

PROTOTIP D'UN SISTEMA *CASHLESS* SOBRE EXPENEDORS AUTOMÀTICS EXISTENTS

Treball de Final de Grau
Presentat a

Escola Tècnica d'Enginyeria de Telecomunicació de
Barcelona

Universitat Politècnica de Catalunya

per
Petar Hlad Colic

En requeriment dels
estudis del grau en

Ciències i Tecnologies de la Telecomunicació

Tutor: Marcel Fernández Muñoz

Barcelona, Setembre 2016

Resum

Aquest document presenta el disseny i el desenvolupament d'un prototip per a un sistema de pagament sense contacte en expenedors automàtics ja fabricats. L'expenedor serà controlat usant un ordinador de placa reduïda per controlar els circuits que engeguen motors de l'expenedor i comunicar-se amb el servidor central, i un lector NFC per poder llegir targetes dels usuaris.

El projecte es centra en el desenvolupament del codi de les aplicacions de servidor i de client, i en el desplegament de hardware que s'integrarà dins de l'expenedor automàtic.

Resumen

Este documento presenta el diseño y el desarrollo de un prototipo para un sistema de pago sin contacto en máquinas expendedoras ya fabricadas. La máquina expendedora será controlada usando un ordenador de placa reducida para controlar los circuitos que encienden los motores de la máquina expendedora y comunicarse con el servidor central, y un lector NFC para poder leer las targetas de los usuarios.

El proyecto se centra en el desarrollo de código de las aplicaciones de servidor y de cliente, y en el despliegue de hardware que se integrará dentro de la máquina expendedora.

Abstract

This document presents the design and development of a prototype for a cashless system on existing vending machines. The machine will be controlled using a small-sized single-board computer to control the circuits that power the motors of the vender and to communicate with the central server, and an NFC reader to read users' cards.

This projects focuses in the code development of the server and client applications, and also on the hardware deployment that will later integrate in the vending machine.

“Agraïments”
Autor Frase

AGRAÏMENTS

Voldria a en Marcel Fernández per creure en mi i acceptar la meva proposta de projecte.

Als companys que anaven a acompanyar-me en l'aventura de desenvolupar aquest projecte abans de decidir presentar-ho com a treball de fi de grau i en especial a l'equip vell de IT de l'associació per tota l'ajuda que m'han prestat.

I sobretot als meus pares, per tot el seu suport infalible les 24 hores del dia, cada dia.

ÍNDEX

Índex	v
List of Figures	vii
Índex de figures	vii
1 Introducció	1
1.1 Objectius del projecte	1
1.2 Planificació del temps	2
2 Requeriments	3
2.1 Requeriments generals de la solució	3
2.2 Requeriments específics	4
2.2.1 Aplicació de Servidor	4
2.2.2 Aplicació de client	4
2.2.3 Implementació del Hardware	5
3 Funcionament d'un expenedor automàtic de begudes fredes	6
3.1 Descripció de les parts	6
3.2 Cas d'ús	6
3.3 Circuit elèctric	7
4 Disseny de software	9
4.1 Aplicació de servidor	9
4.1.1 Model de dades	9
4.1.2 Controlador	10
4.1.3 Vista	11
4.2 Aplicació de client	11
4.2.1 Cas d'ús	12
5 Disseny de hardware	14
5.1 Disseny elèctric	14
5.2 Components	16
5.3 Unitat de prova	17

6	Resultats	19
6.1	Aplicació de servidor	19
6.2	Aplicació de client	20
6.3	Sistema de hardware	20
7	Pressupost	22
8	Conclusions i tasques futures	23
	Appendices	25
A	Descripció de les tasques	25

ÍNDIX DE FIGURES

1.1	Diagrama de Gantt del projecte	2
3.1	Esquema elèctric simplificat de l'expenedor	7
5.1	Esquema elèctric del circuit de sensing	15
5.2	Esquema elèctric proposat de l'expenedor	15
5.3	Diagrama de bloc de la unitat de test	16
5.4	Esquema elèctric del circuit de sensing de la unitat de test	17
5.5	Diagrama de bloc de la unitat de test	18
6.1	Captures de la interfície d'administració de l'aplicació de servidor . . .	19
6.2	Captures de la interfície d'usuari de l'aplicació de client	20
6.3	Imatges de la tapa de la unitat de test	21

CAPÍTOL 1

INTRODUCCIÓ

Els expenedors automàtics de begudes fredes tenen una vida útil bastant llarga però les compres amb efectiu s'estan quedant enrere.

Pels expenedors que originalment no inclouen sistemes de pagament sense monedes existeixen accessoris que se'ls pot afegir com acceptadors de bitllets o de targetes de crèdit.

En aquest projecte el que intentarem és crear una solució que sigui integrable en un expenedor existent que no inclou sistemes de pagament que no sigui amb monedes. La idea és útil per integrar en expenedors que es troben en l'edifici d'una empresa i es vol, per exemple, donar descomptes als seus empleats en funció del consum o de l'hora del dia.

Objectius del projecte

L'objectiu d'aquest projecte és dissenyar i desenvolupar un prototip per a un sistema *cashless* que es pugui integrar dins d'expenedors automàtics existents. El projecte es centrarà en un model específic d'expenedor automàtic de begudes fredes (Dixie-Narco DNCB 386) perquè és l'expenedor que hi ha disponible per al desenvolupament.

Els principals objectius es poden resumir en els següents punts:

Dissenyar i desenvolupar l'aplicació de servidor. El sistema de la solució serà un sistema centralitzat. Tot es gestionarà des d'un mateix lloc, i aquest serà l'aplicació de servidor.

Dissenyar i desenvolupar l'aplicació de client. Per tal de poder controlar l'expenedor automàtic i comunicar-se amb l'aplicació de servidor, és necessària l'aplicació de client.

Dissenyar i desenvolupar el sistema de hardware. Per tal de que l'aplicació de client pugui controlar l'expenedor, és necessari el sistema de hardware que farà d'interfície entre l'aplicació i els circuits de l'expenedor.

La feina feta durant aquests mesos s'ha centrat en desenvolupar el codi de les aplicacions i en desenvolupar el desplegament de hardware que s'integrarà a l'expenedor.

Planificació del temps

La planificació temporal que s’ha seguit durant el projecta es mostra a la Figura 1.1 Diagrama de Gantt.

La feina s’ha repartit en 6 blocs. La descripció detallada de cada bloc es troba a l’Apèndix A.

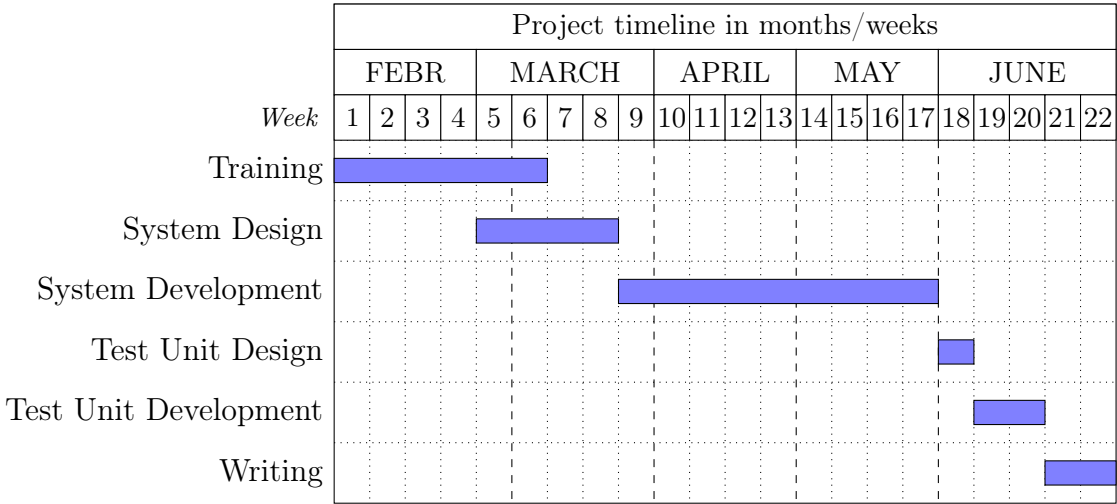


Figura 1.1: Diagrama de Gantt del projecte

CAPÍTOL 2

REQUERIMENTS

L'objectiu del projecte és dissenyar un sistema que es pugui integrar en expenedors automàtics existents i permeti la compra de begudes sense necessitat de fer ús de diners en efectiu en el moment de la compra.

