. D'aquesta manera, es podrà carregar un fitxer G-Code i executar-lo a través de la mateixa aplicació. Permetent així, veure en tot moment on es troba el procés d'impressió i adquirir les dades de temperatura del llit i l'extrusor durant el procés. Finalment, l'últim programa servirà per convertir la matriu de dades que s'obté de la càmera en imatges interpretables en escala de grisos.

A l'hora de dissenyar els codis, s'ha buscat aconseguir programes organitzats, senzills, entenedors i amb una estructura lineal per tal d'evitar errors en la programació. Els dos programes estan basats en funcions creades específicament per aquest projecte. El fet d'utilitzar funcions ajuda a assolir un codi més ben organitzat compacte i robust envers possibles errors.

5.1 Programa d'adquisició d'acceleracions i paràmetres ambientals

El primer programa serà el que es carregarà a l'Arduino DUE. Aquest serà l'encarregat de fer la lectura dels dos acceleròmetres i del sensor de temperatura i humitat ambiental sota demanda.

L'estructura que s'ha seguit per crear el programa és senzilla, es basa en un bucle que s'encarrega de llegir les instruccions que li arriben per port serial. Depenent de la instrucció es farà la lectura d'uns sensors o d'altres.

El codi comença configurant els pins que utilitzaran els sensors, i inicialitzant els protocols de comunicació, I²C i port sèrie. Un cop comprovat que la connexió amb tots els sensors és correcta, es llegeix el port serial. En aquest programa poden arribar dues instruccions, la

demanda de lectura ambiental (amb;) i la demanda de lectura de les acceleracions (acc;). Cal esmentar que al llarg de la programació s'ha escollit el caràcter ';' com a caràcter de finalització, per tal de marcar on acaba la informació desitjada. En llegir una de les dues instruccions, es farà una lectura instantània del valor del sensor i es realitzarà una cadena en format JSON amb els valors de la lectura per enviar a l'ordinador a través del port serial. Vegeu Annex B per explicació format JSON.

Un cop escrita la cadena, s'esperarà nova comanda, si es llegeix una nova comanda, es tornarà a fer el procés de serialització i enviament de dades. En canvi, si no arriba cap comanda el programa s'esperarà a rebre la següent o bé que es tanqui la connexió amb el microcontrolador, si és així, el programa finalitzarà fins a tornar a obrir connexió.

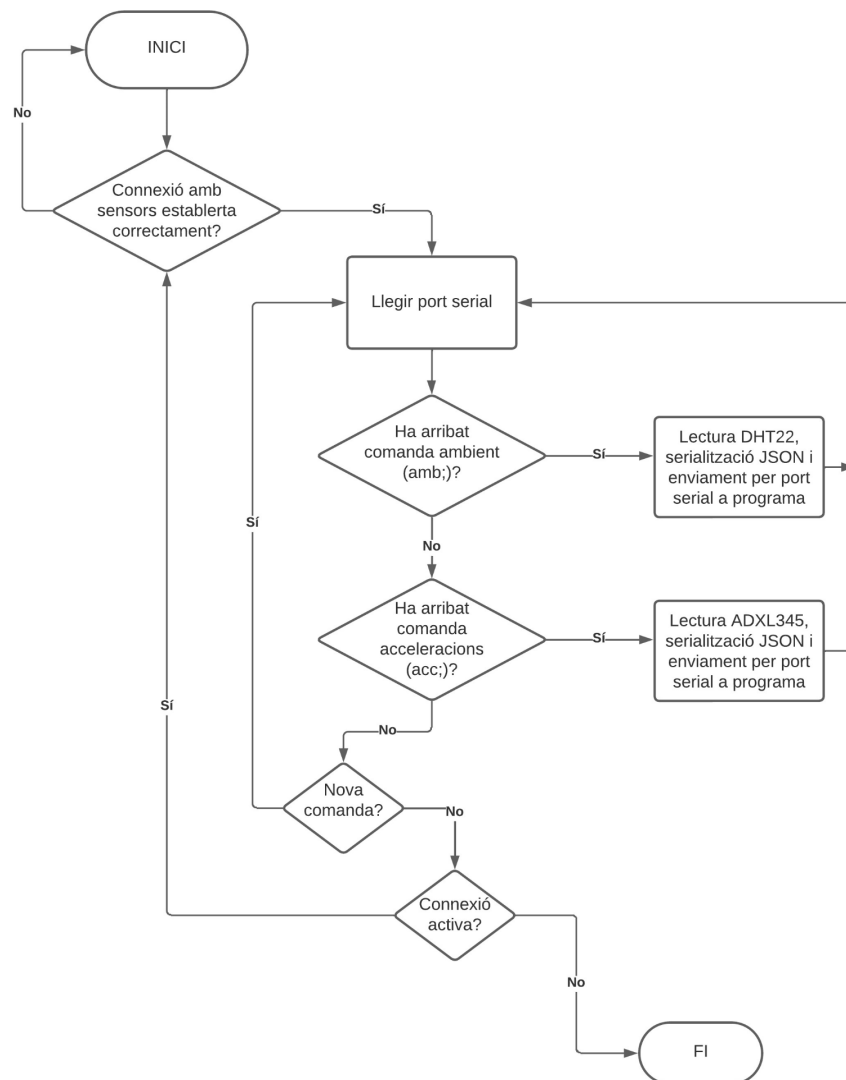


Figura 7: Diagrama de flux programa d'adquisició de dades.

En aquest diagrama de flux (Figura 7) s'exposen totes les possibilitats que es poden experimentar durant l'execució del codi d'adquisició de dades per tal de tenir una idea més clara a l'hora de dissenyar i entendre el codi.

Per tal de dur a terme el codi de la configuració de l'integrat RTC DS3231 s'han utilitzat les següents entrades i sortides (Taula 3).

GPIO Arduino	Variable	D/A	E/S	Component
20	SDA	D	E/S	ADXL345
21	SCL	D	E	ADXL345
2	DHTPIN	D	E	DHT22

Taula 3: Taula connexió de pins Arduino DUE.

5.2 Programa càmera

El segon programa serà el que es carregarà a la placa de desenvolupament ESP32 Feather i llegirà sota demanda la lectura de la càmera tèrmica. L'estructura en aquest segon codi és molt similar al primer amb la diferència que en aquest només es donarà per vàlida una sola instrucció la de lectura de la càmera.

Com en el primer codi, el programa inicia configurant els pins utilitzats i inicialitzant els protocols I²C i serial que utilitzarà la placa de desenvolupament ESP32 per comunicar-se amb l'MLX90640 32x24. Igual que en el codi anterior, un cop validada la connexió amb els sensors es començarà a llegir el port serial. En aquest cas només es llegirà una instrucció, la demanda de lectura de la càmera tèrmica (cam;). En llegir aquesta instrucció es farà la lectura corresponent del sensor i també se serialitzarà el resultat en una cadena JSON.

En aquest programa el resultat que s'obtindrà serà una cadena amb 768 valors (32x24), que s'enviarà per port serial. Igual que el codi anterior, un cop feta la lectura el programa esperarà la següent instrucció fins a rebre una de nova o fins a perdre la connexió amb l'ESP32.

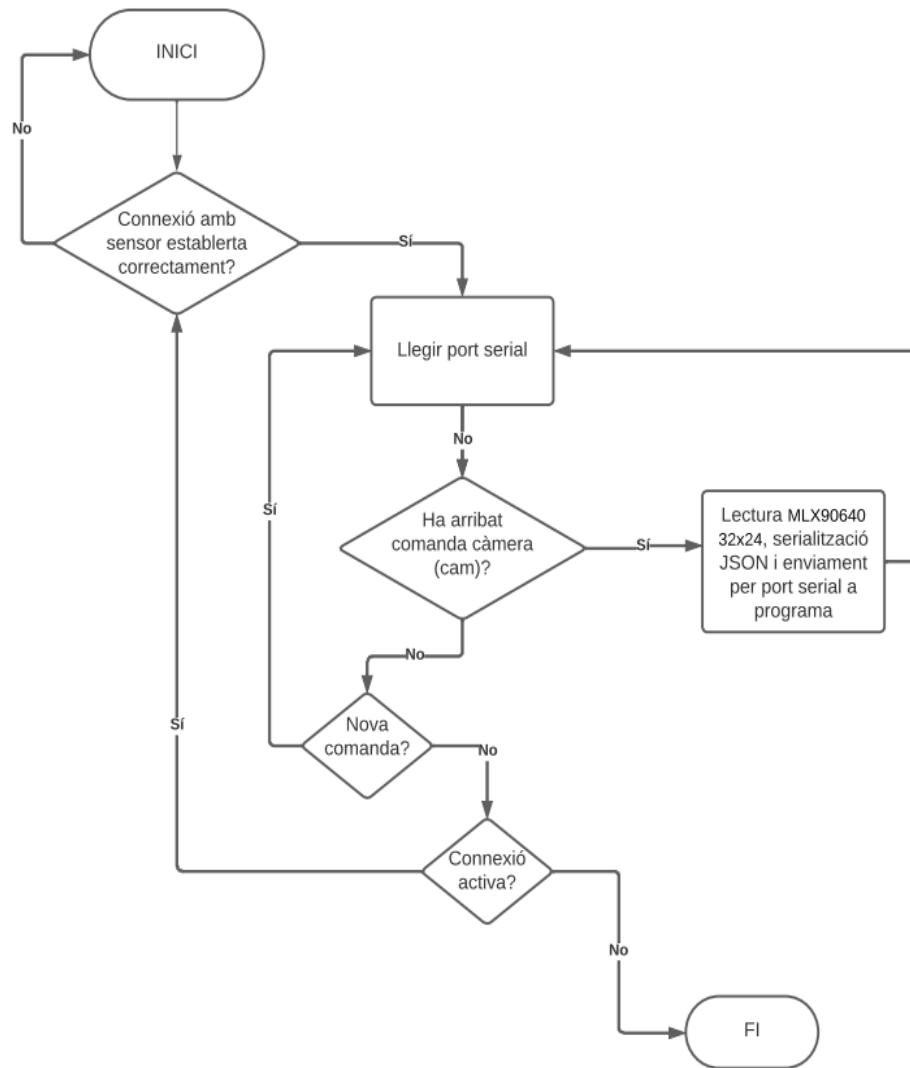


Figura 8: Diagrama de flux programa càmera.

En el diagrama de flux (Figura 8), es mostra l'estructura que segueix el programa que executarà l'ESP32. També s'exposen totes les possibilitats que es poden experimentar durant l'execució del codi. Per tal de dur a terme el codi de la càmera s'han utilitzat diferents pins GPIO de la placa de desenvolupament (Taula 4), per tal que aquesta pugui comunicar-se amb l'integrat i llegir les entrades del sensor.

GPIO ESP32	Variable	D/A	E/S	Component
22	SCL	D	E	MLX90640 32x24
23	SDA	D	E/S	MLX90640 32x24

Taula 4: Taula connexió de pins ESP32.

5.3 Programa interfície de monitoratge

Per tal de recollir i agrupar les dades de tres elements diferents (Arduino, ESP32 i impressora) i representar-les s'ha vist en la necessitat de crear una aplicació per ordinador que compti amb una interfície gràfica. A través d'ella es permetrà controlar en tot moment tant les accions de la impressora com les dels sensors. L'objectiu és crear una interfície senzilla, amable i còmode amb l'usuari final, que permeti guardar tota la informació de manera ràpida per poder-la estudiar posteriorment. Per aquest motiu, a part de mostrar les lectures dels sensors, el programa també permetrà iniciar i controlar el procés d'impressió des de la mateixa pantalla. Tot i que aquest fet comporta elevar la dificultat de programació, permetrà obtenir un programa més complet. Ja que també es podrà observar en tot moment la comunicació entre la impressora i l'ordinador i detectar així possibles errors en el codi d'impressió.

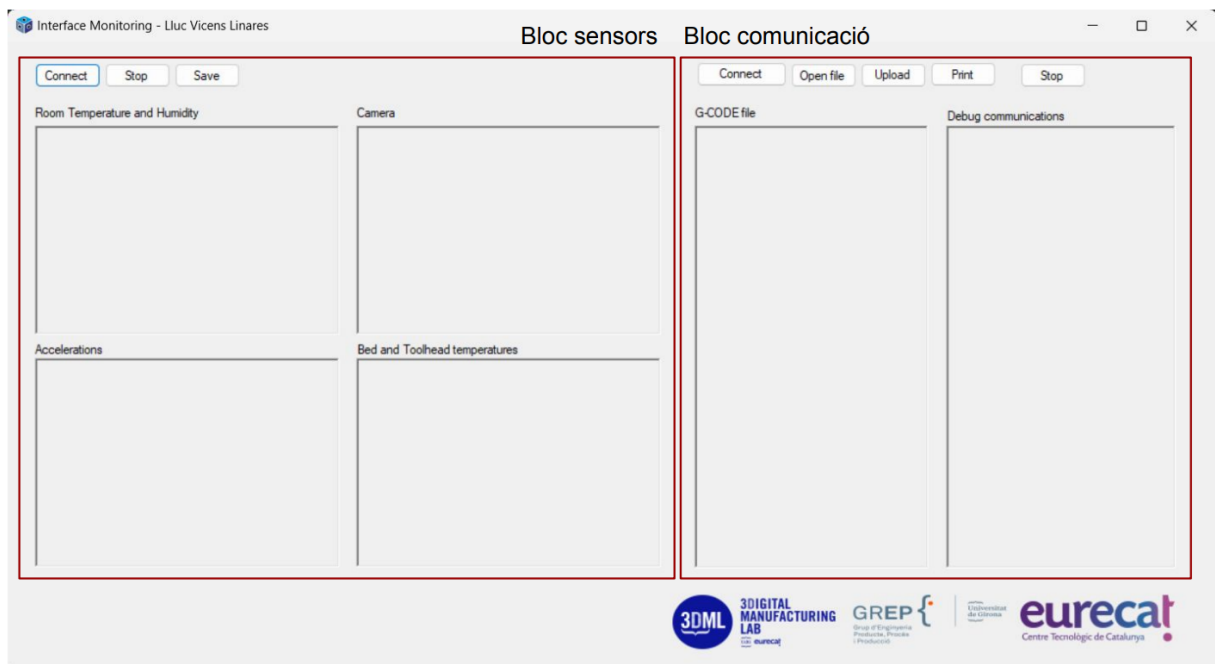


Figura 9: Interfície de monitoratge.

Tal com es pot veure en la Figura 9, la interfície comptarà amb dos blocs, el bloc de dades dels sensors i el bloc de comunicació amb la impressora. En el primer bloc apareixen quatre finestres on es mostrarà la informació captada de temperatura i humitat ambiental, de la càmera tèrmica, de les acceleracions en els tres eixos de moviment i de la temperatura del llit i extrusor.

Per tal d'aconseguir les lectures dels sensors, el programa comptarà amb tres temporitzadors independents, un per a cada instrucció (amb; acc; i cam;). Aquests seran els encarregats d'escriure, cada interval de temps marcat, la instrucció assignada per port serial. Perquè els microcontroladors la llegeixin i retornin una cadena amb els valors que s'escriuran per pantalla juntament amb la data i hora de l'adquisició dels valors.

Com ja s'ha comentat, per tal d'enviar les dades dels sensors a l'ordinador s'ha optat per fer-ho en format JSON. Per tant, un cop arriba la cadena a l'ordinador primer cal desserialitzar-la per obtenir els valors, en aquest cas s'ha fet ús de la biblioteca `Newtonsoft.Json` per `C#`. Un cop desserialitzada la cadena, abans d'escriure el valor per pantalla s'ha optat per separar els valors pel caràcter '@'. D'aquesta manera es podran passar les dades guardades en un fitxer `.txt` a Excel de forma senzilla aprofitant l'arrova com a caràcter delimitador de columnes. Aquest fet permetrà crear gràfics de les dades aconseguides del procés d'impressió per estudiar el comportament de la impressora sensoritzada.

Seguint en el bloc dels sensors, aquest comptarà amb tres botons que permetran tres accions diferents. El primer botó, *Connect*, s'encarregarà d'obrir la comunicació amb els dos microcontroladors que adquiriran les dades dels sensors. D'aquesta manera es permetrà iniciar la lectura dels sensors encara que el procés d'impressió no hagi començat. Per poder capturar valors inicials i comparar-los. El següent botó, *Stop*, parará la lectura dels sensors, però no atura la impressió. Si es vol tornar a llegir el valor dels sensors, es podrà prémer altre cop el botó de connectar que s'haurà tornat a habilitar per continuar adquirint dades dels sensors. I, finalment el botó de *Save*, un cop es premi aquest botó, apareixerà una finestreta on demanarà la ubicació per guardar el fitxer de text. Aquests fitxers de text, contindran totes les dades recollides en les quatre pantalles juntament amb la data i hora de l'adquisició de cada una. Aquest procés es repetirà quatre vegades una per a cada finestra.

Entrant en el segon bloc, el de comunicació amb la impressora, es troben dues grans finestres que permetran controlar en tot moment l'estat d'execució del codi i les comunicacions entre l'ordinador i la impressora.

En el primer requadre es carregarà el fitxer G-Code. Aquest fitxer conté totes les comandes que la impressora haurà d'executar per formar la peça desitjada. I en la segona finestra apareixerà la depuració de la comunicació entre el programa i la impressora. És a dir, primer s'escriurà la línia de codi enviada i posteriorment la resposta de la impressora.

El programa, comptarà amb un temporitzador que s'encarregarà cada mig segon de llegir una instrucció del fitxer G-Code carregat i escriure-la a la impressora, aquesta l'executarà i donarà una de les tres respostes possibles. Si retorna 'ok', significa que la comanda s'ha executat correctament i es passarà a la següent. Si es rep 'busy', voldrà dir que la impressora està ocupada executant l'acció anterior i que cal esperar. En aquest cas s'aturarà el temporitzador fins a rebre un 'ok' per enviar la següent comanda. Finalment, també pot respondre amb la temperatura del llit i de l'extrusor.

