

Desenvolupament d'una interfase sèrie de comunicacions per DSP

Caminal Villodres, G.

Enginyeria Tècnica Telecomunicacions esp. Sistemes Electrònics

Resum

En aquest article es vol presentar l'elecció i disseny d'una plataforma per implementar un panell d'operari o d'usuari que es comunica i visualitza les dades d'una DSP que controla un motor elèctric mitjançant una etapa de potència electrònica o l'equivalent que s'anomena inversor. L'objectiu principal és aconseguir que la plataforma escollida es comuniqui i visualitzi dades correctament amb l'inversor a més d'aplicar nous dispositius que obrin pas a la millora del mateix de forma modular. Per a fer-ho s'estudia el funcionament i característiques dels inversors comercials actuals, es comparen entre ells i s'intentarà preveure el futur d'aquests dispositius en el llarg termini. A partir d'aquí s'escollirà la plataforma més adient i senzilla d'aplicar i els mòduls a utilitzar. Per tal de comprovar el correcte funcionament, s'utilitzarà el software MatLab amb una connexió física Plataforma-PC i després es realitzaran les proves a la DSP directament.

I. Introducció

Els inversors, són màquines molt antigues, des del punt de vista funcional, però que sempre s'han utilitzat degut sobretot a la indústria manufacturera, on avui dia encara s'utilitzen, conjuntament amb noves tècniques i tecnologies, ja que qualsevol procés industrial, encara que no estigui gaire automatitzat, requereix d'un equipament que controli els motors elèctrics existents en aquests i d'altres tipus per obtenir un millor rendiment i eficiència. Avui dia, s'intenta que aquesta tasca de controlar motors sigui el més automàtica possible i per això s'utilitzen dispositius electrònics com les DSP's, que tenen una velocitat i capacitat de càlcul molt gran, que a més de controlar l'activació o no dels transistors de potència, poden dedicar-se a representar dades de forma gràfica, oferir diferents camins o busos de comunicació, càlculs complexes sobre les variables llegides o controlades per les mateixes, etc.

En el context de l'article, l'interessant és la interacció actual entre l'inversor i l'usuari no tècnic, ja que el tècnic pot connectar amb l'ordinador i utilitzant programari molt especialitzat, entendre i comunicar-se amb la DSP.

En aquest context, s'anomena al panell d'operari o d'usuari a aquell personal no tècnic que ha d'utilitzar aquest control, comunicar-se o simplement llegir dades per gestionar el dispositiu, ja sigui per tasques de producció,

rendiment energètic o d'altres. Per tant, s'entén que aquest panell, és aliè a la DSP i que no es requereix d'un coneixement molt especialitzat per utilitzar-lo, és a dir està optimitzat per al benefici de l'usuari.

Per tant i com s'ha comentat anteriorment, l'objectiu principal serà crear un panell d'usuari semblant al de la indústria però utilitzant noves tecnologies o dispositius que encaminin més el context cap a una millor automatització de tasques, aplicable també a altres camps

II. Estat actual dels Inversors

En el context actual, els inversors que alimenten i controlen motors elèctrics en corrent altern, són dispositius molt pesants, degut a tota la electrònica de potència que incorporen i normalment el dispositiu de control i maniobrabilitat es troba incorporat al mateix dispositiu formant tot un sistema o grup conjunt, on avui dia es pot connectar a un PC, aquí no es tindrà en compte ja que no és l'objectiu a consolidar, degut a que es necessita un dispositiu electrònic molt ràpid com són les DSP's; un aspecte nou i que es comença a veure en algunes marques, és la independència de la unitat de maniobrabilitat o control de l'usuari o operari



Figura 1: Exemple d'inversor típic comercial

Gràcies a l'aparició en els últims anys de noves tecnologies o grans millores tècniques, com per exemple: comunicació wifi, bluetooth, LCD's amb tecnologia TFT i pantalles tàctils, smartphones, etc. , es pot millorar molt el conjunt actual inversor-control d'usuari o operari i per tant amb aquest projecte, es començarà a aplicar aquest concepte utilitzant una plataforma amb microcontrolador idependent.

