Universitat de les Illes Balears

escola politècnica superior

Treball Final de Grau

GRAU D'ENGINYERIA ELECTRÒNICA INDUSTRIAL I AUTOMÀTICA

Disseny de node IoT per control de fluxes de persones

JUAN ANTONIO OLIVER GARCIA

Tutor
Bartomeu Alorda Ladaria

Escola Politècnica Superior
Universitat de les Illes Balears
Palma, 8 de setembre de 2017

Als meus pares, els impulsors de tot,
a la meva germana per aguantar-me,
a la meva parella Marina pel suport desinteressat,
a tots els que m'heu recolzat en tots aquests anys
i amb distinció especial al meu padrí.

SUMARI

Sumari iii

Índex de figures vii

Acrònims ix

Resum xi

1 Introducció 1

2 Plataforma Sentilo 3
- 2.1 Introducció . . . 3
- 2.2 Plataforma Sentilo . . . 4
- 2.3 Enregistrament del Node . . . 5
- 2.4 Pujada de dades . . . 6

3 Disseny del hardware 9
- 3.1 Descripció del Hardware implementat . . . 9
- 3.2 Components . . . 10
  - 3.2.1 Microcontrolador: PIC16F1824 . . . 10
  - 3.2.2 Mòdul Wi-Fi: RN-XV . . . 10
  - 3.2.3 Receptor infraroig: TSOP4838 . . . 11
  - 3.2.4 Emissor infraroig: TSAL6200 . . . 13
  - 3.2.5 Placa solar . . . 15
  - 3.2.6 Bateria portàtil . . . 15
  - 3.2.7 Regulador 3.3V: NCP1117 . . . 16
  - 3.2.8 Transistors MOSFET . . . 16
  - 3.2.9 Retroreflectors . . . 18
- 3.3 Disseny del circuit . . . 19
  - 3.3.1 Alimentació . . . 21
  - 3.3.2 Microcontrolador i connector pickit 3 . . . 21
  - 3.3.3 Mòdul Wifi . . . 21
  - 3.3.4 Receptor IR . . . 22
  - 3.3.5 Emissor IR . . . 22
- 3.4 Disseny de la PCB . . . 22

4 Disseny del Sofware 25
- 4.1 Programació PIC16F1824 . . . 25

4.1.1 Hardware i software de programació . . . 25
4.2 Modul Rn-xv 171 . . . 29

5 Prototip **31**
5.1 Descripció . . . 31
5.2 Muntatge . . . 32

6 Conclusions **37**
6.1 Futur desenvolupament . . . 37

A Setup Sentilo **39**
A.1 Prerequisits . . . 39
A.1.1 JDK . . . 39
A.1.2 GIT . . . 39
A.1.3 Maven . . . 40
A.2 Descàrrega i compilació del codi . . . 40
A.3 Infraestructura de la plataforma . . . 41
A.3.1 Redis . . . 42
A.3.2 MongoDB . . . 44
A.3.3 Tomcat . . . 45
A.3.4 ElasticSearch . . . 47
A.3.5 MySQL . . . 48
A.3.6 OpenTSDB . . . 48
A.4 Configuracions . . . 53
A.4.1 Redis . . . 53
A.4.2 MongoDB . . . 55
A.4.3 MySQL . . . 55
A.4.4 Tomcat . . . 56
A.4.5 Elasticsearch . . . 57
A.4.6 OpenTSDB . . . 57
A.5 Instal·lació de la plataforma . . . 57
A.5.1 Instal·lació del Web Application Catalog . . . 58
A.5.2 Instal·lació del servidor de subscripció / publicació . . . 58
A.5.3 Instal·lació dels agents . . . 58
A.5.4 Instal·lació Alarm Agent . . . 59
A.5.5 Instal·lació Relational Agent . . . 59
A.5.6 Instal·lació Location Updater Agent . . . 60
A.5.7 Instal·lació Historian Agent . . . 60
A.5.8 Instal·lació Activity-Monitor Agent . . . 61
A.6 Proba de la plataforma . . . 61

B Codi programació PIC16F1824 **63**
B.1 main . . . 63
B.2 Init.h . . . 67
B.3 UART.h . . . 67
B.4 UART.c . . . 68

C Configuració mòdul WI-FI **69**

D Fabricació dels pals **71**

Bibliografia **77**

ÍNDEX DE FIGURES

1.1 Disseny del circuit 2

2.1 Pàgina inicial de la plataforma Sentilo 4
2.2 Enregistrament del proveïdor 5
2.3 ApiKey assignada al proveïdor. 5
2.4 Codi generat per Postman 6
2.5 Dada registrada al servidor Sentilo 7
2.6 Localització del sensor al mapa 7

3.1 Microcontrolador PIC16F1824 10
3.2 Wi-Fi RN-XV 11
3.3 TSOP4838 11
3.4 block diagram TSOP4838 12
3.5 Freqüència VS Resposta relativa 12
3.6 Longitud d'ona VS Sensibilitat relativa 13
3.7 TSAL6200 13
3.8 Corrent de pols VS duració del pols 14
3.9 Placa solar 15
3.10 Bateria 6600mAh 15
3.11 Regulador NCP1117 16
3.12 Transistor N-MOS i P-MOS 16
3.13 Input vs Output 17
3.14 Retroreflector 18
3.15 Principi d'un Retroreflector 18
3.16 Disseny del circuit 20
3.17 Disseny del PCB 23

4.1 Pàgina principal MPLAB 26
4.2 Programador Pickit 3 27
4.3 diagrama de flux del programa 28
4.4 Codi del microprocessador per a la petició 29

5.1 Universitat de les Illes Balears 31
5.2 Disseny dels pals 32
5.3 Disseny de la caixa 32
5.4 Caixa fabricada 33
5.5 Suport placa solar i bateria 33

5.6 Suport dels sensors . . . 34
5.7 Suport del retrorreflectors . . . 34
5.8 Muntatge complet del pals . . . 35

A.1 Versió MongoDB . . . 44
A.2 Tomcat7 . . . 47
A.3 Zookeeper . . . 49
A.4 Zookeeper . . . 50
A.5 Paràmetres per defecte Redis . . . 53
A.6 Configuració Redis . . . 54
A.7 Configuració de Redis a Sentilo . . . 54
A.8 Paràmetres per defecte MongoDB . . . 55
A.9 Configuració de MongoDB a Sentilo . . . 55
A.10 Paràmetres per defecte MySQL . . . 55
A.11 Configuració de MySQL a Sentilo . . . 56
A.12 Configuració de Tomcat a Sentilo . . . 56
A.13 Arxius de configuració Tomcat a Sentilo . . . 56
A.14 Script de modificació . . . 57
A.15 Arxius de configuració Elasticsearch a Sentilo . . . 57
A.16 Script de modificació . . . 57
A.17 Arxius de configuració OpenTSDB a Sentilo . . . 57
A.18 Subdirectori de l'artefacte WAR . . . 58
A.19 Resposta a la comprovació de la plataforma . . . 61

B.1 main.c . . . 63
B.2 main.c . . . 64
B.3 main.c . . . 65
B.4 main.c . . . 66
B.5 Init.h . . . 67
B.6 UART.h . . . 67
B.7 UART.c . . . 68

C.1 Configuració mòdul . . . 69
C.2 Configuració mòdul . . . 70

D.1 Fabricació de la caixa . . . 72
D.2 Caixa fabricada . . . 72
D.3 Fabricació dels pals . . . 73
D.4 Introducció de la placa als pals . . . 73
D.5 Guia per al suport dels sensors . . . 74
D.6 Sensors muntats al pal . . . 74
D.7 Muntatge dels retroreflectors . . . 75

ACRÒNIMS

HTTP Hypertext Transfer Protocol

IDE Entorno de Desarrollo Integrado

IoT Internet of things

IP Internet Protocol

JSON JavaScript Object Notation

LED Light-Emitting Diode

MOSFET Metal-oxide-semiconductor Field-effect transistor

PCB Printed Circuit Board

PIC Programmable Interrupt Controller

PWM Pulse-Width Modulation

SGBDR Sistema de Gestió de Bases de Dades Relacionals

SMD Surface-Mount Device

SOIC Small-Outline Integrated Circuit

TFG Treball Final de Grau

UART Universal Asynchronous Receiver-Transmitter

USB Universal Serial Bus

RESUM

Clarament estem en un temps on, tecnològicament xerrant, se'ns ofereix molt, exigim molt, i convertim aquesta 'exigència' en una necessitat. Tant és aquesta 'necessitat' que tenim de fer-nos la vida més fàcil, que intencionadament intentam rompre totes les fronteres que ens limiten tenir la informació a l'abast de la nostra mà. Una manera de rompre aquestes fronteres és el que avui en dia moltes ciutats han aconseguit, convertint-se en les anomenades Smart Cities.

Les Smart Cities [1] sorgeixen de proporcionar una bona qualitat de vida als ciutadans, una qualitat de vida que no només depèn de la dotació d'infraestructures físiques, serveis i suports institucionals, sinó que cada vegada més incideix sobre la disponibilitat i qualitat de les comunicacions, la interacció del ciutadà amb els diversos elements urbans , així com la transmissió del saber.

Aquesta interacció amb el ciutadà necessita una interconnexió d' 'objectes quotidians' anomenats sensors amb internet, alternativament, Internet of things (IoT) [2][1] es el punt on es connectaran aquests sensors.

Donat que les Smart Cities [1] es un tema en recent tractament i evolució s'ha volgut dedicar aquest Treball Final de Grau (TFG) al disseny e implementació d'un sensor capaç de quantificar el flux de persones que passen per un accés, enviant aquestes dades a una plataforma anomenat Sentilo[2] al qual es podrà tenir accés des de qualsevol dispositiu amb connexió a xarxa per posteriorment tractar aquestes dades o simplement per obtenir de forma automàtica l'afluència en un moment determinat o orientar-se sobre la concurrència d'un lloc.

[1] Concepte que es refereix a la interconnexió digital d'objectes quotidians amb internet

[2] Arquitectura open-source desenvolupada per recol·lectar, explotar i difondre informació generada per sensors desplegats en una ciutat.

CAPÍTOL 1

INTRODUCCIÓ

Vivim en un món on l'accés a la informació i extreure coneixement a partir d'aquesta es molt important. Actualment es genera informació de moltes fonts diverses. Un exemple es basa en la gestió del flux de persones, de vital importància per la presa de decisions. La mesura de la quantitat de persones que passen per un accés o accedeixen a un lloc té aplicacions en sectors, com per exemple el sector turístic que es mostren interessats en conèixer la afluència del flux de gent en uns mesos o unes hores concretes.

Per tant en aquest projecte l'objectiu principal és el de dissenyar i implementar un prototip capaç de comptar de forma autònoma la quantitat de persones que accedeixen al campus sobre el passeig central entre els edificis de Ramon Llull i Mateu Orfila i així conèixer el flux de persones.Degut a les diferents alternatives de sensorització que existeixen; ultrasons, mesures de pressió, processat d'imatges, infrarojos... en aquest projecte, es defineix l'objectiu d'explorar l'ús de la tecnologia d'infrarojos.

S'utilitzaran dos emissors i dos receptors infrarojos per tal de poder saber la direcció de la persona. Aquests sensors aniran muntats en paral·lel de forma que a l'enfront d'aquests aniran uns reflectors òptics amb els quals es crearà una barrera infraroja que al ser travessada per una persona tallarà aquesta barrera i el receptor detectarà el canvi òptic.

Aquest prototip anirà muntat dins uns pals enfrontats de fusta com es pot veure a la imatge per no danyar l'impacte visual de l'entorn, dins un dels pals anirà incorporat la placa impresa amb els sensors, la placa solar i la bateria que proporcionarà l'energia necessària. Mentre que a l'altre pal tan sola aniran uns reflectors on rebotarà el llum que incideix sobre ells per tal de tornar-ho de volta als receptors òptics del pal.

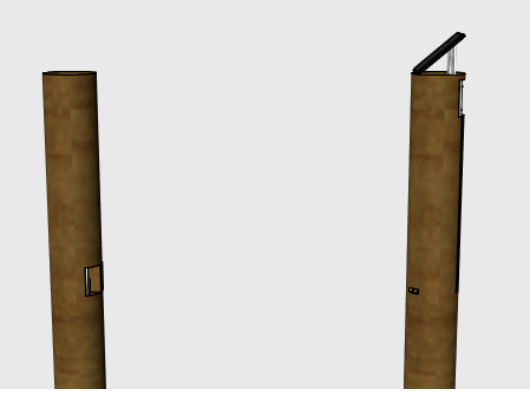


Figura 1.1: Disseny del circuit

Tot aquest muntatge necessitarà de peces fetes a mida amb unes formes no convencionals pel que es farà ús d'una impressora 3d que fabricarà les peces que anteriorment s'haurà dissenyat a ordinador amb les mesures exactes

Una vegada fabricats els pals, inserits els sensors i bateria caldrà fer la instal·lació d'una plataforma la qual serà l'encarregada de mostrar les dades captades per els sensors, aquesta plataforma s'instal·larà a un ordinador de la UIB el qual ens farà de servidor en aquest projecte per rebre les dades. S'utilitzarà la plataforma Sentilo que es va desenvolupar a la ciutat de Barcelona per crear les ciutats intel·ligents.

Finalment per una part tenim el dispositiu amb els sensors i per l'altra el servidor, pel que necessitarem una eina per tal de comunicar les dues part, això serà possible gràcies a un mòdul WI-FI el qual s'incorporarà a la placa de sensors i es donarà d'alta a la xarxa del campus.

Les tasques previstes en aquest projecte es centren en l'anàlisi ,disseny e implementació d'un sistema complet de recollida de dades procedent dels sensors, els quals dipositaran en temps real els diferents valors obtinguts a una base de dades i posteriorment la visualització en un mapa de la localització d'aquest dispositiu, tot això es farà possible gràcies a la implementació de la plataforma Sentilo [3] [1]. El sistema tindrà elements principals com per exemple:

- Plataforma encarregada de rebre i contenir de forma eficient i segura les dades.
- Proces de publicació en temps real mitjançant xarxa tota la informació obtinguda.
- Procés de publicació de dades històriques.
- Mapa web de publicació en temps real i històric.

[1] Arquitectura open-source desenvolupada per recol·lectar, explotar i difondre informació generada per sensors desplegats en una ciutat.

CAPÍTOL 2

PLATAFORMA SENTILO

2.1 Introducció

Inicialment Sentilo [3] va neixer com un projecte intern de l'Ajuntament de Barcelona, que va decidir obrir el producte mitjançant la creació d'una comunitat de software lliure, orientada a la promoció i l'evolució de la plataforma.

Es una arquitectura open-source[1] desenvolupada per recol·lectar, explotar i difondre la informació generada pels sensors desplegats en una ciutat.

L'objectiu de Sentilo es servir com a plataforma de interoperabilitat[2]de la Smart City [1] [3] evitant solucions verticals i actuant com plataforma cross permetent una fàcil interoperabilitat.

Entre les principals característiques destaquen:

- Construida sobre una plataforma sòlida.
- Alt rendiment: Dissenyat per processar infinitat de missatges.
- Modular i ampliable:Arquitectura basada en agents, facilitant la incorporació de noves funcionalitats sense modificar el sistema central.
- Inversions i costos de manteniment reduïts.
- Reduir la dependència de tecnologies específiques, solucions o proveïdors.

[1] (Codi obert) es el terme amb el qual es coneix al software distribuït i desenvolupat lliurement. Es més orientat als beneficis pràctics de compartir codi.

[2] Habilitat de dos o més sistemes o components per intercanviar informació i utilitzar aquesta informació.

[3] Ciutats que apliquen les tecnologies TIC amb l'objectiu d'oferir un desenvolupament sostenible i un increment de la qualitat de vida.

2.2 Plataforma Sentilo

Per al projecte exposat s'ha fet ús de la plataforma Sentilo. Els creadors de la plataforma ens proporcionen a la pàgina web una màquina virtual, la qual es pot descarregar i instal·lar de forma molt senzilla tal com exposen els creadors.

Maldament en aquest cas fos més que necessari la instal·lació d'aquesta màquina virtual, s'ha volgut anar més enfora del necessari i crear un servidor físic en un ordinador proporcionat per la Universitat de les Illes Balears.

La instal·lació de la plataforma Sentilo ha estat tot un repte, ja que s'ha fet servir un ordinador al qual es va instal·lar el sistema operatiu Ubuntu 16.04 basat en Linux. Majorment el repte ha estat començar a fer feina amb aquest tipus de sistema operatiu el qual es molt funcional mitjançant les comandes i deixant enrere tota la interfície gràfica a la qual estam molt acostumats.

Tota la instal·lació d'aquesta plataforma[4] està descrita pas a pas en l'Annex A d'aquesta memòria. Es pot veure totes les comandes necessàries i els programes emprats per replicar en un altre ordinador en el cas de desitjar-se.

Una vegada seguit tot el proces d'instal·lació i configuració de la plataforma Sentilo podem veure la pàgina inicial entrant al navegador a la direcció *localhost:8080/sentilo-catalog-web/*. El port 8080 es el port al qual tenim *Tomcat*. A la figura 2.2 es pot veure aquesta pàgina, amb el color per defecte canviat i el nom dels integrants els quals han instal·lat i configurat aquest servidor a una màquina de la UIB.

