



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA


ETSI Aeroespacial y Diseño Industrial

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Aeroespacial i
Disseny Industrial

Automatització i control inalàmbric dun sistema de
ventilació

Treball Fi de Grau

Grau en Enginyeria Electrònica Industrial i Automàtica

AUTOR/A: Arribas Ríos, Julia Mei

Tutor/a: Rodríguez Ballester, Francisco

CURS ACADÈMIC: 2023/2024



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA
AEROESPACIAL I DISSENY INDUSTRIAL

Automatització i control remot d'un sistema de ventilació

Treball Fi de Grau

Grau en Enginyeria Electrònica Industrial i Automàtica

Autora:
Arribas Ríos, Júlia Mei

Tutor:
Rodríguez Ballester, Francisco

Curs acadèmic 2023/2024



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ETSI Aeroespacial y Diseño Industrial



Resumen

El propósito de este trabajo es realizar el control de un ventilador para que sea capaz de funcionar de manera automática, y poder controlar la temperatura de un habitáculo. Además de controlar el ventilador de manera inalámbrica y monitorizar el estado de las variables que controlan el motor.

En este proyecto se pretende controlar un sistema de ventilación mediante un controlador con conectividad inalámbrica. La automatización de los procesos destinados a gestionar la regulación de la temperatura se realizará a través del ajuste de la velocidad. El control y la monitorización del sistema se realizarán mediante una comunicación inalámbrica, por la cual se recibirá la información real del estado del sistema, como por ejemplo la temperatura capturada, la velocidad del motor, el modo de funcionamiento i la corriente del motor. También se podrá notificar al usuario de posibles problemas que estén ocurriendo, ya sea la parada inesperada del motor, exceso de corriente, un comportamiento inesperado del sistema u otros errores.

Resum

El propòsit d'aquest treball és realitzar el control d'un ventilador per a què siga capaç de funcionar de manera automàtica, per a poder controlar la temperatura d'un habitacle. A més de poder controlar el ventilador de manera remota i monitoritzar l'estat de les variables que controlen el motor.

En aquest projecte es pretén controlar un sistema de ventilació mitjançant un microcontrolador amb connectivitat sense fil. L'automatització dels processos destinats a gestionar la regulació de la temperatura a través de l'ajustament de la velocitat. El control i monitorització del sistema mitjançant una comunicació remota, en el qual es rebrà informació real de l'estat del sistema, com per exemple la temperatura capturada, la velocitat del motor, el mode de funcionament i la corrent del motor. També es podrà notificar a l'usuari de possibles problemes que estiguen ocorrent, ja siga la parada inesperada del motor, l'excés de corrent, un comportament inesperat del sistema, o altres errors.

Abstract

The purpose of this project is to implement the control of a fan so that it can operate automatically, and to be able to control the temperature of a room. And to control the fan wirelessly and motorize the state of the variables that control the motor.

This project aims to control a ventilation system using a microcontroller with wireless connectivity. The automatization of the processes intended to manage the temperature regulation will be carried out by the adjustment of the speed. The control and monitoring of the system will be carried out thorough a wireless connection, through which actual information of the state of the system, for instance the captured temperature, the speed of the moto, the operating mode and the motor current. The user may also be notified of possible problems that are taking place, whether it be an unexpected motor stop, excess of current, unexpected system behavior, or other errors.



Índex general

Document Nº 1: Memòria.....	6
Document Nº2: Plànols.....	83
Document Nº 3: Plec de condicions.....	104
Document Nº 4: Pressupost.....	117



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ETSI Aeroespacial y Diseño Industrial



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA
AEROESPACIAL I DISSENY INDUSTRIAL

Automatització i control remot d'un sistema de ventilació

Treball Fi de Grau

Grau en Enginyeria Electrònica Industrial i Automàtica

Document Nº 1: Memòria

Autora:
Arribas Ríos, Júlia Mei

Tutor:
Rodríguez Ballester, Francisco

Curs acadèmic 2023/2024



Índex de la memòria

1.	Introducció.....	11
2.	Objectiu.....	11
3.	Estudi de necessitats. Factors a tindre en compte: limitacions i condicionants	12
4.	Plantejament de solucions alternatives i justificació de la solució adoptada	13
5.	Descripció detallada de la solució adoptada	16
5.1.	Dispositiu de ventilació	17
5.1.1.	Subsistema electrònic.....	17
5.1.2.	Subsistema d'alimentació.....	20
5.1.3.	Subsistema mecànic	21
5.1.4.	Subsistema de programació	23
5.2.	Quadre de control	23
5.2.1.	Subsistema electrònic.....	23
5.2.2.	Subsistema d'alimentació.....	24
5.2.3.	Subsistema mecànic	24
5.2.4.	Subsistema de programació	26
6.	Justificació detallada dels elements o components de la solució adoptada (càlcul i dimensionament).....	26
6.1.	Diagrama de connexions.....	26
6.2.	Dispositiu de ventilació	29
6.2.1.	Definició de macros, objectes, constants i variables globals	30
6.2.2.	Accés a les dades globals.....	34
6.2.3.	Tasques.....	38
6.2.4.	Controlador	53
6.3.	Quadre de control.....	54
6.3.1.	Definició de macros, objectes, constants i variables globals	57
6.3.2.	Tasques.....	58
6.3.3.	Prioritats	62
6.3.4.	Diagrames de flux de les tasques	63
7.	Conclusions	65
8.	Bibliografia	66
	Annex N°1: Manual d'usuari.....	69
	Annex N°2: Objectiu i desenvolupament sostenible	78



Índex de figures de la memòria

Figura 1: Organigrama tècnic del sistema de ventilació.	16
Figura 2: Placa d'expansió per a la placa de desenvolupament ESP32 de 38 pins.	18
Figura 3: Model 3D de les aspes del prototip.	21
Figura 4: Model 3D de la peça d'unió entre les aspes i l'eix.	21
Figura 5: Model en 3D de l'estructura base de la maqueta.	22
Figura 6: Mòdul ESP32-2432S028.	24
Figura 7: Model 3D de la peça superior de la carcassa que protegeix el quadre de control.	25
Figura 8: Model 3D de la peça inferior de la carcassa que protegeix el quadre de control.	25
Figura 9.	28
Figura 10: Codi exemple per a obtenir la direcció MAC del ESP32.	29
Figura 11: Direcció MAC d'un ESP32.	30
Figura 12: Definició de les macros del programa, associades a sensors i actuadors del sistema.	30
Figura 13: Modificació del document FreeRTOSConfig.h, per a poder utilitzar la funció vTaskDelayUntil().	31
Figura 14: Creació dels objectes que permetran l'accés a les dades obtingudes pels sensors.	31
Figura 15: Constants globals del programa del controlador.	32
Figura 16: Variables globals del control del flux del programa i del control de la velocitat del motor.	33
Figura 17: Variables de per a les tasques, les cues i els semàfors.	33
Figura 18: Creació d'una estructura de dades de tipus struct.	34
Figura 19: Diagrama de flux de la funció readModeActual().	36
Figura 20: Codi exemple de lectura de la variable global "modeActual" a través del mètode del semàfor.	36
Figura 21: Diagrama de flux de la funció writeModeActual().	37
Figura 22: Codi exemple d'escriptura de la variable global "modeActual" a través del mètode del semàfor.	37
Figura 23: Funció pulseDetected().	39
Figura 24: Codi de la tasca taskSendData().	41
Figura 25: Funció onDataRecv().	42
Figura 26: Codi exemple per a la temporització de la funció onDataRecv().	43
Figura 27: Resultat de la temporització de la tasca onDataRecv().	44
Figura 28: Resultat de la temporització de la tasca taskReadMotorSpeed().	45
Figura 29: Resultat de la temporització de la tasca taskReadTemperature().	45
Figura 30: Resultat de la temporització de la tasca taskReadCurrent().	46
Figura 31: Resultat de la temporització de la tasca taskStartAndStop().	46
Figura 32: Resultat de la temporització de la tasca taskControlSpeedAuto().	47
Figura 33: Resultat de la temporització de la tasca taskControlSpeedManual().	47
Figura 34: Resultat de la temporització de la tasca taskSendData().	48
Figura 35: Diagrama de flux de la tasca que llig la velocitat del motor.	48
Figura 36: Diagrama de flux de la tasca que llig la temperatura.	49
Figura 37: Diagrama de flux de la tasca que llig la corrent del motor.	49
Figura 38: Diagrama de flux de la tasca que envia les dades al panell de control.	50
Figura 39: Diagrama de flux de la tasca que controla el motor.	51
Figura 40: Diagrama de flux de la tasca que controlar el motor en mode automàtic.	52
Figura 41: Diagrama de flux de la tasca que controla el motor en mode manual.	53
Figura 42: Diagrama de blocs del controlador per al mode manual.	54



Figura 43: Diagrama de blocs del controlador per al mode automàtic.	54
Figura 44: Definició del driver corresponent.	55
Figura 45: Definició de les macros per al display.	55
Figura 46: Esquema de connexions del display.	55
Figura 47: Icona d'on i off.	56
Figura 48: Modificació de l'arxiu que es va a utilitzar com a icona.	56
Figura 49: Definició de les macros necessàries per a utilitzar la pantalla tàtil.	57
Figura 50: Esquema de connexions de la pantalla tàtil.	57
Figura 51: Objectes per a la utilització de la pantalla i la pantalla tàtil.	57
Figura 52: Constants del programa del panell de control.	58
Figura 53: Variables globals del programa del panell de control.	58
Figura 54: Pantalla d'inici del panell de control.	60
Figura 55: Pantalla menú del panell de control.	60
Figura 56: Pantalla del menú del mode manual que apareix en el quadre de comandaments.	61
Figura 57: Pantalla del menú mode automàtic que es mostra en el display.	61
Figura 58: Missatge d'error de parada inesperada del motor.	62
Figura 59: Diagrama de flux de la tasca OnDataRecv().	63
Figura 60: Diagrama de flux de la tasca timerWirelessCommunication().	63
Figura 61: Diagrama de flux de la tasca pantallaHandler().	64

Índex de taules de la memòria

Taula 1: Connexions dels components del dispositiu de ventilació.	27
Taula 2: Esquema de funcionament de la tasca taskSpeedControlAuto() segons les variables "estatActual" i "modeAcutal".	40
Taula 3: Esquema de funcionament de la tasca taskSpeedControlManual() segons les variables "estatActual" i "modeAcutal".	41
Taula 4: Ordre de prioritats de les tasques en el programa del dispositiu de ventilació.	43
Taula 5: Màxim temps de còmput observat.	44
Taula 6: Pàgina que es mostra pel display segons la variable "page".	59

Índex de les equacions de la memòria

Equació 1: equació per al càlcul de les revolucions de l'eix principal.	39
Equació 2: equació per al càlcul de les revolucions de l'eix secundari.	39



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ETSI Aeroespacial y Diseño Industrial



1. Introducció

La necessitat de regular la temperatura ha vist una desmesurada pujada, degut al incessant ascens de d'aquesta. L'any 2023 ha batut el rècord de les xifres més altes mai registrades, fins a 10 graus més que la mitja, segons l'Aemet. La tendència que es projecta per als pròxims anys és una continuada pujada de les temperatures, fins al punt d'on no es puguem distingir la primavera de l'estiu, o l'estiu de la tardor. De manera que els estius es faran cada vegada més llargs, i la necessitat de l'ús d'aparells de refrigeració més prolongada i intensa, arribant fins i tot a generar un impacte en l'economia de les famílies, per a poder sofocar la calor dins de casa [1].

L'aire condicionat, cada vegada més comú als edificis, consumeix molta energia i contamina l'aire, degut a què expulsa aire calent a l'atmosfera, incrementant la temperatura exterior [2]. A més del fet que disminueix la humitat de l'ambient, provoca malestars a la salut humana, com tos, irritació de gola, constipats, inflamació de les vies respiratòries, deshidratació al veure's reduïda la humitat en l'ambient, migranyes, infeccions virals al mantindre's en circulació el mateix aire contínuament, sequedat i irritació als ulls o problemes en la pell [3].

Algunes persones abusen de l'aire condicionat, i estableixen unes temperatures inferiors a les recomanades pels experts. El que afavoreix encara més la contaminació del medi ambient, i més si la font d'energia elèctrica que alimenta els aparells no és d'origen renovable. De manera que és un cicle viciós, on la temperatura augmenta, les persones engeguen l'aire condicionat, i això provoca que augmenten més les temperatures.

L'ús d'aquest aparell genera un gran impacte en el medi ambient, ja que constitueix un 10% del consum de l'electricitat mundial, i com s'ha mencionat prèviament, emet gasos nocius a l'atmosfera, destruint la capa d'ozó, el que afavoreix l'efecte hivernacle, que provoca un increment de les temperatures [4].

Una de les alternatives és el ventilador, un mitjà menys invasiu per a la salut, menys contaminant, més econòmic, on la instal·lació és més barata, i a més a més consumeix molt menys que l'aire condicionat.

La inspiració de la idea d'aquest treball de fi de grau em va vindre una d'aquestes caloroses nits d'estiu, quan no podia dormir i tenia el ventilador encès, programat per a què després d'un temps s'apagava. Aleshores, passat un temps el ventilador s'apagava, però de sobte jo em despertava perquè començava a tindre calor. Però no volia tindre el ventilador tota la nit encès, perquè pel matí tenia fred i a més, era una despesa innecessària d'energia. Aleshores em va vindre la idea de què el ventilador es regulara a partir de la temperatura, així sols funcionaria quan fora necessari, per a no malgastar energia i a la vegada poder descansar satisfactòriament.

2. Objectiu

En Espanya moltes cases no compten amb aire condicionat, sols amb ventiladors. En aquest projecte es pretén modernitzar aquests aparells de refrigeració convencionals, i transformar-los en dispositius



automatitzats, per a reduir el consum i augmentar l'eficiència, mitjançant l'ús de recursos mínims tant materials, com econòmics.

El l'objectiu general d'aquest projecte és el disseny d'un sistema per a poder controlar un motor, mitjançant l'ús de microcontroladors que es puguen connectar remotament entre sí. Un es connectarà al motor, el controlarà i monitoritzarà les variables com la velocitat. I l'altre s'encarregarà de comunicar-se amb l'usuari, per a què pugua controlar el ventilador, veure l'estat d'aquest i la temperatura de l'habitació. Tots dos es comunicaran a través d'un sistema sense fil.

Més concretament es persegueix la programació d'un microcontrolador, per poder controlar el motor i saber la velocitat que porta, saber la temperatura mitjançant un sensor. Aconseguir una comunicació remota efectiva entre els dos microcontroladors, per a comunicar les dades, mitjançant un protocol de transmissió de dades compacte. La programació de l'altre microcontrolador per a rebre i emetre informació a l'usuari a través d'una pantalla tàctil.

3. Estudi de necessitats. Factors a tindre en compte: limitacions i condicionants

Com s'ha mencionat prèviament la pujada de les temperatures crea la necessitat de refrescar l'ambient. La manera més ràpida és mitjançant l'aire condicionat, no obstant, l'ús d'aquests electrodomèstics consumeix molta energia i contamina, per tant s'ha de buscar altres solucions més amigables per al medi ambient.

També s'ha comentat els avantatges que ofereixen els ventiladors front als aires condicionats. A més, serien més eficients si sols funcionaren quan fora necessari. Encara que no són tan efectius com l'alternativa.

Existeixen certes limitacions físiques que un ventilador no pot superar degut al mètode de funcionament que empra. Un ventilador no redueix la temperatura en sí, sinó que redueix la sensació tèrmica que pot arribar a baixar fins a 5 graus. A més, aquest aparell té una altra limitació, per a reduir la sensació tèrmica, també depèn de la humitat en l'habitació. Es necessita una humitat no molt baixa, per a què siga efectiu, ja que la humitat afecta a la capacitat de l'aire per a retindre o alliberar calor. Si hi ha humitat la transferència de calor és major. A més, s'ha de tindre en compte que si la temperatura de l'habitació és major que la temperatura del nostre cos, engegar el ventilador no serà efectiu, ja que calfarà en compte de refredar el cos [5].

A causa de les limitacions del temps que s'ha de dedicar a aquest treball, el muntatge es realitzarà sobre un maqueta. No obstant el funcionament del sistema dissenyat seria el mateix.

Com el muntatge es va a fer en base a una maqueta, cal tindre en compte que la variació de la temperatura serà menys notable que en un muntatge per a un ventilador real, degut a les dimensions de les aspes. Al ser un model reduït, el caudal d'aire que podrà aconseguir serà molt menor.

Per al control de la temperatura mitjançant un motor s'han d'acomplir diferents aspectes. El control de la velocitat del motor, el coneixement de la velocitat real que porta el motor i la temperatura real que hi ha a l'habitacle.



El motor ha de ser de corrent continua, ja que els d'alterna funcionen diferent. També és necessari que compte amb un encoder, en el seu defecte, s'hauria d'incorporar. D'igual manera haurà de tindre unes revolucions màximes de entre 200 i 400 RPM [6], depenent de les aspes del ventilador, quan més grans les aspes, menys revolucions màximes es requeriran per a emetre un caudal d'aire determinat.

La comunicació entre el controlador i el quadre de comandaments requerirà un protocol ràpid, fiable, amb suport de software, que no interferisca en altres aparells que emeten radiofreqüències ni que altres aparells interferisquen en la comunicació, a més han de poder enviar-se paquets de dades. La comunicació que s'ha escollit ha sigut la ESP-Now, les raons se desenvolupen en el punt 4. Plantejament de solucions alternatives i justificació de la solució adoptada, però cal mencionar una limitació rellevant, abans d'establir la comunicació a distància entre els dos dispositius, s'ha de saber la seua direcció MAC per a poder emparellar-los.

4. Plantejament de solucions alternatives i justificació de la solució adoptada

Per a l'òptim desenvolupament del projecte és fonamental estudiar diverses alternatives per a aconseguir una solució eficient. En aquest apartat es presenta un anàlisi dels diversos enfocaments que es varen donar al desenvolupar el projecte, a més de les diverses alternatives dels components significatius que es van a utilitzar, avaluant les avantatges i els inconvenients, justificant així l'elecció adoptada finalment.

En un inici es va plantejar crear un comandament que permetera comunicar-se amb un ventilador ja existent, el qual es controlés de manera remota a través d'ones de radiofreqüència. Mitjançant comandes que ja existien, un microcontrolador a través d'un emissor de radiofreqüència, podria automatitzar la velocitat del dispositiu, depenent de l'hora i/o de la temperatura, s'establiria certa velocitat. També s'havia pensat la idea que aquest microcontrolador tinguera connectivitat a internet, per poder controlar el ventilador a través d'un altre aparell com poguera ser el mòbil, la tablet o l'ordinador, i que el microcontrolador fera de pont. Un altra aplicació que es va pensar és la de fer funcionar l'aparell amb comandes de veu, mitjançant un mòdul de reconeixement de veu. Finalment, aquesta idea va ser desestimada degut a què el microcontrolador no rebia retroalimentació del ventilador, aleshores era un sistema en llaç obert, que no és recomanable perquè no és capaç de corregir errors al no saber l'estat real del sistema, fet que pot conduir a errors en el control del mateix, fallades de sobrecàrrega o un funcionament malbé en el motor.

Aleshores es va desenvolupar la idea del control d'un motor, ja existent, al qual se li incorpora un microcontrolador i altres components, per a poder controlar-lo remotament des d'un altre microcontrolador. A continuació es desglossaran les alternatives que s'han tingut en consideració per als diferents components i parts del sistema.



Microcontrolador

Per al control del motor varen barallar diferents possibilitats, una d'aquestes era una targeta STM32, però es va descartar pel preu que tenen aquestes targetes, ja que aquest projecte pretén crear un sistema el més econòmic possible, i la targeta escollida ofereix les suficients prestacions per a cobrir les necessitats d'aquest treball. Altra de les alternatives que es varen considerar foren les targetes Arduino, però aquelles que eren econòmicament assequibles no complien amb els requisits necessaris, com l'Arduino nano. Finalment la solució adoptada per a arribar a un compromís entre preu i prestacions és la targeta ESP32-WROOM-32. Altres raons per a l'elecció d'aquesta targeta és la ampla comunitat que té, les llibreries que hi ha disponibles, la gran facilitat i comoditat d'ús que ofereix i a més la possibilitat de comunicació sense fil que ofereix sense afegir ningun mòdul.

Quadre de comandaments

Es varen tantejar diferents idees per a la interacció de l'usuari amb el panell de control. Primerament es va pensar en una pantalla LCD, de 16x2, però aquesta ofereix una molt limitada quantitat de caràcters màxims a la pantalla, motiu pel qual es va descartar. Un altra de les opcions que es varen tindre en compte va ser la pantalla oled SSD1327, amb una densitat de píxels major, de 128x128, en blanc i negre. Es varen fer proves amb aquest display però la resposta que oferia era massa lenta, aleshores també es va desestimar. Després es va intentar amb una pantalla oled ST7789 de 240x240 píxels, però la llibreria no era compatible amb el ESP però sí amb l'Arduino nano, i es va intentar realitzar la connexió de la pantalla controlada per l'Arduino nano, i aquest es connectava a través del protocol I2C al ESP, mitjançant el format Json d'empaquetament de dades. Malauradament, aquesta idea no va funcionar, perquè l'Arduino nano no tenia suficient memòria SRAM i les dades no eren correctament processades. Finalment, es va optar pel ESP32-2432S028R, el qual porta una pantalla tàctil resistiva integrada, la 16BITRGB65K, de manera que desapareix la necessitat d'incorporar botons o qualsevol altre tipus d'input per a què l'usuari interaccione amb el quadre de comandaments.

Motor

Es varen tindre en consideració diversos tipus de motors que podien complir les característiques nomenades en el punt anterior. Una de les opcions va ser un motor pas a pas, que es va descartar pel mètode de control que requereix, i que els ventiladors convencionals no porten d'aquests. Una altra alternativa va ser un motor brushed, que també es va descartar per les desavantatges que es presenten en comparació a l'opció escollida, un motor brushless. Les avantatges que ofereix aquest últim són les següents, és més eficient, més potent, menys sorollós, amb una vida útil molt major, degut al sistema mecànic del qual està compost, ja que les peces suporten menys fregament entre elles, i a més requereix un menor manteniment.

Com el projecte està basat en una maqueta, el motor escollit compta amb una menor potència, no obstant, compleix totes les característiques enumerades prèviament.

Aspes

D'igual manera, l'hèlix escollit és per a la maqueta, motiu pel qual no s'han determinat unes característiques que necessite complir. S'ha escollit un model 3D d'internet per al propòsit de la maqueta.

Sensor de corrent

El consum de corrent del motor havia de ser mesurat, de manera que es varen barallar diferents opcions. Entre aquestes, es va pensar en un ACS712, que hi ha de tres tipus, de $\pm 5A$, $\pm 10A$ i $\pm 20A$, però



cap d'aquests tenia una resolució adequada per a mesurar la corrent del motor, a més, l'eixida que donava estava en un rang de 0 i 5 V, i el ADC del ESP32 sols accepta fins els 3,3V. Una altra de les alternatives que es varen plantejar va ser un ZMCT103C, un sensor no invasiu, que no necessita interrompre el circuit per a mesurar la corrent, però es va descartar també perquè la resolució no era la correcta. El sensor que finalment es va escollir va ser INA 219, encara que aquest és un sensor invasiu, s'ha d'interrompre el circuit per a què mesure la corrent, aquest sí que té una resolució suficientment xicoteta per a mesurar les variacions del motor. A més, utilitza una comunicació I2C per a transmetre les dades.

Sensor de temperatura

En primera instància es va pensar en un LM35. No obstant es va descartar perquè s'alimentava amb 5 V. Aleshores la referència de la qual partia eren de 5 V, motiu pel qual caldria passar aquesta senyal pel level shifter, amb un convertidor de 8 bits que no és capaç de reconèixer les variacions que puga tindre el sensor, el qual té una sensibilitat de 10 mV. També existeix el sensor DHT11 i el DHT22. Finalment es va escollir aquest últim degut a que té millors característiques.

Entorn de programació

Per a poder programar els microcontroladors, existeixen diferents opcions. Entre aquestes es troba el Visual Studio Code mitjançant la extensió de PlatformIO. No obstant la opció escollida, el Arduino IDE 2.0, permet instal·lar un "plugin", que brinda la possibilitat de desxifrar el backtrace del ESP32, una ferramenta molt útil quan sorgeixen problemes i no és possible el mode depuració, i és per això que es va optar per utilitzar el tradicional entorn de programació.

Connectivitat

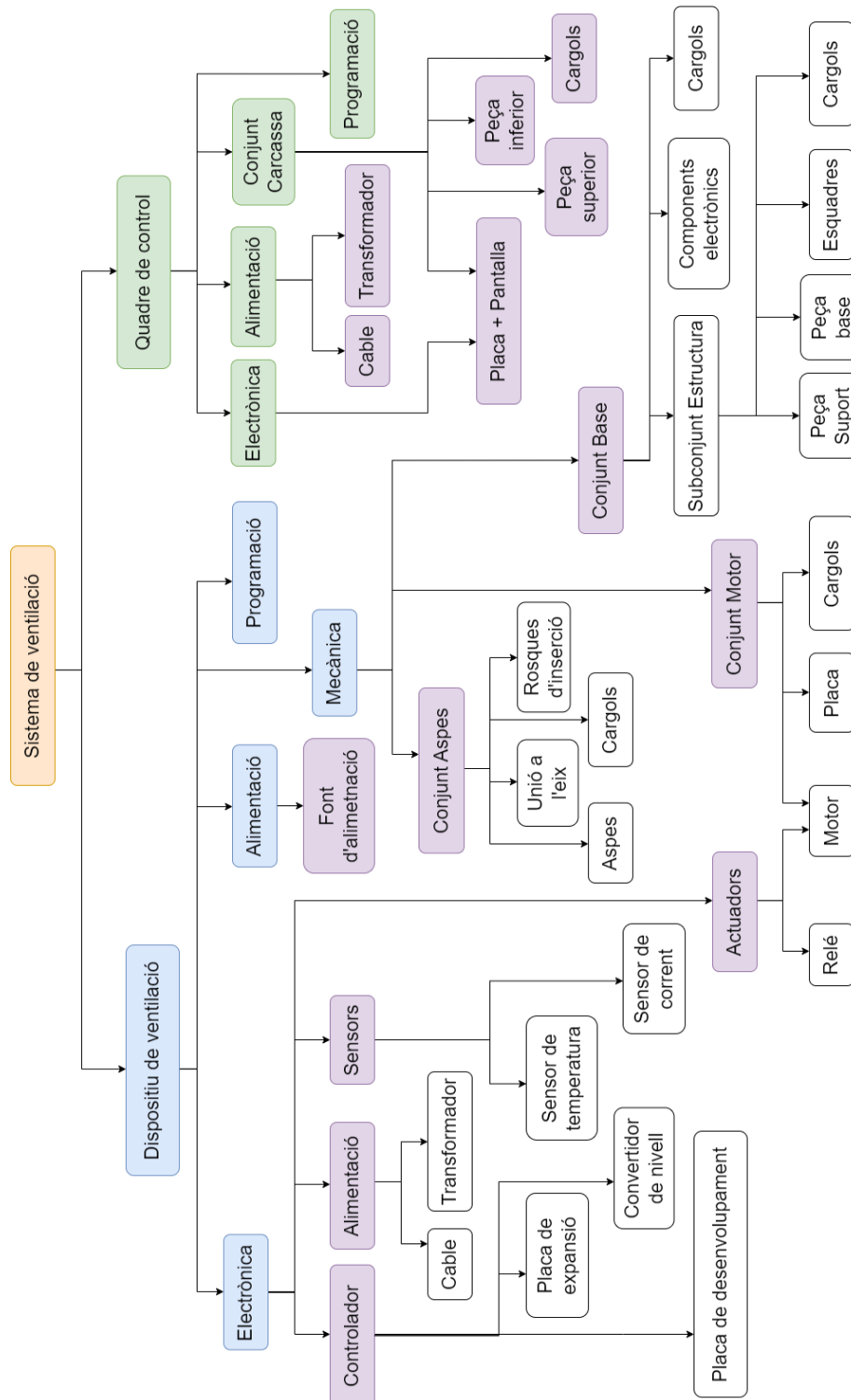
La comunicació a distància amb el controlador del motor es pot realitzar de diverses maneres. El Bluetooth és una d'aquestes, és un protocol que consumeix poca energia, i integra funcions de d'encryptació i seguretat per a l'intercanvi de dades, no obstant això, té un abast de 100 metres, la transmissió de dades és més lenta que amb les altres opcions, i té un número menor de connexions simultànies possibles. El WiFi és una excel·lent opció per a aplicacions que requereixen alta transmissió de dades i grans distàncies, a més soporta múltiples connexions simultànies, en canvi, consumeix més energia, i és necessària una infraestructura addicional per al seu funcionament. Finalment, l'ús de dispositius de la família ESP, permet l'aplicació del protocol desenvolupat per Espressif Systems per a aquests microcontroladors, el qual permet la connexió simultània de múltiples dispositius sense la necessitat de una xarxa WiFi. El protocol ESP-Now ofereix una comunicació a temps real, amb respostes amb baixa latència i un baix consum d'energia, és fàcil d'implementar i soporta un xifrat per a una comunicació segura.

Prèviament s'ha mencionat una limitació important que té aquest protocol, no obstant la comunicació es pot realitzar sense la necessitat de saber la direcció MAC, però aquest mètode funciona diferent. On el mestre envia i o rep dades a través d'un canal de comunicació, lo únic que hi ha que tindre en compte, es que el mestre i l'esclau estiguen connectats al mateix canal. No obstant, aquest mètode és més complex d'implementar i es necessiten més recursos per part del microcontrolador per a poder montar aquesta xarxa.

5. Descripció detallada de la solució adoptada

A continuació, es detallen els components del sistema de la solució adoptada. Cada component es nomena en la Figura 1, en la qual es mostra la estructura que té el sistema. En blau es troba el sistema dispositiu de ventilació i els seus subsistemes, en verd es troba el sistema quadre de control i els seus subsistemes, i en morat es troben els grups dels components que componen el sistema de ventilació.

Figura 1: Organigrama tècnic del sistema de ventilació.





5.1. Dispositiu de ventilació

5.1.1. Subsistema electrònic

5.1.1.1. Controlador

5.1.1.1.1. Placa de desenvolupament

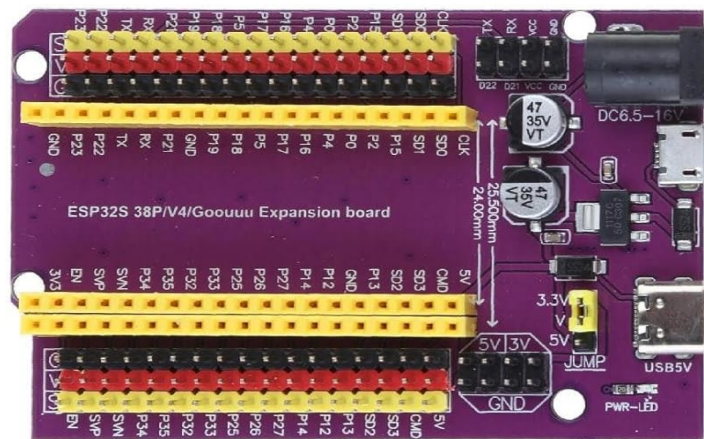
Element que s'encarrega de controlar l'estat del ventilador, rebre informació dels sensors, de temperatura i la velocitat del motor, a més de rebre la informació que es envia des del quadre de control, comandat per l'usuari. Esta tasca la porta a terme la placa de desenvolupament ESP32-WROOM-32, el xip ESP32-D0WDQ6, compta amb connectivitat WiFi i Bluetooth, dos nuclis, una freqüència del rellotge ajustable entre 80 MHz fins a 240 MHz, a més el sistema operatiu escollit per al ESP32 té integrat funcionalitats de FreeRTOS, (LWIP), TLS i acceleració de hardware, també soporta actualitzacions remotes segures (OTA), el qual permet que l'usuari actualitze el microcontrolador remotament. La placa de desenvolupament porta múltiples entrades/eixides de propòsit general (GPIO) per a una còmoda incorporació de sensors, actuadors o altres perifèrics. Les característiques més destacables es nomenen a continuació [7]:

- Connectivitat Wifi, amb suport simultani per a una infraestructura Station i SoftAP. A més de Bluetooth.
- Una microprocessador de Xtensa single/dual-core 32-bit.
- Memòria: 48KB ROM, 520 KB SRAM.
- Perifèrics:
 - o 34 GPIO programables.
 - o 12-bit SAR ADC fins a 18 canals
 - o Dos 8-bit DAC
 - o 4 canals de comunicació per SPI
 - o 2 canals de comunicació per I2S
 - o 2 canals de comunicació per I2C
 - o 3 canals de comunicació per UART
 - o 1 mestre (SD/eMMC/SDIO) i 1 esclau (SDIO/SPI)
 - o Ethernet MAC interfície, amb una pròpia DMA.
 - o Canals PWM

5.1.1.1.2. Placa d'expansió

Per a poder connectar la placa de desenvolupament més fàcilment a tots els altres components, se ha incorporat la placa d'expansió per a 38 pins per a la ESP32, com mostra la Figura 2, que permet alimentar a la placa a través de USB-C, micro-USB, a 5 V o a través d'un jack mascle d'una font d'energia entre 6,5 V o 16 V. A més posseïx connexions per a alimentar als altres components, depenent d'un jumper l'eixida d'alimentació és de 3.3 V o de 5 V, i a més té dos pins que alimenten 5 V i altres dos pins que alimenten a 3.3 V.

