



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

Grau en Enginyeria Electrònica Industrial i Automàtica

**DISSENY I MUNTATGE D' UN SISTEMA D' ADQUISICIÓ DE
DADES PER UN QUADRICÒPTER**



Memòria Tècnica

Autor: Nil Mira Tura
Director: Manuel Andrés Manzanares Brotons
Convocatòria: Juny 2020

Resum

Mitjançant la realització d' aquest Treball Final de Grau, es busca l' aplicació de diferents coneixements adquirits durant la carrera, centrant-se en protocols de comunicació, disseny electrònic i programació d' un microcontrolador.

En el present treball s'ha dissenyat un sistema d' adquisició de dades per a un quadricòpter. Aquest sistema es basa en un microcontrolador PIC, al qual s' hi connecten diversos sensors amb diferents mètodes d' adquisició de dades i comunicació. Aquestes dades adquirides es tracten al propi microcontrolador, es guarden en una memòria externa i s' envien mitjançant un sistema de radiofreqüència.

Per a fer aquest sistema s' ha hagut de dissenyar un hardware tenint en compte totes les especificacions dels diferents dispositius, així com una alimentació adaptada al dron, sense que augmenti molt el pes perquè no afecti a l' estabilitat del vol. Amb tot això fet, també s' ha hagut de dissenyar una placa PCB per a fer les connexions d' una forma més polida i professional.

Finalment, es disposa també d' una estació receptora de dades, la qual mostra en temps real les variables que es van adquirint mitjançant els sensors en diverses pantalles LCD.

Resumen

Mediante la realización de este Trabajo Final de Grado, se busca la aplicación de distintos conocimientos adquiridos durante la carrera, centrándose en protocolos de comunicación, diseño electrónico y programación de un microcontrolador.

En el presente Trabajo se ha diseñado un sistema de adquisición de datos para un cuadricóptero. Dicho sistema se basa en un microcontrolador PIC, al cual se le conectan diversos sensores con diferentes métodos de adquisición de datos y comunicaciones estos datos adquiridos se tratan en el propio microcontrolador, se guardan en una memoria externa y se envían mediante un sistema de radiofrecuencia.

Para hacer este sistema se ha diseñado un hardware teniendo en cuenta todas las especificaciones de los diferentes dispositivos que lo conforman, así como la alimentación adaptada al dron, sin aumentar mucho el peso para que no afecte a la estabilidad de vuelo. Con todo esto hecho, también se ha diseñado una placa PCB para hacer las conexiones de una forma más limpia y profesional.

Finalmente, se dispone también de una estación receptora de datos, la cual muestra en tiempo real las variables que se van adquiriendo mediante los sensores en diversas pantallas LCD.

Abstract

By means of the realization of this Final Degree's Project, it has been looked for the application of the different knowledge that had been acquired during the career, focusing on the different communication protocols, the electronic design and programming a microcontroller.

On the present project it has been designed a data acquisition system applied on a quadcopter. This system is based on a PIC microcontroller, where it is connected to different sensors with varied methods of data acquisition and communication. This acquired data are treated on the microcontroller, they are saved on an external memory and sent with a radiofrequency system based on.

To create this system, a hardware has been designed taking into account all the specifications about the device, as well as the alimentation adapted to the drone, without being too much heavy and avoiding to affect the flight stability. Once all this was done, a PCB has been designed in order the make the different connections more professional and polite.

Finally, there is also a station that is capable of receiving the data, which prints in different LCD screens the real time clock as well as the different variables that the microcontroller is acquiring thanks to the different sensors attached to it.

Agraïments

En primer lloc m' agradaria donar les gràcies al que ha estat el meu tutor del Treball Final de Grau, Manuel Manzanares, per la seva dedicació i voluntat d' ajudar i donar suport sempre que ha estat necessari, juntament amb l' ensenyança que m' ha transmès al llarg de no només el treball, sinó de la carrera universitària.

Per altra banda agraeixo a tots els meus companys de grau i amics per implicar-se i interessar-se en aquest projecte, sempre intentant donar un cop de mà i comentant petites idees que molts cops han servit de suport i han ajudat a obrir la ment en moments complicats i costava avançar.

Per últim i no menys important, agrair a la meva família i parella tota la paciència que han tingut i els ànims que m' han donat, interessant-se en tot moment per l' avanç del projecte i mostrant il·lusió que animava a seguir treballant.

Glossari

- **Arduino:** Companyia de desenvolupament de *hardware* i *firmware* lliures.
- **PCB:** *Printed Circuit Board*. Placa de circuit imprès.
- **LCD:** *Liquid Crystal Display*. Pantalla de cristall líquid.
- **PIC:** *Programmable Integrated Circuit*. Família de microcontroladors de l' empresa Microchip Technology Inc.
- **Datasheet:** Document amb les característiques i especificacions del component al qual va relacionat.
- **LiPo:** Bateria de polímers de Liti.
- **ADC:** Convertidor d' analògic a digital.
- **I2C:** *Inter-Integrated Circuit*. És un bus de comunicacions sèrie de dades.
- **SCL:** *Serial Clock*. Forma part del I2C, sent la línia del rellotge.
- **SDA:** *Serial Data*. Forma part del I2C, sent la línia de dades.
- **EEPROM:** *Electrically Erasable Programmed Read-Only Memory*. Memòria ROM la qual pot ser programada, esborrada i re-programada electrònicament.
- **UART:** *Universal Asynchronous Receiver-Transmitter*. *Hardware* que permet la comunicació sèrie asíncrona.
- **Rx i Tx:** *Receive* i *Transmit* respectivament.

Índex

RESUM	I
RESUMEN	II
ABSTRACT	III
AGRAÏMENTS	V
GLOSSARI	VII
1. PREFACI	1
1.1. Objecte del projecte	1
1.2. Motivació i justificació	2
1.3. Requeriments previs	2
2. INTRODUCCIÓ	5
2.1. Objectius del treball	5
2.2. Abast del treball	6
3. ESTAT DE L' ART	7
4. POSSIBLES SOLUCIONS	9
4.1. Alimentació	9
4.1.1. Piles	9
4.1.2. Bateria LiPo 3,7V	10
4.1.3. Bateria pròpia del quadricòpter	10
4.2. Microcontrolador	10
4.2.1. PIC18F46J50	11
4.2.2. PIC18F46J53	12
4.2.3. PIC18F46K80	13
4.2.4. PIC18F46K42	14
4.3. Sensor de dades ambientals	15
4.3.1. DHT22	15
4.3.2. BMP180	16
4.3.3. BME280	16
4.4. Memòria externa	17
4.4.1. EEPROM 24LC256	18
4.4.2. EEPROM 25LC256	18

4.5.	Rellotge en temps real.....	19
4.5.1.	DS1337	19
4.5.2.	GPS	19
4.5.3.	RTCC del microcontrolador	19
4.6.	Sensor de temperatura dels motors.....	20
4.6.1.	LM35.....	20
4.6.2.	TMP35	21
4.7.	Comunicació.....	21
4.7.1.	Bluetooth HC-05	22
4.7.2.	Transceptor HC-12.....	23
5.	SOLUCIÓ ESCOLLIDA	24
5.1.	Alimentació	24
5.2.	Microcontrolador.....	24
5.3.	Sensor de dades ambientals	26
5.4.	Memòria externa	26
5.5.	Rellotge en temps real.....	26
5.6.	Sensor de temperatura dels motors.....	27
5.7.	Comunicació.....	27
6.	HARDWARE	28
6.1.	Alimentació i condicionament del microcontrolador.....	28
6.2.	Configuració dels sensors.....	30
6.3.	Configuració del transceptor.....	33
6.4.	Altres connexions del microcontrolador	33
7.	FIRMWARE	35
7.1.	Sistema d' adquisició de dades	35
7.1.1.	Funció d' adquisició de la temperatura dels motors.....	38
7.1.2.	Funció d' adquisició de dades del GPS.....	39
7.1.3.	Funció d' adquisició de dades del BME280	39
7.1.4.	Funció d' emmagatzemar les dades a la 24LC256.....	40
7.1.5.	Funció d' enviament de dades al receptor	42
7.2.	Sistema de visualització de dades.....	43
8.	SIMULACIONS	45
8.1.	Alimentació, oscil·lador i reinici	46
8.2.	Adquisició de les dades	47

8.2.1.	Bloc I2C.....	47
8.2.2.	Port sèrie	48
8.2.3.	ADCs	48
8.3.	Emmagatzematge de les dades	49
8.3.1.	Resposta als diferents processos	50
8.4.	Comunicació.....	51
8.5.	Recepció de les dades	53
9.	DISSENY DE LA PCB	55
10.	COMUNICACIONS	59
10.1.	Sincronització	59
10.1.1.	Transmissió asíncrona	59
10.1.2.	Transmissió síncrona	60
10.2.	Modes de comunicació	61
10.2.1.	Símplex	61
10.2.2.	Half – Dúplex	61
10.2.3.	Full – Dúplex.....	62
10.3.	Comunicació UART	62
10.4.	Comunicació I2C	63
11.	NORMATIVA	66
12.	ANÀLISI DE L' IMPACTE MEDIAMBIENTAL	67
	CONCLUSIONS	69
	POSSIBLES MILLORES	71
	BIBLIOGRAFIA	73

1. Prefaci

1.1. Objecte del projecte

La principal finalitat d' aquest Treball Final de Grau és la de realitzar un sistema d' adquisició de dades enfocat a un quadricòpter, recopilant les dades i mesures de diferents sensors per a poder fer una traçabilitat d' aquestes durant el recorregut del dron.

Per a dissenyar aquesta placa d' adquisició de dades, un factor molt important era el pes, ja que la intenció és que es pugui aprofitar pel màxim nombre de drons possibles. Així doncs, l' alimentació era un tema molt important, ja que qualsevol bateria pesa força, i no tots els drons podrien suportar aquest pes. Es va pensar en alimentar la placa amb la mateixa bateria del quadricòpter, en aquest cas s' ha utilitzat una bateria de 3,7V, que alimenta el quadricòpter i el sistema.

La placa d' adquisició de dades compta amb els diferents sensors que permeten capturar dades com ara la temperatura dels 4 motors, la temperatura ambient, humitat, pressió, altura, posició mitjançant GPS i la velocitat, juntament amb el dia i la hora exacta gràcies a un rellotge. S' han seleccionat aquestes dades ja que és una barreja de dades les quals algunes d' elles són útils per la vida del dron i evitar certs problemes de sobreescalfament, també hi ha dades ambientals per a poder registrar variacions d' aquestes en funció de la localització i d' altres més d' interès general com pot ser la velocitat. El fet de tenir el GPS i el rellotge permet saber en quin moment i en quina posició s' han enregistrat totes les dades.

Totes aquestes dades són emmagatzemades a una memòria externa i transferides mitjançant un sistema de radiofreqüència de llarg abast a una estació receptora, la qual està formada per una placa *Arduino* que rep una cadena de dades la qual descomposta i mostra per diverses pantalles LCD la informació corresponent.

Per a desenvolupar el projecte de forma correcta s' han anat complint una sèrie d' aspectes:

- Investigació dels components a utilitzar.
- Profunditzar en la programació i característiques del PIC seleccionat.
- Simulació per parts dels diferents elements que conformen el sistema mitjançant Proteus.
- Simulació completa del sistema mitjançant Proteus.
- Enviament de dades i tractament a l' estació receptora.
- Disseny de placa PCB.
- Muntatge en placa PCB.

Tot i que existeixin plaques de desenvolupament comercialitzades com ara *Arduino* i *Raspberry*, s'ha volgut fer èmfasi en la creació d'una placa pròpia, per així poder desenvolupar habilitats de disseny i entendre millor el funcionament d'aquesta, fent-la totalment adaptable a les nostres necessitats.

Per altra banda, l'estació receptora de dades si que s'ha utilitzat una placa comercial ja que la funció d'aquesta és més secundària i senzilla, ja que només serveix per a visualitzar les dades.

1.2. Motivació i justificació

La idea de realitzar aquest Treball Final de Grau sorgeix de l'augment de l'ús de quadricòpters en l'àmbit professional i no lúdic per a realitzar certes tasques que fins ara no es podien dur a terme o, per contra, es feien d'alguna altra manera. La majoria d'aquestes funcions que han adoptat els quadricòpters es relacionen amb la gravació d'imatges aèries, ja sigui per a vigilar una manifestació, fer una cerca en una zona de difícil accés, etc.

Un cop els drons s'han instaurat a la societat, aquests es poden aprofitar de moltes maneres, i inclús millorar alguna de les seves funcionalitats, així que es va pensar que per a començar aquesta nova fase, seria molt útil poder adquirir dades mitjançant el mateix dron. De dades n'hi ha de molts tipus, ja sigui a nivell mecànic o ambiental. En aquest cas, s'han enregistrat una sèrie de dades que permeten saber informació sobre el propi dron i externes a elles, com és el cas de temperatura, humitat i pressió.

Ja que cada vegada s'utilitzen més drons per part de la policia, per a vigilar manifestacions, per exemple, es va pensar en la possibilitat d'incorporar drons que ajudessin en la meteorologia i altres dades mediambientals que poden ser interessants. Així doncs, aquest sistema d'adquisició de dades permetria a un dron de llarga distància anar capturant les dades ambientals al llarg del seu recorregut, per així poder fer un anàlisi en diferents zones i altures d'una mateixa ciutat. Tot i que en aquest treball només s'han tingut en compte algunes variables, es poden afegir moltes més, com ara la detecció dels nivells de pol·lució, la quantitat de llum que hi ha, i una llarga llista de variables.

1.3. Requeriments previs

Per tal de progressar adequadament en aquest treball, es necessiten una sèrie de coneixements que faciliten molt la feina durant el seu desenvolupament. Aquests requeriments no són de molta profunditat i tecnicisme, però sí que abasten un conjunt de coneixements necessaris per a entrebancar-se el mínim possible.

El més essencial és la programació. Molts cops en programació tens unes funcions que has de mirar d'integrar-les en un conjunt de la manera més eficient possible. El fet d'estar acostumat a programar,

fa que davant un problema puguis afrontar diferents solucions que compleixen el que un demanava. Per altra banda, també és important tenir un mínim coneixement sobre programació aplicada a microcontroladors o similar, per tal de saber controlar coses tan importants com són les interrupcions. També són molt importants els sistemes de comunicació, per tal de crear una cadena de dades i interpretar-la posteriorment, per exemple.

Per altra banda també és necessari tenir un coneixement, o si més no acostumar-se, a llegir *Datasheets*. En un treball de disseny és molt important entendre com funciona cada un dels seus dispositius, tan les especificacions com la seva forma de treballar, per tal de poder connectar-lo i programar-lo d' una forma correcte.

Finalment, i no menys important, és important saber com funcionen les alimentacions i les senyals que proporcionen els diferents sensors, ja que molts cops es poden millorar aportant certs condensadors i resistències que, per exemple, poden servir de desacoblament.

2. Introducció

La modificació i personalització de quadricòpters és cada vegada més habitual. Aquests poden anar des d' un ús més lúdic amb vols a una velocitat més baixa, fins drons adaptats per a participar en carreres d' alta velocitat i d' altres per a un ús professional, majoritàriament en l' enregistrament d' imatges. Amb el present treball es vol afegir una nova funcionalitat a aquest món creixent, el de l' enregistrament de dades.

Aquestes dades poden dependre de l' àmbit en que vagi enfocat el sistema d' adquisició de dades, però es podria crear una nova funcionalitat més ambiental, enfocada a la meteorologia i a certs aspectes del medi ambient, adquirint dades d' evolucions de diferents variables meteorològiques o ambientals per a poder rastrejar una certa zona mitjançant un simple vol a partir d' un dron.

2.1. Objectius del treball

El principal objectiu d' aquest treball és el correcte desenvolupament del sistema d' adquisició de dades per a quadricòpters. Per tal d' assolir aquest objectiu general, n' hi ha d' altres més secundaris però no menys importants.

Es pretén adquirir nous coneixements a nivell de disseny electrònic, començant per una bona selecció de components comparant les diferents opcions de mercat, fins a l' aprenentatge de dissenyar una placa PCB entenent els diversos factors que afecten i generar fitxers per a la seva posterior impressió.

A nivell de *firmware* es vol programar un microcontrolador el qual ve donat sense perifèrics i s' ha d' incorporar al projecte. Això fa que sigui necessari entendre totes les seves característiques i saber els seus límits, cosa que obliga a llegir i entendre el *Datasheet* d' un microcontrolador.

El procés de retroalimentació basada en els resultats obtinguts és un objectiu molt interessant, ja que es pretén anar aprenent conforme va avançant el treball i sobretot, millorant a partir d' errors comesos i observats.

Per últim, el fet d' utilitzar programes nous i molt potents en el seu àmbit, permet adquirir coneixements de noves eines que poden ser útils de cara al futur i, per tant, és molt profitós aprendre a treure el màxim profit possible d' elles i ha estat un dels objectius marcats.

2.2. Abast del treball

Aquest projecte final de grau toca diferents àmbits referents al món de l' electrònica, que tots junts componen un producte final però que s' ha de saber dominar cadascun d' ells per separat ja que son completament diferents a l' hora de tractar amb ells.

Es presenten tasques a nivell d' elecció de components i disseny del *hardware*, que ha de permetre conèixer tots els components utilitzats i seleccionar el millor per a l' aplicació final, ja que en cas contrari, el desenvolupament pot ser més costós, ja sigui per complicació d' haver escollit un component més potent del necessari o perquè s' ha quedat curt.

També afecta, evidentment, diverses tasques de programació. Un bloc molt important d' aquest projecte és la programació del microcontrolador per tal de que efectui les comandes esperades en cada cas per a obtenir el resultat esperat.

El disseny de la placa PCB també és un món on s' ha hagut d' endinsar per tal d' aprendre el funcionament d' aquesta per tal de realitzar un bon disseny final.

Barrejant aquestes tres branques genèriques, s' aconsegueix elaborar un producte, però s' ha de ser conscient que es pot anar molt més enllà, aconseguint diversos resultats. Quan es tracta d' un sistema d' adquisició de dades, les diverses possibilitats de creació són infinites, fent així que es pugui completar el projecte afegint moltes altres funcionalitats que no s' han dut a terme en aquest treball, comentades al final de la memòria en l' apartat de possibles millores.

3. Estat de l' art

És Evident que els drons s' han instaurat en la nostre societat. Tot i existir objectes voladors de caire similar des de fa bastants anys, hi ha hagut una irrupció al mercat al llarg d' aquests últims anys. Això és degut a la seva millora en quan al control i s' ha pogut veure que darrere d' aquests aparells hi ha molt potencial amagat.

A simple vista poden semblar objectes voladors sense masses funcionalitats, simplement tenen un ús lúdic, però si un s' endinsa dins d' aquest món, es poden descobrir com realment els quadricòpters estan preparats per a realitzar un munt de tasques de diferents àmbits. La més coneguda i per la qual es van començar a utilitzar els drons a nivell professional, és per l' enregistrament d' imatges. Aquestes permeten obtenir uns angles i perspectives que fins al moment eren molt complicats d' obtenir, i alhora molt costosos. Es van començar a utilitzar per la gravació d' anuncis ja que era molt senzill i no es necessitava un gran abast de connexió.

Un cop es va millorar el vol i l' abast, quan es va veure que realment podien ser útils, van seguir apareixent noves funcions per als quadricòpters, totes elles seguien relacionades amb el fet d' obtenir imatges aèries. S' han aplicat en camps com els del rescat de persones o l' observació de zones on s' han produït desastres naturals, com bé podria ser un bosc.

També hi ha l' existència de nano drons, els quals tenen un format minúscul, els quals fa que siguin menys resistents al vent i sigui difícil efectuar un vol en exterior, però que en espais tancats funcionen molt bé i tenen una gran maniobrabilitat. En aquests casos amb el pas del temps s' ha vist que poden ser útils per a inspeccionar zones costoses, com poden ser canonades i instal·lacions d' aire.

Avui dia hi ha drons que tenen un abast molt i molt gran, permetent controlar-se a varis kilòmetres de distància i aquest factor pot obrir moltes portes noves. El fet de poder veure el món i desplaçar un objecte volador des d' una gran distància, permet fer coses que també es podien dur a terme, però d' una manera molt més ràpida i senzilla, com podria ser en aquest cas l' enregistrament de diverses dades d' interès en una zona concreta a partir d' un vol, fet plantejat en aquest projecte.

El món dels quadricòpters ha anat a l' alça i cada vegada hi anirà a més. Fins ara la majoria de tasques professionals que es realitzen amb ells tenen a veure amb la gravació d' imatges, però en un futur no molt llunyà aquests tindran moltes altres opcions, com pot ser el repartiment a domicili. Aquests cada vegada són més potents, que fins ara és el punt dèbil dels quadricòpters, ja que no poden suportar un gran pes i mantenir una bona estabilitat de vol. El que si sembla més pròxim, es la inclusió dels drons en l' agricultura. Recarregant en petites quantitats, es podrien adaptar per a realitzar tasques de fumigació i fertilització.

És important tenir en compte que un dron no només permet el vol i la dispensació del líquid, sinó que es podria tenir un micro-ordinador intern capaç de realitzar càlculs precisos per a adaptar les seves accions a cada moment, dotant-se així de pràcticament la seva autosuficiència, sempre tenint en compte el factor de les bateries.

Un últim factor a tenir en compte per a l'evolució exponencial dels quadricòpters té a veure amb les bateries. Fins ara aquestes són molt limitades pel tema ja comentat del pes, però s'estan fent molts avenços i ja hi ha alguna existència de drons que poden tenir un vol de llarga durada gràcies a la càrrega de les bateries mitjançant energia solar, aconseguint així que es carreguin durant el seu vol.

4. Possibles solucions

El disseny del sistema d' adquisició de dades s' ha realitzat de manera que compleixi les següents especificacions d' entre altres:

- Dispositiu lleuger perquè no afecti l' estabilitat de quadricòpter.
- Microcontrolador PIC.
- Adquisició de la temperatura dels 4 motors que formen el quadricòpter.
- Adquisició de la temperatura ambient juntament amb la humitat, la pressió i l' altura.
- Adquisició de la posició.
- Adquisició de data i hora.
- Enviament de dades.
- Emmagatzematge de les dades.
- Visualització de les dades.