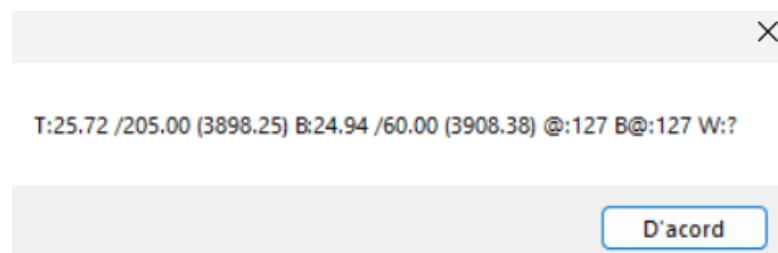


Figura 10: Exemple de resposta a comanda de temperatura llit i extrusor.

En aquest cas interessa guardar aquests valors per estudiar-los. Per això, cada vegada que es llegeix un '@' en la comunicació entre impressora i programa (Figura 10), es guardarà tota la cadena per obtenir el valor i consigna tant del llit com de l'extrusor. Posteriorment, es tractarà per aconseguir el mateix format explicat anteriorment que permet passar el fitxer .txt a Excel.

Pel que fa als botons en aquest bloc es comptarà amb cinc. El seu funcionament és senzill, amb el primer, *Connect Printer*, igual que amb el dels sensors s'establirà connexió amb el port serial de la impressora. Amb el segon botó, *Open*, es permetrà buscar un fitxer G-Code a l'ordinador i obrir-lo. Clicant el següent botó, *Upload File*, aquest fitxer es carregarà al programa, aquest procés tardarà una estona, ja que el programa depurarà el codi de comentaris i línies buides per evitar errors en la comunicació. Per acabar, el botó *Print*, permetrà un cop es tingui el fitxer carregat al programa iniciar el procés d'impressió.

Finalment, s'ha contemplat la inclusió d'un botó que permeti aturar la impressió en qualsevol moment. Per fer-ho s'haurà d'interrompre el temporitzador que s'encarrega de l'enviament de comandes a la impressora, i enviar comandes de G-Code de clausura com: apagar el ventilador, reiniciar les temperatures de llit i extrusor a zero, i retornar el capçal a la posició inicial.

5.4 Programa tractament d'imatge

Per tal de transformar la matriu de valors que s'obté de la càmera tèrmica en una imatge que es pugui interpretar, s'ha desenvolupat un programa independent que s'executarà al mateix ordinador. Aquest s'encarregarà de passar totes les matrius de 32x24 guardades en el fitxer .txt, generat per la interfície, en una imatge en escala de grisos. Associant a cada valor de temperatura una tonalitat diferent depenent de la seva temperatura. Com més calent, més clar serà el píxel i com més fred s'aproximarà més al negre.

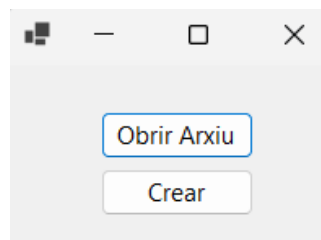


Figura 11: Interfície programa tractament d'imatge.

Per aquest programa també s'ha creat una interfície gràfica (Figura 11), perquè l'usuari pugui pujar els fitxers i crear les imatges. Bàsicament, està composta per dos botons, el d'obrir arxiu i el de crear la imatge. El primer botó obrirà un menú on es podrà seleccionar el fitxer .txt que es vol convertir i la carpeta on es vol que es generin les imatges. Un cop fet això, ja es podrà seleccionar el segon botó per tal que apareguin totes les imatges a la direcció indicada.

Per tal transformar la matriu el primer que farà el programa serà desserialitzar la cadena JSON. S'indicarà per programa la temperatura màxima i mínima obtinguda, i es generaran intervals de valors on se'ls associarà un color. De més fosc pels intervals de valors més petits a més clars per les temperatures més altes. Finalment, es relacionarà cada valor de la matriu amb un dels intervals i es generarà la imatge a escala de grisos.

6. MUNTATGE I VALIDACIÓ

En aquest apartat s'explicarà el muntatge i les proves que s'han dut a terme per validar el sistema de monitoratge del present projecte. Aquest procés consisteix a cablejar tots els sensors, col·locar-los al lloc desitjat, comprovar el seu bon funcionament i finalment estudiar si els valors que s'obtenen són coherents amb la realitat del procés.

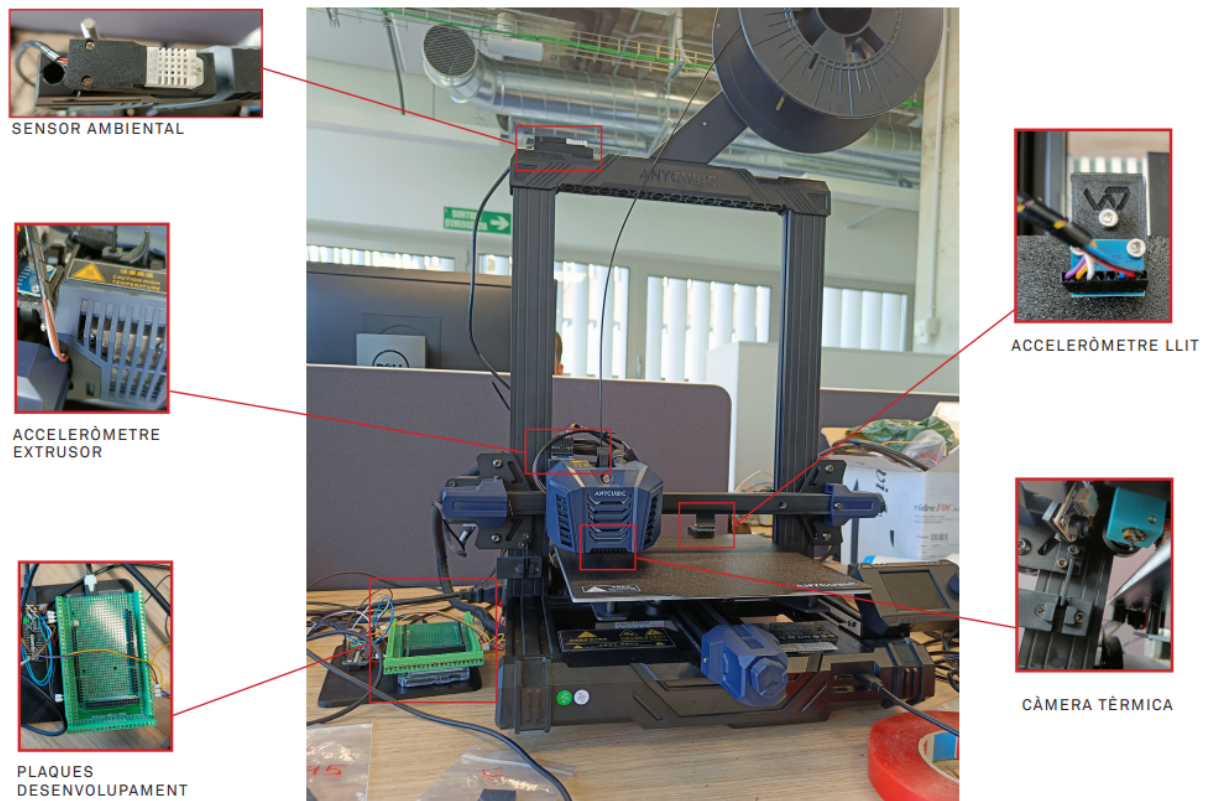


Figura 12: Disposició dels elements del sistema de monitoratge.

En la figura anterior (Figura 12) es pot observar la disposició de tots els sensors i microcontroladors en el muntatge. Tal com s'indica, el sensor ambiental DHT 22 es col·locarà a una part elevada de la impressora, com és la barra transversal superior. Se situarà dins d'un suport específic per aquest tipus de sensors que permet fixar-lo en la posició desitjada per tal d'evitar que l'escalfor del llit o de l'extrusor puguin influir negativament en la mesura dels paràmetres ambientals. Per tal de validar el funcionament d'aquest sensor s'han pres valors durant deu minuts i posteriorment s'han graficat per comprovar-ne els resultats (Figura 13).

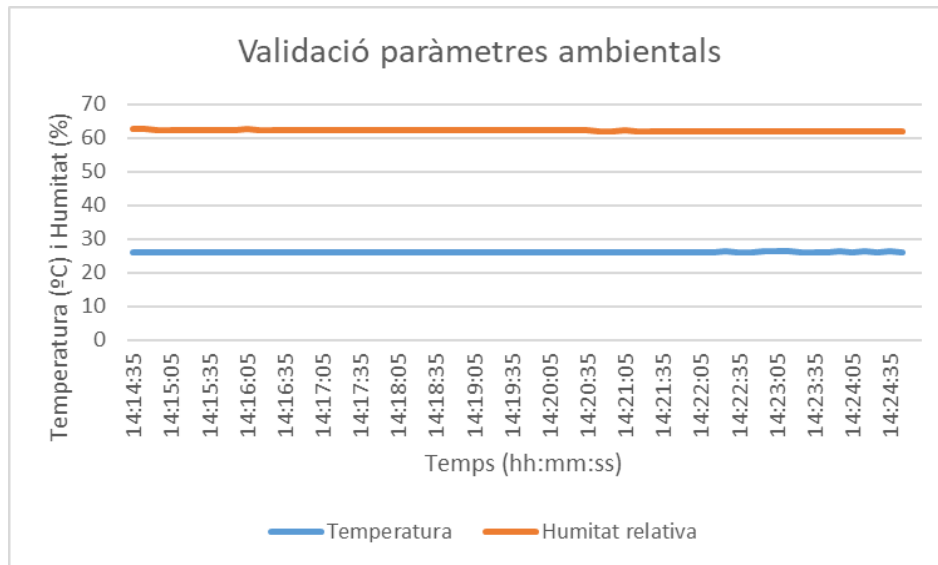


Figura 13: Resultat de la validació del sensor DHT22.

Com es pot comprovar tant per la temperatura com per la humitat s'observen valors precisos i coherents. No es detecten canvis bruscs en els valors recollits, fet que podria indicar un mal funcionament del sensor o la intervenció de la temperatura de l'extrusor en les mesures.

Pel que fa als dos acceleròmetres s'han fixat en dos punts diferents del muntatge per tal de captar les acceleracions en els tres eixos de l'espai. Com en l'Anycubic Kobra Neo el llit és l'encarregat de fer el moviment en l'eix Y, s'ha col·locat un ADXL345 sobre un suport fixat a la part posterior del llit calent. A través d'aquest suport s'aconsegueix mantenir el sensor en una posició fixa i es crea una separació per tal d'evitar que el sensor se sobreescalfi amb les temperatures del llit, ja que a vegades pot arribar a superar els 80°C.

El segon acceleròmetre serà l'encarregat de llegir les acceleracions en X i Z. Per això s'ha situat en un suport darrere del capçal extrusor, perquè és on captarà millor les vibracions i a més, estarà a una distància segura de qualsevol font de calor.

Per tal de validar els sensors, s'han connectat per programa i, posteriorment a través de comandes G-Code s'han provocat moviments en els tres eixos per tal de distingir-los. La impressora permet fer moviments més bruscos en els eixos X i Y, això provoca que les acceleracions produïdes en aquests eixos es representin de forma més clara que en el Z. Això és degut al fet que l'eix Z, es mou a través d'una vareta roscada mentre els altres dos ho fan per una corretja de distribució.

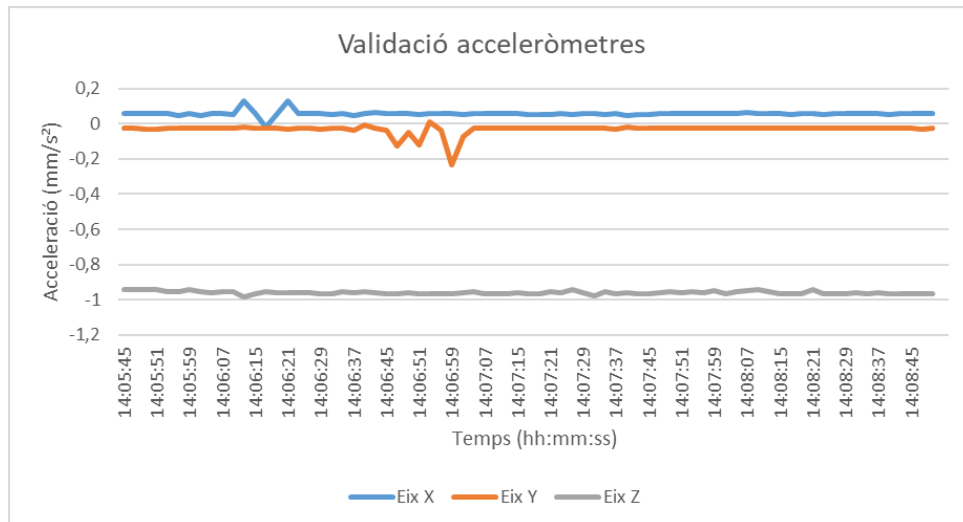


Figura 14: Resultat validació dels acceleròmetres.

A través de la Figura 14, es pot observar el correcte funcionament dels acceleròmetres. Es veu clarament els moments en què s'accelera en X i Y, i les direccions en què ho fan. En canvi, el moment que s'accelera en Z queda molt més dissimulat a causa dels factors comentats anteriorment.

En el cas dels termistors del llit i l'extrusor, aquests ja venen col·locats amb la impressora adquirida. Per aquest motiu, només s'ha de fer la validació de la captura de dades a través del programa. Per fer-ho, s'ha iniciat la comunicació amb la impressora i s'ha executat el programa. D'aquesta manera s'han pogut recollir les dades que apareixen per la pantalla de la impressora i graficar-les.

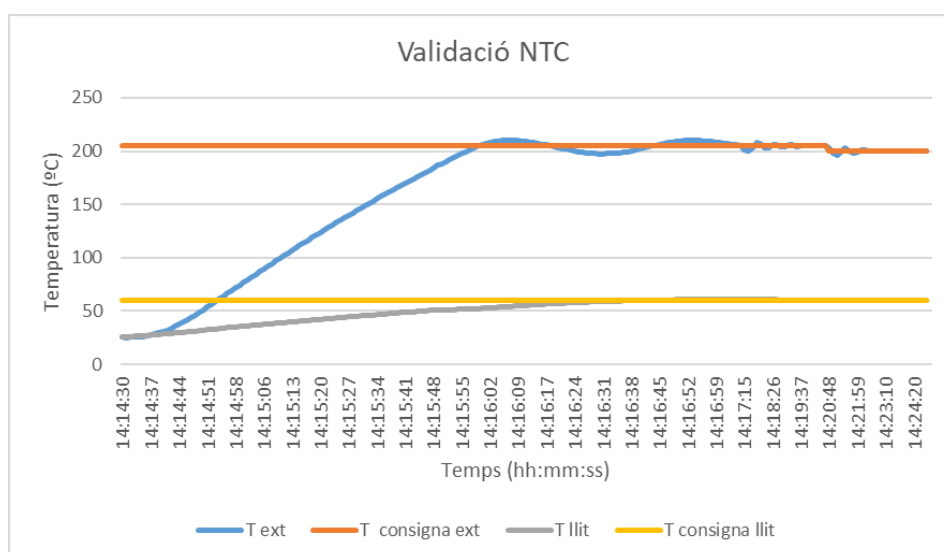


Figura 15: Validació del funcionament dels termistors NTC.

Un cop obtinguts els resultats (Figura 15), s'ha vist que les temperatures representades en el gràfic fet amb les dades aconseguides per programa, i les temperatures observades durant el procés a través de la pantalla LCD de la impressora coincideixen.

Finalment, s'ha col·locat la càmera tèrmica. Per tal de capturar imatges del mateix punt, s'ha dissenyat un suport que permet situar la càmera en la inclinació desitjada per aconseguir veure tots els elements dels quals interessa captar-ne la temperatura, com la punta de l'extrusor i el filament acabat d'extrudir. En aquest cas, s'ha de tenir en compte que en situar la càmera molt pròxima al capçal, el cablejat de les connexions ha de passar per l'exterior d'aquest per tal d'evitar que es malmeti. L'aïllament d'aquests cables és de PVC, material que compta amb una temperatura de fusió entre 170 i 200 °C. Aquests valors de temperatura en condicions normals de producció es donen dins del capçal d'extrusió, i provocarien la fusió dels cables de la càmera i, per tant, un mal funcionament.

Per tal de validar la situació i el funcionament de la càmera, s'han captat imatges amb l'extrusor en funcionament a 205 °C (Figura 16). Posteriorment, amb el programa de tractament d'imatge, s'han transformat les matrius de valor en imatges de 32x24 en escala de grisos.

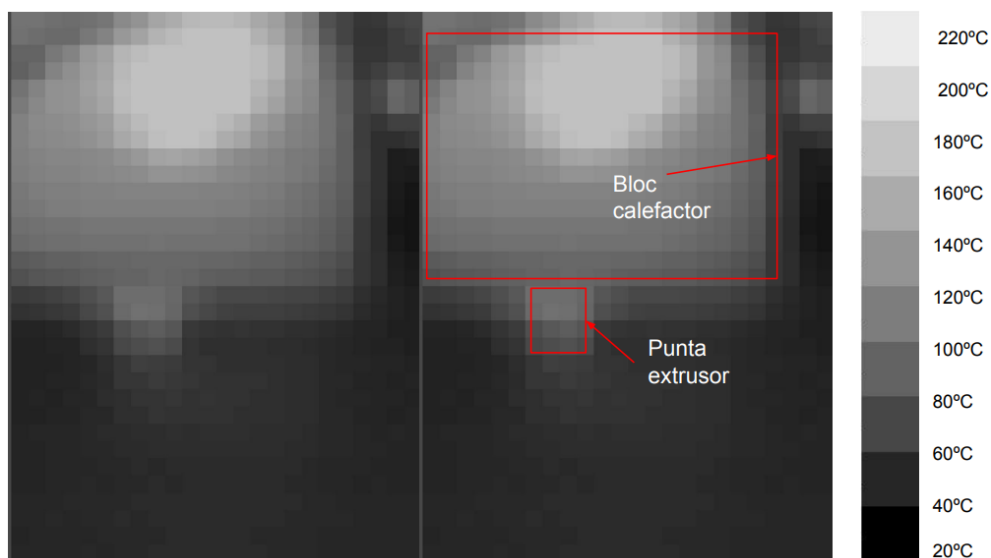


Figura 16: Imatge capturada amb càmera tèrmica.

