



BALANCÍ EN EQUILIBRI AMB ANGLE DESITJAT ACTUAT AMB MOTORS BLDC SENSORLESS

Arnau Quintana Llorens

Gabriel Poch

01-02-2022

En aquesta memòria es detalla el procés de disseny i muntatge d'un balancí, equilibrat mitjançant l'ajust de la velocitat de dos motors BLDC, amb control en llaç tancat, que obeeixen a les instruccions donades gràcies al coneixement de l'angle proporcionat per una IMU.



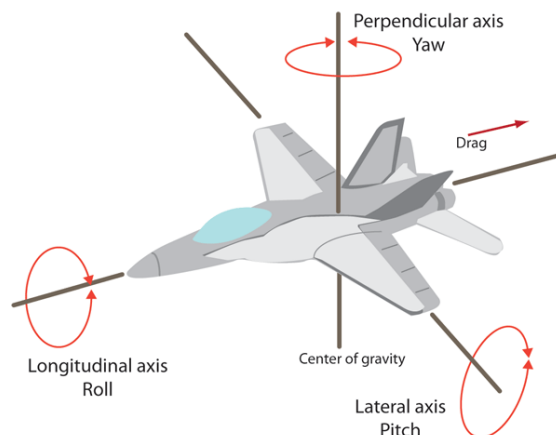
ÍNDIX

| | |
|--|----|
| INTRODUCCIÓ. ESTAT DEL ART. | 3 |
| OBJECTIUS | 4 |
| ENGINYERÍA DE CONCEPCIÓ | 5 |
| ENGINYERÍA DE DETALL..... | 6 |
| ESCs | 6 |
| Control Central | 8 |
| IMU | 8 |
| PID | 9 |
| Comunicació amb els ESCs | 9 |
| RESULTATS | 10 |
| ESC..... | 10 |
| CONTROL..... | 12 |
| BALANCI..... | 13 |
| ARQUITECTURA RTOS..... | 14 |
| VIABILITAT TÈCNICA | 16 |
| VIABILITAT ECONÒMICA | 17 |
| ESCs | 17 |
| CONTROL..... | 17 |
| CRONOGRAMA..... | 18 |
| CONCLUSIONS | 18 |
| BIBLIOGRAFÍA..... | 19 |
| ANNEXOS..... | 19 |
| 1. MEMÒRIA DE LA REALITZACIÓ D'UN ESC PER UN MOTOR BLDC | 19 |
| 2. CODI DEL CONTROL EN LLAÇ TANCAT D'UN BLDC..... | 19 |
| 3. CODI EN RTOS PEL CONTROL DEL ANGLE D'INCLINACIÓ DEL BALANCI | 19 |

INTRODUCCIÓ. ESTAT DEL ART.

Aquest projecte es basa en saber la posició del balancí en cada moment. S'aconsegueix amb una IMU o Unitat de Mesura Inercial.

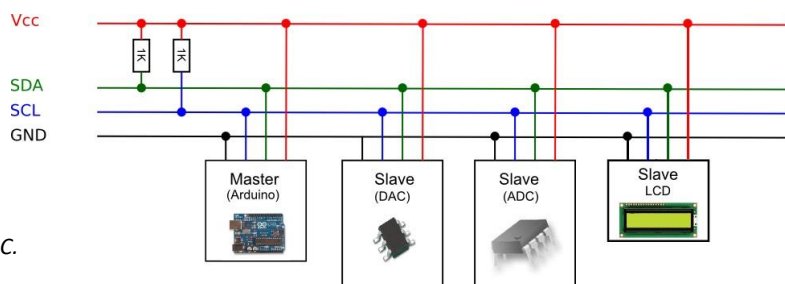
Una IMU és un component principal dels sistemes de navegació inercial utilitzats en avions, naus espacials, vaixells o inclús míssils. La nostra és més senzilla que les utilitzades en aquestes aplicacions, però el principi és el mateix. Detecta l'acceleració fent ús d'acceleròmetres i detecta els canvis en els atributs rotacionals fent ús d'un giroscopi.



Imatge 1. Aplicació d'una IMU

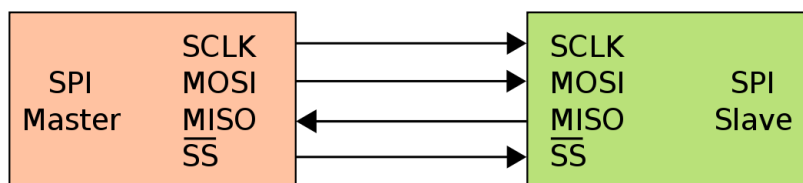
Dos dels protocols que es veuran al llarg del projecte són l'I2C i el SPI. Són dos dels protocols més utilitzats en les comunicacions entre dispositius.

L'I2C es basa en dues línies, una pel rellotge i l'altre per les dades. Per poder triar amb quin dispositiu es vol comunicar el Màster, cada un d'ells té una direcció pròpia, la qual s'envia al principi d'iniciar una comunicació. Aquest protocol té definit de manera exhaustiva les trames i la quantitat de bits a enviar. La gran simplicitat i versatilitat el fan ser un dels preferits pels enginyers que treballen amb sistemes encastats.



Imatge 1. Protocol I2C.

L'SPI es basa en 4 línies: rellotge (CLK), enviament de dades(MOSI), rebuda de dades(MISO) i selector de chip(CS). Les línies MOSI i MISO canvien els seus papers si ho mirem des d'un Slave. El CS ens ajuda a saber amb quin dels dispositius connectats a la mateixa línia es vol comunicar el Màster. Entre algunes de las avantatges del protocol es troba que és una comunicació full-dúplex, és a dir, es pot enviar i rebre dades simultàniament, a diferència del I2C. A més, permet una major velocitat de comunicació. Per contrari, tenim que són necessàries més línies i l'ús de més d'un Màster a una mateixa línia pot ser difícil.



Imatge 3. Connexions del protocol SPI.

OBJECTIUS

En aquest apartat es presenten els objectius del projecte. Està compostat per diverses parts i cada una d'elles compon un objectiu.

El primer objectiu és el **disseny de dues PCB funcionals: un ESC i una per al control del sistema**. El primer disseny està dedicat al controlador d'un motor BLDC sense sensors, per aquest projecte en tenim un total de dos. El segon està dedicat a un petit disseny que farà més accessibles els pins de la launchpad de control del sistema. Passarem per l'esquemàtic i layout, tenint cura de respectar les regles de disseny imposades pel fabricant així com també tenint en compte el placement per evitar problemes d'EMCs, no són crítics en aquest projecte, però és important tenir-ho en ment.

El segon objectiu és el **control dels motors BLDC en laç tancat**, és a dir, controlats per l'ADC i no per un timer. Haurem de llegir la senyal de BEMF recuperada del motor ja que, com s'indica al títol de la memòria, fem ús de motors BLDC sense sensors.

El tercer objectiu és el **disseny i muntatge d'un balancí**. Aquest passa pel disseny a través del 'Autodesk' i per la posterior construcció de l'estructura.

