Universitat de Vic
Escola Politècnica Superior

Treball Final de Carrera

Disseny d'un sistema de control del consum d'aigua automàtic

Albert Boix Comajuan

Enginyeria Tècnica Industrial.
Especialitat en Electrònica Industrial
Director: Dr. Pere Martí Puig
Vic, setembre de 2013

Vull dedicar aquest treball a la meva esposa, als meus pares i a la meva família, per no haver-los pogut prestar més atenció durant aquets anys que he estat formant-me.

La realització, entrega i presentació d'aquest treball, culmina un període de formació a la Universitat de Vic. Volia agrair a tots els professors que han sigut còmplices i part d'aquesta formació. No podia acabar la dedicació sense fer esment dels companys i amics d'estudi. A tots ells, moltes gràcies.

Resum

Títol: Disseny d'un sistema de Control del consum d'aigua automàtic.

Paraula clau: Arduino, Shields Arduino, SMS.

Autor: Albert Boix Comajuan.

Direcció: Dr. Pere Martí Puig

Data: Setembre de 2013

Avui en dia s'ha convertit en una necessitat tenir cura del medi ambient i optimitzar els recursos naturals. En el camp per estalviar energia s'han fet grans progressos i disposem d'un gran ventall de dispositius que ens ajuden i ens faciliten l'optimització del consum d'energia. Però és una realitat que en l'estalvi del consum d'aigua el progrés ha estat molt menor i es limita molt a donar consells i repartir dosificadors d'aigua. Qui no ha vist carrers o jardins o cases inundades amb milers de litres d'aigua?

Aquesta realitat m'ha portat a dissenyar i desenvolupar un prototip que em permeti tenir un millor control del consum d'aigua.

El prototip, a trets principals, consta d'un sensor, una electrovàlvula i una placa Arduino Atmega.

El sensor ens permet mesurar els litres consumits durant un cert període de temps. Passat aquest temps de mostreig es compara els litres consumits amb el consum habitual, en aquell període de temps. En cas de sobrepassar el volum programat es tancarà l'electrovàlvula de forma automàtica i rebrem un SMS al telèfon. L'activació de l'alarma es pot ajustar que sigui al igualar-se els dos valors, litres programats i litres consumits. També es pot programar el percentatge que cal sobrepassar de litres consumits per activar l'alarma, com el temps de mostreig. El fet de poder programar tots aquests valors ens permet fer un ajust ideal per a la instal·lació que es vol tenir controlada.

A més, el prototip es pot utilitzar per enviar a la companyia d'aigua el valor del comptador de forma automàtica. D'aquesta forma la companyia d'aigua també optimitza recursos estalviant-se el desplaçament de personal a la instal·lació per fer la lectura corresponent.

El prototip està basat amb un Arduino Atmega que ens permet el processament de les dades programades i capturades pel sensor. També s'ha incorporat una pantalla TFT Touch 2'8", que permet visualitzar i programar els valors d'una forma molt més intuïtiva. Per enviar els SMS s'utilitza una placa d'Arduino Cel·lular Shield - SM5100B, a la qual només cal afegir una targeta SIM.

A priori, el prototip té un elevat cost al fabricar una sola unitat i pot semblar poc útil. Però ens pot estalviar alguna sorpresa en les factures d'aigua si tenim una fuita i no ens n'adonem fins a veure el rebut de la companyia. Si es fabriqués a grans quantitats es podria abaratir el preu i fer-lo encara més engrescador.

Abstract

Title: Design of Control System Automatic Water Consumption

Keyword: Arduino, Arduino Shields, SMS.

Author: Albert Boix Comajuan.

Director: Dr. Pere Martí Puig

Data: September 2013

Nowadays, it is very important to care for the environment and optimize natural resources. There has been great progress in field of energy saving and we have variety of devices that help and facilitate optimization of energy consumption. But in regards to saving water the progress has been much slower, is limited to giving advice and distributing water dispensers. Who hasn't seen streets, gardens or houses flooded with thousands of gallons of water?

So I designed and developed a prototype that allows better control of water consumption. This prototype consists of a sensor, a solenoid and an Arduino Atmega plate.

This sensor measures the litres consumed over a period of time. After it compares the litres consumed regular water consumption. When overspending occurs, program closes the solenoid automatically and you receive a text message on your mobile phone. The alarm can be set for a programmed number of litters and litres used. Also it can be programmed with a percentage over the limit which will set off the alarm. By allowing for all of these programming options, a better fit can be made according to the installation.

Also the prototype can be used to automatically send the water counter value to the water company, which can also save resources normally used in sending a person to measure the water counter.

The prototype allows processing data programmed and captured by the sensor and it has TFT Touch 2'8" screen. This program lets you view and program values intuitively. To send messages to the phone an "Arduino Cell Shield - SM5100B" plate is used by simply adding a SIM card.

Currently, the prototype manufacturing has a high price, which may seem to make it of litres use, but the cost can be offset by the saving realized the case of an unknown leak, normally leading to a nasty surprise on the water bill it can save some surprises on the water bill. If manufactured in large quantities the price could be lowered, making it much more attractive.

Índex

1.Introducció i presentació del projecte 11

1.1.Motivació *11*

1.2.Objectius *12*

1.3.Metodologia *12*

1.4.Presentació global del prototip. *12*

2.Camps d'aplicació 13

2.1.Camps d'aplicació específic *13*

3.Disseny del sistema 14

3.1.Hardware *14*

3.1.1.Tipus de tecnologia 14

3.1.2.Per què Arduino? 16

3.2.Software *16*

4.Arduino 17

4.1.Introducció a l'Arduino *17*

4.2.Models Plaques Arduino *18*

4.2.1.Elecció placa Arduino 20

4.2.2.Obtenir una placa Arduino 20

4.3.Shields Arduino *21*

4.4.Mòduls connectar i llest *22*

4.5.Entorn de treball Arduino *24*

4.5.1.Programa i estructura. 25

5.Procés de fabricació del prototip 27

5.1.Entrades i sortides digitals. *30*

5.2.Rellotge DS1307 ..... *34*

5.3.Pantalla TFT Touch Shield V2.0 ..... *34*

5.4.SMS *39*

5.5.Comptador d'aigua de polsos i l'electrovàlvula. *43*

5.6.Ensamblatge. *43*

5.7.Resultat final. *47*

5.8.Cost del prototip *48*

6.Instal.lació de l'equip 49

6.1.Centraleta *50*

6.2.Comptador i electrovàlvula *51*

6.3.Connexions elèctriques *52*

7.Posar en marxa 53

7.1.Inicialització *53*

8.Conclusions i futures millores 58

9.Annexos 59

10.Bibliografia 188

Índex d'il·lustracions i taules.

1.1.Motivació 11

1.2.Objectius..... 12

1.3.Metodologia 12

1.4.Presentació global del prototip. 12

2.1.Camps d'aplicació específic 13

3.1.Hardware 14

3.1.1.Tipus de tecnologia 14

Il·lustració 3.1.2.2 Placa Pinguno 15

Il·lustració 3.1.2.1 placa Arduino Uno 15

3.1.2.Per què Arduino? 16

3.2.Software 16

4.1.Introducció a l'Arduino 17

Il·lustració 4.1.3 Diagrama de blocs duna Placa Arduino Uno 17

Il·lustració4.3 Diagrama de blocs placa Arduino Uno 17

4.2.Models Plaques Arduino 18

Taula 4.2.5 18

Taula 4.2.1 Comparacions principals models plaques Arduino. 19

4.2.1.Elecció placa Arduino. 20

4.2.2.Obtenir una placa Arduino 20

4.3.Shields Arduino 21

Il·lustració 4.3.4 ARDUINO SHIELD GSM-GPRS SPARKFUN AMB SM5100B. 21

4.4.Mòduls connectar i llest 22

Il·lustració 4.3.5 ARDUINO SHIELD TFT Touch Shield V2.0. 22

Taula 4.4.2 mòduls connecta i llest inclosos en el kit connectar i llest 23

4.5.Entorn de treball Arduino 24

Il·lustració 4.5.6 Exemple estructura programa 24

4.5.1.Programa i estructura. 25

Taula 4.5.3. Eines del programa Arduino. 25

Taula 5.4 components inclosos en el kit basic 2 Arduino 29

5.1.Entrades i sortides digitals. 30

Il·lustració 5.1.7 Esquema connexió entrades 30
Il·lustració 5.1.8 Esquema connexió sortida 30
Il·lustració 5.1.9 Exemple Led. 31
Il·lustració:5.1.10 Visualització litres per la pantalla LCD i simulació comptador amb polsadors. 32
Il·lustració:5.1.11 Utilització mòduls conectar i llest i polsadors. 33

5.2.Rellotge DS1307 34

5.3.Pantalla TFT Touch Shield V2.0 34

Il·lustració:5.2.12 Rellotge 34
Il·lustració:5.3.13 Pantalla Principal 35
Taula 5.3.5 Pantalles i Transicións programades. 36
Il·lustració:5.3.14 Pantalla principal. 37
Il·lustració:5.3.15 Xarxa de petri 38

5.4.SMS 39

Il·lustració:5.4.16 Proves per enviar SMS 39

5.5.Comptador d'aigua de polsos i l'electrovàlvula. 43

5.6.Ensamblatge. 43

Il·lustració:5.5.17 D'esquerra a dreta Comptador sensor de polsos. I les dues peces ensamblades. 43
Il·lustració:5.7.18 Caixa de protecció finestra i placa per muntatges. 44
Il·lustració:5.7.19 Primera linia D'esquerra a dreta: Connector jakc Alimentació, Porta fusibles interruptor basculant vermell 44
Segona linea d'esquerra a dreta Canaleta Unex, Borna, tapa borna, fre borna i guia Din. 44
Il·lustració:5.7.20 Esquema connexió elèctrica imatge pins TFT 46

5.7.Resultat final 47

Il·lustració:5.6.21 Centraleta 47

5.8.Cost del prototip 48

Taula 5.8.6 Taula Cost del material. 48
Il·lustració:6.22: 1 Centraleta, 2 Transformador Alimentació, 3 Comptador, 4 Sensor medidor Polsus HR-I. 5 juntes 7 Electrovàlvula NO 49

6.1.Centraleta 50

Il·lustració:6.1.23 imatge de la centraleta on es pot veure on i com es col·loca la targeta SIM. 50

6.2.Comptador i electrovàlvula 51

Il·lustració:6.2.24 Comptador i electrovàlvula. 51

6.3.Connexions elèctriques 52

Il·lustració:6.2.25 Sensor Sensus HRI-A1. 52
Taula 6.3.7. Esquema representatiu dels bornes 52

7.1.Inicialització 53
Il·lustració:7.1.26 d'esquerra a dreta pantalla inicialització, pantalla alineació i pantalla recordatori. 53
Il·lustració:7.1.27 Pantalla principal. 54
Il·lustració:7.1.28 Pantalla programació. 54
Il·lustració:7.1.29 Pantalla principal. 55
Il·lustració:7.1.30 D'esquerra a dreta Pantalla programació Dia, pantalla programació minuts i hora i pantalla programació telèfon 55
Il·lustració:7.1.31 Pantalla programació temps mostreig i tan per cent sobre consum. 56
Il·lustració:7.1.32 Pantalla obrir tancar electrovàlvula. 57
Il·lustració:7.1.33 Pantalla Alarma. 57

Índex annexos

1.Annexos Arduino Atmega ADK Android 59

1.2.Desglossament de l'esquema placa Arduino Atmega ADK ANDROID 65

1.3.Mostra de l'assignació de pins per al Atmega 2560 69

1.4.Taula d'assignació dels pins d'Arduino Mega 2560. 70

2.Arduino Shield TFT Tàctil color 2'8" 73

3.Arduino Shield GSM-GPRS Sparkfun SM5100B 76

3.1. SISTEMA DIAGRAMA DE BLOCS 76

3.2. DESCRIPCIÓ DEL PRODUCTE 77

3.3.Llistat de pins placa Shield GSM 77

3.4.Esquema placa Shield GSM 79

3.5.Esquema cablejat d'un teclat extern 80

3.6.Esquema cablejat targeta SIM 80

3.7.Esquema cablejat Altaveu. 82

3.8.Esquema cablejat RTC bateria 82

3.9.Esquema cablejat carregador de bateria 83

4.Programa enviat a l'Arduino Atmega ADK. 84

1.Introducció i presentació del projecte

En aquest primer capítol s'explica la motivació per realitzar aquest projecte, així com els objectius i la metodologia seguida per dur a terme el projecte.

1.1.Motivació

Avui en dia, ja no és una forma de vida sinó cada vegada més una necessitat i urgència el tenir cura del medi ambient. Això implica tractar els recursos naturals, cada vegada més escassos i més demanats, d'una forma més eficient i sense malbaratar-los.

En l'eficiència i estalvi d'energia elèctrica s'han fet molts progressos. Tothom tanca els llums, fem servir la llum natural quan és possible, augmenta la utilització de bombetes de baix consum, i la utilització de sensors de presència per evitar que els llums estiguin sempre encesos, també s'ajusta millor el termòstat, escollim electrodomèstics més eficients, etc.

Per altra banda, en el camp d'estalvi d'aigua, s'han distribuït dosificadors per les aixetes, es demana que la gent es dutxi en comptes de banyar-se, per tal de reduir el consum d'aigua. Però també és una realitat que tot sovint ens trobem una casa on les vàlvules del rec automàtic no han tancat i el jardí està completament inundat o bé per algun altre motiu han patit una fuita d'aigua. Quan se n'ha assabentat, ja hi ha una gran quantitat de litres malbaratats. Per aquest motiu crec necessari fer alguna cosa, per evitar dintre del possible, la pèrdua desmesurada de litres d'aigua de forma inútil. Aquest fet que em va portar a reflexionar en la necessitat de disposar d'un millor control del consum d'aigua. Per la qual cosa vaig començar a pensar en el disseny d'un sistema de control del consum d'aigua automàtic.

Gràcies a Internet, les xarxes socials i els SmartPhones, telèfons intel·ligents, estem més connectats amb les persones. El fet de l'evolució i reducció de costos d'aquesta tecnologia, afavoreix la seva ràpida implementació.

Ara es comença evolucionar en un altre sentit. La connexió tecnològica amb tot el nostre entorn, sorgint així l'Internet des les coses.

Un exemple que mostra l'evolució d'aquesta tecnologia i control són les SmartCities[1], un sector creixent i en plena expansió. Una altre aplicació és la connexió amb la nostra llar, vivendes intel·ligents, on podent tenir el control de molts aparells de la llar. També en aquest apartat, en el camp elèctric s'ha avançat molt però en el control del sistema hidràulic no s'ha quasi avançat, tot i tenir la tecnologia necessària per fer-ho. És evident que tenim poc coneixement i un quasi nul control sobre els litres d'aigua que consumin en una instal·lació hídrica. Un altre motiu de pes per dissenyar aquest projecte.

1 SmartCities o ciutats Intel·ligents són aquelles les quals a través de la tecnologia que tenen implementada es pot saber el carrer amb places d'aparcament lliure en el mòbil, quan estan plens els contenidors de brossa per poder-los buidar ...

1.2.Objectius

Els Objectius que es pretenen assolir en la realització d'aquest projecte són els següents:

- Dissenyar un sistema de control del consum d'aigua automàtic. D'aquesta forma es pot saber en tot moment el consum i reduir al màxim les possibles fuites en el sistema Hidràulic.
- Aprofitar la implementació de la Xarxa GSM per tenir a temps real i immediat l'alarma de problemes en el sistema Hídric que volem tenir controlat.
- Aprofitar el sistema per enviar la lectura del consum mensual a la nostre companyia d'aigua, estalviant temps a un operari el desplaçament fins a la instal·lació per fer la lectura del comptador.
- Com a objectiu final, es pretén la fabricació d'un prototip per poder observar possibles errors i millores del disseny.

1.3.Metodologia

El present treball final de carrera consisteix en les següents parts:

- En el capítol 3, s'estudien les possibilitats que tenim a l'hora de desenvolupar el prototip. En el mercat existeixen moltes possibilitats per poder realitzar prototips i fer un control d'un procés o realitzar una automatització.
- En el capítol 4, s'explica l'Arduino, la base del Hardware escollit. El motiu per haver-me decantat per aquest i les possibilitats que aquest ens ofereix.
- En el capítol 5, s'explica el procés de fabricació del prototip, així com els canvis que aquest ha anat patint durant el procés de fabricació. Al mateix temps s'expliquen totes les seves parts.
- A continuació, en el capítol 6, s'informa com s'han d'instal·lar totes les parts del prototip per poder-lo utilitzar.
- En el capítol 7 s'explica la posta en marxa i programació per part del usuari.
- I finalment, en el capítol 8, es reflexiona sobre les conclusions i futures millores.

1.4.Presentació global del prototip.

Per poder saber el consum d'aigua que tenim en un sistema hidràulic cal disposar d'una part mecànica o electromecànica que ens compti els litres d'aigua. Un cop s'han adquirit aquestes dades cal processar-les i analitzar-les i un cop analitzades cal uns actuadors per actuar en cas de ser necessari.

Queda força evident que el nostre sistema haurà de disposar d'una part de maquinari, compost per sensors, actuadors, com per exemple: comptador, electrovàlvula, alarma, a més, d'un sistema per poder interactuar amb el prototip.

Al mateix temps, és evident que per processar les dades ens caldrà un programari.

- El maquinari, conegut més popularment amb la seva paraula homòloga anglesa hardware. Com ja he dit és la part física del prototip i fa referència a totes les peces que el constitueixen com circuits, cables, sensors, actuadors, suports, caixes, etc.
- El programari, conegut més popularment amb la seva homòloga anglesa software. És la que fa referència al programa que permet al microprocessador realitzar una tasca. EL software utilitzat dependrà en gran part del hardware escollit. Són dues parts ben diferenciades però al mateix temps entrellaçades.

Un cop sabem els litres consumits cal analitzar les dades. Si aquestes estan en uns nivells estàndards no cal fer res. Per al contrari, si el nostre consum està per sobre del consum programat caldrà donar una alerta. D'aquesta forma es pot comprovar possibles problemes i al mateix temps tancar l'electrovàlvula per minimitzar les pèrdues d'aigua en cas d'avaria.

En el capítol 3 s'exposen les diferents possibilitats de hardware per poder desenvolupar la part central del prototip. De les diferents possibilitats m'he decantat per l'Arduino. El software de programació té el mateix nom i l'entorn de programació s'anomena Wiring, és molt semblant el llenguatge de programació C/C++.

2.Camps d'aplicació

Aquest sistema de control del consum d'aigua automàtic pot ser utilitzat en qualsevol sistema hidràulic existent. L'únic que caldria adaptar és el comptador en el cas d'instal·lar-se en una indústria on l'escomesa d'aigua fos d'un diàmetre superior a l'habitual, ja que aquest projecte s'ha desenvolupat per una escomesa estendard de ¾ de polzada.

2.1.Camps d'aplicació específic

El sistema de control del consum d'aigua automàtic pot ser instal·lat en qualsevol sistema hidràulic. Tal i com s'explica en l'apartat de motivació un dels llocs possibles d'aplicació és la vivenda habitual, però el fa molt més útil encara en les segones residències. Aquets habitatges, la major part del temps solen està deshabitats. Això implica que si sorgeix una avaria en el sistema hídric de la casa, el temps transcorregut des de l'aparició d'aquesta fins al tancament de la clau de pas serà molt elevat.

Tot sovint es poden veure parcs i jardins públics on el sistema de rec automàtic s'ha avariat i el parc i el seu entorn queda completament inundat, per això, aquest prototip pot ser també molt útil en els parcs i jardins on hi ha instal·lat uns sistema de rec.

En llocs públics sol ésser molt més gran el temps transcorregut des de l'avaria en el sistema hidràulic fins a tancar la clau de pas i reparar l'avaria. Per aquest motiu també és idoni per escoles, biblioteques, pavellons, poliesportius, etc. En definitiva és idoni per a qualsevol instal·lació pública.

A part dels lloc ja explicats, també és útil per les industries, bars i/o restaurants. Moltes indústries utilitzen l'aigua en els seus processos de fabricació, augmentant així el risc de tenir un problema hidràulic en la seva instal·lació del sistema d'aigua.

Només a tall d'exemple: en alguns models de glaçonera i sistemes de refrigeració s'utilitza un sistema de condensació que llencen l'aigua pel desguàs. En cas d'avaria d'aquesta vàlvula, si queda oberta o parcialment oberta es passar 24 hores evacuant aigua pel desguàs. Amb el greuge que ningú se n'assabentarà fins a l'arribada d'una factura d'aigua desorbitada. Així amb el meu projecte evitaríem tenir una sorpresa en el rebut de l'aigua.

Com ja he escrit en la introducció del capítol 2, el sistema pot servir per controlar el consum de qualsevol instal·lació hídrica.

3.Disseny del sistema

En aquest capítol s'expliquen les diferents possibilitats que tenim per desenvolupar el prototip i el motiu pel qual m'he decantat per un sistema i no un altre.

3.1.Hardware

Una de les parts importants del maquinari és el comptador, que ens permet comptar els litres d'aigua i ens proporciona el valor d'una forma que la podem tractar a posteriori. Per la qual cosa necessitarem un comptador electromecànic. De tots els comptadors que hem analitzat ens serviran els de Siemens WFU i Sensus Hr1.

L'altre part important de Hardware és la utilitzada per poder tractar els senyals del comptador i comunicar-nos amb l'usuari.

L'elecció d'aquest apartat influirà en l'apartat de software, ja que dependrà del tipus de tecnologia escollida per desenvolupar el prototip.

3.1.1.Tipus de tecnologia

Avui en dia existeixen diferents possibilitats de hardware per poder desenvolupar aquest projecte.

Una possibilitat consisteix en utilitzar un petit Autòmat programable. En el mercat existeixen una gran varietat de marques i models amb diferents prestacions. Com per exemple destaquem: Siemens, Omrom,Telemecanique, etc. Aquesta tecnologia està molt implementada per fer automatitzacions industrials, però com a gran inconvenient té el preu que ronda els 350€ els models més bàsics. Com ja he esmentat, la part de software va lligada al Hardware i dependria del model i marca d'Autòmat programable que escollíssim. A vegades, en els models més bàsics d'algunes marques es poden adquirir paquets molt econòmics que inclouen el software i l'autòmat, però si s'escullen per separat el preu s'incrementa.

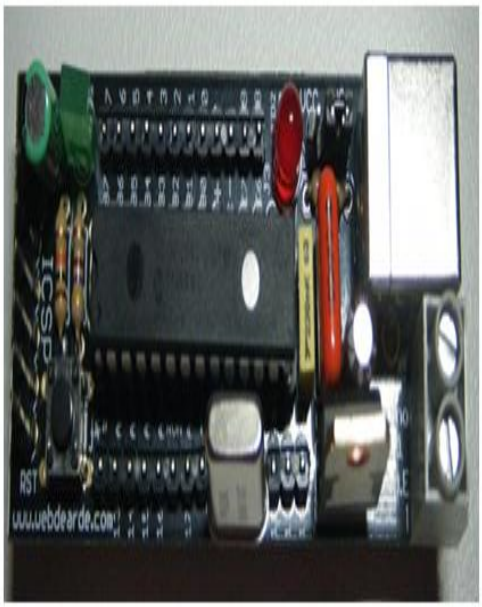
Una altra possibilitat és fabricar-ho amb Pics o amb Microcontroladors. Aquesta tecnologia és més econòmica, però s'ha de dissenyar el prototip des de zero, és a dir, dissenyar l'electrònica, l'alimentació, les entrades/sortides, etc.

Per últim podem optar per una placa on ja s'hi hagi implamentat el Pic o Microcontrolador. Aquestes ens proporcionen una major rapidesa i facilitat, degut a que ja porten incorporats entrades sortides ports USB o COM, per poder-los programar i evitar la necessitat d'adquirir un programador de Pics. La rapidesa que ens aporten aquestes plaques les converteix en un sistema molt atractiu per poder desenvolupar prototips electrònics.

En l'elecció de plaques electròniques per dissenyar prototips també podem trobar diversitat de marques i models. Una de les marques són les plaques Pinguino i una altre exemple les plaques Arduino. A continuació es pot veure a la il·lustració (3.1.2.1) una placa Arduino Uno on s'observa el Microcontrolador, el port USB l'alimentació i els pins per connectar les entrades/sortides. També es pot observar a la il·lustració (3.1.2.2) una placa Pinguino on també es veuen els pins d'entrades/sortides el botó de reset, etc.

Il·lustració 3.1.2.1 placa Arduino Uno

Font: http://www.electan.com/arduino-uno-con-atmega328-p-2977.html



Il·lustració 3.1.2.2 Placa Pinguno

Font:http://tallerarduino.wordpress.com/2011/06/10/mi-placa-pinguino-pic/

Una Placa Arduino, com per exemple l'Uno, és un circuit imprès que incorpora un Microcontrolador Atmega 328 un cristall de 16Mhz, ja disposa un port USB per poder programar i uns pins per poder accedir a les entrades/sortides, a més, incorpora un sistema per prevenir curtcircuits i un botó de resset.

La placa Pinguino és semblant a la plataforma Arduino, però basada en un PIC. Com a principal avantatge és que no necessita una interfase UART a USB addicional per comunicar-se amb el PC, ja que el Microcontrolador PIC18F2550 té el mòdul USB integrat. Això abarateix el Hardware deixant, a més, el port UART del Microcontrolador lliure per les aplicacions.

La seva principal desavantatge envers al seu competidor, Arduino, és la seva baixa popularitat. Això repercuteix amb el nombre de fòrums, tutorials, llibreries, exemples, disponibles per Internet i models de plaques amb característiques molt diverses.

3.1.2.Per què Arduino?

Un cop vistes les diferents possibilitats per poder desenvolupar el prototip calia escollir una de les possibilitats. La primera quedava descartada pel seu elevat cost inicial, encara que facilitava molt la feina i el programa. El fet de dissenyar la placa partint de zero incrementa molt el temps de fabricació, afegint un gran nombre d'hores de disseny i fabricació del prototip, per contra si es domina el tema i es produeixen pocs errors, el cost serà molt més reduït.

L'elecció per la placa vindria a ser l'opció intermèdia de les dues anteriors. No té el preu dels Autòmats programables i no es parteix des de zero com amb l'opció de dissenyar tota la placa electrònica.

3.2.Software

Tal i com he esmentat breument en l'apartat 1.4, el software ve predeterminat pel hardware escollit.

EL software de programació de les plaques Arduino, és un programa lliure que rep el mateix nom Arduino. Podem descarregar la versió per a Windows, Mac o Linux de forma totalment gratuïta des de la pàgina oficial d'Arduino: http://arduino.cc/en/

En aquest enllaç, a més, de descarregar el software de programació, la forma d'instal·lar-lo, trobarem molta informació de gran utilitat, com tutorials, exemples, tipus de models de plaques Arduino, etc.

Un cop descarregat i instal·lat el programa podem obrir l'entorn de treball per poder programar la nostre placa Arduino. Per fer-ho s'utilitza el llenguatge de programació Arduino (basat en el sistema Wiring), com ja he mencionat anteriorment en l'apartat 1.4 és un llenguatge de programació molt semblant al C/C++.

4.Arduino

En aquest capítol s'explica detalladament què significa Arduino. Les parts de les plaques, les diferents plaques que podem trobar al mercat, com aconseguir-les així com accessoris per Arduino com poden ser les plaques Shields i els mòduls connectari llest.

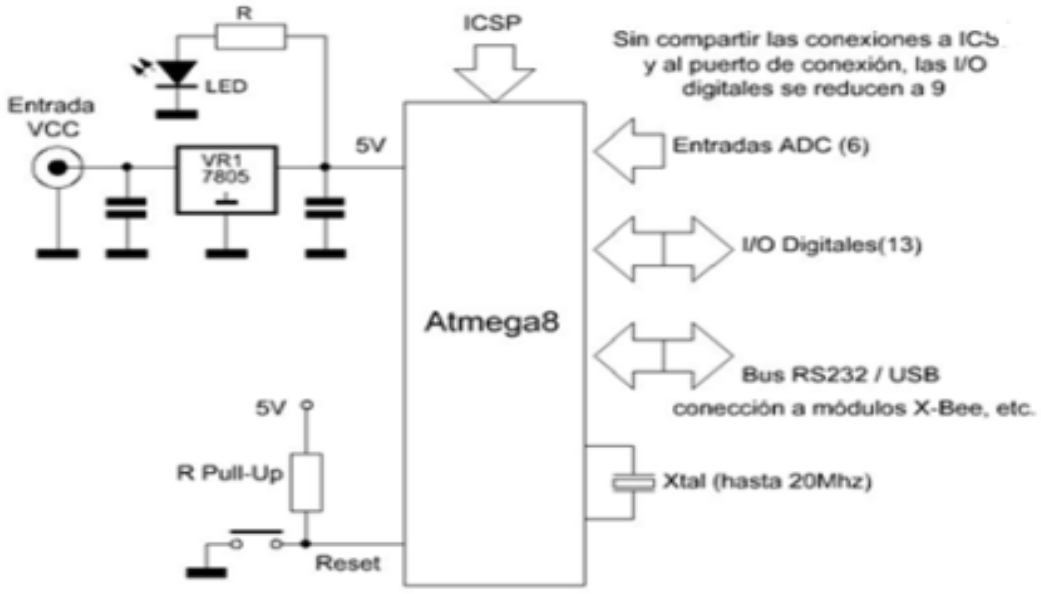
4.1.Introducció a l'Arduino

Arduino[2] és una placa de circuit imprès simple basada en Microcontrolador de codi obert provinent de la plataforma de codi obert Wiring amb l'objectiu de fer més simple i accessible el disseny de circuits electrònics amb Microcontroladors.

El maquinari consisteix en dissenys simples de maquinari lliure amb processadors Atmel AVR en una placa amb pins E/S. L'entorn de desenvolupament implementa el llenguatge Processing de *Wiring*, molt semblant a C++. Arduino es pot utilitzar per desenvolupar objectes interactius autònoms o pot ser connectat a programari de l'ordinador (per exemple: Macromedia Flash, Processing, Max/MSP, Pure Data). Les plaques es poden muntar a mà o adquirir els IDE de font oberta que es poden descarregar gratuïtament.

Arduino és un nom propi masculí italià que significa "gran amic".

La il·lustració (4.1.3), és un diagrama de blocs on es mostren els blocs que podem trobar en un circuit imprès d'una placa Arduino Uno. Entre ells l'entrada d'alimentació, el botó de resset, les entrades i sortides, el port de comunicació USB, etc.



Il·lustració 4.1.3 Diagrama de blocs duna Placa Arduino Uno

Font: http://www.neoteo.com/comparativa-arduino-arduino-vs-el-resto-15399

2 Explicació extreta de la Wikipedia en Català : http://ca.wikipedia.org/wiki/Arduino. versió 16 març 2013

4.2.Models Plaques Arduino

Com he esmentat amb anterioritat, Arduino és la plataforma més popular i això ha fet que en poc temps apareguin en el mercat una gran quantitat de plaques Arduino amb diferents característiques.

Una mostra de gran diversitat de plaques Arduino la trobem al final d'aquest apartat a la taula (4.2.1) on es pot contemplar la gran diversitat de models. A més, tenim les seves principals característiques.

El fet d'existir una gran varietat de plaques Arduino permet poder-ne escollir una que s'adapti millor a les necessitats del projecte que volem desenvolupar. Per aquest motiu, cal primer estudiar bé els requeriments tècnics del nostre projecte. Al tenir tants models per escollir ens obliga a uns coneixements previs de les necessitats del nostre prototip.

Una de les característiques importants és la quantitat d'entrades i sortides digitals analògiques. Les entrades i sortides digitals són les mateixes, l'únic que per programació es configura com entrada o com a sortida. Per exemple, si el prototip necessites utilitzar Ethernet ja disposem d'una placa que porti aquesta funció incorporada i ens estalvia haver d'afegir mòduls evitant connexions cablejats que poden fallar.

Si el que es desitja realitzar són prototips i fer proves de laboratori hi ha al mercat un Arduino que incorpora un protoboard per poder desenvolupar prototips de forma compacte.

És important escollir bé la nostra placa per no fer curt en entrades i sortides o per no pagar un preu molt més elevat per la placa i que aquesta disposi d'unes característiques que pagarem i no utilitzarem.

En algunes de les pàgines web on podem adquirir plaques Arduino, a vegades hi ha tutorials que ens expliquen les característiques de les plaques i ens ajuden a fer una elecció correcta segons les nostres necessitats tècniques.

A continuació, adjunto alguns enllaços on es comparen les diferents plaques Arduino:

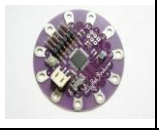
http://blog.bricogeek.com/noticias/arduino/como-comenzar-con-arduino---modelos-y-caracteristicas/#more

http://www.reflexiona.biz/shop/content/10-guia-de-compra-de-arduino

Un altre lloc per troba llocs on comparar plaques és la pàgina web oficial d'Arduino:

A més, de dificultar l'elecció de la placa Arduino cal anar en compte que en algunes webs tenen models en fase de des catalogació i si volem crear més d'un prototip podríem tenir problemes per adquirir una altre placa idèntica.

Tot seguit, trobem la taula (4.2.1), esmentada anteriorment, on es pot veure el de cada model, la imatge, el Microprocessador que incorpora cada model, característiques de rellotge de la CPU, el nombre d'entrades i sortides digitals, especificant quantes poden ser PWM, el número d'entrades analògiques, la memòria Flaix i SRAm, les dimensions i el preu aproximat.