Per tal de complir a terme aquestes especificacions, s' ha fet un estudi previ per tal de seleccionar els components més adequats en cada cas, ja formin part del software emprat o del hardware. A continuació es mostren les diferents alternatives que es van plantejar en un inici per cada cas.

4.1. Alimentació

Ja que el primer requisit de tots és de dissenyar un dispositiu capaç d' acoblar-se al quadricòpter sense afectar en l' estabilitat d' aquest, un dels temes més essencials era l' alimentació, ja que sol ser el que aporta un pes més elevat i pugui perjudicar més.

Així doncs, les idees que es van plantejar per tal d' alimentar el sistema van ser les següents:

4.1.1. Piles

Es van considerar dos tipus de pila: una pila de petaca de 9 V i la unió de 4 piles AA.

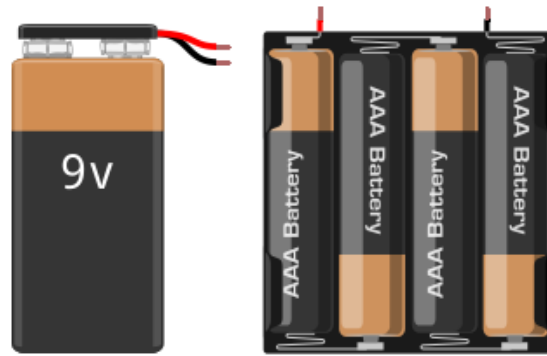


Figura 4-1. Pila de petaca de 9V i 4 piles AA. (Font [27]: www.guimi.net/blogs/hiparco/baterias/)

4.1.2. Bateria LiPo 3,7V

Una altre opció valorada ha estat la d' alimentar el sistema mitjançant una bateria LiPo de 3,7V, que és la mateixa que utilitza el quadricòpter emprat per a les proves (Syma X5C) i és lleugera.



Figura 4-2. Bateria LiPo de 3,7V (Font [28]: <https://www.powerplanetonline.com>)

4.1.3. Bateria pròpia del quadricòpter

Una última opció ha sigut la d' utilitzar la mateixa bateria del quadricòpter per a alimentar ambdós coses, el propi dron i la placa. En aquest cas, tal i com s' ha comentat anteriorment, la bateria seria la del Syma X5C.

4.2. Microcontrolador

El cervell del nostre sistema és el microcontrolador. Així doncs, aquest mateix determinarà les capacitats del projecte. Com a punt de partida s' han escollit els microcontroladors de la marca *Microchip Technology Inc.*, concretament la família PIC18F. Aquesta família de microcontroladors és de 8 bits, però amb més potència que la seva família predecessora PIC16F. Dins d' aquesta família hi ha molts microcontroladors diferents, cadascun amb les seves variants que fan que l' elecció d' aquest

sigui important. Així doncs, s' ha realitzar un primer filtratge d' entre els que oferien millors possibilitats.

4.2.1. PIC18F46J50

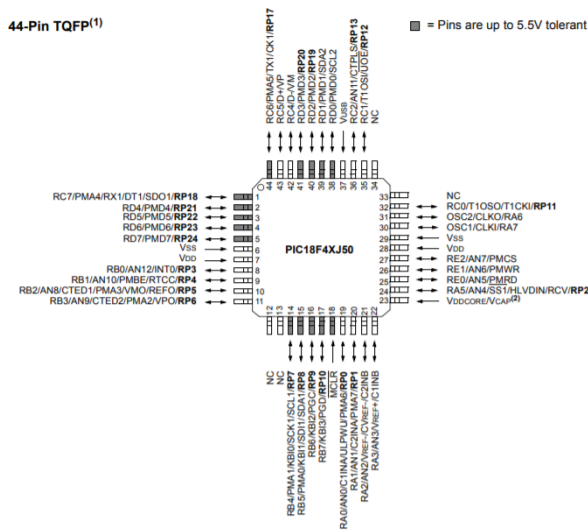


Figura 4-3. Patillatge PIC18F46J50 (Font [7]: www.microchip.com)

Taula 4.1. – Característiques PIC 18F46J50

CARACTERÍSTIQUES	PIC18F46J50
RAM	4kB
ESPAI DEL PROGRAMA	64kB
SERIAL PORTS	2
I2C	Si
CANALS ADC	13
RESOLUCIÓ ADC	10 bits
TIMERS	5
TENSIÓ D' ENTRADA	2.15V / 3.6V
ENTRADES I SORTIDES	33

4.2.2. PIC18F46J53

44-Pin TQFP⁽²⁾

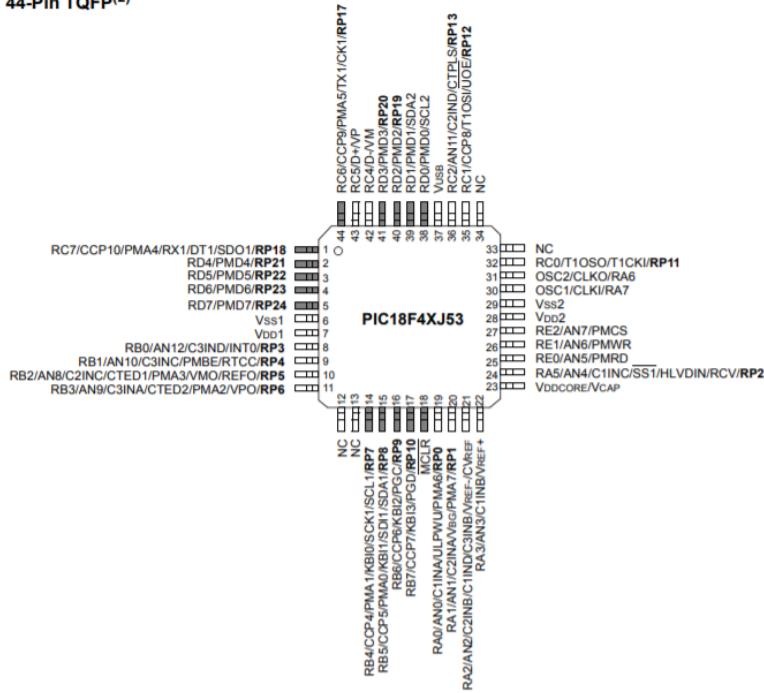


Figura 4-4. Patillatge PIC18F46J53 (Font [7]: www.microchip.com)

Taula 4.2. – Característiques PIC 18F46J53

CARACTERÍSTIQUES	PIC18F46J53
RAM	3.8kB
ESPAI DEL PROGRAMA	64kB
SERIAL PORTS	2
I2C	Si
CANALS ADC	13
RESOLUCIÓ ADC	12 bits
TIMERS	8
TENSIÓ D' ENTRADA	2.15V / 3.6V
ENTRADES I SORTIDES	33

4.2.3. PIC18F46K80

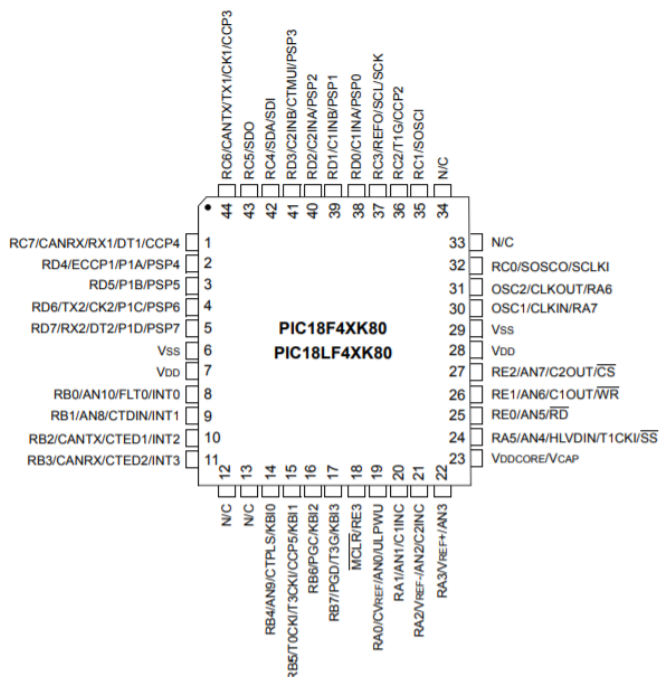


Figura 4-5. Patillatge PIC18F46K80 (Font [7]: www.microchip.com)

Taula 4.3. – Característiques PIC 18F46K80

CARACTERÍSTIQUES	PIC18F46K80
RAM	3.6kB
ESPAI DEL PROGRAMA	64kB
SERIAL PORTS	2
I2C	Si
CANALS ADC	11
RESOLUCIÓ ADC	12 bits
TIMERS	5
TENSIÓ D' ENTRADA	1.8V / 5.5V
ENTRADES I SORTIDES	35

4.2.4. PIC18F46K42

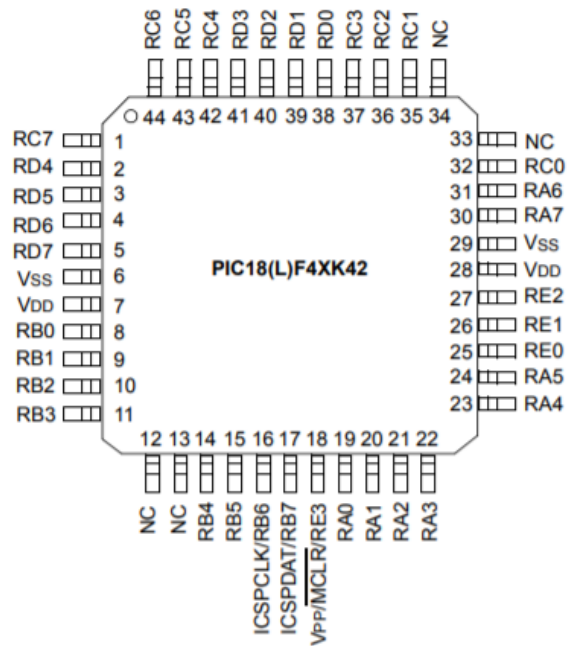


Figura 4-6. Patillatge PIC18F46K42 (Font [7]: www.microchip.com)

Taula 4.4. – Característiques PIC 18F46K42

CARACTERÍSTIQUES	PIC18F46K42
RAM	4kB
ESPAI DEL PROGRAMA	64kB
SERIAL PORTS	2
I2C	Si
CANALS ADC	35
RESOLUCIÓ ADC	12 bits
TIMERS	7
TENSIÓ D' ENTRADA	2.3V / 5.5V
ENTRADES I SORTIDES	36

4.3. Sensor de dades ambientals

Pel que fa a les temperatures que es volen adquirir, es considera que tenim dos temperatures diferents. Per un costat tenim les temperatures del 4 motors que conformen el quadricòpter i per altra banda la temperatura ambiental. En aquest cas s' han buscat sensors per la part d' adquisició de dades ambientals, és a dir, poder obtenir la temperatura ambiental juntament amb la humitat o pressió. S' ha fet un estudi de diverses opcions les quals les resultats han estat les següents.

4.3.1. DHT22

El primer d' ells és el DHT22. Aquest sensor permet mesurar la temperatura i humitat amb una precisió molt acceptable. És de fàcil connexió i les seves característiques són:

Taula 4.5. – Característiques DHT22

CARACTERÍSTIQUES	DHT22
ALIMENTACIÓ	3,3V /6V
RANG DE TEMPERATURA	-40 / 125 °C
PRECISIÓ TEMPERATURA	0,5 °C
RESOLUCIÓ TEMPERATURA	0,1 °C
RANG D' HUMITAT	0 / 100 %
PRECISIÓ HUMITAT	2 %
RESOLUCIÓ HUMITAT	0,1 %
FREQÜÈNCIA DE MOSTREIG	2 Hz

Per a utilitzar aquest sensor, tal i com es pot veure a la Figura 4-7, només s' ha d' alimentar i connectar el pin 2 en un port bidireccional, que posteriorment per codi demanarem la temperatura i/o la humitat i per aquell mateix pin es rebrà.

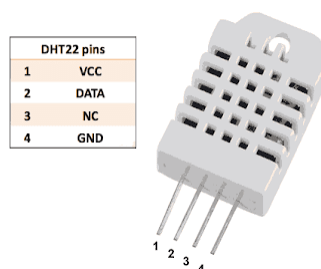


Figura 4-7. Patillatge DHT22 (Font [25]: www.arduinoamue.te.blogspot.com)

4.3.2. BMP180

Aquest altre sensor permet mesurar la temperatura però no la humitat. En canvi, però, sí que té un baròmetre que permet captar la pressió. Les seves característiques són:

Taula 4.6. – Característiques BMP180

CARACTERÍSTIQUES	BMP180
ALIMENTACIÓ	1,8 / 3,6 V
RANG DE TEMPERATURA	0 / 65 °C
PRECISIÓ TEMPERATURA	0,5 °C
RESOLUCIÓ TEMPERATURA	0,1 °C
RANG DE PRESSIÓ	300 / 1100 hPa
PRECISIÓ PRESSIÓ	1 hPa
RESOLUCIÓ DE PRESSIÓ	0,01 hPa
FREQÜÈNCIA DE MOSTREIG	1 Hz

Aquest sensor funciona per I2C, de manera que la seva connexió es basa en l'alimentació del propi sensor i els busos de *Serial Clock (SCL)* i *Serial Data (SDA)*, tal i com es pot veure en la Figura 4-8. D'aquesta manera, mitjançant programació es podran adquirir les dades del sensor amb els propis mètodes del bus I2C, tenint en compte la seva adreça i observant al *Datasheet* les seves adreces internes.



Figura 4-8. Sensor BMP180 (Font [29]: www.cdmxelectronica.com/)

4.3.3. BME280

El BME280 es pot considerar el germà gran del BMP180. Permet mesurar temperatura i pressió, igual que l'anterior, però en aquest cas també es pot obtenir la humitat. Té les següents característiques:

Taula 4.7. – Característiques BME280

CARACTERÍSTIQUES	BME280
ALIMENTACIÓ	1,7 / 3,6 V
RANG DE TEMPERATURA	-40 / 85 °C
PRECISIÓ TEMPERATURA	0,5 °C
RESOLUCIÓ TEMPERATURA	0,01 °C
RANG DE PRESSIÓ	300 / 1100 hPa
PRECISIÓ PRESSIÓ	1 hPa
RESOLUCIÓ DE PRESSIÓ	0,01 hPa
RANG D' HUMITAT	0 / 100 %
PRECISIÓ HUMITAT	3 %
RESOLUCIÓ HUMITAT	0,008 %
FREQÜÈNCIA DE MOSTREIG	1 Hz

El sensor té una interfície que el permet connectar-se mitjançant el bus SPI i el bus I2C. En funció del tipus de comunicació que es vulgui tenir, canvia la forma de connectar-lo i la manera en que es programa. Tal i com es pot veure a la Figura 4-9, té més de 4 pins degut a aquesta variabilitat. Dos d' aquests pins formen part de l' alimentació i els altres es complementen per a utilitzar-lo mitjançant el bus SPI o I2C.

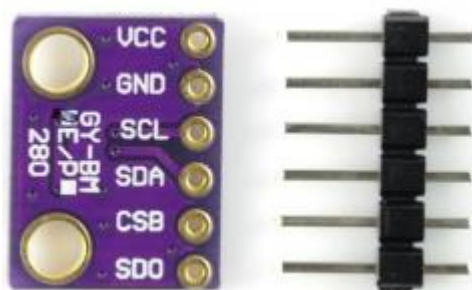


Figura 4-9. Sensor BME280 (Font [30]: <https://botland.com.pl/en>)

4.4. Memòria externa

Aquest sistema també consta d' una memòria externa on emmagatzemar les dades que es van capturant, alhora que es van transmetent a l' estació receptora. És per això que també es varen

proposar un parell de memòries, totes elles EEPROM, d' aquesta manera podem escriure i esborrar mitjançant la programació, a través de les sortides del microcontrolador. S' han proposat memòries de 256 Kbits, però en qualsevol cas seria fàcil modificar-les si es necessités més.

4.4.1. EEPROM 24LC256

Aquesta memòria EEPROM treballa mitjançant el bus I2C. S' alimenta a una tensió d' entre 2,5V i 5,5V. La seva freqüència de treball màxima és de 400 kHz en un rang de temperatures que varia entre els -40 °C i els 125 °C.

El seu cicle d' escriptura i esborrat és de 1.000.000 de cicles i té una retenció de dades superior als 200 anys.

Tal i com es pot veure a la Figura 4-10, el seu patillatge consta de 8 pin, 2 d' ells són els d' alimentació i uns altres 2 es corresponen al bus I2C, sent aquests el *SDA (Serial Data)* i el *SCL (Serial Clock)*. Dels 4 pins restants, 3 d' ells permeten variar la seva direcció I2C, fent així que es puguin connectar fins a 8 memòries EEPROM sense que les seves direccions es trepitgin i puguin funcionar de forma simultània. L' últim connector és un protector d' escriptura *Hardware*, el qual es pot controlar mitjançant el microcontrolador o és pot desactivar connectant-lo a massa.

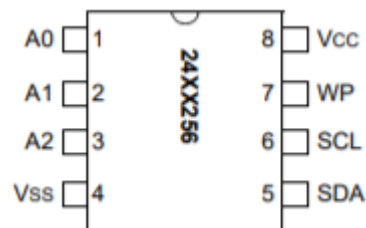


Figura 4-10. Patillatge 24LC256 (Font [7]: www.microchip.com)

4.4.2. EEPROM 25LC256

La EEPROM 25LC256 és la germana bessona de la mencionada anteriorment. Les seves diferències principalment es basen en el seu bus de dades. La 25LC256 treballa mitjançant el bus SPI, és aquí on resideix la seva gran diferència. Pel que fa a la resta de característiques, són idèntiques a la 24LC256, excepte la freqüència de treball, que en aquest cas pot arribar fins a 10 MHz degut a la comunicació que utilitza. Per altra banda aquesta memòria pot treballar a temperatures d' entre -55 °C i 125 °C.

Segons la Figura 4-11, els seu patillatge es correspon a l' alimentació i la protecció d' escriptura. Les altres 5 connexions formen part de la comunicació SPI amb la qual treballa aquesta memòria.

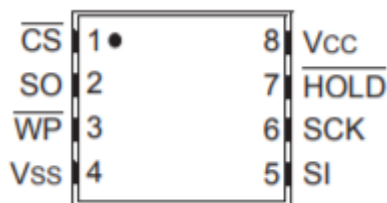


Figura 4-11. Patillatge 25LC256 (Font [7]: www.microchip.com)

4.5. Relotge en temps real

El moment en que s'adquireixen les dades també és important saber-lo, per a poder controlar la seva evolució al llarg del temps, tenint en compte el moment del dia en que s'efectuen les mesures. És per això que és important disposar d'un rellotge que ens permeti saber el moment exacte quan ens sigui necessari.

4.5.1. DS1337

Aquest integrat es comunica mitjançant el protocol I2C. Amb l'ajut d'un cristall i l'alimentació, permet obtenir la data i l'hora. Si el voltatge és superior a 1,3V, aquest manté el temps real sense cap tipus de problema. La seva alimentació està compresa entre 1,8 i 5,5V.

4.5.2. GPS

El GPS també és capaç d'obtenir la data i l'hora a partir dels satèl·lits. És una opció viable ja que ens estalviaria un component a la placa però té l'inconvenient que necessita els satèl·lits per a poder funcionar correctament.

4.5.3. RTCC del microcontrolador

El microcontrolador, de la mateixa manera, també disposa d'un rellotge de temps real que funciona mitjançant la programació del propi microcontrolador. Així doncs tampoc es necessitaria cap element extern, excepte un oscil·lador.

4.6. Sensor de temperatura dels motors

A l' hora d' adquirir la temperatura dels diferents motors que conformen el dron, s' ha de tenir en compte a on han d' anar col·locats. En aquest cas, el model de dron utilitzat té uns compartiments pels motors petits, és per això que en la cerca dels sensors adequats, la mida i la facilitat de manipulació ha sigut un terme molt important.

Finalment, entre els possibles sensors a utilitzar, han acabat destacant dos:

4.6.1. LM35

Aquest sensor de temperatura digital té un circuit de control intern que proporciona una sortida lineal en funció de la temperatura. Les seves principals especificacions són les següents:

Taula 4.8. – Característiques LM35

CARACTERÍSTIQUES	LM35
ALIMENTACIÓ	4 / 30 V
RANG DE TEMPERATURA	-55 / 150 °C
PRECISIÓ TEMPERATURA	0,5 °C
RESOLUCIÓ TEMPERATURA	0,1 °C
FACTOR D' ESCALA LINEAR	10 mV/°C

A la Figura 4-12 es pot veure el sensor amb el seu patillatge. La seva connexió és molt senzilla ja que dos pots serveixen per a l' alimentació i la tercera és la sortida, que ha d' anar connectada a una entrada del ADC del microcontrolador per a convertir el voltatge de sortida en temperatura.

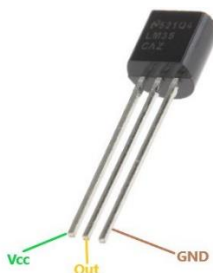


Figura 4-12. LM35 (Font [12]: www.circuits-diy.com/)

4.6.2. TMP35

Pel que fa al TMP35, aquest té molta similitud amb el mencionat anteriorment. Funciona de la mateixa manera, amb el seu control intern que dóna un voltatge de sortida en funció de la temperatura, però la principal diferència recau en l' alimentació d' aquest.

Taula 4.9. – Característiques TMP35

CARACTERÍSTIQUES	TMP35
ALIMENTACIÓ	2,7 / 5,5 V
RANG DE TEMPERATURA	-40 / 125 °C
PRECISIÓ TEMPERATURA	2 °C
RESOLUCIÓ TEMPERATURA	0,1 °C
FACTOR D' ESCALA LINEAR	10 mV/°C

També varia lleugerament el rang de temperatures i la precisió d' aquesta, sent en aquest cas pitjor que l' altre, ja que abasteix un rang una mica menor amb una precisió força pitjor. A simple vista son pràcticament idèntics i el seu patillatge es pot veure a la Figura 4-13.



