



Treball Final de Grau

Programació de codis QR

Carles Oró Arán

Grau en Enginyeria Mecatrònica

Tutor/a: Jordi Surinyac i Ignasi Riera

Vic, setembre de 2014

RESUM TREBALL FINAL DE GRAU

GRAU EN MECATRÒNICA

Títol: Programació de codis QR

Paraules claus: KIT AM335X SK i codis QR

Autor: Carles Oró Arán

Tutors: Jordi Surinyac i Ignasi Riera

Data: Setembre de 2014

Aquest projecte es basa en la programació de codis QR a temps real en el KIT AM335X SK. S'inclou dins l'estudi previ d'una nova sèrie de rentadores de l'empresa Girbau S.A.

Alhora, també consta d'un altre software, compilat per Windows, el qual et permet obtenir els correus electrònics (aquests correus son fruit de generar, escanejar i enviar els codis QR) del servidor i posar-los en un full d'Excel. Posteriorment, aquests son tractats segons el tipus que tenen.

El programa realitzat pel KIT, està desenvolupat mitjançant l'editor QT Creator amb Linux i el per descarregar els correus electrònics del servidor mitjançant el Visual Studio per Windows.

Abstract of Final Degree Project
Mechatronics Engineering

Title: QR codes programation

Key words: KIT AM335X SK and QR codes

Author: Carles Oró Arán

Directorate: Jordi Surinyac i Ignasi Riera

Date: September 2014

This project is based in the real time QR codes programation on the KIT AM335X SK. It is included in the preliminary study for a new series of washing machines for the company Girbau S.A.

In the same time, this project also included another software, compiled by Windows, that allows you to get the e-mails (this e-mails are obtain by generating, scanning and sending the QR codes) from the server and put into a spare sheet. After, this e-mails, are processed depending the type that they have.

The program make for the KIT, it's developed by the QT Creator editor in Linux, and the program to download the e-mails is developed by the Visual Studio.

AGRAÏMENTS:

M'agradaria donar les gràcies a totes les persones que han fet possible que pugues realitzar aquest projecte, sense les quals no crec que hagués sortit tant bé. A totes aquestes persones els hi agraeixo molt tot l'esforç, temps i dedicació que han fet per mi.

En primer lloc, agrair a l'Enric, l'Ignasi i en Jordi, i a l'empresa Girbau per tal d'ajudar-me a realitzar el projecte i també per donar-me la oportunitat de realitzar el meu projecte a l'empresa Girbau S.A.

Donar les gràcies en especial amb en Jordi, per proposar-me un projecte molt divertit, en el qual m'ho he passat molt be realitzant-lo. També donar-li les gràcies pels consells que m'ha donat i totes les reunions que hem fet per tal d'encaminar el projecte cap al bon camí.

També donar les gràcies amb en Gil i l'Alfons, per tal de facilitar-me els mitjans, tant l'ordinador portàtil com el KIT AM335X SK per realitzar el projecte, sense el qual tampoc ves pogut fer-lo.

També agrair moltíssim al meu tutor de la universitat, en Jordi, el qual m'ha encaminat molt per tal de realitzar un bon projecte i ajudat en els moments que estava més perdut. Donar-li també les gràcies per tot el temps que ha invertit en mi, l'esforç i dedicació que sempre ha mostrat i amb les diverses reunions que hem anat fent durant el transcurs d'aquest projecte.

Per últim, donar les gràcies a tota la família, amics i companys, els quals m'han ajudat molt pel suport moral que m'han donat en la realització del projecte. Estic segur que sense ells tampoc ves aconseguit fer un bon projecte.

ÍNDEX:

RESUM

ABSTRACT

CAPÍTOL 1. INTRODUCCIÓ

1.1	Motivació	1
1.2	Objectius i hipòtesis	1
1.2.1	Objectius	1
1.2.2	Hipòtesis	2
1.3	Metodologia	2
1.3.1	Plantejament del projecte	2
1.3.2	Recerca de informació	3
1.3.3	Implementació	3
1.4	Presentació de l'empresa	3

CAPÍTOL 2. INTRODUCCIÓ ALS CODIS QR

2.1	Que són els codis QR	5
2.2	Com funcionen	5
2.3	Avantatges respecte els codis de barres	5
2.4	Guardar la informació	6
2.5	Estructura dels codis QR	7

CAPÍTOL 3. EINES UTILITZADES

3	Eines utilitzades	9
---	-------------------------	---

CAPÍTOL 4. ESQUEMA GENERAL

4	Esquema general	14
---	-----------------------	----

CAPÍTOL 5. ESQUEMA CONCRET

5	Esquema concret	15
---	-----------------------	----

CAPÍTOL 6. PROCÉS DE CONSTRUCCIÓ DE LA SOLUCIÓ

6.1	Buscar llibreria per programar els codis QR	17
6.2	Proves amb la llibreria	17
6.3	Proves amb el KIT AM335X SK	19
6.4	Generar QR al KIT	22
6.5	Programa definitiu	24
6.5.1	Pantalla de informació	25
6.5.2	Pantalla de generar els codis QR	26
6.5.3	Pantalla de generar els codis QR de manteniment	27
6.5.4	Pantalla de generar els codis QR de informació	29
6.5.5	Pantalla de generar els codis QR d'alarma	30
6.5.6	Pantalla de generar els codis QR per obrir el manual	31
6.6	Buscar la API per connectar al servidor Gmail	32
6.7	Proves amb la API	33
6.8	Programa definitiu	35
6.9	Implementar la macro d'Excel	35
6.10	Realitzar un suport pel KIT	38

CAPÍTOL 7. MANUAL D'UTILITZACIÓ

7.1	Procés	40
-----	--------------	----

CAPÍTOL 8. MILLORES

8	Millores	42
---	----------------	----

CAPÍTOL 9. CONCLUSIONS

9	Conclusions	44
---	-------------------	----

CAPÍTOL 10. WEBGRAFIA

10	Webgrafia	46
----	-------------------	----

CAPÍTOL 11. ANNEXES

11.1	Plànol 1. Tallar i doblar la xapa	48
------	---	----

11.2	Plànol 2. Pintar el suport	49
------	----------------------------------	----

ÍNDIX D'IMATGES:

Imatge 1.1 – Rentadora Girbau HS-6017	4
Imatge 2.1 - Diferència entre el codi QR i el codi de barres	6
Imatge 2.2 - Versions dels codis QR amb els seus mòduls	7
Imatge 2.3 - Estructura del codi QR	8
Imatge 3.1 – Programa QT Creator	10
Imatge 3.2 – Programa Visual Studio	11
Imatge 3.3 – Programa Excel	12
Imatge 3.4 – Programa Creo 2	13
Imatge 3.5 – Programa Photoshop CS5	13
Imatge 4.1 – Esquema del projecte	14
Imatge 6.1 – Captura de pantalla del terminal d'Ubuntu amb el qrencode	17
Imatge 6.2 – Codi QR	18
Imatge 6.3 – Codi QR per enviar un e-mail	18
Imatge 6.4 – Captura de pantalla de la primera aplicació amb el QT	19
Imatge 6.5 – Captura de pantalla de l'aplicació executada	20
Imatge 6.6 – Primera aplicació feta amb el KIT AM335X SK	21
Imatge 6.7 – Codi per executar un programa extern	22
Imatge 6.8 – Primera prova del pla B. Es pot observar tots els fitxer “.c” i “.h”	23
Imatge 6.9 – Pantalla d'informació del KIT	26
Imatge 6.10 – Pantalla principal de generar els codis QR	26
Imatge 6.11 – Pantalla per generar el QR de manteniment	27
Imatge 6.12 – Pantalla per escriure un comentari	28
Imatge 6.13 – Pantalla amb el QR de manteniment ja generat	28

Imatge 6.14 – Pantalla amb el QR de informació	29
Imatge 6.15 – Pantalla amb el QR d’alarmes	30
Imatge 6.16 – Pantalla amb el QR del manual	31
Imatge 6.17 – Captura de pantalla del Visual Studio per buscar la API mail	32
Imatge 6.18 – Captura de pantalla del Visual Studio per connectar-se al servidor	33
Imatge 6.19 – Captura de pantalla del Visual Studio per obtenir els e-mails i treure’ls per pantalla	34
Imatge 6.20 – Captura de pantalla d’un e-mail descarregat	34
Imatge 6.21 – Captura de pantalla de la marco d’Excel per mirar l’última posició ocupada	36
Imatge 6.22 – Captura de pantalla de la macro d’Excel per importar el correu electrònic	37
Imatge 6.23 – Captura de pantalla de la macro d’Excel per posar els colors segons el tipus	37
Imatge 6.24 – Captura de pantalla de la macro d’Excel per posar el color al tipus “Retirada”	38
Imatge 6.25 – Foto del suport del KIT AM335X SK	39
Imatge 7.1 – Imatge del procés que segueixen per generar i enviar els codis QR	40
Imatge 11.1 – Plànol de tall i doblat de la peça de xapa	48
Imatge 11.2 – Plànol de pintat del suport del KIT	49

CAPÍTOL 1: Introducció.

En aquest capítol es realitzarà una introducció per començar el treball, definint els objectius claus del projecte i el procés que es farà servir per assolir-los. A la vegada, es mostrarà una explicació de les motivacions que fan que es porti a terme aquest treball.

1.1 Motivació:

Les arrels d'aquest treball sorgeixen per tal de millorar la feina del servei d'assistència tècnica (SAT) i per tenir més control de les averies que hi ha a les màquines de Girbau. Per això, a la nova sèrie de rentadores, es volia que poguessin generar codis QR a temps real per tal de poder enviar informació de la màquina mitjançant un correu electrònic. D'aquí va sorgir la idea del meu TFG, el qual m'he la va presentar Girbau S.A i la vaig trobar molt interessant i del meu agrat, ja que hi havia una part important de programació, però alhora, també una part de gestió dels correus electrònics.

1.2 Objectius i hipòtesis:

1.2.1 Objectius:

L'objectiu principal del treball, com he dit abans, és la generació de codis QR a temps real a la màquina, en concret amb el KIT AM335X SK. A partir d'aquí, una altra part del projecte és la de gestió dels correus que arriben al servidor de Girbau – en el meu cas, per tal de fer proves, he utilitzat el servidor de gmail -. Aquests correus provenen dels codis QR que s'han generat a la màquina, els quals, l'usuari, o bé el SAT (servei d'assistència tècnica), a descodificat (mitjançant aplicacions per smartphone) i enviat per tenir-los ben estructurats i classificats en un excel.

Aquí també has de poder ordenar-los segons models de màquines, bugaderies, números de sèrie i data.

1.2.2 Hipòtesis:

La principal hipòtesi que va sorgir al principi del projecte, era si trobaríem alguna llibreria amb llicència oberta que sigues capaç de generar els codis QR o be, hauríem d'utilitzar la ISO/IEC18004 (és la norma dels codis QR) per tal de generar-los des de zero, cosa que se'ns complicava molt el projecte.

1.3 Metodologia:

Per tal de dur a terme una bona elaboració del projecte s'ha creat una metodologia de treball, per poder desenvolupar de forma continuada cada apartat.

1.3.1 Plantejament del projecte:

Com ja he dit abans, el projecte bàsicament el vam plantejar amb 2 grans grups. El primer, i el principal, era el de generar codis QR a temps real amb el KIT AM335X SK. Durant la realització d'això, l'autor va tenir diverses reunions amb el tutor de l'empresa Girbau S.A per tal de, per exemple, mirar quins tipus de correus electrònics enviàvem, com estava estructurada la pantalla del KIT entre altres. Finalment, com que encara quedava temps, l'autor va poder programa un programa capaç de gestionar els correus que arribaven al servidor de Girbau S.A a causa de l'enviament dels codis QR anteriors, per així poder tenir un control de la informació rebuda.

1.3.2 Recerca de informació:

La primera recerca per la xarxa va estar la de buscar informació per tal de poder, amb Linux, compilar el programa i que sigui executable pel KIT AM335x SK. També s'ha realitzat una recerca per la xarxa per tal de trobar llibreries en llicència oberta per poder generar els codis QR, i finalment per trobar una API que es connectes amb el servidor de gmail per descarregar els correus electrònics.