En els inversors actuals, el panell d'usuari es troba dintre de la mateixa caixa on s'ubica la DSP i la electrònica de potència, mitjançant, en gairebé tots els casos, un LCD de dues línies de text i els botons necessaris per tal de realitzar les accions d'operari més simples, tal com es pot apreciar a l'exemple d'inversor de la figura 1. Aquests panells són completament tancats en el context del firmware i comunicació, e incorporen tecnologia de fa més de 20 anys, ja que es pot entendre que les marques no veuen necessari realitzar una millora en tecnologia en un producte industrial.

III. Elecció i descripció de la plataforma

Per tal d'escollir correctament la plataforma, es necessita saber que es vol fer, i és visualitzar i comunicar amb la DSP de l'inversor, les dades o informació escollida. Aquesta plataforma ha d'estar basada en un microcontrolador, que sigui fàcil de programar, que es realitzi el mínim esforç en el tema hardware, per tant que sigui també un sistema el més modular possible i que el cost també sigui el menys elevat possible, sobretot per l'època econòmica que ens trobem. A la figura 2, es mostra el diagrama de blocs de la plataforma a escollir.

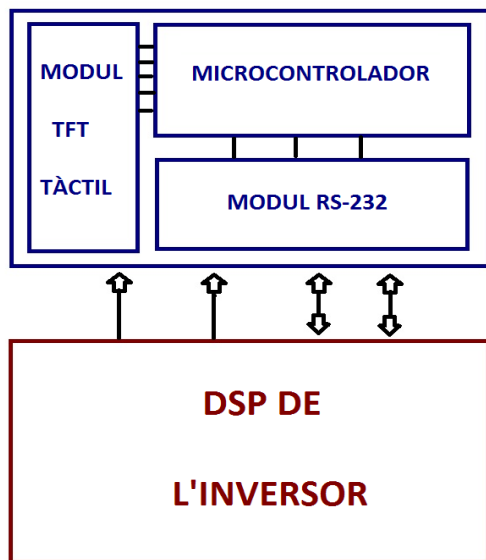


Figura 2: Diagrama de blocs de la plataforma.

el mòdul de visualització està format per un display LCD de tecnologia TFT tàctil i el mòdul de comunicació RS232 que intercanviarà informació amb la DSP i la plataforma.

L'elecció de la plataforma amb microcontrolador, es realitzarà amb Arduino model UNO. La raó principal

d'aquesta elecció ha estat el no haver d'escollir una plataforma tancada com per exemple els microcontroladors PIC on s'ha d'adquirir el programador per a la família de xip escollida, dissenyar e implementar el microcontrolador en una placa de circuit imprès adient i després buscar els diferents mòduls que tenen un cost més elevat i n'hi ha menys per escollir.

IV. Descripció de la placa Arduino UNO

En els últims anys s'ha posat molt de moda, està considerat el millor microcontrolador d'open hardware actual en el context del open source però referent a material electrònic; a més incorpora un entorn de programació anomenat IDE Arduino especialment dissenyat per aquest. Per tant, d'aquesta manera s'estalvia feina i diners en Hardware, ja que com es pot apreciar a la figura 3, bé montat sobre una placa de circuit imprès amb terminals accessibles.