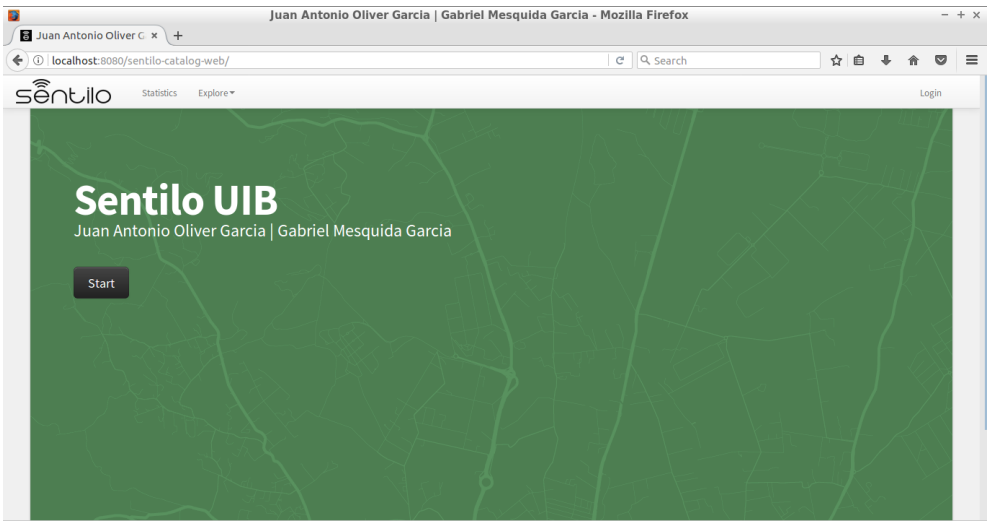


Figura 2.1: Pàgina inicial de la plataforma Sentilo

2.3 Enregistrament del Node

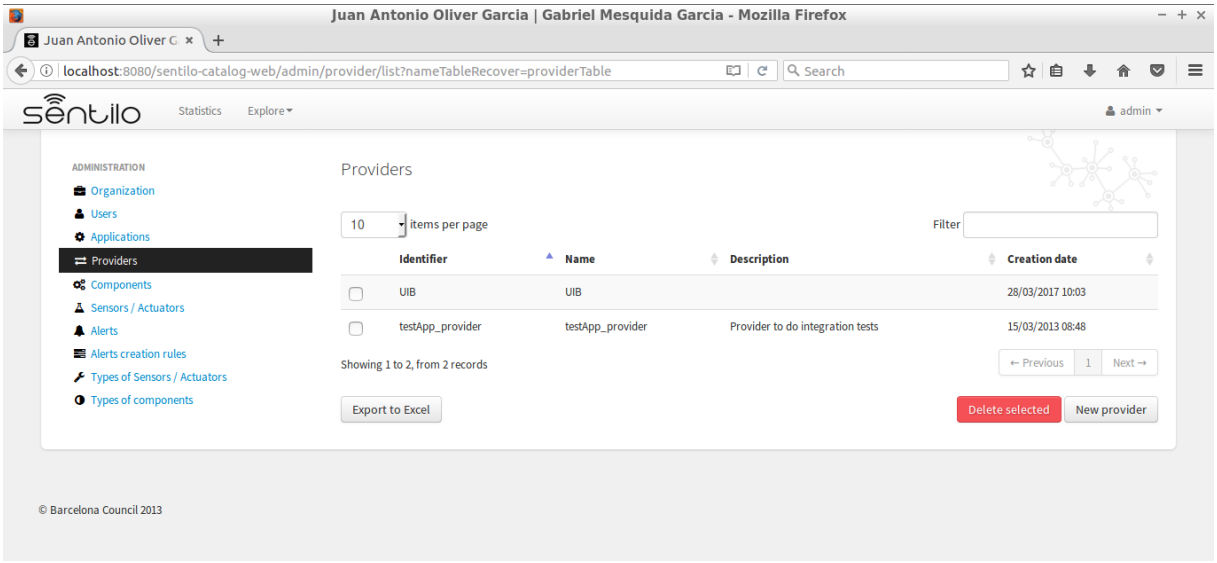


Figura 2.2: Enregistrament del proveïdor

Una vegada Instal·lada i comprovada la plataforma, devem enregistrar el node com a un proveïdor com es veu a la figura 2.2 . Això el dotarà d'un *token* el qual serà el seu identificador dins la plataforma. Aquesta *ApiKey* 2.3 es assignada automàticament per la plataforma i es única per a cada un dels que es creïn.

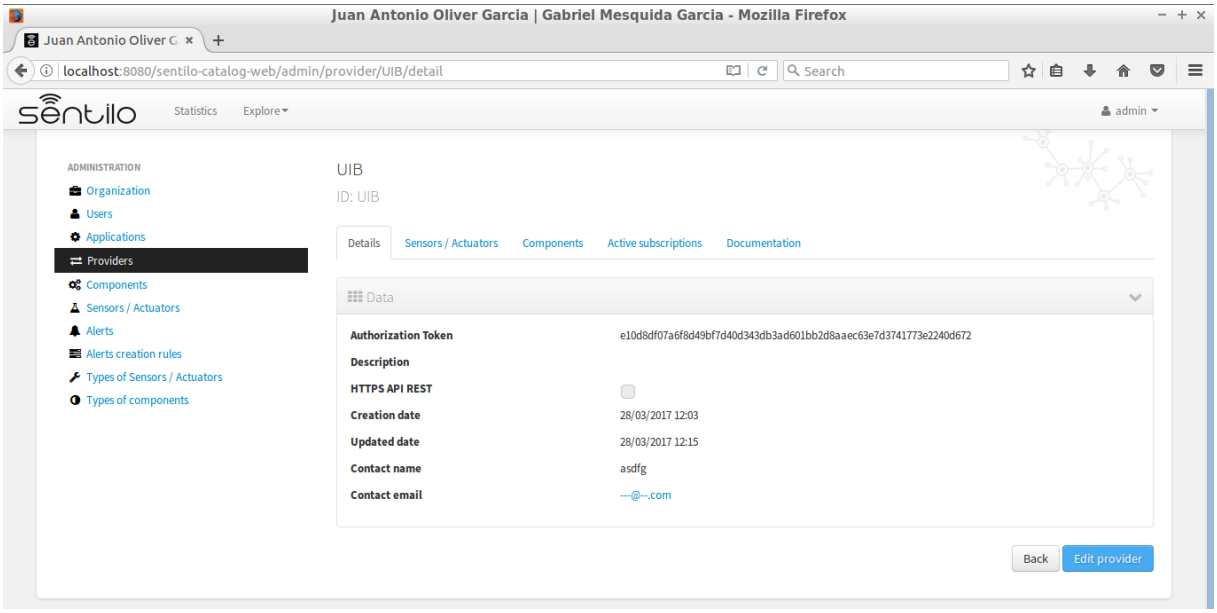


Figura 2.3: ApiKey assignada al proveïdor.

Una vegada donat d'alta el proveïdor a la plataforma serà necessari assignar-li un *Component* el qual representa el node del projecte. Per la creació del component es necessari saber si la posició serà mòbil o fixa, en aquest cas al ser fixa, caldrà introduir les coordenades de la posició, també es necessari associar-lo al *Provider*. També hi ha altres configuracions opcionals com una breu descripció, la forma d'alimentació del component...

La darrera passa es la creació del *sensor*. En aquest pas es donen d'alta els sensors que componen en node, en aquest cas, només es necessari donar-ne un, però hi ha situacions en les quals serà necessari donar d'alta diferents sensors per a un mateix node.

Per donar d'alta els sensor, a la pestanya es selecciona l'etiqueta de *sensors* i es selecciona el tipus de sensor, les unitats en les quals treballa i s'associa al *Provider* corresponent.

2.4 Pujada de dades

Una vegada tenim enregistrat el node a la plataforma Sentilo es poden fer proves de transmissió de dades, per exemple amb la utilització d'un programa anomenat *Postman* [5] el qual es capaç de fer peticions HTTP a una IP donada i en aquest cas també una *apiKey* la qual ha proporcionat la plataforma Sentilo com a identificador del proveïdor de dades. A la següent figura es pot veure el codi generat pel programa esmentat a l'hora de fer la petició:

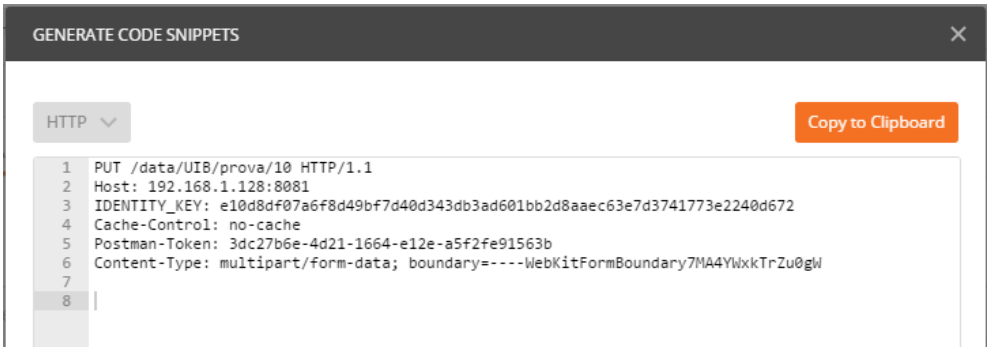


Figura 2.4: Codi generat per Postman

A la figura 2.5 i 2.6 es pot veure la recepció de les dades i la visualització del sensor sobre el mapa respectivament. Una vegada verificada la recepció de les dades al nostre servidor només queda introduir aquesta petició al mòdul WI-FI per tal de fer els enviaments de les dades necessàries i la posterior visualització.

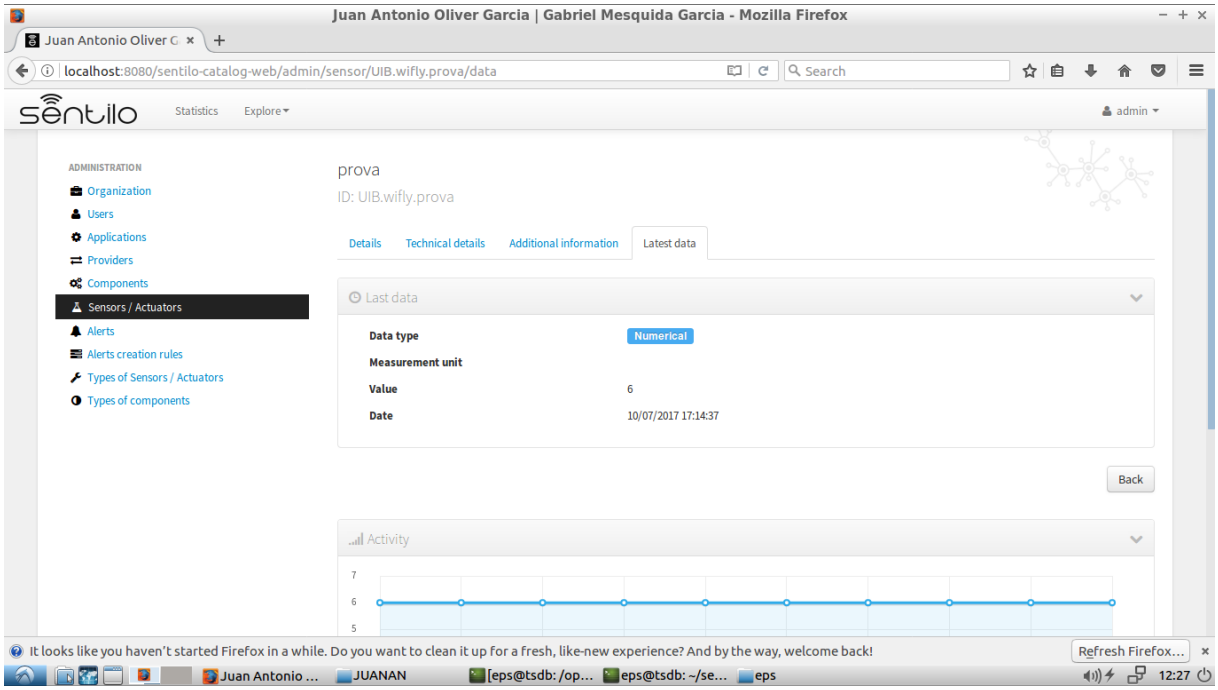


Figura 2.5: Dada registrada al servidor Sentilo

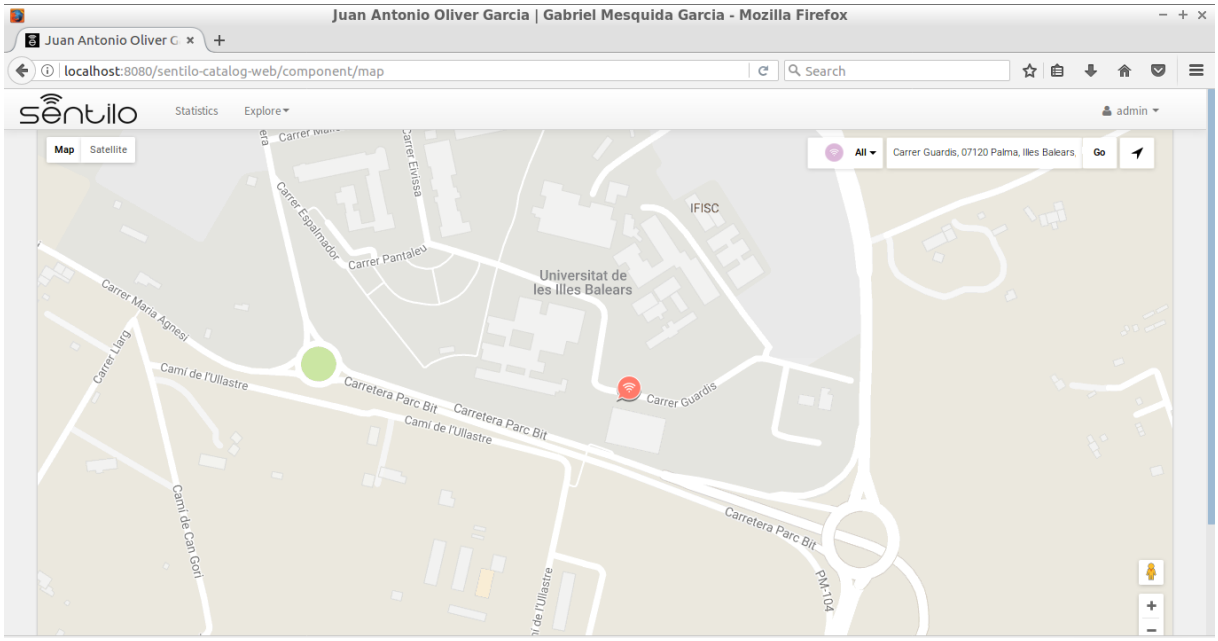


Figura 2.6: Localització del sensor al mapa

CAPÍTOL 3

DISSENY DEL HARDWARE

3.1 Descripció del Hardware implementat

Per al disseny hardware del circuit seran necessaris una sèrie de components tant passius com actius. Serà necessària una etapa reguladora (Font d'alimentació) la qual serà l'encarregada de regular la tensió amb la que s'alimenta la placa a una tensió de funcionament òptima per als components que formen el circuit.

El circuit necessitarà un component el qual podem anomenar cervell, aquest component serà un microcontrolador del fabricant *Microchip* que controlarà els sensors infrarojos, durà a terme l'anàlisi de les lectures dels sensors i serà l'encarregat de comanar a un mòdul WI-FI que vagi l'enviament de certes dades.

El components més restrictiu és el mòdul Wi-Fi, el qual només es pot alimentar a 3,3V, el qual serà l'encarregat de fer una petició al servidor web sempre i quan el microcontrolador li comani.

I finalment tenim els sensors infrarojos com a components importants que componen la placa ls quals seran els encarregats de fer la detecció del flux de persones.
El circuit a més disposa de components com:

- Microcontrolador PIC.
- Mòdul Wi-Fi.
- Regulador.
- Bateria portàtil.
- Receptor infraroig.
- Emissor Infraroig.

- Components pasius (resistències, condensadors...).
- Retroreflectors.
- Sockets, terminals...

3.2 Components

3.2.1 Microcontrolador: PIC16F1824

És necessari un microcontrolador que disposi de les següents característiques com a mínim:

- 6 Entrades/Sortides digitals per tal de poder controlar i llegir una sèrie de components/sensors
- Comunicació UART per interactuar amb el mòdul Wi-Fi.
- Capaç de aportar una modulació per amplada de polsos (PWM) de 38Khz per controlar un emissor infraroig tal com es pot veure a l'apartat3.2.4.

Per fer front a aquestes peticions el microcontrolador triat forma part de la gama PIC16F del fabricant Microchip de 8 bits, els quals incorporen una gran varietat d'opcions d'encapsulat i contactes.

En aquest cas s'ha optat pel PIC16F1824[6] (veure Fig 3.1) amb el disseny Small-Outline Integrated Circuit (SOIC) de 14 pins de muntatge superficial(SMD) amb una possible alimentació entre el rang de valors 1,8 a 3,6V els quals son aptes per la restricció del mòdul Wi-fi (3,3V).

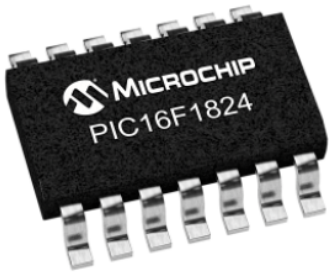


Figura 3.1: Microcontrolador PIC16F1824

3.2.2 Mòdul Wi-Fi: RN-XV

Aquest component (veure Fig 3.2) es el que dotarà de connexió al node, ja que es necessari per fer ús del servidor explicat a l'apartat 2, es un mòdul de Roving Networks el qual es una bona solució Wi-Fi. Aquest mòdul està basat en el robust mòdul RN-171[7] del mateix fabricant. En la configuració més simple tan sols necessita 4 pins per funcionar

(PWR, TX, RX i GND) a part de una pre-configuració. Com ampliació cal destacar que disposa de pins digitals, pins de gestió d'energia i pins analògics per a sensors externs.

Figura 3.2: Wi-Fi RN-XV

Aquest es el component més restrictiu de tota la placa, ja que la seva alimentació només permet aplicar una tensió de 3,3V

3.2.3 Receptor infraroig: TSOP4838

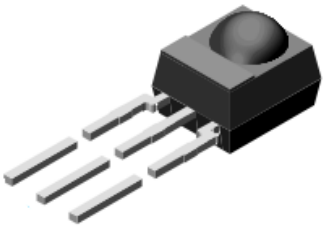


Figura 3.3: TSOP4838

És un mòdul receptor d'infrarojos[8] com el de la figura 3.3 del fabricant Vishay per sistemes de control remot per infrarojos. Primerament podem comentar que la part més externa d'aquest sensor -l'encapsulat- conté un filtre IR a 950nm de longitud d'ona. A més a més, aquest sensor te una implementació molt interessant que s'explica a continuació:

En la figura Fig 3.4 podem veure el block diagram en el qual veiem una sèrie de blocs que en un primer moment ens criden l'atenció.

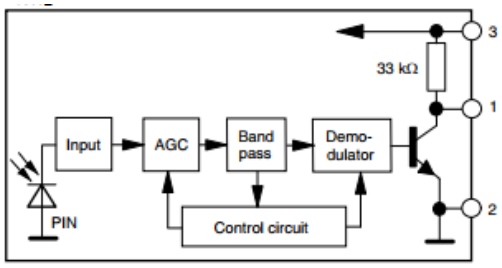


Figura 3.4: block diagram TSOP4838

Es pot observar un diode receptor, un modul de guany (amplificador) el qual introdueix un guany al senyal del LED receptor, un filtre passa bandes i un demodulador, essent aquest dos darrers mòduls els més interessants.