Figura 4-13. Patillatge TMP35 (Font [31]: www.analog.com/)

4.7. Comunicació

Totes les dades adquirides mitjançant la placa d' adquisició de dades, són enviades a una estació receptora per tal de poder visualitzar les dades durant el vol del quadricòpter, ja que la primera placa es trobarà adjunta al dron. És per això que ha d' haver-hi una comunicació sense fils per tal de poder enviar les dades en temps real i així poder visualitzar-les. Per tal de poder dur a terme aquesta comunicació, es van proposar el següents mòduls de comunicació:

4.7.1. Bluetooth HC-05

Aquest dispositiu Bluetooth és molt popular degut a la seva simplicitat. Es comunica amb el microcontrolador mitjançant el port UART i accepta les comandes AT per a programar-se. A continuació es mostren les seves característiques més importants:

Taula 4.10. – Característiques HC-05

CARACTERÍSTIQUES	HC-05
ALIMENTACIÓ	1,8 / 5,5 V
AMPLE DE BANDA	2,4 GHz
SENSIBILITAT TÍPICA	-80 dBm
BAUD RATE	9600 – 460800 bauds
ABASTIMENT	10 m
POTÈNCIA DE TRANSMISSIÓ	4 dBm

Com es pot veure a la Figura 4-14 és un dispositiu amb una velocitat de transmissió força elevada però un abastiment més limitat. El seu patillatge és molt senzill ja que disposa de 6 pins, dos d'ells són d'alimentació i uns altres dos els que reben i transmeten les dades mitjançant UART (Rx i Tx). Els altres dos pins restants són el d' *Enable*, que permet activar o desactivar l' emparellament entre mòduls i el de *Status*, que com el seu nom diu, indica l' estat en que es troba el dispositiu, si està emparellat o no.



Figura 4-14. HC - 05 (Font [32]:www.beatyourbit.wordpress.com/)

4.7.2. Transceptor HC-12

Aquest transceptor té la mateixa idea de cara a l' usuari que l' anterior, amb la diferència que en aquest cas no és un Bluetooth i, per tant, algunes de les seves característiques varien. Té comandes AT i funciona pel port UART exactament igual que l' anterior.

Taula 4.11. – Característiques HC-12

CARACTERÍSTIQUES	HC-05
ALIMENTACIÓ	3,2 / 5,5 V
AMPLE DE BANDA	473 MHz (100 canals)
SENSIBILITAT TÍPICA	-117 dBm
BAUD RATE	1200 – 115200 bauds
ABASTIMENT	1 / 2 Km
POTÈNCIA DE TRANSMISSIÓ	20 dBm

El seu patillatge consta de 5 pins, dos d' ells d' alimentació i dos d' ells de transmissió i recepció de dades. El cinquè pin permet posar el Bluetooth en el mode de comandes AT si aquest es connecta a massa. A la Figura 4-15 es pot veure el dispositiu mencionat.



Figura 4-15. HC - 12 (Font [33]:www.elecrow.com/)

5. Solució escollida

Un cop analitzades totes les opcions, es va pensar que el més adequat per a realitzar el projecte seria comptar amb els següents components:

Taula 5.1. – Components escollits

FUNCIÓ	COMPONENT
ALIMENTACIÓ	Bateria del quadricòpter
MICROCONTROLADOR	PIC18F46J53
SENSOR DE DADES AMBIENTALS	BME280
MEMÒRIA EXTERNA	24LC256
RELLOTGE DE TEMPS REAL	DS1337
SENSOR DE TEMPERATURA DELS MOTORS	TMP35
COMUNICACIÓ	HC-12

5.1. Alimentació

Un aspecte molt important a tenir en compte de cara al disseny final de la placa d'adquisició de dades té a veure amb el seu pes. Aquesta placa es pretén que es pugui incloure en molts drons i, per a fer les proves, s'ha utilitzar un quadricòpter de baixa gamma i amb baixa potència, és per això que el pes que pot aixecar és molt limitat. Ja que les bateries aporten un pes força elevat respecte el conjunt, s'ha optat per aprofitar la pròpia bateria del quadricòpter, ja que la placa no té un consum molt elevat i afectarà poc a la duració del vol. Així doncs, la decisió ha estat presa per tal d'estalviar pes. Al ser aquesta una bateria de 3,7V, això afecta a la resta de seleccions, ja que es necessiten components que puguin operar a aquest voltatge.

5.2. Microcontrolador

L'elecció del microcontrolador va ser la més complicada, ja que tot el projecte girava entorn seu i és molt més difícil de modificar un cop començat el projecte que no pas un sensor. Aquest microcontrolador havia de tenir algunes característiques que ja es van tenir en compte en les possibles solucions, possiblement la més important era que havia de tenir un mínim de dos ports sèrie, ja que tant el GPS com la comunicació sense fils utilitzen aquests ports.

Tot i que tots els microcontroladors pre-seleccionats podrien ser vàlids, ens havíem de decantar per un. La Taula 5.2 ens mostra a simple vista les diferents característiques dels diferents microcontroladors.

Taula 5.2. – Característiques dels microcontroladors

CARACTERÍSTIQUES	PIC18F46J53	PIC18F46J53	PIC18F46K80	PIC18F46K42
RAM	4kB	3.8kB	3.6kB	4kB
ESPAI DEL PROGRAMA	64kB	64kB	64kB	64kB
SERIAL PORTS	2	2	2	2
I2C	Si	Si	Si	Si
CANALS ADC	13	13	11	35
RESOLUCIÓ ADC	10 bits	12 bits	12 bits	12 bits
TIMERS	5	8	5	7
TENSIÓ D' ENTRADA	2.15V / 3.6V	2.15V / 3.6V	1.8V / 5.5V	2.3V / 5.5V
ENTRADES I SORTIDES	33	33	35	36

Tot i que hi ha molta similitud entre els diferents candidats, els quals ja es va buscar que tinguessin una memòria RAM i un espai de programa suficient que no ens pogués donar cap tipus de problema, n' hi ha dos que destaquen d' entre aquests quatre, ja que varien en el nombre de *timers*. Aquests dos són els PIC 18F46J53 i 18F46K42. Aquest segon té moltes més entrades ADC, pràcticament la seva totalitat d' entrades i sortides, ja que és un microcontrolador orientat a tenir-hi molts sensors connectats, però, en canvi, té un *timer* menys i algunes altres característiques més internes són inferiors respecte al PIC18F46J53.

Finalment es va decidir optar pel PIC18F46J53, ja que té més *timers* en cas que aquests siguin necessaris i ens permetin incorporar més funcions, a part de que té moltes altres característiques que permet obtenir una millor funcionalitat i, evidentment, moltes d' altres que no són necessàries per aquest projecte.

Cal comentar que al haver escollit una bateria de 3,7V (la del propi quadricòpter), també s' ha tingut en compte que tots els microcontroladors es puguin alimentar a aquest voltatge.

5.3. Sensor de dades ambientals

En aquest cas la decisió s'ha pres en quin era el sensor més complet. El BME280 es pot considerar com una combinació dels altres dos sensors, ja que ens permet obtenir la temperatura, la humitat i la pressió, sent així l'únic dels tres sensors que combina pressió i humitat.

Era interessant poder obtenir la pressió ja que mitjançant aquesta, es pot fer una estimació força acceptable de l'altitud, fent que durant el vol es pugui enregistrar l'altura a la que ha anat sobrevolant el quadricòpter.

5.4. Memòria externa

L'elecció de la memòria en aquest cas ha estat purament per preferència com a programador. Anteriorment s'havien buscat diferents mides de memòria i es va decidir que amb 256Kbits seria suficient. D'entre les dos proposades es va acabar escollint la 24LC256.

L'única diferència amb l'altra candidata era el bus que utilitzava i la freqüència a la que pot treballar. Tot i que la freqüència de treball de la memòria seleccionada és inferior, per aquest projecte no es necessita una freqüència tan elevada, ja que les dades s'hauran de guardar cada un temps determinat el qual serà de un o varis segons, és per això que no afecta. La 24LC256 treballa per I2C, el qual ha estat la diferència que ha fet que aquesta sigui finalment l'escollida.

5.5. Rellotge en temps real

Si ens fixem en aquest apartat, la decisió és menys determinant ja que, a efectes pràctics, tots donen el mateix resultat, que és la data i l'hora. És cert que en un cas s'hauria d'utilitzar el bus I2C (DS1337) però no és un problema ja que hi ha altres components que l'utilitzen i només seria qüestió d'afegir l'adreça d'aquest.

Tot i que el microcontrolador disposa de rellotge en temps real, si aquest perdés l'alimentació per falta de bateria o alguna altre circumstància, l'hora quedaria desfasada i, a cada canvi de bateria del quadricòpter, aquesta es perdria. Per altra banda el GPS pot obtenir l'hora, però els sistemes que funcionen per satèl·lits poden fallar per pèrdua de comunicació, i si el GPS en algun moment no trobés satèl·lit, a part d'afectar en les coordenades afectaria al rellotge.

Així doncs, per tot això, s'ha decidit optar pel DS1337, que juntament amb una petita alimentació pròpia sembla ser el més efectiu.

5.6. Sensor de temperatura dels motors

El sensor seleccionat en aquest cas depèn d' un únic factor, l' alimentació. Al haver seleccionat la bateria del quadricòpter per a alimentar la placa, l' LM35 té una alimentació mínima de 4V a la qual no hi podem arribar. El TMP35, per contra, es pot alimentar a partir de 2,7V.

5.7. Comunicació

Tot i que l' HC-05 pugui treballar a unes freqüències més elevades, té un rang molt menor. S' ha optat per l' HC-12 ja que el rang d' aquest és molt més elevat i té unes característiques suficientment bones per aquest projecte.

Cal comentar que l' abast màxim del quadricòpter amb el que es fan les proves és de 50m, als quals l' HC-05 no hi arriba i l' HC-12 té un rang superior. Tot i això, la idea d' aquest sistema d' adquisició de dades és que es pugui implementar en diferents drons, així doncs per alguns l' HC-12 seria suficient i per altres es podria implementar algun altre sistema més complex per a tenir un rang de cobertura molt superior.

6. Hardware

En aquest apartat es descriu el disseny de hardware realitzat detalladament així com la justificació de tots els components que en formen part. Per a entendre una mica millor tot l' esquema, s' explicaran les diferents parts que en formen part de manera detallada. Així doncs, es pot resumir que el prototip consta dels següents blocs:

- Alimentació i condicionament del microcontrolador.
- Configuració dels sensors.
- Configuració del transceptor.
- Altres connexions del microcontrolador.

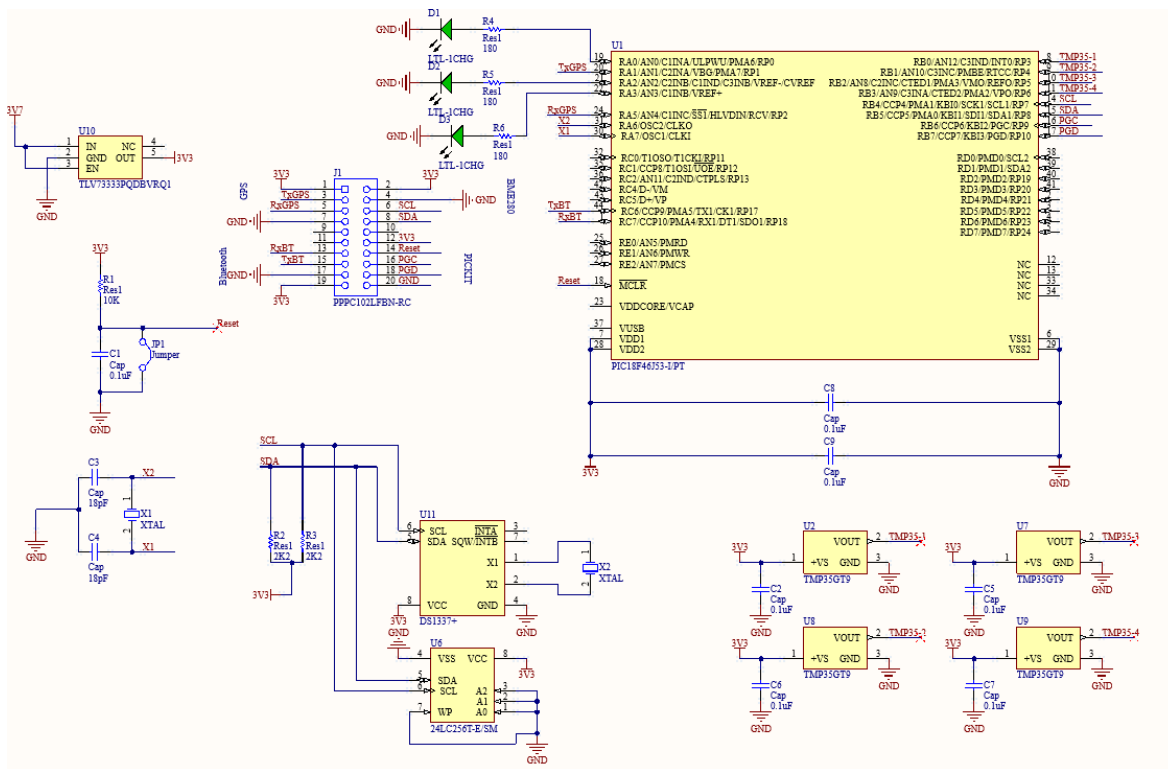


Figura 6-1. Esquema electrònic

6.1. Alimentació i condicionament del microcontrolador

Tal i com s' ha comentat anteriorment, la bateria que alimenta al quadricòpter ens proporciona un voltatge de 3,7V. Per tal de fer més senzilles les connexions posteriors, es va decidir adequar aquest voltatge a un que suportés tots els components que conformen la placa, ja que alguns d' ells tenen un

voltatge d' entrada màxim de 3,6V. Així doncs, aquest voltatge d' entrada que proporciona la bateria del dron, s' envia a un regulador de tensió que ens dóna una sortida estable a 3,3V.

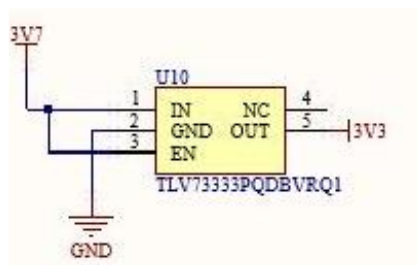


Figura 6-2. Esquemàtic regulador de tensió a 3,3V

A la Figura 6-2 es mostra l' esquemàtic del regulador emprat. S' ha optat per un TLV73333PQDBVRQ1 ja que és un regulador de baixa caiguda (LDO). Els reguladors d' aquest tipus permeten regular el voltatge de sortida quan l' alimentació és un valor molt pròxim a la sortida desitjada. A més a més, no tenen soroll de commutació ja que aquesta no existeix. Aquest model en concret accepta voltatges d' entrada d' entre 1,4 i 5,5V i dóna una intensitat de sortida de 300mA. El pin *EN* (*Enable*) del regulador de tensió va també connectat a l' entrada ja que volem que aquest sempre estigui actiu.

Un cop s' ha obtingut la sortida de 3,3V, aquesta és la que s' utilitzarà per a alimentar tots els components que formen el sistema d' adquisició de dades del quadricòpter, inclòs el microcontrolador, al qual juntament amb l' alimentació es posaran condensadors de 0,1µF per a desacoblar, tal i com recomana el fabricant al seu *Datasheet*.

Per tal de que el microcontrolador funcioni de forma correcte, hi ha dos petits circuits bàsics que són necessaris: el circuit de reinici i el cristall d' oscil·lació.

A la Figura 6-3 es pot observar el circuit de reinici. Cal saber que el pin 18 del PIC18F46J53 és el que permet reiniciar el programa. Aquesta entrada del microcontrolador funciona amb lògica inversa, és a dir, per activar el reinici l' entrada ha de ser un 0 digital. Així doncs, amb aquest esquemàtic aconseguim que l' entrada sigui sempre un 1 digital (no reinici) quan el *Jumper* no està connectat, ja que entren els 3,3V. Per contra, si es connecta el *Jumper*, la resistència consumeix els 3,3V de manera que el pin d' entrada està connectat a massa, forçant així el reinici. Els valors de resistència i condensador s' han obtingut a partir del *Datasheet* del fabricant.

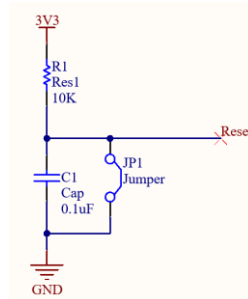


Figura 6-3. Esquemàtic del circuit de reinici

Per altra banda, a la Figura 6-4 hi tenim l' esquema referent a l' oscil·lador extern. Aquest oscil·lador és un cristall de quars que s' encarregarà de portar el *timing* del software, marcant els temps del microcontrolador i dels seus *timers* interns, ja que per software es dirà que en aquest cas l' oscil·lador és extern. L' esquema també es genèric juntament amb el valor dels condensadors, que són els recomanats per a un oscil·lador de 16MHz. S' ha escollit un oscil·lador d' aquesta freqüència per tal de tenir més possibilitats en quan al software per a fer accions ràpides i poder controlar els *timers* d' una forma més senzilla, ja que amb menys freqüència podria haver-hi hagut problemes o complicacions més difícils de solucionar, com ara controlar els *timers* mitjançant bucles.

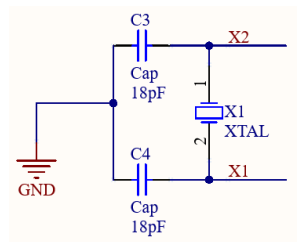


Figura 6-4. Esquemàtic de l' oscil·lador extern

6.2. Configuració dels sensors

De cara a l' adquisició de dades, hi ha varis sensors que entren en joc, els quals s' han comentat anteriorment juntament amb el motiu de la seva elecció. Així doncs, en aquest capítol, es referirà a la manera en que s' ha connectat cada sensor i component, tenint en compte el seu tipus de comunicació amb el microcontrolador.

Per una banda tenim els sensors de temperatura dels 4 motors que conformen el quadricòpter. Aquests són els més senzills ja que cada un ha d' anar connectat a una entrada que pugui treballar amb els ADC del propi microcontrolador, el qual farà la lectura i posteriorment per software s' obtindrà el valor de temperatura. Els 4 sensors tenen un condensador de 0,1 μ F a l' alimentació per tal de desacoblar i que no hi hagi variacions en la sortida.

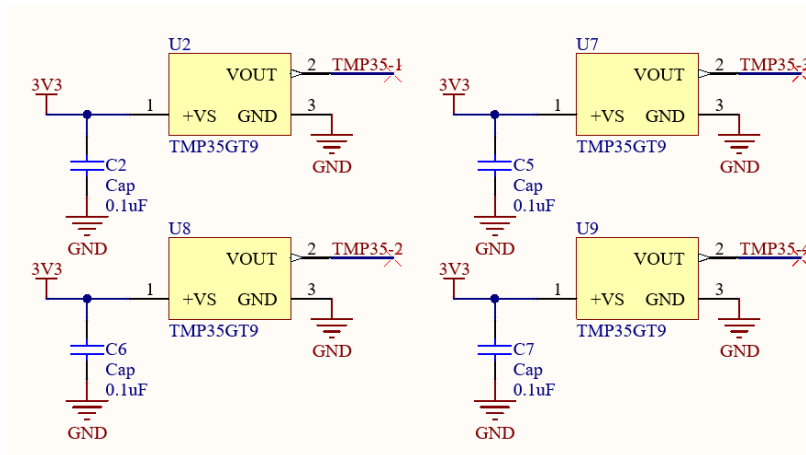


Figura 6-5. Esquemàtic dels sensors de temperatura dels motors

Per altra banda ens trobem amb els components que treballen mitjançant el bus I2C, els quals en aquest cas són el sensor de mesures ambientals (temperatura, humitat i pressió), el rellotge de temps real i la memòria EEPROM externa. Al treballar per I2C, es connecten tots a les mateixes línies de SDA (Serial Data) i SCL (Serial Clock), per tal d' accedir a ells per software a partir de la seva direcció. A la Figura 6-6 es pot veure com s' han connectat el rellotge i la memòria al bus, juntament amb dos resistències de pull-up, ja que el bus I2C és de col·lector obert i es necessiten aquestes resistències per al correcte funcionament.

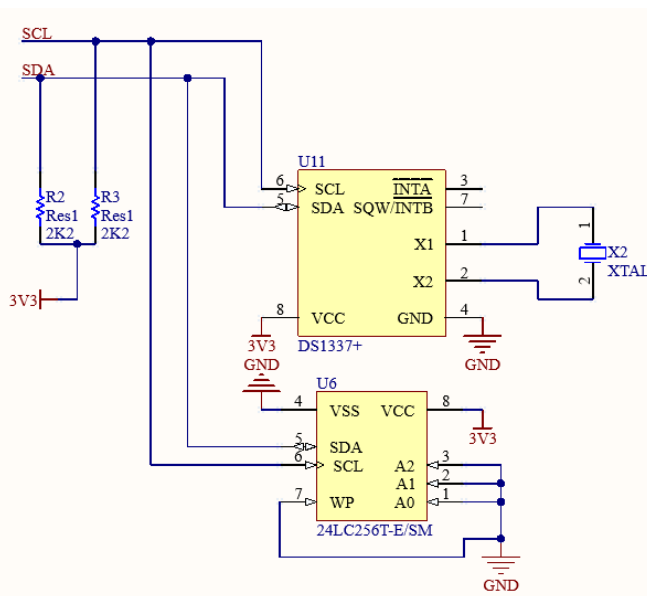


Figura 6-6. Esquemàtic del bus I2C amb el rellotge i la EEPROM

Entrant més en detall en l' esquemàtic de la Figura 6-6, es pot veure com els components tenen altres connexions a part de l' alimentació i la del propi bus. Començant pel rellotge de temps real (DS1337),

té dos patilles que fan referència a interrupcions que es poden executar, però que en aquest cas no s'utilitzen i es deixen a l'aire. Les dos connexions restants són referents a un oscil·lador de quars que, tal i com indica el fabricant, és de 32,768kHz.