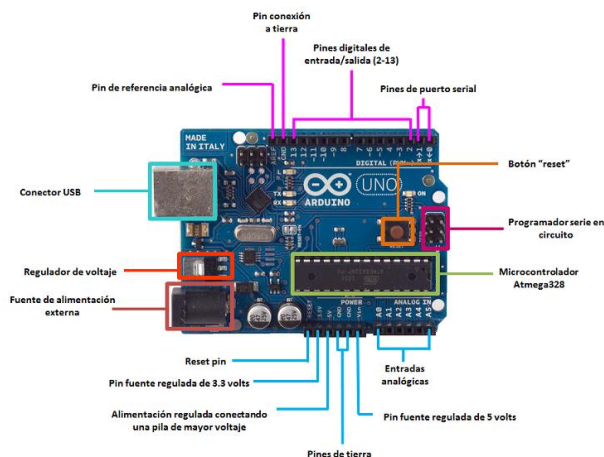


Figura 3: Placa Arduino Uno

A continuació es deixa un resum de les característiques d'aquesta plataforma:

Atmel model ATmega328P

- 14 terminals digitals d'entrada/sortida accessibles a la placa.
- 6 dels 14 terminals digitals anteriors es poden configurar com a sortides per PWM.
- 6 entrades analògiques amb un DAC de 10 bits cadascuna.
- Microcontrolador Atmel ATmega328P: 8 bits, 16 Mhz de rellotge, 32 KBytes de memòria flash, 1 KByte de ROM, 2 KBytes de RAM i arquitectura ARM.
- Programació mitjançant una connexió USB de tipus B accessible a placa.
- Alimentació directa per USB o mitjançant un connector d'alimentació accessible a la placa de 5 (v).
- Botó de reset accessible a la placa.

V. Descripció del mòdul RS-232

L'elecció del mòdul de comunicació RS-232 es senzill d'escollir perquè en realitat només ha de tindre un circuit integrat MAX3232 per convertir els nivells TTL de la USART disponible a la placa Arduino model UNO i els pins de connexió adients, tal com es pot veure a la figura 4, s'escull el mòdul RS-232 de l'empresa DF ROBOT, només utilitza quatre pins de la placa Arduino, el pin 0 de recepció RX, el pin 1 de transmissió TX, pin de massa o GND i el pin d'alimentació de 5 (v).

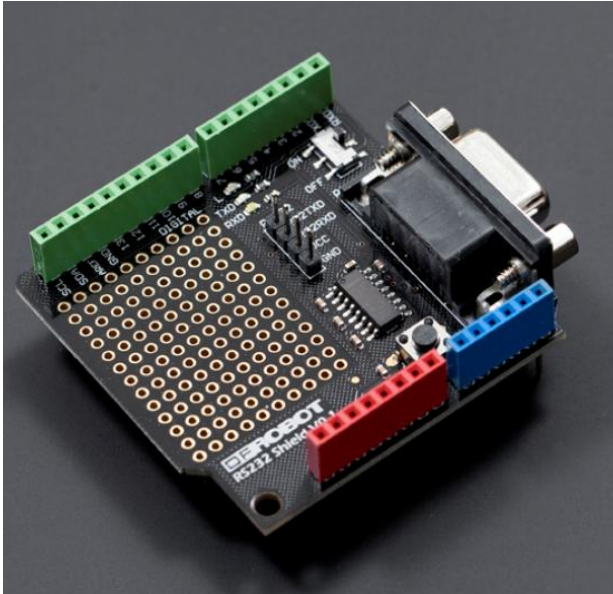


Figura 4: Mòdul RS-232 escollit

Com es pot apreciar a la figura, replica els mateixos terminals de la placa Arduino per tal de connectar un altre mòdul a sobre si s'escau i encaixa força bé.

VI. Descripció del mòdul TFT tàctil.

L'elecció del mòdul de visualització LCD de tipus TFT tàctil, ha estat l'aspecte més complicat, finalment es va decidir utilitzar una placa de l'empresa Xinesa mcufriend, tal que es mostra a la figura 5.

L'elecció d'aquesta pantalla es va realitzar buscant la millor eficiència possible, pel cost (2 o 3 cops inferior a proveïdors europeus) i per portar un driver de LCD conegut compatible amb Arduino, a més d'incorporar un panell tàctil en si mateix.