Com s'ha comentat anteriorment aquest mòdul consta d'un filtre físic per radiació infraroja, però també conté un altre tipus de filtratge depenent de la freqüència d'entrada de la llum, en aquest cas, es xerra d'un filtre passa bandes, el qual te una freqüència de ressonància de 38Khz (freqüència central), amb la qual cosa es pot despreciar un suposat cas de radiació infraroja a la mateixa longitud d'ona provinent d'altres fonts.

Al ser una implementació d'un filtre passa bandes, no existeix un filtratge ideal, per la qual cosa també deixa passar les components freqüencials pròximes a la freqüència de ressonància. Això vol dir que no només deixa passar la freqüència pel que ha estat dissenyat, sinó que també deixa passar freqüències properes però amb una resposta relativa inferior. A la figura 3.5 es pot veure una gràfica de la seva resposta relativa en funció de la freqüència.

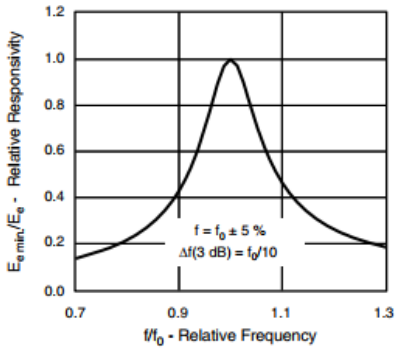


Figura 3.5: Freqüència VS Resposta relativa

Es pot veure que no només deixa passar la freqüència per la qual ha estat dissenyat, sinó que també deixa passar components pròximes a la freqüència de ressonància, però s'observa que la seva resposta relativa es desploma molt ràpidament.

De la mateixa manera que té una variabilitat de la resposta en funció de la freqüència, aquest mòdul també té una variació de la sensibilitat en funció de la longitud d'ona que incideix sobre el receptor, aquesta funció es pot veure a la figura 3.6, la qual es mostra a continuació:

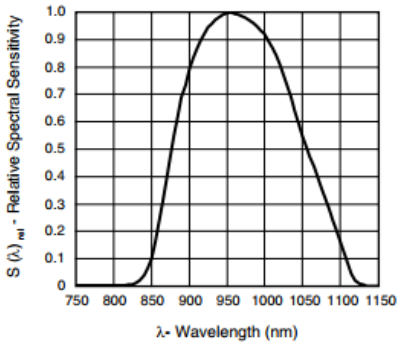


Figura 3.6: Longitud d'ona VS Sensibilitat relativa

El darrer dels mòdul i no menys important és el demodulador, el qual fa la funció de fer un *Output* constant sempre i quan al receptor incideixi una llum infraroja d'un 950nm a una freqüència molt propera als 38Khz, això fa que no es tengui que processar una sortida polsant, sinó que ja es té una sortida lògica constant (0 ò 1) amb la qual es pot fer feina.

3.2.4 Emissor infraroig: TSAL6200

Figura 3.7: TSAL6200

Aquest component[9] tal com el de la figura 3.7 correspon al mateix fabricant que el receptor anterior -veure figura 3.3-, té la particularitat de produir un llum infraroig d'una longitud d'ona de 940nm, la qual es semblant a la que fa feina el receptor.

Aquest emissor al ser un diode es pot modificar la seva potència fent ús del *Duty Cycle* d'un senyal quadrat, amb el qual es pot donar pics de corrent sense fer malbé al component i també es molt positiu alhora de alimentar el circuit amb placa solar i bateria. Amb aquesta regla de funcionament s'aconsegueix tenir una major potència lumínica del component, ja que l'increment de corrent en estat estacionari està limitat

per la màxima dissipació de potència del LED en qüestió.

En aquest mode polsant es pot arribar a aconseguir una major intensitat lumínica sense aver d'arribar al límit de potència dissipable. Tot això es possible perquè la potència mitja dissipada en condicions polsants d'un LED té en compte la intensitat mitja que circula pel dispositiu, la qual es directament proporcional al *Cicle de Treball* tal com es pot veure a la següent expressió:

$$i_{avg} = i_p d_c$$

on i_{avg} es el corrent promig, i_p es el corrent pic i d_c es el Cicle de treball.

on el Cicle de treball que com es pot veure en la següent expressió es la relació que hi ha entre el temps que el senyal es troba en estat actiu i el període de la mateixa expressat en percentatge:

$$d_c = t_{on} / T$$

Una vegada coneixem el funcionament teòric del motiu pel qual es millor el funcionament per polsos per aquest tipus de dispositius podem veure al datasheet que el fabricant ens proporciona una gràfica del corrent que pot suportar el component en funció de la duració del pols, a la qual te en compte el Duty Cycle:

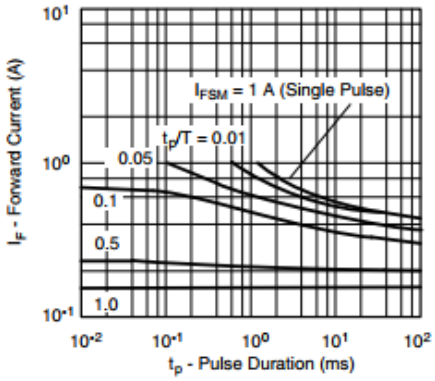


Figura 3.8: Corrent de pols VS duració del pols

En el gràfic es pot veure que en funció de la duració dels polsos que es facin i el duty cycle amb el que es treballi es té un corrent limit amb el que excitar el component sense cremar-ho, en aquest cas en el que necessitem uns 38Khz de període encara que facem feina amb d'un duty proper al 100% ens posicionaríem a la banda de més de l'esquerra del gràfic, i al introduir un cicle de treball, aquesta duració de pols encara seria menor, pel que encara ens posicionaríem a la banda més propera de l'esquerra.

3.2.5 Placa solar

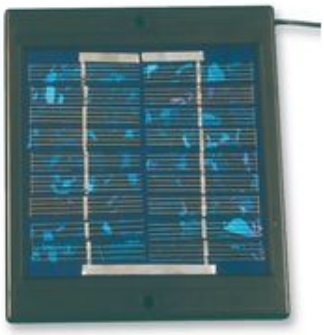


Figura 3.9: Placa solar

En la figura 3.9 es pot veure la placa solar utilitzada en el projecte, la qual s'encarregarà de captar l'energia de la radiació solar. Això dotarà al projecte d'energia sense haver d'estar connectat a xarxa i així posicionar-ho a qualsevol lloc.

3.2.6 Bateria portàtil

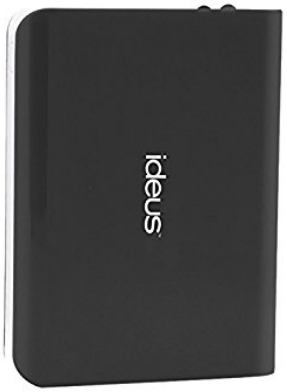


Figura 3.10: Bateria 6600mAh

La bateria de la figura 3.10 serà la que s'encarregarà de dotar d'energia als sensors, la qual es carregarà a partir d'una placa solar. Aquesta bateria te una capacitat de 6600mAh els quals son suficients per garantir una continua alimentació a la placa encara que hagi algun dia de poca energia solar o per passar els vespres.

En aquesta bateria no es necessari introduir un controlador de càrrega entre ella i la placa solar, ya que internament ja duu un controlador integrat el qual s'encarregarà de no sobrecarregar la bateria i de no descarregar-la els moments de poca energia solar o vespres.

3.2.7 Regulador 3.3V: NCP1117

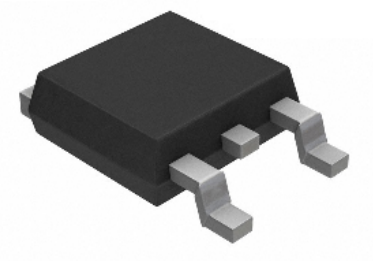


Figura 3.11: Regulador NCP1117

Aquesta serie de reguladors[10] del fabricant ON Semiconductor tal com es pot veure a la figura 3.11 son reguladors de voltatge positius de *Low-Droput* que son capaços de proporcionar un corrent de sortida superior a 1A. Aquesta serie està disponible en vàries tensions de sortida, en aquest cas es treballarà amb el regulador el qual proporciona una tensió de sortida de 3,3V ja que aquesta es la més restrictiva del circuit degut al mòdul WI-Fi.

3.2.8 Transistors MOSFET

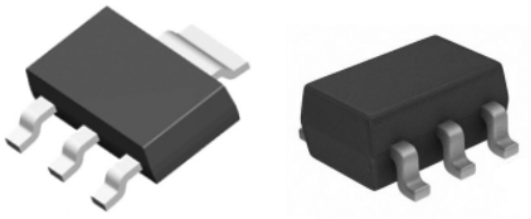


Figura 3.12: Transistor N-MOS i P-MOS

Aquests components son transistors de tecnologia MOSFET els quals a la figura 3.12 es pot veure un transistor N-MOS [11] i un P-MOS[12] respectivament.

En aquest circuit aquests components s'han muntat com un inversor, el qual serà l'encarregat de fer una ona quadrada de potència per alimentar els emissors infrarojos, ja que no és possible alimentar els emissors directament del microprocessador al estar aquest limitat a un corrent de sortida de 25mA. El que fa aquest muntatge es pot veure a la figura 3.13, quasi es podria definir com un seguidor-inversor de potència ja que el que fa es invertir el senyal d'entrada però amb una potència molt superior a la possible pel microprocessador.

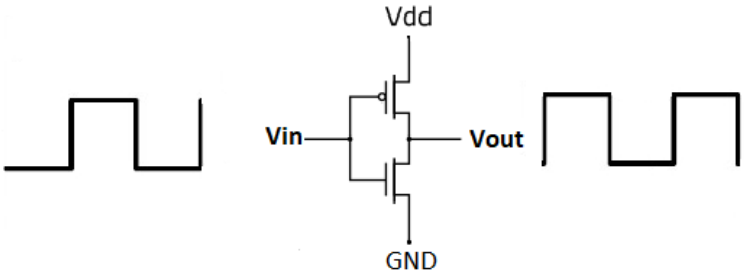


Figura 3.13: Input vs Output

Els motius pels quals s'ha optat a utilitzar transistors de tecnologia MOSFET es que tenen una sèrie d'avantatges respecte als transistors bipolars:

- Control per tensió: Aquests transistors es controlen per tensió i no per corrent pel qual tenen una impedància molt gran a l'entrada de la porta, el qual fa que sigui més senzill el controlar-los ja que el corrent que circula per la porta s'expressa en nanoampers.

- Encara que en aquest cas no es treballi a velocitats altes ,electrònicament parlant, aquests transistors tenen una velocitat de commutació elevada, de l'ordre de nanosegons, la qual serà necessària alhora d'excitar els emissors a la freqüència de treball.

- Molt baix consum en mode estàtic. Aquests transistors fan un pic de corrent en el moment de la commutació, una vegada s'estabilitza aquest consum es mínim els quals son molt recomanables per a sistemes alimentats amb bateries com es aquest cas.

Com s'ha especificat a l'apartat 3.2.4 l'emissor amb el que es treballa permet aplicar un corrent en funció del seu Duty Cycle. Aquest corrent no es pot controlar amb el transistor MOSFET, pel que es controla mitjançant una resistència que s'inclourà a la sortida de l'inversor, la qual limitarà el corrent de sortida sabent la llei d'Ohm i la tensió de caiguda en aquesta resistència que es col·locarà en serie amb l'emissor.

3.2.9 Retroreflectors

Figura 3.14: Retroreflector

Aquest element com el mostrat a la figura 3.14 tenen la particularitat de reflectir el llum que li arriba de volta a la font d'emissió sense importar l'angle d'incidència.

Hi ha diferents tipus de retroreflectors, però en aquest cas els utilitzats son els retroreflectors de cantonada, es pot dir que son tres miralls perpendiculars tots tres entre si de manera que formen una cantonada, el que fa que el feix de llum sigui reflectit de tornada a la font.

A la figura 3.15 es pot veure el principi de funcionament d'aquests elements, senzillament es pot dir que el primer feix incideix en un dels miralls i es reflecteix amb el mateix angle d'entrada cap a l'altre mirall el qual fa el mateix retornant el feix cap a la font d'emissió.

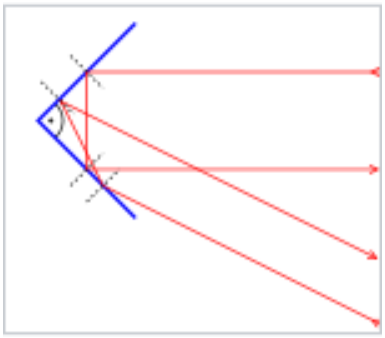


Figura 3.15: Principi d'un Retroreflector

3.3 Disseny del circuit

Per al disseny del circuit d'aquest projecte s'ha fet ús d'un software anomenat EAGLE [13], el qual es un programa especific per al disseny de circuits electrònics y PCBs. EAGLE conté un editor de diagrames electrònics, al qual els components poden ser col·locats senzillament i es poden enrutar a altres components.

Aquest software també conté un editor de PCBs amb enrutador molt eficient, aquest editor es capaç de produir arxius GERBER i demés, els quals son utilitzats en el moment de la producció o fabricació d'aquestes plaques.

A continuació es pot veure el disseny del node on més endavant s'explicaran els *subcircuits*:

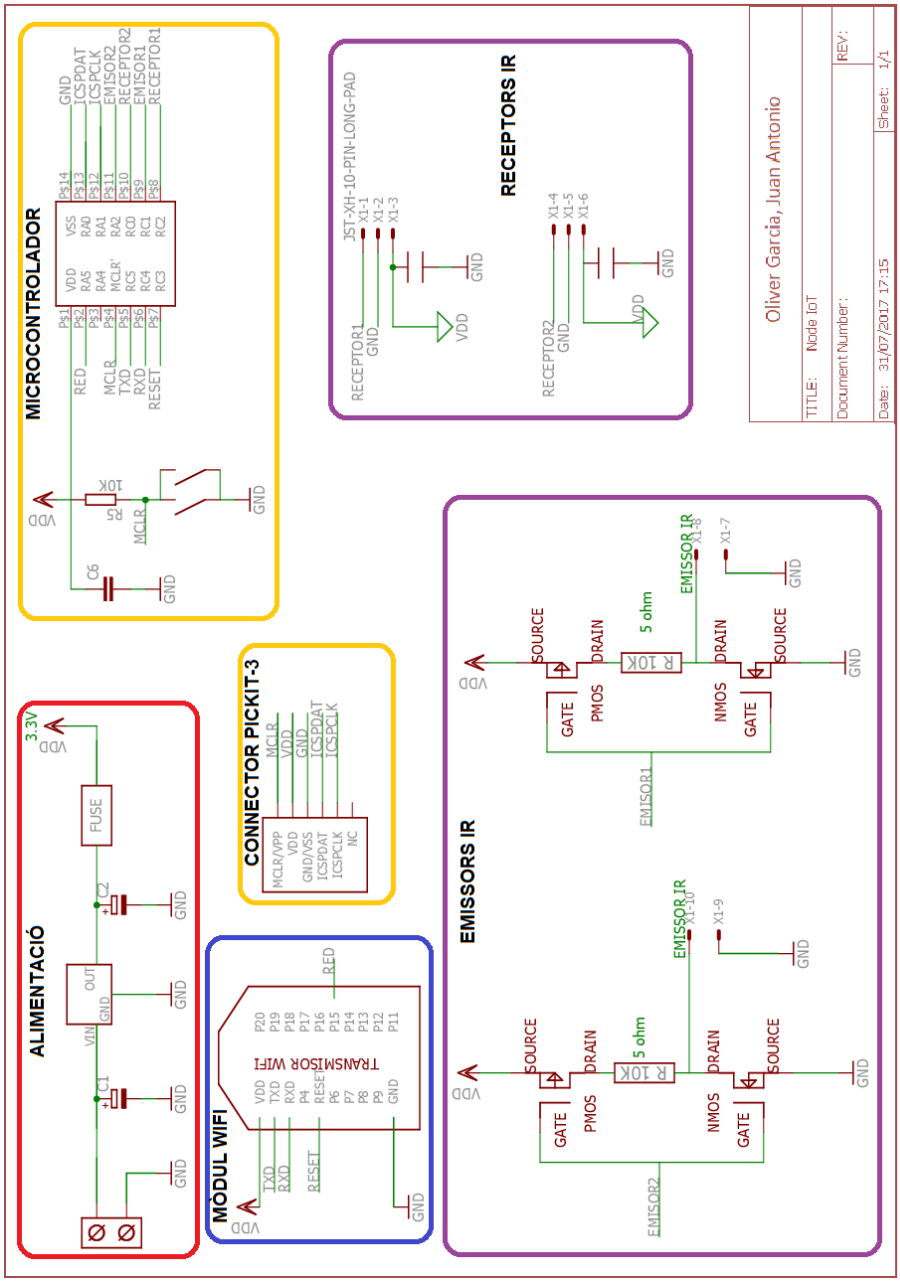


Figura 3.16: Disseny del circuit

3.3.1 Alimentació

A la figura 3.16 podem veure envoltat de color groc la font d'alimentació, la qual s'encarregarà d'alimentar el circuit a una tensió de 3.3V. El dispositiu principal d'aquesta font es el regulador el qual queda explicat a l'apartat 3.2.7 que es el que s'encarrega de regular una tensió d'entrada compresa entre 5.3V i 20V a una tensió regulada de 3.3V.

Tant a l'entrada com a la sortida d'aquest dispositiu podem observar dos condensadors. El condensador d'entrada es requerit per estabilitzar la tensió d'entrada al regulador, aquest condensador reduirà la sensibilitat del circuit i millorarà significativament la resposta transitòria de sortida del regulador. Aquest condensador es deu de muntar el més aprop possible del regulador i fent el camí més curt cap a terra. Segons el fabricant aconsella un condensador ceràmic o de tàntal de 10 uF.

En quant al condensador de sortida s'obliga al seu ús per tal d'assegurar l'estabilització de la sortida del regulador i la compensació de la freqüència. El fabricant aconsella una capacitat mínima de 4.7 uF podent ser ceràmic, de tàntal o electrolític.

A l'esquerra d'aquest es pot veure un connector al qual es connectaràn els cables d'alimentació, aquest connector es de orifici passant i amb rosca com mètode de pinçament del cable.

També es pot veure un fusible, la funció del qual no és més que la protecció del circuit en cas d'un pic de corrent, aquest fusible es auto-rearmable la qual cosa afavoreix a que no es fongui amb qualsevol pic, sinó que es desconnecti amb la pujada i una vegada s'hagi refredat aquest continuï deixant passant el corrent.

