



**INSTITUT
LLUÍS DE
REQUESENS**

DISSENY I CONSTRUCCIÓ D'UN QUADRICÒPTER



**AUTOR: JAUME SÁNCHEZ SALAZAR
TUTOR: LLUÍS NADAL BALANDRAS
2N BATXILLERAT B
CURS 2016-2018**

ÍNDEX

1. Índex d'il·lustracions	3
2. Índex de programes	5
3. Agraïments i cites	6
4. Resum –Abstract	7
4.1 Resum.....	7
4.2 Abstract.....	8
5. Motivacions.....	9
6. Objectiu.....	10
7. Introducció	11
7.1 Definició	11
7.2 Història.....	11
7.2.1 Primera Guerra Mundial.....	12
7.2.2 Segona Guerra mundial.....	14
7.2.3 Guerra del Vietnam.....	15
7.2.4 Actualitat i tipus de drons.....	16
8. Realització	18
8.1 Materials	18
8.2 Dia 1.....	20
8.3 Dia 2.....	21
8.4 Dia 3.....	28
8.5 Dia 4.....	34
8.6 Dia 5.....	38
8.7 Dia 6.....	45
8.8 Dia 7.....	45
8.9 Dia 8.....	48
8.10 Dia 9.....	55
8.11 Dia 10.....	55
8.12 Dia 11.....	55
8.13 Dia 12.....	57
8.14 Dia 13.....	57
8.15 Dia 14.....	59

8.16 Dia 15.....	61
8.17 Configuració i PID definitius	63
9. Resultats.....	68
10. Conclusió.....	69
11. Annexos	70
11.1 Manual motor	70
11.2 Manual acceleròmetre.....	71
11.3 Manual ESC	73
12. Fonts d'informació	77

1. Índex d'il·lustracions

Imatge 1: Quadre dels globus austríacs.....	11
Imatge 2: Hewitt Sperry.....	13
Imatge 3: Siemens Torpedo Planeador.....	13
Imatge 4: GB-1.....	14
Imatge 5: Firebee.....	15
Imatge 6: Luciérnagas.....	16
Imatge 7: Drons missatgers.....	17
Imatge 8: Placa Arduino 101.....	18
Imatge 9: Placa Arduino 101 revers.....	18
Imatge 10: Cable USB.....	18
Imatge 11: Aspecte de l'Arduino Create.....	18
Imatge 12: Tipus de drons segons els motors.....	20
Imatge 13: Sentits de rotació dels motors.....	20
Imatge 14: nRFConnect.....	22
Imatge 15: S'envia el nombre 1.....	23
Imatge 16: S'envia el nombre 0 (apagat).....	23
Imatge 17: Taula de valors de caràcters ASCII.....	24
Imatge 18: nRF enviant caràcter ASCII.....	24
Imatge 19: RealTerm.....	25
Imatge 20: Carregant la bateria LiPo.....	28
Imatge 21: ESC QBrain.....	28
Imatge 22: Parts del motor.....	29
Imatge 23: Motor sense escombretes Turnigy 1000KV.....	29
Imatge 24: Circuit amb motor, bateria, ESC i Arduino.....	29
Imatge 25: Mesura de l'empenta del motor.....	32
Gràfic 26: Gràfica de l'empenta del motor amb diferents hèlixs.....	33
Imatge 27: Mesura del pes dels components.....	33
Imatge 28: Femella de niló.....	34
Imatge 29: Discs DVD.....	34
Imatge 30: Barres de fibra de carboni.....	34
Imatge 31: Caragols de niló.....	34
Imatge 32: Foradant els discs.....	35
Imatge 33: Primer xassís del dron.....	35
Imatge 34: Veta adherent o velcro.....	35
Imatge 35: Muntatge del dron.....	36
Imatge 36: Dron acabat.....	36
Imatge 37: Esquema connexions motors.....	37
Imatge 38: Comprovant les lectures de l'acceleròmetre.....	39
Imatge 39: Perfil d'alumini en L.....	46
Imatge 40: Policarbonat cel·lular.....	46
Imatge 41: Segon dron.....	46

Imatge 42: Muntatge del dron.....	47
Imatge 43: Segon dron acabat.	47
Imatge 44: Teclat.....	48
Imatge 45: Connexions teclat.	49
Imatge 46: Muntant el comandament a distància.	49
Imatge 47: Comandament a distància acabat	50
Imatge 48: Dron atrapat a la cistella de bàsquet.	54
Imatge 49: Nova placa controladora.....	55
Imatge 50: Primera emissora FlySky.....	55
Imatge 51: Primera receptora FlySky.	56
Imatge 52: Regulador de tensió amb un cricuit integrat 7805	56
Imatge 53: Dron amb receptor muntat i l'emissora de ràdio.	56
Imatge 54: Segona emissora de ràdio FlySky	57
Imatge 55: Segon receptor de ràdio FlySky	57
Imatge 56: Mesura de l'amplada del receptor.	57
Imatge 57: Mesura de l'amplada del receptor	57
Imatge 58: Receptor definitiu, sortides en sèrie i paral·lel.....	58
Imatge 59: Connexió de la placa controladora amb el receptor.	58
Imatge 60: Muntatge final del dron.	59
Imatge 61: Dron agafat per cordes.....	60
Imatge 62: Portàtil i dron.	60
Imatge 63: Dron volant	62
Imatge 64: Dron després d'estavellar-se	62
Imatge 65: PID definitiu	63
Imatge 66: Configuració definitiva	63
Imatge 67: Configuració definitiva 2	64
Imatge 68: Configuració definitiva 3.	64
Imatge 69: Ports	65
Imatge 70: Failsafe.....	65
Imatge 71: Receiver.	66
Imatge 72: Motors	66
Imatge 73: Gràfica del moviment.....	67

2. Índex de programes

Programa 1: <i>Blink</i>	22
Programa 2: <i>Led Control</i>	26
Programa 3: <i>Motor Brushless esc serial</i>	30
Programa 4: <i>Motor 4 brushless</i>	38
Programa 5: <i>TR_ Accelerometer</i>	39
Programa 6: <i>TR Dron Definitiu</i>	41
Programa 7: <i>CustomKeypad teclat membrana</i>	51
Programa 8: <i>Control central teclat membrana</i>	52

3. Agraïments i cites

Especials agraïments:

- Als meus pares per motivar-me i donar-me la confiança necessària per a dur a terme aquest treball.
- Al tutor d'aquest treball, per oferir-nos la possibilitat de realitzar aquest treball i ajudar-nos amb els seus coneixements.
- A l'Institut Lluís de Requesens per proporcionar-nos a mi, i als meus companys, material i espai per a executar el nostre treball de recerca.
- A totes aquelles persones que ens ajudaren a facilitar el nostre treball, ja sigui amb préstec de materials o amb proveïment d'informació.

Cites:

- *"No es como empieza, sino como acaba"* (Autor desconegut)

4. Resum –Abstract

4.1 Resum

Aquest projecte, com bé indica el títol, consisteix en el disseny i la construcció d'un quadricòpter o bé, en anglès, "drone". Per a aconseguir-ho s'han hagut de dur a terme diversos intents.

En primer lloc, el primer dron, es va fabricar amb un xassís amb fibra de carboni en forma de X. El quadricòpter s'havia d'estabilitzar mitjançant una placa Arduino-101, amb un programa escrit per nosaltres i enllaçada per Bluetooth amb una altra placa Arduino-101 connectada, a l'ordinador per on es donaven les ordres de navegació.

Posteriorment, la segona placa Arduino-101 es va muntar en una capsa amb un teclat i una bateria de liti. El resultat va ser una mena de comandament a distància que no necessitava l'ordinador. No vam aconseguir que aquest dron volés estable.

En segon lloc, i a la recerca d'una solució als problemes emergents, es va fabricar un xassís diferent amb un rectangle de policarbonat cel·lular i dos perfils d'alumini. El resultat va ser una estructura en H més rígida i amb menys vibracions. La placa controladora encara era un Arduino-101 comandat per Bluetooth. Tampoc es va aconseguir fer volar aquest dron d'una manera estable.

Aprofitant aquest últim xassís, es va canviar la placa controladora per una placa CleanFLight SP Racing F3 Acro, comandada amb una emissora de ràdio FlySky FS-i6 un receptor FlySky FS-iA6B.

Després d'uns quants intents, es va aconseguir ajustar el PID de la placa controladora, mitjançant el programa CleanFLight, a un valor correcte i el dron ja va volar amb una bona estabilització horitzontal, assolint l'objectiu del treball.

4.2 Abstract

This project, as well indicates the title, consists on the design and the construction of a quadcopter. To achieve it, several attempts have been made.

In the first place, the first drone, was manufactured with a carbon fibre chassis in a X form. The drone had to be stabilised by means of a plate Arduino-101, with a program written by us and linked by Bluetooth with another Arduino-101 plate connected to the computer through which the orders of navigation were given.

Later, the second plate Arduino-101 was assembled in a box with a keyboard and a lithium battery. The result was a type of remote control that did not need the computer. We did not achieve that it flew stable.

In the second place, and in search of a solution to emerging problems, we manufactured a different chassis with a rectangle of cellular polycarbonate and two profiles of aluminium. The result was an H shape structure, more rigid and with fewer vibrations. The plate controller was still an Arduino-101 commanded by Bluetooth. We could not achieve the flight of the drone was stable either.

Taking advantage of this last chassis, we changed the plate controller for a plate CleanFLight SP RacingF3 Acro, commanded with radio signal Flysky FS-i6 and a receptor Flysky FS-iA6B.

After a few attempts, we were able to adjust the PID to the plate controller, by means of the program CleanFLight, to a correct value and the drone could already fly with a good horizontal stabilisation, attaining the aim of the work.

5. Motivacions

La motivació principal d'aquest projecte és fruit d'una sortida que vam realitzar en horari escolar, gràcies al departament de Tecnologia, a una conferència sobre drons. En aquesta, els temes que es van exposar van ser d'un àmbit més tècnic i legislatiu i, per tant, no tant atractiu i pràctic com el que ens esperàvem. Per exemple, se centraren més a especificar quins permisos es necessitava per poder volar un dron que no pas en permetre'ns veure com volava un.

A causa d'aquesta experiència, ens vam quedar amb les ganes de saber més sobre aspectes més interessants (pràctics sobretot), d'un món tan ampli com és el dels drons; i vam decidir plantejar al nostre tutor un treball sobre drons.

Personalment, m'agradava la idea d'orientar el nostre projecte cap als drons, perquè és una eina que ha estat molt útil des d'un passat i cada cop ho està sent més en l'actualitat, ja que les seves funcions es van diversificant. A més a més, per a l'elaboració i el funcionament correcte d'un dron es requereixen coneixements físics, tecnològics i informàtics; aspectes que encara que no conegui del tot, m'agrada tractar-los.

6. Objectiu

L'objectiu d'aquest projecte és el disseny i la construcció d'un dron, basats en l'aprofitament d'eines pròpies, excepte certes peces com ara els motors o plaques electròniques, i en un sistema de vol el màxim eficient possible, controlat per una placa controladora, un receptor i una ràdio emissora.

A més a més, és una oportunitat per ampliar coneixements sobre un tema en concret, en aquest cas els drons.

7. Introducció

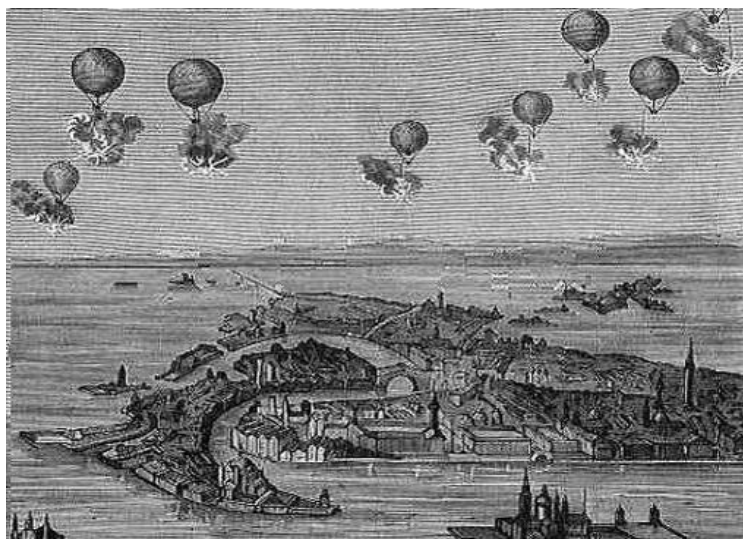
7.1 Definició

Un vehicle aeri no tripulat (VANT), UAV (en anglès: *Unmanned Aerial Vehicle*), o comunament dron, és un vehicle que vola sense tripulació, capaç de mantenir de manera autònoma un nivell de vol controlat i sostingut, i propulsat per un motor d'explosió, elèctric, o de reacció.

7.2 Història

La idea de l'avió no tripulat és antiga. Tot i que sovint associem els drons amb els robots militars d'avui, els avions no tripulats, d'una manera o altra, s'han utilitzat durant dècades. Un dels primers usos registrats va ser pels austríacs, el juliol de 1849, després que posessin en marxa al voltant de dos-cents globus aerostàtics no tripulats muntats amb bombes a la ciutat de Venècia.

Menys de dues dècades després de la Guerra Civil a EE.UU, forces de la Confederació i de la Unió volaven globus per a missions de reconeixement. El 1896, Samuel P. Langley va desenvolupar una sèrie d'aeronaus a vapor, avions sense pilot que van ser traslladats amb èxit al llarg del riu Potomac, prop de Washington DC. La pràctica de la vigilància aèria més tard va sorgir en la Guerra Hispà-Americana de 1898, quan els militars dels EUA van equipar una càmera a un estel, donant lloc a una de les primeres fotografies de reconeixement aeri.



Imatge 1: Quadre dels globus austríacs.
<http://eldrone.es/historia-de-los-drones/>

7.2.1 Primera Guerra Mundial

A la Primera Guerra Mundial, es va utilitzar àmpliament la vigilància aèria. Els militars utilitzaven aquests estels per obtenir fotografies aèries i seguir els moviments de l'enemic formant mapes de situació. Així, aquest seria un dels passos en l'evolució dels avions no tripulats als Estats Units, procés al qual se sumen altres quatre fases posteriors que són les següents.

En primer lloc, el precursor del dron es va utilitzar com a *blanc de pràctica* per a les forces militars de principis del segle XX. En segon lloc, en el període d'entreguerres i en la Segona Guerra Mundial, l'avió no tripulat va ser dissenyat per ser com una mena de *bomba volant* que podria ser enviada després de les línies enemigues. En tercer lloc, durant la Guerra Freda, l'avió no tripulat va ser vist com una *plataforma de vigilància* viable capaç de capturar dades d'intel·ligència en àrees de difícil accés. En quart lloc, el dron, en la guerra contra el terrorisme, s'ha convertit en una arma que fusiona la capacitat de *vigilància i la de matar*, convertint-se en un depredador.

Les aeronaus no tripulades no haurien estat possibles sense grans avenços tecnològics com la ràdio, el telègraf o el comandament a distància o ràdio control d'un vehicle inventat per Nikola Tesla a finals del segle XIX.

El 1916, la idea de les armes guiades de forma remota va despertar l'interès del capità Archibald M. Low, de la Royal Flying Corps al Regne Unit que va supervisar la construcció d'una sèrie d'avions dirigits per control remot que van ser equipats amb explosius.

L'inventor del giroscopi, Elmer Ambrose Sperry, va desenvolupar una plataforma d'aeronaus sense pilot amb un dispositiu per llançar torpedes amb una catapulta. Una publicació del New York Times de 1926 apunta que aquestes aeronaus eren guiades amb gran precisió i que després d'una distància predeterminada giraven i volaven cap avall en vertical, amb una càrrega tal de TNT capaços de volar una població sencera.

La "*Hewitt Sperry*" o "bomba volant va ser capaç de volar 50 milles carregada amb una bomba de 300 lliures de pes. Tot i així, és important destacar que aquest avió no tripulat es va millorar inequívocament amb l'addició de la tecnologia giroscòpica de Sperry.



Imatge 2: Hewitt Sperry.

https://ca.wikipedia.org/wiki/Vehicle_aeri_no_tripulat#/media/File:Kettering-bug-1.jpeg

L'èxit d'aquest projecte va portar l'Exèrcit EUA a posar en marxa un segon projecte, el *Kettering* torpede aeri "*Bug*", desenvolupat per l'empresa Dayton-Wright Airplane Company. "L'insecte" va ser essencialment un torpede aeri, sense pilot i guiat pels controls preestablerts. A Alemanya, un projecte similar estava sent iniciat pel Dr. Wilhelm von Siemens entre 1915 i 1918. El Siemens Torpede Planador era un míssil que es valia d'un Zeppelin i després es guiava cap al seu objectiu per ràdio. La bomba volant, el Bug i el Torpede Planador eren precursors dels míssils de creuer contemporanis.



Imatge 3: Siemens Torpedo Planeador. <http://eldrone.es/historia-de-los-drones/>

7.2.2 Segona Guerra mundial

Durant la Segona Guerra Mundial l'assumpte es va tractar de forma diferent. La influència de les aeronaus en el camp de batalla terrestre i naval va ser objecte de discussió. En aquesta situació els vehicles aeris no tripulats van trobar un altre rol, el de blancs.

La Marina nord-americana va llançar un nou programa, anomenat Operació Anvil, contra búnquers alemanys usant bombarders readaptats per a portar el doble d'explosius guiats per control remot per a estavellar-se als búnquers.

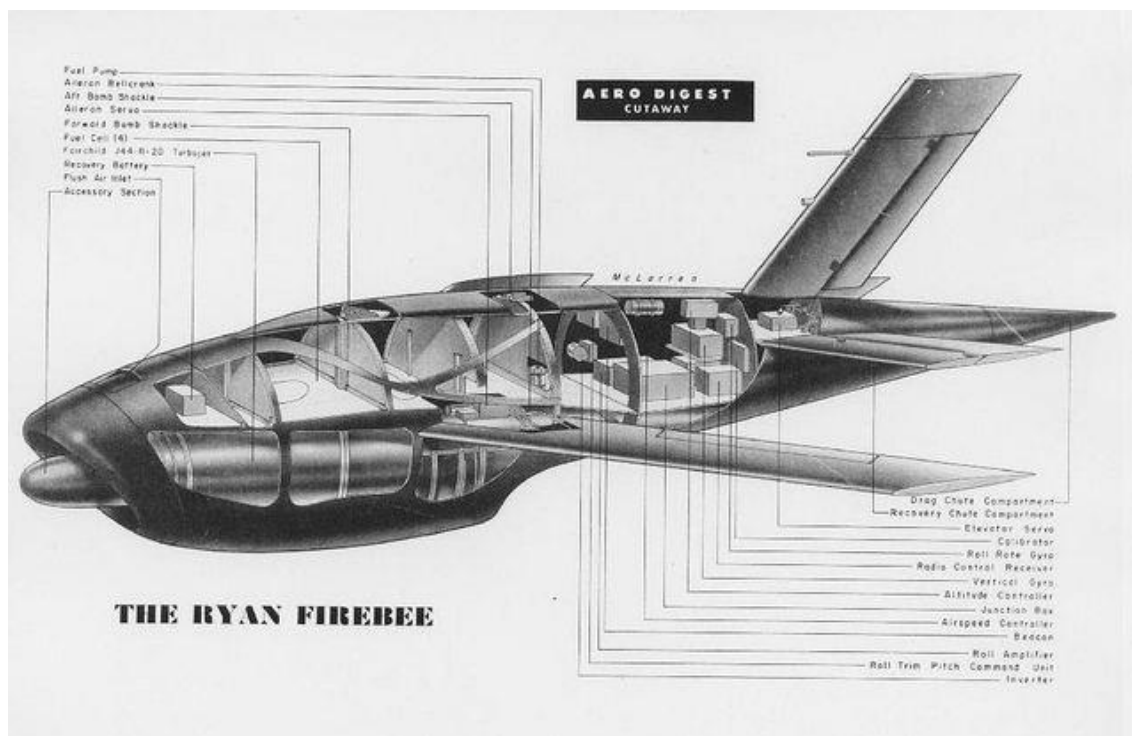
La tecnologia per control remot estava molt limitada, incloent dispositius de ràdio connectats a un motor, de tal manera que els pilots havien de guiar l'artefacte a una alçada determinada per després llençar-se en paracaigudes. A la pràctica, el programa va ser un desastre, moltes aeronaus s'estavellaven. Després de la guerra, EUA i Rússia se sumarien a la febre dels míssils.