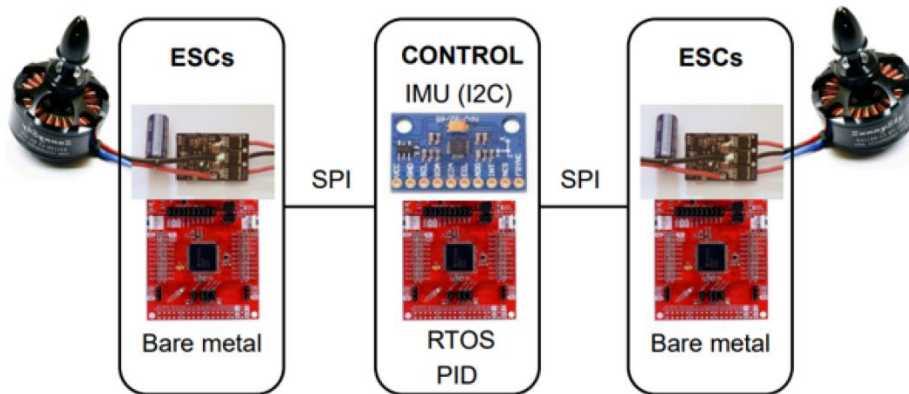
El quart objectiu és la **programació en RTOS de la launchpad de control**. Aquest objectiu inclou tots els protocols usats en el sistema: I2C per la IMU i SPI pels ESCs. A més, inclou també el processament de dades de la IMU així com també l'algoritme del següent objectiu.

El cinquè objectiu és el **disseny d'un sistema PID**. S'usarà per poder equilibrar el balancí i recuperar la posició exigida en cas de desequilibrar-se per algun motiu.

INGENYERÍA DE CONCEPCIÓN

En aquest apartat trobem l'explicació general del sistema: el funcionament i totes les parts que el componen.

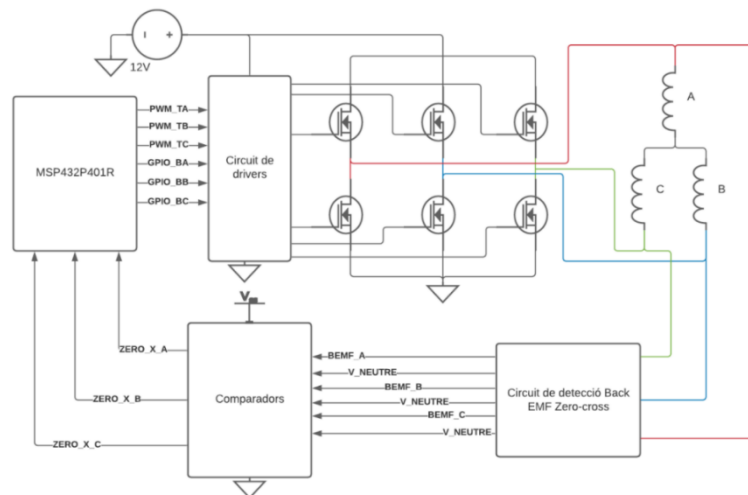
La següent imatge mostra el diagrama de blocs del sistema:



Imatge 4. Diagrama de blocs del projecte.

D'una manera resumida veiem tot el comentat als objectius del projecte. Als ESCs els controla una launchpad cadascun, que obeeixen ordres d'una tercera. Aquesta última controla l'angle del balancí mitjançant una IMU i, amb un control PID, equilibra el balancí a un angle desitjat.

La comunicació entre les launchpad dels ESCs i la de control és mitjançant el protocol SPI; els ESCs com a slaves i la de control com a Master. La comunicació entre la IMU i la launchpad de control es fa a través d'I2C.



Imatge 5. Diagrama de blocs del ESC.

Aquesta imatge mostra el diagrama de blocs dels ESCs. Formats per un pont trifàsic de MOSFETS que van directament a cada una de les bobines del motor. Aquestes també estan connectades a un circuit que s'encarrega d'extreure la senyal de BEMF que fem servir pel control enllaçat.

ENGINYERÍA DE DETALL

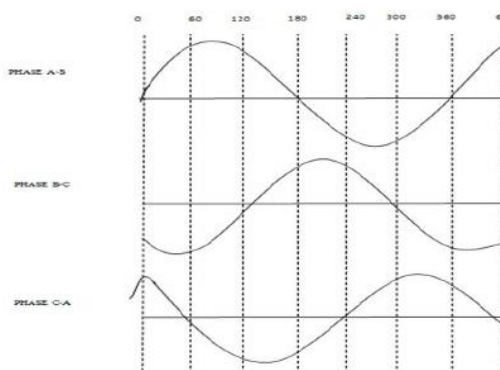
En aquest apartat estudiem en profunditat cada una de les parts del sistema.

ESCs

Adjunt a aquesta memòria es troba la d'un projecte anterior, on es realitza un ESC amb control en llaç obert amb detecció del BEMF. No entrarem en detall en aquesta memòria sobre el hardware ni les simulacions fetes, doncs es troben en aquest document adjuntat; si ho farem en el software, doncs ha patit un seguit de canvis.

Amb respecte al anterior projecte hi ha hagut tres canvis: el control en llaç tancat, l'addició de la hèlice i el protocol SPI pel qual rebrem ordres de variar el duty cycle. A diferència del control en llaç obert, en el control en llaç tancat es controla el gir mitjançant la lectura del BEMF amb el ADC, amb el que es detecta el pas per 0 i es canvia la fase.

En la imatge veiem el que seria la senyal de BEMF de les 3 fases de manera teòrica, a la realitat el sinus no es tan perfecte. Les senyals són les mateixes però desfasades 120° entre elles. Els passos per 0 determinen el moment en que s'ha de canviar de fase, doncs és el moment en que l'imant del rotor passa per davant de la bobina no energitzada del estator. Per a no perdre 'torque', l'ideal seria fer el canvi a l'equivalent en temps a 30° des que l'imant passa per davant la bobina; en el nostre cas hem decidit sacrificar aquest punt per simplificar el codi.



Imatge 6. Senyals del BEMF.

Si entrem en detall en el codi, el qual es troba adjunt a aquest document, el primer que veurem és una rampa d'arrancada del motor; aquesta rampa es realitza per a que el codi sigui funcional per qualsevol motor. A més, ens ajuda a adaptar la velocitat del motor a la necessària per al ADC. Aquí trobem dos aspectes a comentar: la configuració del període i el duty cycle; el període controla els temps entre sinus i el duty cycle l'amplitud del sinus. Hem de quadrar els dos paràmetres per poder fer el control mitjançant l'ADC. A més, en el moment que afegim la hèlice aquests paràmetres canvien ja que estem incorporant pes. Aquí tenim una mostra del codi de la rampa, tenint en compte que la hèlice està incorporada:

```
if(control_ADC == 0)
{
    if(TIM1->CCR[0] == 1900 || TIM1->CCR[0] < 1900)
    {
        rampa_completa = 1; //damos por concluida la rampa y aseguramos que ya hay suficiente BEMF
    }
    else if(TIM1->CCR[0] > 1900)
    {
        TIM1->CCR[0] -= 100;
        /*if(TIM0->CCR[1] < 230)
        {
            TIM0->CCR[1] += 2;
            TIM0->CCR[3] += 2;
            TIM0->CCR[4] += 2;
        }*/
    }
    for(i = 0; i < 25000; i++){}
```

El que fa aquest extracte de codi és simplement reduir el període progressivament i quan assegurem que la senyal de BEMF és suficient com per donar el control al ADC, posar a 1 la variable 'rampa_completa'. L'ADC està sempre actiu, però no té el control fins que aquesta variable ho indica.

Repassem també la interrupció del ADC, doncs hi ha certs punts clau que requereixen d'una explicació més exhaustiva.