Figura 2: Placa d'expansió per a la placa de desenvolupament ESP32 de 38 pins.



Nota. Figura amb llicència © obtinguda de [8].

5.1.1.1.3. Convertidor de nivell

La senyal acceptada per l'etapa de potència del motor és de 0 a 5 V, però la senyal PWM del DAC del ESP32 és d'un màxim de 3,3 V. Per això s'ha introduït un convertidor de nivell, que canvia d'una tensió a un altra, permetent una major compatibilitat entre dispositius que treballen a voltatges diferents. El que s'ha escollit es un TXS108E [9].

Aquest té dues referències, VA, soporta qualsevol tensió entre 1,2 V i 3,6V, i l'altra, VB, soporta tensions entre 1,65 V i 5,5V. A més té huit canals bidireccionals, és a dir que pot transformar tant de 3,3 V a 5 V, com de 5 V a 3,3 V.

Com indica la documentació del component [9], és necessària una resistència de 10 K Ω , R1 entre els terminals VCCA i Output Enable (OE). També recomana l'ús de dos condensadors de 0.1 μ F, C1 i C2, entre els terminals VCCA i GND i entre VCCB i GND.

5.1.1.2. Alimentació

5.1.1.2.1. Cable

Per a pujar el codi es va a utilitzar un cable micro-USB a USB-A o USB-C (depèn de la entrada que puga acceptar l'ordinador), és necessari que aquest no sols done alimentació a la placa, sinó ha de poder transmetre dades per a poder comunicar-se amb l'ordinador. L'elecció d'aquest cable és degut a la compatibilitat amb una gran quantitat de dispositius quotidians, de manera que és amplament disponible i ofereix una solució pràctica i econòmica per a alimentar el dispositiu.

Cal mencionar que el cable d'alimentació per a alimentar la placa d'expansió també pot ser de tipus USB-C, o un cable jack mascle alimentat per una font d'energia, però sols per a la alimentació, per a pujar el codi a la placa de desenvolupament ha de ser micro-USB.



5.1.1.2.2. Transformador

La font d'energia escollida per a alimentar la placa d'expansió és un transformador AC/DC de 5 V, ja que segons la placa d'expansió l'alimentació mitjançant USB-C es de 5 V, com s'observa en la Figura 2.

5.1.1.3. Sensors

5.1.1.3.1. Sensor de temperatura

Per a captar la temperatura de l'ambient s'ha escollit el sensor de temperatura DHT22, a més aquest, permet mesurar la humitat. Utilitza un termistor per a mesurar la temperatura i un sensor capacitiu per a la humitat, oferint dades digitals i fiables. Les característiques més rellevants són nomenades a continuació [10]:

- Temperatura:
 - o Rang de -40 a 125 °C,
 - o Precisió $\pm 0,5$ °C
 - o Resolució de 0,1 °C
- Humitat:
 - o Rang de 0 a 100 % de humitat relativa (HR)
 - o Precisió $\pm 2\%$ HR
 - o Resolució 0,1% HR
- Freqüència de mostra 2 Hz (2 mostres per segon)

5.1.1.3.2. Sensor de corrent

Per a poder mesurar la corrent que consumix el motor s'ha escollit el sensor de corrent INA 219 [11]. Aquest mòdul permet monitoritzar la corrent en tems real a través del protocol I2C. Característiques més rellevants a continuació :

- Corrent: $\pm 3,2$ A
- Resolució: 0,1 mA
- Tensió d'alimentació: 3,3 V
- Tensió de treball: 0 V a 26 V

5.1.1.4. Actuadors

5.1.1.4.1. Relé

S'ha incorporat un relé per a controlar l'alimentació que va directa al motor, aleshores en cas d'emergència es talla la connexió del motor per a garantir una major seguretat. El que s'ha escollit és el Relay ISO9002 [12], el qual s'alimenta amb 5 V, i soporta una tensió contínua de 30 V a un màxim de 5 A. Té dos possibles modes, normalment obert (NO), i normalment tancat (NC), i la commutació es pot realitzar amb una entrada digital en alt (H) o en baix (L).



5.1.1.4.2. Motor

El motor escollit és un motor de corrent continua brushless [13], amb un reductor de 1:19, on la velocitat màxima es de 421 revolucions per minut (RPM), i és capaç de girar en els dos sentits. L'etapa de potència ja està incorporada, de manera que la senyal de velocitat sols depèn d'una senyal analògica de entre 0 i 5V. També porta un encoder, el qual genera 6 polsos per cada volta que dona l'eix principal. L'eix d'eixida té forma de D, aquesta augmenta la força de fricció i facilita la connexió i fixació de l'aspa.

5.1.1.5. Connexions

5.1.1.5.1. Cables

Cal mencionar que la connexió entre els diferents elements del sistema es realitzarà mitjançant cables de pont jumper, amb pins M2M i de tipus mascle a mascle (M/M), mascle a femella (M/F) i femella a femella (F/F), els quals permeten una fàcil, ràpida i no permanent unió de cada un dels components.

5.1.1.5.2. Protoboard

Es va a utilitzar una protoboard per a realitzar el muntatge final de la maqueta. Permet la inserció i extracció de components electrònics i cables sense la necessitat de ferramentes addicionals, facilitant l'experimentació i modificació dels circuits. Està compostat per una matriu de forats connectats internament de una determinada manera, el que permet crear connexions elèctriques simplement insertant els components en els porats corresponents. Aquesta manera de funcionament és molt versàtil ja que permet canviar el disseny de l'esquema elèctric en un moment, és reutilitzable i fàcil d'utilitzar.

5.1.2. Subsistema d'alimentació

5.1.2.1. Font d'alimentació

La font d'alimentació ha sigut escollida en base al motor, el qual necessita 24V per a funcionar, i una corrent de menys de 2 amperes com a màxim. De manera que la font escollida és de 24 V i 5 A [14].

Cal mencionar, que per a donar energia a la font d'alimentació és necessari un cable de corrent alterna, com per exemple el cable d'alimentació estàndard europeu (Tipus F), amb dues terminals és suficient, no obstant, és recomanable que tinga tres, línia, neutre i presa a terra, per més seguretat tant de l'usuari com de l'equip.

5.1.3. Subsistema mecànic

5.1.3.1. *Aspes*

Per a les aspes del prototip es va a imprimir un model escollit d'internet, mostrat en la Figura 3. El material del que s'ha construït l'aspa és àcid polilàctic o PLA, és un material biodegradable, ja que es fabrica a partir de recursos naturals com la dacsca, la canya de sucre o altres materials rics en midó, a més és un material fàcil d'imprimir, ja que la temperatura d'extrusió és relativament baixa (de entre 190 i 220 °C) i a més és molt econòmic.

Figura 3: Model 3D de les aspes del prototip.

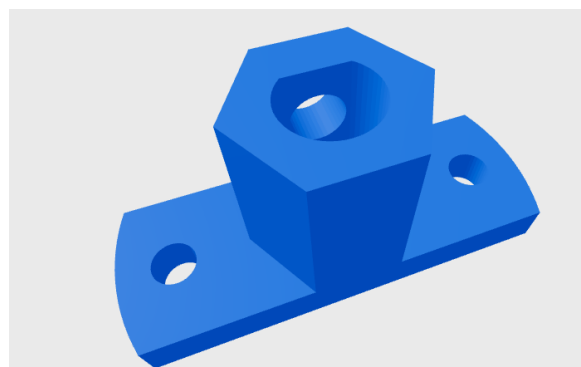


Nota. Figura amb llicència CC obtinguda de [15]

5.1.3.2. *Unió a l'eix*

Per a unir les aspes i l'eix s'ha imprès la peça que es mostra en la Figura 4. S'ha imprès del mateix material que les aspes, PLA.

Figura 4: Model 3D de la peça d'unió entre les aspes i l'eix.



Nota. Figura amb llicència CC obtinguda de [15]



5.1.3.3. Cargols

Per a la unió de les peces és necessari cargols M3x5, en aquest cas el cap serà hexagonal.

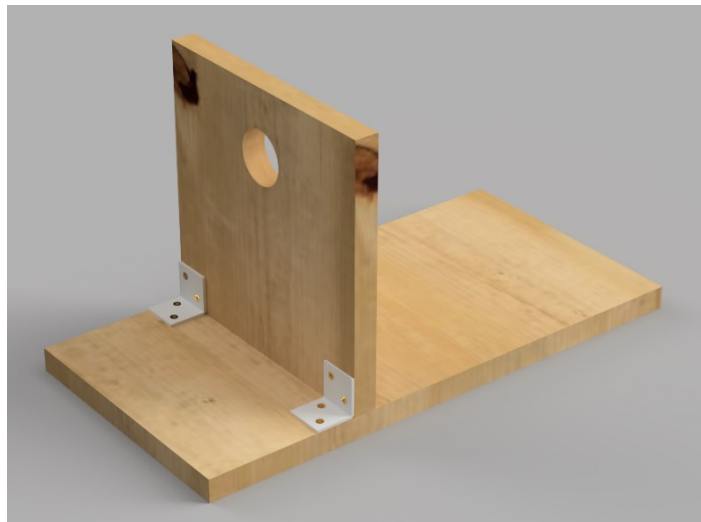
5.1.3.4. Rosques d'inserció

Per a assegurar una unió robusta i precisa de les peces s'ha optat per l'ús de rosques d'inserció de llautó, han de ser compatibles amb cargols de 3 mm de diàmetre.

5.1.3.5. Base

Per a la construcció de la maqueta es pretén construir una base per a albergar tots els components. L'estructura es té intenció de que tinga una forma similar a la de la Figura 5.

Figura 5: Model en 3D de l'estructura base de la maqueta.



5.1.3.5.1. Fusta

Per a l'estructura de la maqueta s'ha seleccionat una fusta d'abet massís, de mesura 200x600x18 mm.

5.1.3.5.2. Esquadres

Per a subjectar l'estructura són necessàries dues esquadres, en aquest cas s'han escollit l'esquadra ampla blanca 30x30x30 mm.



5.1.3.5.3. Cargols

Els cargols per a fixar les esquadres a la estructura han de ser d'un màxim de 3 mm de diàmetre, preferentment per a fusta i amb una longitud no superior al gruix de la fusta, és a dir 18 mm. De manera que s'ha escollit un M3x15 per a fusta amb el cap de punta d'estrela.

5.1.4. Subsistema de programació

El subsistema de programació abasta tot el software necessari per a controlar el sistema de ventilació, incloent la captura de dades dels sensors, la presa de decisions basades en aquestes dades, el control dels actuadors i la comunicació remota.

Com s'ha nomenat abans, per a la programació, la compilació i la pujada del codi a la targeta es va a utilitzar el Arduino IDE 2.0, basat en una variant de C/C++. Aquest és un entorn de desenvolupament integrat de codi obert, amb una interfície d'usuari, intuïtiva, fàcil i ràpida d'utilitzar, compatible amb moltes plaques de desenvolupament i llibreries. A més compta amb una vasta comunitat d'usuaris i desenvolupadors que proporcionen suport i solucions a problemes comuns.

Per al control i l'adquisició de dades del sensor de temperatura es va a utilitzar la llibreria "DHT22.h". Aquesta biblioteca es vastament reconeguda per la seua fiabilitat i precisió de lectura de dades de sensors DHT. La seua integració permet la comunicació eficient amb el microcontrolador, garantit la exactitud en les dades obtingudes i el seu posterior processament.

En el cas del sensor de corrent, s'opta per la llibreria "Adafruit_INA219.h". Aquesta facilita el control del sensor INA219 i assegura que les lectures siguen precises i consistents, la qual cosa es crucial per al monitoratge del consum energètic del motor.

Per a establir una comunicació a distància efectiva entre el controlador i el quadre de comandaments, s'empraran les biblioteques "esp_now.h" i "WiFi.h". La biblioteca "esp_now.h" s'utilitza per a la configuració ràpida i directa entre dispositius ESP32, permetent una transmissió de dades eficient i de baixa latència. Per un altre costat, la llibreria "WiFi.h" s'integra per a habilitar la connectivitat a xarxes WiFi, i que el ESP32 es configure com una estació i pugui connectar-se a un *access point*.

5.2. Quadre de control

5.2.1. Subsistema electrònic

5.2.1.1. Placa + Pantalla

Per a la interacció amb l'usuari s'ha decidit utilitzar una pantalla tàctil, el mòdul ESP32-2432S028 mostrat en la Figura 6, és un mòdul compacte i potent que combina el microcontrolador ESP32 amb una pantalla tàctil resistiva de 2.8 polsades, aquest dispositiu ofereix una solució integrada per a projectes que requereixen d'una interfície d'usuari, facilitant la interacció amb el sistema i visualitzant les dades en temps real. Les característiques més importants són les següents:

- Resolució: 320x240 píxels

- Tipus de pantalla: TFT RGB 65k
- Xip del controlador: IL9341
- Tipus de tacte: resistiu

Figura 6: Mòdul ESP32-2432S028.



Nota. Figura amb llicència © obtinguda de [16].

5.2.2. Subsistema d'alimentació

5.2.2.1. Cable

Per a pujar el codi es va a utilitzar un cable micro-USB a USB-A o USB-C (depèn de l'entrada que puga acceptar l'ordinador)

Per a alimentar la placa i la pantalla i pujar el codi es necessita un cable micro-USB a USB-A o USB-C (depèn de l'entrada que puga acceptar l'ordinador), com s'ha mencionat abans el cable ha de ser de transmissió de dades i no únicament d'alimentació, al menys a l'hora de pujar el codi. No obstant, a diferència del cable per a alimentar a la placa d'expansió del controlador, aquest sols pot ser micro-USB.

5.2.2.2. Transformador

La font d'energia ha de ser de 5 V, segons la documentació tècnica de la placa per al voltatge d'alimentació [16].

5.2.3. Subsistema mecànic

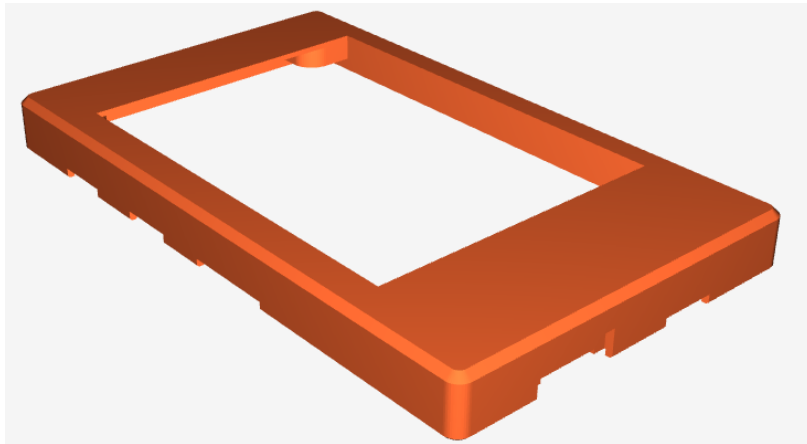
5.2.3.1. Carcassa

Per a encapsular la pantalla i protegir els components de la pols i altres agents, s'ha escollit un model d'internet per a imprimir en una impressora 3D amb PLA. La integració de la carcassa no sols millora la estètica del dispositiu, sinó que assegura la integritat i la durabilitat dels components interns, ampliant així la seua vida útil.

5.2.3.1.1. Peça superior

La part superior de la pantalla es mostra en la Figura 7.

Figura 7: Model 3D de la peça superior de la carcassa que protegix el quadre de control.

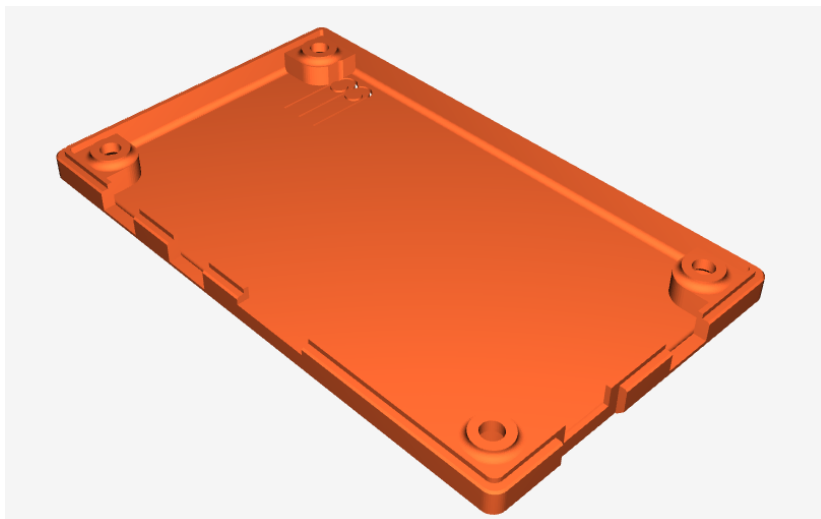


Nota. Figura amb llicència CC obtinguda de [17]

5.2.3.1.2. Peça inferior

La part inferior de la carcassa es mostra en la Figura 8.

Figura 8: Model 3D de la peça inferior de la carcassa que protegix el quadre de control.



Nota. Figura amb llicència CC obtinguda de [17]

5.2.3.1.3. Cargols

Com indica el creador del disseny, per a unir totes les peces es necessiten quatre cargols de M3x6 mm.



5.2.4. Subsistema de programació

Per a la programació del quadre de comandaments, també es va a utilitzar l'Arduino IDE 2.0.

Per mostrar imatges per la pantalla es van a utilitzar les llibreries "SPI.h" i "TFT_eSPI.h" i per a la gestió de la pantalla tàctil es va a utilitzar la llibreria "XPT2046_Touchscreen.h".

Com s'ha explicat prèviament, la comunicació remota entre el controlador i el quadre de comandaments s'implementarà mitjançant les biblioteques "esp_now.h" i "WiFi.h".

6. Justificació detallada dels elements o components de la solució adoptada (càlcul i dimensionament)

En aquesta secció s'aborda la justificació detallada de la solució adoptada, on es desenvolupa el procés seguit per a la realització d'aquest projecte. És un apartat crucial que permet comprendre les decisions preses durant tot el procés, ja que s'expliquen i argumenten les raons darrere de cada element que conforma el projecte, la forma en la qual s'integren en el sistema i com contribueixen al resultat final.

Per a començar es va a comentar l'estructura que va seguir aquest apartat, per poder proporcionar una guia clara i detallada de les seccions d'aquest punt. En primer lloc, es mostra el diagrama de connexions, que proporciona una visió clara de la connexió entre els diversos components del sistema. A continuació, s'explica detalladament el funcionament del programa que controla el dispositiu de ventilació, descrivint les seues funcionalitats i la lògica de programació utilitzada. Finalment, es procedirà a exposar el programa del panell de control, explicant la interacció amb l'usuari i la gestió de les operacions del sistema.

6.1. Diagrama de connexions

El diagrama de connexions és una eina fonamental per a la comprensió del sistema, ja que proporciona una visió clara i detallada de com s'interconnecten els diversos components electrònics. Aquest diagrama és essencial per a garantir la implementació correcta i per a la detecció de possibles errors en les connexions.

En la Taula 1 es mostra un esquema de les connexions de cada component, de manera que es proporciona una vista organitzada i clara de cada connexió, descrivint la relació entre els pins de cada component i les seues connexions corresponents, així com una breu descripció de la funció de cada connexió.

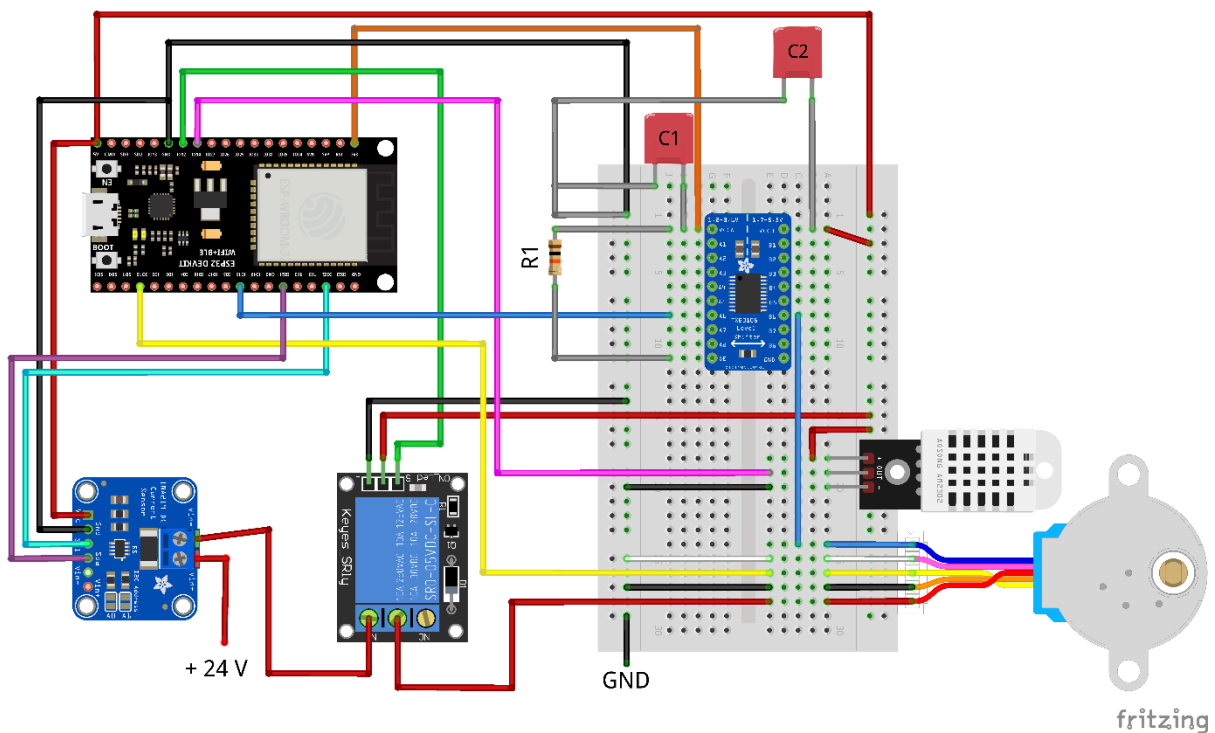
La Figura 9 representa de manera visual les connexions del necessàries per al funcionament correcte del sistema de ventilació. Cada línia de connexió és identificada amb colors diferents per a facilitar la seua comprensió.

Taula 1: Connexions dels components del dispositiu de ventilació.

Component	Pin	Connexió a	Descripció
ESP32	GPIO 12	IN – Relé	Control del relé
	GPIO 14	Data – DHT22	Lector de temperatura
	GPIO 15	Pin groc - Motor	Encoder motor
	GPIO 18	A6 – TXB0108	Commutar el voltatge de eixida del ESP32
	GPIO 21	SDA – INA219	Comunicació I2C de dades
	GPIO 22	SCL – INA219	Comunicació I2C del rellotge
	3V3	VCCA – TXB0108	Senyal A del convertidor de nivell
	3V3	C1	Condensador de 0.1 uF
	3V3	R1	Resistència de 10K Ω
	5V	Vcc – DHT22	Alimentació del sensor de temperatura
	5V	Vcc – INA219	Alimentació del sensor de corrent
	5V	VCCB – TXB0108	Alimentació B del convertidor de nivell
	5V	C2	Condensador de 0.1 uF
	5V	Vcc – relé	Alimentació del relé
	GND	Negatiu comú	Referència
INA219	Vin+	+ 24 V – Font d'alimentació	Mesura de corrent del motor
	Vin-	NO – Relé	Mesura de corrent del motor
	SDA	GPIO 21 – ESP32	Comunicació I2C de dades
	SCL	GPIO 22 – ESP32	Comunicació I2C del rellotge
	Vcc	+5V	Alimentació del sensor de corrent
	GND	Negatiu comú	Referència
Relé ISO9002	DC+	+5 V	Alimentació del relé
	DC-	Negatiu comú	Referència
	IN	GPIO 12 – ESP32	Control del relé
	C	Vin- – INA219	Control del motor
	NO	Vcc - Motor	Control del motor
TXB0108	VCCA	3V3	Referència A del ESP32
	VCCA	C1	Condensador de 0.1 uF
	VCCA	R1	Resistència de 10K Ω
	VCCB	+5V	Referència B del ESP32
	VCCB	C2	Condensador de 0.1 uF
	OE	R1	Output Enable
	GND	Negatiu comú	Referència
	A6	GPIO 18 – ESP32	Senyal A del convertidor de nivell
	B6	Pin blau	Senyal de velocitat del motor

Component	Pin	Connexió a	Descripció
DHT22	Vcc (+)	+5V	Alimentació del sensor
	OUT	GPIO 14 – ESP32	Transmissió de la temperatura
	GND (-)	Negatiu comú	Referència
Motor	Pin roig	C - Relé	Alimentació del motor
	Pin negre	Negatiu comú	Referència
	Pin blau	B6 – TXB0108	Senyal de velocitat
	Pin groc	GPIO 15 – ESP32	Encoder del motor
	Pin blanc	Negatiu comú	Control del sentit de gir
R1	Born 1	VCCA – TXB0108	
	Born 2	OE – TXB0108	
C1	Born 1	VCCA – TXB0108	
	Born 2	Negatiu comú	
C2	Born 1	VCCB – TXB0108	
	Born 2	Negatiu comú	
Font d'alimentació	V+	Vin+ – INA219	
	V-	Negatiu comú	
	L	Línia	
	N	Neutre	
	Pressa a terra	Pressa a terra	

Figura 9: Diagrama de connexions del dispositiu de ventilació





6.2. Dispositiu de ventilació

El dispositiu de ventilació es la part fonamental del sistema, ja que fa possible la circulació de l'aire i el manteniment de les condicions ambientals. Aquest, ha sigut dissenyat per a operar com un sistema de temps real (RTS, per les seues sigles en anglès). Aquest enfocament permet que el sistema gestione simultàniament diverses tasques, assegurant una gestió eficient del sistema per a garantir una resposta ràpida i precisa a les condicions ambientals canviants. Per a la implementació d'aquest RTS s'ha fet ús de FreeRTOS [18], un sistema operatiu de temps real de codi obert, que proporciona les eines necessàries per a la gestió de tasques, sincronització i comunicació entre processos. Per a acomplir la implementació del sistema de temps real s'han seguit els següents punts claus:

- I. Divisió de tasques. S'han identificat les tasques principals que el sistema ha de gestionar. Aquestes inclouen la lectura de l'encoder del motor, la lectura del sensor de temperatura i la del sensor de corrent, el control de l'encès i apagat del motor, el control del motor en mode automàtic i en mode manual i la comunicació remota amb el panell de control.
- II. Planificació i prioritització: a cada tasca ha estat assignada una prioritat basada en la seua importància i la seua freqüència d'execució.
- III. Multitasca. El RTS està configurat per executar diverses tasques en paral·lel mitjançant FreeRTOS. El qual està gestionat pel planificador que distribueix els recursos del microcontrolador entre les tasques segons les seues prioritats.
- IV. Gestió d'interrupció i sincronització entre tasques. S'ha configurat una interrupció per a l'encoder del motor, en la qual es notifica a la tasca que llig la velocitat del motor.
- V. Protecció de variables. Per assegurar la coherència de les dades i evitar la condició de carrera, s'ha implementat el mètode del semàfor proporcionat per FreeRTOS per protegir les variables compartides. De mode que es garanteix que les tasques accedeixen a les mateixes dades no interferisquen entre elles, mantenint la integritat del sistema.

El controlador, per a comunicar-se amb el panell de control utilitza el protocol ESP-Now [19], mitjançant la llibreria `esp_now.h` i `WiFi.h`. Com s'ha comentat prèviament, una limitació que té l'ús d'aquest mitjà, és la necessitat de saber la direcció MAC abans d'establir la primera connexió. En la Figura 10, es mostra un codi exemple que s'ha de pujar al microcontrolador per a saber la seua direcció MAC. El resultat d'aquesta operació es pot veure en la Figura 11, la direcció MAC és un identificador, únic per a cada dispositiu, de 48 bits, representats en sis blocs de dos caràcters en hexadecimal, aquest número ha de ser introduït en el dispositiu anàleg amb el que es vol comunicar.

Figura 10: Codi exemple per a obtenir la direcció MAC del ESP32.

```
#include <WiFi.h>

void setup() {
  // put your setup code here, to run once:

  Serial.begin(115200);
  // Set device as a Wi-Fi Station
  WiFi.mode(WIFI_STA);

  while (!WiFi.STA.started()) {
    delay(100);
  }

  Serial.print(F("Transceiver initialized : "));
  Serial.println(WiFi.macAddress());
}
```

Figura 11: Direcció MAC d'un ESP32.

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1448
load:0x40078000,len:14844
ho 0 tail 12 room 4
load:0x40080400,len:4
load:0x40080404,len:3356
entry 0x4008059c
Transceiver initialized : 48:E7:29:8C:5A:AC
```

Per a poder obtindre les dades dels sensors de temperatura i corrent, es van a utilitzar les llibreries DHT22.h i Adafruit_INA219.h, respectivament.

Una vegada descrita la base del funcionament del controlador, es va a explicar l'estructura que segueix el programa. Primerament es van a explicar les macros, objectes, constants i variables utilitzades, a continuació, es va a explicar com s'accedix a les dades globals, posteriorment es van a descriure les tasques i es van a mostrar els seus diagrames de flux, finalment es va a comentar el controlador utilitzat.

6.2.1. Definició de macros, objectes, constants i variables globals

Macros

Per a definir els diferents pins de connexió s'han utilitzat macros, com es pot veure en la Figura 12, on "temperatureSensorPin" correspon amb el pin data del sensor de temperatura, "relePin" amb el del relé, "yellowPin" amb el de l'encoder del motor i "bluePin" correspon amb el pin de la senyal que regula la velocitat del motor.

Figura 12: Definició de les macros del programa, associades a sensors i actuadors del sistema.

```
#define temperatureSensorPin 14 // pin del sensor de temperatura
#define relePin 13 // pin del relé
#define yellowPin 15 // pin del encoder del motor
#define bluePin 18 // pin a la senyal de velocitat del motor
```

Adicionalment, s'ha de modificar un fitxer, per a poder utilitzar la `vTaskDelayUntil()`. Segons la documentació del FreeRTOS [18], s'ha de definir la macros `INCLUDE_vTaskDelayUntil` a 1, en el fitxer `FreeRTOSConfig.h`, ubicat en la següent localització "C:\Users\usuari\Documents\Arduino\hardware\espressif\esp32\tools\esp32-arduino-libs\esp32s3\include\freertos\esp_additions\include\freertos" (substituir la paraula "usuari", per l'usuari real), com es mostra en la Figura 13, on en la línia 187 s'ha inclòs la macro.

Figura 13: Modificació del document `FreeRTOSConfig.h`, per a poder utilitzar la funció `vTaskDelayUntil()`.

```
C FreeRTOSConfig.h x
C: > Users > julia > Documents > Arduino > hardware > espressif > esp32 > tools > esp32-arduino-libs > esp32s3 > include > freertos > esp_additions > include > freertos > C FreeRTOSConfig.h
173
174 /* ----- API Includes ----- */
175
176 #if CONFIG_FREERTOS_ENABLE_BACKWARD_COMPATIBILITY
177 #define configENABLE_BACKWARD_COMPATIBILITY 1
178 #else /* CONFIG_FREERTOS_ENABLE_BACKWARD_COMPATIBILITY */
179 #define configENABLE_BACKWARD_COMPATIBILITY 0
180 #endif /* CONFIG_FREERTOS_ENABLE_BACKWARD_COMPATIBILITY */
181
182 #define INCLUDE_vTaskPrioritySet 1
183 #define INCLUDE_uxTaskPriorityGet 1
184 #define INCLUDE_vTaskDelete 1
185 #define INCLUDE_vTaskSuspend 1
186 #define INCLUDE_vTaskDelay 1
187 #define INCLUDE_vTaskDelayUntil 1
188 #define INCLUDE_xTaskGetIdleTaskHandle 1
189 #define INCLUDE_xTaskAbortDelay 1
190 #define INCLUDE_xSemaphoreGetMutexHolder 1
191 #define INCLUDE_xTaskGetHandle 1
```

Objectes

Per a la lectura dels sensors, a través de les llibreries nomenades prèviament, és necessari la creació d'objectes. En la Figura 14 es crea l'objecte `dht22`, de tipus `DHT22`, on el paràmetre que se li passa és el pin associat al sensor. També es crea l'objecte `ina219` de tipus `Adafruit_INA219`, aquest no se li passa cap paràmetre, ja que el sensor es connecta per I2C amb una adreça predeterminada, de mode que no és necessari indicar-la.

Figura 14: Creació dels objectes que permetran l'accés a les dades obtingudes pels sensors.

