



**EPS**

Escola Politècnica  
Superior

## Projecte/Treball Fi de Carrera

**Estudi:** Enginyeria Tècn. Ind. Electrònica Ind. Pla 2002

**Títol:** Disseny i implementació d'un sistema de navegació inercial i la seva aplicació a la robòtica mòbil

**Document:** 1. Memòria

**Alumne:** Joel Muñoz Puigdevall

**Director/Tutor:** Lluís Pacheco Valls

**Departament:** Arquitectura i Tecnologia de Computadors

**Àrea:** ATC

**Convocatòria (mes/any):** juliol / 2012

1. INTRODUCCIÓ .....	3
1.1. Antecedents .....	3
1.2. Objecte.....	3
1.3. Especificacions i abast .....	3
2. ENTORN DE DESENVOLUPAMENT USB-PIC'SCHOOL .....	4
2.1. El hardware .....	5
2.2. El Depurador/Gravador .....	6
3. COMPONENTS ELECTRÒNICS BÀSICS .....	7
3.1. Microcontrolador.....	7
3.1.1 El PIC 16F886 .....	9
3.2. Acceleròmetre .....	9
3.2.1 El BMA180.....	11
3.2.2 Funcionament del BMA180.....	12
3.3. Giroscòpic .....	13
3.3.1 L'IXZ500 .....	14
3.4. Brúixola .....	15
3.4.1 L'HMC6352.....	16
3.4.2 Funcionament de l'HMC6352.....	17
3.5. Pantalla LCD .....	17
3.5.1 La pantalla LCD compatible amb HD44780 .....	17
3.5.2 Controlador HD44780 .....	18
3.6. Dispositius electrònics complementaris .....	19
3.6.1. Convertidor MAX232.....	19
3.6.2. Convertidor de nivell lògic.....	19
3.6.3. Reguladors LM317 i LM7805 .....	21
4. PROTOCOLS DE COMUNICACIÓ SÈRIE .....	22
4.1. Comunicació sèrie síncrona I <sup>2</sup> C.....	22
4.2. Comunicació sèrie asíncrona RS-232 .....	25
5. ASSAJOS AMB ELS SENSORS.....	27
5.1. Llenguatge assemblador per PIC .....	27
5.2. Llibreries i funcions desenvolupades .....	29
5.3. Assaig 1: lectura de l'acceleració lineal .....	31
5.4. Assaig 2: lectura de l'orientació .....	35
5.5. Assaig 3: lectura de la velocitat angular.....	38
6. DISSENY I IMPLEMENTACIÓ DEL CIRCUIT IMPRÈS .....	41
6.1. Funcionament.....	41
6.2. Alimentació.....	41

6.3. Adquisició de dades .....	42
6.4. Transmissió de dades .....	46
6.5. Funcionament del programa.....	47
6.6. Disseny del circuit imprès.....	50
7. INTERFÍCIE ENTRE LA PLACA MICROCONTROLADA I EL PC.....	51
7.1. Desenvolupament del programa de comunicacions .....	51
7.2. Processament de les dades i representació .....	52
8. EINES MATEMÀTIQUES PER LA LOCALITZACIÓ ESPAIAL.....	54
8.1. Matrius de transformació homogènia.....	54
8.2. Càlcul de l'orientació inicial.....	54
8.3. Càlcul de la trajectòria .....	56
9. ANÀLISI DELS RESULTATS EXPERIMENTALS OBTINGUTS.....	59
9.1. Experiències amb la placa de prototipatge .....	59
9.2. Resultats finals.....	66
10. RESUM DEL PRESSUPOST.....	70
11. CONCLUSIONS.....	71
12. RELACIÓ DE DOCUMENTS .....	72
13. BIBLIOGRAFIA.....	73
14. GLOSSARI .....	75
A. PROGRAMA INTEGRAT EN EL MICROCONTROLADOR (ASSEMBLADOR).....	77
B. PROGRAMA DE RECEPCIÓ DE DADES (C) .....	81
C. PROGRAMA DE CàLCUL I SIMULACIÓ DE LA TRAJECTÒRIA (MATLAB).....	83

## **1. INTRODUCCIÓ**

### **1.1. Antecedents**

La UdG disposa de diversos robots mòbils per a finalitats docents i de recerca. Actualment, els sistemes de localització incremental emprats utilitzen bàsicament “encoders incrementals” per a dur a terme aquestes tasques. Altrament, en el mercat de l'automoció s'ha desenvolupat una sèrie de dispositius electrònics de baix cost tal com brúixoles electròniques, acceleròmetres, sensors giroscòpics, etc que permeten el desenvolupament de sistemes de navegació inercial.

### **1.2. Objecte**

L'objectiu és dissenyar i construir un sistema de navegació inercial. Per fer-ho s'implementarà un circuit imprès basat en un microcontrolador. Aquest sistema ha de llegir les dades de diversos sensors inercials ja esmentats anteriorment. El llenguatge de programació emprat serà l'assemblador ja que es pretén una execució molt eficient de les rutines creades. Per analitzar el funcionament dels sistema s'utilitzarà la plataforma robotitzada PRIM (robot autònom no humaniode) i es crearà un conjunt de rutines de comunicació via sèrie amb llenguatge d'alt nivell per tal de poder transmetre la informació a un PC extern. Es vol representar i contrastar les dades obtingudes pel sistema de navegació inercial amb les obtingudes pel sistema odomètric actual. Per fer-ho s'utilitzarà l'entorn MATLAB.

També es tindran en compte alguns paràmetres en el disseny del projecte ja que també està pensat aplicar-lo en una altre projecte d'un submarí realitzat per un altre company.

### **1.3. Especificacions i abast**

Per a dur a terme el projecte es farà ús dels dispositius disponibles al laboratori de la UdG, des de sensors i microcontroladors fins a l'entorn de desenvolupament “USB-PIC'School” i la pròpia plataforma PRIM. S'empraran tant els coneixements adquirits en la recerca per al desenvolupament del propi projecte com els adquirits durant la carrera.

## 2. ENTORN DE DESENVOLUPAMENT USB-PIC'SCHOOL

Les raons per les quals s'ha escollit aquest entorn són per la fàcil interacció entre la plataforma i l'usuari, l'ampli ventall d'aplicacions que ofereix en assemblador i, sobretot, per la disponibilitat al laboratori de la universitat. L'elecció del llenguatge assemblador com a llenguatge de programació del PIC és deguda que, entre els tres possibles (Basic, C i assemblador), permet l'aprofitament eficient dels recursos del PIC, el control dels temps i els registres bit a bit, la manipulació d'interrupcions simultànies i una completa optimització en la generació de l'arxiu .hex.

Així doncs, el control total intern del PIC i l'eficiència de les rutines creades permetrà desenvolupar un circuit capaç d'obtenir les dades de l'entorn a temps real de l'ordre de milisegons i fins i tot microsegons.

La universitat ens proporciona la plataforma "USB-PIC'School":



Figura 1. Laboratori USB-PIC'School

Es tracta d'un equip preparat per treballar-hi la major part dels dispositius PIC més representatius de les famílies 12F, 16F, i 18F, amb 8, 18, 28 i 40 pins, de manera que s'hi pugui desenvolupar aplicacions tant a nivell de software com de hardware. Disposa d'un

ampli i representatiu nombre dels perifèrics més utilitzats en les aplicacions reals, així com un mòdul de prototipatge per al muntatge sense soldadura que permet implementar el hardware necessari en un determinat projecte.

És el propi usuari qui, mitjançant un senzill cablejat sense soldadura, connecta i associa els diferents perifèrics a aquestes línies d'E/S segons ho requereixi l'aplicació. Això permet tenir un contacte més directe amb el hardware i un major control del mateix, que és imprescindible a la fase d'aprenentatge de microcontroladors i/o en el desenvolupament d'aplicacions basades en els mateixos.

## 2.1. El hardware

Les característiques més rellevants del laboratori USB-PIC'School són que admet dispositius PIC de 8, 18, 28 i 40 pins de les famílies 12FXXX, 16FXXX i 18FXXX; oscil·lador integrat de cristall de quars (fàcilment intercanviable) per generar la freqüència general de treball del sistema; configuració mitjançant "jumpers" de les senyals CLK/GP5, CLK/RA7, MCL/GP3, MCL/RA5 i MCL/RE3; integra el hardware necessari per la depuració i/o gravació del dispositiu, el qual és compatible amb el PICKIT2 de Microchip i, per tant, es pot controlar mitjançant les seves eines de software originals (MPLAB i PICKIT); tres modes d'operació diferents que es seleccionen fàcilment mitjançant un commutador lliscant, el mode RUN s'empra quan el dispositiu està gravat amb el programa de l'aplicació a executar, el mode USB s'utilitza durant els cicles de depuració/gravació del dispositiu, que es realitza mitjançant un port USB d'un PC dotat de l'entorn de desenvolupament MPLAB de Microchip i, finalment, el mode ICSP permet connectar el laboratori USB-PIC'School amb altres eines externes dotades de les senyals ICSP necessàries per la depuració i/o gravació en circuit, com poden ser l'ICD2 o l'ICD3 de Microchip; no hi ha connexions predeterminades, qualsevol perifèric es pot connectar a qualsevol línia d'E/S; interfície sèrie estàndard RS232; rellotge/calendari en temps real (RTC) i sensor de temperatura per mitjà del protocol I2C i 1-Wire; 8 sortides digitals; dos displays de 7 segments; 4 sortides digitals d'alta intensitat mitjançant drivers de 4 canals a 0.6A cadascun, al qual la tensió de sortida es selecciona a través d'un jumper i pot ser fixa de +5V o bé una tensió externa de referència de fins a 36V; pantalla LCD de 2 x 16 caràcters i connectors que permeten la interfície a 4 o 8 bits; teclat matricial de membrana de 4 x 4 tecles; 8 entrades digitals formades per 4 interruptors i 4 polsadors; 2 entrades per mitjà de potenciòmetres que permeten avaluar i simular variables analògiques; generador lògic d'ona quadrada amb freqüència de sortida ajustable a 1Hz, 10Hz, 100Hz i 1kHz; i mòdul de prototipatge de 550 contactes.

## 2.2. El Depurador/Gravador

Una de les grans aportacions de l'equip és la pròpia integració del hardware i firmware que permet la depuració/gravació en circuit de l'aplicació que s'estigui desenvolupant.

Disposem doncs d'una potent eina per depurar en temps real qualsevol disseny que estiguem desenvolupant. Un cop depurats i funcionant també podem procedir a gravar aquests programes sobre la memòria interna del controlador. Això permet que, com a software de control, es pugui utilitzar el conegut entorn de desenvolupament MPLAB-IDE des del qual es realitzen totes les operacions típiques en el desenvolupament d'un programa: edició del programa font, assemblatge, depuració i gravació. També es pot emprar el software de gravació PICKIT2.

Les capacitats de depuració del USB-PIC'School permeten l'execució del programa d'aplicació en temps real, amb punt de ruptura (BreakPoints) i pas a pas. Així, és possible visualitzar els registres interns i fer un seguiment analitzant el contingut dels mateixos conforme es van executant les instruccions del programa. També és possible modificar aquest contingut per estudiar i seguir la repercussió que això pot provocar en el curs de l'execució. Tot això en temps real i sense extreure el PIC de la plataforma, utilitzant la tècnica coneguda com ICD (In Circuit Debugger).

MPLAB-IDE consisteix en un entorn de treball que permet realitzar, sense sortir d'aquest ni fer ús d'altres eines de software, totes les tasques pròpies del disseny d'aplicacions basades en microcontroladors PIC de 8 (com és el nostre cas), 16 i 32 bits.

### 3. COMPONENTS ELECTRÒNICS BÀSICS

En aquest apartat es definiran els dispositius electrònics utilitzats en el projecte, tant els emprats en els assajos que més tard es comentaran, com en el desenvolupament del circuit imprès final. S'han intentat utilitzar els que hi havia disponibles al laboratori, tot i que en algun cas hi ha hagut la necessitat de comprar-ne d'addicionals com veurem més endavant. Els podem englobar en 6 grups en funció de la importància que adquireixen en l'aplicació final: el microcontrolador, l'acceleròmetre, el giroscòpic, la brúixola, la pantalla LCD i altres dispositius electrònics.

En els següents apartats no es descriu la informació detallada de cada dispositiu ni el seu funcionament detallat (ja que es pot consultar fàcilment al propi full de característiques), sinó la relació que té amb el projecte i les característiques més rellevants.

#### 3.1. Microcontrolador

Un microcontrolador és un circuit programable capaç d'executar ordres gravades a la seva memòria. Està compost per varis blocs funcionals, els quals compleixen una tasca específica. Un microcontrolador inclou al seu interior les tres unitats funcionals d'un ordinador: unitat central de processament, memòria i perifèrics d'entrada/sortida.

Els PIC, Peripheral Interface Controller (controlador d'interfície perifèrica), són una família de microcontroladors tipus RISC fabricats per "Microchip".

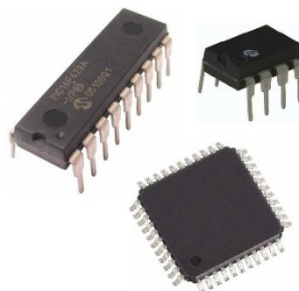


Figura 2. Conjunt de PICs de diferents tipus

Utilitzen un grup d'instruccions tipus RISC, el nombre del qual pot variar des de 35 per PICs da gama baixa a 70 pels de gama alta. Les instruccions es classifiquen entre les que



realitzen operacions entre l'acumulador i una constant, entre l'acumulador i una posició de memòria, instruccions de condicionament i de salt/retorn, implementació d'interrupcions i una per passar a mode de baix consum anomenada "sleep".

L'arquitectura del PIC utilitzat es caracteritza per l'àrea de codi i de dades separades (Arquitectura Harvard); un reduït nombre d'instruccions de longitud fixa; les instruccions s'executen en un únic cicle d'instrucció (4 cicles de rellotge), encara que algunes en 2 cicles; un únic acumulador (W); totes les posicions de la RAM funcionen com a registres d'origen i/o de destí d'operacions matemàtiques i altres; una pila de hardware per emmagatzemar instruccions de retorn de funcions; una petita quantitat d'espai de dades direccional, normalment 256 bytes; l'espai de dades està relacionat amb la CPU, ports, i els registres dels perifèrics; i el comptador de programa també està relacionat dins de l'espai de dades, i és possible escriure-hi.

Els microcontroladors PIC tenen una sèrie de registres que funcionen com una RAM de propòsit general. Els registres de propòsit específic pels recursos de hardware disponibles dins el propi chip també estan adreçats a la RAM. La direccionalitat de la memòria varia depenent la línia de dispositius, i tots els dispositius PIC tenen algun tipus de mecanisme de manipulació de bancs de memòria que poden ser utilitzats per accedir memòria externa o addicional. Per implementar adreçament indirecte, s'empra un registre de "selecció de registres d'arxiu" (FSR) i un de "registre indirecte" (INDF).

Els PICs es caracteritzen per disposar de nuclis de CPU de 8/16 bits amb Arquitectura Harvard modificada; memòria Flash i ROM disponible des de 256 bytes a 256 kilobytes; ports d'E/S (típicament de 0 a 5.5V); temporitzadors de 8/16 bits; tecnologia Nanowatt per modes de control d'energia; perifèrics sèrie síncrons i asíncrons USART, AUSART, EUSART; convertidors A/D de 8, 10 i 12 bits; mòduls de captura i comparació PWM; controladors LCD; perifèric MSSP per comunicacions I<sup>2</sup>C, SPI i I<sup>2</sup>S; memòria EEPROM interna amb duració de fins a un milió de cicles de lectura/escriptura; perifèrics de control de motors; i suports d'interfície USB, de controlador Ethernet, CAN, LIN i Irda.

Així, el microcontrolador s'encarregarà de donar les ordres als diferents perifèrics i processar tota la informació.

### 3.1.1 EI PIC 16F886

Al laboratori de la universitat disposem de varis microcontroladors PIC: 12F508 de la gama baixa; 16F88, 16F886 i 16F887 de la gama mitja; 18F2550 i 18F4550 de la gama alta.

Podem començar suprimint el PIC de gama baixa 12F508 ja que no disposa de cap mòdul de comunicació sèrie. A partir d'aquí, sabent que tots els altres estan dotats d'una EEPROM de 256 bytes, d'un mòdul EUSART i de dos comparadors, s'ha escollit el més adient comparant les característiques de cadascun.

Device	Memòria de programa	Memòria de dades	I/O	Canals A/D	ECCP	MSSP	Timers (8/16 bits)
	Flash (paraules)	SRAM (bytes)					
PIC 16F88	7168	368	16	1	1	0	2/1
PIC 16F886	8192	368	24	11	1/1	1	2/1
PIC 16F887	8192	368	35	14	1/1	1	2/1
PIC 18F2550	32k	2048	24	10	2/0	0	1/3
PIC 18F4550	32k	2048	35	13	1/1	1	1/3

Taula 1. Comparació dels microcontroladors disponibles al laboratori

A simple vista ens podem fixar que els PICs 16F88 i 18F2550 no són aptes per a l'aplicació que volem dur a terme ja que no disposen de mòdul MSSP necessari per la comunicació I<sup>2</sup>C entre el microcontrolador i els sensors. Els altres 3 restants podrien ser perfectament útils, així, s'escollirà el PIC 16F886 ja que (a part de ser més econòmic) ens és suficient i té una mida més petita.

### 3.2. Acceleròmetre

Aquest dispositiu ens és necessari per a determinar el moviment lineal. Quan parlem d'acceleròmetre ens referim a un instrument capaç de mesurar acceleracions i desacceleracions lineals. Els valors obtinguts s'expressen en m/s<sup>2</sup>. Actualment, disposem d'acceleròmetres molt variats i de diferents principis de funcionament en el mercat: mecànic,

piezoelèctric, piezoresistiu, d'efecte Hall, de condensador i de tecnologia MEMS (sistemes microelectromecànics).

L'acceleròmetre mecànic és el més simple, construït a partir de la unió entre una massa ( $m$ ) i un dinamòmetre amb l'eix a la mateixa direcció que l'acceleració que es desitja mesurar. El dinamòmetre permet mesurar el mòdul de la força ( $F$ ), de manera que d'acord amb la Llei Fonamental de la Dinàmica (Segona Llei de Newton) podem obtenir-ne l'acceleració ( $a$ ).

$$a = F / m \quad (\text{Eq. 1})$$

L'acceleròmetre piezoelèctric és el més comú. El principi de funcionament es basa en la compressió d'un reticle cristal·lí piezoelèctric i conseqüent la càrrega elèctrica proporcional a la força aplicada. La dada que n'obtenim també és una força, així que l'acceleració resultant s'obté aplicant l'equació 1. El piezoresistiu en canvi, les deformacions físiques del material provoquen un efecte resistiu.

Altres tipus d'acceleròmetres són els d'efecte Hall, que utilitzen una massa sísmica on s'hi col·loca un imant i un sensor d'efecte Hall que detecta canvis en el camp magnètic. El de condensadors mesuren el canvi de capacitat elèctrica d'un condensador mitjançant una massa sísmica situada entre les plaques d'aquest, i al moure's fa canviar la corrent que circula entre les plaques del condensador.

I com hem dit anteriorment, les noves tecnologies avancen exponencialment i un dels exemples els trobem en la tecnologia MEMS (Sistemes Microelectromecànics) utilitzada per fabricar aquests tipus de sensors. El principi de funcionament d'aquests sensors és utilitzant qualsevol dels observats anteriorment però amb la particularitat que treballen a mides molt petites, unitats de mesura que van dels micròmetres als mil·límetres. Observant la figura 3 ens podem fer una idea de la mida micromètrica que ocupa aquesta tecnologia.

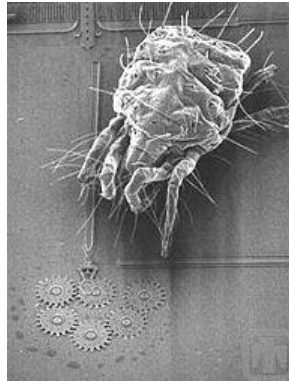


Figura 3. Un àcar a prop d'un grup d'engranatges produïts utilitzant MEMS

### 3.2.1 EI BMA180

Ja que el projecte requereix un acceleròmetre de dimensions reduïdes i màxima precisió, s'ha centrat l'atenció als sensors de tecnologia MEMS. En aquest cas el laboratori disposa d'un únic acceleròmetre, l'ADXL335 (desenvolupat per "Analog Devices"). Aquest sensor utilitza el mètode capacitatiu aplicat en tecnologia MEMS.



Figura 4. Placa amb l'acceleròmetre ADXL335 integrat

S'entén que el sensor és el dispositiu negre situat al centre de la placa de fàcil manipulació proporcionada per "SparkFun" (una empresa dedicada a la venda de tot tipus de dispositius electrònics preparats per una fàcil manipulació de cara a l'usuari, entre d'altres). Aquest sensor disposa d'unes característiques que no acaben de satisfer les necessitats: rang de mesura fix de  $\pm 3g$  i una precisió d' $11mg$ . La precisió és fàcil d'obtenir: alimentant el convertidor A/D del PIC a 3.3V tenim una resolució d'uns 32mV, segons el full de característiques de l'acceleròmetre sabem que la resolució és de 300mV/g, i segons l'equació 3 obtenim la precisió final de la mesura.

$$\text{Resolució A/D} = 3.3 / 1024 = 3.2\text{mV}$$

(Eq. 2)

$$\text{Precisió} = 3.2 / 300 = 11\text{mg} \quad (\text{Eq. 3})$$

Així doncs, s'ha decidit de comprar-ne un altre amb el mateix principi de funcionament: l'acceleròmetre BMA180 (desenvolupat per "Bosch"). Aquest és de tipus digital i es caracteritza per la mesura d'acceleració en els 3 eixos de coordenades; de tipus digital, amb interfície SPI i I<sup>2</sup>C; rang de mesura variable a  $\pm 1\text{g}$ ,  $\pm 1.5\text{g}$ ,  $\pm 2\text{g}$ ,  $\pm 3\text{g}$ ,  $\pm 4\text{g}$ ,  $\pm 8\text{g}$  i  $\pm 16\text{g}$ ; filtre intern programable de 10 a 1200 Hz (estretament lligat al temps d'obtenció de les dades i l'estabilitat); i una precisió de 0.25mg amb el rang de  $\pm 2\text{g}$ .

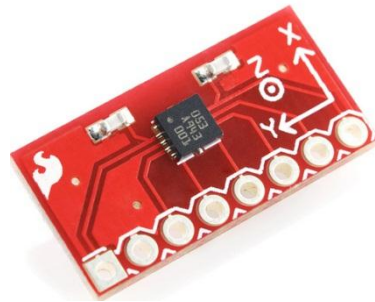


Figura 5. Placa amb l'acceleròmetre BMA180 integrat

### 3.2.2 Funcionament del BMA180

Quan ens referim a sensors digitals no ens referim a sensors que donen uns valors els quals llegim (com els analògics), sinó a sensors capaços de ser programats en diferents modes de funcionament per mitjà de l'escriptura dels propis registres interns. Seguidament es farà una breu descripció del funcionament de l'acceleròmetre BMA180.

Tenim primerament el tipus de memòria i com funciona. La memòria del BMA180 està constituïda per diferents arquitectures físiques. Bàsicament utilitza registres de memòria volàtils per operar. La part volàtil de la memòria pot ser canviada i llegida ràpidament. Part de la memòria volàtil ("imatge") és una còpia de la memòria no volàtil (EEPROM). L'EEPROM pot ser utilitzada per establir els valors per defecte per l'operació del sensor. Aquesta memòria és d'escriptura indirecta únicament. Els valors dels registres de l'EEPROM són copiats al registre imatge després de cada engegada o d'un soft reset. La descàrrega dels bytes de l'EEPROM a la imatge també es fa quan el contingut d'aquesta ha canviat per una comanda d'escriptura. Després de cada comanda d'escriptura l'EEPROM ha de ser reiniciada per un soft reset.

En quan als modes de funcionament: low power mode, low noise mode, ultra low power mode, ultra low noise mode, sleep mode (consum gairebé nul) i wake-up mode (per sortir del mode sleep). Per defecte tenim el low noise mode (que s'adapta a les nostres necessitats), i el fabricant ens aconsella no canviar-lo ja que posteriorment s'haurien de reajustar altres valors de difícil i precís calibratge.

Seguidament els diferents tipus d'interrupcions que ofereix aquest sensor: Low-g interrupt logic (quan l'acceleració baixa d'un valor preestablert), High-g logic (quan l'acceleració sobrepassa un valor preestablert), Slope detection (quan hi ha un canvi bruscat d'acceleració), Tap sensing i Alert mode (aquests dos modes són combinacions dels altres modes d'interrupció). En el nostre cas no ha estat necessària la utilització de cap mode d'interrupció ja que el que ens interessa és saber l'acceleració en cada instant de temps independentment de moviment efectuat. Així, emprarem la interrupció bàsica que ens avisa quan tenim una nova dada.

Pel que fa al rang de funcionament, s'ha optat per el de  $\pm 2g$  ja que és suficient i ens proporcionarà prou precisió en la mesura. El filtre programable s'ha configurat a 10 Hz per la següent raó: al fer proves al laboratori s'ha observat que aquest sensor era molt inestable i tot i que aquest filtre fa augmentar el temps d'adquisició de dades, és suficient per el nostre propòsit (lectura i transmissió de dades al PC cada 50ms).