3.3.2 Microcontrolador i connector pickit 3

Com anteriorment en la figura 3.16 es pot veure de color groc aquest dos *subcircuits*, on com a títol es pot veure el connector pickit 3, el qual es el que ens facilita la connexió del dispositiu *Pickit 3* amb les corresponents potes del microcontrolador.

L'altre component encercat amb el mateix color groc es el microcontrolador explicat a l'apartat 3.2.1, el qual el podem descriure com el cervell d'aquest node, ell serà el que s'encarregarà de controlar els sensors i tractar tant les dades com l'enviament d'aquestes. Visualment es pot veure que com a perifèrics té un condensador, el qual s'encarregarà de regular la tensió d'entrada d'aquest component i un botó denominat com a *master-clear*, el qual no es més que per forçar un *reset* al microcontrolador, el qual faria començar des de el principi el sofware intern.

3.3.3 Mòdul Wifi

Encercat en color blau es pot veure el mòdul WI-FI explicat a l'apartat 3.2.2, el qual serà l'encarregat de connectar a una xarxa i enviar les dades.

Aquest mòdul només consta d'alimentació, dues connexions de transmissió i recepció per UART, una connexió per forçar el *reset* i una pota de control per saber si està correctament connectat a una xarxa

3.3.4 Receptor IR

Assenyalat en color lila es poden veure els sensors emissors i receptors, dels que en aquest apartat explicarem els receptors.

Aquests receptors del fabricant *Vishay* tal com s'ha explicat a l'apartat 3.2.3 son uns components de tres potes en una de les quals es pot observar un condensador el qual s'encarregarà de consumir l'arrissat que li pogués arribar a l'alimentació del receptor, les altres connexions seria *GND* i la sortida del sensor per la qual es detectarien les pertorbacions infraroges. Aquestes connexions anirien a un *socket* per tal de poden connectar els cables, ja que no aniran muntats damunt la mateixa placa.

3.3.5 Emissor IR

Els altres sensors cercats també en un color lila son els emissors de llum infraroja els quals conten amb un petit muntatge d'un inversor a partir de transistors MOSFET els quals s'han explicat a l'apartat 3.2.8.

Aquests emissors infrarojos explicats a l'apartat 3.2.4 es limiten a fer una llum infraroja en la longitud d'ona per la qual han estat dissenyats, la qual coincideix amb la que necessita el receptor de la qual manera es complementen.

3.4 Disseny de la PCB

Una vegada es té dissenyat el circuit en la finestra *schematic* del software esmentat anteriorment, es pot passar tot a un disseny damunt una PCB virtual amb les connexions que s'han fet en el circuit, simplement en aquest punt s'hauran de col·locar els components i crear les pistes.

A l'hora del disseny d'aquesta, s'han de tenir en compte una sèrie de coses, com per exemple:

- Les pistes han de ser el mes curtes i directes possibles, en particular la part analògica o digital d'alta velocitat, on la impedància i els efectes paràsits poden limitar el rendiment del sistema.
- Sempre que sigui possible es tendrà un pla de terra i un altre d'alimentació per tal que la placa actuï com a condensador i així poder fer front a les interferències electromagnètiques.
- Intentar no dibuixar angles tancats de 90º o més, ja que aixo varia l'amplada de la pista fent que canvii la impedància d'aquesta afectant a que l'energia no pugui fluir de la forma més efectiva.

- Evitar el creuament de senyals així com l'efecte *crosstalk* entre pistes properes.
- Tenir en compte l'amplada de la pista ja que aquesta ha de ser suficient per permetre el pas del corrent necessari.

Una vegada s'han tengut en compte aquests punts importants (maldament hi ha més conceptes a tenir en compte) s'ha passat al disseny de la PCB, la qual es pot veure a la figura 3.16.

En la respectiva figura es pot veure de color vermell fosc el que vendria a ser el pla superior de la placa, en blau la part inferior en la que es pot observar que quasi no hi ha pistes, sinó que es tracta de un pla complet de GND, en verd es poden veure totes les vies les quals traspassen la placa, però no necessàriament van connectades a la part inferior i en gris la informació dels components així com les mesures externes d'uns d'ells.

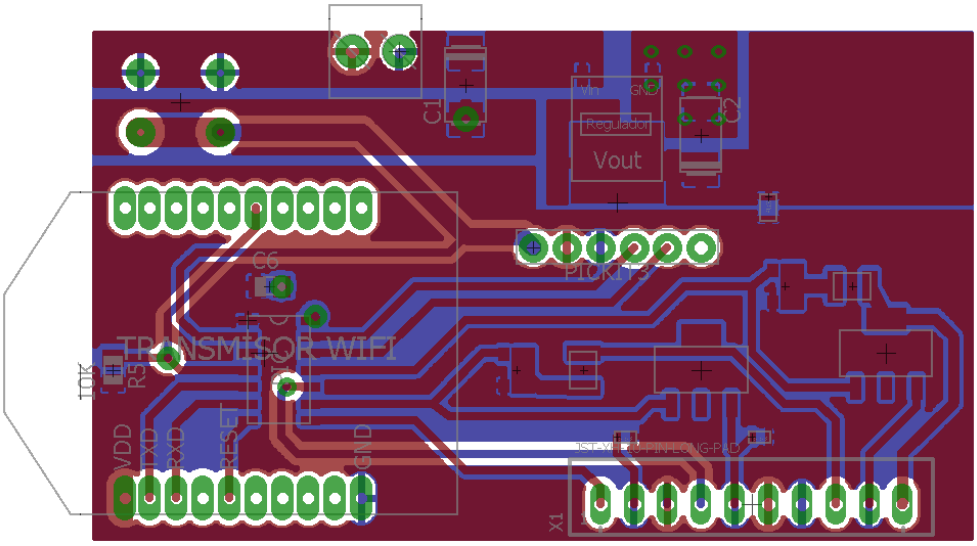


Figura 3.17: Disseny del PCB

CAPÍTOL 4

DISSENY DEL SOFWARE

4.1 Programació PIC16F1824

Per a la programació d'aquest microcontrolador s'ha empleat un software específic del mateix fabricant que el microprocessador utilitzat en aquest projecte i un hardware també del mateix fabricant el qual bolcarà el programa.

4.1.1 Hardware i software de programació

MPLAB X IDE

MPLAB[14] és un editor IDE gratuït, destinat a productes de la marca Microchip. Aquest editor és modular, permet seleccionar els diferents microcontroladors suportats, a més de permetre l'enregistrament d'aquests circuits integrats directament al programador. Es pot veure a la figura 4.1 la pàgina principal d'aquest IDE.

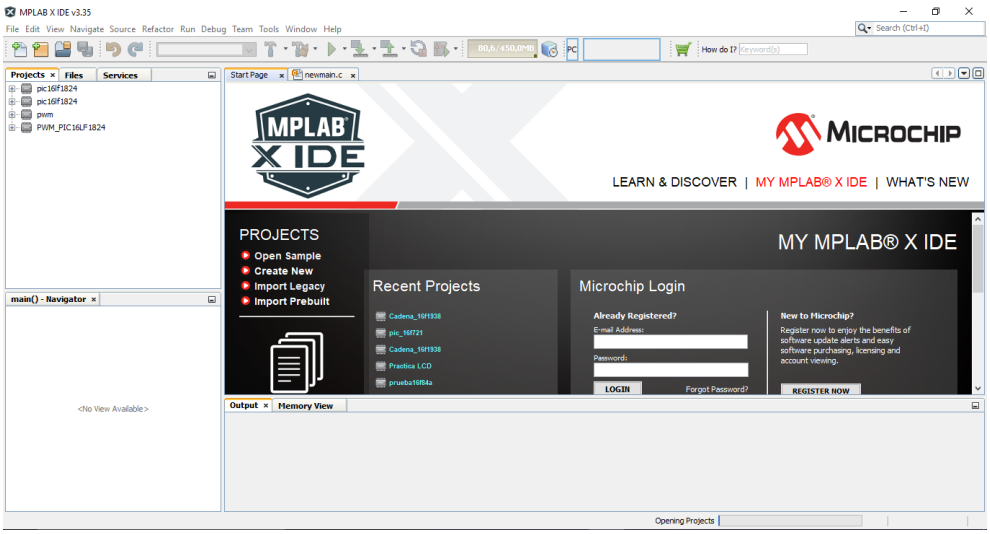


Figura 4.1: Pàgina principal MPLAB

És un programa que corre sota Windows, Mac OS i Linux. L'ambient MPLAB posseeix editor de text, compilador i simulació,no en temps real. Per començar un programa des de zero i gravar-ho al microcontrolador s'han de seguir les següents passes:

- Crear un Projecte nou triant un nom i ubicació.
- Triar el microcontrolador, el compilador.
- Crear el programa principal i començar a escriure el programa.
- Compilar-ho per crear l'arxiu .asm per poder gravar-ho en el PIC.

Aquest editor ens permet triar entr diverses opcions un compilador, en aquest cas s'ha triat el compilador XC8, el qual serà l'encarregat de compilar el codi i crear l'arxiu en *assembler* per poder gravar-ho al microcontrolador.

PICKIT3

Es un programador/depurador com el de la figura 4.2, és un depurador de circuiteria simple i de baix cost, el qual es controlat per un ordinador amb el software MPLAB IDE el qual s'empra en aquest projecte.

El programador PICKIT 3[15] es un sistema de depuració que s'utilitza per al desenvolupament de hardware i software amb els microcontroladors del fabricant *Microchip* basats en la programació en serie.

Figura 4.2: Programador Pickit 3

També té una senzilla connexió *USB Full Speed* amb l'ordinador que no només permet programar i depurar, sinó que també permet fer un *update* del firmware intern del PICKIT3.

Té circuits per protecció de sobre tensió i de curt circuit, permet execució en temps real i suporta tensions des de 2.0V.

Respectant la norma USB pot arribar a alimentar el circuit on està connectat fins a 100mA i té uns leds d'informació ràpida per a l'usuari.

A més d'aquestes funcionalitats, cal destacar una molt interessant i útil, la funcionalitat del *Programmer-To-Go*. Aquesta funcionalitat permet d'una forma molt senzilla programar qualsevol microcontrolador de les fàmilies PIC16, PIC18, PIC24, dsPIC33F i PIC32 del fabricant *Microchip* sense tenir que fer ús d'un ordinador, el qual es bastant útil si estam en un entorn on dur un ordinador es poc pràctic

El PICKIT3[15] permet guardar un còdi defins a 512KB en la seva Flash, això es fa des del software MPLAB i després basta alimentar el programador amb un carregador USB com per exemple els de mòbils i programar "*In Situ*"el nostre dispositiu.

Estructura del programa

A la figura 4.3 es pot veure un senzill diagrama de flux del programa el qual es troba al microprocessador.

En una breu versió explicativa es pot dir que el programa inicialment configura tots els paràmetres necessaris com pots, Timers, mòdul PWM i enviament de dades mitjançant la UART, per a continuació entrar en un bucle infinit, el qual detectarà el pas de les persones.

EL dispositiu es capaç de detectar el sentit de pas mitjançant la implementació de dos sensors iguals, que donen un '1' lògic al pas d'una persona, el sentit es sap per l'ordre en el qual els sensors envien el senyal lògic.

També hi ha una rutina la qual fa ús de la USART per enviar la trama al mòdul WI-FI el qual farà la petició de publicació a la plataforma Sentilo.

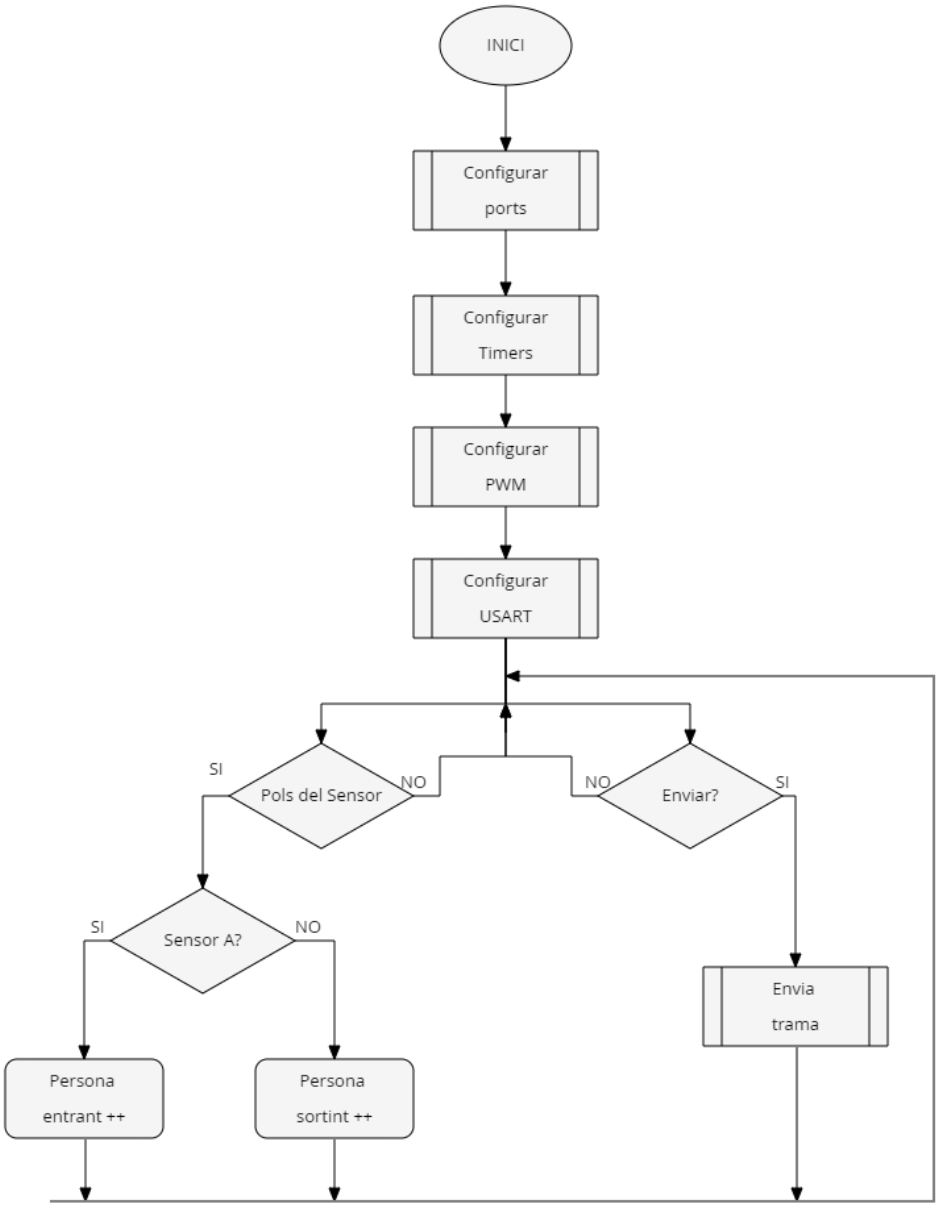


Figura 4.3: diagrama de flux del programa

Una vegada descrit el hardware, el software i l'estructura del programa per a la programació del microprocessador es passa a mostrar com es s'ha considerat fer la petició per pujar una dada al servidor Sentilo ja que es el més interessant d'aquest projecte, donat que tota l'altre part de la programació es resumeix en configuració del PIC i en condicionals per saber l'ordre de detecció de persones per saber el sentit del pas . A la figura 4.4 es pot veure aquesta petita part del codi.

```
sprintf(valor1,"%d", contador_1);
        sprintf(valor2,"%d", contador_2);

            //Enviament del comptador de persones que entren cap el campus
        envia_cadena_usart("PUT /data/UIB/prova/");
        envia_cadena_usart(valor1);
        envia_cadena_usart(" HTTP/1.1\r\n");
        envia_cadena_usart("IDENTITY_KEY: e10d8df07a6f8d49bf7d40d343db3ad"
            "601bb2d8aaec63e7d3741773e2240d672\r\n");
        envia_cadena_usart("\r\nConnection: close\r\n");
```

Figura 4.4: Codi del microprocessador per a la petició

Com es pot veure es una comunicació per la UART amb el mòdul WI-FI, on amb la funció *sprintf* el que fem es passar un valor a *String* per més tard enviar aquesta dada. Seguidament el que es fa és enviar *String's* que no son més que la identificació del sensor a la petició PUT i el identificador del *Proveïdor* majorment. El codi complet es pot veure l'Annex B.

4.2 Modul Rn-xv 171

Per poder configurar el mòdul WI-FI primer cal connectar-nos a ell. Això es fa seguint les següents passes:

Primer haurem de crear una xarxa sense fils en mode ADHOC amb el mòdul, ja que primerament només es possible connectar-se a ell mitjançant Telnet. Per crear aquesta xarxa ADHOC cal connectar físicament un cable amb una tensió de 3,3V al pin 8 del mòdul i desconnectar-ho després. Això haurà creat una xarxa anomenada *Wifly-EZX-131* a la qual ens haurem de connectar.

Una vegada connectat a aquesta xarxa obrim un emulador Telnet o la consola i fem un telnet a la direcció IP que ve per defecte per aquest dispositiu, la qual es *192.168.1.1* i al port *2000*, si tot a anat be rebrem: **HELLO**

Ara que ja estem connectats al modul necessitem entrar al mode configuració, això ho farem introduint *$$$* amb el qual rebrem *CMD* el que voldrà dir que ja estem al mode comandes on podem començar amb la configuració.

Per aquest projecte ha estat necessari introduir els següents paràmetres:

```
set ip host 172.22.39.171:8081
```

Amb el que configurem la direcció IP del *host* remot amb el que volen tenir una comunicació. En aquest cas es l'ordinador on hi ha instal·lat el servidor Sentilo.

```
set wlan auth 4
```

Amb el que configurem el mode d'autenticació, que utilitzam WPA2. Aquest es una especificació del CTI.

```
set wlan phrase <...>
```

Amb aquesta configuració introduïm el *password* de la xarxa a la qual ens volem connectar.

```
set wlan ssid <uib@uib>
```

Configura el nom de la xarxa a la que volem connectar el mòdul wifi.

```
set ip dhcp 1
set ftp mode 0x0
```

i feim que el mòdul reenvii les dades que li passam per la UART:

```
set uart mode 2
```

Amb els que acabam la configuració de la xarxa. Una vegada configurat cal guardar els canvis i reiniciar el dispositiu amb les següents comandes:

```
save
reboot
```

Una vegada reiniciat el led verd parpellejarà lentament en el cas de tenir una IP assignada la qual pot ser donada per el mòdem de la xarxa a la qual es connecta.