```
DHT22 dht22(temperatureSensorPin);
Adafruit_INA219 ina219;
```

Constants

Les constats definides per al funcionament del codi son les mostrades en la Figura 15. Explicades a continuació:

- "broadcasAddress" és l'adreça del ESP que controla la pantalla.
- "velocitatMax" és la velocitat màxima establerta que pot assolir el motor.
- "correntMax" és la corrent màxima permesa en el motor, obtinguda de les característiques del motor [13].
- "queueLength" és el número màxim d'elements en una cua, ha establert en 1, per a que siga una cua d'un element, de manera que si s'actualitza una dada, l'element de la cua siga actualitzat pel nou.
- "poulsesPerRevolution" és la quantitat de polsos que genera el eix del motor principal al donar una volta completa [13].
- "reduction_ratio" és el factor de conversió del reductor que porta el motor, l'eix reduït dona 1 volta cada 19 girs que fa l'eix principal [13].
- "xFrequency_taskReadMotorSpeed" és el període de la tasca que llig la velocitat del motor.
- "xFrequency_taskReadTemperature" és el període de la tasca que llig la temperatura.
- "xFrequency_taskReadCurrent" és el període de la tasca que llig la corrent.

- “xFrequency_taskSartAndStop” és el període de la tasca que controla la parada i l’encés del motor.
- “xFrequency_taskSendData” és el període de la tasca que s’encarrega de enviar les dades a panell de control.

Figura 15: Constants globals del programa del controlador.

```
// adreça MAC del ESP que controla la pantalla
const uint8_t broadcastAddress[] = { 0x10, 0x06, 0x1C, 0x81, 0xDF, 0xA0 };

//velocitat màxima del motor
const float velocitatMax = 300.0;
const float correntMax = 1000.0; // mA

// màxim d'elements que pot tindre una cua
const int queueLength = 1;

// constants per al càlcul de la velocitat del eix reduït
const float pulsesPerRevolution = 6.0;
const float reduction_ratio = 19.0;

// periodos de las tasques
const TickType_t xFrequency_taskReadMotorSpeed = 1000;
const TickType_t xFrequency_taskReadTemperature = 1000;
const TickType_t xFrequency_taskReadCurrent = 300;
const TickType_t xFrequency_taskSartAndStop = 100;
const TickType_t xFrequency_taskSendData = 200;
```

Variables

En la Figura 16, es presenten les variables que permeten el control del programa i el control de la velocitat del motor.

- “estatActual” representa l’estat actual del motor, 1 és encès i 0 és apagat.
- “estatDesitjat” representa l’estat desitjat del motor, 1 és encès i 0 és apagat.
- “modeActual” representa el mode de funcionament actual del motor, 1 és mode manual i 2 mode automàtic.
- “modeDesitjat” representa el mode de funcionament desitjat del motor, 1 és mode manual i 2 mode automàtic.
- “velocitatActual” representa la velocitat actual del motor.
- “velocitatDesitjada” representa la velocitat que es vol donar al motor.
- “senyalVelocitat” és la velocitat que és dona al motor.
- “temperaturaActual” representa la temperatura actual de l’ambient.
- “temperaturaDesitjada” representa la temperatura desitjada.
- “correntActual” és la corrent que té el motor.
- “missatgeError” representa un error sorgit en l’execució del programa, si es 0, no hi ha cap error en el sistema, si és 1 hi ha excés de corrent pel motor i si es 2 ha hagut una parada inesperada del motor.

Figura 16: Variables globals del control del flux del programa i del control de la velocitat del motor.

```
byte estatActual = 0;
byte estatDesitjat = 0;

byte modeActual = 0;
byte modeDesitjat = 0;

float velocitatActual = 0.0;
float velocitatDesitjada = 100;
float senyalVelocitat = 0;

float temperaturaActual = 0.0;
float temperaturaDesitjada = 25.6;

float correntActual = 0;

byte missatgeError = 0;
```

En la Figura 17 apareixen les variables que permeten que el programa tinga tasques, cues i semàfors. Aquelles de tipus “TaskHandle_t” permeten controlar una tasca i aquelles de tipus “SemaphoreHandle_t” permeten accedir al semàfor que protegeix una secció de codi, en aquest cas, l'accés a una variable.

Figura 17: Variables de per a les tasques, les cues i els semàfors.

```
// taskHandle
TaskHandle_t taskReadMotorSpeedHandle;
TaskHandle_t taskReadTemperatureHandle;
TaskHandle_t taskAddValueSpeedHandle;
TaskHandle_t taskSpeedControlManualHandle;
TaskHandle_t taskSpeedControlAutoHandle;
TaskHandle_t taskSartAndStopHandle;
TaskHandle_t taskDelayUntilHandle;

// queueHandle
QueueHandle_t queueDataFromControllerHandler;
QueueHandle_t queueDataFromDisplayHandler;

// semàfors
SemaphoreHandle_t semaphoreModeActual = NULL;
SemaphoreHandle_t semaphoreTemperaturaActual = NULL;
SemaphoreHandle_t semaphoreVelocitatActual = NULL;
SemaphoreHandle_t semaphoreEstatActual = NULL;
SemaphoreHandle_t semaphoreCorrentActual = NULL;
SemaphoreHandle_t semaphoreMissatgeErrorActual = NULL;
SemaphoreHandle_t semaphoreModeDessitjat = NULL;
SemaphoreHandle_t semaphoreTemperaturaDessitjada = NULL;
SemaphoreHandle_t semaphoreVelocitatDessitjada = NULL;
SemaphoreHandle_t semaphoreEstatDessitjat = NULL;
SemaphoreHandle_t semaphoreSenyalVelocitat = NULL;
```

Per a enviar dades a través del protocol Esp-Now era necessari l'ús de variables de tipus struct, que empaqueta diversos tipus de dades en una sola variable, de manera que s'han creat dos tipus d'estructures, mostrades en la Figura 18:

- "struct_dataFromController" conté les dades que s'enviaran al quadre de control, aquest conté el mode actual, la temperatura actual, la velocitat actual, l'estat actual, la corrent del motor i un missatge d'error.
- "struct_dataFromDisplay" conté les dades que es rebran del quadre de control, aquest alberga el mode desitjat, la temperatura desitjada, la velocitat desitjada, l'estat desitjat i la resposta del missatge d'error, per a què desaparega.

Figura 18: Creació d'una estructura de dades de tipus struct.

```
typedef struct struct_dataFromController {  
  
    byte modeActual;  
    float temperaturaActual;  
    float velocitatActual;  
    byte estatActual;  
    float correntActual;  
    byte missatgeError;  
};  
  
typedef struct struct_dataFromDisplay {  
  
    byte modeDesitjat;  
    float temperaturaDesitjada;  
    float velocitatDesitjada;  
    byte estatDesitjat;  
    byte missatgeError;  
};
```

6.2.2. Accés a les dades globals

L'accés d'aquestes ha sigut implementat mitjançant un semàfor, ja que el sistema FreeRTOS permet que sols una tasca a la vegada obtinga les dades.

Un semàfor és un mètode que s'utilitza per a protegir una zona de codi, comunament es protegeix un recurs compartit, de manera que si una tasca necessita aquest recurs, però una altra l'està manipulant, no pot accedir a ell fins que l'altra finalitzi. Per a accedir al recurs primer s'ha d'agafar el mutex, mitjançant xSemaphoreTake(), si el mutex està disponible, la tasca l'agafa i procedeix a accedir al recurs, posteriorment retorna el mutex mitjançant xSemaphoreGive(). En cas contrari, si no està disponible el mutex, la tasca es bloqueja fins que estiga disponible o s'esgoti el temps d'espera especificat.



Llegir i escriure en variables globals a través d'un semàfor és una tasca que consumeix temps i recursos, de manera que s'han creat variables locals, per a què sols es llegeixen una vegada les variables globals per cada iteració del bucle.

En el Figura 19 es presenta el diagrama de flux que segueix el programa en intentar llegir una variable global, i en la Figura 20, es mostra en forma de codi la implementació de la lectura a les variables globals, en aquest cas de la variable "modeActual". En accedir a la funció readmodeActual(), primer es crea la variable de tipus byte, anomenada "mode", igualada a NAN (Not A Number, en anglès), s'inicialitza d'aquesta manera per a què a l'hora de llegir la variable, si ha sorgit algun problema, no es modifique la variable local i no done pas a mal funcionaments en cas de retornar un número anòmal. Després d'inicialitzar la variable, es comprova que el semàfor corresponent amb la variable desitjada existeix, si no existeix directament imprimeix per pantalla un missatge indicant que el semàfor no existeix, i retorna "mode", en aquest cas, com no ha sigut modificada, tornarà NAN. De manera contraria, si sí que existeix el semàfor seguidament, intenta accedir al mutex de la variable "modeActual", mitjançant la funció xSemaphoreTake(), si està disponible agafa el mutex, guarda la variable "modeActual" en la variable local "mode" i torna el mutex, a través de la funció xSemaphoreGive(), posteriorment, retorna la variable "mode". Si pel cas contrari el mutex no està disponible, s'espera vint tics de rellotge i torna a preguntar si està disponible, si aquesta està disponible, agafa el mutex i procedeix a modificar la variable i seguidament torna el mutex. No obstant, si la segona vegada que intenta agafar el mutex però no pot, envia un missatge a la terminal, comunicant que no ha sigut possible llegir la variable "modeActual", i posteriorment torna la variable "mode" sense modificar.

En la Figura 21, es mostra el diagrama de flux que segueix la funció writeModeActual(), i en la Figura 22, hi ha un exemple de la programació de la funció que permet modificar d'una variable global, "modeActual". A aquesta funció, el valor nou es passa com un argument de la funció, de manera que no és necessari definir-la a l'inici, i a més, no retorna cap valor. Posteriorment, de la mateixa manera que en readModeActual(), es comprova que el semàfor existisca, si no existeix s'envia un missatge per pantalla indicant que el semàfor no existeix. Oposadament, si sí que existeix, intenta accedir al mutex de la variable, si aquest està lliure, l'agafa, a continuació modifica la variable "modeActual" per el valor que hi ha en "mode", i posteriorment retorna el mutex. Si pel contrari no està lliure, s'espera vint tics de rellotge i seguidament ho torna a intentar, si està lliure, pot accedir a la modificació de la variable y després retorna el mutex. No obstant, si la segona vegada que intenta accedir al mutex també està ocupat, no accedeix a la variable i imprimeix per pantalla un missatge indicant que la variable no ha pogut ser modificada.

Figura 19: Diagrama de flux de la funció readModeActual().

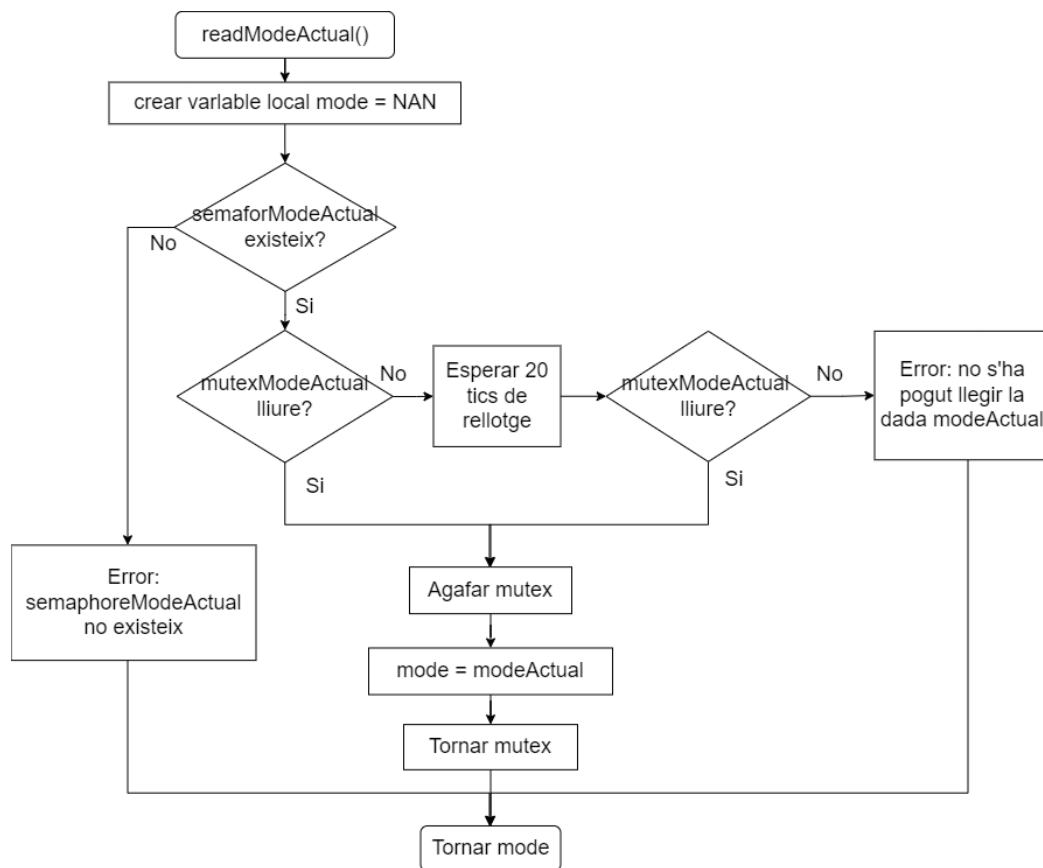


Figura 20: Codi exemple de lectura de la variable global "modeActual" a través del mètode del semàfor.

```

byte readModeActual() {
    byte mode = NAN;

    if (semaphoreModeActual != NULL) {
        if (xSemaphoreTake(semaphoreModeActual, (TickType_t)20) == pdTRUE) {
            mode = modeActual;
            xSemaphoreGive(semaphoreModeActual);
        } else {
            Serial.println("No se ha pogut llegir la dada modeActual");
        }
    } else {
        Serial.println("semaphoreModeActual == NULL");
    }
    return mode;
}
  
```

Figura 21: Diagrama de flux de la funció writeModeActual().

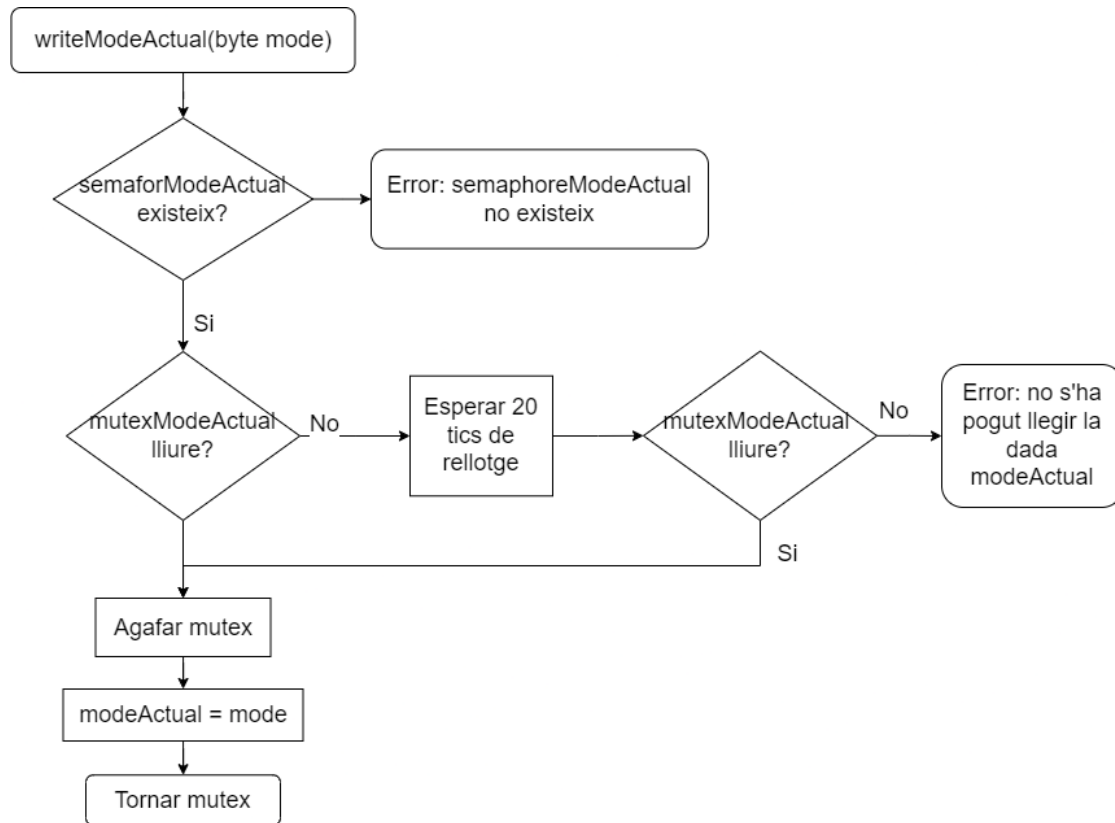


Figura 22: Codi exemple d'escriptura de la variable global "modeActual" a través del mètode del semàfor.

```
void writeModeActual(byte mode) {  
    if (semaphoreModeActual != NULL) {  
        if (xSemaphoreTake(semaphoreModeActual, (TickType_t)20) == pdTRUE) {  
            modeActual = mode;  
            xSemaphoreGive(semaphoreModeActual);  
        } else {  
            Serial.println("No se ha pogut escriure en la dada modeActual");  
        }  
    } else {  
        Serial.println("semaphoreModeActual == NULL");  
    }  
}
```



6.2.3. Tasques

El sistema està dissenyat de manera modular, dividit en diferents tasques, cadascuna ha sigut programada per a complir una funció específica dins del projecte. Aquesta divisió permet una major eficiència i organització, seguint els principis d'un sistema de temps real. Per aconseguir-ho ha utilitzat el sistema operatiu FreeRTOS, que facilita la gestió de múltiples tasques en paral·lel, optimitzant els recursos de la CPU. A continuació, es descriuran les funcions principals de cada tasca, posteriorment s'establirà l'ordre de prioritats de les tasques i finalment es mostraran els diagrames de flux de les tasques.

6.2.3.1. *Descripció de les tasques*

Per a una major organització de la explicació de les tasques, s'han dividit en tres grups:

- Tasques per a la lectura dels sensors
- Tasques per al control del ventilador
- Tasques per al enviament i rebuda de dades

Adicionalment també s'explica la funció de `setup()`, la qual s'encarrega d'inicialitzar les configuracions necessàries per a executar el programa de l'usuari i la funció `pulseDetected()`, la qual s'executa quan ocorre una interrupció del programa associada a un pin.

Funció `setup()`

Aquesta funció no és una tasca, sols s'executa una vegada a l'inici del programa, quan el micro s'inicia o es reinicia. Al principi s'inicia la comunicació amb el terminal, per a enviar missatges al port serial, a continuació s'inicia el pin de l'encoder (`yellowPin`) com a una interrupció associada a la funció `pulseDetected()`, amb `attachInterrupt()`. A continuació, s'estableix el pin `relePin` en mode d'eixida, s'inicia la connexió amb el sensor `INA219`, i s'estableix el mode de connexió com a estació per a la comunicació remota, seguidament s'inicia la connexió. A continuació, s'inicia el protocol `ESP-NOW`, es defineixen les funcions del callback, `OnDatSent()` i `OnDataRecv()`, aquestes s'activaran cada vegada que s'envia i es reb un missatge, respectivament. Seguidament es creen els semàfors amb `xSemaphoreCreateMutex()` i es creen les tasques amb `xTaskCreate()`, no obstant es suspèn amb `vTaskSuspend()`, per a què no comence a funcionar el motor. Posteriorment s'assegura que la velocitat del motor siga 0, a través de la funció `analogWrite()` i també es fica el pin del relé a 0, mitjançant `digitalWrite()`. Finalment, s'encenen les tasques `taskSendDataHandle`, `taskStartAndStopHandle`, `taskReadTemperature` i `taskReadCurrent`.

Funció `pulseDetected()`

Com s'observa en la Figura 23, la funció `pulseDetected()` envia una notificació a la tasca `taskReadMotorSpeed()`, per a contar els polsos, mitjançant `vTaskNotifyGiveFromISR()` i posteriorment es força un canvi de context si fora necessari amb la funció `portYIELD_FROM_ISR()`.

Figura 23: *Funció pulseDetected().*

```
void pulseDetected() {  
  
    BaseType_t xHigherPriorityTaskWoken = pdFALSE;  
  
    // S'envia una notificació a taskReadMotorSpeed()  
    vTaskNotifyGiveFromISR(taskReadMotorSpeedHandle, &xHigherPriorityTaskWoken);  
  
    // Es força un canvi de context si és necessari  
    portYIELD_FROM_ISR(xHigherPriorityTaskWoken);  
}
```

Tasques per a la lectura dels sensors

taskReadMotorSpeed()

Aquesta tasca s'encarrega de calcular la velocitat de l'eix reduït. A través d'una notificació enviada des de la interrupció associada a l'encoder, de manera que mitjançant la funció ulTaskNotifyTake() reentra les vegades que la notificació ha sigut enviada a la variable "pulseCountNotification", cada vegada que s'activa la funció el comptador torna a zero. Seguidament es calcula la velocitat de l'eix principal i es guarda en la variable "speed_shaft". Sabent que una volta de l'eix no reduït genera sis polsos en l'encoder, i que el període de la funció és d'un segon, es poden calcular les revolucions per minut (RPM) amb la fórmula Equació 1, on comptadorPolsos ha de ser substituït per la variable "pulseCountNotification", polsosPerRevolució per la constant "pulsesPerRevolution", i el resultat ha de ser guardat en la variable "speed_shaft".

Equació 1: *equació per al càlcul de les revolucions de l'eix principal*

$$RPM_1 = \frac{\text{comptadorPolsos (unt.)}}{\text{polsosPerRevolució(unt./rev.)}} \cdot \frac{60 \text{ segons}}{1 \text{ minut}}$$

Una vegada coneguda la velocitat que té l'eix principal, i sabent la relació de reducció entre l'eix principal i el secundari, es pot obtenir les RPM de l'eix reduït (per cada 19 voltes que gira l'eix principal, el reduït gira 1 volta), de manera que amb la fórmula Equació 2, **¡Error! No se encuentra el origen de la referencia.** es pot obtenir les revolucions per minut de l'eix secundari. En aquesta fórmula RPM_1 és la velocitat calculada prèviament, guardada a la variable "speed_shaft", ratioReducció ha de ser substituït per la constant "reductionRatio" i RPM_2 ha de ser guardat a la variable "velocitatActual_private", la qual posteriorment s'utilitza per a guardar la velocitat de l'eix en la variable global "velocitatActual", mitjançant writeVelocitatActual().

Equació 2: *equació per al càlcul de les revolucions de l'eix secundari*

$$RPM_2 = RPM_1(\text{revolució eix1}) \cdot \frac{1 (\text{revolució eix2})}{\text{ratioReducció (revolució eix 1)}}$$



taskReadTemperature()

Aquesta tasca s'encarrega d'obtenir la temperatura captada pel sensor DHT22, mitjançant la funció `getTemperature()`, la qual retorna la temperatura en graus centígrads, aquesta és guardada en una variable local i seguidament en la variable global a través de la funció `wirteTemperaturaActual()`.

taskReadCurrent()

Aquesta tasca s'encarrega d'obtenir la temperatura captada pel sensor INA219, mitjançant la funció `getCurrent_mA()`, la qual retorna la corrent en mili ampers, aquesta és guardada en una variable local i seguidament en la variable global a través de la funció `wirteCorrentActual()`.

Tasques per al control del ventilador

taskStartAndStop()

Aquesta tasca s'encarrega d'activar i desactivar el motor, depenent de la variable "estatDesitjat" i "estatActual". Si "estatDesitjat" i "estatActual" són diferents, canvia "estatActual" al "estatDesitjat", mitjançant `writeEstatAcutal()`, a més activa el relé, per a alimentar al motor, i activa la tasca de la lectura de la velocitat, a continuació depenent del mode desitjat activa una tasca del control del motor o un altra, si "modeDesitjat" és ú, s'activa la tasca del control manual; si és dos, s'activa el control automàtic. Posteriorment, comprova si "modeDesitjat" i "modeActual" són diferents, si es així actualitza la variable "modeActual" amb `wirteModeActual()`. Finalment comprova que la variable corrent no excedisca la corrent màxima permesa, si és així, modifica la variable "missatgeError" a través de `writeMissatgeError()` a 1, i a més modifica les variables "estatAcutal" a 0 i "estatDesitjat" a 0, per a que es pare el motor.

taskSpeedControlAuto()

Aquesta tasca s'encarrega de controlar la velocitat en funció de la temperatura desitjada. Depenent de les variables "estatActual" i "modeActual" funciona de una manera o un altra, com es pot veure en la Taula 2, si "estatActual" és ú i "modeActual" dos, es realitza un control de la velocitat depenent de la temperatura, si "estatActual" és ú però "modeActual" és també ú, aleshores es guarda la variable "senyalVelocitat" local en la variable global, a través de `wirteSenyalVelocitat()`, per a que al canviar de tasca al control manual, la senyal de la velocitat no canvie brusquement, posteriorment es suspèn la aquesta tasca, i s'activa la tasca del control manual. Continuant amb els modes de execució que té aquesta tasca, si "estatActual" és zero, independentment del valor en "modeAcutal", la velocitat decandra fins arribar a zero, i es suspendrà la tasca.

Taula 2: Esquema de funcionament de la tasca `taskSpeedControlAuto()` segons les variables "estatActual" i "modeAcutal".

estatActual	modeActual	Descripció de la execució
1	2	Control automàtic de la velocitat en funció de la temperatura
1	1	Canvi de mode a manual
0	-	Descens de la velocitat

taskSpeedControlManual()

Aquesta tasca s'encarrega de controlar la velocitat en funció de la velocitat desitjada. Depenent de les variables "estatActual" i "modeActual" funciona de una manera o un altra, com es pot veure en la Taula 3, si "estatActual" és ú i "modeActual" ú, es realitza un control de la velocitat depenent de la velocitat, si "estatActual" és ú però "modeActual" és dos, aleshores es guarda la variable "senyalVelocitat" local en la variable global, a través de writeSenyalVelocitat(), per a que al canviar de tasca al control automàtic, la senyal de la velocitat no canvie brusquement, posteriorment es suspèn la aquesta tasca, i s'activa la tasca per al control automàtic. Continuant amb els modes de execució que té aquesta tasca, si "estatActual" és zero, independentment del valor en "modeActual", la velocitat decandra fins arribar a zero, i es suspendra la tasca.

Taula 3: Esquema de funcionament de la tasca taskSpeedControlManual() segons les variables "estatActual" i "modeActual".

estatActual	modeActual	Descripció de la execució
1	1	Control manual de la velocitat en funció de la velocitat
1	2	Canvi de mode a automàtic
0	-	Descens de la velocitat

Tasques per al enviament i rebuda de dades

taskSendData()

Aquesta tasca s'encarrega d'enviar les dades al panell de control periòdicament. Primer es crea una variable de tipus struct_dataFromController, nomenada "dataFromController", i a continuació es procedeix a guardar en cada paràmetre l'homòleg corresponent de les variables globals del sistema, com es mostra en la Figura 24. Posteriorment, s'envien les dades mitjançant la funció esp_now_send().

Figura 24: Codi de la tasca taskSendData().

```
void taskSendData(void *pvParameters) {  
  
    TickType_t xLastWakeTime;  
  
    xLastWakeTime = xTaskGetTickCount();  
    for (;;) {  
        vTaskDelayUntil(&xLastWakeTime, xFrequency_taskSendData);  
  
        struct_dataFromController dataFromController;  
  
        dataFromController.modeActual = readModeActual();  
        dataFromController.temperaturaActual = readTemperaturaActual();  
        dataFromController.velocitatActual = readVelocitatActual();  
        dataFromController.estatActual = readEstatActual();  
        dataFromController.correntActual = readCorrentActual();  
        dataFromController.missatgeError = readMissatgeError();  
  
        esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *)&dataFromController, sizeof(dataFromController));  
    }  
}
```

OnDataRecv()

Aquesta funció no és una tasca periòdica, és una funció que s'activa quan es rep un missatge de la pantalla. Primerament, s'utilitza la funció `memcpy()`, per a copiar el missatge rebut en la variable "dataFromDisplay" que es de tipus `struct_dataFromDisplay`. Finalment, cada paràmetre del struct es guarda en la variable global corresponent, com es pot veure en la Figura 25.

Figura 25: *Funció OnDataRecv().*

```
void OnDataRecv(const esp_now_recv_info_t *mac, const uint8_t *incomingData, int len) {  
    struct_dataFromDisplay dataFromDisplay;  
  
    memcpy(&dataFromDisplay, incomingData, sizeof(dataFromDisplay));  
  
    writeModeDesitjat(dataFromDisplay.modeDesitjat);  
    writeVelocitatDesitjada(dataFromDisplay.velocitatDesitjada);  
    writeTemperaturaDesitjada(dataFromDisplay.temperaturaDesitjada);  
    writeEstatDesitjat(dataFromDisplay.estatDesitjat);  
  
    Serial.println("S'han guardat les dades en les variables globals");  
}
```

6.2.3.2. Orde de prioritats de les tasques

En un sistema de temps real totes les tasques no tenen la mateixa importància, ni el mateix patró d'execució. Existeixen diferents tipus de tasques, segons el patró de repetició, en aquests projecte existeixen les tasques periòdiques, aquelles que es repeteixen a intervals regulars; aperiòdiques, les que s'activen davant l'ocurrència d'un esdeveniment determinat; i segons la criticitat, poden ser crítiques, les quals han de complir sempre les restriccions de temps, i acrítiques, poden no complir les restriccions crítiques.

Per a què el sistema funcione correctament, totes les tasques han de satisfer els seus requisits temporals, el component que determina el pla d'execució del programa és el planificador, mitjançant l'aplicació d'un determinat algoritme o política de planificació. En aquest sistema es va a aplicar el planificador per prioritats fixes, aquest és un planificador dinàmic, ja que genera el pla durant l'execució del programa basant-se en les prioritats de les tasques.

En aquest planificador s'assigna una prioritat a totes les tasques, de manera que en cada moment s'executa la tasca més prioritària. Per tant si una tasca de menor prioritat s'està executant i arriba una tasca de major prioritat, la primera és expulsada immediatament fins a l'execució de la tasca més prioritària.

L'ordre de prioritats establert per a les tasques es mostra en la Taula 4, on la més prioritària és la número cinc i la menys la número ú. La més crítica és la que llig la velocitat del motor, ja que per al càlcul de la velocitat és important el temps transcorregut entre un període i el següent, la segona més crítica és la tasca que llig la corrent, per assegurar un correcte funcionament del motor, el següent lloc, és ocupat per la tasca que s'encarrega de la parada i l'inici del motor, i a continuació el de la lectura del sensor de temperatura, per últim les menys crítiques són les del control automàtic i manual i la d'enviament de dades.

Taula 4: *Ordre de prioritats de les tasques en el programa del dispositiu de ventilació.*

Prioritat	Tasca
5	taskReadMotorSpeed
2	taskReadTemperature
4	taskReadCurrent
3	taskStarAndStop
1	taskSpeedControlAuto
1	taskSpeedControlManual
1	taskSendData

Per a la comprovació de que les tasques no es passaven el seu termini es necessita saber el seu temps de còmput, de manera que s'ha executat cada una de les tasques individualment, és a dir, sense que siga interrompuda per ninguna altra tasca. Per a la temporització de les tasques s'ha introduït en el codi timers, com els de la Figura 26, a l' inici de la funció, s'ha guardat en la variable "t1", els micro segons que han passat en eixe instant, posteriorment, al final de les operacions que ha de realitzar, se ha guardat en la variable "t2", els micro segons en eixe instant, posteriorment "time", és la resta entre "t2" menys "t1", que és l'interval de temps que s'utilitza per a realitzar les operacions. Com es pot observar en la Figura 27, el màxim temps de còmput de la tasca OnDataRecv() ha sigut de 639 micro segons, de manera que s'estableix un màxim temps de còmput de menys d'un mili segon.

Figura 26: *Codi exemple per a la temporització de la funció OnDataRecv().*

```
void OnDataRecv(const esp_now_recv_info_t *mac, const uint8_t *incomingData, int len) {
    unsigned long t1 = micros();
    struct_dataFromDisplay dataFromDisplay;

    memcpy(&dataFromDisplay, incomingData, sizeof(dataFromDisplay));

    writeModeDesitjat(dataFromDisplay.modeDesitjat);
    writeVelocitatDesitjada(dataFromDisplay.velocitatDesitjada);
    writeTemperaturaDesitjada(dataFromDisplay.temperaturaDesitjada);
    writeEstatDesitjat(dataFromDisplay.estatDesitjat);

    Serial.println("Se han guardado los datos en las variables globales");

    unsigned long t2 = micros();
    unsigned long time = t2 - t1;

    Serial.print("Time (OnDataRecv) = ");
    Serial.println(time);
}
```

Figura 27: Resultat de la temporització de la tasca OnDataRecv().