En treballar amb escala de grisos cal tenir present com es representen els diferents valors de temperatura. Els colors més clars apareixen en les zones amb més temperatura i els més foscos corresponen a les zones més fredes. Com es pot observar en la imatge presa durant la validació del sistema, a la part superior de la imatge es distingeix en un gris clar el bloc

calefactor juntament amb la punta metàl·lica de l'extrusor, aquests es troben a valors pròxims als 205 °C. Per contra, l'espai que queda entre el llit i l'extrusor queda representat per píxels pràcticament negres, ja que es troben a una temperatura pròxima a l'ambiental, 35°C.

Per últim, s'ha creat un suport per fixar-hi les dues plaques de desenvolupament per aconseguir un millor acabat del sistema. Addicionalment, per facilitar el cablejat sobre la placa Arduino, s'ha inclòs un adaptador de borneres, per poder connectar diversos cables a una mateixa entrada. També s'ha fet ús d'un USB Hub, ja que per executar el programa es necessiten tres ports USB i l'ordinador amb el qual s'ha dut a terme el projecte només compta amb dos.

En aquest projecte no s'ha vist la necessitat de dissenyar una placa PCB, perquè es treballa amb kits de sensors on cada un ja compta amb els seus components i resistències de pull-up. En el sistema de monitoratge tots els sensors es col·loquen en posicions diferents, per aquest motiu s'hauria hagut de desenvolupar una PCB per a cada un, fet que hauria alentit i encarit el procés.

7. DISSENY PLA EXPERIMENTAL

Per tal de validar el sistema desenvolupat en diferents casos d'estudi, s'ha dissenyat un seguit de proves. A través d'una peça amb les mateixes dimensions i geometria es vol variar l'estratègia de fabricació per observar canvis en el comportament de la impressora. Per fer-ho, s'ha dissenyat una peça rectangular de 80x20x4 mm, a la qual se li canviaran diferents paràmetres com la velocitat d'impressió, el patró d'emplenament de les capes superior i inferior (Figura 17), i les temperatures del llit i extrusor.

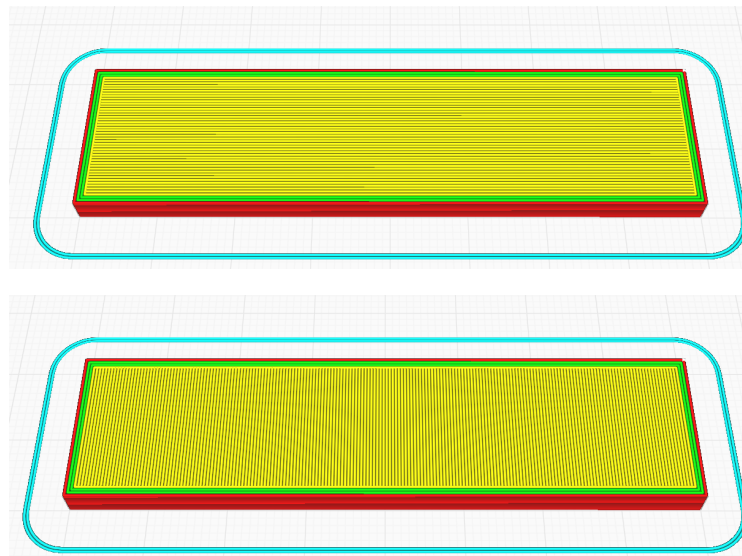


Figura 17: Patrons d'impressió, 90° per la figura superior i 0° per la inferior.

Mitjançant el sistema de monitoratge, es recolliran tots els valors dels sensors durant el procés de fabricació de la peça. Per graficar el resultat i posteriorment fer una anàlisi del funcionament de la impressora. Un cop impreses totes les peces, també es comprovaran les mides i es mesurarà la rugositat superficial per valorar l'acabat final. Totes les peces s'imprimiran amb el mateix material, PLA 1,75 mm de RSO PRO (ref. 832-0214).

En total s'imprimiran sis peces amb diferents combinacions dels paràmetres anteriorment descrits que es volen analitzar, aquestes combinacions venen donades a la Taula 5.

Peça	Patró (°)	Velocitat (mm/s)	T extrusor (°C)	T llit (°C)
A	0	65	200	60
B	0	95	200	60
C	90	65	200	60
D	90	95	200	60

Peça	Patró (°)	Velocitat (mm/s)	T extrusor (°C)	T llit (°C)
E	90	95	220	80
F	90	95	190	50

Taula 5: Paràmetres de les sis peces impreses.

7.1 Mesura de paràmetres del sistema de monitoratge

Mitjançant els resultats adquirits a través del sistema de monitoratge, vegeu Annex A, s'han estudiat els paràmetres observats en els gràfics obtinguts (Figura 18) per extreure explicacions. Respecte a la temperatura i humitat ambiental, s'ha vist que no pateixen cap canvi al llarg del temps i per les proves fetes no es veu cap influència d'aquests paràmetres en el resultat final de la impressió.

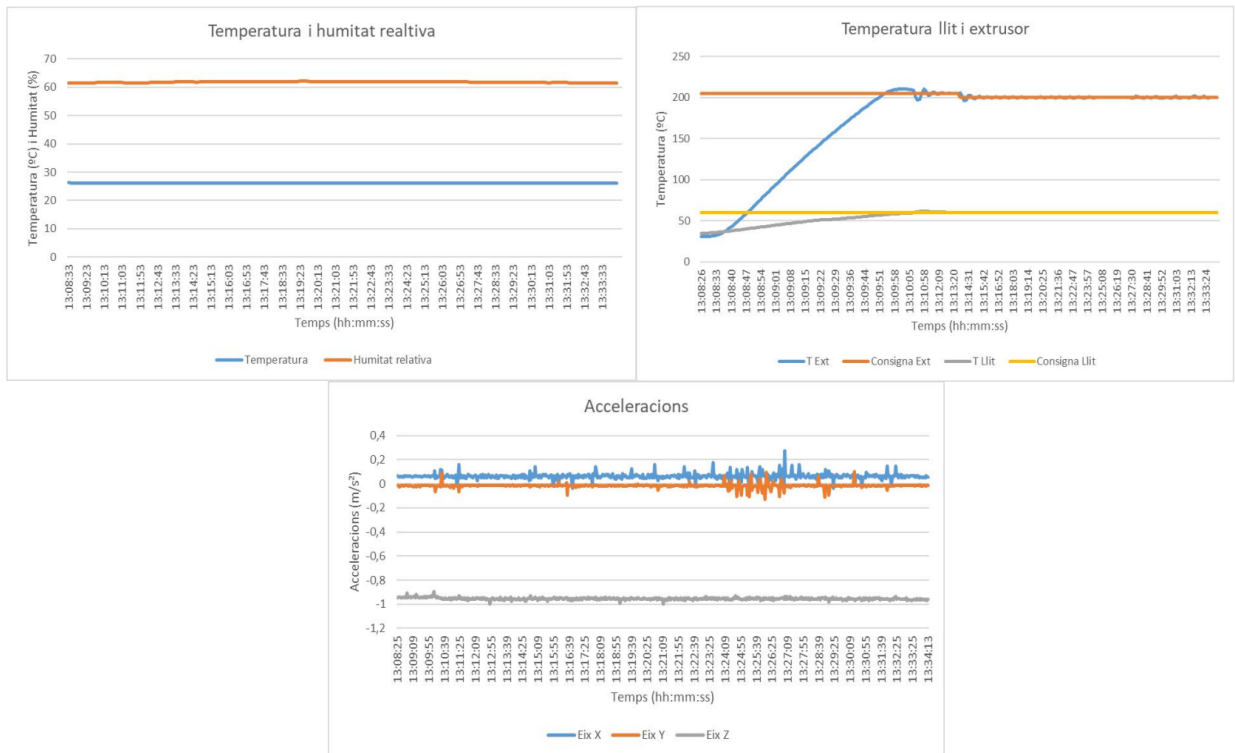


Figura 18: Exemple de gràfics obtinguts amb la informació del sistema de monitoratge peça D.

Pel que fa al control de la temperatura del llit i l'extrusor, a través dels gràfics obtinguts (Figura 18), es pot comprovar el funcionament dels controladors PID de temperatura, tant del llit com de l'extrusor. S'observa com a l'inici de la impressió parteixen de la temperatura ambiental, i augmenten progressivament fins a arribar a la temperatura consigna. Un cop

arribada a aquesta temperatura s'inicia el procés d'impressió, sempre ajustant-se als llindars marcats de $\pm 5^\circ\text{C}$ sobre la temperatura marcada.

En quant a les acceleracions, no s'adverteixen grans diferències entre les peces impreses a 65 mm/s i les impreses a 95 mm/s. Com es mostra en la Figura 19, en augmentar la velocitat, la peça tarda menys temps a imprimir-se i apareixen petites vibracions extremes que comporten un acabat final pitjor.

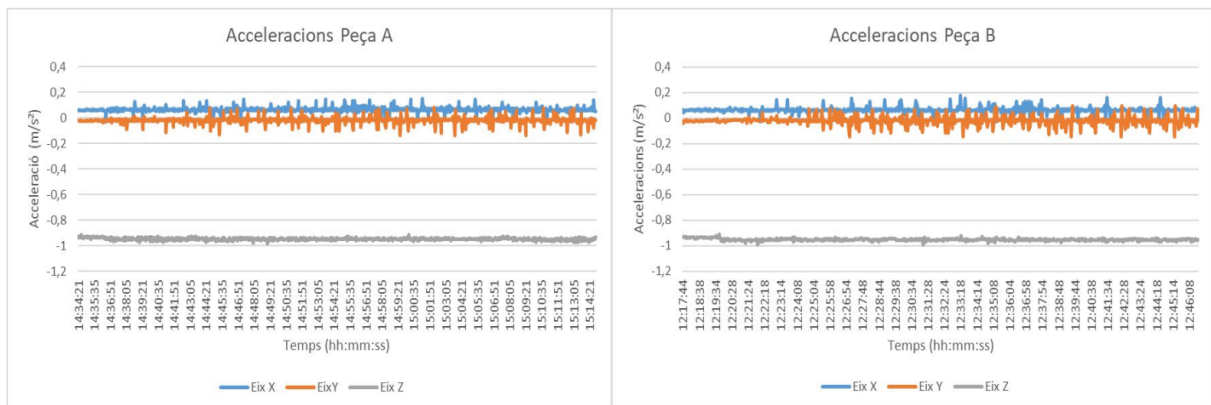


Figura 19: Comparativa peces A i peça B.

On sí que s'observen canvis significatius en les acceleracions, és en la variació dels patrons d'emplenament. S'ha vist que les peces impreses amb 0° tarden més temps a imprimir-se i apareixen més vibracions. El patró de 0° implica fer més canvis de sentit per emplenar les capes, i per tant, l'aparició de més vibracions respecte al patró de 90° (Figura 20). En ser un rectangle s'economitzen més els moviments emplenant amb un patró de 90° , en fer menys moviments també apareixen menys vibracions.

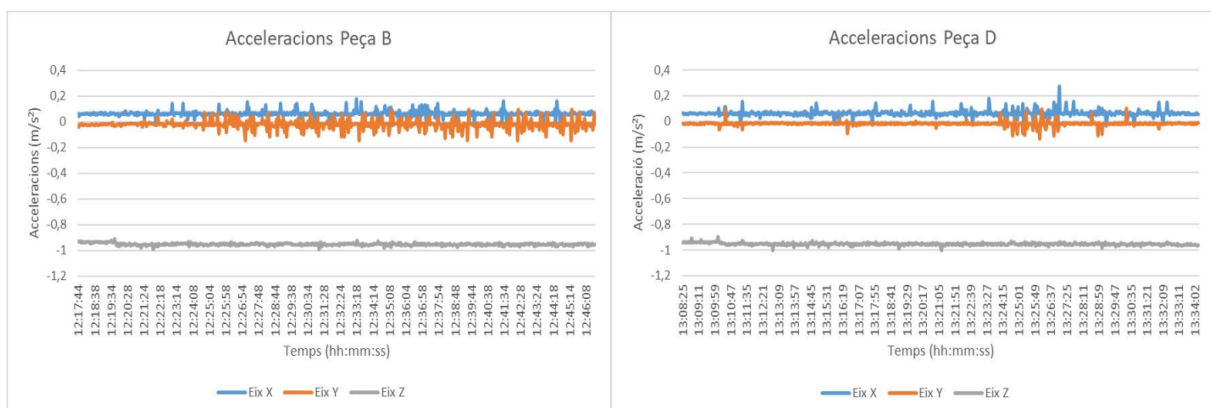


Figura 20: Comparativa peces B i D.

On s'han vist variacions en l'acabat final ha estat en els dos casos extrems, peces E i F. Disminuir la temperatura de l'extrusor a 190 °C i la del llit a 50 °C, peça F, provoca una mala adhesió de la primera capa a la superfície del llit. Això dona peu a l'aparició de vibracions i rugositats a la base de la peça. Per contra, per a la peça E, temperatura llit 80 °C i extrusor 220 °C, s'ha vist una molt bona adhesió tant a la superfície d'impressió com entre capes, comportant un millor acabat en les parets laterals exteriors (Figura 21).

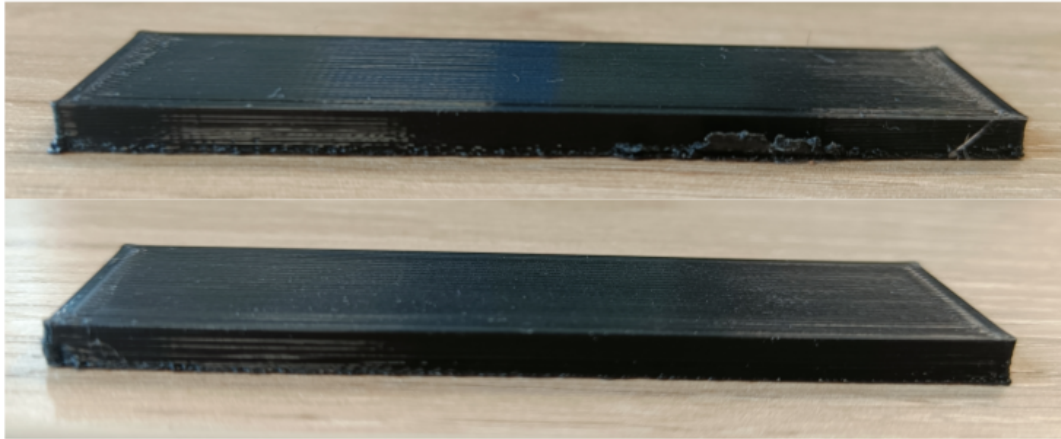


Figura 21: Vibracions en les bases de peça F (superior) i peça E (inferior).

Finalment, comentar que pel que fa a la càmera tèrmica, tot i que tant el muntatge com l'obtenció d'imatges és correcte, el resultat obtingut no és vàlid per fer-ne un estudi. La resolució de la càmera és molt baixa i les imatges aconseguides no permeten diferenciar amb claredat elements com el filament recent extrudit o les diferents capes de la peça (Figura 22). Per aquest motiu no s'han pogut extreure resultats concloents d'aquest apartat.

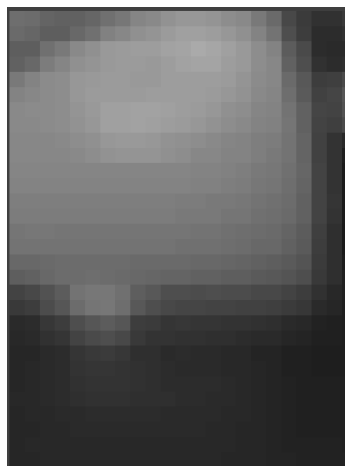


Figura 22: Imatge extrusor en funcionament.

7.2 Mesura de precisió dimensional

A part dels paràmetres obtinguts a través del sistema de monitoratge, també s'han volgut mesurar altres factors per ampliar l'estudi del procés d'impressió. En aquest projecte, s'ha volgut observar les variacions de les dimensions reals de les peces impreses respecte a la dissenyada, 80x20x4mm. Els resultats aconseguits es representen en la Taula 6. Les mesures X, Y i Z s'han obtingut en les zones de la peça més favorables i de més qualitat. Un cop aconseguides, s'han quantificat les diferències respecte al valor objectiu (Di)

Peça	X (mm)	Di	Y (mm)	Di	Z (mm)	Di
A	79,98	0,02	20,03	0,03	4,12	0,12
B	80,12	0,12	20,08	0,08	4,18	0,18
C	80,01	0,01	20,01	0,01	4,06	0,06
D	79,98	0,02	19,94	0,06	4,09	0,09
E	79,94	0,06	20,1	0,10	4,15	0,15
F	79,88	0,12	20,09	0,09	4,03	0,03

Taula 6: Mesures precisió dimensional.

Pel que fa a les dimensions finals de les peces impreses, a través de la taula anterior (Taula 6) es poden observar petites variacions en les dimensions en els eixos X i Y. On divergeixen més és en l'eix Z, fet que podria indicar la necessitat de tornar a calibrar la impressora i ajustar l'offset en Z. Per millorar aquest fet i l'acabat final de les peces impreses, es podria afegir a la impressora un segon motor Z juntament amb una vareta roscada per millorar la precisió en aquest eix.

No s'ha vist un patró concret per poder relacionar les variacions en les dimensions de les diferents peces amb paràmetres com la velocitat d'impressió, la temperatura dels actuadors o les vibracions. Per tal d'aconseguir un patró s'hauria de repetir l'experiment més vegades per aconseguir més mostres i extreure una conclusió més fiable i rigorosa.