1.3.3 Implementació:

Una vegada trobada la llibreria i la API, la implementació del projecte ha estat casi tota pràctica, de poder desenvolupar unes pantalles noves en el KIT a on s'hi generin codis QR i finalment la realització d'una altre aplicació capaç de poder connectar-se al servidor, descarregar els correus electrònics que hi hagi i afegir-los a un full de càlcul d'Excel pel seu processat.

1.4 Presentació de l'Empresa

L'empresa per la qual he realitzat el Treball de final de grau, ha estat la de Girbau S.A. Aquesta és una empresa familiar i multinacional, la qual va ser fundada fa 90 anys per en Pere Girbau. Anteriorment era un taller electromecànic, en el qual reparaven tot tipus d'aparells, com podrien ser ràdios, taules de raigs X, etc. En paral·lel també es van obrir botigues de Radio Girbau, dedicades a la venda d'electrodomèstics i aparells de radio i TV. Ara fa uns 50 anys, en Pere, en Toni i la Teresa Girbau van ser els que van canviar l'empresa i van optar per passar-se a la bugaderia industrial. D'aquí ha sorgit una gran empresa, reconeguda per tot el món, amb moltes filials i amb la seu principal aquí a Vic.

Avui en dia, fabriquen màquines capaces de rentar, assecar, planxar i plegar, de tal manera que pots tenir una peça de roba -com per exemple un llençol- net i plegat sense que l'usuari hagi de dedicar-hi grans esforços.



Imatge 1.1 – Rentadora Girbau HS-6017

CAPÍTOL 2: Introducció als codis QR.

2.1 Que són els codis QR:

El codis QR (Quick Response Barcode) son un sistema per guardar la informació en una matriu de punts, similar als codis de barres, però que permeten guardar la informació bidimensional, podent així tenir molta més informació (fins a 4000 caràcters l'última versió).

Els codis QR van ser creats a Japó l'any 1994 per l'empresa Denso Wave. L'expansió que han tingut els codis QR es deu a que és un estàndard obert, qualsevol persona pot desenvolupar una idea concreta sobre ells.

Inicialment es van aplicar per fer el seguiment de fabricació de peces per a automòbils, però de seguida s'ha anat estenent.

Un part positiva que tenen aquests codis, és que no necessiten un hardware específic per ser descodificats, ja que per exemple amb una simple aplicació per un mòbil que tingui càmera, pot servir per descodificar-lo.

2.2 Com funcionen:

El seu funcionament es base amb una matriu de dades, la qual amb una codificació bidimensional, permet guardar una gran quantitat de informació amb una mida molt reduïda i una alta fiabilitat gràcies a un sistema de correcció d'errors.

2.3 Avantatges respecte els codis de barres:

La principal avantatge respecte el codi de barres, com ja hem dit anteriorment, és que pot guardar informació en les dues direccions, cosa que fa que pugui contenir molta més informació que un codi de barres normal.



Imatge 2.1 – Diferència entre el codi QR i el codi de barres

Un altre avantatge és que tenen un sistema de correcció d'errors basat en Reed Solomon. El Reed Solomon, és un algorisme, el qual és el receptor el que s'encarrega de corregir els errors i no pas l'emissor com en altres casos.

Aquest sistema de correcció, té 4 nivells:

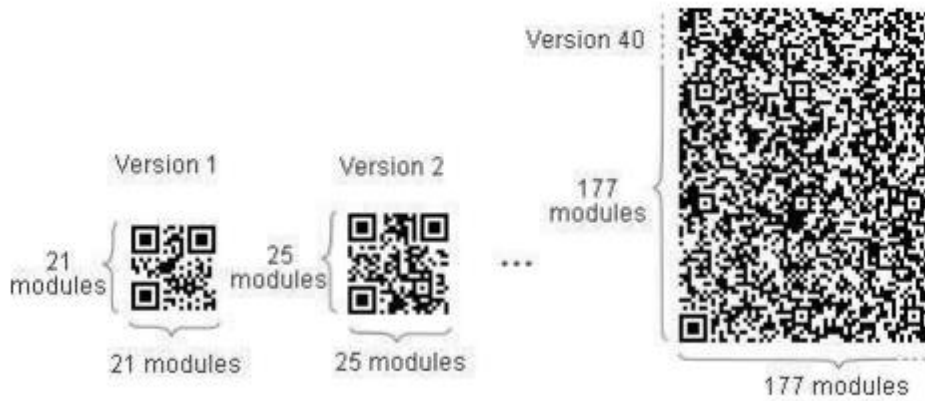
1. L (low): pot corregir fins al 7% dels codewords del símbol.
2. M (medium) pot corregir fins al 15% dels codewords del símbol.
3. Q (quality) pot corregir fins al 25% dels codewords del símbol.
4. H (high) pot corregir fins al 30% dels codewords del símbol.

Ja per acabar, l'últim avantatge més important que té és que es pot llegir a gran velocitat (Quick Response) i des de totes les orientacions (360°) gràcies als patrons de localització de les tres cantonades.

2.4 Guardar la informació:

Fins al moment hi ha 40 versions de codis QR. Cada una d'aquestes versions, té un mòdul diferent de configuració. Un mòdul són els punts blancs i negres que componen els codis QR.

Els mòduls comencen a la versió 1 (21x21 mòduls) fins a la 40 (177x177 mòduls). Cada número més de versió, té 4 mòduls més per cada costat.



Imatge 2.2 – Versions dels codis QR amb els seus mòduls

Els codis QR poden guardar la següent informació:

1. Dades numèriques (0 al 9).
2. Dades alfanumèriques (A a la Z i altres caràcters com: espai, \$, +, *, %, -, /, . i ,).
3. Bytes (Per defecte ISO/IEC 8859-1)
4. Caràcters Kanji (Caràcters de l'escriptura japonesa).

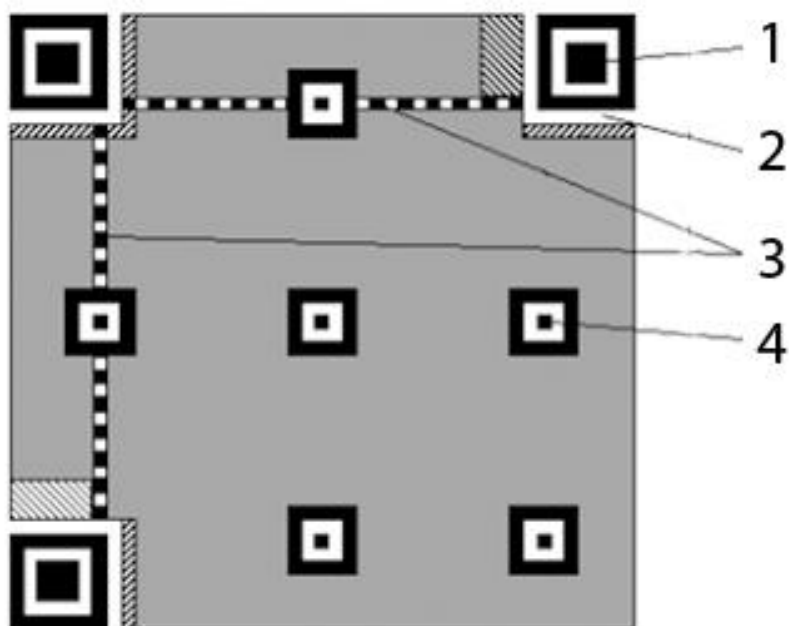
2.5 Estructura dels codis QR:

Els codis QR són formats per mòduls (que son els quadres blancs o negres i representes un 0 i un 1 binari respectivament) posats en una estructura quadrada.

Aquesta estructura té unes determinades zones:

- Patró localitzador (1): Existeix tres vegades en el codi QR col·locat a les cantonades superiors i l'inferior esquerra. Serveix per calcula l'orientació del codi.
- Separador (2): Separa el patró localitzador de la resta de zones del codi QR.

- Patró temporització (3): Seqüència de mòduls blancs i negres que ajuden a calcular les coordenades de la resta de mòduls.
- Patró alineació (4): permet tornar a sincronitzar les coordenades del mapatge de la imatge del codi QR per possibles soroll que pugui ocasionar.



Imatge 2.3 – Estructura del codi QR

CAPÍTOL 3: Eines utilitzades.

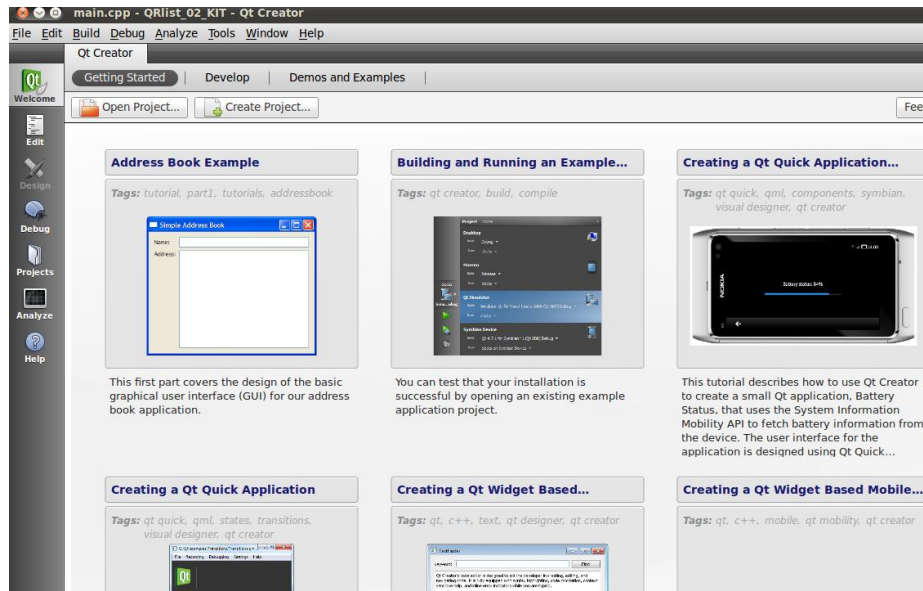
El projecte bàsicament està centrat amb el KIT AM335X SK que pot funcionar amb Sistema Operatiu de Linux o Android. En el nostre cas hem utilitzat el Sistema Operatiu de Linux. Bàsicament és una màquina amb una placa d'Arduino Tre, el qual porta incorporat 2 connexions de xarxa, un USB, una ranura per targeta microSD, wifi, Bluetooth, LEDs i WLAN. Alhora, també porta una pantalla de LCD tàctil de 4,3", la qual cosa vam creure que era una mida molt bona per posar en una rentadora. Si la pantalla fos gaire gran, la rentadora es veuria molt petita i, una pantalla gaire petita, al ser tàctil tots els botons quedarien molt atapeïts i a la rentadora no es veuria bé. Cal mencionar que les rentadores, un cop sortissin de la línia de muntatge ja vindrien amb el KIT posat, ja que és l'única manera que té l'usuari per tal de fer anar la màquina.

Com que vam decidir de programar amb Linux, necessitava una màquina amb aquest sistema operatiu. Primer de tot, vaig instal·lar el Virtual Box al meu PC, un programa el qual hi pots instal·lar diferents màquines virtuals. En el meu cas hi vaig instal·lar el sistema operatiu Ubuntu 10.04 ja que és el que recomanava el fabricant del KIT. A causa de que es necessitava instal·lar part del KIT a dins el PC i utilitzar la targeta SD i el llapis de memòria per fer anar el KIT, amb la màquina virtual vaig tenir bastant problemes, i vaig optar per demanar a Girbau un altre ordinador portàtil. Ells molt amablement, me'l van deixar i en aquest hi vaig instal·lar l'Ubuntu 10.04, el QT Creator 2.4.1 (els que recomanava el proveïdor del KIT) i el software addicional del KIT AM335X SK.

El *QT Creator* és una plataforma de programació que utilitza el llenguatge C++, el CSS i el Javascript, és de codi obert i utilitza una programació orientada a objectes.

Alhora, el *QT Creator* disposa d'un editor de codi avançat, un depurador i una interfície gràfica d'usuari.

Un cop instal·lat el QT, l'havíem de poder configurar perquè poguéssim compilar el programa i que funcionés pel KIT. Per sort, a internet, a la mateixa web on vam comprar el KIT, hi havia explicat com fer-ho.