- Pantalla LCD gràfica TFT de 2,8" de 262000 colors.
- Xip controlador ILI9341 (segons fabricant) realment incorpora un controlador Sitronix ST7781.
- Resolució de 320x240 píxels.
- Panell tàctil resistiu de 4 fils.
- Slot per a targetes de tipus microSD amb un regulador de 3,3(v) a 2,7 (v). (No l'utilitzarem en aquest projecte).
- Connexió i comunicació en paral·lel compatible pin a pin amb Arduino UNO.



Figura 5: Mòdul TFT tàctil del fabricant mcufriend.

VII. Funcionament bàsic de la plataforma

La DSP permet connectar altres dispositius mitjançant una comunicació asíncrona de tipus Non Return to Zero o NRZ, això significa que els nivells lògics tipus TTL es transformen a nivells de tensió positius i negatius sense passar pel valor de 0 (v), amb aquests valors de tensió transformats, el senyal pot arribar molt lluny i és el format que veurem més endavant que compleix amb el RS-232

Per veure si l'Arduino és compatible amb aquest senyal, es realitza un programa senzill, figura 6, que envia el caràcter "a" pel port de sortida del mòdul RS-232, així mesurarem amb l'oscil·loscopi si és el mateix tipus de senyal que determina la DSP. A la figura 7 es mostra la captura de la mesura de l'oscil·loscopi.

A screenshot of the Arduino IDE 1.6.5 interface. The main window shows a program named 'LLETRA' with the following code:

```
//Programa envia caràcter A
//Per Genís Caminal

char lletra;
void setup() {
  Serial.begin(9600);
  lletra='A';
}

void loop() {
  Serial.print lletra;
}
```

The bottom status bar shows: 'Global variables use 183 bytes (8%) of dynamic memory, leaving 1.865 bytes for local variables. Maximum is 2.048 bytes.'

Figura 6: Programa senzill utilitzat per mesurar amb l'oscil·loscopi

Com es pot apreciar en la captura, el senyal de color verd o canal 2 de la placa Arduino, al començament es troba per defecte en nivell alt i després ens envia el bit de Start que és un zero, després ens envia sencer el byte, i per últim torna a nivell alt que indica el bit de Stop, la informació enviada te els següents valors: 10000010 que es el mateix que: 1000 0010, com representa el codi ASCII, els primers 4 bits són els més petits (el primer zero equival a 1 en decimal) i els altres quatre, els més grans, com és un byte, el codi va des de 0 fins 255 i si passem a decimal cada grup de 4 bits tenim que 1000 = 1 i 0010 = 64, sumats dona 65 que en el codi ASCII és el caràcter A, per tant el programa és correcte. Ara només cal veure que en el canal 1 o senyal groc de la captura, tenim el mateix resultat, però invertit, a la sortida del mòdul RS-232. Aquesta inversió resulta dels nivells de tensió RS-232 i NRZ on el 0 lògic val 9,6 (v) i el 1 lògic val -9,6 (v) sense passar en cap moment per 0 (v), on es veu clarament la discontinuïtat; per tant el tipus de senyal que surt del mòdul és completament compatible en nivell de tensions i format al port de la DSP.

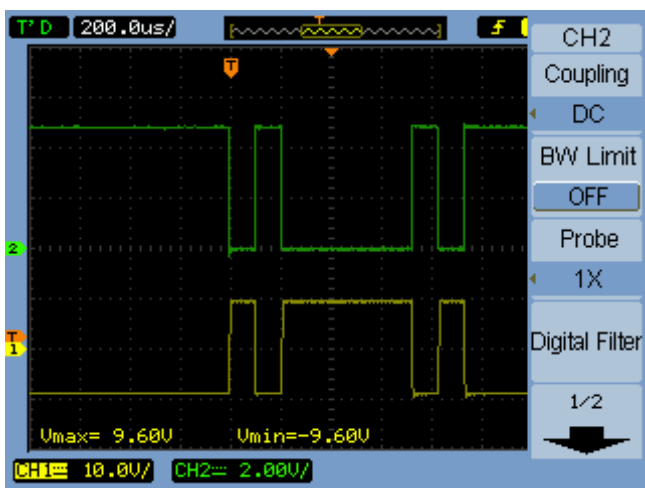


Figura 7: Captura de l'oscil·loscopi a la sortida del port d'Arduino.

