

FRIDGES

Food-waste Reduction with Intelligent Decisions Generating Energy Savings

Phase 1 Report

CS413 Embedded Systems

Paul McGurk, Alex McBride, Daniel Rafferty,
Andrew Mortimer, and Scott Henderson

October 20, 2015

Contents

1 Introduction	3
1.1 About the Project	3
1.2 Motivation	3
1.2.1 Environmental Impact	3
1.2.2 Our Goal	3
2 Assessment of Capabilities	4
2.1 Requirements	4
2.2 Arduino	4
2.3 Raspberry Pi	5
2.4 Choice	5
3 Hardware Design	6
3.1 Main Parts	6
3.1.1 RGB LED	6
3.1.2 Resistor	6
3.1.3 Thermistor	7
3.1.4 Capacitor	7
3.1.5 Hall Effect Sensor and Magnet	7
3.1.6 Raspberry Pi Camera	8
3.1.7 Raspberry Pi Touch Screen Display	9
3.2 Component List	10
3.3 Prototype Temperature Circuit	11
3.3.1 Circuit Diagram	11
3.3.2 Result	11
3.4 Prototype Door State Circuit	12
3.4.1 Circuit Diagram	12
3.4.2 Circuit Schematic	12

4 Software Design	13
4.1 Operating System	13
4.2 Accessing the hardware	13
4.3 Web server	13
4.3.1 Why use a Web Server?	13
4.3.2 Technology Stack	13
4.4 Temperature Measurement	13
4.5 Barcode Scanning	14
4.5.1 Barcode Scanning Library	14
4.5.2 Identifying Products using the Barcode	14
4.6 Door State Detection	14
4.7 Interface	15
4.7.1 Local	15
4.7.2 Remote	15
5 GANTT Chart	16
6 Organisation	17
7 Current Progress	17
8 Stretch Features	18
9 Conclusion	18

1 Introduction

1.1 About the Project

The project is to create a product using an Arduino and/or Raspberry Pi that will be a part of the Internet of things. The Internet of things is about taking everyday objects and embedding them with software, electronics and networking capabilities so that they can sense the environment around them and then send and receive data. The idea, after much thought and discussion will be to create a smart fridge. This idea will not only have a physical benefit to its application but also an environmental one, as from the start the design is centred on trying to reduce food waste and energy consumption.

1.2 Motivation

According to statistics from LoveFoodHateWaste (a website run by WRAP - Waste and Resources Action Programme) UK households throw away 7 million tonnes of food every year. One of the main reasons for this is not using the food before it goes out of date. The average cost per household of this food wastage is £470 per year.

1.2.1 Environmental Impact

The same website claims that if this food wastage was reduced to 0, it would be equivalent to taking a quarter of the cars in the UK off the road. According to studies from WRAP, by ensuring our fridges are at the right temperature (between 0 and 5 °C), as a country there is a potential of £200 million to be saved. Apparently, 70% of fridges are kept too warm, meaning food becomes inconsumable much more quickly.

1.2.2 Our Goal

By allowing a user to know more easily when foods are soon to be out of date, we hope to reduce food waste and the environmental impact it has.

2 Assessment of Capabilities

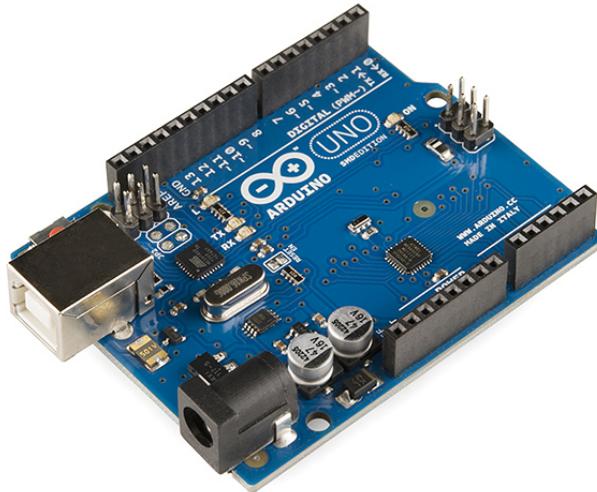
2.1 Requirements

The device we end up choosing must be able to support all the hardware and software we want to incorporate into our smart fridge. One of things we want to be able to support is a touchscreen to allow the user to interact with the fridge directly and on top of this a camera to act as a barcode scanner and then display the product on the screen. Another device that we also want to support is a thermometer that will be positioned inside the fridge allowing us to monitor the temperature in the fridge by taking readings. The final thing our device must have is internet capabilities, this is so that we will be able to set up and run our own Web server for the fridge.