Requeriments generals de la solució

Millors prestacions La solució no pot empitjorar les prestacions que tenia l'expenedor automàtic en l'estat original. Només n'ha de millorar.

Reversibilitat En cas de necessitat, s'ha de poder retrocedir en la configuració de l'expenedor automàtic i deixar-lo en l'estat original.

Seguretat El sistema ha de ser prou segur per evitar frau o robatoris de diners.

Per assolir aquests requeriments el projecte es dividirà en tres parts:

Aplicació de Servidor És el sistema central de la solució. Serà la part que gestionarà les compres, transaccions monetàries, usuaris.

Aplicació de Client És el sistema que controlarà directament l'expenedor. Quan l'usuari s'identifiqui i faci una selecció, l'aplicació es comunicarà amb el sistema central perquè autoritzi la compra i pugui fer que l'expenedor serveixi una beguda.

Implementació i integració del hardware És tota la part del projecte que tracta la part física del projecte. En altres paraules, la integració del dispositiu que contindrà l'aplicació de client, el/s dispositiu/s que servira/n per identificar l'usuari i tots els circuits elèctrics per controlar l'expenedor i el seu estat.

Un cop dividit el projecte en tres blocs, perquè la solució sigui prou segura contra frauds, es proposa que el sistema sigui un sistema de prepagament on el saldo de cada usuari l'emmagatzema i gestiona el sistema central. També es proposa que l'usuari s'identifiqui mitjançant una etiqueta NFC.

Requeriments específics

Aplicació de Servidor

Per tal de tenir una solució que compleixi els requeriments generals, es proposen una sèrie de requisits específics per a l'aplicació de servidor:

Gestió d'expedadors Ha de tenir control sobre els expedadors. Ha de gestionar quin producte tindrà l'expedidor a cada carril i a quin preu. Ha de poder deshabilitar la venda d'un cert carril. Ha de poder visualitzar l'estat en el que es troba un expedidor (s'ha quedat sense monedes per donar canvi, s'ha avariat, s'ha quedat sense producte ...).

Gestió d'usuaris Ha de poder crear nous usuaris, eliminar-ne, modificar-ne les dades. També ha de poder identificar un usuari a partir d'algun identificador amb què aquest s'identifica quan fa ús d'un expedidor.

Gestió del saldo disponible Ha de controlar el saldo disponible del que disposa cada usuari. L'ha de poder consultar, modificar-lo quan hi ha un pagament o una recàrrega de saldo.

Gestió de les compres Ha d'autoritzar la compra que fa un usuari tenint en compte el seu saldo disponible i la disponibilitat del producte a l'expedidor.

Seguretat en les transaccions El sistema ha de ser segur en termes de transaccions de diners i recàrregs de saldo per compra. S'han d'evitar suplantacions d'identitat, intrusions en el sistema, frauds i demés possibles accions que perjudiquin al propietari de l'expedidor o als usuaris.

Detecció i gestió d'incidències En sistemes que involucren parts mecàniques, elèctriques, i parts que es comuniquen per xarxa, no és tan estrany que en certs moments les coses no funcionin com haurien de funcionar. Així el sistema hauria de ser capaç de detectar la gran part d'incidències produïdes per errors de comunicació o fallades en l'expedidor i solucionar-les en la mesura que sigui possible.

Aplicació de client

De la mateixa manera que amb l'aplicació de servidor, es proposen un seguit de requeriments específics per l'aplicació de client:

Control sobre l'expedidor Ha de poder controlar els circuits de l'expedidor per exendre llaunes. També ha de poder analitzar els paràmetres dels circuits per a poder preveure la quantitat de producte restant que conté l'expedidor i a més detectar i evitar fallades.

Comunicació amb el sistema central S'ha de comunicar amb el sistema central per a recollir la informació necessària per funcionar i per a poder realitzar les vendes. També ha de comunicar al servidor la informació sobre l'estat de l'expedidor.

Interfície fàcil d'usar L'ús de l'aplicació de cara a l'usuari es farà a través d'una interfície gràfica d'usuari que ha de ser senzilla d'utilitzar de la mateixa manera que ho és un expedidor automàtic usual.

Seguretat en les comunicacions Les comunicacions han de tenir un mecanisme per evitar intents de frau i demés accions perjudicials per al propietari de l'expenedor i altres usuaris.

Implementació del Hardware

Tenint en compte els requisits generals de la solució i els requisits proposats per a l'aplicació de client, es proposen un seguit de requeriments específics per a la implementació del hardware dins l'expenedor:

Reversibilitat de la integració Un cop integrat el sistema, en cas de necessitat, s'ha de poder extreure el hardware i deixar l'expenedor en l'estat original. Això vol dir que no es pot alterar irreversiblement el cablejat de l'expenedor.

Volum reduït No pot ocupar molt espai. Ha de caber dins de la porta de l'expenedor.

Implementació modular Ha de ser prou fàcil de posar i treure de l'expenedor. Fàcil de reparar i reposar blocs defectuosos.

No interferir amb el funcionament mecànic de l'expenedor La implementació no ha d'interferir en el funcionament mecànic usual que té un expenedor com són l'obertura de l'expenedor per recollir les monedes o reposar producte.

Evitar creuament amb una compra amb monedes Es pot donar el cas que es vulgui efectuar una compra sense efectiu i s'hagin introduït monedes a l'expenedor al mateix temps. Ha d'evitar que es produeixin problemes per aquest creuament.

Aïllament físic amb l'exterior Ha d'estar físicament de l'exterior per protegir el sistema i evitar possibles intents de manipulació.

Interfície d'usuari Ha de tenir una interfície de manera que l'usuari pugui visualitzar la informació que dona l'aplicació de client i també alguna manera d'introduir la seva selecció.

Lector NFC Ha d'incloure un lector NFC per a la identificació dels usuaris.

CAPÍTOL 3

FUNCIONAMENT D'UN EXPENEDOR AUTOMÀTIC DE BEGUDES FREDES

Per a poder dissenyar la solució i poder entendre les millores, primer s'ha d'esbrinar i entendre el funcionament original d'un expenedor automàtic.

Descripció de les parts

La descripció de les parts es pot trobar al manual d'instruccions de l'expenedor a l'Apèndix ??.

Cas d'ús

Presentem el cas d'ús usual:

1. Un usuari s'acosta fins l'expenedor i observa quins productes estan disponibles i quin preu tenen. Cada botó de selecció té una etiqueta al costat que indica el producte i el seu preu. La disponibilitat del producte la indica un indicador lluminós sota el botó (si emet llum, no està disponible).
2. Un cop l'usuari ha fet la seva elecció, introdueix monedes amb un valor igual o superior a l'import del producte.
3. L'usuari apreta el botó de selecció del producte que ha escollit.
4. L'expenedor retorna el canvi (si n'hi ha) i serveix la beguda que l'usuari ha seleccionat.

Exposem els casos d'ús quan alguna cosa va com no hauria d'anar:

L'usuari apreta el botó d'un producte que no està disponible: No passa res. És com si no hagués apretat cap botó.