Ara ja tenim el mòdul configurat, connectat a la xarxa i preparat per enviar les dades que li passem per la UART.

Es pot veure la configuració completa del dispositiu introduint *get everything* en el mode comandes tal com es pot veure a l'Annex

CAPÍTOL 5

PROTOTIP

5.1 Descripció

El projecte descrit anteriorment es un prototip del qual es faran proves per tal provar l'error que pot arribar a tenir, per exemple en el cas de passar dues persones en paral·lel lo que faria que una persona tapàs a l'altre i no es comptàs aquesta, o si dues persones entren al rang de detecció al mateix moment però en sentits oposats.

Aquest prototip es muntarà en uns pals de fusta de geometria circular de 8cm de diàmetre. Un del pals tendrà la placa amb els sensor, la bateria i la placa solar, mentre que l'altre tendrà els dos retroreflectors, els quals han quedat explicats a l'apartat 3.2.9.

Aquests pals es disposaran en un àmbit real, concretament a l'entrada exterior del metro d'aquesta entitat la Universitat de les Illes Balears, a la següent imatge es por veure on s'instal·lara el pal amb els sensors.

Figura 5.1: Universitat de les Illes Balears

5.2 Muntatge

A la següent figura es pot veure la idea inicial del muntatge dels sensors al pal:

Figura 5.2: Disseny dels pals

Per tal d'integrar la placa al pal s'ha dissenyat una caixa amb un programa en 3D, per què un disseny d'una caixa standard? Perquè es volia integrar el més possible el dispositiu al pal, i donat que era de geometria circular es va dissenyar una caixa amb la mateixa circumferència del pal per tal de passar el mes desapercebut possible.

La caixa te unes incisions per tal de poder atorrollar uns perns i unes mosques per inserir una junta de goma i d'aquesta manera sigui estanca per poder fer front als efectes ambientals ja que va exposada a l'exterior.

El programa utilitzat es 123D Design del fabricant *Autodesk*, un programa senzill i gratuït per al disseny de peces senzilles, el disseny de la caixa es pot veure a continuació:

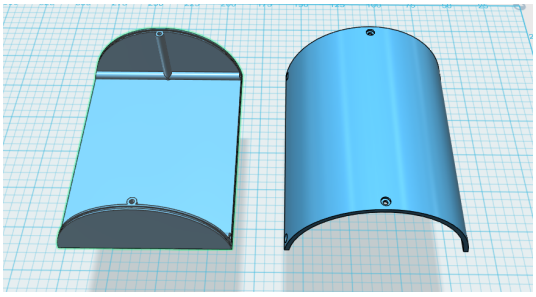


Figura 5.3: Disseny de la caixa

Una vegada dissenyada la caixa l'hem de portar a la realitat, s'ha optat de crear-la amb una impressora 3D, la qual utilitza un sistema de dipositació de plàstic el qual va fonent i va formant la peça capa a capa la qual es pot veure a la figura 5.4. Es poden elegir diferents tipus de plàstics, en aquest cas s'ha fabricat en plàstic PLA. A l'Annex D.

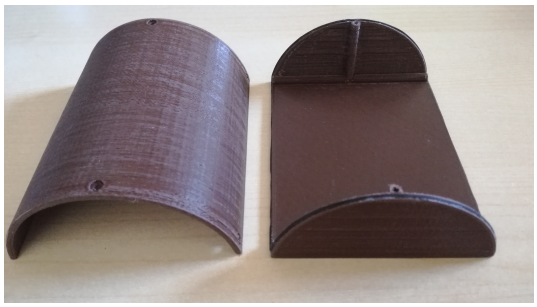


Figura 5.4: Caixa fabricada

Ara que ja es té dissenyada la caixa que portarà la placa on es trobarà el cervell del dispositiu serà necessari dissenyar i fabricar el suport on es col·locarà la placa solar amb la bateria, el qual també s'ha dissenyat amb el mateix software. A aquest suport s'ha creat un espai on s'introduirà la bateria al qual també s'ha dotat d'estanquitat ama una junta de goma i unes incisions per a atorrollar uns perns com a la caixa anterior. A continuació es pot veure el disseny i el producte final:

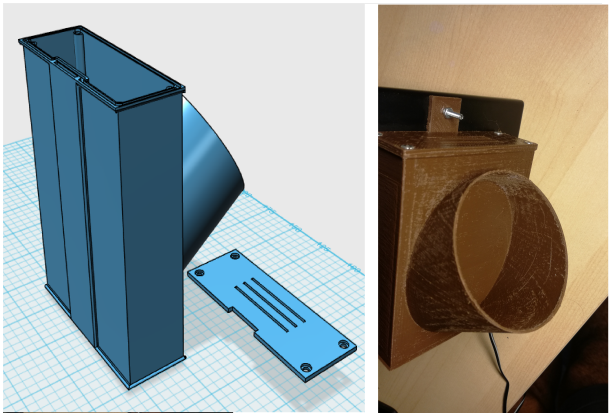


Figura 5.5: Suport placa solar i bateria

Darrerament només faltarà el disseny i la fabricació del lloc on aniran els sensors. Igualment que als apartats anteriors s'ha dissenyat amb l'ordinador. En aquest cas s'ha dissenyat una peça cilíndrica que portarà els sensors, però sense que l'emissor i el receptor del mateix costat no es vegin. A més s'ha inclòs una petita endidada on s'introduirà un petit vidre de metacrilat el qual farà que no passi la humitat ni la pols, dotant també aquesta petita part d'estanquitat. A continuació es pot veure gràficament com ha quedat aquest disseny:

Figura 5.6: Suport dels sensors

Per finalitzar el muntatge d'aquest producte només faltarà incorporar els retrorreflectors a l'altre pal. El retrorreflectors son quadrats de 5 cm de costat, mentre el pal es rodó i té un diàmetre de 8 cm, pel que s'ha procedit fer una incisió al pal, de la mateixa manera que per la caixa, i a crearli un petit suport per protegir millor els retrorreflectors que sobresortiran un poc. A continuació es por veure el suport dissenyat i fabricat:

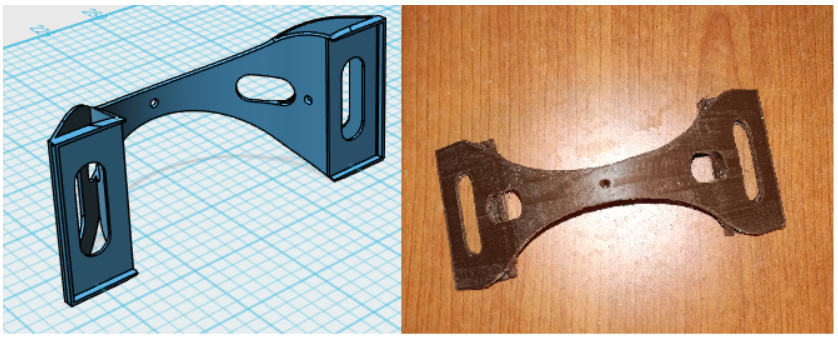


Figura 5.7: Suport del retrorreflectors

Una vegada fet i muntat totes les parts que componen els pals es té els sensors infrarojos muntats amb tots els seus perifèrics, placa solar, bateria reflectors... tal com es mostra a la figura 5.8

Figura 5.8: Muntatge complet del pals

CAPÍTOL 6

CONCLUSIONS

En aquest projecte s'ha dissenyat i fabricat un prototip el qual es troba emmarcat en les tecnologies actuals de la informació i difusió. En aquest node s'han duit a terme els objectius que es van proposar a l'inici d'aquest projecte com es el disseny d'aquest juntament amb la monitorització a la plataforma Sentilo, ítem del qual es el referent de les noves tecnologies a la ciutat de Barcelona.

Els pals s'han fabricat realitzant una feina manual casera i pensant en totes les possibles accions de l'intemperat al estar aquest a l'exterior, com pot esser les condicions de pluja que poden fixar problemes d'humitat i corrosió fabricant les caixes on es troben les plaques totalment estanques i fixant uns vidres de metacrilat transparent als sensors òptics.

Una vegada que ja es tenia la plataforma Sentilo instal·lada i tot el hardware muntat s'ha procedit a fer les proves de detecció dins un àmbit real, els pals muntats al seu lloc, els quals han conclòs una bona detecció als diferents obstacles que infereixen sobre la barrera òptica i la posterior publicació de les deteccions sobre la plataforma.

Després de concloure aquest prototip cal mencionar que la feina realitzada i descrita anteriorment es totalment funcional, tant la placa dissenyada i muntada com el servidor funcionen correctament, per tant la motivació al realitzar aquest projecte ha estat desmesuradament satisfactòria.

6.1 Futur desenvolupament

Com s'ha esmentat anteriorment això es un prototip el qual ha quedat totalment funcional i muntat sobre la seva estructura, amb la qual s'ha comprovat el correcte funcionament de la plataforma Sentilo i tots els components que conformen la placa. Es pot dir que es detecta completament el pas de persones, on les proves fetes ens

validen que la distància de cobertura dels sensors d'infrarojos es troba al voltant d'una vintena de metres.

D'aquest prototip es faran proves, es corregiran possibles problemes i errors que sempre compaginen de la mà en la vida real per tal de convertir-ho a un producte final totalment comprovat i amb uns possibles errors documentats i tractats com poden ser:

- Estimació dels marges d'errors en funció de la lluminositat de l'entorn.
- Estimació dels marges d'error en la compta de grups de persones.
- Estimació en velocitat de pas.
- Valorar el consum efectiu del transmissor per allargar-lo al màxim.

APÈNDIX A

SETUP SENTILO

A.1 Prerequisits

Abans de començar amb la instal·lació de la plataforma Sentilo, es necessari que instal·lem o verifiquem que tenim disponibles una sèrie de requisits:

- JDK 1.6.x ò superior.
- Git, el qual usarem per descarregar el codi des de Github.
- Maven 2.2.1 ò superior.

A.1.1 JDK

Primer instal·larem Java Runtime Environment(JRE) i Java Development Kit (JDK) del repositori, el qual ja haurem actualitzat, amb les següents instruccions:

```
$ sudo apt-get install default-jre
$ sudo apt-get install default-jdk
```

Podem veure la versió de java instal·lada en el nostre sistema de la següent manera:

```
$ java -version
```

A.1.2 GIT

Una vegada instal·lat JDK instal·larem GIT[16] dels repositoris del nostre sistema. Aquesta es la forma més ràpida, però la versió que s'instal·larà pot ser no sigui la versió més recent. Si es volgués la versió més recent s'haurà de compilar git del seu origen, però

en aquest cas no es necessària la més recent, per la qual cosa s'instal·larà i es sabrà la versió amb les següents comandes:

```
$ sudo apt-get install git
$ git - -version
```

A.1.3 Maven

Molt semblant a la instal·lació dels requisits anteriors introduirem les següents comandes[17]:

```
$ sudo apt-get install maven
```

I per verificar la versió instal·lada executarem:

```
$ mvn -version
```

A.2 Descàrrega i compilació del codi

El codi Sentilo[4] s'ha de descarregar des de Github, el qual una vegada descarregat es podrà compilar fent ús d'un script el qual construirà els artefactes de Sentilo.

Ja que es té descarregat el software 'Git' podem descarregar el codi amb la següent comanda, la qual clonarà el projecte en un directori local que s'ha anomenat ,obviament, Sentilo:

```
$ git clone https://github.com/sentilo/sentilo.git Sentilo
```

Una vegada descarregat el compilarem sense canviar els valors per defecte executant un script anomenat buildSentilo.sh, el qual es troba a la carpeta scripts:

```
$ sudo sh Sentilo/scripts/buildSentilo.sh
```

Aquest script compila el codi i construeix els artefactes de Sentilo des de zero, però no els desplega en els entorns d'execució, la qual cosa s'ha de fer manualment per diverses raons, per exemple:

- No tots els agents son obligatoris
- Podem voler un agent històric, relacional, ...
- Canviar els paswords o usuaris per defecte

A.3 Infraestructura de la plataforma

En el present apartat es descriurà la instal·lació i la configuració que han de tenir per defecte de cada un dels elements de la infraestructura.

Els elements que componen la infraestructura Sentilo es poden dividir en dues categories:

- Obligatoris
  - Redis: motor de base de dades.
  - MongoDB: Enorme sistema de base de dades en open-source
  - Tomcat: Funciona com un contenidor de servlets[1]
- Opcionals
  - MySQL: Necessari si el vol instal·lar l'agent relacional.
  - ElasticSearch: Necessari si es vol instal·lar l'agent activity-monitor.
  - OpenTSDB: Necessari si es vol instal·lar l'agent històric.
    Prerequisits:
    - Java Development Kit 1.6 ò posterior. Veure instal·lació a l'apartat A.1.1.
    - Zookeeper
    - Hbase 0.92 ò posterior
      - Hadoop
    - GnuPlot 4.2 ò posterior
    - Autotools
    - Python
    - Git. Veure instal·lació a l'apartat A.1.2.

Per tal de poder descarregar i configurar tots aquests elements es farà ús de les eines que a continuació es mostra com poder obtenir-les:

Wget

Wget es una eina lliure que mitjançant la consola permet la descàrrega de continguts des de servidors web d'una forma molt simple.

Entre les característiques mes destacades que ofereix Wget està la possibilitat de fàcil descàrrega de mirrors (miralls) complexes de forma recursiva.

Es pot instal·lar amb la següent comanda:

[1] Classe en el llenguatge de programació Java, utilitzada per ampliar les capacitats d'un servidor.

```
$ sudo apt-get install wget
```

Gedit

Gedit es un editor de textos compatible per GNU/Linux, Mac OS X y Microsoft Windows. Dissenyat com un editor de textos de propòsit general, Gedit emfatitza la simplicitat i facilitat d'us. Inclou eines per l'edició del codi font i textos estructurats.

Es pot instal·lar i descarregar de manera similar a l'anterior:

```
$ sudo apt-get install gedit
```

Bash

Bash es un programa informàtic, la funció del qual consisteix en interpretar ordres. Es l'interpret de comandos per defecte en la majoria de les distribucions linux.

De la mateixa manera que els casos anteriors el podem instal·lar:

```
$ sudo apt-get install bash
```

A.3.1 Redis

Es un motor de base de dades en memòria, basat en el emmagatzematge en taules de hashes (clave/valor) però que opcionalment pot ser usada com una base de dades durable o persistent. Està alliberat sota llicència BSD pel que és considerat programari de codi obert.

Es donarà peu a la instal·lació[18] de `Redis` des de els repositoris de Ubuntu, pel que no es necessari seguir una gran quantitat de de passes, ja que es bastant sencill i ràpid perquè els paquests entreguen un entorn preconfigurat de propòsit general que cobreix en general les necessitats de qualsevol projecte. Per instal·lar la darrera versió de Redis des de els repositoris només serà necessari fer una actualització dels paquets i instal·lar aquest servei introduint les següents comandes:

```
$ sudo apt update
$ sudo apt install redis-server
```

Després d'això es pot configurar per a que s'inicii el servici al arrencar el sistema.

```
$ sudo systemctl enable redis-server
```

Una vegada fet això es pot passar a configurar Redis per Ubuntu 16.04 (com es aquest cas).

- Configuració

  Per la configuració de Redis es necessari copiar l'a, però per això, primer crearem un directori com es mostra a continuació:

```
$ sudo mkdir /etc/redis
```

  Ara copiarem l'arxiu mencionat anteriorment al directori que s'ha creat mitjançant la següent comanda:

```
$ sudo cp /tmp/redis/redis.conf /etc/redis
```

  Una vegada copiat aquest arxiu serà necessari modificar-ho. La modificació tracta on s'emmagatzemaran els arxius de la base de dades, això no ha de ser accessible per altres usuaris o des de internet. Per això s'ha seleccionat el directori /var/lib/redis. Seguidament podem veure com modificar aquest arxiu i el que s'ha de modificar:

```
$ sudo gedit /etc/redis/redis.conf
```

```
# The working directory.
#
# The DB will be written inside this directory, with the filename specified
# above using the 'dbfilename' configuration directive.
#
# The Append Only File will also be created inside this directory.
#
# Note that you must specify a directory here, not a file name.
dir /var/lib/redis
```

  Una vegada fets els canvis assegurar-se de guardar-los.

Ara ja es té Redis instal·lat i configurat. Podem provar que el servei de Redis està funcionant amb la següent comanda:

```
$ redis-cli
```

Amb la qual s'obrirà una interfície de text Redis que es podrà identificar per un 'promt' que es semblant a: "127.0.0.1:6379>". On per veure que respon correctament es pot enviar un PING. Aquí es pot veure això i la seva resposta:

```
$ 127.0.0.1:6379>PING
PONG
```

Si retorna el 'PONG' anterior es que Redis està funcionant al teu servidor i tot ha sortit correctament.

A.3.2 MongoDB

Es un sistema de base de dades orientat a documents, desenvolupat baix el concepte de codi obert. Forma part d la nova família de sistemes de bases de dades NoSQL [2]. En lloc de guardar les dades en taules tal com es fa en les bases de dades relacionals, MongoDB guarda estructures de dades en documents similars a JSON [3] amb un esquema dinàmic fent que la integració de les dades en certes aplicacions sigui més fàcil i ràpida.

Per a la instal·lació[19] d'aquesta base de dades podem accedir als repositoris del sistema amb la següent comanda:

```
$ sudo apt-get install mongodb-org
```

Per a instal·lar una versió en concret es pot procedir de la següent manera, indicant la versió que es vol instal·lar, en el nostre cas ha estat la versió 2.6.10:

```
$ sudo apt-get install mongodb-org=2.6.10 mongodb-org-server=2.6.10 mongodb-org-shell=2.6.10 mongodb-org-mongos=2.6.10 mongodb-org-tools=2.6.10
```

Una vegada instal·lat podem verificar la correcta instal·lació introduint la següent petició que en indicarà la versió instal·lada:

```
$ mongo –version
```

Si tot ha estat correcta s'obtindra una resposta a la consola com la que es pot veure a la següent figura:

Figura A.1: Versió MongoDB

[2] Es una àmplia classe de sistemes de gestió de dades que difereixen del model clàssic de SGBDR. Les dades emmagatzemades no requereixen d'estructures fixes com taules, i habitualment escalen molt be horitzontalment

[3] Format de text lleuger per l'intercanvi de dades.