2.2 Arduino

The Arduino Uno V3 is a small microcontroller board sporting an ATmega328p processor. It is designed for prototyping embedded products and as a result has good support for driving ICs and other low-level peripherals. The 6 PWM outputs allows analogue components to be operated easily, and the lean 16MHz processor is extremely low power meaning it could potentially be powered from a battery.

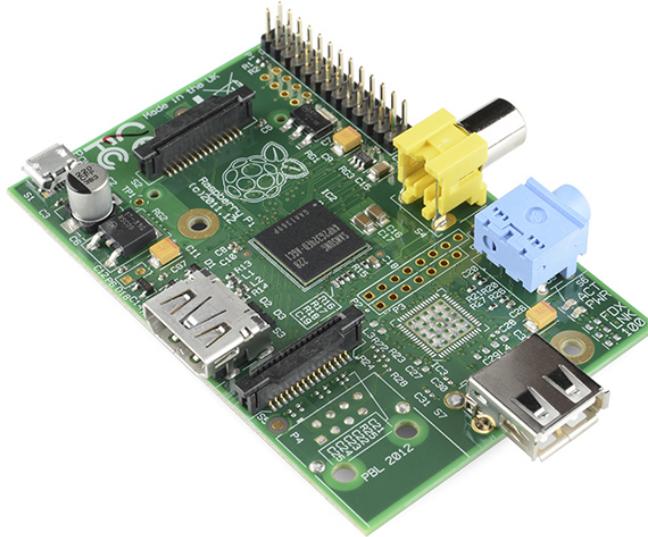
Figure 1: Arduino Uno V3



Arduino Uno V3	
Microcontroller	16 MHz ATmega328p
SRAM	2KB
Flash Memory	32KB
USB 2.0	None
HDMI	None
Audio Output	None
Digital I/O Pins	14
Analogue input pins	6

2.3 Raspberry Pi

Figure 2: Raspberry Pi



The Raspberry Pi is a small form-factor ARM based computer. It is a very popular system that has a healthy development ecosystem. The RPi can run a custom Linux-based operating system, called Raspbian, allowing fairly high level development and integration with lower-level peripherals such as GPIOs. The Raspberry Pi also has official peripherals or "modules" for devices such as a camera or a touchscreen.

There are several models of Raspberry Pi available:

	Raspberry Pi Model A	Raspberry Pi 2
CPU	700 MHz Low Power ARM1176JZ-F Applications Processor	900MHz quad-core ARM Cortex-A7
RAM	256MB	1GB
Ethernet port	None	One
USB 2.0	1 port	4 ports
HDMI	Full HDMI port	Full HDMI port
Audio Output	3.5mm audio jack	3.5mm audio jack
Number of GPIOs	17	40

2.4 Choice

We have chosen to go for the Raspberry Pi 2. The reason we chose a RPi over an Arduino was that we needed the higher performance that the RPi line offers in order to serve our Web pages and do the barcode scanning in software. Furthermore we wanted internet connectivity and we felt this would be easier to achieve in the Linux based Pi. The reason we chose to use the model 2 instead of the model A was the better connectivity for peripherals in terms of USB ports and available GPIOs, with the higher performance an added bonus.