L'usuari apreta el botó d'un producte pel qual no té saldo suficient: No passa res. És com si no hagués apretat cap botó.

L'indicador de "import exacte" està encès: L'expenedor només accepta monedes del tipus que pot retornar i no retorna canvi.

Circuit elèctric

L'expenedor automàtic està dissenyat per funcionar amb un sistema electromecànic sense necessitat de circuits electrònics.

El mecanisme consistia en un petit interruptor mecànic amb una palanqueta allargada que quedava premuda quan s'introduïa un tipus de moneda concret i activava la resta del circuit que era controlat per un relé de tres commutadors, els botons de selecció i els interruptors que es troben on els carrils per controlar la posició dels motors.

Avui dia, amb expenedors que retornen el canvi, aquest relé és substituït per la màquina de canvi, que és l'aparell que classifica les monedes segons el seu valor, i després retorna canvi si cal. A més, dona la possibilitat de poder definir diversos preus.

Per tal de tenir una visió més clara del funcionament per tal de poder dissenyar el nostre sistema, presentem un circuit simplificat de l'expenedor juntament amb la màquina de canvi que s'ha extret del manual d'instruccions (llegint les explicacions i analitzant l'esquema elèctric complet).

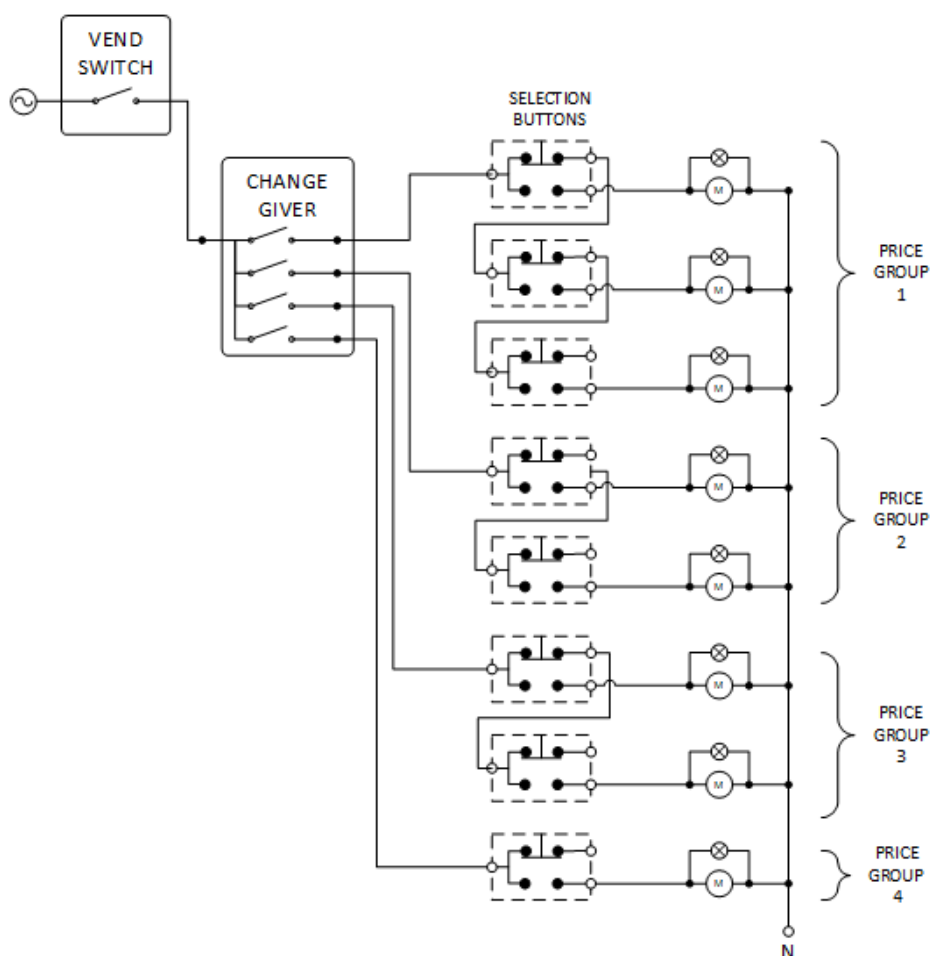


Figura 3.1: Esquema elèctric simplificat de l'expenedor

Quan s'introdueixen les monedes, la màquina de canvi dona corrent als botons pels quals s'ha introduït un import suficient. Un cop premut el botó de selecció, es tanca el circuit del motor i el motor s'acciona. Hi ha més factors a tenir en compte, però per a dissenyar el nostre sistema ja en tenim suficient.

Els botons de selecció estan connectats en serie d'aquesta manera en concret per evitar que el circuit el tanqui més d'un botó al mateix temps.

Com podem veure a l'esquema, abans de la màquina de canvi hi ha un interruptor. Aquest interruptor obre el circuit quan es deshabilita la venda en casos com quan no queda producte a cap carril de l'expenedor, o s'està servint una beguda o bé quan hi ha algun problema amb el mecanisme de l'expenedor i s'ha encallat.

Quan es deshabilita la venda, la màquina de canvi ja no accepta més monedes però l'import de les monedes que s'hagi introduït abans no es perd. Quan es torna a habilitar la venda es pot seguir comprant amb el saldo anterior. Si s'apreta la palanca de retorn mentre la venda està deshabilitada la màquina de canvi retorna els diners introduïts.

Malauradament, si l'expenedor es queda sense energia durant el procés de compra, es perd l'import introduït.

Les làmpades que hi ha en paral·lel amb els motots tenen dues funcions:

1. Si estan enceses durant el període entre que s'ha apretat el botó de selecció i s'ha servit la beguda, indica que el circuit s'ha tancat correctament i que el motor d'aquell carril està girant.
2. Si estan enceses durant de forma permanent indiquen que ja no queda producte en aquell carril.

CAPÍTOL 4

DISSENY DE SOFTWARE

En aquesta part es presenta el disseny de l'aplicació amb les decisions de disseny explicades.

Per al projecte sencer s'ha escollit programar en llenguatge *Python* perquè és molt senzill de programar i és molt més llegible.

Aplicació de servidor

Per a l'aplicació de servidor s'ha fet servir un *framework* per a desenvolupar webs en Python anomenat Django.

S'ha seguit el patró MVC (Model-Vista-Controlador).

Model de dades

Per a desenvolupar una aplicació web mínimament complexa, primer s'ha de començar dissenyant el model de dades.

Cada *model* en Django representa una taula a la base de dades.

User Cada usuari està associat a un **User** i contindrà les dades que l'identificaran com el seu nom, correu electrònic, nom d'usuari i contrassenya, DNI ...

Credit Conté el saldo disponible de l'usuari i, per tant, està associat a un **User**, i cada **User** té associat com a molt un **Credit**. Està en una taula separada perquè està pensat perquè l'aplicació pugui ser integrada en una base de dades d'usuaris existent.

Card Representa la targeta amb la què s'identificarà l'usuari. Per tant està associada a un usuari. Igual que les targetes de crèdit, un usuari pot tenir més d'una targeta associada.

Vender Representa un expenedor. Conté la informació bàsica de l'expenedor (model, número de sèrie, localització geogràfica), i també la informació sobre el seu estat.

ApiKeyToken Per a garantir un mínim de seguretat en les comunicacions, l'aplicació de client usará un codi per a identificar l'expenedor des del qual s'està comunicant. Aquest identificador serà únic per a cada expenedor. **Vender** i **ApiKeyToken** estan en taules separades per raons d'implementació de codi d'autenticació en Django.

Column Representa un carril d'un expenedor. Cada carril té un preu de venda i un tipus de producte. També pot estar actiu o inactiu, i també pot estar buit o ple.