La comunicació es pot dur a terme per mitjà dels protocols I<sup>2</sup>C o SPI. S'ha escollit el primer protocol perquè és el mateix que s'utilitzarà per la brúixola i es poden aprofitar les mateixes línies de comunicació (a nivell de hardware) i llibreries (a nivell de firmware). L'adreça que té establert l'acceleròmetre és 80h i 81h (escriptura i lectura, respectivament) connectant el pin SDO a terra, i 82h i 83h (escriptura i lectura, respectivament) connectant el pin SDO a 3.3V.

### 3.3. Giroscòpic

Aquest dispositiu ens és necessari per a determinar el moviment rotatiu. El giroscòpic és un mecanisme capaç de detectar el moviment giratori. Hi ha varis tipus de giroscòpics, tots ells diferenciats pel seu principi de funcionament: piezoelèctrics, capacitius i de tecnologia MEMS.

Els piezoelèctrics estan formats bàsicament per un cristall oscil·lador. Aquesta oscil·lació sorgeix quan hi actua una acceleració que fa que els elements que el suporten

(piezoelèctrics) es deformin i per tant variï la seva resistència elèctrica. Aquests canvis de resistència elèctrica són mesurables i permeten saber el valor de la força que ha actuat i també l'acceleració (relacionant-les mitjançant Coriolis).

El giroscòpic capacitiu està format per dues parts: una de mòbil que realitza u moviment de vibració respecte a l'eix de rotació del giroscòpic i una altra de fixa perpendicular a aquest moviment. Aquestes dues parts formen una estructura capacitiva que pot emmagatzemar una carrega elèctrica (el condensador). Aquesta càrrega reflecteix la velocitat de rotació del sistema.

Finalment, igual que amb l'acceleròmetre, apareix la tecnologia MEMS per reduir al màxim aquests sistemes mecànics i elèctrics. Utilitzant materials a nivell molecular aconseguint integrar en un chip de silici petites parts tridimensionals.

### 3.3.1 L'IXZ500

El laboratori disposa d'un giroscòpic fabricat en tecnologia MEMS de tipus analògic, l'IXZ500 (desenvolupat per "Honeywell") com es pot observar a la figura 6. S'ha observat el full de característiques i en un principi sembla que pot ser apte per l'aplicació que es vol dur a terme, per tant, s'ha testejat al laboratori i s'ha observat que és vàlid. En relació a la precisió, el full de característiques ens informa que ofereix, amb uns rangs de mesura de  $\pm 110^\circ/\text{s}$  i  $\pm 500^\circ/\text{s}$ , unes sensibilitats de  $9.1\text{mV}/^\circ/\text{s}$  i  $2\text{mV}/^\circ/\text{s}$  respectivament. Així, combinant l'equació 1 amb les següents equacions 4 i 5 sabrem la precisió de la mesura.

$$\text{Precisió } (\pm 110^\circ/\text{s}) = 3.2 / 9.1 = 0.35^\circ/\text{s} \quad (\text{Eq. 4})$$

$$\text{Precisió } (\pm 500^\circ/\text{s}) = 3.2 / 2 = 1.6^\circ/\text{s} \quad (\text{Eq. 5})$$

Com s'ha fet anteriorment amb l'acceleròmetre, definirem les característiques bàsiques d'aquest sensor i el mode de funcionament. Es tracta d'un sensor basat en un principi de funcionament piezoelèctric i capacitiu, on a partir de les vibracions produïdes es carreguen uns condensadors. Es caracteritza per tenir la capacitat de mesurar la rotació dels eixos X i Z en un únic chip; ser tipus analògic; tenir 2 rangs de mesura a escollir,  $\pm 110^\circ/\text{s}$  (alta precisió) i  $\pm 500^\circ/\text{s}$  (alta velocitat); estar dotat d'una precisió de  $0.35^\circ/\text{s}$  (pel mode de  $\pm 110^\circ/\text{s}$ )

i de  $1.6^{\circ}/s$  (pel mode de  $\pm 500^{\circ}/s$ ); i per disposar de la funció Auto Zero pel calibratge de l'offset.



Figura 6. Placa amb el giroscòpic IXZ500 integrat

A diferència de l'acceleròmetre, aquest sensor és mol més fàcil d'utilitzar ja que tots els valors ja venen calibrats de fabrica i al ser més simple, no disposa de la configuració de gaires paràmetres. Al alimentar-lo, es fa un Auto Zero i seguidament es llegeix el valor de les sortides analògiques.

### 3.4. Brúixola

La brúixola o compàs magnètic és un instrument que serveix d'orientació i té el seu fonament en la propietat de les agulles magnetitzades. Mer mitjà d'una agulla imantada senyala el Nord magnètic, així, permet saber l'angle respecte al nord. Ja que el laboratori de la universitat disposava d'aquest sensor, s'ha decidit implementar-lo com per contrastar les dades del giroscòpic



Figura 7. Brúixola magnètica convencional

El principi físic pel que funcionen les brúixoles radica en l'atracció dels pols magnètics i la tendència dels cossos magnetitzats a orientar-se en la mateixa direcció. La terra actua com



si tingués un gran imant al seu interior, amb els seus pols al llarg de l'eix de rotació terrestre. Per altra banda, l'agulla de la brúixola està imantada longitudinalment. Com que els pols magnètics s'atrauen, el nord de la brúixola va cap al pol sud magnètic de la terra (que coincideix amb el nord geogràfic), de manera que l'agulla de la brúixola senyala on està realment el nord. Un inconvenient molt important a tenir en compte és que a causa de la sensibilitat als camps magnètics, aquest sensor no és apte per algunes aplicacions on hi intervinguin variacions de camps magnètics.

Es poden diferenciar sis grans grups de brúixoles: de caixa de rellotge, cartogràfica, d'espill, de mirall, nàutiques i digitals. No s'entrarà en detall per a cada grup de brúixola ja que el principi de funcionament és el mateix o molt semblant. Si que mencionarem que la tecnologia MEMS s'ha fet càrrec d'aplicar el funcionament de la brúixola i així aconseguir ampliar el mercat d'aquesta i les seves aplicacions.

### 3.4.1 L'HMC6352

La UdG té a l'abast una brúixola fabricada amb tecnologia MEMS i que ens serà molt útil pel projecte. És de tipus digital, el protocol de comunicació és el mateix que el de l'acceleròmetre, I<sup>2</sup>C. Es tracta d'un compàs digital integrat en un petit encapsulat que dona el valor de l'angle respecte el nord.



Figura 8. Placa amb la Brúixola digital HMC6352 integrada

Les principals característiques són la mesura de l'angle respecte el nord; de tipus digital, interfície per I<sup>2</sup>C; 3 modes d'operació seleccionables, a part, un mode de calibratge per ajustar els canvis produïts pels camps magnètics; i una precisió de 0.1 °.

### **3.4.2 Funcionament de l'HMC6352**

Com s'ha comentat anteriorment, aquest sensor ofereix 3 modes d'operació següents: Standby mode, Query mode i Continuous mode.

El primer mode funciona enviant la comanda de lectura. El sensor calcula el valor de l'angle, l'envia i espera per qualsevol nova comanda. El segon mode en canvi, calcula el valor de l'angle i espera a la comanda de lectura per enviar-la, així quan rebí una nova comanda de lectura enviarà la dada calculada prèviament i calcularà un nou angle. Finalment, el mode continu (el que s'utilitzarà pel projecte) es basa en el càlcul de l'angle contínuament i al rebre la comanda de lectura envia la dada calculada més recent.

La comunicació es du a terme per mitjà dels protocols I2C. L'adreça que té establerta la brúixola és 42h per operacions d'escriptura, i 43h per operacions de lectura.

## **3.5. Pantalla LCD**

### **3.5.1 La pantalla LCD compatible amb HD44780**

El fet de poder visualitzar els paràmetres i els resultats obtinguts en temps real de l'execució d'un programa és de gran importància ja que permet comprovar fàcilment i ràpidament el funcionament de l'aplicació que s'estigui provant. Estem parlant d'una pantalla LCD (pantalla de cristall líquid), prima i plana formada per un nombre de píxels en color o monocromes col·locats davant d'una font de llum o reflectora. Aquest perifèric permet representar les dades rebudes en forma de lletres i nombres.

Actualment existeixen varis tipus de pantalles LCD i de varies mides, però com que la plataforma que s'ha utilitzat (USB-PIC'School) per experimentar amb els sensors i altres dispositius ja portava implementada una pantalla LCD s'ha utilitzat aquesta, representada a la figura 9.



Figura 9. Pantalla LCD amb controlador compatible amb HD44780

És una pantalla alfanumèrica estàndard de dues línies de setze caràcters cada una, gestionada per un microcontrolador compatible amb l'Hitachi 44780.

### 3.5.2 Controlador HD44780

Com que quan treballem amb la pantalla LCD ho farem per mitjà del controlador HD44780, s'estudiarà el comportament d'aquest i el protocol de comunicació a seguir.

Aquest controlador disposa de 3 tipus de memòria: CGROM, CGRAM i DDRAM. A la CGROM (generador de caràcter de la ROM) hi ha emmagatzemats els caràcters que es poden visualitzar a la pantalla, a la CGRAM (generador de caràcter de la RAM) s'hi poden emmagatzemar els dissenys del propi usuari i la DDRAM (visualitzador de les dades de la RAM) és la memòria encarregada de guardar les dades visualitzades per pantalla.

El HD44780 té dos registres de 8 bits, un de comandes (per esborrar el display, el desplaçament del cursor o adreces de la CGRAM o DDRAM) i un de dades (emmagatzema temporalment les dades escrites o llegides a la CGRAM o DDRAM).

El HD44780 disposa de tres línies de control: l'activació de la línia d'activació (que permet a l'LCD llegir les altres línies de control), la línia d'escriptura/lectura (per llegir o escriure en els registres) i la línia selectora de registre (per triar el registre de comandes o de dades). Segons el valor d'aquests bits, el controlador pot saber si es vol llegir o escriure i on, mitjançant el bus de dades.

La comunicació pot efectuar-se per una interfície de dades de 4 o 8 bits. Pel nostre cas s'ha utilitzat la interfície de 4 bits. També cal saber que s'ha de seguir una rutina d'inicialització

(un conjunt de comandes separades per determinats retards) abans de començar la comunicació entre la pantalla LCD i la nostra aplicació. La informació detallada es troba al full de característiques del controlador HD44780, tot i que més tard es donarà més informació a l'apartat dels assajos.

### 3.6. Dispositius electrònics complementaris

Fins ara hem definit els dispositius electrònics més importants del projecte. En aquest apartat es farà referència a components no tan específics i més convencionals, però que també han estat imprescindibles pel desenvolupament del treball.

#### 3.6.1. Convertidor MAX232

El MAX232 és un circuit integrat (fabricat per "Maxim") que converteix els senyals d'un port sèrie RS-232 a senyals compatibles amb nivells TTL de circuits lògics. Aquest dispositiu serveix d'interfície de transmissió i recepció pels senyals RX, TX, CTS i RTS.



Figura 10. Circuit integrat MAX232

Aquest circuit integrat té sortides per manipular nivells de tensió RS-232 (aproximadament  $\pm 7.5V$ ) produïts a partir d'una tensió d'alimentació de 5V. Les entrades de recepció d'RS-232 poden arribar a  $\pm 25V$ . Aquestes es converteixen al nivell estandarditzat de 5V de la lògica TTL.

#### 3.6.2. Convertidor de nivell lògic

El convertidor de nivell lògic no es tracta d'un integrat, sinó d'un simple circuit construït a partir de resistències i MOSFETs (transistors d'efecte de camp) amb la funció de convertir 3.3V a 5V i el mateix a l'invers. Això és degut que tots els sensors treballen a 3.3V (tant l'alimentació com la interfície) i el microcontrolador ho fa a 5V. Per la comunicació a través

del bus I<sup>2</sup>C es necessita convertir la senyal que prové del microcontrolador de 5V a 3.3V pels sensors i el mateix a la inversa.

Pels assajos s'ha utilitzat el convertidor de nivell lògic disponible a "Sparkfun". Pel disseny de la placa final s'utilitzaran els MOSFETS i resistències per separat a causa de la millor maniobrabilitat que ofereix.

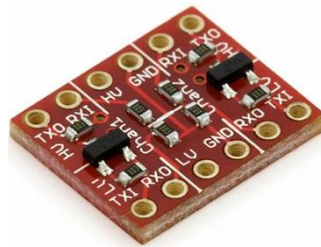


Figura 11. Convertidor de nivell lògic de dos ports (3.3V-5V)

Es basa en un circuit molt simple però de gran utilitat, així que s'ha vist la necessitat de descriure el funcionament d'aquest. Com s'observa a l'esquema de la figura 12, el MOSFET de canal N (significa que condueix per valors de  $V_{gs}$  positius) serveix d'intermediari entre els senyals  $V_1$  i  $V_2$ . Aquest dos senyals es configuren físicament per una resistència de pull-up a 3.3V i 5V per assegurar l'estat inactiu. El dispositiu que treballa a 3V correspondria al senyal  $V_1$ , el que treballa a 5V a  $V_2$ . En el cas que el dispositiu a  $V_1$  enviés un 1 lògic (3.3V),  $V_{gs}$  seria igual a 0V i el MOSFET es trobaria en estat de tall ( $V_2=5V$ ); si el dispositiu enviés un 0 lògic (0V),  $V_{gs}$  seria igual a 3.3V i el MOSFET es trobaria en estat de saturació ( $V_2=0V$ ). En el cas contrari, que el dispositiu a  $V_2$  enviés un 1 lògic (5V), l'estat del MOSFET no varia (estat de tall i  $V_1=3.3V$ ); si el dispositiu enviés un 0 lògic (0V), el díode intern del MOSFET condueix els 0V al sortidor i fa canviar el MOSFET a l'estat de saturació ( $V_1=0V$ .)

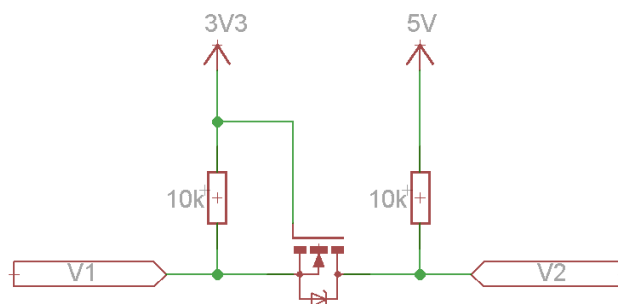


Figura 12. Esquema d'un port del convertidor de nivell lògic

### 3.6.3. Reguladors LM317 i LM7805

Tot circuit electrònic necessita una alimentació estable i amb el menor soroll possible. Aquesta és la funció dels reguladors LM317 i LM7805 utilitzats en el projecte. El primer ens subministra una tensió estable de 3.3V i el segon una tensió de 5V.

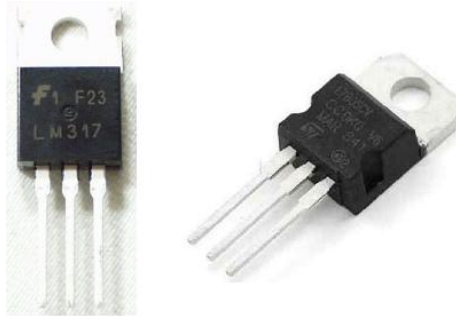


Figura 13. Esquema d'un port del convertidor de nivell lògic

Al mercat podem distingir dos grans grups de fonts d'alimentació: les lineals (transformen l'energia elèctrica mitjançant transistors polaritzats a la regió de polarització activa) i les commutades (treballen a la regió de tall i saturació a altes freqüències).

Per una banda, les fonts lineals tenen els avantatges de la seva simplicitat en el disseny, poc soroll de sortida, resposta dinàmica molt ràpida i cost dels components menor que el de les commutades (per potències menors a 10W). Però tenen el gran inconvenient de tenir un rendiment del 30 al 60%.

Per altra banda, les fonts commutades tenen un rendiment del 68 al 90% i poden funcionar com a dispositiu reductor i/o elevador de tensió.

Si tenim en compte que pel nostre projecte necessitem simplicitat en el disseny sense importar el rendiment, ja que la placa no s'alimenta per mitjà d'una bateria sinó directament de la xarxa elèctrica, la font d'alimentació lineal serà la millor opció.

## 4. PROTOCOLS DE COMUNICACIÓ SÈRIE

Quan parlem de protocol ens referim a un conjunt de regles utilitzades per establir la connexió entre 2 o més dispositius a través d'una xarxa o medi físic. En el projecte es fa ús del protocol de comunicació sèrie, on l'intercanvi d'informació es pot fer per mitjà d'un únic cable per rebre i/o transmetre, ja que els bits són transmesos un rere l'altre.

El microcontrolador és l'encarregat de comunicar-se tant amb els sensors (per a l'adquisició de dades) com amb l'ordinador (per enviar i tractar la informació). Així, utilitzarem el protocol de comunicació sèrie síncron I<sup>2</sup>C per a l'intercanvi d'informació entre el microcontrolador i els sensors i el protocol de comunicació sèrie asíncron RS-232 per enviar les dades del microcontrolador a l'ordinador.

### 4.1. Comunicació sèrie síncrona I<sup>2</sup>C

I<sup>2</sup>C (Circuits Inter-Integrats) és un bus de comunicacions en sèrie. Va ser desenvolupat per "Philips" al 1992. És un bus molt utilitzat en la indústria, principalment per comunicar microcontroladors i els seus perifèrics. La principal característica d'aquest bus és que utilitza dos línies per transmetre la informació: una per les dades i una altra per la senyal de rellotge. Per tant, es tracta d'una comunicació síncrona (les dades es sincronitzen juntament amb una senyal de rellotge) i Half-dúplex (comunicació en un únic sentit).

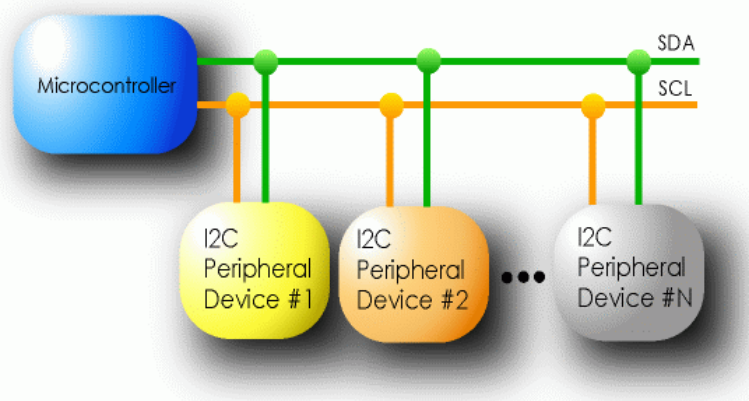


Figura 14. Diagrama del bus I<sup>2</sup>C i les connexions entre mestre i esclaus

El bus es basa en tres senyals: SDA (per la qual viatgen les dades entre els dispositius), SCL (per la qual transiten els polsos del rellotge que sincronitzen el sistema) i GND (interconnectada entre tots els dispositius del bus).

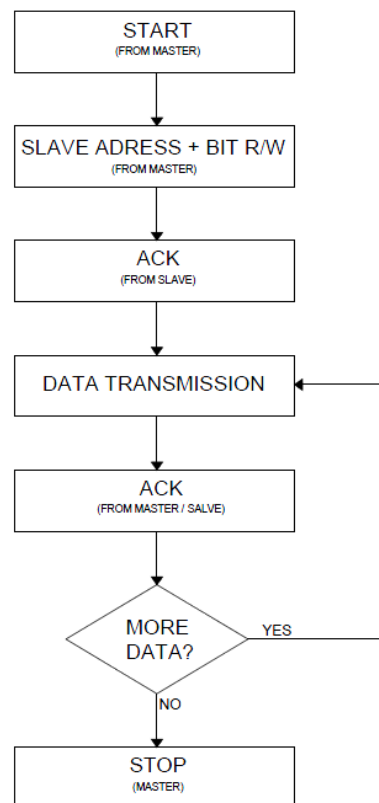
Les línies SDA i SCL són de tipus drenador obert, similars a les de col·lector obert però associades a un FET (transistor d'efecte de camp). Això significa que quan el bus estigui lliure, es pot produir un estat "flotant", és a dir, que no sigui estat alt ni estat baix. Per solucionar-ho s'ha de connectar una resistència de pull-up a cada una de les línies SDA i SCL per assegurar l'estat alt quan els bus estigui lliure.

La jerarquia que aplica aquest bus està definida de la següent manera: un Mestre ("Master") i un Esclau ("Slave"). El Master és el dispositiu que determina la temporització i la direcció del tràfic de dades al bus. És l'únic que aplica els polsos de rellotge a la línia SCL. Quan es connecten varis dispositius mestres a un mateix bus, la configuració obtinguda es denomina "multi-mestre". En aquest cas ho serà el microcontrolador. El Slave (Esclau) és qualsevol dispositiu connectat al bus incapaç de generar polsos de rellotge. Reben senyals de comanda i de rellotge provinent del dispositiu mestre. Els sensors compliran aquesta tasca.

El protocol del bus I<sup>2</sup>C disposa de les següents funcions: Start, Stop i ACK. L'Start (Inici) succeeix quan un dispositiu mestre ocupa el bus (inici de la comunicació), generant aquesta condició. La línia de dades (SDA) passa a estat baix mentre que la línia de rellotge (SCL) roman alta. L'Stop (Parada) pot ser generat per un dispositiu mestre per deixar lliure el bus (fi de la comunicació). La línia de dades passa a estat alt mentre que la de rellotge roman alta. L'ACK (bit de reconeixement): és un pols (alt-baix-alt) enviat tant per l'esclau com pel mestre. Aquest bit es transmet un cop rebuda la dada per indicar que aquesta ja s'ha rebut.

El protocol que permet aquesta comunicació està format per un seguit de passos, com es pot veure al diagrama de flux de la figura 15.



Figura 15. Diagrama de flux del protocol I<sup>2</sup>C

Tan aviat com el bus està lliure, un dispositiu mestre pot ocupar-lo generant una condició d'inici (Start). El primer byte transmès després de la condició d'inici conté set bits que componen l'adreça del dispositiu de destí seleccionat i un vuitè bit corresponent a l'operació desitjada (lectura o escriptura). Si la direcció coincideix amb la d'algun dispositiu connectat al bus, aquest respon enviant un bit de coneixement (ACK). Seguidament pot començar l'intercanvi d'informació entre els dispositius. Quan l'esclau rep una dada, aquest envia l'ACK al mestre (i el mateix procediment a l'invers). I finalment, quan el mestre decideix acabar la comunicació, genera una condició de parada (Stop). Si es desitja seguir transmetent, el dispositiu mestre pot generar una altra condició d'inici (Start) en lloc de la de parada (Stop). Aquesta nova condició d'inici es denomina "inici repetitiu" i es pot generar per adreçar un dispositiu esclau diferent o per alterar l'estat del bit de lectura/escriptura.

La velocitat de transmissió estàndard és de 100kHz, tot i que també s'utilitzen velocitats de 400kHz i 1MHz.

## 4.2. Comunicació sèrie asíncrona RS-232

El protocol RS-232 és una norma o estàndard mundial que regeix els paràmetres d'un dels modes de comunicació sèrie. Per mitjà d'aquest protocol s'estandarditzen les velocitats de transferència de dades, la forma de control que utilitza aquesta transferència, els nivells de voltatge utilitzats, el tipus de cable permès, les distàncies entre dispositius, les connectors, etc. A part de les línies de transmissió (TX) i recepció (RX), les comunicacions sèrie disposen d'altres línies de control de flux, l'ús del qual és opcional depenent del dispositiu a connectar. Pel projecte només són necessàries tres línies: RX, TX i GND.

Es tracta d'una comunicació asíncrona (les dades no es sincronitzen juntament amb una senyal de rellotge, sinó que cada dispositiu fixa la mateixa velocitat de transmissió) i Full-dúplex (comunicació en doble sentit).

A nivell de software, la configuració principal que s'ha de donar a una connexió a través dels ports sèrie RS-232 és bàsicament la selecció de la velocitat en bauds (1200, 2400, 4800, etc.), la verificació de dades o paritat (paritat parell, paritat imparell o sense paritat), els bits de parada després de cada dada (1 o 2) i la quantitat de bits per dada (7 o 8) que s'utilitza per cada símbol o caràcter enviat.

Com s'ha comentat anteriorment, aquest protocol permetrà al microcontrolador enviar les dades a l'ordinador. Així, el DTE (Equip Terminal de Dades) correspondrà al microcontrolador, i el DCE (Equip de finalització del Circuit de Dades) a l'ordinador.

Les senyals que utilitza aquest protocol són: Transmit Data, Recieve Data, Request To Send, Clear To Send, Data Terminal Ready, Data Set Ready, Recieve Signal Ready, Recieve Signal Line Detect i Ground. Transmit Data (TD o TX) és la línia pel qual la dada es transmet bit per bit. Recieve Data (RD o RX) és la línia pel qual la dada es rep bit per bit. Request To Send (RTS) s'envia, en el nostre cas, del microcontrolador (DTE) a l'ordinador (DCE) per indicar que es volen transmetre dades. Si l'ordinador hi està d'acord, contesta per la línia CTS. A partir d'aquí, el microcontrolador començarà a enviar dades. Clear To Send (CTS) és la senyal afirmada per l'ordinador després de rebre la senyal de RTS. Indica que el microcontrolador pot transmetre. Data Terminal Ready (DTR) és afirmada pel microcontrolador, i informa a l'ordinador que el microcontrolador està preparat per rebre dades. Data Set Ready (DSR) és afirmada per l'ordinador com a resposta a una senyal de DTR del microcontrolador. Aquest supervisa l'estat d'aquesta línia després d'afirmar DTR

per descobrir si l'ordinador està encès. Receive Signal Line Detect (RSLD) és afirmada per l'ordinador i informa al microcontrolador que s'ha establert una connexió física amb l'ordinador. I Ground (GND) serveix per referenciar el terra del bus.