Hem de tenir en compte que les dues interrupcions que tenim activades són les de la finestra de comparació: la que saltarà quan el valor llegit per l'ADC sigui superior al 'threshold' i la que saltarà quan aquest valor sigui inferior a un altre 'threshold'. És important entendre això, doncs el primer que fem quan salta una de les dues, a part de baixar la bandera, és desactivar les dues interrupcions, per a que no segueixi saltant mentre executem la interrupció; hem de pensar que l'ADC està pendent de totes les fases a no ser que nosaltres ho impedim. Per tant, el que fem cada cop que salta una interrupció, com sabem la fase en la que ens trobem i la següent, és desactivar els canals que no ens interessa llegir i activar al que hem d'estar atents. Per poder modificar aquesta configuració, hem de deshabilitar i parar les conversions del ADC; un cop hagi acabat la configuració nova, reactivem l'ADC per a la següent fase.

En l'extracte que veiem més avall, també podem apreciar la condició que s'executa quan la rampa ha acabat: desactivar el timer, indicar mitjançant una variable que el control és mitjançant l'ADC, canviar de fase doncs s'ha detectat un pas per zero i indicar el proper canal a mirar. Ens fixem que per la bobina B canviem el threshold; això és degut a que quan donem el control al ADC, la fase B i C guanyen un offset, simplement el tenim en compte augmentant els valors de threshold.

```
void ADC14_IRQHandler(void)
{
    if (ADC14->IFGR1 & ADC14_IFGR1_HIIFG) //Si estamos por encima del valor de threshold
    {
        ADC14->CLRIFGR0 = 0xFFFFFFFF; //Bajamos todas las banderas posibles
        ADC14->CLRIFGR1 = 0xFFFFFFFF;
        ADC14->IER1 &= ~ADC14_IER1_HIIE; //Desactivamos las interrupciones para que no siga saltando
        ADC14->IER1 &= ~ADC14_IER1_LOIE;
        if(control_ADC == 0) //Si estamos realizando la rampa de arranque
        {
            ADC14->CTL0 &= ~(ADC14_CTL0_ENC | ADC14_CTL0_SC); //deshabilitamos el ADC para cambiar la bobina a mirar
            ADC14->MCTL[2] &= ~ADC14_MCTLN_WINC; //Seleccionamos el canal 22 para mirar la bobina C
            ADC14->MCTL[1] &= ~ADC14_MCTLN_WINC;
            ADC14->MCTL[0] |= ADC14_MCTLN_WINC;
            ADC14->CTL0 |= (ADC14_CTL0_ENC | ADC14_CTL0_SC);

            if(rampa_completa == 1) //Una vez se haya completado la rampa de arranque
            {
                TIMER_A1->CTL &= MC_0; //Desactivamos el timer
                control_ADC = 1; //Activamos la condición de lazo cerrado
                control_de_fases(); //Cambiamos de fase pues ha habido un paso por cero
                switch(FASE) //No sabemos de que fase venimos así que las definimos todas por evitar fallos
                {
                    case 1:
                        ADC14->CTL0 &= ~(ADC14_CTL0_ENC | ADC14_CTL0_SC);
                        ADC14->MCTL[0] &= ~ADC14_MCTLN_WINC;
                        ADC14->MCTL[2] &= ~ADC14_MCTLN_WINC;
                        ADC14->MCTL[1] |= ADC14_MCTLN_WINC | ADC14_MCTLN_WINTCH;
                        ADC14->CTL0 |= (ADC14_CTL0_ENC | ADC14_CTL0_SC);
                        break;
                    case 2:

```

Hem de comentar el codi del SPI, que va dins el main, doncs hi ha certs detalls que requereixen d'una atenció especial. No fem ús de les interrupcions del mòdul doncs podem portar problemes a l'hora d'executar-les. La nostra prioritat és el canvi de fase i si afegim una altra interrupció podríem arribar a causar pèrdues d'algun canvi de fase amb la conseqüent parada del motor.

Per tant, l'estratègia seguida és que comprovem si hi ha alguna bandera pendent dins del main i si és així llegim el buffer, sinó, el mòdul no entra en joc. Des de la launchpad de control rebrem un caràcter que va des de A fins a K, ja que podem tenir 10 valors diferents de duty cycle. Enviem a través de SPI caràcters doncs imposar les condicions es fa més fàcil, però podrien enviar un nombre si això ho desitgessim.

```
if(EUSCI_B0 -> IFG & EUSCI_B_IFG_RXIFG) //si se ha recibido algun mensaje por SPI
{
    duty_actual = EUSCI_B0->RXBUF; //guardamos el mensaje en una variable bajando el flag de lectura
    EUSCI_B0->IFG &= ~EUSCI_B_IFG_RXIFG;
    P1OUT ^= 0x01;
    if((duty_actual == 'A' || duty_actual == 'B' || duty_actual == 'C' || duty_actual == 'D' ||
        duty_actual == 'E' || duty_actual == 'F' || duty_actual == 'G' || duty_actual == 'H' ||
        duty_actual == 'I' || duty_actual == 'J' || duty_actual == 'K') && duty_actual != duty_anterior)
    {
        check_duty(); //comprobamos que hemos recibido por si debemos variar el duty cycle
        P2OUT ^= 0x01;
    }
}
```

Control Central

Aquesta launchpad s'encarrega de recollir les dades del acceleròmetre i el giroscopi, processar-les i extreure l'angle d'inclinació, aplicar un control PID amb aquest angle i corregir les velocitats dels motors per tal d'arribar al angle desitjat.

IMU

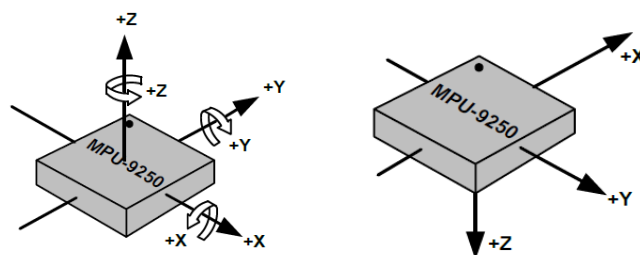
Una 'IMU' és una unitat de mesura inercial. És un dispositiu que mesura la velocitat, orientació o forces gravitacionals d'un aparell, fent ús d'acceleròmetres i giroscopis. Tenen una desavantatge, que ja veurem com resoldre-la, i és que acumulen un error a mesura que prenen dades.

Ens comuniquem amb ella mitjançant el protocol I2C. Hem de llegir un total de 12 registres: 6 pel giroscopi i 6 per l'acceleròmetre. Les dades de cada un dels tres eixos ocupa 16 bits i cada registre és de 8.

El codi de lectura per I2C i el processament de les dades rebudes es troba adjunt a la memòria però cal comentar un parell d'aspectes.

El giroscopi mesura la velocitat angular en $^{\circ}/s$. L'acceleròmetre ens dona la sortida en 'g'. El valor de la sortida depèn de la configuració que imposen; però serà proporcional a la velocitat angular que tinguem en cada moment.

En aquesta imatge s'aprecia l'orientació de cada un dels eixos. A nosaltres el que més ens interessa és el Y, doncs és el que ens dirà l'angle d'inclinació del balancí.

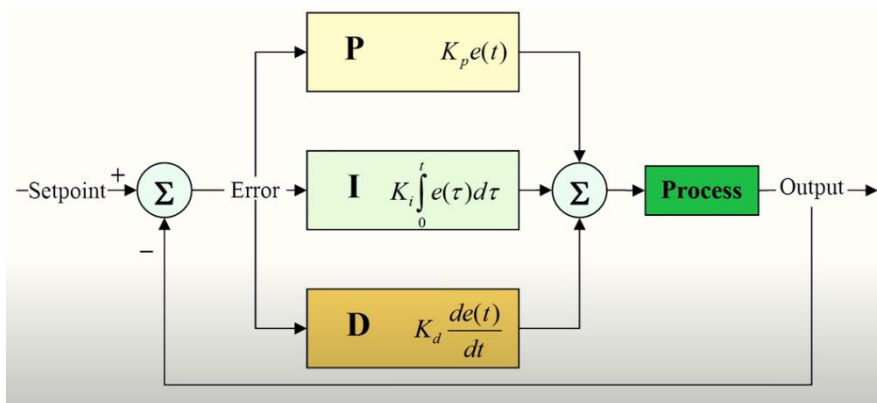


Imatge 7 i 8. Eixos del giroscopi i del acceleròmetre.