Purchase Cada cop que l'usuari fa una selecció i tot va bé (la informació proporcionada per realitzar la compra és correcta i l'usuari té suficient saldo disponible) es crea una instància de **Purchase** que representa la compra que ha fet l'usuari amb la informació corresponent (targeta amb què s'ha comprat, producte, carril, preu, data i hora, identificador de la compra, i si ha estat pagada). L'identificador de la compra és un identificador aleatori que assigna l'aplicació de client per raons que comentarem més endavant. Tot i que **Column** ja conté la informació de carril, producte i preu, es guarda per separat ja que al llarg del temps aquests paràmetres podrien canviar.

Payment Quan un **Purchase** ha estat creat, s'ha de procedir al pagament de la compra, per tant un pagament està associat directament a una compra. **Payment** representa l'acció de pagar la compra. Inclou el preu de la compra, la quantitat descomptada (si es fa descompte) i el preu final a pagar. També inclou l'usuari a qui se li ha de cobrar, el dia i hora i la informació per determinar si el pagament està pagat i és vàlid, o si l'han hagut de reemborsar.

RefundPetition Quan un procés de compra es queda a mitges, o l'expenedor pateix una fallada tècnica, o per qualsevol altre motiu és necessari reemborsar l'import d'una compra a l'usuari, primer s'ha de crear una petició de reemborsament. És així perquè d'aquesta manera les peticions es poden generar de manera automàtica mentre que l'autorització de la petició es realitza amb més cura (un sistema més curós o bé amb supervisió humana). Conté l'identificador del pagament que es vol reemborsar, el sol·licitant (ja sigui un empleat o bé un expenedor de manera automàtica), la raó de la petició, l'hora i la data, i la informació per determinar si ha estat denegada o finalment s'ha realitzat el reemborsament.

Refund Representa el reemborsament d'un pagament un cop la petició és acceptada. Així doncs conté l'entitat que ha autoritzat el reemborsament, la petició que ha estat autoritzada i l'hora i la data de l'autorització.

Controlador

El controlador és on es troba tota la lògica en si del sistema.

En Django, quan es desenvolupa una aplicació de servidor que respon a peticions, la funció de controlador ho desenvolupen les classes que de l'API.

Les següents són les classes que fan de control·lador i cadascuna respon a un tipus diferent de petició:

GetUserDataFromCard Retorna la informació de l'usuari (nom i saldo disponible) a partir de l'identificador de la seva targeta.

GetVenderColumns Retorna la informació de cada carril de l'expenedor que l'ha demanat (tots els paràmetres que conté el model **Column**).

ProductPurchase Quan l'usuari ha fet la selecció del producte, es comprova que el producte estigui disponible, que el producte del carril seleccionat coincideixi amb el producte que conté, que el preu també coincideixi i que el saldo disponible de l'usuari sigui suficient. Si tot és correcte, es crea una instància de **Purchase** i el seu **Payment** associat i se li descompta l'import de la compra al saldo de l'usuari. Finalment se li retorna una resposta de que tot ha anat bé a l'expenedor perquè serveixi una beguda o bé se li retorna una resposta d'error informant de què és el que ha anat malament.

RequestCancelPayment Quan alguna cosa ha anat malament a la banda de l'expenedor (s'ha encallat el mecanisme, ha fallat la xarxa, ...) pot ser que s'hagi cobrat la compra a l'usuari i no s'hagi servit la beguda. És per això que cal un mecanisme per demanar reemborsament sota certes circumstàncies. Quan es rep una sol·licitud de reemborsament aquesta s'emmagatzema si fins el moment no existia cap altra sol·licitud associada al mateix pagament.

Vista

La vista en si és la informació que es pot visualitzar de la resposta d'una petició. En la nostra aplicació hi ha dos possibilitats d'origen de petició: des de l'aplicació de client, i des de la interfície d'administració.

Django té per defecte implementada una interfície d'administració per gestionar manualment les dades de l'aplicació. Amb això ens estalviem haver de desenvolupar aplicacions d'interfície gràfica d'usuari per tal de poder gestionar nous usuaris, incidències, etcètera.

Aleshores, la interfície d'administració és un tipus de vista, i la informació que retorna el controlador a l'aplicació de client és l'altre tipus de vista de l'aplicació.

Aplicació de client

Per a l'aplicació de client, com ha de tenir una interfície gràfica d'usuari, s'ha decidit utilitzar **PyQt4** que és una llibreria gràfica. Té integrada la part gràfica i també la comunicació entre fils d'execució.

L'aplicació de client es divideix en 6 grans blocs diferenciables:

comms S'encarrega d'implementar la comunicació amb el servidor.

card_reader S'encarrega de controlar el lector de targetes. Cada cop que detecta una targeta avisa al fil d'execució principal juntament amb l'identificador de la targeta.

column_controller S'encarrega de controlar els mecanismes dels carrils de l'expenedor i de detectar les possibles fallades mecàniques.

db_controller S'encarrega de controlar la base de dades local. En aquesta base de dades s'emmagatzema la informació dels carrils i també la informació de compres fallides per les quals el servidor encara no ha confirmat la creació d'una petició de reemborsament.

graphic_ui : S'encarrega de crear la interfície gràfica d'usuari i de gestionar totes les accions de l'usuari i derivar-les cap el fil d'execució principal.

cocacolero : S'encarrega d'instanciar, controlar i coordinar la resta de blocs. Aquest bloc es troba en el fil d'execució principal

Cas d'ús

A continuació s'exposa l'explicació del cas d'ús usual de l'aplicació per part de l'usuari.

1. L'usuari arriba i apropa la seva targeta al lector.
2. L'usuari és correctament identificat i el seu nom i el seu saldo disponible apareixen per pantalla. Els botons de selecció dels productes disponibles queden habilitats.
3. L'usuari fa la selecció pitjant amb el dit sobre el botó del producte que ha escollit
4. L'usuari disposa de suficient saldo disponible i la compra s'efectua amb èxit. L'expenedor serveix la beguda, es deshabiliten els botons i finalitza el procés.

Exposem el mateix cas d'ús des del punt de vista de l'aplicació.

1. **card_reader** detecta una targeta i comunica l'identificador a **cocacolero**.
2. **cocacolero** demana la informació de l'usuari associat a la targeta detectada a l'aplicació del servidor. El servidor l'identifica amb èxit i retorna la informació demanada (nom i saldo). **cocacolero** habilita els botons de **graphic_ui** dels productes disponibles.
3. Quan l'usuari fa la selecció, **graphic_ui** comunica a **cocacolero** la selecció.
4. **cocacolero** genera un identificador aleatori per a la compra, envia al servidor la petició de la compra. El servidor comprova que la informació és correcta i que l'usuari té suficient saldo. El servidor respon amb èxit. **cocacolero** comunica a **column_controller** el carril que s'ha d'activar i aquest li respon amb èxit quan la beguda s'hagi servit. **cocacolero** deshabilita els botons de **graphic_ui** i finalitza el procés.

Ara exposem els casos d'ús quan alguna cosa no funciona com està previst:

L'usuari no vol realitzar cap compra després d'identificar-se: L'usuari pitja el botó Log Out i finalitza el procés.

L'usuari no s'ha identificat amb èxit: S'ignora

Es detecta una targeta durant un procés de compra: S'ignora

L'usuari fa una selecció per la qual no disposa de saldo suficient: el servidor ho detecta i ho comunica a l'aplicació de client. L'aplicació descarta la compra i finalitza el procés.

La informació de la compra és incorrecta: El mateix cas que el de saldo insuficient.

El servidor no pot processar la compra per qualsevol altre motiu: El mateix cas que el de saldo insuficient.

El servidor no respon dins del termini establert: S'anul·la el procés. S'emmagatzema la informació de la compra i s'envien peticions al servidor periodicament fins que el servidor crea la petició de reemborsament amb èxit o la descarta per qualsevol motiu.