A.3.3 Tomcat

Tomcat funciona com un contenidor de servlets, és a dir, és un mòdul per executar servlets i/o pàgines JSP en les aplicacions Web. Tomcat pot funcionar com a servidor web per si mateix. L'avantatge més gran es que es tracta de codi obert.

Com anteriorment es donarà peu a la instal·lació de `Tomcat`[20] des de els repositoris de Ubuntu, pel que es farà de manera molt senzilla. Només introduirem la següent comanda que donarà començament a la instal·lació:

```
$ sudo apt-get install tomcat7
```

Una vegada introduïda aquesta petició començarà la instal·lació a la que segurament haurem de respondre amb una 'Y' per poder dur-la a terme. Això instal·larà Tomcat i les seves dependències com Java i també crearà l'usuari 'tomcat7'.

Una vegada acabada aquesta instal·lació s'haurà de fer un canvi a les opcions de Java que utilitza Tomcat quan s'inicia. Per això s'haurà d'editar l'arxiu Tomcat7 com es mostra a continuació:

```
$ sudo gedit /etc/default/tomcat7
```

Una vegada dins l'editor cercar la linea que posa '`JAVA_OPTS=`' i introduir el següent. Aquests paràmetres afecten a la quantitat de memòria que Tomcat farà servir:

```
JAVA_OPTS=-Djava.security.egd=file:/dev/./urandom -Djava.awt.headless=true -
Xmx512m -XX:MaxPermSize=256m -XX:+UseConcMarkSweepGC"
```

Després de fer aquesta edició guardarem els canvis i sortirem de l'editor. Ara serà necessari reiniciar el servidor, la qual cosa es farà mitjançant la següent comanda:

```
$ sudo service tomcat7 restart
```

Tomcat no s'ha establert completament encara, però es pot accedir a la pàgina de benvinguda per defecte anant al seu domini o l'adreça IP seguida d'un :8080 al navegador web. Veurà una pàgina de benvinguda que diu Ït works!", a més d'una altra informació.Ara s'aprofundirà més en la instal·lació de Tomcat.

Ara serà necessari introduir dades d'accés al nostre servidor Tomcat, això es farà mitjançant l'edició de l'arxiu 'tomcat-users.xml':

```
$ sudo gedit /etc/tomcat7/tomcat-users.xml
```

Aquest arxiu està ple de comentaris que descriuen com configurar-ho, en aquest cas volem introduir un usuari que pugui accedir al 'manager-gui' i al 'admin-gui', això es pot fer localitzant les etiquetes "tomcat-usersï introduint les definicions d'usuaris. Tot això es pot fer com es mostra:

```
<tomcat-users>
  <role rolename="manager-gui»</role>
  <user username="sentilo"password="sentilo"roles="manager-gui»</user>
</tomcat-users>
```

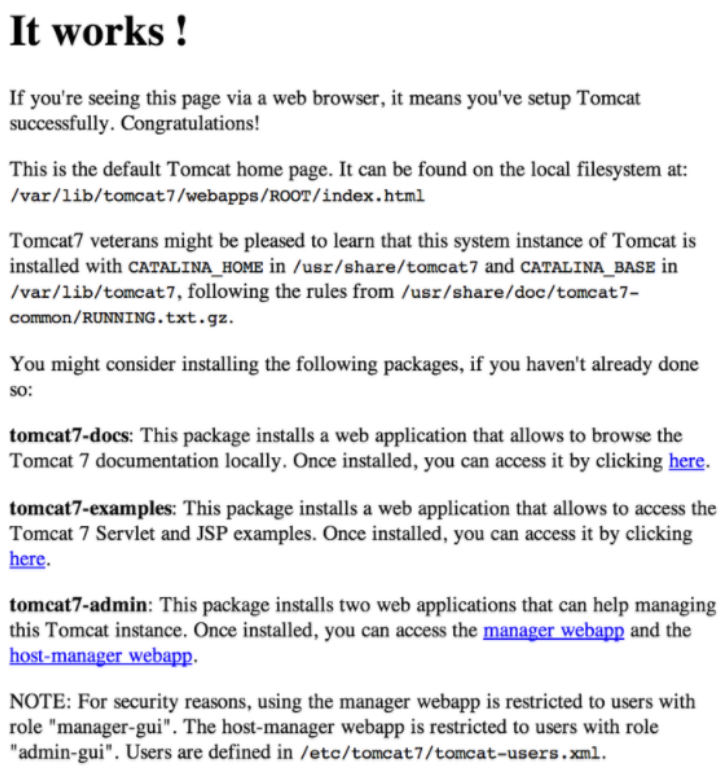
Una vegada fets els canvis sortir i guardar els canvis. Per posar en pràctica aquests canvis s'haura de reiniciar el servei de Tomcat:

```
$ sudo service tomcat7 restart
```

Ara que hem configurat un usuari administrador, tendrem accés a la interfície d'administració web al navegador. Al nostre 'web browser' introduirem el següent:

```
http://server_IP_address:8080
```

On `server_IP_address` serà la nostra direcció IP. Una vegada introduït això al nostre navegador si tot ha anat be obtindrem alguna cosa semblant a la següent imatge:



It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat7/webapps/ROOT/index.html`

Tomcat7 veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat7` and `CATALINA_BASE` in `/var/lib/tomcat7`, following the rules from `/usr/share/doc/tomcat7-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

tomcat7-docs: This package installs a web application that allows to browse the Tomcat 7 documentation locally. Once installed, you can access it by clicking here.

tomcat7-examples: This package installs a web application that allows to access the Tomcat 7 Servlet and JSP examples. Once installed, you can access it by clicking here.

tomcat7-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the manager webapp and the host-manager webapp.

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined in `/etc/tomcat7/tomcat-users.xml`.

Figura A.2: Tomcat7

A.3.4 ElasticSearch

Es un servidor de cerca. Proveeix un motor de cerca de text complet, distribuït i amb una interfície web. Està desenvolupat en Java i està publicat com a codi obert baix les condicions de la llicència Apache.

La instal·lació[21] es put dur a terme mitjançant l'eina *wget*, descarregant l'arxiu desde la pròpia pàgina del proveïdor amb la següent comanda:

```
$ wget https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-1.7.2.deb
```

El qual una vegada descarregat el podem instal·lar fent ús de la comanda *dpkg* tal com es mostra a continuació:

```
$ sudo dpkg -i elasticsearch-1.7.2.deb
```

De moment per defecte, *ElasticSearch* s'hauría d'estar executant al port 9200. Això es pot comprovar amb la comanda *CURL*, fent una simple sol·licitud GET com aquesta:

```
$ curl -X GET 'http://localhost:9200'
```

A.3.5 MySQL

Es un sistema de gestió de bases de dades relacional desenvolupat baix la llicencia dual GPL/Llicencia comercial per *Oracle Corporation* i està considerada com la base de dades *open source* més popular del món, i una de les més populars en general juntament a *Oracle* i *Microsoft SQL Server*, sobretot pel desenvolupament web.

La instal·lació[22] d'aquest paquet es molt senzill, basta en accedir als repositoris del sistema com s'ha fet altres vegades, sempre després d'haver fet un *update* en qualque moment anterior:

```
$ sudo apt-get install mysql-server
```

Una vegada instal·lat, MySQL s'haurà començat a executar-se automàticament. Es pot provar si el seu funcionament es correcte executant el següent script:

```
$ systemctl status mysql.service
```

A.3.6 OpenTSDB

Es una Base de dades se Series Temporals distribuïda i escalable sobre HBase. Es un producte Open-Source i està pensada per emmagatzemar i servir mètriques recopilades d'aplicacions, sistemes operatius, bases de dades,... a gran escala i pensant en fer aquestes dades accessibles i gràfiques.

Per a la instal·lació[**?**] d'aquesta base de dades cal descarregar i configurar els següents prerequisits:

Zookeeper

Per començar descarregarem l'arxiu de Zookeeper[23], això es pot fer entrant a internet a la pàgina d'apache al següent link `http://zookeeper.apache.org/releases.html`, i descarregam la darrera versió estable, o mitjançant la consola linux fent ús de l'eina 'wget' (Instruccions apartat A.3) introduint la següent comanda:

```
$ wget http://apache.rediris.es/zookeeper/zookeeper-3.4.9/zookeeper-3.4.9.tar.gz
```

Una vegada descarregat l'arxiu el podem descomprimir posicionant-nos al directori de descàrrega i introduint el següent script:

```
$ tar -zxf zookeeper-3.4.9.tar.gz
```

...el qual ens crearà la carpeta descomprimida de zookeeper. Una vegada descomprimida podem moure la carpeta a altre directori de la següent forma:

```
$ sudo mv zookeeper-3.4.9 /var/
```

(Si es volés moure al directori /var/)
Una vegada mogut a la carpeta desitjada anam a aquest directori i ens posicionam dins la carpeta i cream una altre carpeta anomenada 'data', per exemple. Tot això es pot fer de la següent forma:

```
$ cd /var/zookeeper-3.4.9
$ sudo mkdir data
```

Una vegada fet això es crea un arxiu de configuració dins la carpeta 'conf' amb l'enida Gedit (Instruccions apartat A.3) amb la següent comanda:

```
$ sudo gedit zoo.cnf
```

Al qual introduïm els següents paràmetres:

```
tickTime = 2000
dataDir = /var/zookeeper-3.4.9/data
clientPort=2181
```

Una vegada realitzat totes les passes anteriors es pot provar que Zookeeper ha quedat ben instal·lat i configurat amb l'execució del següent script fent ús de l'eina bash (Instruccions apartat A.3),tot això sempre des de dins d la carpeta zookeeper-3.4.9:

```
$ sudo bash bin/zkServer.sh start
```

Amb el que, una vegada executat, obtindrem una resposta similar a la Fig. A.3 si tot s'ha produït satisfactòriament.

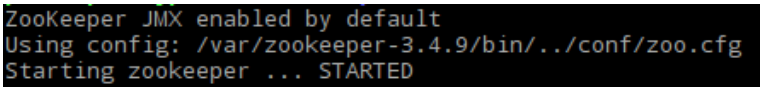


Figura A.3: Zookeeper

Una vegada instal·lat, configurat i provat ens hem d'assegurar que es accessible, això ho farem de la següent forma:

```
$ telnet localhost 2181
```

On obtendrem la següent resposta i una vegada dins executarem la comanda 'stats' tal com mostra la Fig. A.4.

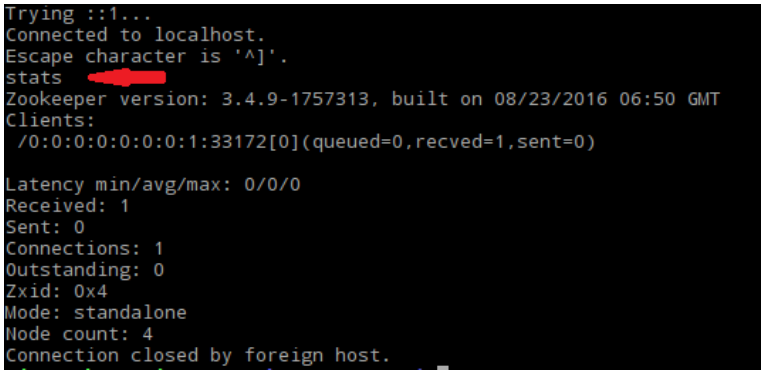


Figura A.4: Zookeeper

En el cas de no poder-se connectar a Zookeeper, comprovar direccions IP ò ports. També tenir en compte que si HBase s'estàs executant podria donar problemes.

Una vegada connectat el servidor, es pot aturar executant el següent script(sempre dins el directori de zookeeper):

```
$ sudo bash bin/zkServer.sh stop
```

HBase

Per poder instal·lar i configurar correctament HBase[24] necessitarem tenir instal·lada una versió estable de Hadoop.

- Hadoop

  Per instal·lar Hadoop haurem de instal·lar el següent software a més de tenir Java instal·lat (veure apartat A.1.1):.

```
$ sudo apt-get install ssh
$ sudo apt-get install rsync
```

  Una vegada instal·lat els requisits anteriors hem de descarregar una versió estable de Hadoop, la qual la podem descarregar de la pàgina web d'apache al següent link `http://apache.org/dyn/closer.cgi/hadoop/common`, i descarregam la darrera versió estable, o mitjançant la consola introduint la següent comanda:

```
$ wget http://apache.rediris.es/hadoop/common/hadoop-2.7.3/
hadoop-2.7.3.tar.gz
```

Una vegada s'ha descarregat l'arxiu ens posicionam a la carpeta de descàrrega per poder descomprimir-ho, el que es pot fer introduint el següent:

```
$ tar -zxf hadoop-2.7.3.tar.gz
```

Amb el que es crearà la carpeta descomprimida de Hadoop.

Ara que es té la carpeta descomprimida s'ha d'editar un arxiu existent dins la carpeta mencionada el qual es nom hadoop-env.sh. Dins aquest arxiu s'ha d'editar el directori root on es troba la instal·lació JAVA, tal com es mostra a continuació:

```
export JAVA_HOME = /usr/
```

Una vegada instal·lada i configurada una versió estable de Hadoop ja es pot instal·lar HBase. Es pot trobar la darrera versió estable de HBase a la pàgina http://apache.org/dyn/closer.cgi/hbase on a l'inici de la pàgina es veurà un link suggerit per la descàrrega. Una vegada dins aquest link entrarem a la carpeta de nom 'stable' on es troba la darrera versió per descarregar i es descarregarà un arxiu binari acabat en `.tar.gz`. A continuació es pot veure un script per poder descarregar l'arxiu des de la consola Linux:

```
$ wget http://apache.rediris.es/hbase/stable/hbase-1.2.4-bin.tar.gz
```

Una vegada acabada la descàrrega es pot descomprimir l'arxiu des de el directori de descàrrega introduint la següent comanda:

```
$ tar -zxf hbase-1.2.4-bin.tar.gz
```

el qual crearà la carpeta descomprimida de HBase. Una vegada creada aquesta carpeta es necessari fer una sèrie de modificacions.

- Configuració HBase.

  Una configuració necessària que s'ha de fer es establir la variable d'entorn `JAVA_HOME`. Es pot establir aquesta variable amb el mecanisme habitual, però HBase proporciona un mecanisme central mitjançant la modificació de l'arxiu hbase-env.sh.

La variable `JAVA_HOME` deu ajustar-se a un directori que conté l'arxiu executable `bin/java`. La majoria dels sistemes operatius moderns Linux proporcionen un mecanisme, tal com `/usr/bin` per la commutació entre versions executables com JAVA. En aquest cas l'enllaç simbòlic es `/usr/`.

Per establir això s'ha de posicionar dins el directori on es localitzi l'arxiu hbase-env.sh, en aquest cas:

```
$ cd  /hbase-1.2.4/conf
```

i una vegada dins aquest directori utilitzant l'editor de text 'Gedit'(Instruccions apartat A.3) podem incloure l'enllaç de la variable d'entorn com es mostra a continuació:

```
export JAVA_HOME = /usr/
```

Una vegada editada aquesta variable s'haurà d'editar el document de configuració de HBase que es hbase-site.xml. Aquí haurem d'especificar el directori en el sistema d'archius local on HBase i Zookeeper escriuen les dades, ja que per defecte es crea un directori `/tmp` , però molts de servidors estan configurats per eliminar el contingut d'aquest directori, pel que s'haurà d'emmagatzemar les dades a un altre lloc.

Per editar aquest document podem fer el mateix que s'ha fet a l'anterior pas o de manera més ràpida amb la següent comanda:

```
$ sudo gedit  /hbase-1.2.4/conf/hbase-site.xml
```

i introduïm el següent dins aquest document:

```
<configuration>
   <property>
      <name>hbase.rootdir</name>
      <value>file:///home/eps/hbase-1.2.4</value>
   </property>
   <property>
      <name>hbase.zookeeperproperty.dataDir</name>
      <value>/home/eps/zookeeper</value>
   </property>
</configuration>
```

Una vegada instal·lats i configurats els softwares anterior podem passar a la instal·lació i configuració de OpenTSDB. Per obtenir l'arxiu al nostre sistema local podem fer ús de la utilitat *git clone* tal com es mostra a continuació:

```
$ git clone git://github.com/OpenTSDB/opentsdb.git
```

Una vegada fet això ens posicionem al directori clonat i es comença el proces d'instal·lació mitjançant els següents scripts:

```
$ cd opentsdb
$ sudo ./build.sh
```

Ara ens movem al directori *build* el qual queda localitzat dins la carpeta *"opentsdb"* i completam la instal·lació introduint el següent:

```
$ cd /opt/opentsdb/build
$ sudo make install
```

Una vegada seguides aquestes passes ja ha quedat instal·lat OpenTSDB.

A.4 Configuracions

Una vegada tenim tot el software necessari instal·lat passem a configurar-lo. A continuació es mostraran els valors de configuració dels diferents softwares:

A.4.1 Redis

Els paràmetres que Sentilo espera trobar son els següents:

```
jedis.pool.host = 127.0.0.1
jedis.pool.port = 6379
jedis.pool.password = sentilo
```

Figura A.5: Paràmetres per defecte Redis

Els paràmetres *port* i *host* que venen per defecte a Redis coincideixen amb els que cerca tenir Sentilo, per tant això no farà falta modificar-ho.

Per altre banda si que farà falta incloure una breu modificació, ja que Sentilo vol que aquest utilitzi una contrasenya, la qual s'ha d'activar al fitxer *redis.conf*. A continuació es mostrarà un breu fragment del que ve dins aquest arxiu per veure un es troba aquesta configuració i la modificació d'un paràmetre, concretament *requirepass sentilo*:

```
################################## SECURITY ###################################

# Require clients to issue AUTH <PASSWORD> before processing any other
# commands.  This might be useful in environments in which you do not trust
# others with access to the host running redis-server.
#
# This should stay commented out for backward compatibility and because most
# people do not need auth (e.g. they run their own servers).
#
# Warning: since Redis is pretty fast an outside user can try up to
# 150k passwords per second against a good box. This means that you should
# use a very strong password otherwise it will be very easy to break.
#
 requirepass sentilo
```

Figura A.6: Configuració Redis

En el cas de voler canviar qualsevol dels paràmetres dins el software de Sentilo que fan referència a Redis es poden trobar a les següents carpetes:

```
sentilo-platform/sentilo-platform-service/src/main/resources/properties/jedis-config.properties
sentilo-agent-alert/src/main/resources/properties/jedis-config.properties
sentilo-agent-relational/src/main/resources/properties/jedis-config.properties
sentilo-agent-location-updater/src/main/resources/properties/jedis-config.properties
```

Figura A.7: Configuració de Redis a Sentilo

A.4.2 MongoDB

Sentilo espera trobar els següents paràmetres:

```
catalog.mongodb.host = 127.0.0.1
catalog.mongodb.port = 27017
catalog.mongodb.user = sentilo
catalog.mongodb.password = sentilo
```

Figura A.8: Paràmetres per defecte MongoDB

Com anteriorment els paràmetres *port* i *host* que venen per defecte a MongoDB coincideixen amb els que cerca tenir Sentilo, per tant això no farà falta modificar-ho.