## 5. ASSAJOS AMB ELS SENSORS

Un cop seleccionats els components electrònics i definit els protocols de comunicació per a la connexió entre tots els dispositius, es procedirà a posar a prova els tres sensors encarregats de recollir la informació del medi físic. Es durà a terme un assaig per a cada sensor per dominar el propi funcionament i aplicar-lo a la placa final que formarà el projecte. Tots els assajos es desenvoluparan sobre la plataforma USB-PIC'School, mitjançant l'entorn de treball MPLAB-IDE.

En primer lloc, s'experimentarà amb l'acceleròmetre BMA180 fent una petita pràctica de l'adquisició de l'acceleració lineal en l'eix Z, es a dir, el valor de la gravetat, i posteriorment la seva representació en la pantalla LCD. Aquesta part ens servirà per obtenir la translació del robot, per així calcular la posició en cada instant de temps.

En segon lloc, amb la brúixola HMC6352 determinarem l'orientació respecte el nord i comprovarem els efectes causats pels camps magnètics pròxims.

Finalment, es provarà el giroscòpic IXZ500 mostrant per pantalla LCD el valor de la velocitat angular en l'eix X (pitch) i en l'eix Z (yaw). Aquestes dades permetran saber la orientació en cada moment, d'aquesta manera sabrem la direcció del moviment.

Per a desenvolupar el firmware capaç de manejar els diferents protocols de comunicació i les diferents connexions ha estat necessari la creació de llibreries per a facilitar la programació i aportar molta més claredat al programa. Així que abans nombrarem i definirem les llibreries desenvolupades i les seves funcions corresponents.

### 5.1. Llenguatge ensamblador per PIC

Les instruccions per programar el microcontrolador es poden agrupar en tres grups: operacions d'orientació de byte, operacions d'orientació de bit i operacions literals i de control. Com s'observa a la taula 2, per les instruccions d'orientació de byte, "f" representa un registre i "d" representa una destinació. El registre especifica quin registre ha de ser utilitzat per la instrucció. La destinació especifica on ha de ser col·locat el resultat de l'operació. Si "d" és zero, l resultat es col·loca a l'acumulador (el registre d'operacions).

<b>OPERACIONS D'ORIENTACIÓ DE BYTE</b>		
<b>Instrucció</b>	<b>Descripció</b>	<b>Cicles</b>
ADDWF f,d	Suma W i f	1
ANDWF f,d	Funció AND de W amb f	1
CLRF f	Borra f	1
CLRWF -	Borra W	1
COMF f,d	Complementa f	1
DECF f,d	Decrementa f	1
DECFSZ f,d	Decrementa f i salta si zero	1
INCF f,d	Incrementa f	1
INCFSZ f,d	Incrementa f i salta si zero	1
IORWF f,d	Funció OR de W amb f	1
MOVF f,d	Mou f	1
MOVWF f	Mou W a f	1
NOP -	No opera	1
RLF f,d	Rotació a esquerra de f a través de Carry	1
RRF f,d	Rotació a dreta de f a través de Carry	1
SUBWF f,d	Resta W a f	1
SWAPF f,d	Commuta quartets a f	1
XORWF f,d	Funció XOR de W amb f	1

Taula 2. Operacions d'orientació de byte.

Per les instruccions d'orientació de bit, "b" representa el bit afectat per l'operació, localitzat a l'adreça del registre "f".

<b>OPERACIONS D'ORIENTACIÓ DE BIT</b>		
<b>Instrucció</b>	<b>Descripció</b>	<b>Cicles</b>
BCF f,b	Borra bit b de f	1
BSF f,b	Activa bit b de f	1
BTFSC f,b	Comprova bit b de f i salta si 0	1
BTFSS f,b	Comprova bit b de f i salta si 1	1

Taula 3. Operacions d'orientació de bit.

Per les instruccions literals i de control, “k” representa una constant de 8 o 11 bits, o un valor literal.

<b>OPERACIONS LITERALS I DE CONTROL</b>			
<b>Instrucció</b>		<b>Descripció</b>	<b>Cicles</b>
ADDLW	k	Suma W i k	1
ANDLW	k	Funció AND de W amb k	1
CALL	k	Crida a subrutina	2
CLRWDT	-	Borra el WTD	1
GOTO	k	Salta a la direcció k	2
IORLW	k	Funció OR de W i k	1
MOVLW	k	Mou k a W	1
RETFIE	-	Retorn d'interrupció	2
RETLW	k	Retorn carregant k a W	2
RETURN	-	Retorn de subrutina	2
SLEEP	-	Mode repòs	1
SUBLW	k	Resta W de k	1
XORLW	k	Funció XOR de W i k	1

Taula 4. Operacions literals i de control.

Un cycle d'instrucció consisteix en quatre períodes d'oscil·lació; per una freqüència d'oscil·lació de 4MHz, un cycle equivaldrà a 1us. La majoria de les instruccions són executades en un cycle d'instrucció, d'altres ho fan en dos.

## 5.2. Llibreries i funcions desenvolupades

A la taula 5 s'hi representen els arxius que conté cada assaig, tal com les llibreries i els programes. Com que el llenguatge de programació serà l'assemblador, els arxius que formen les llibreries portaran l'extensió .inc, i els propis programes principals portaran l'extensió .asm. Per incloure les rutines d'una llibreria, s'utilitza la funció include “nom\_de\_la\_libreria.inc” a l'inici del programa principal.

<b>Assaig acceleròmetre</b>	<b>Assaig brúixola</b>	<b>Assaig giroscòpic</b>
BMA180.asm	HMC6352.asm	IXZ500.asm
P16F886.inc		
BIN_BCD.inc		
LCD_4bits.inc		
I2C.inc		-

Taula 5. Llibreries de cada assaig.

Els arxius dels programes font de l'assaig de l'acceleròmetre, la brúixola i el giroscòpic són el BMA180.asm, el HMC6352.asm i el IXZ500.asm respectivament. Cada arxiu té induïdes les llibreries que necessita i el programa principal encarregat d'executar les instruccions i cridar les rutines pertinents.

La llibreria P16F886.inc és utilitzada per cada assaig ja que conté definits els registres del PIC 16F886 i els bits de cadascun dels registres.

La llibreria BIN\_BCD.inc (de fet és una funció) també està integrada en tots els sensors perquè s'utilitza per fer la conversió de les dades que arriben dels sensors en format binari a format decimal. Això és necessari ja que posteriorment es volen mostrar les resultats en una pantalla LCD per mitjà de la llibreria LCD\_4bits.inc, la qual està configurada per una comunicació per interfície de 4 bits entre el PIC i el controlador HD44780 de la pantalla LCD. I la llibreria I2C.inc només la comparteixen l'acceleròmetre i la brúixola, ja que estem parlant de sensors digitals on la comunicació es du a terme per mitjà del protocol sèrie síncron I<sup>2</sup>C. Aquesta llibreria conté totes les funcions necessàries que utilitza aquest protocol per la transmissió i recepció de dades.

Si ens fixem amb la taula 6, veiem les funcions que conté cada llibreria i la seva descripció. Més endavant veurem cada assaig amb l'esquema del circuit i les rutines desenvolupades, així que s'explicarà amb més detall. A més, la mateixa llibreria conté cada línia de codi comentada.

Llibreria	Funció	Descripció
BIN_BCD.inc	Iniciar_conversió	Converteix una dada de format binari a format decimal, amb un rang de 0 a 9999 en decimal.
LCD_4bits.inc	LCD_I	Inicialització de la pantalla LCD. Es configuren els registres del PIC per la comunicació amb el perifèric i s'inicialitza aquest amb la rutina corresponent.
	LCD_BF	Comprovem que l'LCD no estigui executant cap comanda perquè la pròxima instrucció s'executi correctament.
	LCD_dada	Esriptura de dades a la DDRAM o CGRAM.
	LCD_comand	Esriptura de comandes a l'LCD.
	LCD_val	Generar puls per a validar comanda.
	LCD_retard	Rutina de temporització (utilitzada en la rutina d'inicialització de l'LCD).
I2C.inc	I2C_I	Inicialització del protocol I <sup>2</sup> C. Ajusta el mòdul MSSP al mode "Master" amb freqüència de treball a 100kHz.
	I2C_Start	Envia la seqüència d'inici per la comunicació I <sup>2</sup> C.
	I2C_RStart	Envia la seqüència de reinici.
	I2C_Stop	Envia la seqüència de parada.
	I2C_Enviar_Byte	Transmet el byte que es troba a l'acumulador per I <sup>2</sup> C. La rutina finalitza quan es rep el bit de reconeixement (ACK).
	I2C_Llegir_Byte	Llegeix el byte procedent del dispositiu I <sup>2</sup> C i el retorna a l'acumulador.
	I2C_nACK	Envia el bit de reconeixement (ACK=0)
	I2C_ACK	Envia el bit de reconeixement (ACK=1)

Taula 6. Rutines de cada llibreria.

### 5.3. Assaig 1: lectura de l'acceleració lineal

En aquest primer assaig s'ha experimentat amb el sensor d'acceleració lineal BMA180. La figura 16 mostra el muntatge de l'aplicació:



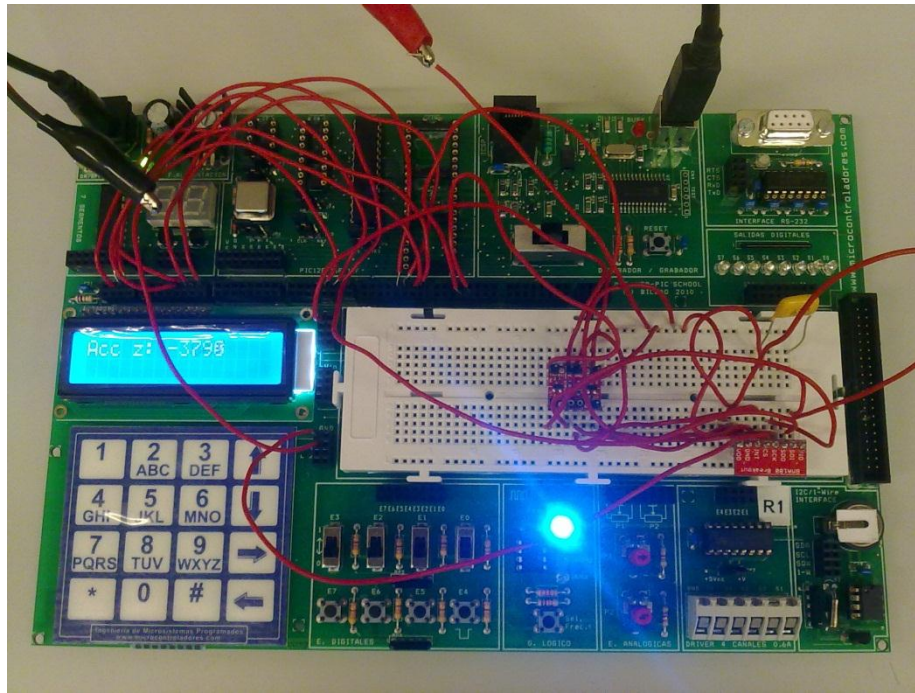


Figura 16. Muntatge del primer assaig (acceleròmetre BMA180)

En relació al material utilitzat, tenim la plataforma USB-PIC'School (amb la pantalla LCD integrada, la placa de prototipatge, etc.), el PIC 16F886, el BMA180, el convertidor de nivell lògic i un condensador de 22nF. A part, s'ha fet ús d'una font d'alimentació de 3.3V.

El microcontrolador es comunica per I<sup>2</sup>C amb l'acceleròmetre. S'han configurat els paràmetres d'operació del sensor per a la posterior adquisició de les dades d'acceleració. Com que únicament es vol comprovar el funcionament d'aquest, s'ha decidit llegir únicament la dada d'acceleració vertical (l'eix Z), es a dir, el valor de la gravetat. La dada obtinguda és el valor del convertidor analògic/digital del propi sensor en format binari. Per poder interpretar-la es converteix en format decimal. Després es mostra el valor de la gravetat a la pantalla LCD per mitjà d'un bus de 4 bits. Tot aquests passos formen un cycle infinit, és a dir, una lectura contínua de l'acceleració.

Per tenir el resultat expressat en  $m/s^2$  són necessàries operacions de multiplicació i divisió, però fer-ho amb assemblador és una tasca llarga i complicada, i no és aquesta la finalitat de l'assaig. Així que sabent que una acceleració de  $\pm 1g$  equival a  $\pm 8192$  LSB (dada obtinguda directament del sensor i convertida a format decimal), la dada mostrada a la pantalla LCD ( $-1g$ ) hauria de ser  $-4096$ . Podem comprovar que a la imatge s'hi veu un valor de  $-3790$  (molt

aproximat a l'esperat). La diferència d'error és deguda a la inestabilitat que ofereix aquest sensor i que la placa no es troba totalment vertical.

Per aquesta experiència, s'ha necessitat el programa font BMA180.asm i les llibreries BIN\_BCD.inc, I2C.inc, LCD\_4bits.inc i P16F886.inc. Degut a les dimensions del codi dels arxius, s'ha decidit adjuntar-ho a part. Tampoc s'ha vist la necessitat de comentar-lo en aquest document ja que ho esta fet junt amb el codi.

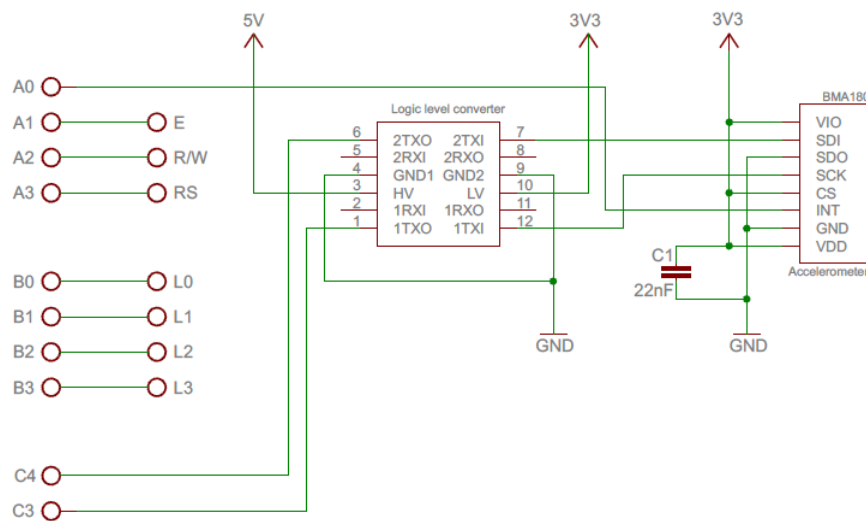


Figura 17. Esquema del primer assaig (acceleròmetre BMA180)

Segons el circuit de la figura 17 es pot apreciar que l'acceleròmetre està connectat a 3.3V i GND a través dels pins que ho necessiten (informació extreta del propi full de característiques), i també a un condensador de 22nF. Cal remarcar que "Sparkfun" (l'empresa subministradora dels sensors juntament amb els components externs de cada un en una placa de prototipatge) entrega aquest sensor amb un condensador integrat a la placa de 0.22nF. Aquest valor és erroni, ja que el full de característiques especifica un condensador de 22nF. Aquest petit detall cal tenir-lo en compte ja que es va dedicar molt de temps en buscar la causa de les lectures tan inestables que proporcionava el sensor BMA180 fins que no es va trobar l'error d'aquest condensador. La connexió d'aquest nou component en paral·lel va solucionar el problema. Seguidament tenim el convertidor de nivell lògic que permet la connexió I<sup>2</sup>C entre el sensor i el microcontrolador, les senyals del qual són SCK (SCL, dades) i SDI (SDA, rellotge). Després s'observa que aquestes dues senyals es connecten a la plataforma USB-PIC'School als pins C4 (SCL) i C3 (SDA), a més, tenim la senyal d'interrupció de l'acceleròmetre al pin A0 (ens informarà de la disponibilitat

de dades noves). I finalment la connexió entre el PIC i l'LCD. Com que tots dispositius estan integrats a la plataforma, les connexions són directes: A1, A2 i A3 per la interfície de control i B0, B1, B2 i B3 per la interfície de dades. Els pins A0-A7 corresponen al port A del PIC, els pins B0-B7 al port B i els pins C0-C7 al port C. Aquestes condicions són vàlides en tots els assajos.

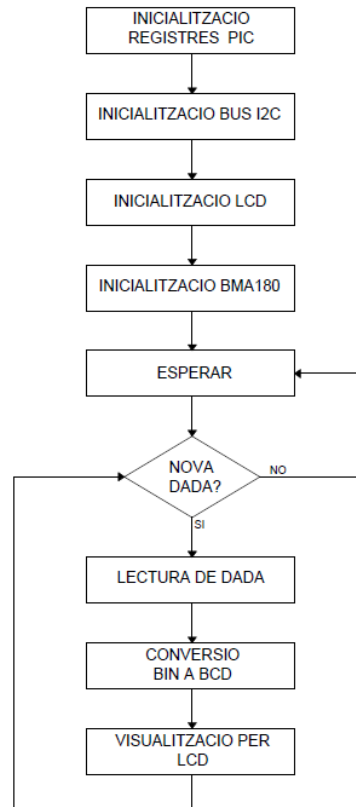


Figura 18. Diagrama de flux del primer assaig (acceleròmetre BMA180)

El procés comença amb una inicialització dels registres del PIC, tals com els ports digitals i analògics. També es configura el mòdul MSSP del microcontrolador en mode “Master” i freqüència de treball de 100kHz per la comunicació sèrie I<sup>2</sup>C. Seguidament es configura la comunicació amb la pantalla LCD i alhora s’executa la rutina d’inicialització que especifica el fabricant. Pel sensor BMA180 no és necessària cap rutina però sí que s’han de configurar alguns registres interns per la selecció dels diferents modes d’operació: habilitem l’escriptura de registres, inhabilitem interrupcions (necessari per la configuració de l’ample de banda), seleccionem ample de banda de 10Hz (lectures menys freqüents però més estables), activem interrupció per disponibilitat de nova dada i habilitem el bloqueig del MSB de la dada d’acceleració durant la lectura del LSB.

A partir d'aquí ja podem començar a llegir dades, de manera que quan l'acceleròmetre ens avisa que ja hi ha nova dada, la rebem i la mostrem per pantalla (un procés continu). Així, al provar de moure la plataforma en diferents direccions s'ha observat que el valor de l'acceleració variava, ja que experimentava un moviment.

#### 5.4. Assaig 2: lectura de l'orientació

En aquest segon assaig s'ha experimentat amb la brúixola electrònica HMC6352.

En relació al material utilitzat, tenim la plataforma USB-PIC'School (amb la pantalla LCD integrada, la placa de prototipatge, etc.), el PIC 16F886, l'HMC6352 i el convertidor de nivell lògic. A part, s'ha fet ús d'una font d'alimentació de 3.3V per alimentar la brúixola.

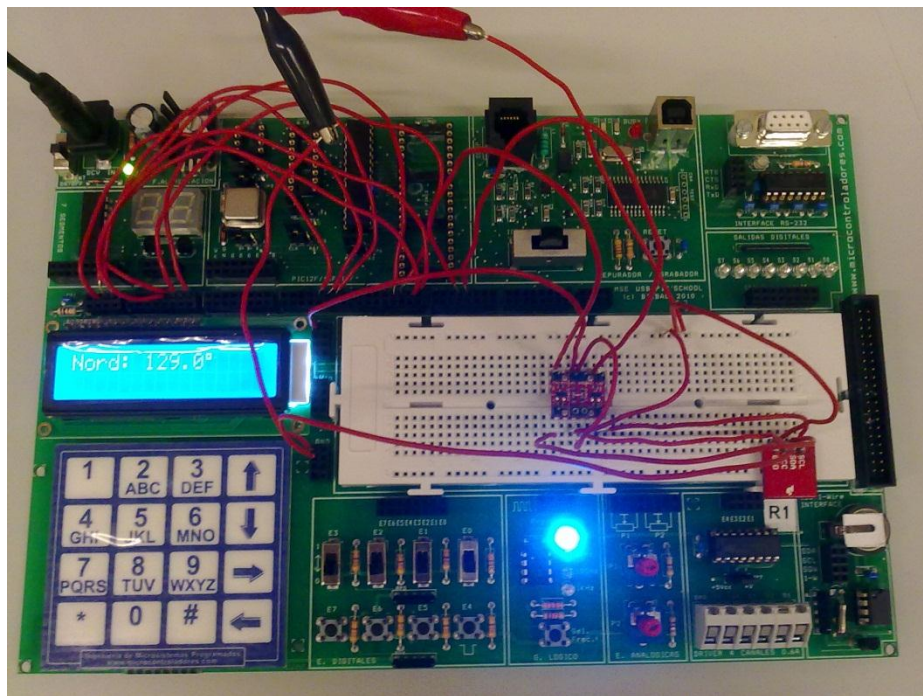


Figura 19. Muntatge del segon assaig (brúixola IXZ500)

El microcontrolador es comunica per I<sup>2</sup>C amb la brúixola. S'han configurat els paràmetres d'operació del sensor per a la posterior adquisició de les dades, en aquest cas la lectura de l'angle respecte el nord. Anteriorment s'ha comentat que aquest sensor és molt sensible als camps magnètics i per tal de solucionar el problema incorpora un mode de calibratge en el qual l'usuari ha de rotar el sensor unes dues voltes durant 20 segons. Així, el sensor recull les dades dels camps magnètics del voltant per tal de compensar els offsets interns i donar

una dada vàlida. El sistema de calibratge es seleccionarà per mitjà d'un interruptor. La dada obtinguda és en format binari, que posteriorment es converteix en format decimal per representar-lo a la pantalla LCD i l'usuari pugui interpretar-lo fàcilment. Tot aquests passos formen un cicle infinit, és a dir, una lectura contínua de l'angle.

Per tenir el resultat expressat en graus s'ha de dividir el valor entre 10, de manera que el resultat final té una precisió de dècimes de grau. Per evitar implementar una rutina que divideixi entre 10 en ensamblador, s'imprimeix per l'LCD un punt (per representar les dècimes) entre les dues últimes xifres. Podem comprovar que a la imatge s'hi distingeix un valor de 129.0° respecte el nord.

Per aquesta experiència, s'ha necessitat el programa font HMC6352.asm i les llibreries BIN\_BCD.inc, I2C.inc, LCD\_4bits.inc i P16F886.inc. Degut a les dimensions del codi dels arxius, s'ha decidit adjuntar-ho a part. Tampoc s'ha vist la necessitat de comentar-lo en aquest document ja que ho esta fet junt amb el codi. La següent figura 20 mostra l'esquema del circuit:

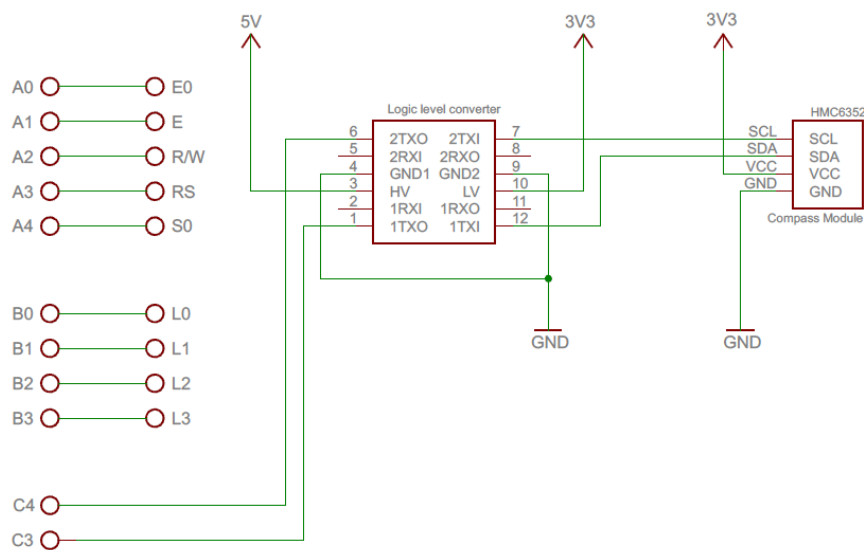


Figura 20. Esquema del segon assaig (brúixola HMC6352)

Segons el circuit de la figura 20 es pot apreciar que la brúixola està alimentada a 3.3V. Seguidament tenim el convertidor de nivell lògic que permet la connexió I<sup>2</sup>C entre el sensor i el microcontrolador, les senyals del qual són SCL (dades) i SDA (rellotge). Després s'observa que aquestes dues senyals es connecten a la plataforma USB-PIC'School als pins

C4 (SCL) i C3 (SDA), a més, tenim l'interruptor E0 connectat al pin A0 per activar el mode de calibratge. L'indicador LED del pin S0 connectat al pin A4 s'activarà per informar-nos que estem en el mode de calibratge. I finalment la connexió entre el PIC i l'LCD. Com que tots dispositius estan integrats a la plataforma, les connexions són directes: A1, A2 i A3 per la interfície de control i B0, B1, B2 i B3 per la interfície de dades. Els pins A0-A7 corresponen al port A del PIC, els pins B0-B7 al port B i els pins C0-C7 al port C.