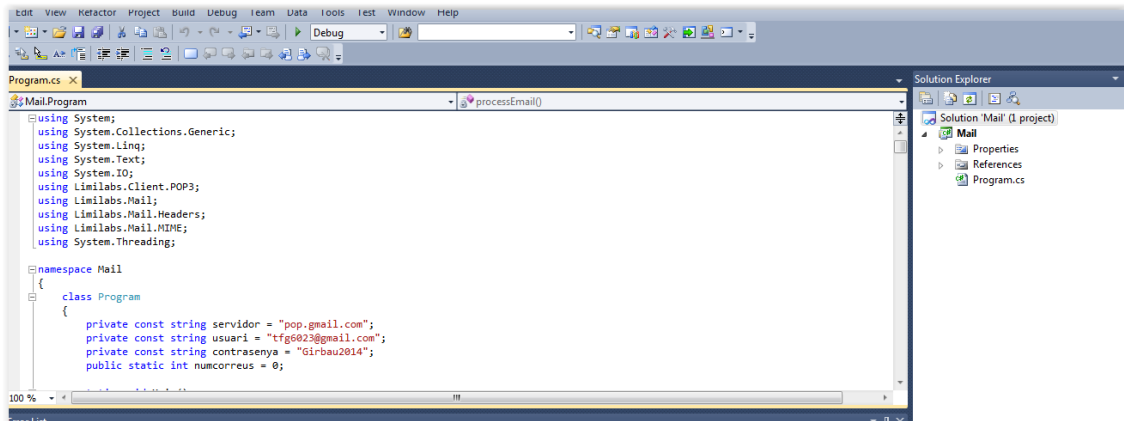
Imatge 3.1 – Programa Qt Creator

Per tal de poder generar el codi QR, vam optar que la millor manera, seria trobar una llibreria en llenguatge C, per l'Ubuntu i de llicència lliure, la qual poguéssim utilitzar i d'aquesta manera facilitar-nos la feina. En vam trobar varies, però ens vam quedar amb la de "qrencode", ja que semblava la més ben documentada i de la qual la gent parlava millor.

Qrencode: És una llibreria desenvolupada per Kentaro Fukuchi el 2006, de llicència lliure i com a conseqüència, que l'autor no té responsabilitats si no funciona bé. Per altre banda, hi ha l'aspecte positiu i és que vam trobar tot el codi font i el vam poder utilitzar.

Per tal de generar la segona part del projecte, la gestió dels correus que rebem enviats pels responsables de les bugaderies o be el SAT (servei d'assistència tècnica), vam decidir de fer aquesta part amb Windows, ja que és el sistema

operatiu per defecte que fem servir a l'empresa. Aquesta part, la vaig fer amb el meu propi ordinador portàtil, el qual hi vaig instal·lar el Visual Studio, una aplicació per programar en molts llenguatges.



Imatge 3.2 – Programa Visual Studio

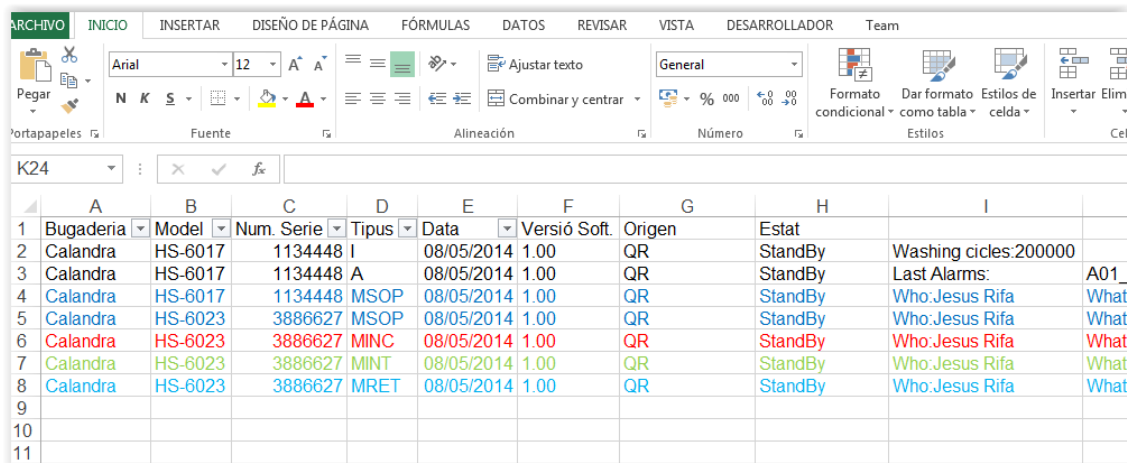
Bàsicament vaig trobar tres APIs, la mail, la email i la Gmail API, que em permetien fer la connexió amb el servidor de correu electrònic (en el meu cas gmail) per saber si hi havia algun correu nou, descarregar-lo i posar la informació amb un excel. La API utilitzada en el meu cas, ha estat la Mail, una API que et permet connectar al servidor (no importa de quin servidor es tracta) i obtenir els correus nous. Aquesta API, la vaig trobar directament a les extensions del Visual Studio, cosa que va anar molt bé perquè ja t'ho instal·la ell directament i no t'has de preocupar per res més.

A causa de que la API només era per C# o NET i com que no havia utilitzat mai cap d'aquests 2, vaig pensar que el millor seria fer servir el C#, ja que s'assembla més al C++ que ja havia utilitzat.

D'aquesta manera, només em falta posar els correus amb un fitxer de text per després poder-los portar a l'Excel.

Per poder fer la gestió ho vaig fer a través de l'Excel, un programa d'Office, el qual via macros, ens permet posar els correus descarregats anteriorment, fer filtres, ordenar (per si ens interessa tenir els correus ordenats per model de

màquina, bugaderia, etc.) i poder fer estadístiques, per si volem saber quin % de màquines han tingut una certa alarma.



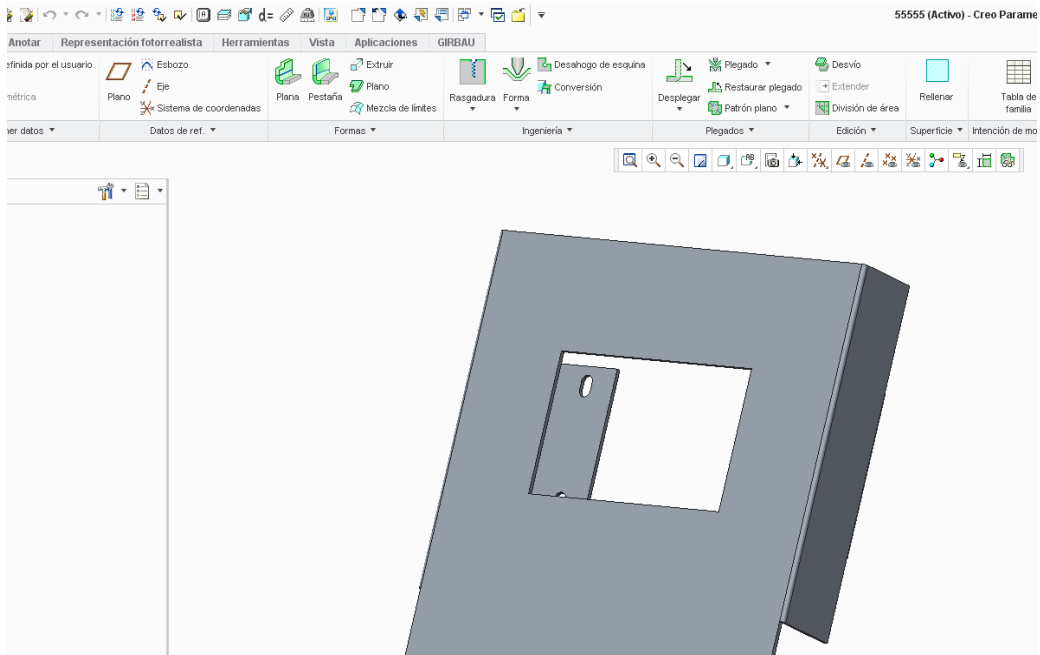
	A	B	C	D	E	F	G	H	I
1	Bugaderia	Model	Num. Serie	Tipus	Data	Versió Soft.	Origen	Estat	
2	Calandra	HS-6017	1134448	I	08/05/2014	1.00	QR	StandBy	Washing cycles:200000
3	Calandra	HS-6017	1134448	A	08/05/2014	1.00	QR	StandBy	Last Alarms: A01_
4	Calandra	HS-6017	1134448	MSOP	08/05/2014	1.00	QR	StandBy	Who:Jesus Rifa What
5	Calandra	HS-6023	3886627	MSOP	08/05/2014	1.00	QR	StandBy	Who:Jesus Rifa What
6	Calandra	HS-6023	3886627	MINC	08/05/2014	1.00	QR	StandBy	Who:Jesus Rifa What
7	Calandra	HS-6023	3886627	MINT	08/05/2014	1.00	QR	StandBy	Who:Jesus Rifa What
8	Calandra	HS-6023	3886627	MRET	08/05/2014	1.00	QR	StandBy	Who:Jesus Rifa What
9									
10									
11									

Imatge 3.3 – Programa Excel

Ja per últim, cal mencionar el Creo2 i el Photoshop CS5.

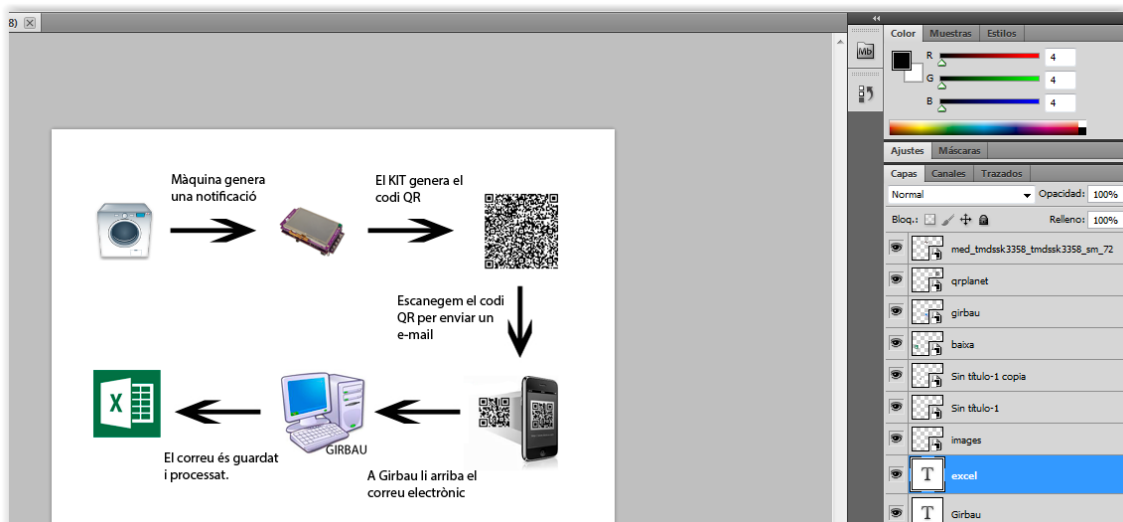
El Creo és un programa de dibuix en 3D molt potent. Aquest, el fem servir a l'empresa per tal de dibuixar les màquines i fer el plànols de les peces. Per tal de dibuixar el suport del KIT, vaig utilitzar aquest programa.

En aquest cas, només vaig haver de fer 2 plànols, un el de tallar la peça de xapa i doblar-la i l'altre perquè pintin la peça i deixar-la acabada.