3 Hardware Design

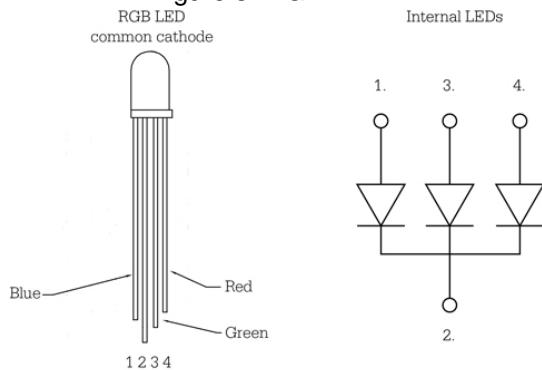
3.1 Main Parts

3.1.1 RGB LED

An RGB LED (Red Green Blue Light Emitting Diode) is a small light which has three different diodes within it, capable of emitting three types of light. Each pin controls a different colour, with the longest pin being the PLUS. Using PWM (Pulse Width Modulation), we can dim and lighten the different colours to give a range of colours. The operating temperature of the LED covers the expected temperature in the fridge.

This is being used within the project as a visual display of the temperature in the fridge, with blue representing when the reading is near the ideal temperature, and red when it is too warm.

Figure 3: RGB LED



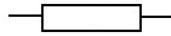
Colour	Wave Length	Forward Voltage	Forward Current	Luminosity	Operating Temperature
Red	623nm	2.0V	20ma	2800mcd	-25 ~ 85°C
Green	517.5nm	3.2V	20ma	6500mcd	-25 ~ 85°C
Blue	466nm	3.2V	20ma	1200mcd	-25 ~ 85°C

3.1.2 Resistor

A resistor is a device which reduces the flow of current within a circuit. This can be used for many applications such as protecting elements from high current.

This is being used within the project to protect the RGB LED, as a pull-up for the hall effect sensor, and within the temperature part of the circuit to work out the resistance of the thermistor (see below).

Figure 4: Resistor



Value	Quantity
10kΩ	1
1kΩ	2
470Ω	3

3.1.3 Thermistor

A thermistor is a resistor which is sensitive to temperature. The resistance of the thermistor is used to work out the temperature of its surroundings via the Steinhart-Hart equation.

This is being used within the project to work out the temperature within the fridge, allowing us to monitor it in real-time.

Figure 5: Thermistor



Value	Quantity
$1\text{k}\Omega$ (at 25°C)	1

3.1.4 Capacitor

A capacitor is a device which stores electrical energy temporarily.

This is used within the project to work out the temperature of the fridge, which is done by measuring how long it takes to drain the capacitor, with the resistance of the thermistor being dependent on the temperature.

Figure 6: Capacitor

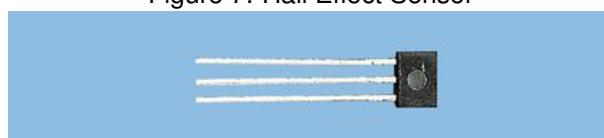


Value	Quantity
330nF	1

3.1.5 Hall Effect Sensor and Magnet

A Hall Effect sensor detects magnetic fields. It is often used as a non-contact switch for applications such as doors. We intend to mount the sensor on the chassis of the fridge next to the door, and mount a magnet on the door itself, aligned with the sensor. When the door is closed the sensor will detect the magnetic field from the magnet and the output signal will go low. When the door is opened, the magnetic field will not be detected by the sensor and the signal will go high.

Figure 7: Hall Effect Sensor



3.1.6 Raspberry Pi Camera

The Raspberry Pi Camera Board connects to any Raspberry Pi and allows for high definition photography. It has several useful features, including: high data capability, 5 mega-pixel fixed focus, support for 1080p, 720p60 and VGA90. It also has automatic control functions including white balance, exposure control, and luminance detection. The Pi Camera Board has a 15cm ribbon cable which slots into the Pi Camera Serial Interface Port. Via Raspbian, there are several applications that can be used to take photos, including Raspistill.

Figure 8: Raspberry Pi Camera



Technical Specifications	
Dimensions	8.5 x 8.5 x 5mm
Maximum frame rate capture	30fps @ 1080p, 60fps @ 720p
Maximum supported resolution	2592 x 1944
Supported Bus Interfaces	I2C
Supported Video Ports	Raspberry Pi
Minimum Operating Temperature	-30°C

The minimum operating temperature of -30°C means we could mount the camera inside the fridge if we so desire.