PID

L'algoritme PID intenta reduir el valor del error a 0 i va modificant els paràmetres per tal d'aconseguir-ho. El podem aplicar a qualsevol magnitud, en el nostre cas és l'angle d'inclinació del balancí.

El funcionament és el següent: el sistema genera una sortida, la qual recollim y comparem amb la sortida desitjada. La diferència entre el valor real i el desitjat serà el paràmetre que l'algoritme intentarà reduir 0 mitjançant la suma de 3 formules, les quals tenim més avall.



Imatge 9. Esquema del algoritme PID.

Trobem una part 'proporcional', una part 'integral' i una 'derivativa'.

La part proporcional, tal com el nom indica, proporciona una sortida proporcional al error. Si apliquem només aquesta part podríem controlar un sistema estable, però el seu exercici serà limitat amb un offset permanent.

La part integral dona una sortida que és proporcional el error acumulat, el que implica que és una manera lenta de controlar el sistema. La sortida d'aquesta part és diferent de zero quan el error és zero, el que implica que donada una referencia constant, o pertorbacions, el error en règim permanent és zero. Ajuda a reaccionar davant canvis lleugers en la sortida.

La part derivativa aporta una caràcter de previsió, per tant, fa el sistema més ràpid davant canvis bruscs. La principal desavantatge és que amplifica el soroll, el que podria provocar una saturació en el controlador. Aquesta part quasi mai s'utilitza sola, doncs és eficaç només durant períodes transitoris.

Comunicació amb els ESCs

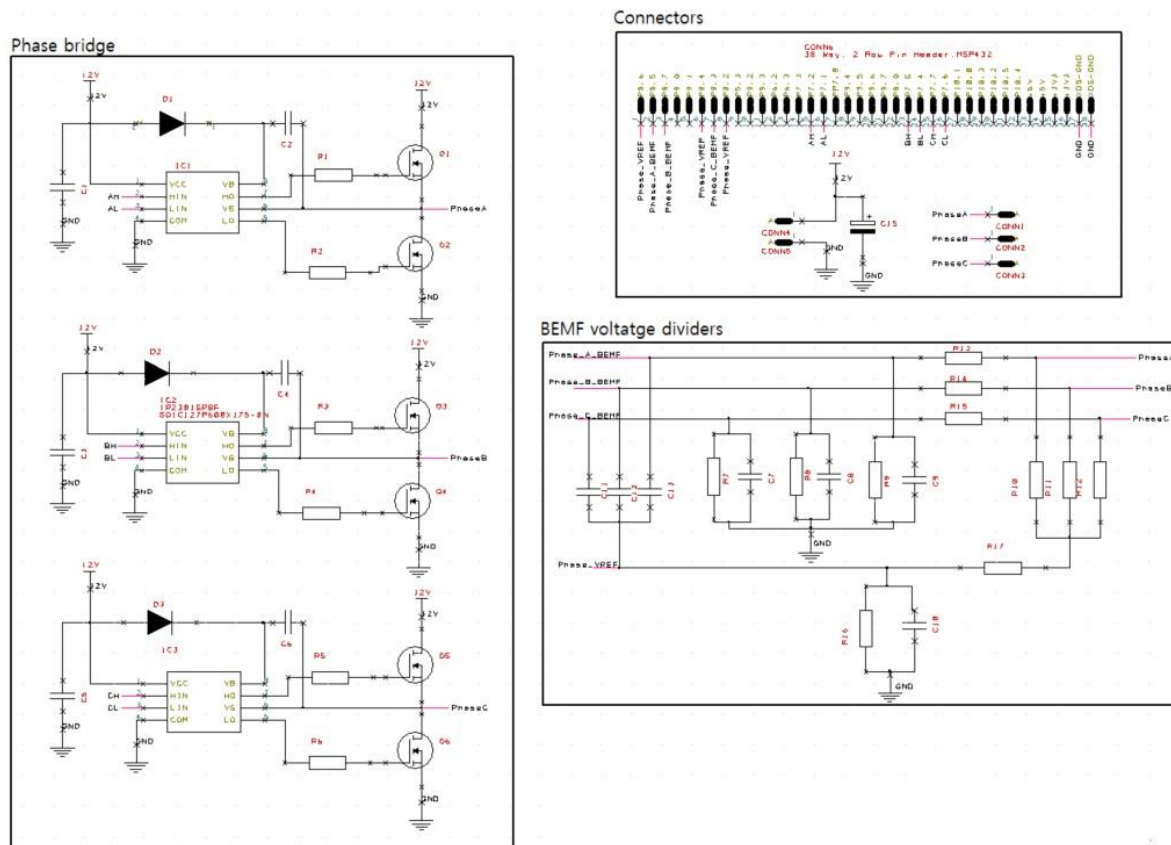
La comunicació es fa mitjançant el protocol SPI. En el nostre cas fem una comunicació amb 3 cables, sense el 'chip select'. Sense el 'chip select' precisem de dos mòduls diferents per poder comunicar-nos amb els motors. A més, no precisem de resposta del ESC, per tant, només utilitzem 2 cables per la comunicació, el MOSI (sortida del master i entrada del slave) i el relloige.

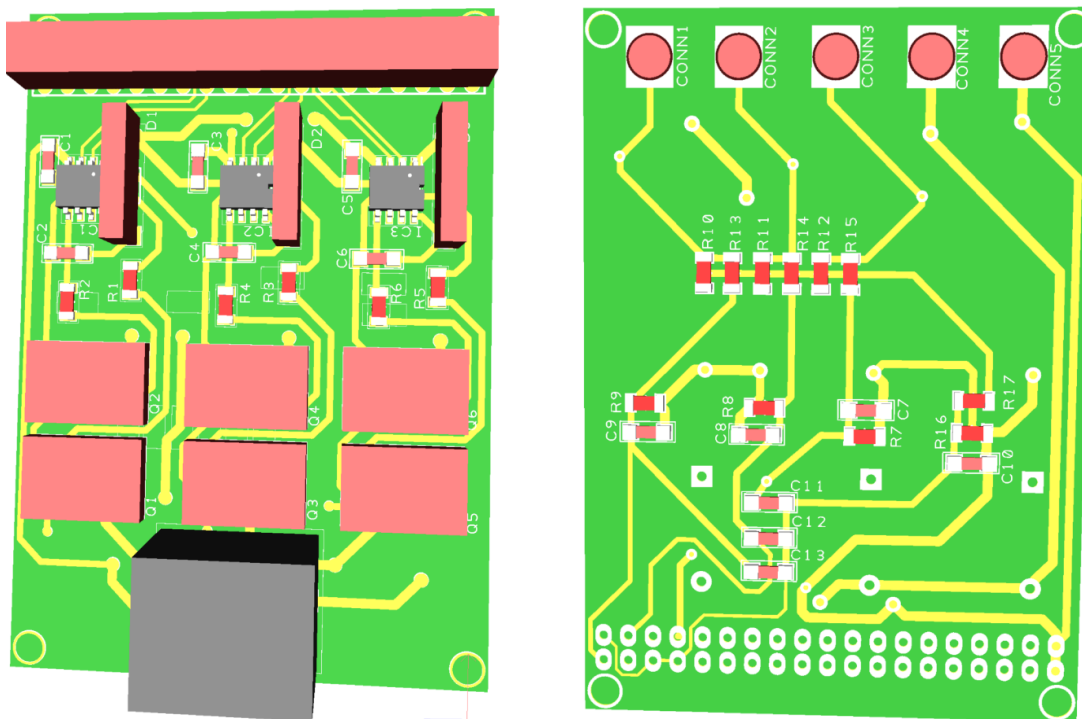
La comunicació és senzilla, cada cop que es recull una dada i es calcula l'error, s'envia el valor de duty cycle que hauria de tenir cada un dels motors. En els ESCs ens encarreguem d'interpretar si és necessari canviar o no la velocitat. Tal com hem comentat anteriorment, enviem caràcters i no números, les condicions i les comparacions han estat menys complicades de quadrar. Simplement 'A' seria el valor de duty cycle més baix i 'K' el més alt (tenint en compte que la diferencia entre el valor més baix i més alt és 10 unitats).