```
--- START ---  
Se han guardado los datos en las variables globales  
Time (OnDataRecv) = 383  
Se han guardado los datos en las variables globales  
Time (OnDataRecv) = 116  
Se han guardado los datos en las variables globales  
Time (OnDataRecv) = 639  
Se han guardado los datos en las variables globales  
Time (OnDataRecv) = 153  
Se han guardado los datos en las variables globales  
Time (OnDataRecv) = 102  
Se han guardado los datos en las variables globales  
Time (OnDataRecv) = 1012  
Se han guardado los datos en las variables globales  
Time (OnDataRecv) = 696  
Se han guardado los datos en las variables globales  
Time (OnDataRecv) = 103  
Se han guardado los datos en las variables globales  
Time (OnDataRecv) = 557  
Se han guardado los datos en las variables globales  
Time (OnDataRecv) = 103  
Se han guardado los datos en las variables globales  
Time (OnDataRecv) = 102  
Se han guardado los datos en las variables globales  
Time (OnDataRecv) = 103
```

De manera que s'han temporitzat totes les tasques, donant com a resultat la Taula 5.

Taula 5: Màxim temps de còmput observat.

Tasca	Tems d'execució màxim (μs)	Figura
taskReadMotorSpeed	1308	Figura 28
taskReadTemperature	7204	Figura 29
taskReadCurrent	2209	Figura 30
taskStarAndStop	28	Figura 31
taskSpeedControlAuto	487	Figura 32
taskSpeedControlManual	2111	Figura 33
taskSendData	609	Figura 34

Cal nomenar que el temps de còmput de les tasques és insignificant en comparació al període de les tasques, el temps d'execució màxim que s'ha arribat a observar ha sigut de uns pocs mili segons, per tant en comparació a 100 mili segons, que es el període més curt establert en una de les tasques, és insignificant.

Figura 28: Resultat de la temporització de la tasca `taskReadMotorSpeed()`.

```
Reduced Shaft Speed (RPM): 0.00
Time (taskReadMotorSpeed)= 304
Reduced Shaft Speed (RPM): 0.00
Time (taskReadMotorSpeed)= 304
Reduced Shaft Speed (RPM): 0.00
Time (taskReadMotorSpeed)= 304
Reduced Shaft Speed (RPM): 0.00
Time (taskReadMotorSpeed)= 1308
Reduced Shaft Speed (RPM): 0.00
Time (taskReadMotorSpeed)= 304
Reduced Shaft Speed (RPM): 0.00
Time (taskReadMotorSpeed)= 304
Reduced Shaft Speed (RPM): 0.00
Time (taskReadMotorSpeed)= 304
Reduced Shaft Speed (RPM): 0.00
Time (taskReadMotorSpeed)= 304
Reduced Shaft Speed (RPM): 0.00
Time (taskReadMotorSpeed)= 305
Reduced Shaft Speed (RPM): 0.00
Time (taskReadMotorSpeed)= 304
```

Figura 29: Resultat de la temporització de la tasca `taskReadTemperature()`

```
temperatura = 25.70
*****
Time (taskReadTemperature)= 7204
Temperatura = 25.70
*****
Time (taskReadTemperature)= 311
Temperatura = 25.70
*****
Time (taskReadTemperature)= 6300
Temperatura = 25.70
*****
Time (taskReadTemperature)= 2156
Temperatura = 25.80
*****
Time (taskReadTemperature)= 7204
Temperatura = 25.80
*****
Time (taskReadTemperature)= 311
Temperatura = 25.80
*****
Time (taskReadTemperature)= 7170
Temperatura = 25.80
*****
Time (taskReadTemperature)= 2163
```

Figura 30: Resultat de la temporització de la tasca `taskReadCurrent()`.

```
Current:      23.20 mA
*****
Time (taskReadCurrent)= 2208
Current:      23.20 mA
*****
Time (taskReadCurrent)= 1329
Current:      23.20 mA
*****
Time (taskReadCurrent)= 2209
Current:      23.40 mA
*****
Time (taskReadCurrent)= 1328
Current:      23.20 mA
*****
Time (taskReadCurrent)= 2209
Current:      23.30 mA
*****
Time (taskReadCurrent)= 1327
Current:      23.30 mA
*****
Time (taskReadCurrent)= 1327
Current:      23.70 mA
*****
Time (taskReadCurrent)= 1327
Current:      23.50 mA
*****
Time (taskReadCurrent)= 2209
Current:      23.20 mA
*****
Time (taskReadCurrent)= 1328
Current:      23.60 mA
*****
Time (taskReadCurrent)= 2209
Current:      23.30 mA
*****
Time (taskReadCurrent)= 1328
Current:      23.40 mA
*****
Time (taskReadCurrent)= 2209
```

Figura 31: Resultat de la temporització de la tasca `taskStartAndStop()`.

```
Time (taskSartAndStop)= 28
Time (taskSartAndStop)= 26
Time (taskSartAndStop)= 26
Time (taskSartAndStop)= 27
Time (taskSartAndStop)= 26
Time (taskSartAndStop)= 26
Time (taskSartAndStop)= 26
Time (taskSartAndStop)= 26
Time (taskSartAndStop)= 27
Time (taskSartAndStop)= 28
Time (taskSartAndStop)= 28
Time (taskSartAndStop)= 26
Time (taskSartAndStop)= 26
Time (taskSartAndStop)= 27
Time (taskSartAndStop)= 28
Time (taskSartAndStop)= 26
Time (taskSartAndStop)= 26
Time (taskSartAndStop)= 27
```

Figura 32: Resultat de la temporització de la tasca `taskControlSpeedAuto()`.

```
Accion de control (auto)= 0
*****
Time (taskControlSpeedAuto)= 397
Error = -25.60
Accion de control (auto)= 0
*****
Time (taskControlSpeedAuto)= 487
Error = -25.60
Accion de control (auto)= 0
*****
Time (taskControlSpeedAuto)= 486
Error = -25.60
Accion de control (auto)= 0
*****
Time (taskControlSpeedAuto)= 400
Error = -25.60
Accion de control (auto)= 0
*****
Time (taskControlSpeedAuto)= 486
Error = -25.60
Accion de control (auto)= 0
*****
Time (taskControlSpeedAuto)= 397
Error = -25.60
Accion de control (auto)= 0
*****
Time (taskControlSpeedAuto)= 397
Error = -25.60
Accion de control (auto)= 0
```

Figura 33: Resultat de la temporització de la tasca `taskControlSpeedManual()`.

```
accionDeControl_private = 200
*****
Time (taskControlSpeedManual)= 2107
Se han guardado los datos en las variables globales
Time (OnDataRecv) = 769
Error = 230.00
Senyal de velocidad = 318
accionDeControl_private = 200
*****
Time (taskControlSpeedManual)= 2111
Error = 230.00
Senyal de velocidad = 318
accionDeControl_private = 200
*****
Time (taskControlSpeedManual)= 2107
Error = 230.00
Senyal de velocidad = 318
accionDeControl_private = 200
*****
Time (taskControlSpeedManual)= 2106
Error = 230.00
Senyal de velocidad = 318
accionDeControl_private = 200
*****
Time (taskControlSpeedManual)= 2107
Error = 230.00
Senyal de velocidad = 318
accionDeControl_private = 200
*****
Time (taskControlSpeedManual)= 2107
```


Figura 34: Resultat de la temporització de la tasca `taskSendData()`.

```
Time (taskSendData)= 609
Time (taskSendData)= 343
Time (taskSendData)= 348
Time (taskSendData)= 348
Time (taskSendData)= 390
Time (taskSendData)= 350
Time (taskSendData)= 349
Time (taskSendData)= 342
Time (taskSendData)= 387
Time (taskSendData)= 347
Time (taskSendData)= 388
Time (taskSendData)= 388
Time (taskSendData)= 383
```

6.2.3.3. Diagrames de flux de les tasques

Per a una major comprensió del flux del programa, a continuació es presenten en forma de diagrama els fluxos de cada tasca. Aquests diagrames permeten visualitzar clarament el funcionament de les diverses parts del sistema, oferint una perspectiva més detallada i estructurada. Les figures Figura 35, Figura 36, Figura 37, Figura 38, Figura 39, Figura 40 i Figura 41 il·lustren els processos seqüencials de cada tasca, facilitant la comprensió de les operacions realitzades.

Figura 35: Diagrama de flux de la tasca que llig la velocitat del motor.

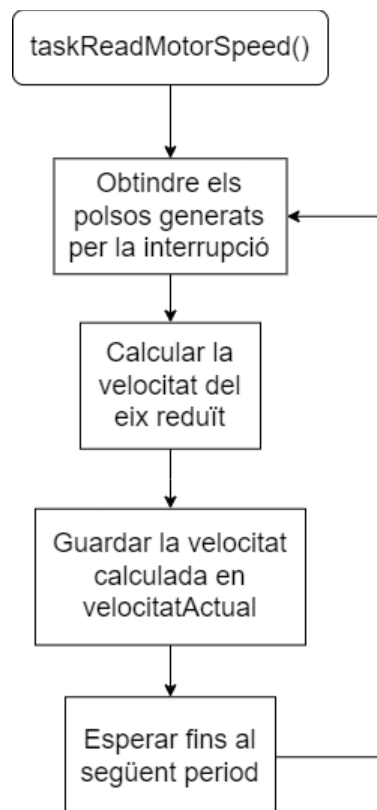


Figura 36: Diagrama de flux de la tasca que llig la temperatura.

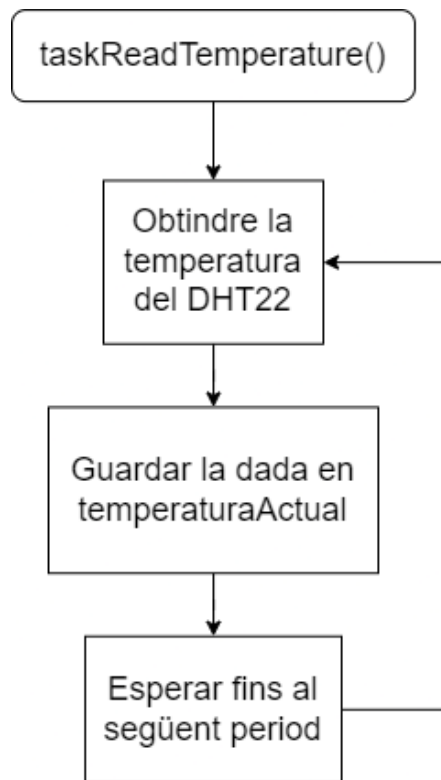


Figura 37: Diagrama de flux de la tasca que llig la corrent del motor.

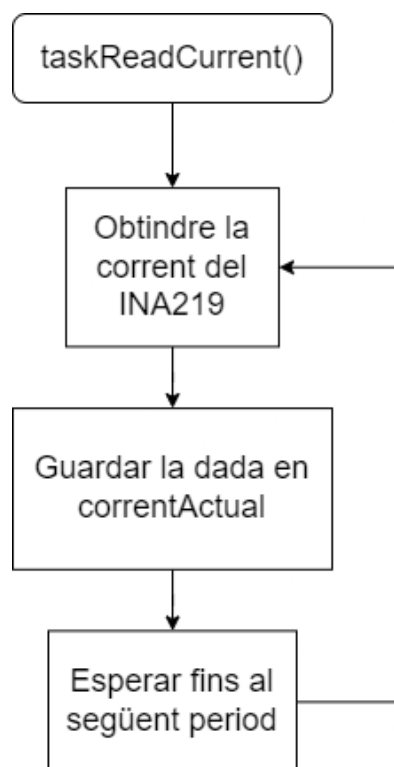


Figura 38: Diagrama de flux de la tasca que envia les dades al panell de control.

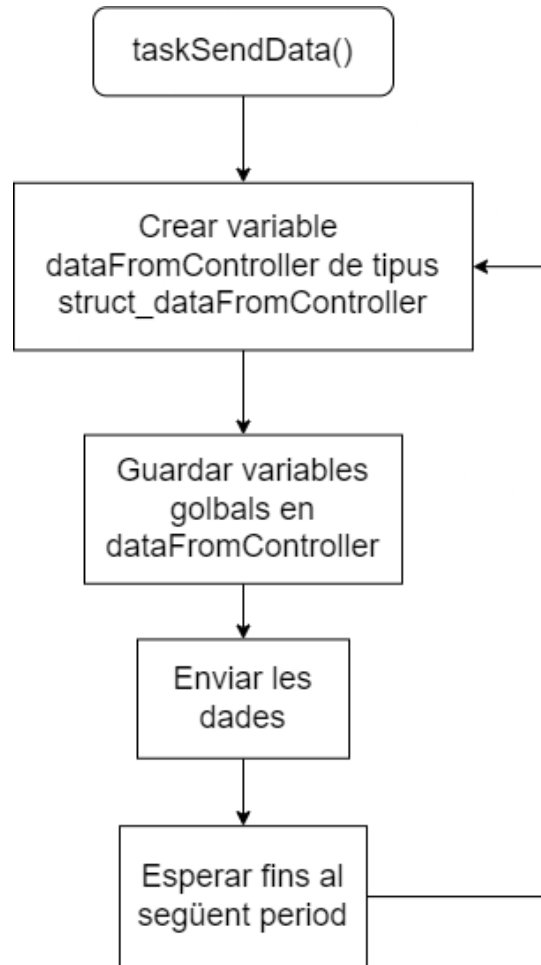


Figura 39: Diagrama de flux de la tasca que controla el motor.

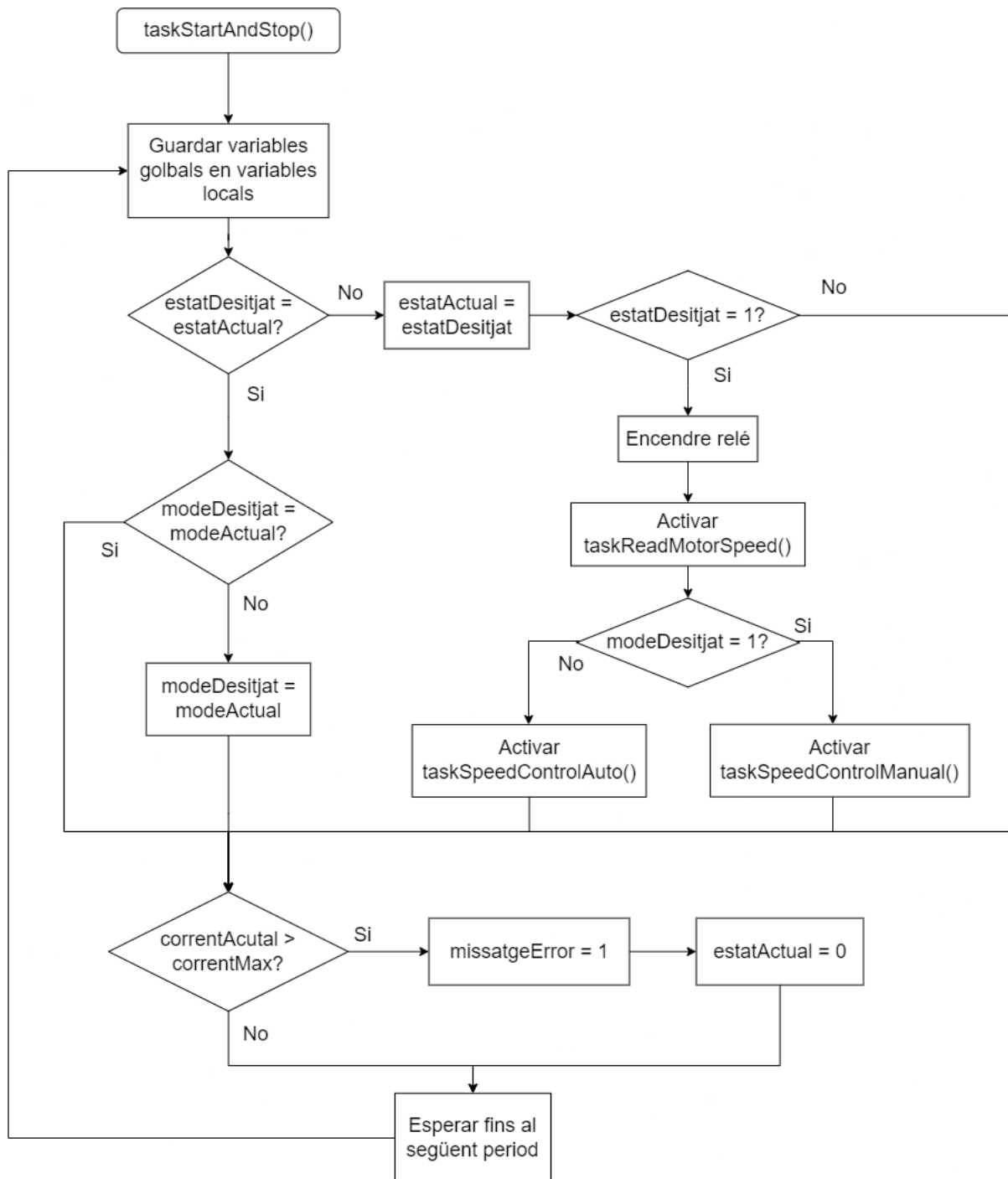


Figura 40: Diagrama de flux de la tasca que controlar el motor en mode automàtic.

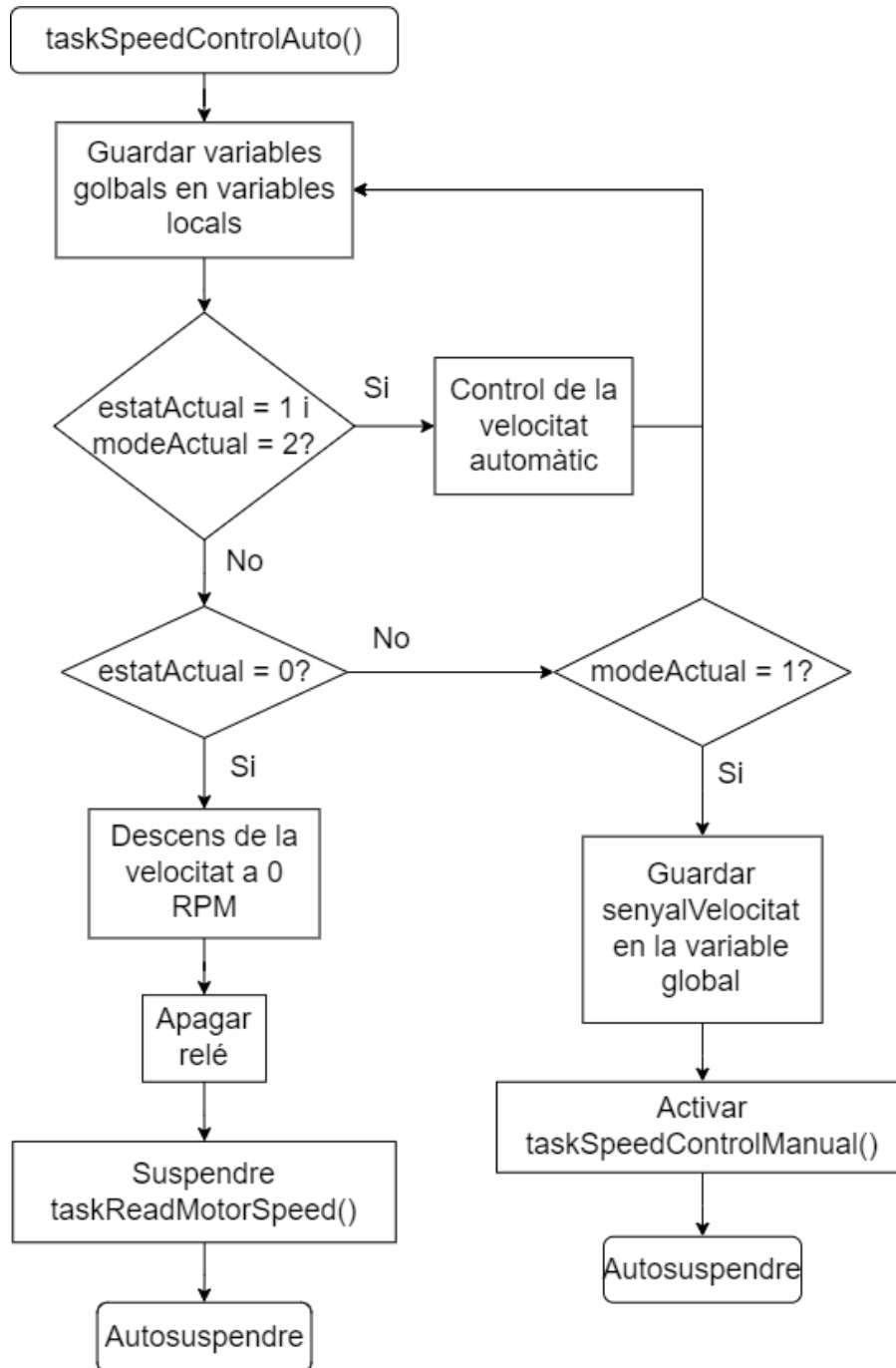
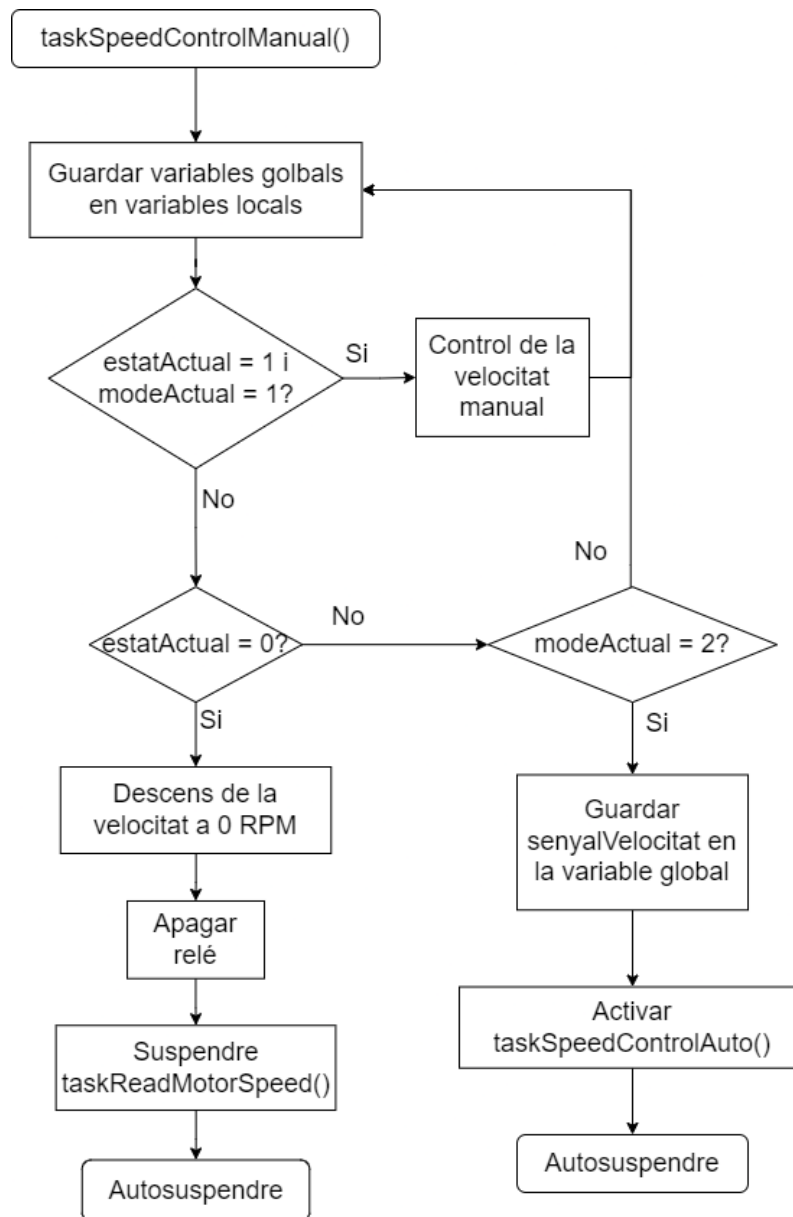


Figura 41: Diagrama de flux de la tasca que controla el motor en mode manual.



6.2.4. Controlador

Per al control del motor, no ha sigut possible obtindre la funció de transferència ja que no ha sigut proporcionada pel fabricant, de manera que s'ha decidit implementar un controlador proporcional. En la Figura 42, es mostra el diagrama del controlador proporcional per al mode manual, on K_p és la constant proporcional, igual a 0.08, aquesta xifra ha sigut obtinguda de manera experimental. A la Figura 43, es mostra el controlador proporcional per al mode automàtic, on K_t , és la constant proporcional, igual a -2, aquesta també ha sigut obtinguda de manera experimental, és negativa perquè l'increment que es va a realitzar és contrari a la magnitud que es va a controlar, per exemple si la temperatura actual és de 28 °C, i es vol una temperatura de 25°C, l'error és de -3°C, per lo que al

multiplicar-lo per la constata K_t , es converteix en positiu l'error, i per tant l'acció de control també és positiva. Amb aquest controlador hi ha un 24% d'error, ja que al donar-li una senyal de velocitat de 300 revolucions per minut, aplega a unes revolucions de 258 revolucions per minut, aquesta no és la situació ideal, no obstant no s'ha pogut obtenir un controlador millor, ja que, com s'ha comentat abans, el fabricant no ha proporcionat la funció de transferència del motor.

Figura 42: Diagrama de blocs del controlador per al mode manual.

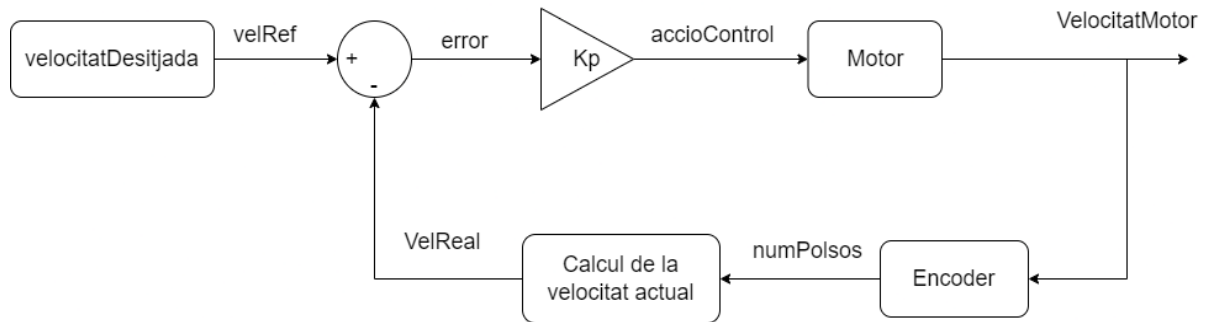
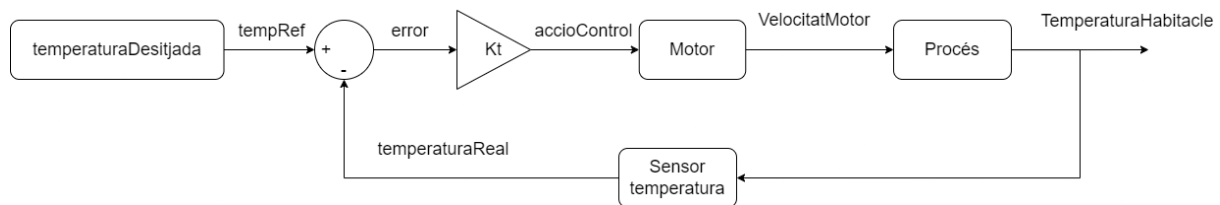


Figura 43: Diagrama de blocs del controlador per al mode automàtic.



6.3. Quadre de control

La funció principal del panell de control és permetre a l'usuari monitoritzar l'estat del sistema i interactuar amb ell a través de la pantalla tàctil, aquesta mostra informació rellevant sobre el dispositiu de ventilació, com la velocitat del motor, el corrent que el travessa, la temperatura de l'estança i el mode de funcionament actual. A més, permet que l'usuari reba missatges d'errors del sistema.

El microcontrolador de la pantalla també es un ESP32, com s'ha mencionat prèviament, de manera que també es va a fer funcionar com un sistema de temps real, dividit en diferents tasques, i seguint els principis o claus nomenades l'apartat del dispositiu de ventilació.

Es necessiten les següents llibreries: `esp_now.h` i `WiFi.h`, per a la comunicació remota; `SPI.h`, `XPT2046_TouchScreen.h` i `TFT_eSPI.h`, per al control de la pantalla i la detecció de la pantalla tàctil.

Per a poder utilitzar la llibreria `TFT_eSPI.h` és necessari la modificació del fitxer `User_Setup.h`, que es troba en la següent direcció "C:\Users\usuari\Documents\Arduino\libraries\TFT_eSPI" (s'ha de substituir "usuari" per l'usuari real). Aquest fitxer es divideix en seccions, en la primera cal seleccionar el driver adequat per a la pantalla, en aquest cas és el `ILI9341_2_DRIVER`, com es mostra en la Figura 44. En la secció 2, s'ha de trobar la secció que alberga les macros per als pins del ESP32, com s'observa en la Figura 45, segons el esquema electrònic mostrat la Figura 46, el pin d'eixida de dades (SDO) està connectat al pin 12 del ESP, el d'entrada de dades (SDI) al pin 13, la senyal de rellotge (SCK) al pin 14, el del chip select (cs) al pin 15, el de reset (RS) al pin 2, que correspon amb la macro nomenada `TFT_DC`.

Figura 44: Definició del driver corresponent.

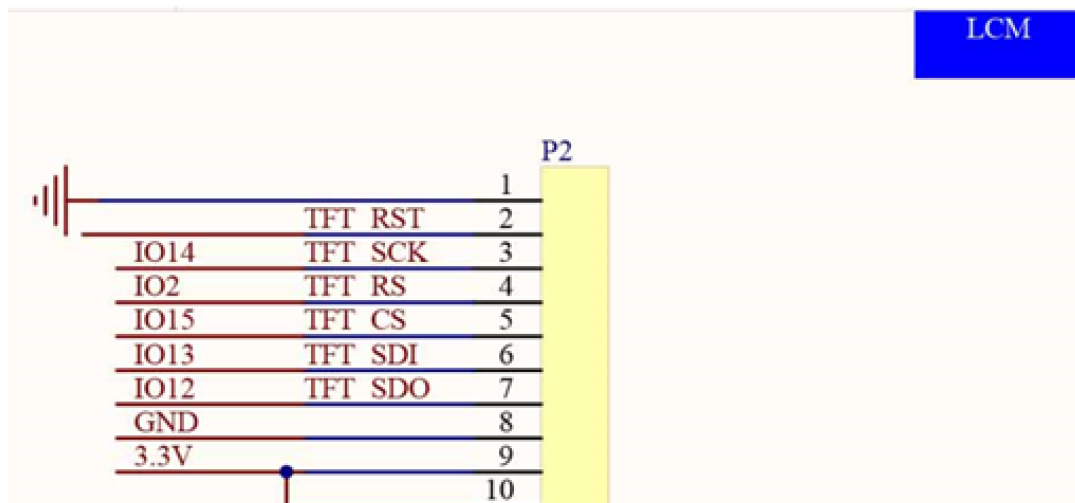
```
// Display type - only define if RPi display
// #define RPI_DISPLAY_TYPE // 20MHz maximum SPI

// Only define one driver, the other ones must be commented out
// #define ILI9341_DRIVER // Generic driver for common displays
#define ILI9341_2_DRIVER // Alternative ILI9341 driver, see https://github.com/Bodmer/TFT_eSPI/issues/1172
// #define ST7735_DRIVER // Define additional parameters below for this display
// #define ILI9163_DRIVER // Define additional parameters below for this display
// #define S6D02A1_DRIVER
```

Figura 45: Definició de les macros per al display.