La memòria EEPROM també té els pins A2, A1 i A0 que fan referència a la seva direcció amb la qual s'accedirà mitjançant el bus I2C. Modificant aquests pins permet connectar fins a 8 memòries EEPROM del mateix model de forma simultània. En aquest cas, al tenir-ne només una, s'han connectat a massa. Per últim, el WP (Write Protection) defineix si es pot escriure a la memòria ('0') o només es pot llegir ('1'). Com que en aquest cas es vol escriure i llegir a la memòria, també s'ha connectat a massa. Així doncs, tal i com es veu a la Figura 6-7, l'adreça resultant de la memòria és 10100000, equivalent a A0 en hexadecimal per a escriure i A1 per a llegir.

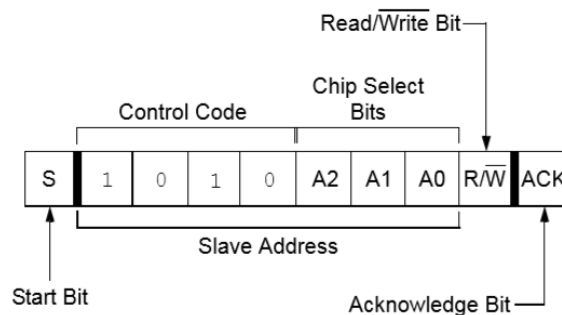


Figura 6-7. Format d'adreça de la EEPROM 24LC256 (Font [7]: www.microchip.com)

A l'esquemàtic de la Figura 6-6 s'hi troba a faltar el sensor de dades ambientals (BME280), que també treballa per I2C. En aquest cas s'ha separat al esquemàtic dels altres dos components ja que el seu encapsulat és diferent i en aquest cas no va soldat a la placa, tal i com s'explicarà posteriorment. Així doncs, aquest dispositiu juntament amb uns altres, van connectats a una capçalera femenina degut a que el model obtingut porten els pins incorporats del tipus mascle.

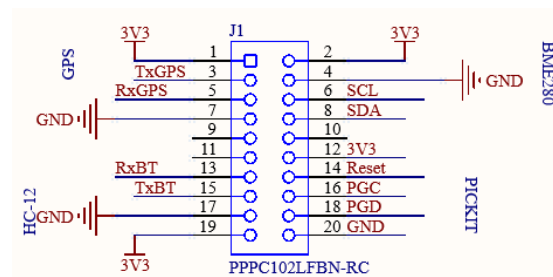


Figura 6-8. Esquemàtic de la capçalera femenina

A la part superior dreta de la capçalera, tal i com es pot veure a la Figura 6-8, hi tenim el BME280. En aquest cas la connexió és encara més senzilla ja que consta de l'alimentació i la connexió al bus I2C

mencionat anteriorment, que ja porta les resistències de pull-up incorporades a l' altre part de l' esquemàtic.

Aquí també s' hi pot trobar un altre component molt important com és ara el GPS. Aquest va connectat directament al port sèrie del microcontrolador, ja que funciona per UART. De la mateixa manera que el BME280, s' ha connectat mitjançant aquest *header* ja que l' encapsulat ve adaptat d' aquesta manera.

6.3. Configuració del transceptor

Per tal de comunicar-se amb l' estació receptora, es necessita el transceptor, el dispositiu HC-12. A la part inferior esquerra de la Figura 6-8 s' hi pot veure el seu esquemàtic. Aquest funciona exactament de la mateixa manera que el GPS, dos pins d' alimentació juntament amb els que van connectats al port sèrie del microcontrolador.

Cal afegir que ambdós dispositius no poden anar connectats al mateix port sèrie, ja que s' han d' utilitzar de forma simultània. És per això que una condició indispensable que havia de tenir el microcontrolador era que aquest disposés d' un mínim de dos ports sèrie. Per aquest motiu la nomenclatura de les connexions és diferent, distingint els Tx (Transmit) i Rx (Receive) del GPS i del transceptor, en aquest cas anomenat BT.

6.4. Altres connexions del microcontrolador

Per al disseny de la placa del sistema d' adquisició de dades, també s' han tingut en compte la incorporació d' altres components més secundaris i connexions essencials per tal de poder tenir un correcte funcionament.

Un primer factor important és el fet de tenir uns pins per a programar el microcontrolador, així un cop es té la placa amb el microcontrolador incorporat, es pot carregar el firmware sense necessitat de treure'l de la pròpia placa. Així doncs, tal i com es pot veure a la Figura 6-8, s' han reservat uns pins de la capçalera (part inferior dreta) per a connectar el programador a les potes corresponents del microcontrolador. Aquestes connexions s' han fet mitjançant el header ja que és la forma més senzilla de connectar-hi els diferents programadors existents, els quals es poden connectar mitjançant cables mascle.

Finalment, i no menys important, s' han afegit 3 LEDs que es programaran per a donar certa informació al usuari, com ara l' enviament de dades i la correcte recepció dels satèl·lits del GPS. Aquests LEDs tenen una resistència en sèrie de 180 Ω per evitar que aquests es fonguin.

A la Figura 6-9 s'hi poden veure totes les connexions que s'han anat comentant anteriorment juntament amb els 3 LEDs, connectats a les sortides 19, 21 i 22, corresponents a RA0, RA2 i RA3 respectivament.

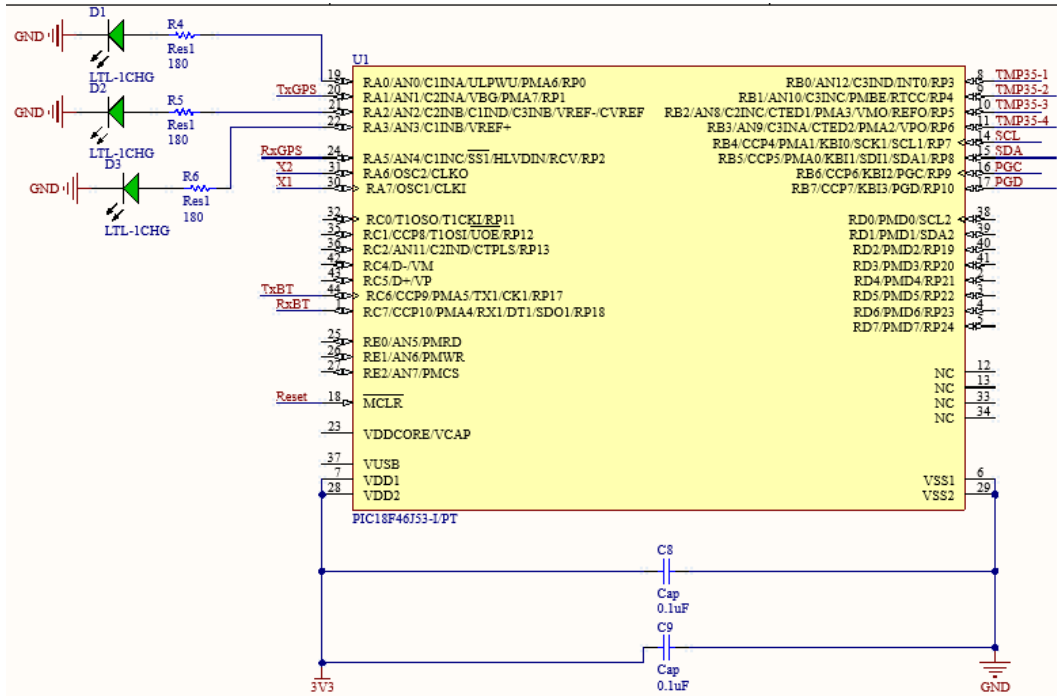


Figura 6-9. Esquemàtic del microcontrolador PIC18F46J53

La selecció d'entrades i sortides a utilitzar per part del microcontrolador, s'ha establert en funció dels ports que s'han d'utilitzar, amb la preferència que estiguessin en entrades/sortides contigües els dispositius del mateix caràcter, com ara els sensors de temperatura dels motors.

Com es pot observar a la Figura 6-9, no s'utilitzen totes les entrades i sortides del microcontrolador. Es podria haver escollit un encapsulat amb menys pins però aquest era inexistent pel PIC seleccionat. Així doncs, aquests pins restants podrien ser utilitzats en un futur per a afegir altres entrades, sortides o comunicacions.

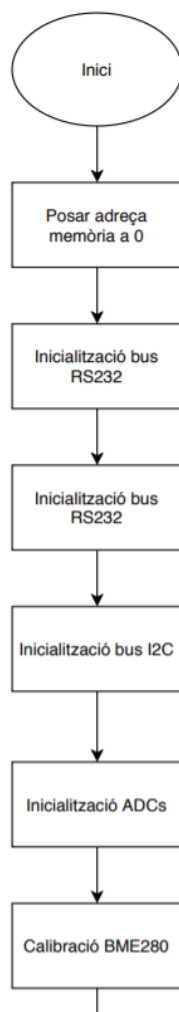
7. Firmware

7.1. Sistema d' adquisició de dades

Un cop es té la certesa que la implementació del hardware és la correcta, s' ha de procedir a la programació del microcontrolador, el que anomenarem firmware.

En aquest cas, el microcontrolador és l' encarregat d' obtenir les diferents dades i paràmetres a estudiar, empaquetar-los i enviar-los per tal que l' estació receptora desempaqueti les dades rebudes i sigui capaç d' analitzar-les i mostrar-les. Al ser un procés automàtic, que s' inicia un cop hi ha comunicació amb els satèl·lits i entre els transceptors, no hi ha cap tipus de menú ni botons que modifiquin el comportament del microcontrolador.

Així doncs, l' estructura que ha de seguir el firmware del microcontrolador que s' incorpora al dron és la següent:



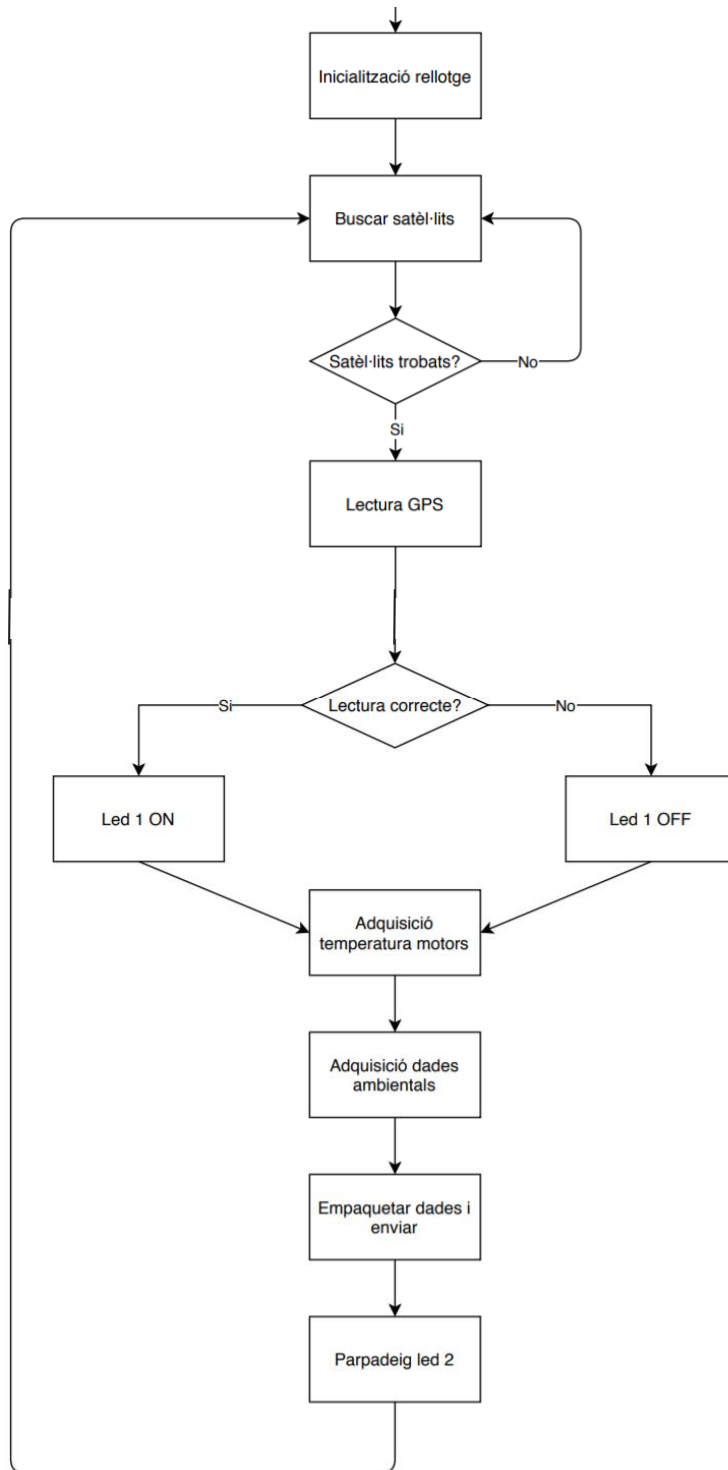


Figura 7-1. Diagrama de flux principal del firmware del microcontrolador

Paral·lelament al programa principal, hi ha un temporitzador que executa una subrutina. Aquest temporitzador és de 2 segons i permet que les últimes dades obtingudes es guardin a la memòria EEPROM externa.

Aquests 2 segons han estat calculats de la següent manera: sabent que la memòria té 256 Kbits, és un equivalent a 32.768 bytes. Per a cada cicle d'emmagatzematge, totes les dades sumen un total de 39 bytes.

Si s'emmagatzemen les dades cada segon, això equivaldria a omplir la memòria en 14 minuts aproximadament: $\frac{32.768 \text{ bytes}}{39 \text{ bytes/cicle}} \cdot 1 \frac{\text{segon}}{\text{cicle}} \cdot 1 \frac{\text{minut}}{60 \text{ segons}} \approx 14 \text{ minuts}$.

En canvi, si el temporitzador el posem cada 2 segons, s'obté un emmagatzematge complet en 28 minuts. S'ha optat per aquest segon ja que alguns drons poden superar els 14 minuts de vol i, en canvi, 28 minuts hi ha pocs drons que ho superin. Si sigues el cas es podria modificar aquest *timer* per a no omplir la memòria abans que finalitzi el vol. El diagrama de flux d'aquest temporitzador és molt senzill, i s'executa paral·lelament amb el principal:

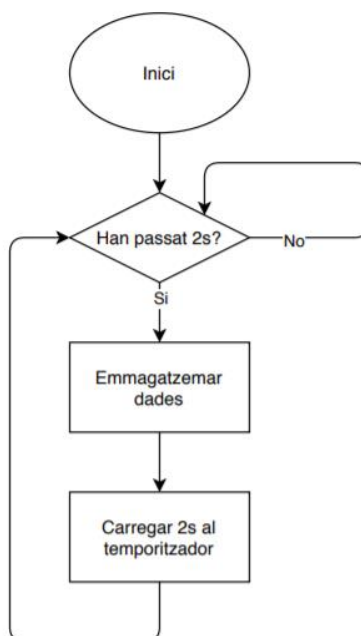


Figura 7-2. Diagrama de flux del temporitzador del microcontrolador

Per tal de tenir un codi ben ordenat, el qual es veurà als annexes, s'ha treballat mitjançant la implementació de funcions i *drivers* per tal de saber en tot moment quina part del codi s'ha de modificar en el cas que sigui necessari, també ajudant al seu enteniment i al seu desenvolupament. Així doncs, s'ha optimitzat el codi de tal manera que a la funció principal (el *main*) pràcticament només hi ha execucions de funcions, que aquestes es poden trobar dins del fitxer principal o en altres fitxers diversos.

A banda de les funcions d' inicialització, es comentarà el funcionament d' aquelles funcions que permeten adquirir dades i treballar amb elles.

7.1.1. Funció d' adquisició de la temperatura dels motors

Com bé és sabut, els sensors de temperatura dels 4 motors que formen el quadricòpter funcionen mitjançant l' ADC del microcontrolador. Així doncs, segons la Figura 6-9, aquests es troben als pins 8, 9, 10 i 11, els quals estan referenciats a les entrades analògiques 12, 10, 8 i 9 respectivament, les quals tenen el convertidor analògic – digital.

Per tal d' adquirir la dada i convertir-la, s' ha de seleccionar el canal corresponent al ADC del qual es vol obtenir la temperatura. S' introdueix un petit temps d' espera ja que la selecció del canal no és immediata. Posteriorment, s' ha de fer la conversió de la lectura a temperatura en graus Celsius.

Per tal de fer la correcta conversió, s' ha de saber que l' ADC està configurat a 10 bits, això fa que el valor màxim que obtindrà el convertidor és de 1024, que es correspon quan l' entrada és de 3,3V. Finalment, l' equació resultant és:

$$Temperatura = \frac{ADC}{1024} \cdot 3,3 \cdot 100 \quad (\text{Eq. 7.1})$$

On el 100 fa referència als 100 °C que equivalen a 1V, ja que al *Datasheet* del fabricant es pot observar que la variació de la sortida és de 10mV/°C. Amb tot això, la funció d' adquisició de temperatura queda tal i com es pot veure a la Figura 7-3.

```
void get_temp()
{
    //Temperatura motor 1
    set_adc_channel(12);
    delay_us(20);
    temp1=((Read_adc()/1024)*3.3)*100;
    delay_us(20);
    //Temperatura motor 2
    set_adc_channel(10);
    delay_us(20);
    temp2=((Read_adc()/1024)*3.3)*100;
    delay_us(20);
    //Temperatura motor 3
    set_adc_channel(8);
    delay_us(20);
    temp3=((Read_adc()/1024)*3.3)*100;
    delay_us(20);
    //Temperatura motor 4
    set_adc_channel(9);
    delay_us(20);
    temp4=((Read_adc()/1024)*3.3)*100;
    delay_us(20);
}
```

Figura 7-3. Funció d' adquisició de temperatura dels motors

7.1.2. Funció d' adquisició de dades del GPS

El GPS és un component complex de programar, el qual té un *driver* que s' importa al fitxer principal. En aquest *driver* és on s' agafen totes les dades i es creen funcions per tal de poder consultar-les en qualsevol moment des del fitxer principal.

Tot i que el *driver* que proporciona la companyia del GPS és més genèric, aquest s' ha adequat per a aquest projecte, únicament amb el necessari i modificant alguna línia de la recepció per UART ja que en aquest cas comptem amb dos dispositius que funcionen pel mateix protocol, de manera que s' ha de concretar.

Malgrat poder obtenir alguna variable que no interessa en aquest projecte, com ara la hora (que ja la proporciona el rellotge per raons esmentades anteriorment) i la orientació, aquestes no s' han inclòs en la funció que relaciona el *driver* amb les variables del fitxer principal.

A la es pot veure com aquesta funció és molt senzilla, ja que únicament se li assigna a les variables corresponents la funció que s' encarrega de retornar el valor obtingut en l' altre fitxer *driver*.

```
void get_gps()
{
    lat = Latitude();
    lon = Longitude();
    altGPS = Altitude();
    sat = Satellites();
    spd = Speed();
}
```

Figura 7-4. Funció d' adquisició de dades del GPS

7.1.3. Funció d' adquisició de dades del BME280

Les dades ambientals referents com són la temperatura ambient, la humitat relativa i la pressió, s' aconsegueixen mitjançant el BME280. En aquest cas es compta amb una llibreria que segueix els passos per a aconseguir una bona adquisició referits al *datasheet*. Aquest sensor es pot configurar de diferents formes mitjançant *firmware*, s' ha escollit el mode normal, el quan té un ràtio d' adquisició de dades mitjà, sense portar-lo al extrem ni fer que el seu procés sigui massa lent.

S' ha creat una funció que executa els diferents mètodes de la llibreria i tracta les dades rebudes per a que s' adequin a les unitats necessitades.

```

void get_bme()
{
  BME280_readTemperature(&Tdec);
  BME280_readPressure(&Pdec);
  BME280_readHumidity(&Humdec);
  T_Cent = Tdec/100 + (float)((Tdec%100)/100);
  P_mBar = Pdec/100 + (float)((Pdec%100)/100);
  H_rel = Humdec/1024 + (float)((((Humdec * 100)/1024) % 100)/100);
  altBMP = BMP280Altitude(P_mBar);
}

```

Figura 7-5. Funció d'adquisició de dades ambientals del BME280

A la Figura 7-5 es pot observar com es llegeixen les dades i s'emmagatzemen a la variable corresponent. Posteriorment, la temperatura i la pressió es divideixen entre 100 ja que el sensor dóna el valor en centenars i, la humitat relativa s'ha de dividir entre 1024.

Posteriorment, es calcula l'altitud a partir de la pressió obtinguda, sabent la pressió superficial del lloc on es duran a terme les proves, sent aquesta de 1012,7 hPa, calculada a partir de la fórmula que es veu a la Figura 7-6.

```

float BMP280Altitude(float p)
{
  altitudbme = 44330*(1-pow(p/1012.7,1/5.255));
  return(altitudbme);
}

```

Figura 7-6. Càlcul de l'altura a partir de la pressió

7.1.4. Funció d'emmagatzemar les dades a la 24LC256

Per tal d'emmagatzemar dades a la memòria EEPROM externa, s'ha de seguir una sèrie de passos relacionats amb el protocol del bus I2C el qual es comentarà més endavant. Per tal de poder accedir a ella i escriure, el seu *datasheet* ens proporciona el procediment a seguir, tal i com es pot veure a la Figura 7-7.