This will be used within the project to scan the barcode of the products being entered into the fridge. It could possibly automatically detect the use-by date as well.

3.1.7 Raspberry Pi Touch Screen Display

The Raspberry Pi 7" Touchscreen Display connects to a Raspberry Pi, and allows it to display graphics in a self-contained package. It also allows the user to interact with the GUI through touch, omitting the need for additional peripherals such as mouse and keyboard.

We chose this screen because other options were too small to be able to display a useful amount of information and still be able to type on.

Figure 9: Raspberry Pi Touchscreen



Technical Specifications	
Dimensions	194mm x 110mm x 20mm
Viewable screen size	155m x 86mm
Screen resolution	800 x 480

This will be used within the project to display a custom GUI which will have a number of uses, such as; display all products which are near their use-by date, adding custom products (homemade food, products which do not scan properly, etc). From this screen, the user will also be able to see the temperature of the fridge, and possibly configure the Web server.

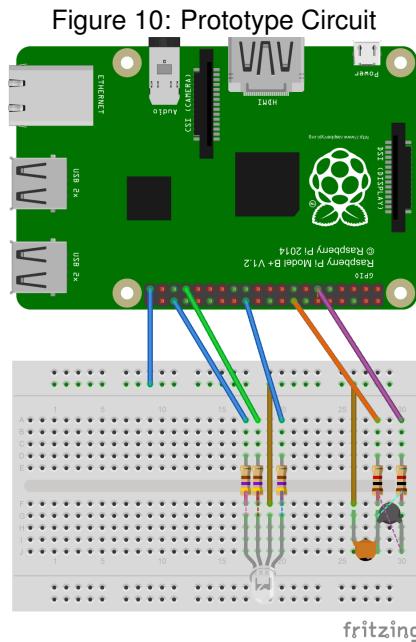
3.2 Component List

Quantity	Component Name	PriceUnit	Total Price
1	Raspberry Pi Touch Screen	£42.99	£42.99
1	Raspberry Pi Camera	£16.20	£16.20
1	330nF Capacitor	£0.09	£0.09
1	RGB LED	£2.76	£2.76
1	Thermistor	£0.60	£0.60
1	10kΩ	£0.053	£0.053
2	1kΩ	£0.027	£0.054
3	470Ω	£0.005	£0.02
1	Digital Hall Effect Sensor	£1.53	£1.53
1	Magnet	£1.055	£1.055
		Total	£65.37

3.3 Prototype Temperature Circuit

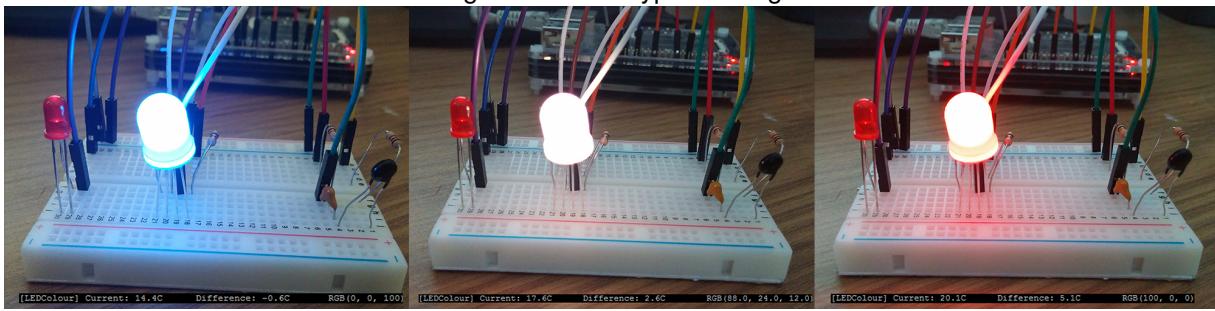
To accurately assess the feasibility of certain parts of this project, we created a prototype circuit of the temperature sensitive LED component. This was completed with a breadboard connecting the GPIO pins to a range of devices, most notable of which being a thermistor, and a RGB LED.