RESULTATS

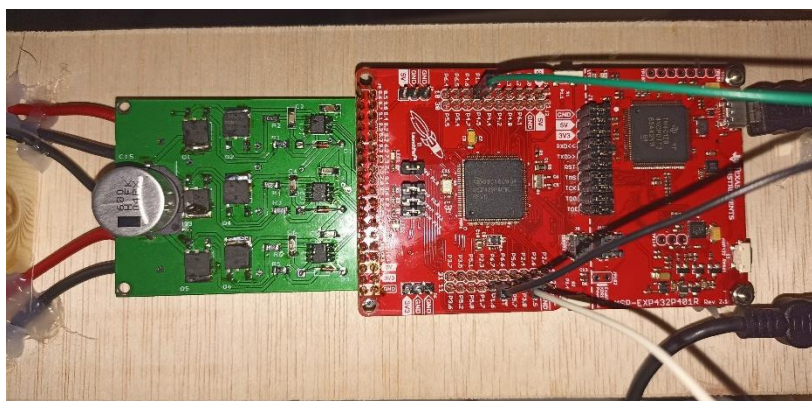
En aquest apartat trobem la presentació del esquemàtic així com també les PCB realitzades. A més, trobem l'explicació del problemes que han sorgit durant la realització del projecte així com també la solució implementada.

ESC





Imatge 10 i 11. PCB del ESC.



Imatge 12. Muntatge del ESC juntament amb la launchpad que el controla.

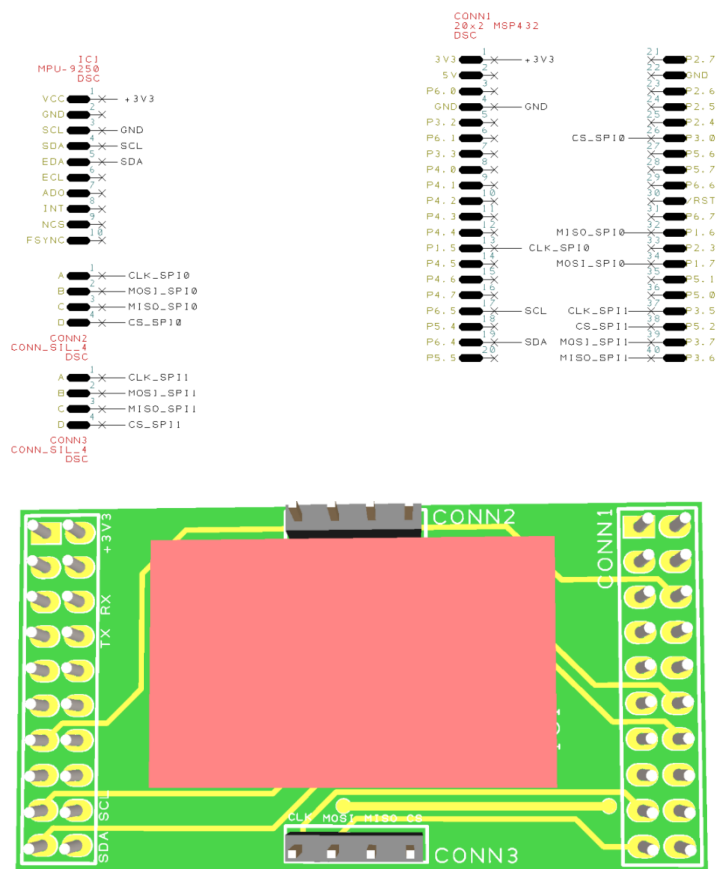
Un punt a comentar és que s'ha vist la necessitat de saber si està funcionant bé la comunicació SPI sense debugar el codi. Es fan servir 2 leds, el P1.0 i el P2.0 per aquest fi. El primer farà pampallugues sempre que rebem quelcom per SPI, el segon farà pampallugues quan rebem alguna cosa que haguem de revisar, és a dir, quelcom que faci canviar el duty del motor.

El P1.0, abans d'avisar d'aspectes del SPI, ens ajuda a saber quan l'ADC ha pres el control del gir. El primer cop que s'encén i es queda fix és quan sabem que el timer ja està fora de joc.

El botó S1 reinicia la rampa d'arrancada en cas que hi hagi hagut un error al inici.

Una del problemes més crítics que s'ha trobat durant la realització d'aquesta part va ser que una setmana abans de l'entrega del projecte un dels ESCs va deixar de funcionar. Debugant la PCB es va veure que la fase B tenia el transistor alt obert permanentment. No era qüestió de codi ni del components utilitzats, ja que es va canviar tota la fase sense èxit. Finalment es va culpar a una pujada de tensió que va provocar que la fase B deixés de treballar com hauria; per solucionar-ho, es va muntar una altra PCB amb components aprofitats d'un disseny antic.

CONTROL



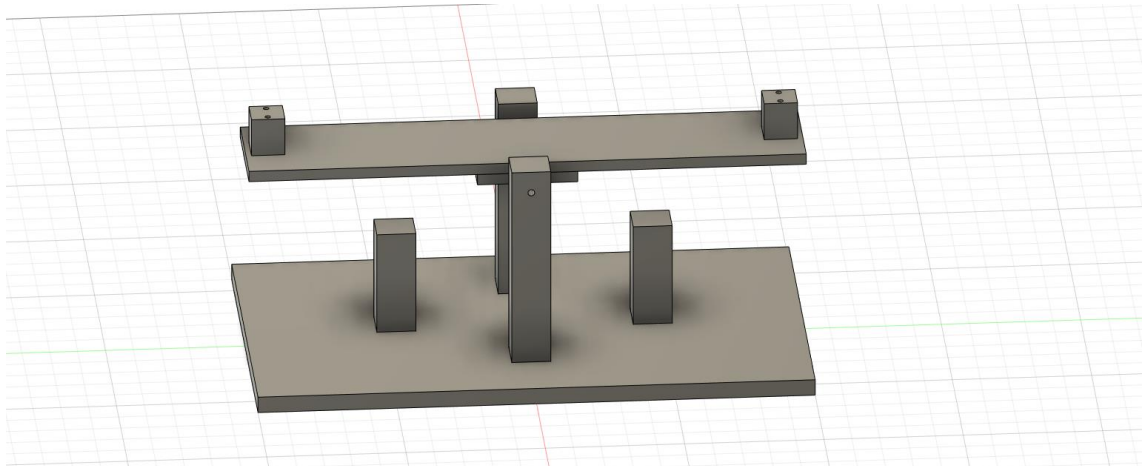
Imatge 13 i 14. Esquemàtic i PCB de la placa de control.