FIGURE 6-1: BYTE WRITE

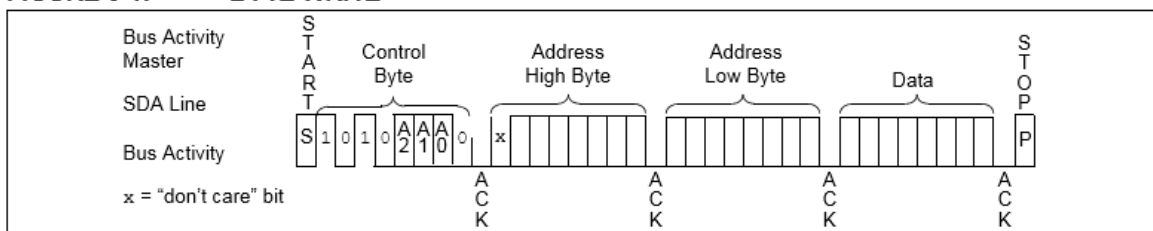


Figura 7-7. Escripció a la 24LC256 (Font [7]: www.microchip.com)

Seguint amb el procediment proporcionat pel fabricant, s'ha creat un *driver* que conté la funció d'escriptura per a executar-la diverses vegades al programa principal. Aquesta funció es pot veure a la Figura 7-8.

```
void write_ext_eeprom(long int address, byte data){
    short int status;
    i2c_start(); //Inici comunicació escriptura
    i2c_write(0xa0); //Paraula de control escriptura
    i2c_write(address>>8); //Part alta direcció
    i2c_write(address); //Part baixa direcció
    i2c_write(data); //Dada a escriure
    i2c_stop(); //Final comunicació escriptura
    i2c_start(); //Reiniciem
    status=i2c_write(0xa0); //Lectura del bit ACK

    while (status == 1) //Si és 1 es segueix esperant
    {
        i2c_start();
        status=i2c_write(0xa0);
    }
    i2c_stop(); //Si és 0, finalitza la comunicació
}
```

Figura 7-8. Funció d'emmagatzematge a la EEPROM

Comparant la Figura 7-7 amb la Figura 7-8, es pot veure com es segueix el mateix patró. La funció *write_ext_eeprom* rep dos paràmetres els quals són l'adreça a on s'ha d'emmagatzemar i la dada a escriure. Un cop rebuts s'inicia la comunicació I2C i s'escriu A0, que es correspon a l'escriptura tal i com s'ha comentat anteriorment. Posteriorment s'escriu la part alta de la direcció rebuda i a continuació la part baixa, tal i com demana el fabricant. Finalment s'envia la dada a emmagatzemar i s'atura la comunicació.

A posteriori, es comprova el bit d'ACK, explicat en l'apartat de Comunicació I2C, per tal de comprovar si el receptor ha rebut les dades. En cas contrari s'espera fins que aquest retorna el bit de forma correcte, en aquest cas com a 0 binari.

En el cas que la dada sigui un valor decimal, s'ha d'emprar un altre mètode aprofitant però la funció ja escrita. En aquest cas, la dada ocuparà 4 posicions de memòria. Per tal d'aconseguir-ho s'agafa la dada que es rep i es converteix a *int8*, de tal manera que es va trossejant la dada per tal d'encabir-la en 4 posicions de memòria, aplicat tal i com està a la Figura 7-9.

```
WRITE_FLOAT_EXT_EEPROM(long int n, float data) {
    int i;

    for (i = 0; i < 4; i++)
        write_ext_eeprom(i + n, *((int8 *)&data + i));
}
```

Figura 7-9. Funció d'emmagatzematge a la EEPROM una variable decimal

Un cop les dos funcions estan implementades als *drivers*, aquests s' utilitzen al programa principal per a poder escriure a la memòria. S' ha creat una funció la qual escriu totes les dades adquirides a la EEPROM. A la Figura 7-10 es veu una petita mostra on s' emmagatzema una variable *integer* i una variable *float*. En funció del tipus d' aquestes l' adreça on s' ha d' emmagatzemar la següent variable ha d' augmentar 1 o 4 respectivament, ja que en el cas del *float*, tal i com s' acaba de veure, ocupa 4 posicions de memòria.

```
write_ext_eeprom(address,sec);
address = address+1;
write_float_ext_eeprom(address,T_Cent);
address = address+4;
```

Figura 7-10. Fragment d' exemple d' emmagatzematge d' un *integer* i un *float*

7.1.5. Funció d' enviament de dades al receptor

Per tal d' enviar les dades mitjançant el transceptor, s' ha creat una funció per tal de disminuir les línies de codi al bucle principal. Aquesta funció s' encarrega d' enviar les dades totes seguides, cadascuna amb el format que interessa, per tal de descompondre tota la cadena a l' estació receptora.

Mitjançant la funció *fprintf* es pot enviar cada dada segons el tipus de variable que sigui i, en el cas dels *float*, es pot especificar el nombre de decimals. Un cop s' ha valorat el nombre de decimals de cada variable i l' ordre en què s' envien, la funció resultant es la que s' observa a la Figura 7-11.

```
void send_data()
{
    fprintf(BT,"%u,",day);
    fprintf(BT,"%u,",mth);
    fprintf(BT,"%u,",year);
    fprintf(BT,"%u,",hr);
    fprintf(BT,"%u,",min);
    fprintf(BT,"%u,",sec);
    fprintf(BT,"%0.1f",T_Cent);
    fprintf(BT,"%0.1f",P_mBar);
    fprintf(BT,"%0.1f",altBMP);
    fprintf(BT,"%0.1f",H_rel);
    fprintf(BT,"%0.6f",lat);
    fprintf(BT,"%0.6f",lon);
    fprintf(BT,"%0.1f",altGPS);
    fprintf(BT,"%d",sat);
    fprintf(BT,"%d",temp1);
    fprintf(BT,"%d",temp2);
    fprintf(BT,"%d",temp3);
    fprintf(BT,"%dF \n",temp4);
}
```

Figura 7-11. Funció d' enviament de dades

La terminació "F\r" de l' última línia, fa referència a que s' ha finalitzat l' enviament del paquet sencer de dades, per tal de poder interpretar-ho correctament a l' estació receptora.

7.2. Sistema de visualització de dades

Per tal de visualitzar les dades correctament, s' ha de dur a terme la programació de l' estació receptora. En aquest cas s' actua sobre 5 pantalles LCD que van connectades directament a les línies SDA i SCL del bus I2C. Aquest codi s' ha emprat mitjançant *Arduino* ja que la placa és de la pròpia empresa.

En aquest cas es tractarà la funció que s' encarrega de rebre les dades i adjudicar-les a la variable corresponent, ja que el fet de mostrar-les no necessita tant detall, és una qüestió de format sobre les pantalles LCD.

Així doncs, a l' inici del programa es declaren totes les variables que s' envien a través del radiotransmissor i s' inicia la comunicació sèrie a 9600 bauds, de la mateixa manera que es fa al microcontrolador PIC. Un cop s' ha inicialitzat, la funció que detecta quan es reben dades a través del port sèrie, on hi ha connectat el radiotransmissor, és *serialEvent()*.

Aquí dins és on s' assignaran els valors que es van rebent a les variables corresponents. Per tal de fer això es divideix en si es tracta de la última dada del paquet total o no. Quan no es correspon a la última dada, sabem que el separador és una coma (","), en cas contrari, una "F". Això ho determinem mitjançant una altre variable auxiliar anomenada *paquet*. Aquesta variable auxiliar també permet assignar el valor a les variables corresponents.

Iniciant per la primera dada, seguint el mateix ordre d' enviament, es fa l' adjudicació dels valors i s' incrementa en 1 el valor de *paquet*, per tal de saber que quan s' analitzi la pròxima dada es correspondrà a una variable diferent, i així successivament (Figura 7-12).

```

void serialEvent()
{
  while (Serial.available())
  {
    if (paquet != 18)
    {
      data = Serial.readStringUntil(',');
      Serial.read();
    }
    else
    {
      data = Serial.readStringUntil('F');
      Serial.read();
    }
    switch (paquet)
    {
      case 1:
        dia = data;
        paquet += 1;
        break;
    }
  }
}

```

Figura 7-12. Primera part de la funció receptora de dades

Quan s' arriba a la última dada rebuda (la número 18), es llegeix aquesta, s' indica que s' ha completat la lectura de totes les dades rebudes i es torna a reiniciar el comptador *paquet* a 1 (Figura 7-13).

```

      case 18:
        temp4 = data;
        altura = (altBMP.toDouble() + altGPS.toDouble()) / 2;
        stringComplete = true;
        paquet = 1;
        break;
    }
  }
}

```

Figura 7-13. Part final de la funció receptora de dades

A la Figura 7-13 també es pot veure com una línia de codi fa referència a l' altura. Això és degut a que quan s' ha completat tota l' adquisició de dades, s' obtenen dos valors d' altura, un subministrat pel BME280 mitjançant el càlcul amb la pressió, i un altre provinent del mòdul GPS. Com que tots dos poden tenir petites variacions, es fa una mitjana aritmètica per a intentar corregir les desviacions.

8. Simulacions

Prèviament al disseny de la PCB i del prototip de proves, s' han realitzant una sèrie de simulacions per tal de verificar el correcte funcionament del sistema complet i comprovar que no hi ha errors al disseny de connexió abans de muntar la placa de proves.

Quan es fa una simulació d' aquest estil, es comprova alhora el *hardware* i el *firmware*, ja que per tal de que el sistema funcioni correctament, ambdós han de combinar-se d' una forma adequada. En aquestes simulacions s' han tingut en compte diversos aspectes que s' han anat verificant a mesura que es simulava. Aquests aspectes que s' havien de superar són els següents:

- Verificació del bus I2C.
- Verificació de dos ports sèrie de forma simultània.
- Correcte adquisició de les dades.
- Correcte emmagatzematge de les dades.
- Correcte enviament i recepció de les dades.
- Respostes en funció de les dades obtingudes i processos.

Per tal de dur a terme aquestes simulacions, s' ha utilitzat el *Proteus*. Aquesta eina permet simular dissenys electrònics amb diferents microcontroladors amb una similitud a la realitat força precisa (en funció de la CPU del PC, l' execució pot ser més ràpida o més lenta). Aquest *software* també accepta el disseny de modelats creats per l' usuari en cas que algun componen específic no estigui incorporat a la llibreria del programa.

Un cop superades les fases de verificació, es va dissenyar el *firmware* final que incorpora tot el necessari per al sistema d' adquisició de dades. És aquí on totes les fases de simulació han de funcionar correctament en una mateixa simulació del sistema sencer. Cal comentar que en aquest cas, per tal de poder simular la recepció de dades, hi ha dos variants respecte el disseny del prototip. El sensor BME280 no existeix al *Proteus*, és per això que s' ha utilitzat el BMP180, malgrat no tenir l' adquisició d' humitat, era el més fidel a la realitat. Per altra banda, els sensors del motor són el TMP-36 enlloc del TMP-35. En aquest cas només varia molt poc la fórmula de conversió, ja que la seva única diferència és el rang de temperatures.

Tot i que en simulació es podria representar d' una forma molt més gràfica totes les dades obtingudes, s' ha volgut fer la interpretació de les dades de la mateixa manera que al prototip final, d' aquesta manera les variacions de *firmware* eren mínimes i s' aconseguia minimitzar l' error. Així doncs, per tal d' entendre i validar la simulació, el més important és saber com es comuniquen les dos estacions, per tal d' interpretar la cadena que s' envia per a mostrar per pantalles posteriorment a l' estació receptora.

Aquesta cadena d'informació separa les dades mitjançant comes i, a l'última dada, s'hi adhiereix una "F" i un canvi de línia tal i com s'ha comentat anteriorment. Un cop sabent això, l'ordre determinat és el següent: dia, mes, any, hora, minut, segon, temperatura (ambient), pressió, altura (a partir de la pressió), humitat, latitud, longitud, altura (a partir del GPS), satèl·lits, temperatura motor 1, temperatura motor 2, temperatura motor 3, temperatura motor 4.

Un cop és conegut això, es fa la simulació total del sistema en la qual, per a comprovar que les dades s'han enregistrat correctament a la memòria, aquestes també s'enviaran amb l'única finalitat de poder llegir el que hi ha a la memòria.

8.1. Alimentació, oscil·lador i reinici

Una primera part més senzilla, per ser el màxim fidel possible a la realitat, és la de l'alimentació. De la mateixa manera que al prototip final, no s'ha optat per a posar l'alimentació predeterminada del programa, que també funcionaria, sinó que s'ha simulat una bateria de 3,7V i mitjançant un regulador de tensió s'obtenen els 3,3V que alimenten tot el circuit.

L'oscil·lador de cristall s'ha connectat de la mateixa manera i finalment també es fa la prova del circuit de reinici, que quan aquest està actiu (*Jumper* connectat) el microcontrolador s'atura, i quan es desconnecta el microcontrolador comença el bucle principal des del principi, anul·lant tot l'anterior. En el cas que no es toqui el *Jumper*, el programa segueix sense problemes.

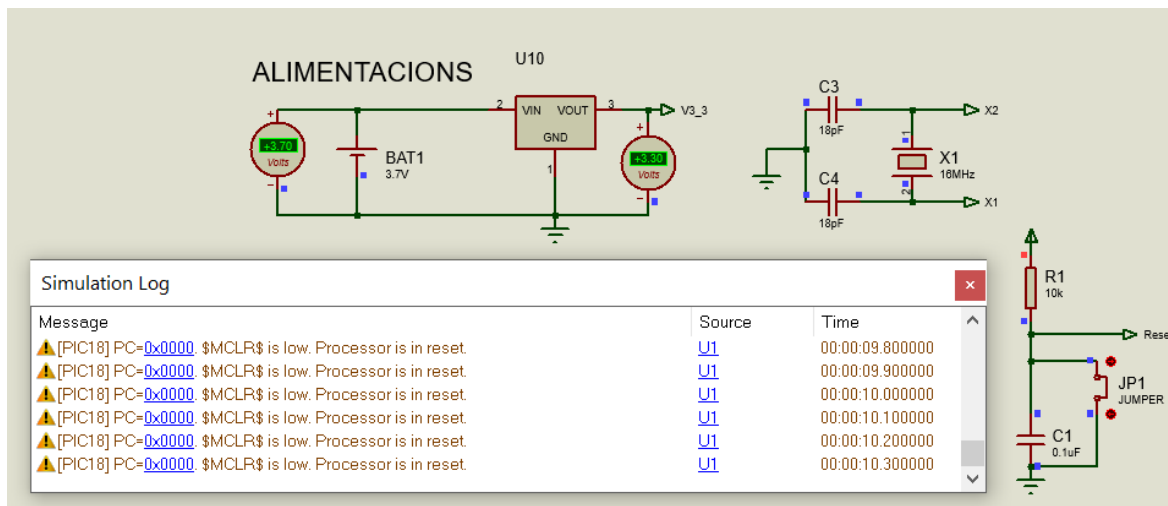


Figura 8-1. Simulació de l'alimentació i el reinici.

A la Figura 8-1 es pot observar com, efectivament, amb el regulador de tensió s'obté la sortida de 3,3V que alimentarà tot el circuit i, en aquest cas, el *Jumper* es troba connectat de manera que el *Log* de la simulació ens mostra que el microcontrolador es troba en l'estat de reinici de forma contínua.

8.2. Adquisició de les dades

Una de les parts essencials del programa és la correcta adquisició de les dades, ja que és la primera funcionalitat del sistema, adquirir les dades del quadricòpter durant el seu vol. Per tal de certificar que s' obtenen les dades de la forma esperada, primer es presentarà els diferents valors a adquirir per a comparar-los posteriorment amb la cadena generada.

Aquestes dades s' obtenen a partir de diferents protocols de comunicació, concretament mitjançant el bus I2C, el port sèrie i la conversió mitjançant els ADCs.

8.2.1. Bloc I2C

En aquest bus hi tenim connectats el sensor de mesures ambientals (BMP180 en simulació, BME280 en prototip), el rellotge de temps real i la memòria EEPROM. Així doncs, per a l' adquisició de dades la memòria no entra en joc, ja que la seva lectura es farà a posteriori.

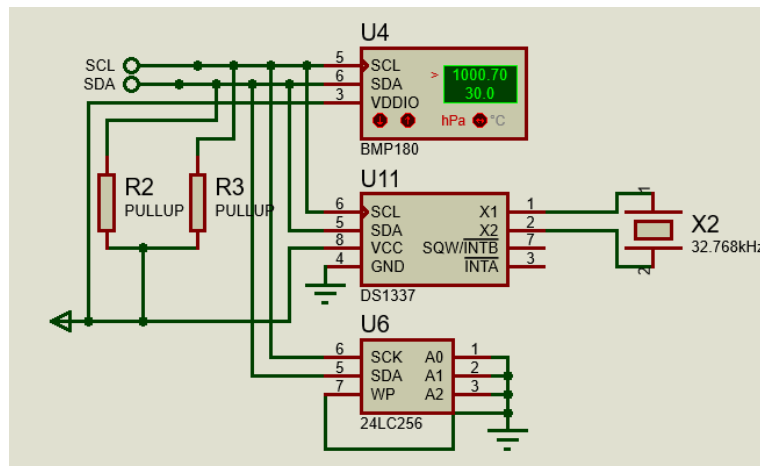


Figura 8-2. Simulació I2C.

A la Figura 8-2 es pot veure el bloc de simulació referent al protocol de comunicació I2C. A la part superior s' observa com el valor de temperatura ambient és de 30°C i la pressió de 1000,7 hPa. Aquestes són les dues primeres dades que s' hauran de contrastar amb la cadena que s' envia a l' estació receptora.

Per altra banda, també hi ha el rellotge de temps real. És evident que en aquest cas s' ha d' inicialitzar en una hora, la qual es fa la primera vegada que es programa el microcontrolador mitjançant el *firmware*, ja que posteriorment s' actualitzarà automàticament gràcies a l' alimentació. Així doncs, l' hora introduïda per a simular es pot veure a la Figura 8-3.

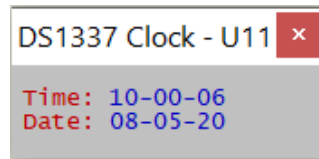


Figura 8-3. Data i hora del rellotge DS1337.

8.2.2. Port sèrie

Com bé és sabut, aquest microcontrolador disposa de dos ports sèrie, tot i que en l'adquisició de dades només un en forma part, ja que l'altre és l'encarregat de transmetre les dades mitjançant els mòduls HC-12.

En aquest cas ens centrarem en el port sèrie que capta dades, ja que està connectat al GPS. La configuració del GPS amb les coordenades simulades es pot veure a la Figura 8-4.

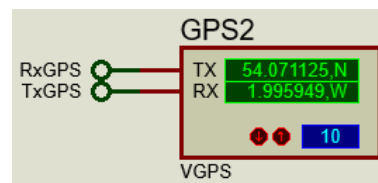


Figura 8-4. Coordenades i satèl·lits del GPS.

Malgrat a la Figura 8-4 no es veu, a les seves propietats també es poden configurar altres factors com és ara l'altura, la qual s'ha establert en 100 metres, per tal de quadrar-ho amb la obtinguda mitjançant la pressió. Cal dir que s'envien les dos dades d'altura per a tenir més precisió, ja que es farà una tria d'aquestes a l'estació receptora.

8.2.3. ADCs

Finalment, en l'adquisició de dades, també s'han de tenir en compte les diferents entrades ADC del microcontrolador, on hi van a parar les temperatures corresponents als 4 motors del quadricòpter. Els valors establerts en aquest cas són consecutius, basats en l'ordre d'aparició en la cadena que s'envia, per tal de facilitar la interpretació de les dades a simple vista. Tal i com es veu a la Figura 8-5, s'han escollit els valors de 26, 27, 28 i 29°C.

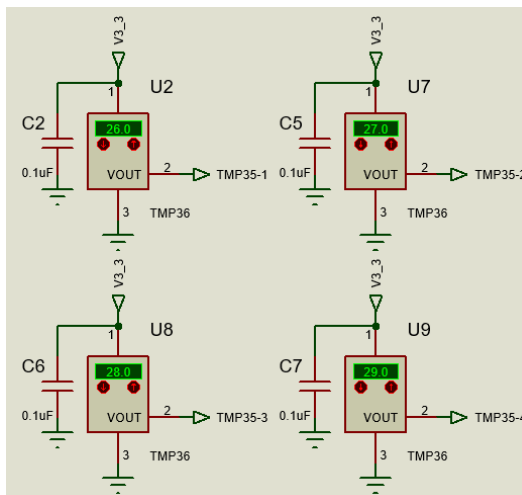


Figura 8-5. Sensors de temperatura dels 4 motors.

Un cop s' han mostrat tots els valors que s' han d' adquirir per a simular, aquests es mostren per pantalla escrivint el missatge que s' enviaria a l' estació receptora, comprovant un a un que els valors coincideixin. Aquest missatge és el que es desxifrarà a l' estació receptora excepte la part que posa "Dades Enviades", de la Figura 8-6, que serveix per a diferenciar aquestes dades de les emmagatzemades.

DADES ENVIADES: 8,5,20,10,0,6,30.0,1000.7,100.1,54.071124,-1.995948,100.0,10,26,27,28,29F

Figura 8-6. Cadena de dades a enviar.

Tal i com es veu a la Figura 8-6, els valors enviats, separats per comes i en l' ordre esmentat en la introducció de l' apartat de simulacions, coincideixen exactament amb els valors que s' han simulat mitjançant *Proteus* i que s' acaben de comentar. Així doncs, es dona per satisfactòria la part de la simulació de l' adquisició de dades.

8.3. Emmagatzematge de les dades

Al llarg de la iteració principal del programa que es carrega al microcontrolador, hi ha un emmagatzematge de les dades periòdic cada dos segons. Per tal de poder verificar que aquest emmagatzematge s' està fent de forma correcte, s' ha creat una funció especial per a llegir les dades de la memòria i mostrar-les pel port sèrie, la qual cosa en el prototip final no es farà ja que afectaria a la recepció d' aquestes. Així doncs, el funcionament per a simular i comprovar l' emmagatzematge, és que les dades s' envien cada segon i s' emmagatzemen cada dos (tal i com es produeix al prototip final),

però s' ha afegit que, quan s' emmagatzemen, es mostra per pantalla una lectura de 35 posicions de la memòria, ja que s' emmagatzemen 35 dades en simulació (39 al prototip final, ja que apareix la humitat).

```

DADES ENVIADES: 8,5,20,10,0,6,30.0,1000.7,100.1,54.071124,-1.995948,100.0,10,26,27,28,29F
DADES ENVIADES: 8,5,20,10,0,7,30.0,1000.7,100.1,54.071124,-1.995948,100.0,10,26,27,28,29F
DADES MEMORIA: 8,5,20,10,0,7,30.0,1000.7,100.1,54.071124,-1.995948,100.0,10,26,27,28,29F
DADES ENVIADES: 8,5,20,10,0,9,30.0,1000.7,100.1,54.071124,-1.995948,100.0,10,26,27,28,29F
DADES ENVIADES: 8,5,20,10,0,10,30.0,1000.7,100.1,54.071124,-1.995948,100.0,10,26,27,28,29F
DADES MEMORIA: 8,5,20,10,0,10,30.0,1000.7,100.1,54.071124,-1.995948,100.0,10,26,27,28,29F

```

Figura 8-7. Cadena de dades enviades i emmagatzemades.