L'aplicació detecta una fallada tècnica i no s'ha servit el producte: El mateix cas que quan es perd la resposta del servidor.

CAPÍTOL 5

DISSENY DE HARDWARE

En aquest capítol comentarem tot el disseny del hardware que s'ha seguit per poder controlar l'expenedor automàtic assolint els requisits definits anteriorment.

Disseny elèctric

A partir de l'esquema elèctric simplificat de l'expenedor, la idea del nostre disseny elèctric és d'afegir un circuit paral·lel a la màquina de canvi i als botons de selecció per poder accionar el motor del carril desitjat i que pugui deshabilitar la màquina de canvi durant el transcurs d'una venda en el nostre sistema.

Per aconseguir deshabilitar la màquina de canvi, afegim un relé commutador abans d'aquesta i així també dirigir el corrent cap els relés que accionaran els motors.

Els relés es connecten en serie de la mateixa manera que els botons de selecció per evitar que més d'un d'ells pugui tancat el circuit al mateix temes.

Per tal de poder controlar l'estat de l'expenedor per saber si no queda producte en algun carril o si la venda ha quedat deshabilitada per algun motiu, s'ha de poder verificar si hi ha tensió en certs punts del circuit.

Per verificar que el carril no està buit i hi queda producte per vendre, es pot observar si la làmpada esta encesa mesurant la tensió que cau entre els seus borns. Això ho farem amb un circuit que hem trobat en un blog de projectes casolans i hem adaptat al nostre sistema:

Un rectificador d'alta impedància connectat a un opto-acoblador per separar els circuits de l'expenedor dels circuits del nostre hardware.

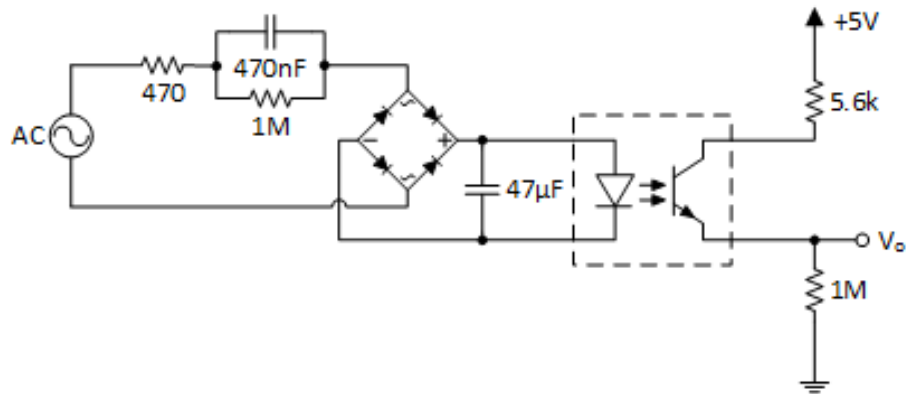


Figura 5.1: Esquema elèctric del circuit de sensing

Per verificar que la venda no ha quedat deshabilitada, mesurarem la tensió al node comú del relé de by-pass amb el mateix circuit anterior.

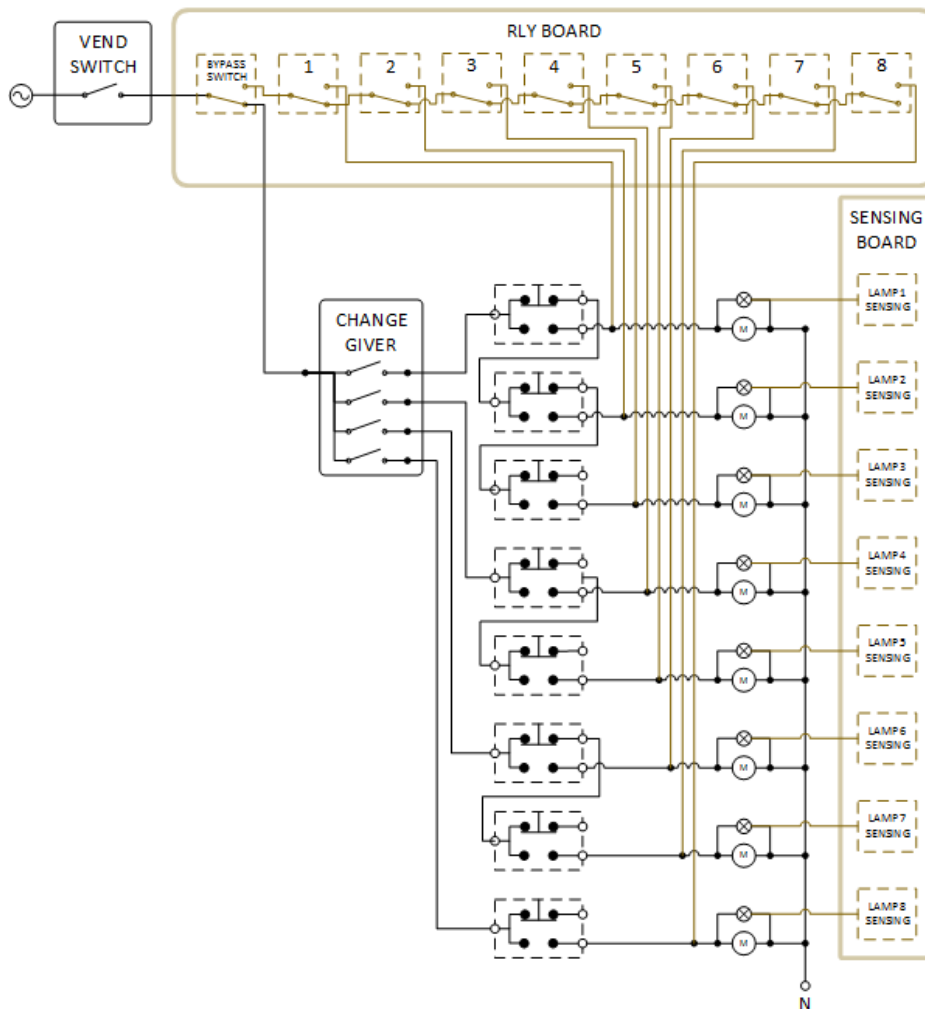


Figura 5.2: Esquema elèctric proposat de l'expenedor

Tot i que pel funcionament del nostre sistema no és gaire rellevant, també mesurarem la tensió a la làmpada de *canvi exacte* per poder notificar al servidor que

l'exenedor gairebé s'ha quedat sense canvi.

Components

Com a controlador central de l'exenedor es farà servir un ordinador de placa petita *RaspberryPi 2 B* (a partir d'ara, RPi), ja que disposa de bastants pins d'entrada/sortida, se li pot instal·lar un sistema operatiu basat en linux (que fa molt més fàcil el disseny de l'aplicació de client), té connexió a internet, possibilitat de connectar-hi una pantalla per HDMI o bé una pantalla tàctil.

Per a la interfície d'usuari usarem la pantalla tàctil *Raspberry Pi Touch 7"*.

Per a poder accionar els circuits que accionen els motors de l'exenedor des de la RPi farem servir una placa de relés activats per nivell baix.

RPi necessitarà una alimentació de 5V i 1A. La pantallà tàctil necessita una alimentació de 5V i 1A. La placa de relés necessita una alimentació de 12V.