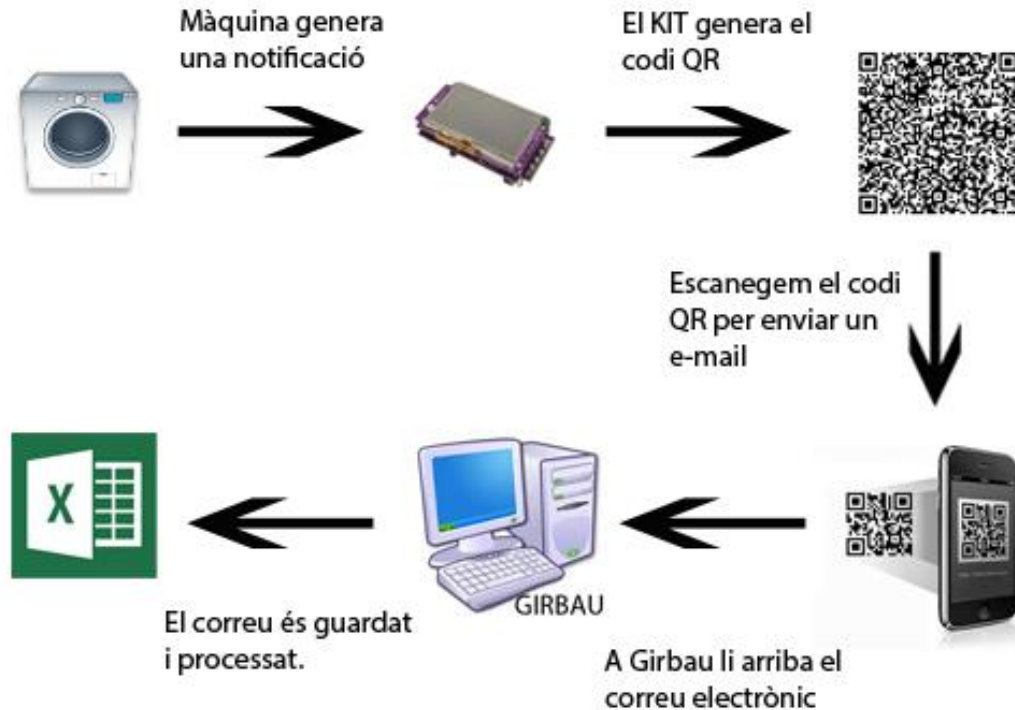
Imatge 3.4 – Programa Creo 2

El Photoshop CS5, és un programa per poder editar fotos. Aquest l’he utilitzat per modificar, retocar i canviar de mida les imatges les quals van a la pantalla del KIT. També l’he fet servir alhora de fer la memòria per tal de fer esquemes del procés.



Imatge 3.5 – Programa Photoshop CS5

CAPÍTOL 4: Esquema general.



Imatge 4.1 – Esquema del projecte

Tal i com es pot veure en el dibuix, el procés el qual segueix és el següent:

1. La màquina genera una notificació la qual és processada pel software (el KIT).
2. A partir d'aquí, el KIT genera un codi QR.
3. L'usuari de la bugaderia o el SAT (servei d'assistència tècnica) escaneja el codi QR i envia el correu electrònic.
4. Aquest correu, arriba a Girbau, el qual mitjançant un programa, es col·loca amb un fitxer d'Excel.
5. En l'Excel, es processa el correu i queda guardat per tal de tenir historial del que ha succeït a les diferents màquines.

CAPÍTOL 5: Esquema concret.

Abans ja hem explicat alt per alt el procés pel qual passa fins a arribar a Girbau i estar guardat en un Excel. Ara, entrarem en més detall tècnic de com funciona.

Quan, per exemple a la màquina de rentar, es genera una alarma, l'usuari té l'opció de poder generar un codi QR. Aquest, arriba a Girbau i al SAT (servei d'assistència tècnic) per tal de ser avisats que, la màquina de model - per exemple HS-6017 – i el número de sèrie, 1111111, ha tingut aquella, o aquelles alarmes. Una altra manera de generar els codis QR, és, per exemple quan no hi ha una alarma, però l'usuari de la bugaderia vol avisar el servei d'assistència tècnic perquè s'han de canviar els coixinets de la màquina. Després l'usuari fa un codi QR de incidència, el qual amb un petit comentari, per exemple, posaria que a la màquina es necessita canviar els coixinets. Un cop el SAT els hagués canviat, per tal de tancar aquesta incidència, faria un altre codi QR el qual també, amb un petit comentari, per exemple, posaria que ha canviat els coixinets.

Tots aquests tipus de codi QR (els expliquem més endavant) i més, s'han d'escanejar mitjançant un smartphone, el qual després envies el correu. Aquest correu, sempre arriba a una adreça de correu de Girbau, i en depèn de quin tipus, també arriba al SAT (hi arriba els tipus el qual ell després ha de fer una intervenció, per tant seria el tipus alarma i el tipus incidència).

Un cop a Girbau, mitjançant un programa, descarreguem (1 o 2 cops al dia) tots els correus rebuts, els quals els guardem amb un fitxer de text per seguidament ser importats a un Excel.

L'Excel, bàsicament et permet fer 2 coses. Una la fa automàtica que és, segons el tipus de correu que sigui, ho pinta d'un color o d'un altre, perquè sigui més fàcil de veure visualment. Per exemple, el tipus incidència, ho pinta de color vermell, i el d'intervenció de color verd. Un altre cas seria el tipus S.O.P (start of production) que vol dir que la màquina s'ha posat en funcionament per primer cop.

Aquest cas el pinta de color blau fort, i el seu contrari, la retirada, que és quan es dóna per acabat la vida útil de la màquina, que en aquest cas es pinta blau clar.

CAPÍTOL 6: Procés de construcció de la solució.

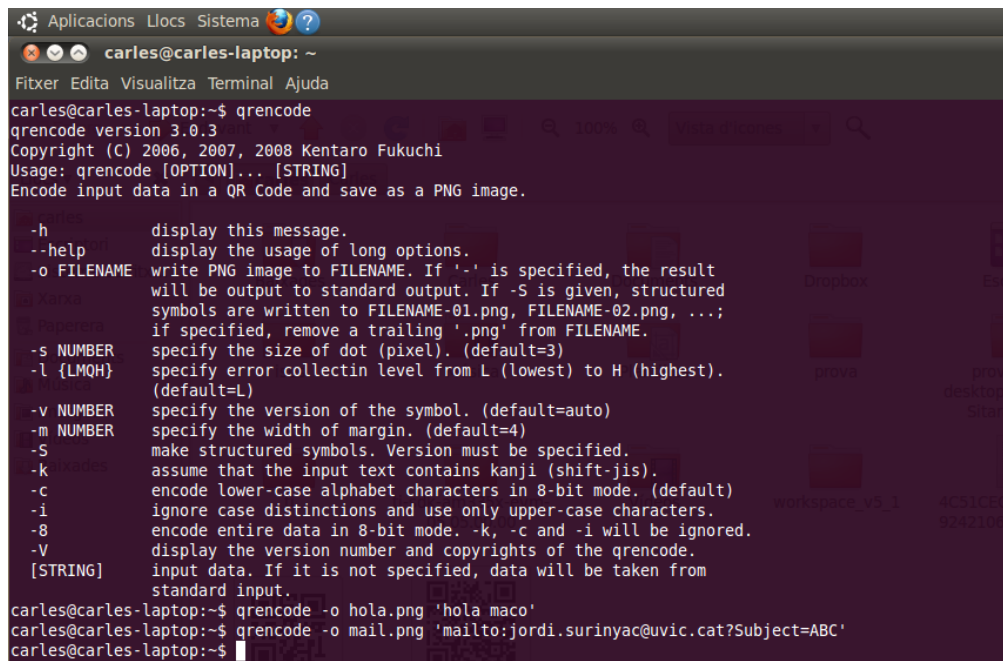
Per tal de desenvolupar el projecte, ho vaig estructurar bàsicament en 10 punts.

6.1 Buscar llibreria per programar els codis QR:

Al començament del projecte, el primer que vaig fer va ser buscar diverses llibreria per Linux (Ubuntu) les qual em permetessin programar codis QR. Dins les que vaig trobar, com ja he dit abans, vaig fer servir la qrencode, ja que era lliure i semblava la més ben documentada i que funcionava bé.

6.2 Proves amb la llibreria:

Per tal de comprovar i saber com funcionava la llibreria, em vaig dedicar a fer proves. Vaig veure al principi que t'instal·lava un software al PC el qual el podies cridar des del terminal. Posant directament "qrencode" t'obria el menú d'ajuda, on a partir d'aquí vam començar a fer proves.



```
carles@carles-laptop:~$ qrencode
qrencode version 3.0.3
Copyright (C) 2006, 2007, 2008 Kentaro Fukuchi
Usage: qrencode [OPTION]... [STRING]
Encode input data in a QR Code and save as a PNG image.

  -h          display this message.
 --help      display the usage of long options.
 -o FILENAME write PNG image to FILENAME. If '-' is specified, the result
            will be output to standard output. If -S is given, structured
            symbols are written to FILENAME-01.png, FILENAME-02.png, ...;
            if specified, remove a trailing '.png' from FILENAME.
 -s NUMBER   specify the size of dot (pixel). (default=3)
 -l {LMQH}   specify error collectin level from L (lowest) to H (highest).
            (default=L)
 -v NUMBER   specify the version of the symbol. (default=auto)
 -m NUMBER   specify the width of margin. (default=4)
 -S          make structured symbols. Version must be specified.
 -k          assume that the input text contains kanji (shift-jis).
 -c          encode lower-case alphabet characters in 8-bit mode. (default)
 -i          ignore case distinctions and use only upper-case characters.
 -8          encode entire data in 8-bit mode. -k, -c and -i will be ignored.
 -V          display the version number and copyrights of the qrencode.
 [STRING]   input data. If it is not specified, data will be taken from
            standard input.
carles@carles-laptop:~$ qrencode -o hola.png 'hola maco'
carles@carles-laptop:~$ qrencode -o mail.png 'mailto:jordi.surinyac@uvic.cat?Subject=ABC'
carles@carles-laptop:~$
```

Imatge 6.1 – Captura de pantalla del terminal d'Ubuntu amb el qrencode

Tal i com es pot veure a la captura de pantalla, escrivint `-o hola.png 'hola maco'` ens genera un codi QR amb el nom `hola`, amb format `.png` i que ens diu `hola maco`.



Imatge 6.2 – Codi QR

Després d'aquesta prova, vam mirar quin era el format per escriure correus electrònics, el qual vam trobar que era el següent:

```
mailto: "Direcció de correu electrònic."  
?subject: "Assumpte del correu."  
&body: "Cos del correu electrònic."
```

Amb aquest format vam fer el primer QR que enviava un correu electrònic. Al principi només hi vam posar subjecte, fent la següent codificació; `-o mail.png mailto:jordi.surinyac@uvic.cat?subject:'ABC'`.



Imatge 6.3 – Codi QR per
enviar un e-mail

Al començar a fer proves amb el qrencode, va sortir el primer problema, i és que programant des de el terminal, no es podia posar un punt i a part amb el cos del correu electrònic. Per tal de solucionar això, vaig veure que ha

partir d'un fitxer de text, també et podia generar el codi QR amb el que contenia aquest. Per tant, vaig optar per crear un simple fitxer de text, i que després, el qrencode ens generés el codi QR a partir d'aquest.

A partir d'aquí, i al veure que això funcionava, vam optar per fer una aplicació amb el QT Creator, que pogués executar un programa extern (en aquest cas seria el qrencode) per tal que des de el KIT AM335X SK ens generés el codi QR.

6.3 Proves amb el KIT AM335X SK:

Un cop vam tenir totes les proves fetes amb la llibreria qrencode, i abans de fer proves amb el KIT AM335X SK, vaig haver de fer algun programa per tal de saber com funcionava el QT Creator. Primer vaig fer un petit programa que quan s'accionés un botó, em surtis un missatge que digués: Hola món!. El codi font és el següent:

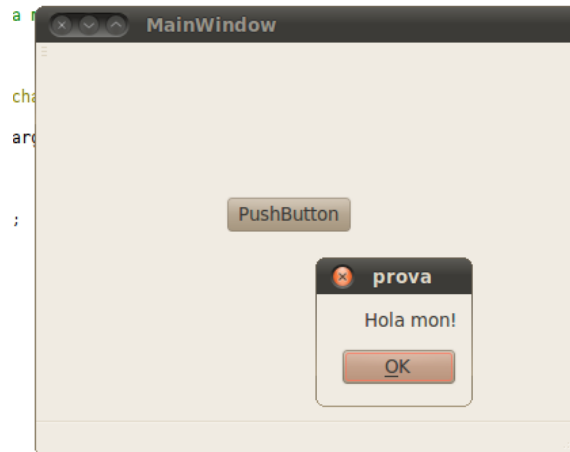
```
#include <QtGui/QApplication>
#include "mainwindow.h"
#include <QMessageBox>

void MainWindow::hola(){
    QMessageBox mb;
    mb.setText("Hola mon!");
    mb.exec();
}

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

    return a.exec();
}
```

Imatge 6.4 – Captura de pantalla de la primera aplicació amb el QT



Imatge 6.5 – Captura de pantalla de l'aplicació executada

Per tal de fer un programa que el KIT fos capaç d'executar, com que ja ho tenia tot instal·lat, només vaig haver de posar la versió del KIT corresponent, junt amb el seu cgg i gdb (el compilador i el depurador).