Però, abans d'això, a mitjans de la dècada de 1940 assistim al desenvolupament del *GB-1 Glide* sistema de bombardeig aeri ideat per eludir les defenses aèries alemanyes. Era un planador viable equipat amb una bomba estàndard de 1.000 o 2.000 lliures de pes. Fet de fusta contraxapada amb les ales, timons, i controlat per ràdio, eren guiats cap a línies enemigues. El 1943, cent vuit GB-1 van ser llançats sobre Colònia, causant greus danys.



Imatge 4: GB-1. <http://eldrone.es/historia-de-los-drones/>

A la fi de 1946 un programa de la Força Aèria dels EUA va ser aprovat per desenvolupar tres tipus d'avions no tripulats per al seu ús com a objectius de formació. Dels tres, el que es llançava des de l'aire, el model Q-2, va ser el més important, convertint-se en el pare d'una classe d'avions "diana" construïts per la Companyia Aeronàutica Ryan. El "*Firebee*" es va provar per primera vegada el 1951 a la base aèria de Holloman. Aquest model podia mantenir-se en vol durant dues hores i va ser capaç d'aconseguir altures de fins a 60.000 peus.



Imatge 5: Firebee. <http://eldrone.es/historia-de-los-drones/>

El desenvolupament dels drons es va col·lapsar durant dècades a causa de la poca necessitat que hi havia d'ells com a conseqüència de l'èxit i els desenvolupaments en el camp dels míssils.

7.2.3 Guerra del Vietnam

La guerra del Vietnam, sens dubte, va ser molt important en aquesta cursa històrica dels drons per ser el que són avui dia. El conflicte va donar a llum el programa més sofisticat de vigilància amb avions no tripulats en la història de l'aviació. D'altra banda, la guerra del Vietnam va ser la primera "guerra tecnològica" de la història: una guerra duta a terme d'acord amb principis tècnics, models estadístics i sistemes electrònics.

És difícil apreciar la rapidesa amb què avança la tecnologia militar durant la Guerra Freda. Al maig de 1964, els EUA van començar a considerar l'enviament d'avions no tripulats per reemplaçar els seus O-2 en missions d'espionatge sobre Cuba. "Luciérnagas" operats per EUA van ser posteriorment

utilitzats per a vigilància en les denominades "zones sensibles", escenaris protagonistes de la Guerra Freda: entre ells Cuba, Corea del Nord i la República Popular de la Xina. El novembre de 1964, el Washington Post va informar que Xina afirmava haver abatut un avió de reconeixement nord-americà sense pilot. L'exèrcit dels Estats Units va mantenir silenci sobre aquests fets, de la mateixa manera que faria dècades més tard després que els iranians capturessin un avió avançat de la CIA.



Imatge 6: Lucièrnagas. <http://eldrone.es/historia-de-los-drones/>

Entre 1964 i 1975, aquests *Lighting Bugs* van realitzar més de 3.500 missions de combat a Vietnam.. El 1972 avions teledirigits de vigilància van ser equipats amb tecnologia LORAN (Long Range Navigation o navegació de llarg abast) que van millorar dràsticament la capacitat operacional d'aquestes aeronaus no tripulades gràcies a un sistema d'ajuda a la navegació electrònic que utilitza l'interval transcorregut entre la recepció de senyals de ràdio transmeses des de tres o més transmissors per determinar la posició del receptor (cosa semblant a la triangulació que utilitzen els GPS d'avui en dia).

A Vietnam, avions no tripulats "Lucièrnagas" es van utilitzar extensament i per a gran varietat de missions. Entre 1964 i 1975, més de 1.000 Lighting Bugs van realitzar més de 34.000 missions de vigilància en tot el sud-est d'Àsia.

7.2.4 Actualitat i tipus de drons

Gradualment, aquesta diversificació dels UAV's es va anar ampliant fins a la gran varietat de vehicles aeris no tripulats actuals. En els darrers anys també s'ha de destacar l'aparició de vehicles aeris no tripulats per aplicacions civils.

Actualment, aquesta tecnologia s'ha començat a utilitzar en l'àmbit civil per a labors més útils per al desenvolupament de la humanitat. Hi ha una gran

varietat de camps en els quals es poden fer servir els drons avui en dia, diferenciada per a dos usos: militars o civils.

Els d'ús militar solen anar armats i amb capacitat de bombardeig; per altra part, els civils són aquells que faciliten la vida a la societat, com ara:

Internet: distribució de senyal gratuïta d'internet.

Serveis forestals: seguiment de les àrees boscoses, control d'incendis.

Cartografia: realització d'ortofotomapes i de models d'elevacions del terreny d'alta resolució.

Recerca, rescat i salvament de persones

Geologia.

Monitorització d'instal·lacions.

Medi ambient: estat de l'atmosfera.

Transport i lliurament de mercaderies.

Seguiment de la planificació urbanística.

Agricultura: gestió de cultius.

Seguretat i control fronterer.



Imatge 7: Drons missatgers. <https://i.blogs.es/06c13f/drones/original.jpg>

8. Realització

8.1 Materials

Els estris necessaris per a dur a terme tot el treball de programació són: un ordinador, una placa Arduino 101, un cable USB i la plataforma de programació Arduino (un "IDE"), que pot ser local instal·lant el programa a l'ordinador o bé "online" amb l'Arduino Create.



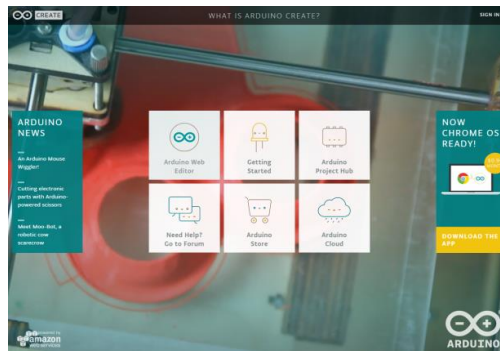
Imatge 8: Placa Arduino 101.
<https://www.sparkfun.com/products/13787>



Imatge 9: Placa Arduino 101 revers.
<https://www.sparkfun.com/products/13787>



Imatge 10: Cable USB.
<https://www.startech.com/Cables/USB-2.0/USB-2.0-Cables/20m-65ft-Active-USB-2-A-to-B-Cable-M-M-USB2HAB65AC>



Imatge 11: Aspecte de l'Arduino Create.
<https://create.arduino.cc/>

Arduino és una companyia de maquinari lliure i una comunitat tecnològica que dissenya i crea plaques de desenvolupament de maquinari, compostes per microcontroladors, elements passius i actius. L'elecció d'aquesta plataforma és deguda al fet que Arduino s'enfoca en apropar i facilitar l'ús de l'electrònica i programació gràcies a la gran comunitat d'internautes que l'envolta.

La placa Arduino 101, va ser escollida pel seu preu econòmic i per contenir el que és bàsic pel control del dron: acceleròmetre de sis eixos, giroscopi i connectivitat Bluetooth BLE.

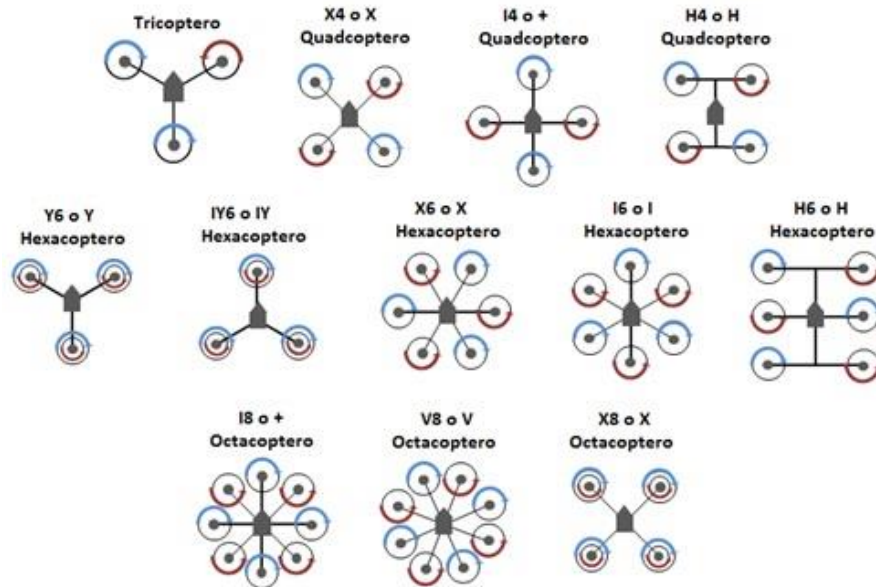
Els acceleròmetres mesuren l'efecte de la força de gravetat sobre ells mateixos, és a dir, mesuren l'acceleració, la inclinació i les vibracions amb la finalitat de trobar un equilibri. Un giroscopi, en canvi, és un dispositiu que funciona per mesurar velocitats angulars basant-se en el manteniment de l'impuls de rotació. Finalment, la connectivitat per Bluetooth, permet controlar el dron a distància.

No obstant, la placa Arduino 101, també presenta moltes altres característiques, com ara:

- Mòdul Intel Curie.
- Tensió de funcionament 3.3V (5V tolerant I / O).
- Voltatge d'entrada (recomanat) 7-12V.
- Voltatge d'entrada (límit) 7-20V.
- 14 pins digitals I / O (dels quals 4 proporcionen sortida PWM).
- 6 pins d'entrada analògica
- Corrent continu per Pin I / O 20 Ma.
- Longitud: 68,6 mm.
- Amplada: 53,4 mm.
- Pes: 34 gr.

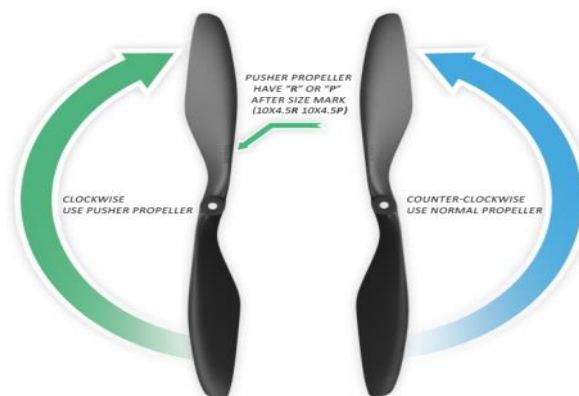
8.2 Dia 1

Existeixen diferents tipus de drons segons el nombre de motors i la distribució d'aquests. Volíem fabricar un quadricòpter i, per tant, la quantitat de motors ja ens venia regida, és a dir, necessitariem quatre motors. Només calia decidir quina distribució faríem servir i, comparant entre les més convenientes i útils, vam decidir que, probablement, emprariem el quadricòpter X o bé el quadricòpter H.



Imatge 12: Tipus de drons segons els motors.
<https://www.sparkfun.com/products/13787>

Podem distingir dos tipus de motors segons la seva rotació. En anglès, *clockwise* o *counterclockwise* i, en català, sentit horari o sentit antihorari. Generalment, si el nombre de motors és parell, la rotació d'aquests sol anar per parelles, com s'aprecia a la fotografia nº12.



Imatge 13. Sentits de rotació dels motors.
<http://www.pyramidmodels.com/shop/shopimages/props%20rotation.jpg>

L'explicació d'això, és que cada motor genera un moment angular en sentit horari o antihorari que faria girar el dron horitzontalment en sentit contrari. Si es posen aparellats girant en sentits contraris, els dos moments angulars es contraresten.

De la mateixa manera, existeixen dos tipus d'hèlixs segons la rotació, horari o antihorari, i el nombre de pales que tinguin, les més comunes són les bipala (dues pales) i les tripala (tres pales).

Els pins de la placa Arduino es poden configurar com a entrades (llegir un voltatge) o com a sortides (donar voltatge i corrent).

Com s'ha esmentat abans, la placa Arduino 101 disposa de connectivitat Bluetooth BLE que no és fàcil de configurar. Com que teníem dues plaques, amb la idea que una controlés l'altra per Bluetooth, vam haver d'aconseguir que es comunicessin.

Per últim, vam acordar el disseny del dron. Es basaria en un quadricòpter X, que seria controlat per dues plaques Arduino 101 enllaçades per Bluetooth. Una actuaria de receptora i l'altra actuaria d'emissora de les ordres, enviades mitjançant l'ordinador. El nostre treball tindria com a referència un altre trobat a Internet, en el qual, aparentment, s'especifiquen les eines i els processos necessaris a seguir per a elaborar-lo.

La pàgina web és la següent:

<http://www.instructables.com/id/Make-Your-Own-Removable-Drone-With-Arduino/>

8.3 Dia 2

Primerament, i per posar en pràctica els nous coneixements, vam copiar un programa de la plataforma Arduino Create. Aquest, consistia a encendre i apagar un LED que portava introduït la placa Arduino 101, abans esmentada.

Per tal d'encendre el LED, tenint la placa connectada a l'ordinador mitjançant el cable USB, s'havia d'enviar un número de l'1 al 255, és a dir, un byte com a màxim, i per apagar-lo, s'havia d'enviar el dígit 0.

A continuació, es mostra el programa emprat, anomenat *Blink*.

```

1  /*
2  Blink
3
4  Turns an LED on for one second, then off for one second, repeatedly.
5
6  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
7  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
8  the correct LED pin independent of which board is used.
9  If you want to know what pin the on-board LED is connected to on your Arduino
10 model, check the Technical Specs of your board at:
11 https://www.arduino.cc/en/Main/Products
12
13 modified 8 May 2014
14 by Scott Fitzgerald
15 modified 2 Sep 2016
16 by Arturo Guadalupi
17 modified 8 Sep 2016
18 by Colby Newman
19
20 This example code is in the public domain.
21
22 http://www.arduino.cc/en/Tutorial/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34   delay(1000); // wait for a second
35   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
36   delay(1000); // wait for a second
37 }
38

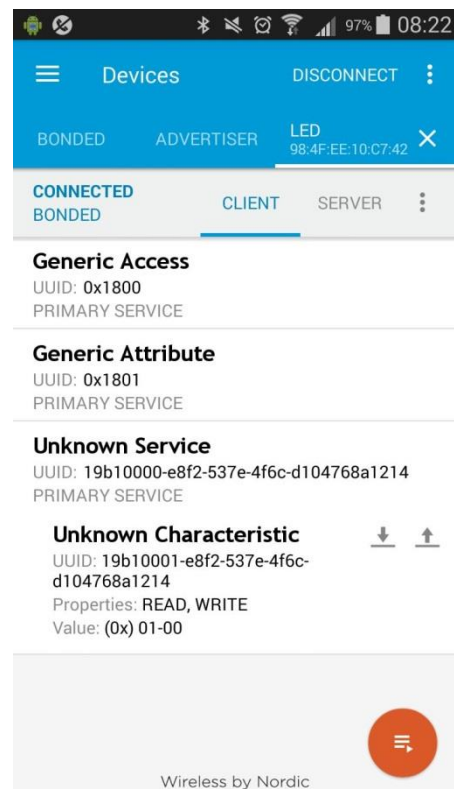
```

També, amb l'objectiu d'aprofitar la connectivitat Bluetooth de la qual disposava la nostra placa, vam connectar-la a un mòbil mitjançant l'aplicació nRFConnect per tornar a realitzar la mateixa acció que abans.

L'aplicació nRFConnect és una eina genèrica que permet escanejar, anunciar i explorar els dispositius amb baix consum d'energia (BLE) i comunicar-se amb ells.

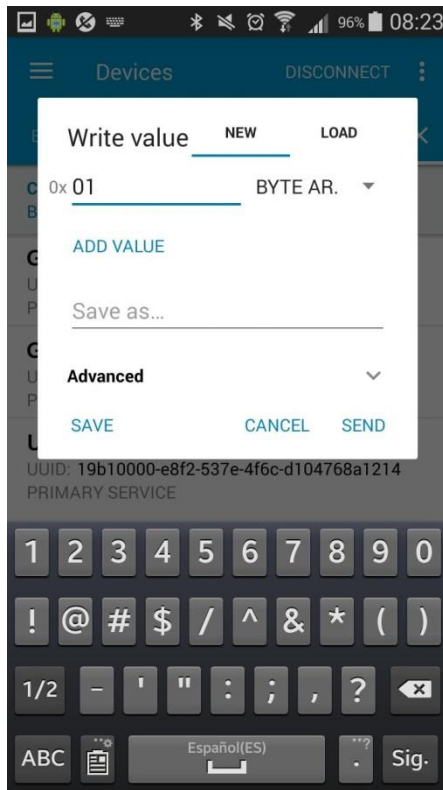
El primer pas va ser enllaçar ambdós dispositius, tal com es mostra a la fotografia següent.

A continuació, seguint el mateix raonament que abans, s'enviaven números de l'1 al 255 per encendre'l, i un 0 per apagar-lo. En aquest cas, a més a més d'enviar números, també vam poder enviar caràcters ASCII.