7.3 Mesura de rugositat superficial

Per últim, també s'ha volgut mesurar la rugositat superficial de les sis peces impreses per veure la qualitat final dels acabats. Per fer-ho, s'ha utilitzat el rugosímetre de sobretaula Mitutoyo SurfTest SV-2000 del laboratori del GREP (Figura 23).

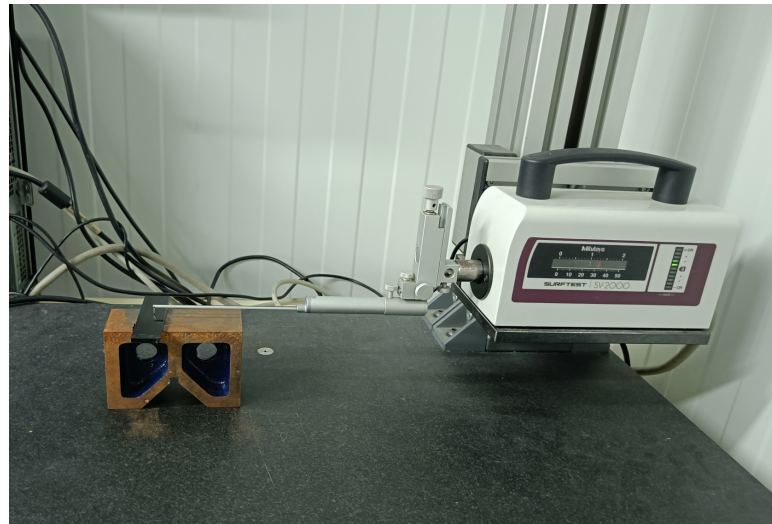


Figura 23: Mesura amb Mitutoyo Surftrac SV-2000.

Tant la calibració com les mesures, s'han fet seguint la norma UNE ISO 4288:1996. Per tal de calibrar la màquina amb la nova agulla introduïda, la Mitutoyo 12AAC741 60°, s'ha utilitzat un bloc patró amb una Ra de 3 µm. Amb el resultat de la calibració i la taula de Cut-off de la UNE ISO 4288:1996 (Taula 7) s'ha determinat que per les peces a mesurar i l'agulla utilitzada el λc a introduir ha de ser de 2,5 mm.

Recommended Cut-off (ISO 4288-1996)				
Periodic Profiles	Non-Periodic Profiles		Cut-off	Sampling Length/Evaluation Length
	Rz (µm)	Ra (µm)	λc (mm)	λc (mm)/L
Spacing Distance RSm (mm)				
>0.013-0.04	To 0.1	To 0.02	0.08	0.08/0.4
>0.04-0.13	>0.1-0.5	>0.02-0.1	0.25	0.25/1.25
>0.13-0.4	>0.5-10	>0.1-2	0.8	0.8/4
>0.4-1.3	>10-50	>2-10	2.5	2.5/12.5
>1.3-4.0	>50	>10	8	8/40

Taula 7: Cut-off ISO 4288:1996.

Un cop calibrada la màquina, s'ha iniciat el mesurament de la rugositat superficial de les diferents peces. Totes les mesures s'han fet al centre de cada peça, amb la mateixa agulla i fent dues passades. Per tal d'obtenir un patró en totes les mesures s'ha passat l'agulla a 90° respecte a la direcció de fabricació. Aquesta direcció de mesura, proporciona el valor més representatiu de la distribució de Ra respecte a altres angles (0° i 45°) per aconseguir

dades (Alsoufi i Elsayed, 2017). Els paràmetres que s'han aconseguit dels testos són Ra i Rq. Amb la rugositat mitjana, Ra, s'obté la desviació promig de la superfície respecte a la seva línia central, permetent valorar la qualitat de la superfície. I amb la rugositat quadràtica mitjana, Rq, s'aconsegueix l'arrel de la mitjana dels valors quadrats de les desviacions de la superfície respecte de la línia de referència. Amb aquesta es pot observar la influència dels pics i les valls en la mesura final. Un valor alt de les dues implica una superfície més rugosa, aquests es poden veure representats en la Taula 8.

Peça	Ra1 (μm)	Ra2 (μm)	Ra3 (μm)	Ra mitjana (μm)	Rq1 (μm)	Rq2 (μm)	Rq3 (μm)	Rq mitjana (μm)
A	7,013	6,872	7,365	7,083	8,041	8,118	8,334	8,164
B	8,958	9,55	9,957	9,488	10,779	11,906	12,57	11,752
C	5,295	5,569	5,583	5,482	6,185	6,449	6,615	6,416
D	5,109	5,837	5,92	5,622	6,123	6,899	6,897	6,640
E	7,798	7,248	7,424	7,490	8,899	8,213	8,415	8,509
F	5,175	4,956	5,094	5,075	6,232	6,143	6,258	6,211

Taula 8: Resultats prova de rugositat.

A través dels resultats obtinguts, es pot veure com a menys velocitat s'obté una superfície menys rugosa. El fet d'extrudir el material lentament comporta una reducció en la rugositat de les capes, i millora l'adhesió entre elles. També s'ha detectat una forta influència del patró d'emplenament sobre l'acabat superficial. S'observa com les peces amb un patró de 90° (C,D,E,F) tenen menys rugositat, que les de 0° (A,B). Com s'ha comentat la peça impresa és un rectangle, per tant, emplenant amb un patró de 90° s'economitzen més els moviments i es generen menys vibracions. Aquesta reducció de les vibracions és la que permet assolir menys rugositat mitjana a la cara superior.

Finalment, s'ha pogut comparar l'efecte de la temperatura del llit i l'extrusor en l'acabat superficial. Estudiant les tres últimes peces, que només es diferencien per la temperatura dels actuadors, s'ha observat com un augment de la temperatura influeix negativament en la superfície. Es pot comprovar com la peça F, temperatura llit de 50°C i extrusor 180°C, és la que té millor acabat de totes les sis, i la peça E, temperatura llit de 80°C i extrusor 220°C, la pitjor.

Amb tot això, es pot dir que la velocitat d'impressió, la temperatura del llit i extrusor i el patró d'emplenament són factors que afecten l'acabat superficial de les peces impreses. D'aquest experiment s'ha deduït que per tal d'aconseguir la menor rugositat possible s'ha de fer una impressió a baixa velocitat, patró de 90° i baixes temperatures. Tot i això, cal remarcar que

per falta de temps no s'ha pogut repetir aquest procediment múltiples vegades. Per tal d'obtenir resultats més rigorosos caldria imprimir més mostres per comprovar si les tendències observades es mantenen.

8. RESUM DEL PRESSUPOST

A través del pressupost es pot consultar la quantitat de material necessari i les hores de feina que es necessiten invertir per la realització del projecte juntament amb els preus unitaris dels materials, el de les hores de cada treballador i el cost total del projecte.

Aplicades totes les unitats requerides, tant de treball com de material, finalment queda que l'import del projecte és de tres mil vuit-cents seixanta-quatre euros amb vint-i-vuit cèntims, sense IVA.

9. CONCLUSIONS

Amb la realització del projecte, s'ha aconseguit crear un sistema capaç de monitorar els factors marcats en un inici. Permetent estudiar el comportament de la impressora FDM escollida i relacionar les dades obtingudes amb el resultat de les peces impreses. Dotant així, al GREP i a Eurecat d'una nova eina per a l'estudi de nous processos de fabricació.

Un cop finalitzat el projecte es pot dir que el resultat és satisfactori, ja que s'han complert tots els objectius plantejats de manera correcta, s'han aconseguit observar i capturar la temperatura i humitat ambiental, la temperatura del llit i extrusor, les acceleracions en els tres eixos de moviment i la temperatura de la broqueta de l'extrusor en temps real. També s'ha desenvolupat una interfície que permet iniciar el procés de fabricació i obtenir-ne les dades per poder-les graficar i estudiar posteriorment. Finalment, s'ha desenvolupat un pla experimental per validar el sistema i s'ha comprovat els efectes dels paràmetres observats en els acabats finals de les peces impreses. Un aspecte a millorar seria la càmera tèrmica, ja que a causa de la seva mala resolució no s'han pogut extreure resultats vàlids. Tot i això, amb aquesta prova s'ha vist que té sentit la seva implementació dins del mateix muntatge. S'ha desenvolupat tant el programa d'adquisició de la matriu de dades com el de transformació d'aquesta a imatge, per tant, només caldria canviar la càmera actual per una amb més resolució i de dimensions similars.

El projecte també ha estat pensat per tal que sigui senzill d'aplicar-hi millores futures. Una de les propostes més ambicioses que podria sorgir d'aquest seria la creació d'un sistema de control en llaç tancat. Aquest aprofitaria totes les dades captades pels sensors per analitzar els paràmetres en temps real i permetre a la impressora ajustar-los automàticament, i així evitar possibles errors en el procés d'impressió. Una altra ampliació podria ser la recaptació de dades per crear un banc d'experiments que permeti aplicar intel·ligència artificial per assajar nous materials o optimitzar el procés de fabricació additiva.

Lluc Vicens Linares

Graduat en Enginyeria Electrònica Industrial i Automàtica

Cornellà del Terri, 1 de setembre de 2023.

10. RELACIÓ DE DOCUMENTS

Aquest projecte està format pel conjunt de cinc documents: memòria, plànols, plec de condicions, estat d'amidaments i pressupost.

11. BIBLIOGRAFIA

ALSOUFI, S. , ELSAYED, E. How Surface Roughness Performance of Printed Parts Manufactured by Desktop FDM 3D Printer with PLA+ is Influenced by Measuring Direction. American Journal of Mechanical Engineering. Vol. 5. No. 5. p. 211-222. 2017.

COLLINS, P. Complete guide: Configure Marlin 2.0.X from scratch and not die trying. 3DWork. (<https://3dwork.io/en/complete-guide-configure-marlin-2-0-x-fromscratch>, 15 de maig de 2023).

HEATFIELD , D. Far Infrared Thermal Sensor Array (32x24 RES). Melexis. (<https://www.melexis.com/en/product/mlx90640/far-infrared-thermal-sensor-array>, 10 de juny de 2023).

MARLIN COMMUNITY. GCode. Marlin Firmware. (<https://marlinfw.org/meta/gcode/>, 6 juny 2023).

MICROSOFT SOFTWARES. Introducción a Visual Studio. Visual Studio. (<https://visualstudio.microsoft.com/es/vs/getting-started/>, 2 juny de 2023).

NOSONOWITZ, D. ADXL345 Digital Accelerometer. Adafruit Learning System. (<https://learn.adafruit.com/adxl345-digital-accelerometer/programming>, 25 de maig de 2023).

RODRIGUEZ, A. Tutorial Sensor de Temperatura y humedad DHT11 Y DHT22. Naylamp Mechatronics-Perú. (https://naylampmechatronics.com/blog/40_tutorial-sensor-de-temperatura-y-humedad-dht11-y-dht22.html, 14 maig 2023).

ROSENBERG, N. Designing 3D printers: Essential knowledge. Independently published. 2019.

TIUSHKA, N. A Beginner's guide to visual studio professional. MarketSplash. (<https://marketsplash.com/tutorials/visual-studio/visual-studio-professional/>, 2 de juny de 2023).

12. GLOSSARI

ADC: Analog-to-Digital Converter

CNC: Computer Numerical Control

DAQ: Sistema d'adquisició de dades

FDM: Fused Deposition Modelling

FFF: Fused Filament Fabrication

GPIO: General Purpose Input/Output

I²C: Inter-Integrated Circuit

IoT: Internet of Things

JSON: JavaScript Object Notation

NTC: Termistors de Coeficient de Temperatura Negativa

PLA: Àcid Polilàctic

RAM: Random Access Memory

RTD: Resistance Temperature Detector

SFF: Solid free-form Fabrication

SLA: Stereo Lithography Apparatus

SLS: Selective Laser Sintering

STL: Standard Triangle Language

USB: Universal Serial Bus

A. RESULTAT MESURA DE PARÀMETRES DEL SISTEMA DE MONITORATGE

En aquest apartat es presentaran els resultats obtinguts de la impressió de les sis peces incloses en el disseny d'experiment. Per a cada peça s'inclou el resultat gràficat de la temperatura i humitat ambiental, de les acceleracions en els tres eixos i de les temperatures de l'extrusor i el llit.

A.1 Peça A

Els resultats obtinguts són per un patró d'emplenament de 0, una velocitat d'impressió de 65 mm/s, una temperatura d'extrusor de 200 °C i una temperatura de llit de 60 °C.

En el següent gràfic, Figura 24, es mostren les variacions en les acceleracions del capçal extrusor en els tres eixos de moviment de la impressora:

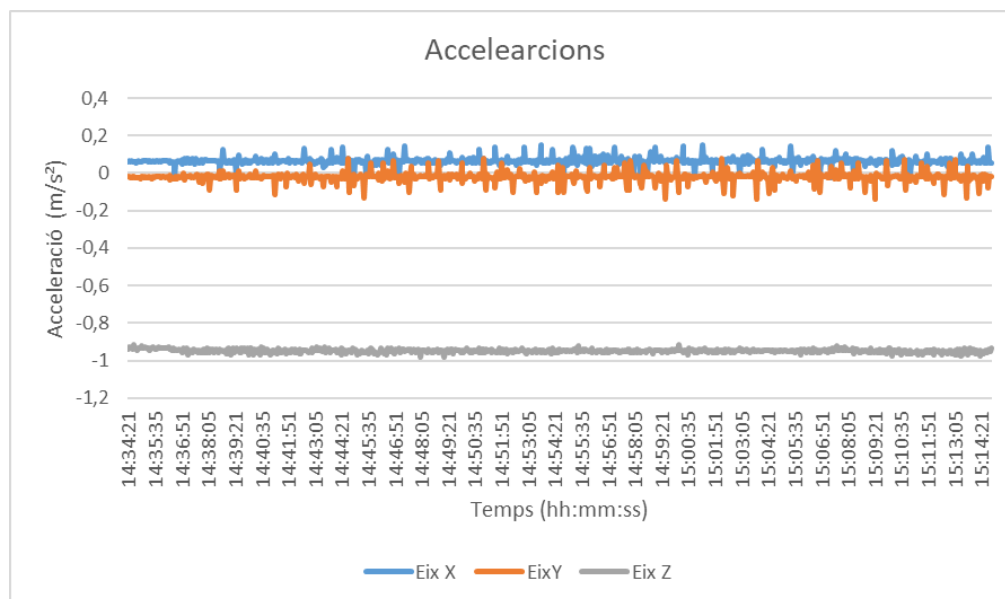


Figura 24: Acceleracions en els tres eixos peça A.

El comportament de les temperatures de l'extrusor i del llit es poden observar en la Figura 25.

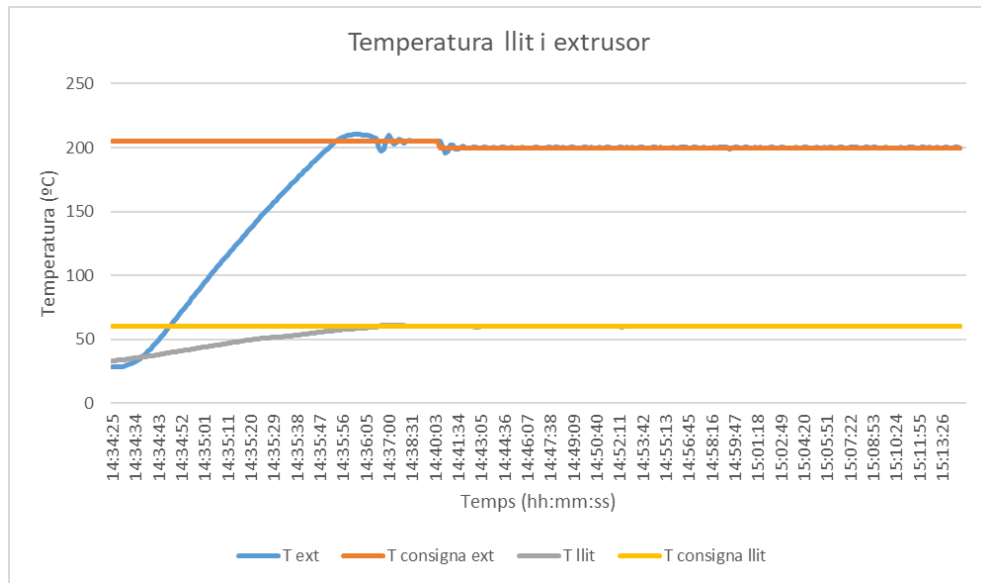


Figura 25: Temperatures llit i extrusor peça A.

Finalment, els paràmetres ambientals de l'entorn on s'ha desenvolupat l'experiment queden representats en la Figura 26.

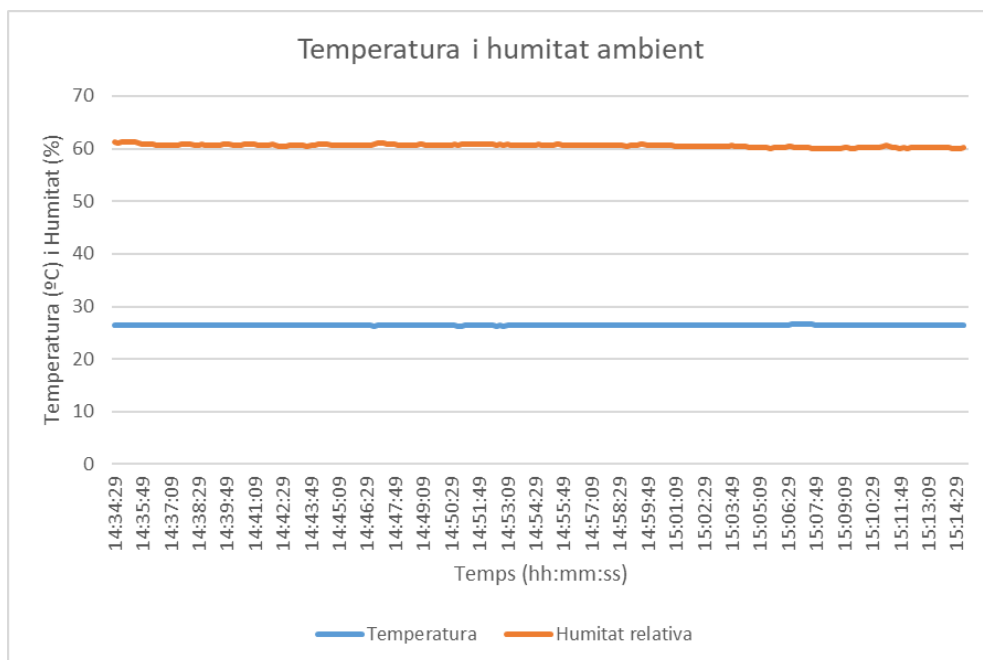


Figura 26: Paràmetres ambientals peça A.