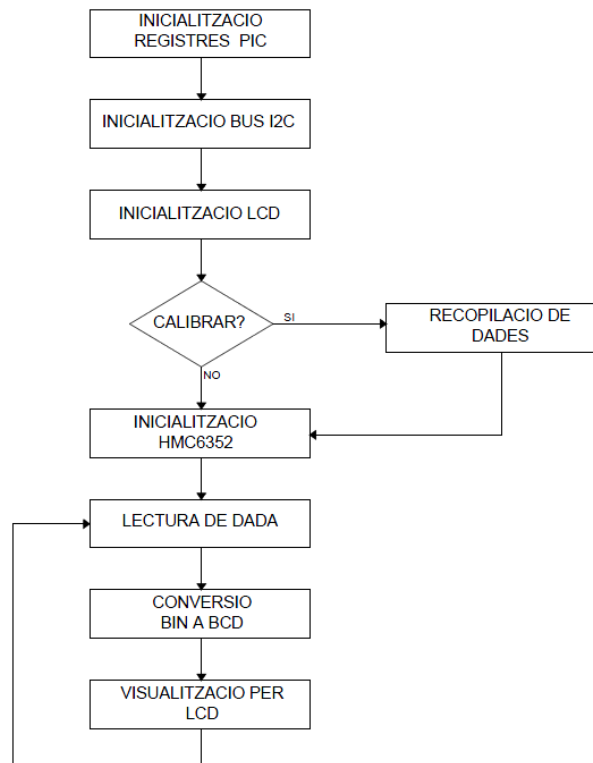


Figura 21. Diagrama de flux del segon assaig (brúixola HMC6352)

El procés comença amb una inicialització dels registres del PIC, tals com els ports digitals i analògics. També es configura el mòdul MSSP del microcontrolador en mode "Master" i freqüència de treball de 100kHz per la comunicació sèrie I<sup>2</sup>C. Seguidament es configura la comunicació amb la pantalla LCD i alhora s'executa la rutina d'inicialització que especifica el fabricant. Si sabem que l'entorn està pertorbat per camps magnètics, activem l'interruptor (abans d'activar el funcionament de l'aplicació) i iniciem la rutina de calibratge. Un cop acabada, col·loquem l'interruptor a la posició inicial. Pel sensor HMC6352 no és necessària cap rutina però sí que s'ha de configurar el registre intern per la selecció del mode d'operació: "Standby mode".



A partir d'aquí ja podem començar a llegir dades, de manera que anem rebent les dades i les mostrem per pantalla. Així, al provar de moure la plataforma en diferents direccions s'ha observat que el valor de l'angle ha experimentat una variació real de la direcció del nord. També s'ha comprovat que el valor de l'angle representat per l'LCD ha variat considerablement en efectuar un moviment lineal, conseqüència deguda a la inestabilitat respecte els camps magnètics, com per exemple els creats pels ordinadors del laboratori.

### 5.5. Assaig 3: lectura de la velocitat angular

En aquest segon assaig s'ha experimentat amb el sensor de velocitat angular IXZ500. En relació al material utilitzat, tenim la plataforma USB-PIC'School (amb la pantalla LCD integrada, la placa de prototipatge, etc.), el PIC 16F886, l'IXZ500 i dues resistències de 2k2 ohms i 1k ohm. A part, s'ha fet ús d'una font d'alimentació de 3.3V per alimentar el sensor.

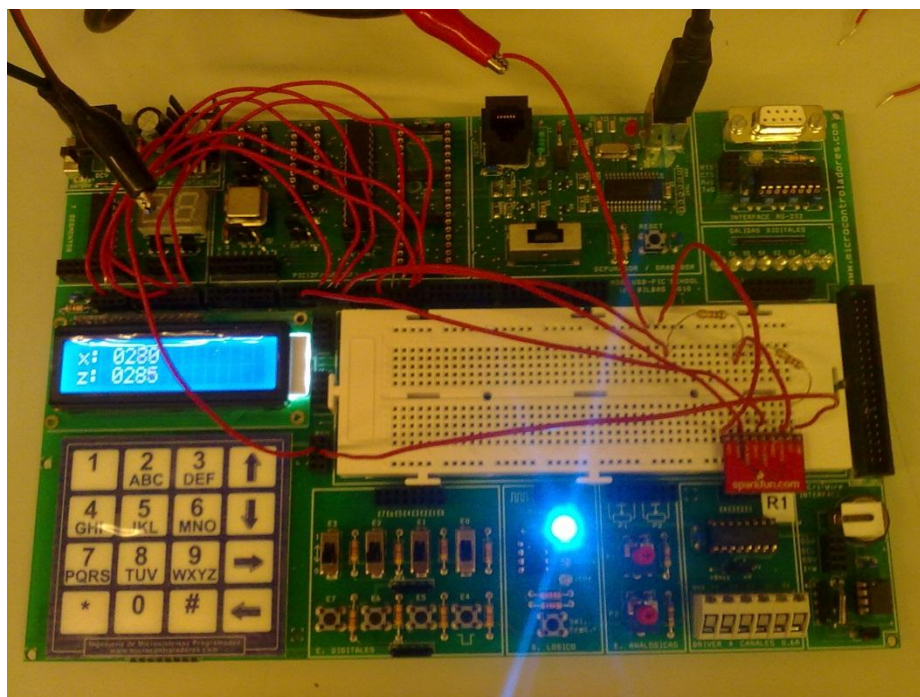


Figura 22. Muntatge del tercer assaig (giroscòpic IXZ500)

El microcontrolador es comunica per mitjà d'entrades analògiques amb el giroscòpic. S'han configurat els paràmetres d'operació del sensor per a la posterior adquisició de les dades, en aquest cas la lectura de la rotació vertical (eix Z) i la rotació horitzontal (eix X). Aquest sensor necessita un petit calibratge inicial per mitjà de l'aplicació d'un pols de 3.3V durant uns 200us al pin AZ (AutoZero). D'aquesta manera fixem com a velocitat rotativa de 0°/s a

1.35V, així l'entrada analògica podrà llegir velocitats en els dos sentits (negatives i positives). La dada obtinguda és en format binari, i posteriorment es converteix en format decimal. El resultat que s'obté és el valor del convertidor A/D del microcontrolador on el rang de 108 a 729 es tradueix a  $\pm 110$ . Com en l'assaig de l'acceleròmetre, no ens interessa invertir temps en desenvolupar una rutina d'operacions aritmètiques en ensamblador. Després es representa a la pantalla LCD, on l'usuari pot interpretar-lo fàcilment. Tot aquests passos formen un cicle infinit, és a dir, una lectura contínua de la velocitat angular dels dos eixos. Podem comprovar que a la imatge s'hi distingeix un valor de 280 i 285, que equivaldria a uns 0°/s.

Per aquesta experiència, s'ha necessitat el programa font IXZ500.asm i les llibreries BIN\_BCD.inc, LCD\_4bits.inc i P16F886.inc. Degut a les dimensions del codi dels arxius, s'ha decidit adjuntar-ho a part. Tampoc s'ha vist la necessitat de comentar-lo en aquest document ja que ho esta fet junt amb el codi.

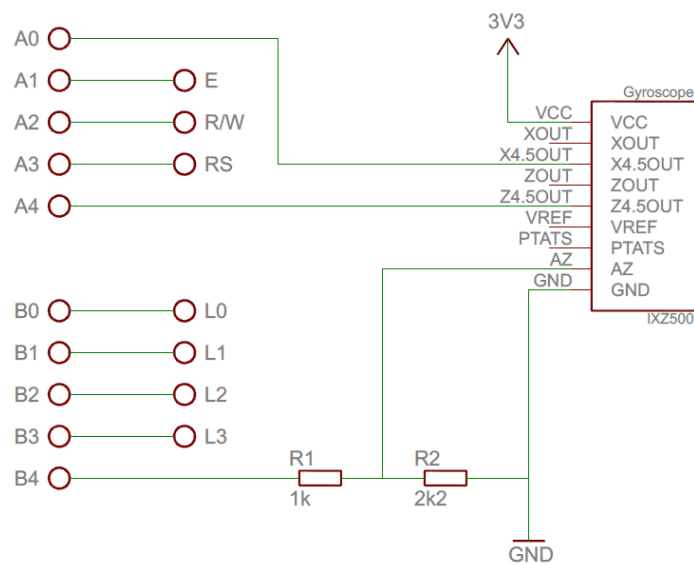


Figura 23. Esquema del tercer assaig (giroscòpic IZX500)

Segons el circuit de la figura 23 es pot apreciar que el giroscòpic està alimentat a 3.3V. Seguidament, tenim les sortides X4.5OUT i Z4.5OUT connectades als pins A0 (entrada analògica AN0) i A4 (entrada analògica AN4) respectivament. Al pin B4 hi va connectada la funció AutoZero pel calibratge del sensor per mitjà de dues resistències que converteixen els 5V del PIC a 3.3V. I finalment la connexió entre el PIC i l'LCD. Com que tots dispositius estan integrats a la plataforma, les connexions són directes: A1, A2 i A3 per la interfície de



control i B0, B1, B2 i B3 per la interfície de dades. Els pins A0-A7 corresponen al port A del PIC, els pins B0-B7 al port B i els pins C0-C7 al port C.

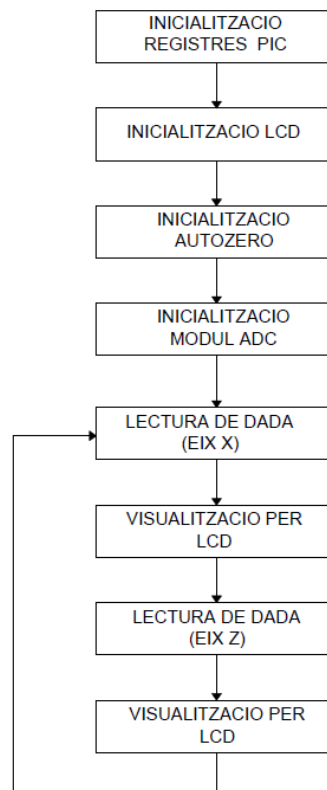


Figura 24. Diagrama de flux del tercer assaig (giroscòpic IXZ500)

El procés comença amb una inicialització dels registres del PIC, tals com els ports digitals i analògics. També es configura la comunicació amb la pantalla LCD i alhora s'executa la rutina d'inicialització que especifica el fabricant. Seguidament s'activa el pols d'AutoZero per fixar els 0°/s a 1.35V i posteriorment s'inicialitza el mòdul convertidor A/D. Al ser un sensor analògic, no és tan complex com els digitals que necessiten, alguns, rutines d'inicialització i configuració de registres. Només consistirà en la lectura analògica dels dos pins de dades.

A partir d'aquí ja podem començar a llegir dades i les mostrant-les per pantalla. Així, al provar de moure la plataforma en diferents direccions s'ha observat que el valor de la velocitat angular canvia, ja que experimenta un gir.

## 6. DISSENY I IMPLEMENTACIÓ DEL CIRCUIT IMPRÈS

### 6.1. Funcionament

És el moment de combinar tots els coneixements adquirits utilitzant l'entorn USB PIC SCHOOL per tal de aplicar-los i desenvolupar un prototip PCB per la corresponent implementació final. Altrament, es pretén representar la informació en un PC.

Es vol utilitzar els tres sensors descrits fins ara (el giroscòpic, l'acceleròmetre i la brúixola) per adquirir les dades del moviment del robot. Sabem que per a determinar la trajectòria necessitem informació sobre els moviments de translació i de rotació. Aquestes dades s'enviaran al microcontrolador, encarregat de processar la informació i enviar-la a l'ordinador. Alhora s'emmagatzemaran les dades en un document .txt mitjançant un programa desenvolupat amb l'entorn C Sharp. Per accedir a les dades d'aquest document i traduir-les en moviments, posicions i trajectòries, s'emprarà l'entorn MATLAB capaç de dur a terme una representació visual en 3D. Les dades seran contrastades amb les obtingudes amb el sistema odomètric, desenvolupat anteriorment per un altre projectista. Amb els resultats s'analitzaran l'efectivitat dels dos sistemes de navegació amb els avantatges i inconvenients de cada un. Els diferents programes desenvolupats s'adjuntaran al final d'aquest document.

Aquest apartat es dividirà en diferents subapartats per descriure el funcionament i la relació que tenen cadascun dels dispositius electrònics utilitzats. Hi haurà una part on s'hi descriurà l'alimentació del circuit, en una altra l'adquisició de dades, i en l'altre la transmissió de dades. També constarà d'un últim subapartat relacionat amb el disseny del circuit i la PCB.

### 6.2. Alimentació

La placa de navegació inercial necessita dues alimentacions: 5V per alimentar el PIC i l'integrat MAX232, i 3.3V per alimentar els sensors. Aquestes dues tensions s'extreuen de la font d'alimentació de 12V que ja està integrada a la plataforma PRIM. La figura 25 mostra els esquemes de les fonts d'alimentació amb els dispositius utilitzats.

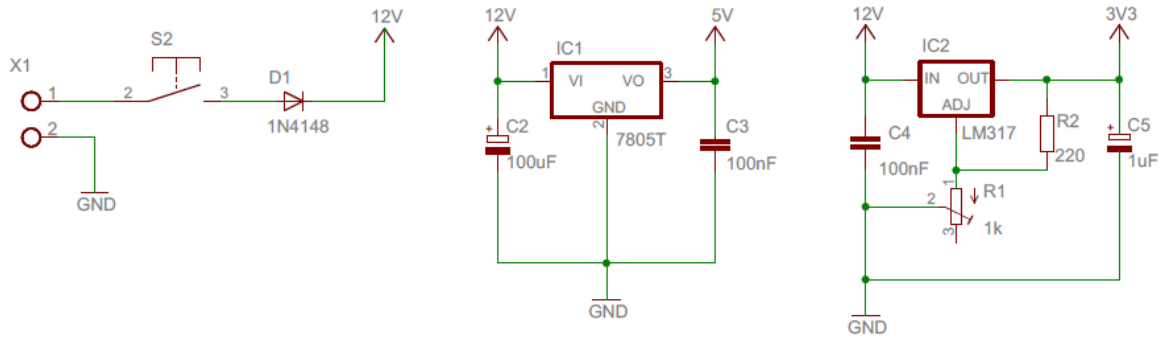


Figura 25. Esquemes electrònics de les fonts d'alimentació

A l'esquerra hi distingim el connector X1 on s'hi col·locaran els dos borns de la font de 12V del robot mòbil del laboratori. Per mitjà de l'interruptor S2 activarem o desactivarem la funcionalitat de la placa, i el díode 1N4148 ens servirà per evitar el pas de corrent inverses.

L'integrat LM7805 disposa dels condensadors que especifica al full de característiques i permet subministrar 5V estables i amb el mínim de soroll a partir dels 12V.

L'integrat LM317, a part de la connexió dels dos condensadors especificats pel fabricant, necessita dues resistències amb els valors dels quals es pot obtenir una tensió de sortida de 1.2V a 37V. A partir de l'equació 5 i suposant que  $I(R2)$  és pràcticament nul·la podem obtenir el valor de la resistència R1 fixant R2 a 220 ohms.

$$V_{out} = 1.25 \cdot (1 + (R1 / R2)) + I(R2) \quad (\text{Eq. 5})$$

$$R1 = ((V_{out} / 1.25) - 1) \cdot R2 = ((3.3 / 1.25) - 1) \cdot 220 = 360.8 \text{ ohms} \quad (\text{Eq. 6})$$

Per tal d'eliminar els errors inevitables que comporta transportar el model teòric a la pràctica com poden ser les toleràncies de les resistències, s'ha utilitzat un potenciòmetre i així adaptar el valor de la resistència per generar els 3.3V.

### 6.3. Adquisició de dades

Arribat a aquest punt, ja es poden alimentar els sensors i el Microcontrolador i fer les connexions necessàries entre ells per dur a terme una bona comunicació.

Tot seguit es pot observar l'esquema següent, format pel microcontrolador PIC, el giroscòpic IXZ500, l'acceleròmetre BMA180 i la brúixola HMC6352. També s'hi localitza el dispositiu integrat MAX232 que es definirà el seu funcionament en el següent subapartat.

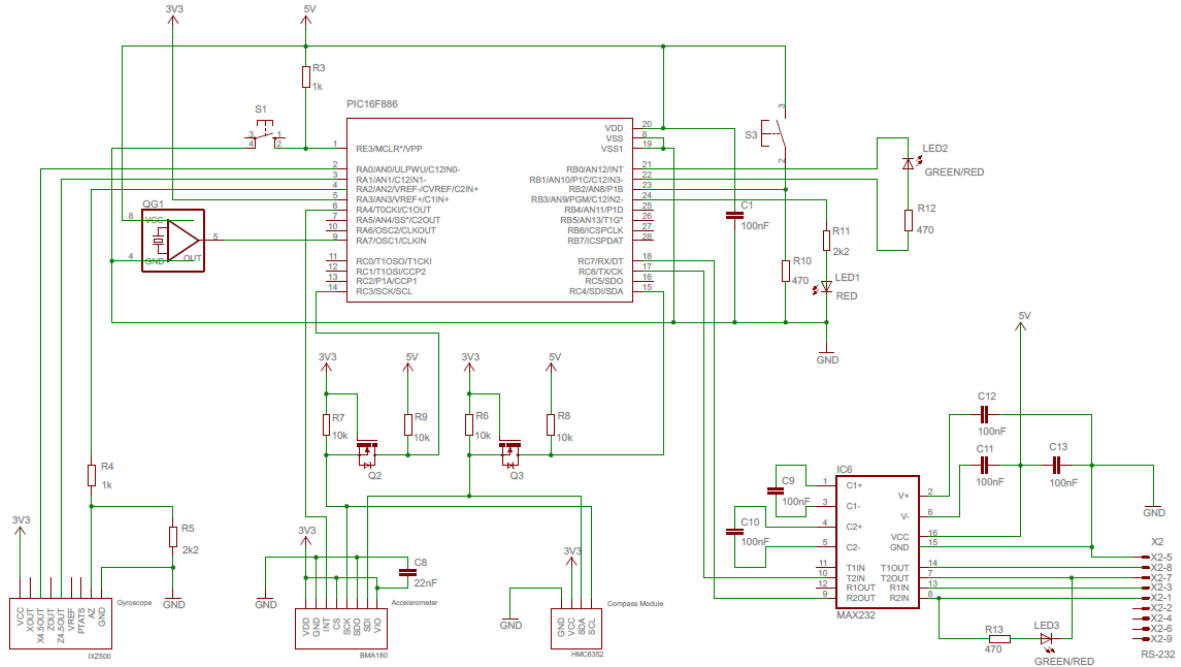


Figura 26. Esquema general de la placa final

La figura superior permet fer-nos una idea general de les diferents parts del circuit i dels diferents components electrònics que s'utilitzen. Per entrar en detall al funcionament de cada part, es subdividirà en diferents figures.

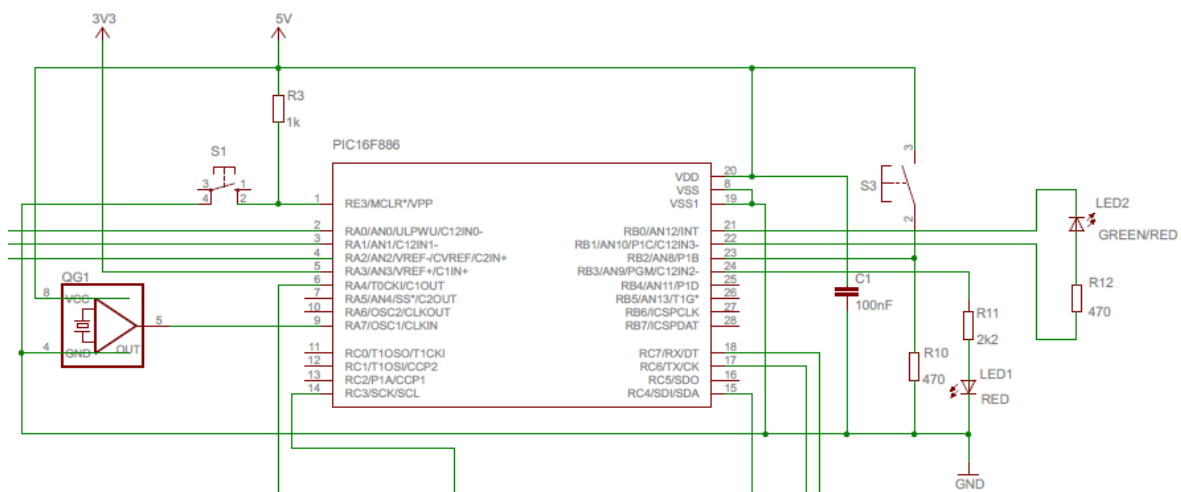


Figura 27. Esquema del microcontrolador PIC 16F886

A la figura 27, observem que al pin 1 hi ha connectat el circuit de reset, de manera que es manté en estat alt (5V) mentre no es prem el pulsador S1. Als pins 2 i 3 (entrades analògiques) s'hi col·loquen els pins X4.5OUT i Z4.5OUT del giroscòpic, respectivament. El 4 actua com a sortida digital per efectuar un pols d'AutoZero al mateix sensor. Seguidament ens trobem el número 5, el qual està configurat per connectar-hi la tensió de referència pel mòdul convertidor A/D, en aquest cas 3.3V. Això és degut a que permet aproximar i augmentar la precisió del convertidor. El següent pin, RA4, es configura com a entrada digital per detectar la interrupció enviada per l'acceleròmetre, indicant que es pot efectuar una nova lectura. El circuit d'oscil·lació es porta a terme per mitjà d'un oscil·lador alimentat a 5V, que dóna la sortida al pin 9 del PIC amb una freqüència de 4MHz. Els pins SCL i SCA són utilitzats per la comunicació I<sup>2</sup>C amb l'acceleròmetre i la brúixola. L'altre protocol, per la comunicació RS-232 amb l'ordinador, fa ús dels pins RX i TX del port C del microcontrolador. Finalment ens trobem amb l'interruptor S3 connectat a RB2 i configurat per iniciar el mode de calibratge de la brúixola. En la posició inicial es troba inactiu, quan el desplaçem comença el calibratge de la brúixola durant uns 20 segons fins que el tornem a desplaçar a la posició inicial altre cop, on surt d'aquest mode i continua executant el programa principal. Els leds 1 i 2 ens indiquen l'estat en què es troba el funcionament de la placa. El primer led, de color vermell, està programat per fer intermitència mentre s'estigui en el mode de calibratge. El segon led es pot il·luminar de color verd (indicant el correcte funcionament el circuit) o de color vermell (indicant que s'ha produït un error i que s'ha d'efectuar el conseqüent reset).

La figura 28 ens mostra amb detall l'esquema del giroscòpic. S'alimenta el sensor a 3.3V, es connecten els pins X4.5OUT i Z4.5OUT al PIC i s'adapta el pols d'AutoZero que envia el microcontrolador a 3.3V per mitjà d'un simple divisor de tensió format per dues resistències.

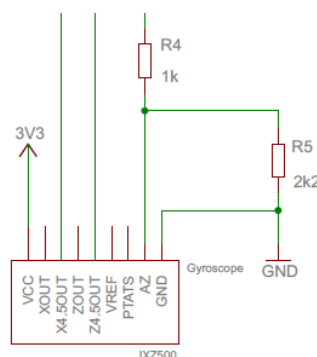


Figura 28. Esquema del giroscòpic IXZ500

L'acceleròmetre i la brúixola necessiten un circuit més complex ja que es comuniquen amb el microcontrolador a través del bus I<sup>2</sup>C. Però no ho fan directament, sinó per mitjà d'un parell de transistors d'efecte de camp (MOSFETs) encarregats d'adaptar el nivell lògic de tensions (5V pel PIC i 3.3V pels sensors).

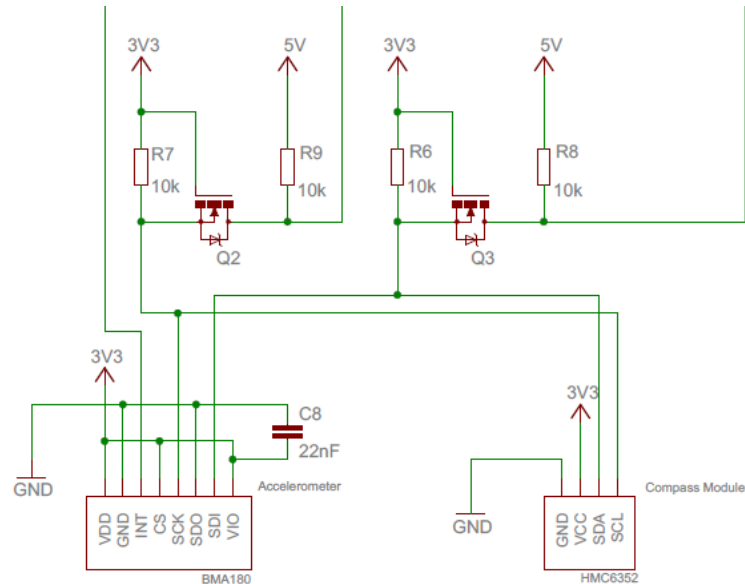


Figura 29. Esquema de l'acceleròmetre BMA180 i la brúixola HMC6352 mitjançant dos MOSFETs

L'acceleròmetre s'alimenta a 3.3V pels pins VDD, CS i VIO, i a terra pels pins GND i SDO. Segons el full de característiques del BMA180, si SDO es connecta a 3.3V l'adreça és 41h més el bit de lectura/escriptura, en canvi, si es connecta a 0V l'adreça és 40h més el bit de lectura/escriptura. El mateix passa amb el pin CS, si es connecta a 3.3V s'habilita la interfície per I<sup>2</sup>C, en canvi, si es connecta a 0V la comunicació s'ha de dur a terme per SPI. Els pins SCK (rellotge) i SDI (dades) s'utilitzen per la comunicació amb el PIC. El condensador C8 és un component imprescindible, ja que sense ell les lectures són molt inestables. Referent a la brúixola, consta de l'alimentació a 3.3V i la comunicació sèrie síncrona. L'avantatge que ofereix aquest protocol de comunicació és que els mateixos pins de rellotge i de dades poden establir comunicació en diferent perifèrics. Així que els pins SCK i SDI van connectats amb els pins SCL i SDA respectivament. Com que els sensors treballen amb una lògica de 3.3V i el microcontrolador a 5V, s'utilitzen els MOSFETS treballant en les zones de tall i saturació.