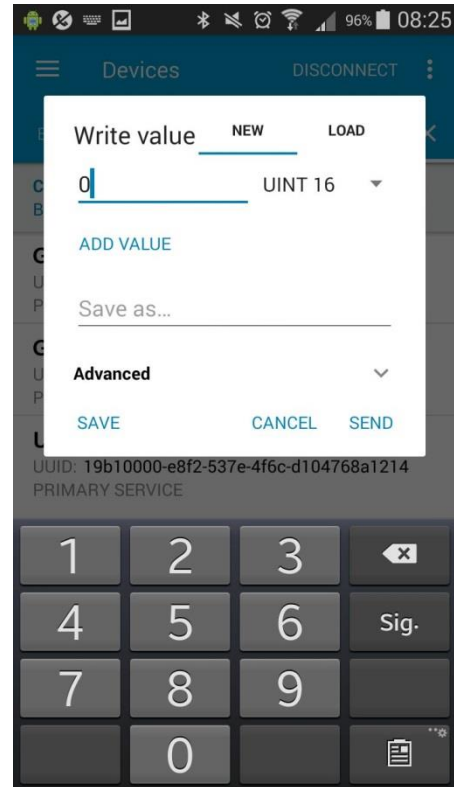


Imatge 14: nRFConnect

L'ASCII, acrònim anglès de *American Standard Code for Information Interchange*, o en català, Codi Estàndard Americà per a l'Intercanvi d'Informació, és un codi numèric que representa els caràcters, mitjançant números del 0 al 255. No tots són imprimibles, del 0 al 31 i també el 127, són caràcters de control. Per tant, cadascuna de les lletres que s'escriguin correspondran a un d'aquests codis.



Imatge 15: S'envia el nombre 1.



Imatge 16: S'envia el nombre 0 (apagat).

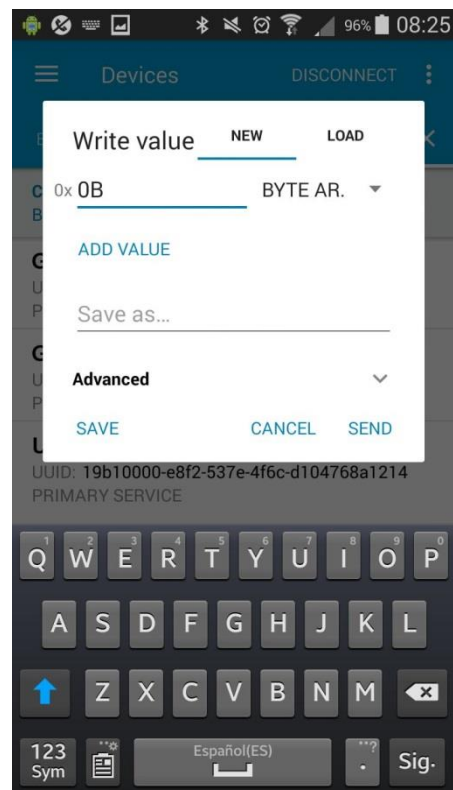
Aquest codi va ser proposat per Robert W. Bemer, buscant crear codis per a caràcters alfa-numèrics (lletres, símbols, nombres i accents). D'aquesta manera seria possible que els ordinadors de diferents fabricants aconseguissin entendre els mateixos codis.

ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo
0 0 NUL	16 10 DLE	32 20 (espacio)	48 30 0
1 1 SOH	17 11 DC1	33 21 !	49 31 1
2 2 STX	18 12 DC2	34 22 "	50 32 2
3 3 ETX	19 13 DC3	35 23 #	51 33 3
4 4 EOT	20 14 DC4	36 24 \$	52 34 4
5 5 ENQ	21 15 NAK	37 25 %	53 35 5
6 6 ACK	22 16 SYN	38 26 &	54 36 6
7 7 BEL	23 17 ETB	39 27 '	55 37 7
8 8 BS	24 18 CAN	40 28 (56 38 8
9 9 TAB	25 19 EM	41 29)	57 39 9
10 A LF	26 1A SUB	42 2A *	58 3A :
11 B VT	27 1B ESC	43 2B +	59 3B ;
12 C FF	28 1C FS	44 2C ,	60 3C <
13 D CR	29 1D GS	45 2D .	61 3D =
14 E SO	30 1E RS	46 2E /	62 3E >
15 F SI	31 1F US	47 2F /	63 3F ?
ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo
64 40 @	80 50 P	96 60 `	112 70 p
65 41 A	81 51 Q	97 61 a	113 71 q
66 42 B	82 52 R	98 62 b	114 72 r
67 43 C	83 53 S	99 63 c	115 73 s
68 44 D	84 54 T	100 64 d	116 74 t
69 45 E	85 55 U	101 65 e	117 75 u
70 46 F	86 56 V	102 66 f	118 76 v
71 47 G	87 57 W	103 67 g	119 77 w
72 48 H	88 58 X	104 68 h	120 78 x
73 49 I	89 59 Y	105 69 i	121 79 y
74 4A J	90 5A Z	106 6A j	122 7A z
75 4B K	91 5B [107 6B k	123 7B {
76 4C L	92 5C \	108 6C l	124 7C
77 4D M	93 5D]	109 6D m	125 7D }
78 4E N	94 5E ^	110 6E n	126 7E ~
79 4F O	95 5F _	111 6F o	127 7F ¯

Imatge 17: Taula de valors de caràcters ASCII.
<http://trinisoft.blogspot.com.es/2011/09/validar-campos-con-java.html>

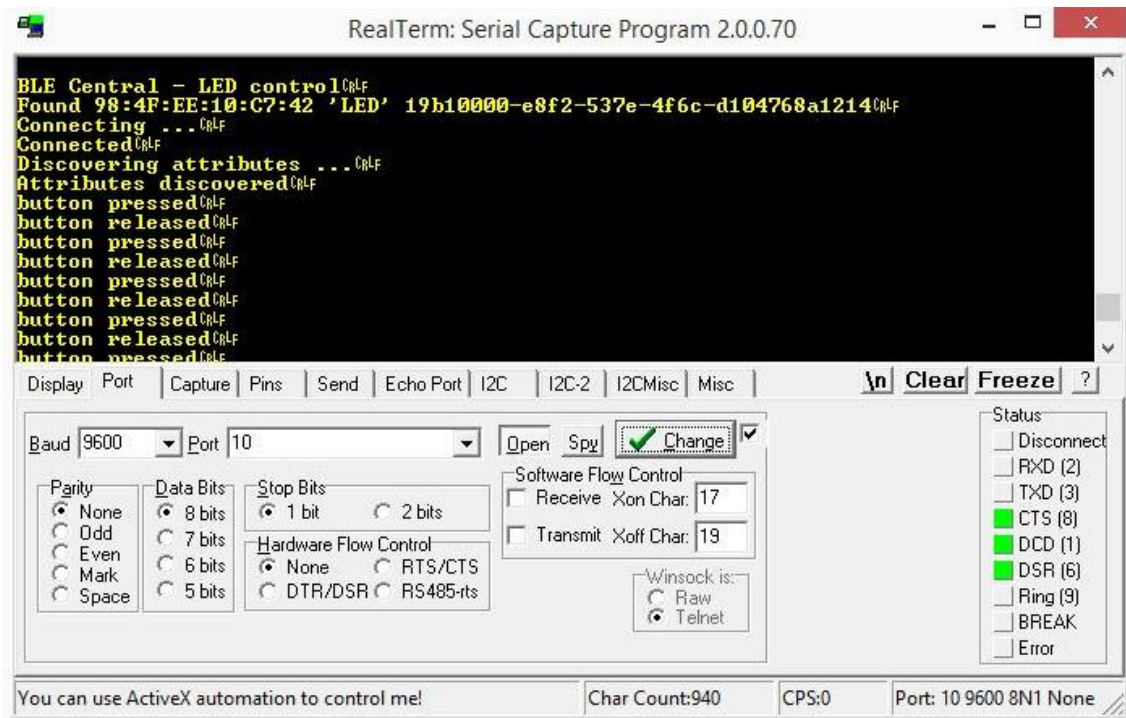
Com es pot apreciar a la captura de pantalla, vam enviar el caràcter en hexadecimal "0B", que equival al decimal 11, segons la imatge nº17.

A més a més, vam realitzar una última prova amb el LED de la placa. Aquesta vegada seria amb el programa RealTerm que va permetre comunicar dos ordinadors mitjançant connexió Bluetooth.



Imatge 18: nRF enviant caràcter ASCII.

Un dels ordinadors faria la funció d'emissor, el qual podia enviar números o caràcters ASCII. L'altre ordinador tindria connectada la placa per un cable USB i faria la funció de receptor. Doncs, depèn del que rebés (0 ó 1) encendria o apagaria el LED de la placa.



Imatge 19: RealTerm.

El programa utilitzat va ser: *Led Control*.

```

1  /*
2  * Copyright (c) 2016 Intel Corporation. All rights reserved.
3  * See the bottom of this file for the license terms.
4  */
5
6  /*
7  * Sketch: LedControl.ino
8  *
9  * Description:
10 * This is a Central sketch that looks for a particular Service with a
11 * certain Characteristic from a Peripheral. Upon succesful discovery,
12 * it reads the state of a button and write that value to the
13 * Peripheral Characteristic.
14 *
15 * Notes:
16 *
17 * - Expected Peripheral Service: 19b10000-e8f2-537e-4f6c-d104768a1214
18 * - Expected Peripheral Characteristic: 19b10001-e8f2-537e-4f6c-d104768a1214
19 * - Expected Peripheral sketch:
20 *
21 */
22
23 #include <CurieBLE.h>
24
25 // variables for button
26 int estatLed;
27
28
29 void setup() {
30   Serial.begin(9600);
31
32   // configure the button pin as input
33   //pinMode(buttonPin, INPUT);
34
35   // initialize the BLE hardware
36   BLE.begin();
37
38   Serial.println("BLE Central - LED control");
39
40   // start scanning for peripherals
41   BLE.scanForUuid("19b10000-e8f2-537e-4f6c-d104768a1214");
42 }
43
44 void loop() {
45   // check if a peripheral has been discovered
46   BLEDevice peripheral = BLE.available();
47
48   if (peripheral) {
49     // discovered a peripheral, print out address, local name, and advertised service
50     Serial.print("Found ");
51     Serial.print(peripheral.address());
52     Serial.print(" ");
53     Serial.print(peripheral.localName());
54     Serial.print(" ");
55     Serial.print(peripheral.advertisedServiceUuid());
56     Serial.println();
57
58     // stop scanning
59     BLE.stopScan();
60
61     controlledLed(peripheral);
62
63     // peripheral disconnected, start scanning again
64     BLE.scanForUuid("19b10000-e8f2-537e-4f6c-d104768a1214");
65   }
66 }
67
68 void controlledLed(BLEDevice peripheral) {
69   // connect to the peripheral
70   Serial.println("Connecting ...");
71
72   if (peripheral.connect()) {
73     Serial.println("Connected");

```

```

74   } else {
75     Serial.println("Failed to connect!");
76     return;
77   }
78
79   // discover peripheral attributes
80   Serial.println("Discovering attributes ...");
81   if (peripheral.discoverAttributes()) {
82     Serial.println("Attributes discovered");
83   } else {
84     Serial.println("Attribute discovery failed!");
85     peripheral.disconnect();
86     return;
87   }
88
89   // retrieve the LED characteristic
90   BLECharacteristic ledCharacteristic =
peripheral.characteristic("19b10001-e8f2-537e-4f6c-d104768a1214");
91
92   if (!ledCharacteristic) {
93     Serial.println("Peripheral does not have LED characteristic!");
94     peripheral.disconnect();
95     return;
96   } else if (!ledCharacteristic.canWrite()) {
97     Serial.println("Peripheral does not have a writable LED characteristic!");
98     peripheral.disconnect();
99     return;
100  }
101
102  while (peripheral.connected()) {
103    // while the peripheral is connection
104
105    // read the button pin
106    if (Serial.available() > 0) {
107      estatLed = Serial.read();
108    }
109
110
111
112    if (estatLed > 0) {
113      Serial.println("led encès");
114
115      // button is pressed, write 0x01 to turn the LED on
116      ledCharacteristic.writeByte(0x01);
117    } else {
118      ledCharacteristic.writeByte(0x00);
119      Serial.println("led apagat");
120
121    }
122  }
123  Serial.println("Peripheral disconnected");
124 }
125
126
127
128
129
130 /*
131  Arduino BLE Central LED Control example
132  Copyright (c) 2016 Arduino LLC. All right reserved.
133
134  This library is free software; you can redistribute it and/or
135  modify it under the terms of the GNU Lesser General Public
136  License as published by the Free Software Foundation; either
137  version 2.1 of the License, or (at your option) any later version.
138
139  This library is distributed in the hope that it will be useful,
140  but WITHOUT ANY WARRANTY; without even the implied warranty of
141  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
142  Lesser General Public License for more details.
143
144  You should have received a copy of the GNU Lesser General Public
145  License along with this library; if not, write to the Free Software
146  Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
147  */
148
149
150
151

```

8.4 Dia 3

El dia següent, vam muntar un petit circuit del qual formaven part:

- La placa Arduino 101.
- Un cable USB.
- Una bateria (Turnigy nano-tech 1500mah 3S 35~70C Lipo Pack).
- Un ESC (regulador electrònic de velocitat) Q BrainQuattro 4 x 20 A.
- Un motor sense escombretes (Brushless Motor) Turnigy D2830-11 1000kv.
- Una placa d'aglomerat per a fixar el motor.



Imatge 20: Carregant la bateria LiPo.



Imatge 8: ESC QBrain.

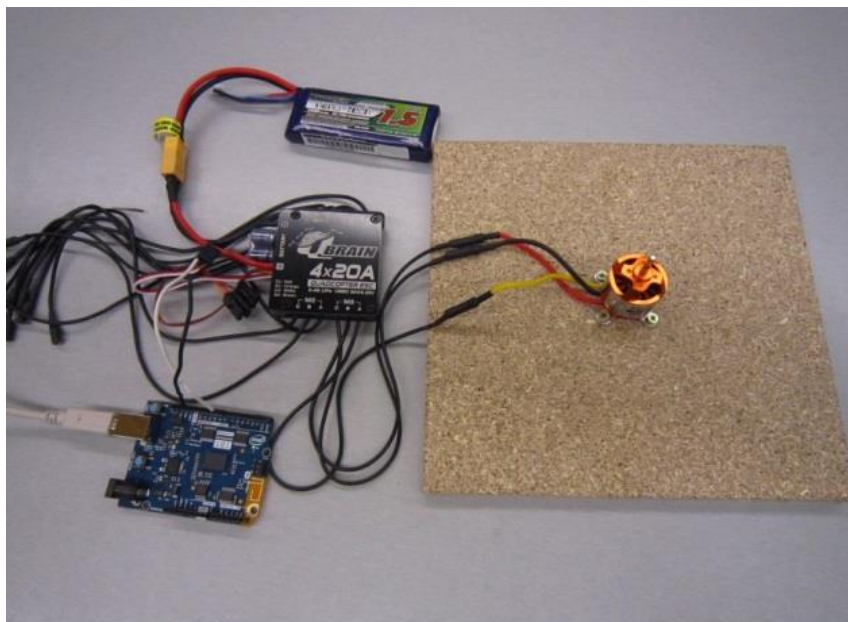
<http://www.ebay.fr/itm/Q-Brain-Quattro-4-x-20A-Brushless-Quadcopter-ESC-2-4S-3A-SBEC-orangerRX-Drone-/191847152360>



Imatge 22: Parts del motor.
https://hobbyking.com/en_us/d2830-11-1000kv-brushless-motor.html



Imatge 23: Motor sense escobretes Turnigy 1000KV.
https://hobbyking.com/en_us/d2830-11-1000kv-brushless-motor.html



Imatge 24: Circuit amb motor, bateria, ESC i Arduino.

```

1  /* Important: la placa d'Arduino ha d'estar funcionant abans de connectar la bateria i
2  un impuls de 1000 microsegons ha d'estar actiu al control del ESC.
3  amb 1000 microsegons el motor està aturat. Amb 1200 s'engega i amb 2000 agafa la
4  velocitat màxima
5
6  *****
7  Si s'empra el RealTerm, el valor dels impulsos s'ha d'enviar en format ASCII i no
8  en format nombre
9  *****
10 */
11 #include<Servo.h> //Afegeix la biblioteca servo
12 Servo ESC; //Crea un servo de nom ESC
13 int vel = 1000; //Amplada dels impulsos inicials en microsegons (motor aturat)
14
15 void setup()
16 {
17   pinMode(LED_BUILTIN, OUTPUT);
18   //Asigna un pin a l'ESC
19   ESC.attach(9, 1000, 2000);
20
21   //Activa l'ESC. Pot dependre de l'Esc
22   ESC.writeMicroseconds(vel); //1000 microsegons
23   delay(2000); //Un retard de 2 segundos abans d'emprar l'ESC
24   //Ara ja es podria connectar la bateria
25
26   //Inicia el port sèrie
27   Serial.begin(9600);
28   Serial.setTimeout(10);
29
30 }
31
32 void loop()
33 {
34   if(Serial.available() >= 1)
35   {
36     vel = Serial.parseInt(); //Llegeix una cadena ASCII pel port sèrie i la
37     transforma en un enter
38     digitalWrite(LED_BUILTIN, HIGH); //Encèn el LED incorporat per a senyalar que
39     s'ha rebut
40     //algun caracter pel port sèrie
41     delay(300); //Està encès 0.3 segons
42     digitalWrite(LED_BUILTIN, LOW); //apaga el LED
43     Serial.println(vel); //Per a comprobar el valor rebut
44     ESC.writeMicroseconds(vel); //Genera impulsos amb l'amplada rebuda
45
46   }
47 }
48
49

```

El programa que vam escriure, va ser: *motor brushless esc serial*.

L'objectiu d'aquest circuit i programa és fer variar la velocitat del motor des de zero fins al valor màxim. Vam intentar diferents mètodes que, finalment, van acabar funcionant.

El primer va ser, de la mateixa manera que abans, connectar dos ordinadors via Bluetooth, un emissor i l'altre receptor, i mitjançant l'aplicació RealTerm i amb el programa que havíem escrit amb l'Arduino Create, donar valors al motor perquè girés a diferents velocitats. Fins a l'endemà no vam aconseguir que funcionés. Va ser quan ens vam adonar que havíem d'enviar els valors amb caràcters ASCII i no numèricament.

L'altre mètode que vam emprar, havent-nos trobat amb aquell problema, va ser fer-ho tot directament des d'un mateix ordinador, és a dir, comunicar-nos amb el motor sense Bluetooth, enviant i rebent-ho tot des del mateix ordinador.

El funcionament és aquest: per Bluetooth s'envia un caràcter ASCII a la placa Arduino 101 i aquesta segons el valor rebut genera una sèrie d'impulsos (PWM) d'una durada compresa entre 1000 i 2000 μ s que van al regulador electrònic de velocitat (ESC). Aquest regula la velocitat del motor: els impulsos de 1000 μ s corresponen a motor aturat i els de 2000 μ s a la velocitat màxima.

Per motius de seguretat (les hèlixs són perilloses), quan s'alimenta l'ESC aquest ha de trobar impulsos de 1000 μ s si no, no permet engegar el motor. Així s'evita que es pugui engegar sense voler amb la velocitat màxima.

Necessitàvem saber quant pes era capaç d'aixecar el nostre dron i, per tant, havíem de calcular l'empenta dels motors.

Vam aixecar el motor sobre la base mitjançant un prisma de fusta. Vam posar el conjunt sobre una balança i vam anar mesurant la disminució de massa que indicava la balança amb les tres hèlixs diferents que teníem a diferents velocitats.