Per tal de comprovar el funcionament, carreguem a l'Arduino un programa que si rep el caràcter "a", apagarà un led de la seva placa i l'Arduino respondrà amb els caràcters "OFF", i si rep b s'encendrà el led i respondrà amb els caràcters "ON". Aquest enviament i recepció de dades, es realitzarà mitjançant el programa MatLab que farà de DSP. El codi del programa de MatLab és:

```
PS
for i=1:10
fwrite (PS,'a','uchar');
k1=[];
for j=1:3
k1=[k1 fread(PS,1,'uchar')];
end;
k1=char(k1);
pause(1);
fwrite (PS,'b','uchar');
```

```
c1=[];
for z=1:2
c1=[c1 fread(PS,1,'uchar')];
end;
c1=char(c1);
pause(1);
end;
```

Com es pot comprovar a la figura 8, tenim el resultat vist des de MatLab.

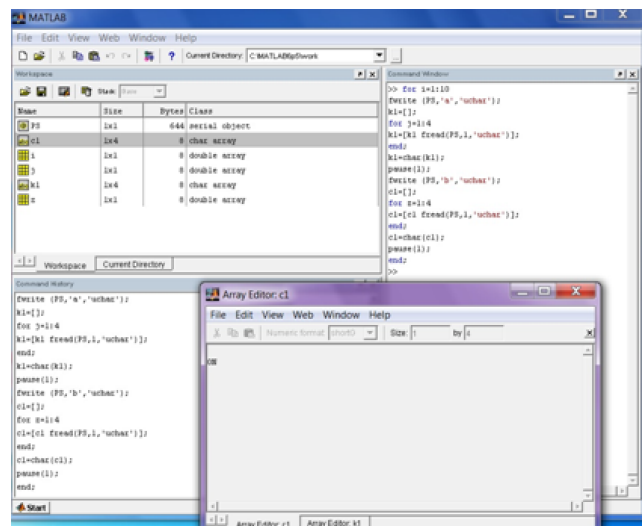


Figura 8: Resultat comunicació MatLab-mòdul RS-232 Arduino.

Com anteriorment, per entendre les funcions e instruccions per dibuixar a la pantalla LCD, ens ajudarem d'un exemple que funciona correctament, aquest exemple s'anomena graphicstest i es troba dins la llibreria SWTFT [ArDSP-1]. S'ometen els passos necessaris per inicialitzar la pantalla i el codi de l'exemple. Una de les funcions executades per aquest exemple es mostra a la figura 9.

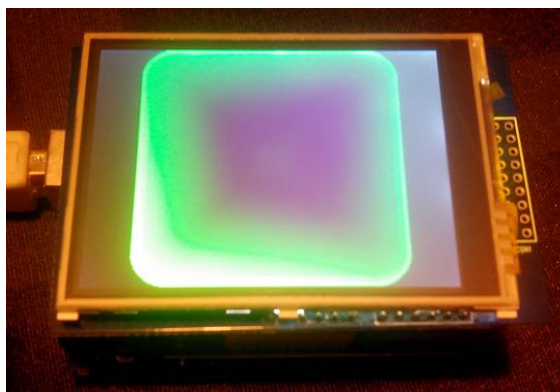


Figura 9: Funció executada per l'exemple esmentat

VIII. Programa Final

Al ésser un codi molt extens no es mostrarà sino que s'explicarà de forma resumida, però si algunes captures de demostració.