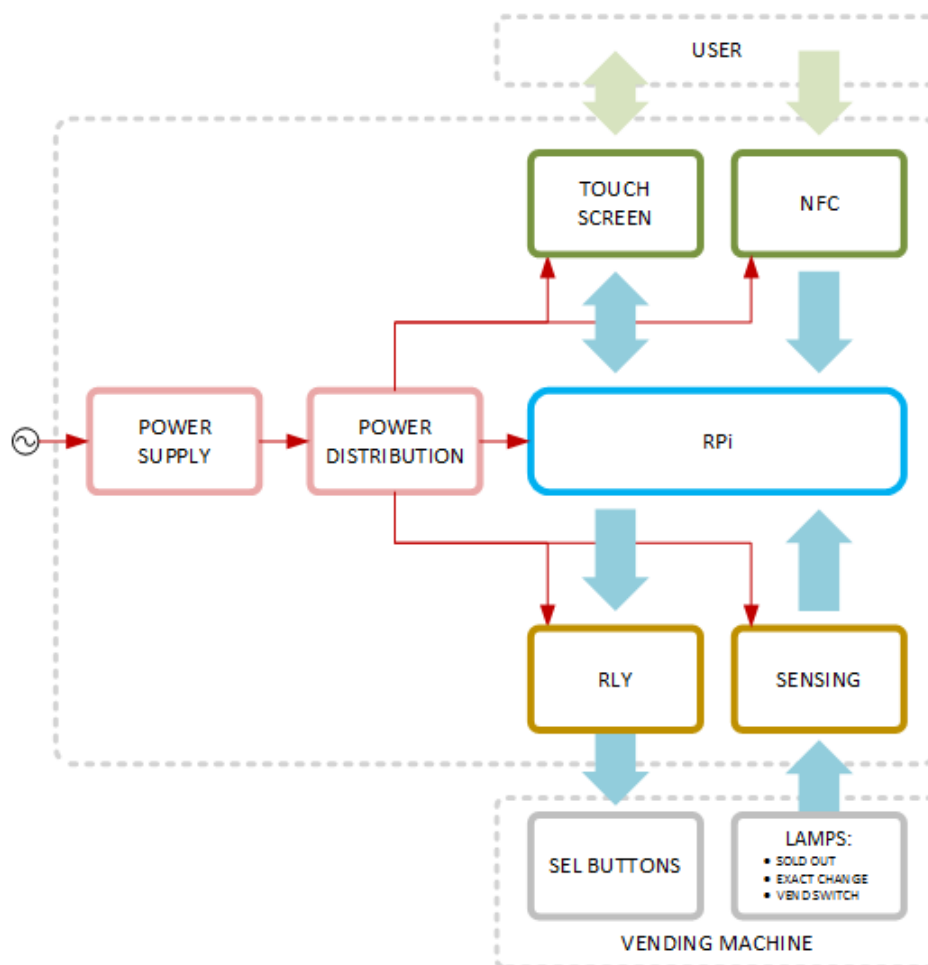


Figura 5.3: Diagrama de bloc de la unitat de test

Unitat de prova

Per tal de poder fer proves amb el sistema prèvies a la integració final, s'ha desenvolupat una unitat de test que consisteix en una capsa de metacrilat que conté tots els components fixats a dins i interconnectats de manera que sigui fàcil de transportar.

A més, s'ha desenvolupat una placa de sensing a part per a la unitat de test ja que el sensing sobre alta tensió no té gaire sentit fer-lo si no és sobre l'expenedor.

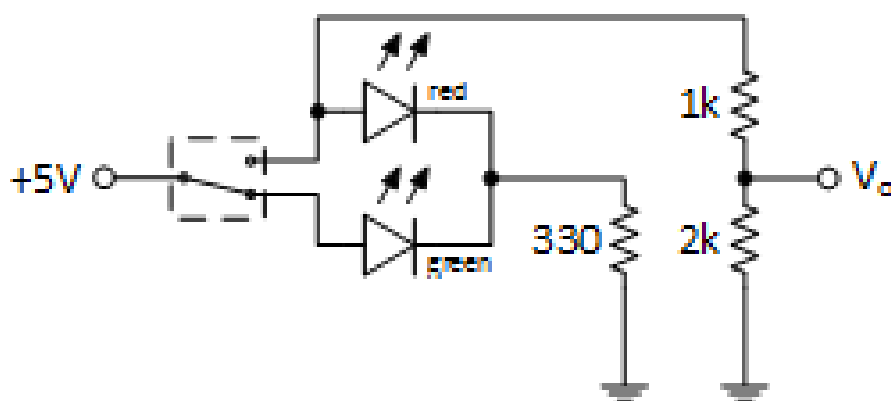


Figura 5.4: Esquema elèctric del circuit de sensing de la unitat de test

Aquesta placa té uns pins d'entrada on hi van connectats les sortides dels relés i quan s'activa un relé s'encén un LED per poder visualitzar des de fora de la capsa quins relés estan activats.

També té uns polsadors interruptors que són per a simular l'estat de les làmpades dels botons de selecció, que estan connectats als pins de sortida de la placa que aniran connectats als pins d'entrada de la RPi.

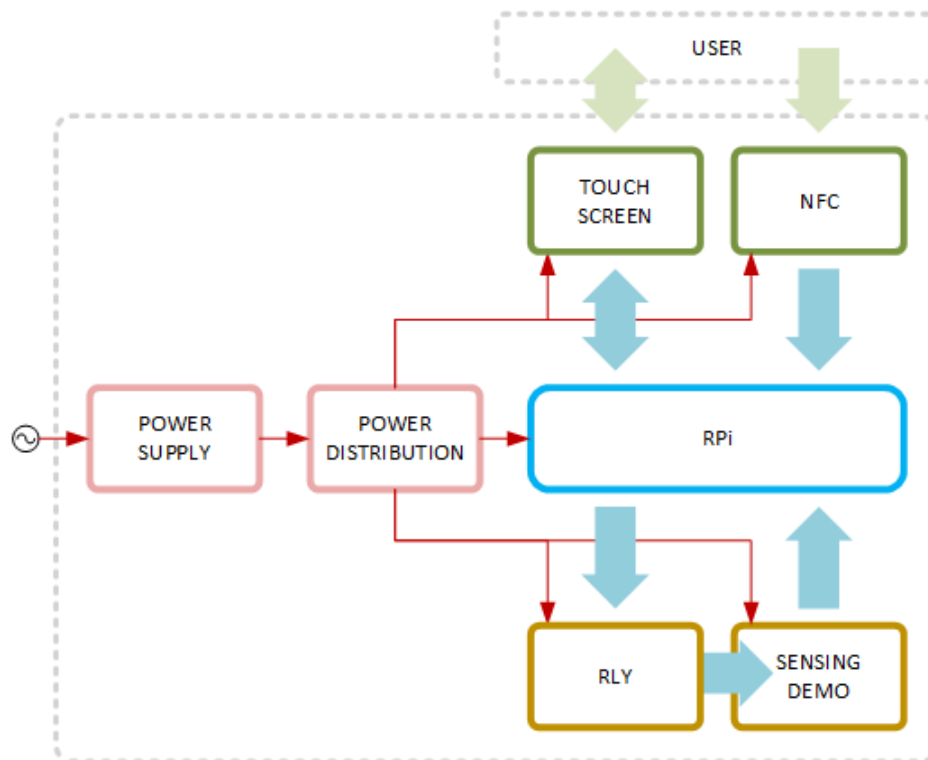


Figura 5.5: Diagrama de bloc de la unitat de test

CAPÍTOL 6

RESULTATS

En aquest capítol exposem els resultats del disseny i el desenvolupament del sistema sencer.

Aplicació de servidor

L'aplicació de servidor és la que més s'ha trigat en dissenyar i desenvolupar perquè el disseny de l'aplicació de client depen directament de la de servidor.

L'aplicació funciona com s'esperava.

A través de la interfície d'administració es poden fer fàcilment les gestions necessàries com són: afegir usuaris nous, afegir expenedors nous, gestionar els carrils dels expenedors ..., tot el que es requeria de poder gestionar en remot.

A la interfície d'administració només hi poden accedir els usuaris que tinguin permís d'administració, i això es pot gestionar des de la mateixa interfície.