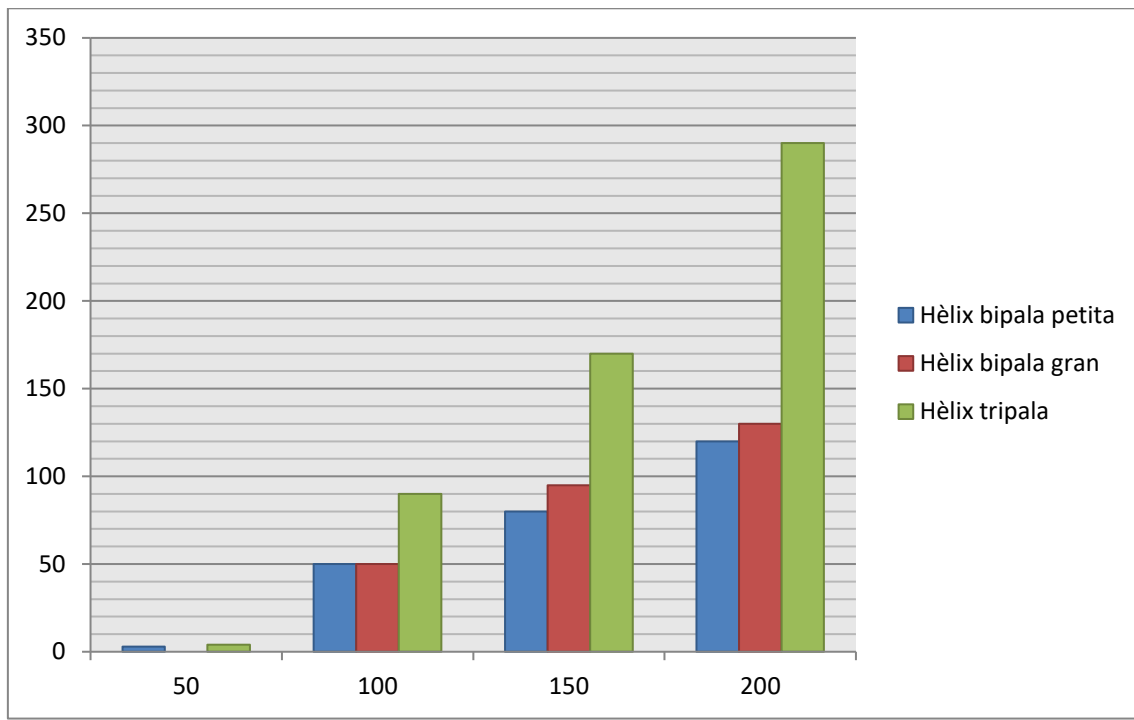
Com que teníem més d'un tipus de pales, vam provar-les totes. Vam enviar diferents valors mitjançant el RealTerm, mitjançant el programa *motor brushless esc serial* (utilitzat anteriorment):(50-100-150-200) i vam apuntar el pes que ens indicava la bàscula, construint així una taula de valors per a cada hèlix i, finalment, un gràfic.



Imatge 25: Mesura de l'empenta del motor.

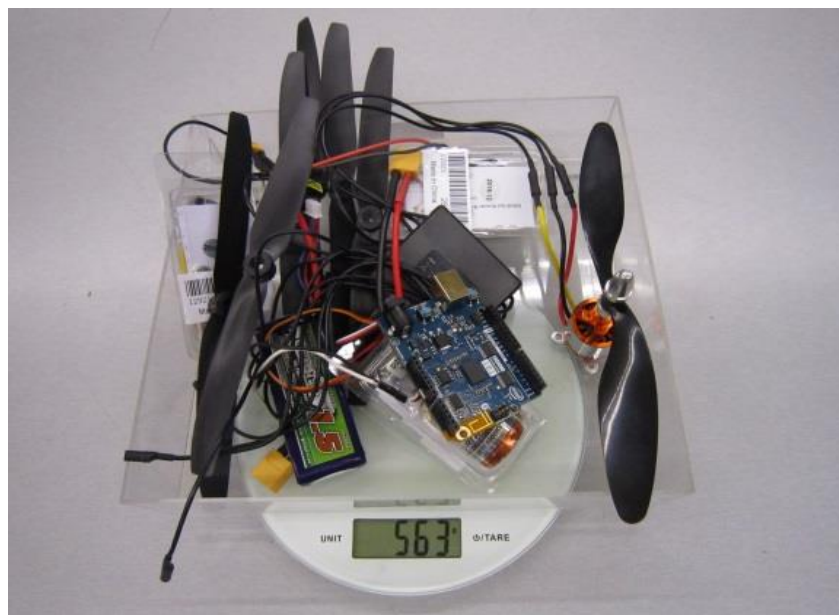
Hèlix bipala petita	
Valors (bytes)	Massa (grams)
50	3
100	50
150	80
200	120
Hèlix bipala gran	
Valors	Massa
50	0
100	50
150	95
200	130
Hèlix tripala	
Valors	Massa
50	4
100	90
150	170
200	290

Gràfic 26: Gràfic de l'empenta del motor amb diferents hèlixs.



Veient els resultats a la gràfica, vam triar les hèlixs tripala perquè tenien una gran empenta, en comparació a les altres, i perquè ens permetria afegir-hi una mica més de pes al dron, si s'esqueia.

Un cop calculada l'empenta del motor i, en conseqüència, l'empenta total, vam col·locar tots els components del dron, a excepció del xassís, en una balança. Com es pot veure el pes total és de 563 grams. Finalment, ens disposarem a realitzar el muntatge del dron.



Imatge 9: Mesura del pes dels components.

8.5 Dia 4

El primer xassís del dron consta de:

- Barres de fibra de carboni.
- Discs DVD o CD.
- Caragols blancs de niló per a estalviar pes.
- Femelles de niló.



Imatge 28: Femella de niló.
<http://www.shoptronica.com/tornilleria-especial-tuercas/2578-tornillos-de-nylon-m2.html>



Imatge 29: Discs DVD.
<https://www.theodysseyonline.com/5-reasons-will-always-use-cds>



Imatge 30: Barres de fibra de carboni.
<http://www.nauticexpo.es/prod/tech/product-21624-111858.html>



Imatge 31: Caragols de niló.
<http://www.shoptronica.com/tornilleria-especial-tuercas/2578-tornillos-de-nylon-m2.html>

Vam ajuntar dues barres de fibra de carboni en X, al mig d'aquestes, tres CD (un a sobre de l'altre) per a donar rigidesa al xassís i al mateix temps tenir plataformes per a posar-hi la placa d'Arduino, l'ESC i la bateria. Per a poder unir els CD amb la fibra mitjançant els caragols, s'hi van haver de fer prèviament uns forats amb el trepant de columna.



Imatge 32: Foradant els discs.



Imatge 33: Primer xassís del dron.

A continuació, vam col·locar els quatre motors als extrems de cada barra units a aquesta mitjançant caragols i femelles de niló, l'ESC i la placa Arduino superposades unides per sobre del CD per un tros de velcro adhesiu i, finalment, la bateria enganxada per sota del CD també amb velcro.

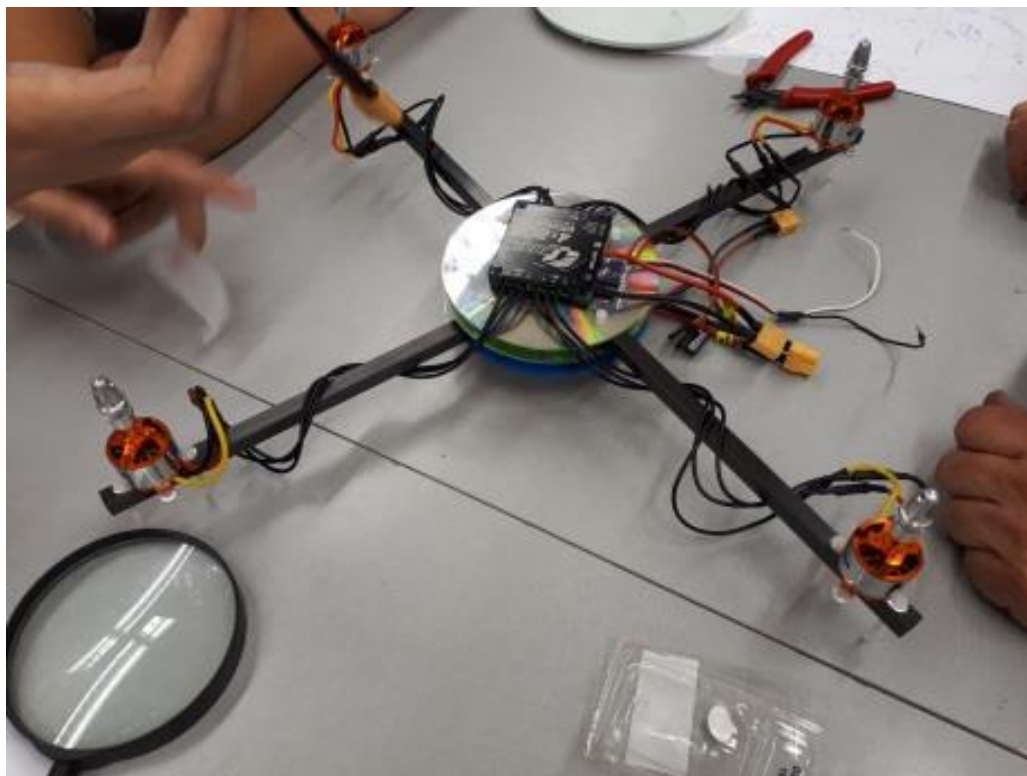
Es va tenir cura de centrar cada component al seu lloc. La ubicació de cada peça té la seva explicació:

- La bateria es troba situada a sota dels CD, així fa baixar el centre de masses i facilita l'extracció i la inserció d'aquesta.
- La placa controladora es troba sota la placa Arduino perquè no l'hem de manipular i, per tant, pot estar obstruïda.
- La placa Arduino està sobre la placa controladora per a poder-la connectar fàcilment amb l'ordinador mitjançant un cable USB.

Imatge 34: Veta adherent o velcro. <https://www.colourbox.de/bild/haken-klettverschluss-aka-klettverschluss-in-nahaufnahme-bild-1310843>

Per acabar la construcció del dron, vam enrotllar minuciosament els cables dels motors al llarg de les barres i vam afegir-hi dos LEDs, un de blanc al davant per a comprovar la connexió Bluetooth i un de roig al darrere per a saber quina era la part del davant i quina la del darrere.

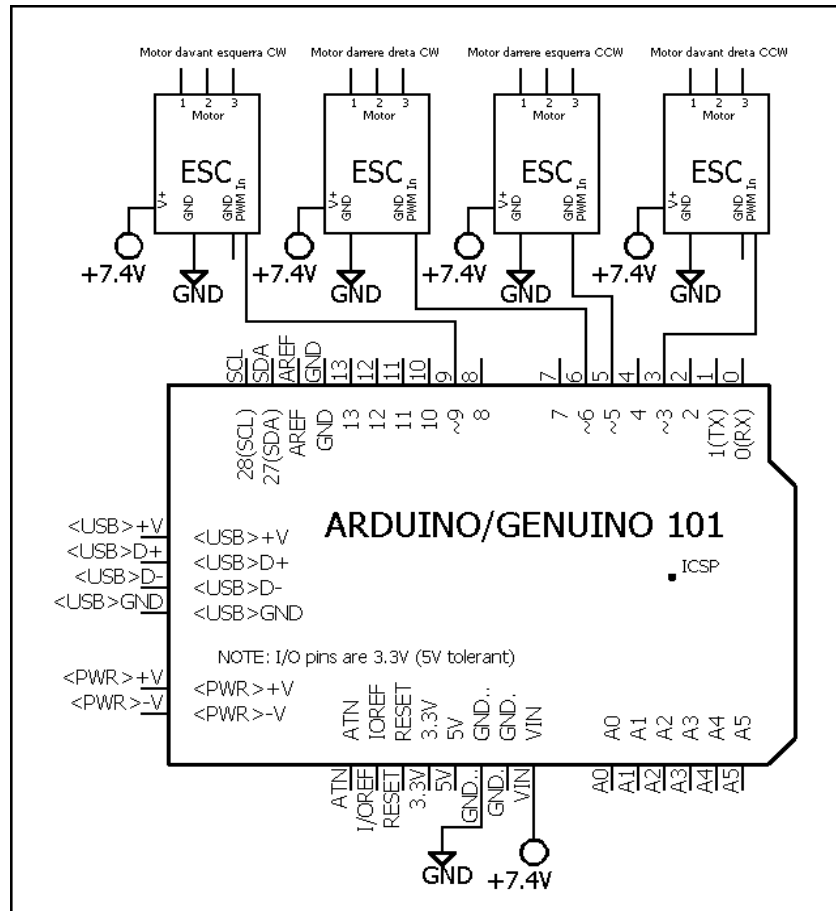
Com es pot apreciar a la imatge n°36, s'han afegit etiquetes adhesives a diferents parts del dron amb l'objectiu de reconèixer cada motor i saber on s'ha de connectar i en quin sentit ha de girar.



Imatge 35: Muntatge del dron.



Imatge 36: Dron acabat.



Imatge 37: Esquema de connexions dels motors, l'ESC i la placa Arduino-101.

Si un motor gira al revés només cal intercanviar les connexions d'un parell de fils qualsevol dels tres que té cada motor.

Realitzant proves, vam comprovar que el dron es començava a enlairar quan els impulsos tenien una amplada de 1400 μ s, si les hèlixs eren tripala i a partir de 1520 μ s si les hèlixs eren bipala. Sabent això, vam escriure un programa per a veure si el dron s'aixecava i, en definitiva, comprovar si havíem fet bé els càlculs fins aquell moment.

De fet, el dron s'elevà, però malauradament es va estavellar contra la paret; trencant així una barra de fibra de carboni i una hèlix. Després, un cop arreglat el desperfecte i, per poder seguir realitzant proves, vam agafar el dron amb dues cordes lligades a una taula per tal que no es tornés a descontrolar.

El programa utilitzat es deia: *motor 4 brushless*.

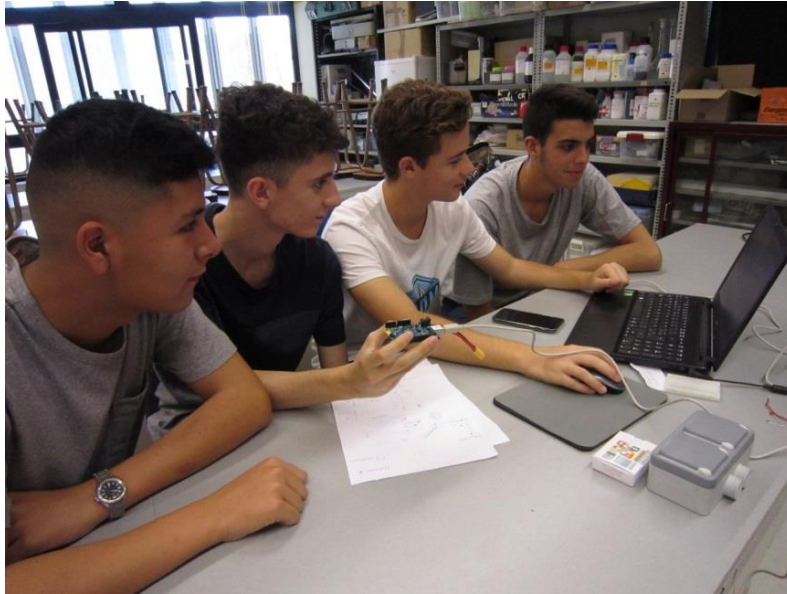
```

1  /* Important: la placa d'Arduino ha d'estar funcionant abans de connectar la bateria i
2  un impuls de 1000 microsegons ha d'estar actiu al control del Motor.
3  amb 1000 microsegons el motor està aturat. Amb 1200 s'engega i amb 2000 agafa la
4  velocitat màxima
5
6  *****
7  Si s'empra el RealTerm, el valor dels impulsos s'ha d'enviar en format ASCII i no
8  en format nombre
9  *****
10 */
11 #include<Servo.h> //Afegeix la biblioteca servo
12 Servo Motor1; //Crea un servo de nom Motor1
13 Servo Motor2;
14 Servo Motor3;
15 Servo Motor4;
16 int vel = 1000; //Amplada dels impulsos inicials en microsegons (motor aturat)
17
18 void setup()
19 {
20   pinMode(LED_BUILTIN, OUTPUT);
21   //Asigna un pin al Motor
22   Motor1.attach(5, 1000, 2000);
23   Motor2.attach(6, 1000, 2000);
24   Motor3.attach(3, 1000, 2000);
25   Motor4.attach(9, 1000, 2000);
26
27   //Activa el Motor. Pot dependre del Motor
28   Motor1.writeMicroseconds(vel); //1000 microsegons
29   Motor2.writeMicroseconds(vel);
30   Motor3.writeMicroseconds(vel);
31   Motor4.writeMicroseconds(vel);
32
33   delay(2000); //Un retard de 2 segundos abans d'emprar el motor
34   //Ara ja es podria connectar la bateria
35
36   //Inicia el port sèrie
37   Serial.begin(9600);
38   Serial.setTimeout(10);
39
40 }
41
42 void loop()
43 {
44   if(Serial.available() >= 1)
45   {
46     vel = Serial.parseInt(); //Llegeix una cadena ASCII pel port sèrie i la
47     transforma en un enter
48     digitalWrite(LED_BUILTIN, HIGH); //Encèn el LED incorporat per a senyalar que
49     s'ha rebut
50     //algun caracter pel port sèrie
51     delay(300); //Està encès 0.3 segons
52     digitalWrite(LED_BUILTIN, LOW); //apaga el LED
53     Serial.println(vel); //Per a comprobar el valor rebut
54     Motor1.writeMicroseconds(vel); //Genera impulsos amb l'amplada rebuda
55     Motor2.writeMicroseconds(vel);
56     Motor3.writeMicroseconds(vel);
57     Motor4.writeMicroseconds(vel);
58
59   }
60 }
61 }
62 }

```

8.6 Dia 5

Veient que el dron se'ns estavellà contra la paret, pensàrem que alguna cosa havíem d'estar fent malament. Vam comprovar els eixos (ax, ay, az) de l'acceleròmetre que portava incorporat la placa Arduino, i ens vam adonar que estaven invertits en relació a la documentació de l'Arduino.



Imatge 38: Comprovant les lectures de l'acceleròmetre

```

1  /*
2
3  */
4  #include "CurieIMU.h" //Incluye libreria para acelerometro
5
6  void setup() {
7      Serial.begin (9600);
8      CurieIMU.begin();
9      CurieIMU.setAccelerometerRange(2);
10
11 }
12 void loop() {
13
14     //Acelerometro
15     int axRaw, ayRaw, azRaw; //Valores acelerometro
16     float ax, ay, az;
17
18     CurieIMU.readAccelerometer(axRaw, ayRaw, azRaw);
19     ax=axRaw*2/32768;
20     ay=ayRaw*2/32768;
21     az=azRaw*2/32768;
22
23     Serial.print(ax);
24     Serial.print("\t");
25     Serial.print(ay);
26     Serial.print("\t");
27     Serial.print(az);
28     Serial.print("\t");
29     Serial.println();
30     Serial.println();
31     delay(2000);
32 }
33

```

El programa emprat, s'anomenà: *TR_Accelerometer*.