## 6.4. Transmissió de dades

Un cop s'han obtingut totes les dades, cal enviar-les a l'ordinador seguint un ordre lògic i coherent pel posterior emmagatzematge en un document .txt. Aquesta funció es porta a terme per mitjà de l'integrat MAX232, que adapta la lògica TTL que utilitza el PIC a nivells de tensió RS-232 de  $\pm 7.5V$ . A la següent figura s'observa l'esquema de les connexions entre el MAX232 i el PIC 16F886.

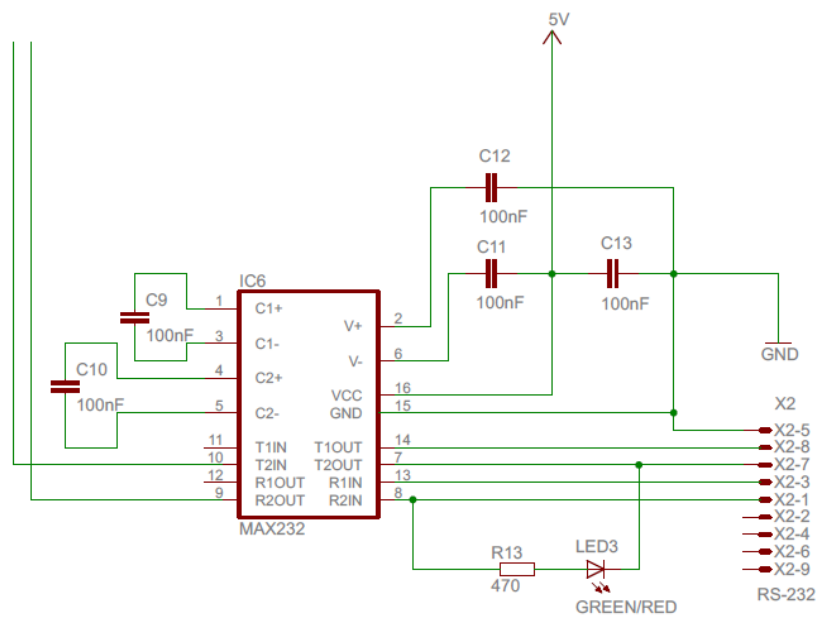


Figura 30. Esquema del dispositiu integrat MAX232

El fabricant aconsella col·locar els condensadors C9-C13 pel correcte funcionament de l'integrat. Per una banda, els pin 9 i 10 s'encarreguen de la recepció i transmissió de les dades a nivells lògics TTL. Per altra banda, els pins 7 i 8 tenen la funció de rebre i transmetre la informació a nivells estàndards RS-232, aquests es connecten a un connector femella RS-232. El LED3 té doble funcionalitat, es a dir, es pot encendre de color verd o vermell depenent de si es transmeten o es reben dades. Com que la velocitat de transmissió és molt ràpida en comparació a la visualització de l'ull humà, l'únic que es podrà veure és com el LED3 va canviant de color simultàniament.

## 6.5. Funcionament del programa

Per adquirir totes les dades dels sensors es necessita un temps determinat, igual que per enviar-les. El diagrama de flux següent permet entendre de manera més fàcil i més visual el funcionament del programa.

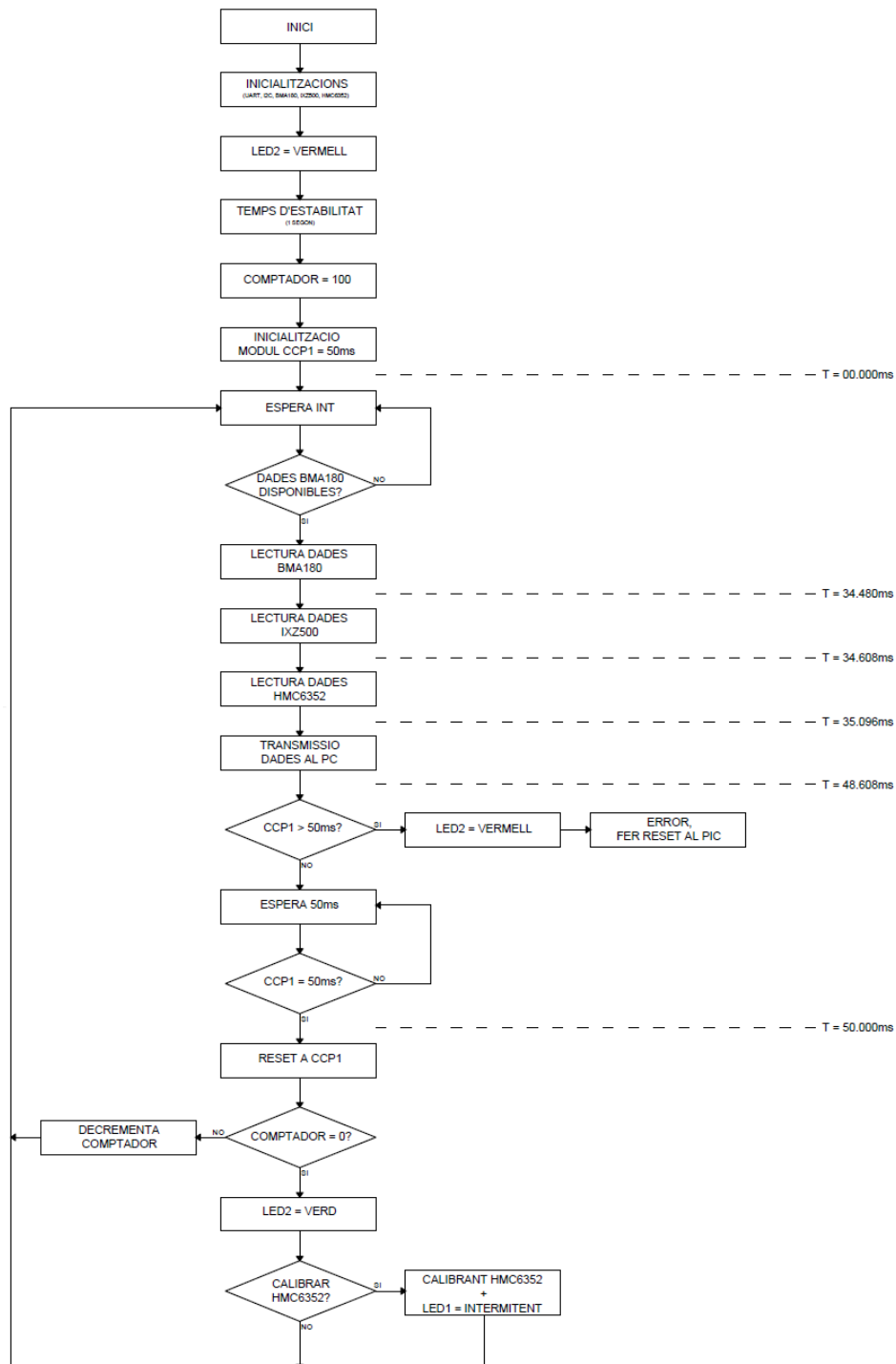


Figura 31. Diagrama de flux del funcionament del programa de la placa de navegació inercial



Es comença inicialitzem el mòdul UART del PIC 16F886 a 19200 bits/s en mode asíncron, inicialitzant el mòdul MSSP del PIC pel protocol I<sup>2</sup>C a 100kHz, inicialitzant el sensor BMA180 amb lectura de dades per interrupció (entre d'altres), inicialitzant el sensor IXZ500 (activació de la funció AutoZero i configuració del mòdul ADC del PIC) i inicialitzant el sensor HMC6352 en mode continu. Seguidament activem el LED2 de color vermell per indicar que el procés està en marxa però no preparat per enviar dades vàlides de moviment, ja que abans s'ha d'esperar un temps d'estabilització dels sensors (a la pràctica es va observar que els sensors donaven errors imprecisos els primers milisegons) i un temps de calibratge a través del MATLAB. Aquest calibratge consisteix (amb el robot parat) en enviar les dades dels 5 primers segons a l'ordinador perquè amb el MATLAB es pugui obtenir la posició inicial del robot. Passats aquests 5 segons, el LED2 s'il·luminarà de verd per indicar que el robot ja pot començar a moure's.

El primer sensor a ser llegit és l'acceleròmetre, quan aquest envia la senyal d'interrupció s'envien cap al microcontrolador les acceleracions dels 3 eixos de coordenades. Després es llegeix el giroscòpic i seguidament la brúixola. S'ha establert aquest ordre de manera que el sensor que necessita un temps més llarg per tenir una dada vàlida es col·loqui a la posició inicial i la diferència de temps de lectura entre els 3 sensors sigui el mínim (ja que es considera que totes les dades són adquirides cada 50ms exactes). Després es transmet la informació a l'ordinador per RS-232 i es comprova que totes les operacions efectuades, des de la lectura del BMA180 fins aquí, no hagin tardat més de 50ms. Com que anteriorment s'ha calculat que aquestes operacions tardaven poc més de 48ms no hi hauria d'haver cap error, però si passés, el LED2 quedaria permanentment en vermell indicant que algun sensor ha deixat de funcionar correctament i per tant les mesures ja no són vàlides. O simplement indicaria algun error de soroll, de manera que fent un reset al microcontrolador el problema ja quedaria solucionat. Si s'ha comprovat que no s'ha accedit de 50ms, significa que les dades són vàlides i el procés continua executant-se. Un cop arribats exactament a 50ms, es resseteja el mòdul CCP1 per reiniciar el Temporitzador de 50ms i començar de nou el procés d'adquisició de dades i transmissió. Com es pot comprovar, el control del temps és essencial pel correcte funcionament de la placa de navegació inercial.

Finalment, si es desitja calibrar la brúixola, es pot fer actuant sobre l'interruptor S3 i donant unes 2 voltes al robot durant uns 20 segons. Durant aquest procés el LED1 estarà en intermitència. Passats els 20 segons, es desplaça l'interruptor a la posició inicial. Aquest procés es pot efectuar en qualsevol moment, amb la condició que posteriorment s'ha de fer un reset a la placa i començar el procés de lectura de nou.

En un principi s'havia decidit enviar les dades a l'ordinador cada 100ms, igual que el sistema odomètric. Però abans de res, cal fer un petit estudi dels temps d'adquisició de les dades i de transmissió. Segons els fulls de característiques de l'acceleròmetre BMA180 i l'equació 7, amb un ample de banda de 10Hz necessita 50.04ms per adquirir les dades vàlides d'acceleració.

$$T(\text{BMA180}) = 0.417 \cdot (1200 / \text{ample de banda}) = 50.04\text{ms} \quad (\text{Eq. 7})$$

La brúixola HMC6352 necessita 6ms per calcular el nou angle i deixar-lo disponible per transmetre. Però com que la configurem en mode continu, eliminem aquest temps, ja que executa càlculs continus i sempre hi ha disponible la última dada calculada. Pel que fa al giroscòpic IXZ500, al ser analògic, el temps d'adquisició vindrà determinat pel temps que tarda el convertidor A/D del PIC 16F886 a transformar la dada, format per temps d'adquisició més el temps de conversió. L'equació 8 mostra el resultat.

$$T(\text{ADC}) = T_{\text{acq}} + T_{\text{cnv}} = (0.04 \cdot 11) + 0.0115 = 0.45\text{ms} \quad (\text{Eq. 8})$$

També hem de tenir en compte els temps de transmissió del bus I<sup>2</sup>C configurat a una freqüència de treball de 100kHz. Es compona d'una condició d'Start (0.0087ms), una condició d'Stop (0.0087ms) i el nombre de bits a transmetre (0.0087ms cada un). L'equació 9 reflecteix un temps aproximat de la transmissió de 2 bytes (seleccionem 2 bytes perquè cada dada enviada pels sensors està formada per 2 bytes, de manera que simplifica els càlculs finals).

$$T(\text{I}^2\text{C}) = 0.0087 + (0.0087 \cdot (8 \cdot 2)) + 0.0087 = 0.16\text{ms} \quad (\text{Eq. 9})$$

La transmissió per UART la tenim configurada a 19200 bits/s i amb un bit de parada. L'equació següent mostra el temps necessari per transmetre 2 bytes.

$$T(\text{UART}) = (2 \cdot (8 + 1)) / 19200 = 0.94\text{ms} \quad (\text{Eq. 10})$$

Amb les equacions 7, 8, 9 i 10 podem fer una estimació del temps necessari per adquirir totes les dades i transmetre-les a l'ordinador. En les equacions 11 i 12 es mostra el temps total del procés.

$$T = T(\text{BMA180}) + T(\text{IXZ500}) + T(\text{HMC6352}) + T(\text{RS-232}) \quad (\text{Eq. 11})$$

$$T = 50.52 + 0.9 + 0.16 + 5.64 = 57.22 \quad (\text{Eq. 12})$$

Fins ara s'han portat a terme els càlculs teòrics i s'observa que la transmissió s'hauria d'efectuar cada 100ms. En la pràctica els temps varien lleugerament. El temps del BMA180 es tradueix a 34.48ms, el de l'IXZ500 passa a valer 0.128ms, el de l'HMC6352 passa a ser 0.16ms i el de transmissió a l'ordinador es converteix en 13.512 ms. Fent un total de 48.608ms. Així que s'ha decidit transmetre les dades cada 50ms, d'aquesta forma disposem de més dades i conseqüentment permet acostar-nos més al moviment real del robot.

Pel que fa a la part informàtica, el projecte consta de l'arxiu Navegacio\_inercial.asm com a programa principal, i les següents llibreries: BMA180.inc, IXZ500.inc, HMC6352.inc, I2C.inc, UART.inc i P16F886. Tots aquest arxius es troben adjuntats al final d'aquest document suficientment comentats.

## 6.6. Disseny del circuit imprès

El circuit imprès s'ha dissenyat amb l'entorn CadSoft EAGLE (Easily Applicable Graphical Layout Editor). És un programa de disseny de diagrames i PCBs amb autoenrutador. És molt conegut arreu del món pels projectes electrònics DiY ("Do It Yourself"), degut a que moltes versions d'aquest programa tenen una llicència gratuïta i amb gran quantitat de biblioteques de components disponibles al voltant de la xarxa.

EAGLE conté un editor de diagrames electrònics. Els components poden ser col·locats al diagrama amb un sol clic i fàcilment enrutables amb altres components mitjançant cables o etiquetes. Aquest entorn també conté un editor de PCBs amb un autoenrutador bastant eficient. L'editor és capaç de produir arxius GERBER i a més, que són utilitzats en el moment de la producció. EAGLE porta un gran ventall de biblioteques de components incloses, senzilles de fer i disponibles per part d'empreses com SparkFun o aficionats que les distribueixen a través de la xarxa de forma gratuïta. Es poden trobar fàcilment molts fòrums i molts tutorials aptes per començar a utilitzar l'EAGLE. L'aprenentatge es facilita gràcies a la intuïtiva interfície gràfica.

## 7. INTERFÍCIE ENTRE LA PLACA MICROCONTROLADA I EL PC

### 7.1. Desenvolupament del programa de comunicacions

L'entorn utilitzat per adquirir les dades enviades pel PIC al port sèrie de l'ordinador és Microsoft Visual C Sharp. La principal raó de la seva utilització és perquè es trobava disponible al laboratori i era suficientment apte per la nostra aplicació.

C Sharp és un llenguatge de programació orientat a objectes desenvolupat i estandarditzat per Microsoft com a part de la seva plataforma .NET. La seva sintaxis bàsica deriva de C/C++.

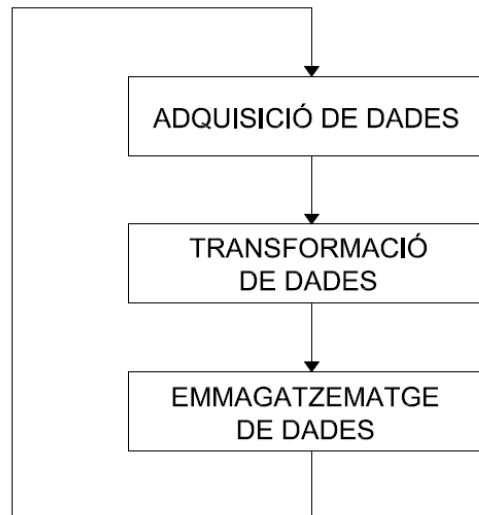


Figura 32. Diagrama de flux del funcionament del programa de comunicacions

El programa desenvolupat està configurat amb una velocitat de transmissió de 19200bits/s a 8 bits sense paritat i un bit de parada. Mentre es van llegint les dades que van arribant pel port sèrie es transformen: per l'acceleròmetre apliquem els càlculs necessaris per deixar l'acceleració expressada en  $m/s^2$ , pel giroscòpic també transformem les dades de manera que situem els 0°/s a 1.35V, i per la brúixola expressem les dades en graus respecte el nord. Seguidament es guarden les dades en el fitxer "dades.txt". El programa està adjuntat al final d'aquest document i comentat.

## 7.2. Processament de les dades i representació

L'entorn utilitzat per adquirir les dades del document "dades.txt" i representar-les en un gràfic 3D és el MATLAB.

Aquest entorn és un software matemàtic que ofereix un entorn de desenvolupament integrat amb un llenguatge de programació propi (llenguatge M). Entre les prestacions bàsiques s'hi troben: la manipulació de matrius, la representació de dades i funcions, la implementació d'algoritmes, la creació d'interfícies d'usuari i la comunicació amb programes amb altres llenguatges i amb altres dispositius hardware. És un software molt utilitzat a les universitats i centres d'investigació i desenvolupament.

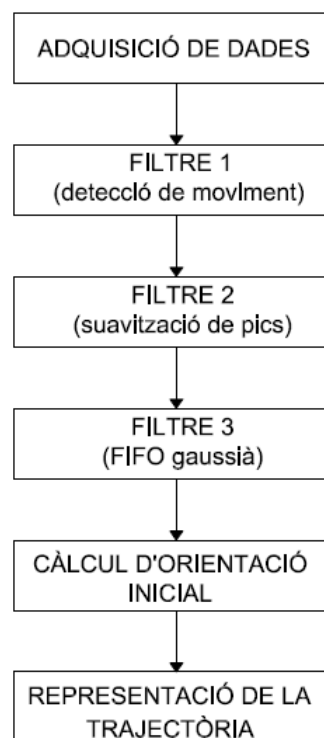


Figura 33. Diagrama de flux del funcionament del programa de processament i representació de dades

S'ha desenvolupat un programa en llenguatge M capaç de llegir les dades del document "dades.txt", filtrar-les i interpretar-les en un gràfic 3D. Inicialment, les dades passen per un filtre que anul·la l'acceleració quan el robot està parat, després per un altre filtre el qual suavitza els canvis bruscs d'acceleració lineal i velocitat angular, i finalment per un filtre FIFO gaussià. Aquest últim filtre, es basa en una taula de 5 posicions amb diferents ponderacions cada una (gauss): 0.25, 0.5, 1, 0.5 i 0.25 per les posicions 1,2,3,4 i 5

respectivament, de manera que cada dada queda influïda per les del seu voltant però donant menys importància a aquestes. Totes les dades passen per aquest filtre mitjançant el mètode FIFO, és a dir, a mesura que entren per un costat, en van sortint per l'altre. Així les dades es suavitzen les unes amb les altres i s'eliminen possibles pics. Totes aquestes manipulacions de dades són necessàries a causa de la gran quantitat de soroll que s'infiltra.

Seguidament, es porten a terme els diferents càlculs matricials. Primer es calcula la orientació inicial amb les 100 primeres dades de cada eix (els 5 segons d'estabilització del programa del PIC). A partir d'aquí, les pròximes dades representen la trajectòria del robot i es guarden en forma de llista de matrius en el document "matrius.txt". A partir d'aquest document, on cada matriu indica la posició i la orientació cada 50ms, s'obté la gràfica fent ús de funcions pròpies de MATLAB com `patch()`, `line()` i `plot3()`. Aquest programa també està adjuntat juntament amb els altres al final d'aquest document.

## 8. EINES MATEMÀTIQUES PER LA LOCALITZACIÓ ESPAIAL

### 8.1. Matrius de transformació homogènia

Les dades d'acceleració lineal i velocitat angular són fàcilment convertibles a posicions i orientacions mitjançant les lleis de la dinàmica. En el projecte, el robot descriurà un moviment en l'espai tridimensional, que es pot descomposar en els tres eixos de coordenades i treballar amb tres Moviments Rectilinis Uniformement Accelerats (MRUA). A partir de l'acceleració lineal es pot obtenir la velocitat lineal per mitjà de la integració, i a partir de la velocitat es pot obtenir la posició integrant altra vegada.

$$v = v_0 + \int_{t_0}^t a \, dt \quad [\text{m/s}] \quad (\text{Eq. 13})$$

$$p = p_0 + \int_{t_0}^t v \, dt \quad [\text{m}] \quad (\text{Eq. 14})$$

Però únicament amb la translació no es pot calcular cap posició vàlida, ja que tot cos té una orientació. El giroscòpic dona la velocitat angular, de manera que equival a un Moviment Circular Uniforme (MCU). L'equació 15 representa la posició angular o angle a partir de la velocitat angular.

$$\theta = \theta_0 + \int_{t_0}^t \omega \, dt \quad [\text{rad o graus}] \quad (\text{Eq. 15})$$

A partir d'aquestes fórmules i utilitzant les matrius de transformació homogènia podem representar conjuntament la posició i la orientació, per tant, la trajectòria.

$$T = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{Rotació} & \text{Translació} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. 16})$$

### 8.2. Càlcul de l'orientació inicial

En el moment de posar en pràctica el funcionament de la placa, es va observar que l'acceleròmetre havia d'estar totalment alineat amb els tres eixos de coordenades per obtenir

dades vàlides. Com que es requereix de molta precisió per aconseguir-ho i ofereix molt poca portabilitat de cara a l'aplicació en diferents tipus de robots i plataformes, es va decidir solucionar-ho per software. S'ha implementat codi addicional al programa de l'entorn MATLAB de manera que es calcula la inclinació de l'acceleròmetre respecte un pla horitzontal. Amb això aconseguim una total portabilitat de la placa desenvolupada en qualsevol robot, però amb la condició d'esperar cinc segons inicials perquè es puguin agafar suficients dades pel càlcul de la orientació inicial. Durant aquest temps inicial, s'adquireixen 100 dades de cada una de les sis variables (tres d'acceleracions, dues de velocitats angulars i una de l'angle respecte el nord) i es fa una mitjana, la qual serveix per efectuar els càlculs. Com s'observa, tots els valors per dur a terme les operacions s'agafen en conjunts o mitjanes a causa del soroll infiltrat en les lectures.

Les eines matemàtiques emprades en aquest subapartat estan més encarades a vectors i plans en l'espai tridimensional. Amb les tres mitjanes calculades provinents de les dades d'acceleració lineal es calcularà la inclinació. Com que tots tres eixos són perpendiculars entre ells, amb la informació que ens proporcionen els eixos X i Y és suficient. Així, hem de trobar el valor del vector  $X = (nx, ny, nz)$ , del vector  $Y = (ox, oy, oz)$  i del vector  $Z = (ax, ay, az)$ . En primer lloc, calculem els angles d'inclinació dels vectors mitjançant trigonometria tal i com es mostra a les equacions 17 i 18.

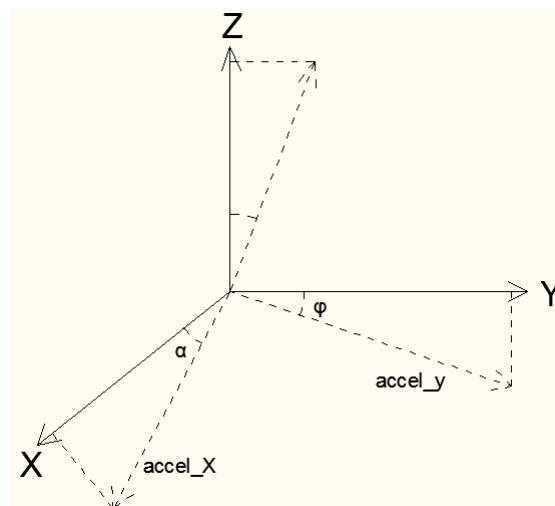


Figura 34. Representació de la inclinació real de l'acceleròmetre

$$\alpha = \sin^{-1} (\text{accel}_x / -9.81) \quad (\text{Eq. 17})$$

$$\varphi = \sin^{-1} (\text{accel}_y / -9.81) \quad (\text{Eq. 18})$$



Les variables  $accel\_x$  i  $accel\_y$  són el valor de l'acceleració mesurada en els eixos X i Y respectivament. Seguidament, fixem l'orientació X ( $ox$ ) de l'eix Y a 0 per establir una direcció inicial i fixem el centre de coordenades al punt  $P = (0, 0, 0)$ . Les següents equacions mostren els valors dels vectors X i Y.

$$X = (nx, ny, nz) = (nx, ny, \sin\alpha) \quad (\text{Eq. 19})$$

$$Y = (ox, oy, oz) = (0, \cos\varphi, \sin\varphi) \quad (\text{Eq. 20})$$

Si sabem que el producte de dos vectors perpendiculars és 0, podem aïllar  $ny$ :

$$ny = -(\sin\alpha \cdot \sin\varphi) / \cos\varphi \quad (\text{Eq. 21})$$

I aplicant el teorema de Pitàgores obtenim  $nx$ :

$$nx = \text{sqr}(1 - (-\sin\alpha \cdot \sin\varphi / \cos\varphi)^2 - \sin^2\alpha) \quad (\text{Eq. 22})$$

Ara ja podem calcular el vector Z. Com que tenim el punt  $P = (0, 0, 0)$  i els vectors X i Y, el vector normal en el punt P coincidirà amb el vector Z. Les següents equacions desenvolupen les operacions:

$$\text{Det} \begin{vmatrix} x - P_1 & X_1 & Y_1 \\ y - P_2 & X_2 & Y_2 \\ z - P_3 & X_3 & Y_3 \end{vmatrix} = \text{Det} \begin{vmatrix} x & nx & ox \\ y & ny & oy \\ z & nz & oz \end{vmatrix} = 0 \quad (\text{Eq. 23})$$

$$Z = (ny \cdot oz - oy \cdot nz, nz \cdot ox - nx \cdot oz, nx \cdot oy - ox \cdot ny) \quad (\text{Eq. 24})$$

$$T(\text{inicial}) = \begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. 25})$$

### 8.3. Càlcul de la trajectòria

La figura 35 mostra els angles de navegació. L'eix X equival a la rotació "Pitch", l'eix Y a la rotació "Roll" i l'eix Z a la rotació "Yaw".