MODEL	IMATGE	MICROPROCESSADOR	RELLOTGE CPU	E/S DIGITALS	ENTRADES ANALOGIQUES	FLAIX MEMORIA SRAM	DIMENSIONS mm	PREU APROX.
Arduino Pro mini 328		Atmega328	8 MHz	14(6 PROPORCIONEN PWM)	6	16Kb (2 son BootLoader) 1kbSRAM 512 Byte EEPROM	18X33	14,75€
Arduino Pro.		Atmega328	8 MHz	14 (6 PROPORCIONEN PWM)	6	32Kb (2 son BootLoader) 2kbSRAM 1Kb EEPROM	53.3x52.1	15,75€
Arduino Leonardo		ATmega32u4	16MHZ	20 (7 PROPORCIONEN PWM)	12	32Kb (4 son BootLoader) 2,5kbSRAM 1Kb EEPROM	75 x 53	19,70€
Arduino Micro		ATmega32u4	16MHZ	20 (7 PROPORCIONEN PWM)	12	32Kb (4 son BootLoader) 2,5kbSRAM 1Kb EEPROM	48X18	19,70€
Arduino Fio		Atmega328	8 MHz	14(6 PROPORCIONEN PWM)	8	32Kb (2 son BootLoader) 2kbSRAM 1Kb EEPROM	40.6x27.9	19,90€
Arduino Uno		Atmega328	16MHZ	14(6 PROPORCIONEN PWM)	6	32Kb (2 son BootLoader) 2kbSRAM 1Kb EEPROM	68.6X53.3	21,90€
Arduino Due		CPU de 32-bit Atmel	84MHZ	54(12 PROPORCIONEN PWM)	12	32Kb (2 son BootLoader) 2kbSRAM 1Kb EEPROM	52x29	39,00€
Arduino Ethernet			84MHZ	Com Arduino uno incorpora WizNet W5100 TCP/IP	6	32Kb (2 son BootLoader) 2kbSRAM 1Kb EEPROM	68.6X53.3	39,90€
Arduino Mega		ATmega 2560		53(4 HW UARTs, 14 PWMs, I2C bus)	16	32Kb (2 son BootLoader) 2kbSRAM 1Kb EEPROM	101,6x53.3	41,00€
Arduino Mega ADKte un USB per connectar amb l'Androide		"	"	"	"	"	"	59,00€
Arduino Lilypad *		Atmega328	8 MHz	14(6 PROPORCIONEN PWM)	6	16Kb (2 son BootLoader) 1kbSRAM 512 Byte EEPROM	50 diàmetre	16,95€

*l' Lilypad es pot rentar. Ideal per fer aplicacions sobre tèxtil.

Taula 4.2.1 Comparacions principals models plaques Arduino.

Font: http://www.electan.com/arduino-arduino-c-337_342.html http://www.bricogeek.com/shop/5-arduino?p=2

http://www.bricogeek.com/shop/5-arduino

4.2.1.Elecció placa Arduino

Com he esmentat en l'apartat anterior, un punt clau és el nombre d'entrades/sortides que necessitem per desenvolupar el nostre prototip.

En el meu prototip es necessiten 13 entrades per a la connexió de la TFTTouch de 2'8" polzades, una per a l'alarma, una per al sensor i 2 mínim per a l'electrovàlvula i una per l'interuptor de la pantalla. És evident que necessito una placa Arduino de més de 18 entrades. Per tan podria utilitzar un Arduino Leonardo, l' Micro, l' Atmega o l' ATmega ADK.

Finalment, m'he decidit per l'últim model, ja que vaig trobar molt interessant poder-lo connectar amb l'Android. Aquesta funció no la utilitzo en aquest projecte en concret. Però sí considero aquesta funció molt interessant i és un camp que desitjo aprofundir després de fer el projecte. D'aquesta forma ja disposo de la placa.

Si desitges instal·lar el prototip en algun lloc ja hi sóc a temps a comprar un model de placa una mica més econòmica.

4.2.2.Obtenir una placa Arduino

Avui en dia existeixen moltes botigues d'electrònica que disposen de material Arduino, els Shields i mòduls a connectar i llest, els qual en parlarem més endavant. Una altra forma d'adquirir material Arduino és a través d'Internet com per exemple:

- Electan, o Bricogeek, d'on s'ha extret informació de models i preus d' Arduino.
- http://www.electan.com/index.php
- http://www.bricogeek.com/shop/

Aquesta segona Web té uns ports més cars que l'anterior, però disposa d'un blog molt interessant. No és tan sols d'Arduino sinó d'Electrònica en general amb tutories i reportatges molt interessants.

Comprar a través de portals d'Internet té l'inconvenient d'haver de pagar ports per rebre el producte. Com a contrapartida tenim ja informació de molts webs on sovint podem trobar: Bolgs, projectes, multitud d'exemples diferents, llibreries, per la simplificació de la programació, fòrums o podem preguntar dubtes i trobar usuaris amb problemes semblants als nostres, i on a vegades, ens pot ser de gran ajuda per trobar solucions als nostres problemes.

Parlant de Webs, no podia deixar d'esmentar la web oficial d'Arduino. http://www.arduino.cc/

Aquesta web és de gran ajuda per informar-nos de les diferències de models. Esquemes, Datasheet dels Microcontroladors, llocs on poder adquirir material Arduino, un fòrum, tallers Arduino, etc.

En definitiva una web molt útil, per no dir imprescindible, si utilitzem l'Arduino.

El fet de què Arduino sigui una plataforma oberta, implica que si es desitja des de la mateixa web es pot descarregar els esquemes de la placa per poder-la fabricar.

4.3.Shields Arduino

Les Shields Arduino són plaques que poden ser connectades directament sobre, d'aquesta forma augmentant les seves caracterísitques.

Al mercat existeixen diverses plaques Shields que ens poden facilitar molt la feina a l'hora de desenvolupar un prototip. Algunes de les plaques que podem trobar el mercat són: plaques per afegir targetes SD, Relés, displays, Teclats, controladors per motors, etc.

Les plaques Shields són de gran utilitat per afegir prestacions al nostre Arduino d'una forma molt més ràpida, eficaç i fiable. El fet d'anar connectada directament sobre les plaques Arduino evita que possibles cables puguin fer fallar el prototip i perdre un munt d'hores per un fil que fa mal contacte.

Tot seguit, podem contemplar les dues plaques Shields utilitzades en el meu prototip. La primera il·lustració (4.3.4) és una placa Arduino Shield GSM-GPRS SM5100B, que entre d'altres funcions permet enviar i rebre els SMS.

Il·lustració 4.3.4 ARDUINO SHIELD GSM-GPRS SPARKFUN AMB SM5100B.

Font: http://www.bricogeek.com/shop/arduino/286-arduino-cellular-shield-sm5100b.html

L'altre placa Shield utilitzada per desenvolupar el prototip il·lustració (4.3.5) és una TFT Touch Shield V2.0, que permet una millor interacció entre el prototip i l'usuari. A la il·lustració de l'esquerre es pot veure els pins que s'utilitza per la connexió amb l'Arduino.

Il·lustració 4.3.5 ARDUINO SHIELD TFT Touch Shield V2.0.

Font: http://www.bricogeek.com/shop/arduino/286-arduino-cellular-shield-sm5100b.html

4.4.Mòduls connectar i llest

Els mòduls connectar i llest són plaques més petites que les plaques Shields, però a diferència d'aquestes últimes són únicament sensors o actuadors. Aquets mòduls es caracteritzen per portar un connector soldat de forma que la seva connexió amb l'Arduino és molt ràpida.

A més, del connector esmentat també porten la resistència incorporada, d'aquesta forma la senyal del sensor ja està condicionada per connectar amb l'Arduino. D'aquesta forma si tenim mòduls connectar i llest, una placa Shield connectar i llest i els cables, podem utilitzar aquests sensors de forma molt ràpida.

Aquets mòduls es poden adquirir de forma individual si en necessitem un en concret o també tenim l'opció d'adquirir un kit que incorpora els cables, placa Shields i diferents mòduls connectar i llest. Aquesta opció és ideal si tenim intenció de fer diferents prototips amb l'Arduino, ja que surt més econòmic que comprar-los per separat.

Seguidament, podem observar la taula (4.4.2) on es mostra la imatge i nom dels mòduls connectar i llest que incorpora aquest kit.

Imatge	Descripció
	Arduino UNO amb ATMega328
	Arduino Shield Connectar i llest

	Mòdul Boto Connectar i llest
	Mòdul Led RGB Connectar i llest
	Mòdul Sensor Inclinació
	Mòdul Zumbador.
	Mòdul Potenciòmetre (Analògic)
	Mòdul Sensor temperatura.
	Mòdul Relè.
	Mòdul Placa prototips.
	Mòdul LCD 16*2 Caràcters.
	Cable 4 conductores. 10 unitats
	Adaptador Pila 9V.

Taula 4.4.2 mòduls connecta i llest inclosos en el kit connectar i llest

Font: http://www.electan.com/arduino-shield-kit-modulos-conectar-listo-p-2987.html

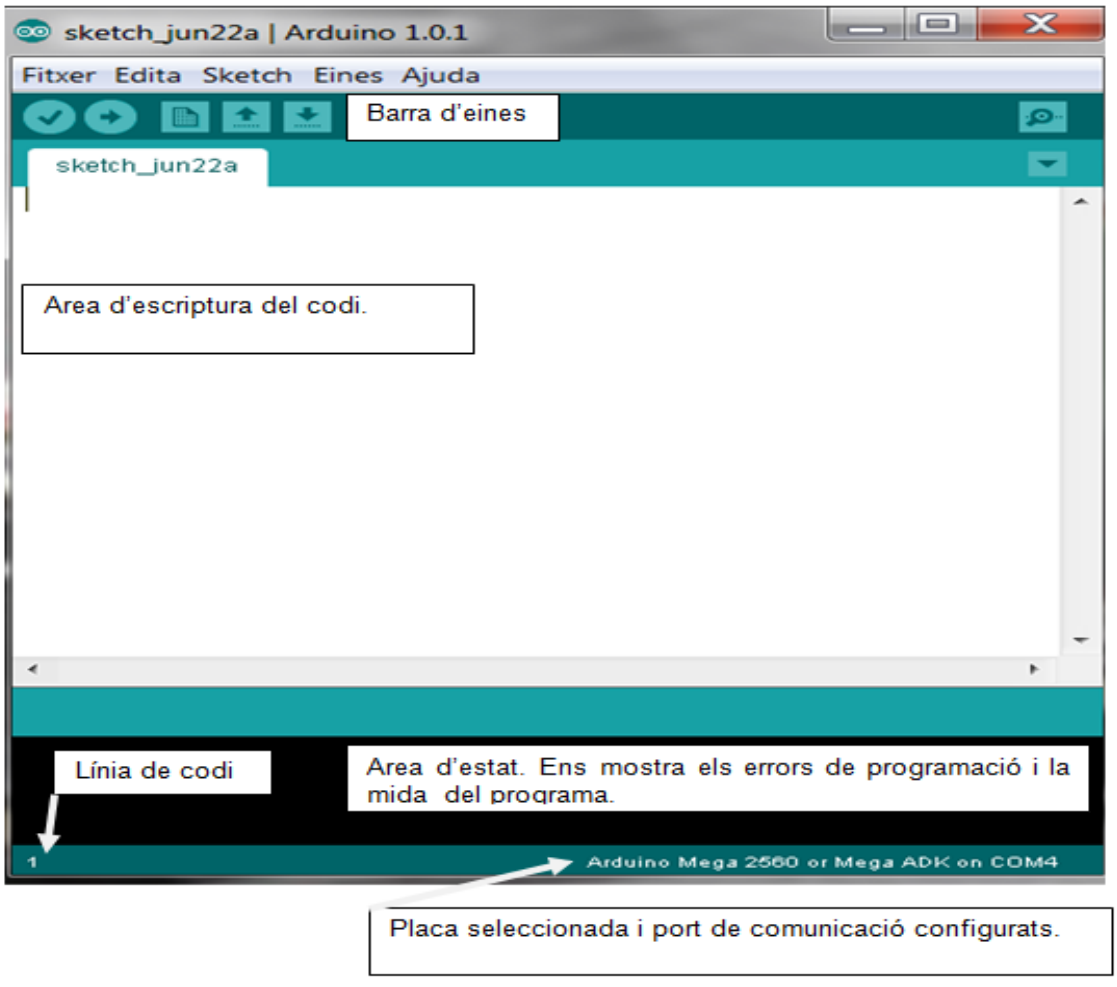
Per utilitzar aquest mòduls cal disposar de la placa Shield connectar i llest. Aquesta placa té dues versions una per al model Arduino Uno i per a l'Atmega.

4.5.Entorn de treball Arduino

A continuació, és pot veure la il·lustració (4.5.6) de l'entorn de programació Arduino. Aquest entorn de treball és el que s'utilitza per editar el programa que volem enviar al Microcontrolador del Arduino. També per compilar i comprovar possibles errors. Al mateix temps podem obrir el port sèrie i observar valors del Arduino així com enviar i rebre dades.

Abans de començar a fer el programa cal haver seleccionat el model de placa Arduino que estem utilitzant i el mode de comunicació que s'utilitza.

Les plaques poden estar connectades el PC, però també poden funcionar de forma independents i autònomes.



Il·lustració 4.5.6 Exemple estructura programa

A continuació, a la taula (4.5.3) és pot observar una imatge i una breu explicació del botons que podem trobar a la barra d'eines de l'entorn de treball d'Arduino.

	S'utilitza per compilar el nostre programa i saber si hi ha errors abans d'enviar-lo a la placa Arduino.
	S'utilitza per enviar el programa a la placa a traves del port USB.
	S'utilitza per crear un nou programa.
	S'utilitza per visualitzar tots els programes guardats i poder seleccionar-ne un per poder-lo obrir.
	S'utilitza per poder guardar el programa.
	S'utilitza per poder obrir una finestra nova per motoritzar el bus que està connectat a la placa Arduino USB o RS323. Així com per enviar i rebre dades de la placa.

Taula 4.5.3. Eines del programa Arduino.

4.5.1.Programa i estructura.

En primer lloc, es criden les llibreries que utilitzarà el projecte, com poden ser les de configuració del i2c, les de la pantalla TFT i així totes les que el programa utilitzi. Per cridar una llibreria es fa de la següent forma: amb coixinet include i el nom de la llibreria entre signes de <> i amb l'extensió. Per exemple: #include<nomllibreria.h>

El programa Arduino ja porta algunes llibreries incorporades. A més, la última versió les porta quasi totes les llibreries que podem necessitar. Per saber més sobre el que realitza cada llibreria i com poder-ne afegir-ne de noves adjunto un enllaç de la pàgina Arduino molt útil.

http://arduino.cc/en/Tutorial/HomePage

Cal anar en compte a l'hora d'utilitzar les llibreries. Si són d'una placa Shield cal comprovar que sigui adequada per aquell model i versió, ja que si no podem tenir molts problemes i perdre molt de temps per solucionar-los.

Un cop incorporades les llibreries s'han de definir les variables i constants que desitgem utilitzar en el nostre programa. Primer s'indica el tipus, separat per un espai el nom, després un igual i finalment el valor inicial d'aquesta variable. L'exemple de declaració d'una variable seria el següent: int dia = 1;

Les variables poden ser definides de la següent forma:

Byte, té un valor numèric de 8 bits sense decimal. Té un rang de 0 a 255.

Int, té un valor numèric de 16 bits sense decimal. Té un rang de -32767 a 32768.

Long, té un valor numèric de 32 bits sense decimal. Té un rang de -2147483648 a 2147483647.

Float, té un valor numèric de 32 bits amb decimal. Té un rang de -3.4028235E+38 a 3.4028235E+38.

Arrays és una cadena de caràcters, amb el que tan sols amb un índex es pot accedir a qualsevol valor de la cadena. Una array de defineix de la següent forma: int miArray[3]= ´{valor0,valor1,valo2...}. El tres indica la longitud de la cadena.

Després de definir les variables utilitzades s'ha de definir els pins utilitzats per l'. Exemple:

Les variables poden canviar el seu valor mitjançant el programa. Les constants, com el seu mateix nom indica sempre tindran el mateix valor. Una de les constants són les definicions dels pins utilitzats.

```
const int buttonPin = 2;     //  El pin 2 se li assigna el nom de buttonPin

const int ledPin =  13;      //El pin 13 se li assigna el nom  de ledPin.
```

Dins el programa Arduino per poder escriure comentaris i explicacions per tal de que no doni errors i diferenciar-lo del programa purament aquests han d'anar precedits de //. Davant de cada fila. L'altre opció es col·locar el comentari entre barres i asterisc de la següent forma. /*comentari */

Seguit de la declaració de variables i constants s'escriuen les funcions que volem utilitzar en el nostre programa principal. Utilitzar funcions no és imprescindible, però és molt útil per programes llargs i així poder-lo dividir en subprogrames. També és molt útil per parts del programa que s'executen de forma repetitiva i evitar haver d'escriure les instruccions més cops i d'aquesta forma simplificar el programa.

La utilització de funcions es realitzaria de la següent manera:

```
void nom de la funció()
        {
        //Accions que volem realitzar dins d'aquesta funció
        }
```

Un cop creada la funció aquesta ja està a punt per ser utilitzada dins el menú void loop(). Per ser utilitzada només hem de fer la cridar de la funció escrivint el seu nom. Un cop fem la crida del programa escrit a void loop() es deixa d'executar i es comença executar el programa escrit dins la funció cridada. Un cop s'ha finalitzat l'execució d'aquesta, el programa es continua executant des del punt en què s'havia parat perquè s'havia demanat de fer la crida de la funció.

No tots els programes tenen funcions creades, només s'utilitza en funcions que es repeteixen de forma reiterada o per realitzar un programa molt llarg i dividir-lo en diferents parts. El que sí ha de tenir tot programa Arduino és la part del Void Setup() i el Void Loop().

La primera, només s'executa en el moment que l'Arduino rep alimentació per primer cop o després d'un resset. S'utilitza per fer configuracions com per exemple del port sèrie i dir si els pins definits a les variables són entrades o sortides, etc.

```
Void setup()
        {
        // inicialització Led com a sortida
        pinMode(ledPin, OUTPUT);
        // inicialització del pinmode com entrada.
        pinMode(buttonPin, INPUT);
       }
```

Seguidament del Void Setup() trobem el Void loop() i és on col·locarem el nostre programa principal. A diferència del setup que només s'executa al rebre alimentació la placa Arduino o després d'un resset aquest bucle es repeteix de forma infinita. Un cop el programa que s'està executant arriba al seu final torna a començar a principi del loop.

```
Void loop()
       {
       //inici i Cos principal del programa;
       //Cridem la nostre funció
       //Continua el programa després d'executar la funció
       //Torna al inici del loop i torna executar aquesta part de programa.
       //Sempre i quan no es retiri l'alimentació del Arduino o es realitzi un reset
       }
```

El mateix enllaç que es pot utilitzar per informar-se sobre la referència del llenguatge de programació. On podem trobar estructures que controlin la sintaxis d'aquesta. Com ja he esmentat la programació d' utilitza una estructura i llenguatge semblant al C C++.

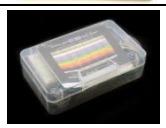
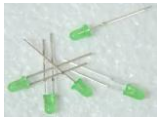
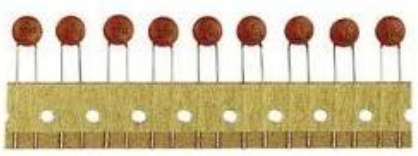
5.Procés de fabricació del prototip

En aquest capítol explico tot el procés de fabricació del prototip.

Un cop avaluades les possibilitats per fabricar el prototip em vaig decantar per l'opció de la placa Arduino.

Al començar a mirar preus de les diferents plaques Arduino vaig veure l'opció d'adquirir uns pacs, els quals a més d'incorporar la placa incorporen material divers, molt útil per començar a desenvolupar prototips. De pacs hi ha diferents versions i preus. Jo em vaig decantar pel més

complert. Aquest paquet incorpora els mòduls connectar i llest del capítol 4.4, taula (4.4.2) i l'altre inclou els elements de la taula (5.4) .

Imatge	Descripció
	1 Mòdul Board
	50 Cables Jumper
	1 Mini Servo
	1 Zumbador
	1 Potenciòmetre
	1 Fulla de valors de les resistències
	1 Caixa de plàstic
	1 Led RGB
	10 Leds vermell
	10 Leds verd
	10 Condensadores Ceràmics 10nF

	10 Condensadores Ceràmics 100nF
	5 Condensadores Electrolítics 100uF
	10 Resistències 330 ohm
	10 Resistències 1k ohm
	10 Resistències 10k ohm
	1 Sensor de inclinació
	1 Termistor
	1 Fotoresistència
	1 Diode
	5 Polsadors
	5 Interruptors

Taula 5.4 components inclosos en el kit basic 2 Arduino

Font: http://www.electan.com/arduino-shield-kit-modulos-conectar-listo-p-2987.html

El meu prototip ha anat evolucionant i millorant en forma i prestacions a mida que els meus coneixements del món del Arduino anaven en augment.

Un cop em van arribar tot el material era hora de començar a fer les primeres passes amb l'Arduino Uno fent un programa senzill, per aprendre a programar-lo i a connectar-lo.

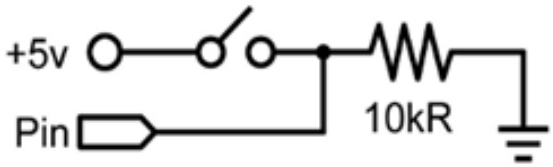
5.1.Entrades i sortides digitals.

El primer pas va ser fer encendre i apagar un Diode Led mitjançant un polsador. Aquest senzill programa apareix al principi dels tutorials d'Arduino i a més, està inclòs en el software de programació d'Arduino a l'apartat d'exemples. En aquest podem trobar-ne de molts temes els quals ens poden ser de gran ajuda abans de començar a desenvolupar el nostre projecte.

La part de programació es limita a obrir l'exemple, compilar-lo i enviar-lo a la placa. No té cap complicació però és de gran utilitat per entendre el funcionament i poder realitzar els primers passos amb l'Arduino. A més, és útil per saber si hem connectat bé les entrades i sortides.

Per connectar el Led i el polsador tenim dues opcions. La primera és utilitzar els mòduls connectar i llest amb la placa Shield Connectar i llest. Tal i com fa referència el seu nom es simplement connectar-los a l'entrada i sortida que haguem definit per programa i ja funcionarà.

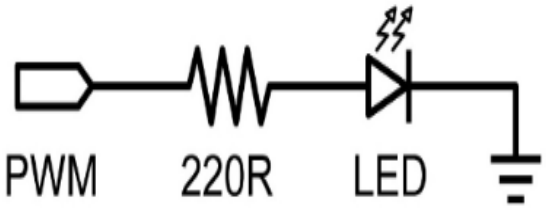
La segona opció és l'utilització d'un polsodor sense el mòdul, en aquest cas hem de posar una resistència de 10kΩ. Per tant en comptes de tenir el circuit fet amb un mini circuit imprès com són els mòduls conectar i llest, haurem de fer totes les connexions com es mostra en la il·lustració (5. 1.7).



Il·lustració 5.1.7 Esquema connexió entrades

La resistència de 10KΩ és negre, marró, taronja.

Per la connexió de les sortides hem d'utilitzar una resistència de 220Ω i realitzar el connexió tal i com es mostra a la il·lustració(5.1.8).



Il·lustració 5.1.8 Esquema connexió sortida

La resistència de 220Ω correspon als colors vermell, vermell i negre.

Tot seguit, a la il·lustració (5.1.9) es pot veure l'exemple per encendre el led.

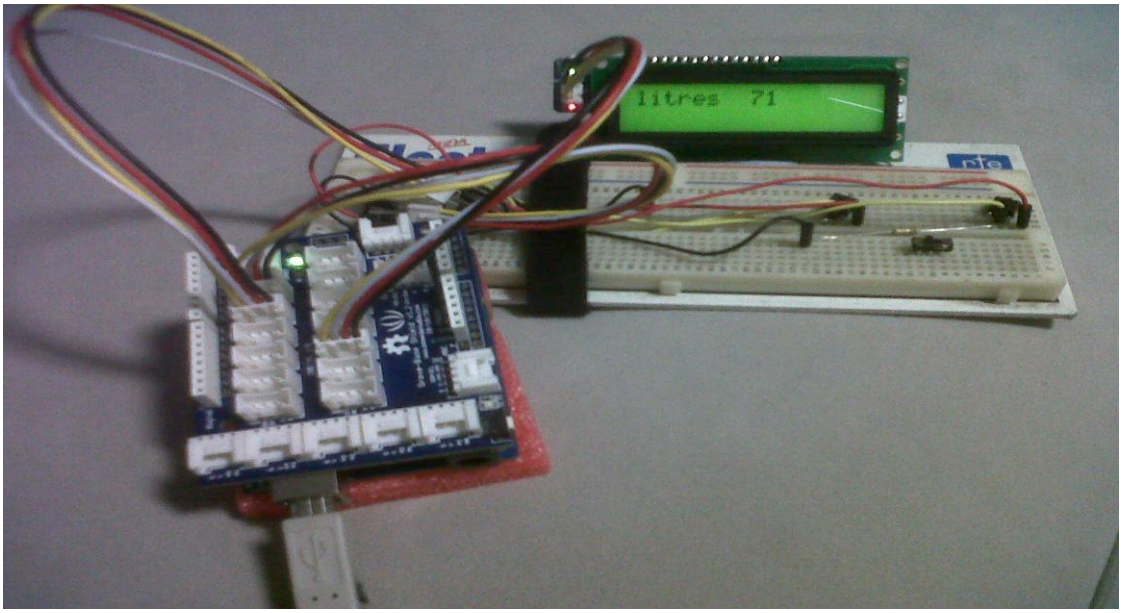
Il·lustració 5.1.9 Exemple Led.

El meu primer prototip era molt més senzill, a mida que he anat ampliant coneixements i m'he anat introduint en el món Arduino el meu projecte ha crescut amb prestacions de forma paral·lela.

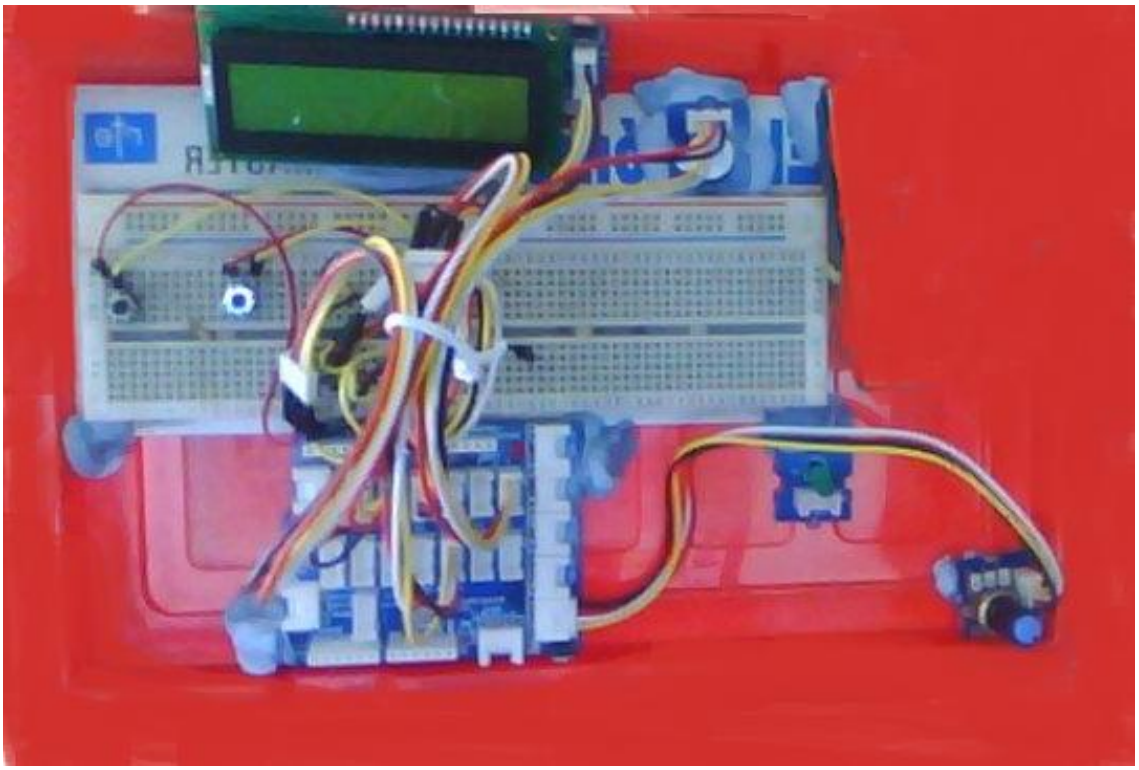
En la realització del meu primer prototip utilitzava una entrada pel comptador de litres, la qual en un primer moment vaig simular amb un polsador. Un polsador per realitzar un resset quan s'ha disparat l'alarma. Una sortida per l'alarma, una pantalla LCD 16x2 caràcters i dos sortides per activar el relé per activar i desactivar l'electrovàlvula.

Com que vaig adquirir l'Arduino juntament amb els dos kits disposava de mòduls connectar i llest per muntar el prototip. També de resistències i polsadors per dissenyar el circuit. El fet que el meu comptador de litres no fos un mòdul connectar i llest i haver de condicionar el senyal per l'Arduino vaig utilitzar una combinació del dos sistemes.

A la següent il·lustració (5.1.10) i il·lustració (5.1.11) es mostra unes fotografies dels primers muntatges on es pot visualitzar la placa Shiled connectar i llest, un protoboard amb els polsadors per simular el comptatge de litres i un altre per fer resset un cop activada a l'alarma. La placa Arduino no es veu, ja que les plaques Shield van col·locades directament sobre de l'Arduino.



Il·lustració:5.1.10 Visualització litres per la pantalla LCD i simulació comptador amb polsadors.



Il·lustració:5.1.11 Utilització mòduls conectar i llest i polsadors.

Per desenvolupar el meu projecte he utilitzat els mòduls connectar i llest mòdul zumbado per utilitzar d'alarma i el mòdul relé per activar l'electrovàlvula. Aquests dos mòduls es poden contemplar a la taula Il·lustració (4.4.2.) La placa Shield connecta i llest no l'he utilitzat, ja que era incompatible amb la pantalla TFTtouch 2'8".

Un cop funcionava tot això calia fer referència els litres programats en funció del mes i de l'any, ja que no es solen consumir els mateixos litres d'aigua a l'estiu que a l'hivern i, a més, en cases de segona residència. Com que el prototip necessitava saber el mes l'hi havia de proporcionar la data d'alguna forma, per això vaig equipar el prototip amb un DS1307, un microxip que té la funció de rellotge.

5.2.Rellotge DS1307

Per proveir la data el meu prototip em vaig decantar pel mòdul connectar i llest RTC Rellotge a temps real DS1307. Aquest mòdul va connectar a l'entrada, però no a una qualsevol, ja que s'utilitza una comunicació Ic2.

Aquest rellotge em va funcionar uns dies, però de sobte va deixar de funcionar. Primer vaig pensar que eren problemes en la programació, però el final va ser la placa. Després de tornar a comprar una segona placa i trobar-me en el mateix problema em vaig decidir per comprar una placa diferent, que era molt més econòmica que la primera. El rellotge utilitzat finalment va ser el que es mostra a la següent il·lustració (5.2.12) aquest mòdul també utilitza una comunicació i2c, a més, de disposa d'una petita eprom per si volem guardar algunes dades que no volem perdre per falta d'alimentació. El fet d'utilitzar una comunicació i2C em va portar haver d'aprofundir més en aquest tipus de comunicació.

Il·lustració:5.2.12 Rellotge

Font:http://www.dfrobot.com/wiki/index.php?title=Real_Time_Clock_Module_(DS1307)_(SKU:DFR0151)

5.3.Pantalla TFT Touch Shield V2.0

Avui en dia, si observem el nostre entorn ens adonarem que estem envoltats per pantalles tàctils, caixers electrònics, telèfons mòbils i els ordinador. Per aquest motiu en veure que hi havia pantalles tàctils per l'Arduino, vaig decidir substituir la meva pantalla LCD 16x2 per una pantalla TFT Touch de 2'8" model que es pot veure a la il·lustració (4.3.4) on es parla de les plaques Shield d'Arduino.

D'aquesta forma, s'aconsegueix una millor interactuació entre l'usuari i el prototip, a més de fer-ho més atractiu visualment.

El canvi de la pantalla LCD per una TFTTàctil va significar haver de canviar el programa del Arduino del tot. Tot i això, vaig creure que el repte era molt motivant i complicat alhora i vaig decidir seguir endavant.

A part de la dificultat de dissenyar la imatge de la pantalla l'altre dificultat és programar els botons que es dibuixen a la pantalla tàctil.

A continuació, es pot veure la il·lustració (5.3.13) una de les pantalles creades en el meu prototip. Concretament la pantalla principal.

Il·lustració:5.3.13 Pantalla Principal

El primer pas ha estat el de dissenyar la imatge de totes les pantalles necessàries i crear les instruccions per accedir a cada una de les pantalles. A continuació es mostra la taula (5.3.5) on s'anomenen totes les pantalles i transicions creades en el prototip. També es mostra la il·lustració (5.3.5) on es pot observar una Xarxa Petri, amb els itineraris possibles. La P significa pantalla i va precedida d'un número d'identificació. La T indica transició que s'ha de complir per poder accedir a la pantalla següent i també va precedida d'un número per identificar la transició.

Etiqueta	Nom pantalla	Descripció
P1	Pantalla inicialització	Indica versió i carrega els paràmetres del programa
P2	Alineació Eixos	S'utilitza per referència l'eix X i l'eix Y als punts 0,0
P3	Recordatori data hora i telèfon	Recordatori ajustar i comprovar data hora i telèfon
P4	Principal	Mostra el consum actual i el consum programat l'alarma.
P5	Programació	Permet accedir els paràmetres programables per l'usuari.
P6	Dia, Mes i Any	Programació del dia mes i any
P7	Hora i minuts	Programació de l'hora i minuts
P8	Telèfon	Programació del número de telèfon.
P9	Programació Consums	Programació alarma litres consumits
P10	Temps Mostreig i tan x cent	Programació del temps mostreig i percentatge excés del consum.
P11	Manual	Pantalla obrir tancar la Vàlvula forma manual
P12	Alarma	Pantalla Alarma tanca Vàlvula i envia SMS
T0	Transició inicial	Acció que es compleix al alimentar l' Arduino o després de polsar el botó de resset.
T1,T2,T3	Transició immediates	Es compleixen sempre
T4	Tecla programació	Es compleix al polsar la tecla verda programació
T5,T12	Tecla Sortir	Es compleix al polsar la tecla vermella Sortir
T6,T7,T8	Tecla Següent	Es compleix al polsar la tecla grisa següent i estem al final dels paràmetres programables de la pantalla actual.
T9	Tecla Temps i sobre consum	Es compleix al polsar la tecla verda Temps i sobre consums.
T10	Tecla Consums	Es compleix al polsar la tecla Grisa Consums.
T11	Tecla Manual	Es compleix al polsar la tecla groga Manual.
T13	Transició alarma activada	S'activa de forma automàtica al sobrepassar el consum programat.
T14	Transició Resset Alarma	Es compleix al polsar la sobre qualsevol part de la pantalla vermella un cop activada l'alarma.

Taula 5.3.5 Pantalles i Transicións programades.