The screenshot shows the 'Add user' form in the Django administration interface. The form includes fields for Username, Password, Password confirmation, User profile (User profile: #1, Birth date, National id number), and Credits (Credit: #1, Amount, Active checkbox). The URL bar shows 'https://localhost:8000/admin/auth/user/'.

(a) Interfície d'administració per crear usuaris

The screenshot shows the 'Change vender' form in the Django administration interface. The form includes fields for Name, Vender model, Location, and a checkbox for 'Need change'. Below the form is a table with columns: CONTENT, CAPACITY, PRICE, NEED REFILL, ACTIVE, and DELETE? The table lists various vending machines with their details. The URL bar shows 'https://localhost:8000/admin/vendingServer/v'.

CONTENT	CAPACITY	PRICE	NEED REFILL	ACTIVE	DELETE?
Dispositiu - 0. Random	70	0.34	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Dispositiu - 1. Coca-cola	70	0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Dispositiu - 2. Estrella Damm	80	0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Dispositiu - 3. Random	70	0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Dispositiu - 4. Guarana	30	0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Dispositiu - 5. Nestle	30	0.6	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Dispositiu - 6. Fanta Orange	30	0.5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Dispositiu - 7. Fanta Lemon	30	0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

(b) Interfície d'administració per gestionar els expenedors

Figura 6.1: Captures de la interfície d'administració de l'aplicació de servidor

Aplicació de client

L'aplicació de client compleix els requeriments que s'han especificat.

No obstant, té un problema amb el lector de targetes: si l'aplicació es tanca de manera inesperada (per exemple, degut a un error d'execució inesperat), es perd la comunicació amb el lector, i no es pot recuperar fins que no se li fa un reset de l'alimentació al lector. Això és un problema de la llibreria que utilitzem per controlar-lo (nfcpy), ja que amb altres llibreries que s'han provat això no passava.

La interfície gràfica d'usuari no ha estat dissenyada per un dissenyador



(a) Interfície d'administració per crear usuaris

(b) Interfície d'administració per gestionar els expenedors

Figura 6.2: Captures de la interfície d'usuari de l'aplicació de client

Sistema de hardware

El hardware en conjunt està desenvolupat. No obstant, per temes de calendari de l'associació (durant el mes de març l'expenedor està inaccessible, durant l'agost i part de setembre el despatx està tancat), no s'han pogut fer totes les proves que es volia fer.

Durant el quadrimestre s'han fet petites proves de concepte bàsiques amb èxit:

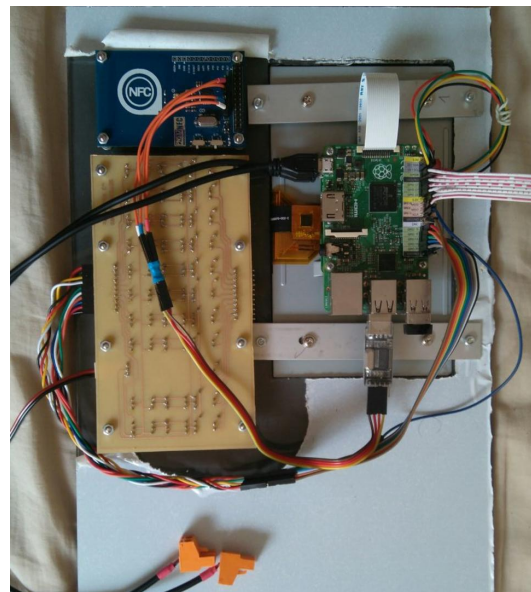
1. Provar d'expendre una llauna amb un relé activant-lo manualment.
2. Provar d'expendre una llauna amb els relés activats per l'aplicació de client.

Més enllà d'això no s'ha pogut integrar el sistema dins de l'expenedor per problemes de calendari i falta de temps.

La unitat de prova ha estat desenvolupada amb èxit i és molt útil per transportar el sistema d'una manera còmode.



(a) Front de la tapa



(b) Revers de la tapa

Figura 6.3: Tapa de la unitat de test

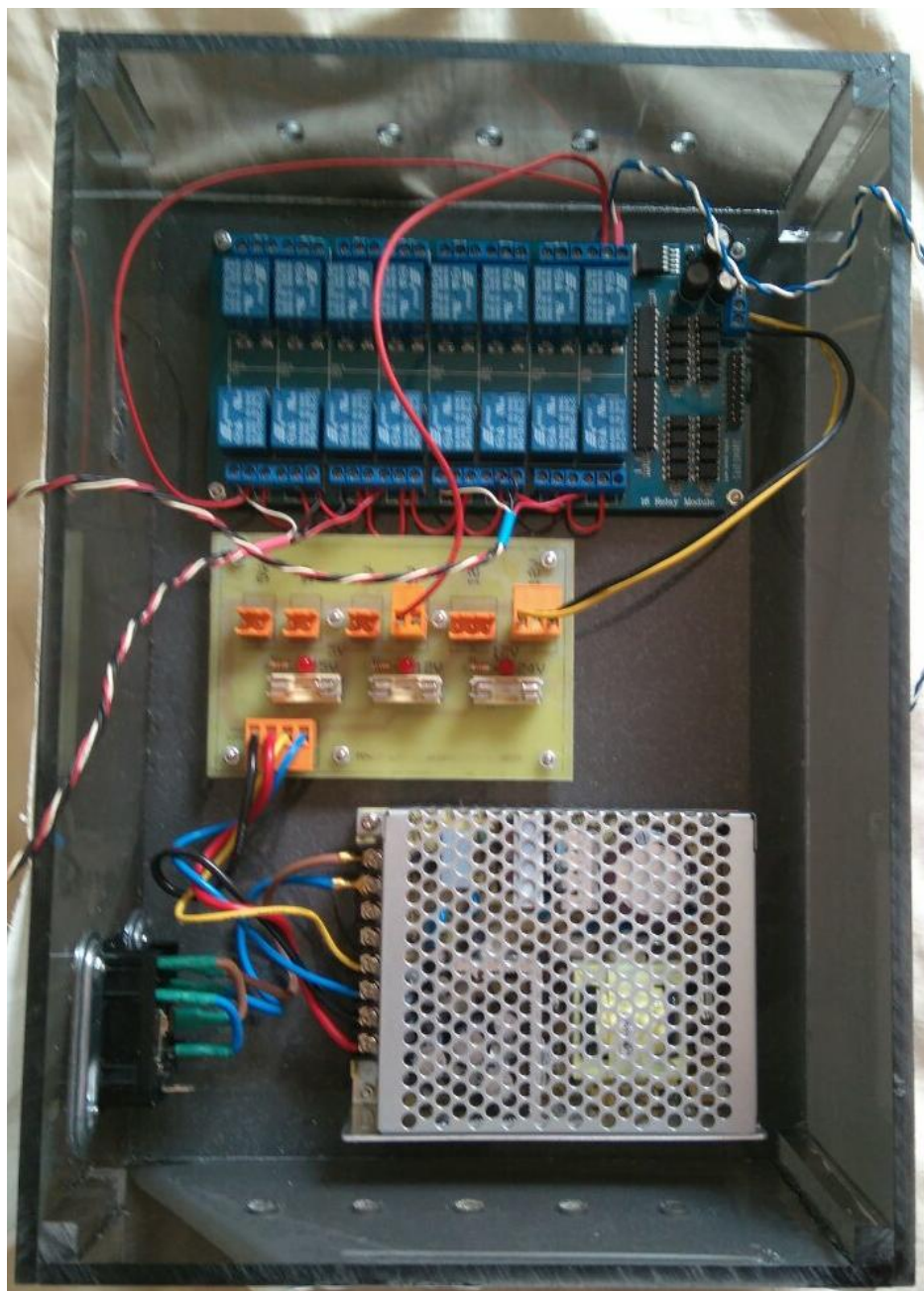


Figura 6.4: Interior de la unitat de test

CAPÍTOL 7

PRESSUPOST

This TFG project has been developed jointly with the GPI team. The budget presented is divided in Activities. Each Activity corresponds to an specific task, and the division has been done according to the equipment and staff needed to develop each tasks. In this budget are considered not only the activities developed in this project scope, but also the activities that are required to perform this project development.