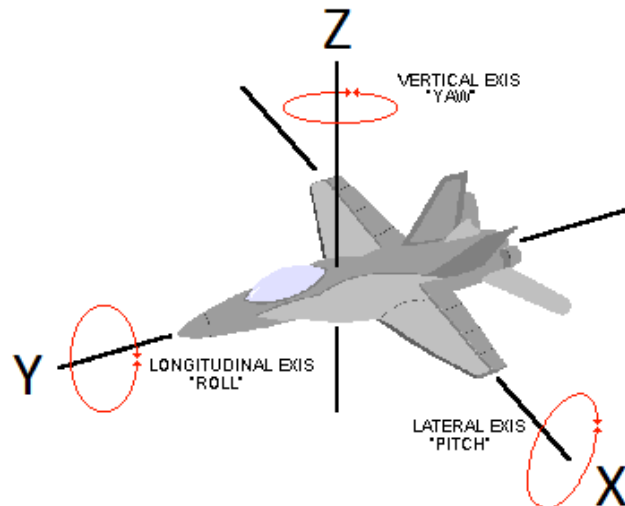


Figura 35. Angles de navegació

Es té previst que el robot circularà en un terra pla i es mourà en direcció a l'eix Y, podent girar a dretes i/o a esquerres. La placa desenvolupada també permet detectar pendents, en el cas que el robot pugés o baixés una rampa. Però es pot deixar com a una millora del projecte per més endavant, que podria ser portada a terme per futurs projectistes. Observant l'equació 26 veiem com s'expressa una translació.

$$T(p) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. 26})$$

Les equacions 27, 28 i 29 mostren les matrius de rotació al voltant dels eixos X (amb un angle de rotació  $\alpha$ ), Y (amb un angle de rotació  $\Phi$ ) i Z (amb un angle de rotació  $\theta$ ) respectivament.

$$T(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. 27})$$

$$T(y, \Phi) = \begin{bmatrix} \cos \Phi & 0 & \sin \Phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \Phi & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. 28})$$

$$T(z, \theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. 29})$$

Com s'ha esmentat anteriorment, la única matriu de rotació que ens interessa és la del gir, és a dir, la de l'eix Z. Així tindrem una matriu de translació, una de rotació. No s'ha d'oblidar que la trajectòria ve determinada per tot un conjunt de punt o posicions (adquirides cada 50ms), aquestes posicions es tradueixen en matrius. En cada instant de temps es calcularà la nova matriu que determinarà la nova posició i orientació, per tant, s'haurà de tenir en compte la matriu anterior per arrossegar tots els càlculs. La matriu resultant de cada nova posició s'expressa així:

$$T = T_0 \cdot T(p) \cdot T(z, \theta) = T_0 \cdot \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. 30})$$

## 9. ANÀLISI DELS RESULTATS EXPERIMENTALS OBTINGUTS

En aquest apartat es pretén fer una interpretació de les dades finals obtingudes amb el MATLAB. Abans de desenvolupar la PCB i soldar-hi els components, es van dur a terme petites proves del funcionament del circuit sobre la plataforma USB-PIC'School per tal d'assegurar que la placa final actuï com es pretén. Aquestes petites experiències es basen en petits moviments rectilinis sobre la pròpia taula de treball.

### 9.1. Experiències amb la placa de prototipatge

Es tracta de petites trajectòries per comprovar el funcionament de la placa. S'han efectuat moviments de translació pura i de roto-translació. Inicialment, s'ha efectuat un moviment rectilini a través de l'eix X. Consisteix en moure, amb el màxim de suavitat, la plataforma linealment en direcció a l'eix X durant un recorregut d'un metre.

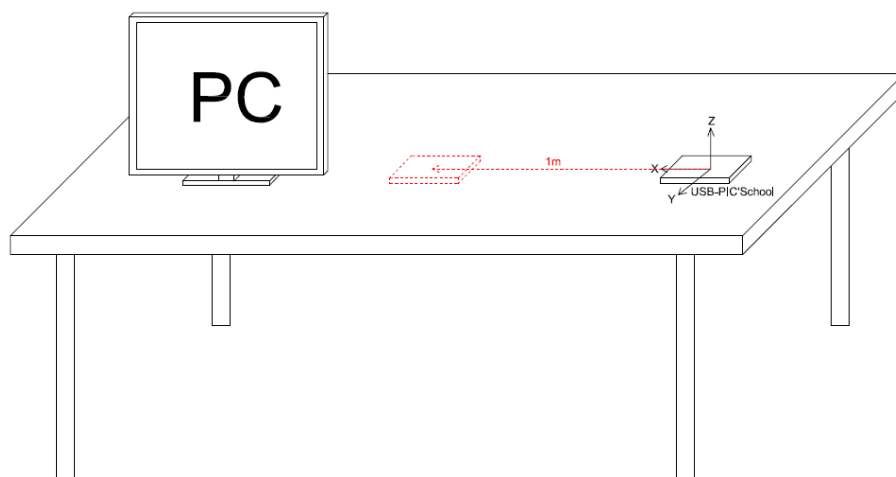


Figura 36. Representació del moviment de la primera experiència

El primer detall que s'ha observat ha estat que les proves efectuades a prop d'aparells electrònics els quals generen un camp magnètic considerable fan variar lleugerament les lectures dels sensors.

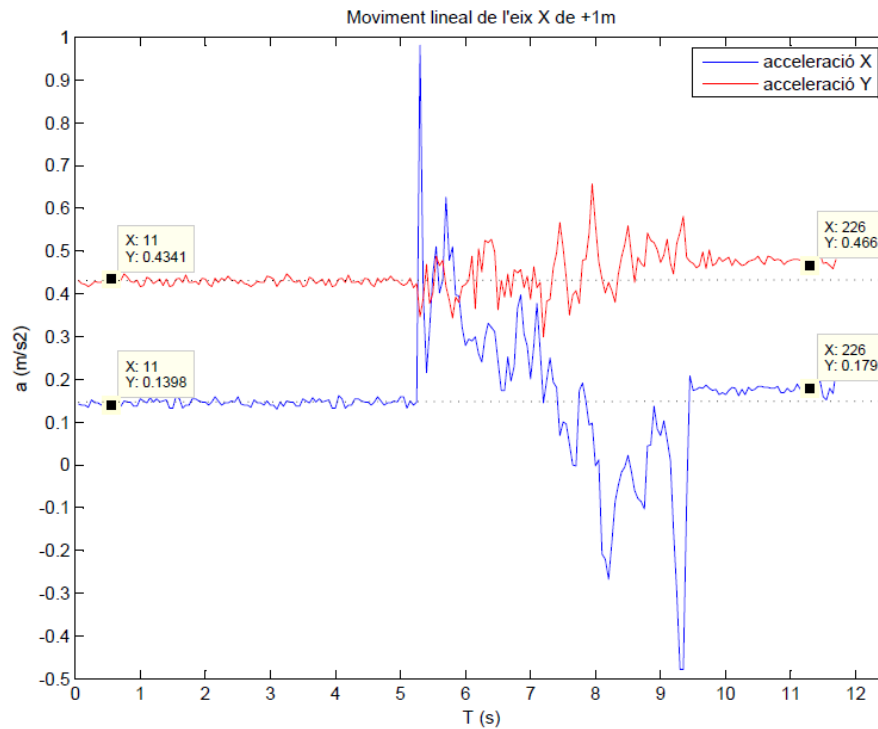


Figura 37. Resposta de l'acceleròmetre (eixos X i Y) en la primera experiència

Aquesta figura mostra les dades obtingudes del moviment. S'observa una considerable inestabilitat de les dades, raó pel qual han de passar per varis filtres abans de ser tractades. També veiem que les dues acceleracions no són  $0\text{m/s}^2$ , sinó  $0.14\text{ m/s}^2$  en l'eix X i  $0.43\text{ m/s}^2$  en l'eix Y. Aquí es reflecteix la dificultat que hi ha en col·locar l'acceleròmetre totalment alineat amb els eixos de coordenades. La solució per aquest problema ja s'ha comentat anteriorment: fer un calibratge inicial per software, millorant les prestacions de la placa en quant a portabilitat en altres plataformes de manera eficaç i senzilla. I l'altre detall a destacar és l'efecte dels camps magnètics exteriors sobre les dades. Ens fixem que el valor mig de l'acceleració en estat de repòs en els dos eixos ha augmentat al final del moviment, ja que la placa s'ha mogut i conseqüentment la percepció dels camps magnètics. Com és obvi, al efectuar un moviment positiu en l'eix X, el sensor detecta una acceleració a l'inici (l'arrencada) i una desacceleració al final (la frenada).

L'acceleració en l'eix Z (la gravetat) sembla que es manté a  $-9.45\text{ m/s}^2$  a causa de la inclinació de la placa. La figura 38 ho representa.

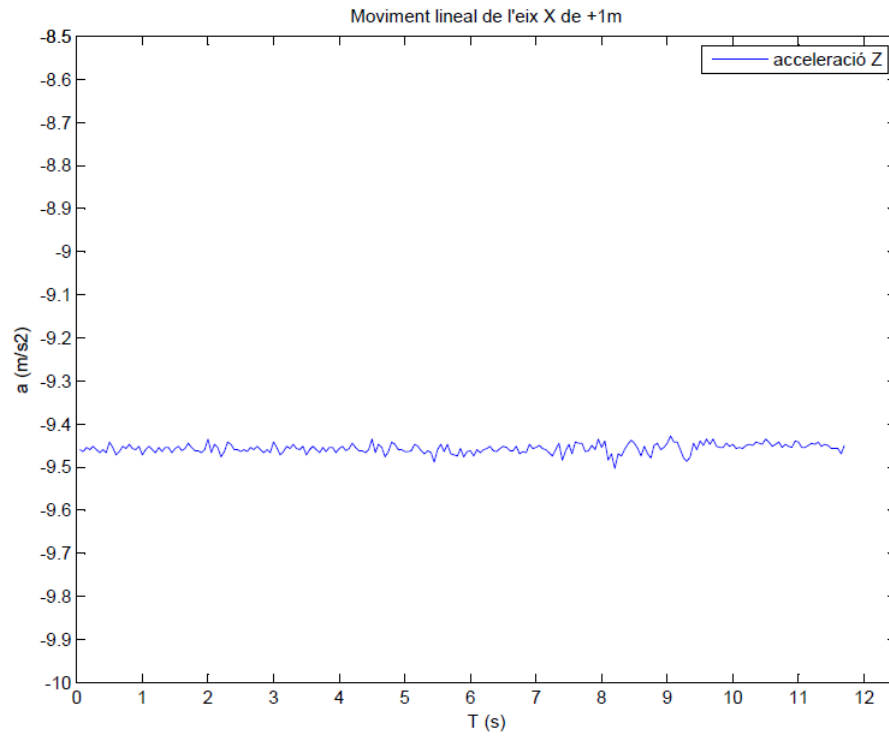


Figura 38. Resposta de l'acceleròmetre (eix Z) en la primera experiència

La següent figura 38 mostra les dades del giroscòpic.

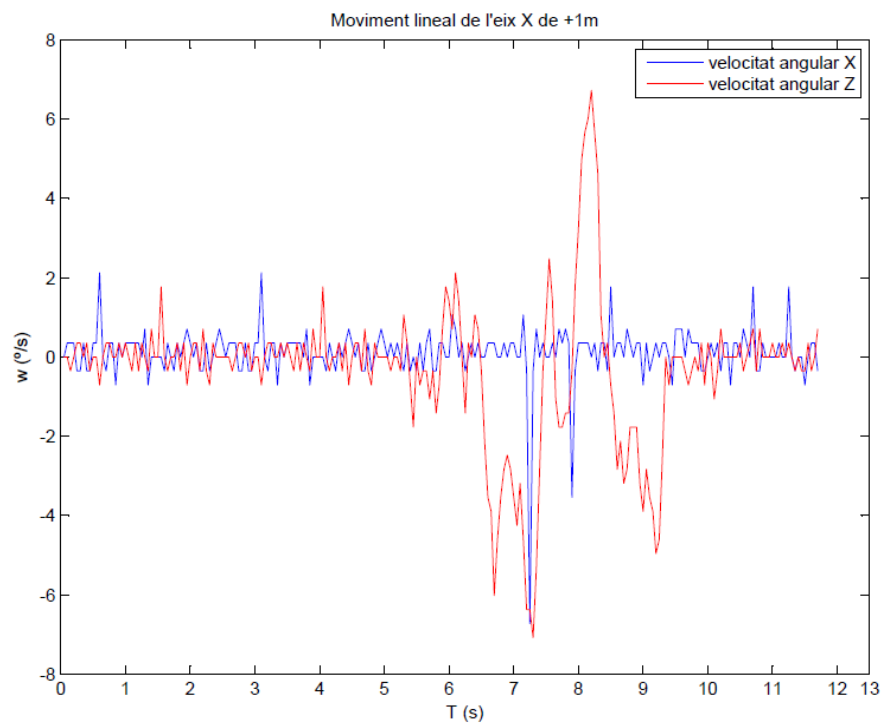


Figura 38. Resposta del giroscòpic en la primera experiència

S'observa que els valors són força variables, però sempre dins un rang. Els pics que es poden apreciar a partir dels 5s a l'eix Z (eix de rotació), és a dir, durant el moviment, són causa de la trajectòria efectuada manualment (no és totalment recte).

Les dades de la brúixola es mostren molt pertorbades pels camps magnètics. Tot i realitzar un moviment rectilini, aquest sensor detecta un moviment de  $12^\circ$ , amb  $130^\circ$  a l'inici i  $141^\circ$  al final de l'execució.

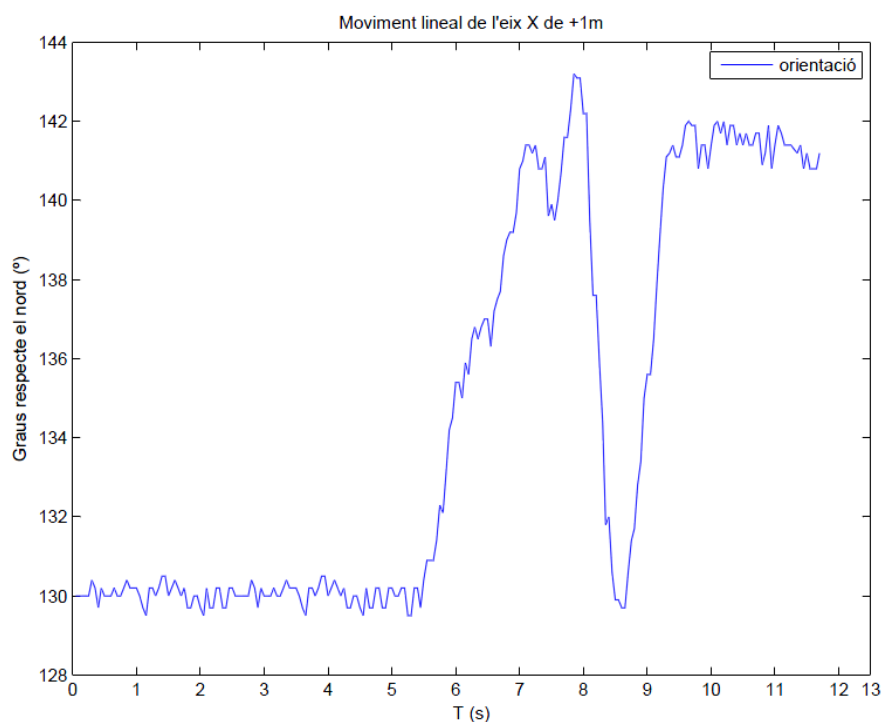


Figura 39. Resposta de la brúixola en la primera experiència

Per contrastar aquests valors dels sensors s'ha realitzat el mateix moviment però en sentit contrari, representat en la figura 40. Consisteix en moure, amb el màxim de suavitat, la plataforma linealment en direcció (però sentit contrari) a l'eix X durant un recorregut d'un metre.

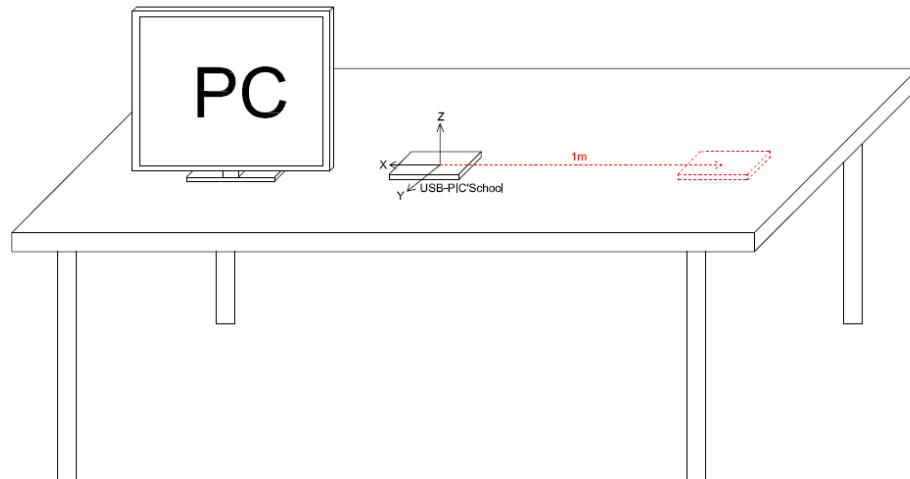


Figura 40. Representació del moviment de la segona experiència

La figura 41 mostra les dades obtingudes del moviment. En aquesta experiència ens fixem que les dues acceleracions s'aproximen a la primera, amb  $0.17 \text{ m/s}^2$  en l'eix X i  $0.48 \text{ m/s}^2$  en l'eix Y.

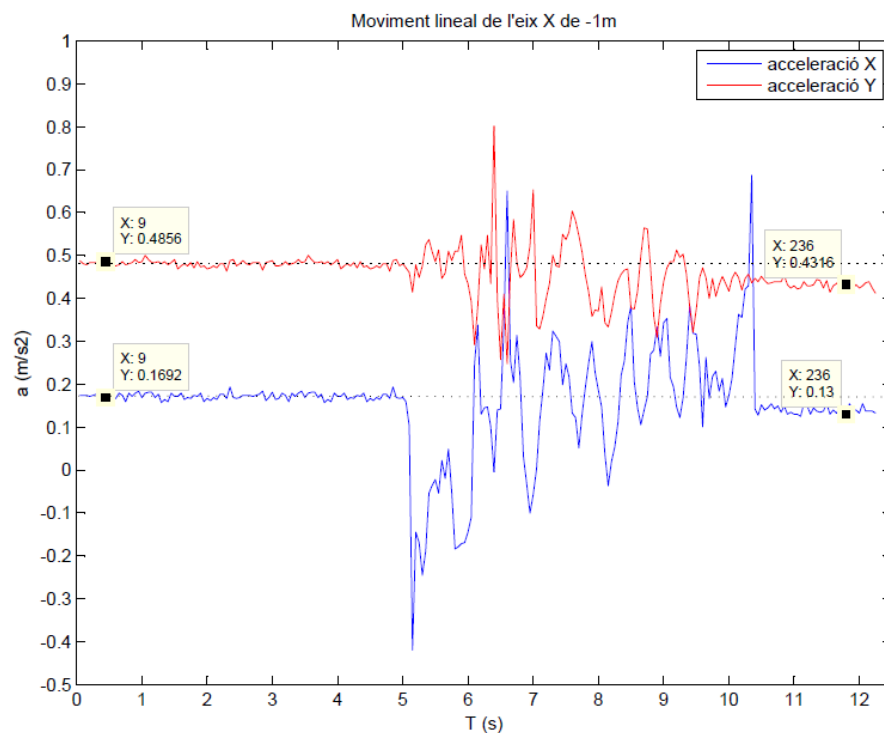


Figura 41. Resposta de l'acceleròmetre (eixos X i Y) en la segona experiència



L'aspecte més important a tenir en compte és que l'efecte dels camps magnètics exteriors sobre les dades és invers a l'anterior trajectòria. Ens fixem que el valor mig de l'acceleració en estat de repòs en els dos eixos ha disminuït al final del moviment, ja que la placa s'ha mogut i conseqüentment la percepció dels camps magnètics. Completament invers a l'experiència 1, al efectuar un moviment negatiu en l'eix X, el sensor detecta una desacceleració a l'inici (l'arrencada) i una acceleració al final (la frenada).

Per que fa a l'eix de la gravetat, es manté estable a  $-9.45\text{m/s}^2$ . Idèntica que a l'experiència anterior, ja que es tracta de la mateixa placa amb la mateixa inclinació inicial.

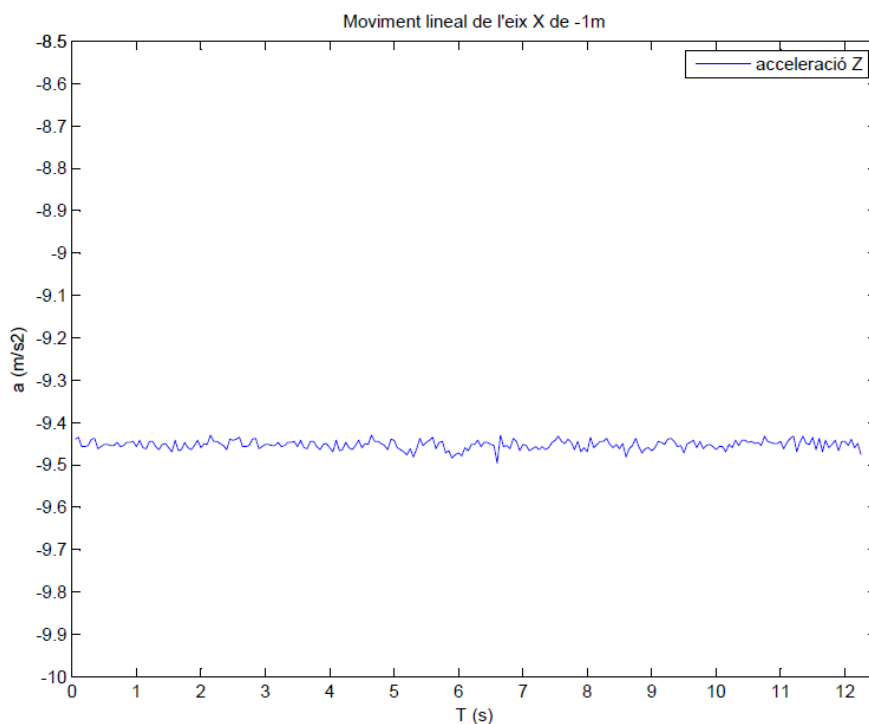


Figura 42. Resposta de l'acceleròmetre (eix Z) en la segona experiència

El sensor de velocitat angular es comporta de manera molt semblant a la trajectòria anterior, es manté a prop dels  $0^\circ/\text{s}$  però amb petites variacions indicant que el moviment efectuat no ha estat completament lineal. La figura 43 mostra les dades.

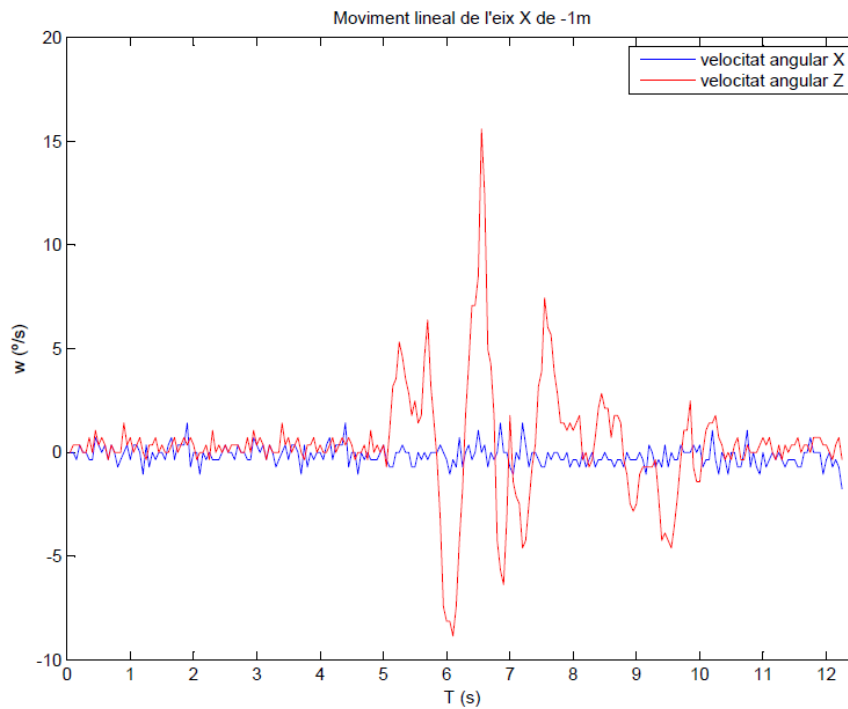


Figura 43. Resposta del giroscòpic en la segona experiència

Les dades que recull la brúixola varien de  $137^{\circ}$  a  $125^{\circ}$ , tot i ser un moviment no rotatiu. Aquí també es veu reflectit la variació de les dades davant de camps magnètics exteriors.