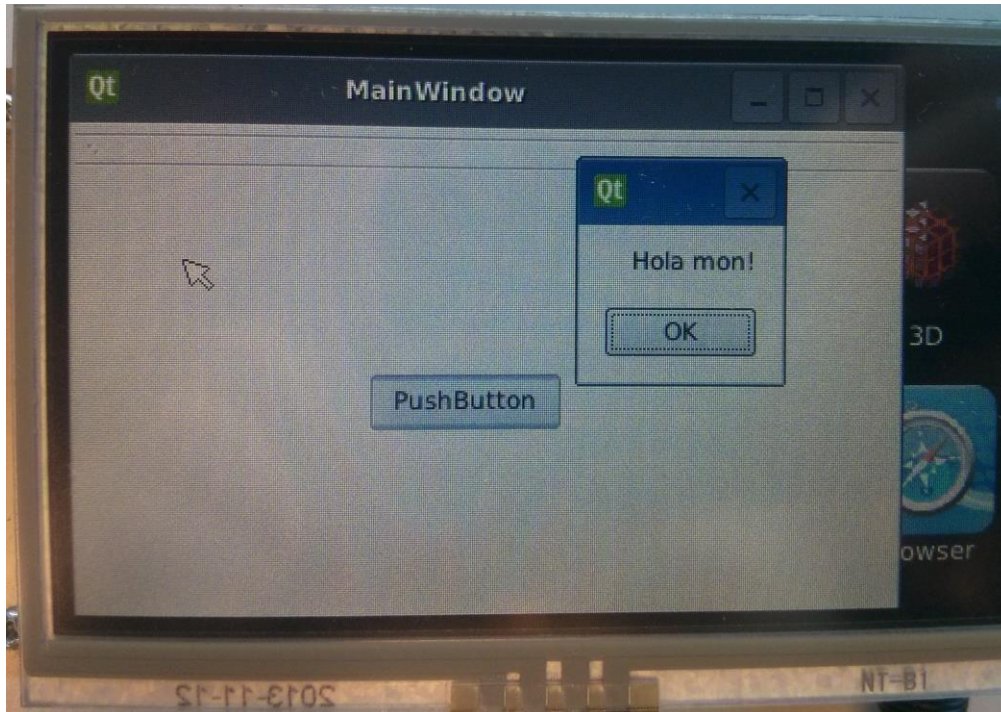
Amb tot això llest, l'únic que cal fer (i això s'ha de fer cada vegada que vols obrir el QT Creator –si vols programa pel KIT-) és obrir el terminal i posar les dues següents línies:

- **`source/home/carles/ti-sdk-am335x-evm-05.05.00.00/linux-devkit/environment-setup`**
- **`/home/carles/qtcreator-2.4.1/bin/qtcreator`**

Bàsicament el que fa és anar a la carpeta on hi ha guardat tot el software del KIT i ho executa per tal de que la versió que hem posat en el QT Creator sigui vàlida. La última línia, l'únic que fa és obrir el QT Creator.

Ara si que ja es podia programar i que els programes fets fossin executables pel KIT. Per tal de fer el primer programa, vaig pensar que seria millor començar per un programa fàcil i veure si realment funcionava, per tant vaig fer el mateix programa que havia fet prèviament per l'ordinador. Quan s'accionés un botó, que sortir un missatge que digués: Hola món!

El codi font és el mateix que abans.



Imatge 6.6 – Primera aplicació feta amb el KIT AM335X SK

Un cop comprovat que funciona, vaig anar a fer el programa el qual el KIT em generés el codi QR.

6.4 Generar QR al KIT:

Com ja he dit abans, en el punt 6.2, el que ens va semblar més fàcil de fer, va ser de fer un programa el qual cridés el qrencode per tal que ell ens generés el codi QR. Per tal de mirar si feia be la crida a un programa extern, el primer que vaig fer va ser agafar un programa que havia fet abans pel KIT AM335X (el del missatge que deia hola món!) i que s'havia segur que funcionava i executar-lo. El codi en C++ seria el següent:

```
1 #include <QtGui/QApplication>
2 #include "mainwindow.h"
3 #include <QProcess>
4
5
6 void MainWindow::obra(){
7
8     QString program = "home/carles/UBUNTU/prova";
9
10
11     QProcess *myProcess = new QProcess(this);
12     myProcess->start(program);
13 }
14
15 int main(int argc, char *argv[])
16 {
17     QApplication a(argc, argv);
18     MainWindow w;
19     w.show();
20
21     return a.exec();
22 }
```

Imatge 6.7 – Codi per executar un programa extern

Un cop comprovat que això funcionava, vaig anar a fer el mateix, però en comptes de cridar el meu programa, cridant el qrencode. Per mala sort, no va funcionar be. Al principi, vaig suposar que era per culpa de que el qrencode necessitava unes llibreria les qual no estaven al KIT i per culpa d'això no funcionava. Les vaig posar a on ens semblava que havien d'anar, perquè quan les busques el programa, les trobés, cosa que, al cap de varis intents d'anar-ho provant, i de posar les llibreria a diferents llocs no funcionava.

Després de fer diverses proves, vam arribar a la conclusió, de que el problema no eren les llibreries, si no que el qrencode, tot i està compilat per Linux, el KIT no el podia executar. Per tant, vaig haver de passar el pla B.

El pla B consistia en buscar el codi font del qrencode, i a partir d'aquest, programar el nostre programa per tal que generes ell el codi QR.

Al principi, per no haver d'afegir molts fitxers ".c" i ".h" en el projecte, vaig mirar d'anar agafant just el tros de codi que necessitava. Com que vaig veure que cada vegada necessitava més codi i es complicava més, vaig optar per posar tots els fitxers i d'aquesta manera no tenir problemes ni complicacions (una línia de futur del projecte, seria treure els fitxers i posar just el codi de programa necessari).



```

1  #include <QtGui/QApplication>
2  #include "mainwindow.h"
3  #include "qrencode.h"
4  #include "ui_mainwindow.h"
5  #include <QImage>
6  #include <QRect>
7  #include <QMessageBox>
8
9  void MainWindow::QR(){
10
11     const char* st;
12     st="mailto:carles.oro@uvic.cat\n?subject=ABC\n&body=jgnkjgn";
13     int x,y,w,l;
14     unsigned char *p;
15
16
17
18     QRcode *qrcode;
19
20     QSize s(200,200);
21
22     ui->label->setText(st);
23
24     qrcode=QRcode_encodeString(st, 7, QR_ECLEVEL_M, QR_MODE_8, 1);
25
26     p = qrcode->data;
27     w = qrcode->width;
28
29     QImage image(w, w, QImage::Format_RGB888);
30     image.fill(QColor(Qt::white).rgb());
31
32
33     for(y=0; y<w; y++) {
    
```

Imatge 6.8 – Primera prova del pla B. Es pot observar tots els fitxer ".c" i ".h"

Un cop comprovat que el programa funciona i que ens generava el codi QR, vaig anar a ajuntar el meu programa amb el de l'Eudald, un company de feina que també havia fet el seu projecte a Girbau i amb el KIT AM335X SK.

Ell havia fet el projecte del control de la rentadora, i, en una part d'aquest, hi ha la part de informació de la màquina, i on volem que l'usuari tingui l'opció de, si la màquina està connectada a internet, poder enviar un correu electrònic, i si no, que tingui l'opció de generar un codi QR que després mitjançant un smartphone, es pugui escanejar el QR i enviar el mateix correu electrònic amb l'smartphone.

6.5 Programa definitiu:

Per tal d'ajuntar la part feta per mi, i la feta per l'Eudald, vaig agafar el mateix format de pantalla que ja hi havia per tal de fer-ho igual.

Bàsicament, s'han de poder generar 4 tipus de codis QR, el de manteniment, el de informació, el d'alarmes i un que pugui obrir el manual de les rentadores mitjançant una pàgina web. S'ha de dir, que el de manteniment, no és un tipus, si no, que a dins d'aquest, hi ha un desplegable on tries quin tipus vols, entre el de S.O.P, incidència, intervenció i retirada.

Els tipus que codifiquem son els següents:

- Incidència: si la màquina no funciona del tot be i vols que vingui el SAT a arreglar-la. Aquest correu electrònic l'enviaria l'usuari de la bugaderia.
- Intervenció: serveix per tancar una incidència. Quan el SAT ha anat a la bugaderia en qüestió i ha arreglat la màquina.
- S.O.P: és el mateix que start of production. Quan el SAT posa per primer cop la màquina en funcionament a una bugaderia.
- Retirada: Fi de vida útil de la màquina. Quan la màquina deixa de funcionar i el SAT la desmunta.

- Informació: Per saber la informació de la màquina. Per exemple els cicles de rentat, quantitat de programes executats, etc. Avui en dia només hi tenim la informació de cicles de rentat.
- Alarma: Per saber les últimes alarmes que han sortit a la màquina. A la pantalla d'alarmes, expliquem les que hem definit que es puguin enviar.
- Manual: Per tal de poder obrir el manual de les rentadores. En cas de que es tingui dubte de com funciona alguna part de la màquina o algun control.

Les pantalles finals queden de la següent manera:

6.5.1 Pantalla de informació:

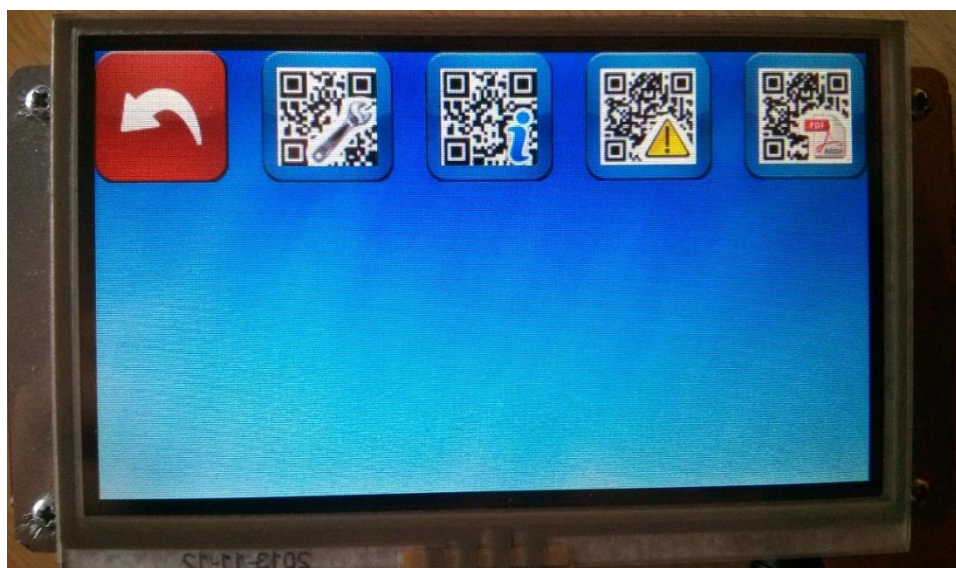
Et diu el model, número de sèrie, versió de software i en quin estat està la màquina. Com es pot veure als botons de dalt, hi ha el de generar el codi QR (que està sempre actiu) i el d'enviar un correu electrònic (aquest només està actiu en cas que la màquina tinguiés internet). Al clicar per generar codis QR surt a la pantalla per generar-los.