3.3.1 Circuit Diagram



3.3.2 Result

Figure 11: Prototype working

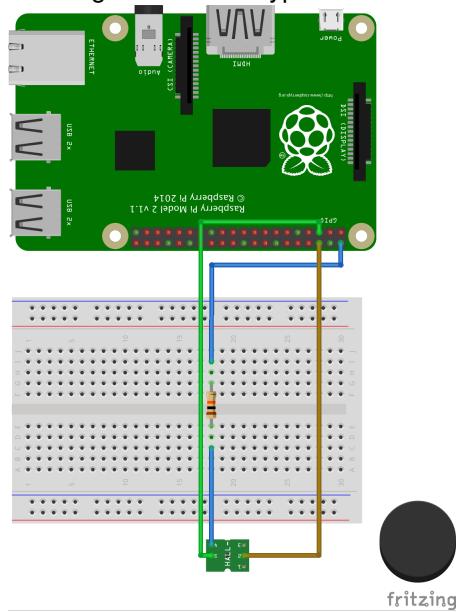


3.4 Prototype Door State Circuit

The prototype for detecting the state of the door consists of a breadboard, the hall effect sensor and magnet, and a resistor. We could do away from the breadboard since it is attached almost directly to the Pi.

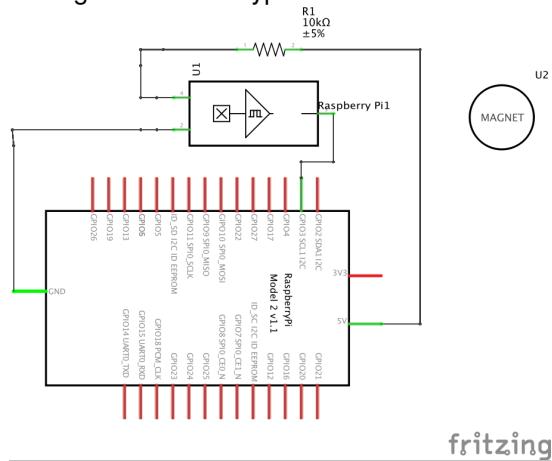
3.4.1 Circuit Diagram

Figure 12: Prototype Circuit



3.4.2 Circuit Schematic

Figure 13: Prototype Circuit Schematic



4 Software Design

4.1 Operating System

Raspbian is a Linux Debian-based operating system which is designed to run on a Raspberry Pi. It optimizes the full power of the Linux desktop environment for the Raspberry Pi's hardware. It allows us to focus on writing the functionality of the software that is unique to our smart fridge, leveraging the rich software ecosystem of 35000 packages that are offered by Raspbian to implement some of the more "standard" technologies such as the Web server.

4.2 Accessing the hardware

RPi.GPIO is a library that allows high level access to the GPIOs on the board through python bindings. Since we currently do not have plans to utilize the hardware PWM, SPI, or I2C functionalities we are not limited by the fact that RPi.GPIO does not support these. If we choose to use any of these functionalities then we can implement them in C and call down from Python fairly easily.

4.3 Web server

4.3.1 Why use a Web Server?

Most of the functionality of our project requires us to have an interface to the fridge to administrate and view information. We have chosen to go with a Web-based approach, with the RPi itself hosting the Web server. Going with this approach allows us to write the client application once and access it from a phone or a desktop without having to write client applications directly for each.

4.3.2 Technology Stack

Since the RPi.GPIO library provides access to the GPIOs via Python, we thought it prudent to serve the dynamic content of our application from Python also. However, we had originally implemented the server using Apache and PHP5, so we had to change this to a Python-based approach. We achieved this using the Flask framework. Flask features a Web server and a templating engine to allow us to write static HTML with wildcards that allow us to dynamically insert relevant data. This also allows us to access the sensors on the Pi to provide the user with up-to-date data straight from the fridge. Although none of us had any experience with this technology before, we thought it was a natural fit and did not have too much trouble implementing it.

4.4 Temperature Measurement

The temperature measurement element of this project is currently written in Python. This program was modified from a script written by Simon Monk for his "Raspberry Pi Starter Pack". The program works by measuring the time it takes to empty a 330nF capacitor, using the variable resistance from the thermistor, which depends on the temperature.