A la Figura 8-7 es mostra la seqüència, d' enviament de dades cada segon i, cada dos vegades que s' envia (2 segons), s' emmagatzemen les dades. Es pot veure com la fila de "Dades Memòria" coincideix sempre amb la que es troba just a sobre seu, ja que són les últimes dades que s' han emmagatzemat. També es pot comprovar el funcionament al llarg del temps observant la columna marcada en vermell, la qual correspon a l' adquisició dels segons del DS1337.

8.3.1. Resposta als diferents processos

A l' apartat de *Hardware* es comenta que s' incorporen LEDs per a poder veure a simple vista si el funcionament de la placa d' adquisició de dades és l' esperat. En aquests LEDs se'ls atorga la funcionalitat de determinar si el GPS està connectant correctament amb els satèl·lits, de manera que el programa continua, i si els mòduls s' estan transferint dades, de manera que no hi hagi problemes de comunicació si no s' actualitzen les dades a l' estació receptora.

Quan el GPS encara no ha adquirit dades, aquest actua com si estigués rebent 0, de manera que és la dada que s' envia i s' emmagatzema, per molt que sigui errònia. Així doncs, és molt important saber quan el GPS ha adquirit el satèl·lits per tal de poder començar el vol sense perill a que la posició del quadricòpter sigui errònia, és per això que un LED s' il·lumina quan les dades de posició no són 0, de manera que vol dir que el GPS ha adquirit els satèl·lits necessaris. Per simulació també, mitjançant el *Proteus*, es pot configurar un temps d' adquisició de satèl·lits, per tal de poder simular correctament aquests casos (Figura 8-8).

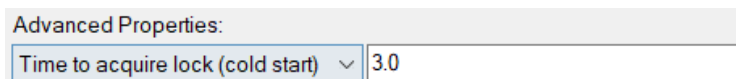


Figura 8-8. Temps d' adquisició dels satèl·lits del GPS a *Proteus*.

Un cop s' ha configurat tot, a la Figura 8-9 es veu com les dades del GPS són nul·les degut a que no han transcorregut els 3 segons i, en conseqüència, el LED està apagat.

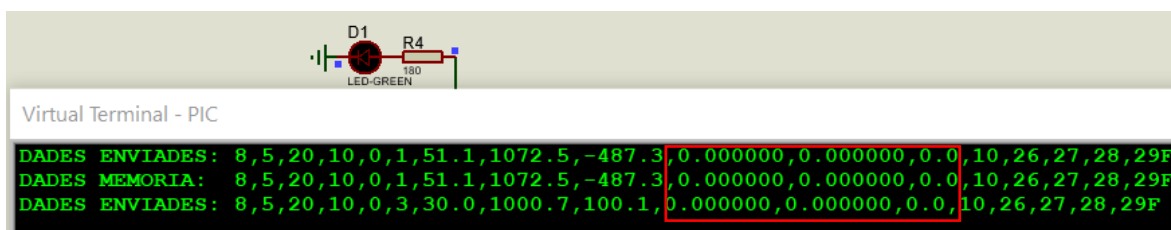


Figura 8-9. LED apagat per falta de comunicació del GPS amb els satèl·lits.

Si el temps de simulació continua avançant, un cop han passat els 3 primers segons representa que el GPS ja ha localitzat els satèl·lits i és capaç de posicionar-se, fent així que s'activi el LED per a donar l'avis a l'usuari de la correcta comunicació, resultat que es pot veure a la Figura 8-10.

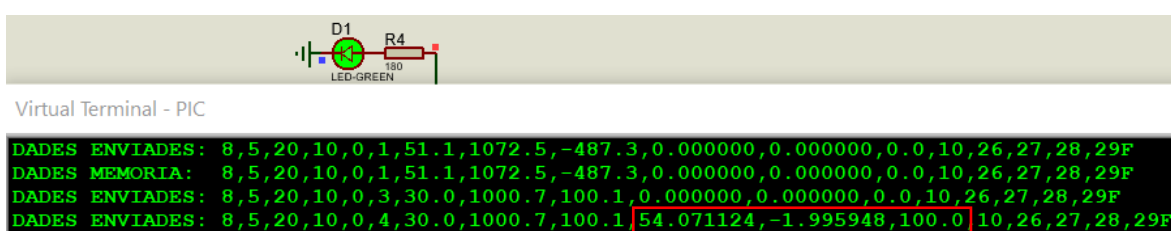


Figura 8-10. LED encès per correcta comunicació del GPS amb els satèl·lits.

Per altra banda, també es vol notificar a l'usuari quan hi ha comunicació entre les dos estacions. Per a fer això, s'ha programat un LED per a que faci un parpelleig molt ràpid quan s'envia la trama, de manera que es pugui observar aquesta fase del procés tal i com es veu a la Figura 8-11.

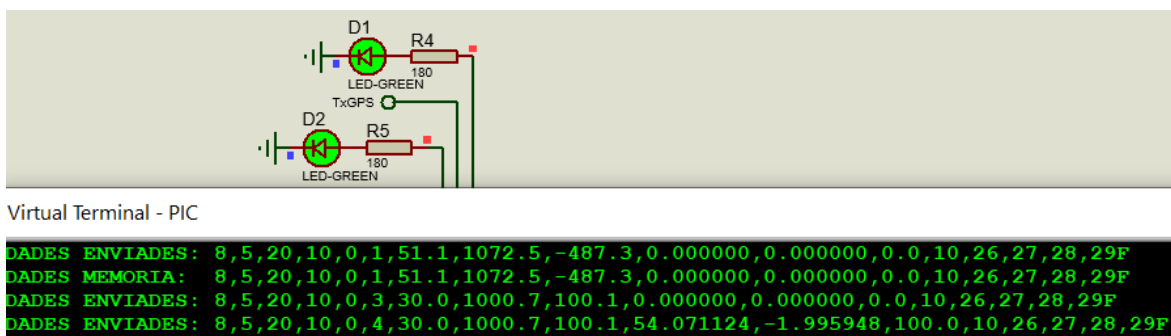


Figura 8-11. LEDs encesos per correcta comunicació del GPS amb els satèl·lits i l'enviament de dades.

8.4. Comunicació

L'apartat referent a la simulació de la comunicació entre els dos mòduls, per tal de comprovar que el que envia un és exactament el que rep l'altre, és molt senzilla. Al Proteus s'han col·locat dos dispositius HC-06 (funcionen amb el mateix protocol que l'HC-12, de manera que per a la simulació serveix

igualment), un d'ells va connectat a un port sèrie del microcontrolador (el superior de la Figura 8-12) i l'altre en aquest cas actua com a receptor, que al prototip final anirà connectat al port sèrie de l'estació receptora.

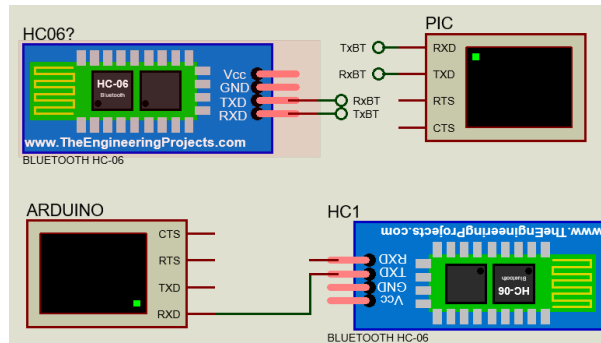


Figura 8-12. Esquema de *Proteus* dels dispositius de comunicació.

Com que en aquest cap ha d'haver-hi un emparellament virtual entre dispositius, això s'ha aconseguit amb l'ajuda d'un *software* extern, anomenat "VSPE" (*Virtual Serial Port Emulator*). Aquest *software* permet crear diversos ports virtuals amb diferents tipus de dispositiu. En aquest cas, tal i com es veu a la Figura 8-13, els ports són del tipus "Pair", simulats al COM10 i COM11. Un cop s'ha creat aquest emparellament virtual, al *Proteus* només s'ha de vincular cada dispositiu amb un dels ports simulats.

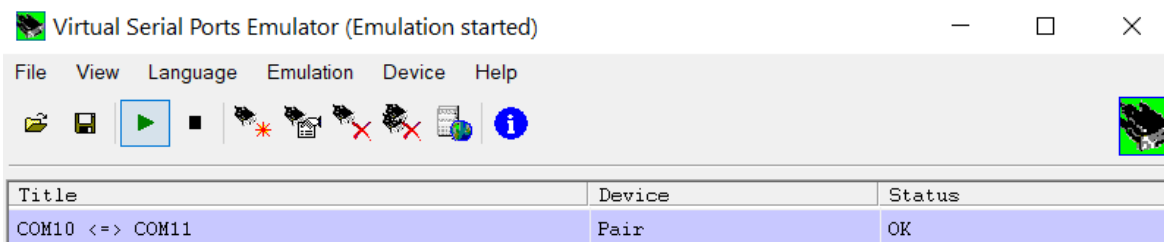


Figura 8-13. Ports virtuals emparellats mitjançant VSPE.

Un cop s'ha creat aquest emparellament virtual, al *Proteus* només s'ha de vincular cada dispositiu amb un dels ports simulats dins de les seves propietats (Figura 8-14) per a que puguin establir comunicació entre ells durant la simulació.

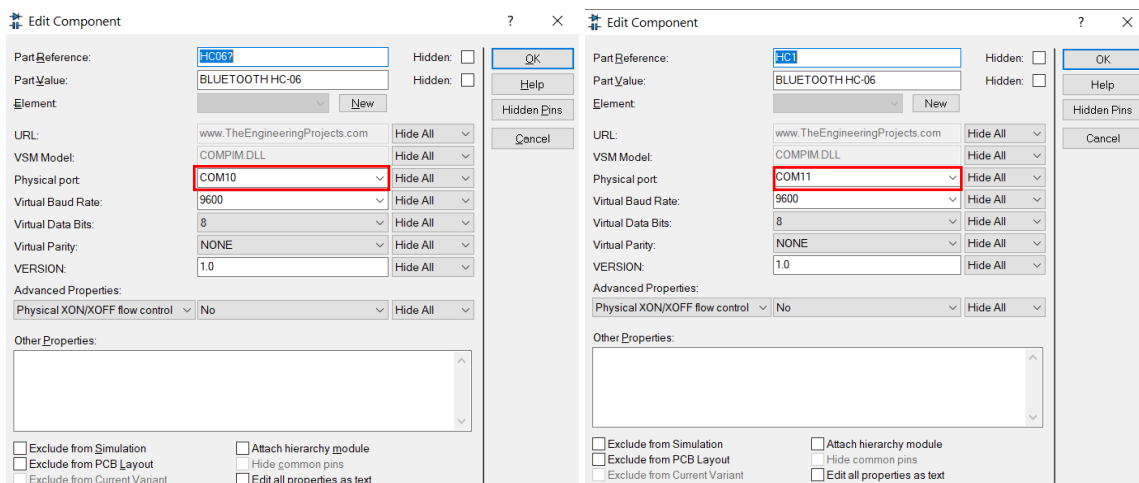


Figura 8-14. Propietats amb els ports assignats dels mòduls de comunicació de Proteus.

Per tal d' observar que la comunicació és correcta, s' han col·locat dos *Virtual Terminal* als mateixos dispositius HC-12 per a poder veure la trama que passa per cada un d' ells. Com a resultat es pot veure que en ambdós terminals es veu exactament el mateix on, el que està catalogat com a "PIC" és el referent al microcontrolador, que envia les dades, i el que s' anomena "Arduino" és el receptor, segons la Figura 8-15.

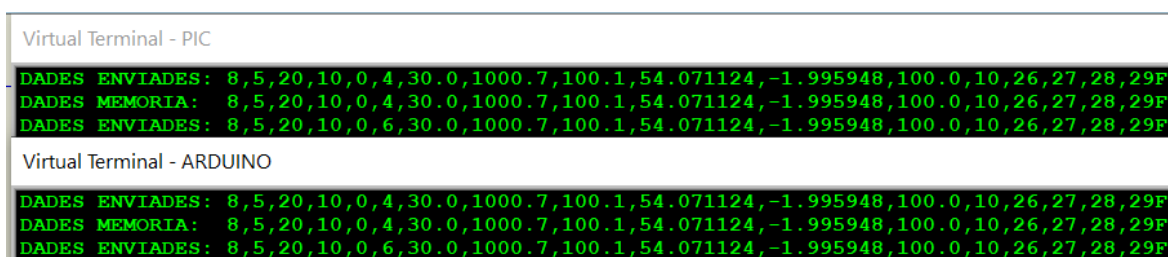


Figura 8-15. Vista de la trama enviada pel microcontrolador i rebuda al receptor.

8.5. Recepció de les dades

Finalment, es va fer una simulació de com tractar la cadena de dades a l' estació receptora. Aquesta simulació no s' ha fet mitjançant Proteus, ja que el seu muntatge físic era molt senzill i més verídica, ja que Proteus no incorpora llibreries d' Arduino. Així doncs, es diu que és una simulació perquè l' estructura de visualització no és la definitiva, i també s' ha de tenir en compte que es prova amb la mateixa cadena de dades que s' ha vist posteriorment que envia el microcontrolador, de manera que no hi apareix la humitat.

Amb aquesta prova es pretén comprovar que es té la capacitat de separar les dades i detectar el final de cadena, per tal de guardar cada dada a la variable corresponent per tal de poder mostrar-les en diferents pantalles, tal i com s' observa a la Figura 8-16.

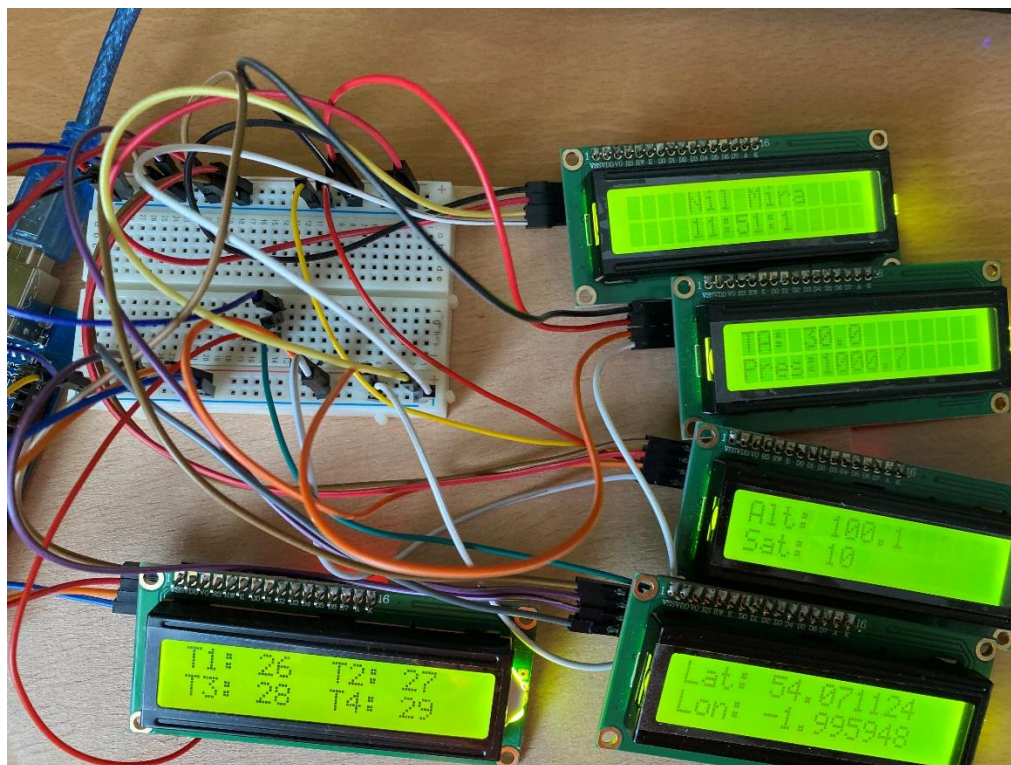


Figura 8-16. Simulació de la visualització de les dades.

9. Disseny de la PCB

Per tal de desenvolupar aquest projecte fins la final, s' ha creat una placa PCB (*Printed Circuit Board*) on fer-hi l' assemblatge de tots els components per tal de crear un prototip més professional. A l' hora de dissenyar aquest tipus de plaques, es tenen molts factors en compte, com ara el nombre de cares, gruix de les pistes, diàmetre dels pads, etc.

Al ser un prototip el qual no seria el que es comercialitzés finalment, alguns d' aquests aspectes s' han determinat de la manera més senzilla, per tal d' evitar tenir problemes amb la impressió, ja que aquesta no ha de ser el màxim refinada possible.

El disseny d' aquesta placa s' ha dut a terme mitjançant el *software Altium Designer*. Aquest programa és molt reconegut a nivell mundial, utilitzar per un gran nombre d' empreses degut a la seva professionalitat. Un cop s' ha bolcat l' esquema electrònic al *software*, el més important era la col·locació i l' encaminat dels components. A la Figura 9-1 es pot veure com ha quedat el disseny en 2D.

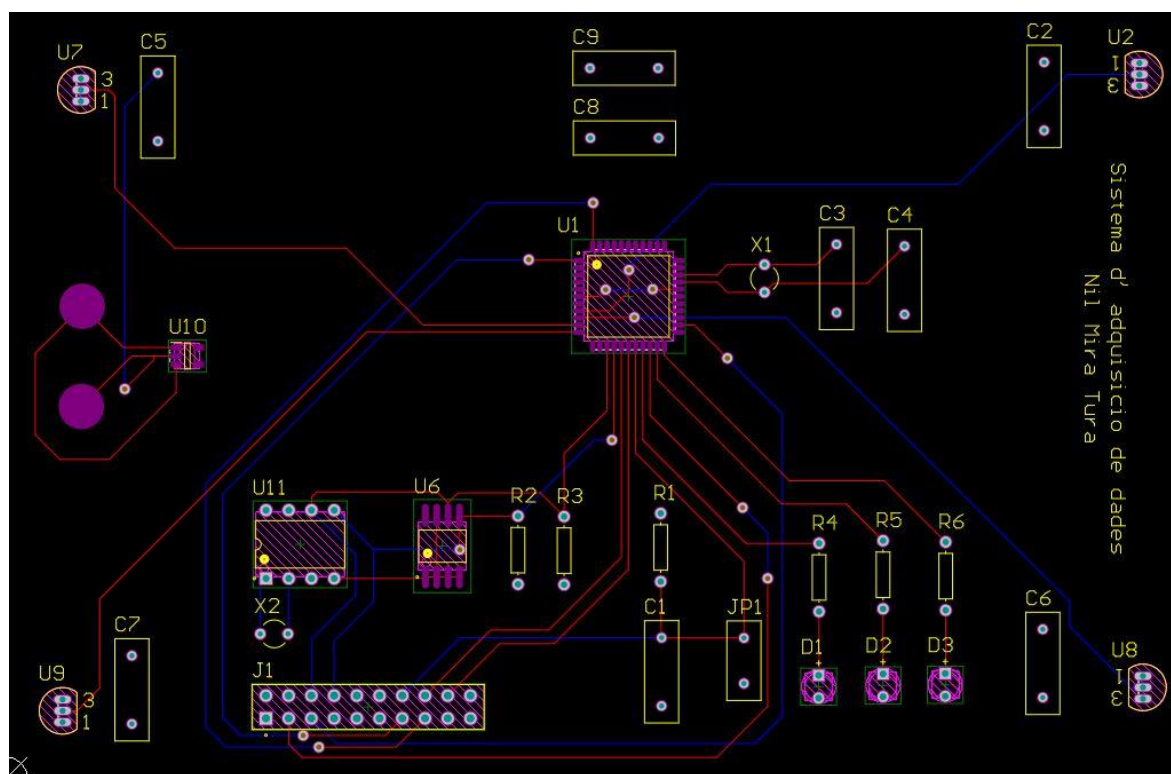


Figura 9-1. Traçat de la PCB.

Malgrat el disseny podria haver sigut molt més compacte, s' ha buscat una mínima separació entre els components per tal de facilitar l' encaminat i la posterior soldadura d' aquests, ja que al no comptar amb una màquina professional, la soldadura es durà a terme manualment. És per aquest motiu, que

també s' han evitat els components SMD com ara les resistències i condensadors, malgrat en un producte final es faria amb aquest altre tipus de soldadura.

Per tal d' establir la ubicació dels diversos components que formen la PCB, hi ha hagut una sèrie de factors que s' han tingut en compte. Primerament, s' ha volgut col·locar els 4 sensors de temperatura dels motors a les 4 cantonades de la placa, per tal de que estiguin el més pròxim possible als motors als quals han d' obtenir-hi la temperatura. El microcontrolador s' ha posat a la part central de la placa per tal de facilitar les connexions de les pistes, ja que si estigués molt escorçat, en alguns casos seria difícil accedir-hi. La resta de components s' han ubicat per proximitat al pin on es connecten al microcontrolador, agrupant-los així per busos com ara pot ser l' I2C i grups de funcionalitats, per exemple els 3 LEDs col·locats junts i en una aresta de la placa per tal de facilitar la seva visualització.

També s' ha optat per col·locar un connector del tipus femella per tal d' acoblar-hi diversos sensors que per facilitat s' han obtingut amb aquest encapsulat. En aquest connector també s' han deixat uns pins disponibles per a la programació del microcontrolador, de manera que connectant el programador als pins adequats la programació es pot fer sense haver de treure el microcontrolador de la placa.