Imatge 6.9 – Pantalla d'informació del KIT

6.5.2 Pantalla de generar els codis QR:

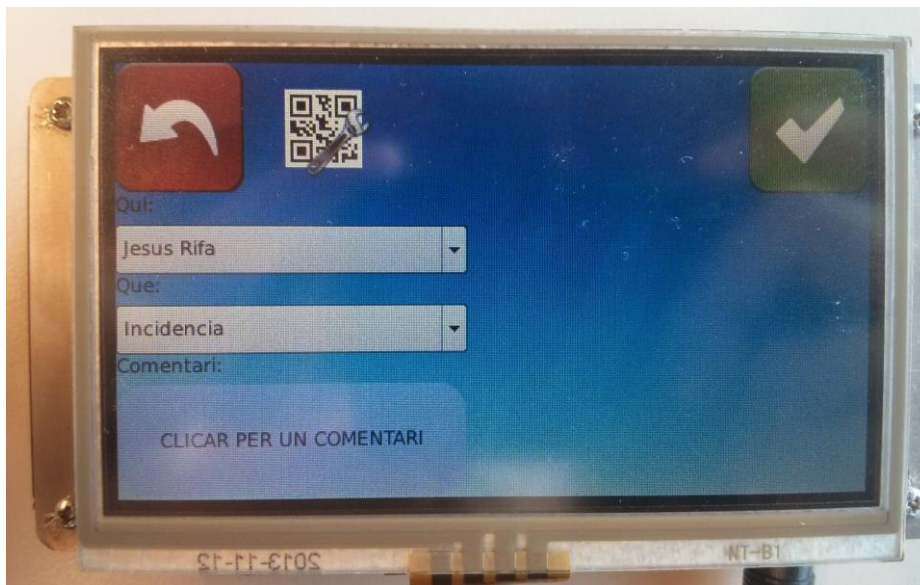
Tal i com es pot observar, hi ha els 4 codis QR a fer (manteniment, informació, alarma i manual) i òbviament el botó de tirar enrere. Si triem el cas de generar el QR de manteniment, anem a una altre pantalla, i en els altres casos, es genera directament el codi QR.



Imatge 6.10 – Pantalla principal de generar els codis QR

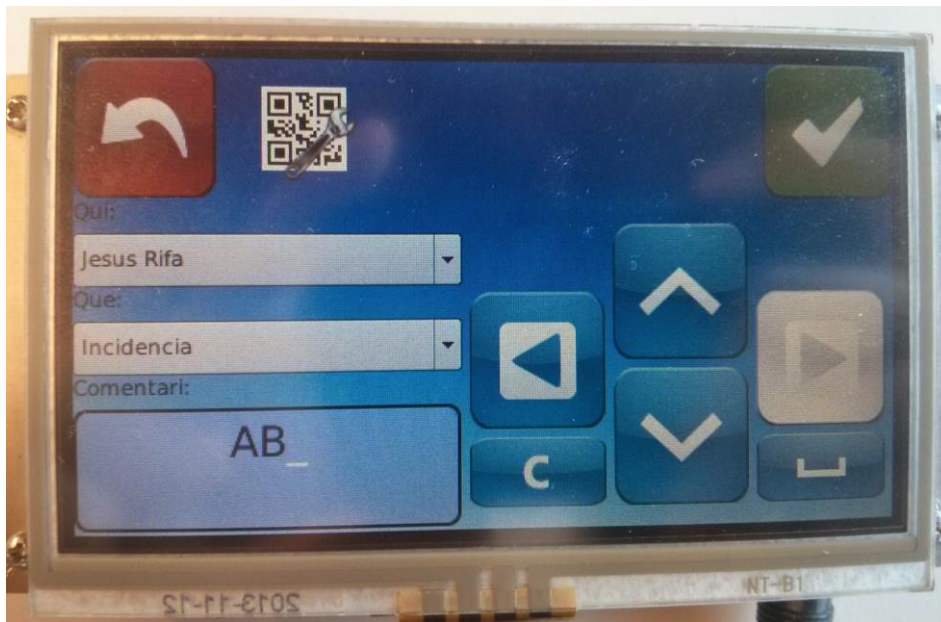
6.5.3 Pantalla de generar els codis QR de manteniment:

En aquesta pantalla, tenim dos desplegable. Amb el primer, hem de triar qui és el que generar el codi QR (en una línia de futur del projecte, quan el SAT posaria en funcionament una màquina per primer cop, aquí hi afegiria el seu nom, ja que el més probable és que sigui sempre ell el que vingui a arreglar la màquina i també hi posaria el nom de l'usuari més freqüent –o en cas de que hi hagi un responsable, el del responsable- que fa servir la màquina. Actualment, el noms venen fixats amb un desplegable que no es pot modificar). El següent desplegable, és el tipus de QR que vols enviar. Com ja hem dit abans, en aquest cas hi ha 4 tipus, el de incidència, el de intervenció, el de S.O.P i el de retirada.



Imatge 6.11 – Pantalla per generar el QR de manteniment.

Ja per anar acabant el QR de manteniment, en cas de que vulguis escriure un comentari, només cal que cliquis el botó de 'PRESS TO ENTER A COMMENT' i t'apareix els botons per començar a escriure. Només pots posar lletres i números i sempre amb majúscules.



Imatge 6.12 – Pantalla per escriure un comentari

Finalment cliques el botó d'acceptar (pots no haver entrat un comentari i també clicar-lo) i et genera el codi QR. Cal mencionar que després aquest botó queda inutilitzat i que l'únic que pots fer és tirar enrere i tornar a començar.



Imatge 6.13 – Pantalla amb el QR de manteniment ja generat.

6.5.4 Pantalla de generar els codis QR de informació:

Per tal d'enviar la informació, has de clicar el botó que conté la I. Després ja t'apareix el codi QR i a la part esquerra un petit text d'informació de com ho has de fer per enviar el codi QR. Aquest també serveix per assegurar de que realment envies el correu electrònic que vols, ja que depenen del tipus que sigui, et diu quin QR has generat.

Per exemple, tal i com podem veure a les altres pantalles, si generes un QR de tipus informació, el text et diu com envio la informació, si generes un QR d'alarma, et diu com envio l'alarma i així en totes.



Imatge 6.14 – Pantalla amb el QR de informació.

6.5.5 Pantalla de generar els codis QR d'alarma:

La part de les alarmes funciona igual que la de informació. L'únic que has de fer tu és accionar el botó on hi ha el triangle de perill i immediatament et surt el codi QR amb les últimes alarmes rebudes per la màquina.

Cal mencionar de que si t'etiboques alhora de generar un codi QR, no cal que surtis de la pantalla per genera un altre, ja que el programa ja s'encarrega de borrar de memòria l'anterior codi QR i et treu el nou.



Imatge 6.15 – Pantalla amb el QR d'alarmes.

6.5.6 Pantalla de generar els codis QR per obrir el manual:

Finalment, pel cas del manual, se't obrirà una pagina web. Aquesta web, que estarà en funcionament en breu i no forma part de la realització del meu projecte, hi haurà posades totes les màquines, on cada usuari tindrà un nom d'usuari i una contrasenya amb la qual hi podrà accedir. Allà triarà quina màquina, amb quin idioma, quin model i finalment quin manual vol obrir. Un cop amb tot això triat, se't obrirà el manual amb format PDF.

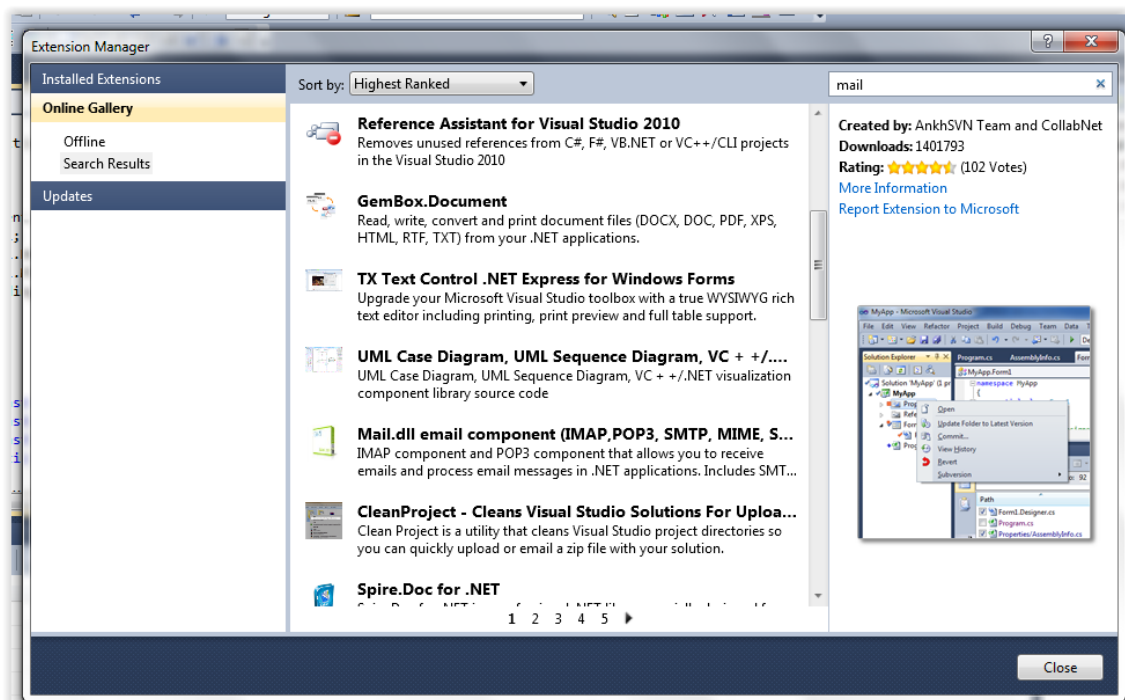


Imatge 6.16 – Pantalla amb el QR del manual.

6.6 Buscar la API per connectar al servidor Gmail:

Per tal de buscar la API per connectar-me al servidor, ho vaig fer de dues maneres diferents. Primer des de la web, la qual vaig trobar la *Gmail API* i després directament des de el Visual Studio. En aquest cas en vaig trobar dues, la *mail* i la *Email*. Finalment em vaig decantar per la *mail*, ja que es veia més ben documentada i amb més exemples i no només servia pels casos que tinguis un servidor gmail (com és el cas de la Gmail API).

A part d'això, una cosa que em va agradar va ser que era una que ja s'instal·lava en el programa.



Imatge 6.17 – Captura de pantalla del Visual Studio per buscar la API mail.

6.7 Proves amb la API:

Per tal de fer proves amb la API, primer de tot vaig mirar com s'havia de fer per connectar-se amb el servidor POP3. Era bastant immediat, i només calia connectar-se al servidor corresponent, en el meu cas és el "pop.gmail.com" i després posar l'adreça de correu electrònic i la contrasenya. Ja finalment, només calia mirar quins correus hi havia de nous i treure'ls per pantalla.

Cal recordar que està escrit en llenguatge C# i no C++.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using Limilabs.Client.POP3;
using Limilabs.Mail;
using Limilabs.Mail.Headers;
using Limilabs.Mail.MIME;
using System.Threading;

namespace Mail
{
    class Program
    {
        private const string servidor = "pop.gmail.com";
        private const string usuari = "tfg6023@gmail.com";
        private const string contrasenya = " ";

        static void Main()
        {
            using (Pop3 pop3 = new Pop3())
            {
                pop3.ConnectSSL(servidor);
                pop3.Login(usuari, contrasenya);

                List<string> uidList = pop3.GetAll(); //Obtenir tots els missatges
            }
        }
    }
}
```

Imatge 6.18 – Captura de pantalla del Visual Studio per connectar-se al servidor.