A.2 Peça B

Els resultats obtinguts són per un patró d'emplenament de 0, una velocitat d'impressió de 95 mm/s, una temperatura d'extrusor de 200 °C i una temperatura de llit de 60 °C.

En el següent gràfic, Figura 27, es mostren les variacions en les acceleracions del capçal extrusor en els tres eixos de moviment de la impressora:

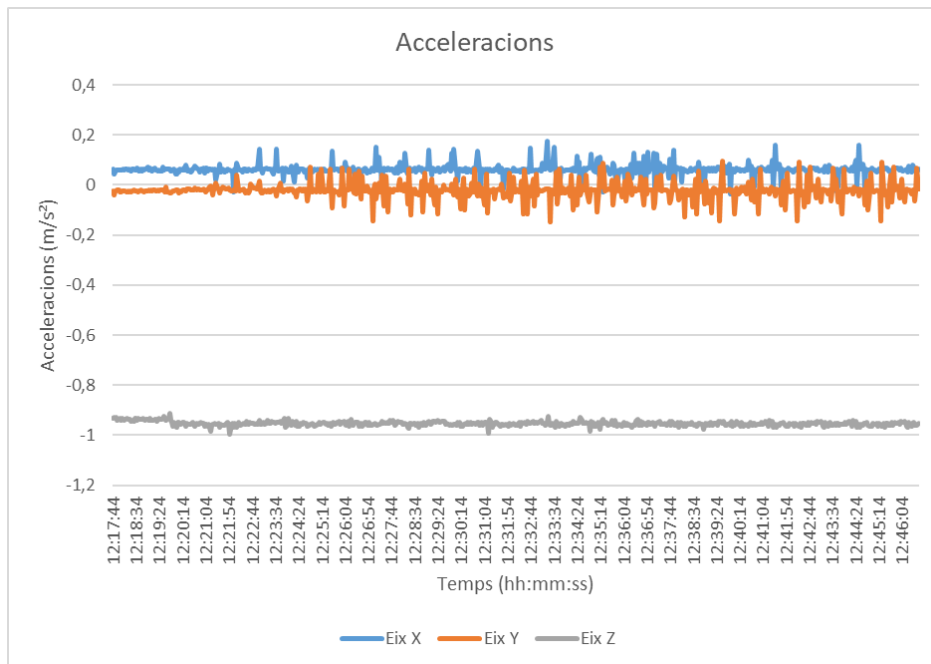


Figura 27: Acceleracions en els tres eixos peça B.

El comportament de les temperatures de l'extrusor i del llit es poden observar en la Figura 28

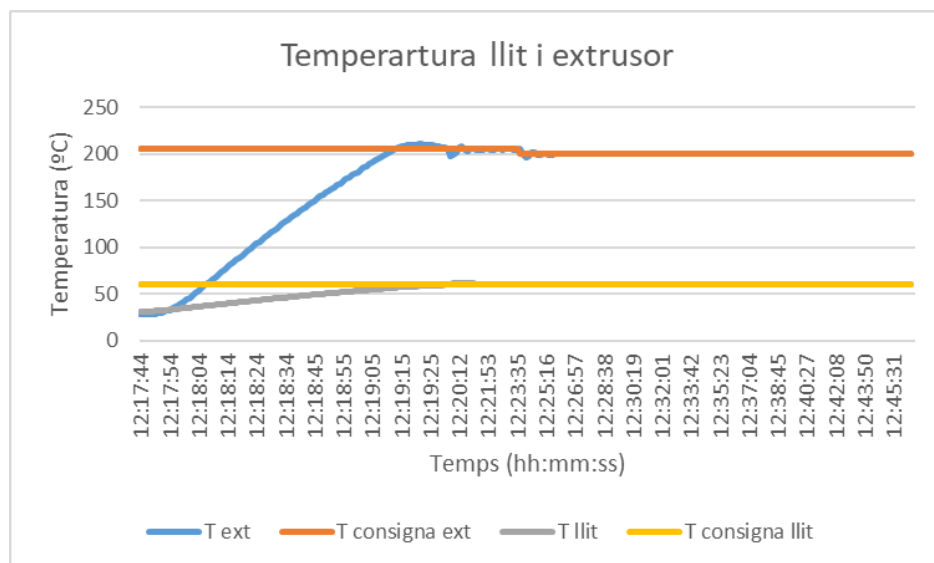


Figura 28: Temperatures llit i extrusor peça B.

Finalment, els paràmetres ambientals de l'entorn on s'ha desenvolupat l'experiment queden representats en la Figura 29.

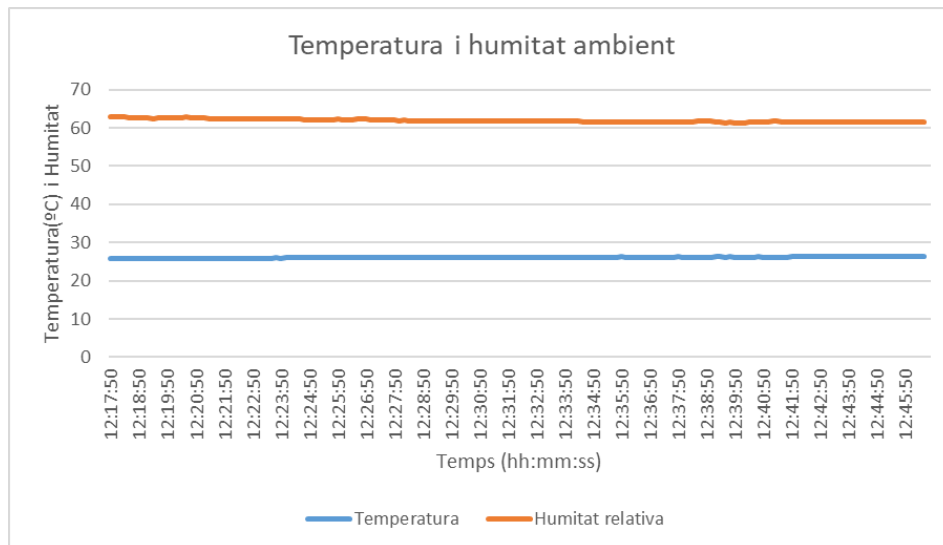


Figura 29: Paràmetres ambientals peça B.

A.3 Peça C

Els resultats obtinguts són per un patró d'emplenament de 90, una velocitat d'impressió de 65 mm/s, una temperatura d'extrusor de 200 °C i una temperatura de llit de 60 °C.

En el següent gràfic, Figura 30, es mostren les variacions en les acceleracions del capçal extrusor en els tres eixos de moviment de la impressora:

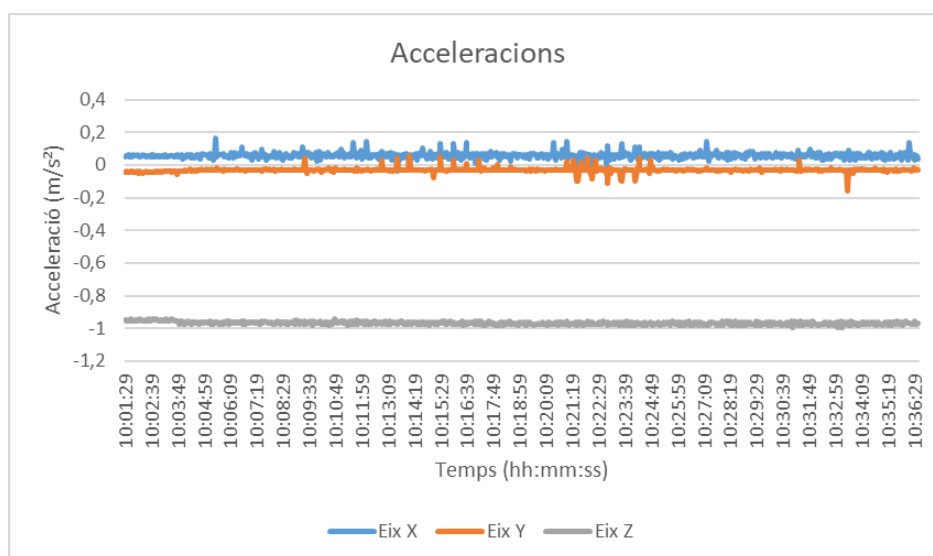


Figura 30: Acceleracions en els tres eixos peça C.

El comportament de les temperatures de l'extrusor i del llit es poden observar en la Figura 31.

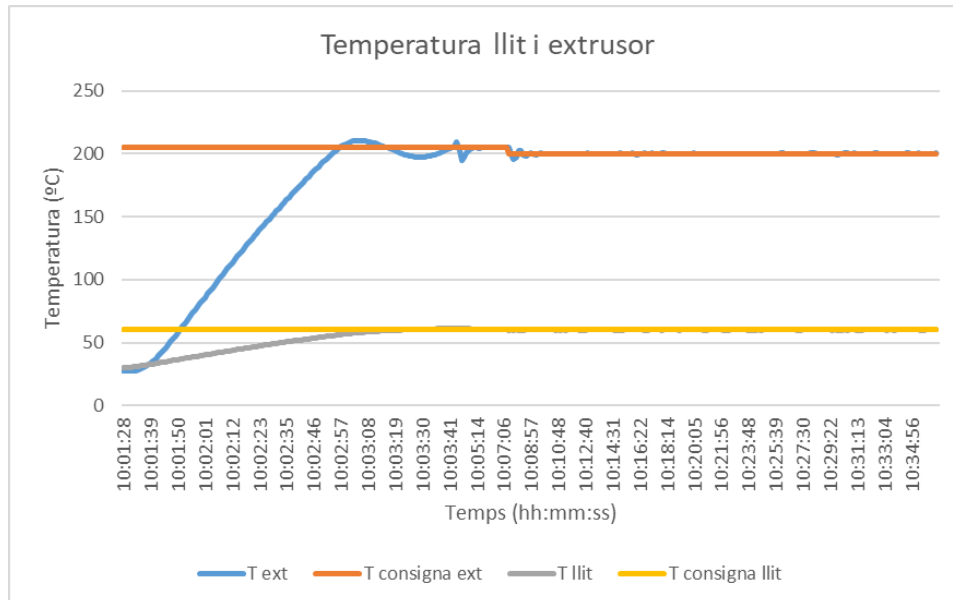


Figura 31: Temperatures llit i extrusor peça C.

Finalment, els paràmetres ambientals de l'entorn on s'ha desenvolupat l'experiment queden representats en la Figura 32.

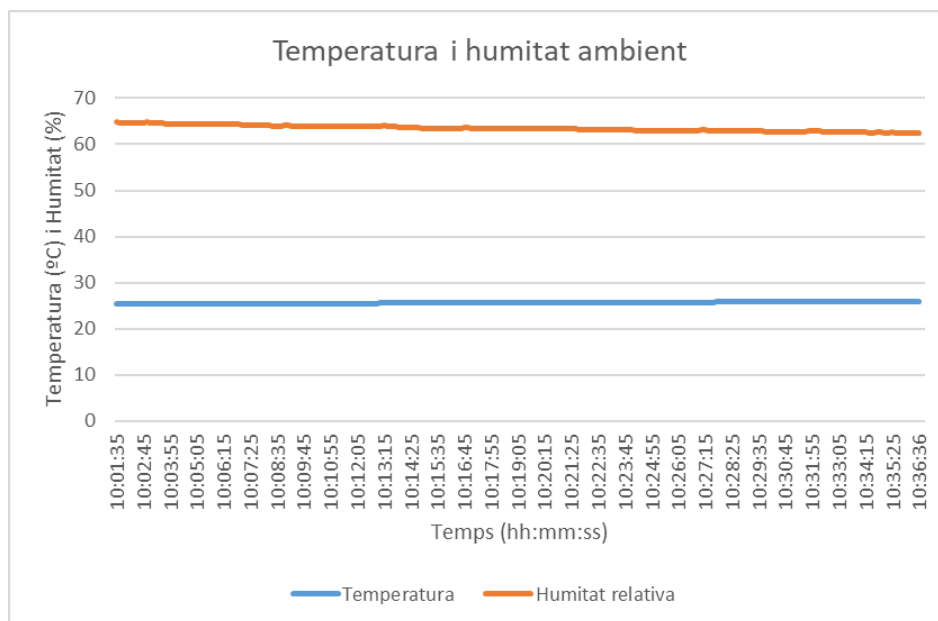


Figura 32: Paràmetres ambientals peça C.

A.4 Peça D

Els resultats obtinguts són per un patró d'emplenament de 90, una velocitat d'impressió de 95 mm/s, una temperatura d'extrusor de 200 °C i una temperatura de llit de 60 °C.

En el següent gràfic, Figura 33, es mostren les variacions en les acceleracions del capçal extrusor en els tres eixos de moviment de la impressora.

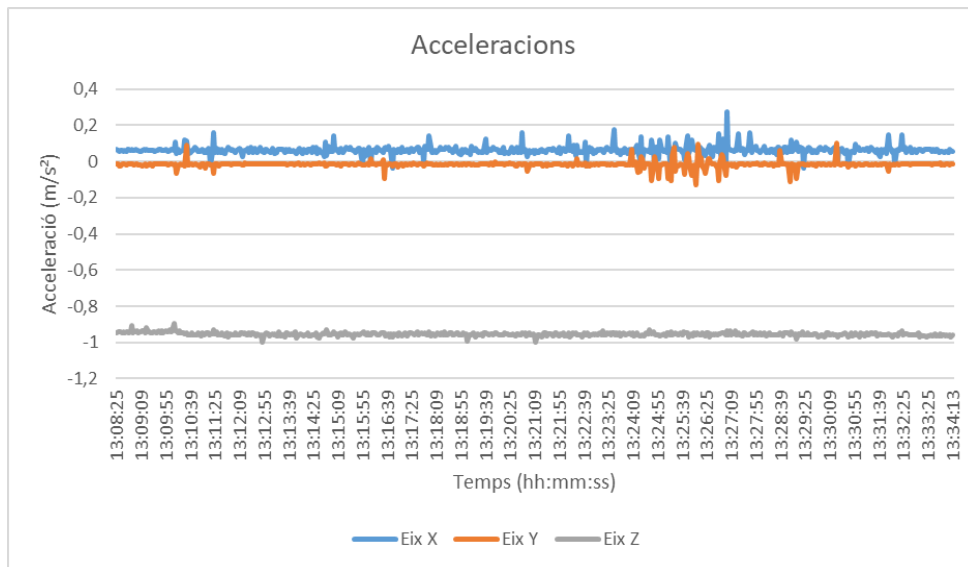


Figura 33: Acceleracions en els tres eixos peça D.

El comportament de les temperatures de l'extrusor i del llit es poden observar en la Figura 34.

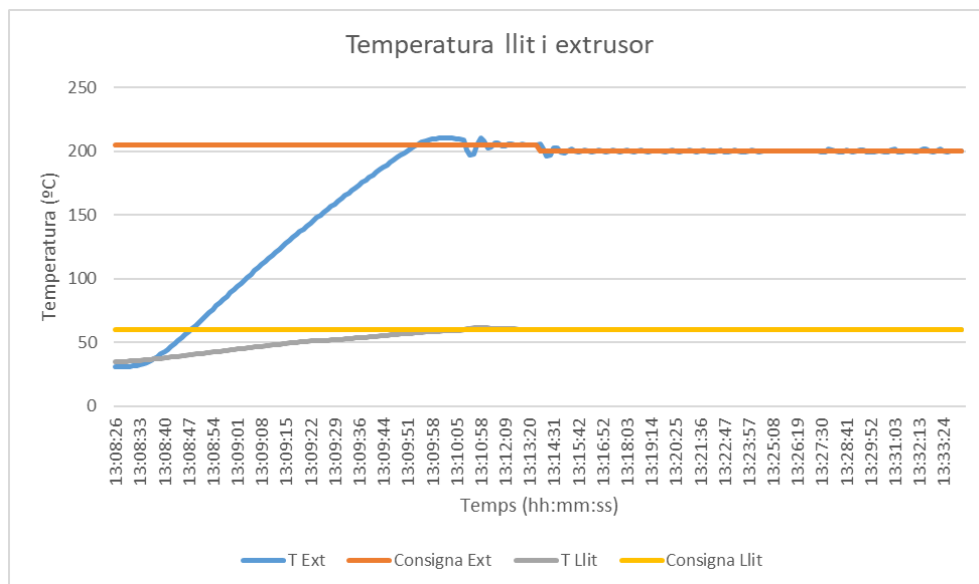


Figura 34: Temperatures llit i extrusor peça D.

Finalment, els paràmetres ambientals de l'entorn on s'ha desenvolupat l'experiment queden representats en la Figura 35.