Un cop teníem aclarit el tema de l'acceleròmetre, només calia continuar el programa que vam començar per la primera elevació del dron, és a dir, la versió definitiva del programa.

Per començar, havíem de delimitar les ordres que enviaríem al dron per a poder desplaçar-se, les quals són:

Moviment	Número assignat
Aturat	0
Augmenta velocitat (↑)	1
Minva velocitat (↓)	2
Dreta	3
Esquerra	4
Endavant	5
Endarrere	6
Rotació dreta	7
Rotació esquerra	8

Després, basant-nos en un exemple, vam acabar de programar la versió definitiva, *TR Dron Definitiu*, la qual es mostra a continuació.

```

1  /*El quadricòpter s'ha dissenyat per a volar en X.
2  Motors vistos des de dalt:
3  Devant:
4  esquerra CW (p9 marró) 4-----3 dreta CCW (p3 blanc)
5  Darrera:
6  esquerra CCW (p5 roig) 1-----2 dreta CW (p6 taronja)
7
8  Control per BLE mitjançant una altra placa Arduino 101 que actua com a "Central"
9
10 *Copyright (c) 2016 Intel Corporation. All rights reserved.
11 * See the bottom of this file for the license terms.
12 */
13
14 /*
15 * Sketch: led.ino
16 *
17 * Description:
18 * This is a Peripheral sketch that works with a connected Central.
19 * It allows the Central to write a value and set/reset the led
20 * accordingly.
21 */
22 #include "CurieIMU.h"
23 #include <CurieBLE.h>
24 #include <Servo.h>
25 //introduim les biblioteques
26
27 BLEService ledService("19B10000-E8F2-537E-4F6C-D104768A1214"); // BLE LED Service
28
29 // BLE LED Switch Characteristic - custom 128-bit UUID, read and writable by central
30 BLEUnsignedCharCharacteristic
31 switchCharacteristic("19B10001-E8F2-537E-4F6C-D104768A1214", BLERead | BLEWrite);
32
33 Servo motor1;
34 Servo motor2;
35 Servo motor3;
36 Servo motor4;
37 //introduim els motors
38
39 int impulsMin = 1000;
40 int impulsMax = 2000;
41 //variables d'impuls mínim i màxim
42
43 int vel=1000;
44 int vell;
45 int vel2;
46 int vel3;
47 int vel4;
48 //la variable vel afecta a tots els motors; les altres a cada motor per separat
49
50 int LedPinBlanc = 2;
51 int LedPinVermell = 12;
52 //pin on estan connectats els leds
53
54
55 void setup() {
56   pinMode(LedPinBlanc, OUTPUT);
57   pinMode(LedPinVermell, OUTPUT);
58   //es diu que els led actuen com a sortida per cadascun dels pins
59
60   digitalWrite(LedPinBlanc, HIGH);
61   //encén el led blanc
62
63   Serial.begin(9600);
64
65   //BLUETOOTH
66   // begin initialization
67   BLE.begin();
68
69   // set advertised local name and service UUID:
70   BLE.setLocalName("TR_DRONE");// qualsevol nom fins 8 caracters
71   BLE.setAdvertisedService(ledService);
72

```

```

73 // add the characteristic to the service
74 ledService.addCharacteristic(switchCharacteristic);
75
76 // add service
77 BLE.addService(ledService);
78
79 // set the initial value for the characteristic:
80 switchCharacteristic.setValue(0); // valor inicial. Ha de ser un valor que no
    molesti.
81
82 // start advertising
83 BLE.advertise();
84
85 Serial.println("BLE TR_DRONE Peripheral");
86
87 //ACCELEROMETRE
88 Serial.println("Initializing IMU service");
89 CurieIMU.begin();
90 CurieIMU.setAccelerometerRange(2);
91 //configuració del accelerometre
92
93 //MOTORS
94 motor1.attach(5, impulsMin, impulsMax);
95 motor2.attach(6, impulsMin, impulsMax);
96 motor3.attach(3, impulsMin, impulsMax);
97 motor4.attach(9, impulsMin, impulsMax);
98 //configuració individual de cada motor
99 motor1.write(vel);
100 motor2.write(vel);
101 motor3.write(vel);
102 motor4.write(vel);
103 delay(2000);
104
105 } // Fi del setup()
106
107
108
109
110 void loop() {
111 // listen for BLE peripherals to connect:
112 BLEDevice central = BLE.central();
113
114
115
116 // if a central is connected to peripheral:
117 if (central){
118
119
120 Serial.print("Connected to central: ");
121 // print the central's MAC address:
122 Serial.println(central.address());
123
124 // while the central is still connected to peripheral:
125 while (central.connected()) {
126 digitalWrite(LedPinVermell, HIGH);
127 //encén el led vermell mentre estigui connectat
128
129 //ACCELEROMETRE
130 int axRaw, ayRaw, azRaw;
131 float ax, ay, az;
132
133 CurieIMU.readAccelerometer(axRaw, ayRaw, azRaw);
134
135 ax = convertRawAcceleration(axRaw);
136 ay = convertRawAcceleration(ayRaw);
137 az = convertRawAcceleration(azRaw);
138
139 //*****Estabilització *****
140
141 //Es manté horitzontal
142 if (ax>-0.05 && ax<0.05 && ay>-0.05 && ay<0.05 && az>0.95 && az<1.05){
143 motor1.write(vel);
144 motor2.write(vel);

```

```

145     motor3.write (vel);
146     motor4.write (vel);
147 }
148
149 //Corregeix si puja de l'esquerra i baixa de la dreta
150 if(ax>-0.5 && ax<0.05 && ay>0.05 && ay<1.0 && az>0.0){
151     vel2=vel+25;
152     vel3=vel+25;
153     motor2.write (vel2);
154     motor3.write (vel3);
155 }
156
157 //Corregeix si puja de la dreta i baixa de l'esquerra
158 if(ax>-0.5 && ax<0.05 && ay>-1.0 && ay<0.05 && az>0.0){
159     vel1=vel+25;
160     vel4=vel+25;
161     motor1.write (vel1);
162     motor4.write (vel4);
163 }
164
165 //Corregeix si puja del davant i baixa del darrera
166 if(ax>0.05 && ax<1.0 && ay>-0.05 && ay<0.05 && az>0.0){
167     vel1=vel+25;
168     vel2=vel+25;
169     motor1.write (vel1);
170     motor2.write (vel2);
171 }
172
173 //Corregeix si puja del darrera i baixa del davant
174 if(ax>-1.0 && ax<-0.05 && ay>-0.05 && ay <0.05 && az>0.0){
175     vel3=vel+25;
176     vel4=vel+25;
177     motor1.write (vel3);
178     motor2.write (vel4);
179 }
180
181
182
183
184 //BLUETOOTH
185 if(switchCharacteristic.written() ) {
186
187     if(switchCharacteristic.value() == 0) { //Atura
188         Serial.println(switchCharacteristic.value());
189         vel=1000;
190         motor1.write (vel);
191         motor2.write (vel);
192         motor3.write (vel);
193         motor4.write (vel);
194         Serial.println("Aturat");
195     }
196
197     if(switchCharacteristic.value() == 1) { //Augmenta la velocitat
198         Serial.println(switchCharacteristic.value());
199         vel=1250;
200         vel=vel+25;
201         if(vel>2000) vel=2000;
202         motor1.write (vel);
203         motor2.write (vel);
204         motor3.write (vel);
205         motor4.write (vel);
206         Serial.println("Puja velocitat");
207     }
208
209     if(switchCharacteristic.value() == 2) { //Minva la velocitat
210         Serial.println(switchCharacteristic.value());
211         vel=vel-25;
212         if(vel<1000) vel=1000;
213         motor1.write (vel);
214         motor2.write (vel);
215         motor3.write (vel);
216         motor4.write (vel);
217         Serial.println("Baixa velocitat");

```

```

218     }
219
220     if (switchCharacteristic.value ()==3) { //Gira a la dreta
221         vel1=vel+50;
222         vel4=vel+50;
223         motor1.write (vel1); //Augmenta la velocitat dels motors de l'esquerra
224         motor4.write (vel2);
225         delay (200); //Gira durant aquest temps
226         motor1.write (vel); //Retorna a la velocitat anterior
227         motor4.write (vel);
228     }
229
230     if (switchCharacteristic.value ()==4) { //Gira a l'esquerra
231         vel2=vel+50;
232         vel3=vel+50;
233         motor2.write (vel2); //Augmenta la velocitat dels motors de la dreta
234         motor3.write (vel3);
235         delay (200); //Gira durant aquest temps
236         motor2.write (vel); //Retorna a la velocitat anterior
237         motor3.write (vel);
238     }
239
240     if (switchCharacteristic.value ()==5) { //Va cap endavant
241         vel1=vel+50;
242         vel2=vel+50;
243         motor1.write (vel1);
244         motor2.write (vel2);
245         delay (200);
246         motor1.write (vel);
247         motor2.write (vel);
248     }
249     if (switchCharacteristic.value ()==6) { //Va cap endarrere
250         vel3=vel+50;
251         vel4=vel+50;
252         motor3.write (vel3);
253         motor4.write (vel4);
254         delay (200);
255         motor3.write (vel);
256         motor4.write (vel);
257     }
258
259     if (switchCharacteristic.value ()==7) { //Rotació esquerra
260         vel2=vel+50;
261         vel4=vel+50;
262         motor2.write (vel2);
263         motor4.write (vel4);
264         delay (200);
265         motor2.write (vel);
266         motor4.write (vel);
267     }
268
269     if (switchCharacteristic.value ()==8) { // Rotació dreta
270         vel1=vel+50;
271         vel3=vel+50;
272         motor1.write (vel1);
273         motor3.write (vel3);
274         delay (200);
275         motor1.write (vel);
276         motor3.write (vel);
277     }
278     }
279     }
280
281
282
283
284
285 } // F1 de if (central)
286
287 digitalWrite (ledPinVermell, LOW);
288
289 // when the central disconnects, print it out:
290 Serial.print (F ("Disconnected from central: "));

```

```
291     Serial.println(central.address());
292
293     } // Fi de loop()
294
295     float convertRawAcceleration(int aRaw){
296         float a = (aRaw*2.0)/32768.0;
297         return a;
298     }
299
300
301     /*
302     Copyright (c) 2016 Intel Corporation. All rights reserved.
303
304     This library is free software; you can redistribute it and/or
305     modify it under the terms of the GNU Lesser General Public
306     License as published by the Free Software Foundation; either
307     version 2.1 of the License, or (at your option) any later version.
308
309     This library is distributed in the hope that it will be useful,
310     but WITHOUT ANY WARRANTY; without even the implied warranty of
311     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
312     Lesser General Public License for more details.
313
314     You should have received a copy of the GNU Lesser General Public
315     License along with this library; if not, write to the Free Software
316     Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
317     */
318
```

8.7 Dia 6

Quan vam tenir enllestit el programa, vam anar al pati de l'institut per a fer volar el dron. Desgraciadament, el dron no s'estabilitzava horitzontalment, en comptes d'això, per poc que s'aixequés, girava i s'estavellava contra el terra.

Encara que vam intentar canviar alguns paràmetres del programa per tal que funcionés, no vam aconseguir que s'estabilitzés de la manera desitjada.

8.8 Dia 7

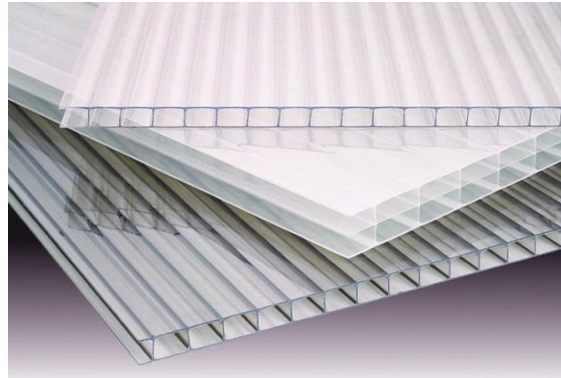
Vam pensar que els problemes eren deguts a que el xassís no era prou rígid i feia que l'acceleròmetre captés vibracions massa fortes per a donar una lectura correcta. Vam decidir fer un xassís nou.

Aquest segon xassís diferia completament de l'antic, i estava format per:

- Una placa de policarbonat cel·lular (10 mm de gruix).
- Dos perfils d'alumini en L (45 cm de llargària).
- Els altres components principals anteriorment esmentats: motors, ESC, placa Arduino, hèlixs.

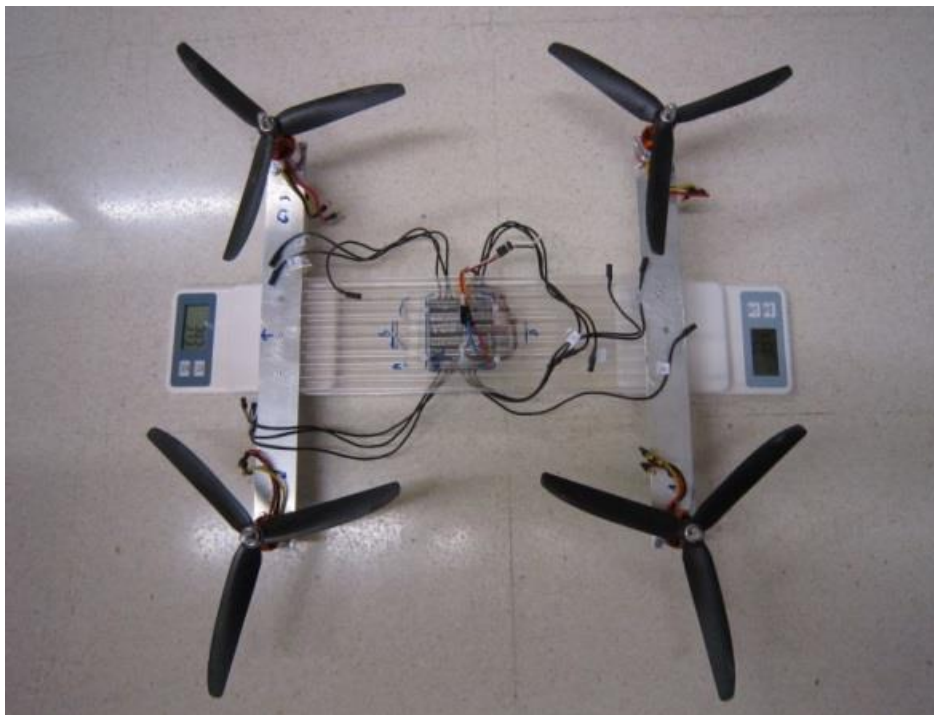


Imatge 39: Perfil d'alumini en L.
<http://www.ebay.de/itm/Winkelstahl-verzinkt-ungleichschenkelig-L-Winkel-500-1500mm-30x20x3-100x50x10mm-/191669874604>



Imatge 40: Policarbonat cel·lular.
<http://www.plastiastur.com/>

Vam tallar un rectangle (10 x 35'8cm) de policarbonat cel·lular i el vam unir amb els dos perfils d'alumini amb caragols, de manera equidistant. A continuació, vam posar-lo sobre dues balances per tal de trobar un equilibri de pes i vam marcar amb un retolador el lloc on aniria cada peça.



Imatge 41: Segon dron.

Aquesta vegada vam organitzar de manera diferent les peces: la placa Arduino aniria a dalt del policarbonat cel·lular damunt d'una esponja (per a reduir vibracions) i sota hi anirien l'ESC i la bateria. Vam tornar a fixar els motors amb caragols de niló però aviat els vam haver de canviar per caragols d'acer perquè no resistien els xocs amb terra. Tots els motors estaven a una distància de 45 cm els uns dels altres. Finalment, hi vam posar els dos LEDs i vam haver de fer retocs al programa.



Imatge 42: Muntatge del dron.



Imatge 43: Segon dron acabat.

8.9 Dia 8

No era gaire pràctic controlar el dron des de l'ordinador, així que vam fer un comandament a distància portàtil format per:

- Una placa Arduino 101.
- Una bateria de líti.
- Una capsa de plàstic.
- Teclat de membrana.
- Un LED.

A continuació es detallen les connexions del teclat:



Imatge 44: Teclat.

<http://www.instructables.com/id/Connecting-a-4-x-4-Membrane-Keypad-to-an-Arduino/>

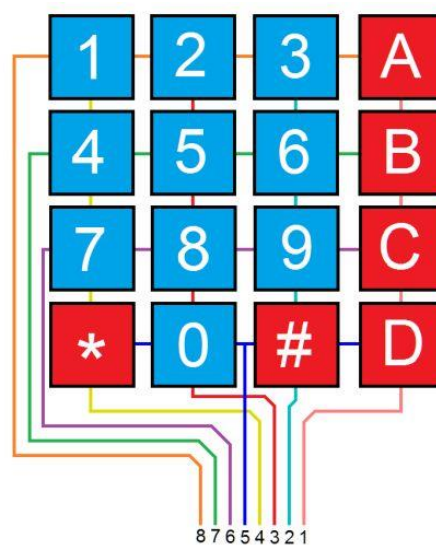
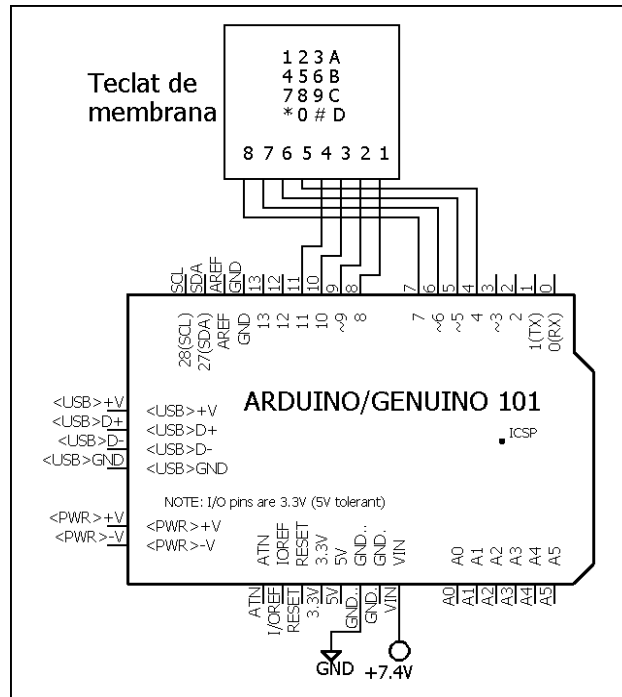


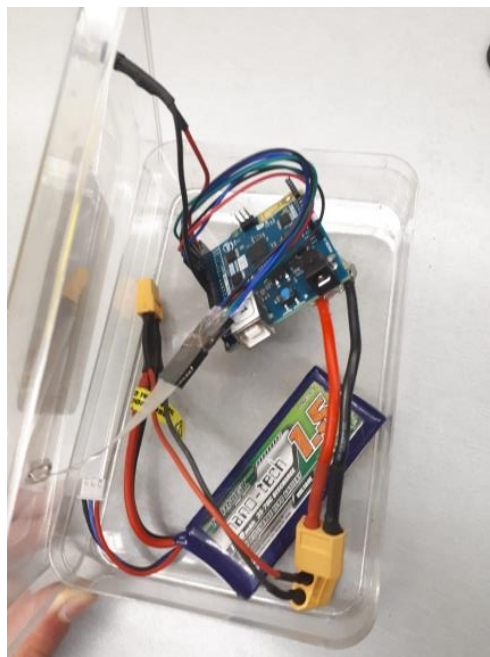
Figure 1: Matrix Keypad Connections



Imatge 45: Connexions del teclat.
<http://www.instructables.com/id/Connecting-a-4-x-4-Membrane-Keypad-to-an-Arduino/>

Dins de la capsula de plàstic vam introduir la bateria i la placa Arduino, per fora el teclat i el LED en un forat fet a la capsula. La funció del LED era indicar si s'havia pogut establir la connexió Bluetooth entre les dues plaques Arduino.