La primera pantalla és d'inicialització (P1) a la xarxa i serveix per carregar paràmetres de la pantalla TFTTouch. La segona pantalla (P2) serveix d'alineació per definir els eixos X=0 i Y=0 a la part superior. La tercera (P3) és una pantalla recordatori per ajustar i comprovar la data i programar el número de telèfon. La pantalla (P4) és la principal, que es mostraria en funció automàtic. A l'alimentar-se l'Arduino o polsar el botó de resset sempre s'inicia a la pantalla d'inicialització seguidament i de forma automàtica passa a la pantalla p2 p3 fins arribar a la pantalla principal. En aquest punt l'usuari ja pot utilitzar la pantalla per desplaçar-se pels diferents apartats.

La pantalla Principal ens permet accedir a la pantalla de programació (P5). D'aquesta pantalla és pot tornar a la principal o bé anar a la pantalla (P6) de programació del dia, hora. O bé a la pantalla (P9) programació litres consumits i també a la pantalla (P10) programació temps consum i percentatge sobre consum litres i a la pantalla (P11) obrir i tancar l'electrovàlvula de forma manual.

Per accedir a la pantalla (P7) de programació hora i minuts només és possible després d'haver ajustat el dia, l'hora i l'any polsant la tecla de següent. Al mateix succeeix per accedir a la

pantalla (P8) programació del número de telèfon només és possible després de programar els minuts, l'hora i polsat sobre la tecla de següent.

Per sortir d'aquesta pantalla i tornar a la pantalla principal només cal polsar la tecla vermella, sortir. El mateix serveix per a les altres pantalles on hi podem actuar, on sempre es pot tornar a la pantalla principal. Si en algun moment l'Arduino es queda sense alimentació o es polsa el botó de resset de l' sempre s'iniciarà amb la pantalla inicialització (P1).

Queda la pantalla alarma (P12) que no és accessible per l'usuari. S'hi arriba un cop disparada l'alarma. Aquesta pantalla es passa a tancar l'electrovàlvula, a fer sonar l'alarma i a enviar un SMS al número programat prèviament. Per sortir d'aquesta pantalla i anar a la pantalla principal només cal polsar qualsevol part de la pantalla vermella d'alarma.

La Transició T0 és la que s'inicia en rebre alimentació l'Arduino o després de polsar el botó de resset. Les transicions T1,T2,T3, que és veuen en el diagrama de petri són directes. La transició T4 és el polsador de programació, T6 és la tecla de programació dia, hora i telèfon. La Transició T7, T8 és el polsador de següent. La transició T5,T12, és el polsador de sortir. La transició T11 és el polsador de manual. EL T10 és consums i T9 polsador temps i sobre consums. La transició T13 és automàtica al sobrepassar els consums programats. La transició T14 passa al polsar sobre la pantalla un cop disparada l'alarma.

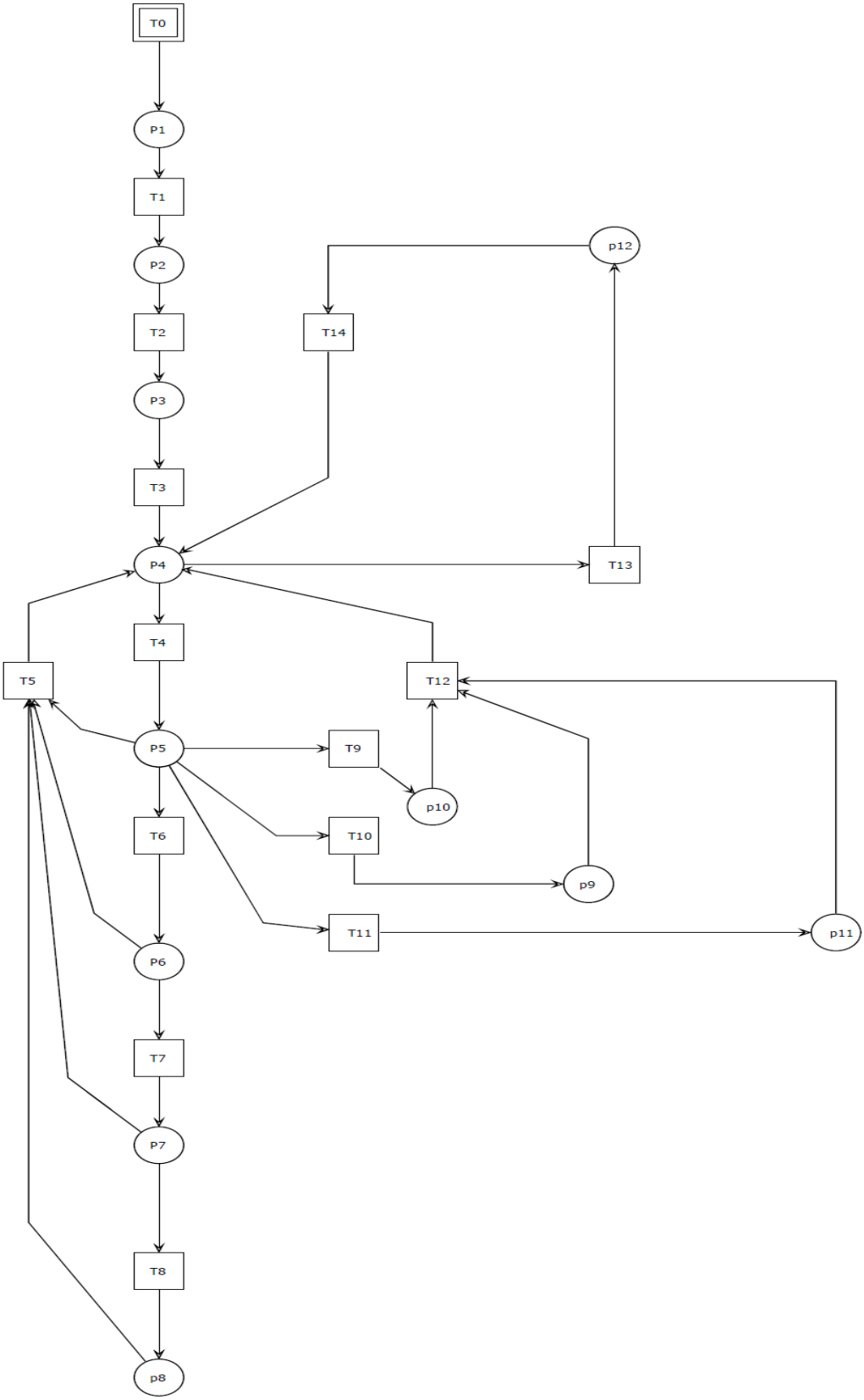
La transició T10, T17 són el polsador de sortir. La transició T15 és polsant la pantalla d'alarma. La T6, és el polsador data, hora i telèfon. La T8 i T9 és el polsador de següent. La transició T12 és el polsador temps i sobre consums. La transició T11 és el posador manual.

El fet de tenir tantes pantalles també ha provocat haver de tenir forces variables per saber en tot moment en quina pantalla es troba el programa i quins són els botons programats que ens interessa fer servir. Perquè una cosa és la imatge que es mostra per pantalla i una altra cosa són els botons que es programen. Al principi, fer coordinar les dues coses va ser una mica complicat.

En funcionament normal el programa mostra de forma continua la pantalla principal P4 il·lustració (5.3.14) on es mostra la data, l'hora, valor del comptador, els litres consumits, el temps de mostreig i el sobre consum en el qual s'activa l'alarma. Un cop sobrepassat el temps de mostreig els litres consumits tornen a zero. Tornant-se a iniciar un nou mostreig.

Il·lustració:5.3.14 Pantalla principal.

Tot seguit, és mostra la xarxa de petri a la il·lustració (5.3.15) on és pot comprovar de forma molt més visual el diagrama de flux que permet anar d'una pantalla a l'altra.

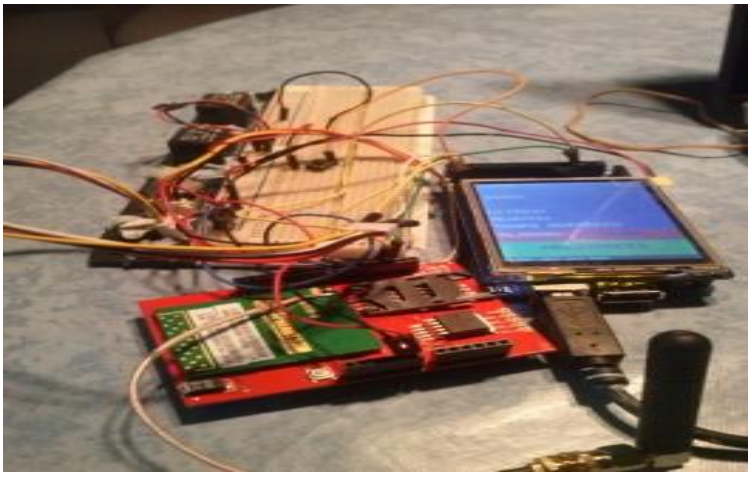


Il·lustració:5.3.15 Xarxa de petri

Seguidament, es mostra un petit exemple d'una de les funcions que trobem en el programa sencer a l'annex (4). Aquesta funció permet mostrar per pantalla el que es permet mostrar la pantalla de programació.

5.4.SMS

L'enviament de SMS es realitza de forma automàtica al disparar-se l'alarma, a la pantalla P12. Per poder enviar el SMS cal prèviament haver programat el número de telèfon en la pantalla de programació després de programar el dia, el mes, l'any, el minut i l'hora. El número s'ha de programar amb el 00 i el codi del país, Espanya el 34 seguit pel número que volem que rebi el SMS, sempre i quan s'hagi introduït una targeta SIM. La targeta SIM ha d'estar des bloquejada i que no sol·liciti el número PIN. La Shield GSM-GPRS Sparkfun a més de permetre funcions de SMS, també permet totes les funcions GSM/GPRS i TCP/IP. És a dir, a part de poder enviar i rebre SMS pot realitzar trucades telefòniques connectant un altaveu i un micròfon en els pins que ja venen predisposats. La il·lustració (5.4.16) es mostra una imatge de les primeres proves d'enviar SMS.



Il·lustració:5.4.16 Proves per enviar SMS

Ara s'adjunta el programa per enviar SMS. Cal configurar les variables per poder escriure el número de telèfon des de l'exterior de la programació d'Arduino. Si no es realitzés l'accés al número de telèfon des de fora l'Arduino el programa seria més simple. Aquest programa és una part del programa sencer que es pot robar a l'annex (4)

```
#include <SerialGSM.h>
#include <SoftwareSerial.h>
#include <String.h>
SerialGSM cell(48,49);
char Str4[12] = "";
int tf1=0;
int tf2=0;
int tf3=3;
int tf4=4;
int tf5=6;
int tf6=5;
int tf7=5;
int tf8=4;
int tf9=5;
int tf10=3;
int tf11=3;
int tf12=2;
int tf13=5;
char an;
char bn;
char cn;
char dn;
char en;
char fn;
char gn;
char hn;
char in;
char jn;
char kn;
char ln;
char mn;
```

```
void setup()
        {
         Serial.begin(9600);
         cell.begin(9600);
         cell.Verbose(true);
         cell.Boot();
         cell.FwdSMS2Serial();
         cell.Rcpt("AT+CCLK?");
         tf1=0+48;
         an=tf1;
         tf2=0+48;
         bn=tf2;
         tf3=3+48;
         cn=tf3;
         tf4=4+48;
         dn=tf4;
         tf5=6+48;
         en=tf5;
         tf6=5+48;
         fn=tf6;
         tf7=5+48;
         gn=tf7;
         tf8=4+48;
         hn=tf8;
         tf9=5+48;
         in=tf9;
         tf10=3+48;
         jn=tf10;
         tf11=3+48;
         kn=tf11;
         tf12=2+48;
         ln=tf12;
```

```
        tf13=5+48;
        mn=tf13;
        Str4[0]=an;
        Str4[1]=bn;
        Str4[2]=cn;
        Str4[3]=dn;
        Str4[4]=en;
        Str4[5]=fn;
        Str4[6]=gn;
        Str4[7]=hn;
        Str4[8]=in;
        Str4[9]=jn;
        Str4[10]=kn;
        Str4[11]=ln;
        Str4[12]=mn;
        cell.Rcpt(Str4);
        cell.Message(" Valvula Tancada");
        cell.SendSMS();
       }
void loop()
      {
      if (cell.ReceiveSMS())
                  {
                  Serial.println(cell.Message());
                  cell.DeleteAllSMS();
                  Serial.println("AT+CCLK");
                  }

}
```

5.5.Comptador d'aigua de polsos i l'electrovàlvula.

La part del comptador d'aigua consta de dues parts, una és un comptador d'aigua Sensus 120 i l'altre el sensor HRI. El comptador d'aigua bé ja preparat per incorporar el Sensor HRI. S'ha adquirit un comptador per poder fer les proves tot i que el sensor HRI ve preparat per ser instal·lat en la majoria de comptadors d'aigua. En canviar el comptador d'aigua cal tenir present que si substituïm el nostre comptador d'aigua de l'entrada general, aquest ha de complir una reglamentació específica. Aquest la compleix i es pot canviar sense problemes avisant a la companyia d'aigua. El número de sèrie del comptador apareix a la factura d'aigua i per això s'ha d'avisar a la companyia d'aigua. L'altre opció és posar-lo a continuació del comptador ja existent. A la il·lustració (5.5.17) podem observar la imatge del comptador del sensor i de les dues peces ensamblades. Com es pot veure a la foto de més a la dreta, encara que hi hagi el sensor col·locat és pot llegir el valor del comptador de forma manual.

Il·lustració:5.5.17 D'esquerra a dreta Comptador sensor de polsos. I les dues peces ensamblades.

El Sensor és el model HRI-A1, aquest model compta impulsos i està ajustat a un impuls 1 litre. El model B en comptes de marcar els polsos a través dels cables es consulta el valor actual del comptador.

5.6.Ensamblatge.

Un cop programat i provat el prototip calia fer-lo més compacta i protegir totes les connexions i parts del projecte. Es podia haver realitzat el prototip d'una forma molt més compacta i tot en una sola peça. Però la decisió de fer-ho en diferents parts és molt més encertada. Moltes vegades els comptadors d'aigua estan col·locats en una bateria de comptadors d'aigua, fet que dificultaria la instal·lació d'un equip més compacta. L'altre motiu per construir el prototip en diferents parts és la de tenir més distància entre la part elèctrica i la part hidràulica, fent més difícil que la part elèctrica entri en contacta amb l'aigua i evitar així futurs problemes.

Per ensamblar el prototip he comprat una caixa de plàstic de 190x240x90 IP56. Amb aquesta caixa queden protegides totes les connexions de l'Arduino. Una placa de muntatge per poder

collar totes les peces a dins la caixa i una finestra plàstica 78x130x15. Aquesta finestra un cop instal·lada a la tapa de la caixa ens permetrà un fàcil accés a la pantalla Tàctil. Aquestes peces es mostren a la Il·lustració (5.7.18).

Il·lustració:5.7.18 Caixa de protecció finestra i placa per muntatges.

A més, del material vist anterior, també he utilitzat un Portafusibles de panell per Fusible 5x20mm, per tal de prevenir curtcircuits, un connector Jack DC per a panell per poder alimentar l'Arduino, bornes per a guiar din, per realitzar totes les connexions, frens per bornes de guia din, guia din i una canaleta Unex 60x40 per poder collar la placa Arduino i la placa Shield GSM. A continuació, es mostra a la il·lustració (5.7.19) una imatge de totes les peces utilitzades per l'assemblatge esmentades anteriorment.

Il·lustració:5.7.19 Primera linia D'esquerra a dreta: Connector jakc Alimentació, Porta fusibles interruptor basculant vermell

Segona linea d'esquerra a dreta Canaleta Unex, Borna, tapa borna, fre borna i guia Din.

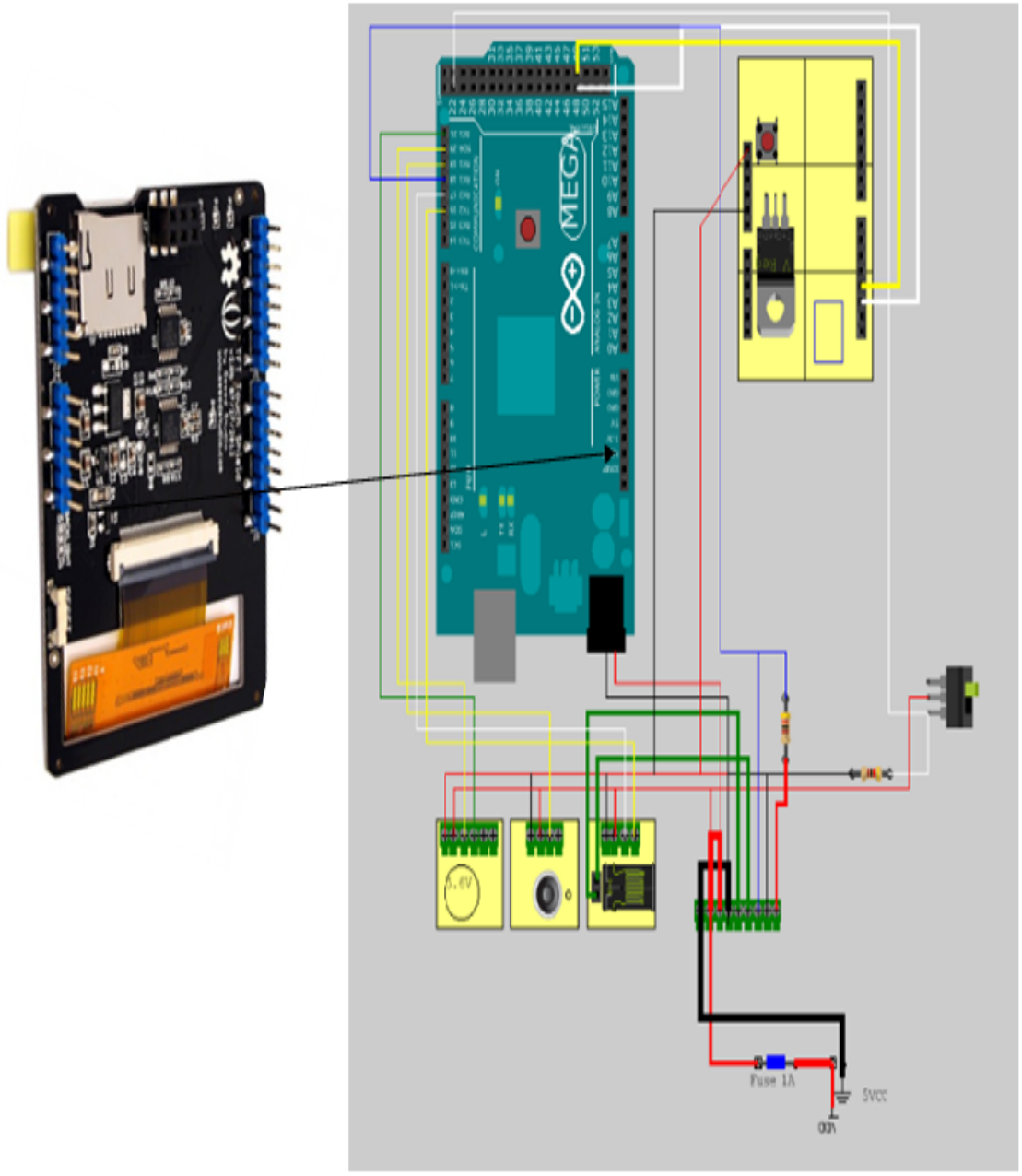
Per l'assemblatge també s'ha utilitzat angelets, visos i petit material. Aquest no surt reflectit a la fotografia per tenir poca importància, però si ha estat comptabilitzat en el preu del material del prototip. A més del material utilitzat per al muntatge del comptador i electrovàlvula que estan reflectits en l'apartat que es parla del comptador i l'electrovàlvula. Aquesta part és la part central del projecte i per això aquesta part l'anomeno com a centraleta.

Un cop col·locats totes les parts dins la caixa és hora de fer les connexions elèctriques de totes les parts. A la il·lustració (5.7.20) es pot veure un esquema amb totes les connexions realitzades. La TFT Touch Shield V2.0 i la placa GSM-GPRS SM5100B són plaques Shield, per tant poden anar col·locades directament sobre de l'Arduino. Tot i que mecànicament s'hi poden posar les dues plaques una sobre no funciona, per això he optat per col·locar la pantalla TFT directament a sobre l'ATmega i la placa GSM-GPRS SM5100B fer el cablejat elèctric necessari. és millor fer aquest cablejat perquè aquesta placa necessita menys fils que la pantalla, només en necessita 2 d'alimentació i 2 de comunicació.

A la il·lustració (5.7.20) es pot veure com he utilitzat la part alta d'entrades i sortides ja que la TFT utilitza des de la 1 fins a la 13 connectada directament. A la il·lustració es pot veure la part de darrera de la TFt poden contemplar els pins que s'utilitza per col·locar aquesta placa sobre l'Arduino i al mateix temps que queda connectada es queda subjectada. Una fletxa de color negra marca els pins que queden units al girar i connecta la pantalla amb l'Arduino. El fet de tenir els pins soldats fa que la pantalla tingui poques posicions però sí que es pot col·locat la pantalla d'una altre forma sense que aquesta funcioni correctament.

Totes les plaques estan alimentades a 5V amb el cable vermell i negre d'alimentació.

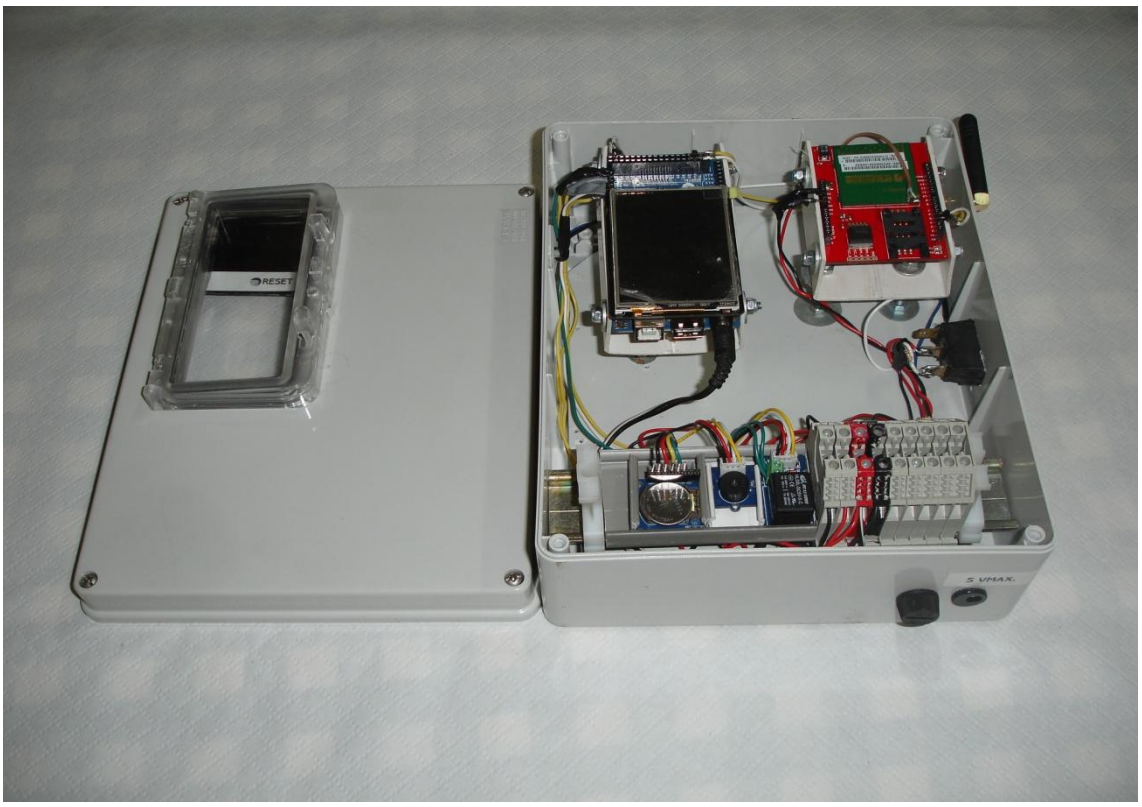
El rellotge utilitza la comunicació I2C, per tant, està connectat als pins 20 SDA i 21 SCL del Arduino. Les altres connexions només cal connectar-les als pins que hem decidit a l'hora de fer el programa. Els pins 48 i 49 del Arduino van els pins 2,3 de la placa GSM GPRS SM5100B. El mòdul connectar Zumbador i el mòdul relè van connectats els pins 19 i 16 respectivament. El pin 18 va col·locat al bornes per la connexió del sensor HRI-A. Tenint dos bornes més per l'alimentació del sensor. També es por veure la imatge de la resistència de la qual en fem referència en el apartat (5.1) on es parla de les entrades i sortides de l'Arduino.



Il·lustració:5.7.20 Esquema connexió elèctrica imatge pins TFT

5.7.Resultat final

A la il·lustració (5.6.21) es mostra una imatge de la centraleta on es pot observar una part de l'Arduino Atmega ADK, la pantalla TFT, la placa Shield GSM, antena, etc. També es pot observar la tapa de la caixa amb la finestra instal·lada.



Il·lustració:5.6.21 Centraleta

A la part inferior de la imatge és pot observar el portafusibles i el jac per connectar l'alimentació.

5.8.Cost del prototip

A la taula (5.8.6) es mostra el nom de tots els components utilitzats, la quantitat utilitzada i el preu.

Descripció	Quantitat	Preu en Euros
Arduino Atmega ADk	1	59'00
Arduino TFT Touch Shield V2.0	1	41'00
Arduino Cel·lular Shield - SM5100B	1	81'30
Antena GSM con connector SMA	1	7'25
Connector Femella Arduino - 8 pines	2	0'50
Connector Femella Arduino - 6 pines	2	0'50
Mòdul Relé 5V Connectar i Llest	2	4'35
Mòdul Zumbador Connectar i Llest	1	3'75
DS1307 I2C RTC DS1307 24C32 Real Time	1	2'75
Cables 4 Conductors Shield Connectar i llest	1	5'85
Connector Jack DC para panel	1	3'30
Fuente de Alimentación Estabilitzada 1000 mA	1	10'16
Portafusibles panel para Fusible 5x20mm.	1	0'78
Placa Scame -686208 190x240x90	1	8'70
Armari Hazemeyer	1	64'21
Finestra plàstica 78x130x15	1	11'10
BornesWK4/U	10	2'90
Comptador Dn15-115 A.F ¾"	1	27'48
Emissor de impulsos HRi-Polsus	1	65'27
Electrovàlvula	1	62'01
Angelets sense ganxo m4	4	0'65
Canal Unex 60x40	1 mt	6'63
Prensastopes	2	6,25
Interruptor basculant vermell	1	1'12
Total Preu Material		**295'51€**

Taula 5.8.6 Taula Cost del material.

6.Instal.lació de l'equip

En aquest capítol s'explica com realitzar el muntatge de tot el prototip. El primer pas és identificar totes les peces. A continuació hi ha una il·lustració (6.22) on es poden veure enumerades totes les parts que component aquest projecte.

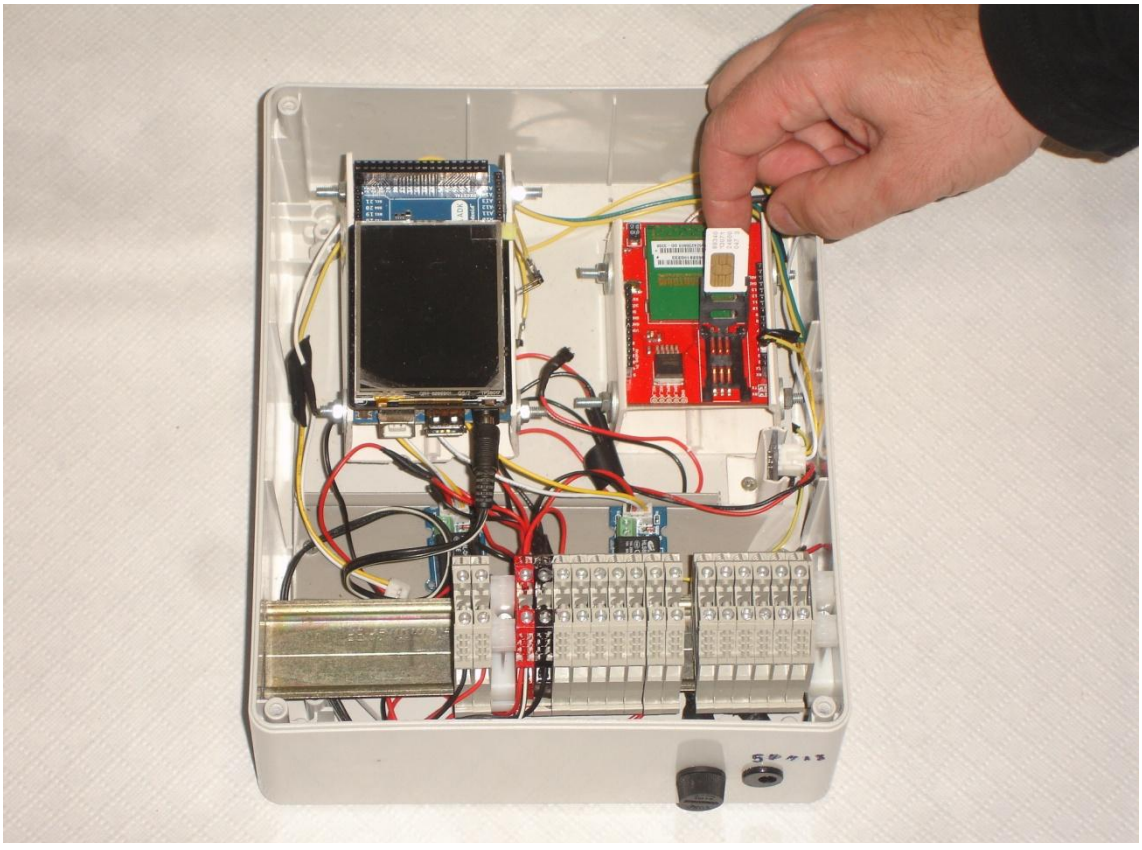
Il·lustració:6.22: 1 Centraleta, 2 Transformador Alimentació, 3 Comptador, 4 Sensor medidor Polsus HR-I. 5 juntes 7 Electrovàlvula NO

El número 1 és la centraleta, part principal del projecte, el 2 és un transformador regulable de 4 fins 12V. L'alimentació de l'Arduino pot ser de 5 a 12V, però per protegir les plaques Shield és recomanable tenir el transformador ajustat a 5V, el 3 és el comptador, que va instal·lat a l'entrada del sistema hidràulic que es vol controlar, el 4 és el sensor de Polsos HR-I, el 5 són juntes de goma ¾ per col·locar els extrems del comptador, el 6 és un tros de tub de coure de 22 i dos entronques solda ¾ per poder adaptar el prototip en una aixeta per poder realitzar proves i finalment, el 7 és una electrovàlvula normalment oberta. Al alimentar la bobina la electrovàlvula passarà del seu estat oberta a tancada.

6.1.Centraleta

El primer pas és col·locar la centraleta. Per col·locar-la cal buscar una zona propera al comptador d'aigua, el sensor HR-I disposa d'un cable de 1,5mts i si s'allarga pot provocar algun problema. A més, la zona on posem la centraleta ha de disposar de cobertura GSM, la qual podem comprovar amb el nostre telèfon mòbil. En cas de no tenir cobertura GSM el projecte funcionarà, però no enviarà el SMS. L'única forma de saber si la vàlvula està tancada serà escoltant l'alarma i mirant la pantalla.

Un cop collada la caixa a la paret cal incerta una targeta SIM a la placa SHIELD GSM. En la Il·lustració (6.1.23) es pot veure a on i com s'ha de col·locar la targeta. Perquè la targeta funcioni de forma correcta cal que estigui des bloquejada i no sol·liciti el codi PIN. Per des bloquejar-la o provar si la targeta demana el codi PIN, podem utilitzar el telèfon mòbil.



Il·lustració:6.1.23 imatge de la centraleta on es pot veure on i com es col·loca la targeta SIM.

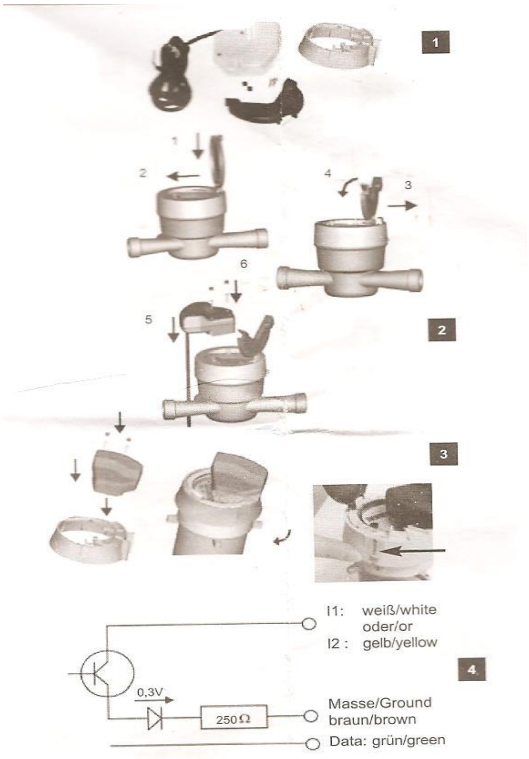
6.2.Comptador i electrovàlvula

Un cop muntada la centraleta és el torn del comptador i de l'electrovàlvula. A la il·lustració (6.2.24) es poden observar aquestes peces. Al realitzar aquesta instal·lació cal anar en molt de compte, el comprador té un sentit de connexió i l'electrovàlvula també. Per saber el sentit del comptador cal fixar-se en la imatge, sota de la part de plàstic blanca, tal i com es pot veure a la imatge apareix una fletxa que ens marca el sentit del flux d'aigua. L'electrovàlvula està indicat amb les paraules Out i IN

Il·lustració:6.2.24 Comptador i electrovàlvula.

Si instal·lem el comptador de forma equivocada, el comptador de polsos no marcarà, a més, el comptador porta un sistema de memorització de polsos invertits. Això vol dir que si passen de polsos en sentit invers hauran de passar onze polsos per marcar un pols.