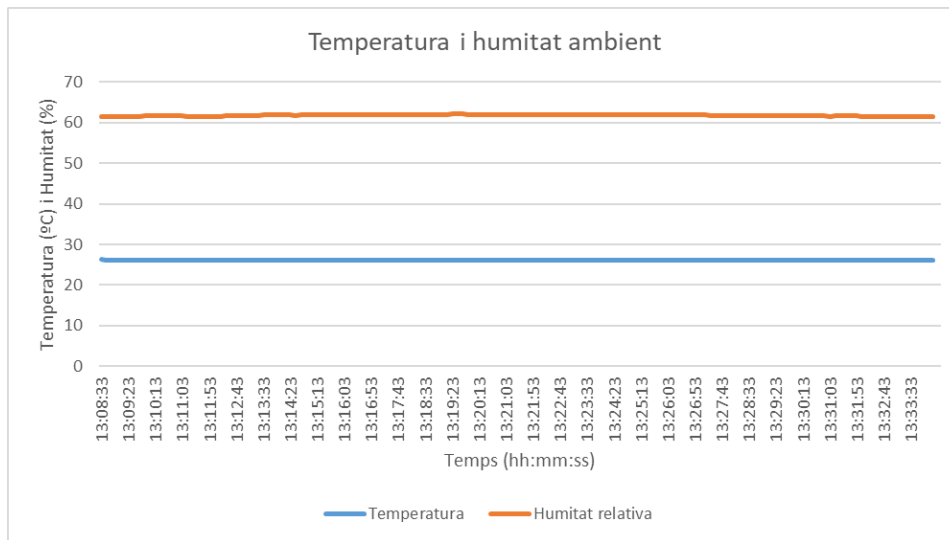


Figura 35: Paràmetres ambientals peça D.

A.5 Peça E

Els resultats obtinguts són per un patró d'emplenament de 90, una velocitat d'impressió de 95 mm/s, una temperatura d'extrusor de 220 °C i una temperatura de llit de 80°C.

En el següent gràfic, Figura 36, es mostren les variacions en les acceleracions del capçal extrusor en els tres eixos de moviment de la impressora:

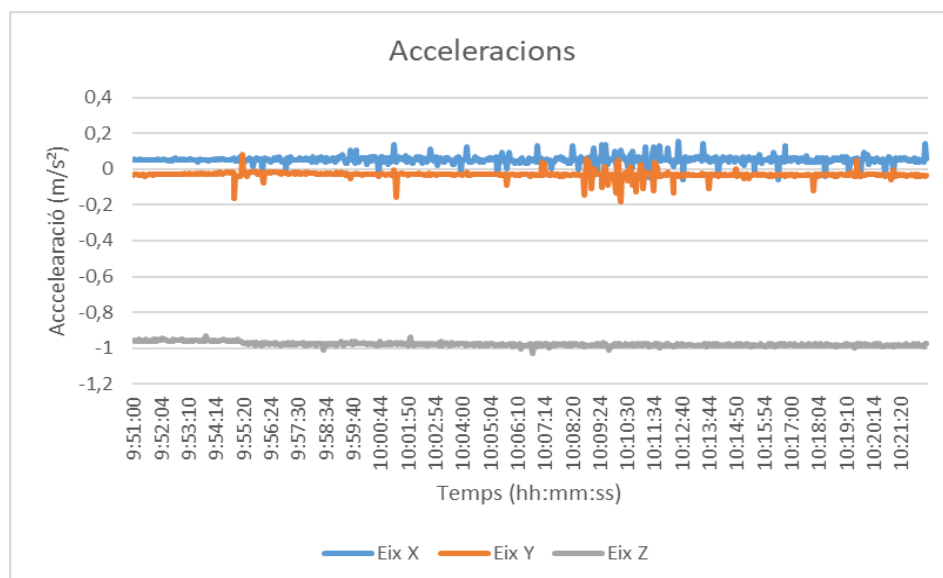


Figura 36: Acceleracions en els tres eixos peça E.

El comportament de les temperatures de l'extrusor i del llit es poden observar en la Figura 37.

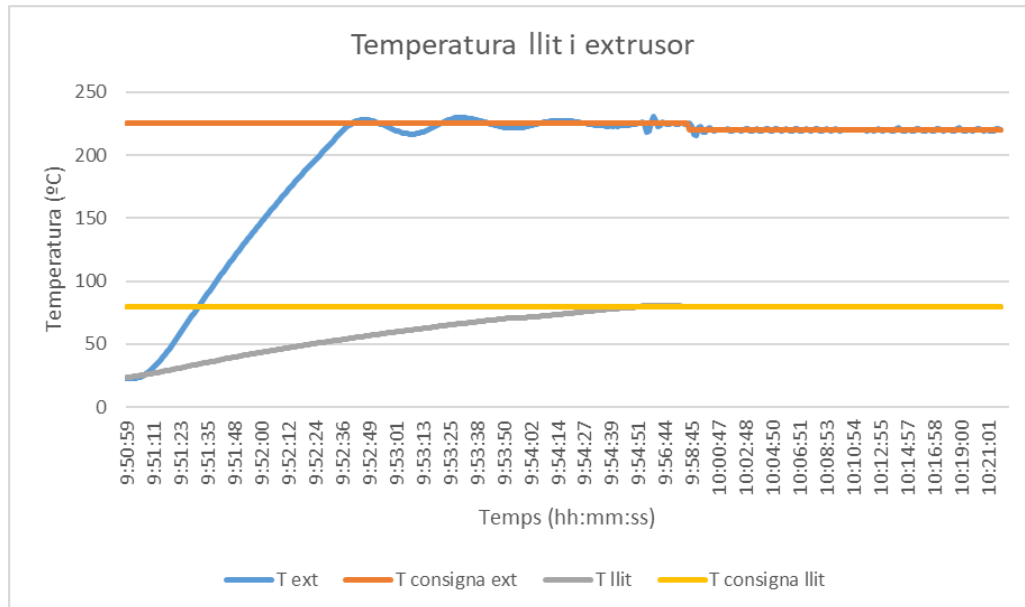


Figura 37: Temperatures llit i extrusor peça E.

Finalment, els paràmetres ambientals de l'entorn on s'ha desenvolupat l'experiment queden representats en la Figura 38.

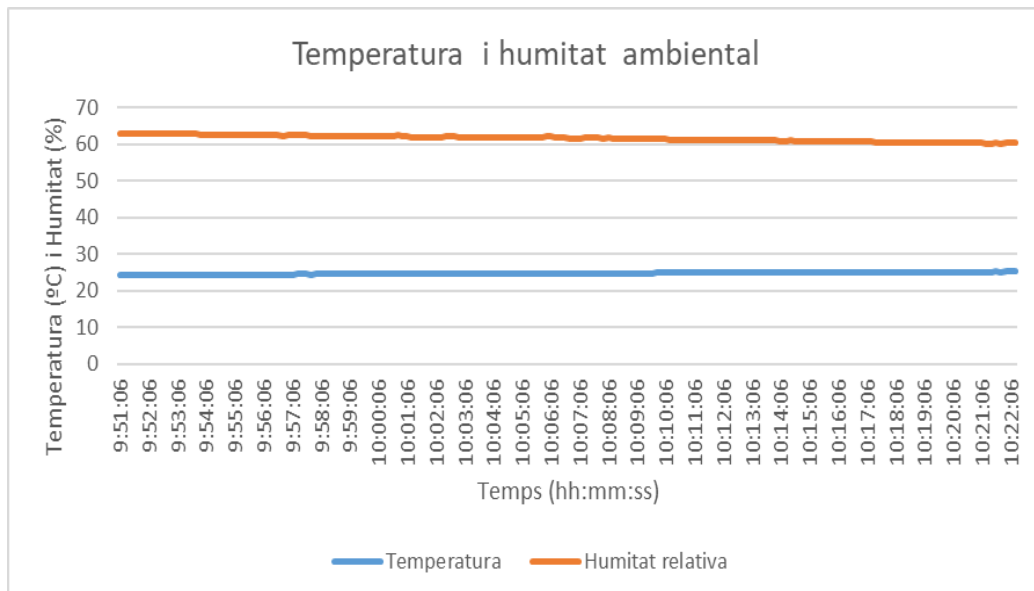


Figura 38: Paràmetres ambientals peça E.

A.6 Peça F

Els resultats obtinguts són per un patró d'emplenament de 90, una velocitat d'impressió de 95 mm/s, una temperatura d'extrusor de 190 °C i una temperatura de llit de 50 °C.

En el següent gràfic, Figura 39, es mostren les variacions en les acceleracions del capçal extrusor en els tres eixos de moviment de la impressora.

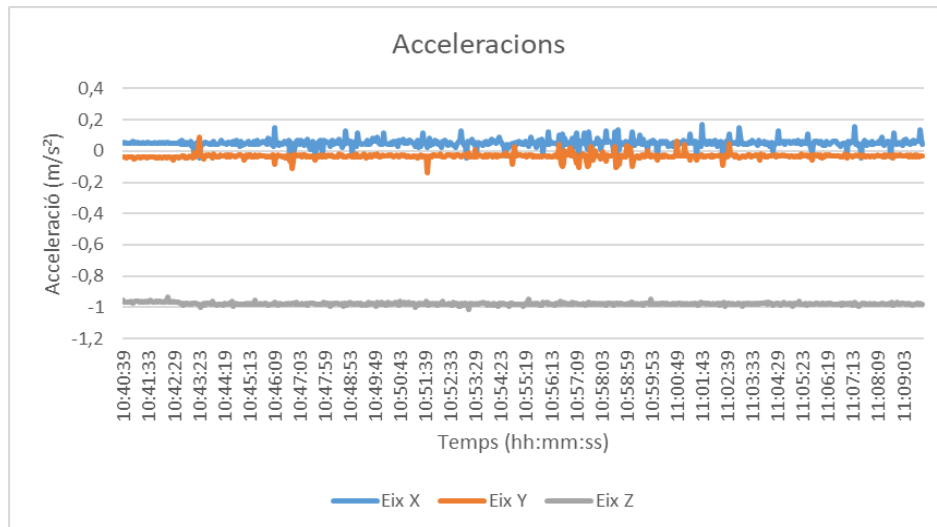


Figura 39: Acceleracions en els tres eixos peça F.

El comportament de les temperatures de l'extrusor i del llit es poden observar en la Figura 40.

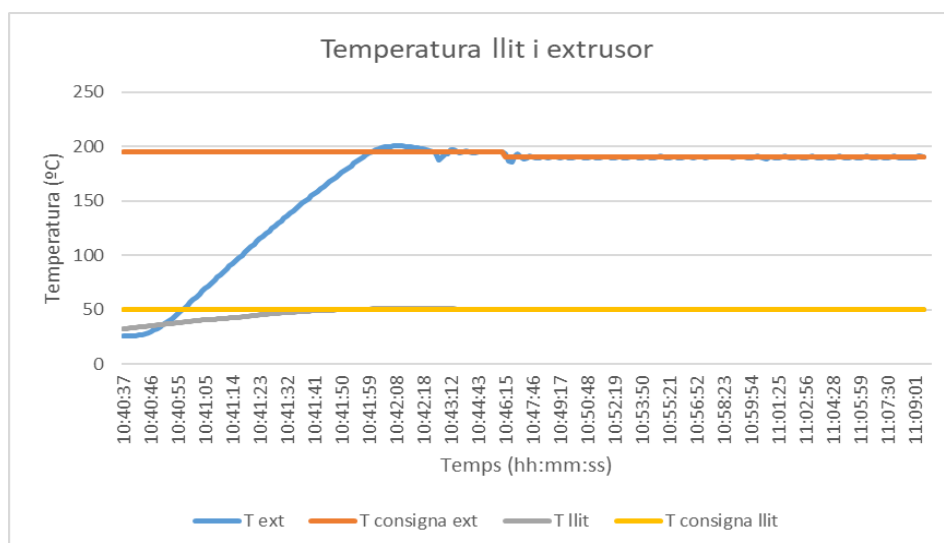


Figura 40: Temperatures llit i extrusor peça F.

Finalment, els paràmetres ambientals de l'entorn on s'ha desenvolupat l'experiment queden representats en la Figura 41.

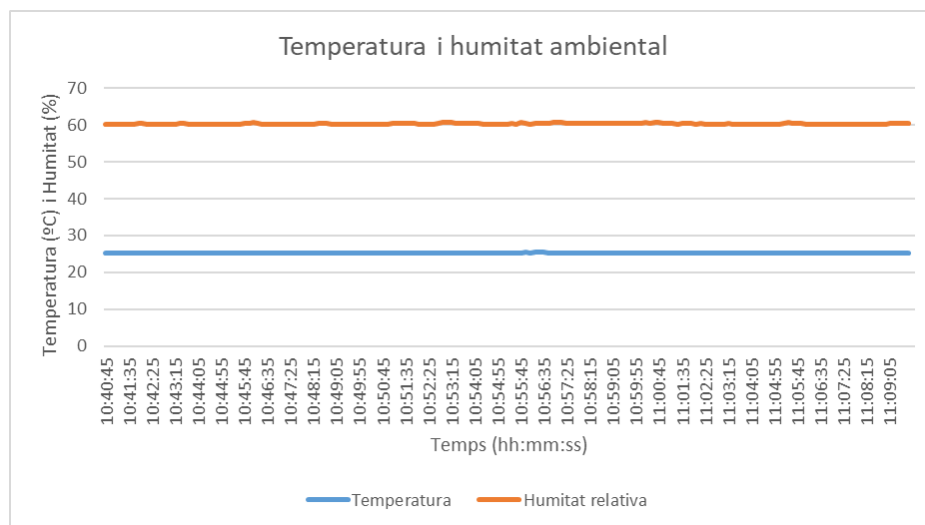


Figura 41: Paràmetres ambientals peça F.

B. Format JSON

JSON (JavaScript Object Notation) és un format lleuger d'intercanvi de dades. És un estàndard àmpliament utilitzat per representar dades estructurades de manera llegible, tant per màquines com per humans. Fet que el converteix en un format ideal per intercanviar eficientment dades entre sistemes i aplicacions. JSON es basa en un subconjunt del llenguatge de programació JavaScript, i tot i ser un format de text independent, els convenis que usa fa que sigui familiar als programadors de llenguatges com C, C++, C#, JavaScript, Python i molts altres.

La popularitat d'aquest format ha crescut en el desenvolupament d'aplicacions web, serveis al núvol, intercanvi de dades i emmagatzematge d'informació. En aquest context, s'ha triat el format JSON com la base per transmetre i emmagatzemar dades del present projecte per la seva simplicitat, compatibilitat amb diferents llenguatges, estructura flexible i eficiència en el processament, fet que provoca que cada vegada sigui més usat a la indústria.

Cal remarcar també que moltes APIs i serveis web utilitzen JSON com el format predeterminat per a l'intercanvi de dades. Treballar en JSON en aquest projecte, facilitarà la posterior integració de possibles millores de tercers, accelerant així la millora del sistema de sensorització.

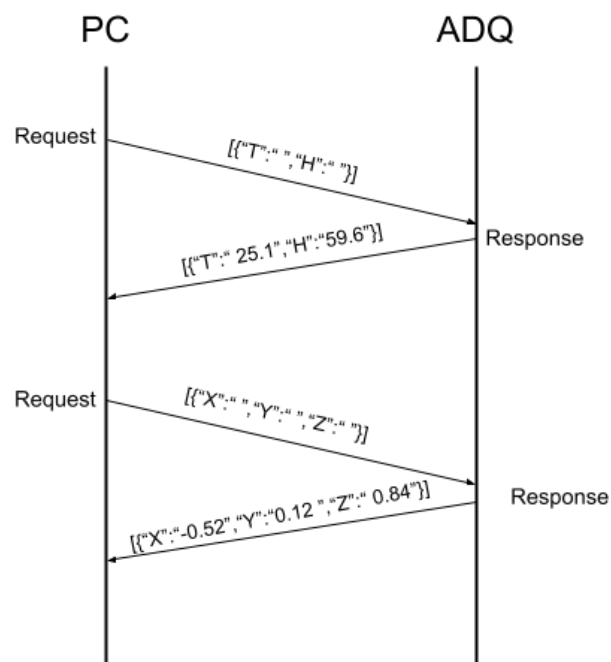


Figura 42: Exemple de comunicació i sintaxi JSON entre dos sistemes.

Tal com es mostra en la Figura 42, el procés de transmissió de dades s'inicia amb la codificació de les dades. En el sistema emissor es codifiquen les dades estructurades en format JSON. Per fer-ho, es representen en parelles clau-valor. Les claus són cadenes de text que actuen com a identificadors, i els valors poden ser cadenes, integers, booleans o matrius. Per tal de serialitzar les dades s'ha de fer en la sintaxi JSON, afortunadament, actualment existeixen una gran varietat de llibreries que compten amb funcions que permeten serialitzar i desserialitzar la informació de manera ràpida i senzilla.

Un cop la informació s'ha transformat ha format JSON, ja es pot transmetre a través del canal desitjat, en el cas del projecte el port serial. Posteriorment, el sistema receptor analitza la cadena JSON i la converteix en una estructura de dades nativa del llenguatge de programació que s'estigui utilitzant. Un cop desserialitzades i les dades, el sistema receptor ja pot utilitzar-les com a variables dins del seu propi codi.

C. CÀLCULS

En aquest apartat s'exposen els diferents càlculs que s'han utilitzat per tal d'obtenir l'àrea de visió segons la distància de les diferents versions del sensor de temperatura MLX90614. Els resultats s'han emprat posteriorment, per valorar quin model s'adapta millor en l'aplicació del present projecte.

Com s'ha comentat anteriorment, l'MLX90614 és un sensor de temperatura infraroig que compta amb set versions diferents segons el seu angle d'obertura, veure Taula 9.

	xAA	xBA	xCC	xCK	xCH	xCF	xCI
Angle d'obertura (°)	90	70	35	13	12	10	5

Taula 9: Angle d'obertura segons versió del sensor MLX90614.

Totes les versions proporcionen una lectura puntual, és a dir, d'un sol punt específic del seu camp de visió. Si en aquesta àrea s'hi acumulen diversos objectes a diferents temperatures, el sensor extreu la mitjana de totes aquestes. Pel sistema de monitoratge es vol evitar aquest fet, per això és important calcular els diferents camps de visió per veure a quina distància es pot col·locar el sensor, i evitar mesures d'elements no desitjats.