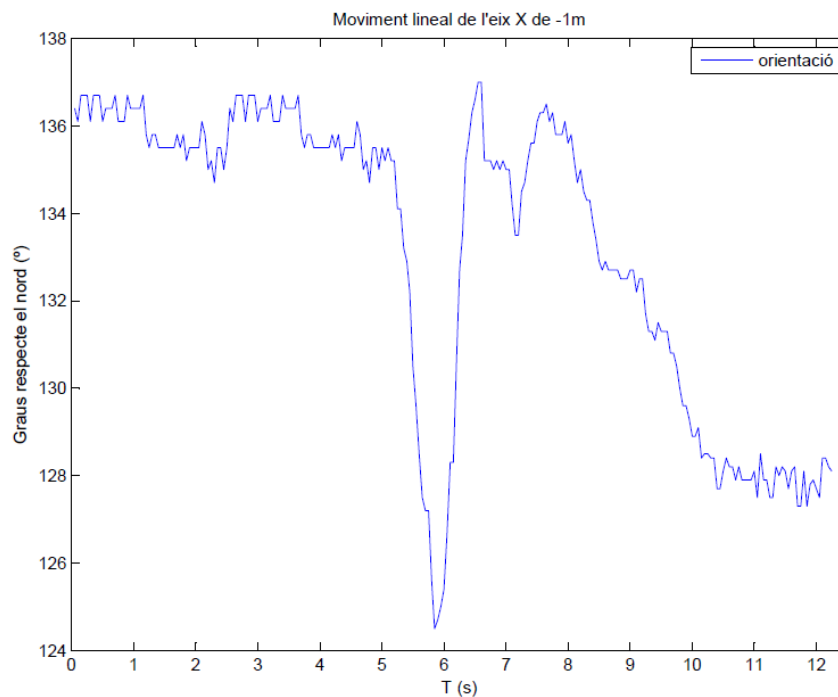


Figura 44. Resposta de la brúixola en la segona experiència

Les respostes dels sensors ens mostren una considerable inestabilitat de les dades i una influència dels camps magnètics amb alguns d'ells. D'entre els tres sensors utilitzats per obtenir les dades del medi físic, es pot concloure que l'acceleròmetre i el giroscòpic ens poden proporcionar dades fiables a l'hora de calcular la trajectòria del robot del laboratori. En canvi, la brúixola ofereix una gran dependència amb els camps magnètics i no es considera apte per obtenir-ne dades, ja que el medi on es prendran les mesures serà dins el laboratori. Això no significa que l'exclourem de la placa final, ja que en medis amb absència de camps magnètics la brúixola ens pot ser de gran utilitat en diverses aplicacions.

Així, es procedirà a fabricar la placa final amb tots els components necessaris i experimentar amb el nou producte desenvolupat.

## 9.2. Resultats finals

La fabricació de la placa s'ha dut a terme de manera ordenada i s'ha anat provant el funcionament per parts a mesura que s'anaven soldant els components per tal de solvantar eficaçment els possibles errors de fabricació. El resultat final es pot apreciar a la següent figura:

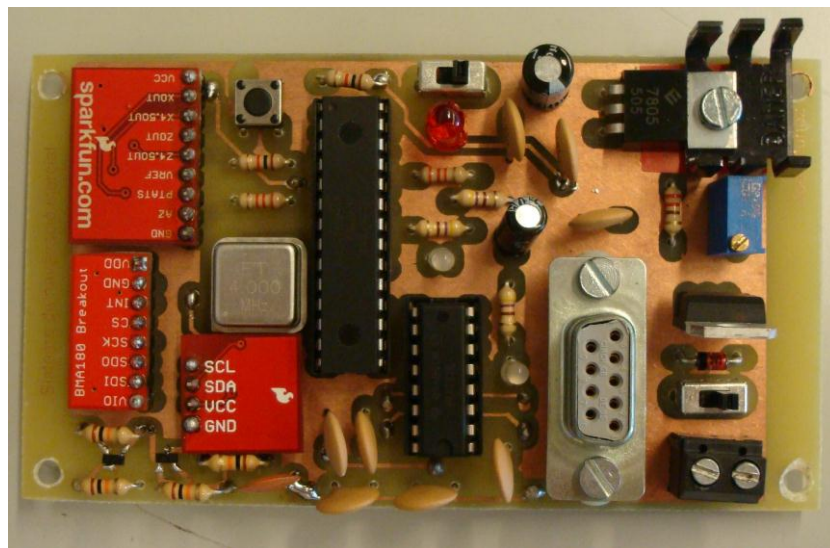


Figura 45. Placa final del projecte

Al laboratori, la placa s'ha col·locat amb la orientació correcta sobre la plataforma PRIM per dur a terme les simulacions.

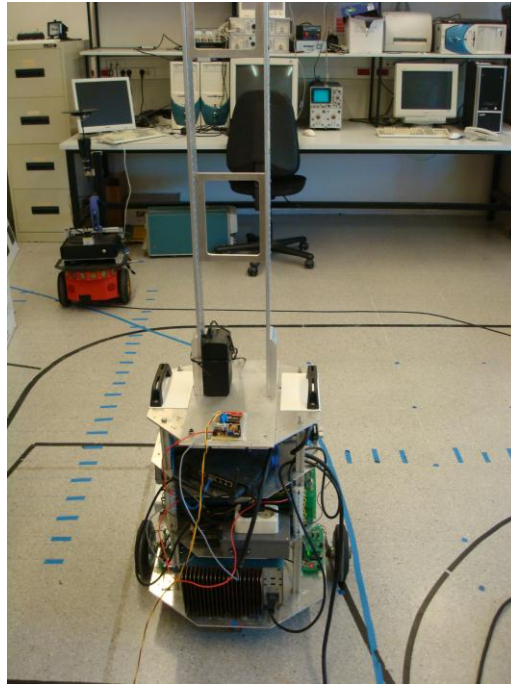


Figura 46. Robot del laboratori amb la placa inercial integrada

Amb la placa ja desenvolupada, s'ha traçat una trajectòria amb el robot en un pla horitzontal i seguint moviments rectilinis i giratoris. S'ha avançat mig metre endavant seguit d'un gir de  $90^\circ$  a la dreta i mig metre més endavant. Finalment un gir de  $90^\circ$  a l'esquerra i posteriorment un avanç de mig metre.

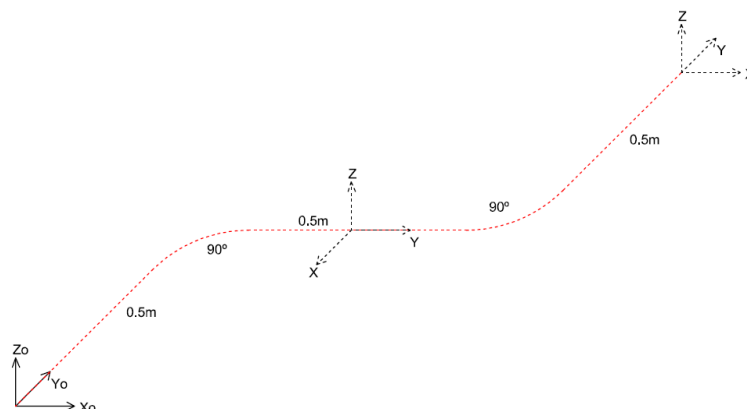


Figura 47. Representació de la trajectòria del robot

El sistema odomètric ha mesurat una posició final de 1m en l'eix Y i de 0.5m en l'eix X. Les dades són les esperades. Alhora s'han comparat amb les dades obtingudes del sistema de navegació inercial.

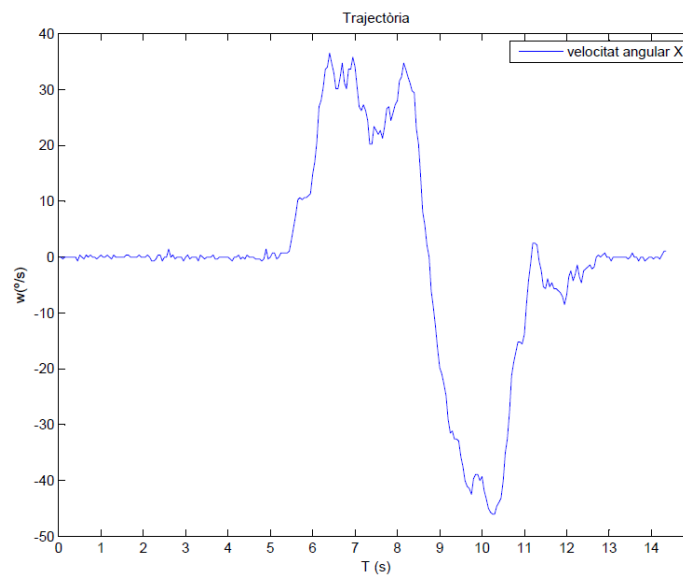


Figura 48. Representació de la velocitat angular de la trajectòria

La gràfica anterior ens mostra las dades del giroscòpic. S'observa com s'efectuen dos moviments de semblant magnitud però de sentit contrari.

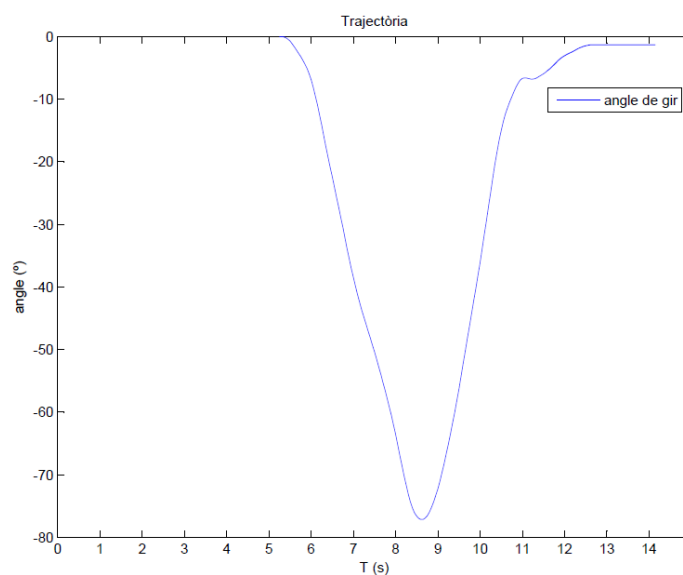


Figura 49. Representació de l'angle de gir de la trajectòria

Aquestes velocitats angulars, es tradueixen en un primer gir de  $-80^{\circ}$  (aproximadament  $90^{\circ}$ ) i un segon gir de  $80^{\circ}$  cap a la orientació inicial. Per tant, apreciem que el giroscòpic ens proporciona dades prou fiables com per calcular la trajectòria del moviment.

Pel que fa l'acceleròmetre, la següent gràfica mostra com l'acceleració de l'eix Y (en direcció al sentit del moviment) experimenta una primera fase d'acceleració inicial, una segona de moviment irregular i una tercera fase de desacceleració final de parada. En l'eix X, es pot observar de forma més clara que l'acceleració inicial correspon al gir a dretes que efectua el robot, i una acceleració negativa corresponent al gir a esquerres.

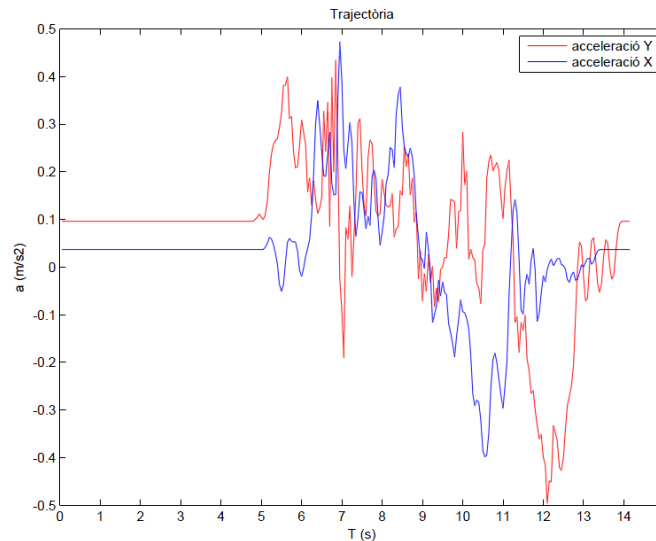


Figura 50. Representació de les acceleracions de la trajectòria

Amb el programa desenvolupat amb MATLAB per filtrar les dades i operar-les mitjançant matrius, s'ha aconseguit una simulació de la trajectòria en 3D.

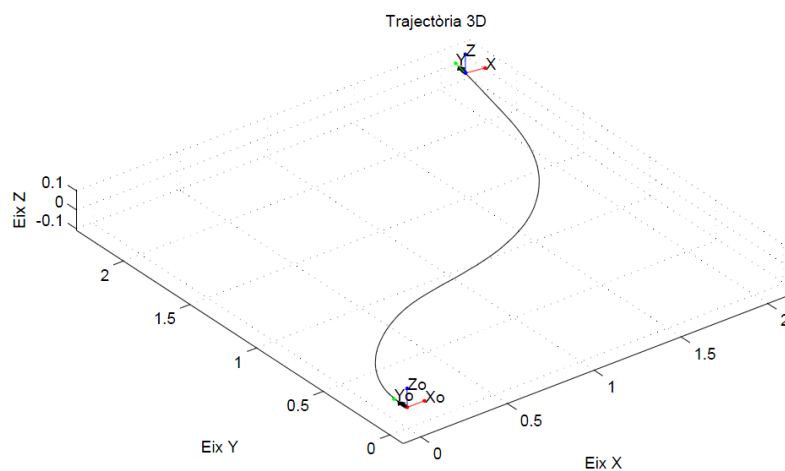


Figura 51. Representació de la trajectòria del robot mitjançant MATLAB

## **10. RESUM DEL PRESSUPOST**

L'import d'aquest projecte és de mil vuit cents nou euros i dos cèntims sense iva.

## 11. CONCLUSIONS

La realització del projecte s'ha dut a terme tal i com s'ha especificat a la introducció. S'ha dissenyat i construït un sistema de navegació inercial format per un acceleròmetre, un giroscòpic i una brúixola. Mitjançant diferents protocols de comunicació s'han enviat les dades i a través de diferents entorns informàtics s'han tractat i representat les dades. S'ha comprovat el correcte funcionament del sistema i s'ha comparat amb el sistema odomètric actual. El circuit imprès s'ha desenvolupat amb èxit, igual que els resultats. A més, es deixa camí per a un futur projecte per a una manipulació de les dades més acurada i amb la corresponent reducció de soroll.

Es tracta d'un projecte molt complet, ja que es toquen molts camps diferents, com diferents entorns informàtics i llenguatges de programació, disseny i desenvolupament d'un circuit electrònic, diferents eines matemàtiques, etc. A més, s'han ampliat considerablement els coneixements en electrònica.

Joel Muñoz Puigdevall

Enginyer Tècnic Industrial Especialitat en Electrònica Industrial

Girona, 4 de juny del 2012



## **12. RELACIÓ DE DOCUMENTS**

Els documents que formen el projecte són: Memòria, Plànols, Plec de condicions, Estat d'amidaments i Pressupost.

### 13. BIBLIOGRAFIA

BARRIENTOS, ANTONIO, FELIPE PEÑÍN, LUIS. Fundamentos de robòtica. McGraw-HILL. 1997.

GARCÍA BREIJO, EDUARDO. Compilador C CCS y simulador Proteus para microcontroladores PIC. Marcombo. cop. 2009 2a ed.

GARRIDO GONZÁLEZ, ANTONI. Física II. EDEBÉ. 2007 5a ed.

VALDÉS PÉREZ, FERNANDO E., PALLÀS ARENY, RAMÓN. Microcontroladores, fundamentos y aplicaciones con PIC. Marcombo. 2007.

BOSCH, BMA180 Data Sheet. (<http://irtfweb.ifa.hawaii.edu/~tcs3/jumpman/jumpppc/1107-BMA180/BMA180-DataSheet-v2.5.pdf>, 30 de maig de 2012)

FAIRCHILD SEMICONDUCTOR, BSS138 Data Sheet.  
(<http://www.adafruit.com/datasheets/BSS138.pdf>, 30 de maig de 2012)

FAIRCHILD SEMICONDUCTOR, LM7805 Data Sheet.  
(<http://www.fairchildsemi.com/ds/LM/LM7805.pdf>, 30 de maig de 2012)

HITACHI, HD44780U (LCD-II) Datasheet. (<http://pdf1.alldatasheet.com/datasheet-pdf/view/63673/HITACHI/HD44780.html>, 30 de maig de 2012)

HONEYWELL, HMC6352 Data Sheet.  
(<http://www.sparkfun.com/datasheets/Components/HMC6352.pdf>, 30 de maig de 2012)

INVENSENSE, IXZ500 Data Sheet. (<http://invensense.com/mems/gyro/documents/PS-IXZ-0500B-00-03.pdf>, 30 de maig de 2012)

MICROCHIP, PIC16F88X Data Sheet.  
(<http://ww1.microchip.com/downloads/en/DeviceDoc/41291D.pdf>, 30 de maig de 2012)

NATIONAL SEMICONDUCTOR, LM317 Data Sheet.

(<https://www.national.com/ds/LM/LM117.pdf>, 30 de maig de 2012)

TEXAS INSTRUMENTS, MAX232 Data Sheet.

(<http://www.datasheetcatalog.org/datasheet/texasinstruments/max232.pdf>, 30 de maig de 2012)

WIKIPEDIA, Acelerómetro. (<http://es.wikipedia.org/wiki/Aceler%C3%B3metro>, 30 de maig de 2012)

WIKIPEDIA, Giróscopo. (<http://es.wikipedia.org/wiki/Gir%C3%B3scopo>, 30 de maig de 2012)

WIKIPEDIA, Brújula. (<http://es.wikipedia.org/wiki/Br%C3%BAjula>, 30 de maig de 2012)

WIKIPEDIA, I<sup>2</sup>C. (<http://es.wikipedia.org/wiki/I%C2%B2C>, 30 de maig de 2012)

WIKIPEDIA, RS-232. (<http://es.wikipedia.org/wiki/RS-232>, 30 de maig de 2012)

## 14. GLOSSARI

Autoenrutador: Eina de connexió entre components automàtica.

BreakPoints: punts de ruptura per a la depuració de programes.

Dinamòmetre: instrument utilitzat per mesurar forces o pesar objectes.

Encoders incrementals: aparell per mesurar velocitat, posició angular i posició lineal.

E/S: Entrada/Sortida.

FET: Transistor d'Efecte de Camp.

ICSP: In-Circuit Serial Programming.

I<sup>2</sup>C: Inter-Integrated Circuit. Protocol de comunicació sèrie síncron.

LSB: Least Significant Bit.

MSB: Most Significant Bit.

PCB: Printed Circuit Board.

RAM: Random-Access Memory.

Resistència de pull-up: resistència connectada a +Vcc per assegurar l'estat alt.

RISC: Reduced Instruction Set Computer.

RS-232: protocol de comunicació sèrie amb nivells de tensió normalitzats de  $\pm 7.5V$ .

Sistema odomètric: estimació de la posició i orientació del robot a partir de les voltes efectuades per les rodes d'aquest.

TTL: lògica que utilitza 5V per l'estat alt i 0V per l'estat baix.

USB: Bus Sèrie Universal.

1-Wire: protocol de comunicació sèrie per un sol cable.

## A. PROGRAMA INTEGRAT EN EL MICROCONTROLADOR (ASSEMBLADOR)

```

;*****
;*****

;COMUNICACIÓ DEL PIC 16F886 AMB L'ACCELERÒMETRE BMA180 PER MITJÀ DEL PROTOCOL DE COMUNICACIÓ
;I2C

;*****
;*****

List    p=16F886                ;Tipus de microcontrolador
include "P16F886.inc"          ;Definicions de registres interns

__config    _CONFIG1, _LVP_OFF&_PWRTE_ON&_WDT_OFF&_EC_OSC&_FCMEN_OFF    ;Registre 1 de
                                                ;configuració
__config    _CONFIG2, _WRT_OFF&_BOR40V                ;Registre 2 de configuració

Memoria     equ            0x20                ;Variable de memòria
Comptador   equ            0x21                ;Variable de temporització
Retard_var  equ            0x22                ;Variables de memòria (Rutina de
temporització)

                org            0x00                ;Vector RESET
                goto           Inici
                org            0x05                ;Vector CODI PROGRAMA

include "BMA180.inc"           ;Rutines per l'acceleròmetre BMA180
include "IXZ500.inc"           ;Rutines pel giroscòpic IXZ500
include "HMC6352.inc"         ;Rutines per la brúixola HMC6352
include "I2C.inc"             ;Rutines per la comunicació I2C
include "UART.inc"            ;Rutines per la comunicació per UART

;*****
;FUNCIONS
;*****
;Inicialització del mòdul CCP1 per enviar dades al PC cada 50ms:
CCP1_I      clrf    T1CON
                clrf    CCP1CON
                bsf    STATUS,RP0
                bcf    PIE1,TMR1IE
                bcf    PIE1,CCP1IE
                bcf    STATUS,RP0
                clrf    PIR1
                movlw  b'00110001'
                movwf  T1CON                ;Seleccionar rellotge intern y preescaler de 8
                movlw  0x0B
                movwf  CCP1CON            ;Seleccionar mode comparador, amb reset al Timer1
                movlw  0x18
                movwf  CCPR1H

```

```

        movlw  0x6A
        movwf  CCP1L          ;CCPR1 = 50ms
        bsf   T1CON,TMR1ON
        return

;*****
;Inicialització del LED que indica que la placa ja està inicialitzada i preparada per enviar
valors vàlids de trajectòria:
LED_I      bsf           STATUS,RP0
           bsf           STATUS,RP1      ;Banc 3
           bcf           ANSELH,ANS12    ;RB0 digital
           bcf           ANSELH,ANS10    ;RB1 digital
           bcf           STATUS,RP1      ;Banc 1
           bcf           TRISB,0         ;RB0 sortida
           bcf           TRISB,1         ;RB1 sortida
           bcf           STATUS,RP0      ;Banc 0
           bcf           PORTB,0
           bsf           PORTB,1         ;Estat inicial, RB0=0(led verd) i RB1=1(led
vermell)
           return

;*****
;Esperem 1 segon tant per a l'estabilitat de la plataforma com l'alimentació dels sensors:
ESTABILITAT  movlw  .20
              movwf  Memoria
ESTAB_W      movlw  .250
              movwf  Retard_1
              call   Retard
              decfsz Memoria
              goto   ESTAB_W
              return

;*****
;PROGRAMA PRINCIPAL
;*****
Inici        call   UART_I          ;Inicialitzar comunicació per UART
              call   I2C_I          ;Inicialitzar bus I2C
              call   BMA180_I       ;Inicialitzar connexió entre PIC-BMA180
              call   IXZ500_I       ;Inicialitzar connexió entre PIC-IXZ500
              call   HMC6352_I      ;Inicialitzem connexió entre PIC-HMC6352
              call   LED_I          ;Inicialitzar el LED indicador
              call   ESTABILITAT    ;Esperem un temps prudencial d'estabilitat
              movlw  .100
              movwf  Comptador      ;100x50ms=5s (temps d'espera per la calibració
                                   inicial de la placa
                                   ;amb el MATLAB)

              ;Configuració direccionament indirecte de dades:
              bcf   STATUS,IRP      ;Bancs 0 i 1
              call  CCP1_I         ;Inicialitzar el mòdul del CCP1
Loop         movlw  0x30

```

```

movwf FSR ;Direcció a partir de la qual s'aniran guardar
;totes les dades (0x30)

;Lectura d'acceleració:
BMA180_W btfss PORTA,4 ;Dades d'acceleració disponibles?
goto BMA180_W ;No, esperar
movlw 0x02
movwf Memoria ;Eix X
call BMA180_W ;Llegir dades d'acceleració
movlw 0x04
movwf Memoria ;Eix Y
call BMA180_W ;Llegir dades d'acceleració
movlw 0x06
movwf Memoria ;Eix Z
call BMA180_W ;Llegir dades d'acceleració

;Lectura de velocitat angular:
movlw b'01000001'
movwf Memoria ;Eix X (XOUT)
call IXZ500_W ;Llegir dades de velocitat angular
movlw b'01000101'
movwf Memoria ;Eix Z (ZOUT)
call IXZ500_W ;Llegir dades de velocitat angular

;Lectura de la brúixola:
call HMC6352_W ;Llegir dades de la brúixola

;Transmissió de dades al PC per UART:
movlw 0x30
movwf FSR ;Direcció a partir de la qual s'aniran envian
;totes les dades (0x30)

movlw .12
movwf Memoria ;Enviar 12 dades (6 acceleròmetre + 4 giroscòpic
call UART_T ; + 2 brúixola)

;Bucle d'error, la lectura de les dades dels sensors han tardat més de 50ms:
bucle bsf PORTB,1
bcf PORTB,0
btfsc PIR1,CCP1IF ;Han passat 50ms o més?
goto bucle ;Si, error! Reiniciar el PIC.

;Comprovació dels 5s per a la calibració de les dades per MATLAB:
movf Comptador,W
sublw .0
btfss STATUS,C ;Comptador = 0?
decf Comptador ;No, decrementa
btfss STATUS,C ;Comptador = 0?
goto Compara ;No, seguim amb el led vermell activat
bcf PORTB,1 ;Si, desactivem led vermell i activem led verd.
;Calibració concluida! Tot
bsf PORTB,0 ;preparat per enviar dades de la trajectòria