4.5 Barcode Scanning

4.5.1 Barcode Scanning Library

Zbar is an open source barcode reading library that supports a wide range of barcode formats. It has Python bindings allowing us to read the barcodes in Python.

4.5.2 Identifying Products using the Barcode

Outpan is a web service that provides an API for identifying products from their GTIN. It is free to use and has a database of millions of products. We can make simple HTTP requests with the barcode we just scanned and get information such as the name of the product back.

4.6 Door State Detection

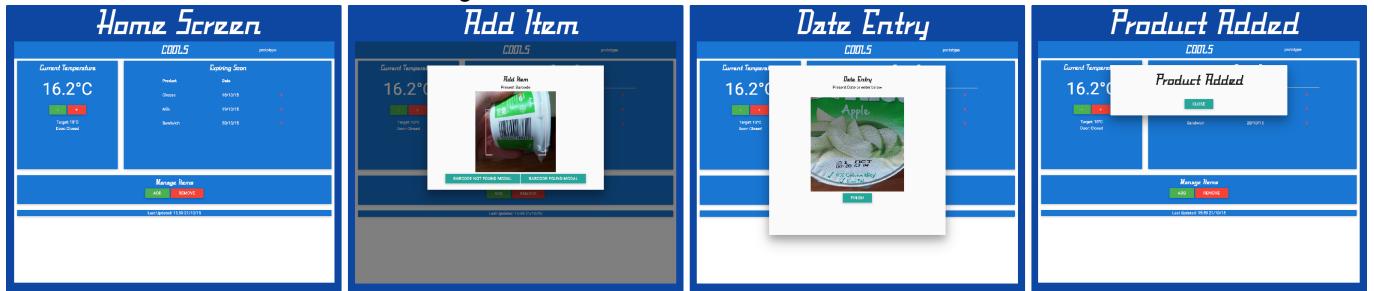
Using the hall effect sensor attached to a GPIO, we will be able to use the RPi.GPIO library to detect whether the door is open or closed. This information might be tied in with the temperature measurement in order to indicate to a user when they are leaving the fridge door open for too long.

4.7 Interface

4.7.1 Local

The touchscreen interface will be written in HTML5, JavaScript and CSS3, using the Materialize CSS for the visuals. As for the keyboard for entering data, we will use Matchbox keyboard.