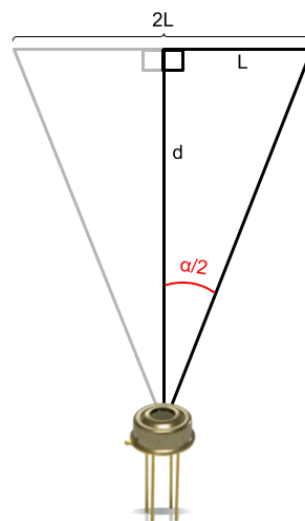


Figura 43: Exemple gràfic del càlcul de L.

Com es pot veure a la figura anterior, Figura 43, per tal d'aconseguir l'àrea depenent de la distància i de l'angle d'obertura primerament cal obtenir el radi de visió, L. Per fer-ho cal fer ús de l'Equació 1.

$$L = \tan(0,5 \cdot \alpha) \cdot d \quad (\text{Eq.1})$$

Un cop extreta la L, ja es pot calcular l'àrea aplicant l'Equació 2.

$$A = \pi \cdot (2L)^2 \quad (\text{Eq.2})$$

A continuació s'adjunta la Taula 10 amb els resultats de les set versions.

Distància (cm)	Àrea visió xAA (cm ²)	Àrea visió xBA (cm ²)	Àrea visió xCC (cm ²)	Àrea visió xCK (cm ²)	Àrea visió xCH (cm ²)	Àrea visió xCF (cm ²)	Àrea visió xCI (cm ²)
1,00	3,14	1,54	0,31	0,04	0,03	0,02	0,01
2,00	12,57	6,16	1,25	0,16	0,14	0,10	0,02
3,00	28,27	13,86	2,81	0,37	0,31	0,22	0,05
4,00	50,27	24,64	5,00	0,65	0,56	0,38	0,10
5,00	78,54	38,51	7,81	1,02	0,87	0,60	0,15
6,00	113,10	55,45	11,24	1,47	1,25	0,87	0,22
7,00	153,94	75,47	15,30	2,00	1,70	1,18	0,29
8,00	201,06	98,58	19,99	2,61	2,22	1,54	0,38
9,00	254,47	124,76	25,30	3,30	2,81	1,95	0,49
10,00	314,16	154,03	31,23	4,08	3,47	2,40	0,60
11,00	380,13	186,38	37,79	4,93	4,20	2,91	0,72
12,00	452,39	221,80	44,97	5,87	5,00	3,46	0,86
13,00	530,93	260,31	52,78	6,89	5,87	4,06	1,01
14,00	615,75	301,90	61,21	7,99	6,80	4,71	1,17
15,00	706,86	346,57	70,27	9,18	7,81	5,41	1,35
16,00	804,25	394,32	79,95	10,44	8,88	6,16	1,53
17,00	907,92	445,14	90,26	11,79	10,03	6,95	1,73
18,00	1.017,88	499,06	101,19	13,21	11,24	7,79	1,94
19,00	1.134,11	556,05	112,75	14,72	12,53	8,68	2,16
20,00	1.256,64	616,12	124,93	16,31	13,88	9,62	2,40
21,00	1.385,44	679,27	137,73	17,98	15,30	10,60	2,64
22,00	1.520,53	745,50	151,16	19,74	16,80	11,64	2,90
23,00	1.661,90	814,82	165,22	21,57	18,36	12,72	3,17
24,00	1.809,56	887,21	179,89	23,49	19,99	13,85	3,45

Distància (cm)	Àrea visió xAA (cm ²)	Àrea visió xBA (cm ²)	Àrea visió xCC (cm ²)	Àrea visió xCK (cm ²)	Àrea visió xCH (cm ²)	Àrea visió xCF (cm ²)	Àrea visió xCI (cm ²)
25,00	1.963,50	962,68	195,20	25,49	21,69	15,03	3,74
26,00	2.123,72	1.041,24	211,13	27,57	23,46	16,26	4,05
27,00	2.290,22	1.122,87	227,68	29,73	25,30	17,53	4,37
28,00	2.463,01	1.207,59	244,86	31,97	27,21	18,85	4,70
29,00	2.642,08	1.295,39	262,66	34,30	29,19	20,22	5,04
30,00	2.827,43	1.386,26	281,08	36,70	31,23	21,64	5,39
31,00	3.019,07	1.480,22	300,14	39,19	33,35	23,11	5,76
32,00	3.216,99	1.577,26	319,81	41,76	35,54	24,62	6,13
33,00	3.421,19	1.677,38	340,11	44,41	37,79	26,19	6,52
34,00	3.631,68	1.780,58	361,04	47,14	40,12	27,80	6,92
35,00	3.848,45	1.886,86	382,59	49,96	42,51	29,46	7,34
36,00	4.071,50	1.996,22	404,76	52,85	44,98	31,16	7,76
37,00	4.300,84	2.108,66	427,56	55,83	47,51	32,92	8,20
38,00	4.536,46	2.224,18	450,98	58,89	50,11	34,72	8,65
39,00	4.778,36	2.342,79	475,03	62,03	52,79	36,57	9,11
40,00	5.026,55	2.464,47	499,71	65,25	55,53	38,47	9,58
41,00	5.281,02	2.589,23	525,00	68,55	58,34	40,42	10,07
42,00	5.541,77	2.717,08	550,93	71,94	61,22	42,42	10,56
43,00	5.808,80	2.848,00	577,47	75,41	64,17	44,46	11,07
44,00	6.082,12	2.982,01	604,64	78,95	67,19	46,55	11,59
45,00	6.361,73	3.119,09	632,44	82,58	70,28	48,69	12,13
46,00	6.647,61	3.259,26	660,86	86,29	73,44	50,88	12,67
47,00	6.939,78	3.402,51	689,91	90,09	76,66	53,12	13,23
48,00	7.238,23	3.548,84	719,58	93,96	79,96	55,40	13,80
49,00	7.542,96	3.698,24	749,87	97,92	83,33	57,74	14,38
50,00	7.853,98	3.850,73	780,79	101,96	86,76	60,12	14,97

Taula 10: Resultat àrees de visió calculades per a cada versió.

D. PROGRAMA

En aquest annex es presenten els quatre codis descrits en el projecte necessaris per a la implementació del sistema de monitoratge.

D.1 Codi programa d'adquisició acceleracions i paràmetres ambientals

```
//Llibreries. https://www.circuitgeeks.com/arduino-dht11-and-dht22-sensor-tutorial/
//https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/
#include <Wire.h>
#include "DHT.h" //See de Arduino DHT
#include <ArduinoJson.h> //Oficial de ArduinoJson

/*****DECLARACIÓ VARIABLES*****/

//Definició del sensor DHT 22
#define DHTTYPE DHT22 // DHT 22 (AM2302)
#define DHTPIN 53 // Definir pin on connectem DHT22
DHT dht(DHTPIN, DHTTYPE); // DHT22

//Variables acceleració en els 3 eixos
float xa;
float ya;
float za;

/*****CREACIÓ OBJECTES JSON*****/

//Creació del JSON per sensor DHT22
const int capacity = JSON_OBJECT_SIZE(2); //Definim tamany màxim del JSON
StaticJsonDocument<capacity> doc; //Creació d'objecte StaticJsonDocument
char Json_DHT22[128];

//Creació del JSON per sensor ADXL345
const int capacity2 = JSON_OBJECT_SIZE(3); //Definim tamany màxim del JSON
StaticJsonDocument<capacity2> doc2; //Creació d'objecte StaticJsonDocument
char Json_ADXL345[256];

String serial_rec;// String on es guardaran dades buffer entrada port serial
int serial_char;

/*****SET UP*****/

void setup() {
  // Obrim comunicació serial i esperem a que s'obri
  adxlconfig();
  Serial.begin(9600);
  while (!Serial) {
```

```
    ; // Espera al port serial a que es connecti.
}

// Iniciem DHT22
dht.begin();

//jSon Ambiental
doc["T"] = "";
doc["H"] = "";

//jSon Acceleracions
doc2["X"] = "";
doc2["Y"] = "";
doc2["Z"] = "";

}

/*****LOOP*****/
void loop() {

    serial_rec = Serial.readString(); //Llegeix la cadena de text del port serial

    if (serial_rec.indexOf("amb;") != -1) { //Busquem subcadena dins de String

        doc["T"] = dht.readTemperature();
        delay(50);
        doc["H"] = dht.readHumidity();
        delay(50);
        //Convertim el JSON en una cadena de text
        serializeJson(doc, Json_DHT22);
        //Imprimim el JSON
        Serial.print(Json_DHT22);
        Serial.println(";"); //Escribim el caràcter de finalització

        serial_rec = ""; //Netegem String

    } else if (serial_rec.indexOf("acc;") != -1) { //Busquem subcadena dins de String
        readadxl();
        doc2["X"] = xa;
        delay(50);
        doc2["Y"] = ya;
        delay(50);
        doc2["Z"] = za;
        delay(50);
        //Convertim el JSON en una cadena de text
        serializeJson(doc2, Json_ADXL345);
        //Imprimim el JSON
        Serial.print(Json_ADXL345);
        Serial.println(";"); //Escribim el caràcter de finalització
    }
}
```



```

    serial_rec = ""; //Netegem String
  }
}

/*****FUNCTIONS*****/

//Funció per configurar els dos acceleròmetres,
//adreça extrusor : 0x53
//adreça llit: 0x1D
void adxlconfig() {
  Wire.begin();

  Wire.beginTransmission(0x53);
  Wire.write(0x2C);
  Wire.write(0x08);
  Wire.endTransmission();

  Wire.beginTransmission(0x53);
  Wire.write(0x31);
  Wire.write(0x08);
  Wire.endTransmission();

  Wire.beginTransmission(0x53);
  Wire.write(0x2D);
  Wire.write(0x08);
  Wire.endTransmission();

  Wire.beginTransmission(0x1D);
  Wire.write(0x2C);
  Wire.write(0x08);
  Wire.endTransmission();

  Wire.beginTransmission(0x1D);
  Wire.write(0x31);
  Wire.write(0x08);
  Wire.endTransmission();

  Wire.beginTransmission(0x1D);
  Wire.write(0x2D);
  Wire.write(0x08);
  Wire.endTransmission();
}

//Funció lectura dels acceleròmetres
void readadxl() {
  Wire.beginTransmission(0x53);
  Wire.write(0x32);
  Wire.endTransmission();
  Wire.requestFrom(0x53, 1);
  byte x0 = Wire.read();
}

```

```
Wire.beginTransaction(0x53);
Wire.write(0x33);
Wire.endTransmission();
Wire.requestFrom(0x53, 1);
byte x1 = Wire.read();
x1 = x1 & 0x03;

uint16_t x = (x1 << 8) + x0;
int16_t xf = x;
if(xf > 511)
{
xf = xf - 1024;
}
xa = xf * 0.004;

Wire.beginTransaction(0x1D);
Wire.write(0x34);
Wire.endTransmission();
Wire.requestFrom(0x1D, 1);
byte y0 = Wire.read();

Wire.beginTransaction(0x1D);
Wire.write(0x35);
Wire.endTransmission();
Wire.requestFrom(0x1D, 1);
byte y1 = Wire.read();
y1 = y1 & 0x03;

uint16_t y = (y1 << 8) + y0;
int16_t yf = y;
if(yf > 511)
{
yf = yf - 1024;
}
ya = yf * 0.004;

Wire.beginTransaction(0x53);
Wire.write(0x36);
Wire.endTransmission();
Wire.requestFrom(0x53, 1);
byte z0 = Wire.read();

Wire.beginTransaction(0x53);
Wire.write(0x37);
Wire.endTransmission();
Wire.requestFrom(0x53, 1);
byte z1 = Wire.read();
```

```

    z1 = z1 & 0x03;

    uint16_t z = (z1 << 8) + z0;
    int16_t zf = z;
    if(zf > 511)
    {
        zf = zf - 1024;
    }
    za = zf * 0.004;

    delay(50);
}

```

D.2 Codi programa càmera

```

//Llibreries
#include <Adafruit_MLX90640.h> // Sensor MLX90640

/*****
*****/
//Inicialització variables i llibreria càmera

Adafruit_MLX90640 mlx;
float frame[32*24]; // creem buffer on s'escriura matriu de càmera

String serial_rec; // String on es guarden dades buffer entrada port serial
int serial_char;

/*****SET
UP*****/

void setup() {
    // Obrir comunicació serial i esperar a que le port es connecti.
    while (!Serial) delay(10);
    Serial.begin(115200);
    delay(100);

    //Imprimim configuració càmera
    camconfig();
}

/*****LOOP*****
*****/
void loop() {

    serial_rec = Serial.readString(); //Llegeix la cadena de text del port serial

    //Comprovació que el sensor llegeix correctament

```

```

if (mlx.getFrame(frame) != 0) {
    Serial.println("Failed");
    return;
    //Si llegeix correctament llegim buffer entrada port serial
} else if (serial_rec.indexOf("cam;") != -1) { //Busquem subcadena dins de String
    //Creem cadena Json amb els 768 valors adquirits
    Serial.print("{\"C\":");
    for (uint8_t h=0; h<24; h++) {
        for (uint8_t w=0; w<32; w++) {
            float t = frame[h*32 + w];

            Serial.print(t, 1);
            Serial.print(", ");
        }
    }
    Serial.println("};"); //Caràcter finalització cadena Json
    serial_rec = ""; //Netegem String
}

delay(500);
}
/*****FUNCIONS*****/

//Funció per configurar càmera, aquí podem canviar resolució i taxa de refresc
void camconfig(){
    Serial.println("Adafruit MLX90640 Simple Test");
    if (! mlx.begin(MLX90640_I2CADDR_DEFAULT, &Wire)) {
        Serial.println("MLX90640 not found!");
        while (1) delay(10);
    }
    Serial.println("Found Adafruit MLX90640");

    Serial.print("Serial number: ");
    Serial.print(mlx.serialNumber[0], HEX);
    Serial.print(mlx.serialNumber[1], HEX);
    Serial.println(mlx.serialNumber[2], HEX);

    //mlx.setMode(MLX90640_INTERLEAVED);
    mlx.setMode(MLX90640_CHESS);
    Serial.print("Current mode: ");
    if (mlx.getMode() == MLX90640_CHESS) {
        Serial.println("Chess");
    } else {
        Serial.println("Interleave");
    }
}

mlx.setResolution(MLX90640_ADC_18BIT);
Serial.print("Current resolution: ");
mlx90640_resolution_t res = mlx.getResolution();
switch (res) {

```

```

    case MLX90640_ADC_16BIT: Serial.println("16 bit"); break;
    case MLX90640_ADC_17BIT: Serial.println("17 bit"); break;
    case MLX90640_ADC_18BIT: Serial.println("18 bit"); break;
    case MLX90640_ADC_19BIT: Serial.println("19 bit"); break;
}

mlx.setRefreshRate(MLX90640_2_HZ);
Serial.print("Current frame rate: ");
mlx90640_refreshrate_t rate = mlx.getRefreshRate();
switch (rate) {
    case MLX90640_0_5_HZ: Serial.println("0.5 Hz"); break;
    case MLX90640_1_HZ: Serial.println("1 Hz"); break;
    case MLX90640_2_HZ: Serial.println("2 Hz"); break;
    case MLX90640_4_HZ: Serial.println("4 Hz"); break;
    case MLX90640_8_HZ: Serial.println("8 Hz"); break;
    case MLX90640_16_HZ: Serial.println("16 Hz"); break;
    case MLX90640_32_HZ: Serial.println("32 Hz"); break;
    case MLX90640_64_HZ: Serial.println("64 Hz"); break;
}
}
}

```

D.3 Codi interfície de monitoratge

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.IO.Ports;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Xml;
using Newtonsoft.Json;

namespace Conjunt1
{
    public partial class Form1 : Form
    {
        //Incialització de variables globals
        String textArd;
        String textSP;
        String text3DP;
        // String filePath;
    }
}

```

```
String textFile;
int numLin;
int countLin;
List<String> linesFile;
String filePath;

private int connected;
private string path;
private IEnumerable<string> lines;
private List<string> list_lines;
private string msg_rec;

private int ttimer;
private int count_line;
private int send_next;
private bool isPrinting;

string output;
public Form1()
{
    InitializeComponent();

    //Permet canviar interval dels timers d'adquisició de dades
    timerTest.Interval = 50;
    timerAmb.Interval = 10000;
    timerAcc.Interval = 2000;
    timerCam.Interval = 4000;

    textArd = "";
    textSP = "";
    text3DP = "";

    connected = 0;
    ttimer = 0;
    msg_rec = "";
    send_next = 0;

    isPrinting = false;
}

//Generem classe per deserialitzar cadena JSON de temperatura i humitat
ambient
public class Ambient
{
    public float T { get; set; }
    public float H { get; set; }
}

//Generem classe per deserialitzar cadena JSON d'acceleracions
public class Acc
```

```

    {
        public float X { get; set; }
        public float Y { get; set; }
        public float Z { get; set; }
    }

//-----BUTTONS-----
-----//

//Botó per obrir comunicació amb Arduino i ESP32
private void buttonConnectSens_Click(object sender, EventArgs e)
{
    serialPortArd.BaudRate = 9600;
    serialPortArd.PortName = "COM8";

    serialPortESP.BaudRate = 115200;
    serialPortESP.PortName = "COM5";

    serialPortArd.Open();
    serialPortESP.Open();

    timerAmb.Enabled = true;
    timerAcc.Enabled = true;
    timerCam.Enabled = true;

    MessageBox.Show("Sensors connected succesfully");

    buttonConnectSens.Enabled = false;
}

//Botó per aturar la comunicació amb Arduino i ESP32
private void buttonStop_Click(object sender, EventArgs e)
{
    timerAmb.Enabled = false;
    timerAcc.Enabled = false;
    timerCam.Enabled = false;
    buttonConnectSens.Enabled = true;

    serialPortArd.Close();
    serialPortESP.Close();
}

//Botó per guardar la informació de les pantalles en un fitxer .txt
private void buttonSave_Click(object sender, EventArgs e)
{
    string fileName = "THAmbient_" + DateTime.Now.ToString("ddMMyy_HHmm") +
".txt";

    SaveFileDialog saveAmb = new SaveFileDialog();
    saveAmb.Filter = "Text document|*.txt";

```