```
195 // ##### EDIT THE PIN NUMBERS IN THE LINES FOLLOWING TO SUIT YOUR ESP32 SETUP #####
196
197 // For ESP32 Dev board (only tested with ILI9341 display)
198 // The hardware SPI can be mapped to any pins
199
200 #define TFT_MISO 12
201 #define TFT_MOSI 13
202 #define TFT_SCLK 14
203 #define TFT_CS 15 // Chip select control pin
204 #define TFT_DC 2 // Data Command control pin
205 // #define TFT_RST 4 // Reset pin (could connect to RST pin)
206 #define TFT_RST -1 // Set TFT_RST to -1 if display RESET is connected to ESP32 board RST
207
```

Figura 46: Esquema de connexions del display.

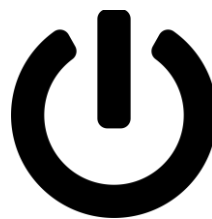


Nota: Figura amb llicència © obtinguda de [16]

Cal comentar l'ús de fitxers addicionals que han sigut afegits per a mostrar un tipus de lletra seleccionada i icones per a la interfície d'usuari. Per a l'objectiu primer s'ha inclòs el fitxer Free_Fonts.h, aquest es pot trobar en alguns dels exemples de la llibreria TFT_eSPI.h, només cal copiar-lo i pegar-lo en la carpeta que conté el projecte i incloure'l, amb #include "Free_Fonts.h".

Per a fer que pel display s'imprimisquen icones, primer s'ha descarregat una imatge d'internet, com la de la Figura 47, després s'ha reduït de grandària a una més xicoteta, mitjançant una pàgina web, com per exemple imatge.online-convert.com [20], la qual permet convertir imatges d'un format a un altre, a més es pot indicar el tamany en píxels que es desitja, en aquest cas, s'ha establert un tamany de 60x60 píxels. A continuació, aquesta imatge en png, s'ha de convertir a un arxiu de text, la pròpia llibreria TFT_eSPI.h recomana una pàgina web per a fer-ho, online-utility.org [21], que permet passar múltiples tipus d'arxius a xbm, de manera que s'ha de pujar la imatge a aquesta pàgina web, i descarregar-se l'arxiu que retorna. No obstant, l'Arduino IDE no pot llegir arxius en format xbitmap, de manera que cal canviar l'extensió de ".xbm" a ".h", posteriorment incloure el header en la carpeta del programa i incloure el nom del fitxer ".h" en el projecte, mitjançant #include "nom_del_arxiu.h".

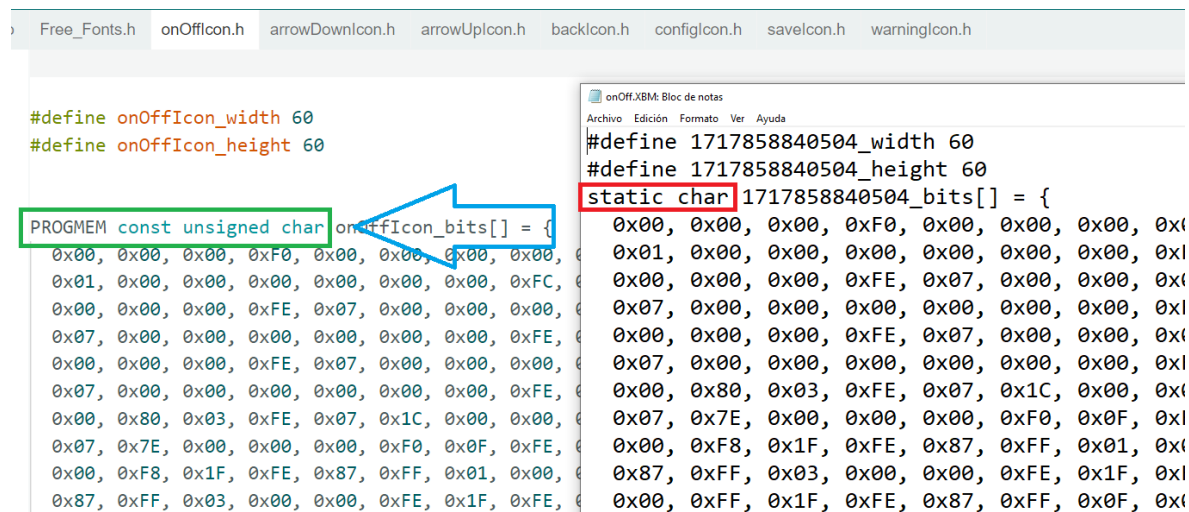
Figura 47: Icona d'on i off.



Nota: Figura amb llicència CC obtinguda de [22].

A continuació, s'ha de modificar l'arxiu, ja que el tipus de dada que ha generat la pàgina web no és compatible amb la llibreria. Com es mostra en la Figura 48, en l'arxiu onOff_XBM.XBM, que és l'arxiu descarregat de la pàgina per a transformar arxius a xbitmap, el tipus de dada "static char" cal canviar-lo per "PROGMEM const unsigned char". Opcionalment s'ha canviat el nom de les dades per un més llegible.

Figura 48: Modificació de l'arxiu que es va a utilitzar com a icona.



```

Free_Fonts.h  onOffIcon.h  arrowDownIcon.h  arrowUpIcon.h  backIcon.h  configIcon.h  saveIcon.h  warningIcon.h

```

```

#define onOffIcon_width 60
#define onOffIcon_height 60

PROGMEM const unsigned char onOffIcon_bits[] = {
  0x00, 0x00, 0x00, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0xFE, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFE, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0xFE, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFE, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x80, 0x03, 0xFE, 0x07, 0x1C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x07, 0x7E, 0x00, 0x00, 0x00, 0xF0, 0x0F, 0xFE, 0x00, 0xF8, 0x1F, 0xFE, 0x87, 0xFF, 0x01, 0x00, 0x00,
  0x00, 0xF8, 0x1F, 0xFE, 0x87, 0xFF, 0x01, 0x00, 0x00, 0x87, 0xFF, 0x03, 0x00, 0x00, 0xFE, 0x1F, 0x00,
  0x87, 0xFF, 0x03, 0x00, 0x00, 0xFE, 0x1F, 0xFE, 0x87, 0xFF, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```

Una vegada realitzada la configuració del sistema, es va a explicar la estructura que va a seguir la explicació d'aquest punt. Primerament, es van a explicar les macros, els objectes, les constants i les variables globals del projecte, posteriorment es van a descriure les tasques que conté el programa, i finalment es van a mostrar els diagrames de flux.

6.3.1. Definició de macros, objectes, constants i variables globals

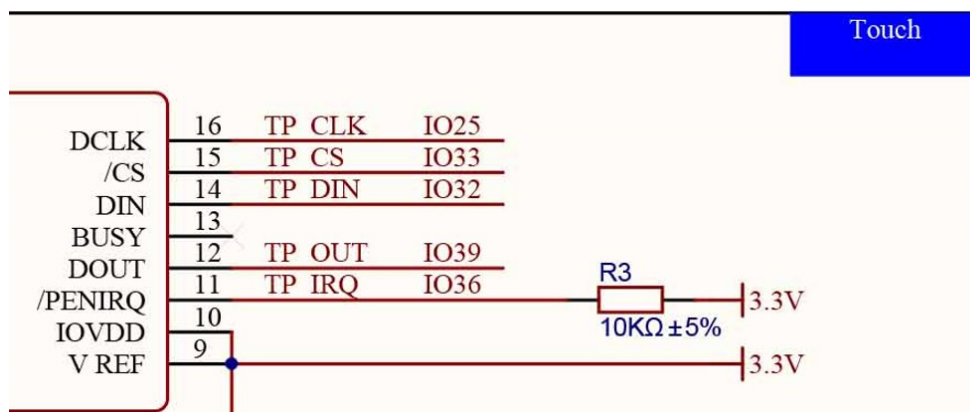
Macros

Per a la utilització de la pantalla tàctil s'han definit les macros que apareixen en la Figura 49, basant-se en el diagrama proporcionat pel fabricant, mostrat en la Figura 50.

Figura 49: Definició de les macros necessàries per a utilitzar la pantalla tàctil.

```
// touch screen pins
#define XPT2046_IRQ 36
#define XPT2046_MOSI 32
#define XPT2046_MISO 39
#define XPT2046_CLK 25
#define XPT2046_CS 33
```

Figura 50: Esquema de connexions de la pantalla tàctil.



Nota: Figura amb llicència © obtinguda de [16]

Objectes

Per a l'ús del display i la pantalla tàctil es necessiten crear objectes, com els que s'observen en la Figura 51, els quals s'utilitzen per accedir a les funcions per a controlar el display i la pantalla tàctil.

Figura 51: Objectes per a la utilització de la pantalla i la pantalla tàctil.

```
SPIClass mySpi = SPIClass(VSPI); // SPI communication
XPT2046_Touchscreen ts(XPT2046_CS, XPT2046_IRQ); // Touch Screen
TFT_eSPI tft = TFT_eSPI(); // Display
```

Constants

S'han definit les constants que es mostren en la Figura 52. “broadcastAddress” correspon amb l'adreça MAC del ESP que controla el dispositiu de ventilació, “queue_length” amb el número màxims d'elements que poden haver en una cua i “xFrequency_timerWirlessCommunication” és el període en mili segons de la tasca timerWirlessCommunication().

Figura 52: Constants del programa del panell de control.

```
const uint8_t broadcastAddress[] = { 0x48, 0xE7, 0x29, 0x8D, 0x5C, 0x6C };  
const uint8_t queue_length = 1;  
const TickType_t xFrequency_timerWirelessCommunication = 5000;
```

Variables globals

Les variables de tipus QueueHandle_t són variables per a crear cues, per a les dades “dataFromController”, “dataFromDisplay” i “isControllerOnline”, aquesta última es utilitzada per a saber si s'ha rebut dades des del controlador. Les variables de tipus TaskHandle_t s'utilitzen per al control de les tasques, pantallaHandler() i timerWirelessCommunication(). La variable SemaphoreHandle_t s'utilitza per a crear un semàfor per a protegir la variable global “missatgeError”, com es mostra en la Figura 53. “missatgeError” serà zero si no hi ha cap error, en canvi, serà diferent de zero si hi ha algun error. La codificació dels missatges és la mateixa que per al controlador.

Figura 53: Variables globals del programa del panell de control.

```
QueueHandle_t queueDataFromControllerHandler;  
QueueHandle_t queueDataFromDisplayHandler;  
QueueHandle_t queueIsControllerOnline;  
  
TaskHandle_t pantallaHandler_Handle;  
TaskHandle_t timerWirelessCommunication_Handle;  
  
SemaphoreHandle_t semaphoreMissatgeError = NULL;  
  
uint8_t missatgeError = 0;
```

Cal mencionar la creació del tipus de dades struct utilitzades també per al programa del controlador, ja que també es necessita per a l'empaquetament i desempaquetament de dades enviades i rebudes a través del protocol ESP-NOW.

6.3.2. Tasques

Aquest programa sols utilitza una tasca periòdica crítica, dues tasques aperiòdiques, a banda del setup(). Primer s'explicarà la funció setup(), i posteriorment les altres tasques i funcions.



Funció setup()

En aquesta funció s’inicialitza la comunicació remota en mode estació, per a poder enviar i rebre informació, s’inicia el protocol ESP-NOW, amb la seua configuració corresponent i les funcions del callback. Posteriorment, s’inicia la comunicació SPI, la pantalla tàctil i el display, i es configura també, la pantalla tàctil i el display de manera desitjada. A continuació, es creen les cues, el semàfor i per últim les tasques.

Tasques aperiòdiques

pantallaHandler()

Aquesta tasca és una tasca aperiòdica ja que no compleix amb un període fixe d’activament, aquesta tasca és la principal de tot el programa, per tant està tot el temps en funcionament excepte per una espera relativa de 200 mili segons, per a què la pantalla tàctil no done un fals positiu. L’objectiu d’aquesta tasca és mostrar els elements per el display, i obtindre les coordenades de la pantalla tàctil si ha sigut pressionada, i actual en conseqüència.

A l’ inici de la tasca aquesta comprova si hi ha algun element en la cua “queueDataFromController”, si és així, guarda les dades en la variable local “dataFromController” i actualitza la variable local estatActual, amb dataFromController.estatActual, a més, actualitza la variable global missatgeError, amb wirtMissatgeError(). A continuació, comprova que la variable missatgeError que siga diferent de zero, si es cert, significa que hi ha un error, per tant actualitza la variable page a 4. La variable “page” indica la “pàgina” que mostrar pel display. Seguidament, s’utilitza un switch() en funció de la variable “page”, en la Taula 6, es mostra un esquema entre el valor de “page” i el que es mostra per pantalla.

Taula 6: Pàgina que es mostra pel display segons la variable “page”.

page	Funció
0	iniciPage()
1	configPage()
2	manualPage()
3	autoPage()
4	messageErrorPage()

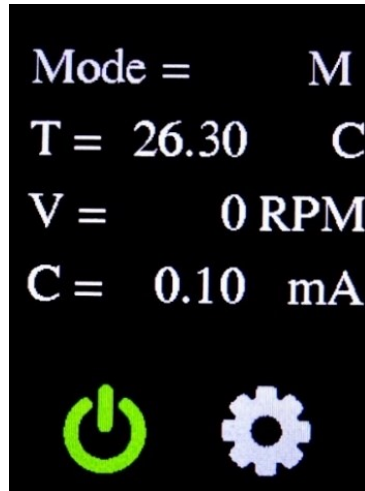
Seguidament, es comprova que la pantalla tàctil ha sigut pressionada o no, si no ha sigut així, hi ha un retard relatiu de 200 milisegons, i després torna a començar. En canvi, si s’ha detectat una pulsació en la pantalla, es guarden les coordenades de la pantalla que han sigut pressionades, amb getPoint(), a continuació depenent de la variable “page” es realitzaran unes operacions o unes altres. Una vegada atesa la petició de la pantalla tàctil, fa un retard relatiu de 200 mili segons, i torna a començar.

A continuació es comenten les funcions, nomenades en la Taula 6, que permeten mostrar pel display la informació.

iniciPage()

Aquesta mostra la informació més rellevant que concerneix al controlador, mostra el mode de funcionament, la temperatura actual, la velocitat que porta el motor en cada instant i la corrent que travessa aquest, a més mostra dos icones, una per a encendre i apagar el motor, i l'altra permet entrar en el menú de configuració, com es mostra en la Figura 54.

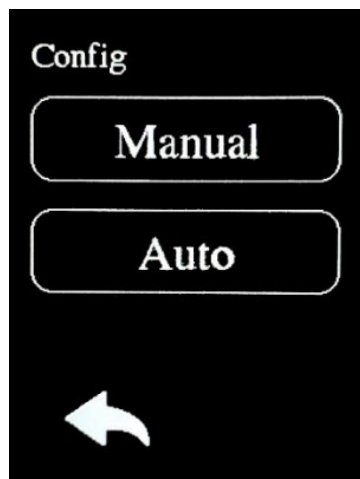
Figura 54: Pantalla d'inici del panell de control.



configPage()

En el menú, es mostren dues opcions per a poder canviar de mode, el manual i l'automàtic, com s'observa en la Figura 55, i una última opció per a tornar a la pàgina principal.

Figura 55: Pantalla menú del panell de control.



manualPage()

En entrar en el mode manual es mostra una pantalla com la de la Figura 56, on es permet establir la velocitat desitjada, directament 100, 200 o 300 revolucions per minut, o en escalons de 10 en deu mitjançant les fletxes. Per a guardar la velocitat desitjada s'ha de pressionar la icona de disquet, a l'inferior dreta. Si no es vol guardar la xifra establerta, es pot pressionar la icona de retorn, baix a l'esquerra, la qual retorna a la pàgina de configuració.

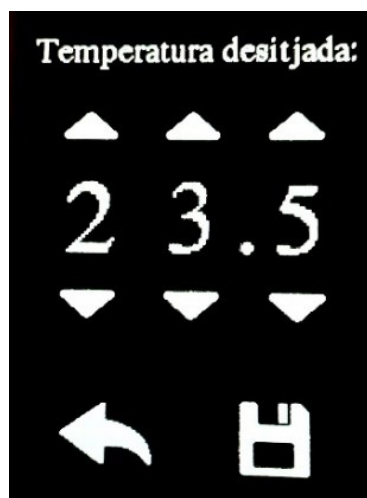
Figura 56: Pantalla del menú del mode manual que apareix en el quadre de comandaments.



autoPage()

El menú mode automàtic es pot observar en la Figura 57, on es pot establir la temperatura desitjada a través de la modificació de les desenes, unitats i decimals. Al igual que en el menú manual, es guarda la temperatura pressionant la icona en forma de disquet, i pressionant la fletxa baix a l'esquerra es pot tornar al menú configuració.

Figura 57: Pantalla del menú mode automàtic que es mostra en el display.



messageErrorPage()

En canviar la variable "missatgeError" a un número diferent del zero, apareix un missatge com el que es pot veure en la Figura 58, aquest desapareix quan es pressiona qualsevol part de la pantalla.

Figura 58: Missatge d'error de parada inesperada del motor.



OnDataRecv()

Aquesta es una tasca aperiòdica, ja que s'activa quan es rep un missatge des del controlador.

A l' inici es guarda la informació rebuda en la variable local creada "dataFromController", de tipus struct_dataFromController, a continuació es sobreescriu en la cua queueIsControllerOnline, mitjançant xQueueOverwrite(), per a notificar que s'ha rebut un missatge des del controlador. Posteriorment, envia les dades rebudes a la cua "queueDataFromController", per a actualitzar les dades.

Per a la notificació es podria haver utilitzat una notificació, a través de xTaskNotify(), no obstant, l'ús d'una cua d'un únic valor requereix menys capacitat de còmput, que es trasllada a una major eficiència de l'ús del processador.

Tasca periòdica crítica

timerWirelessCommunication()

Aquesta tasca és una tasca crítica, la qual s'activa cada 5 segons. El seu objectiu es comprovar si ha arribat un missatge del controlador. Si ha arribat, a la cua hi haurà un valor, i no es farà res més, en canvi, si la cua està buida, significa que no s'ha rebut un missatge des de l'última vegada que s'ha comprovat la cua. De manera que la variable "missatgeError" és actualitzada a tres, indicant un error de que no es rep comunicació des del controlador.

6.3.3. Prioritats

S'ha establert com la tasca més prioritària la tasca crítica, ja que comprova si existeix comunicació entre el quadre de comandaments i el controlador. I la menys prioritària és la tasca que controla la pantalla, no perquè no siga important, sinó perquè és la que sempre se està executant, de manera que si fora la més prioritària no deixaria temps de còmput per a les altres.

6.3.4. Diagrames de flux de les tasques

En les figures Figura 59, Figura 60 i Figura 61, es mostren els diagrames de flux de les tasques que s'executen en el panell de control.

Figura 59: Diagrama de flux de la tasca *OnDataRecv()*.

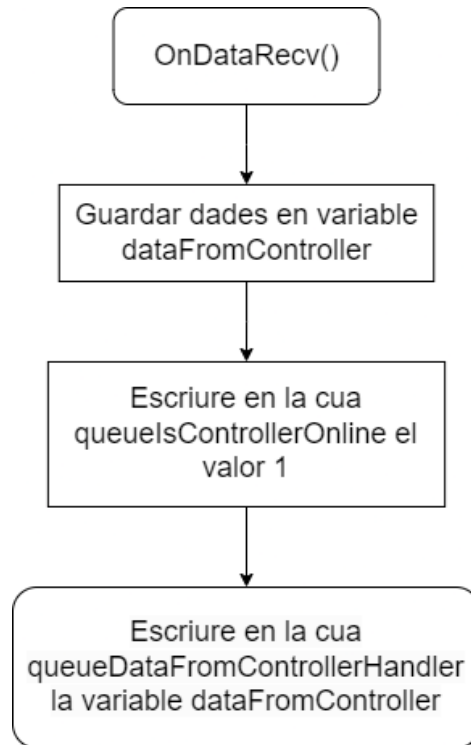


Figura 60: Diagrama de flux de la tasca *timerWirelessCommunication()*.

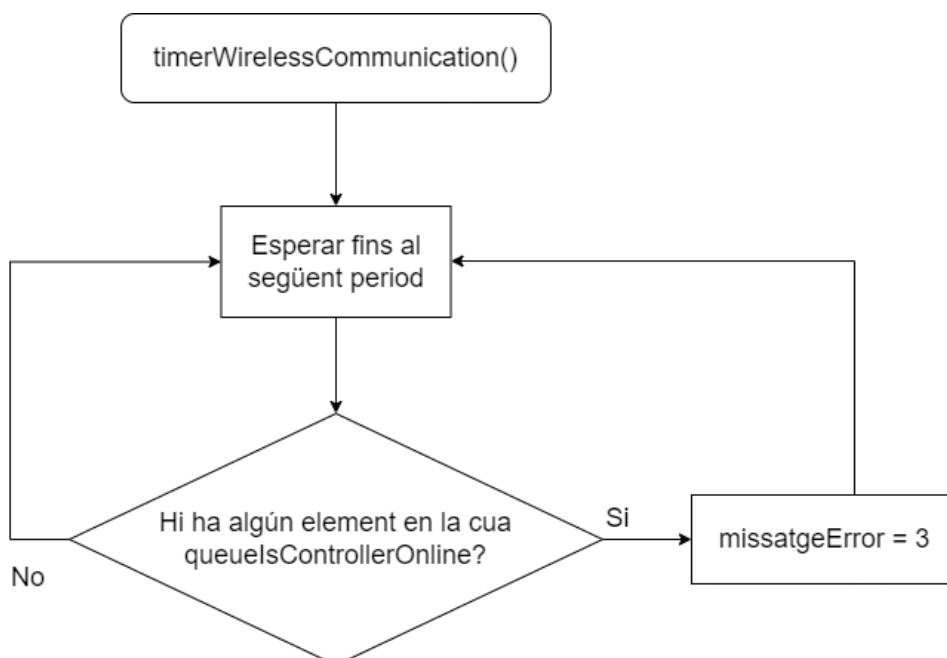
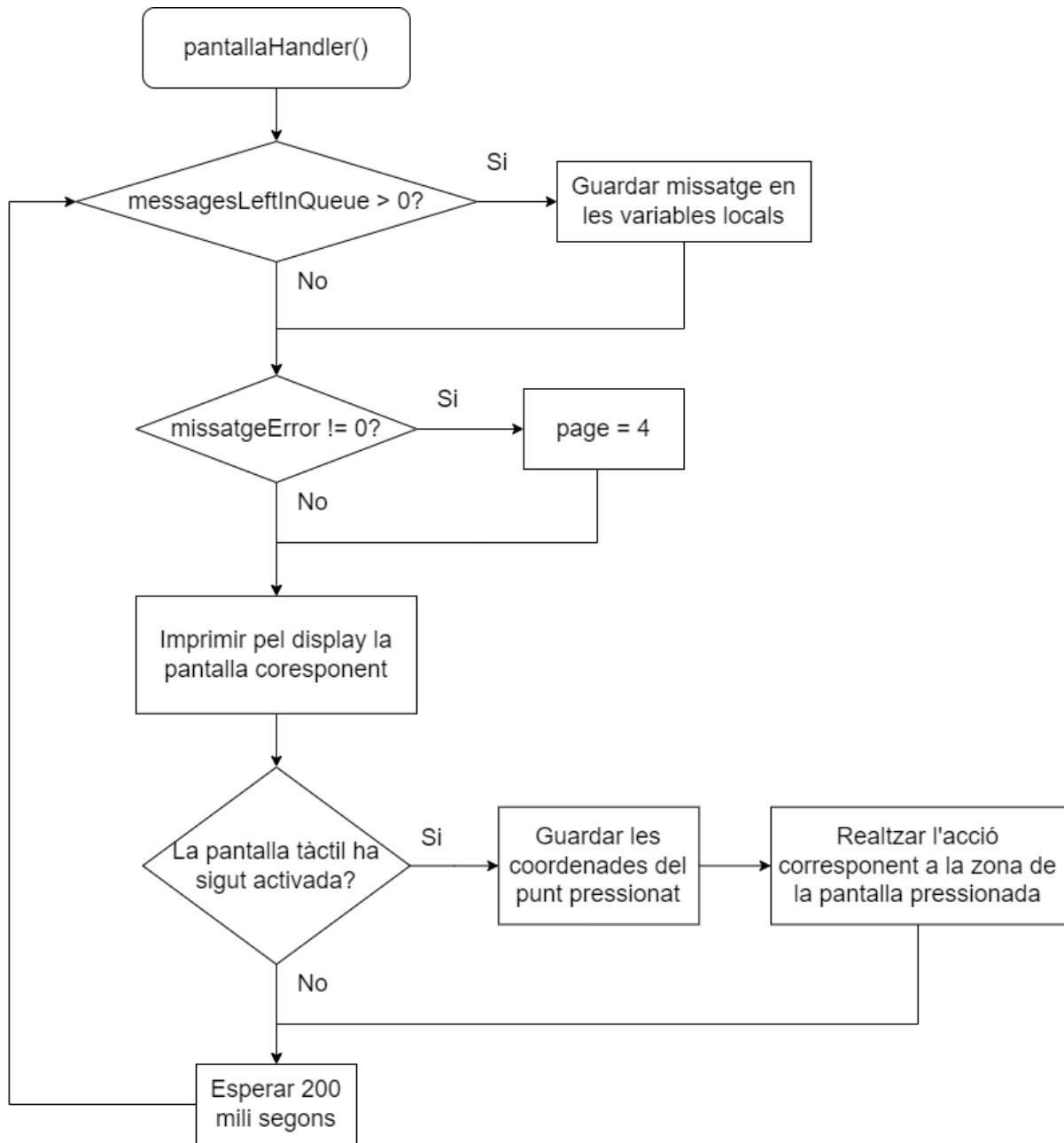


Figura 61: Diagrama de flux de la tasca pantallaHandler().





7. Conclusions

En aquest projecte s'ha desenvolupat un sistema intel·ligent, implementat un controlador i un quadre de comandaments interactiu. L'ús del microcontrolador ESP32-WROOM-32 ha demostrat ser una solució eficient i econòmica, proporcionant la capacitat necessària per a gestionar les tasques del sistema, com el control del motor, la monitorització dels sensor i la gestió de l'enviament i rebuda de dades, mitjançant el sistema operatiu FreeRTOS, que ha permès una gestió efectiva de les tasques en temps real, assegurant que les funcions crítiques tinguen la prioritat necessària. Gràcies al protocol ESP-NOW, és possible una comunicació remota, ràpida, amb baixa latència, i consumint pocs recursos.

S'han seleccionat els components òptims i més econòmics per a la creació del sistema, combinant accessibilitat, eficiència i robustesa, de manera que es facilita l'actualització d'un ventilador tradicional, automatitzant-lo, millorant la seua funcionalitat i allargant la seua vida útil, per a aportar major confort i comoditat amb la única contaminació generada que l'energia elèctrica que consumeix. De manera que es pot dir que els objectius d'aquest projecte han sigut superats de manera satisfactòria.

Les millores que es podrien introduir en quant al panell de control podria ser la incorporació d'una bateria, per fer-lo portàtil, la implementació de més funcionalitats, com la de visualitzar l'historial de variables i missatges d'errors. En relació al dispositiu de ventilació el disseny del controlador del motor es podria millorar, mitjançant un controlador PID, si es disposa de la funció de transferència del motor. I respecte al sistema de ventilació es podrien agregar més funcionalitats, com per exemple connectivitat a la xarxa WiFi per a poder ser controlat des del mòbil a través d'una pàgina web o a través d'una aplicació, també es podria dissenyar per a que funcionara amb el protocol Matter per a permetre el seu control a través d'altres dispositius intel·ligents, com Alexa d'Amazon, Siri d'Apple o Google Home Assistant de Google, una altra opció podria ser incloure un servidor TCP/IP controlat per un ordinador, per a poder veure un historial de les variables al llarg del temps i el missatges d'errors que han aparegut, i implementar un algoritme que detecte possibles futures falles del motor i poder anticipar-se.

Finalment, aquest projecte ha permès la possibilitat de desenvolupar una idea i portar-la a terme, a més de poder aplicar tots els coneixements adquirits durant els anys d'aprenentatge en el grau d'Enginyeria Electrònica Industrial i Automàtica, des dels coneixements de programació de sistemes embebuts com l'elecció adequada dels components electrònics, i el disseny d'un controlador. Les etapes transcorregudes al llarg del desenvolupament del projecte han permès tindre una idea d'allò que és realitzar un projecte d'enginyeria.



8. Bibliografia

- [1] A. Ruiz i A. Somanilla, «El Confidencial,» 10 març 2023. [En línia]. Available: https://www.elconfidencial.com/medioambiente/clima/2023-10-03/aumento-temperaturas-aire-acondicionado_3746530/. [Últim accés: maig 2024].
- [2] F. Martín León, «Tiempo,» 30 juny 2022. [En línia]. Available: <https://www.tiempo.com/ram/sobreuso-aire-acondicionados-aumento-emisiones-de-gases-efecto-invernadero.html>. [Últim accés: maig 2024].
- [3] N. A. Muiños Fernandez, «Generali,» sense data. [En línia]. Available: <https://www.generali.es/blog/tuasesorsalud/efectos-aire-acondicionado-salud/>. [Últim accés: maig 2024].
- [4] S. Khatsenkova, «Euro news,» 28 agost 2023. [En línia]. Available: <https://es.euronews.com/green/2023/08/28/the-cube-el-aire-acondicionado-calienta-mas-las-ciudades>. [Últim accés: maig 2024].
- [5] P. Martínez Juárez, «Xataka,» 22 juny 2022. [En línia]. Available: <https://www.xataka.com/energia/aire-acondicionado-ventilador-que-dice-ciencia-mejor-metodo-para-enfriarnos-calor>. [Últim accés: maig 2024].
- [6] M. Scott, «The Tibble,» maig 2022. [En línia]. Available: <https://thetibble.com/average-ceiling-fan-speed/>. [Últim accés: maig 2024].
- [7] Espressif Systems®, «Espressif Systems®,» 2023. [En línia]. Available: www.espressif.com
https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [Últim accés: abril 2024].
- [8] Liusasmue, «Amazon,» 2023. [En línia]. Available: <https://www.amazon.es/ESP32-DevKitC-32-ESP-WROOM-32-Procesador-actualizado-Desarrollo/dp/B0CMSC7XS3>. [Últim accés: febrer 2024].
- [9] Texas Instruments, «Texas Instruments,» abril 2024. [En línia]. Available: https://www.ti.com/lit/ds/symlink/txs0108e.pdf?ts=1718881928084&ref_url=https%253A%252F%252Fwww.google.com%252F. [Últim accés: maig 2024].
- [10] AdaFruit, «Mouser,» 8 març 2024. [En línia]. Available: <https://www.mouser.com/datasheet/2/737/dht-932870.pdf>. [Últim accés: maig 2024].
- [11] Texas Instruments, «Texas Instruments,» desembre 2015. [En línia]. Available: https://www.ti.com/lit/ds/symlink/ina219.pdf?ts=1718870706175&ref_url=https%253A%252F%252Fwww.google.com%252F. [Últim accés: juny 2024].
- [12] Songle relay, «Songle relay,» 2019. [En línia]. Available: <https://html.alldatasheet.es/html-pdf/1132639/SONGLERELAY/SRD-05VDC-SL-C/1713/2/SRD-05VDC-SL-C.html>. [Últim accés: juny 2024].



- [13] ZENG WHCD, «ZENG WHCD,» sense data. [En línia]. Available: https://es.aliexpress.com/item/1005005955881251.html?spm=a2g0o.order_list.order_list_m ain.45.62d8194dF8MLpM&gatewayAdapt=glo2esp. [Últim accés: febrer 2024].
- [14] BTF-LIGHTING, «Amazon,» 2018. [En línia]. Available: <https://www.amazon.es/dp/B01D8FM3W6?psc=1>. [Últim accés: febrer 2024].
- [15] parpeCejas, «Thingiverse,» maig 2024. [En línia]. Available: <https://www.thingiverse.com/thing:6637589>. [Últim accés: maig 2024].
- [16] Apklvsr, «Amazon,» gener 2024. [En línia]. Available: <https://www.amazon.es/dp/BOCSYPG716?psc=1>. [Últim accés: juny 2024].
- [17] mdkendall, «Printables,» gener 2024. [En línia]. Available: <https://www.printables.com/es/model/685845-enclosure-for-sunton-esp32-2432s028r-cheap-yellow->. [Últim accés: juliol 2024].
- [18] Barry, Richard; FreeRTOS Team, «FreeRTOS,» 2023. [En línia]. Available: <https://www.freertos.org/Documentation/Mastering-the-FreeRTOS-Real-Time-Kernel.v1.0.pdf>. [Últim accés: maig 2024].
- [19] Espressif Systems, «docs espressif,» 2024. [En línia]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_now.html. [Últim accés: juny 2024].
- [20] «image.online-convert.com,» [En línia]. Available: <https://image.online-convert.com/convert/jpg-to-bmp>. [Últim accés: juny 2024].
- [21] «online-utility.org,» [En línia]. Available: <https://www.online-utility.org/image/convert/to/XBM>. [Últim accés: juny 2024].
- [22] Anònim, «<https://iconduck.com/>,» 31 Desembre 2020. [En línia]. Available: <https://iconduck.com/icons/22267/power-off>. [Últim accés: juny 2024].
- [23] Á. Guitérrez, B. Lagarra y F. Monasterui-Huelin, «[robolabo.upm.es](http://www.robolabo.etsit.upm.es/),» 22 gener 2024. [En línia]. Available: <http://www.robolabo.etsit.upm.es/>. [Último acceso: juliol 2024].