Cal comentar que totes les connexions de massa i d' alimentació a 3,3V, s' han fet mitjançant plans d' alimentació i de massa. En el cas del pla de massa molt sovint es fa per a disminuir el soroll, però en aquest cas al tenir una alimentació global a 3,3V per a tots els components, també s' ha optat per a crear-la mitjançant un pla de massa. El pla corresponent a l' alimentació es troba a la cara superior (on també hi ha totes les connexions marcades en vermell), mentre que el pla de massa està situat a la cara inferior (juntament amb les connexions blaves). Finalment, a la part esquerra de la placa hi ha dos pads que és on es connectaria la bateria del dron per a alimentar el regulador de tensió que dona la sortida de 3,3V.

Un cop el disseny de la placa està complert, el programa ens permet verificar que totes les connexions s' han fet correctament i que es compleixen les especificacions que s' han determinat al principi, anomenades com a "rules". El mateix *software* també permet generar els fitxes de fabricació i tenir una visualització en 3 dimensions de la placa amb els components seleccionats (alguns d' ells els permet veure en 3D i d' altres només mostra el *footprint* a la PCB), la qual va molt bé per tenir una referència de com quedaria el disseny final. A la Figura 9-2 i la Figura 9-3 es poden veure respectivament els models en 3D de la cara superior i inferior.

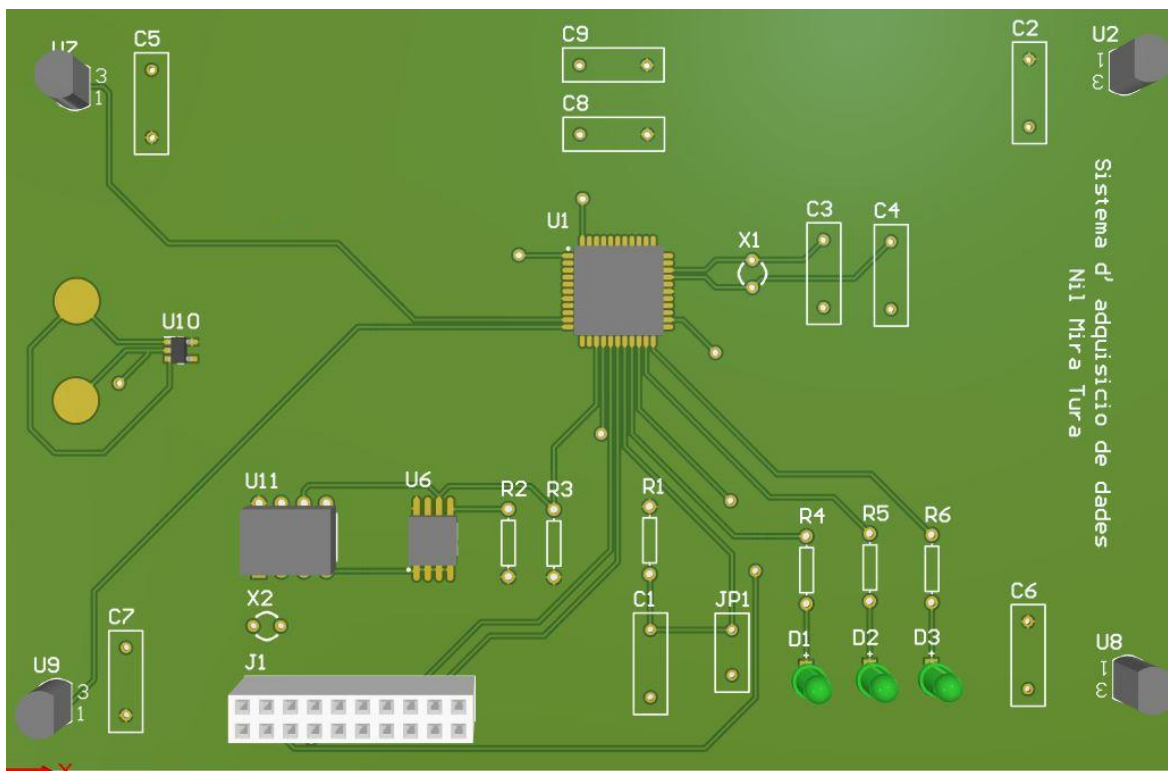


Figura 9-2. Visualització en 3D de la cara superior de la PCB.

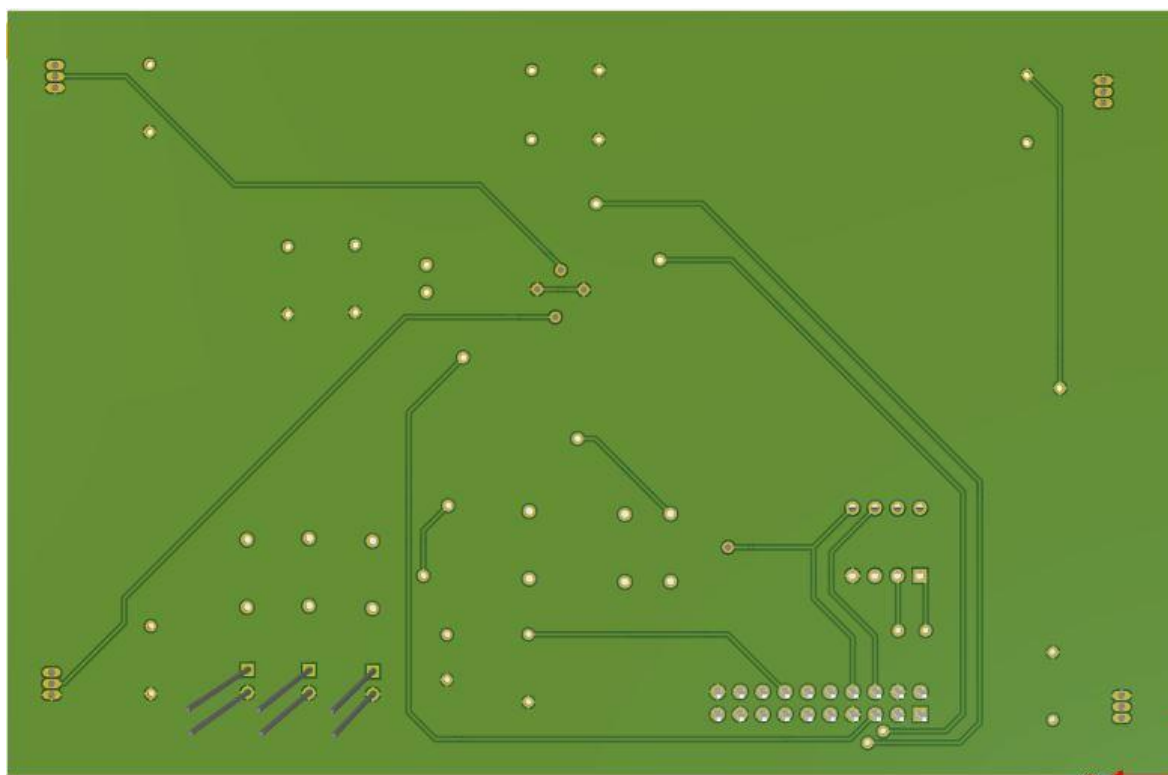


Figura 9-3. Visualització en 3D de la cara inferior de la PCB.

Per tal d' aconseguir assemblejar els components a la PCB s' han generat una sèrie d' arxius amb el propi *Altium* que especifiquen les diferents accions que s' han de dur a terme per aconseguir la PCB. Un cop s' han obtingut aquests fitxes, s' han enviat a una empresa externa perquè s' encarreguin de la impressió de la placa.

Així doncs, un cop s' han muntat tots els components a la PCB, s' obté el resultat observat a la Figura 9-4.

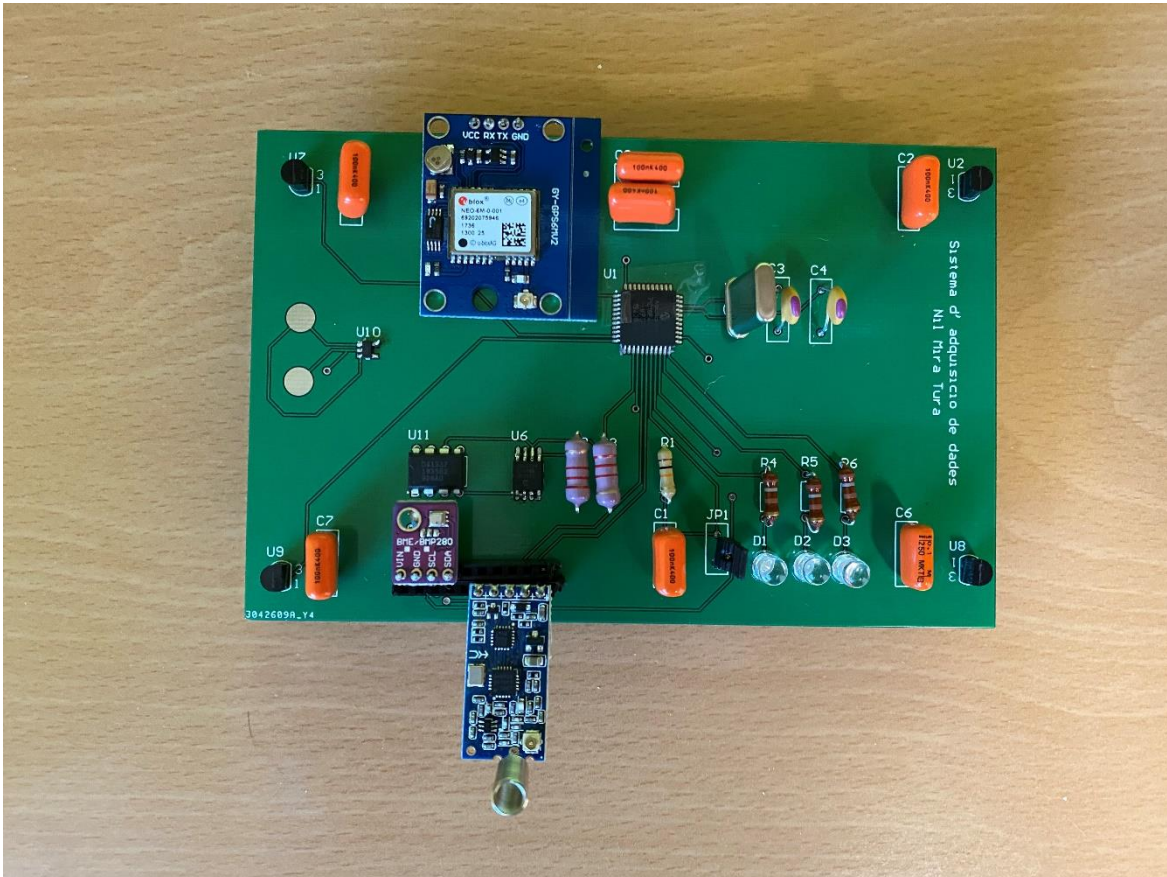


Figura 9-4. Components soldats sobre la PCB.

10. Comunicacions

Les comunicacions emprades en aquest projecte es basen totes en el mateix protocol, sent aquest del tipus sèrie. Aquestes comunicacions sèrie, com bé diu el seu nom, envien les dades bit a bit a través d' una mateixa línia. La comunicació sèrie va néixer com a alternativa a la connexió en paral·lel, la qual malgrat ser més ràpida, ofereix més problemes de sincronització ja que cada bit s' envia per una línia diferent.

Tal i com s' ha comentat, les comunicacions d' aquest projecte es basen en el protocol sèrie. Aquest protocol en molts casos pot constar majoritàriament de dos línies. La funcionalitat d' aquestes depèn del tipus de comunicació sèrie que s' estigui duent a terme. Independentment d' això, es considera que els paràmetres essencials de la comunicació sèrie són les següents:

- Velocitat de transmissió: relaciona la quantitat d' informació transmesa en una comunicació (normalment bits) amb el temps que s' ha trigat.
- Bits de dades: quantitat de bits a transmetre durant la comunicació. Aquest número depèn del tipus d' informació a transmetre.
- Bits de paritat: bits opcionals els quals permeten detectar si la comunicació s' ha realitzat de forma correcta o s' ha perdut algun bit durant la transmissió.
- Bits de sincronització: marquen el principi i el final d' un paquet de dades.

A part d' aquestes característiques relacionades amb la velocitat i els bits que componen el missatge, hi ha altres factors que determinen els diferents tipus de comunicació sèrie existent. En aquest projecte s' han utilitzat el RS-232 basat en UART pel GPS i la comunicació mitjançant radiofreqüència, i el I2C per a diversos components. Per tal d' entendre el funcionament d' aquests dos protocols, es procedirà a una explicació basada en les seves diferències.

10.1. Sincronització

Tot i que la norma RS-232 permeti la comunicació síncrona, la utilitzada en aquest cas i, la més emprada en general, és la asíncrona. Per contra, el bus I2C treballa mitjançant la comunicació síncrona.

10.1.1. Transmissió asíncrona

Quan s' estableix una transmissió asíncrona, no hi ha un rellotge comú que sincronitzi els bits per tal de tenir una correcta comunicació, cada dispositiu té el seu rellotge. La sincronització dels dispositius ve determinada per la velocitat de transmissió que s' ha establert prèviament entre ambdós dispositius

que es volen comunicar. El rellotge de cada dispositiu actua a una freqüència molt més elevada que la velocitat de transmissió per tal de poder detectar els canvis en els flancs dels diferents bits.

Hi ha certes regles de la transmissió asíncrona que vénen donades per la norma RS-232:

- Si no s'envien dades, s'envia el que s'anomena el bit de marca, el qual és un "1" binari.
- Per tal d'iniciar la transmissió, primerament s'envia el bit d'inici ("0" binari), per tal d'enviar el caràcter posteriorment.
- Els bits del caràcter s'envien de menys important (LSB) a més important (MSB).
- Un cop s'ha transmès el caràcter, s'envien opcionalment els bits de paritat i el de final, sent aquest un "1" binari.

A la Figura 10-1 es pot observar el procediment d'enviament del número 61 en hexadecimal, on l'ordre de lectura del receptor, en aquest cas una impressora, és l'invers a l'ordre de transferència.

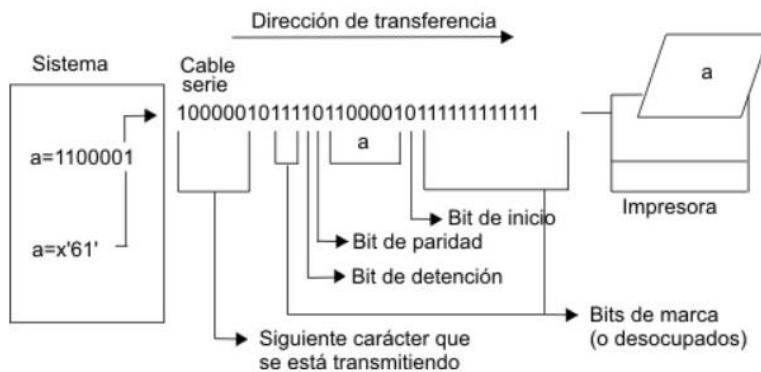


Figura 10-1. Exemple de transmissió asíncrona de dades (Font [17]: www.ibm.com).

10.1.2. Transmissió síncrona

En aquest tipus de transmissió, si que apareix una senyal de rellotge comuna entre els participants en la comunicació, hi ha una comunicació contínua. Aquesta comunicació s'utilitza per a enviar grans blocs de forma més ràpida, ja que tots els bits que s'envien són d'informació, al no haver-hi bits d'inici i d'aturada.

Per tal de sincronitzar la transmissió de dades, abans d'enviar els paquets corresponent s'envien el que es diuen bytes de sincronisme, els quals serveixen per a sincronitzar la comunicació i que aquesta es pugui iniciar. Aquesta sincronització s'ha de mantenir tot i que no s'estiguin enviant dades, de manera que es van enviant bytes de sincronisme contínuament per tal de tenir la transmissió llesta per a quan sigui necessària, tal i com es pot veure a la Figura 10-2.

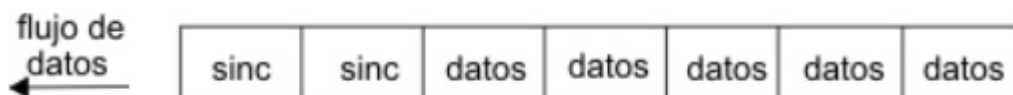


Figura 10-2. Exemple de transmissió síncrona de dades (Font [17]: www.ibm.com).

10.2. Modes de comunicació

Els sistemes de transmissió també es poden classificar segons la seva direccionalitat. En aquest cas el bus I2C té un mode de comunicació Half – Dúplex mentre que el UART del microcontrolador es pot configurar en diferents modes de comunicació, només s' utilitza el mode símplex, ja que en el cas del GPS només s' ha de rebre dades i no s' ha d' enviar cap informació cap el GPS i en el cas de la radiocomunicació tot el contrari, s' envien dades però en cap cas s' han de tenir en compte les dades que arriben, de manera que són unidireccionals. Els diferents modes de comunicació existents són:

10.2.1. Símplex

En aquest mode de comunicació només es permet la transmissió en un únic sentit, originant-se en l' emissor, que és qui envia la informació, i enviant-se al receptor (Figura 10-3). Només pot haver-hi un únic transmissor i un receptor. Com que el receptor no pot enviar cap tipus d' informació, no es poden dur a terme certs procediments de detecció i correcció d' errors.



Figura 10-3. Mode de comunicació símplex (Font [26]: www.blackbox.com.mx).

Un exemple d' aquest mode de comunicació es podria trobar en les emissores de radiodifusió convencionals, ja que s' envia la informació des de l' antena emissora fins a totes les ràdios receptores.

10.2.2. Half – Dúplex

El mode Half – Dúplex permet la transmissió bidireccional però no de forma simultània (Figura 10-4). Tant el transmissor com el receptor comparteixen la mateixa freqüència, però a través d' aquesta només hi pot circular informació en un sentit alhora.



Figura 10-4. Mode de comunicació half – dúplex (Font [26]: www.blackbox.com.mx).

Un exemple d' aquest mode de comunicació poden ser les ràdios de comunicació de l' estil *Walkie Talkie*. En aquests casos emissor i receptor no poden parlar alhora, de manera que quan un ha acabat de transmetre el missatge, l' altre pot començar a fer-ho, però mai de manera simultània.

10.2.3. Full – Dúplex

Aquest mode de comunicació és el més emprat actualment, ja que permet la comunicació bidireccional i de forma simultània (Figura 10-5). Existeixen dos freqüències per a fer-ho, una per a transmetre i l' altre per a rebre la informació.

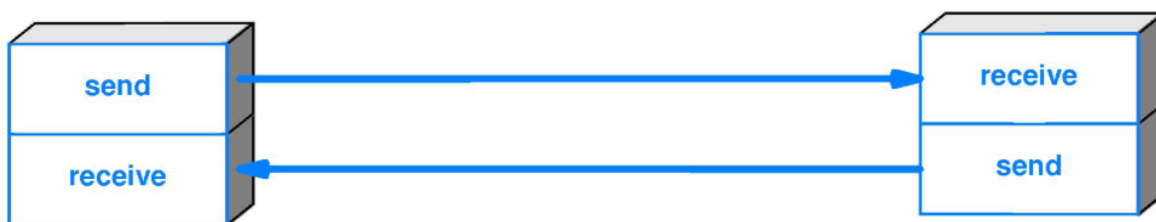


Figura 10-5. Mode de comunicació Full – dúplex (Font [26]: www.blackbox.com.mx).

Aquest mode de comunicació es pot veure molt sovint, un clar exemple en poden ser les comunicacions telefòniques, ja que ambdós persones poden parlar alhora sense necessitat d' esperar a que l' altre acabi i s' hagin d' establir torns.

10.3. Comunicació UART

En les explicacions anteriors s' ha comentat que els dos ports sèrie del microcontrolador s' utilitzen de forma asíncrona i amb el mode de comunicació símplex. Això fa que la seva comunicació sigui senzilla, ja que mitjançant *firmware* només s' han d' enviar o rebre les dades sense necessitat d' un protocol de comunicació molt elaborat.

Així doncs aquestes comunicacions disposen de dos fils, anomenades "Tx" i "Rx" els quals fan referència a "Transmit" i "Receive" respectivament. La connexió d' aquests dos fils és creuada, és a dir,

la sortida "Tx" d' un component enviarà les dades, de manera que a l' altre component, el receptor, ha de connectar-se al "Rx" per tal de rebre aquesta informació, tal i com s' observa a la Figura 10-6.

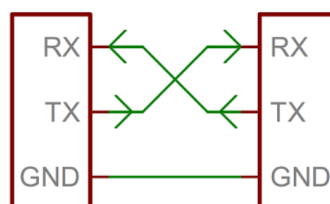


Figura 10-6. Connexió Tx i Rx del UART (Font [34]: www.docs.onion.io/omega2-docs/uart1.html).

Mitjançant la comunicació UART només es poden comunicar dos dispositius, a diferència de la I2C.

10.4. Comunicació I2C

Aquesta altra comunicació que apareix en aquest projecte és totalment diferent a l' anterior, malgrat que també segueixi un protocol sèrie. En aquest cas estem davant una comunicació síncrona i half – dúplex, seguint el protocol mestre-esclau.

Les connexions d' aquesta línia també són senzilles ja que es disposa únicament de dos cables, el SDA (*Serial Data*) i el SCL (*Serial Clock*). Com bé indica el seu nom, el SCL s' encarrega de gestionar els temps fent de rellotge mentre per SDA hi circula la informació. La connexió es pot veure a la Figura 10-7, on es pot deduir que, a diferència de la comunicació UART, en aquest cas es poden comunicar varis dispositius en un mateix parell de línies.