```

        foreach (string uid in uidList)
        {
            IMail email = new MailBuilder().CreateFromEml( // Descarregar cada e-mail.
                pop3.GetMessageByUID(uid));

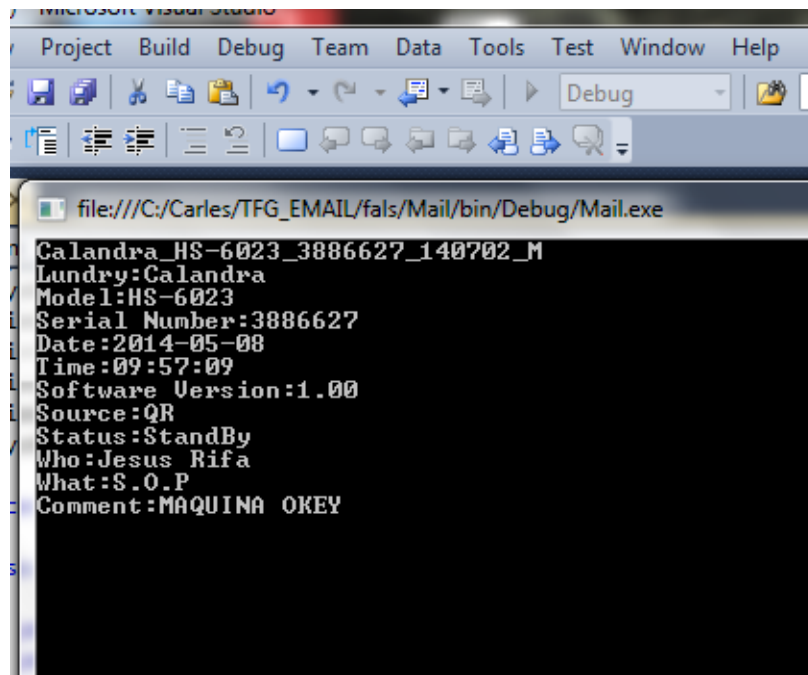
            saveMessage(email);

        }
        pop3.Close();
    }
}

private static void saveMessage(IMail email)
{
    Console.WriteLine(email.Subject);
    Console.WriteLine(email.Text);
}
}

```

Imatge 6.19 – Captura de pantalla del Visual Studio per obtenir els e-mails i treure'ls per pantalla.



Imatge 6.20 – Captura de pantalla d'un e-mail descarregat.

6.8 Programa definitiu:

Per tal de fer el programa definitiu, vaig utilitzar el que havia fet servir per fer les proves, i l'únic que vaig haver de canviar va ser que en comptes de que et treies els missatges per pantalla, me'ls guardés amb un fitxer de text. Després, un cop els tenia guardats, venia el processat dels correus per tal de que un cop exportats al excel, tinguin tots el mateix format.

Segons si hi ha més d'un correu, o bé, només un, hi ha un processat o un altre (en cas de que sigui el primer, s'ha de crear el fitxer de text, en canvi, si és el segon o endavant, s'ha de modificar el fitxer per posar els altres a continuació).

Bàsicament el que faig aquí és agafar, tant de l'assumpte com del cos del correu electrònic, i per aquest ordre, la bugaderia, el model, el número de sèrie, el tipus, la data, la versió de software, l'origen i l'estat (aquesta part seria la comuna per tots els tipus de correu) i després també agafo, depenen del tipus, els cicles de rentat (en cas de que sigui de informació), les alarmes (en cas de que sigui d'alarma) i qui ha enviat el correu electrònic, el que i el comentari (en cas que sigui de tipus manteniment).

D'aquesta manera, i separats cada un per un punt i coma, i entre els diferents correus, separats per un punt i a part (d'aquesta manera l'Excel ho tracta com un format csv), tens el format pel processat de l'Excel.

6.9 Implementar la macro d'Excel:

Per tal d'implementar la macro d'Excel, el primer que faig fer va ser una macro la qual em permetés importar un fitxer de text en format csv. El format csv és un fitxer de text, en el qual les dades que vols separar en diferents cel·les a l'Excel, les separen per un punt i coma. En cas de que a l'Excel vulguis tenir diferents files, el que has de fer és posar-ho en diferents línies del fitxer de text.

Quan vaig veure que això ja anava be, vaig anar a fer la macro, que mires quina és l'última posició ocupada de l'Excel, i que allà i importes el fitxer. Ja per acabar, vaig fer que segons el tipus que fos (informació, alarma, intervenció, incidència, retirada i S.O.P) pintes la fila d'un color o d'un altre. D'aquesta manera és molt més visual de veure quan hi ha una incidència i quan hi ha hagut la seva intervenció. També per quan es posa en funcionament per primer cop una màquina (S.O.P) i quan es fa la seva retirada.

```

Sub Macro1 ()
'
' Macro1 Macro
'
'
'
Dim x, r, ul, pos, ultima As Integer
x = 1

    ul = Cells(Rows.Count, 1).End(xlUp).Row 'A ul hi ha lultima posició ocupada
    If (ul = 1) Then           ' Si la ul és la posició 1, ho deixem a aquesta posició
        pos = ul
    Else                       ' En cas que no, hi hem de sumar 1
        pos = ul + 1
    End If

```

Imatge 6.21 – Captura de pantalla de la marco d'Excel per mirar l'última posició ocupada.

```

email_macro.xls - Mòdul01 (Código)
(General) Macro1

With ActiveSheet.QueryTables.Add(Connection:=_
"TEXT;C:\Carles\IFG_EMAIL\txt\02email.txt", Destination:=Range(("A") & pos)) 'Importem el txt
    .Name = "02email"
    .FieldNames = True
    .RowNumbers = False
    .FillAdjacentFormulas = False
    .PreserveFormatting = True
    .RefreshOnFileOpen = False
    .RefreshStyle = xlInsertDeleteCells
    .SavePassword = False
    .SaveData = True
    .AdjustColumnWidth = False
    .RefreshPeriod = 0
    .TextFilePromptOnRefresh = False
    .TextFilePlatform = 850
    .TextFileStartRow = 1
    .TextFileParseType = xlDelimited
    .TextFileTextQualifier = xlTextQualifierDoubleQuote
    .TextFileConsecutiveDelimiter = False
    .TextFileTabDelimiter = False
    .TextFileSemicolonDelimiter = True
    .TextFileCommaDelimiter = False
    .TextFileSpaceDelimiter = False
    .TextFileColumnDataTypes = Array(1, 1, 1, 1, 1, 1, 1, 1, 1)
    .Refresh BackgroundQuery:=False
End With
    
```

Imatge 6.22 – Captura de pantalla de la macro d’Excel per importar el correu electrònic.

```

ultima = Cells(Rows.Count, 1).End(xlUp).Row 'Tornem a mirar lultima posició ocupada

For x = 0 To ultima Step 1 ' Recorrem totes les files que hi ha, posant el color edient.
    Range("D" & x).Select
    If Range("D" & x) = "MSOP" Then ' Si és MSOP hi posem blau fort
        Range("A" & x & ":L" & x).Select

        With Selection.Font
            .Color = -4165632
            .TintAndShade = 0
        End With

    End If

    If Range("D" & x) = "MINC" Then ' Si és MINC hi posem vermell
        Range("A" & x & ":L" & x).Select

        With Selection.Font
            .Color = -16776961
            .TintAndShade = 0
        End With

    End If

    If Range("D" & x) = "MINT" Then 'Si és MINT hi posem verd
        Range("A" & x & ":L" & x).Select

        With Selection.Font
            .Color = -11480942
            .TintAndShade = 0
        End With

    End If
End For
    
```

Imatge 6.23 – Captura de pantalla de la macro d’Excel per posar els colors segons el tipus.

```
    If Range("D" & x) = "MRET" Then           ' Si és MRET hi posem blau fluix
        Range("A" & x & ":L" & x).Select

        With Selection.Font
            .Color = -1003520
            .TintAndShade = 0
        End With