València, juliol de 2024

Grau en Enginyeria Electrònica Industrial i Automàtica

Júlia Arribas Ríos



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ETSI Aeroespacial y Diseño Industrial



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA
AEROESPACIAL I DISSENY INDUSTRIAL

Automatització i control remot d'un sistema de ventilació

Treball Fi de Grau

Grau en Enginyeria Electrònica Industrial i Automàtica

Annex N°1: Manual d'usuari

Autora:
Arribas Ríos, Júlia Mei

Tutor:
Rodríguez Ballester, Francisco

Curs acadèmic 2023/2024



Índex de l'annex nº 1

1. Introducció.....	72
2. Iniciació al dispositiu	72
3. Guia d'inici.....	72
4. Funcionament	73
4.2. Pantalla d'inici.....	73
4.2. Pantalla de configuració o menú	74
4.3. Pantalla del mode manual	75
4.4. Pantalla del mode automàtic	75
4.5. Pantalla de missatge d'error	76

Índex d'imatges del annex nº 1

Figura 1: Monitor serial que mostra l'inici del programa que controla la pantalla i mostra la adreça MAC.....	73
Figura 2: Programa del controlador, que mostra la constant "broadcastAdress"	73
Figura 3: Pàgina principal del panell de control.....	74
Figura 4: Pantalla de configuració del panell de control	74
Figura 5: Pantalla del mode manual, per a configurar la velocitat desitjada	75
Figura 6: Pantalla del mode automàtic, per a configurar la temperatura desitjada	76
Figura 7: Pantalla de missatge d'error	77



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ETSI Aeroespacial y Diseño Industrial



1. Introducció

Aquest dispositiu de ventilació està dissenyat per a millorar la comoditat de l'usuari mitjançant la regulació de la temperatura i el control de la velocitat del ventilador. A continuació, es relata una guia bàsica que explica el funcionament del sistema, no es tractarà el muntatge del dispositiu.

El sistema compta de dues parts, el dispositiu de ventilació, que conté el motor, els sensors i el microcontrolador; i el panell de control, que consta d'una pantalla connectada a un altre microcontrolador.

2. Iniciació al dispositiu

Per a alimentar el dispositiu de ventilació s'ha de connectar el cable d'alimentació a la xarxa elèctrica, també cal connectar la placa d'expansió a la llum a través d'un cable micro USB a USB-A o C i un transformador compatible de 5 V.

Per a alimentar el panell de control s'ha de connectar un cable micro USB a USB-A o C i un transformador compatible de 5 V.

3. Guia d'inici

Material requerit:

- Microcontrolador del dispositiu de ventilació
- Microcontrolador del panell control
- Un cable micro USB a USB-A o USB-C (segons siga compatible amb l'ordinador utilitzat)
- Ordinador

Per a emparellar els dispositius s'ha de saber primer la seua adreça MAC, per a açò s'han de connectar els dos microcontroladors a un ordinador que tinga instal·lat un programa que permeta veure el Serial port, com l'Arduino IDE.

Una vegada connectat un microcontrolador, s'ha de establir la velocitat de transferència a 115200 baud, com mostra la Figura 1, es pot observar que enquadrat taronja s'indica la velocitat de transferència i enquadrat en verd es mostra l'adreça MAC de la pantalla, aquest número cal introduir-lo en el programa del controlador, en la constant "broadcastAdress", com es pot veure en la Figura 2, enquadrat en roig. El mateix procés s'ha de seguir per a saber la MAC de l'altre dispositiu.

A continuació, ja es poden endollar a la llum.

Figura 1: Monitor serial que mostra l'inici del programa que controla la pantalla i mostra la adreça MAC.

The screenshot shows a serial monitor window titled "Output Serial Monitor x". The baud rate is set to 115200. The output text includes boot parameters and a MAC address: "10:06:1C:81:DF:A0".

```
ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configisp: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1288
load:0x40078000,len:13872
load:0x40080400,len:4
ho 8 tail 4 room 4
load:0x40080404,len:3048
entry 0x40080590
Transceiver initialized 10:06:1C:81:DF:A0
----- START -----
```

Figura 2: Programa del controlador, que mostra la constant "broadcastAdress".

The screenshot shows a code editor with the file name "pregramaControlador7.ino". The code defines a MAC address array named "broadcastAddress".

```
pregramaControlador7.ino
17
18 DHT22 dht22(temperatureSensorPin);
19
20 Adafruit_INA219 ina219;
21
22 // adreça MAC del ESP que controla la pantalla
23 const uint8_t broadcastAddress[] = { 0x10, 0x06, 0x1C, 0x81, 0xDF, 0xA0 };
24
25 //velocitat màxima del motor
26 const float velocitatMax = 300.0;
27 const float correntMax = 1000.0; // mA
28
29 // màxim d'elements que pot tindre una cua
```

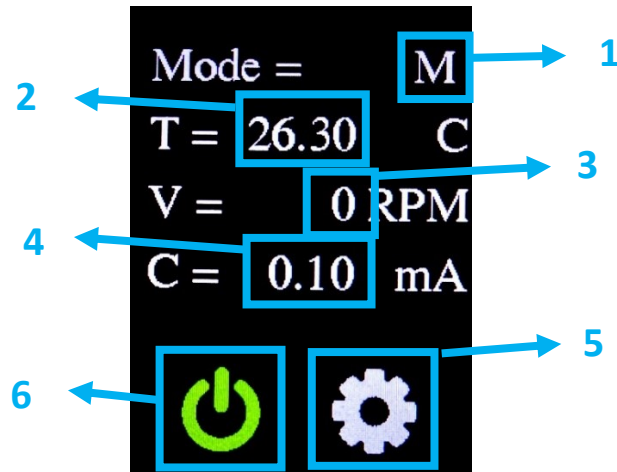
4. Funcionament

4.2. Pantalla d'inici

En iniciar el panell de control apareix una pantalla com es mostra en la Figura 3, on apareixen diferents elements que es descriuen a continuació:

1. Mode actual (M: mode manual | A: mode automàtic)
2. Temperatura actual en graus centígrads
3. Velocitat del motor actual en revolucions per minuts (RPM)
4. Corrent actual del motor en mili amperis (mA)
5. Botó per a entrar a la pantalla de configuració o menú
6. Botó d'encès i apagat del motor (verd: encès | roig: apagat)

Figura 3: Pàgina principal del panell de control.

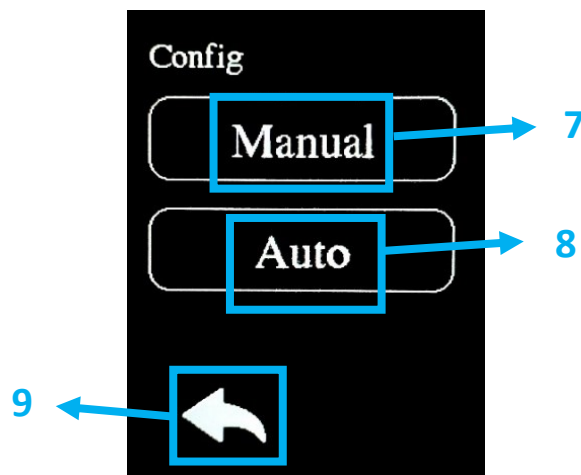


4.2. Pantalla de configuració o menú

En aquesta pantalla, que es mostra en la Figura 4, es pot seleccionar el mode desitjat, les diferents opcions que es poden seleccionar es descriuen a continuació:

7. Manual: botó per a seleccionar la velocitat desitjada
8. Auto: botó per a seleccionar la temperatura desitjada
9. Botó per a tornar a la pàgina anterior

Figura 4: Pantalla de configuració del panell de control.



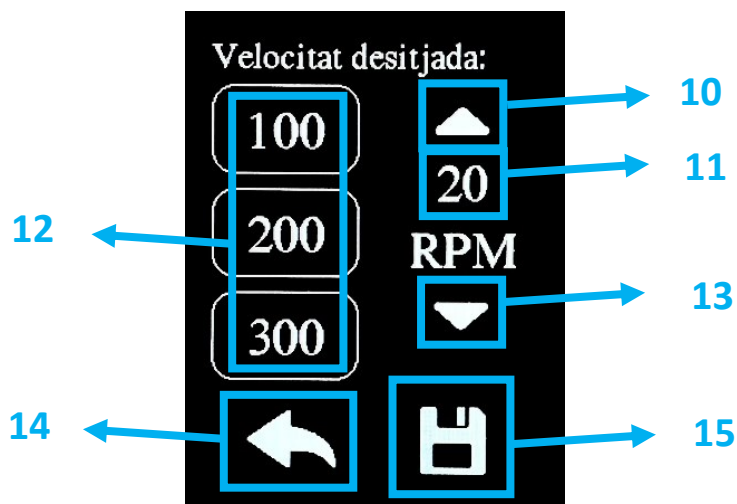
4.3. Pantalla del mode manual

Aquesta pantalla permet seleccionar la velocitat desitjada, a través dels diferents botons, que es poden veure en la Figura 5, es pot determinar la velocitat. Per a guardar el valor desitjat s'ha de pressionar el botó de guardar. En cas de no voler guardar-lo es pot pressionar el botó tornar.

Els elements de la pantalla es descriuen a continuació:

10. Botó per a pujar la velocitat desitjada 10 RPM
11. Velocitat desitjada establerta
12. Botons per a establir la velocitat desitjada en 100, 200 o 300 RPM
13. Botó per a baixar la velocitat desitjada 10 RPM
14. Botó per a tornar a la pàgina anterior, pàgina del menú (si es pressiona aquest botó la velocitat desitjada establerta no es guardarà)
15. Botó per a guardar la velocitat desitjada

Figura 5: Pantalla del mode manual, per a configurar la velocitat desitjada.



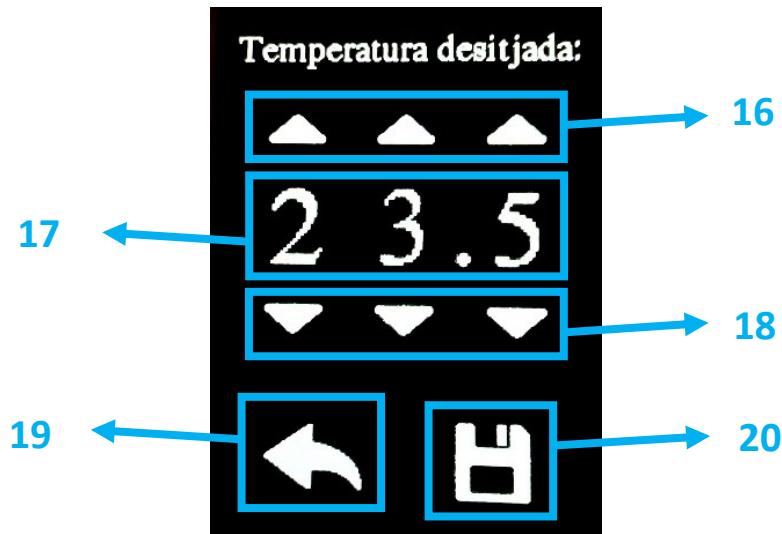
4.4. Pantalla del mode automàtic

En aquesta pàgina es pot seleccionar la temperatura desitjada, mitjançant els botons que es mostren en la Figura 6. Una vegada establert el valor volgut es pot guardar mitjançant el botó "guardar", si no es vol guardar la xifra, es pot seleccionar el botó tornar.

Descripció dels elements d'aquesta pantalla:

16. Botó per a pujar les desenes, unitats i centenes individualment
17. Temperatura desitjada en graus centígrads
18. Botó per a baixar les desenes, unitats i centenes individualment
19. Botó per a tornar a la pàgina anterior, al menú de configuració
20. Botó per a guardar la temperatura desitjada establerta

Figura 6: Pantalla del mode automàtic, per a configurar la temperatura desitjada.



4.5. Pantalla de missatge d'error

Aquesta pantalla apareix per a notificar un ús anòmal del sistema, per a notificar l'error detectat. Poden aparèixer els següents missatges:

- Parada inesperada del motor. Aquest error sorgeix en cas de que el senyal de velocitat siga major que 0, però l' encoder del motor no detecta cap pols. En aquest cas, es deuria de revisar l'estat del ventilador, desconnectant-lo de la llum i comprovant que no està obstruït. També es pot donar el cas de què algun cable estiga solt, de manera que és recomanable comprovar que tots els cables estiguen connectats.
- Excés de corrent pel motor. Aquest error sorgeix en cas de què es detecte una corrent superior a 1 A, en aquest cas, es deté el motor automàticament, és convenient revisar el circuit electrònic i comprovar que no hi haja cap cable solt o defectuós.
- No es rep comunicació del controlador. Aquest error sorgeix en cas de què no es reba informació des del controlador des dels últims cinc segons. En aquest cas es deuria comprovar que el controlador del motor estiga encès, si no ho està, s'hauria d'encendre connectant-lo a la llum, en cas de que estiga encès, es deu assegurar de l'adreça d'emissió ("broadcastAdress") siga correcta en ambdós microcontroladors. En cas de que el problema persistisca s'hauria de comprovar que els microcontroladors puguen enviar i rebre dades correctament.



Figura 7: Pantalla de missatge d'error.



València, juliol de 2024

Grau en Enginyeria Electrònica Industrial i Automàtica

Júlia Arribas Ríos



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA


ETSI Aeroespacial y Diseño Industrial

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA
AEROESPACIAL I DISSENY INDUSTRIAL

Automatització i control remot d'un sistema de ventilació

Treball Fi de Grau

Grau en Enginyeria Electrònica Industrial i Automàtica

Annex N^o2: Objectiu i desenvolupament sostenible

Autora:
Arribas Ríos, Júlia Mei

Tutor:
Rodríguez Ballester, Francisco

Curs acadèmic 2023/2024



Índex de l'annex nº 2

1. Objectius del desenvolupament sostenible.....	81
2. Objectius del desenvolupament sostenible del projecte	81

Índex de taules de l'annex nº 2

Taula 1: Objectius de desenvolupament sostenible que compleix el projecte de automatització i control remot d'un sistema de ventilació.....	82
---	----



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ETSI Aeroespacial y Diseño Industrial



1. Objectius del desenvolupament sostenible

L'agenda 2030 per al desenvolupament sostenible, adoptada per tots els Estats membres de les Nacions Unides en setembre del 2015, és un ambiciós pla global que busca abordar alguns dels reptes més urgents que enfronta la humanitat. En el nucli d'aquesta agenda es troben els 17 Objectius de Desenvolupament Sostenible (ODS), dissenyats per a ser una fulla de ruta que abaste des de l'erradicació de la pobresa i la fam, fins la promoció de la igualtat de gènere, l'acció contra el canvi climàtic i la construcció de societats més justes i sostenibles.

Els ODS són una crida universal a l'acció, destinats a guiar els esforços nacionals i internacionals cap a un futur més equitatiu, inclusiu i respectuós amb el medi ambient. Cadascun dels objectius aborda un aspecte específic del desenvolupament sostenible i està interconnectat amb els demés, reconeixent la complexitat i la interdependència dels reptes globals. La Agenda 2030 representa un compromís col·lectiu per a transformar el nostre món, promovent el progrés econòmic, social i ambiental de manera equitativa assegurant que ningú es quede enrere.

2. Objectius del desenvolupament sostenible del projecte

Aquest projecte no contempla la contribució directa a l'objectiu número 1. Fi de la pobresa; ni el número 2. Fam zero; ni el número 4. Educació de qualitat; ni el 5. Igualtat de gènere; ni el número 6. Aigua neta i sanejament; ni el 8. Treball decent i creixement econòmic; ni el 16. Pau, justícia i institucions sòlides, ni la 17. Aliances per aconseguir els objectius, ja que aquests tracten temes socioeconòmics, i aquest projecte està més enfocat en un àmbit tecnològic per a millorar la qualitat de vida de les persones d'una manera assequible.

L'aportació d'aquest treball a l'objectiu 7. Energia assequible i no contaminant, tampoc és apreciable, ja que amb l'ús de l'aparell dissenyat no es pretén generar energia, sinó consumir-la.

De la mateixa manera que tampoc col·labora amb els objectius 14. Vida submarina, 15. Vida d'ecosistemes terrestres, perquè no aborda temes relacionats amb aquests.

La contribució d'aquest projecte sí és apreciable amb l'objectiu número 3. Salut i benestar, ja que mitjançant l'ús de ventiladors permet disminuir la sensació de calor, sense deixar seqüeles. A diferència de l'aire condicionat, el qual pot arribar a ser perjudicial per a la salut.

També contribueix notablement a l'objectiu 11. Ciutats i comunitats sostenibles; i a l'objectiu 12. Producció i consum responsables, ja que aquest projecte està pensat per a la reutilització d'algunes peces i el poder allargar la vida útil dels aparells de ventilació tradicionals, evitant així que es generen més residus, i per la no compra de més electrodomèstics.

L'objectiu número 9. Indústria, innovació i infraestructures, es cobert al implementar funcionalitats que permeten una connectivitat remota i processos d'automatització, el que possibilita un ús eficient de la energia, cobrint l'objectiu número 13. Acció pel clima, aquest objectiu també es complit pel fet



de què la substitució d'aire condicionat per altres aparells menys contaminants, ja que d'aquesta manera no afavoreixen l'escalfament global.

L'elecció dels materials i components d'una manera econòmica fa que aquest projecte puga ser realitzat per persones de qualsevol nivell econòmic, podent accedir a aparells automatitzats d'una manera assequible, contribuint amb l'objectiu número 10, reducció de les desigualtats.

En la Taula 1, apareix una recapitulació de les aportacions d'aquest projecte als objectius de desenvolupament sostenible.

Taula 1: Objectius de desenvolupament sostenible que compleix el projecte de automatització i control remot d'un sistema de ventilació.

Objectius de desenvolupament sostenible	Gens apreciable	Apreciable	Prou apreciable	Molt apreciable
1. Fi de la pobresa	X			
2. Fam zero	X			
3. Salut i benestar		X		
4. Educació de qualitat	X			
5. Igualtat de gènere	X			
6. Aigua neta i sanejament	X			
7. Energia assequible i no contaminant	X			
8. Treball decent i creixement econòmic	X			
9. Indústria, innovació i infraestructures			X	
10. Reducció de les desigualtats				X
11. Ciutats i comunitats sostenibles.				X
12. Producció i consum responsables				X
13. Acció pel clima			X	
14. Vida submarina	X			
15. Vida d'ecosistemes terrestres	X			
16. Pau, justícia i institucions sòlides	X			
17. Aliances per aconseguir els objectius	X			

València, juliol de 2024

Grau en Enginyeria Electrònica Industrial i Automàtica

Júlia Arribas Ríos



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA
AEROESPACIAL I DISSENY INDUSTRIAL

Automatització i control remot d'un sistema de ventilació

Treball Fi de Grau

Grau en Enginyeria Electrònica Industrial i Automàtica

Document N°2: Plànols

Autora:
Arribas Ríos, Júlia Mei

Tutor:
Rodríguez Ballester, Francisco

Curs acadèmic 2023/2024



Índex dels plànols

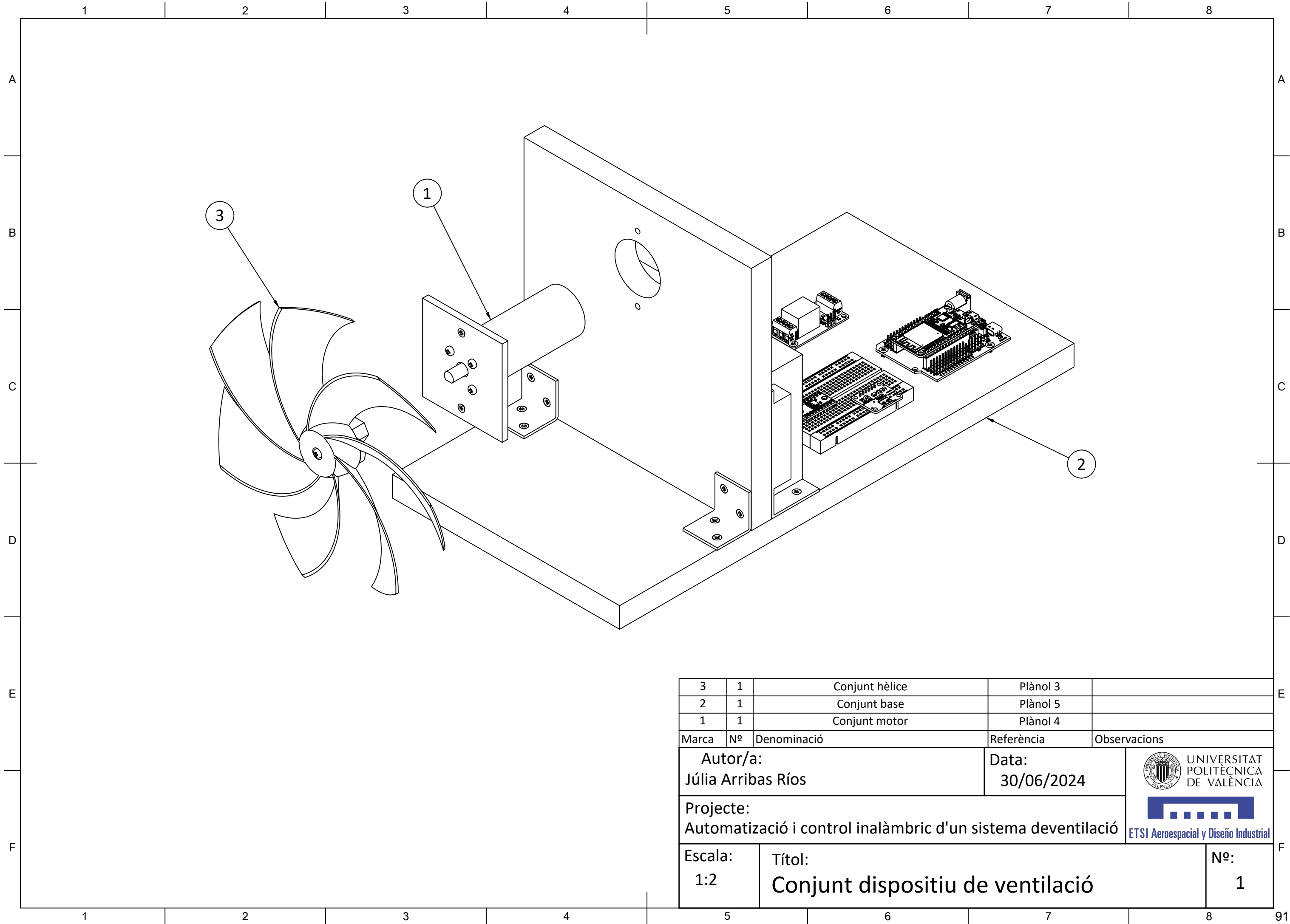
1.	Plànol 1 – Conjunt dispositiu de ventilació	86
2.	Plànol 2 – Conjunt panell de control.....	88
3.	Plànol 3 – Conjunt panell de control.....	90
4.	Plànol 4 – Conjunt motor	92
5.	Plànol 5 – Conjunt base.....	94
6.	Plànol 6 – Conjunt estructura	96
7.	Plànol 7 – Placa	98
8.	Plànol 8 – Peça suport.....	100
9.	Plànol 9 – Peça base.....	102





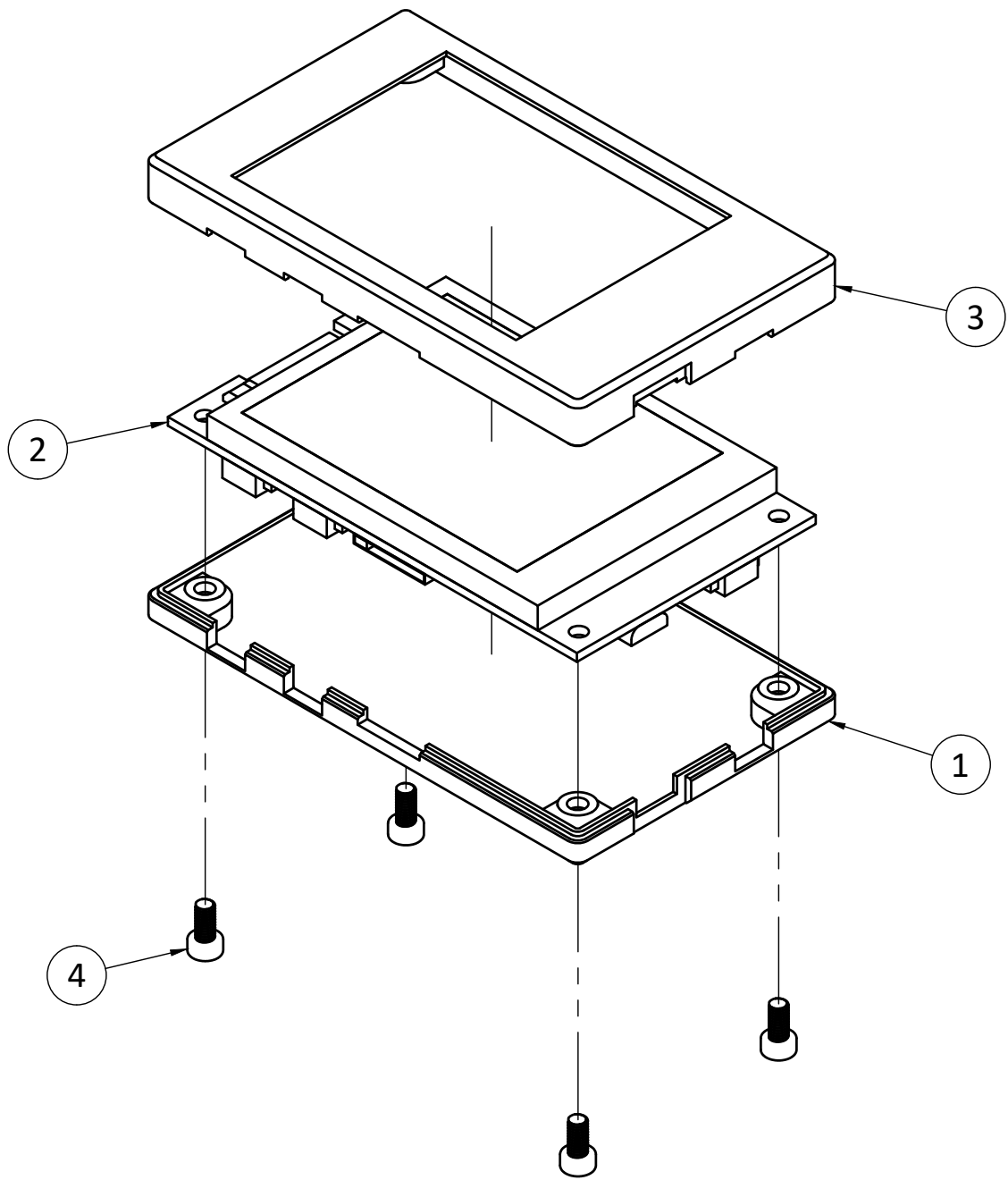
UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



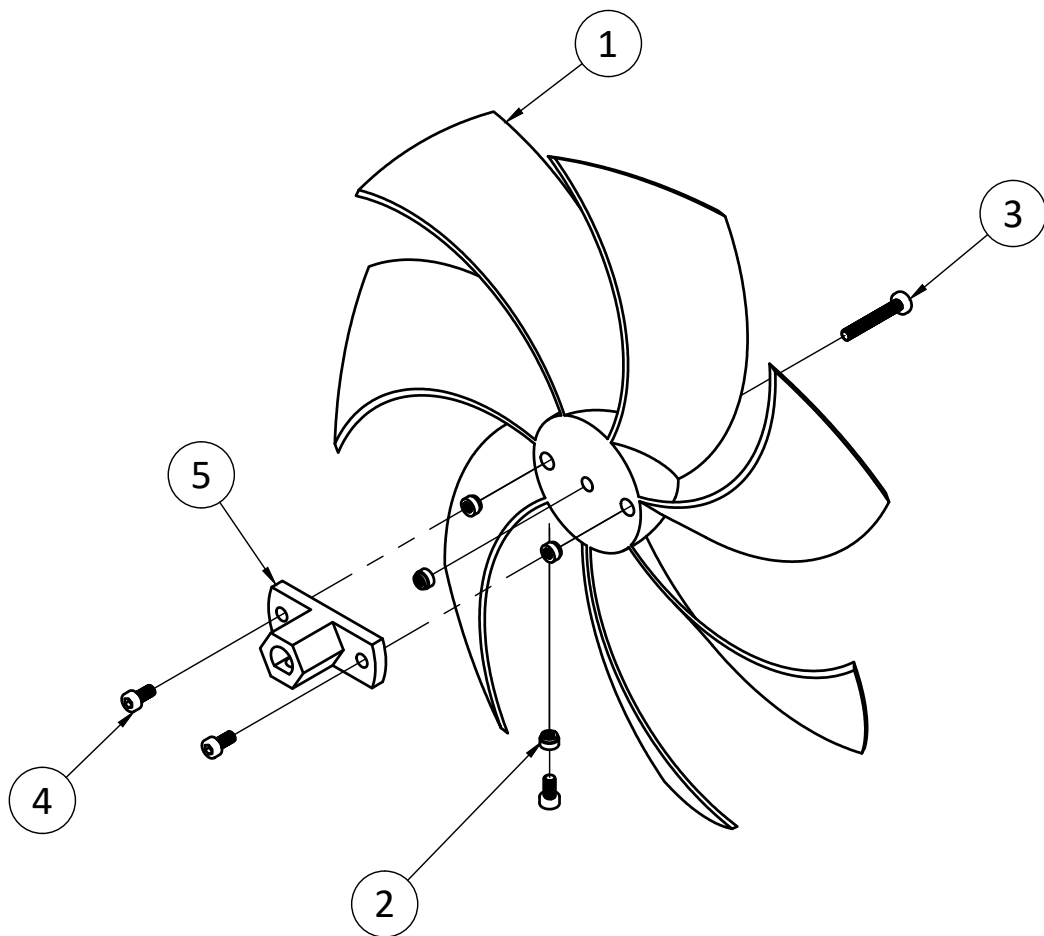
ETSI Aeroespacial y Diseño Industrial



3	1	Conjunt hèlice	Plànol 3	
2	1	Conjunt base	Plànol 5	
1	1	Conjunt motor	Plànol 4	
Marca	Nº	Denominació	Referència	Observacions
Autor/a: Júlia Arribas Ríos			Data: 30/06/2024	
Projecte: Automatització i control inalàmbric d'un sistema de ventilació			 UNIVERSITAT POLITÈCNICA DE VALÈNCIA  ETSI Aeroespacial y Diseño Industrial	
Escala: 1:2		Títol: Conjunt dispositiu de ventilació		





4	4	Cargols M3x6	ISO 4762	Acer inoxidable
3	1	Peça superior de la carcassa	Model 3D	PLA
2	1	Pantalla + Controlador	ASIN: B0CSYPG716	
1	1	Peça inferior de la carcassa	Model 3D	PLA
Marca	Nº	Denominació	Referència	Observacions
Autor/a: Júlia Arribas Ríos			Data: 30/06/2024	
Projecte: Automatització i control inalàmbric d'un sistema de ventilació			 UNIVERSITAT POLITÈCNICA DE VALÈNCIA  ETSI Aeroespacial y Diseño Industrial	
Escala: 1:1	Títol: Conjunt panell de control			