Tal com s'ha comentat, l'objectiu d'aquest disseny és el de fer més accessibles els pins de les comunicacions. És una PCB que va muntada just a sobre de la launchpad.

En aquesta part cal comentar com s'han extret els paràmetres PID. S'ha aplicat una diferència de duty cycle a cada un motor, un al màxim i un al mínim. Cada cop que es passava per l'angle 0 es canviava aquesta excitació, el que estava al mínim es passava al màxim i a l'inrevés. Fent aquesta petita prova s'ha extret el temps d'oscil·lació, és a dir, des que es canviaven les tornes fins que el balancí tornava a passar per 0. L'últim paràmetre que ens queda és l'angle màxim, que simplement s'ha aconseguit fent un bucle acumulatiu que es quedés amb el valor més gran i l'imprimís per consola. Un cop els havíem aproximat, s'han fets diferents tests per poder ajustar-los a base de prova i error.

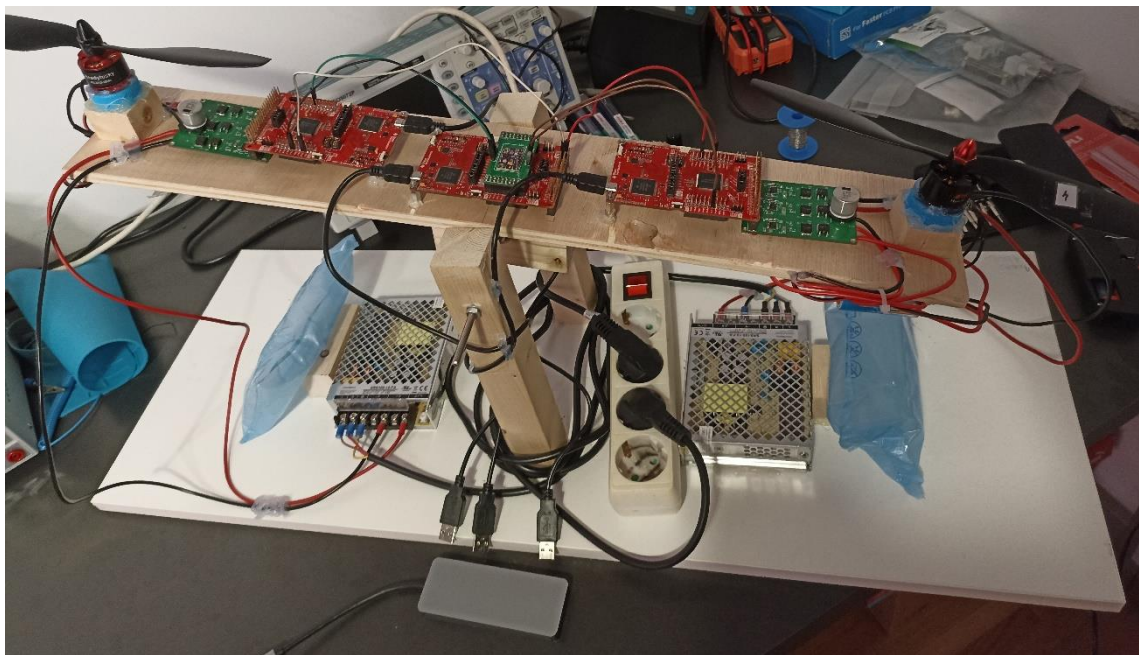
Una dels problemes trobats en la comunicació SPI entre el control i els ESCs és que depenent del moment en que s'iniciés els mòduls trobàvem un desfasament en l'enviament de les dades. És a dir, els ESCs detectaven que s'estava rebent algun missatge per SPI però al comprovar què s'ha rebut, res coincidia amb l'esperat. Sempre s'ha d'iniciar primer el Màster i després els Slaves.

BALANCI



Imatge 15. Disseny en 3D del balanci.

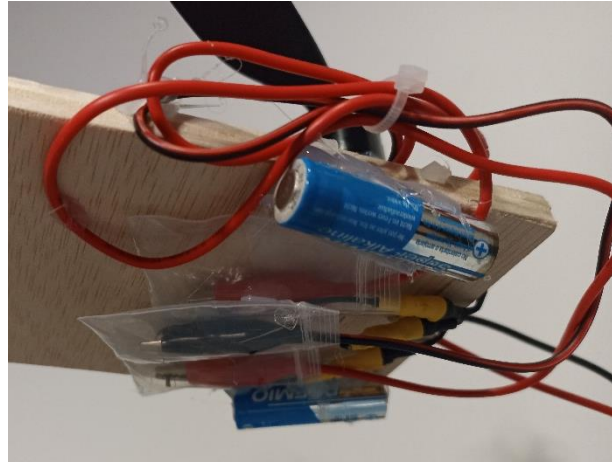
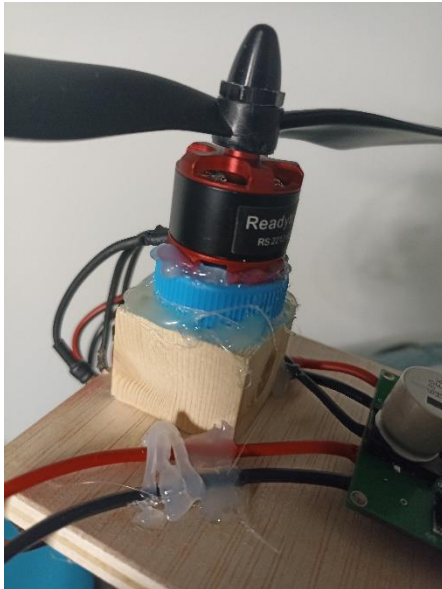
Com es pot apreciar, el disseny inicial i el final disten una mica. Es va haver d'adaptar el que es volia fer amb el que es podia, tal com es comentarà al apartat de viabilitat tècnica.



Imatge 16. Disseny final del balanci.

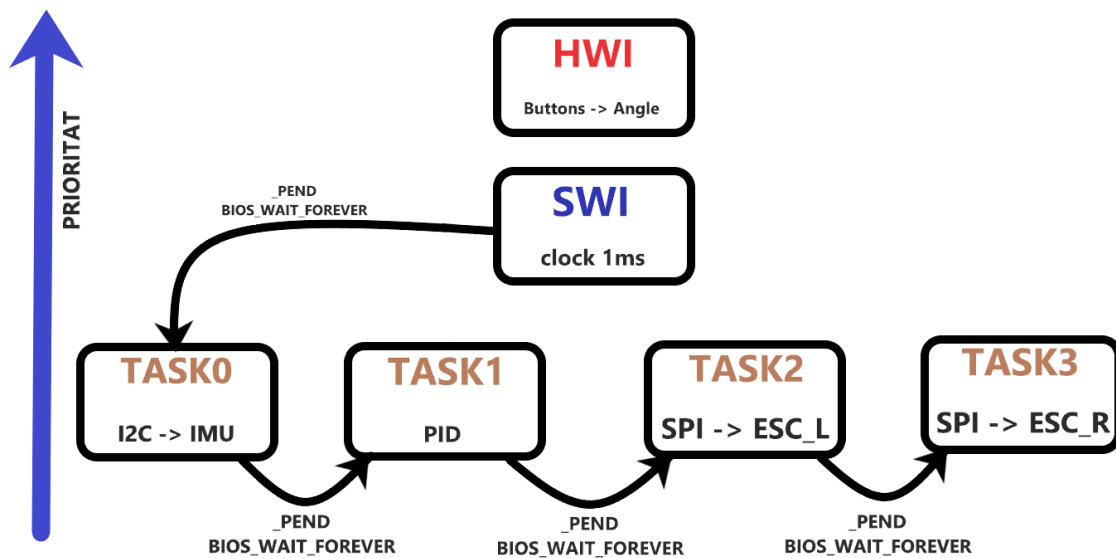
Uns dels problemes més rellevants és que es va realitzar el braç incorrectament, un costat és més llarg que l'altre i, per tant, el pes dels dos costats és diferent. No seria un problema si els motors fossin iguals, ja que es compensaria imposant un duty cycle màxim més alt al costat que pesi més. La solució ha estat la d'incorporar pes per sota del braç a cada un dels costats per compensar la diferència, no es perfecte, doncs al resultat final es nota cert desplaçament a la dreta, però per l'objectiu principal del projecte no és un fet rellevant. Sí ho és si volguéssim recuperar la posició 0° i partim des d'un angle molt gran, ja que els motors no tenen suficient força com per executar-ho i, com hem comentat, són diferents.