Per altre banda si que farà falta crear una base de dades anomenada *sentilo* amb les característiques *username:sentilo, password:sentilo*

En el cas de voler canviar qualsevol dels paràmetres dins el software de Sentilo que fan referència a MongoDB es poden trobar a les següents carpetes:

```
sentilo-agent-alert/src/main/resources/properties/catalog-config.properties
sentilo-catalog-web/src/main/resources/properties/catalog-config.properties
```

Figura A.9: Configuració de MongoDB a Sentilo

Per altre part, es necessari carregar en la base de dades sentilo el conjunt bàsic de dades necessàries per executar la plataforma. Per fer això s'haurà de carregar les dades definides al següent arxiu:

```
./scripts/mongodb/init_data.js
```

Per fer això dins la màquina MongoDB, s'ha d'executar la següent comanda des del directori on es localitza l'arxiu:

```
$ mongo -u sentilo -p sentilo sentilo init_data.js
```

A.4.3 MySQL

Sentilo espera trobar els següents paràmetres:

```
sentiloDs.jdbc.driverClassName = com.mysql.jdbc.Driver
sentiloDs.url = jdbc:mysql://127.0.0.1:3306/sentilo
sentiloDs.username = sentilo_user
sentiloDs.password = sentilo_pwd
```

Figura A.10: Paràmetres per defecte MySQL

Els paràmetres *port* i *host* que venen per defecte a MySQL coincideixen amb els que cerca tenir Sentilo, per tant això no farà falta modificar-ho.

Per altre banda, una vegada més, farà falta crear una base de dades anomenada *sentilo* amb les característiques *username:sentilo_user, password:sentilo_pwd*

En el cas de voler canviar qualsevol dels paràmetres dins el software de Sentilo que fan referència a MySQL es poden trobar a les següents carpetes:

```
sentilo-agent-relational/src/main/resources/properties/relational-config.properties
```

Figura A.11: Configuració de MySQL a Sentilo

Una vegada que s'hagi configurat MySQL i creat les bases de dades Sentilo, el següent pas es crear les taules de bases de dades necessàries per mantenir les dades històriques de la plataforma. On següent directori es troba la instal·lació de les taules i el *scrip*:

```
sentilo-agent-relational/src/main/resources/bd
```

A.4.4 Tomcat

Per defecte Sentilo considera que Tomcat estarà al host i port: 127.0.0.1:8080, que es la configuració que ve per defecte a Tomcat, pel que no s'han modificat els valors.

En el cas de voler-se modificar, cal canviar la següent propietat:

```
catalog.rest.endpoint = http://127.0.0.1:8080/sentilo-catalog-web/
```

Figura A.12: Configuració de Tomcat a Sentilo

La qual es pot trobar als següents directoris:

```
sentilo-platform/sentilo-platform-service/src/main/resources/properties/integration.properties
sentilo-agent-location-updater/src/main/resources/properties/integration.properties
```

Figura A.13: Arxius de configuració Tomcat a Sentilo

Tomcat també es deu iniciar amb la variable d'entorn de temps del usuari establerta com UTC. Per establir *Timezone* en Tomcat cal modificar el script d'inici *catalina.sh* amb el següent codi:

```
-Duser.timezone=UTC
```

A.4.5 Elasticsearch

Com a altres apartats Sentilo considera que Elasticsearch es troba al port *localhost:9200* per defecte, paràmetre que no s'ha modificat. Per modificar aquest comportament cal anar introduir el següent script al directori corresponent:

```
elasticsearch.url = http://localhost:9200
```

Figura A.14: Script de modificació

```
sentilo-agent-activity-monitor/src/main/resources/properties/monitor-config.properties
```

Figura A.15: Arxius de configuració Elasticsearch a Sentilo

A.4.6 OpenTSDB

Com a l'apartat anterior, en aquest cas només cal fixar-se al port on estarà OpenTSDB, si no es vol modificar res, sentilo escoltarà per defecte a *127.0.0.1:4242* que es la configuració amb la que es descarrega OpenTSDB. Si es volgués modificar caldrà modificar el següent script al directori corresponent:

```
opentsdb.url = http://127.0.0.1:4242
```

Figura A.16: Script de modificació

```
sentilo-agent-historian/src/main/resources/properties/historian-config.properties
```

Figura A.17: Arxius de configuració OpenTSDB a Sentilo

A.5 Instal·lació de la plataforma

Una vegada descarregat el codi, compilat i construït, el següent pas es el desplegament dels artefactes de Sentilo. La plataforma té cinc artefactes:

- Web Application Catalog (obligatori).
- Server publication and subscription (obligatori).
- Agents interns els quals son opcionals, però en aquest cas s'han desplegats tots:
  - alarms agent

- relational database agent
- location updater agent

A.5.1 Instal·lació del Web Application Catalog

Després de crear Sentilo, per instal·lar l'aplicació Web sols es necessari desplegar l'artefacte *WAR* en el servidor Tomcat, es a dir, copiar aquest artefacte en el subdirectori *webapps* del servidor Tomcat. Aquest artefacte es pot trobar al següent subdirectori:

```
./sentilo-catalog-web/target/sentilo-catalog-web.war
```

Figura A.18: Subdirectori de l'artefacte WAR

A.5.2 Instal·lació del servidor de subscripció / publicació

Després de construir Sentilo, per instal·lar el servidor PubSub es deuen copiar dos directoris denominats *repo* i *bin* els quals es troben en el directori *./sentilo-platform/sentilo-platform-server/target/appassembler*.

Aquests directoris s'han copiat a una carpeta creada a la màquina, concretament s'ha creat la carpeta *sentilo-server* dins *opt*.

Per copiar aquestes carpetes cal executar els següents scripts:

```
$ sudo cp -r repo /opt/sentilo-server/
$ sudo cp -r bin /opt/sentilo-server/
```

Al escriure cp es fa una còpia de la carpeta, i -r indicam recursivitat, es a dir, que fagi el mateix amb tots els elements interns a la carpeta que copiam.

Una vegada copiat, per iniciar el procés només es necessari executar el següent script:

```
$ sudo sh /opt/sentilo-server/bin/sentilo-server
```

A.5.3 Instal·lació dels agents

Com s'ha comentat anteriorment, tots els agents son opcionals, depenent de les necessitats. Però com també s'ha esmentat, en aquesta màquina s'han instal·lat tots, fent una màquina portadora del servidor Sentilo complet.

Hi ha cinc agents principals:

- Alarms agent, el qual es el responsable de processar cada alerta interna definida, i publicr una notificació quan no es compleixen les regles d'integritat configurades.

- Relational agent, es el responsable d'emmagatzemar tota la informació rebuda del servidor PubSub en un conjunt de bases de dades relacionals.
- Location updater agent, es el responsable d'actualitzar automàticament la ubicació del component d'acord amb la ubicació de les observacions publicades.
- Historian agent, es el responsable d'emmagatzemar tota la informació rebuda del servidor PubSub en una base de dades temporal.
- Activity monitor agent, es el responsable d'indexar tota la informació rebuda del servior PubSub en un servidor de cerca

Tots els agents s'instal·len de manera similar al servidor PubSub, maldament a continuació es detalla la instal·lació de tots aquests.

A.5.4 Instal·lació Alarm Agent

Per instal·lar aquest agent també es deuen copiar dos directoris denominats *repo* i *bin* els quals es troben en el directori *./sentilo-agent-alert/target/appassembler*.

Aquests directoris s'han de copiar a la carpeta creada a la màquina *opt* dins la qual s'ha creat una altre carpeta denominada *sentilo-agent-alert*.

Per copiar aquestes carpetes cal executar els següents scripts:

```
$ sudo cp -r repo /opt/sentilo-agent-alert/
$ sudo cp -r bin /opt/sentilo-agent-alert/
```

Una vegada copiat, per iniciar el procés només es necessari executar el següent script:

```
$ sudo sh /opt/sentilo-agent-alert/bin/sentilo-agent-alert-server
```

A.5.5 Instal·lació Relational Agent

Com abans, copiar els dos directoris denominats *repo* i *bin* els quals es troben en el directori *./sentilo-agent-relational/target/appassembler* i s'han copiat a una carpeta creada denominada *sentilo-agent-relational* amb el següent script:

```
$ sudo cp -r repo /opt/sentilo-agent-relational/
$ sudo cp -r bin /opt/sentilo-agent-relational/
```

Una vegada copiat, una altre vegada cal executar el següent:

```
$ sudo sh /opt/sentilo-agent-relational/bin/sentilo-agent-relational-server
```

A.5.6 Instal·lació Location Updater Agent

A l'hora de anar a cercar els arxius *repo* i *bin* dels agents *Location* i *Historian* va aver un problema, i es que aquests no existien, cercant informació, es va trobar la solució, generar-los amb *Maven*. Per generar aquests arxius amb Maven es va executar el següent script:

```
$ sudo ~/sentilo-agent-location-updater# mvn package appassembler:assemble -P
dev
```

El qual va generar els arxius *repo* i *bin* dins el directori *./sentilo-agent-relational/target/appassembler*. Una vegada generats s'actua de la mateixa manera que als apartats anteriors, es copien a una carpeta creada denominada *sentilo-agent-location-updater* amb el següent script:

```
$ sudo cp -r repo /opt/sentilo-agent-location-updater/
$ sudo cp -r bin /opt/sentilo-agent-location-updater/
```

I s'executa el script que arrancarà el procés:

```
$ sudo sh /opt/sentilo-agent-location-updater/bin/sentilo-agent-location-updater-
server
```

A.5.7 Instal·lació Historian Agent

Com s'ha comentat abans, els paquets d'aquest agent també s'han tengut que generar amb el mateix script utilitzat anteriorment:

```
$ sudo ~/sentilo-agent-historian# mvn package appassembler:assemble -P dev
```

el qual generarà els paquets *repo* i *bin*, els quals es copien a una carpeta creada denominada *sentilo-agent-historian* amb el següent script:

```
$ sudo cp -r repo /opt/sentilo-agent-location-updater/
$ sudo cp -r bin /opt/sentilo-agent-location-updater/
```

Una vegada copiats només faltarà executar el procés:

```
$ sudo sh /opt/sentilo-agent-historian/bin/sentilo-agent-historian-server
```

A.5.8 Instal·lació Activity-Monitor Agent

En aquest darrer agent no va ser necessari generar els arxius, per tant, com abans, tornem copiar els dos directoris denominats *repo* i *bin* els quals es troben en el directori *./sentilo-agent-activity-monitor/target/appassembler* i els copem a una carpeta creada denominada *sentilo-agent-activity-monitor* amb el següent script:

```
$ sudo cp -r repo /opt/sentilo-agent-activity-monitor/
$ sudo cp -r bin /opt/sentilo-agent-activity-monitor/
```

Una vegada copiat, una altre vegada cal executar el següent:

```
$ sudo sh /opt/sentilo-agent-activity-monitor/bin/sentilo-agent-activity-monitor-server
```

A.6 Proba de la plataforma

Una vegada tot instal·lat i configurat es pot validar que tots els serveis de Redis, MongoDB, Tomcat Server i PubSub estan en funcionament.

Primerament cal anar al directori on es té el codi Sentilo i executar el servidor tal com es mostra a continuació:

```
$ sudo sh ./scripts/sentilo-server.sh
```

Una vegada arrencat el servidor serà necessari posar en marxa un arxiu el qual farà una prova per tal de comprovar el funcionament de la plataforma, això es fa introduint el següent script:

```
$ sudo sh ./scripts/testServerStatus.sh
```

Si tot ha anat be, el script ha de retornar el codi *1111* tal com es pot veure a la figura A.19 cada un dels quals fa referència a cada un dels serveis, en cas de rebre qualsevol '0' cal mirar a quin servei es refereix i aquest serà el que tendrà l'error.

```
eps@tsdb:~/sentilo/scripts$ sudo bash testServerStatus.sh
1111
```

Figura A.19: Resposta a la comprovació de la plataforma

APÈNDIX B

CODI PROGRAMACIÓ PIC16F1824

B.1 main

```
/*
 * File:   main.h
 * Author: Oliver Garcia, Juan Antonio
 * Created on 15 Juny de 2017.
 */

#define _XTAL_FREQ 32000000

#include <stdlib.h>
#include <stdio.h>
#include <xc.h>
#include "Init.h"
#include "UART.h"


//Variable global.
unsigned int timer_counter=0;
unsigned int contador_1=0;
unsigned int contador1_anterior=0;
unsigned int contador_2=0;
unsigned int contador2_anterior=0;

void interrupt _ISR_TIM0 (){

    timer_counter++;
    INTCONbits.T0IF = 0;
}
```

Figura B.1: main.c

```
28
29  void start_Timer(){
30      INTCONbits.GIE=0;   //Habilitem les interrupcions
31      TMR0=0;           //Inicielitzem el timer a un valor prdefinit.
32  }
33
34  void eviament_dades(){
35      char valor1[];
36      char valor2[];
37      sprintf(valor1,"%d", contador_1);
38              sprintf(valor2,"%d", contador_2);
39
40                  //Enviament del comptador de persones que entren cap el campus
41              envia_cadena_usart("PUT /data/UIB/prova/");
42              envia_cadena_usart(valor1);
43              envia_cadena_usart(" HTTP/1.1\r\n");
44              envia_cadena_usart("IDENTITY_KEY: e10d8df07a6f8d49bf7d40d343db3ad"
45                      "601bb2d8aaec63e7d3741773e2240d672\r\n");
46              envia_cadena_usart("\r\nConnection: close\r\n");
47
48                  //Enviament del comptdor de persones que surten del campus
49              envia_cadena_usart("PUT /data/UIB/prova/");
50              envia_cadena_usart(valor2);
51              envia_cadena_usart(" HTTP/1.1\r\n");
52              envia_cadena_usart("IDENTITY_KEY: e10d8df07a6f8d49bf7d40d343db3ad"
53                      "601bb2d8aaec63e7d3741773e2240d672\r\n");
54              envia_cadena_usart("\r\nConnection: close\r\n");
55
56                  //Reinicilaització de les variables i el Timer
57              contador1_anterior=0;
58              contador_1=0;
59              contador2_anterior=0;
60              contador_2=0;
61              timer_counter=0;
62  }
63
64  void main(void) {
65      OPTION_REGbits.PSA=1;
66      OPTION_REGbits.PS=0b000;
67
68     //Inicialització del mòdul USART en mode asincron, 8 bits i 9600 baudis
69      TRISCbits.TRISC5=1;//pin RX como entrada digital
70      TRISCbits.TRISC4=0;//pin TX com sortida digital
71      APFCON0bits.TXCKSEL=0; //Habilita RC4 como TX
72      SPBRGH=1; //Aquest parell de registres determinen el periode del baudrate
73      SPBRGL=1;
74      TXSTA=0b00100110;//8bits, Transmissió habilitada, asíncron, alta velocitat
75      RCSTA=0b10010000;
76      BAUDCONbits.BRG16=0;
77      RCSTA=0b10010000;//Habilita USART PIC, recepció 8 bits
78      SPBRG=207;//Per una velocitst de 9600baudis amb un oscilador de 32Mhz
79
80  //Variables per el conteig de persones
81      int Actual_RC1=1;
82      int Anterior_RC1=1;
83      int Actual_RC2=1;
84      int Anterior_RC2=1;
```


Figura B.2: main.c

```
85
86      OSCCONbits.IRCF=0b1110; //Internal Oscillator Frequency Select bits(32MHz))
87      OSCCONbits.SCS=0b00;    //System Clock Select bits (FOSC))
88
89      ANSELA=0;
90      ANSELC=0;
91      TRISAbits.TRISA2=0;     //OUT
92      TRISCbits.TRISC0=0;
93      TRISCbits.TRISC1=1;     //IN
94      TRISCbits.TRISC2=1;
95      TRISCbits.TRISC3=0;     //OUT
96
97      PORTCbits.RC1=0;
98      PORTCbits.RC2=0;
99      PORTAbits.RA2=1;    //Assignació de valors als pins
100     PORTCbits.RC0=1;
101     PORTCbits.RC3=0;
102     PORTCbits.RC4=0;
103     PORTCbits.RC5=0;
104
105     INTCONbits.GIE=0;       //Global interrupts enable
106     INTCONbits.TMR0IE=1;    //Timer 0 interrupt enable
107     INTCONbits.TMR0IF=0;    //Timer 0 interrupt flag
108     INTCONbits.PEIE=0;      //Peripheral Interrupts enable
109
110     OPTION_REGbits.nWPUEN=0;    //NOT WEAK PULL-UPS
111     OPTION_REGbits.TMR0CS=0;    //Timer0 Clock Source Select bit
112     OPTION_REGbits.TMR0SE=0;    //Timer0 Source Edge Select bit
113     OPTION_REGbits.PSA=1;       //Prescaler Assignment bit
114     OPTION_REGbits.PS=0b000;    //Prescaler Rate Select bits
115
116     //configuracio PWM
117     MDCARHbits.MDCH=0b0110;
118     //Configuració del mòdul PWM
119     CCP3CON = 0B00111100;
120             /*||||||||
121               ||||-----> 11XX: Mode PWM activat.
122               ||-------> 00: Dos bits menys significatius del Duty Cycle
123               ---------> 00: Mode PWM simple.*/
124     CCPTMRS = 0X00;       //Es selecciona TMR2 com a temporitzador de control.
125     T2CON = 0X06;         //Prescaler per el TMR2
126     PR2 = 12;
127     CCPR3Lbits.CCPR3L = 9;//Duty de 25%
128     __delay_ms(500);
129
```