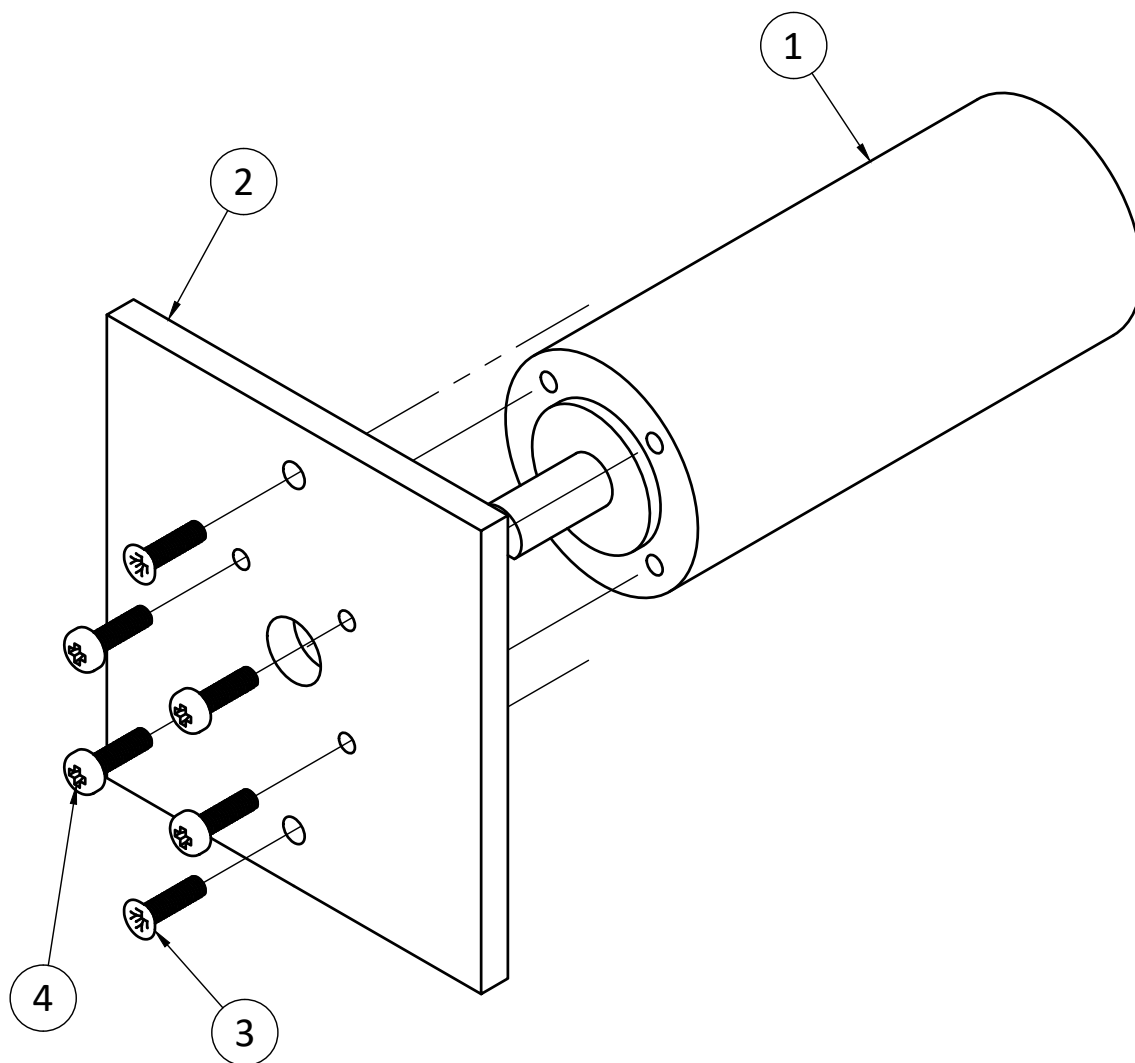
5	1	Peça unió eix aspes	Model 3D	PLA
4	3	Cargol M3x6	ISO 4762	Acer inoxidable
3	1	Cargol M3x20	ISO 4762	Acer inoxidable
2	4	Rosca d'inserció M3	ISO 5821	Llautó
1	1	Peça hèlix	Model 3D	PLA

Marca	Nº	Denominació	Referència	Observacions
-------	----	-------------	------------	--------------

Autor/a: Júlia Arribas Ríos		Data: 30/06/2024		 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
---------------------------------------	--	----------------------------	--	---


Projecte: Automatització i control inalàmblic d'un sistema de ventilació			 ETSI Aeroespacial y Diseño Industrial
--	--	--	--


Escala: 1:2	Títol: Conjunt hèlix	Nº: 3
-----------------------	--------------------------------	-----------------



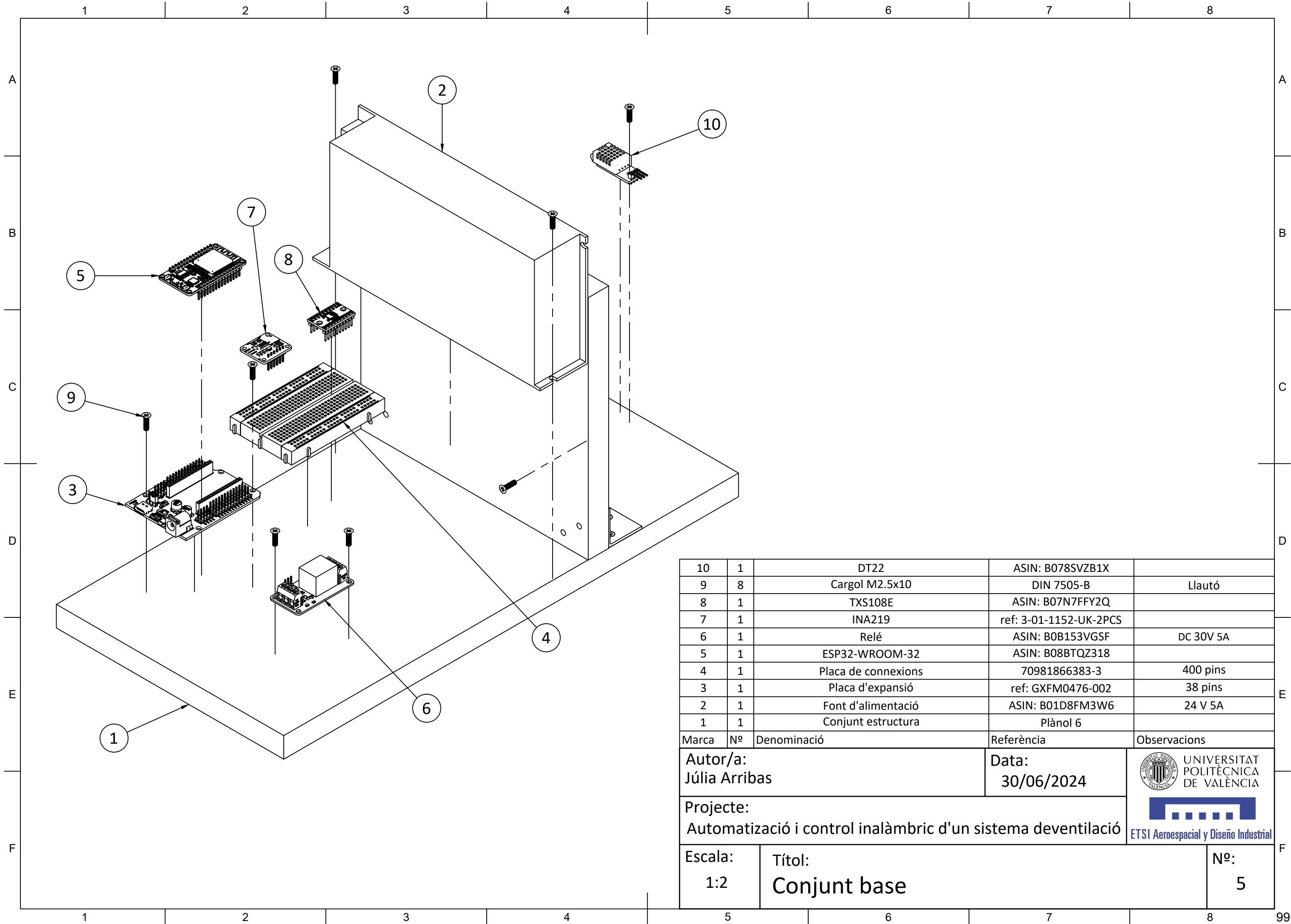
4	4	Cargol M3x10	ISO 5821	Acer inoxidable
3	2	Cargol M3x15	ISO 5821	Acer inoxidable
2	1	Placa	Plànol 7	
1	1	Motor	36GP-3650	24 V 421 RPM

Marca	Nº	Denominació	Referència	Observacions
-------	----	-------------	------------	--------------

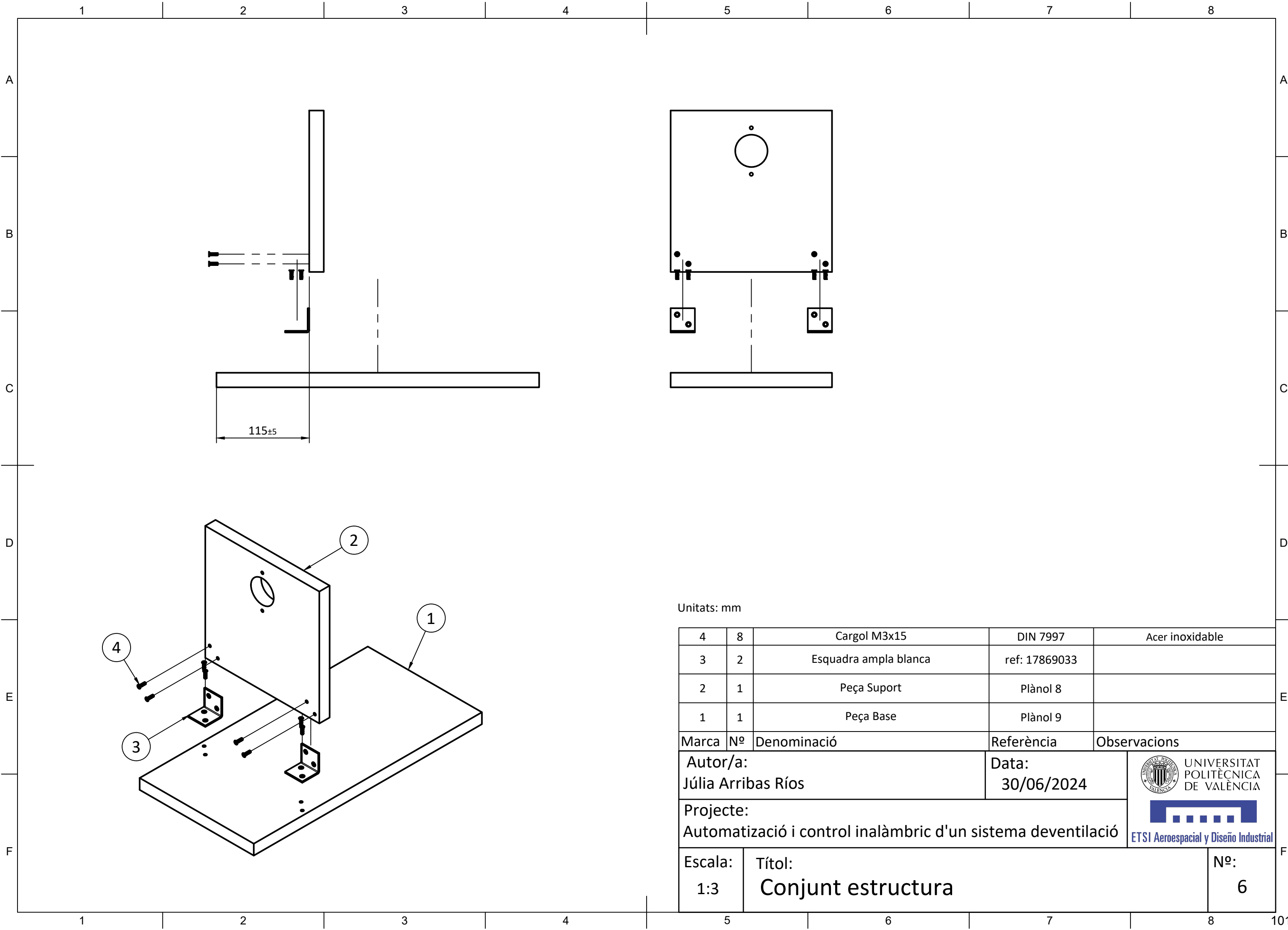
Autor/a: Júlia Arribas Ríos		Data: 30/06/2024		 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
--------------------------------	--	---------------------	--	---

Projecte: Automatització i control inalàmbric d'un sistema de ventilació			 ETSI Aeroespacial y Diseño Industrial	
---	--	--	--	--



Escala: 1:1	Títol: Conjunt Motor	Nº: 4
----------------	--------------------------------	----------

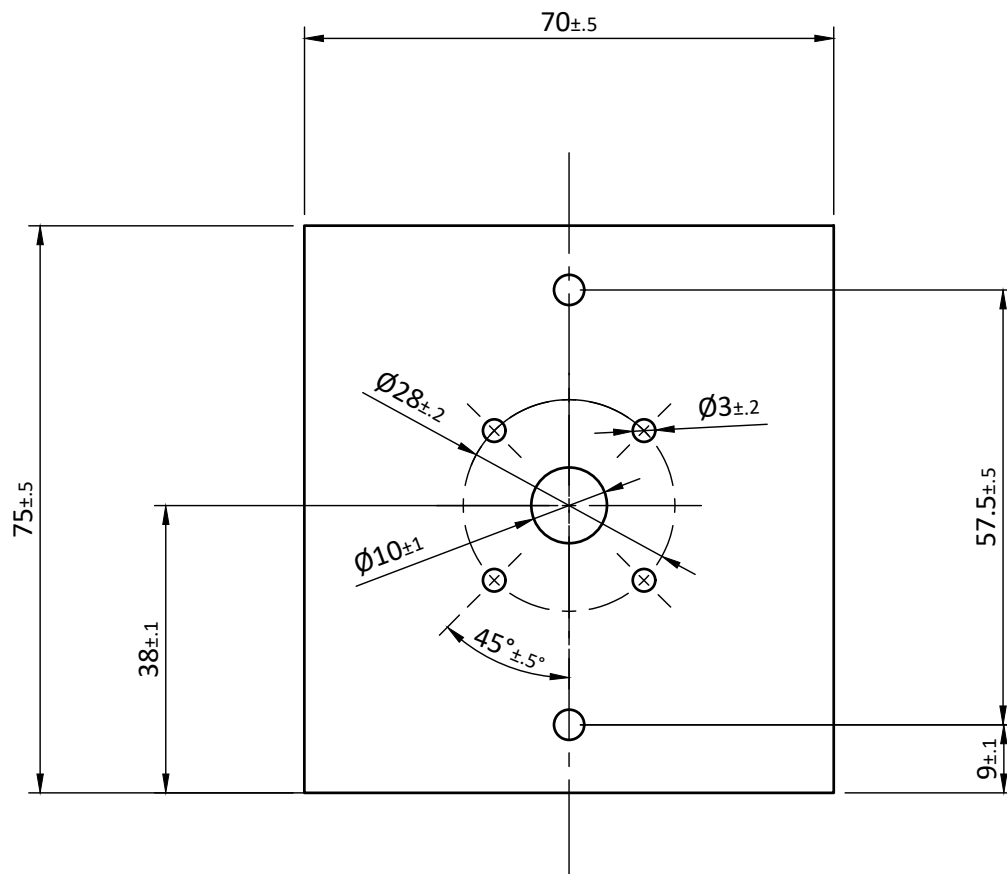


10	1	DT22	ASIN: B078SVZB1X	
9	8	Cargol M2.5x10	DIN 7505-B	Llautó
8	1	TXS108E	ASIN: B07N7FFY2Q	
7	1	INA219	ref: 3-01-1152-UK-2PCS	
6	1	Relé	ASIN: B0B153VGSF	DC 30V 5A
5	1	ESP32-WROOM-32	ASIN: B08BTQZ318	
4	1	Placa de connexions	70981866383-3	400 pins
3	1	Placa d'expansió	ref: GXFM0476-002	38 pins
2	1	Font d'alimentació	ASIN: B01D8FM3W6	24 V 5A
1	1	Conjunt estructura	Plànol 6	
Marca	Nº	Denominació	Referència	Observacions
Autor/a: Júlia Arribas			Data: 30/06/2024	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA  ETSI Aeroespacial y Diseño Industrial
Projecte: Automatització i control inalàmbric d'un sistema de ventilació				
Escala: 1:2	Títol: Conjunt base			Nº: 5





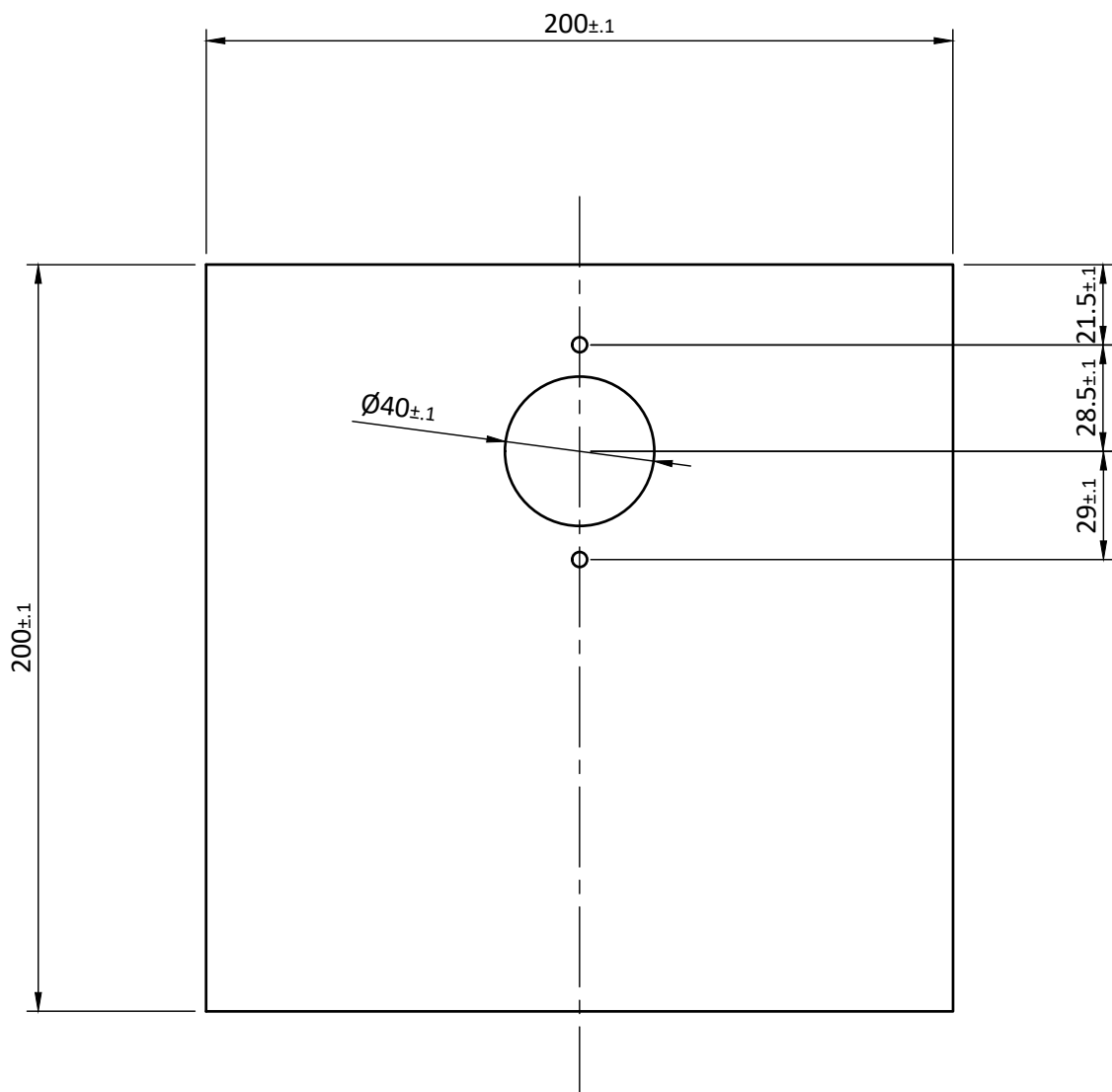
Unitats: mm

4	8	Cargol M3x15	DIN 7997	Acer inoxidable
3	2	Esquadra ampla blanca	ref: 17869033	
2	1	Peça Suport	Plànol 8	
1	1	Peça Base	Plànol 9	
Marca	Nº	Denominació	Referència	Observacions
Autor/a: Júlia Arribas Ríos			Data: 30/06/2024	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA  ETSI Aeroespacial y Diseño Industrial
Projecte: Automatització i control inalàmbric d'un sistema de ventilació				
Escala: 1:3	Títol: Conjunt estructura			Nº: 6





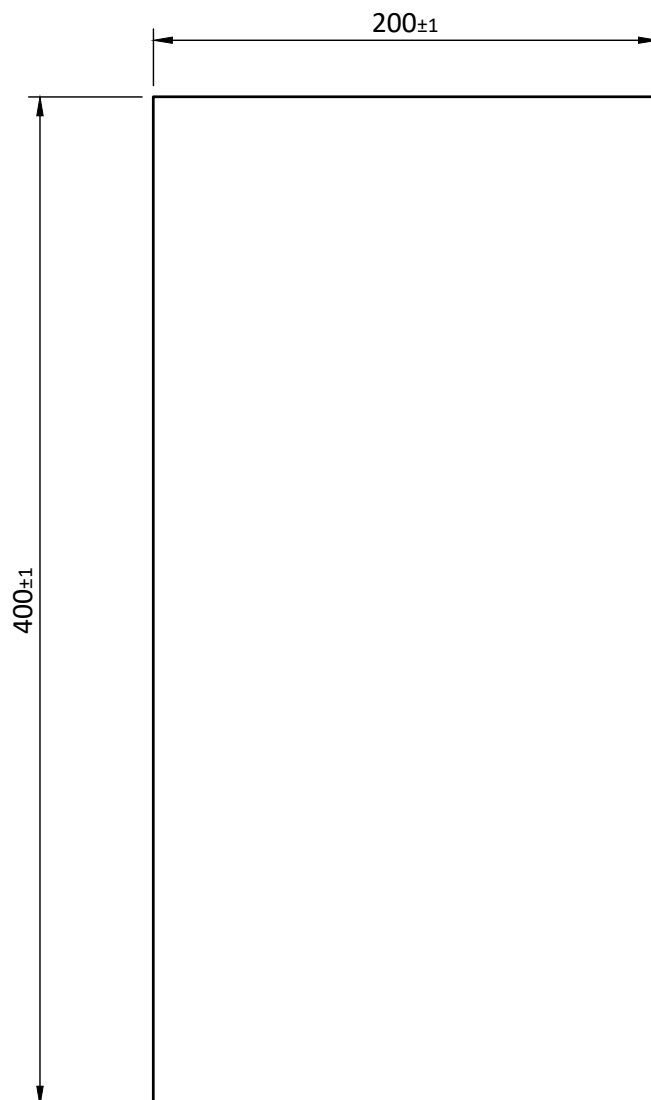
Unitats: mm

Autor/a: Júlia Arribas Ríos		Data: 30/06/2024	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA  ETSI Aeroespacial y Diseño Industrial
Projecte: Automatització i control inalàmbic d'un sistema de ventilació			
Escala: 1:1	Títol: Placa	Nº: 7	





Unitats: mm

Autor/a: Júlia Arribas Ríos		Data: 30/06/2024	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA  ETSI Aeroespacial y Diseño Industrial
Projecte: Automatització i control automàtic d'un sistema de ventilació			
Escala: 1:2	Títol: Peça suport		Nº: 8



Unitats: mm

Autor/a: Júlia Arribas Ríos		Data: 30/06/2024	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA  ETSI Aeroespacial y Diseño Industrial
Projecte: Automatització i control inalàmbric d'un sistema de ventilació			
Escala: 1:3	Títol: Peça base		Nº: 9



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA
AEROESPACIAL I DISSENY INDUSTRIAL

Automatització i control remot d'un sistema de ventilació

Treball Fi de Grau

Grau en Enginyeria Electrònica Industrial i Automàtica

Document N^o 3: Plec de condicions

Autora:
Arribas Ríos, Júlia Mei

Tutor:
Rodríguez Ballester, Francisco

Curs acadèmic 2023/2024



Índex del plec de condicions

1.	Definició i abast del plec	107
2.	Especificacions tècniques.....	107
2.1.	Condicions dels materials	107
2.1.1.	Subsistema del dispositiu de ventilació	107
2.1.2.	Subsistema del panell de control.....	108
2.2.	Condicions de les ferramentes.....	110
3.	Condició de l'execució de la maqueta	111
3.1.	Control de l'execució del dispositiu de ventilació.....	111
3.1.1.	Subsistema electrònic	111
3.1.2.	Subsistema d'alimentació	112
3.1.3.	Subsistema mecànic.....	113
3.2.	Control de l'execució del panell de control	114
4.	Proves prèvies a la posada en funcionament	114

Índex taules del plec de condicions

Taula 1:	Llistat de materials per al subsistema de ventilació.	107
Taula 2:	Llistat de materials del panell de control.	108



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ETSI Aeroespacial y Diseño Industrial



1. Definició i abast del plec

La següent especificació es refereix al muntatge de la maqueta del sistema de ventilació, incloent tant el dispositiu de ventilació com el panell de control. S'especifiquen els materials, les ferramentes, les execucions, proves necessàries per a la posta en funcionament.

Queden exemptes d'aquesta especificació el muntatge del prototip final, així com l'especificació del motor i aspes del prototip final, i la de la fabricació de les peces impreses en 3D.

2. Especificacions tècniques

2.1. Condicions dels materials

2.1.1. Descripció dels materials del subsistema del dispositiu de ventilació

En la Taula 1 es llisten tots els materials necessaris per al subsistema del dispositiu de ventilació.

Taula 1: Llistat de materials per al subsistema de ventilació.

Subsistema	Component	Quantitat	Model	Detalls
Electrònic	Microcontrolador	1	ESP32-WROOM-32	Xip ESP32-D0WDQ6-V3
	Placa d'expansió de 38-Pins	1	ESP32S 38P-V4-Goouu Expansion Board	
	Convertidor de nivell	1	TXS108E	3.3V a 5V, bidireccional
	Cable USB	1	Micro USB a USB-A 2.0	
	Transformador AC/DC	1	CellularLine ACHUSBQUAD4AK	DC5V 4200mA max
	Sensor de temperatura	1	DHT22	
	Sensor de corrent	1	INA219	Corrent $\pm 3.2A$, Resolució 0.1 mA
	Resistència 10 K Ω	1	Series E24	
	Condensador 0.1 μF	2	Series E12	
	Relé	1	Relay ISO9002	DC 30V 5A
	Motor	1	36GP-3650	DC24V 421RPM
	Cables M/M	5	22AWG	20 cm
	Cables M/M	5	22AWG	10 cm
	Cables M/F	11	22AWG	20 cm
Cables F/F	6	22AWG	20 cm	

Subsistema	Component	Quantitat	Model	Detalls
	Placa de connexions	1	400 Pins	70981866383-3
Alimentació	Font d'alimentació	1	24V 5A	ASIN B01D8FM3W6
	Cable alimentació estàndard europeu tipus F	1	3 terminals	
	Terminal connexió aïllat	3	AWG 22-16	Forma de u
Mecànic	Aspes	1	Peça fabricada en 3D	Material PLA
	Unió a l'eix	1	Peça fabricada en 3D	Material PLA
	Cargol M3x10	4	ISO 4762	Acer inoxidable
	Cargol M3x6	3	ISO 4762	Acer inoxidable
	Cargol M3x20	1	ISO 4762	Acer inoxidable
	Rosca d'inserció M3	4	ISO 5821	inserció roscada per a plàstics de llautó
	Cargol M3x15	10	DIN 7997	Acer inoxidable
	Cargol M2.5x10	8	DIN 7505-B	Llautó
	Esquadra	2	Esquadra ampla blanca de 30x30 cm de Leroy Merlin	Ref. 17869033
	Fusta 20x60x1.8 cm	1	Tauler massís d'avet de de Leroy Merlin	Ref. 12678785
Fusta contraxapat 3mm	1	Tauler A5 148x210	Ref. 90544985	

2.1.2. Descripció dels materials del subsistema del panell de control

En la Taula 2, es troba un llistat dels materials que componen el sistema del panell de control.

Taula 2: Llistat de materials del panell de control.

Subsistema	Component	Quantitat	Model	Detalls
Electrònic	Pantalla + controlador	1	ESP32-2432S028	ASIN B0CSYPG716
Alimentació	Cable USB	1	Micro USB a USB-A 2.0	
	Transformador AC/DC	1	CellularLine ACHUSBQUAD4AK	DC5V 4200mA max
Mecànic	Peça superior de la carcassa	1	Peça immersa en 3D	Material PLA



Subsistema	Component	Quantitat	Model	Detalls
	Peça inferior de la carcassa	1	Peça immersa en 3D	Material PLA
	Cargols M3x6	4	ISO 4762	Acer inoxidable

2.1.3. Control de qualitat

Comprovar que els microcontroladors funcionen correctament, pujant un codi d'exemple. També s'ha de comprovar que la comunicació remota entre els dos microcontroladors funciona correctament. D'aquesta manera també es comprova que els cables micro-USB funcionen correctament, tant com a l'hora de transmetre dades, com per a alimentar les plaques. Rebutjar els components que no funcionen correctament.

Comprovar que la placa d'expansió funcione adequadament, col·locar la placa d'expansió ESP32-WROOM-32 en els forats adequadament, assegurant que els pins del micro corresponen amb els xerografiats en la placa d'expansió. Comprovar continuïtat amb el multímetre, de tots els pins entre el micro i la placa, a més, comprovar que el voltatge subministrat a la placa i als pins és adequat al connectar-se una font d'alimentació, tant pel jack, com per la entrada micro-USB, com per la USB tipus C. Rebutjar la placa en cas de que alguna de les comprovacions no resulte positiva.

Comprovar que el convertidor de nivell funcione correctament, connectant el convertidor de nivell adequadament, i subministrant una tensió de 5 V per cada canal i comprovant que pel canal anàleg hi haja una tensió de 3.3 V. El mateix procediment per a comprovar el canvi de nivell de 3.3 V a 5 V. Desestimar el component en cas de que el resultat siga negatiu.

Comprovar que els transformadors AC/DC funcionen com s'indica, engegant-los a la corrent, i comprovant la tensió d'eixida que ofereixen. Rebutjar els components que no satisfacen les condicions establertes.

Comprovar que el sensor de temperatura mesure correctament la temperatura, mitjançant un termòmetre de referència. Connecta adequadament el sensor a un microcontrolador mitjançant els cables F/F, a continuació puja el codi "dht22test.ino", aquest es troba en els exemples de la llibreria <DHT22.h>. Comparar els resultats. Si la discrepància és de més de ± 0.5 °C, desestimar el component.

Comprovar que el sensor de corrent mesure correctament la corrent, mitjançant un circuit i un multímetre. Connecta un circuit amb una font de tensió, de per exemple 5 V, un LED i una resistència, de per exemple 10K Ω , comprova que la resistència tinga el valor que correspon amb el multímetre, rebutjar la resistència si el resultat no està dins del marge de tolerància de la resistència. Connecta el sensor i a continuació el multímetre entre la pila i el LED, interrompent el circuit. Comparar els resultats, i si la discrepància és de més de $\pm 0,004$ mA, rebutjar el component.

De la mateixa manera que s'ha comprovar el valor de la resistència, procedir amb el valor dels condensadors, rebutjar els valors dels condensador que no es torben dins del marge de toleràncies.



Comprovar que el relé funciona correctament. Connectar el component a un circuit, es pot fer servir el del LED, connectar els pins C i NO. El pin DC+ a la part positiva de la font de tensió, i el pin DC- a GND o part negativa de la font d'alimentació, assegurar-se de que el jumper estiga en posició high (H). Per a comprovar que el relé commuta, connectar el pin de dades, IN, amb un voltatge superior de 3.3 V, com Vcc de la font d'alimentació, si el LED s'encén, el relé funciona correctament. Desestimar el component en cas de que resultat de la prova siga negatiu.

Comprovar que la font d'energia de 24 V, done el voltatge indicat, connectant-lo a la corrent mitjançant el cable d'alimentació estàndard europeu tipus F, i mesurant, amb el multímetre el voltatge que subministra per l'eixida, entre -V i +V. Si es diferent de 24 V, es pot variar el voltatge mitjançant un potenciòmetre que es troba en la font d'alimentació, al costat de les eixides i entrades del dispositiu, variant-lo fins a trobar el voltatge desitjat. En cas de que no s'arribe al voltatge desitjat, rebutjar el component.

Comprovar que el motor funcione correctament. Connectant-lo a la font de voltatge de 24 V, i a una font d'alimentació variable, de rang entre 0 i 5 V. Variar el voltatge d'aquesta comprovant que el motor gira com s'espera. Desestimar el motor, en cas de que es comporte de manera no desitjada.

Revisar que els peces impreses en 3D, els cargols, les esquadres i les fustes, no presenten ningun defecte de ruptura o de fabricació, mitjançant una ullada. Desestimar les peces en cas de que presenten algun defecte.

Comprovar que la pantalla funciona correctament. Primerament, pujar un codi que mostre pel display alguna imatge, i comprovar amb la vista que es mostra correctament. A continuació, pujar un codi per a comprovar que la pantalla tàctil respon correctament a la interacció amb aquesta. Rebutjar la pantalla en cas de que alguna de les dues proves no donen un resultat positiu.