```
saveAmb.Title = "Save RichTextBox1";
saveAmb.FileName = fileName;

string fileName2 = "Acceleracions_" +
DateTime.Now.ToString("ddMMyy_HHmm") + ".txt";
SaveFileDialog saveAcc = new SaveFileDialog();
saveAcc.Filter = "Text document|*.txt";
saveAcc.Title = "Save RichTextBoxAcc";
saveAcc.FileName = fileName2;

string fileName3 = "Camera_" + DateTime.Now.ToString("ddMMyy_HHmm") +
".txt";

SaveFileDialog saveCam = new SaveFileDialog();
saveCam.Filter = "Text document|*.txt";
saveCam.Title = "Save RichTextBoxCam";
saveCam.FileName = fileName3;

string fileName4 = "T_Llit&Extrussor_" +
DateTime.Now.ToString("ddMMyy_HHmm") + ".txt";
SaveFileDialog saveBed = new SaveFileDialog();
saveBed.Filter = "Text document|*.txt";
saveBed.Title = "Save RichTextBox3D";
saveBed.FileName = fileName4;

if (saveAmb.ShowDialog() == DialogResult.OK && saveAcc.ShowDialog() ==
DialogResult.OK && saveCam.ShowDialog() == DialogResult.OK && saveBed.ShowDialog()
== DialogResult.OK)
{
    //Funció per escriure a document de dades ambientals
    StreamWriter writeAmb = new StreamWriter(saveAmb.FileName);
    foreach (object line in richTextBox1.Lines)
    {
        writeAmb.WriteLine(line);
    }
    writeAmb.Close();

    //Funció per escriure a document d'acceleracions
    StreamWriter writeAcc = new StreamWriter(saveAcc.FileName);
    foreach (object line in richTextBoxAcc.Lines)
    {
        writeAcc.WriteLine(line);
    }
    writeAcc.Close();

    //Funció per escriure a document de càmera
    StreamWriter writeCam = new StreamWriter(saveCam.FileName);
    foreach (object line in richTextBoxCam.Lines)
    {
        writeCam.WriteLine(line);
    }
}
```



```

    }
    writeCam.Close();

    //Funció per escriure a document de temperatura llit i extrusor
    StreamWriter write3D = new StreamWriter(saveBed.FileName);
    foreach (object line in richTextBox3D.Lines)
    {
        write3D.WriteLine(line);
    }
    write3D.Close();
}
}

private void buttonConnect3D_Click(object sender, EventArgs e)
{
    if (connected == 0)
    {
        //serialPort3D.BaudRate = 115200;
        //serialPort3D.PortName = "COM11";
        serialPort3D.Open();
        connected = 1;
        MessageBox.Show("Printer connected successfully");
    }
    else
    {
        serialPort3D.Close();
        connected = 0;
    }
}

private void buttonOpen_Click(object sender, EventArgs e)
{
    using (OpenFileDialog openFileDialog = new OpenFileDialog())
    {
        openFileDialog.Filter = "txt files (*.txt)|*.txt|All files
(*.*)|*.*";
        openFileDialog.FilterIndex = 2;
        openFileDialog.RestoreDirectory = true;

        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            //Obtenim direcció del fitxer
            path = openFileDialog.FileName;
            MessageBox.Show("File opened");
        }
    }
}

private void buttonPrint_Click(object sender, EventArgs e)

```

```
{
    if (ttimer == 0)
    {
        MessageBox.Show("Printing...");
        timerTest.Enabled = true;
        timerTest.Start();
        ttimer = 1;
        send_next = 1;
        isPrinting = true;
    }
    else
    {
        isPrinting = false;
        timerTest.Stop();
    }
}

private void buttonUpload_Click(object sender, EventArgs e)
{
    MessageBox.Show("Uploading file, wait until it's done");
    buttonConnect3D.Enabled = false;
    buttonOpen.Enabled = false;
    buttonPrint.Enabled = false;
    lines = File.ReadAllLines(path);
    foreach (string line in lines)
    {
        richTextBox3.AppendText(line);
    }
    count_line = 0;

    list_lines = lines.ToList();

    MessageBox.Show("Text uploaded");
    buttonConnect3D.Enabled = true;
    buttonOpen.Enabled = true;
    buttonPrint.Enabled = true;
    send_next = 1;
}

//Botó per aturar el procés d'impressió en marxa
private void buttonStop3D_Click(object sender, EventArgs e)
{
    DialogResult result = MessageBox.Show("Are you sure to stop printing?",
"Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

    if (result == DialogResult.Yes)
    {
        //Seguit d'accions que farà la impressora al parar la impressió
    }
}
```

```

        serialPort3D.WriteLine("M107");//Apagar ventilador
        serialPort3D.WriteLine("M104 S0");//Apagar extrusor
        serialPort3D.WriteLine("M140 S0");//Apagar llit calent
        serialPort3D.WriteLine("G91");//Agafar posició relativa
        serialPort3D.WriteLine("G1 E-5 F300");//Retraure una mica
        serialPort3D.WriteLine("G1 Z+10 E-5");//Pujar punta de l'extrusor
        serialPort3D.WriteLine("G28 X0 Y0");//Homing
        serialPort3D.WriteLine("M84");//apagar motors
        serialPort3D.WriteLine("M300 S440 P200");//Fer sonar impressora per
indicar la finalització

        serialPort3D.WriteLine("M82");//Mode d'extrusió
        serialPort3D.WriteLine("M2");//Finalitzar programa

        timerTest.Enabled = false;
        serialPort3D.Close();

        connected = 0;
        // Missatge que apareixerà per confirmar cancel·lació de la
impressió
        MessageBox.Show("Stopping printing", "Information",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else if (result == DialogResult.No)
    {
        // Missatge que apareixerà per confirmar cancel·lació acció
        MessageBox.Show("Action canceled", "Information",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

//-----RichTextBox
Functions-----//
//A continuació es creen les funcions per escriure valors dels sensors per
pantalla
delegate void SetTextCallback(string text);

//Funció per escriure Humitat i Temperatura ambiental per pantalla
private void SetText(String text)
{
    if (richTextBox1.InvokeRequired)
    {
        SetTextCallback d = new SetTextCallback(SetText);
        this.Invoke(d, new object[] { text });
    }
    else
    {
        richTextBox1.AppendText(text.Split(';')[0]);
    }
}

```

```

    }

    //Funció per escriure Acceleracions per pantalla
    private void SetTextAcc(String text)
    {
        if (richTextBoxAcc.InvokeRequired)
        {
            SetTextCallback d = new SetTextCallback(SetTextAcc);
            this.Invoke(d, new object[] { text });
        }
        else
        {
            richTextBoxAcc.AppendText(text.Split(';')[0]);
        }
    }

    //Funció per escriure Càmera per pantalla
    private void SetTextCam(String text)
    {
        if (richTextBoxCam.InvokeRequired)
        {
            SetTextCallback d = new SetTextCallback(SetTextCam);
            this.Invoke(d, new object[] { text });
        }
        else
        {
            richTextBoxCam.AppendText(text);
        }
    }

    //Funció per escriure Càmera per pantalla
    private void SetText3D(String text)
    {
        if (richTextBox3D.InvokeRequired)
        {
            SetTextCallback d = new SetTextCallback(SetText3D);
            this.Invoke(d, new object[] { text });
        }
        else
        {
            richTextBox3D.AppendText(text);
        }
    }
}

//-----SerialPort
Reading-----//

//Funció per llegir text del port serial del Arduino

```

```

private void serialPortArd_DataReceived(object sender,
SerialDataReceivedEventArgs e)
{
    textArd += serialPortArd.ReadExisting();
    String temporal;

    if (textArd.Contains(";")) //Busquem caràcter finalització cadena JSON
    {
        if (textArd.Contains("H"))//Busquem caràcter indicador del tipus de
dades, Humitat i temperatura
        {
            try//evitem que si hi ha algun error en la cadena s'escrigui
per pantalla
            {
                //Un cop rebut la cadena cal eliminar el ";" de l'string
per poder deserialitzar el JSON
                textArd = textArd.Replace(";", "").Replace("\n",
"").Replace("\r", "");

                Ambient ambient =
JsonConvert.DeserializeObject<Ambient>(textArd);//Deserealitzem cadena JSON, i
guardem valors a la classe creada
                SetText(DateTime.Now.ToString("HH:mm:ss.fff@") + ambient.T
+ "@" + ambient.H);//Escrivim data i dades separades per @

                SetText("\n");
                textArd = "";//Netegem string
            }
            catch { textArd = ""; }
        }
        else if (textArd.Contains("X"))//Busquem caràcter indicador del
tipus de dades, acceleracions
        {
            if (textArd.Length >= 10)
            {
                try
                {
                    //Un cop rebut la cadena cal eliminar el ";" de
l'string per poder deserialitzar el JSON
                    textArd = textArd.Replace(";", "").Replace("\n",
"").Replace("\r", "").Replace("{", "");

                    Acc acc =
JsonConvert.DeserializeObject<Acc>(textArd);//Deserealitzem cadena JSON, i guardem
valors a la classe creada
                    SetTextAcc(DateTime.Now.ToString("HH:mm:ss.fff@") +
acc.X + "@" + acc.Y + "@" + acc.Z);
                }
            }
        }
    }
}

```

```

        SetTextAcc("\n");
        textArd = ""; //Netegem string
    }
    catch { textArd = ""; }

    }

    }

}

//Funció per llegir text del port serial del ESP32
private void serialPortESP_DataReceived(object sender,
SerialDataReceivedEventArgs e)
{
    textSP += serialPortESP.ReadExisting();

    if (textSP.Contains(";")) //Busquem caràcter finalització cadena JSON
    {
        if (textSP.Contains("C"))
        {
            SetTextCam(DateTime.Now.ToString("HH:mm:ss.fff@"));
            SetTextCam(textSP);

            textSP = "";
        }
    }
}

//Funció per llegir text del port serial de la impressora
private void serialPort3D_DataReceived(object sender,
SerialDataReceivedEventArgs e)
{
    //Inicialitzem variables
    char initialValue = 'A';
    char terminationChar = '\n';
    char inChar = initialValue;

    while (serialPort3D.BytesToRead > 0) //Mentres hi hagin caràcters per
llegir al port serial
    {
        inChar = (char)serialPort3D.ReadChar(); //Llegeix caràcter per
caràcter
        msg_rec += inChar; //Inclou el nou caràcter per formar resposta
impressora
    }
}

```

```

        if (inChar == terminationChar)// si es rep el caràcter de
finalització "\n"
        {
            //Escriu per pantalla resposta impressora
            richTextBox2.BeginInvoke(new Action(() =>
richTextBox2.AppendText("<- " + msg_rec));

            if (msg_rec.Contains("@"))//Si la resposta conté @, ens està
retornant T de llit i extrusor
            {

                string[] words = msg_rec.Split(' ');
                output =words[1].Remove(0, 2) + "@" + words[2].Remove(0, 1)
+ "@" + words[4].Remove(0, 2) + "@" + words[5].Remove(0, 1);
                SetText3D(DateTime.Now.ToString("HH:mm:ss.fff@")+output+
"\n");

                timerTest.Enabled = true;
                msg_rec = "";
                output= "";

            }
        }
        if (msg_rec.Contains("ok"))//Si retorna ok, passem a la següent
instrucció
        {
            send_next = 1;
            timerTest.Enabled = true;
            msg_rec = "";
            break;
        }

        if (msg_rec.Contains("busy"))//Si rebem busy esperem a que
acabi l'acció fins rebre ok
        {
            MessageBox.Show("ocupat");
            timerTest.Enabled = false;//Aturem timer
            send_next = 0;
            msg_rec = "";
            break;
        }

        if (msg_rec.Contains("echo"))//Si rebem busy esperem a que
acabi l'acció fins rebre ok
        {
            timerTest.Enabled = true;//Aturem timer
            send_next = 1;
            msg_rec = "";

```

```

                break;
            }
        }
    }
}

//-----TIMERS-----
-----//

//Timer per demanar lectura de temperatura i humitat ambient per port
serial
private void timerAmb_Tick(object sender, EventArgs e)
{
    serialPortArd.WriteLine("amb;");
}
//Timer per demanar lectura d'acceleracions per port serial
private void timerAcc_Tick(object sender, EventArgs e)
{
    serialPortArd.WriteLine("acc;");
}
//Timer per demanar lectura de càmera per port serial
private void timerCam_Tick(object sender, EventArgs e)
{
    serialPortESP.WriteLine("cam;");
}
//Aquest timmer llegirà una línia de codi i l'enviarà a la impressora cada
interval de temps determinat a l'inici
private void timerTest_Tick(object sender, EventArgs e)
{
    string line;
    if (isPrinting)
    {
        if (send_next > 0)
        {
            line = list_lines.ElementAt(count_line);

            if (line.Length > 0)
            {
                if (!(line[0] == ';' || (line.Length <= 3)))// No llegim
comentaris ni línies en blanc
                {
                    richTextBox2.BeginInvoke(new Action(() =>
richTextBox2.AppendText("-> " + line)); //Escriu línia per pantalla
                    serialPort3D.WriteLine(line); //Escriu resposta
impressora per pantalla
                    send_next = 0;
                }
            }
        }
    }
}

```



```
        count_line += 1;
    }
}

private void button1_Click(object sender, EventArgs e)
{
    serialPort3D.WriteLine("M105");
}
//-----Funcions per fer auto scroll en les TextBox
desitjades-----//
private void richTextBox1_TextChanged(object sender, EventArgs e)
{
    richTextBox1.ScrollToCaret();
}

private void richTextBoxAcc_TextChanged(object sender, EventArgs e)
{
    richTextBoxAcc.ScrollToCaret();
}

private void richTextBoxCam_TextChanged(object sender, EventArgs e)
{
    richTextBoxCam.ScrollToCaret();
}

private void richTextBox3D_TextChanged(object sender, EventArgs e)
{
    richTextBox3D.ScrollToCaret();
}

private void richTextBox2_TextChanged(object sender, EventArgs e)
{
    richTextBox2.ScrollToCaret();
}

private void Form1_Load(object sender, EventArgs e)
{
}
}

}
```

D.4 Codi programa tractament d'imatge

```
using System.Windows.Forms;
using System.Drawing;
using System.Drawing.Imaging;
using System.Text.Json;
using System.Text.Json.Serialization;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.Rebar;

namespace TermicalCam
{
    public partial class Form1 : Form
    {
        private string fileContent = string.Empty;
        private string filePath = string.Empty;
        private string direPath = string.Empty;
        private string[] lines;

        private int height = 24;
        private int width = 32;

        private double minTemp = 30.0;
        private double maxTemp = 220.0;

        public class CameraTemp
        {
            public string C { get; set; }
        }

        public Form1()
        {
            InitializeComponent();
        }

        private void buttonObrir_Click(object sender, EventArgs e)
        {
            using (OpenFileDialog openFileDialog = new OpenFileDialog())
            {
                openFileDialog.InitialDirectory = "c:\\\\";
            }
        }
    }
}
```

```
        openFileDialog.Filter = "txt files (*.txt)|*.txt|All files  
(*.*)|*.*";  
  
        openFileDialog.FilterIndex = 2;  
        openFileDialog.RestoreDirectory = true;  
  
        if (openFileDialog.ShowDialog() == DialogResult.OK)  
        {  
            //Obtén la ruta del fitxer especificat  
            filePath = openFileDialog.FileName;  
            direPath = System.IO.Path.GetDirectoryName(filePath);  
  
            direPath = direPath.Replace("\\", "\\");  
  
            // Llegim el contingut del fitxer en cascada  
            var fileStream = openFileDialog.OpenFile();  
  
            using (StreamReader reader = new StreamReader(fileStream))  
            {  
                fileContent = reader.ReadToEnd();  
            }  
  
        }  
  
        MessageBox.Show("Open");  
    }  
}  
  
private void buttonImage_Click(object sender, EventArgs e)  
{  
    int c = 0;  
    int a = 0;  
  
    try  
    {  
  
        lines = fileContent.Split(new string[] { "\r\n", "\r", "\n" },  
StringSplitOptions.None);  
  
        for (int i = 0; i < lines.Length; i++)  
        {
```

```
        string line = lines[i];
        string[] sublines;
        string date;
        double value;
        string info;
        string[] tempString;
        double tempVal;

        a = i;

        if (line.Length > 10)
        {
            try
            { //Filtrem possibles matrius amb errors d'escriptura
              sublines = line.Split(new string[] { "@", ";" },
StringSplitOptions.None);
              date = sublines[0].Replace(":", "").Replace(":",
"");

              info = sublines[1];
              info = info.Substring(5, info.Length - 6);
              tempString = info.Split(new string[] { ", " },
StringSplitOptions.None);

              Bitmap bmp = new Bitmap(width, height);

              c = 0;

              for (int y = 0; y < height; y++)
              {
                  for (int x = 0; x < width; x++)
                  {
                      //Generem un valor ARGB aleatori
                      int alpha = 255;
                      int red = 0;
                      int green = 0;
                      int blue = 0;
                      value = double.Parse(tempString[c]);
                      value = Math.Round(value);
                      value = value / 10;
```

```

minTemp) * 255.0;

tempVal = (value - minTemp) / (maxTemp -
minTemp) * 255.0;

int grey = Convert.ToInt32(tempVal);

//Passem a escala de grisos
if (grey > 255) { grey = 255; }

red = Convert.ToInt32(0.333 *
Convert.ToDouble(grey));

green = Convert.ToInt32(0.333 *
Convert.ToDouble(grey));

blue = Convert.ToInt32(0.333 *
Convert.ToDouble(grey));

//establim valor ARGB
bmp.SetPixel(x, y, Color.FromArgb(alpha,
red, green, blue));

c++;

    }
}
bmp.Save(direPath + "\\\\" + date + ".png");
}
catch { }

}

}

}
catch
{
    MessageBox.Show(c.ToString());
    MessageBox.Show(a.ToString());
}

MessageBox.Show("Finish");

}

```

```
private void Form1_Load(object sender, EventArgs e)
{
}
}
```