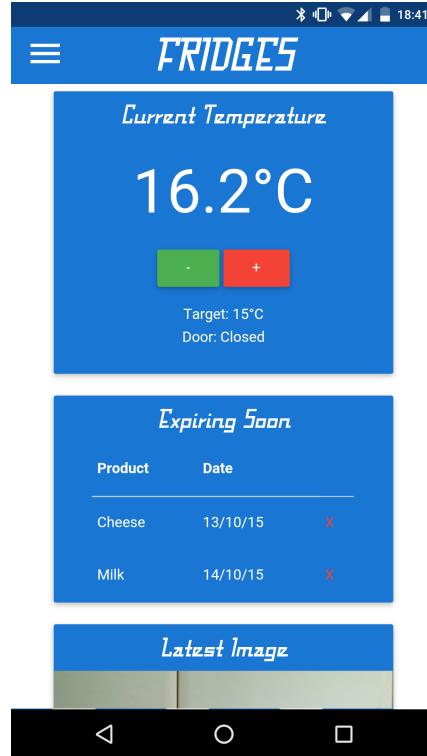
Figure 14: Touchscreen GUI flow



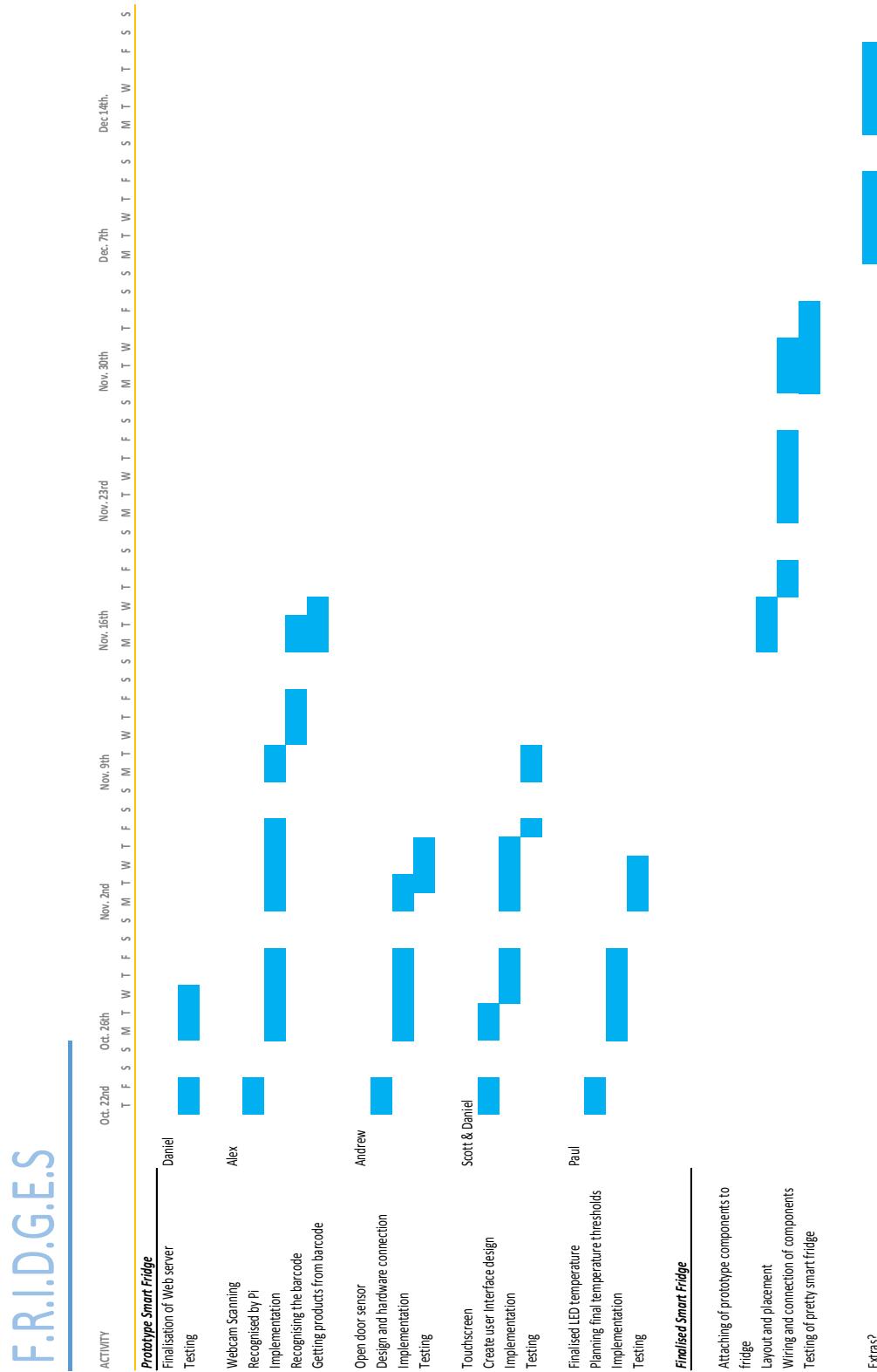
4.7.2 Remote

The mobile interface will also be written in HTML5, JavaScript and CSS3, again, using the Materialize CSS library. This interface will be similar to the touchscreen interface, visually, but will not be able to add products, at least to begin with.