A la il·lustració (6.2.25) es mostra la forma d'acoplar el sensor Sensus HRI-A1 al comptador.



Il·lustració:6.2.25 Sensor Sensus HRI-A1.

Font: Manual d'istruccions Sensus HRI-A1

6.3.Connexions elèctriques

A continuació, s'adjunta la il·lustració (6.3.7) on es veu un esquema dels bornes de la centraleta. S'indica el que hem de connectar a cada borna. Els bornes 1,2,3 i 4 s'han utilitzat per realitzar connexions internes i no s'hi ha de connectar cap cable extern. El borna 5 i 6 els intercalarem a la fase que va a l'electrovàlvula, ja que són els que van al relè. El borna 6 el connectarem al cable blanc del sensor. La borna 7 hi col·locarem el marró i el borna 9 hi col·locarem el cable verd.

1	2	3	4	5	6	7	8	9
⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
+	-	+ 1A	-	Relè	Relè	Entrada polsos	- Sensor	+ Sensor

Taula 6.3.7. Esquema representatiu dels bornes

L'alimentació de la centraleta es realitza connectant el transformador de 5V al jack. Aquest està situat a la part exterior inferior esquerra de la centraleta.

7.Posar en marxa

En aquest capítol s'explica el que l'usuari pot visualitzar a cada pantalla i el que pot fer a cada una d'elles. Aquest és molt semblant al capítol 5.7 on es parlava de la Placa Shield TFT V2. Tot i assemblar-se molt en l'apartat 5.7, està més centrat en la part de programació mostrant un diagrama de flux i les transicions a complir. En aquest cas es mostra el que podria ser el manual d'instruccions del prototip i explicant tot el que es veu a cada pantalla.

7.1.Inicialització

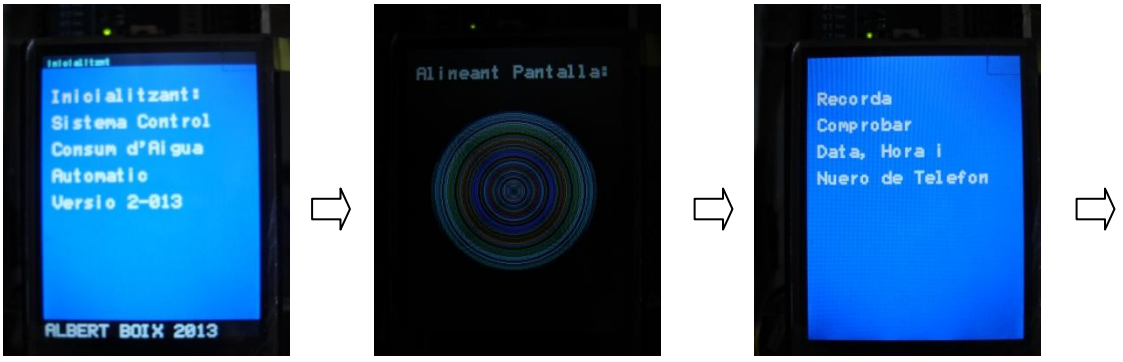
Un cop instal·lades totes les parts i connectats els cables podem passar a connectar la centraleta a la font d'alimentació amb el transformador. Cal comprovar tenir-lo ajustat a 5V.

Només d'alimentar la centraleta s'il·luminarà la pantalla. En cas de no il·luminar-se polsar l'interruptor vermell, aquest serveix per desconnectar la pantalla un cop programat i ajustat. D'aquesta forma s'allarga la vida útil de la pantalla. Si tot i així no s'il·luminés cal comprovar que el transformador tingui tensió. Si tot i així no s'il·lumina comprovem l'estat del fusible. El porta fusibles està al costat del jack d'alimentació. Aquest fusible ha de ser 1A màxim. Un cop alimentat de forma automàtica apareixerà la pantalla d'inicialització. Es mostra la paraula inicialitzant sistema de control d'aigua automàtic i el número de versió del programa,2-2013.

La versió 1 va ser el primer programa que vaig fer. Però després la pantalla TFT se'm va trancar i la nova TFT és programava de forma diferent, per això vaig posar versió 2.

Tot seguit i de forma automàtica, apareix un cercle i marquen els 0,0 del eix X i Y. També apareix la tercera pantalla on es mostra un recordatori per ajustar i comprovar la data, hora i número de telèfon on volem que s'enviï el SMS.

La il·lustració (7.1.26) es una imatge de les tres pantalles que es mostren al iniciar o reiniciar l'Arduino.



Il·lustració:7.1.26 d'esquerra a dreta pantalla inicialització, pantalla alineació i pantalla recordatori.

Aquestes pantalles es mostren de forma automàtica sense intervenció de l'usuari. Tot seguit s'arriba a la pantalla principal en la il·lustració (7.1.27). En aquesta pantalla es mostra la data i hora. A sota es mostra el número de telèfon on s'enviarà el SMS, a sota el valor del comptador,

els litres consumits, el temps de marxa, el temps de mostreig transcorregut amb la paraula minuts i el temps de mostreig programat. Seguidament trobem de color vermell un valor que ens indica quan s'activarà l'alarma. Aquest valor depèn dels consums programats, del mes actual i del percentatge programat. Per últim, a la part final de la pantalla trobem un polsador verd que ens permet entrar a la pantalla de programació.

Il·lustració:7.1.27 Pantalla principal.

En el cas que l'usuari polsi sobre el botó de color verd accedirem a la pantalla programació a la il·lustració (7.1.28) on amb diferents botons podrem accedir els diferents paràmetres programables. En aquesta pantalla podem trobar un polsador per programar els consum, un altre per programar el dia, l'hora i el número de telèfon. El botó de color verd és per programar el temps de mostreig i el sobre consum que volem activar l'alarma. També trobem un polsador de color groc per tancar i obrir la vàlvula de forma manual. Al final hi trobem un polsador vermell per tornar a la pantalla principal.

Il·lustració:7.1.28 Pantalla programació.

A continuació, a la il·lustració (7.1.29) es mostra la pantalla la programació de consum. En aquesta pantalla es visualitza un mes i un valor que posa consum. El valor que programem a

consums serà el valor de referència per disparar l'alarma d'excés de consum d'aigua. Aquest valor s'ha de programa per tots els mesos, ja que el consum d'aigua sol ser diferent l'estiu de l' hivern. Per modificar el valor de referència programat s'ha d'utilitzar (+ / –)**.** Un cop programat el primer mes podem passar el següent polsant el botó de següent. Un cop programat els dotze mesos podem polsar la tecla vermella de sortir per retornar el la pantalla principal.

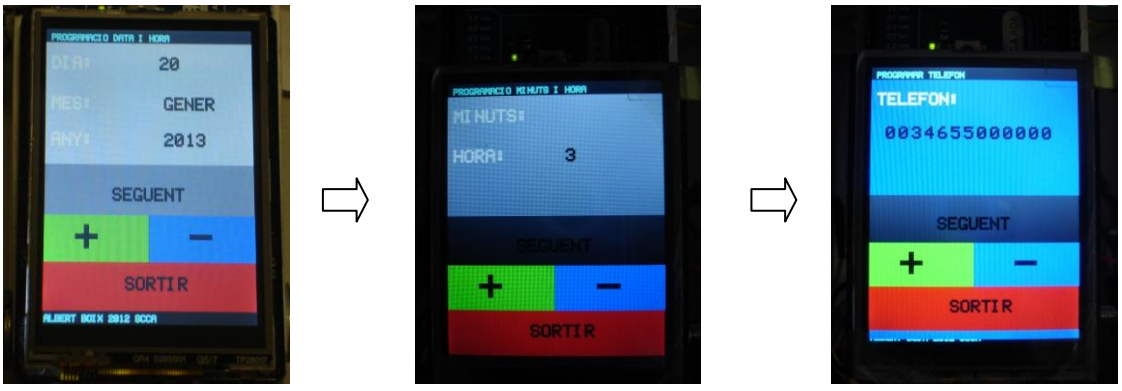
Els consums s'han de programar tots els mesos, perquè el consum d'aigua sol ser diferent depenent del mes de l'any.

Il·lustració:7.1.29 Pantalla principal.

Cal recordar que el valor programat a consums és de referència i no el valor exacte on s'activarà l'alarma. Aquest valor d'activació també dependrà del percentatge programat a la pantalla excés litres %. Si, programem 0 seria el valor exacte, si i programem un 5% el valor hauria de sobrepassar un 5% del consum.

Un cop tornat el menú principal tornem a polsar programació i passem a programar la data, l'hora i el número de telèfon.

Per programar aquests valors polsem la tecla DIA, HORA I TELÈFON de la pantalla programació, accedirem a la pantalla programació dia, mes i any. Primera imatge de la il·lustració (7.1.30).



Il·lustració:7.1.30 D'esquerra a dreta Pantalla programació Dia, pantalla programació minuts i hora i pantalla programació telèfon

En aquesta pantalla hem d'ajustar primer el dia amb les tecles (+ | -). Un cop ajustat el dia actual polsarem la tecla següent i ens permetrà escollir el mes, utilitzant també les tecles (+ | -). Un cop seleccionat tornem a polsar la tecla següent i ens permetrà ajustar l'any.

Un cop ajustat tornem a polsar la tecla següent i accedirem a la pantalla programació minuts i hora, segona imatge de la il·lustració (7.1.30) Primer ajustem els minuts amb les tecles (+ | -). Un cop ajustat polsem la tecla següent i ens permetrà amb les tecles (+ | -) ajustar l'hora amb format 24 hores. Tot seguit polsem següent i accedirem a l'última pantalla d'aquest apartat. La pantalla programació telèfon. En aquesta pantalla hem d'escriure el número de telèfon on volem enviar el SMS d'alarma vàlvula tancada. El número de telèfon ha de començar amb 0034 per Espanya. Els números es programen amb les tecles (+ | -) i següent per accedir al dígit del costat. Per sortir d'aquesta pantalla s'ha de fer amb el botó vermell de sortir i retornarem a la pantalla principal.

Un cop a la pantalla principal podem tornar a entrar al menú programació i polsar la tecla temps i sobre consum accedirem a la pantalla programació temps i sobre consums il·lustració (7.1.31). El temps ha d'estar programat amb minuts, per exemple si posem 120 serien dues hores.

Això vol dir que la centraleta estarà dues hores contant els litres consumits. Si en aquest interval de temps els litres programat sobrepassen els litres consumits s'activarà l'alarma. Per el contrari un cop esgotat el temps els litres consumits es tornarà a posar a zero i començarà a comptar de nou.

Si polsem la tecla de següent passem a programar el percentatge que volem que sobrepassi el valor de litres consumits.

Il·lustració:7.1.31 Pantalla programació temps mostreig i tan per cent sobre consum.

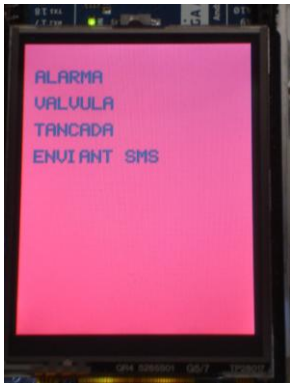
Per sortir d'aquesta pantalla polsem sortir i tornem anar a la pantalla principal.

Finalment tenim si tornem entrar a la pantalla programació podem accedir a la pantalla manual, la il·lustració (7.1.32). Aquesta pantalla ens permet de forma manual obrir i tancar l'electrovàlvula.

Il·lustració:7.1.32 Pantalla obrir tancar electrovàlvula.

Per sortir d'aquesta pantalla només hem de polsar la tecla vermella per sortir i tornarem a la pantalla principal. Si sortim d'aquesta pantalla es torna a l'estat automàtic de forma immediata. L'estat de manual només és vigent en aquesta pantalla.

Si en mode funcionament s'activa l'alarma automàticament es mostrarà la pantalla alarma. La Il·lustració (7.1.33). Aquesta pantalla ens avisa que s'ha tancat l'electrovàlvula a més d'aparèixer un text que indica que s'està enviant un SMS. Un cop enviat les paraula enviant canviarà per SMS enviat. Per esborrar l'alarma cal polsar a qualsevol part vermella de la pantalla durant un temps mínim de 3 segons.



Il·lustració:7.1.33 Pantalla Alarma.

8.Conclusions i futures millores

Un cop dut a terme la construcció del prototip amb l'entorn Arduino he comprovat que és un entorn molt versàtil ideal per desenvolupar tota mena de prototips electrònics. Aquestes plaques ens permet un gran estalvi de temps en hores de disseny i de fabricació, evitan al mateix temps possibles errors que ens encariran els costos del prototip.

Ja sigui amb informació de pàgines oficials o no oficials, aquesta és molt extença. Al mateix temps, el fet de ser una plataforma amb una expansió tan ràpida implica poder disposar de tecnologia molt actual per poder afegir el nostre prototip. Com per exemple: Bluethooth wiffi, les pantalles tàctils, etc.

El substituir la pantalla LCD per la pantalla TFTTouch ha aportat una dificultat extra al desenvolupament del prototip. Al mateix temps ha augmentat el tamany del programa i ha fet augmentar el seu cost, tant pel preu de la pantalla com l'augment d'hores de programació. Tot i aquests desevantages, el canvi de pantalla crec que ha set molt positu. A contrïbuit una millor interactuació en el projecte entre l'usuari i el prototip.

Les plaques Shield són de gran utilitat per augmentar les característiques del meu prototip. Però cal vigilar amb el model i la versió. Si les llibreries no són les de la versió que estem utilitzant ens pot donar més d'un mal de cap. Em va costa una mica realitzar els primers passos amb la pantalla tàctil i quan la començava a dominar em va caure i es va trencar. La segona pantalla funcionava de forma diferent tot i ser igual, però una versió nova i vaig haver de tornar quasi els inicis del meu programa.

Com ja he dit, a internet existeix molta informació i exemples sobre les plaques Arduino, les plaques Shields i els mòduls connectar i llest, però no tenim cap certesa de que aquesta estigui contrestada i funcioni de forma correcta.

Personalment m'ha agradat desenvolupar aquest projecte amb l'Arduino fet que m'ha permès aprofundir i ampliar els meus coneixemets d'aquest entorn. Aquest amb poden ser molt útils per a desenvolupar futurs prototips electrònics.

Totes les coses sempre és poden millorar i molt més un primer prototip. Aquest ha evolucionat molt des dels seus inicis i té poc a veure en el primer prototip que vaig pensar i dissenyar i, fins i tot, començar a fabricar. Tot i això, un cop finalitzat i feria algunes modificacions i millores.

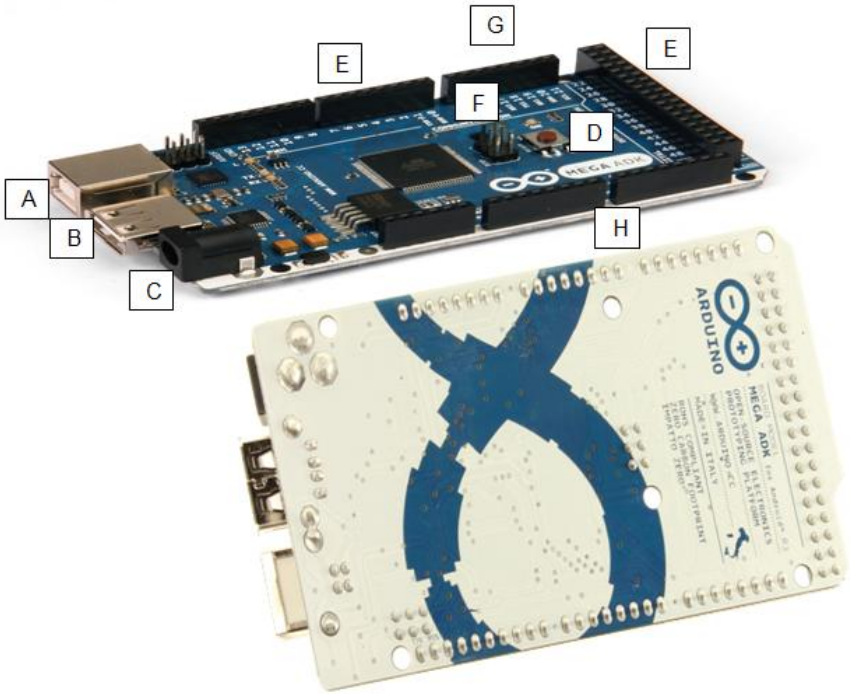
Una de les primers coses que modificaria seria el comptador de Polsos HR-I, el substituiria per el model HR-I B, la diferència és que en comptes de mesurar polsos es pot consultar els litres consumits. Amb el comptador tipus B, a més, de saber el consum fent comparacions amb la lectura anterior sabríem el consum total i podríem enviar aquest valor a la companyia d'aigua. En aquest model ho he dut a terme a través de les variables, però ja no és el valor real del comptador, si nó que depent de la programació.

Una altre possible millora seria la utilització de una placa Shield XBEE, la qual ens permetria connectar el comptador de polsos de forma inalàmbrica. També podrien incloure un mòdul WIFII per poder fer consultes des d'internet. A part, es podria afegir una aplicació per poder connectar-se amb l'Arduino i tancar l'electrovàlvula a distància enviant un SMS.

9.Annexos

1.Annexos Arduino Atmega ADK Android

A la taula 4.2.1 es mostrava la imatge i les principals característiques de les diferents plaques Arduino. A continuació es pot veure una imatge ampliada i amb meés resolució de la placa utilitzada per dur a terme el prototip. Al mateix temps s'explica les característiques de la placa de forma més detallada.



Lletra	Descripció
A	**Micro USB Socket**: connexions placa principal al PC. S'utilitza per esbós càrrega utilitzant Arduino IDE.
B	**USB A Plug**: es connecta a dispositius mòbils Android.
C	**Connector JST / DC Jack**: per la font d'alimentació externa de CC. No connecteu PC en utilitzar DC extern.
D	**Botó de reinici**: situat al costat per permetre l'ús de reajustament durant l'ús d'escuts.
E	**I / O Pins**
F	**ICSP**: per a la programació d' Bootloader usant AVR ICSP
G	Comunicació TX RX SDA SCL
H	Entrades Analògiques

Mides de la placa 8,64x5,33cm

L'ADK és una placa electrònica basat en el Microcontrolador Atmega 2560.

El datasheet Atmel té una extensió de 444 pàgines. A continuació trobem dos enllaços per poder descarregar la documentació:

www.**atmel**.com/images/doc2549.pdf

http://www.arduino.cc/en/Main/ArduinoBoardADK

Compta amb una interfície host USB per connectar amb els telèfons basats en Android, basat en el MAX3421E IC. Té 54 pins digitals d'entrada / sortida (dels quals 15 es poden utilitzar com a sortides PWM), 16 entrades analògiques, 4 UARTs (ports sèrie de maquinari), un rellotge de 16 MHz oscil·lador de vidre, una connexió USB, un connector d'alimentació, una capçalera ICSP i un botó de reinici. També diposa amb una ATmega8U2 programat com convertidor USB a sèrie i una resistència pulling WW 8U2 per facilita posar en mode DFU.

Disposa de pins SDA i SCL permeten una comunicació i2c

Per obtenir informació sobre l'ús de la targeta amb el sistema operatiu Android, consulteu la documentació de Google ADK . En el següent enllaç:

http://developer.android.com/tools/adk/index.html

Resum d'especificacions tècniques:

Microcontroladors	Atmega2560
Voltatge de funcionament	5V
Voltatge d'entrada (recomanat)	7-12V
Voltatge d'entrada (límits)	6-20V
Pines E/S digitals	54 (dels quals 15 proporcionen PWM)
Pins d'entrada analògica	16
DC Corrent per I / O Pin	40 mA
Corrent CC per Pin 3.3V	50 mA
Memòria Flash	256 KB dels quals 8 KB usats per bootloader
SRAM	8 kB
EEPROM	4 kB
Velocitat del rellotge	16 MHz

Potència

l'ADK pot ser alimentat a través de la connexió USB o amb una font d'alimentació externa. La font d'alimentació es selecciona automàticament.

(No USB) Font d'alimentació externa pot venir amb un adaptador d'AC-DC o la bateria. L'adaptador es pot connectar al connectar un endoll de 2,1 mm de centre positiu en el connector d'alimentació de la placa. Cables de la bateria es poden inserir en els capçals de pin GND i Vin del connector d'alimentació.

Nota: Com que el ADK és un Host USB, el telèfon intentarà extreure energia d'ella quan es carrega. Quan el ADK és alimentat a través de USB, el consum total és de 500mA disponible per al telèfon i el regulador d'alimentació placa externa. Aquestes poden subministrar fins 1500mA. 750mA està disponible per al telèfon i la targeta ADK. Un 750mA addicional s'assigna per als actuadors i sensors connectats a la targeta. Una font d'alimentació ha de ser capaç de proporcionar 1.5A per utilitzar aquesta quantitat de corrent.

La targeta pot funcionar amb un subministrament extern de 5,5-16 volts. Si es proporcionen menys de 7V, però el pin de 5V pot subministrar menys de cinc volts i la junta pot ser inestable. Si s'utilitza més d'12V, el regulador de voltatge es pot sobreescalfar i malmetre la placa. El rang recomanat és de 7 a 12 volts.

Els pins d'alimentació són els següents:

- **VIN**, el voltatge d'entrada a la placa Arduino quan es tracta d'utilitzar una font d'alimentació externa (en lloc de 5 volts de la connexió USB o una altra font d'alimentació regulada) es pot subministrar la tensió a través d'aquest pin o del subministrament de tensió a través de la presa d'alimentació.

- **5V**, aquest pin ens don com a sortida una tensió de 5V regulada des del regulador incorporat a la placa. Si en lloc d'utilitzar aquesta alimentació utilitzem la del USB (5V) o directament del alimentador de la placa, aquesta alimentació no passa a través del regulador i pot malmetre la placa.

- **3V3**, un subministrament de 3,3 volts generats pel regulador de la placa i el drenatge actual màxim és de 50 mA.

- **GND.** Pins de terra.

- IOREF, aquest pin de la placa Arduino proporciona la referència de tensió amb la qual opera el microcontrolador. Un escut configurat pot llegir el voltatge pin IOREF i seleccioneu la font d'alimentació adequada o habilitar traductors voltatge en les sortides per treballar amb els 5V o 3.3V.

La memòria del ADK té 256 KB de memòria flash per a l'emmagatzematge de codi (de les que 8 kB s'utilitza per al carregador), 8 KB de SRAM i 4 KB de EEPROM (que pot ser llegit i escrit amb la llibreria EEPROM).

Entrada i sortida, cadascun dels 50 pins digitals sobre el ADK es pot utilitzar com una entrada o sortida, utilitzant pinMode () , digitalWrite () , i digitalRead () funcions. Aquets funcionen a 5 volts. Cada pin pot proporcionar o rebre un màxim de 40 mA i té una resistència pull-up interna (desconnectada per defecte) de 20 a 50 kOhm. A més, alguns pins tenen funcions especialitzades:

Sèrie: 0 (RX) i 1 (TX); sèrie 1: 19 (RX) i 18 (TX); sèrie 2: 17 (RX) i 16 (TX); sèrie 3: 15 (RX) i 14 (TX). S'utilitza per rebre (RX) i de transmissió (TX) de dades en sèrie TTL. Pines 0 i 1 també estan connectats a les clavilles corresponents de la ATmega8U2 de USB a TTL xip de sèrie.

Interrupcions externes: 2 (interrupció 0), 3 (alarma 1), 18 (alarma 5), 19 (alarma de 4), 20 (interrupció 3) i 21 (alarma 2). Aquests pins poden ser configurats per activar una interrupció en un valor baix, un flanc ascendent o descendent, o un canvi en el valor.

PWM: 2 a 13 i 44 a 46 proporcionen PWM de 8 bits amb l'analogWrite () funció.

SCI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Aquests pins admet la comunicació SPI utilitzant la biblioteca de la SPI. Els pins SCI també es desglossen a la capçalera ICSP, que és físicament compatible amb l'U, Duemilanove i Diecimila.

USB Host: MAX3421E . L' MAX3421E informeu Arduino amb el bus SPI. Per tant, s'utilitzen els següents pins:

Digital:. 7 (RST), 50 (MISO), 51 (MOSI), 52 (SCK) Nota: Si us plau no fer servir el pin digital 7 com entrada o sortida, perquè s'utilitza en la comunicació amb MAX3421E.

No esclatat als encapçalats: PJ3 (GP_MAX), PJ6 (INT_MAX), PH7 (SS).

LED: 13 ha un predefinit LED connectat al pin digital 13. Quan el pin és un valor alt, el LED està encès, quan el passador és baix, és apagat.

TWI: 20 (SDA) i 21 (SCL) donar suport a la comunicació amb el TWI llibreria Wire. Tingueu en compte que aquests pins no estan en la mateixa ubicació que els pins a la placa Arduino Uno o en l'Arduino Duemilanove o Diecimila.

El ADK té 16 entrades analògiques, cadascun dels quals proporcionen 10 bits de resolució, és a dir, 1.024 valors diferents. Per defecte es mesuren des del sòl a 5 volts, encara que és possible canviar l'extrem superior del seu rang amb el pin AREF i la funció analogReference ().

Hi ha un parell de potes de la placa:

- AREF, voltatge de referència per a les entrades analògiques. S'utilitza amb analogReference ().

- Restablir, aquesta línia LOW per reiniciar el microcontrolador. Normalment s'utilitza per afegir un botó de reinici per escuts que bloquegen la que està a la placa.

La comunicació de ADK té un nombre d'instal·lacions per a la comunicació amb un ordinador, un altre Arduino, o altres Microcontroladors. El Atmega2560 proporciona quatre maquinari UART per a (5V) de comunicació sèrie TTL. Un ATmega8U2 en els canals de pujar a un d'aquests a través de USB i proporciona un port COM virtual amb el programari en l'equip (màquines Windows necessitaran un arxiu. inf, però les màquines OSX i Linux reconeixerà la targeta com un port COM automàticament. El programari de Arduino inclou un monitor de sèrie que permet que les dades de text simples per ser enviats cap i des de la pantalla. L'RX i TX LED al tauler parpellejarà quan les dades es transmeten a través del ATmega8U2 de xip / 16U2 i la connexió USB a l'ordinador, però no per a la sèrie comunicació en els pins 0 i 1).

Una biblioteca SoftwareSerial permet la comunicació en sèrie en qualsevol dels pins digitals del ADK.

El Atmega2560 també dóna suport TWI i la comunicació SPI. El programari d'Arduino inclou una llibreria Wire per simplificar l'ús del bus TWI, veure la biblioteca de cominucació per als detalls. Per a la comunicació SPI, utilitzeu la biblioteca de la SPI .

La interfície de host USB donada per MAX3421E IC permet l'ADK Arduino per connectar i interactuar amb qualsevol tipus de dispositiu que tingui un port USB. Per exemple, et permet interactuar amb molts tipus de telèfons, control de càmeres Cànon, que interactua amb el teclat, ratolí i controladors de jocs com el Wiimote i PS3 .

La programació del ADK es pot programar amb el programari de Arduino. Per obtenir més informació, consulteu la referència de la web Arduino i tutorials .

El ATmega2560 al Arduino ADK ve amb un carregador d'arrencada (el mateix en Mega 2560), que li permet carregar nou codi sense l'ús d'un programador de maquinari extern. Es comunica amb l'original STK500v2 protocol (de referència , arxius de capçalera C). També pot passar per alt el carregador i el programa del Microcontrolador a través del ICSP (programació en circuit sèrie) veure les instruccions per a més detalls.

El ATmega8U2 codi font del firmware està disponible a la web d' Arduino . El ATmega8U2 es carrega amb un carregador d'arrencada DFU, que pot ser activat de les següents formes:

En Rev1 juntes: connectar el pont de soldadura a la part posterior de la placa. Després reiniciar 8U2.

En rev2 o posteriors juntes: hi ha una resistència que tirant de la línia HWB 8U2/16U2 a terra, pel que és més fàcil de posar en mode DFU. A continuació, pot utilitzar el programari FLIP de Atmel (Windows) o el programador DFU (Mac OS X i Linux) per carregar un nou firmware. O pot utilitzar la capçalera ISP amb un programador extern (sobreescriure el carregador DFU). Veure aquest tutorial aportat pels usuaris per obtenir més informació.

El restabliment automàtic (Programari), en lloc de requerir polsar físicament el botó de reinici abans d'un procés de càrrega, el Arduino ADK està dissenyat d'una manera que permet que es restableix per programari que s'executa en un ordinador connectat. Una de les línies de control de flux de maquinari (DTR) de la ATmega8U2 està connectat a la línia de posada a zero del ATmega2560 a través d'un condensador de 100 nanofarads. Quan s'afirma aquesta línia de restabliment passa el temps suficient per restablir el xip. El programari d'utilitza aquesta capacitat que li permet carregar codi amb només prémer el botó de pujada a l'entorn Arduino. Això vol dir que el gestor d'arrencada pot tenir un temps d'espera més curt, com la reducció de DTR pot ser ben coordinada amb l' inici de la càrrega.

Aquesta configuració té altres implicacions. Quan el ADK està connectat ja sigui a un ordinador amb Mac OS X o Linux, es restableix cada vegada que es realitza una connexió a la mateixa des del programari (a través de USB). Per al següent mig segon o menys, el gestor d'arrencada s'executa en el ADK. Mentre que està programat per ignorar dades amb errors (és a dir, res, a més d'una pujada del nou codi), es interceptarà els primers bytes de dades enviades a la targeta després d'obrir una connexió. Si un dibuix s'executa a la placa rep la configuració d'una sola vegada o d'altres dades quan s'inicia per primera vegada, assegureu-vos que el programari amb el qual es comunica s'espera un segon després d'obrir la connexió i abans d'enviar les dades.

L'ADK conté una traça es pot tallar per desactivar l'auto-reset. Els coixinets a banda i banda de la traça es poden soldar junts per tornar a habilitar. Ha marcat "RESET-ES". També pot ser capaç de desactivar l'auto-reset connectant una resistència de 110 ohm de 5V a la línia de reinici, vegeu aquest fil del fòrum per a més detalls.

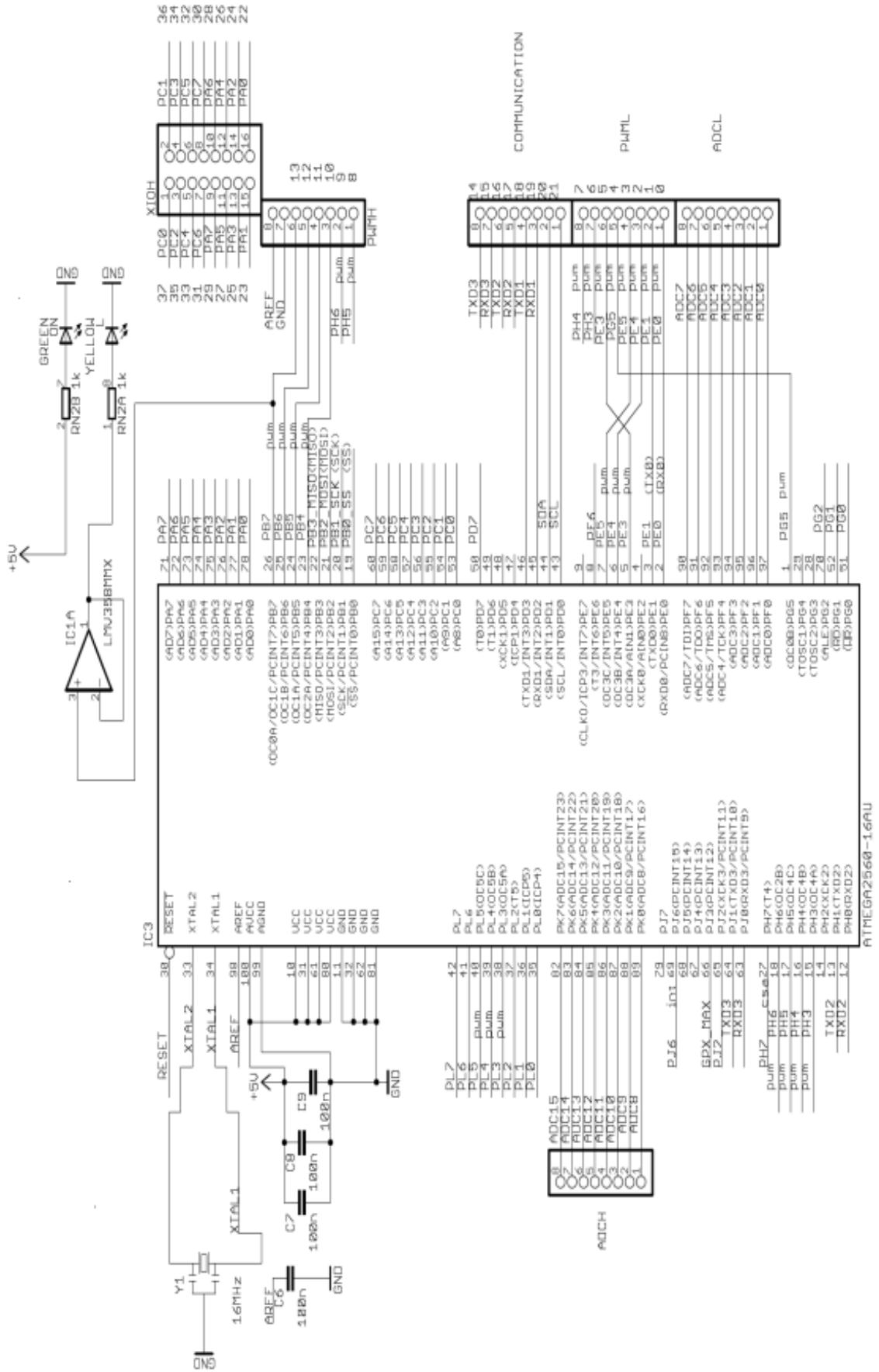
La protecció de sobrecorrent USB d'ADK té una POLYFUSE resetejable que protegeix els ports USB del teu ordinador de curtcircuits i sobretensions. Encara que la majoria dels ordinadors proporcionen la seva pròpia protecció interna, el fusible proporciona una capa addicional de

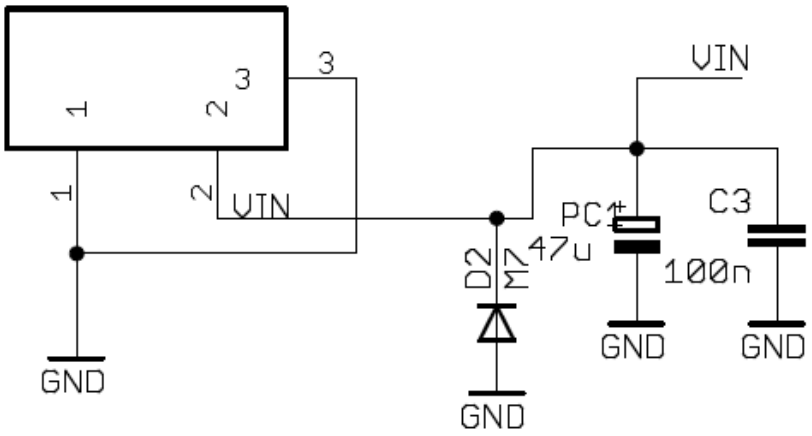
protecció. Si s'aplica més de 500 mA al port USB, el fusible es trencarà automàticament la connexió fins que s'elimina el curtcircuit o una sobrecàrrega.