2.2. Condicions de les ferramentes

A continuació s'enumeren les ferramentes necessàries per al muntatge de la maqueta.

- Ordinador compatible amb Arduino, els requisits per a instal·lar aquest programa son 1GB d'espai en el disc dur, 256 MB de RAM, processador Pentium 4 o superior i un sistema operatiu amb Windows, MacOS o Linux.
- Multímetre
- Serra
- Trepant
- Broca per a forat de 40 mm de diàmetre
- Broca de 10 mm de diàmetre
- Broca de 4 mm de diàmetre
- Clau Allen CrV 2.5
- Tornavís PH1 i un PH2 o puntes per a trepant de tipus PH1 i PH2.
- Crimpadora
- Pela cables
- Soldador
- Llapis
- Compàs
- Regle



3. Condició de l'execució de la maqueta

3.1. Control de l'execució del dispositiu de ventilació

3.1.1. Subsistema electrònic

Per al muntatge del subsistema electrònic s'han de seguir les següents indicacions:

1. Col·locar la placa de desenvolupament en la placa d'expansió, els pins han d'inserir-se de manera perpendicular al pla horitzontal, i s'ha de pressionar de manera uniforme fins al final. Col·locar el jumper de la placa d'expansió entre GND i 5V.
2. Connectar un cable mascle/femella (M/F) de 20 cm, per a unir la columna positiva esquerra, la més propera a la columna a, amb 5 V de la placa d'expansió. Connectar un cable M/F de 20 cm, per a unir la columna positiva dreta, la més propera a la columna j, amb 3.3 V de la placa d'expansió. Connectar un cable mascle/mascle (M/M) de 10 cm, per a unir les dues columnes negatives de la protoboard. I col·locar un cable M/F de 20 cm, el terminal mascle ha d'anar a una de les dues columnes negatives de la protoboard, i el terminal femella a un pin GND de la placa d'expansió. La fila en la qual es situen els cables no té rellevància, però s'ha d'assegurar que no quede en tensió cap cable, dels prèviament mencionats, ni els futurs que es col·loquen.
3. Col·locar el convertidor de nivell en la protoboard, de manera que els pins de la zona B es situen en la columna d i els pins de la zona a es situen en la columna g, els pins GND i OE han de quedar en la fila 3 i el pin VB i VA en la fila 12. S'han d'inserir de manera perpendicular a la placa i amb una pressió uniforme en tots els pins fins al final.
4. Col·locar un cable M/M de 10 cm unint la fila de GND del convertidor de nivell amb la columna negativa més pròxima. Col·locar un cable M/M de 10 cm, unint VB amb la columna de 5 V, i un altre cable M/M de 10 cm, unint VA amb la columna de 3.3 V.
5. Col·locar el condensador de 0.1uF, el terminal negatiu ha de connectar-se amb la columna negativa (-) de la protoboard, més pròxima del costat dels pins B, i l'altre terminal ha de connectar-se a la fila corresponent al pin VB, del convertidor de nivell, és a dir la fila 12. Connectar l'altre condensador, inserint el terminal positiu en la fila corresponent al pin VA, del convertidor de nivell, en la fila 12, i l'altre terminal en la columna negativa (-), més pròxima als pins de la zona A. Col·locar la resistència de 10 KΩ, un terminal en la fila corresponent al pin OE i l'altre en la fila corresponent a VA.
6. Col·locar un cable M/F de 20 cm unint la fila corresponent al pin A6, del convertidor de nivell, amb el pin P18, de la placa d'expansió. Col·locar el cable blau en la fila corresponent al pin B6 del convertidor de nivell.
7. Col·locar el sensor de corrent en la columna b, de manera que el pin Vin+ quede en la fila 18 i el pin Vcc en la fila 23. S'han d'inserir de manera perpendicular a la placa i amb una pressió uniforme en tots els pins fins al final.
8. Connectar el pin Vin- del sensor de corrent amb el terminal V+ de la font d'alimentació, mitjançant un cable M/M de 20 cm.
9. Per a connectar un cable a de les entrades de la font d'alimentació, primer s'ha d'alçar una tapa que cobreix cargols que assegura la unió, posteriorment hi ha que descargolar el cargol amb el tornavis PH2, lo suficient per a que pugui entrar el cable. Seguidament cal inserir el cable de manera que sols els fils de coure estiguen en contacte amb les plaques que asseguruen la subjecció, en cap cas la part aïllant ha de ser pressionada per aquestes dos plaques, si és així s'ha de descargolar el cargol i corregir la posició del cable. Una vegada el cable estiga en la posició correcta s'ha de cargolar el cargol fins que el cable estiga fixe. Aquest pas s'ha de

seguir sempre que es necessite connectar un cable a un terminal de la font d'alimentació o del relé, en el cas de relé no és necessari alçar cap tapa.

10. Connectar Vin+ del sensor de corrent amb el terminal COM del relé, mitjançant un cable M/M de 20 cm. Per a connectar el cable a l' entrada COM, s' ha de seguir el mateix procediment que amb la font d'alimentació, explicat en el pas número 9.
11. Connectar el pin Sda del sensor de corrent amb el pin P21 de la placa d'expansió, i el pin Scl del sensor de corrent amb el pin 22 de la placa d'expansió, ambdós amb cables M/F de 20 cm, el terminal masculí s'ha d'inserir en la protoboard, i el terminal femení en els pins de la placa d'expansió. Connectar el pin GND del sensor de corrent amb GND de la placa d'expansió i el pin Vcc amb la columna connectada a 3.3 V, ambdós amb cables M/M de 10 cm.
12. Connectar el cable roig del motor a l' entrada NO del relé. Per a connectar-lo s' ha de seguir el mateix procediment que amb la font d'alimentació, explicat en el pas número 8.
13. Connectar DC+, del relé a un pin de 5V de la placa d'expansió, DC- del relé a un pin de GND de la placa d'expansió i IN del relé al pin P13 de la placa d'expansió, tots mitjançant cables M/F de 20 cm.
14. Connectar el sensor de temperatura mitjançant tres cables femella/femella (F/F) de 20 cm a uns cables M/M de 20 cm i aquests connectar-los a uns altres F/F de 20 cm, els terminals que queden lliure s'han de connectar a la placa d'expansió del següent mode, el pin + del sensor s'ha de connectar amb 5 V de la placa d'expansió, el pin – s'ha de connectar amb GND, i el pin OUT amb el pin P14 de la placa d'expansió.
15. Connectar el cable blanc i el negre del motor a GND de la protoboard.
16. Connectar el cable groc del motor a la protoboard, en la fila 27 i en la columna i. Connectar en la fila 27 en la columna g, el terminal mascle d'un cable M/F de 20 cm i connectar el pin femella al pin P15 de la placa d'expansió.
17. Connectar un cable M/M de 20 cm a una de les dues columnes de GND de la protoboard, i l'altre connectar-lo a V- de la font d'alimentació.

A continuació es van a relatar les pautes que s'han de seguir per a pujar el codi al microcontrolador:

1. Connectar el ESP-32 a l'ordinador amb el cable micro USB.
2. Pujar el codi controlador.ino amb el IDE d'Arduino
3. Desconnectar el ESP-32 de l'ordinador

3.1.2. Subsistema d'alimentació

A continuació es va a detallar el montatge necessari per alimentar la font d'alimentació:

1. Per a unir un terminal elèctric aïllat amb un dels tres cables del cable d'alimentació estàndard, primer s'ha d'assegurar que el cable no estiga connectat a la xarxa elèctrica. A continuació, s'ha de pelar el cable, mitjançant el pela cables, seguidament s'introdueix el cable en el l'orifici del terminal elèctric, assegurant que la part de coure no se'n ix per cap extrem del terminal, seguidament es crimpa amb la crimpadora en la zona roja que està sobre la part metàl·lica amb força, assegurant-se que el cable no balla. Repetir aquest procés en els dos cables restants.
2. Unir el cable marró amb el terminal L de la font d'alimentació, el cable blau amb el terminal N, i el cable verd i groc amb el terminal pressa a terra.



3.1.3. Subsistema mecànic

A continuació es va a explicar com montar l'estructura:

1. Tallar el tauler massís d'abet de 20x60x1.8 cm de manera que es dividisquen en dues peces, la primera de 20x20x1.8 cm, aquesta serà nomenada d'ací endavant com el suport (plànol 8), i la segona de 20x40x1.8 cm, aquesta serà nomenada d'ací endavant com la base (plànol 9).
2. Foradar el suport amb la broca de 40 mm, com es mostra en el plànol 8.
3. Col·locar la base amb la cara amb més superfície en horitzontal. Col·locar el suport formant una T invertida i alineada amb l'amplària de 20 cm de la base, com es mostra en el plànol 6.
4. Col·locar les esquadres pegades les vores exteriors en la part frontal, cada cara exterior de l'esquadra ha d'estar en contacte amb una de les dues peces, com mostra el plànol 6.
5. Caragolar els cargols DIN 7997 M3x15 travessant els orificis de les escudares a la fusta, assegurant que les vores de l'esquadra, la base i el suport estiguen alineades, mitjançant un tornavís PH2 o un trepant amb una punta PH2.
6. Tallar la fusta contraxapada en forma de rectangle de 70x75 mm. A continuació s'ha de marcar el centre i fer una circumferència de 14 mm de radi, posteriorment cal realitzar línies a 45 graus de l'horitzontal, es pot fer mitjançant la bisectriu entre la vertical i l'horitzontal. En la intersecció entre les línies diagonals i la circumferència s'han de realitzar 4 forats de 4 mm de diàmetre, amb el trepant i la broca de 4 mm. Realitzar un forat de 10 mm de diàmetre com es mostra en el plànol 7. Addicionalment s'han de realitzar dos forats de 3 mm de diàmetre al llarg de la línia vertical, en els extrems, a 1 cm aproximadament de la vora, aquests es poden fer directament amb el cargol DIN 7997 M3x15 a l'hora de unir la fusta contraxapada amb el suport.
7. Fer coincidir els forats realitzats en el pas previ amb els orificis del motor i inserir els cargols ISO 4762 d'acer inoxidable de M3x10 amb el tornavís PH1.
8. Fer passar el motor per l'orifici del suport fins que la fusta contraxapada topete amb el suport, fent que el motor quede centrat en l'orifici. A continuació, fixar els dos conjunts amb els cargols DIN 7997 M3x15, fent pressió per a què els cargols puguin inserir-se en la fusta, en aquest pas és recomanable utilitzar el trepant, amb la punta PH2.

A continuació, es va a indicar com col·locar els components electrònics en l'estructura, per a formar el conjunt base:

1. Col·locar els components com es mostra en el plànol 5, i fixar-los amb cargols DIN 7505-B M2.5x10 mitjançant un PH1, excepte la protoboard que porta un adhesiu, hi ha que desapegar l'adhesiu i fixar-lo a la fusta. La posició exacta dels components no és rellevant, però han de seguir l'estructura i l'orientació que es mostra en el plànol, a excepció de la font d'alimentació.
2. La part foradada de la font d'alimentació ha de mirar cara l'exterior.

A continuació es van a nomenar les directrius per a montar l'hèlice:

1. Col·locar la peça unió amb l'eix de manera que l'orifici en una de les cares, del prisma hexagonal, mire cara amunt. A continuació inserir una rosca d'inserció, amb el soldador calent, de manera vertical, fins que la rosca quede a la mateixa altura que la superfície de la peça.



2. Col·locar la unió amb l'eix verticalment de manera que els tres orificis miren cara dalt. Inserir una rosca d'inserció en el forat del mig, de manera vertical, amb el soldador calent.
3. Col·locar l'hèlice de manera que la cara que queda mirant cara amunt mostre els tres forats. Aquesta ha de quedar totalment horitzontal. A continuació, inserir les rosques d'inserció amb el soldador calent de manera vertical, fins que la rosca quede al ras de la superfície de l'hèlice, en els dos forats exteriors.
4. Col·locar la unió amb l'eix sobre les aspes fent coincidir els tres forats. Roscar els cargols ISO 4762 d'acer inoxidable de M3x6, en els dos forats que queden en l'anell exterior, passant primer per la peça d'unió i enroscant-los amb la rosca d'inserció de les aspes.
5. Donar la volta al conjunt. Roscar el cargol ISO 4762 d'acer inoxidable de M3x20 en l'orifici central, amb el tornavís PH1.

A continuació es van a descriure les indicacions per a unir el subconjunt de les aspes amb el subconjunt del motor:

1. Introduir el orifici en forma de D de la unió del eix, amb l'eix.
2. Inserir el cargol ISO 4762 d'acer inoxidable de M3x6 en la rosca d'inserció en el lateral amb la clau Allen CrV 2.5, fins que faça tope amb l'eix, assegurant-se de que queda fixe.

3.2. Control de l'execució del panell de control

Per al muntatge del panell de control s'han de seguir els següents passos:

1. Col·locar la peça superior cara avall, de manera que l'interior de la peça mire cara amunt.
2. Col·locar sobre aquesta la pantalla, assegurant que els connectors coincideixen amb els de la carcassa.
3. Col·locar la peça inferior sobre aquestes, de manera que l'interior mire cara avall, fent coincidir els orificis de les tres peces.
4. Inserir els cargols ISO 4762 d'acer inoxidable de M3x6 en els quatre orificis amb la clau Allen CrV 2 fins que facen tope.

A continuació es van a relatar les pautes que s'han de seguir per a pujar el codi al microcontrolador:

1. Connectar el ESP-32 a l'ordinador amb el cable micro USB
2. Pujar el programa menu.ino amb el IDE d'Arduino
3. Desconnectar el ESP-32 de l'ordinador

4. Proves prèvies a la posada en funcionament

S'ha de comprovar que l'hèlix està ben subjectada mitjançant l'intent d'extracció d'aquesta amb força moderada, si es percep que es mou, cal assegurar-la roscant més el cargol que subjecta l'hèlix l'eix del motor. En cas de què romanga ballant es rebutjarà el muntatge.



S'ha de comprovar que els microcontroladors es comuniquen entre sí. Connectar el controlador a l'ordinador, i veure els missatges que s'imprimeixen per pantalla comuniquen que l'enviament de dades ha sigut satisfactori. En cas de què no es comuniquen es rebutjarà el muntatge.

S'ha de comprovar que la pantalla tàctil del panell de control funciona, mitjançant l'ús d'aquesta. Si al tocar-la en els llocs pertinents no reacciona com s'espera, es rebutjarà el muntatge.

S'ha de comprovar que el sensor de temperatura funciona correctament, mitjançant el mesurament de la temperatura amb aquest, i comprovant els resultats amb dades de referència. En cas de ser discrepants en un ± 2 °C, es rebutjarà el muntatge.

S'ha de comprovar que el muntatge electrònic siga correcte, que no hi haja cables solts, ni ningun curtcircuit.

València, juliol de 2024

Grau en Enginyeria Electrònica Industrial i Automàtica

Júlia Arribas Ríos



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ETSI Aeroespacial y Diseño Industrial



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA
AEROESPACIAL I DISSENY INDUSTRIAL

Automatització i control remot d'un sistema de ventilació

Treball Fi de Grau

Grau en Enginyeria Electrònica Industrial i Automàtica

Document N^o 4: Pressupost

Autora:
Arribas Ríos, Júlia Mei

Tutor:
Rodríguez Ballester, Francisco

Curs acadèmic 2023/2024



Índex del pressupost

1. Quadre de preus elementals.....	120
2. Quadre de preus descompostos	122
3. Resum del pressupost.....	130

Índex de les taules del pressupost

Taula 1: Quadre de preus elementals.....	120
Taula 2: Quadre de preus descompostos del subsistema electrònic.	123
Taula 3: Quadre de preus descompostos del conjunt estructura.....	124
Taula 4: Quadre de preus descompostos del conjunt base.....	125
Taula 5: Quadre de preus descompostos del conjunt motor.	126
Taula 6: Quadre de preus descompostos del conjunt hèlice.....	126
Taula 7: Quadre de preus descompostos del sistema dispositiu de ventilació.	127
Taula 8: Quadre de preus descompostos del panell del sistema panell de control.	128
Taula 9: Quadre de preus descompostos del conjunt del disseny del projecte.	128
Taula 10: Valoració del projecte.	129
Taula 11: Resum del pressupost del projecte.....	130

Índex de les equacions del pressupost

Equació 1: càlcul del cost per hora de l'ordinador	121
Equació 2: càlcul del consum elèctric de l'ordinador per hora.....	121
Equació 3: càlcul de l'amortització de l'ordinador per hora.....	122



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ETSI Aeroespacial y Diseño Industrial

1. Quadre de preus elementals

En la Taula 1, es llisten tots els elements que han intervingut per a l'execució del projecte Automatització i control remot d'un sistema de ventilació. Es llisten els materials, la mà d'obra, l'amortització dels equips utilitzats, i les llicències de software, per preu unitari, a més es determina un percentatge del 10% que s'aplicarà sobre els costos directes, que representa els medis auxiliars, aquests fan referència a costos difícils de quantificar, com la utilització de les ferramentes de preu poc significatiu, pèrdua o malbaratament de material.

Taula 1: *Quadre de preus elementals.*

1. Quadre de preus elementals			
Referència	Unitat	Descripció	Preu (€)
<u>Materials</u>			
m1	u.	ESP32-WROOM-32	8,66
m2	u.	ESP32 38P-V4- Gooou Expansion Board	6,99
m3	u.	Convertidor de nivell TXS108E	2,33
m4	u.	Cable micro USB a USB-A	5,60
m5	u.	Transformador AC/DC	9,89
m6	u.	Sensor de temperatura DHT22	5,49
m7	u.	Sensor de corrent INA219	6,33
m8	u.	Resistència 10KΩ	0,10
m9	u.	Condensador 0.1 uF	0,10
m10	u.	Relé Relay ISO9002	2,60
m11	u.	Cable M/M 20 cm	0,10
m12	u.	Cable M/M 10 cm	0,09
m13	u.	Cable M/F 20 cm	0,10
m14	u.	Cable F/F 20 cm	0,10
m15	u.	Motor 36GP-3650	35,00
m16	u.	Placa de connexions 400 Pins	2,33
m17	u.	Font d'alimentació 24 V 5A	32,99
m18	u.	Cable d'alimentació estàndard europeu tipus F	7,39
m19	u.	Terminal de connexió aïllat AWG 22-16 en forma de u	0,30
m20	u.	Peça hèlix	1,57
m21	u.	Peça unió a l'eix	0,25
m22	u.	Cargol M3x10	0,05
m23	u.	Cargol M3x6	0,05
m24	u.	Cargol M3x20	0,05
m25	u.	Rosca d'inserció M3	0,05
m26	u.	Cargol M3x15	0,05



Referència	Unitat	Descripció	Preu (€)
m27	u.	Esquadra	1,39
m28	u.	Fusta 20x60x1.8 cm	7,99
m29	u.	Fusta contraxapat 3 mm	2,60
m30	u.	ESP32-2432S028	24,99
m31	u.	Peça superior de la carcassa	0,21
m32	u.	Peça inferior de la carcassa	0,21
m33	u.	Cargol M2.5x10	0,05
Mà d'obra			
h1	h.	Enginyer desenvolupador	20
Equip			
e1	h.	Ordinador	0,03
Software			
s1	Llicència mensual	Arduino	0
s2	Llicència mensual	Autodesk Fusion 360	0
s3	Llicència mensual	Microsoft Office 365	0
s4	Llicència mensual	Fritzing	0
Mitjans auxiliars			
	%	Medis auxiliars sobre costos directes	10

Per al càlcul del cost de l'equip e1, en la Taula 1, s'ha tingut en compte l'amortització de l'ordinador, el preu de compra va ser de uns 1320,9 €, i la vida mitjana d'un ordinador es de 6 anys. El preu per hora de l'ordinador és de 0,025 €/h, com mostra la Equació 1. A més, l'ordinador consumeix energia, aquesta es cobra a un preu de 0,152 €/kWh i l'ordinador gasta una mitjana de 0,33 kWh, de manera que el cost de l'energia consumida per l'ordinador és de 0,05 €/h, com mostra la Equació 2. Per lo tant l'amortització de l'ordinador per hora és de 0,03 €/h, com mostra la Equació 3.

Equació 1: càlcul del cost per hora de l'ordinador

$$\frac{\text{cost ordinador}}{h} = \frac{\text{cost total}(\text{€})}{\text{vida útil}(\text{anys})} \cdot \frac{1 \text{ any}}{365 \text{ dies}} \cdot \frac{1 \text{ dia}}{24 \text{ h}} = \frac{1320,9 \text{ €}}{6 \text{ anys}} \cdot \frac{1 \text{ any}}{365 \text{ dies}} \cdot \frac{1 \text{ dia}}{24 \text{ h}} = 0,025 \frac{\text{€}}{\text{h}}$$

Equació 2: càlcul del consum elèctric de l'ordinador per hora

$$\begin{aligned} \frac{\text{cost consum}}{h} &= \text{consum elèctric de l'ordinador (kW)} \cdot \text{preu electricitat} \left(\frac{\text{€}}{\text{kWh}} \right) = \\ &= 0,33 \text{ kW} \cdot 0,152 \frac{\text{€}}{\text{kWh}} = 0,05 \frac{\text{€}}{\text{h}} \end{aligned}$$



Equació 3: càlcul de l'amortització de l'ordinador per hora

$$\frac{\text{amortització de l'ordinador}}{h} = \frac{\text{cost ordinador}}{h} + \frac{\text{cost consum}}{h} = 0,025 \frac{\text{€}}{h} + 0,05 \frac{\text{€}}{h} = 0,03 \frac{\text{€}}{h}$$

Per un altre costat, el cost de les llicències mensuals per a desenvolupar el projecte es de 0 euros en tots els elements, ja que Arduino es de llicència gratuïta, la llicència mensual per als estudiants és gratuïta en Autodesk Fusion 360 i en Microsoft Office 365, i el software de Fritzing és gratis.

2. Quadre de preus descompostos

Per al càlcul dels costos del projecte s'utilitza el mètode de preus descompostos, on es divideix l'execució del material en diferents sistemes, subsistemes, conjunts o subconjunts, els quals són el subsistema electrònic, que apareix desglossat en la Taula 2, el conjunt de l'estructura (Taula 3), el conjunt base (Taula 4), el conjunt motor (Taula 5), el conjunt hèlice (Taula 6), el sistema del dispositiu de ventilació (Taula 7), el sistema del panell de control (Taula 8) i el conjunt del disseny del sistema (Taula 9). Posteriorment es realitza una valorització del l'execució del projecte (Taula 10).

Taula 2: Quadre de preus descompostos del subsistema electrònic.

2. Quadre de preus descompostos I					
Referència	Unitat	Descripció	Preu (€)	Quantitat	Total (€)
d1	u.	Muntatge del subsistema electrònic del dispositiu de ventilació, incloent la font d'alimentació, els cables d'alimentació, el transformador, el controlador, el sensor de temperatura, el sensor de corrent, el convertidor de nivell, tots els cables necessaris per a realitzar el muntatge i la protoboard. No s'inclou els cargols que fixen els components a la estructura ni el muntatge del motor.			
m1	u.	ESP32-WROOM-32	8,66	1	8,63
m2	u.	ESP32 38P-V4 Gooou Expansion Board	6,99	1	6,99
m3	u.	Convertidor de nivell TXS108E	2,33	1	2,33
m4	u.	Cable micro USB a USB-A	5,60	1	5,6
m5	u.	Transformador AC/DC	9,89	1	9,89
m6	u.	Sensor de temperatura DHT22	5,49	1	5,49
m7	u.	Sensor de corrent INA219	6,33	1	6,33
m8	u.	Resistència 10KΩ	0,10	1	0,1
m9	u.	Condensador 0.1 uF	0,10	2	0,2
m10	u.	Relé Relay ISO9002	2,60	1	2,59
m11	u.	Cable M/M 20 cm	0,10	5	0,49
m12	u.	Cable M/M 10 cm	0,09	5	0,46
m13	u.	Cable M/F 20 cm	0,10	11	1,08
m14	u.	Cable F/F 20 cm	0,10	6	0,59
m16	u.	Placa de connexions 400 Pins	2,33	1	2,33
m17	u.	Font d'alimentació 24 V 5A	32,99	1	32,99
m18	u.	Cable d'alimentació estàndard europeu tipus F	7,39	1	7,39
m19	u.	Terminal de connexió aïllat AWG 22-16 en forma de u	0,30	3	0,9
h1	h.	Enginyer desenvolupador	20,00	4	80,00
	%	Medis auxiliars sobre costos directes	10	174,41	17,44
Total preu d'execució material					191,85

Taula 3: Quadre de preus descompostos del conjunt estructura.

2. Quadre de preus descompostos II					
Referència	Unitat	Descripció	Preu (€)	Quantitat	Total (€)
d2	u.	Muntatge del conjunt estructura, incloent treball de serrar el tauler de fusta de 20x60x1.8 cm, el de foradar la peça superior amb un diàmetre de 40 mm i el de cargolar els cargols de les esquadres a la estructura.			
m26	u.	Cargol M3x15	0,05	8	0,4
m27	u.	Esquadra	1,39	2	2,78
m28	u.	Fusta 20x60x1.8 cm	7,99	1	7,99
h1	h.	Enginyer desenvolupador	20,00	2	40,00
	%	Medis auxiliars sobre costos directes	10,00	51,17	5,12
Total preu d'execució material					56,29



Taula 4: Quadre de preus descompostos del conjunt base.

2. Quadre de preus descompostos III					
Referència	Unitat	Descripció	Preu (€)	Quantitat	Total (€)
d3	u.	Muntatge del conjunt base, aquest engloba la estructura i el subsistema electrònic, i la fixació d'aquest dos amb cargols.			
d1	u.	Muntatge del subsistema electrònic del dispositiu de ventilació, incloent la font d'alimentació, els cables d'alimentació, el transformador, el controlador, el sensor de temperatura, el sensor de corrent, el convertidor de nivell, tots els cables necessaris per a realitzar el muntatge i la protoboard. No s'inclou els cargols que fixen els components a la estructura ni el muntatge del motor.	191,85	1	191,85
d2	u.	Muntatge del conjunt estructura, incloent treball de serrar el tauler de fusta de 20x60x1.8 cm, el de foradar la peça superior amb un diàmetre de 40 mm i el de cargolar els cargols de les esquadres a la estructura.	56,29	1	56,29
m33	u.	Cargol M2.5x10	0,05	9	0,45
h1	h.	Enginyer desenvolupador	20,00	0,50	10,00
	%	Mitjans auxiliars sobre costos directes excepte d1 i d2	10,00	10,45	1,05
Total preu d'execució material					259,64

Taula 5: Quadre de preus descompostos del conjunt motor.

2. Quadre de preus descompostos IV					
Referència	Unitat	Descripció	Preu (€)	Quantitat	Total (€)
d4	u.	Muntatge del conjunt motor, aquest inclou el treball de serrar la fusta contraxapada, foradar-la i fixar la peça al motor.			
m15	u.	Motor 36GP-3650	35,00	1,00	35,00
m22	u.	Cargol M3x10	0,05	4,00	0,20
m29	u.	Fusta contraxapat 3 mm	2,60	1,00	2,60
h1	h.	Enginyer desenvolupador	20,00	1,00	20,00
	%	Medis auxiliars sobre costos directes	10,00	57,80	5,78
Total preu d'execució material					63,58

Taula 6: Quadre de preus descompostos del conjunt hèlice.

2. Quadre de preus descompostos V					
Referència	Unitat	Descripció	Preu (€)	Quantitat	Total (€)
d5	u.	Muntatge del conjunt hèlice, aquest inclou la inserció de les rosques d'inserció i la fixació de les peces mitjançant els cargols.			
m20	u.	Peça hèlix	1,57	1	1,57
m21	u.	Peça unió a l'eix	0,25	1	0,25
m23	u.	Cargol M3x6	0,05	3	0,15
m24	u.	Cargol M3x20	0,05	1	0,05
m25	u.	Rosca d'inserció M3	0,05	4	0,18
h1	h.	Enginyer desenvolupador	20,00	0,5	10,00
	%	Medis auxiliars sobre costos directes	10	12,20	1,22
Total preu d'execució material					13,42



Taula 7: Quadre de preus descompostos del sistema dispositiu de ventilació.

2. Quadre de preus descompostos VI					
Referència	Unitat	Descripció	Preu (€)	Quantitat	Total (€)
d6	u.	Muntatge del sistema dispositiu de ventilació. Aquest inclou la unió del conjunt base, el conjunt motor i el conjunt hèlice.			
d3	u.	Muntatge del conjunt base, aquest engloba la estructura i el subsistema electrònic, i la fixació d'aquest dos amb cargols.	259,64	1	259,64
d4	u.	Muntatge del conjunt motor, aquest inclou el treball de serrar la fusta contraxapada, foradar-la i fixar la peça al motor.	63,58	1	63,58
d5	u.	Muntatge del conjunt hèlice, aquest inclou la inserció de les rosques d'inserció i la fixació de les peces mitjançant els cargols.	13,42	1	13,42
m26	u.	Cargol M3x15	0,05	2	0,10
h1	h.	Enginyer desenvolupador	20,00	0,5	10,00
	%	Medis auxiliars sobre costos directes excepte d3, d4 i d5	10	10,10	1,01
Total preu d'execució material					347,75

Taula 8: Quadre de preus descompostos del panell del sistema panell de control.

2. Quadre de preus descompostos VII					
Referència	Unitat	Descripció	Preu (€)	Quantitat	Total (€)
d7	u.	Muntatge del sistema panell de control. Aquest inclou l'assemblatge de la pantalla + el controlador, amb les carcasses, mitjançant cargols.			
m23	u.	Cargol M3x6	0,05	4	0,2
m30	u.	ESP32-2432S028	24,99	1	24,99
m31	u.	Peça superior de la carcassa	0,21	1	0,21
m32	u.	Peça inferior de la carcassa	0,21	1	0,21
h1	h.	Enginyer desenvolupador	20,00	0,5	10,00
	%	Medis auxiliars sobre costos directes	10	35,61	3,56
Total preu d'execució material					39,17

Taula 9: Quadre de preus descompostos del conjunt del disseny del projecte.

2. Quadre de preus descompostos VIII					
Referència	Unitat	Descripció	Preu (€)	Quantitat	Total (€)
d8	u.	Disseny del projecte. Aquest inclou la cerca d'informació, la programació del controlador i del panell de control, així com la realització de la memòria, els plànols, el plec de condicions i el pressupost.			
h1	h.	Enginyer desenvolupador	20,00	291,00	5820,0
e1	h.	Ordinador	0,03	291,00	8,73
s1	Llicència mensual	Arduino	0	1	0
s2	Llicència mensual	Fusion 360	0	1	0
s3	Llicència mensual	Microsoft Office 365	0	1	0
s4	Llicència mensual	Fritzing	0	1	0
	%	Medis auxiliars sobre costos directes	10	5820,00	582,00
Total preu d'execució material					6410,7



Taula 10: Valoració del projecte.

3. Valoració					
Referència	Unitat	Descripció	Preu (€)	Quantitat	Total (€)
d6	u.	Muntatge del sistema dispositiu de ventilació. Aquest inclou la unió del conjunt base, el conjunt motor i el conjunt hèlice.	347,75	1	347,75
d7	u.	Muntatge del sistema panell de control. Aquest inclou l'assemblatge de la pantalla + el controlador, amb les carcasses, mitjançant cargols.	39,17	1	39,17
d8	u.	Disseny del projecte. Aquest inclou la cerca d'informació, la programació del controlador i del panell de control, així com la realització de la memòria, els plànols, el plec de condicions i el pressupost.	6410,73	1	6410,73
Total pressupost del desenvolupament i de l'execució material abans d'impostos					6797,65
	%	IVA	21	6797,65	1427,51
Total pressupost del desenvolupament i de l'execució material després d'impostos					8225,15



3. Resum del pressupost

Projecte: Automatització i control remot d'un sistema de ventilació

En la Taula 11, es troba el resum del pressupost general del projecte, sense IVA s'eleva a la quantitat de 6797,65 € i amb IVA s'eleva a la quantitat de 8225,15€. El benefici industrial és del 0%, ja que aquest projecte no està pensat per a ser comercialitzat, sinó que ha sigut pensat baix una llicència de codi obert, ja que ofereix una manera a tothom de poder actualitzar i automatitzar un sistema de ventilació.

Taula 11: Resum del pressupost del projecte.

Concepte	Import (€)
Pressupost de l'execució del material	6797,65
Beneficis (0%)	0
Suma	6797,65
IVA (21%)	1427,51
Total pressupost general	8225,15

El pressupost general del projecte s'eleva a la quantitat expressada de VUIT MIL DOS-CENTS VINT-I-CINC EUROS AMB QUINZE CÈNTIMS.

València, juliol de 2024

Grau en Enginyeria Electrònica Industrial i Automàtica

Júlia Arribas Ríos