Les ordres que es podien enviar per a controlar el dron eren les mateixes que es van detallar quan es construïa la versió definitiva del primer model. (*TR Dron Definitiu*)



Imatge 46: Muntant el comandament a distància.



Imatge 47: Comandament a distància acabat.

A continuació es mostren els programes del comandament a distància i el del programa definitiu, respectivament: *Customkeypad teclat membrana* i *Control central teclat membrana*

```

1  /* @file CustomKeypad.pde
2  || @version 1.0
3  || @author Alexander Brevig
4  || @contact alexanderbrevig@gmail.com
5  ||
6  || @description
7  || | Demonstrates changing the keypad size and key values.
8  || #
9
10 Teclat de membrana: es comença a comptar per l'esquerra: 8----1
11 de 1'1 al 4 són columnes. Començant de dreta a esquerra i de dalt a baix:
12 1) ABCD; 2) 3, 6, 9, #; 3) 2, 5, 8, 0; 4) 1, 4, 7, *
13 del 5 al 8 són files. Començant de baix a dalt i llegint d'esquerra a dreta:
14 5) *, 0, #, D; 6) 7, 8, 9, C; 7) 4, 5, 6, B; 8) 1, 2, 3, A
15
16
17 */
18 #include <Keypad.h>
19
20
21
22 const byte ROWS = 4; //four rows
23 const byte COLS = 4; //four columns
24 //define the cymbols on the buttons of the keypads
25 char hexaKeys[ROWS][COLS] = { // Així és tal com es veu el teclat
26   {'1', '2', '3', 'A'},
27   {'4', '5', '6', 'B'},
28   {'7', '8', '9', 'C'},
29   {'*', '0', '#', 'D'}
30 };
31 byte rowPins[ROWS] = {11, 10, 9, 8}; //connect to the row pinouts of the keypad
32 byte colPins[COLS] = {7, 6, 5, 4}; //connect to the column pinouts of the keypad
33
34 //initialize an instance of class NewKeypad
35 Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
36
37
38 int customKey; // Si es defineix com a int envia: 49, 50, 51, 65 els codis ASCII.
39 //Si es defineix com a char dóna: 1, 2, 3, A... els caràcters.
40 // Si es defineix com a byte dóna: 49, 50, 51, 65... els codis ASCII.
41
42
43 void setup(){
44   Serial.begin(9600);
45 }
46
47 void loop(){
48   customKey = customKeypad.getKey();
49
50   if (customKey){
51     Serial.println(customKey); // Ho envia com a ASCII
52   }
53 }
54

```

```

1  /* Actua com a CENTRAL BLE i pot enviar qualsevol dels 16 caràcters del teclat
2
3  Teclat de membrana: es comença a comptar per l'esquerra: 8----1
4  de 1'1 al 4 són columnes. Començant de dreta a esquerra i de dalt a baix:
5  1) ABCD; 2) 3, 6, 9, #; 3) 2, 5, 8, 0; 4) 1, 4, 7, *
6  del 5 al 8 són files. Començant de baix a dalt i llegint d'esquerra a dreta:
7  5) *, 0, #, D; 6) 7, 8, 9, C; 7) 4, 5, 6, B; 8) 1, 2, 3, A
8
9
10
11
12  * Copyright (c) 2016 Intel Corporation. All rights reserved.
13  * See the bottom of this file for the license terms.
14  */
15
16  /*
17  * Sketch: LedControl.ino
18  *
19  * Description:
20  * This is a Central sketch that looks for a particular Service with a
21  * certain Characteristic from a Peripheral. Upon succesful discovery,
22  * it reads the state of a button and write that value to the
23  * Peripheral Characteristic.
24  *
25  * Notes:
26  *
27  * - Expected Peripheral Service: 19b10000-e8f2-537e-4f6c-d104768a1214
28  * - Expected Peripheral Characteristic: 19b10001-e8f2-537e-4f6c-d104768a1214
29  * - Expected Peripheral sketch:
30  *
31  */
32
33  #include <CurieBLE.h>
34  //aquí s'inclou la biblioteca del Bluetooth
35  #include <Keypad.h>
36
37  const byte ROWS = 4; //four rows
38  const byte COLS = 4; //four columns
39  //define the cymbols on the buttons of the keypads
40  char hexaKeys[ROWS][COLS] = { // Així és tal com es veu el teclat
41    {'1', '2', '3', 'A'},
42    {'4', '5', '6', 'B'},
43    {'7', '8', '9', 'C'},
44    {'*', '0', '#', 'D'}
45  };
46  byte rowPins[ROWS] = {7, 6, 5, 4}; //connect to the row pinouts of the keypad
47  byte colPins[COLS] = {11, 10, 9, 8}; //connect to the column pinouts of the keypad
48
49  //initialize an instance of class NewKeypad
50  Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
51
52  char customKey; // Si es defineix com a char dóna: 1, 2, 3, A... els caràcters
53  // si es defineix com a byte dóna: 49, 50, 51, 65... els codis ASCII.
54  //com que operem amb BLE(Bluetooth Low Energy), com a màxim podem enviar un byte
55
56  int LedIncorporat = 13;
57
58
59  void setup() {
60    Serial.begin(9600);
61    while(!Serial.available());
62
63    // initialize the BLE hardware
64    BLE.begin();
65
66    Serial.println("BLE Central - Led control");
67
68    // start scanning for peripherals
69    BLE.scanForUuid("19B10000-E8F2-537E-4F6C-D104768A1214");
70  } // Fi de setup()
71
72
73  //aquest paràgraf serveix per iniciar el Bluetooth i per a escanejar perifèrics

```

```

74
75 void loop() {
76
77     customKey = customKeypad.getKey();
78
79     // check if a peripheral has been discovered
80     BLEDevice peripheral = BLE.available();
81
82     if (peripheral) {
83         // discovered a peripheral, print out address, local name, and advertised service
84         Serial.print("Found ");
85         Serial.print(peripheral.address());
86         Serial.print(" ");
87         Serial.print(peripheral.localName());
88         Serial.print(" ");
89         Serial.print(peripheral.advertisedServiceUuid());
90         Serial.println();
91
92         // stop scanning
93         BLE.stopScan();
94
95         controlled(peripheral);
96
97         // peripheral disconnected, start scanning again
98         BLE.scanForUuid("19B10000-E8F2-537E-4F6C-D104768A1214");
99     } // Fi de peripheral()
100
101 //comprova si un perifèric ha estat connectat. si ho ha estat, imprimeix les dades
102 //d'aquest: l'adreça,
103 //el nom, i el servei anunciat
104 } // Fi de loop()
105
106
107
108 void controlled(BLEDevice peripheral) {
109     // connect to the peripheral
110     Serial.println("Connecting ...");
111
112     if (peripheral.connect()) {
113         Serial.println("Connected");
114     } else {
115         Serial.println("Failed to connect!");
116         return;
117     }
118     /*si s'esta connectant, imprimeix 'Connecting...'; si sha connectat,
119     imprimeix 'Connected', i si no, imprimeix 'Failed to connect!*/
120
121
122
123     Serial.println("Discovering attributes ...");
124     if (peripheral.discoverAttributes()) {
125         Serial.println("Attributes discovered");
126     } else {
127         Serial.println("Attribute discovery failed!");
128         peripheral.disconnect();
129         return;
130     }
131     // Descobrir els atributs del perifèric
132     BLECharacteristic ledCharacteristic =
133     peripheral.characteristic("19B10001-E8F2-537E-4F6C-D104768A1214");
134
135     if (!ledCharacteristic) {
136         Serial.println("Peripheral does not have LED characteristic!");
137         peripheral.disconnect();
138         return;
139     } else if (!ledCharacteristic.canWrite()) {
140         Serial.println("Peripheral does not have a writable LED characteristic!");
141         peripheral.disconnect();
142         return;
143     }
144     // recuperar les característiques del LED

```

```
145 while (peripheral.connected()) {
146     // mentre el perifèric està connectat
147     digitalWrite(LedIncorporat, HIGH);
148     if(customKey){
149         ledCharacteristic.writeByte(customKey);
150     }
151 } // Fi de while
152
153     Serial.println("Peripheral disconnected");
154     //si no està connectat, imprimeix Peripheral disconnected
155     digitalWrite(LedIncorporat, LOW);
156 } // Fi del procediment controlled
157
158 /*
159 Arduino BLE Central LED Control example
160 Copyright (c) 2016 Arduino LLC. All right reserved.
161
162 This library is free software; you can redistribute it and/or
163 modify it under the terms of the GNU Lesser General Public
164 License as published by the Free Software Foundation; either
165 version 2.1 of the License, or (at your option) any later version.
166
167 This library is distributed in the hope that it will be useful,
168 but WITHOUT ANY WARRANTY; without even the implied warranty of
169 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
170 Lesser General Public License for more details.
171
172 You should have received a copy of the GNU Lesser General Public
173 License along with this library; if not, write to the Free Software
174 Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
175 */
176
177
178
179
```

Un cop teníem enllestit el nou fuselatge i el nou programa vam fer volar el dron al pati de l'institut. Aquest cop va volar millor que abans però no acabava d'estabilitzar bé. Vam fer diferents canvis al programa però no van donar resultat. Finalment, se'ns va quedar atrapat a la cistella de bàsquet.



Imatge 48: Dron atrapat a la cistella de bàsquet.

Revisant el programa que ens havia servit de guia, vam arribar a la conclusió que era una enganyifa: només feia servir les lectures de l'acceleròmetre quan totes les fonts consultades deien que s'havien d'emprar conjuntament les dades de l'acceleròmetre i del giroscopi, fins i tot que eren més importants aquestes últimes. No era estrany que no aconseguíssim estabilitzar el dron.

Com que se'ns acabava el temps, vam decidir emprar una controladora de vol comercial per a substituir l'Arduino-101.

8.10 Dia 9

Es va comprar una nova controladora més potent, SP RACING F3 Acro. Aquesta, es pot programar per a ajustar diferents paràmetres mitjançant el programa CleanFLight, una aplicació gratuïta per Google Chrome.

8.11 Dia 10

Vam pensar de connectar la placa controladora amb l'Arduino per a aprofitar part del treball que havíem elaborat, però fent proves amb la controladora, ens vam adonar que no es podia comunicar amb la placa Arduino-101. En realitat no tenia cap sentit fer-ho, ja que no la podíem controlar per Bluetooth; la controladora estava dissenyada per a ser controlada per ràdio.

8.12 Dia 11

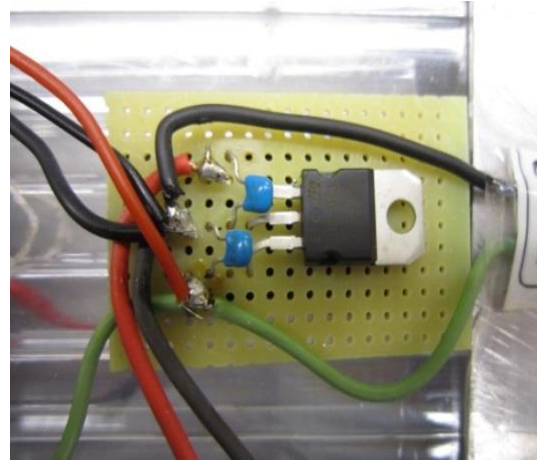
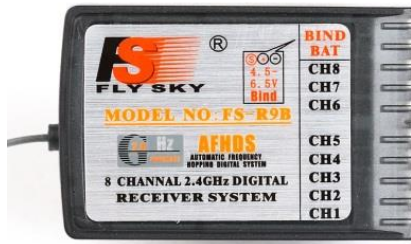
Vam haver de comprar una emissora de ràdio amb receptor que tampoc no els vam encertar a la primera.



Imatge 49: Nova placa controladora.
<http://www.canarias-stock.es/es/inicio/20-pro-sp-racing-regulador-de-vuelo-acro-6-dof-para-diy-mini-250-280-210-rc.html>



Imatge 50: Primera emissora FlySky.
<http://netwhole.ocnk.net/product/2290>



Imatge 51: Primera receptora FlySky.
<https://www.rcgroups.com/forums/showthread.php?2500523-Need-help-with-connecting-FlySky-FS-R9B-to-Arducopter>

Imatge 52: Regulador de tensió amb un circuit integrat 7805.

La bateria de liti donava un voltatge massa alt per a la controladora i el receptor, així que vam haver de muntar un circuit regulador que donés 5V a partir de la bateria.



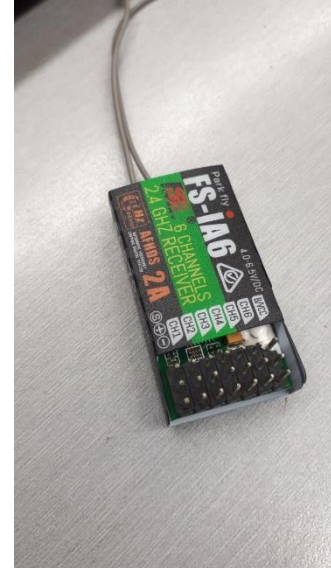
Imatge 53: Dron amb el receptor muntat i l'emissora de ràdio.

8.13 Dia 12

Desafortunadament, l'emissora que teníem no ens funcionava amb la placa controladora. Això implicava una altra emissora compatible. Per sort la vam poder canviar sense problemes.



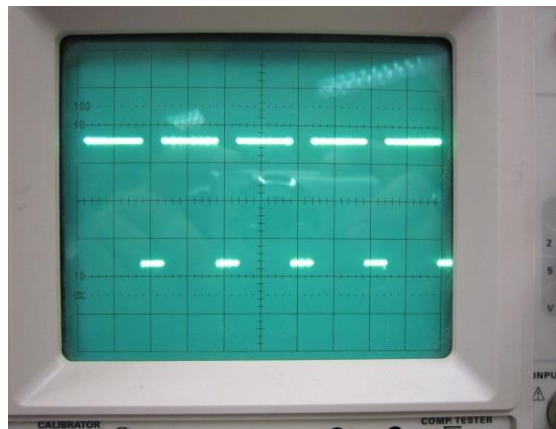
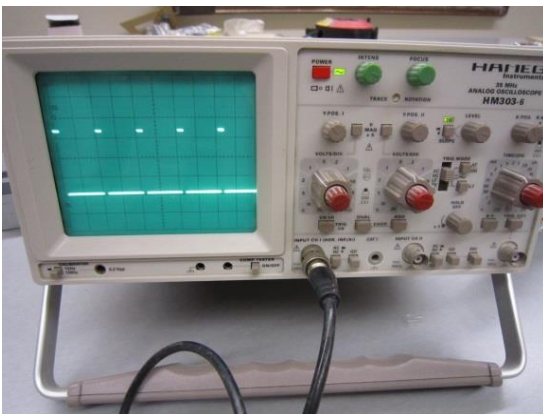
Imatge 54: Segona emissora de ràdio FlySky.



Imatge 55: Segon receptor de ràdio FlySky.

8.14 Dia 13

Encara no funcionava bé. Vam comprovar la modulació d'amplada d'impuls del receptor amb un oscil·loscopi i era correcta (imatges 56 i 57).



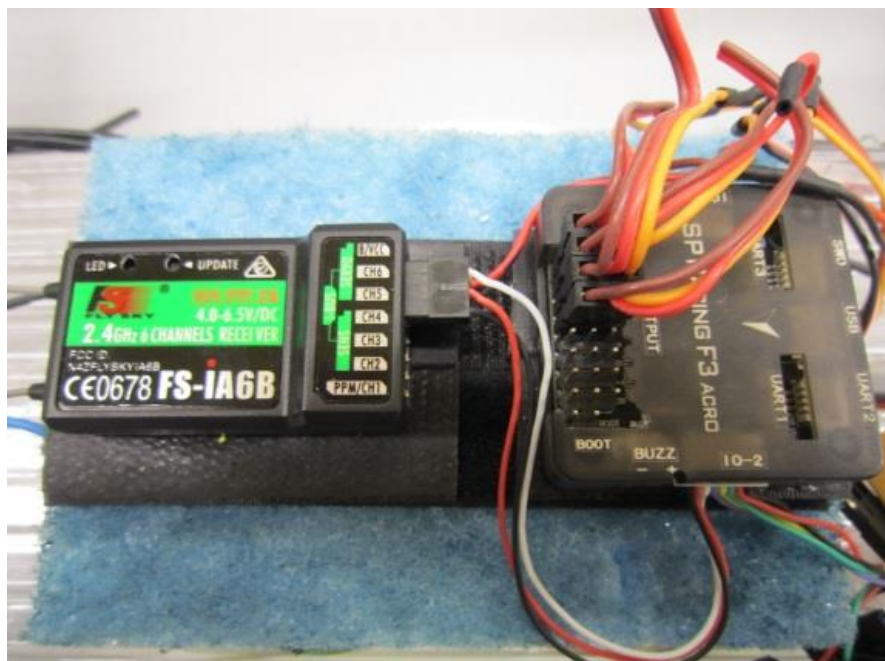
Imatges 56 i 57: Mesura de l'amplada dels impulsos generats pel receptor.

El receptor inclòs amb l'emissora només tenia sortida en paral·lel. La controladora de vol necessitava un receptor que tingués sortida en sèrie (port iBUS).