Aquí tenim dues imatges que mostren com s'han muntat els motors i els cables als braços. Com es pot veure, s'han utilitzat piles per poder afegir pes i quadrar els dos costats.



Imatge 17. Muntatge dels motors al braç del balancí.

ARQUITECTURA RTOS



Imatge 18. Arquitectura RTOS implementada.

Hwi: vinculada als botons integrats a la launchpad.

1. Es redueix o augmenta l'angle desitjat segons el botó pitjat, fins a un màxim de 20° o -20°.

Swi: vinculada a un clock d'1ms.

1. Fem un post d'un semàfor que desbloqueja la tasca 0.



Task0: tasca dedicada a la IMU.

1. Inicialitzem el mòdul. (1 sol cop)
2. Enviem la configuració desitjada a la IMU. (1 sol cop)
3. Demanem dades a la IMU.
4. Processem les dades rebudes.

Task1: dedicada al PID.

1. Calibrem el giroscopi i l'acceleròmetre.
2. Apliquem les fórmules de Euler per obtenir les dades del acceleròmetre.
3. Apliquem un filtre complementari amb les dades del giroscopi i les del acceleròmetre.
4. Calculem l'error.
5. Apliquem les fórmules PID i obtenim el valor dels dutys.
6. Desbloquegem amb un semàfor la tasca 2.

Task2: dedicada a la comunicació SPI amb el motor dret.

1. Inicialitzem el mòdul. (1 cop).
2. Enviem el caràcter pertinent segons el duty necessari un cop s'hagi desbloquejat la tasca.
3. Desbloquegem la tasca 3.

Task3: dedicada a la comunicació SPI amb el motor esquerra.

1. Inicialitzem el mòdul. (1 cop)
2. Enviem el caràcter pertinent segons el duty necessari un cop s'hagi desbloquejat la tasca.

VIABILITAT TÈCNICA

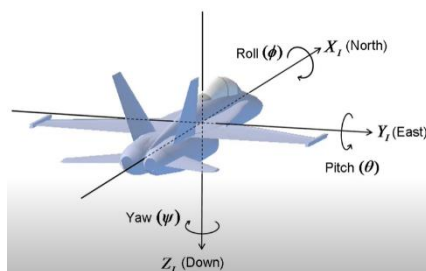
En aquest apartat es troba els aspectes tècnics d'algunes parts del projecte: limitacions a l'hora de realitzar les PCBs, mesures del balanç, configuracions dels mòduls dins del uC, etc.

Un dels primers passos va ser redissenyar les PCB recuperades del projecte adjunt a la memòria. Tot i que l'esquemàtic no canvia, es va voler millorar el layout per poder tenir una PCB més petita i millor implementada. Es va enviar a fabricar externament, a 'JLCPCB', i és aquí on trobem el primer detall tècnic a comentar. Per poder fabricar les PCB amb l'empresa mencionada, primer es va haver de revisar els requeriments[2]: mida de les vies, capes, 'clearance', serigrafies, etc. No és un procés complicat, simplement s'ha de modificar les regles del programa perquè al passar el DRC s'avisí d'un incompliment de les normes, si escau.

Un altre aspecte tècnic a comentar és el muntatge del balanç. Tal com s'ha vist al apartat de resultats, el disseny final dista una mica del disseny original. Vam haver de quadrar el que volíem amb el que es podia fer. L'estructura està formada per dos braços de fusta verticals que subjecten un braç horitzontal; s'uneixen a través d'una peça de fusta a la que s'ha afegit un rodament per poder fer oscil·lar el braç horitzontal. Tot això subjecte a una base, també de fusta, a la que s'ha afegit dos peces que serveixen de protecció contra situacions en les que els motors es poden descontrolar per una mala execució del programa. A més, s'ha col·locat els motors una mica per sobre del nivell del braç, per evitar xocs de les hèlices contra les plaques i aprofitar el vent generat per refrigerar de manera forçada els ESC.

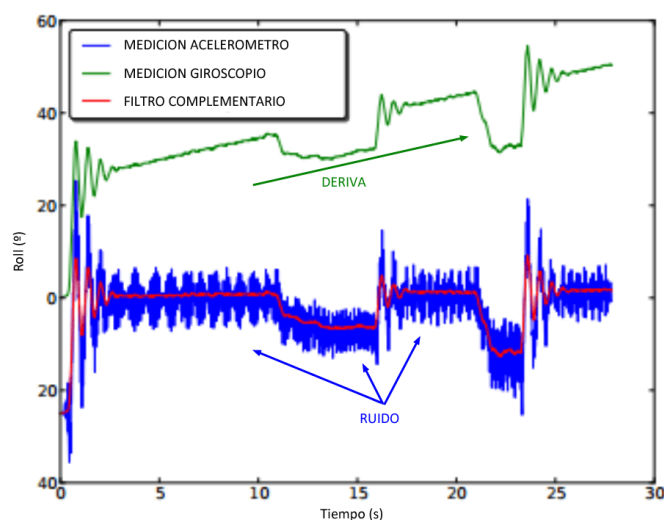
La configuració de la IMU és una mica complexa. Començant per giroscopi, aquest mesura la velocitat angular en $^{\circ}/s$. Si al resultat que llegim el dividim per l'escala escollida, tindrem el valor del angle en graus. Un altre aspecte a comentar és que el dispositiu necessita ser calibrat abans de tenir en compte les dades que proporciona; ho fem de manera senzilla, mirant el offset de les 2000 primeres mesures que agafi i restant el valor trobat a cada una de les mesures posteriors. I un cop sabem el temps entre mesures podem saber quanta distància ha recorregut.

Si només tinguéssim en compte les dades del giroscopi ens trobaríem que ens faltarien dades si, per exemple, el dispositiu girés 90° en el pla Y. El giroscopi no ho detectaria perquè no hi ha un canvi de la velocitat angular en els eixos que mesura (pitch i roll en la imatge del avió).



Imatge 19. Eixos que mesura el giroscopi.

Resolem aquesta qüestió amb les dades que ens proporciona l'acceleròmetre. Això sí, l'aportació total de la informació del acceleròmetre és molt menor, doncs si ens fixem en la imatge de més avall, varia molt el seu valor. Hem d'aconseguir que simplement sigui una petita aportació per tal de tenir la informació més exacte possible aplicant un filtre complementari. Un filtre complementari s'encarrega de sumar dues senyals sorolloses i cancel·lar-les entre sí.



Imatge 20. Representació de les dades del giroscopi, l'acceleròmetre i el resultat al aplicar el filtre.