Aquí els dos mòduls estan connectats conjuntament a la placa Arduino i no s'ha apreciat cap tipus d'incompatibilitat. El codi del programa, a part d'inicialitzar correctament els dos mòduls, inicia amb la representació d'un logo on està escrit UPC amb lletres blaves i fons blanc. Aquesta representació dura dos segons d'execució per disseny del programador, i ho realitza mitjançant la funció upcLogo(), on s'ecriuen les funcions per dibuixar a la pantalla, el codi d'aquesta es mostra a continuació:

```
void upcLogo(){
tft.fillScreen(WHITE);
tft.setCursor(90, 80);
tft.setTextColor(BLUE);
tft.setRotation(3);
tft.setTextSize(9);
tft.print("UPC");
delay(2000);
```

Com es pot observar al codi, la funció no retorna cap valor, sino que només dibuixa. Les instruccions com per exemple tft.fillScreen(WHITE), ens dibuixen tota la pantalla de color blanc com es pot entendre pel seu nom.

Acte seguit, s'encamina a l'usuari fins una pantalla de selecció tàctil, on aquest haurà d'escollir, pressionant amb el dit, si vol llegir dades de la DSP o les vol escriure cap a la DSP. A la figura 10, es mostra la captura d'aquesta pantalla.

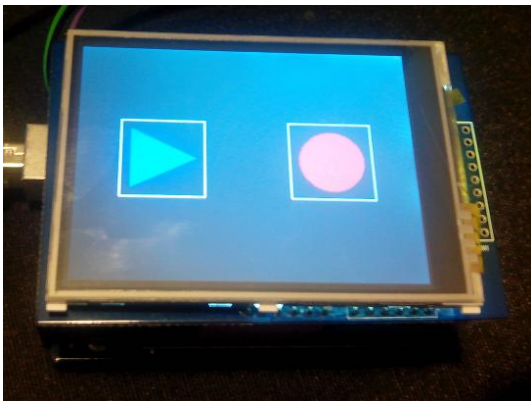


Figura 10: Pantalla de selecció tàctil, llegir triangle blau i escriure cercle vermell.

A continuació, es mostra el codi per a detectar la pressió en alguna d'aquestes dues icones:

```
for (int i=0 ; i<1000; i++){
digitalWrite(13, HIGH);
TSPoint p = ts.getPoint();
digitalWrite(13, LOW);
pinMode(XM, OUTPUT);
pinMode(YP, OUTPUT);
if (p.z > MINPRESSURE && p.z < MAXPRESSURE) {
p.x = map(p.x, TS_MINX, TS_MAXX, 0, tft.height());
p.y = map(p.y, TS_MINY, TS_MAXY, 0, tft.width());
if ((p.x >= tritochx1) && (p.x <= tritochx2) && (p.y >=
tritochy1) && (p.y <= tritochy2)) {
resultat=1;
delay (200);
}
if ((p.x >= cirtochx1) && (p.x <= cirtochx2) && (p.y >=
cirtochy1) && (p.y <= cirtochy2)){
resultat=0;
delay (200);
```

Com es pot apreciar en el codi inicialitzem la funció TSPoint p, que ens desarà a p les coordenades presionades (a p.x i a p.y), si aquestes coordenades es troben dins els marges de pressió i les coordenades del triangle, encendrà el motor i començarem a llegir o si es troben dins les coordenades del cercle només l'apagarem mitjançant la variable resultat que la funció retornarà al codi principal.

Després de la pantalla de selecció, si escollim llegir, l'Arduino començarà a llegir les dades i desar-les als arrays corresponents. La trama i protocol de comunicació que seguirà l'Arduino, es mostra a la figura 11.