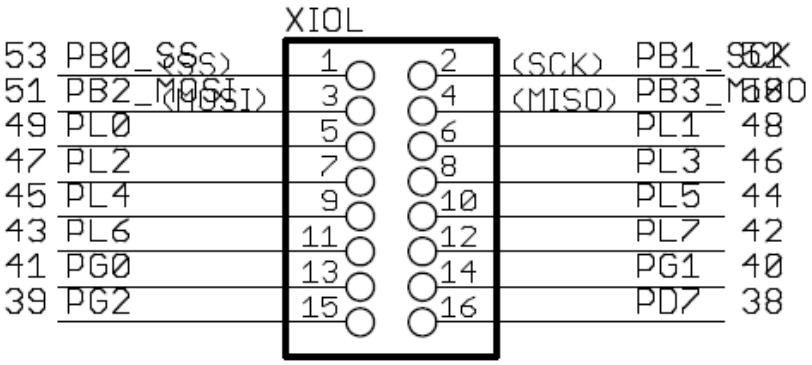
Les característiques físiques de la placa són: la longitud màxima i l'amplada de la PCB ADK són 4 i 2,1 polzades, respectivament, amb el connector USB i el connector d'alimentació que s'estén més enllà de la dimensió anterior. Tres orificis dels cargols que la Junta pugui fixar-se a una superfície o caixa. Tingueu en compte que la distància entre els pins digitals 7 i 8 és de 160 mil·lèsimes de polzada (0,16 "), no és un múltiple parell de mil·lèsimes de polzada espaiament dels altres perns 100.

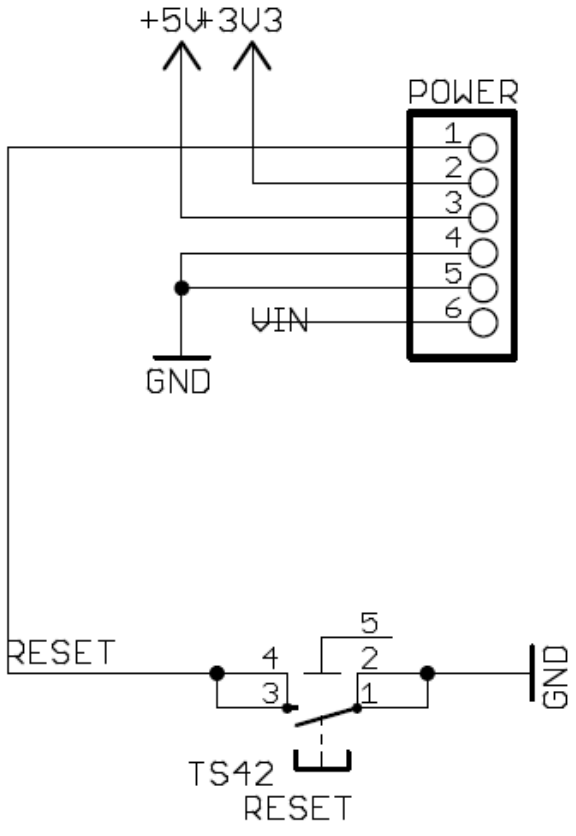
El ADK està dissenyat per a ser compatible amb la majoria dels protectors dissenyats per l'U, Diecimila o Duemilanove. Pines digitals 0 a 13 (i les adjacents pins AREF i GND), entrades analògiques de 0 a 5, la capçalera d'energia, i la capçalera ICSP estan en punts equivalents. A més, la UART principal (sèrie) es troba en els mateixos pins (0 i 1), igual que les interrupcions externes 0 i 1 (pins 2 i 3, respectivament). SCI està disponible a través de la capçalera ICSP tant en la ADK i Duemilanove/Diecimila. *Tingueu en compte que ^{2} C no es troba en les mateixes patilles del ADK (20 i 21) com el Duemilanove / Diecimila (entrades analògiques 4 i 5).*

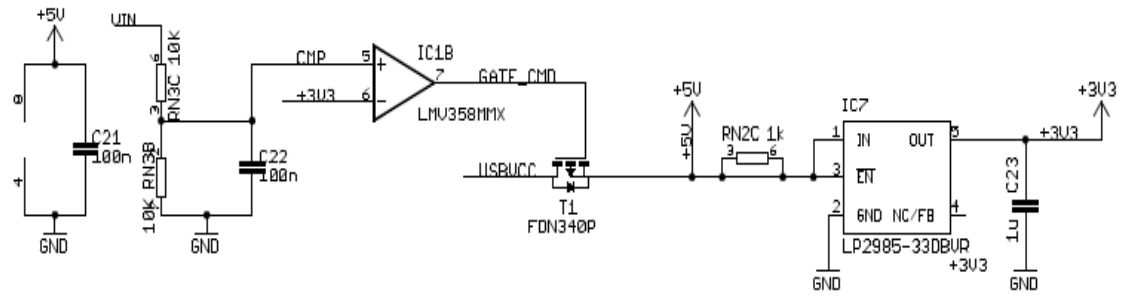
1.2.Desglossament de l'esquema placa Arduino Atmega ADK ANDROID

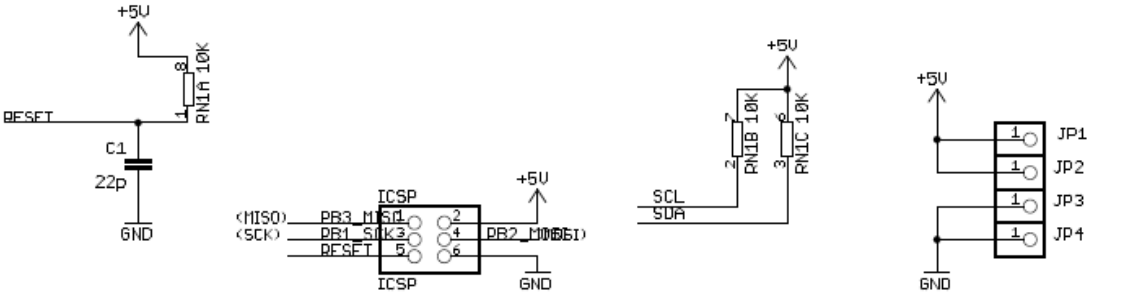


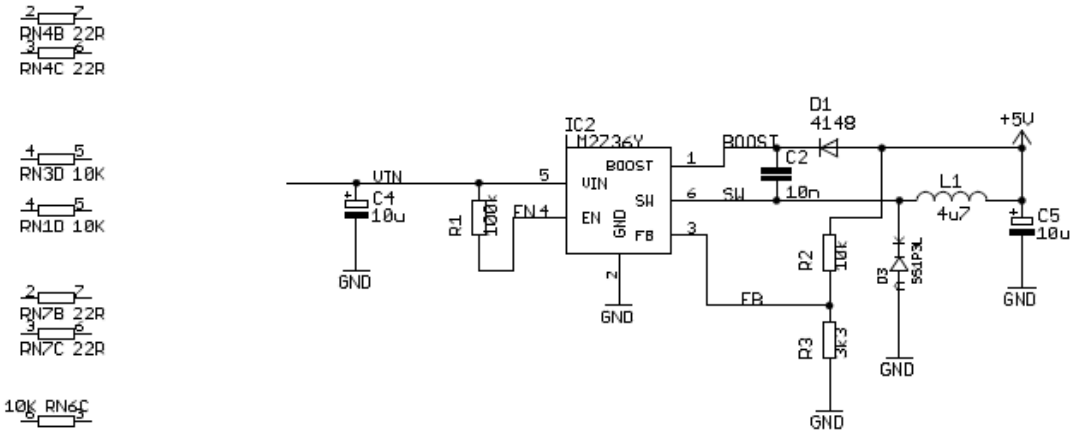

3
1
2
3
1
2
VIN
GND
VIN
PC1
47u
C3
100n
D2
M7
GND
GND
GND

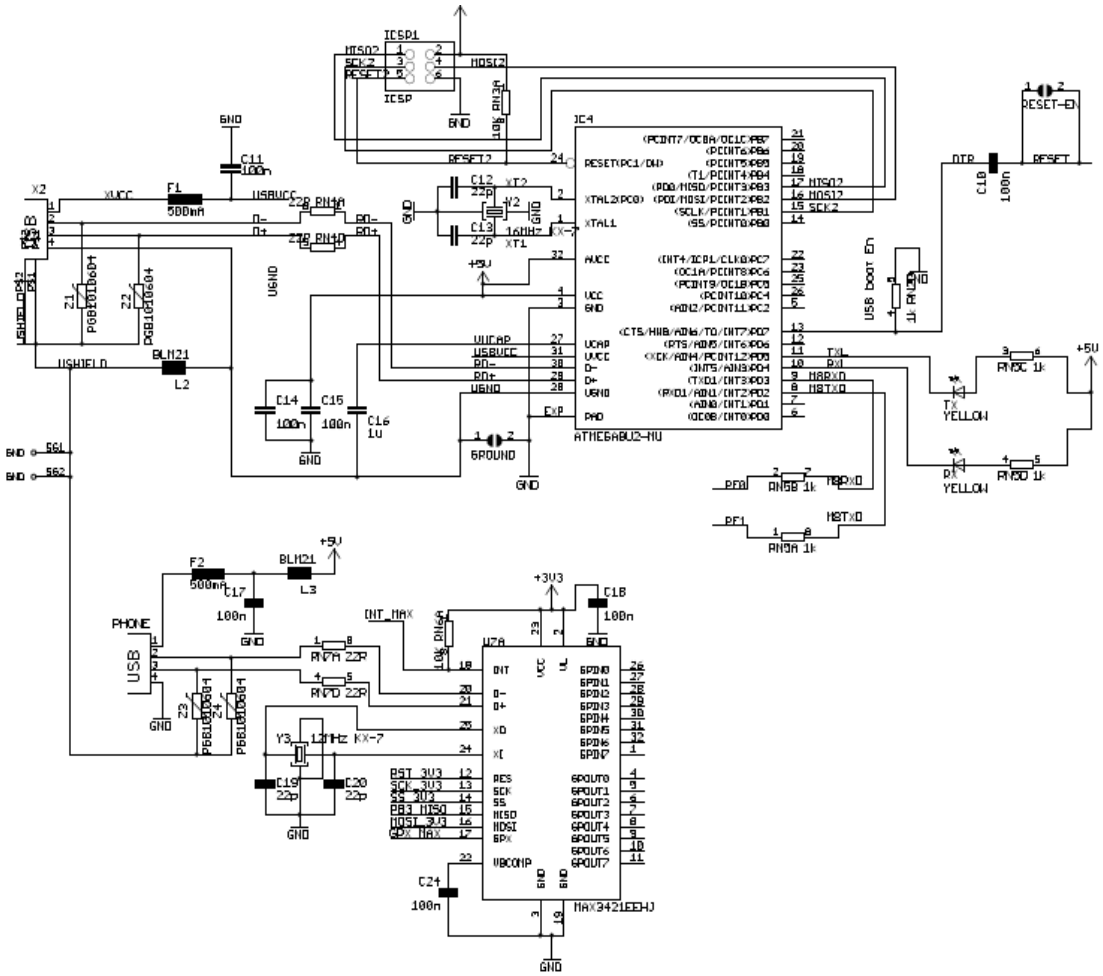

XIOL
53 PB0_SS
51 PB2_MOSI
49 PL0
47 PL2
45 PL4
43 PL6
41 PG0
39 PG2
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
(SCK)
(MISO)
PB1_SCK
PB3_MISO
PL1 48
PL3 46
PL5 44
PL7 42
PG1 40
PD7 38


+5V
3V3
POWER
1
2
3
4
5
6
VIN
GND
RESET
4
3
5
2
1
GND
TS42
RESET

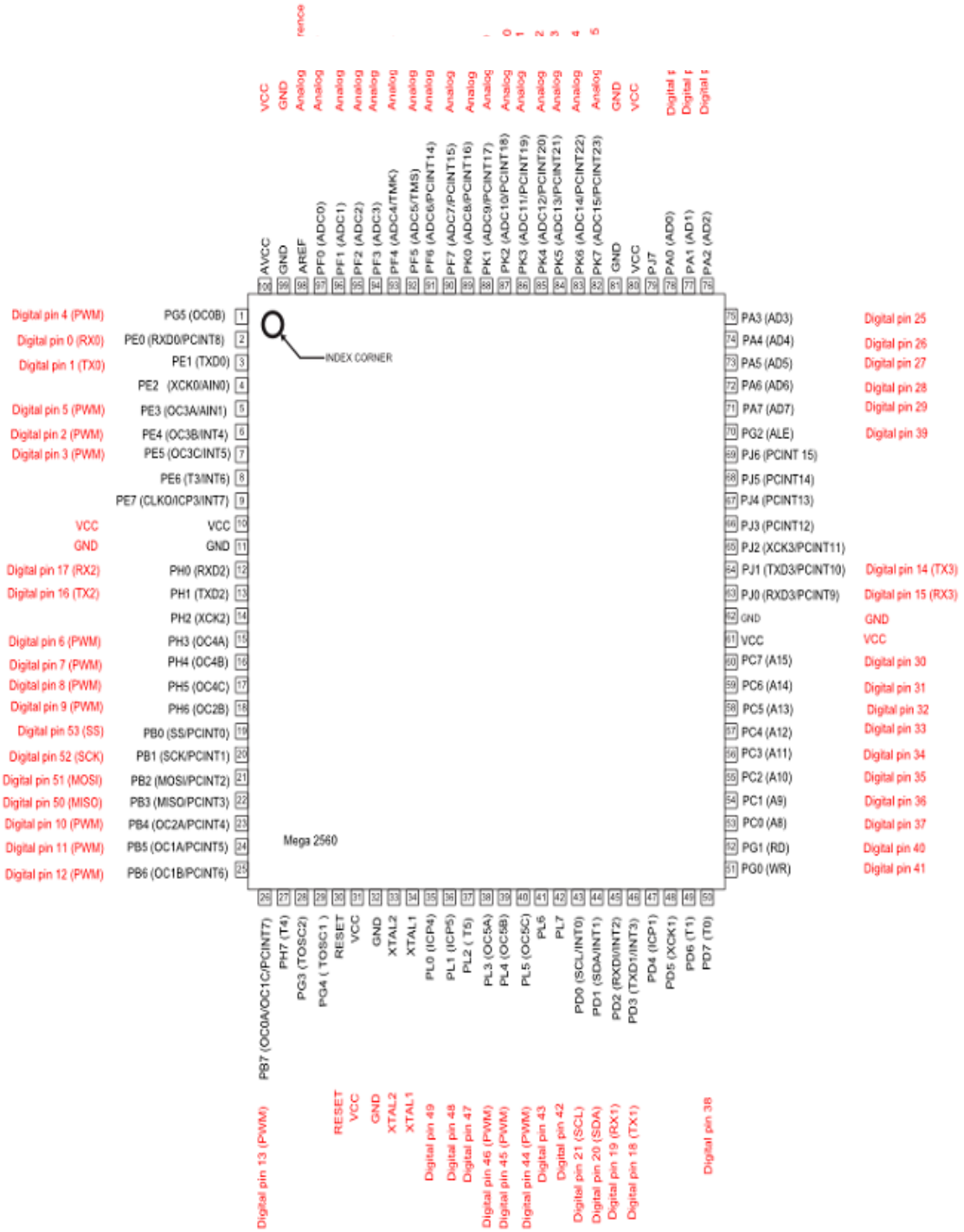

+5V
VIN
RN3C 10K
CMP
IC1B
+3V3
GATE_CMD
LMV358MMX
C21
100n
RN3B
10K
C22
100n
GND
GND
USBVCC
T1
FDN340P
+5V
RN2C 1k
IC7
IN
OUT
EN
GND NC/FB
LP2985-33DBVR
+3V3
+3V3
C23
1u
GND
GND


+5V
RESET
RN1A 10K
C1
22p
GND
ICSP
(MISO)
(SCK)
RESET
ICSP
GND
+5V
SCL
SDA
RN1B 10K
RN1C 10K
+5V
JP1
JP2
JP3
JP4
GND


RN4B 22R
RN4C 22R
RN3D 10K
RN1D 10K
RN7B 22R
RN7C 22R
10K RN6C
IC2
M2236Y
BOOST
VIN
SW
EN
GND
FB
D1
4148
C2
10n
L1
4u7
+5V
C5
10u
GND
VIN
C4
10u
R1
100K
EN4
GND
GND
FB
R2
10K
R3
3K3
D3
SS1P3L
GND
GND


ATMEGA8U2-MU
MAX3421EEHJ
74LVC16125DGK
+3V3
+5V
GND
RST_5V
SCK_3V3
SS_3V3
MOSI_3V3
INT_MAX
GPX_MAX
PHONE
USB

1.3.Mostra de l'assignació de pins per al Atmega 2560



1.4.Taula d'assignació dels pins d'Arduino Mega 2560.

Nombre de Pin	Pin Nom	Assignat Nom Pin
1	PG5 (OC0B)	Pin digital 4 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	Digital pin 5 (PWM)
6	PE4 (OC3B/INT4)	Pin 2 Digital (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
8	PE6 (T3/INT6)	
9	PE7 (CLKO/ICP3 / INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Pin digital 17 (RX2)
13	PH1 (TXD2)	Pin digital 16 (TX2)
14	PH2 (XCK2)	
15	PH3 (OC4A)	Digital pin 6 (PWM)
16	PH4 (OC4B)	Digital pin 7 (PWM)
17	PH5 (OC4C)	Pin digital 8 (PWM)
18	PH6 (OC2B)	Pin setembre Digital (PWM)
19	PB0 (SS/PCINT0)	Pin digital 53 (SS)
20	PB1 (SCK/PCINT1)	Pin 52 Digital (SCK)
21	PB2 (MOSI/PCINT2)	Pin digital 51 (MOSI)
22	PB3 (MISO/PCINT3)	Pin digital 50 (MISO)
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)
24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)
25	PB6 (OC1B/PCINT6)	Pin digital 12 (PWM)
26	PB7 (OC0A/OC1C / PCINT7)	Pin digital 13 (PWM)
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	
30	REAJUST	REAJUST
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2
34	XTAL1	XTAL1
35	PL0 (ICP4)	Pin digital 49
36	PL1 (ICP5)	Pin digital 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Pin digital 46 (PWM)
39	PL4 (OC5B)	Pin digital 45 (PWM)
40	PL5 (OC5C)	Pin digital 44 (PWM)
41	PL6	Pin digital 43
42	PL7	Pin digital 42
43	PD0 (SCL/INT0)	Pin 21 Digital (SCL)
44	PD1 (SDA/INT1)	Pin digital 20 (SDA)
45	PD2 (RXDI/INT2)	Pin digital 19 (RX1)

46	PD3 (TXD1/INT3)	Pin digital 18 (TX1)
47	PD4 (ICP1)	
48	PD5 (XCK1)	
49	PD6 (T1)	
50	PD7 (T0)	Pin digital 38
51	PG0 (WR)	Pin digital 41
52	PG1 (RD)	Pin digital 40
53	PC0 (A8)	Pin digital 37
54	PC1 (A9)	Pin digital 36
55	PC2 (A10)	Pin digital 35
56	PC3 (A11)	Pin digital 34
57	PC4 (A12)	Pin digital 33
58	PC5 (A13)	Pin digital 32
59	PC6 (A14)	Pin digital 31
60	PC7 (A15)	Pin digital 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Pin digital 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Pin digital 14 (TX3)
65	PJ2 (XCK3/PCINT11)	
66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCInt 15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Pin digital 29
72	PA6 (AD6)	Pin digital 28
73	PA5 (AD 5)	Pin digital 27
74	PA4 (AD4)	Pin digital 26
75	PA3 (AD3)	Pin digital 25
76	PA2 (AD2)	Pin digital 24
77	PA1 (AD1)	Pin digital 23
78	PA0 (AD0)	Pin digital 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15/PCINT23)	Pin analògic 15
83	PK6 (ADC14/PCINT22)	Pin analògic 14
84	PK5 (ADC13/PCINT21)	Pin analògic 13
85	PK4 (ADC12/PCINT20)	Pin analògic 12
86	PK3 (ADC11/PCINT19)	Pin analògic 11
87	PK2 (ADC10/PCINT18)	Pin analògic 10
88	PK1 (ADC9/PCINT17)	Pin analògic setembre
89	Pk0 (ADC8/PCINT16)	Pin analògic 8
90	PF7 (ADC7)	Pin analògic 7
91	PF6 (ADC6)	Pin analògic 6
92	PF5 (ADC5 / TMS)	Pin analògic 5
93	PF4 (ADC4 / TMK)	Pin analògic 4
94	PF3 (ADC3)	Pin analògic 3
95	PF2 (ADC2)	Pin analògic 2
96	PF1 (ADC1)	Pin analògic 1

97	PF0 (ADC0)	Pin analògic 0
98	AREF	Referència analògica
99	GND	GND
100	AVCC	VCC

2.Arduino Shield TFT Tàctil color 2'8"

Es tracta d'una pantalla tàctil resistiva multi funcional Arduino/Seeeduino/Arduino compatible amb l'Arduino i l'Atmega. Pot ser utilitzat com a dispositiu de visualització o bloc de dibuix. En comparació amb la versió anterior, 2.8" TFT Touch Protector V1.0, s'ha millorat el controlador de pantalla amb un xip professional, ILI9341 conductor, proporcionant estalvi de pins SCI comunicació sense sacrificar la velocitat de transmissió de dades. Amb un mòdul de targeta SD integrat també en aquest escut, aquest escut es reserva gran espai per a altres expansions per al seu projecte.

Característiques

Gran pantalla per a una experiència fàcil i còmoda

Backlight controlable a través de la programació

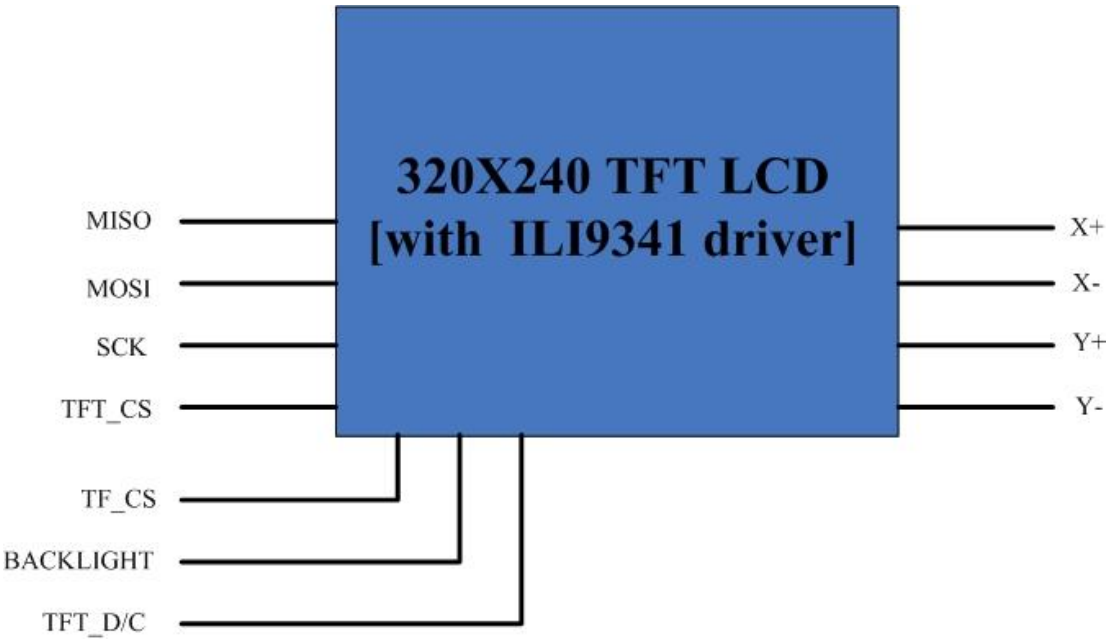
65.535 colors.

Mètode de comunicació Pin d'estalvi de SCI

Nom	Min	Típic	Max	Unitat
Voltatge	4.5	5	5.5	VDC
Corrent	/	/	250	mA
Panell LCD Mida	2.8	polzada		
Angle de visió	60 ~ 120	Graus		
Resolució	320x240	/		
LCD a color	65k	/		
Tipus de llum de fons	LED	/		
Controlador LCD IC	ILI9341	/		
Tipus d'interfície	SCI	/		
Pantalla tàctil	Pantalla tàctil resistiva de 4 fils	/		
Àrea activa	43.2 * 57.3	mm		
Descàrrega de contacte ESD	± 4	KV		
De descàrrega d'aire EDS	± 8	KV		
Dimensió	72.5x54.7x18	mm		
Pes	24 ± 2	g		

No premeu amb massa força a la pantalla per evitat que pugui provocar una distorsió de la pantalla.

Pins d'ús al Arduino



Pines utilitzats per al control de la pantalla TFT:

D4: TF_CS, seleccioneu pin d'entrada de targeta SD

D5: TFT_CS, seleccioneu pin d'entrada de xip TFT

D6:, pin de control TFT de dades / TFT_D / C

D7: LLUM DE FONS, pin de control de contrallum TFT

Pines utilitzats per a la interfície de SCI

D10: SCI de selecció de xip

D11: pin de dades SCI

D12: pin de dades SCI

D13: pin rellotge serial SCI

Pines utilitzats per a la funció Touch

A0 - I-pin d'entrada de la pantalla tàctil.

A1 - X-pin d'entrada de la pantalla tàctil.

A2 - Touch Screen I + pin d'entrada.

A3 - pantalla tàctil X pin d'entrada +.

Instal·lació del maquinari

Connecteu l'escut en el seu Arduino i Connecteu la targeta al PC mitjançant un cable USB ..

Instal·lació del programari

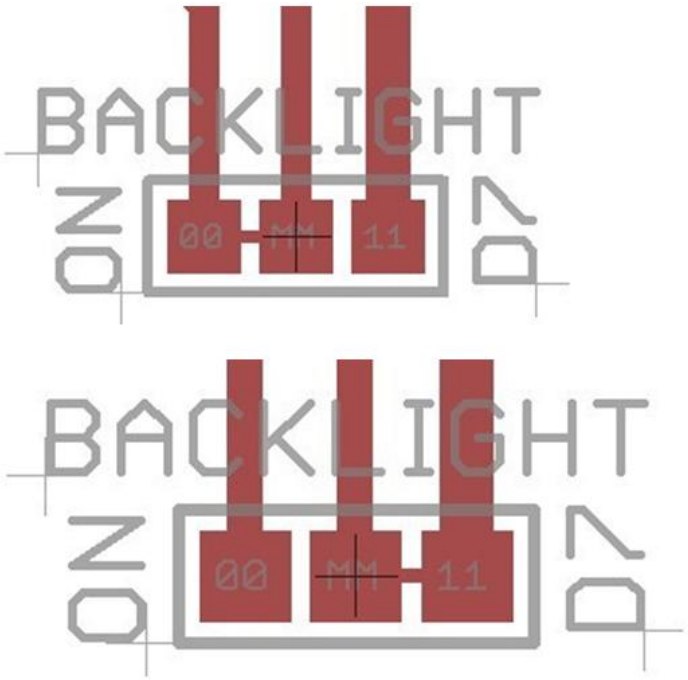
1. Descarregueu la Biblioteca SeeedTFTV2.0 per Arduino 1.0 i SeeedTouchScreen Biblioteca d' 1.0

2. descomprimir a l'arxiu de biblioteques de Arduino IDE pel camí: .. \ arduino-1.0 \ llibreries

4. Si vols començar a patrons de pintura, només ha de prémer el botó de reinici o apagat i encesa de l'Escut.

Control de llum de fons

En l'estat per defecte, la il·luminació de la pantalla TFT tàctil Escut està connectat a 5V. En altres paraules, la llum de fons s'encén quan el poder de la pantalla. De fet, vostè pot utilitzar un port per controlar el seu estat donant el port de control alt / baix nivell. A la part posterior, que és la següent: Ara el que necessita fer això handling: Desconnecteu la connexió amb el pin, i connecti el pin retroil·luminació amb pin D7. De cource, el control del seu estat és tan simple com controlar un LED.



3.Arduino Shield GSM-GPRS Sparkfun SM5100B

Aquest document ofereix una visió general del mòdul SM5210: una targeta petita, d'un sol costat, quad-band GSM

Mòdul de 850/EGSM 900/DCS 1800/PCS 1900, a punt per a la integració en diversos tipus de telèfons sense fil Fixi altres dispositius sense fils.

3.1. SISTEMA DIAGRAMA DE BLOCS

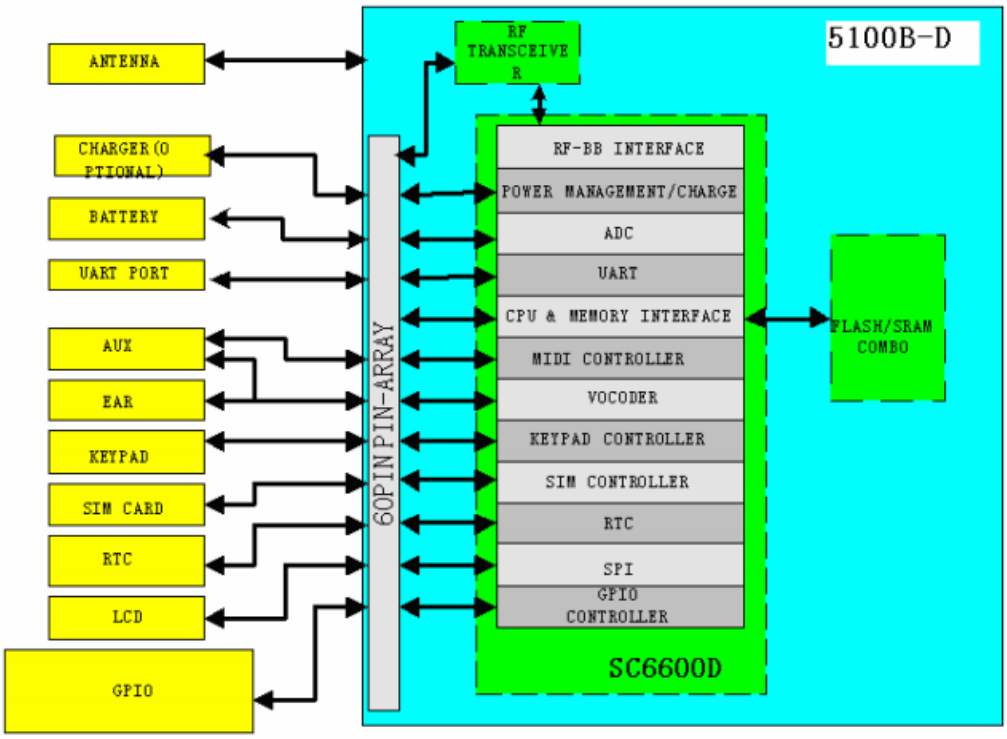


Figure 1: System block diagram

Figura 1: diagrama de blocs del sistema

3.2. DESCRIPCIÓ DEL PRODUCTE

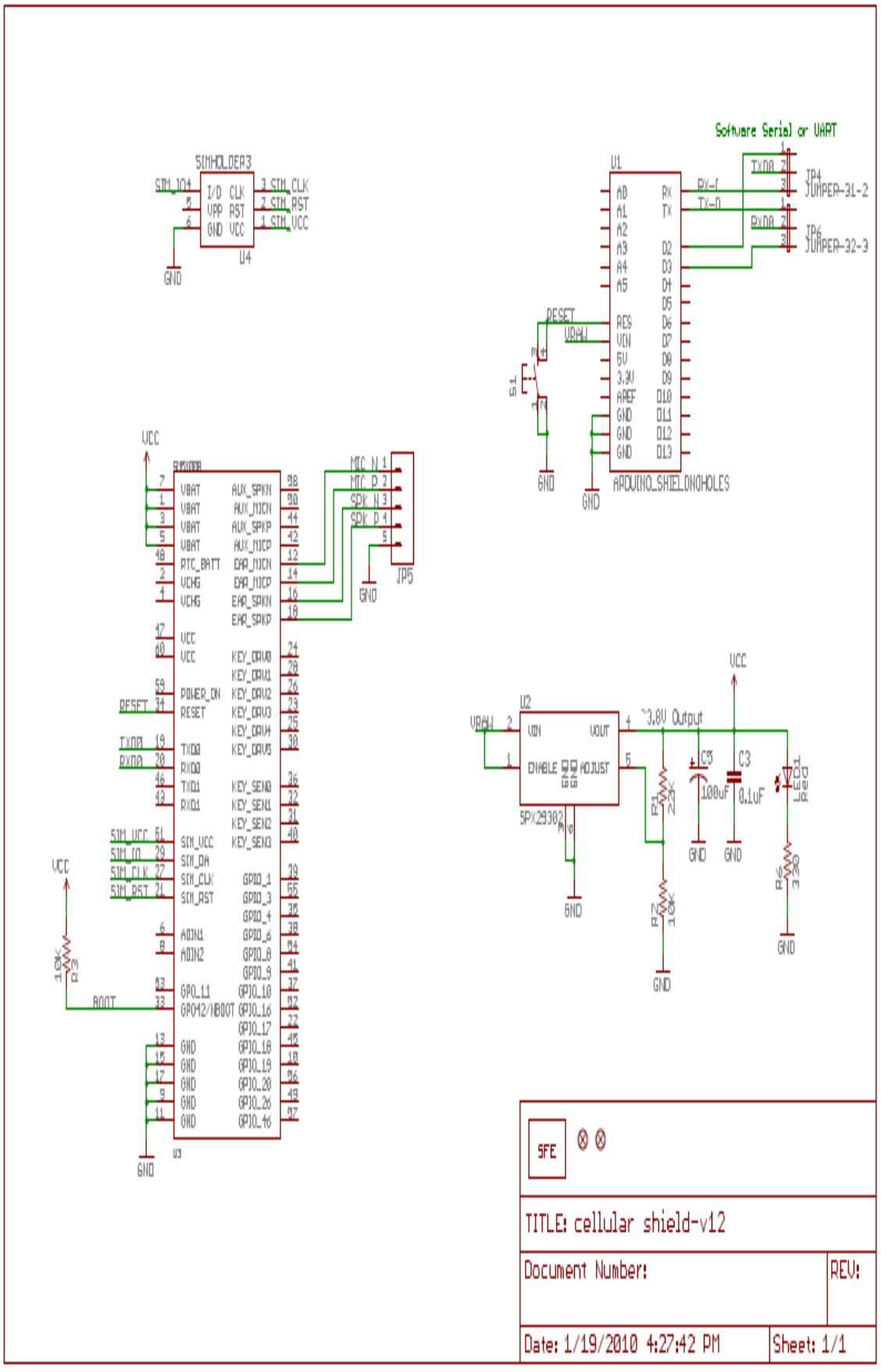
Temperatura de funcionament	-10 ° C a +55 ° C
Dimensions físiques	35.0X39.0X2.9 mm
Connexió	60 Pins
Alimentació	3,3 V a 4,2 V gamma, típica 3.6V.
Consum d'energia mode apagat	<100uA
Consum d'energia mode suspensió	<2.0mA
Consum d'energia mode d'espera	<7.0mA (mitjana)
Consum d'energia mode comunicació	350 mA (mitjana, GSM)
Consum d'energia mode comunicació	2000mA (màxim típic en TX ranura, GSM)
Gestió de càrrega de la bateria Li-ió i interface (OPCIONAL)	S'inclou la gestió de càrrega de la bateria Li-ió. El carregador interfície està disponible al connector de 60 pins. (Només per 3.7V Li-ió Bateria)
Les bandes de freqüències	EGSM900+GSM850+DCS1800+ PCS1900
Targeta SIM compatibles	3V/1.8V targeta SIM. (Reconeix automàtic)
Potencia	4 (2W) per EGSM900/GSM850 Classe 1 (1W) per DCS1800/PCS1900
Es proporciona interfície de teclat	interfície de teclat 4x6
UART0 amb control de flux	2-WIRE UARTFins 460 kbps
Interfície de LCD	Suport interfície estàndard SPI,