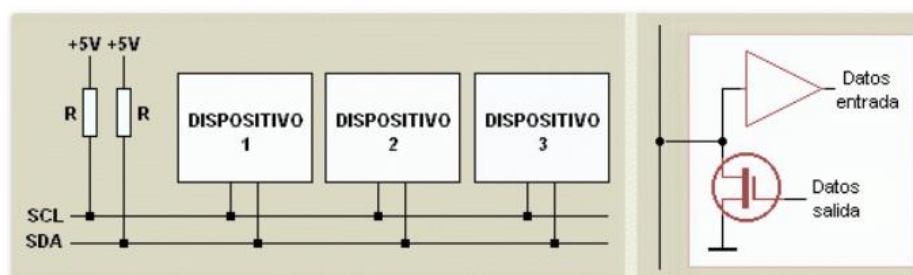


Figura 10-7. Connexió teòrica del bus I2C (Font [19]: <http://robots-argentina.com.ar/>).

Aquest bus necessita de dos resistències de pull-up amb alimentació ja que aquestes línies són de col·lector obert, de manera que per al seu correcte funcionament s' han d' incloure aquestes resistències.

L' inici de la comunicació la determina el mestre. Quan SDA i SCL es troben en un estat lògic alt, el mestre pot ocupar la línia, de manera que posarà la línia de dades en un estat de 0 binari mantenint el relloatge a l' estat alt, tal i com es veu a la **Figura 10-8**.

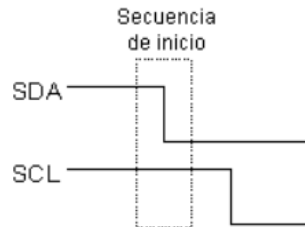


Figura 10-8. Inici de la seqüència I2C (Font [19]: <http://robots-argentina.com.ar/>).

Posteriorment, un cop s' ha iniciat la comunicació, s' ha d' enviar l' adreça (7 bits) a la qual va dirigida el missatge. Cal saber que en les comunicacions I2C, cada dispositiu té una adreça pròpia la qual ve donada pel fabricant i en alguns casos es poden modificar alguns bits en concret per a poder tenir dispositius iguals connectats a la mateixa línia. El vuitè bit del byte que compon el primer missatge un cop s' ha iniciat la comunicació fa referent a la operació que es vol realitzar sobre l' esclau, determinant així si és una lectura o una escriptura.

Si l' adreça que s' ha introduït és correcta, l' esclau respondrà amb el que s' anomena el bit ACK (*Acknowledge*). Si aquest bit és un 0 lògic, significarà que l' esclau ha reconegut la crida que li ha fet el mestre i la comunicació pot continuar. En aquest moment, depenent de si el vuitè bit (R/W) és un 0 o un 1, el mestre enviarà una dada cap a l' esclau o generarà els polsos de relloatge lliures a l' esclau per a que retorni una dada.

Finalment si es vol seguir transmetent es torna a generar una condició d' inici i en cas contrari es generarà una condició d' aturada, la qual consisteix en posar en estat alt les línies SDA i SCL (Figura 10-9).

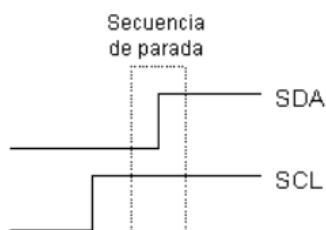


Figura 10-9. Final de la seqüència I2C (Font [19]: <http://robots-argentina.com.ar/>).

Un exemple complet del protocol de comunicació que s' ha incloure al *firmware* del microcontrolador que vol governar la comunicació mitjançant el bus I2C, seria seguint la base explicada que es pot veure reflectida en la Figura 10-10.

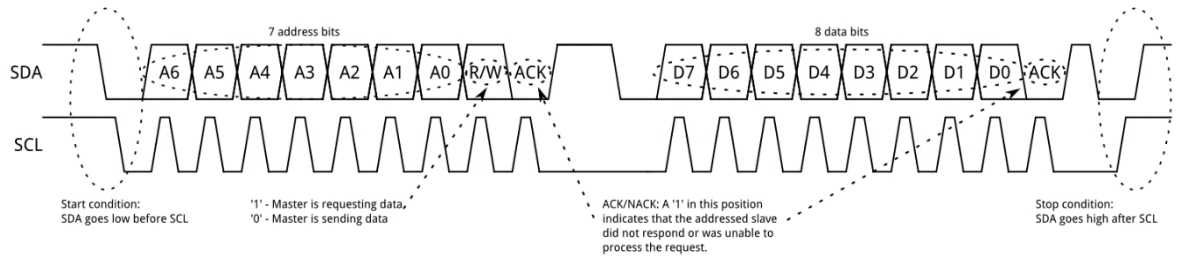


Figura 10-10. Exemple de de la seqüència de comunicació amb I2C (Font [35]: <https://learn.sparkfun.com/>).

11. Normativa

Una primera normativa a tenir en compte relacionada en l' àmbit d' aquest projecte és la legislació de drons a territori espanyol. En aquest camp apareix:

- **Reial decret 1036/2017**, de 15 de desembre, pel qual es regula la utilització civil de les aeronaus pilotades per control remot, i es modifiquen el Reial decret 552/2014, de 27 de juny, pel qual es desplega el Reglament de l'aire i les disposicions operatives comunes per als serveis i els procediments de navegació aèria, i el Reial decret 57/2002, de 18 de gener, pel qual s'aprova el Reglament de circulació aèria.

Aquest decret estableix les bases del pilotatge de drons en un entorn lúdic amb mesures com ara no superar els 120 metres d' altura, no apropar-se a menys de 8 km d' un aeroport o espai aeri controlat, el quadricòpter sempre ha d' estar a l' abast visual, ha d' estar identificat i s' ha de protegir la intimitat i la seguretat de les persones. En aquest cas concret, el dron utilitzar ha de complir la normativa de quadricòpter de pes inferior als 250 grams.

Per altra banda, també s' ha de tenir en compte diferents normatives referents als components electrònics com poden ser:

- **Directiva 2014/53/UE DEL PARLAMENT EUROPEU I DEL CONSELL** del 16 d' abril de 2014 relativa a l' harmonització de les legislacions dels Estats membres sobre la comercialització d' equips radioelèctrics, i per la que es deroga la Directiva 1999/5/CE.
- **Directiva 2014/30/UE DEL PARLAMENT EUROPEU I DEL CONSELL** del 26 de febrer de 2014 sobre l' harmonització de les legislacions dels Estats membres en matèria de compatibilitat electromagnètica (refosa).
- **Directiva 2011/65/UE DEL PARLAMENT EUROPEU I DEL CONSELL** del 8 de juny de 2011 sobre restriccions en la utilització de determinades substàncies perilloses en aparells elèctrics i electrònics (refosa).
- **Directiva 2012/19/UE DEL PARLAMENT EUROPEU I DEL CONSELL** del 4 de juliol de 2012 sobre residus d' aparells elèctrics i electrònics (RAEE) (refosa).

12. Anàlisi de l' impacte mediambiental

Durant la realització del projecte i en la seva posterior comercialització, s' ha hagut de tenir en compte l' impacte mediambiental del dispositiu dissenyat. En aquest cas, tal i com ja s' ha comentat, el prototip dissenyat és d' unes dimensions superiors a la versió que es comercialitzaria i seria més efectiva de cara al públic. Tot i això, l' impacte seria aproximadament el mateix.

S' ha assegurat que tots els components utilitzats no són nocius per al medi ambient, complint així la directiva RoHS, inclosa la PCB on s' ha muntat el prototip. Aquesta directiva permet la regulació a uns nivells definits de certes substàncies perjudicials com ara el plom, mercuri, cadmi, crom VI, PBB, PBDE i és la mencionada en l' apartat **Normativa** amb nom **Directiva 2011/65/UE**. El fet no utilitzar condensadors electrolítics també ajuda a disminuir l' impacte mediambiental.

Malgrat tots els components compleixen aquesta directiva, el més crític és la bateria. Aquesta bateria és la mateixa que utilitza el dron per a la seva propulsió, de manera que l' eficiència s' augmenta degut a que es redueix l' ús d' una segona bateria. Al tractar-se d' una bateria LiPo, si aquesta té un mal cuidat es pot inflamar i causar alguna petita explosió causant així certa contaminació i deixant anar partícules tòxic.

Tot i això, aquestes possibilitats són menors i l' alimentació de la pròpia placa depèn del tipus de bateria que s' utilitzi en el dron on la placa vagi instal·lada, fent així que l' impacte mediambiental de la placa en si dissenyada sigui mínim, ja que els seus components no són perillosos per al medi ambient.

Conclusions

Mitjançant la realització del present treball, s' ha aconseguit desenvolupar amb èxit el principal objectiu marcat, el de desenvolupar un sistema d' adquisició de dades orientat a un quadricòpter amb una petita estació receptora. Per tal de dur a terme el projecte, s' han empleat diversos coneixements adquirits durant el transcurs del grau.

En primer lloc, s' ha pogut desenvolupar un projecte que requeria una gran quantitat d' hores i tocava diversos àmbits de l' enginyeria, per tal d' aconseguir un procés complet de disseny de producte, passant per l' etapa de l' elecció dels components adequats per a abordar de la millor manera l' objectiu, fins el disseny d' un prototip amb la seva placa PCB inclosa.

S' han assolit els diversos objectius específics de programació, com el fet d' entendre el funcionament d' un microcontrolador de forma més exhaustiva, consultant tots els seus requeriments i els límits d' on es pot arribar amb el microcontrolador seleccionat. S' ha programat mitjançant C amb ús de llibreries pròpies entenent el protocol de diversos dispositius com pot ser ara el bus I2C o el guardat en la memòria EEPROM, tot això d' una manera ordenada i amb un codi senzill, fàcil d' entendre i alhora efectiu. Aquesta part de programació ha estat la més complicada sens dubte. El fet de treballar amb un microcontrolador molt específic, tot i formar part d' una família més gran, dificulta la cerca d' informació, fent així que cada pas aconseguit hagi sigut un repte, per a aconseguir ajuntar tot el que s' ha anat fent en un *firmware* final. Per tal d' aconseguir-ho s' han anat desenvolupant petits programes per parts entenent el funcionament total d' aquest per a saber d' on poden venir els errors en un programa més complet i solucionar-los.

Per altra banda també s' han desenvolupat i adquirit nous coneixements a nivell de *hardware* referents al disseny del prototip, seleccionant els components adequats per a cada cas i amb la millor manera de connectar-los possible per a millorar els seu rendiment. Dins d' aquest apartat també es valora el disseny i impressió d' una placa PCB adaptada al disseny especificat i entenent com es dissenya una placa des d' un àmbit més professional, tenint en compte molts dels factors que poden afectar en l' acabat d' aquesta i aconseguint un bon traçat de pistes.

Els fet de tractar amb diferents *softwares* que no s' havia treballat amb anterioritat, ha estat un factor molt interessant que també pot ajudar de cara al futur. A banda d' *Arduino* per a l' estació receptora, el qual ja s' havia utilitzat anteriorment i és conegut per la seva senzillesa i facilitat, els principals programes emprats han estat El *PIC C Compiler*, *Proteus* i *Altium*. El primer d' aquests ha servit per a la programació del microcontrolador i, per tant, el més senzill, tot i que no s' havia utilitzat mai; aquest permet redactar el codi que posteriorment s' inclourà al microcontrolador. Gràcies a *Proteus* s' han pogut fer les diferents simulacions que facilitaven molt la programació, ja que a cada pas que es feia es

podia veure el resultat d' aquest i comparar-lo amb l' esperant, fent que el procés de programació sigui més àgil ja que en tot moment es podia veure el comportament del microcontrolador a mesura que s' anava programant. Per últim, el *Proteus s'* ha utilitzat per a dissenyar la PCB. Malgrat hi ha programes molt més senzill, aquest és el més potent i més utilitzat a nivell professional arreu del món. És per això que s' ha volgut utilitzar aquest *software*, ja que pot servir en un futur molt llunyà i ha servit com a aprenentatge. El fet de ser un programa tan potent, també ha dificultat una mica el seu ús, tot i que només s' hagin utilitzat les funcions més bàsiques.

Per últim i no menys important, es valora molt com s' han anat superant els diversos reptes i entrebancs que han anat sorgint al llarg del treball, alguns per falta de coneixement que amb investigació s' han anat resolent, i alguns d' altres que han hagut de donar un gir en certes parts del treball adaptant-se a situacions externes com ara la pandèmia del COVID-19.

Possibles millores

Un cop finalitzat el projecte, és evident que es poden implementar certes millores les quals algunes no s' han pogut incloure per temes pressupostaris o de temps. El fet d' enfocar aquest sistema d' adquisició de dades als quadricòpters per tal de recollir informació que pot ser útil per a diversos factors, com ara el registre de temperatura al llarg d' un recorregut per la ciutat, fa que hi hagi multitud de possibles millores a incloure.

Una d' elles i possiblement la més evident, és la reducció de la dimensionalitat. Es podrien utilitzar components SMD per tal d' aconseguir reduir considerablement les dimensions de la PCB. Fent això s' aconseguiria incorporar el sistema a l' interior del quadricòpter. Lligat amb això, es podria fer que el don actués en funció de certes dades, com pot ser la temperatura dels motors. Si es tingués accés a la codificació del quadricòpter, es podria establir una connexió entre el sistema d' adquisició de dades i aquest per tal de poder reaccionar a certes mesures. Tornant a l' exemple de la temperatura dels motors, es podria disminuir l' altura si es detecta que aquests s' estan sobreescalfant.

Per altra banda es podrien afegir més sensors depenent de la orientació del comprador final, ja que es poden adquirir més dades d' interès a nivell ambiental, com pot ser la concentració de certs gasos determinats o la intensitat de llum que hi ha en aquell moment.

Pel que fa a la programació, aquesta es podria professionalitzar més afegint modes d' ús com pot ser el *Sleep Mode*, el quan permet que el microcontrolador descansi i consumeixi menys energia en certs moments determinats. Juntament amb potenciar més algunes facetes que el microcontrolador ens permet, es podria substituir l' emmagatzematge de dades, per tal de canviar la EEPROM per una targeta micro SD. Això permetria emmagatzemar moltes més dades a part d' una extracció de dades més senzilla i ràpida.

Finalment, l' estació receptora és un aspecte a millorar. En aquest cas la base del projecte estava centrada en el sistema d' adquisició de dades i l' estació permetia veure el correcte enviament i recepció d' aquestes. Es podria fer una aplicació per tal de visualitzar les dades a la computadora. Aquesta aplicació es podria dissenyar de tal forma que s' anessin graficant les dades al llarg del recorregut i que es poguessin exportar fitxers Excel per a guardar-los a posteriori. També es podrien afegir funcions a l' aplicació per tal d' actuar sobre el sistema d' adquisició de dades, modificant el temps d' adquisició o eliminant les dades de la memòria per exemple a nivell remot, incloent una connexió mitjançant un servidor web.

Bibliografia

- [1] **Castaño Giraldo, Sergio Andrés.** Control Automático Educación. [En línia] 26 / Juny / 2016. [Data: 17 / Març / 2020.] <https://controlautomaticoeducacion.com/microcontroladores-pic/comunicacion-i2c/>.
- [2] **TeslaBem.** TeslaBem. [En línia] 1 / Octubre / 2016. [Data: 29 / Gener / 2020.] <https://teslabem.com/blog/como-elegir-el-microcontrolador-correcto/>.
- [3] **Syed Zain, Nasir.** The Engineering Projects. [En línia] 29 / Gener / 2016. [Data: 15 / Febrer / 2020.] <https://www.theengineeringprojects.com/2016/01/interfacing-lm35-pic-microcontroller.html>.
- [4] **Electronic Wings.** Electronic Wings. [En línia] 26 / Juny / 2017. [Data: 20 / Febrer / 2020.] <https://www.electronicwings.com/pic/pic18f4550-watchdog-timer>.
- [5] **Joshi, Neha i Patil, VR.** ResearchGate. *Data Acquisition, processing and plotting using PIC Microcontroller-Matlab interface.* [En línia] Febrer / 2012. [Data: 20 / Febrer / 2020.] https://www.researchgate.net/publication/315487186_Data_Acquisition_processing_and_plotting_using_PIC_Microcontroller-Matlab_interface.
- [6] **Ghosh, Mousam, et al.** IEEE Computer Society. *Design and Implementation of PIC16F877A Microcontroller Based Data.* [En línia] 2016. [Data: 20 / Febrer / 2020.] <https://uksim.info/isms2016/CD/data/0665a419.pdf>.
- [7] **Microchip.** [En línia] 2020. www.microchip.com.
- [8] **CCS.** Custom Computer Services. [En línia] 2020. www.ccsinfo.com.
- [9] **Simple Projects.** Simple-Circuit. [En línia] 9 / Desembre / 2018. [Data: 3 / Març / 2020.] <https://simple-circuit.com/pic16f877a-bme280-sensor-ccs-c/>.
- [10] **Explore Embedded.** Explore Embedded. [En línia] 1 / Gener / 2016. [Data: 10 / Març / 2020.] [https://exploreembedded.com/wiki/Interfacing_DS1307\(RTC\)_with_PIC16F877A](https://exploreembedded.com/wiki/Interfacing_DS1307(RTC)_with_PIC16F877A).
- [11] **Castaño Giraldo, Sergio Andrés.** Control Automático Educación. [En línia] 26 / Juliol / 2019. [Data: 15 / Març / 2020.] <https://controlautomaticoeducacion.com/microcontroladores-pic/timer-pic/>.
- [12] **Circuit Basics.** DIY Electronics. [En línia] 2020. [Data: 16 / Març / 2020.] <https://www.circuitbasics.com/diy-electronics/>.

- [13] **Engineers Garage**. Measuring Room temperature using LM35 Temperature Sensor with Arduino. [En línia] 25 / Juny / 2019. [Data: 20 / Març / 2020.] <https://www.engineersgarage.com/arduino/measuring-room-temperature-with-lm35-using-arduino/>.
- [14] **Dejan**. How To Mechatronics. *Arduino and HC-12 Long Range Wireless Communication Module*. [En línia] 12 / Juliol / 2017. [Data: 7 / Abril / 2020.] <https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-12-long-range-wireless-communication-module/>.
- [15] *Normas de Comunicación en Serie: RS-232, RS-422 y RS-485*. **Forero Saboya, Néstor Gabriel**. 13, Bogotá : Ingenio Libre, 2012, Vol. XI.
- [16] **Estrada Marmolejo, Rubén**. Hetpro. *Puerto Serial – protocolo y su teoría*. [En línia] 27 / Octubre / 2017. [Data: 2 / Maig / 2020.] <https://hetpro-store.com/TUTORIALES/puerto-serial/>.
- [17] **IBM**. Comunicaciones asíncronas. [En línia] 15 / Gener / 2016. [Data: 4 / Maig / 2020.] https://www.ibm.com/support/knowledgecenter/es/ssw_aix_72/network/asynch_intro.html.
- [18] **Robots-Argentina**. Robots Didácticos. *¿Qué es la comunicación serie?* [En línia] 20 / Juny / 2019. [Data: 5 / Maig / 2020.] <http://robots-argentina.com.ar/didactica/que-es-la-comunicacion-serie/>.
- [19] **Robots-Argentina**. Robots Didácticos. *Descripción y funcionamiento del Bus I2C*. [En línia] 7 / Desembre / 2018. [Data: 5 / Maig / 2020.] <http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/>.
- [20] **CEMDAL**. Consultoría en Compatibilidad Electromagnética. *La Directiva de Baja Tensión 2014/35/UE*. [En línia] 7 / Novembre / 2018. [Data: 10 / Maig / 2020.] <https://www.cemdal.com/2018/11/07/la-directiva-de-baja-tensi%C3%B3n-2014-35-ue/>.
- [21] **OneAir**. NORMATIVA DE DRONES EN ESPAÑA 2020. [En línia] 14 / Gener / 2020. [Data: 11 / Maig / 2020.] <https://www.oneair.es/normativa-drones-espana-aesa/>.
- [22] **Huang, Han-Way**. *PIC Microcontroller: An Introduction to Software and Hardware Interfacing*. Minnesota : Thomson Delmar Learning, 2004. ISBN 9781401839673.
- [23] **García Breijo, Eduardo**. *Compilador C CCS y Simulador PROTEUS para Microcontroladores PIC*. Barcelona : Marcombo, 2008. ISBN 9788426714954.
- [24] **Ibrahim, Dogan**. *PIC Microcontroller Projects in C: Basic to Advanced*. s.l. : Newnes, 2008. ISBN 9780080999241.

- [25] **Díaz Gómez, Miguel Angel.** [En línia] <http://arduinoamujete.blogspot.com/>.
- [26] **Black Box.** [En línia] <https://www.blackbox.com.mx/mx-mx/page/25065/Recursos/Technical/black-box-explica/Fibre-Optic-Cable/Simplex-vs-duplex-fiber-patch-cable>.
- [27] **GUIMI.** [En línia] <http://guimi.net/blogs/hiparco/baterias/>.
- [28] **PowerPlanet.** [En línia] <https://www.powerplanetonline.com/bateria-syma-x5c>.
- [29] **CDMX Electrónica.** [En línia] <https://www.cdmxelectronica.com/producto/sensor-de-presion-barometrica-bmp180/>.
- [30] **Botland.** [En línia] https://botland.com.pl/en/pressure-sensors/11803-bme280-humidity-temperature-and-pressure-sensor-110-kpa-i2c-spi-33v.html?search_query=BME280&results=7.
- [31] **Analog Devices.** [En línia] <https://www.analog.com/en/products/tmp35.html#product-overview>.
- [32] **Beat Your Beat.** [En línia] <https://beatyourbit.com/arduinointerface/hc05/>.
- [33] **ELECROW.** [En línia] <https://www.elecrow.com/433mhz-serial-rf-module-hc12-1000m-p-874.html>.
- [34] **Onion DOCS.** [En línia] <https://docs.onion.io/omega2-docs/uart1.html>.
- [35] **SparkFun.** [En línia] <https://learn.sparkfun.com/tutorials/i2c/all>.
- [36] **Simple Projects.** Simple-Circuit. [En línia] 25 / Agost / 2018. [Data: 20 / Març / 2020.] <https://simple-circuit.com/pic-microcontroller-gps-module-mikroc/>.