Figura B.3: main.c

```
130     while(1){
131
132         Actual_RC1=PORTCbits.RC1;
133         Actual_RC2=PORTCbits.RC2;
134                                 //Si s`activa el sensor 1 abans que el sensor 2
135          if(Anterior_RC1==1 && Actual_RC1==0){
136              Anterior_RC1=Actual_RC1;
137
138              for(int a=100;a>0;a--){
139                  __delay_ms(15);
140                 Actual_RC2=PORTCbits.RC2;//i posteriorment s`activa el sensor 2
141                  if(Anterior_RC2==1 && Actual_RC2==0){
142                      Anterior_RC2=Actual_RC2;
143                      contador_1++;          //contador_1 ++
144                      a=0;
145                      __delay_ms(500);
146                  }
147                 Anterior_RC2=Actual_RC2;
148              }
149          }
150         Anterior_RC1=Actual_RC1;
151         //Si s'activa el sensor 2 abans que el sensor 1
152           if(Anterior_RC2==1 && Actual_RC2==0){
153              Anterior_RC2=Actual_RC2;
154              for(int a=100;a>0;a--){
155               __delay_ms(15);
156               Anterior_RC1=Actual_RC1;
157               if(PORTCbits.RC1==0){  // i el sensor 1 s'activa posteriorment
158                 Anterior_RC1=Actual_RC1;
159                 contador_2++;           //contador_2 ++
160                 a=0;
161                 __delay_ms(500);
162               }
163               Anterior_RC1=Actual_RC1;
164              }
165           }
166           Anterior_RC2=Actual_RC2;
167
168          //Si varia qualsevol dels comptadors es posa en marxa el Timer0
169          if (contador_1!=contador1_anterior || contador_2!=contador2_anterior){
170              start_Timer();
171              contador1_anterior=contador_1;
172              contador2_anterior=contador_2;
173              timer_counter=0;
174          }
175
176           //Una vegada el Timer a desbordat una sèrie de vegades,
177           //qual cosa vol dir que s'ha complit un temps sense haver
178           //passat ningú s'envien les dades
179          if(timer_counter==100){
180             INTCONbits.GIE=0;   //Deshabilitem les interrupcions
181             eviament_dades();
182             }
183     }
184 }
185
```


Figura B.4: main.c

B.2 Init.h

```
/*
 * File:   Init.h
 * Author: Oliver Garcia, Juan Antonio
 * Created on 15 Juliol de 2017.
 */

#include <xc.h>

// CONFIG1
#pragma config FOSC = INTOSC  // Oscillator Selection (INTOSC oscillator)
#pragma config WDTE = OFF     // Watchdog Timer Enable (WDT disabled)
#pragma config PWRTE = ON     // Power-up Timer Enable (PWRT enabled)
#pragma config MCLRE = ON     // MCLR Pin Function Select (pin function is MCLR)
#pragma config CP = OFF       // Flash Program Memory Code Protection
#pragma config CPD = OFF      // Data Memory Code Protection
#pragma config BOREN = OFF    // Brown-out Reset Enable(Brown-out disabled)
#pragma config CLKOUTEN = OFF // Clock Out Enable (CLKOUT function is disabled)
#pragma config IESO = OFF     // Internal/External Switchover (disabled)
#pragma config FCMEN = OFF    // Fail-Safe Clock Monitor Enable (Disabled)

// CONFIG2
#pragma config WRT = OFF      // Flash Memory Self-Write Protection off
#pragma config PLLEN = ON     // PLL Enable (4x PLL enabled)
#pragma config STVREN = ON    // Stack Overflow/Underflow Reset Enable
#pragma config BORV = LO      // Brown-out Reset Voltage Selection
#pragma config LVP = OFF      // Low-Voltage Programming Enable
```

Figura B.5: Init.h

B.3 UART.h

```
/*
 * File:   UART.h
 * Author: Oliver Garcia, Juan Antonio
 * Created on 15 Juliol de 2017.
 */

#include <xc.h>
#include "Init.h"

void envia_caracter_usart(unsigned char caracter);
void envia_cadena_usart(char* cadena);
```

Figura B.6: UART.h

B.4 UART.c

```
/*
 * File:   UART.c
 * Author: Oliver Garcia, Juan Antonio
 * Created on 15 Juliol de 2017.
 */


#include <xc.h>
#include "UART.h"


    void envia_caracter_usart(unsigned char caracter){
      while(TXSTAbits.TRMT==0);// Espera mentre el registre TSR està ple.
    TXREG = caracter;//Quan el registre TSR sigui buit enviarà el caràcter.
}

void envia_cadena_usart(char* cadena){// Cadena de caracters tipus 'char'
            //Mentre el darrer valor de la cadena sigui diferent del caracter nul
    while(*cadena !=0x00){

        envia_caracter_usart(*cadena);//Envia els caràcters de la cadena
        //Incrementa la posició de la cadena (punter)
        //per enviar els següents caràcters
        cadena++;
    }
}
```

Figura B.7: UART.c

APÈNDIX C

CONFIGURACIÓ MÒDUL WI-FI

```
<4.41> get everything
wifly-EZX Ver: 4.41 Build: r1057, Jan 17 2014 10:23:56 on RN-171
SSID=WiFly-EZX-df
Pass=
Linkmon=3600
Beacon=102
Reboot=0
IF=DOWN
DHCP=ON
IP=10.164.33.3:2000
NM=255.255.248.0
GW=10.164.32.1
HOST=172.22.39.171:8081
PROTO=TCP,HTTP,
MTU=1524
FLAGS=0x7
TCPMODE=0x0
BACKUP=0.0.0.0
OPEN=*OPEN*
CLOSE=*CLOS*
REMOTE=
FlushSize=1420
MatchChar=0
FlushTimer=5
IdleTimer=0
CmdChar=$
DNS=130.206.33.1
Name=172.22.39.171
Backup=rn.microchip.com
Lease=86400
```

Figura C.1: Configuració mòdul

```
FTP=0.0.0.0:21
File=wifly-EZX.img
User=roving
Pass=Pass123
Dir=public
Timeout=200
FTP_mode=0x0
SSID=uib@uib
Chan=0
ExtAnt=0
Join=1
Auth=WPA2
Mask=0x1fff
Rate=12, 24 Mb
Linkmon=30
Passphrase=MjH&34Dg#p9@
TxPower=12
EAP_Id=userid
EAP_User=peap-user
SleepTmr=0
WakeTmr=0
Trigger=0x1
Autoconn=0
IoFunc=0x20
IoMask=0x21d0
IoValu=0x0
DebugReg=0x0
PrintLvl=0x1
LaunchStr=web_app
TimeEna=0
TIMEADR=64.90.182.55:123
Zone=7
Baudrate=9600
Flow=0x0
Mode=0x2
Cmd_GPIO=0
JoinTmr=1000
Replace=0x24
DeviceId=WiFly-EZX
Password=
Format=0x1
Signal=0
Average=5
BCAST=255.255.255.255:55555
Interval=0x7
Backup=0.0.0.0:0
Sensor=0x0
SensePwr=0x0
```

Figura C.2: Configuració mòdul

APÈNDIX D

FABRICACIÓ DELS PALS

La fabricació de la caixa s'ha duit a terme amb una impressora 3D, exactament una Anet A8. La decisió de fabricar-la a mida i no comprar una estàndard ve donada per encaixar el més perfectament possible en el pal on aniran els sensors, per tal d'integrar-la el més possible i així produir un impacte visual menor.

El material emprat per la fabricació d'aquesta caixa es el PLA. El PLA es un material el qual té unes propietats semblants al que s'utilitza per fabricar els envasos dels comerços, però a més es biodegradable. Aquest plàstic es un polímer que s'obté de substàncies orgàniques. Aquesta gran virtut de ser biodegradable i per lo qual no contaminant es la major causa d'aquesta elecció.

També hi ha una condició en contra, i es que al ser biodegradable li afecten les condicions mediambientals, tant la humitat com els raigs UV, per lo qual s'ha de protegir, això es pot aconseguir mitjançant una capa de pintura o qualsevol emprimació. En aquest cas s'ha optat per una pintura aïllant que produeix un revestiment plàstic al voltant de l'objecte.

El procés de fabricació es molt senzill, una vegada tenim dissenyada la peça, aquest disseny es passa a un software el qual depenent de les característiques de la impressora divideix l'objecte en punts a l'espai, els qual seran seguits per la boca d'aquesta.

El plàstic utilitzat necessita una temperatura de fusió d'uns 200ºC i, maldament no es necessari un suport calent, s'ha optat per encalentir la base a uns 50ºC i a més dipositar cola blanca dissolta en aigua damunt la base calenta per a que el material quedi mes firme sobre la base. Aquesta base calenta s'anomena *llit calent*.

A la següent figura es pot veure un moment del procés de fabricació de la caixa sobre la impressora:

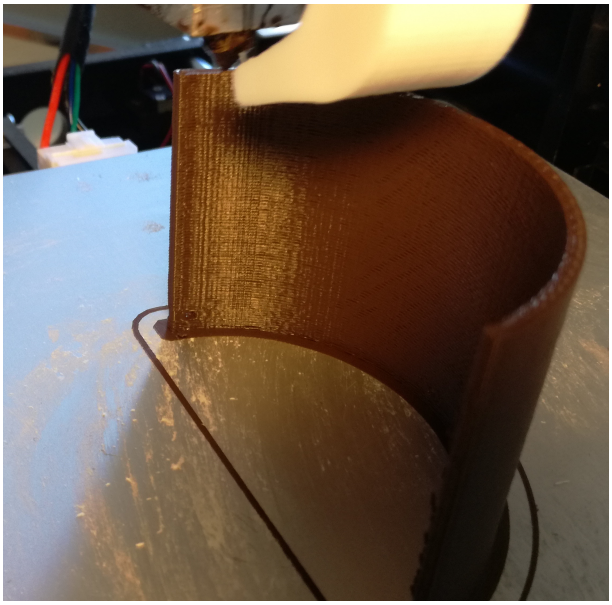


Figura D.1: Fabricació de la caixa

Ja que aquesta caixa ha de estar a la intempèrie, al disseny de la caixa es varen fer unes mosques on anirà una junta de goma, la qual al posar-se els perns, estrenyeran aquesta junta, fent la caixa hermètica a la intempèrie. La caixa fabricada es pot veure a continuació, on també es pot visualitzar al lloc de la junta la goma que dotarà d'estanquitat a la caixa :

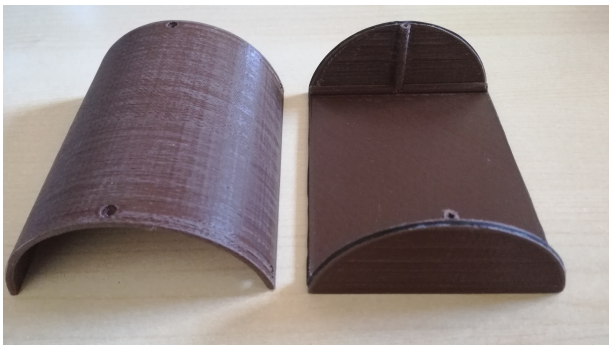


Figura D.2: Caixa fabricada

Aquesta caixa té un radi exactament igual que el pal on anirà muntada. El procediment sobre el pal, ha estat extreure una part d'ell exactament igual que les dimensions de la caixa. La feina realitzada a estat a fer uns tall paral·lels de l'amplada de la caixa i una profunditat igual a la de la caixa i seguidament poc a poc i amb un poc de paciència anar buidant amb un punta-corrent. A continuació es pot veure la feina de tallar el pal i el posterior muntatge de la caixa sobre ell:

Figura D.3: Fabricació dels pals

Un altre muntatge a fer sobre el pal ha estat fabricar i dissenyar el suport que fixarà la placa solar a una inclinació d'uns 30º sobre l'horitzontal i inclourà la bateria d'alimentació juntament amb la regulació de càrrega. Aquesta inclinació es la recomanada per la localització de Mallorca, i en el cas de voler afavorir l'hivern es recomanable augmentar aquesta inclinació, que es el que s'ha fet augmentant la inclinació uns 4º més. També cal dir que al igual que en la caixa anterior, aquesta també es estanca amb l'ajuda d'una junta de goma incrustada.

A continuació es pot veure com s'ha incrustat aquest suport al pal, el que s'ha fet es devastar la part superior del pal per tal de introduir aquest dins el suport, d'aquesta manera tota la superfície del pal serà llissa sense cap abruptament que sobresurti. Per passar l'alimentació des de la part superior del pal fins la caixa de la placa s'ha fet un orifici amb una broca de 50 cm, així s'aconsegueix no rompre l'estètica del pal amb cables a la vista o amb regates que impacten visualment.

Figura D.4: Introducció de la placa als pals

Per a la inserció dels sensors i els retrorreflectors s'ha fet ús dels suports esmentats a l'apartat 5.2. Els suports dels sensors son cilíndrics pel qual motiu només es necessari fer uns forats amb una broca del mateix diàmetre que els suports. A això va haver un problema i era que els suports estaven dissenyats per anar aun lloc específic donat que se'ls va donar una curvatura d'un punt exacte, per el que es va dissenyar una plantilla també en la impressora 3d, la qual cosa va facilitar la feina, ja que la plantilla indicava on es tenia que clavar l'inici de la broca. Aquesta plantilla es pot veure a la següent figura:

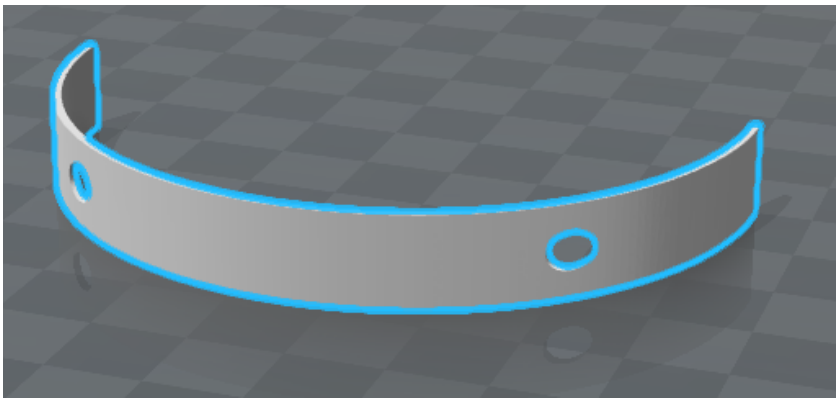


Figura D.5: Guia per al suport dels sensors

Aquesta guia es va col·locar al pal i només es van tenir que fer els forats al lloc on indicava, quedant com a resultat:

Figura D.6: Sensors muntats al pal

Fet això només queda el darrer, inserir els retrorreflectors a l'altre pal. Això va dur un petit problema, aquest va ser que els dos retrorreflectors eren més amples que el mateix pal i el pal era rodó, pel que es va fer una endidada fins a la meitat del pal per tal de agafar la major amplada possible i per el sobrant dels retrorreflectors dels costats es va dissenyar un petit suport estètic per millorar aquest petit impacte visual. Aquesta feina es pot veure a la següent imatge:

Figura D.7: Muntatge dels retroreflectors

BIBLIOGRAFIA

[1] A. G. S. Laura Liliana Moreno Herrera, "Ciudades inteligentes: Oportunidades para generar soluciones sostenibles," *Estudios Sectoriales CINTEL - Proyectos TIC Innovadores*, 2012. [Online]. Available: http://cintel.org.co/wp-content/uploads/2013/05/01.Ciudades_Inteligentes_CINTEL.pdf (document), 2.1

[2] F. Bosquiazzo, "Internet and social," 2015. [Online]. Available: https://tecno.americaeconomia.com/opinion/por-que-hoy-es-el-momento-para-implementar-la-internet-de-las-cosas (document)

[3] M. Carulla, "Sentilo," 2015. [Online]. Available: http://www.sentilo.io/xwiki/bin/view/Sentilo.About.Product/Whatis 1, 2.1

[4] Administrator, "Setup sentilo," 2017. [Online]. Available: http://www.sentilo.io/xwiki/bin/view/Sentilo.Community.Documentation/Setup 2.2, A.2

[5] I. Postdot Technologies, "Postman," 2017. [Online]. Available: https://www.getpostman.com/ 2.4

[6] *PIC16(L)F1824/1828*, Microchip Technology Inc., 2010-2011. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/41419C.pdf 3.2.1

[7] *datasheet RN-XV*, Roving Networks, 2013. [Online]. Available: https://cdn.sparkfun.com/datasheets/Wireless/WiFi/rn-wiflycr-ug-v1.2r.pdf 3.2.2

[8] *datasheet TSOP4838 Receiver modules*, VISHAY INTERTECHNOLOGY, INC., 2017. [Online]. Available: http://www.vishay.com/docs/82459/tsop48.pdf 3.2.3

[9] *datasheet TSAL6200, High Power Infrared Emitting Diode*, VISHAY INTERTECHNOLOGY, INC., 2017. [Online]. Available: https://www.vishay.com/docs/81010/tsal6200.pdf 3.2.4

[10] *datasheet REGULADOR NCV1117*, Semiconductor Components Industries, LLC, 2017. [Online]. Available: https://www.onsemi.com/pub/Collateral/NCP1117-D.PDF 3.2.7

[11] *datasheet STN2NF10*, STMicroelectronics, 2003. [Online]. Available: http://docs-europe.electrocomponents.com/webdocs/0478/0900766b8047858c.pdf 3.2.8

[12] *datasheet CPH6341*, SANYO Semiconductor Co., 2012. [Online]. Available: http://docs-europe.electrocomponents.com/webdocs/11a6/0900766b811a640c.pdf 3.2.8

[13] *Eagle pcb design.*, Autodesk, Inc, 2017. [Online]. Available: https://cadsoft.io/ 3.3

[14] *MPLAB X IDE*, Microchip Technology Inc., 2017. [Online]. Available: http://www.microchip.com/mplab/mplab-x-ide 4.1.1

[15] *PICkit 3 In-Circuit Debugger/Programmer Users Guide*, Microchip Technology Inc., 2013. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/52116A.pdf 4.1.1, 4.1.1

[16] *Download GIT*. [Online]. Available: https://git-scm.com/download/linux A.1.2

[17] *Download Maven*. [Online]. Available: http://www.happyminds.es/maven-instalar-y-probar-maven/ A.1.3

[18] *Download Redis*. [Online]. Available: https://redis.io/download A.3.1

[19] *Install MongoDB Community Edition on Linux*. [Online]. Available: https://docs.mongodb.com/manual/administration/install-on-linux/ A.3.2

[20] *How To Install Apache Tomcat on Ubuntu*. [Online]. Available: https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-7-on-ubuntu-14-04-via-apt-get A.3.3

[21] *How To Install and Configure Elasticsearch*. [Online]. Available: https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-elasticsearch-on-ubuntu-14-04 A.3.4

[22] *How To Install MySQL*. [Online]. Available: https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-16-04 A.3.5

[23] *ZooKeeper Getting Started Guide*. [Online]. Available: https://zookeeper.apache.org/doc/r3.1.2/zookeeperStarted.html A.3.6

[24] *Insttall HBase*. [Online]. Available: https://www.tutorialspoint.com/es/hbase/hbase_installation.htm A.3.6