3.3.Llistat de pins placa Shield GSM

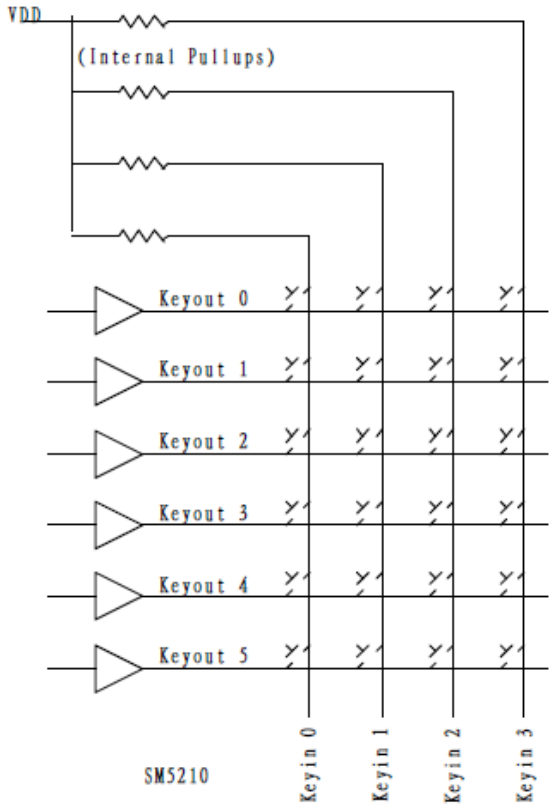
Nº Pin	Nom	Descripció
1	VBAT	Voltatge bateria
2	VCHG	Carrega Voltatge addicional
3	VBAT	Voltatge bateria
4	VCHG	Carrega Voltatge addicional
5	VBAT	Voltatge Bateria
6	ADIN1	Detecta falta Volts de la bateria
7	VBAT	Voltatge Bateria
8	ADIN2	Detecta falta Volts de la bateria
9	GND	Terra Digital
10	GPIO19/UO_CTS/JTAG TMS	Esborrar enviament
11	GND	Terra Digital
12	EAR_MICN	Micròfon entrada negativa
13	GND	Terra Digital
14	EAR_MICP	Micròfon entrada positiva
15	GND	Terra Digital
16	EAR_SPKN	Altaveu sortida negativa
17	GND	Terra Digital
18	EAR_SPKP	Altaveu sortida positiva
19	TXD0	Transmissió de dades

20	RXD0	Recepció de dades
21	SIM RST	Resset SIM
22	GPIO17/U0_RTSN/JTAG_DI	Sol·licitar enviament
23	KEY_DRV3	Sortida teclat
24	KEY_DRV0	Sortida teclat
25	KEY_DRV4	Sortida teclat
26	KEY_DRV2	Sortida teclat
27	SIM_CLK	Rellotge SIM
28	KEY_DRV1	Sortida teclat
29	SIM_DA	SIM Serial Data
30	KEY_DRV5	Sortida teclat
31	KEY_SEN2	Entrada teclat
32	KEY_SEN1	Entrada teclat
33	GPO42/LCD_DA/NBOOT	LCD_DATA_SPI
34	RESSET	Resset
35	GPIO4/HF_CTRL	GPIO
36	KEY_SEN0	Entrada teclat
37	GPIO10/CHARGE IN	GPIO
38	GPIO6/LCD_CLK	GPIO
39	GPIO1/LCD_RST	GPIO
40	KEY_SEN3	Entrada teclat
41	GPIO9/LCD_RS	LCD_RS_SPI
42	AUX_MICP	Auxiliar micròfon entrada positiva
43	RXD1	Recepció dades
44	AUX_SPKP	Altaveu auxiliar entrada positiva
45	GPIO18/U0_DSR/JTAG_CK	GPIO18
46	TXD1	Transmissió dades
47	VCC	Alimentació auxiliar
48	RTC_BAT	RTCVDD 3.0V
49	GPIO26/LED_BK_CTRL	GPIO
50	AUX_MICN	Auxiliar micròfon entrada negativa
51	SIM_VCC	SIM alimentació 3V
52	GPIO16/U0_DTR/JTAG_DO	Terminal llest
53	GPO11/CS3	ARM extern CS3
54	GPIO8/LCD CS	LCD_CS_SPI
55	GPIO3	GPIO
56	GPIO20Ç7JTAG_RST/PWMB	JTAT resset
57	GPIO46	GPIO
58	AUX_SPKN	AUXILIAR
59	POWER_ON	Alimentació boto
60	VCC	Alimentació auxiliar

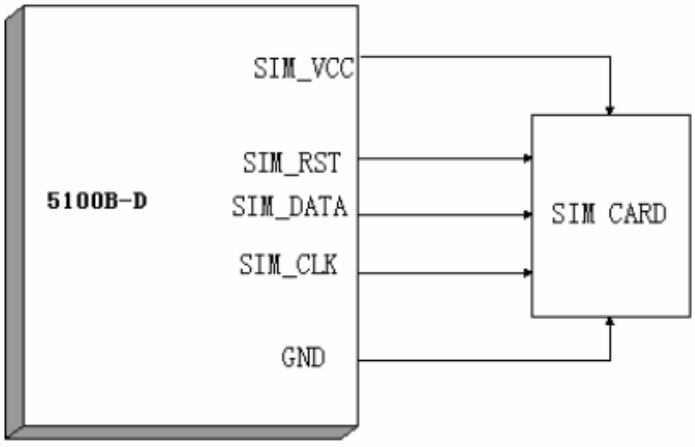
3.4.Esquema placa Shield GSM



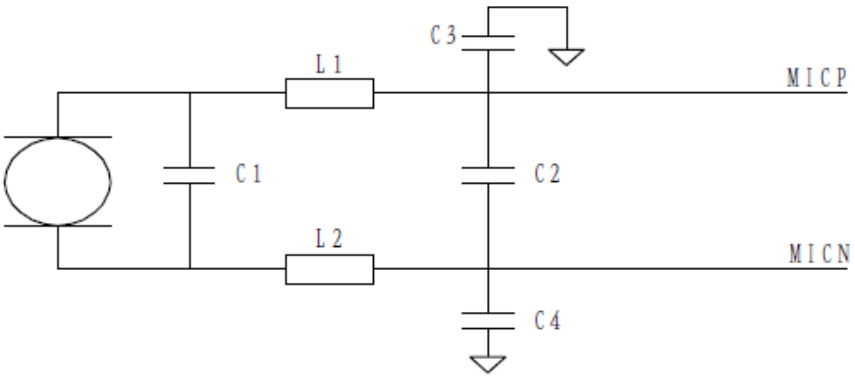
3.5.Esquema cablejat d'un teclat extern



3.6.Esquema cablejat targeta SIM



Esquema cablejat micròfon extern.

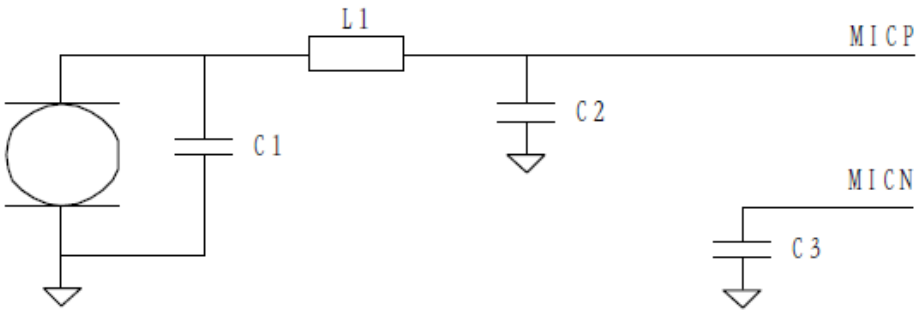


C1= 22pF a 47 pF

C2=C3=C4=47pF a 100pF

L1=L2=100nH

Es recomana realitzar la connexió anterior.



C1= 22pF a 47 pF

C2=C3=47pF a 100pF

L1=100nH

3.7.Esquema cablejat Altaveu.

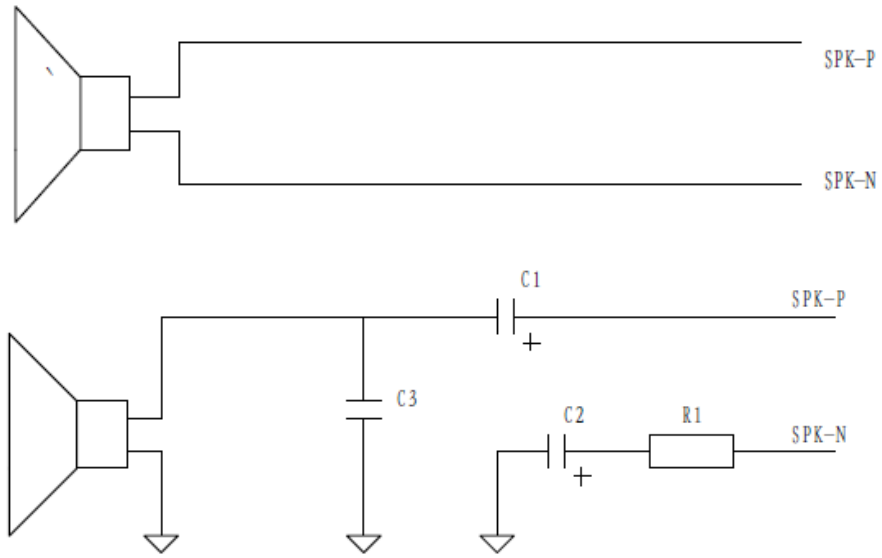


Figure 8: SPK output (single-end output)

C1=C2=4,7µK a 47 µF

C3=33pF a 100pF

R1= resistència altaveu.