Figure 15: Remote GUI accessed on a Mobile



5 GANTT Chart

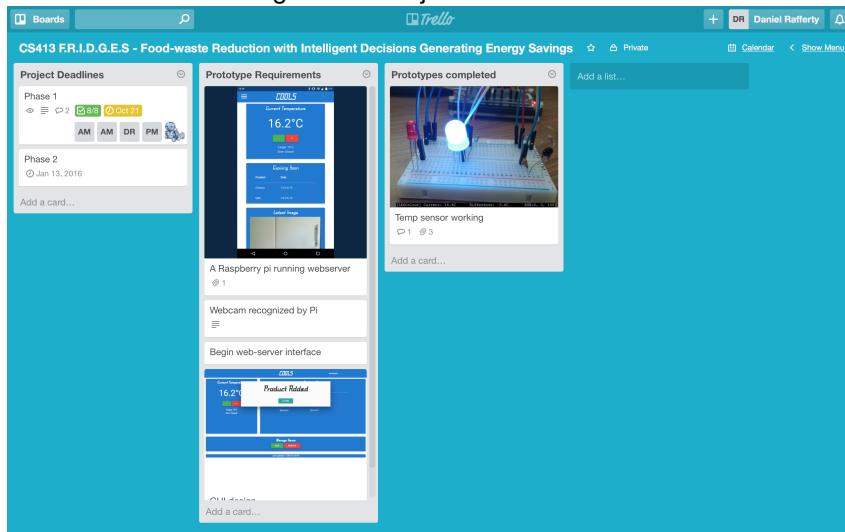


6 Organisation

To organise our group we used a few online tools. We are using a GitHub repository to keep track of changes in our code and to allow multiple people to work on the same file. With Git, we can see who is working on each part of the project, preventing people from doing the same part accidentally or doing nothing at all. This will also give us version control in case we have to roll back to a previous part of the project.

We are also using the project management application Trello. This enables us to assign members of the group to different "tasks" and keep track of what we have already done. This also allows us to keep track of due dates so our work will always be on time. Trello's commenting system means we can communicate easily as a group, without having to email everyone separately. This is useful as communication is one of the most important aspects of group organisation.

Figure 16: Project Trello Board



7 Current Progress

Our excellent group organisation has allowed us to have already made good progress on Phase 1 of our project. To begin with, we have researched all of the necessary components to complete our project, including deciding on a Raspberry Pi 2 as the basis for our embedded system. We have also created a demo to sense temperature and change the colour of an LED using this Pi, as this is one of the main features of our project. Another feature that we have already implemented is a Flask Web server running on the Pi. Flask will give us an easy way to store data and check the values of the sensors remotely via a Web interface. We have also planned a prototype that will allow us to identify the major challenges in our design/software before moving onto the final build of our project. The basis of a Web interface has been created to allow a user to view temperature, display current contents of the fridge (including expiration dates), and also to show the latest image from the Pi camera.

8 Stretch Features

If we meet all of our deadlines, there are some extra features we would like to implement. One of these features would be for the user to be able to change the temperature of the fridge from both the touch interface and remotely. This would be useful as it would give the user greater control over the fridge and enhance the energy saving aspects of the fridge.

One other feature would be to recommend meals based on what is currently in your fridge. These meals could be taken from an open-source recipe database, like Big Oven, and can be filtered based on what may be going out of date in your fridge. This further reduces food wastage.

Ordering more food would also be a great feature to aid in food reduction. The fridge could recognise when you are low on a specific product, and order more for you from your preferred supplier (Asda/Tesco/Amazon). This would result in less over-buying, as more food would only be ordered when the current stock is low.

User profiles are another optional feature we would like to add. This would allow a group of users (families, housemates) to mark which food is theirs, and receive personalised information. This would also allow user customisation, like changing the interface colours/backgrounds. This would also allow us to have the fridge only open for recognised users, either by login or NFC chip.

Our final extra feature would be to allow users to add widgets to the interface of their fridge. These could include Reminders, Weather forecasts, Daily News and even information from Twitter/Facebook feeds.

9 Conclusion

We have outlined a solid plan to go forward and start developing our smart fridge. By staying as organised as we have been so far, we should meet all the deadlines (submission ones as well as ones we have set ourselves) with no trouble. Building a smart fridge will be very interesting, not only because we will have a useful product by the end of the project, but also because of the environmental implications. By being able to monitor the temperature of the fridge, as well as alerting a user if the door has been left open, we will ensure the fridge will use the least amount of power possible. Secondly, by giving the user an obvious visual representation of which foods are going out of date soon, there is the hope that this will reduce food wastage. Furthermore, it is generally a useful system which will be applicable to the real world. We look forward to making good progress, and our strong group dynamic will allow us to overcome any obstacles with minimal stress.