Trama i Protocol entre DSP-Arduino

"r"
1 o 0
Espectura N° entre 0 i 4
Espectura N° entre 0-9
Espectura N° entre 0-9
Espectura N° entre 0-9
Lectura N° entre 0 i 4
Lectura N° entre 0 i 9
Lectura N° entre 0 i 9
Lectura N° entre 0 i 9
Lectura N° entre 0 i 5
Lectura N° entre 0 i 9
Lectura N° entre 0 i 6
Lectura N° entre 0 i 9
No Determinat-Extra
No Determinat-Extra
No Determinat-Extra
No Determinat-Extra
No Determinat-Extra
"@"

Figura 11: Trama i protocol de comunicació

Com es pot veure a la figura, cada posició o color horitzontal, equival a 1 byte, com tenim 20 posicions horitzontals, la trama ocupa un total de 20 Bytes. Aquesta s'executa a més de dalt a baix, primer ens trobem amb el caràcter "r" que es rep i es confirma pels dos extrems de la comunicació (Arduino-DSP), en la següent posició ens trobem el resultat de la pressió a la pantalla de selecció, comentat anteriorment. Després ens trobem 4 caràcters de tipus enter que estan reservats per a que l'Arduino escrigui la velocitat en revolucions per minut, quan seleccionem el cercle vermell (no implementat al codi final) i per tant s'atura el motor fins que no indiquem la velocitat i l'enviem a la DSP. Les quatre següents posicions de color verd, són idèntiques que les anteriors a diferència que és la dada que ens envia la DSP per llegir la velocitat provinent del motor. Després tenim quatre dades més de lectura de tipus enter, les dues primeres posicions de color groc són les dades del corrent que flueix pel motor i les dues següents de color blau són les dades de la fase de conducció dels transistors que alimenten el motor, indicat en graus. Les cinc següents posicions de la trama de color blanc, és un espai reservat buit per ampliar el codi de la comunicació en un futur si cal i crear noves funcionalitats, i per últim tenim el caràcter @ que s'envia i confirma per les dues parts i que finalitza la trama, per tornar a començar des de d'alt de forma indefinida, fins que s'acabi la comunicació o es pressioni un botó en forma de rectangle de color blanc per tornar a la pantalla de selecció. A la figura 12, es mostra el resultat de la pantalla TFT al llegir una trama.



Figura 12: Pantalla llegint dades.

El codi del programa també informa si no hi ha comunicació sèrie dient que no està operativa i aquí acaba el codi del programa.

IX. Millores que es poden implementar

Aquest dispositiu es pot millorar, realitzant una miniaturització, dissenyant en una mateixa placa de circuit imprès el xip de l'Arduino i el MAX232 que incorpora el mòdul de comunicació RS-232. També, es pot prescindir

del mòdul de comunicació i substituir-lo fàcilment per un altre que utilitzi comunicació sense fils com el Xbee per Arduino que realitza la comunicació mitjançant les senyals Wifi domèstiques i lliures segons la ETSI. I per últim, millorant el codi, per incloure gràfiques de rendiment del motor o d'altres representacions de varies variables a l'hora aprofitant el TFT amd 262K colors. Respecte l'elecció del mòdul de visualització, a estat un error adquirir-lo d'aquest fabricant degut al temps perdut intentant que funcionés de forma correcte, gràcies al poc o inexistent suport del mateix, i per tant no compensa l'estalvi de diners en la compra comparant amb el temps perdut d'optimització, per un projecte d'empresa aquest aspecte és inviable si es vol innovar i treure productes en poc temps a l'aparició de noves tècniques i tecnologies.

X. Conclusions.

Com s'ha pogut observar en aquest article, s'ha consolidat l'objectiu principal a mitges, ja que el codi del programa principal no està suficientment depurat i calen més hores de feina per tal d'aconseguir-ho, però les bases principals, d'escollir la plataforma i seguir unes línies de treball si s'ha aconseguit.

En l'aspecte personal, m'ha agradat molt participar en aquest projecte, malauradament no s'ha pogut millorar tot el que tenia en ment i que els dispositius poden realitzar a plena capacitat. Respecte aquests mòduls he après molt sobre comunicacions asíncrones i visualitzacions gràfiques i tàctils en pantalles LCD-TFT.

Per acabar, donar els meus agraïments, per la paciència que han tingut amb mi, al docent Balduí Blanqué i el futur doctor Marc Gomila.