VIABILITAT ECONÒMICA

En aquest apartat trobem la BOM final del projecte. Per una millor comparativa s'adjunta la referència de 2 webs on es possible adquirir el material.

ESCs

| Referència | Component | Mouser | Farnell | Preu |
|-------------------|------------------------|----------------------|-----------|---------|
| IC1,IC2,IC3 | Driver de porta | 942-IR2301SPBF | 3155038 | 10,56 € |
| C1, C3, C5 | Capacitat 10uF 1206 | 581-1206YD106K | 2281113 | 1,14€ |
| C11,C12,C13 | Capacitat 220nF 1206 | 603-CC126KRX7R7BB224 | 644250 | 1,50€ |
| C7,C8,C9,C10 | Capacitat 47nF 1206 | 581-12065C473K | 499377 | 1,12 € |
| C15 | Capacitat 600uF | 667-EEV-TG1E681UQ | 2326384RL | 3,14 € |
| R13,R14,R15,R17 | Resistència 33k 1206 | 667-ERJ-P08F3302V | 1750856 | 2,72 € |
| R10,R11,R12 | Resistència 33k 1206 | 667-ERJ-P08F3302V | 1750856 | 2,02 € |
| R7,R8,R9,R16 | Resistència 12.4k 1206 | 279-RP73D2B12K4BTG | 1501718 | 4,16 € |
| R1,R2,R3,R4,R5,R6 | Resistència 33 1206 | 667-ERJ-T08J330V | 1577436 | 1,37 € |
| Q1,Q2,Q3,Q4,Q5,Q6 | MOSFET tipus N | 942-IRLR2905TRLPBF | 2781154 | 14,88 € |
| D1,D2,D3 | Díode | 512-1N4148TR | 9843680 | 0,66 € |
| CONN6 | Connector | 710-61303621121 | 2356147 | 1,48€ |
| MSP432 | Lauchpad amb uC | 595-MSP-EXP432P401R | 3009876 | 20,16€ |
| | | | | 64,91€ |

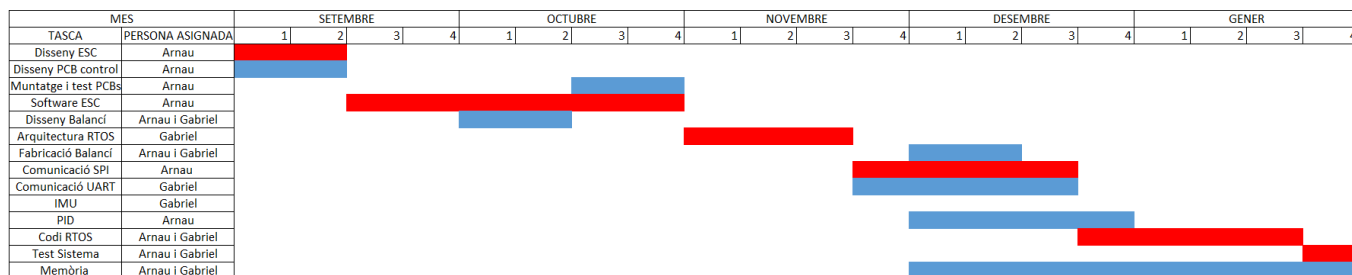
CONTROL

| Referència | Component | Digi-key | Amazon | Preu |
|--------------|---------------------|---------------------|---|--------|
| CONN1 | Header tipus mascle | 732-5317-ND | | 6,38€ |
| CONN2, CONN3 | Connector 10x10 | SAM15296-ND | | 0.32€ |
| IC1 | IMU MPU9250 | | https://amzn.to/3q6ng2z | 12,99€ |
| MSP432 | Lauchpad amb uC | 595-MSP-EXP432P401R | 3009876 | 20,16€ |
| | | | | 39,85€ |

Taula 1 i 2. BOM dels ESC i la placa de control.

CRONOGRAMA

En aquest apartat es presenta el diagrama de Gantt del projecte.



Imatge 21. Diagrama de GANTT del projecte.

Com veiem, s'ha dividit el projecte en diverses tasques i cadascuna té una persona assignada, o les dues si es una feina que precisa de quatre mans. En vermell veiem el camí crític i, com és natural, en paral·lel a aquest s'han realitzat altres tasques doncs moltes d'elles no requerien d'una anterior. En alguns casos sí, per exemple, no podríem muntar o programar els ESCs sense les PCB o no podríem fer tests al sistema sense tenir el balanci muntat.

CONCLUSIONS

En aquest apartat es repassen els objectius i es valora l'èxit o fracàs dels mateixos.

Revisant el primer objectiu, el qual fa referència al disseny i implementació de dues PCBs, una pel control i una altra pels ESC, podem afirmar que s'ha complert. Els nostres dissenys son funcionals.

Respecte al segon objectiu, que era el control dels motors BLDC en llaç tancat, també s'ha complert. Es realitza una rampa d'arrancada, amb un botó de reinici en cas de fallida, per posteriorment donar el control del gir al ADC i deixar al timer foca del joc.

Fent referència al tercer objectiu, que era el disseny i muntatge del balanci, també s'ha aconseguit. Com s'ha comentat, ens hem trobat certs problemes, com que un costat pesava més que l'altre, però que hem pogut anar resolent.

El quart objectiu proposava realitzar el codi en RTOS de la launchpad de control. Respecte als 3 protocols de comunicació: I2C per la comunicació amb la IMU, SPI per la comunicació entre la launchpad de control amb els ESC i UART per la comunicació entre el PC i la launchpad de control. Els dos primers han estat un èxit, però el protocol UART no s'ha pogut realitzar per qüestió de temps. S'ha realitzat un prototip del codi, però no s'ha pogut provar; tot i que no dista molt dels ja realitzats. Tot i així, el codi és funcional, per tant, podem dir que s'ha complert, però no al 100%.

El cinquè i últim objectiu, que era la generació d'un sistema PID, podem afirmar que ha estat un èxit. El nostre sistema és controlat per l'algorisme que es preocupa de corregir l'error en l'angle d'inclinació.

Analitzant els resultats i el diagrama de Gantt podem qualificar el projecte d'èxit ja que l'objectiu principal, que era equilibrar un balanci mitjançant l'ajust de la velocitat dels motors a través del coneixement de l'angle d'inclinació proporcionat per una IMU,

Per acabar, ens agradaria fer una menció especial als professors de l'assignatura que sempre han estat disponibles per qualsevol dubte o qüestió.



BIBLIOGRAFÍA

- [1] <https://www.youtube.com/watch?v=jh9dXNzTzz8>
- [2] <https://jlcpcb.com/capabilities/Capabilities>
- [3] <https://www.eng.newcastle.edu.au/~jhb519/teaching/caut1/Apuntes/PID.pdf>
- [4] <https://www.youtube.com/watch?v=uN8SYfGwYVw>
- [5] <https://www.youtube.com/watch?v=4BoIE8YQwM8>
- [6] <https://longnight975551865.wordpress.com/2018/02/11/how-to-read-data-from-mpu9250/>
- [7] <https://e2e.ti.com/support/microcontrollers/msp-low-power-microcontrollers-group/msp430/f/msp-low-power-microcontroller-forum/653367/msp432p401r-msp432-spi-3-wire-increment-issue>

ANNEXOS

1. MEMÒRIA DE LA REALITZACIÓ D'UN ESC PER UN MOTOR BLDC
2. CODI DEL CONTROL EN LLAÇ TANCAT D'UN BLDC
3. CODI EN RTOS PEL CONTROL DEL ANGLE D'INCLINACIÓ DEL BALANCI