3.8.Esquema cablejat RTC bateria

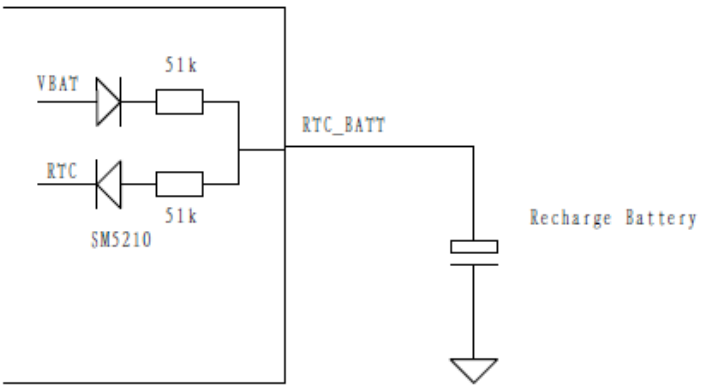


Figure 9: RTC battery connection

3.9.Esquema cablejat carregador de bateria

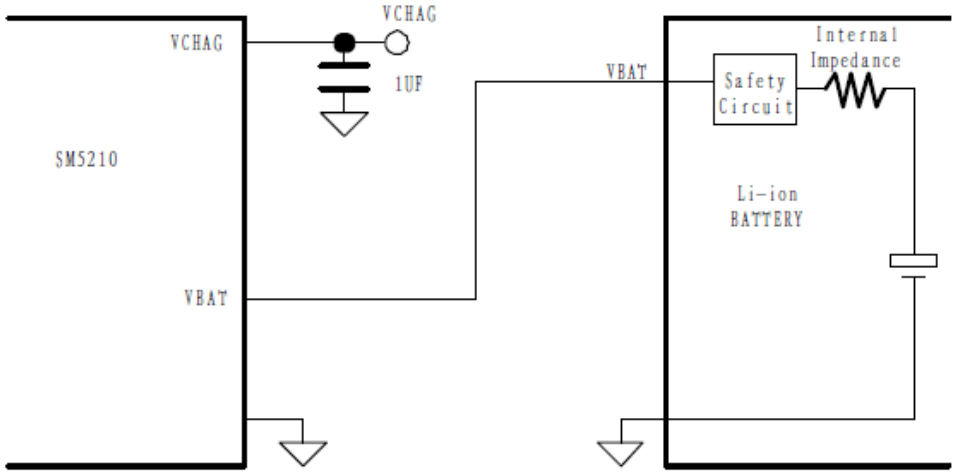


Figure 11: Battery Charging Connection

4.Programa enviat a l'Arduino Atmega ADK.

```
/*

Projecte final de carrera Enginyeria Tècnica Industrila. Especialitat en Electrònica Industrila. Albert Boix Comajuan. Universitat de Vic Any 2013. Sistema Control Consum d'aigua Automàtic.

*/

//_____________________________INICIALITZAR LLIBRERIES_____________________________

#include <EEPROM.h>

#include <SerialGSM.h>

#include <SoftwareSerial.h>

SerialGSM cell(48,49);

#include <TFTv2.h>

#include <SPI.h>

#include <stdint.h>

#include <SeeedTouchScreen.h>

#include <Wire.h>

#include "DS1307.h"

#define Backlignt 7

DS1307 clock;

#if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)

#define YP A2

#define XM A1

#define YM 54

#define XP 57

#elif defined(__AVR_ATmega32U4__)

#define YP A2

#define XM A1

#define YM 18

#define XP 21
```

```
#else
#define YP A2
#define XM A1
#define YM 14
#define XP 17
#endif
#define TS_MINX 116*2
#define TS_MAXX 890*2
#define TS_MINY 83*2
#define TS_MAXY 913*2
TouchScreen ts = TouchScreen(XP, YP, XM, YM);

//_____________________________________________________________________________

//_____________________________DEFINIR VARIABLES_____________________________

int y=0;  //  Variable per saber si toquem la pantalla cordenada X.
int x=0; //  Variable per saber si toquem la pantalla cordenada Y.
int sensorpin = 18;//  Assignem el sensorpin el pin 1.
int alarmapin = 19;//  Assignemt el pim 6 a l'alarma.
int innterpantalla = 22;
int relevalvula = 16;//  Assignemt el pim 6 al rele.
int sensorstat = 0;// creem variable per gurdar l'estat del pulsador i la posem a 0.
int alarmastat =0;//  Creem una variable per guardar l'estat del l'alarma i la posem a 0.
int sortida=0;
int mes=1;
int alarmalitres=0;
int consumlitres=0;
long comptadorlitres=0;
```

```
int alarmagener=0;
int alarmafebrer=0;
int alarmamarc=0;
int alarmaabril=0;
int alarmamaig=0;
int alarmajuny=0;
int alarmajuliol=0;
int alarmaagost=0;
int alarmasetembre=0;
int alarmaoctubre=0;
int alarmanovembre=0;
int alarmadesembre=0;
int alarmames=1;
int pantalles=0;
int minut=00;
int start=1;
int diames=1;
int pantalla=0;
int principal=1;
int programacio=0;
int consums=0;
int dataihora=0;
int ressetalarma=0;
int manual=0;
int al=0;
int month1=1;
int progdm=0;
int progm=0;
int proga=0;
```

```
int progminut=0;
int proghora=0;
int data=0;
int tempstrans=0;
int cronominut=0;
int tempsalarma=5;
int progtempsalarma=0;
float ecxessconsums=0;
int progecxessconsums=0;
float resultat1=0;
float resultat2=0;
float resultat3=0;
int a=0;
int b=0;
int sms=0;
int tf1=0;
int tf2=0;
int tf3=3;
int tf4=4;
int tf5=0;
int tf6=0;
int tf7=0;
int tf8=0;
int tf9=0;
int tf10=0;
int tf11=0;
int tf12=0;
int tf13=0;
int d1=0;
```

```
int d2=0;
int d3=0;
int d4=0;
int d5=0;
int d6=0;
int d7=0;
int d8=0;
int mem1=41;
int mem2=42;
int mem3=43;
int mem4=44;
int mem5=45;
int mem6=46;
int mem7=47;
int mem8=48;
char an;
char bn;
char cn;
char dn;
char en;
char fn;
char gn;
char hn;
char in;
char jn;
char kn;
char ln;
char mn;
int progtelf1=0;
```

```
int progtelf2=0;

int progtelf3=0;

int progtelf4=0;

int progtelf5=0;

int progtelf6=0;

int progtelf7=0;

int progtelf8=0;

int progtelf9=0;

int progtelf10=0;

int progtelf11=0;

int progtelf12=0;

int progtelf13=0;

int n=0;

int d=0;

int numtf=13;

char Str4[12] = "";

Point p = ts.getPoint();


//_____________________________FUNCIÓ INICIALITZAR_____________________________

// s'executa al iniciar l'arduino.


void funcioinicialitzant()

                    {

                     Tft.fillRectangle(0, 0, 239,319,BLACK);// BORRAR PANTALLA

                     //DIBUIX RECTANGLE NEGRE PER BORRAR TEXT ANTERIOR

                     Tft.fillRectangle(0, 0, 239,15,BLACK);

                     // TITUL, CAPÇALERA PANTALLA

                     Tft.drawString("inicialitznt",5,5,1,CYAN);

                     //DIBUIX RECTANGLE BLAU
```

```
                Tft.fillRectangle(0, 15, 239,300,BLUE);
                // TEXT PART BLAVA
                Tft.drawString("Inicialitzant:",10,40,2,WHITE);
                Tft.drawString("Sistema Control",10,70,2,WHITE);
                Tft.drawString("Consum d'Aigua",10,100,2,WHITE);
                Tft.drawString("Automatic",10,130,2,WHITE);
                Tft.drawString("Versio 2-013",10,160,2,WHITE);
                // RECTANGLE VERD
                Tft.fillRectangle(0, 300, 239,55,BLACK);
                //Tft.drawString("PROGRAMACIO",5,265,2,BLACK);
                // PEU DE PANTALLA
                Tft.drawString("ALBERT BOIX 2013",0,305,2,WHITE);
                principal=1;
                programacio=1;
                y=0;
                x=0;
                delay(4000);
                pantalla++;
               }

//_____________________________________________________________________________

//_____________________________FUNCIÓ ALINEAR_____________________________

//S'executa quan iniciem el arduino.
void funcioalinear()
          {
           Tft.fillRectangle(0, 0, 239,319,BLACK);
           pantalla++;
```

```
Tft.drawString("Alineant Pantalla:",10,20,2,WHITE);
delay (1000);
for(int r=0;r<115;r=r+2)
                    {
                     Tft.drawCircle(119,160,r,random(0xFFFF));
                     }
Tft.fillRectangle(0, 0, 239,319,BLUE);
Tft.drawString("Recorda",10,40,2,WHITE);
Tft.drawString("Comprobar",10,70,2,WHITE);
Tft.drawString("Data, Hora i ",10,100,2,WHITE);
Tft.drawString("Numero de Telefon",10,130,2,WHITE);
delay(1000);
Point p = ts.getPoint();
p.x = map(p.x, TS_MINX, TS_MAXX, 0, 240);
p.y = map(p.y, TS_MINY, TS_MAXY, 0, 320);
}

//_____
_____FUNCIÓ VISUALITZAR MÉS PROGRAMACIÓ_____
void funciomesprog()
                {
                 switch (clock.month)
                                {
                                 case 1:
                                 Tft.drawString("GENER",135,75,2,BLACK);
                                 break;

                                 case 2:
                                 Tft.drawString("FEBRER",135,75,2,BLACK);
```

```
break;

case 3:
Tft.drawString("MARC,",135,75,2,BLACK);
break;

case 4:
Tft.drawString("ABRIL",135,75,2,BLACK);
break;

case 5:
Tft.drawString("MAIG",135,75,2,BLACK);
break;

case 6:
Tft.drawString("JUNY",135,75,2,BLACK);
break;

case 7:
Tft.drawString("JULIOL",135,75,2,BLACK);
break;

case 8:
Tft.drawString("AGOST",135,75,2,BLACK);
break;

case 9:
Tft.drawString("SETEMBRE",135,75,2,BLACK);
break;
```

```
                    case 10:
                    Tft.drawString("OCTUBRE",135,75,2,BLACK);
                    break;

                    case 11:
                    Tft.drawString("NOVEMBRE",135,75,2,BLACK);
                    break;

                    case 12:
                    Tft.drawString("DESEMBRE",135,75,2,BLACK);
                    break;
                    default:
                    break;
                   }
            }

//_____________________________________________________________________________

//_____________________FUNCIÓ PPROGRAMAR DIA, MES I ANY_____________________

void funciodia()
          {
           Tft.fillRectangle(0, 0, 239,320,BLACK);
           pantalla++;
           pantalla++;
           progdm=1;
           data=1;
```

```
e:
        //DIBUIX RECTANGLE NEGRE PER BORRAR TEXT ANTERIOR
        Tft.fillRectangle(0, 0, 239,25,BLACK);
        //TEXT PANTALLA CAPÇALERA
        Tft.drawString("PROGRAMACIO DATA I HORA",5,5,1,CYAN);
        //DIBUIX RECTANGLE GRIS
        Tft.fillRectangle(0, 15, 239,50,GRAY1);
        //TEX RECTANGLE GRIS
        Tft.drawString("DIA:",5,30,2,WHITE);
q:
        Tft.drawNumber(clock.dayOfMonth,130,30,2,BLACK);
        //DIBUIX RECTANGLE GRIS 2
        Tft.fillRectangle(0, 55, 239,50,GRAY1);
        Tft.drawString("MES:",5,75,2,WHITE);
        //DIBUIX RECTANGLE GRIS
        funciomesprog();
        Tft.fillRectangle(0, 100, 319,50,GRAY1);
        //TEX RECTANGLE GRIS
        Tft.drawString("ANY:",5,115,2,WHITE);
        Tft.drawString("20",135,115,2,BLACK);
        Tft.drawNumber(clock.year,160,115,2,BLACK);
        //DIBUIXAR RECTANGLE GRIS
        Tft.fillRectangle(0, 150, 319,55,GRAY2);
        // Tft.drawString("HORA:",5,175,2,WHITE);
        Tft.drawString("SEGUENT",75,175,2,BLACK);
        //DIBUIX RECTANGLE + / -
        Tft.fillRectangle(0, 205, 119,50,GREEN);
        Tft.fillRectangle(119, 205, 220,50,BLUE);
        // ESCRIURE TEXT + / -
```

```
Tft.drawString("+",30,210,5,BLACK);
Tft.drawString("-",160,210,5,BLACK);
// DIBUIX RECTANGLE
Tft.fillRectangle(0, 255, 239,50,RED);
// TEXT RECTANGLE VERMELL
Tft.drawString("SORTIR",90,270,2,BLACK);
a=135;  //  a valor de posicio x
b=75;  //  b valor de posicio y
funciomes();
//TEXT PEU DE PANTALLA
Tft.drawString("ALBERT BOIX 2012 SCCA",0,310,1,CYAN);
dataihora=0;
p.x = map(p.x, TS_MINX, TS_MAXX, 0, 240);
p.y = map(p.y, TS_MINY, TS_MAXY, 0, 320);
while (progdm==1)
            {
             Tft.fillRectangle(120, 20, 50,30,GRAY1);
             Tft.drawNumber(clock.dayOfMonth,130,30,2,BLACK);
             delay(1000);
             Tft.drawNumber(clock.dayOfMonth,130,30,2,GRAY1);
             delay(1000);
             Point p = ts.getPoint();
             if (p.z > __PRESURE) {
                          x=(p.x);
                          y=(p.y);
                          if (y>1500)
                                  {
                                   x=0;
                                   y=0;
```

```
                progdm=0;
                pantalla=2;
               }
            if (y>1230 && y<1480 && x>1 && x<980)
                                {
                                 x=0;
                                 y=0;
                                 diames++;
                                 if (clock.month!=2 || clock.month!=4
||clock.month!=6 ||clock.month!=9 ||clock.month!=11)
{
                                                                  if
(diames==32 ||clock.dayOfMonth==32)
{
diames=1;
clock.dayOfMonth=1;
}
}
{
                                                    if(clock.month==4
||clock.month==6 || clock.month==9 || clock.month==11)
{
                                                                  if
(diames==31 ||clock.dayOfMonth==31)
{
```

```
diames=1;

clock.dayOfMonth=1;

}

}
                                                    if (clock.month==2)
                                                        {
                                                        if          (diames==30
||clock.dayOfMonth==30)
                                                                                {

diames=1;

clock.dayOfMonth=1;
                                                                                }
                                                        }
                                                    clock.dayOfMonth=diames;
                                                 }
                                                }


                                        if (y>1230 && y<1480 && x>980 )
                                                {
                                                  x=0;
                                                  y=0;
                                                  diames--;
                                                  if (clock.month!=2 || clock.month!=4
|| clock.month!=6 ||clock.month!=9 ||clock.month!=11){

if (diames==0 ||clock.dayOfMonth==0)
```

```
{
diames=31;
clock.dayOfMonth=31;
}
}
{
                                                    if (clock.month==4 ||
clock.month==6 || clock.month==9 || clock.month==11)
                                                                {
                                                                 if
(diames==1 ||clock.dayOfMonth==1)
{
diames=30;
clock.dayOfMonth=30;
}
                                                                }
                                                    if (clock.month==2)
                                                          {
                                                            if          (diames==1
||clock.dayOfMonth==1)
                                                                       {
diames=29;
clock.dayOfMonth=29;
```

```
                                                                    }
                                                                }
                                                            }
                                                             }
                                                              clock.dayOfMonth=diames;
                                        }
                                Tft.drawNumber( clock.dayOfMonth,130,30,2,BLACK);
                                 if (y>956 && y<1194)
                                                {
                                                 progdm=0;
                                                 progm=1;
                                                 x=0;
                                                 y=0;
                                                 while (progm==1)
                                                            {
                                                             switch (clock.month)
                                                                        {
                                                                         case 1:
Tft.drawString("GENER",135,75,2,BLACK);
                                                                         delay(500);
Tft.drawString("GENER",135,75,2,GRAY1);
                                                                         delay(500);
                                                                         break;

                                                                         case 2:
Tft.drawString("FEBRER",135,75,2,BLACK);
                                                                         delay(500);
```

```
Tft.drawString("FEBRER",135,75,2,GRAY1);
                                        delay(500);
                                        break;

                                        case 3:
Tft.drawString("MARC,",135,75,2,BLACK);
                                        delay(500);
Tft.drawString("MARC,",135,75,2,GRAY1);
                                        delay(500);
                                        break;

                                        case 4:
Tft.drawString("ABRIL",135,75,2,BLACK);
                                        delay(500);
Tft.drawString("ABRIL",135,75,2,GRAY1);
                                        delay(500);
                                        break;

                                        case 5:
Tft.drawString("MAIG",135,75,2,BLACK);
                                        delay(500);
Tft.drawString("MAIG",135,75,2,GRAY1);
                                        delay(500);
                                        break;
```

```
                                        case 6:
Tft.drawString("JUNY",135,75,2,BLACK);
                                        delay(500);
Tft.drawString("JUNY",135,75,2,GRAY1);
                                        delay(500);
                                        break;

                                        case 7:
Tft.drawString("JULIOL",135,75,2,BLACK);
                                        delay(500);
Tft.drawString("JULIOL",135,75,2,GRAY1);
                                        delay(500);
                                        break;

                                        case 8:
Tft.drawString("AGOST",135,75,2,BLACK);
                                        delay(500);
Tft.drawString("AGOST",135,75,2,GRAY1);
                                        delay(500);
                                        break;

                                        case 9:
Tft.drawString("SETEMBRE",135,75,2,BLACK);
                                        delay(500);
```

```
Tft.drawString("SETEMBRE",135,75,2,GRAY1);
                                        delay(500);
                                        break;

                                        case 10:
Tft.drawString("OCTUBRE",135,75,2,BLACK);
                                        delay(500);
Tft.drawString("OCTUBRE",135,75,2,GRAY1);
                                        delay(500);
                                        break;

                                        case 11:
Tft.drawString("NOVEMBRE",135,75,2,BLACK);
                                        delay(500);
Tft.drawString("NOVEMBRE",135,75,2,GRAY1);
                                        delay(500);
                                        break;

                                        case 12:
Tft.drawString("DESEMBRE",135,75,2,BLACK);
                                        delay(500);
Tft.drawString("DESEMBRE",135,75,2,GRAY1);
                                        delay(500);
                                        break;
                                        default:
```

```
                                        break;
                                       }

                                Point p = ts.getPoint();
                                if (p.z > __PRESURE) {
                                        x=(p.x);
                                        y=(p.y);
                                       }

                                if (y>1488)
                                      {
                                       x=0;
                                       y=0;
                                       progm=0;
                                       pantalla=2;
                                       }

//*********************************************************************************************************
if (y>1230 && y<1480 && x>1 && x<980)
                                {
                                 delay(25);
                                 month1++;
                                 if (month1==13)
                                        {
                                         month1=1;
                                        }
                                 delay(25);
                                 x=0;
                                 y=0;
```

```
                    }
 clock.month=month1;
 mes=clock.month;
 if (y>1230 && y<1480 && x>980 )
                   {
                    delay(25);
                    month1 --;
                    if (month1==0)
                          {
                           month1=12;
                          }
                    delay(25);
                    x=0;
                    y=0;
                    }
if (y>956 && y<1194)
              {
               x=0;
               y=0;
               progdm=0;
               progm=0;
               proga=1;
               a=135;  //  a valor de posicio x
               b=75;  //  b valor de posicio y
               funciomes();
              }

               while (proga==1)
                     {
```

```
funciomesprog();
delay(1000);
Tft.drawString("20",135,115,2,BLACK);
Tft.drawNumber(clock.year,160,115,2,BLACK);
delay(1000);
Tft.drawString("20",135,115,2,GRAY1);
Tft.drawNumber(clock.year,160,115,2,GRAY1);
Point p = ts.getPoint();
if (p.z > __PRESURE) {
                delay(25);
                x=(p.x);
                y=(p.y);
               }
if (y>1230 && y<1480 && x>1 && x<980)
                        {
                        delay(25);
                        clock.year++;
                        if (clock.year==99)
                                {
                                 clock.year=00;
                                 }
                        delay(25);
                        x=0;
                        y=0;
                        }
if (y>1230 && y<1480 && x>980 )
                       {
                       delay(25);
                       clock.year--;
```

```
                        if (clock.year==0)
                                {
                                 clock.year=99;
                                 }
                        delay(25);
                        x=0;
                        y=0;
                        }
 if (y>1500)
        {
         x=0;
         y=0;
         proga=0;
         pantalla=2;
        }
 progminut=1;
if  (progminut=1)
            {
             if (y>956 && y<1194)
                        {
                         progminut=1;
                         progminuthora();
                        }
             }

}
```

```
//*************************************************************************************************
```

```
                                        }
                                         clock.month=month1;



                                       }

                    }



    data=0;
    clock.setTime();
        }

//_____________________________________________________________________________

//_____________________FUNCIÓ PROGRAMACIÓ MINUTS I HORES_____________________

void progminuthora()
                {
                 Tft.fillRectangle(0, 0, 239,320,BLACK);
                 //DIBUIX RECTANGLE NEGRE PER BORRAR TEXT ANTERIOR
                 Tft.fillRectangle(0, 0, 239,25,BLACK);
                 //TEXT PANTALLA CAPÇALERA
                 Tft.drawString("PROGRAMACIO MINUTS I HORA",5,5,1,CYAN);
                 //DIBUIX RECTANGLE GRIS
                 Tft.fillRectangle(0, 15, 239,135,GRAY1);
                 //TEX RECTANGLE GRIS
```

```
Tft.drawString("MINUTS:",5,30,2,WHITE);
//DIBUIX RECTANGLE GRIS 2
Tft.fillRectangle(0, 55, 239,50,GRAY1);
Tft.drawString("HORA:",5,75,2,WHITE);
Tft.drawNumber(clock.hour,130,75,2,BLACK);
//DIBUIXAR RECTANGLE GRIS
Tft.fillRectangle(0, 150, 319,55,GRAY2);
Tft.drawString("SEGUENT",75,175,2,BLACK);
//DIBUIX RECTANGLE + / -
Tft.fillRectangle(0, 205, 119,50,GREEN);
Tft.fillRectangle(119, 205, 220,50,BLUE);
// ESCRIURE TEXT + / -
Tft.drawString("+",30,210,5,BLACK);
Tft.drawString("-",160,210,5,BLACK);
// DIBUIX RECTANGLE
Tft.fillRectangle(0, 255, 239,50,RED);
// TEXT RECTANGLE VERMELL
Tft.drawString("SORTIR",90,270,2,BLACK);
// PEU DE PANTALLA
Tft.fillRectangle(0, 305, 239,10,BLACK);
Tft.drawString("ALBERT BOIX 2012 SCCA",0,310,1,CYAN);
while(progminut==1)
                {
                 Tft.drawNumber(clock.minute,130,30,2,BLACK);
                 delay(500);
                 Tft.drawNumber(clock.minute,130,30,2,GRAY1);
                 delay(500);
                 Point p = ts.getPoint();
                 if (p.z > __PRESURE) {
```

```
delay(25);
x=(p.x);
y=(p.y);
if (y>1500)
        {
          x=0;
          y=0;
          proga=0;
          progminut=0;
          pantalla=2;
        }
        if (y>1230 && y<1480 && x>1 && x<980)
                                {
                                  delay(25);
                                  clock.minute++;
                                  if (clock.minute==60)
                                          {
clock.minute=00;
                                          }
                                  delay(25);
                                  x=0;
                                  y=0;
                                }

        if (y>1230 && y<1480 && x>980 )
                                {
                                 delay(25);
```

```
                                                clock.minute--;
                                                if (clock.minute==0)
                                                        {
clock.minute=60;
                                                        }
                                                delay(25);
                                                x=0;
                                                y=0;
                                               }
                                        if (y>956 && y<1194)
                                                {
                                                 progminut=0;
                                                 proghora=1;
                                                 pantalla=2;
                                                }
                                                                }
                        }
                        Tft.drawNumber(clock.minute,130,30,2,BLACK);
                        while (proghora==1)
                                {
                                 delay(500);
                                 Tft.drawNumber(clock.hour,130,75,2,BLACK);
                                 delay(500);
                                 Tft.drawNumber(clock.hour,130,75,2,GRAY1);
                                 Point p = ts.getPoint();
                                 if (p.z > __PRESURE) {
                                                delay(25);
                                                x=(p.x);
```

```
y=(p.y);
if (y>956 && y<1194)
                {
                 x=0;
                 y=0;
                 progdm=0;
                 progm=0;
                 proga=0;
                 funtelf();
                }

if (y>1500)
        {
          x=0;
          y=0;
          proghora=0;
          proga=0;
          progminut=0;
          pantalla=2;
        }
if (y>1230 && y<1480 && x>1 && x<980)
                        {
                         delay(25);
                         clock.hour++;
                         if (clock.hour==24)
                                {

clock.hour=00;

                                }
```

```
                                                    delay(25);
                                                    x=0;
                                                    y=0;
                                                    }


                                          if (y>1230 && y<1480 && x>980 )
                                                   {
                                                    delay(25);
                                                    clock.hour--;
                                                    if (clock.hour==0)
                                                            {
clock.hour=23;
                                                            }
                                                    delay(25);
                                                    x=0;
                                                    y=0;
                                                   }

                                                }
                                    }

    }

//_____________________________________________________________________

//_____________________________FUNCIÓ PROGRAMAR TELÈFON_____________________
void  funtelf()
```

```
{

                    {
                     Tft.fillRectangle(0, 0, 239,350,BLACK);
                     Tft.drawString("PROGRAMAR TELEFON",5,5,1,CYAN);
                     //DIBUIX RECTANGLE BLAU
                     Tft.fillRectangle(0, 15, 239,300,BLUE);
                     Tft.drawString("TELEFON:",5,30,2,WHITE);
                     numtf=13;
                     tf1 = EEPROM.read(numtf);
                     Tft.drawNumber(tf1,15,70,2,BLACK);
                     numtf=14;
                     tf2 = EEPROM.read(numtf);
                     Tft.drawNumber(tf2,30,70,2,BLACK);
                     numtf=15;
                     tf3 = EEPROM.read(numtf);
                     Tft.drawNumber(tf3,45,70,2,BLACK);
                     numtf=16;
                     tf4 = EEPROM.read(numtf);
                     Tft.drawNumber(tf4,60,70,2,BLACK);
                     numtf=17;
                     tf5 = EEPROM.read(numtf);
                     Tft.drawNumber(tf5,75,70,2,BLACK);
                     numtf=18;
                     tf6 = EEPROM.read(numtf);
                     Tft.drawNumber(tf6,90,70,2,BLACK);
                     numtf=19;
                     tf7 = EEPROM.read(numtf);
```

```
Tft.drawNumber(tf7,105,70,2,BLACK);
numtf=20;
tf8 = EEPROM.read(numtf);
Tft.drawNumber(tf8,120,70,2,BLACK);
numtf=21;
tf9 = EEPROM.read(numtf);
Tft.drawNumber(tf9,135,70,2,BLACK);
numtf=22;
tf10 = EEPROM.read(numtf);
Tft.drawNumber(tf10,150,70,2,BLACK);
numtf=23;
tf11 = EEPROM.read(numtf);
Tft.drawNumber(tf11,165,70,2,BLACK);
numtf=24;
tf12 = EEPROM.read(numtf);
Tft.drawNumber(tf12,180,70,2,BLACK);
numtf=25;
tf13 = EEPROM.read(numtf);
Tft.drawNumber(tf13,195,70,2,BLACK);
//TEXT PANTALLA CAPÇALERA
//DIBUIXAR RECTANGLE GRIS
Tft.fillRectangle(0, 150, 319,55,GRAY2);
Tft.drawString("SEGUENT",75,175,2,BLACK);
//DIBUIX RECTANGLE + / -
Tft.fillRectangle(0, 205, 119,50,GREEN);
Tft.fillRectangle(119, 205, 220,50,BLUE);
// ESCRIURE TEXT + / -
Tft.drawString("+",30,210,5,BLACK);
Tft.drawString("-",160,210,5,BLACK);
```

```
                    // DIBUIX RECTANGLE
                    Tft.fillRectangle(0, 255, 239,60,RED);
                    // TEXT RECTANGLE VERMELL
                    Tft.drawString("SORTIR",90,270,2,BLACK);
                    // PEU DE PANTALLA
                    Tft.fillRectangle(0, 305, 239,10,BLACK);
                    Tft.drawString("ALBERT BOIX 2012 SCCA",0,310,1,CYAN);
                    progtelf1=1;


//**********************************************************TF1***********************************************
                    while(progtelf1==1)
                            {
O:
                             numtf=13;
                             tf1 = EEPROM.read(numtf);
                             Tft.drawNumber(tf1,15,70,2,BLACK);
                             delay(500);
                             Tft.drawNumber(tf1,15,70,2,BLUE);
                             delay(500);
                             Point p = ts.getPoint();
                             if (p.z > __PRESURE) {
                                      x=(p.x);
                                      y=(p.y);
                                      if (y>1500)
                                             {
                                              progtelf1=0;
                                              x=0;
                                              y=0;
```

```
            proga=0;
            progminut=0;
            pantalla=2;
           }
n=tf1;
if (y>1230 && y<1480 && x>1 && x<980)
                                {
                                 delay(25);
                                 n++;
                                 if (n==10)
                                        {
                                         n=0;
                                         }
                                 delay(25);
                                 x=0;
                                 y=0;
                                }

tf1=n;
if (y>1230 && y<1480 && x>980 )
                                {
                                delay(25);
                                n--;
                                if (n==-1)
                                        {
                                         n=9;
                                        }
                                delay(25);
```

```
                                                    x=0;
                                                    y=0;
                                                   }
                                        tf1=n;
                                        if (y>956 && y<1194)
                                                  {
                                                   delay(500);
                                                   progtelf1=0;
                                                   progtelf2=1;
                                                   y=0;
                                                   }
                                              }
                                Tft.drawNumber(tf1,15,70,2,BLACK);
                                EEPROM.write(numtf, tf1);


                                }

//*************************************************************TF2*************************************************
 while(progtelf2==1)
                                                {
                                                 numtf=14;
                                                 tf2 = EEPROM.read(numtf);
                                                 Tft.drawNumber(tf2,30,70,2,BLACK);
                                                 delay(500);
                                                 Tft.drawNumber(tf2,30,70,2,BLUE);
                                                 delay(500);
                                                 Point p = ts.getPoint();
                                                 if (p.z > __PRESURE) {
```

```
x=(p.x);
y=(p.y);
if (y>1500)
        {
         progtelf2=0;
         x=0;
         y=0;
         proga=0;
         progminut=0;
         pantalla=2;
        }
n=tf2;
if (y>1230 && y<1480 && x>1 && x<980)
                                {
                                 delay(25);
                                 n++;
                                 if (n==10)
                                        {
                                         n=0;
                                         }
                                 delay(25);
                                 x=0;
                                 y=0;
                                }

tf2=n;
if (y>1230 && y<1480 && x>980 )
                                {
```

```
                                                    delay(25);
                                                    n--;
                                                    if (n==-1)
                                                            {
                                                             n=9;
                                                            }
                                                    delay(25);
                                                    x=0;
                                                    y=0;
                                                }
                                        tf2=n;
                                        if (y>956 && y<1194)
                                                {
                                                 delay(500);
                                                progtelf2=0;
                                                progtelf3=1;
                                                y=0;
                                                }
                                    }
                                Tft.drawNumber(tf2,30,70,2,BLACK);
                                EEPROM.write(numtf, tf2);
                                }
//*********************************************************TF3*************************************************
 while(progtelf3==1)
                                {
                                numtf=15;
                                tf3 = EEPROM.read(numtf);
                                Tft.drawNumber(tf3,45,70,2,BLACK);
                                delay(500);
```

```
Tft.drawNumber(tf3,45,70,2,BLUE);
delay(500);
Point p = ts.getPoint();
if (p.z > __PRESURE) {
        x=(p.x);
        y=(p.y);
        if (y>1500)
            {
             progtelf3=0;
             x=0;
             y=0;
             proga=0;
             progminut=0;
             pantalla=2;
            }
        n=tf3;
        if (y>1230 && y<1480 && x>1 && x<980)
                {
                 delay(25);
                 n++;
                 if (n==10)
                    {
                     n=0;
                    }
                 delay(25);
                 x=0;
                 y=0;
                }
```

```
                tf3=n;
                if (y>1230 && y<1480 && x>980 )
                                {
                                delay(25);
                                n--;
                                if (n==-1)
                                        {
                                         n=9;
                                        }
                                delay(25);
                                x=0;
                                y=0;
                        }
                tf3=n;
                if (y>956 && y<1194)
                                {
                                 delay(500);
                                progtelf3=0;
                                progtelf4=1;
                                y=0;
                                }
                        }
          Tft.drawNumber(tf3,45,70,2,BLACK);
          EEPROM.write(numtf, tf3);
        }

//*************************************************************************************************************
```

```
//*********************************************************TF4*****************************************
 while(progtelf4==1)
                                                {
                                                 numtf=16;
                                                 tf4 = EEPROM.read(numtf);
                                                 Tft.drawNumber(tf4,60,70,2,BLACK);
                                                 delay(500);
                                                 Tft.drawNumber(tf4,60,70,2,BLUE);
                                                 delay(500);
                                                 Point p = ts.getPoint();
                                                 if (p.z > __PRESURE) {
                                                                 x=(p.x);
                                                                 y=(p.y);
                                                                 if (y>1500)
                                                                        {
                                                                         progtelf4=0;
                                                                         x=0;
                                                                         y=0;
                                                                         proga=0;
                                                                         progminut=0;
                                                                         pantalla=2;
                                                                        }
                                                                 n=tf4;
                                                                 if (y>1230 && y<1480 && x>1 && x<980)
                                                                                                 {
                                                                                                  delay(25);
                                                                                                  n++;
                                                                                                  if (n==10)
```

```
                                    {
                                     n=0;
                                     }
                                  delay(25);
                                  x=0;
                                  y=0;
                                 }

tf4=n;
if (y>1230 && y<1480 && x>980 )
                                 {
                                 delay(25);
                                 n--;
                                 if (n==-1)
                                        {
                                         n=9;
                                        }
                                 delay(25);
                                 x=0;
                                 y=0;
                          }
tf4=n;
if (y>956 && y<1194)
                            {
                             delay(500);
                            progtelf4=0;
                            progtelf5=1;
                            y=0;
                            }
```

```
                                        }
                                    Tft.drawNumber(tf4,60,70,2,BLACK);
                                    EEPROM.write(numtf, tf4);
                                  }

//*************************************************************************************************************

//*********************************************************TF5*************************************************
 while(progtelf5==1)
                                    {
                                     numtf=17;
                                     tf5 = EEPROM.read(numtf);
                                     Tft.drawNumber(tf5,75,70,2,BLACK);
                                     delay(500);
                                     Tft.drawNumber(tf5,75,70,2,BLUE);
                                     delay(500);
                                     Point p = ts.getPoint();
                                     if (p.z > __PRESURE) {
                                                  x=(p.x);
                                                  y=(p.y);
                                                  if (y>1500)
                                                        {
                                                         progtelf5=0;
                                                         x=0;
                                                         y=0;
                                                         proga=0;
                                                         progminut=0;
                                                         pantalla=2;
                                                        }
```

```
n=tf5;
if (y>1230 && y<1480 && x>1 && x<980)
                {
                 delay(25);
                 n++;
                 if (n==10)
                        {
                         n=0;
                         }
                 delay(25);
                 x=0;
                 y=0;
                }

tf5=n;
if (y>1230 && y<1480 && x>980 )
                {
                delay(25);
                n--;
                if (n==-1)
                        {
                         n=9;
                        }
                delay(25);
                x=0;
                y=0;
            }
tf5=n;
```

```
                                        if (y>956 && y<1194)
                                                {
                                                 delay(500);
                                                progtelf5=0;
                                                progtelf6=1;
                                                y=0;
                                                }
                                        }
                                Tft.drawNumber(tf5,75,70,2,BLACK);
                                EEPROM.write(numtf, tf5);
                               }

//*************************************************************************************************
//*************************************************TF6*************************************
 while(progtelf6==1)
                                        {
                                         numtf=18;
                                         tf6 = EEPROM.read(numtf);
                                         Tft.drawNumber(tf6,90,70,2,BLACK);
                                         delay(500);
                                         Tft.drawNumber(tf6,90,70,2,BLUE);
                                         delay(500);
                                         Point p = ts.getPoint();
                                         if (p.z > __PRESURE) {
                                                x=(p.x);
                                                y=(p.y);
                                                if (y>1500)
                                                        {
                                                         progtelf6=0;
```

```
                x=0;
                y=0;
                proga=0;
                progminut=0;
                pantalla=2;
               }
        n=tf6;
        if (y>1230 && y<1480 && x>1 && x<980)
                                {
                                 delay(25);
                                 n++;
                                 if (n==10)
                                        {
                                         n=0;
                                         }
                                 delay(25);
                                 x=0;
                                 y=0;
                                }

        tf6=n;
        if (y>1230 && y<1480 && x>980 )
                                {
                                delay(25);
                                n--;
                                if (n==-1)
                                        {
                                         n=9;
```

```
                                                                    }
                                                                delay(25);
                                                                x=0;
                                                                y=0;
                                                            }
                                                        tf6=n;
                                                        if (y>956 && y<1194)
                                                            {
                                                             delay(500);
                                                            progtelf6=0;
                                                            progtelf7=1;
                                                            y=0;
                                                            }
                                                    }
                                                Tft.drawNumber(tf6,90,70,2,BLACK);
                                                EEPROM.write(numtf, tf6);
                                            }

//*************************************************************************************************************
//*********************************************************TF7*************************************************
 while(progtelf7==1)
                                                {
                                                 numtf=19;
                                                 tf7 = EEPROM.read(numtf);
                                                 Tft.drawNumber(tf7,105,70,2,BLACK);
                                                 delay(500);
                                                 Tft.drawNumber(tf7,105,70,2,BLUE);
                                                 delay(500);
                                                 Point p = ts.getPoint();
```

```
if (p.z > __PRESURE) {
        x=(p.x);
        y=(p.y);
        if (y>1500)
            {
             progtelf7=0;
             x=0;
             y=0;
             proga=0;
             progminut=0;
             pantalla=2;
            }
        n=tf7;
        if (y>1230 && y<1480 && x>1 && x<980)
                {
                 delay(25);
                 n++;
                 if (n==10)
                    {
                     n=0;
                     }
                 delay(25);
                 x=0;
                 y=0;
                }

        tf7=n;
        if (y>1230 && y<1480 && x>980 )
```

```
                                                    {
                                                    delay(25);
                                                    n--;
                                                    if (n==-1)
                                                            {
                                                             n=9;
                                                            }
                                                    delay(25);
                                                    x=0;
                                                    y=0;
                                                }
                                        tf7=n;
                                        if (y>956 && y<1194)
                                                {
                                                 delay(500);
                                                progtelf7=0;
                                                progtelf8=1;
                                                y=0;
                                                }
                                    }
                                Tft.drawNumber(tf7,105,70,2,BLACK);
                                EEPROM.write(numtf, tf7);
                            }

//*************************************************************************************************
//*********************************************************TF8*************************************
 while(progtelf8==1)
                                {
                                 numtf=20;
```

```
tf8 = EEPROM.read(numtf);
Tft.drawNumber(tf8,120,70,2,BLACK);
delay(500);
Tft.drawNumber(tf8,120,70,2,BLUE);
delay(500);
Point p = ts.getPoint();
if (p.z > __PRESURE) {
        x=(p.x);
        y=(p.y);
        if (y>1500)
            {
             progtelf8=0;
             x=0;
             y=0;
             proga=0;
             progminut=0;
             pantalla=2;
            }
        n=tf8;
        if (y>1230 && y<1480 && x>1 && x<980)
                {
                 delay(25);
                 n++;
                 if (n==10)
                    {
                     n=0;
                    }
                 delay(25);
```

```
                                        x=0;
                                        y=0;
                                       }

                    tf8=n;
                    if (y>1230 && y<1480 && x>980 )
                                        {
                                        delay(25);
                                        n--;
                                        if (n==-1)
                                                {
                                                 n=9;
                                                }
                                        delay(25);
                                        x=0;
                                        y=0;
                                }
                    tf8=n;
                    if (y>956 && y<1194)
                                        {
                                         delay(500);
                                        progtelf8=0;
                                        progtelf9=1;
                                        y=0;
                                        }
                                }
          Tft.drawNumber(tf8,120,70,2,BLACK);
          EEPROM.write(numtf, tf8);
        }
```

```
//*************************************************************************************************************

//*********************************************************TF9*************************************************
 while(progtelf9==1)
                                                {
                                                 numtf=21;
                                                 tf9 = EEPROM.read(numtf);
                                                 Tft.drawNumber(tf9,135,70,2,BLACK);
                                                 delay(500);
                                                 Tft.drawNumber(tf9,135,70,2,BLUE);
                                                 delay(500);
                                                 Point p = ts.getPoint();
                                                 if (p.z > __PRESURE) {
                                                                 x=(p.x);
                                                                 y=(p.y);
                                                                 if (y>1500)
                                                                        {
                                                                         progtelf9=0;
                                                                         x=0;
                                                                         y=0;
                                                                         proga=0;
                                                                         progminut=0;
                                                                         pantalla=2;
                                                                        }
                                                                 n=tf9;
                                                                 if (y>1230 && y<1480 && x>1 &&
x<980)
                                                                                 {
```

```
                delay(25);
                n++;
                if (n==10)
                    {
                     n=0;
                     }
                delay(25);
                x=0;
                y=0;
               }

tf9=n;
if (y>1230 && y<1480 && x>980 )
                    {
                    delay(25);
                    n--;
                    if (n==-1)
                        {
                         n=9;
                        }
                    delay(25);
                    x=0;
                    y=0;
                }
tf9=n;
if (y>956 && y<1194)
                  {
                   delay(500);
                  progtelf9=0;
```

```
                                                progtelf10=1;
                                                y=0;
                                                }
                                        }
                                Tft.drawNumber(tf9,135,70,2,BLACK);
                                EEPROM.write(numtf, tf9);
                               }


//*************************************************************************************************************
//*********************************************************TF10*************************************************
 while(progtelf10==1)
                                        {
                                         numtf=22;
                                         tf10 = EEPROM.read(numtf);
                                         Tft.drawNumber(tf10,150,70,2,BLACK);
                                         delay(500);
                                         Tft.drawNumber(tf10,150,70,2,BLUE);
                                         delay(500);
                                         Point p = ts.getPoint();
                                         if (p.z > __PRESURE) {
                                                        x=(p.x);
                                                        y=(p.y);
                                                        if (y>1500)
                                                                {
                                                                 progtelf10=0;
                                                                 x=0;
                                                                 y=0;
                                                                 proga=0;
```

```
            progminut=0;
            pantalla=2;
           }
n=tf10;
if (y>1230 && y<1480 && x>1 && x<980)
                                {
                                 delay(25);
                                 n++;
                                 if (n==10)
                                        {
                                         n=0;
                                         }
                                 delay(25);
                                 x=0;
                                 y=0;
                                }

tf10=n;
if (y>1230 && y<1480 && x>980 )
                                {
                                delay(25);
                                n--;
                                if (n==-1)
                                        {
                                         n=9;
                                        }
                                delay(25);
                                x=0;
```

```
                                                    y=0;
                                                }
                                            tf10=n;
                                            if (y>956 && y<1194)
                                                {
                                                 delay(500);
                                                progtelf10=0;
                                                progtelf11=1;
                                                y=0;
                                                }
                                        }
                                    Tft.drawNumber(tf10,150,70,2,BLACK);
                                    EEPROM.write(numtf, tf10);
                                }

//*************************************************************************************************
//*********************************************************TF11*********************************
 while(progtelf11==1)
                                        {
                                         numtf=23;
                                         tf11 = EEPROM.read(numtf);
                                         Tft.drawNumber(tf11,165,70,2,BLACK);
                                         delay(500);
                                         Tft.drawNumber(tf11,165,70,2,BLUE);
                                         delay(500);
                                         Point p = ts.getPoint();
                                         if (p.z > __PRESURE) {
                                                    x=(p.x);
```

```
y=(p.y);
if (y>1500)
        {
         progtelf11=0;
         x=0;
         y=0;
         proga=0;
         progminut=0;
         pantalla=2;
        }
n=tf11;
if (y>1230 && y<1480 && x>1 && x<980)
                                {
                                 delay(25);
                                 n++;
                                 if (n==10)
                                        {
                                         n=0;
                                         }
                                 delay(25);
                                 x=0;
                                 y=0;
                                }

tf11=n;
if (y>1230 && y<1480 && x>980 )
                                {
                                delay(25);
```

```
                                        n--;
                                        if (n==-1)
                                                {
                                                 n=9;
                                                }
                                        delay(25);
                                        x=0;
                                        y=0;
                                }
                        tf11=n;
                        if (y>956 && y<1194)
                                {
                                 delay(500);
                                progtelf11=0;
                                progtelf12=1;
                                y=0;
                                }
                        }
                Tft.drawNumber(tf11,165,70,2,BLACK);
                EEPROM.write(numtf, tf11);
               }

//*****************************************************************************************************
//*******************************************************TF12****************************************
 while(progtelf12==1)
                {
                 numtf=24;
                 tf12 = EEPROM.read(numtf);
                 Tft.drawNumber(tf12,180,70,2,BLACK);
```

```
delay(500);
Tft.drawNumber(tf12,180,70,2,BLUE);
delay(500);
Point p = ts.getPoint();
if (p.z > __PRESURE) {
                x=(p.x);
                 y=(p.y);
                 if (y>1500)
                        {
                         progtelf12=0;
                         x=0;
                         y=0;
                         proga=0;
                         progminut=0;
                         pantalla=2;
                        }
                n=tf12;
                if (y>1230 && y<1480 && x>1 && x<980)
                                        {
                                         delay(25);
                                         n++;
                                         if (n==10)
                                                {
                                                 n=0;
                                                 }
                                         delay(25);
                                         x=0;
                                         y=0;
```

```
                                                    }

                                tf12=n;
                                if (y>1230 && y<1480 && x>980 )
                                                    {
                                                    delay(25);
                                                    n--;
                                                    if (n==-1)
                                                            {
                                                             n=9;
                                                            }
                                                    delay(25);
                                                    x=0;
                                                    y=0;
                                                }
                                tf12=n;
                                if (y>956 && y<1194)
                                                   {
                                                    delay(500);
                                                    progtelf12=0;
                                                    progtelf13=1;
                                                    y=0;
                                                    }
                                              }
                             Tft.drawNumber(tf12,180,70,2,BLACK);
                             EEPROM.write(numtf, tf12);
                           }

//*************************************************************************************************
```

```
//************************************************************TF13*****************************************
 while(progtelf13==1)
                                                {
                                                 numtf=25;
                                                 tf13 = EEPROM.read(numtf);
                                                 Tft.drawNumber(tf13,195,70,2,BLACK);
                                                 delay(500);
                                                 Tft.drawNumber(tf13,195,70,2,BLUE);
                                                 delay(500);
                                                 Point p = ts.getPoint();
                                                 if (p.z > __PRESURE) {
                                                                 x=(p.x);
                                                                 y=(p.y);
                                                                 if (y>1500)
                                                                        {
                                                                         progtelf13=0;
                                                                         x=0;
                                                                         y=0;
                                                                         proga=0;
                                                                         progminut=0;
                                                                         pantalla=2;
                                                                        }
                                                                 n=tf13;
                                                                 if (y>1230 && y<1480 && x>1 && x<980)
                                                                                 {
                                                                                  delay(25);
                                                                                  n++;
```

```
                    if (n==10)
                        {
                         n=0;
                         }
                    delay(25);
                    x=0;
                    y=0;
                   }

tf13=n;
if (y>1230 && y<1480 && x>980 )
                        {
                        delay(25);
                        n--;
                        if (n==-1)
                            {
                             n=9;
                            }
                        delay(25);
                        x=0;
                        y=0;
                }
tf13=n;
if (y>956 && y<1194)
                    {
                     delay(500);
                    progtelf13=0;
                    progtelf1=1;
                    y=0;
```

```
Tft.drawNumber(tf13,195,70,2,BLACK);
                                                    goto O;
                                                  }
                                              }

                                    EEPROM.write(numtf, tf13);
                                    }



                              }
          }

//_____________________________________________________________________________

//_____________________________FUNCIÓ MES_____________________________

void funciomes()
          {
           resultat1=ecxessconsums/100;
           alarmames=clock.month;
           switch (clock.month)
                    {
                     case 1:
                     //Tft.drawString("GENER",a,b,2,BLACK);
                     if ( data!=1)
                              {
```

```
                alarmagener = EEPROM.read(alarmames);
                resultat2=resultat1*alarmagener;
                resultat3=resultat2+alarmagener;
                Tft.drawFloat(resultat3,140,215,2,BLACK);
                 }
break;

case 2:
// Tft.drawString("FEBRER",a,b,2,BLACK);
if ( data!=1)
          {
          alarmafebrer = EEPROM.read(alarmames);
          resultat2=resultat1*alarmafebrer;
          resultat3=resultat2+alarmafebrer;
          Tft.drawFloat(resultat3,140,215,2,BLACK);
         }
break;

case 3:
//Tft.drawString("MARC,",a,b,2,BLACK);
if ( data!=1)
          {
           alarmamarc= EEPROM.read(alarmames);
           resultat2=resultat1*alarmamarc;
           resultat3=resultat2+alarmamarc;
           Tft.drawFloat(resultat3,140,215,2,BLACK);
          }
break;
```

```
case 4:
//Tft.drawString("ABRIL",a,b,2,BLACK);
if ( data!=1)
        {
        alarmaabril= EEPROM.read(alarmames);
        resultat2=resultat1*alarmaabril;
        resultat3=resultat2+alarmaabril;
        Tft.drawFloat(resultat3,140,215,2,BLACK);
        }
break;

case 5:
// Tft.drawString("MAIG",a,b,2,BLACK);
if ( data!=1)
        {
        alarmamaig= EEPROM.read(alarmames);
        resultat2=resultat1*alarmamaig;
        resultat3=resultat2+alarmamaig;
        Tft.drawFloat(resultat3,140,215,2,BLACK);
        }
break;

case 6:
//Tft.drawString("JUNY",a,b,2,BLACK);
if ( data!=1)
        {
        alarmajuny= EEPROM.read(alarmames);
        resultat2=resultat1*alarmajuny;
        resultat3=resultat2+alarmajuny;
```

```
            Tft.drawFloat(resultat3,140,215,2,BLACK);
            }
break;

case 7:
//Tft.drawString("JULIOL",a,b,2,BLACK);
if ( data!=1)
            {
            alarmajuliol = EEPROM.read(alarmames);
            resultat2=resultat1*alarmajuliol;
            resultat3=resultat2+alarmajuliol;
            Tft.drawFloat(resultat3,140,215,2,BLACK);
            }
break;

case 8:
//Tft.drawString("AGOST",a,b,2,BLACK);
if ( data!=1)
            {
            alarmaagost = EEPROM.read(alarmames);
            resultat2=resultat1*alarmaagost;
            resultat3=resultat2+alarmaagost;
            Tft.drawFloat(resultat3,140,215,2,BLACK);
            }
break;

case 9:
//Tft.drawString("SETEMBRE",a,b,2,BLACK);
if ( data!=1)
```

```
        {
        alarmasetembre = EEPROM.read(alarmames);
        resultat2=resultat1*alarmasetembre;
        resultat3=resultat2+alarmasetembre;
        Tft.drawFloat(resultat3,140,215,2,BLACK);
        }
break;

case 10:
//Tft.drawString("OCTUBRE",a,b,2,BLACK);
if ( data!=1)
        {
        alarmaoctubre= EEPROM.read(alarmames);
        resultat2=resultat1*alarmaoctubre;
        resultat3=resultat2+alarmaoctubre;
        Tft.drawFloat(resultat3,140,215,2,BLACK);
        }
break;

case 11:
//Tft.drawString("NOVEMBRE",a,b,2,BLACK);
if ( data!=1)
        {
        alarmanovembre = EEPROM.read(alarmames);
        resultat2=resultat1*alarmanovembre;
        resultat3=resultat2+alarmanovembre;
        Tft.drawFloat(resultat3,140,215,2,BLACK);
        }
break;
```

```
                    case 12:
                   // Tft.drawString("DESEMBRE",a,b,2,BLACK);
                    if ( data!=1)
                            {
                            alarmadesembre = EEPROM.read(alarmames);
                            resultat2=resultat1*alarmadesembre;
                            resultat3=resultat2+alarmadesembre;
                            Tft.drawFloat(resultat3,140,215,2,BLACK);
                            }
                    break;
                    default:
                    break;
                        }
            }


//_____________________________FUNCIÓ PANTALLA PROGRAMACIÓ_____________


void pantprogramacio()

                   {
p:                  principal==0;
                   pantalla++;
                   //DIBUIX RECTANGLE NEGRE PER BORRAR TEXT ANTERIOR
                   Tft.fillRectangle(0, 0, 239,25,BLACK);
                   //TEXT PANTALLA CAPÇALER
                   Tft.drawString("PROGRAMACIO",5,5,1,CYAN);
                   //DIBUIX RECTANGLES GRIS
```

```
Tft.fillRectangle(0, 25, 239,65,GRAY2);
// TEXT RECTANGLE GRIS
Tft.drawString("CONSUMS",70,40,2,BLACK);
//DIBUIX RECTANGLE GRIS1
Tft.fillRectangle(0, 80, 239, 65,GRAY1);
// TEXT RECTANGLE GRIS
Tft.drawString("DIA I HORA I TELF.",20,100,2,BLACK);
//DIBUIX RECTANGLE GRIS
Tft.fillRectangle(0, 130, 239, 65,GREEN);
// ESCRIURE TEXT
Tft.drawString("TEMPS I SOBRECONSUM",5,150,2,BLACK);
//DIBUIX RECTANGLE GRIS
Tft.fillRectangle(0, 195, 239,55,YELLOW);
//TEXT RECTANGLE VERD
Tft.drawString("MANUAL",75,210,2,BLACK);
//DIBUIX RECTANGLE VERMELL
Tft.fillRectangle(0, 250, 239,55,RED);
//TEXT RECTANGLE VERMELL
Tft.drawString("SORTIR",90,270,2,BLACK);
//TEXT PEU DE PANTALLA
Tft.drawString("ALBERT BOIX 2012 SCCA",0,310,1,CYAN);
programacio=0;
y=0;//RESETEGEM EL PUNT DE TOUCH
x=0;//RESETEGEM EL PUNT DE TOUCH
consums=1;//proba
dataihora=1;
manual=1;
}
//_____
```

```
//_____________________________FUNCIÓ TEMPS I SOBECONSUM_____________________

void tempsobreconsums()

                {

                  Tft.fillRectangle(0, 0, 239,320,BLACK);

                  //DIBUIX RECTANGLE NEGRE PER BORRAR TEXT ANTERIOR

                  Tft.fillRectangle(0, 0, 239,25,BLACK);

                  //TEXT PANTALLA CAPÇALERA

                  Tft.drawString("PROGRAMACIO TEMPS ALARMA I
SOBRECONSUM",5,5,1,CYAN);

                  //DIBUIX RECTANGLE GRIS

                  Tft.fillRectangle(0, 15, 239,135,GRAY1);

                  //TEX RECTANGLE GRIS

                  Tft.drawString("TEMPS MOSTREIG:",5,30,2,WHITE);

                  //DIBUIX RECTANGLE GRIS 2

                  Tft.fillRectangle(0, 55, 239,50,GRAY1);

                  Tft.drawString("LITRES EXCES %:",5,85,2,WHITE);

                  //DIBUIXAR RECTANGLE GRIS

                  Tft.fillRectangle(0, 150, 319,55,GRAY2);

                  Tft.drawString("SEGUENT",75,175,2,BLACK);

                  //DIBUIX RECTANGLE + / -

                  Tft.fillRectangle(0, 205, 119,50,GREEN);

                  Tft.fillRectangle(119, 205, 220,50,BLUE);

                  // ESCRIURE TEXT + / -

                  Tft.drawString("+",30,210,5,BLACK);

                  Tft.drawString("-",160,210,5,BLACK);

                  // DIBUIX RECTANGLE

                  Tft.fillRectangle(0, 255, 239,50,RED);

                  // TEXT RECTANGLE VERMELL

                  Tft.drawString("SORTIR",90,270,2,BLACK);
```

```
// PEU DE PANTALLA
Tft.fillRectangle(0, 305, 239,10,BLACK);
Tft.drawString("ALBERT BOIX 2012 SCCA",0,310,1,CYAN);
Tft.drawNumber(ecxessconsums,115,115,2,BLACK);
progtempsalarma=1;
while( progtempsalarma==1)
                    {
                     delay(500);
                     Tft.fillRectangle(100, 50, 100,30,GRAY1);
                     Tft.drawNumber(tempsalarma,110,55,2,BLACK);
                     delay(500);
                     Tft.drawNumber(tempsalarma,110,55,2,GRAY1);
                     Point p = ts.getPoint();
                     if (p.z > __PRESURE) {
                                    delay(25);
                                    x=(p.x);
                                    y=(p.y);
                                    if (y>1500)
                                          {
                                           x=0;
                                           y=0;
                                           proga=0;
                                           progminut=0;
                                           pantalla=2;
                                           progtempsalarma=0;
                                          }
                                   if (y>1230 && y<1480 && x>1 && x<980)
                                                        {
                                                         tempsalarma++;
```

```
                                                    x=0;
                                                    y=0;
                                                  }


                                        if (y>1230 && y<1480 && x>980 )
                                                  {
                                                    tempsalarma--;
                                                    if (tempsalarma==-1)
                                                          {
tempsalarma=999;
                                                          }
                                                    delay(25);
                                                    x=0;
                                                    y=0;
                                                  }
                                              if (y>956 && y<1194)
                                                    {
                                                      progtempsalarma=0;
progecxessconsums=1;
                                                      }
                                                                  }
                                    }
                              while(progecxessconsums==1)
                                          {
                                           delay(500);
```

```
                                    Tft.fillRectangle(90, 110, 130,30,GRAY1);

Tft.drawNumber(ecxessconsums,115,115,2,BLACK);
                                    Point p = ts.getPoint();
                                if (p.z > __PRESURE) {
                                        delay(25);
                                        x=(p.x);
                                        y=(p.y);
                                        if (y>1500)
                                                {
                                                  progecxessconsums=0;
                                                  x=0;
                                                  y=0;
                                                  proga=0;
                                                  progminut=0;
                                                  pantalla=2;
                                                  progtempsalarma=0;
                                                 }
                                         if (y>1230 && y<1480 && x>1 && x<980)
                                                                {
                                                                  ecxessconsums++;
                                                                  x=0;
                                                                  y=0;
                                                                 }


                                        if (y>1230 && y<1480 && x>980 )
                                                                {
                                                                  ecxessconsums--;
```

```
                                                    if (ecxessconsums==-1)
                                                        {
ecxessconsums=999;
                                                        }
                                                    delay(25);
                                                    x=0;
                                                    y=0;
                                                }
                                            if (y>956 && y<1194)
                                                {
                                                    progtempsalarma=0;
progecxessconsums=1;
                                                }
                                                        }
                                    }
            }

//_____________________________________________________________________
//_____________________________FUNCIÓ MANUAL_____________________________

void pantmanual()
        {
```

```
pantalla++;
pantalla++;
//DIBUIX RECTANGLE NEGRE PER BORRAR TEXT ANTERIOR
Tft.fillRectangle(0, 0, 239,25,BLACK);
//TEXT PANTALLA CAPÇALERA
Tft.drawString("MANUAL",5,5,1,CYAN);
//DIBUIX RECTANGLE VERD
Tft.fillRectangle(0, 25, 239,70,YELLOW);//
//TEX RECTANGLE VERD
Tft.drawString("TANCAR VALVULA",5,50,2,BLACK);
//DIBUIX RECTANGLE BLAU
Tft.fillRectangle(0, 95, 239,70,GREEN);//
// TEXT RECTANGLE BLAU
Tft.drawString("OBRIR VALVULA",5,110,2,BLACK);
//DIBUIX RECTANGE GRIS
Tft.fillRectangle(0, 160, 239,80,GRAY2);
// DIBUIX RECTANGLE
Tft.fillRectangle(0, 239, 239,60,RED);
// TEXT RECTANGLE VERMELL
Tft.drawString("SORTIR",90,260,2,BLACK);
//TEXT PEU DE PANTALLA
Tft.drawString("ALBERT BOIX 2012 SCCA",0,310,1,CYAN);
while (pantalla==6)
        {
        #if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__) // mega
        #define YP A2
        #define XM A1
        #define YM 54
```

```
#define XP 57
#elif defined(__AVR_ATmega32U4__)
#define YP A2
#define XM A1
#define YM 18
#define XP 21
#else
#define YP A2
#define XM A1
#define YM 14
#define XP 17
#endif
#define TS_MINX 116*2
#define TS_MAXX 890*2
#define TS_MINY 83*2
#define TS_MAXY 913*2
TouchScreen ts = TouchScreen(XP, YP, XM, YM);
Serial.begin(9600);
Point p = ts.getPoint();
p.x = map(p.x, TS_MINX, TS_MAXX, 0, 240);
p.y = map(p.y, TS_MINY, TS_MAXY, 0, 320);
if (p.z > __PRESURE) {
        y=p.y;
        }
if (y>1&& y<95)
    {
    digitalWrite(relevalvula, HIGH);
    }
if (y>=95 && y<164)
```

```
                    {
                     digitalWrite(relevalvula, LOW);
                        }
            if (y>237)
                  {
                   digitalWrite(relevalvula, LOW);
                   pantalla=4;
                   manual=0;
                  }


 delay(100);
            }
         }

//_____________________________FUNCIÓ ALARMA_____________________________

void funAlarma()
            {
            digitalWrite(relevalvula, HIGH);

            }
//_____________________FUNCIÓ ENVIAR SMS_____________________

void funEnviarSMS()
            {
             Tft.drawString("ENVIANT SMS",10,120,2,BLACK);
             cell.begin(9600);
             cell.Verbose(true);
```

```
cell.Boot();
cell.FwdSMS2Serial();
tf1=tf1+48;
an=tf1;
tf2=tf2+48;
bn=tf2;
tf3=tf3+48;
cn=tf3;
tf4=tf4+48;
dn=tf4;
tf5=tf5+48;
en=tf5;
tf6=tf6+48;
fn=tf6;
tf7=tf7+48;
gn=tf7;
tf8=tf8+48;
hn=tf8;
tf9=tf9+48;
in=tf9;
tf10=tf10+48;
jn=tf10;
tf11=tf11+48;
kn=tf11;
tf12=tf12+48;
ln=tf12;
tf13=tf13+48;
mn=tf13;
Str4[0]=an;
```

```
                Str4[1]=bn;
                Str4[2]=cn;
                Str4[3]=dn;
                Str4[4]=en;
                Str4[5]=fn;
                Str4[6]=gn;
                Str4[7]=hn;
                Str4[8]=in;
                Str4[9]=jn;
                Str4[10]=kn;
                Str4[11]=ln;
                Str4[12]=mn;
                cell.Rcpt(Str4);
                cell.Rcpt(Str4);
                cell.Message("Alarma Valvula Tancada");
                Tft.drawString("ENVIANT SMS",10,150,2,BLACK);
                cell.SendSMS();
                sms=1;
               }
//_____________________________________FUNCIÓ SETUP_____________________________________

void setup(void) {
                Serial.begin(9600);
                Wire.begin();
                Tft.TFTinit();
                pinMode(sensorpin,INPUT);//  Declerem el sensor com entrada.
                pinMode(innterpantalla,INPUT);//  Declerem el sensor com entrada.
                pinMode(alarmapin,OUTPUT);//  Declerem el sensor com a sortida.
                pinMode(relevalvula,OUTPUT);//  Declerem la sortida del rele que activa la valvula.
```

```
 if (start==1)
        {
        start=0;
        }
clock.getTime();
```