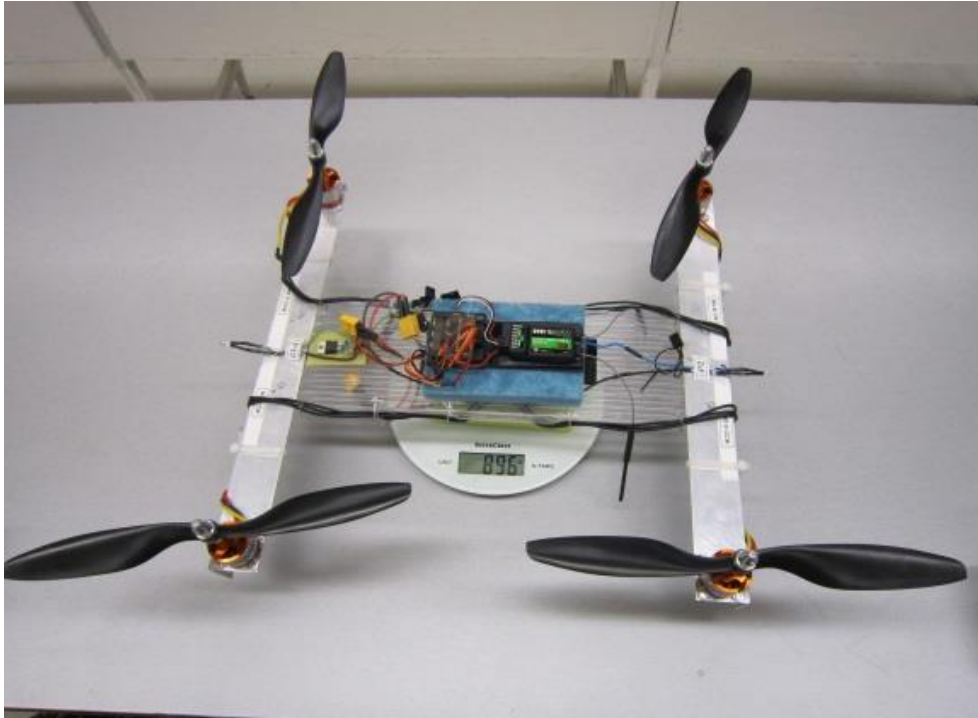
Imatge 58: Receptor definitiu amb sortides en sèrie i en paral·lel.

Amb aquest nou receptor funcionava però amb irregularitats. Buscant informació ens vam assabentar que calia “armar” el dron amb l'emissora.



Imatge 59: Connexió de la placa controladora amb el receptor.

Això vol dir que, per seguretat, els motors es bloquegen per evitar qualsevol tipus de danys i destrosses. Per tal de desbloquejar-los, cal moure cap avall i després, cap a la dreta la palanca (Esquerra). I per tal de tornar-lo a bloquejar el procediment invers, és a dir, cap avall i cap a la dreta.

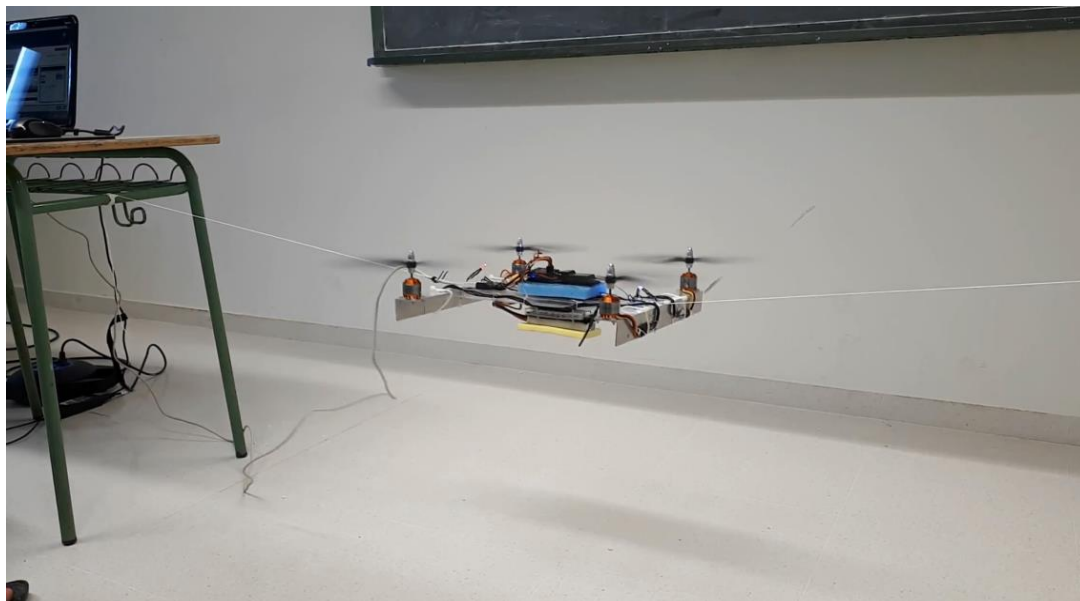


Imatge 60: Muntatge final del dron.

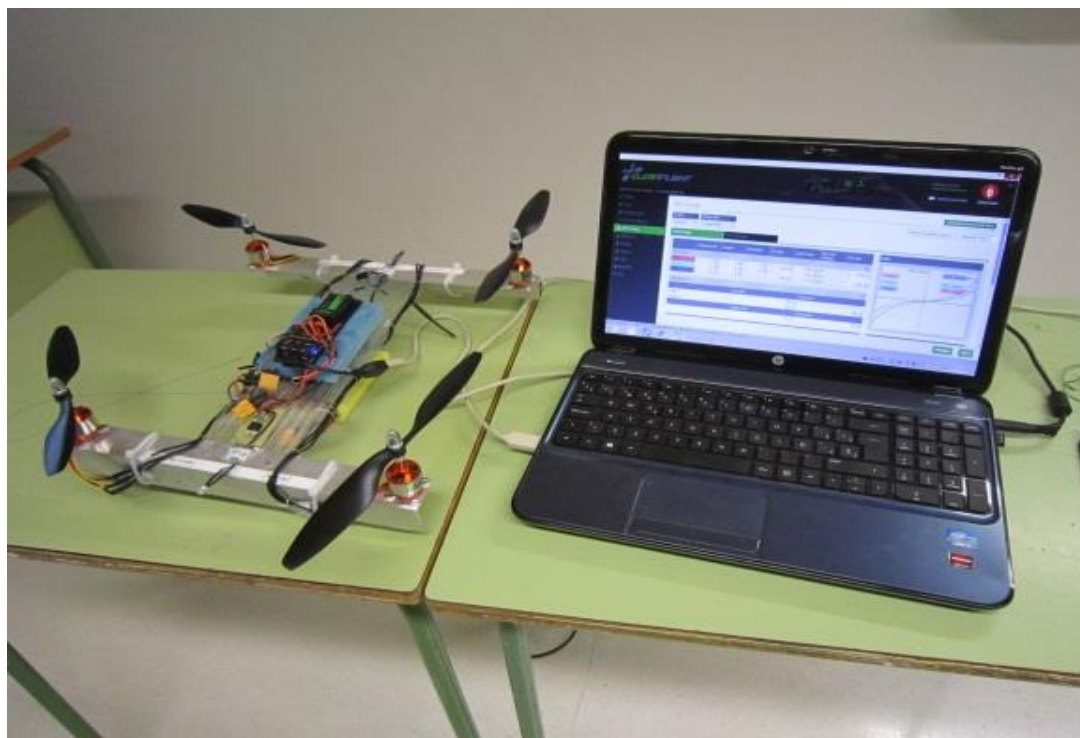
8.15 Dia 14

Un cop el dron ens funcionava adequadament, havíem d'ajustar els diferents paràmetres del PID (proporcional, integral i derivatiu) que serveixen per a la correcta estabilització del dron.

Per tal d'ajustar el PID vam agafar dues cordes, dues taules i el dron. Vam lligar les cordes a les taules i les cordes al dron de manera que les cordes quedessin més o menys tenses. D'aquesta manera podríem veure com el dron s'estabilitzava i, convenientment, anar canviant el PID mitjançant el programa CleanFlight. El PID més adequat va ser: (12,0,0)



Imatge 61: Dron agafat per cordes.



Imatge 62: Portàtil i dron.

8.16 Dia 15

Finalment, un cop teníem el PID ajustat, vam sortir al pati per volar el dron. Com esperàvem, aquest cop va volar estable. Vam provar de variar lleugerament els paràmetres del PID, però amb els paràmetres inicials ja anava bé.

Vam poder realitzar 4 vols. El primer dels vols va ser de prova, i no va arribar a molta altura. El segon, amb més confiança, ja va agafar més altura i, fins i tot, es va aconseguir aterrar amb bastant control. El tercer va arribar més alt que l'anterior, aterrant de nou sense cap xoc. Finalment, al quart i últim vol, per a evitar el xoc amb un llum del pati vam fer una maniobra equivocada reduint massa la velocitat dels motors. El dron es va estavellar contra el terra. Es va partir la placa de policarbonat, es van torçar els perfils d'alumini i es van trencar dues hèlixs bipala. Cal dir que controlar el dron des de l'emissora no és fàcil, necessita pràctica que encara no teníem. A més el dron que havíem fet no tenia magnetòmetre de manera que encara era més difícil de controlar (els drons sense magnetòmetre s'anomenen acrobàtics).



Imatge 63: Dron volant.



Imatge 64: Dron després d'estavellar-se.

8.17 Configuració i PID definitius

The screenshot shows the 'PID Tuning' interface. At the top, there are dropdowns for 'Profile' (set to 'Profile 1') and 'Rateprofile' (set to 'Rateprofile 1'). Below these are 'Reset all profile values' and 'Show all PIDs' buttons. The main area is divided into 'PID Settings' and 'Filter Settings'. The 'PID Settings' section contains a table for 'Basic/Acro' settings:

	Proportional	Integral	Derivative	RC Rate	Super Rate	Max Vel [deg/s]	RC Expo
ROLL	10	0	0	0,67	0,65	383	0,00
PITCH	13	0	0		0,70	447	
YAW	17	0		1,00	0,70	667	0,00

Below this table is the 'Angle/Horizon' section with settings for Strength, Transition, Angle Limit, and Sensitivity. On the right, the 'Rates' graph shows three curves for ROLL, PITCH, and YAW. The YAW curve is the highest, reaching 667 deg/s. The PITCH curve is in the middle, reaching 447 deg/s. The ROLL curve is the lowest, reaching 383 deg/s. A horizontal line is drawn at 800 deg/s. At the bottom right, there are 'Refresh' and 'Save' buttons.

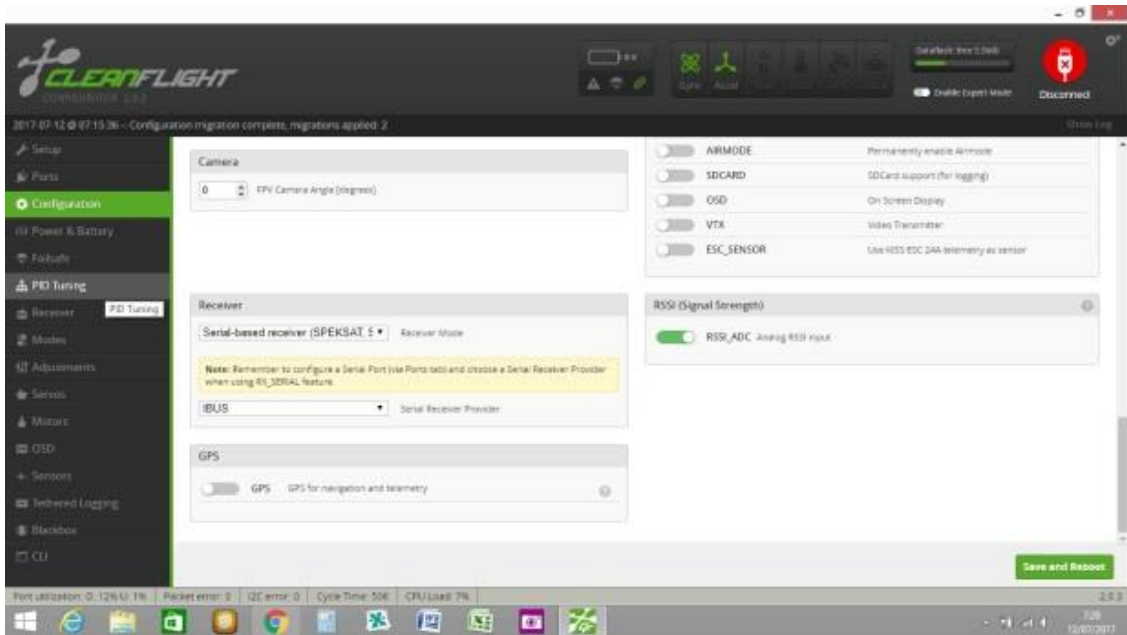
Imatge 65: PID definitiu.

Després de volar el dron i canviar el PID per a l'estabilitat completa, la configuració d'aquest va restar de la següent manera.

The screenshot shows the 'Configuration' interface. At the top, there is a 'Note' about feature combinations. The 'Mixer' section shows a 'Quad X' configuration with four motors labeled 1, 2, 3, and 4. The 'ESC/Motor Features' section shows settings for PWM (set to 480), MOTOR_STOP (checked), and various throttle and command values. At the bottom right, there is a 'Save and Reboot' button.

Imatge 66: Configuració definitiva.

En aquest apartat es veu la distribució dels motors (4,2,3,1) i les diferents opcions que s'escolliren: PWM (*Pulse-width modulation*) i la seva freqüència (490) i el MOTOR STOP, és a dir, que els motors no gressin quan estigués armat.



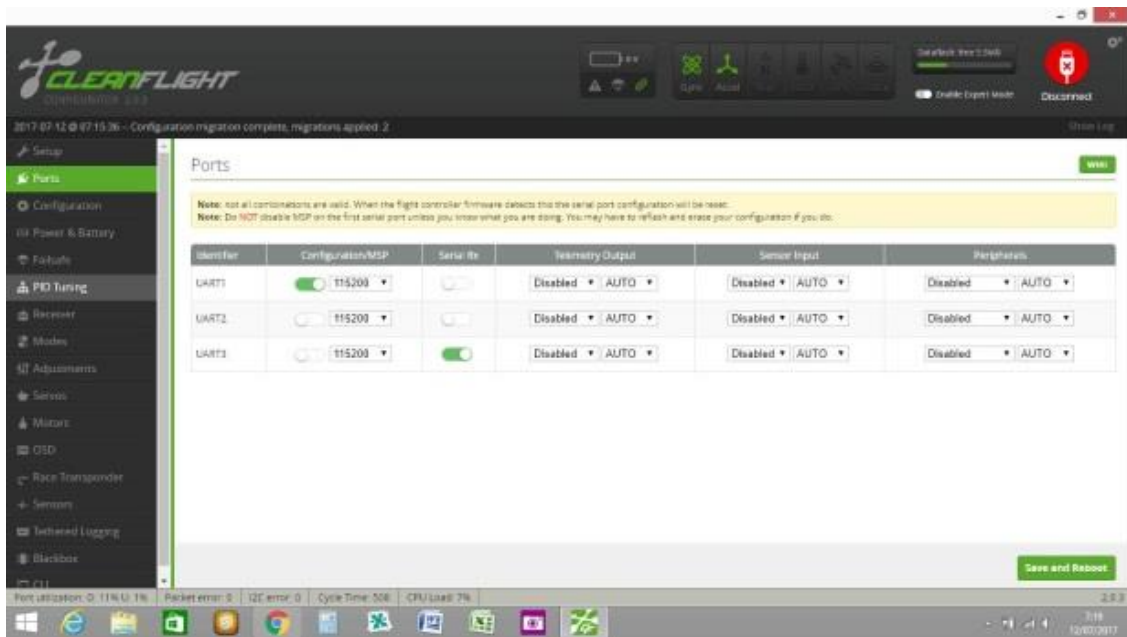
Imatge 67: Configuració definitiva 2.

S'aprecia la tria del IBUS



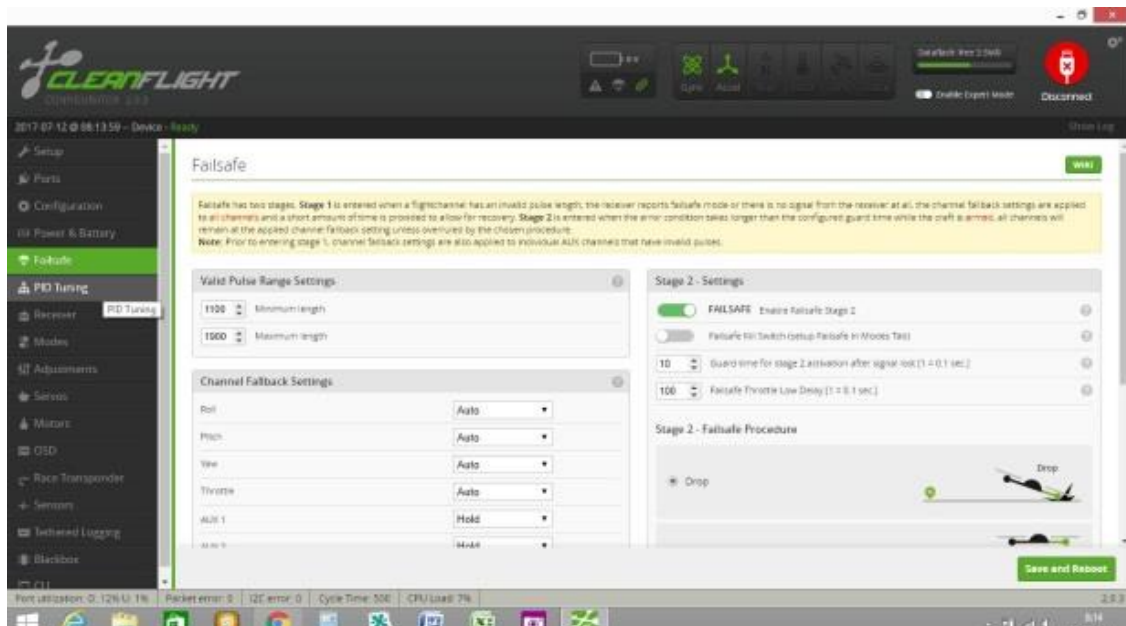
Imatge 68: Configuració definitiva 3.

En aquest apartat de la programació, es podia calibrar l'acceleròmetre deixant el dron sobre una superfície plana i sempre tenint en compte la direcció de la fletxa verda.



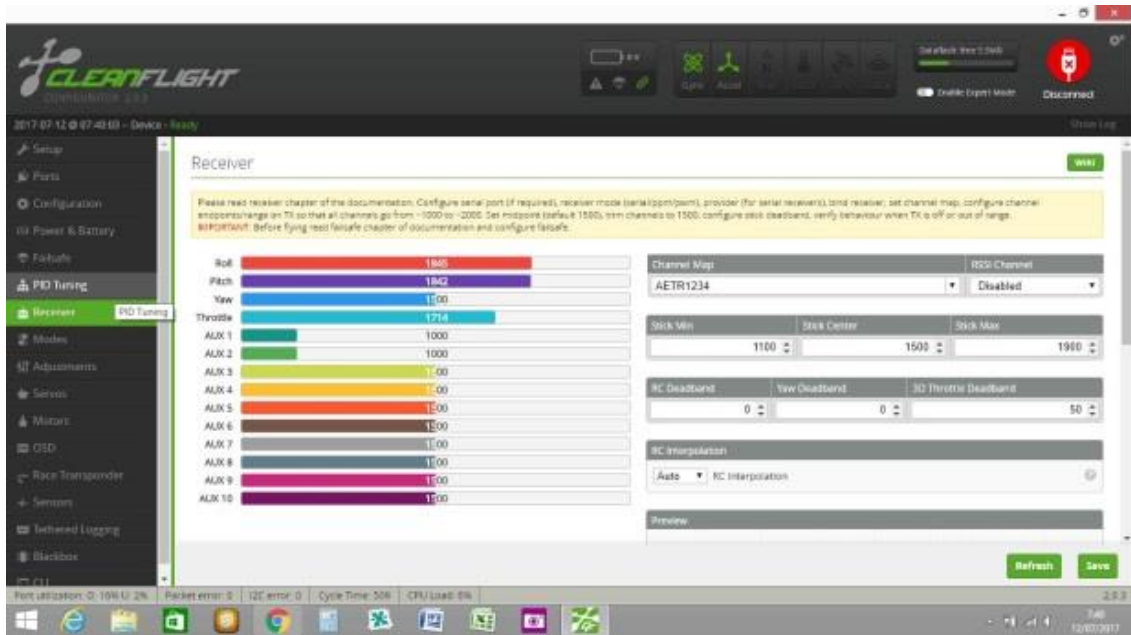
Imatge 69: Ports.