```



```
                ;Espera que hagin passat 50ms. Si és així, es resseteja el CCP1 i es repeteix
                ;el cicle:
Compara        btfs   PIR1,CCP1IF           ;Han passat justament 50ms?
                goto   Compara              ;No, espera.
                bcf    PIR1,CCP1IF         ;Si, re-set a CCP1 i repetim cicle
                btfsc  PORTB,2             ;Calibrar brúixola?
                call   HMC6352_C          ;Si
                goto   Loop

                end                          ;Fi del programa

;*****
```

## B. PROGRAMA DE RECEPCIÓ DE DADES (C)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO.Ports;

namespace proves
{
    class Program
    {
        static bool acabar = false;

        static void Main(string[] args)
        {
            // Declarar variables
            byte[] rebut = new byte[12];
            String liniaPerEscriure = String.Empty;

            // Fitxer

            // Port sèrie
            SerialPort ps = new SerialPort("COM1", 19200, Parity.None, 8, StopBits.One);
            ps.Open();

            byte[] dadesRebudes = new byte[12];           // Dades rebudes
            double[] dadesTransformades = new double[6];  // Dades transformades
            bool primeraLectura = true;
            double[] zeroVelocitatAngular = new double[2];

            // Fer el programa ppal
            while (!acabar)
            {
                liniaPerEscriure = String.Empty;
                for (int i = 0; i < 12; i++)
                {
                    ps.Read(rebut, 0, 1);
                    dadesRebudes[i] = rebut[0];
                }

                // dadesRebudes està plena amb els valors llegits. Transformem valors
                // Pels eixos treiem els 2 últims bits, fem el C2 i multipliquem pel factor de
                // conversió
                for (int i = 0; i < 3; i++)
                {
                    byte v1, v2;
                    v1 = dadesRebudes[2 * i];           // Byte alt
                    v2 = dadesRebudes[(2 * i) + 1];    // Byte baix
                    v2 = (byte)(v2 & 0xFC);           // Posar els 2 últims bits a 0

                    dadesTransformades[i] = (complementA2(v1, v2) >> 2) * 0.00025 * 9.81;
                }
                //Apliquem els valors de calibració de l'acceleròmetre
                dadesTransformades[0] = (dadesTransformades[0] + 0.06) * 9.81 / 9.95;
                dadesTransformades[1] = (dadesTransformades[1] + 0.03) * 9.81 / 9.97;
                dadesTransformades[2] = (dadesTransformades[2] - 0.55) * 9.81 / 10.05;
                // Pels valors analògics apliquem el factor de conversió i situem el zero real
                for (int i = 3; i < 5; i++)
                {
                    dadesTransformades[i] = ((dadesRebudes[2 * i] << 8) + dadesRebudes[(2 * i) +
1]) * 3.3 / (1024 * 0.0091);
                    //Dades de calibració del giroscòpic a 0°/s
                    if (primeraLectura == true)
                    {
                        zeroVelocitatAngular[i - 3] = dadesTransformades[i];
                    }
                    dadesTransformades[i] = dadesTransformades[i] - zeroVelocitatAngular[i -
3];
                }
                primeraLectura = false;

                // Pels valors de la brúixola apliquem el factor de conversió
            }
        }
    }
}

```

```
        dadesTransformades[5] = ((dadesRebudes[10] << 8) + dadesRebudes[11]) * 1.0 /
10;

        for (int i = 0; i < 6; i++)
        {
            liniaPerEscriure = liniaPerEscriure +
dadesTransformades[i].ToString().Replace(',', '.');
            if (i < 5) liniaPerEscriure = liniaPerEscriure + ", ";
        }

        Console.WriteLine("Valors finals: " + liniaPerEscriure);

        System.IO.StreamWriter fitxer = new System.IO.StreamWriter(@"d:\JOEL\log.txt",
true);
        fitxer.WriteLine(liniaPerEscriure);
        fitxer.Close();
    }

    ps.Close();
}

// Altres funcions
static short complementA2(byte byteH, byte byteL)
{
    byte[] data = new byte[2];
    data[0] = byteL;
    data[1] = byteH;
    short res = System.BitConverter.ToInt16(data, 0);
    return (res);
}
}
```

## C. PROGRAMA DE CÀLCUL I SIMULACIÓ DE LA TRAJECTÒRIA (MATLAB)

```

close all
clear all
clc
%Carreguem fitxer i borrem possibles fitxers anteriors
D=load('dades.txt');
delete('matrius.txt')
%Dades originals
D_AX=D(:,2);
D_AY=D(:,1);
D_AZ=D(:,3);
D_WX=D(:,4);
D_WZ=D(:,5);
    %Filtrem dades
    Filtre=zeros(100,1);
    DF_AX=zeros(length(D_AX),1);
    DF_AY=zeros(length(D_AY),1);
    DF_AZ=zeros(length(D_AZ),1);
    DF_WX=zeros(length(D_WX),1);
    DF_WZ=zeros(length(D_WZ),1);
    FFG=5;
    FiltreFifoGaussia=zeros(FFG,1);
    DFG_AX=zeros(length(D_AX)-(FFG-1),1);
    DFG_AY=zeros(length(D_AY)-(FFG-1),1);
    DFG_AZ=zeros(length(D_AZ)-(FFG-1),1);
    DFG_WX=zeros(length(D_WX)-(FFG-1),1);
    DFG_WZ=zeros(length(D_WZ)-(FFG-1),1);
    %Filtrem l'acceleració en l'eix X
    %Filtrem el senyal quan el robot està immòbil
    for i=1:100
        Filtre(i,1)=D_AX(i,1);
    end
    MaxFiltre=max(Filtre)+0.05;
    MinFiltre=min(Filtre)-0.05;
    DFM_AX=mean(Filtre);
    DFDM_AX=(max(Filtre)-min(Filtre))/2;
    for i=1:length(D_AX)
        if ((D_AX(i,1)>MinFiltre)&&(D_AX(i,1)<MaxFiltre)) || (i>(length(D_AX)-10))
            DF_AX(i,1)=DFM_AX;
        else
            DF_AX(i,1)=D_AX(i,1);
        end
    end
    %Filtrem el senyal quan hi ha moviment
    for i=2:length(DF_AX)
        if (DF_AX(i,1)>DF_AX(i-1,1)) && (abs(DF_AX(i,1)-DF_AX(i-1,1))>(DFDM_AX+0.001))
            DF_AX(i,1)=DF_AX(i,1)-DFDM_AX;
        else
            if (DF_AX(i,1)<DF_AX(i-1,1)) && (abs(DF_AX(i-1,1)-
            DF_AX(i,1))>(DFDM_AX+0.001))
                DF_AX(i,1)=DF_AX(i,1)+DFDM_AX;
            end
        end
    end
    %Apliquem Filtre Fifo Gaussià
    for i=1:length(DF_AX)
        if i<=FFG
            FiltreFifoGaussia(i,1)=DF_AX(i,1);
        else
            DFG_AX(i-
            FFG,1)=(FiltreFifoGaussia(1,1)/10)+(FiltreFifoGaussia(2,1)/5)+(FiltreFifoGaussi
            a(3,1)/2.5)+(FiltreFifoGaussia(4,1)/5)+(FiltreFifoGaussia(5,1)/10);
            for j=1:FFG-1
                FiltreFifoGaussia(j,1)=FiltreFifoGaussia(j+1,1);
            end
            FiltreFifoGaussia(FFG,1)=DF_AX(i,1);
        end
        if i==length(DF_AX)
            DFG_AX(i-(FFG-1),1)=DFM_AX;
        end
    end
    %Filtrem l'acceleració en l'eix Y
    %Filtrem el senyal quan el robot està immòbil

```

```

for i=1:100
    Filtre(i,1)=D_AY(i,1);
end
MaxFiltre=max(Filtre)+0.05;
MinFiltre=min(Filtre)-0.05;
DFM_AY=mean(Filtre);
DFDM_AY=(max(Filtre)-min(Filtre))/2;
for i=1:length(D_AY)
    if (D_AY(i,1)>MinFiltre)&&(D_AY(i,1)<MaxFiltre)|| (i>(length(D_AY)-10))
        DF_AY(i,1)=DFM_AY;
    else
        DF_AY(i,1)=D_AY(i,1);
    end
end
end
%Filtrem el senyal quan hi ha moviment
for i=2:length(DF_AY)
    if (DF_AY(i,1)>DF_AY(i-1,1))&&(abs(DF_AY(i,1)-DF_AY(i-1,1))>(DFDM_AY+0.001))
        DF_AY(i,1)=DF_AY(i,1)-DFDM_AY;
    else
        if (DF_AY(i,1)<DF_AY(i-1,1))&&(abs(DF_AY(i-1,1)-
DF_AY(i,1))>(DFDM_AY+0.001))
            DF_AY(i,1)=DF_AY(i,1)+DFDM_AY;
        end
    end
end
end
%Apliquem Filtre Fifo Gaussià
for i=1:length(DF_AY)
    if i<=FFG
        FiltreFifoGaussia(i,1)=DF_AY(i,1);
    else
        DFG_AY(i-
FFG,1)=(FiltreFifoGaussia(1,1)/10)+(FiltreFifoGaussia(2,1)/5)+(FiltreFifoGaussi
a(3,1)/2.5)+(FiltreFifoGaussia(4,1)/5)+(FiltreFifoGaussia(5,1)/10);
        for j=1:FFG-1
            FiltreFifoGaussia(j,1)=FiltreFifoGaussia(j+1,1);
        end
        FiltreFifoGaussia(FFG,1)=DF_AY(i,1);
    end
    if i==length(DF_AY)
        DFG_AY(i-(FFG-1),1)=DFM_AY;
    end
end
end
%Filtrem l'acceleració en l'eix Z
%Filtrem el senyal quan el robot està immòbil
for i=1:100
    Filtre(i,1)=D_AZ(i,1);
end
MaxFiltre=max(Filtre)+0.05;
MinFiltre=min(Filtre)-0.05;
DFM_AZ=mean(Filtre);
DFDM_AZ=(max(Filtre)-min(Filtre))/2;
for i=1:length(D_AZ)
    if (D_AZ(i,1)>MinFiltre)&&(D_AZ(i,1)<MaxFiltre)|| (i>(length(D_AZ)-10))
        DF_AZ(i,1)=DFM_AZ;
    else
        DF_AZ(i,1)=D_AZ(i,1);
    end
end
end
%Filtrem el senyal quan hi ha moviment
for i=2:length(DF_AZ)
    if (DF_AZ(i,1)>DF_AZ(i-1,1))&&(abs(DF_AZ(i,1)-DF_AZ(i-1,1))>(DFDM_AZ+0.001))
        DF_AZ(i,1)=DF_AZ(i,1)-DFDM_AZ;
    else
        if (DF_AZ(i,1)<DF_AZ(i-1,1))&&(abs(DF_AZ(i-1,1)-
DF_AZ(i,1))>(DFDM_AZ+0.001))
            DF_AZ(i,1)=DF_AZ(i,1)+DFDM_AZ;
        end
    end
end
end
%Apliquem Filtre Fifo Gaussià
for i=1:length(DF_AZ)
    if i<=FFG
        FiltreFifoGaussia(i,1)=DF_AZ(i,1);
    else

```

```

        DFG_AZ(i-
FFG,1)=(FiltreFifoGaussia(1,1)/10)+(FiltreFifoGaussia(2,1)/5)+(FiltreFifoGaussi
a(3,1)/2.5)+(FiltreFifoGaussia(4,1)/5)+(FiltreFifoGaussia(5,1)/10);
        for j=1:FFG-1
            FiltreFifoGaussia(j,1)=FiltreFifoGaussia(j+1,1);
        end
        FiltreFifoGaussia(FFG,1)=DF_AZ(i,1);
    end
    if i==length(DF_AZ)
        DFG_AZ(i-(FFG-1),1)=DFM_AZ;
    end
end
%Filtrem la velocitat angular en l'eix X
%Filtrem el senyal quan el robot està immòbil
for i=1:100
    Filtre(i,1)=D_WX(i,1);
end
MaxFiltre=max(Filtre)+0.36;
MinFiltre=min(Filtre)-0.36;
DFDM_WX=(max(Filtre)-min(Filtre))/2;
for i=1:length(D_WX)
    if (D_WX(i,1)>MinFiltre)&&(D_WX(i,1)<MaxFiltre)|| (i>(length(D_WX)-10))
        DF_WX(i,1)=0;
    else
        DF_WX(i,1)=D_WX(i,1);
    end
end
%Filtrem el senyal quan hi ha moviment
for i=2:length(DF_WX)
    if (DF_WX(i,1)>DF_WX(i-1,1))&&(abs(DF_WX(i,1)-DF_WX(i-1,1))>(DFDM_WX+0.001))
        DF_WX(i,1)=DF_WX(i,1)-DFDM_WX;
    else
        if (DF_WX(i,1)<DF_WX(i-1,1))&&(abs(DF_WX(i-1,1)-
DF_WX(i,1))>(DFDM_WX+0.001))
            DF_WX(i,1)=DF_WX(i,1)+DFDM_WX;
        end
    end
end
%Apliquem Filtre Fifo Gaussià
for i=1:length(DF_WX)
    if i<=FFG
        FiltreFifoGaussia(i,1)=DF_WX(i,1);
    else
        DFG_WX(i-
FFG,1)=(FiltreFifoGaussia(1,1)/10)+(FiltreFifoGaussia(2,1)/5)+(FiltreFifoGaussi
a(3,1)/2.5)+(FiltreFifoGaussia(4,1)/5)+(FiltreFifoGaussia(5,1)/10);
        for j=1:FFG-1
            FiltreFifoGaussia(j,1)=FiltreFifoGaussia(j+1,1);
        end
        FiltreFifoGaussia(FFG,1)=DF_WX(i,1);
    end
    if i==length(DF_WX)
        DFG_WX(i-(FFG-1),1)=0;
    end
end
%Filtrem la velocitat angular en l'eix Z
%Filtrem el senyal quan el robot està immòbil
for i=1:100
    Filtre(i,1)=D_WZ(i,1);
end
MaxFiltre=max(Filtre)+0.36;
MinFiltre=min(Filtre)-0.36;
DFDM_WZ=(max(Filtre)-min(Filtre))/2;
for i=1:length(D_WZ)
    if (D_WZ(i,1)>MinFiltre)&&(D_WZ(i,1)<MaxFiltre)|| (i>(length(D_WZ)-10))
        DF_WZ(i,1)=0;
    else
        DF_WZ(i,1)=D_WZ(i,1);
    end
end
%Filtrem el senyal quan hi ha moviment
for i=2:length(DF_WZ)
    if (DF_WZ(i,1)>DF_WZ(i-1,1))&&(abs(DF_WZ(i,1)-DF_WZ(i-1,1))>(DFDM_WZ+0.001))
        DF_WZ(i,1)=DF_WZ(i,1)-DFDM_WZ;
    else

```

```

        if (DF_WZ(i,1)<DF_WZ(i-1,1)) && (abs(DF_WZ(i-1,1)-
DF_WZ(i,1))>(DFDM_WZ+0.001))
            DF_WZ(i,1)=DF_WZ(i,1)+DFDM_WZ;
        end
    end
end
%Apliquem Filtre Fifo Gaussià
for i=1:length(DF_WZ)
    if i<=FFG
        FiltreFifoGaussia(i,1)=DF_WZ(i,1);
    else
        DFG_WZ(i-
FFG,1)=(FiltreFifoGaussia(1,1)/10)+(FiltreFifoGaussia(2,1)/5)+(FiltreFifoGaussi
a(3,1)/2.5)+(FiltreFifoGaussia(4,1)/5)+(FiltreFifoGaussia(5,1)/10);
        for j=1:FFG-1
            FiltreFifoGaussia(j,1)=FiltreFifoGaussia(j+1,1);
        end
        FiltreFifoGaussia(FFG,1)=DF_WZ(i,1);
    end
    if i==length(DF_WZ)
        DFG_WZ(i-(FFG-1),1)=0;
    end
end
%Inicialitzem valors inicials
posicionsX=zeros(length(DFG_AX),1);
posicionsX(1,1)=0;
posicionsY=zeros(length(DFG_AY),1);
posicionsY(1,1)=0;
posicionsZ=zeros(length(DFG_AZ),1);
posicionsZ(1,1)=0;
velocitatsX=zeros(length(DFG_AX),1);
velocitatsX(1,1)=0;
velocitatsY=zeros(length(DFG_AY),1);
velocitatsY(1,1)=0;
velocitatsZ=zeros(length(DFG_AZ),1);
velocitatsZ(1,1)=0;
anglesX=zeros(length(DFG_WX),1);
anglesX(1,1)=0;
anglesZ=zeros(length(DFG_WZ),1);
anglesZ(1,1)=0;
yxtotal=0;
yzttotal=0;
VXant=0;
VYant=0;
VZant=0;
%Inicialitzem matriu inicial
%Calculem angle eix X, eix Y i eix Z respecte la horitzontal
angle_eixX=asind(DFM_AX/(-9.81));
angle_eixY=asind(DFM_AY/(-9.81));
angle_eixZ=acosd(DFM_AZ/(-9.81));
%Calculem vectors X=(nx,ny,nz), Y=(ox,oy,oz) i Z=(ax,ay,az)
nx=sqrt(1-(-sind(angle_eixX)*sind(angle_eixY)/cosd(angle_eixY))^2-(sind(angle_eixX))^2);
ny=-sind(angle_eixX)*sind(angle_eixY)/cosd(angle_eixY);
nz=sind(angle_eixX);
ox=0;
oy=cosd(angle_eixY);
oz=sind(angle_eixY);
ax=(ny*oz)-(oy*nz);
ay=(nz*ox)-(nx*oz);
az=(nx*oy)-(ox*ny);
Minicial=[nx ox ax 0;ny oy ay 0;nz oz az 0;0 0 0 1];
Mant=Minicial;
%Guardem a matrius.txt la matriu inicial
save('matrius.txt','Minicial','-ascii','-double','-append')
% Càlcul de acceleració, velocitat i posició en cada instant de temps (AT=0.05s)
for i=2:length(DFG_AX)
    %Restem la gravetat per obtenir el moviment pur
    AX=DFG_AX(i,1)-(-9.81*nz);
    AY=DFG_AY(i,1)-(-9.81*oz);
    AZ=0;
    %Calculem translació en eix Y
    VY=VYant+(AY*0.05);
    PY=(VYant*0.05)+(AY*0.00125);
    VYant=VY;
    %Calculem translació en eix Z
    VZ=VZant+(AZ*0.05);

```

```

PZ=(VZant*0.05)+(AZ*0.00125);
VZant=VZ;
%Calculem rotació en eix X (Pitch)
yx=(-DFG_WX(i,1)*0.05);
yxtotal=yxtotal+yx;
%Calculem rotació en eix Z (Yaw)
yz=(-DFG_WZ(i,1)*0.05);
yzttotal=yzttotal+yz;
%Calculem acceleració normal
AN=VY*(-DFG_WZ(i,1))*0;
AX=AX-AN;
%Ara (un cop calculada AN) podem calcular translació en eix X
VX=VXant+(AX*0.05);
PX=(VXant*0.05)+(AX*0.00125);
VXant=VX;
%Calculem matrius
%Matriu de translació
Mtrans=[1 0 0 PX;0 1 0 PY;0 0 1 PZ;0 0 0 1];
%Matriu de rotació en eix X (Pitch)
MrotX=[1 0 0 0;0 cosd(yx) -sind(yx) 0;0 sind(yx) cosd(yx) 0;0 0 0 1];
%Matriu de rotació en eix Z (Yaw)
MrotZ=[cosd(yz) -sind(yz) 0 0;sind(yz) cosd(yz) 0 0;0 0 1 0;0 0 0 1];
%Matriu resultant
Mresult=Mant*Mtrans*MrotX*MrotZ;
Mant=Mresult;
nz=Mresult(3,1);
oz=Mresult(3,2);
az=Mresult(3,3);
%Guardem la matriu en un .txt i les posicions en una altre
save('matrius.txt','Mresult','-ascii','-double','-append')
posicionsX(i,1)=Mresult(1,4);
posicionsY(i,1)=Mresult(2,4);
posicionsZ(i,1)=Mresult(3,4);
velocitatsX(i,1)=VX;
velocitatsY(i,1)=VY;
velocitatsZ(i,1)=VZ;
anglesX(i,1)=yxtotal;
anglesZ(i,1)=yzttotal;
end
%Configurem gràfica
axis([min(posicionsX)-0.1,max(posicionsX)+0.1,min(posicionsY)-
0.1,max(posicionsY)+0.1,min(posicionsZ)-0.1,max(posicionsZ)+0.1])
xlabel('Eix X')
ylabel('Eix Y')
zlabel('Eix Z')
grid on
axis equal
hold on
%Dibuixem posició inicial
%Dibuixem els eixos de coordenades
nx=Minicial(1,1);
ny=Minicial(2,1);
nz=Minicial(3,1);
ox=Minicial(1,2);
oy=Minicial(2,2);
oz=Minicial(3,2);
ax=Minicial(1,3);
ay=Minicial(2,3);
az=Minicial(3,3);
PX=Minicial(1,4);
PY=Minicial(2,4);
PZ=Minicial(3,4);
line([PX PX+(nx*0.1)], [PY PY+(ny*0.1)], [PZ PZ+(nz*0.1)], 'Marker', '.', 'LineStyle', '-
', 'Color', 'r');
line([PX PX+(ox*0.1)], [PY PY+(oy*0.1)], [PZ PZ+(oz*0.1)], 'Marker', '.', 'LineStyle', '-
', 'Color', 'g');
line([PX PX+(ax*0.1)], [PY PY+(ay*0.1)], [PZ PZ+(az*0.1)], 'Marker', '.', 'LineStyle', '-
', 'Color', 'b');
text(PX+(nx*0.1)+0.01, PY+(ny*0.1)+0.01, PZ+(nz*0.1)+0.01, 'Xo')
text(PX+(ox*0.1)+0.01, PY+(oy*0.1)+0.01, PZ+(oz*0.1)+0.01, 'Yo')
text(PX+(ax*0.1)+0.01, PY+(ay*0.1)+0.01, PZ+(az*0.1)+0.01, 'Zo')
%Dibuixem el robot (fletxa amb orientació i sentit)
robotX=[-0.01*nx -0.01*nx+0.04*ox -0.02*nx+0.04*ox 0.07*ox 0.02*nx+0.04*ox 0.01*nx+0.04*ox
0.01*nx];
robotY=[-0.01*ny -0.01*ny+0.04*oy -0.02*ny+0.04*oy 0.07*oy 0.02*ny+0.04*oy 0.01*ny+0.04*oy
0.01*ny];

```



```

    robotZ=[-0.01*nz -0.01*nz+0.04*oz -0.02*nz+0.04*oz 0.07*oz 0.02*nz+0.04*oz 0.01*nz+0.04*oz
0.01*nz];
    patch(PX+robotX,PY+robotY,PZ+robotZ,'k');
%Dibuixem trajectòria del moviment
plot3(posicionsX,posicionsY,posicionsZ,'k');
%Dibuixem posició final
%Dibuixem els eixos de coordenades
nx=Mresult(1,1);
ny=Mresult(2,1);
nz=Mresult(3,1);
ox=Mresult(1,2);
oy=Mresult(2,2);
oz=Mresult(3,2);
ax=Mresult(1,3);
ay=Mresult(2,3);
az=Mresult(3,3);
PX=Mresult(1,4);
PY=Mresult(2,4);
PZ=Mresult(3,4);
line([PX PX+(nx*0.1)],[PY PY+(ny*0.1)],[PZ PZ+(nz*0.1)'],'Marker','.', 'LineStyle','-
','Color','r');
line([PX PX+(ox*0.1)],[PY PY+(oy*0.1)],[PZ PZ+(oz*0.1)'],'Marker','.', 'LineStyle','-
','Color','g');
line([PX PX+(ax*0.1)],[PY PY+(ay*0.1)],[PZ PZ+(az*0.1)'],'Marker','.', 'LineStyle','-
','Color','b');
text(PX+(nx*0.1)+0.01,PY+(ny*0.1)+0.01,PZ+(nz*0.1)+0.01,'X')
text(PX+(ox*0.1)+0.01,PY+(oy*0.1)+0.01,PZ+(oz*0.1)+0.01,'Y')
text(PX+(ax*0.1)+0.01,PY+(ay*0.1)+0.01,PZ+(az*0.1)+0.01,'Z')
%Dibuixem el robot (fletxa amb orientació i sentit)
robotX=[-0.01*nx -0.01*nx+0.04*ox -0.02*nx+0.04*ox 0.07*ox 0.02*nx+0.04*ox 0.01*nx+0.04*ox
0.01*nx];
robotY=[-0.01*ny -0.01*ny+0.04*oy -0.02*ny+0.04*oy 0.07*oy 0.02*ny+0.04*oy 0.01*ny+0.04*oy
0.01*ny];
robotZ=[-0.01*nz -0.01*nz+0.04*oz -0.02*nz+0.04*oz 0.07*oz 0.02*nz+0.04*oz 0.01*nz+0.04*oz
0.01*nz];
    patch(PX+robotX,PY+robotY,PZ+robotZ,'k');

```