// S'utilitza per ajustar el mateix valor el comptador digital i analògic. sempre ha d estar en comentaris execpta el primer cop

```
/*
d1=8;
d2=8;
d3=5;
d4=1;
d5=0;
d6=0;
d7=0;
d8=0;
EEPROM.write(mem1, d1);
EEPROM.write(mem2, d2);
EEPROM.write(mem3, d3);
EEPROM.write(mem4, d4);
EEPROM.write(mem5, d5);
EEPROM.write(mem6, d6);
EEPROM.write(mem7, d7);
EEPROM.write(mem8, d8);
*/
 }
```

```
//_____________________________________________________________________________

//_____________________________FUNCIÓ PRINCIPAL_____________________________


void loop(void) {
              if (digitalRead(innterpantalla) ==LOW )
            {
              digitalWrite(Backlignt,HIGH); // connecta la pantalla
            }
            else
            {
             digitalWrite(Backlignt,LOW);
            }
              clock.getTime();

             if (pantalla==0)
                           {
                            funcioinicialitzant();
                            }
             if (pantalla==1)
                          {
                          funcioalinear();
                          }


//*****************************Incrementació dels litres consumits*****************************************

              clock.getTime();
              if (digitalRead(sensorpin) ==LOW )//H
```

```
{
 delay(100);
 consumlitres++;
 comptadorlitres++;
 d1++;
 if (d1==10)
        {
        d1=0;
        d2++;
        }
         EEPROM.write(mem1, d1);
         if (d2==10)
                {
                d2=0;
                d3++;
                }
        EEPROM.write(mem2, d2);
        if (d3==10)
                {
                 d3=0;
                 d4++;
                }
        EEPROM.write(mem3, d3);
        if (d4==10)
                {
                d4=0;
                 d5++;
                }
        EEPROM.write(mem4, d4);
```

```
if (d5==10)
        {
        d5=0;
         d6++;
        }
EEPROM.write(mem5, d5);
if (d6==10)
        {
        d6=0;
         d7++;
        }
 EEPROM.write(mem6, d6);
 if (d7==10)
        {
        d7=0;
         d8++;
        }
  EEPROM.write(mem7, d7);
  if (d8==10)
        {
        d8==0;
          }
   EEPROM.write(mem8, d8);


if (principal==1 && pantalla==3)//principal= 0 c&& pantalla==3
        {
        goto prin;
        }
```

```
                                    }


//*****************************************PANTALLA PRINCIPAL*************************************
            if (principal==1 && pantalla==2)
                                    {
ini:
                                    // RECTANGLE VERMELL
                                    Tft.fillRectangle(0, 190, 239, 65,RED);
                                    //TEXT PAR VERMELLA
                                    Tft.drawString("ALARMA:",5,215,2,WHITE);
                                    // RECTANGLE VERD
                                    Tft.fillRectangle(0, 250, 239,55,GREEN);
                                    // TEXT RECTANGLE VERD
                                    Tft.drawString("PROGRAMACIO",50,265,2,BLACK);
                                  // PEU DE PANTALLA
                                    Tft.fillRectangle(0, 305, 239,20,BLACK);
                                    Tft.drawString("ALBERT BOIX 2012 SCCA",0,310,1,CYAN);
                                    pantalla++;
                                    //DIBUIX RECTANGLE NEGRE PER BORRAR TEXT
ANTERIOR
                                    Tft.fillRectangle(0, 0, 239,15,BLACK);
                                    // TITUL, CAPÇALERA PANTALLA

                                    Tft.drawString("PRINCIPAL",5,5,1,CYAN);
din:
                                    //DIBUIX RECTANGLE BLAU
                                    Tft.fillRectangle(0, 15, 239,190,BLUE);
                                    a=50;  //  a valor de posicio x
                                    b=25;  //  b valor de posicio y
```

```
funciomes();
Tft.fillRectangle(0,15,130,30,BLUE);//  Rectangle per evitar sobreposar valors.
Tft.drawNumber(clock.dayOfMonth,5,20,2,BLACK);
Tft.drawString("/",30,20,2,BLACK);
Tft.drawNumber(clock.month,45,20,2,BLACK);
Tft.drawString("/",67,20,2,BLACK);
Tft.drawString("20",80,20,2,BLACK);
Tft.drawNumber(clock.year,105,20,2,BLACK);
```

minu:

```
clock.getTime();
Tft.fillRectangle(175,15, 80,30,BLUE);//  Rectangle per evitar sobreposar valors
if (clock.hour<=9)
        {
        Tft.drawString("0",175,20,2,BLACK);
        Tft.drawNumber(clock.hour,188,20,2,BLACK);
        }
Tft.drawNumber(clock.hour,175,20,2,BLACK); //
Tft.drawString(":",200,20,2,BLACK);
Tft.drawNumber(clock.minute,210,20,2,BLACK);
if (clock.minute<=9)
        {
        Tft.drawString("0",210,20,2,BLACK);
        Tft.drawNumber(clock.minute,223,20,2,BLACK);
        }
 else
   {
```

```
                Tft.drawNumber(clock.minute,210,20,2,BLACK);
              }

              diames=clock.dayOfMonth;
              minut=clock.minute;
              Tft.drawString("MINUTS MARXA:",1,150,2,WHITE);
              Tft.fillRectangle(180,150, 130,30,BLUE);//  Rectangle per evitar
sobreposar valors.
              Tft.drawNumber( cronominut,185,150,2,BLACK);
              Tft.drawString("TEMPS MOSTREIG:",1,180,2,WHITE);
              Tft.drawNumber(tempsalarma,185,180,2,BLACK);
prin:
              Tft.drawString("TELF:",1,55,2,WHITE);
              numtf=13;
              tf1 = EEPROM.read(numtf);
              Tft.drawNumber(tf1,72,55,2,BLACK);
              numtf=14;
              tf2 = EEPROM.read(numtf);
              Tft.drawNumber(tf2,84,55,2,BLACK);
              numtf=15;
              tf3 = EEPROM.read(numtf);
              Tft.drawNumber(tf3,96,55,2,BLACK);
              numtf=16;
              tf4 = EEPROM.read(numtf);
              Tft.drawNumber(tf4,108,55,2,BLACK);
              numtf=17;
              tf5 = EEPROM.read(numtf);
              Tft.drawNumber(tf5,120,55,2,BLACK);
              numtf=18;
```

```
tf6 = EEPROM.read(numtf);
Tft.drawNumber(tf6,132,55,2,BLACK);
numtf=19;
tf7 = EEPROM.read(numtf);
Tft.drawNumber(tf7,144,55,2,BLACK);
numtf=20;
tf8 = EEPROM.read(numtf);
Tft.drawNumber(tf8,156,55,2,BLACK);
numtf=21;
tf9 = EEPROM.read(numtf);
Tft.drawNumber(tf9,168,55,2,BLACK);
numtf=22;
tf10 = EEPROM.read(numtf);
Tft.drawNumber(tf10,180,55,2,BLACK);
numtf=23;
tf11 = EEPROM.read(numtf);
Tft.drawNumber(tf11,192,55,2,BLACK);
numtf=24;
tf12 = EEPROM.read(numtf);
Tft.drawNumber(tf12,204,55,2,BLACK);
numtf=25;
tf13 = EEPROM.read(numtf);
Tft.drawNumber(tf13,216,55,2,BLACK);
Tft.drawString("COMPTADOR:",1,85,2,WHITE);
Tft.fillRectangle(125,85, 120,15,BLUE);// Rectangle per evitar
```

sobreposar valors.

```
d1 = EEPROM.read(mem1);
d2 = EEPROM.read(mem2);
d3 = EEPROM.read(mem3);
```

```
                    d4 = EEPROM.read(mem4);
                    d5 = EEPROM.read(mem5);
                    d6 = EEPROM.read(mem6);
                    d7 = EEPROM.read(mem7);
                    d8 = EEPROM.read(mem8);
                    Tft.drawNumber(d1,209,85,2,BLACK);//
                    Tft.drawNumber(d2,197,85,2,BLACK);//
                    Tft.drawNumber(d3,185,85,2,BLACK);//
                    Tft.drawNumber(d4,173,85,2,BLACK);//
                    Tft.drawNumber(d5,161,85,2,BLACK);//
                    Tft.drawNumber(d6,149,85,2,BLACK);//
                    Tft.drawNumber(d7,137,85,2,BLACK);//
                    Tft.drawNumber(d8,125,85,2,BLACK);//
                    Tft.fillRectangle(135, 120, 100,30,BLUE); // Rectangle pero
poder evitar la superposició de valors.
                    Tft.drawString("LITRES:",1,120,2,WHITE);
                    Tft.drawNumber(consumlitres,185,120,2,BLACK);// Valor de
litres consumits.
                    programacio=1;
                    y=0;
                    x=0;
                    }
    if (minut!=clock.minute && pantalla==3)
                {
                  cronominut++;
                  if (tempsalarma==cronominut)
                                {
                                consumlitres=0;
                                cronominut=0;
```

```
                                        }


                    if(diames!=clock.dayOfMonth&& pantalla==3)
                                {
                                goto din;
                                }
                    goto minu;
                    }


        clock.getTime();
        delay(500);
        Point p = ts.getPoint();
        p.x = map(p.x, TS_MINX, TS_MAXX, 0, 240);
        p.y = map(p.y, TS_MINY, TS_MAXY, 0, 320);
        if (p.z > __PRESURE) {
                     x=(p.x);
                     y=(p.y);
                    }
            if (y>250 && programacio==1 && pantalla==3)
                                        {
                                         pantprogramacio();


                                        }
//******
if (y>5 && y<80 && consums==1 && pantalla==4)
//*****************************************PANTALLA CONSUMS*************************************
                    {
pantalla++;
//DIBUIX RECTANGLE NEGRE PER BORRAR TEXT ANTERIOR
```

```
Tft.fillRectangle(0, 0, 239,25,BLACK);
//TEXT PANTALLA CAPÇALERA
Tft.drawString("PROGRAMACIO CONSUMS",5,5,1,CYAN);
//DIBUIX RECTANGLE + / -
Tft.fillRectangle(0, 125, 119,60,GREEN);
Tft.fillRectangle(119, 125, 120,60,BLUE);
// ESCRIURE TEXT + / -
Tft.drawString("+",30,130,5,BLACK);
Tft.drawString("-",160,130,5,BLACK);
//DIBUIX RECTANGE GRIS
Tft.fillRectangle(0, 180, 239,60,GRAY2);
// TEXT RECTANGLE GROC
Tft.drawString("SEGUENT",90,195,2,BLACK);
// DIBUIX RECTANGLE
Tft.fillRectangle(0, 239, 239,60,RED);
z:
y:
t:
//DIBUIX RECTANGLE GRIS 1
Tft.fillRectangle(0, 85, 239,45,GRAY2);
Tft.drawString("CONSUM:",5,95,2,WHITE);
// TEXT RECTANGLE VERMELL
Tft.drawString("SORTIR",90,260,2,BLACK);
//DIBUIX RECTANGLE GRIS 1
Tft.fillRectangle(0, 25, 239,60,GRAY1);
//TEX RECTANGLE GRIS
Tft.drawString("MES:",5,50,2,WHITE);
alarmames=mes;
switch (mes){
```

```
case 1:
//  Gener.
Tft.drawString("GENER",100,50,2,BLACK);
alarmagener = EEPROM.read(alarmames);
Tft.drawNumber(alarmagener,170,95,2,BLACK);
break;
case 2:
//  Febrer.
Tft.drawString("FEBRER",100,50,2,BLACK);
alarmafebrer = EEPROM.read(alarmames);
Tft.drawNumber(alarmafebrer,170,95,2,BLACK);
break;
case 3:
//  Març
Tft.drawString("MARC,",100,50,2,BLACK);
alarmamarc = EEPROM.read(alarmames);
Tft.drawNumber(alarmamarc,170,95,2,BLACK);
break;
case 4:
//  Abril.
Tft.drawString("ABRIL",100,50,2,BLACK);
alarmaabril = EEPROM.read(alarmames);
Tft.drawNumber(alarmaabril,170,95,2,BLACK);
break;

case 5:
//  Maig.  .
Tft.drawString("MAIG",100,50,2,BLACK);
alarmamaig = EEPROM.read(alarmames);
```

```
Tft.drawNumber(alarmamaig,170,95,2,BLACK);
break;
case 6:
//  Juny.
Tft.drawString("JUNY",100,50,2,BLACK);
alarmajuny = EEPROM.read(alarmames);
Tft.drawNumber(alarmajuny,170,95,2,BLACK);
break;
case 7:
//  Juliol.
Tft.drawString("JULIOL",100,50,2,BLACK);
alarmajuliol = EEPROM.read(alarmames);
Tft.drawNumber(alarmajuliol,170,95,2,BLACK);
break;
case 8:
//  Agost.
Tft.drawString("AGOST",100,50,2,BLACK);
alarmaagost = EEPROM.read(alarmames);
Tft.drawNumber(alarmaagost,170,95,2,BLACK);
break;
case 9:
//  Setembre.
Tft.drawString("SETEMBRE",100,50,2,BLACK);
alarmasetembre = EEPROM.read(alarmames);
Tft.drawNumber(alarmasetembre,170,95,2,BLACK);
break;
case 10:
//  Octubre.
Tft.drawString("OCTUBRE",100,50,2,BLACK);
```

```
            alarmaoctubre = EEPROM.read(alarmames);
            Tft.drawNumber(alarmaoctubre,170,95,2,BLACK);
            break;
            case 11:
            //  Novembre.
            Tft.drawString("NOVEMBRE",100,50,2,BLACK);
            alarmanovembre = EEPROM.read(alarmames);
            Tft.drawNumber(alarmanovembre,170,95,2,BLACK);
            break;
            case 12:
            //  Desembre.
            Tft.drawString("DESEMBRE",100,50,2,BLACK);
            alarmadesembre = EEPROM.read(alarmames);
            Tft.drawNumber(alarmadesembre,170,95,2,BLACK);
            break;
            default:
            break;
           }
  //TEXT PEU DE PANTALLA
  Tft.drawString("ALBERT BOIX 2012 SCCA",0,310,1,CYAN);
  consums=0;
}
if (y>180 && y<240 && pantalla==5)
                {
                delay(1000);
                mes++;
                if (mes==13){
                        mes=1;
                        }
```

```
                y=0;
                x=0;
                goto t;
                        }


//*****************************************MES CONSUMS GENER*********************************
alarmames=mes;
if (y>125 && y<185 && x>1 && x<119 && pantalla==5 && mes==1)
                                        {
                                         delay(50);
                                         alarmagener++;
                                         EEPROM.write(alarmames, alarmagener);
                                         y=0;
                                         x=0;
                                         goto z;
                                        }

 if (y>125 && y<185 && x>120 && x<239 && pantalla==5 && mes==1)
                                        {
                                         delay(50);
                                         alarmagener--;
                                         EEPROM.write(alarmames, alarmagener);
                                         y=0;
                                         x=0;
                                         goto y;
                                        }
//*****************************************MES CONSUMS FEBRER*********************************
if (y>125 && y<185 && x>1 && x<119 && pantalla==5 && mes==2)
```

```
                                    {
                                     delay(25);
                                     alarmafebrer++;
                                     EEPROM.write(alarmames, alarmafebrer);
                                     y=0;
                                     x=0;
                                     goto z;
                                    }
 if (y>125 && y<185 && x>120 && x<239 && pantalla==5 && mes==2)
                                    {
                                      delay(25);
                                      alarmafebrer--;
                                      EEPROM.write(alarmames, alarmafebrer);
                                      y=0;
                                      x=0;
                                      goto y;
                                    }

//*****************************************MES CONSUMS MARÇ*****************************************
if (y>125 && y<185 && x>1 && x<119 && pantalla==5 && mes==3)
                                    {
                                     delay(25);
                                     alarmamarc++;
                                     EEPROM.write(alarmames, alarmamarc);
                                     y=0;
                                     x=0;
                                     goto z;
                                    }
```

```
 if (y>125 && y<185 && x>120 && x<239 && pantalla==5 && mes==3)
                                        {
                                         delay(25);
                                         alarmamarc--;
                                         EEPROM.write(alarmames, alarmamarc);
                                         y=0;
                                         x=0;
                                         goto y;
                                        }

//*********************************************MES CONSUMS ABRIL*********************************
if (y>125 && y<185 && x>1 && x<119 && pantalla==5 && mes==4)
                                       {
                                        delay(25);
                                        alarmaabril++;
                                        EEPROM.write(alarmames, alarmaabril);
                                        y=0;
                                        x=0;
                                        goto z;
                                       }
 if (y>125 && y<185 && x>120 && x<239 && pantalla==5 && mes==4)
                                        {
                                         delay(25);
                                         alarmaabril--;
                                         EEPROM.write(alarmames, alarmaabril);
                                         y=0;
                                         x=0;
                                         goto y;
                                        }
```

```
//*****************************************MES CONSUMS MAIG*************************************
if (y>125 && y<185 && x>1 && x<119 && pantalla==5 && mes==5)
                                        {
                                         delay(25);
                                         alarmamaig++;
                                         EEPROM.write(alarmames, alarmamaig);
                                         y=0;
                                         x=0;
                                         goto z;
                                        }

 if (y>125 && y<185 && x>120 && x<239 && pantalla==5 && mes==5)
                                        {
                                         delay(25);
                                         alarmamaig--;
                                         EEPROM.write(alarmames, alarmamaig);
                                         y=0;
                                         x=0;
                                         goto y;
                                        }

//*************************************MES CONSUMS JUNY*************************************
if (y>125 && y<185 && x>1 && x<119 && pantalla==5 && mes==6)
                                        {
                                         delay(25);
                                         alarmajuny++;
                                         EEPROM.write(alarmames, alarmajuny);
                                         y=0;
                                         x=0;
```

```
                                goto z;
                               }

 if (y>125 && y<185 && x>120 && x<239 && pantalla==5 && mes==6)
                               {
                                delay(25);
                                alarmajuny--;
                                EEPROM.write(alarmames, alarmajuny);
                                y=0;
                                x=0;
                                goto y;
                               }

//************************************************MES CONSUMS JULIOL*****************************
if (y>125 && y<185 && x>1 && x<119 && pantalla==5 && mes==7)
                              {
                               delay(25);
                               alarmajuliol++;
                               EEPROM.write(alarmames, alarmajuliol);
                               y=0;
                               x=0;
                               goto z;
                              }

 if (y>125 && y<185 && x>120 && x<239 && pantalla==5 && mes==7)
                               {
                                delay(25);
                                alarmajuliol--;
                                EEPROM.write(alarmames, alarmajuliol);
```

```
                                    y=0;
                                    x=0;
                                    goto y;
                                   }

//*****************************************MES CONSUMS AGOST*********************************
if (y>125 && y<185 && x>1 && x<119 && pantalla==5 && mes==8)
                                   {
                                    delay(25);
                                    alarmaagost++;
                                    EEPROM.write(alarmames, alarmaagost);
                                    y=0;
                                    x=0;
                                    goto z;
                                   }

 if (y>125 && y<185 && x>120 && x<239 && pantalla==5 && mes==8)
                                   {
                                    delay(25);
                                    alarmaagost--;
                                    EEPROM.write(alarmames, alarmaagost);
                                    y=0;
                                    x=0;
                                    goto y;
                                   }

//*****************************************MES CONSUMS SETEMBRE*****************************
if (y>125 && y<185 && x>1 && x<119 && pantalla==5 && mes==9)
                                   {
```

```
                                    delay(25);
                                    alarmasetembre++;
                                    EEPROM.write(alarmames, alarmasetembre);
                                    y=0;
                                    x=0;
                                    goto z;
                                    }

 if (y>125 && y<185 && x>120 && x<239 && pantalla==5&& mes==9)
                                   {
                                    delay(25);
                                    alarmasetembre--;
                                    EEPROM.write(alarmames, alarmasetembre);
                                    y=0;
                                    x=0;
                                    goto y;
                                   }

//*********************************************MES CONSUMS OCTUBRE*****************************
if (y>125 && y<185 && x>1 && x<119 && pantalla==5 && mes==10)
                                  {
                                   delay(25);
                                   alarmaoctubre++;
                                   EEPROM.write(alarmames, alarmaoctubre);
                                   y=0;
                                   x=0;
                                   goto z;
                                  }
```

```
if (y>125 && y<185 && x>120 && x<239 && pantalla==5 && mes==10)
                                        {
                                         delay(25);
                                         alarmaoctubre--;
                                         EEPROM.write(alarmames, alarmaoctubre);
                                         y=0;
                                         x=0;
                                         goto y;
                                        }

//************************************MES CONSUMS NOVEMBRE************************************
if (y>125 && y<185 && x>1 && x<119 && pantalla==5 && mes==11)
                                       {
                                         delay(25);
                                         alarmanovembre++;
                                         EEPROM.write(alarmames, alarmanovembre);
                                         y=0;
                                         x=0;
                                         goto z;
                                        }

if (y>125 && y<185 && x>120 && x<239 && pantalla==5 && mes==11)
                                        {
                                         delay(25);
                                         alarmanovembre--;
                                         EEPROM.write(alarmames, alarmanovembre);
                                         y=0;
                                         x=0;
                                         goto y;
```

```
                                    }


//*************************************MES CONSUMS DESEMBRE**********************************
if (y>125 && y<185 && x>1 && x<119 && pantalla==5 && mes==12)
                                   {
                                    delay(25);
                                    alarmadesembre++;
                                    EEPROM.write(alarmames, alarmadesembre);
                                    y=0;
                                    x=0;
                                    goto z;
                                   }

 if (y>125 && y<185 && x>120 && x<239 && pantalla==5 && mes==12)
                                   {
                                    delay(25);
                                    alarmadesembre--;
                                    EEPROM.write(alarmames, alarmadesembre);
                                    y=0;
                                    x=0;
                                    goto y;
                                   }
//*****************************************************************************************************************
if (y>80 && y<129 && dataihora==1 && pantalla==5){
//*****************************************************************************************************************
//*********************************PANTALLA DATA I HORA*************************************
 pantalla++;
 pantalla++;
//DIBUIX RECTANGLE NEGRE PER BORRAR TEXT ANTERIOR
```

```
Tft.fillRectangle(0, 0, 239,25,BLACK);
//TEXT PANTALLA CAPÇALERA
Tft.drawString("PROGRAMACIO DATA I HORA",5,5,1,CYAN);
//DIBUIX RECTANGLE GRIS
Tft.fillRectangle(0, 15, 239,50,GRAY1);
//TEX RECTANGLE GRIS
Tft.drawString("DIA:",5,30,2,WHITE);
//DIBUIX RECTANGLE GRIS 2
Tft.fillRectangle(0, 65, 239,50,GRAY2);
Tft.drawString("MES:",5,90,2,WHITE);
//DIBUIX RECTANGLE GRIS
Tft.fillRectangle(0, 115, 319,50,GRAY1);
//TEX RECTANGLE GRIS
Tft.drawString("ANY:",5,135,2,WHITE);
//DIBUIXAR RECTANGLE GRIS
Tft.fillRectangle(0, 165, 319,50,GRAY2);
//TEXT RECTANGLE GRIS
Tft.drawString("HORA:",5,175,2,WHITE);
//DIBUIX RECTANGLE + / -
Tft.fillRectangle(0, 205, 119,50,GREEN);
Tft.fillRectangle(119, 205, 220,50,BLUE);
// ESCRIURE TEXT + / -
Tft.drawString("+",30,210,5,BLACK);
Tft.drawString("-",160,210,5,BLACK);
// DIBUIX RECTANGLE
Tft.fillRectangle(0, 255, 239,50,RED);
// TEXT RECTANGLE VERMELL
Tft.drawString("SORTIR",90,270,2,BLACK);
//TEXT PEU DE PANTALLA
```

```
Tft.drawString("ALBERT BOIX 2012 SCCA",0,310,1,CYAN);
dataihora=0;
  }

//*************************************************************************************************************
 //*************************************PANTALLA MANUAL*************************************************
  if (y>195 && y<250 && manual==1 && pantalla==4)
                                        {
                                         pantmanual();
                                        }

if (y>130 && y<194 && manual==1 && pantalla==4)
                                        {
                                        tempsobreconsums();// pantmanual();
                                        }
 //*************************************************************************************************************
if (tempsalarma==cronominut)
                        {
                        consumlitres=0;
                        cronominut=0;
                        }


        if (y>250)
        {
        principal=1;
        pantalla--;
        }
        if ( resultat3<consumlitres)
```

```
            {
             al=1;
            }
while (al==1)
      {
       digitalWrite(relevalvula, HIGH);
       //al=0;
       //DIBUIX RECTANGLE NEGRE PER BORRAR TEXT ANTERIOR
       Tft.fillRectangle(0, 0, 239,25,BLACK);
       delay(1000);
       digitalWrite(alarmapin,HIGH);
       delay(1000);
       digitalWrite(alarmapin,LOW);
       Tft.fillRectangle(0, 0, 239,319,RED);
       Tft.drawString("ALARMA",10,30,2,BLACK);
       Tft.drawString("VALVULA ",10,60,2,BLACK);
       Tft.drawString("TANCADA",10,90,2,BLACK);
       if (sms==0)
            {
             funEnviarSMS();
             Tft.drawString("SMS ENVIAT AL ",10,280,2,BLACK);
            }
       Point p = ts.getPoint();
       p.x = map(p.x, TS_MINX, TS_MAXX, 0, 240);
       p.y = map(p.y, TS_MINY, TS_MAXY, 0, 320);
       if (p.z > __PRESURE) {
                    x=(p.x);
                    y=(p.y);
                    al=0;
```

```
                    consumlitres=0;
                    cronominut=0;
                    principal=1;
                    pantalla=2;
                   }
              }
if (y>80 && y<145 && dataihora==1 && pantalla==4)
                                    {
                                    funciodia();
                                    }

if (y>250 && pantalla==4 || y>250 && pantalla==5)  //  BOTO SORTIR
                                   {
                                    Tft.drawNumber(pantalla,5,300,4,BLUE);//  Valor  de  litres
consumits.
                                    pantalla=2;
                                    }

//*************************************************************************************************************

          } //  Corxeta Final Programa.

//*************************************************************************************************************
```

10.Bibliografia

[1] Electan electrònica i robótica (setembre 2012) lloc web: http://www.electan.com/index.php

[2] Brico Geek botiga (setembre 2012) lloc web: http://www.bricogeek.com/shop/

[3] Adafruit industries (setembre 2012) lloc web: http://www.adafruit.com/

[4] Dealextreme (setembre 2012) lloc web: http://dx.com/

[5] Arduino (setembre 2012) lloc web: http://arduino.cc/en/

[6] Henning Karlsen (setembre 2012/maig 2013) lloc web: http://henningkarlsen.com/electronics/index.php

[7] Wiki Seeed Studio (setembre 2012) lloc web: http://www.seeedstudio.com/depot/

[8] Alpha Crucis (desembre 2012) lloc web: http://www.alpha-crucis.com/es/

[9] TrionixStuff (març 2013) lloc web: http://tronixstuff.wordpress.com/

[10] GSM note 19 (abril 2013) lloc web: http://note19.com/category/gsm/

[11] Tu electró(3 maig 2013)nica lloc web: http://www.tuelectronica.es/

[12] Dfrobot (maig 2013) lloc web: http://www.dfrobot.com/index.php

[13] Reflexiona (maig2013) lloc web:http://www.reflexiona.biz/