    End If

    x = x + 1
Next r

End Sub
```

Imatge 6.24 – Captura de pantalla de la macro d'Excel per posar el color al tipus "Retirada"

6.10 Realitzar un suport pel KIT:

Ja finalment, i per deixar el projecte acabat, amb el programa Creo 2, que ja havia utilitzat molt durant les meves pràctiques a Girbau S.A, vaig realitzar un suport pel KIT.

Bàsicament, i per utilitzar el mateix sistema que fa servir l'empresa, vaig haver de fer 2 plànols, un per tallar la xapa i doblar-la, i l'altre per pintar-lo.

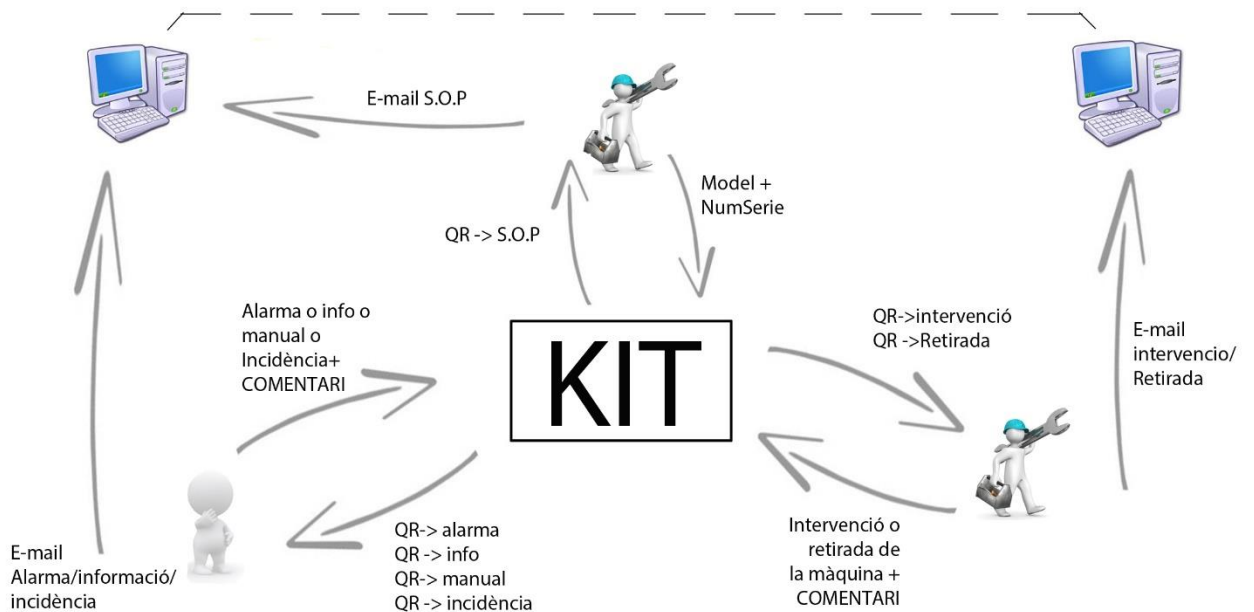
D'aquesta manera, per tal de presentar el projecte, es disposa d'un sistema de suport i molt més presentable.



Imatge 6.25 – Foto del suport del KIT AM335X SK.

CAPÍTOL 7: Manual d'utilització.

7.1 Procés



Imatge 7.1 – Imatge del procés que segueixen per generar i enviar els codis QR.

Tal i com es pot observar en la imatge, el primer que passaria, seria que el servei d'assistència tècnica (SAT), entraria a la màquina els valors del model i número de sèrie. Un cop això, generaria un codi QR de tipus S.O.P (start of production) el qual aquí Girbau, sabria que la màquina, per exemple, HS-6023, amb número de sèrie, per exemple 1111111, en la data que estiguem ha començat a entrar en funcionament a la bugaderia, per exemple Calandra.

A partir d'aquí, podrien passar dues coses, o que el responsable de la bugaderia generi un QR del tipus alarma, informació, manual o incidència (hi pot incloure un comentari) –que seria un dels casos més típics, ja que per exemple vol informar de que a la màquina s'hi ha de canviar els coixinets- el qual el SAT rebria un correu electrònic i aniria a la bugaderia en qüestió per tal de solucionar el problema.

En cas que el correu electrònic sigues el de informació, en aquest cas no hauria de fer res ja que aniria al servidor de Girbau i prou.

Un altre cas és quan el SAT ha reparat alguna cosa de la màquina, hauria de fer un QR tipus intervenció, el qual si vol, hi posaria un petit comentari on escriuria el que ha reparat –per exemple canviar coixinets màquina-.

Un altre cas, seria per tancar la vida útil de la màquina, i és que el SAT, abans de desballestar la màquina, generes un QR de retirada, el qual també hi posaria un comentari, per exemple; fi de vida útil de la màquina. D'aquesta manera, a Girbau sabríem que la màquina amb número de sèrie –per exemple; 1111111- ja no està en funcionament.

Això seria un exemple de com l'usuari de la bugaderia i el SAT, podrien estar en contacte mitjançant els correus electrònics i en cas de que s'hagués d'arreglar alguna màquina poder-hi anar el més ràpid possible. Per altre banda, també es manté a l'empresa Girbau S.A informada dels moviments que hi ha, quines son les averies més freqüents i també amb la rapidesa que treballa el servei d'assistència tècnic.

CAPÍTOL 8: Millores.

Per tal de millorar el projecte, el primer que hi hauria a fer, i que de fet, ja és una línia de futur a l'empresa Girbau S.A, és modificar el programa, per tal de que no hi hagi d'haver tants fitxers .h i .c al projecte. La idea seria agafar estrictament el necessari per tal de poder generar els codis QR, ja que actualment amb aquests fitxers que hi ha, contenen molt codi font que nosaltres no fem servir (per exemple hi ha per generar micro codis QR).

Una altre millora possible, seria de que a part de generar codis QR que permetin enviar correus electrònics, hi hagués la opció de generar codis QR que permetin enviar sms. D'aquesta manera, si hi hagués una bugaderia que estigués situada a un lloc on hi hagués cobertura però no internet, es pogués fer el mateix però enviant un sms. El petit inconvenient d'això és que has de pagar, en front els correus electrònics que són gratis si disposes de connexió a internet.

En aquest cas, s'hauria de refer el programa i que amb algun lloc, hi hagués l'opció de, o be generar un QR per enviar un correu electrònic, o bé per enviar un sms.

Altres millores que hi podria haver, serien amb el tractament dels correus electrònics ja aquí a Girbau. La primera seria buscar un sistema d'emmagatzematge dels correus que no fos l'Excel. Aquest, a no se que es trobés un programa semblant, s'hauria de fer a la casa i es tractaria d'un programa el qual bàsicament tindria dues parts. La pantalla estaria dividida en dos, a l'esquerra, hi hauria uns desplegable, els quals podries classificar per; bugaderia, model, número de sèrie i data. Alhora, et sortiria tota la informació del correu electrònic, ja sigui per exemple les alarmes, la intervenció, etc. A la part dreta de la pantalla, hi hauria gràfics, estadístiques, etc tot això depenen de, per exemple la bugaderia que hi hagués seleccionada, o bé el model, o fins i tot el número de sèrie.

Amb aquests gràfics i estadístiques, per exemple, es podria veure de manera molt visual, per exemple, quina és l'alarma que li ha sortit més a la màquina amb número de sèrie 1111111.

Una altre part a millorar, en el cas que volguéssim millorar el tractament de les dades en l'Excel, seria posar també gràfics i estadístiques. D'aquesta manera, tal i com he dit abans, ràpid i visualment pots veure la quantitat d'alarmes, i de quin tipus han sigut en un model de màquina concret.

CAPÍTOL 9: Conclusions.

El projecte ha complert les expectatives plantejades al principi d'ell. S'ha aconseguit poder generar codis QR en el sistema operatiu de Linux i en concret amb el KIT AM335X SK. Alhora, també s'ha realitzat un software per Windows, capaç de connectar-se al servidor de gmail i de obtenir el correus electrònics nous, generats per un codi QR i escanejats i enviats per un smartphone, i ja finalment, fer un tractament amb un full d'Excel per tenir tot l'historial de correus rebuts.

La realització d'aquest projecte, m'ha servit molt per tal de posar en pràctica el que he après durant els 4 anys a la carrera universitària. També m'ha servit per conèixer l'entorn de programació amb el QT Creator, un entorn molt potent i flexible pel que fa a la programació amb objectes i alhora tenint unes pròpies llibreries que et faciliten bastant la programació. Tot i ser un editor que no havia utilitzat mai, he comprovat que es bastant fàcil i que permet fer moltes coses.

Entrar en l'entorn de haver de programar per un KIT crec que també ha estat una bona experiència, ja que veus que el programa que fas no només té aplicació davant d'un ordinador, si no que pot anar amb màquines i és una solució molt més pràctica.

La part realitzada a Windows, que es la gestió dels correus electrònics, també m'ha servit de molt, ja que la programació, tant amb el llenguatge de C#, com amb l'editor de Visual Studio no ho havia utilitzat mai, i d'aquesta manera també ho he pogut tocar una mica. El Visual Studio, he vist que també era un editor molt potent, que ell mateix et permet buscar i descarregar diferents APIs per tal de facilitar-te la programació. També he pogut comprovar que el llenguatge de C# no és molt diferent al C++ que havia utilitzat sempre, i una cosa molt positiva, que sabent programar en C++ els altres llenguatges son bastant més fàcils.

La majoria del projecte s'ha realitzat mitjançant l'autoaprenentatge, tot hi que també hi ha hagut diverses reunions, tant amb el tutor de la UVIC com amb el tutor de Girbau, per tal d'acabar de definir alguns aspectes.

El fet de poder realitzar el projecte a una empresa, m'ha ajudat molt, tant per tenir més ajuda (econòmica i professional), com per treballar en equip (a causa de diferents reunions) i perquè el projecte tingui una aplicació pràctica real.

Ja per acabar, comentar que m'ha agradat molt realitzar el projecte i que m'ho he passat molt bé. A part d'això, també dir que he après molt i que estic molt satisfet amb els objectius aconseguits.

CAPÍTOL 10: Webgrafia.

Introducció als codis QR.

http://ca.wikipedia.org/wiki/Codi_QR

(Consulta el 24 de març 2014)

Imatges i documentació.

<http://www.google.cat/>

(Última consulta el 27 d'agost 2014)

Descarregar el S.O de Ubuntu.

<http://www.ubuntu.com/>

(Consulta el 25 d'abril 2014)

Descarregar el QT Creator i documentació varia.

<https://qt-project.org/search/tag/qt-creator>

(Última consulta el 30 de juliol 2014)

Descarregar el codi font del "qrencode" i documentació varia.

<http://fukuchi.org/works/qrencode/manual/>

(Última consulta el 30 de juliol 2014)

Documentació per configurar el KIT AM335X SK amb el QT.

<http://www.ti.com/>

(Consulta el 15 de maig 2014)

Descarregar i documentació del Visual Studio.

<http://msdn.microsoft.com/es-es/vstudio/aa718325.aspx>

(Consulta el 31 de juliol 2014)

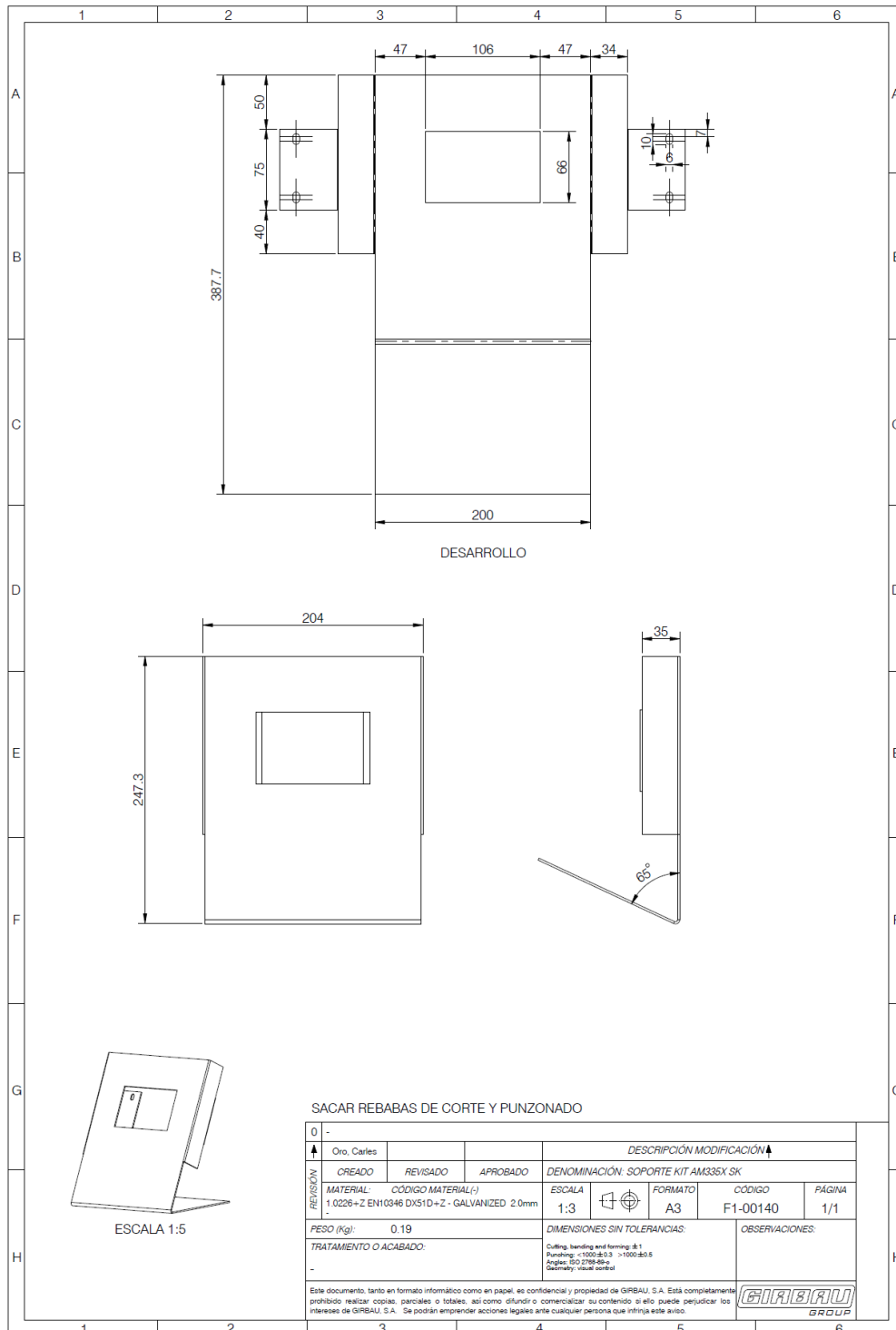
Documentació de la API "mail".

<http://www.limilabs.com/mail/help>

(Última consulta el 8 d'agost 2014)

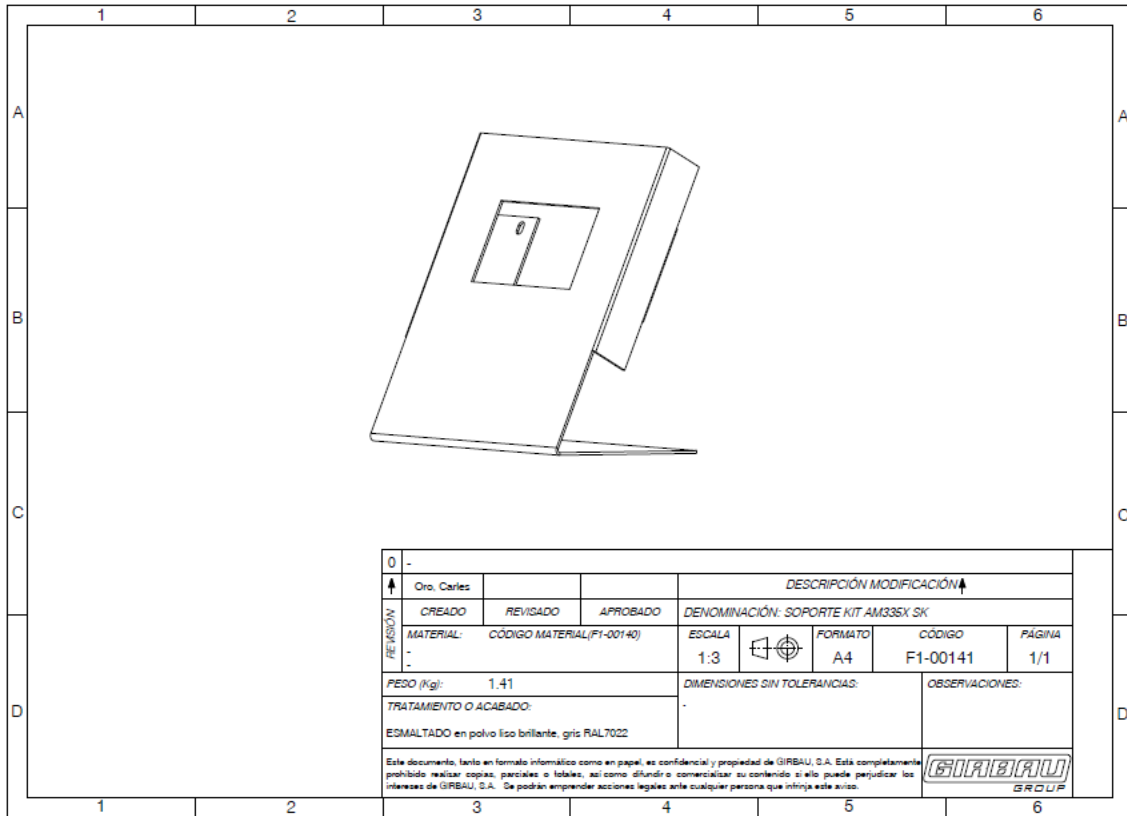
CAPÍTOL 11: ANNEXES

11.1 Plànol 1. Tallar i doblar la xapa.



Imatge 11.1 – Plànol de tall i doblat de la peça de xapa.

11.2 Plànol 2. Pintar el suport



Imatge 11.2 – Plànol de pintat del suport del KIT.