The Activities that had been taken into account are:

Activity 1: Recording I. Captures done using cones as a pattern.

Activity 2: Recording II. Captures done using a chessboard as a pattern.

Activity 3: Undistortion. Test the different undistortion algorithms.

Activity 4: Registration, warping and blending. This Activity includes all the algorithms and methods tested in this TFG project scope and are directly related to the Arcol project.

Activity 5: Documentation. Additional activity that includes all the tasks done outside the Arcol project scope -TFG writing, additional tests, etc.-.

Each Activity summarized cost is the following:

Activity	Equipment	Workers	Subtotal
1 Cone recording	197.06 €	290.00 €	487.06 €
2 Chessboard recording	205.83 €	290.00 €	495.83 €
3 Undistortion	46.70 €	1,030.00 €	1,076.70 €
4 Registration. warping and blending	442.00 €	21,000.00 €	21,442.00 €
5 Documentation	66.00 €	3,800.00 €	3,866.00 €
TOTAL			27.367,59€

Each activity detailed costs can be found in Appendix ???. This Appendix also contains each equipment detailed amortization and each worker salary.

CAPÍTOL 8

CONCLUSIONS I TASQUES FUTURES

This project has described the procedure followed to obtain an specific image stitching algorithm. This algorithm had to merge the image from four cameras located around a bus in a single 360° view image. The work described in this project has been done in a multidisciplinary UPC team working in a project commissioned by the Arcol company.

First of all, all the requirements and specifications inherited from the Arcol project have been stated. In basis on these requirements, a research in the current stitching methods has been done, including both commercial systems and high-level algorithms.

The next part included in this project have shown the used methods in the steps inside this TFG scope. From the four different stitching steps –undistortion, registration, warping and blending–, only warping have been fully developed in this project. Table 8.1 shows each step internal tasks and its final status.

<i>Stitching step</i>	<i>Internal task</i>	<i>Status</i>
Undistortion	Search and test undistortion algorithms	Done by another team member
	Apply the final algorithm	Done by the project author
Registering	Automatic registering	Not done
	Manual registering	Done by the project author
Warping	Search and test warping algorithms	Done by the project author
	Apply the final algorithm	Done by the project author
	Apply provisional algorithm	Done by the project author
Blending	Search and test blending algorithm	Not done
	Apply the final algorithm	Not done
Unghosting	Segment above ground objects	Not done
	Define an algorithm	Not done
	Apply the algorithm to the stitching	Not done

Taula 8.1: Internal tasks status

Finally, the last part of this project shows the results obtained in the warping process. It can be extracted from the results that the developed algorithm beats the state-of-the-art algorithms for this specific application. As it can be seen in Chapter ??, the *Homography merging* algorithm developed in this project obtains better results than simple homography in this specific application.

Regarding to the initial goals of this project, the results obtained in each point can be summed up as follows:

Representative points automatic detection. The automatic key points detection has been left outside this project scope, and has been left to a future development.

Estimate the warping parameters. The warping parameters and warping algorithm had been estimated successfully.

Blending the results on the final stitching. Blending algorithms have been tested and the feathering selected offers an acceptable performance for the project.



Follow the requirements stated by the Arcol project. Regarding the Arcol specifications, this project is fully adapted to be used with the specific hardware and software. Moreover, the land lines discontinuity has been fully fulfilled. Exposure compensation and avoiding host images has been left to the Arcol project future development.

The overall project assessment is that, although not completely fulfilling all the goals stated at this project beginning, the results obtained in the developed parts are fully satisfactory. At the project start, the warping process has stated to be a trivial part, by simply applying the homography method. However, the development has arose many issues a priori not taken into account. These difficulties had recommended a change in the project direction, focusing on obtaining a more precise warping algorithm. This development sprung good results –and many discarded algorithms– and establish a base-line for the future development.

This project had left the development in a solid stage, with several work strands to be followed. First of all, an automatic registration algorithm can be implemented on the final product. This algorithm should automatically detect the chessboard corners and automatically do the key points registration. Another opened work strand is the definitive blending algorithm, directly bounded up with the above ground object treatment. In this part, an algorithm has to be defined to manage this objects and obtain the minimum image distortion.

APÈNDIX A

DESCRIPCIÓ DE LES TASQUES

Document: critical_review.pdf	Project Critical Review Prototype of a Cashless System	 
Date: 09/05/2016		
Rev: 01		
Page 6 of 8		



2. UPDATED WORK PLAN

2.1. Updated Work Packages, Tasks and Milestones

Project: Prototype of a cashless system	WP ref: WP1	
Major constituent: Programming Learning	Sheet 1 of 6	
Short description: Learn python language and other software concepts that will be needed in the software development in order to design the software system.	Planned start date: 01/02/2016 Planned end date: 12/03/2016	
	Start event: T1 End event: T2	
Internal task T1: Learn python. Internal task T2: Learn to use other needed tools.	Deliverables: None	Dates: None

Project: Prototype of a cashless system	WP ref: WP2	
Major constituent: System design	Sheet 2 of 6	
Short description: Design of the whole system (software and hardware) fulfilling the project requirements and specifications Also will have to do all proofs of concept to ensure that the design is feasible.	Planned start date: 28/02/2016 Planned end date: 26/02/2016	
	Start event: T1 End event: T4	
Internal task T1: Design server application Internal task T2: Design client application Internal task T3: Design hardware system Internal task T4: Proofs of concept	Deliverables: None	Dates: None

Project: Prototype of a cashless system	WP ref: WP3	
Major constituent: System development	Sheet 3 of 6	
Short description: Develop the system following the system design.	Planned start date: 27/03/2016 Planned end date: 28/05/2016	
	Start event: T1 End event: T3	
Internal task T1: Develop the server application Internal task T2: Develop the client application Internal task T3: Develop the hardware system	Deliverables: None	Dates: None

Document: critical_review.pdf	Project Critical Review Prototype of a Cashless System	 
Date: 09/05/2016		
Rev: 01		
Page 7 of 8		

Project: Prototype of a cashless system	WP ref: WP4	
Major constituent: Design the test unit	Sheet 4 of 6	
Short description: Design the test and demonstration unit that will be portable to show the results of the project	Planned start date: 29/05/2016	
	Planned end date: 02/06/2016	
	Start event: T1	
	End event: T1	
Internal task T1: Design the test unit	Deliverables: None	Dates: None

Project: Prototype of a cashless system	WP ref: WP5	
Major constituent: Develop the test unit	Sheet 5 of 6	
Short description: Develop the test unit that will be used in the presentation following the design.	Planned start date: 03/06/2016	
	Planned end date: 09/06/2016	
	Start event: T1	
	End event: T2	
Internal task T1: Develop the test unit	Deliverables: None	Dates: None
Internal task T2: Test the system on the test unit		

Project: Prototype of a cashless system	WP ref: WP6	
Major constituent: Writing	Sheet 6 of 6	
Short description: Write down the final degree project	Planned start date: 10/06/2016	
	Planned end date: 27/06/2016	
	Start event: T1	
	End event: T1	
Internal task T1: Write the final document	Deliverables: Final degree project	Dates: 27/06/2016

Milestones

WP#	Task#	Short title	Milestone / deliverable	Date (week)
2	4	Proofs of concept	Validated Design	26/02/2016
3	3	System development	System developed	07/05/2016
5	2	Test Unit Development	System tested on test unit	09/06/2016
6	1	Final Degree Project	Document	27/06/2016