

# FRIDGES

*Food-waste Reduction with Intelligent Decisions Generating Energy Savings*

Phase 2 Report

CS413 Embedded Systems

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

13th January 2016

## Contents

<b>1</b>	<b>User Guide</b>	<b>2</b>
1.1	Barcode Scanning . . . . .	2
1.2	Temperature Reading . . . . .	2
1.3	Open Door Alert . . . . .	2
1.4	Stock Count . . . . .	2
<b>2</b>	<b>Technical Manual</b>	<b>2</b>
2.1	Web Server . . . . .	2
2.2	Interface . . . . .	3
2.2.1	Local . . . . .	3
2.2.2	Remote . . . . .	3
2.3	Barcode Scanning . . . . .	3
2.4	Temperature Reading . . . . .	3
2.5	Open Door Alert . . . . .	3
<b>3</b>	<b>Final Project Plan</b>	<b>3</b>
3.1	The final project plan that identifies the contributions made by each member of the group. . .	3
3.1.1	Paul McGurk . . . . .	3
3.1.2	Daniel Rafferty . . . . .	3
3.1.3	Alex McBride . . . . .	3
3.1.4	Andrew Mortimer . . . . .	3
3.1.5	Scott Henderson . . . . .	3

## 1 User Guide

The idea of our Smart Fridge is to reduce food wastage and power consumption. By doing this, the environmental impact will be lessened. In order to achieve this goal, we added several features to an ordinary mini-fridge.

### 1.1 Barcode Scanning

When an item is placed in the fridge, the barcode can be scanned and its best-before date recorded. In order to do this, select the "Barcode Scanner" option using the touchscreen. Then hold the barcode up to the Pi Camera. This will scan the barcode and store it. Select the product if it appears in the product list that will be generated, otherwise it can be manually added to the database so that it will be remembered for next time. Once this is done, the best-before date can be entered on the touchscreen. This means, the fridge will know when each item of food in the fridge will be nearing its best-before date, and will display the food that is going out of date soon on the home menu of the touchscreen.

### 1.2 Temperature Reading

The thermometer takes constant readings of the temperature inside the fridge. An LED will change colour based on the current temperature of the fridge. Blue indicates a normal (or lower) operating temperature. This will transition to green then red as the temperature rises above normal. This will most likely be an indicator that the fridge door has been left open and the fridge will have to use more energy to bring the temperature back down to the normal operating temperature.

### 1.3 Open Door Alert

If the door has been left open for more than one minute, it will send a notification to the web application that the door has not been closed.

### 1.4 Stock Count

Select the "Stock" menu on the touchscreen to be given a display of what is currently in the fridge and how much of each item there is.

## 2 Technical Manual

### 2.1 Web Server

We decided to use a Web-based approach for our interface. The Pi itself hosted the Web Server. This meant we only had to write the client application once, meaning it could be accessed from a phone or desktop without client applications having to be written for each.

We served the dynamic content of our application using Python, seeing as the RPi.GPIO library provides access to the GPIO pins using Python. We used the Flask framework. Flask features a Web server and a templating engine to allowed us to write static HTML with wildcards that allowed us to dynamically insert relevant data. This also allowed 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 had any experience with this technology before, we thought it was a natural fit and did not have too much trouble implementing it.

## 2.2 Interface

### 2.2.1 Local

### 2.2.2 Remote

## 2.3 Barcode Scanning

The barcode scanner uses a camera which is attached to a Rasperry Pi. The camera scans for barcodes every fraction of a second. When the barcode of a product is held up to the camera, a Python script is run to search the open-source barcode library, Zbar. Originally, the plan was to use Outpan, which is a web service that provides an API for identifying products from their GTIN. However, Outpan didn't seem to recognise many products, so we decided to use TescoLabs instead. (One disadvantage of this is that it limits us to using products sold from Tesco only). Using the Python code, a simple HTTP request could be made with the barcode just scanned and information on the current product would be returned.

## 2.4 Temperature Reading

In order to detect the temperature of the fridge, a Rasperry Pi was connected to a bread board which housed a range of devices. The most notable of these were the thermistor and an RGB LED. As a thermistor's resistance changes based on the current temperature, we set up the LED to change colour based on the current resistance of the thermistor. If the resistance is above the threshold temperature, the LED will turn red. Otherwise it will remain blue. The code to allow this to happen was written in Python, modified from a script written by Simon Monk for his "Rasperry Pi Starter Pack". The program worked by measuring the amount of time taken for a 300nF capacitor (using the variable resistance from the thermistor) to empty. This was depended on the temperature.

## 2.5 Open Door Alert

Using some Python code and a hall effect sensor, it was possible to constantly check for the presence of a magnetic field, which was from a magnet attached to the door of the fridge. If no magnetic field was detected, after 60 seconds an alert was raised in the Python code. This was relayed to the touchscreen display.

In order for the software on the Pi to access the hardware used, we utilised the RPi.GPIO library to allow access to the GPIO pins on the board.

## 3 Final Project Plan

### 3.1 The final project plan that identifies the contributions made by each member of the group.

#### 3.1.1 Paul McGurk

#### 3.1.2 Daniel Rafferty

#### 3.1.3 Alex McBride

#### 3.1.4 Andrew Mortimer

#### 3.1.5 Scott Henderson