Es va seleccionar el port: UART1, que era el que vam emprar.



Imatge70:Failsafe.

Aquí es configurà l'abast vàlid dels impulsos enviats mitjançant dos límits, és a dir, des de quin valor el motor s'encenia i des de quin valor deixava d'accelerar la velocitat dels motors.



Imatge 71: Receiver.

En aquest apartat es podien veure manifestades les ordres que enviàvem des de l'emissora de ràdio.



Imatge 72: Motors.

Mitjançant aquest panell, es poden modificar la velocitat de tots els motors a la vegada o únicament la d'un.



Imatge 73: Gràfica del moviment.

La gràfica representa les diferents equacions que segueix el dron segons diferents paràmetres.

9. Resultats

Després de resoldre molts problemes, el dron aconseguí tenia una molt bona estabilització horitzontal.

El control, era una mica difícil per diferents raons:

- 1) Falta de pràctica amb les palanques de l'emissora i aconseguir aquesta pràctica era a costa de trencar hèlixs o altres peces.
- 2) El dron anava molt ràpid i responia massa de pressa. No hi havia temps per a pensar i un petit error suposava trencar alguna peça.
- 3) El pati on fèiem les proves estava ple d'obstacles: cistelles de bàsquet, llums i arbres. L'ideal hauria estat una zona ampla amb gespa per a amortir els cops, com ara un camp de futbol.
- 4) El fet que la controladora de vol no tingués magnetòmetre feia que el dron no mantingués una orientació fixa i s'havia d'anar corregint.
- 5) Tal com era de suposar, la bateria es descarregava en pocs minuts i, en canvi, tardava bastant de temps en carregar-se.

10. Conclusió

Després d'haver realitzat el disseny i la construcció d'un quadricòpter es conclouen diferents aspectes.

No ha estat tan fàcil com semblava al començament.

Vam perdre molt de temps seguint una informació falsa. Aquest és un dels problemes de buscar informació a Internet. De totes maneres va ser la mateixa informació d'Internet la que ens va portar pel camí correcte.

Hem hagut de resoldre molts problemes, però tenim la satisfacció d'haver-los resolt amb perseverança i una actitud positiva. Aquest pensament personal explica la cita esmentada al principi del treball: “ *No es como empieza, sino como acaba.*”

La conclusió final és que hem ampliat coneixements sobre el món dels drons i assolit l'objectiu de dissenyar i construir un quadricòpter.

11. Annexos

11.1 Manual motor

02830 Out Runner Brushless Motor Instruction

Dimensions

connection

Congratulations for your purchase of high performance brushless outrunner motor series. Our motors are designed to use perfectly with our star electronics based control kits. But it is also possible to use them with most of other common electronic speed controllers. The motors are designed for mainly clockwise or counter-clockwise. To change the turning direction you just simply and change the red and yellow connection wire. If you want to use outer speed controller, please refer to your controller manual or manual.

02830 (2212) SERIES OUTRUNNER BRUSHLESS MOTOR

Model	Voltage	KV (rpm/v)	Max Pull	Weight	Motor size	Shaft size	Max power	ESC	Rotation rpm
02830-8		1300	935g				275watt		LIP062:383 LIP064:730
02830-11	7.4-11V	1000	800g	62g	42x39mm	0.1x7.754mm	215watt		LIP062:1137 LIP064:234
02830-12		850	875g				187watt	37A	LIP062:1137 LIP064:390
02830-14		750	860g				165watt		LIP062:1230 LIP064:395


Product description

1. Mount your motor with the include screws tightly on your frame cover of model.
2. Our motors provide high efficiency stable design.
3. Small size, lightweight and long life.

ATTENTION:

1. Please make sure your motor has enough cooling when running. Otherwise to put a wide hole your model motor cover to improve the ventilation.
2. Our motors working best performance with the recommended rpm's or rps. If you consider it strange the propeller size be sure the motor can over heat and damaged.
3. Keep tracks away from the slush, dust, soap and oil. Terms be avoid car tracks. We wish you many happy arrivals with our star life quality equipment.

11.2 Manual acceleròmetre

 BOSCH	BMI160 Data sheet	Page 103
--	-----------------------------	----------

5.2 Sensing axes orientation

If the sensor is accelerated and/or rotated in the indicated directions, the corresponding channels of the device will deliver a positive acceleration and/or yaw rate signal (dynamic acceleration). If the sensor is at rest without any rotation and the force of gravity is acting contrary to the indicated directions, the output of the corresponding acceleration channel will be positive and the corresponding gyroscope channel will be "zero" (static acceleration).

Example: If the sensor is at rest or at uniform motion in a gravity field according to the figure given below, the output signals are:

- $\pm 0g$ for the X ACC channel and $\pm 0^\circ/\text{sec}$ for the Ω_x GYR channel
- $\pm 0g$ for the Y ACC channel and $\pm 0^\circ/\text{sec}$ for the Ω_y GYR channel
- $+ 1g$ for the Z ACC channel and $\pm 0^\circ/\text{sec}$ for the Ω_z GYR channel

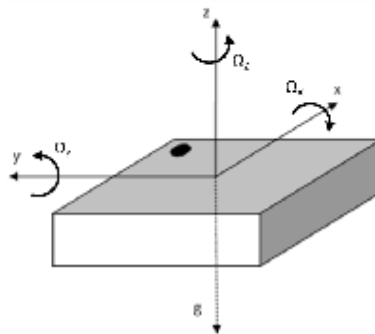


Figure 42: definition of sensing axes orientation

For reference the figure below shows the Android device orientation with an integrated BMI160.

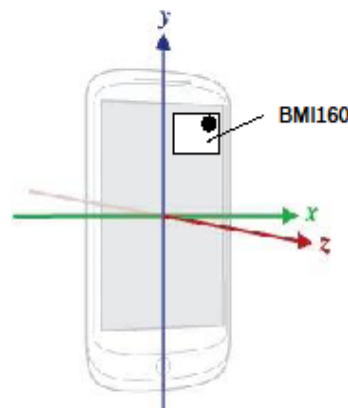



Figure 43: Android axis definition with BMI160

	<p align="center">BMI160 Data sheet</p>	<p align="right">Page 30</p>
---	--	------------------------------

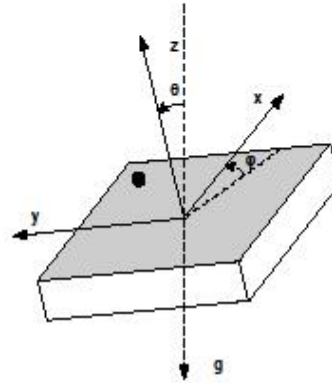


Figure 8: definition of coordinate system with respect to pin 1 marker

The measured acceleration vector components look as follows:

$$acc_x = 1g \cdot \sin \theta \cdot \cos \varphi \tag{1}$$

$$acc_y = -1g \cdot \sin \theta \cdot \sin \varphi \tag{2}$$

$$acc_z = 1g \cdot \cos \theta \tag{3}$$

$$(2)/(1): \quad \frac{acc_y}{acc_x} = -\tan \varphi$$

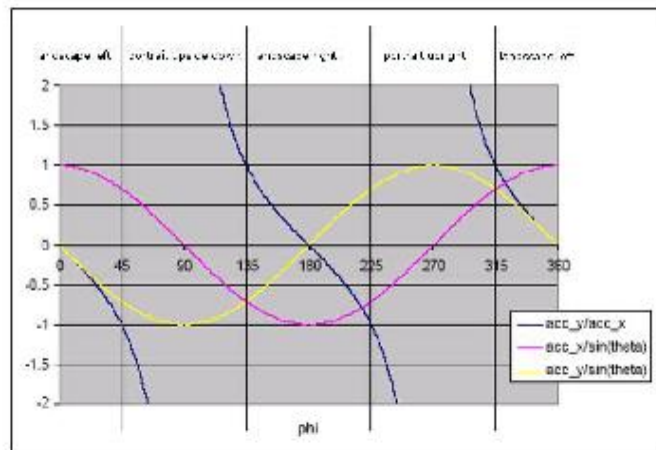


Figure 9: Angle-to-Orientation Mapping

11.3 Manual ESC

Manual of Brushless Motor Speed Controller

REV. 01.01.2016

When the speed controller is connected to the motor, the following parameters must be set:

Rated brushless motor speed (rpm)	Rated brushless motor voltage (V)	Rated brushless motor current (A)	When the speed controller is connected to the motor, the following parameters must be set:
1500	24	10	

PRECAUTIONS:

- Do not connect the speed controller to the motor until the motor is completely stopped.
- Do not connect the speed controller to the motor until the motor is completely stopped.
- Do not connect the speed controller to the motor until the motor is completely stopped.

Applications:

- Use the speed controller to control the speed of the motor.
- Use the speed controller to control the speed of the motor.

Connections:

- Connect the speed controller to the motor.
- Connect the speed controller to the motor.

Technical Specifications:

Parameter	Value
Rated brushless motor speed (rpm)	1500
Rated brushless motor voltage (V)	24
Rated brushless motor current (A)	10

Manual of Brushless Motor Speed Controller

REV. 01.01.2016

When the speed controller is connected to the motor, the following parameters must be set:

Rated brushless motor speed (rpm)	Rated brushless motor voltage (V)	Rated brushless motor current (A)	When the speed controller is connected to the motor, the following parameters must be set:
1500	24	10	

PRECAUTIONS:

- Do not connect the speed controller to the motor until the motor is completely stopped.
- Do not connect the speed controller to the motor until the motor is completely stopped.
- Do not connect the speed controller to the motor until the motor is completely stopped.

Applications:

- Use the speed controller to control the speed of the motor.
- Use the speed controller to control the speed of the motor.

Connections:

- Connect the speed controller to the motor.
- Connect the speed controller to the motor.

Technical Specifications:

Parameter	Value
Rated brushless motor speed (rpm)	1500
Rated brushless motor voltage (V)	24
Rated brushless motor current (A)	10

Manual of Brushless Motor Speed Controller

REV. 01.01.2016

When the speed controller is connected to the motor, the following parameters must be set:

Rated brushless motor speed (rpm)	Rated brushless motor voltage (V)	Rated brushless motor current (A)	When the speed controller is connected to the motor, the following parameters must be set:
1500	24	10	

PRECAUTIONS:

- Do not connect the speed controller to the motor until the motor is completely stopped.
- Do not connect the speed controller to the motor until the motor is completely stopped.
- Do not connect the speed controller to the motor until the motor is completely stopped.

Applications:

- Use the speed controller to control the speed of the motor.
- Use the speed controller to control the speed of the motor.

Connections:

- Connect the speed controller to the motor.
- Connect the speed controller to the motor.

Technical Specifications:

Parameter	Value
Rated brushless motor speed (rpm)	1500
Rated brushless motor voltage (V)	24
Rated brushless motor current (A)	10

User Manual of AE-20A, 25A, 30A Electronic Speed Controller (ESC)

Thanks for purchasing our Electronic Speed Controller (ESC). High power system for RC model can be very dangerous, so we strongly suggest you read this manual carefully. In that we have no control over the correct use, installation, application, or maintenance of our products, we do not take the liability for the damage or injury caused by any damages, losses or costs resulting from the use of the product. Also, please do not use the speed controller for any other applications, i.e., will be voided. We reserve the liability for personal injury, property damage or consequential damage resulting from our product or our workmanship. As this is highly technical, the difference to computer or is limited to the precise amount of the affected product.

Features:

- High performance microprocessor using and can be compatible with all kinds of motors and the right driving efficiency, short and long, suitable for most brushless motor on market;
- Low current, low impedance controller, providing good anti-stalling capacity; possibly reduce the possibility of loss of current;
- Sequential motor speed (Microtrim): 10000rpm to 20000rpm to power 25000rpm (1-pole);
- Brushless motor can be compatible with all motors currently on the market, smooth, linear, quick and precise throttle response;
- Wide speed and torque: Made for helicopters, even 4.5s with 7.5mV/turn, when it used BLHeli-chor.
- With precision over-temperature function, ensure the safety of using ESC and providing good anti-jamming capability;
- With the design of cooling fins, increase using temperature;
- The screw is small, properly easy for installation.

Direction for ESC:

- The range of input voltage: 3.5V-12V (up to 2 leads with 6-22cells). Don't surpass voltage to use, otherwise it will break the product.
- The power should be as inserted in source very lead to engine and this is a guide.
- Setting for high throttle response speed, current will be increased as expected in ambience.
- With continuous heavy current working, it may be necessary to use ESC with fan.

Working Diagram: Attention to the positive and negative of the power and connection the power for testing (DO NOT connect the battery connection to the controller)



For normal start process direction

Start procedure	Connect ESC with battery when it is working normally. If the motor does not start, check the battery voltage. If the battery voltage is low, charge it or use a new one.	1. Check the battery voltage. 2. Check the motor connection. 3. Check the motor speed. 4. Check the motor direction.
-----------------	--	--

Fig. When use 11.1V LiPo battery, the voltage protection should set 11.1V. Turn off the motor.

11.1V LiPo battery, the full charge is 12.6V, the ESC is required as 200mAh LiPo battery. Error in the "start" function on voltage is 2.4V or less. When the voltage protection function is set to 2.4V, the motor will stop when the voltage is 2.4V. When the voltage is 2.4V, the motor will stop when the voltage is 2.4V.

Additional functions are available only to the Turnigy ESCs with enough power and functioning normally.

When you start to use the battery in a rough power, you can find the power of motor according to the ESC's response.

4. Plane Mode:

Normal plane mode (default 1):

Suitable for normal plane, see manual.

Helicopter mode (default 2):

How to set the start condition, suitable for normal, but it will be flying mode. When change to the other mode, if the lighter RPM makes the gyroscopes have slight noise, for this moment should be down the sensitivity of gyroscopes.

Helicopter mode 2 (default 3):

Suitable for the low flying of RC, it will not suitable for Normal flying mode. Choose Governor mode function, it will be suitable for the use. If the high RPM makes the gyroscopes have slight noise, when it is down the sensitivity of gyroscopes. Because of this, when RPM makes the gyroscopes have slight noise, when it is down the sensitivity of gyroscopes setting, increase or decrease the setting, all of this will be because the function of the Governor, even cause the sensitivity of gyroscopes will be suitable for the use.

5. Throttle response speed settings: Normal, Medium, High

Default is Normal. The function is according to the throttle position to adjust properly. The normal, 10 plane and violent 10 helicopter mode for Multi-rotor. High speed, make the throttle more quick and precise. But pay attention when you increase the throttle response speed, the voltage and current power supply will be increased.

Other functions:

Start protection function: The ESC checks the throttle lever is set to the bottom position when starting, motor and lock will be released. If you set the throttle to the top position, the motor will not start. At the same time, a warning tone will be heard. Please refer to the manual for other settings, and set the power to 1.

Over-heat protection:

When the motor is overheated, the ESC will stop the motor. The over-heat protection will be activated in 100 degrees. The ESC will stop the motor when the temperature is 100 degrees. In order to reach the best efficiency of ESC, we suggest to install the ESC to cross ventilation position in the cabin.

to the setting table), each setting includes 3 options and each option's setting value is according to the opt., zero-, zero- and bottom position of function.

Throttle stick and touch functions match table: (in Sinos' dmp', all can setting of below)

Function	Low	Medium	High
1. Brake setting	off	soft	hard
2. Battery Protection Voltage Threshold	Low	Medium	High
3. Battery Protection Voltage Threshold	High cut-off threshold	Medium cut-off threshold	Low cut-off threshold
4. Brake Force	Low	Medium	High
5. Throttle response speed setting	Normal	Medium	High

Setting means defaults

3. Check after setting
The motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

The buzzer "ding-ding" self-checking tone when the throttle is on the bottom position.

The buzzer "ding-ding" self-checking tone when the throttle is on the bottom position.

The buzzer "ding-ding" self-checking tone when the throttle is on the bottom position.

Caused the battery voltage for ESC being connected correctly.

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

Menu

1. Brake Settings: off-soft-hard

2. Settings: Low / Medium / High

3. Battery Protection Voltage Threshold: High / Medium / Low

Solution is according to pole of motor. Low setting value for 2 pole motor, for high speed, medium for 3 pole motor, medium for 4 pole motor. For motor with more than 6 poles, get a high efficiency for higher speed. High setting value can be chosen. When revision frame, it will cause current increase and ESC loses voltage regulation in the discharge current value, and the motor speed will increase.

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

ESC will start by "ding-ding-ding"

reconnect the power.
Connected ESC with battery, when it is working normally. Connect two speed control units (ESC) to the ESC. When the ESC is working normally, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

Motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

Motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

Motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

Throttle range setting description:

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

1	2	3	4	5
Start motor	ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.	ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.	ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.	ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

Function setting mode:

1. After throttle range setting, enter the function set up screen.

2. Setting mode option.

The tone of entering function setting mode is Sinos' dmp' and 2 tones of "ding-ding-ding" self-checking tone. Before confirmed tone finish, the throttle stick on the corresponding position of setting item that you choose. When the throttle stick hit the confirmation tone is correct, then this setting is finished. When the confirmation tone is correct, the setting is finished. When the confirmation tone is correct, the setting is finished.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

ESC will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position. When the throttle is on the bottom position, the motor will start by "ding-ding-ding" self-checking tone when the throttle is on the bottom position.

12. Fonts d'informació

Fonts principals per a dur a terme la realització:

<http://www.instructables.com/id/Make-Your-Own-Removable-Drone-With-Arduino/?ALLSTEPS> (Pàgina web de referència)

https://hobbyking.com/es_es (Pàgina web on vam comprar els components del dron.)

<https://www.arduino.cc/> (Pàgina web d'Arduino.)

<http://cleanflight.com/> (Pàgina web de CleanFLight.)

Informació sobre la història dels drons:

<http://eldrone.es/historia-de-los-drones/>

<http://eldrone.es/que-es-un-drone/>

<https://en.wikipedia.org/wiki/Quadcopter>

<https://es.wikipedia.org/wiki/Quadcopter>

(https://ca.wikipedia.org/wiki/Vehicle_aeri_no_tripulat)

<http://mundrone.blogspot.com.es/p/historia-de-los-drones.html>

Treball d'inspiració:

<http://academica->

e.unavarra.es/bitstream/handle/2454/19208/TFG%20Jose%20Etxeberria.pdf?sequence=1

