Designing a System to Quantify Waste in Domestic Heating

Joel Hutton

BEng Hons Project Report
Computer Science and Electronics (BEng Hons)
School of Informatics
University of Edinburgh

2017

Abstract

The aim of the project was to build a sensor system to assess the viability and usefulness of a smart heating control system that takes into account individual user preferences in a shared home. In order to do this devices were designed to collect raw data from volunteer properties. Validation data was collected over SMS with the inhabitants by sending reminder texts with a link to a web interface three times per day at random times between 08:00 and 23:59. The data was then examined to determine if such a system would be viable and useful. Some success was found in determining which users were in or out of the property at a given time as well as being able to track when specific users were usually home. The users were found to prefer significantly different temperatures, indicating that a system that takes into account specific users' preferences could save energy.

Table of Contents

Chapter 1

Introduction

As growing concerns around climate change gain traction worldwide, focus has increasingly been placed on domestic heating, which has been identified as a major contributor to energy consumption and carbon emissions in Europe (Dounis and Caraiscos, 2009). Although the United Kingdom (UK) is home to only 1% of the worlds population, it is responsible for 2.3% of global carbon emissions, with 25% of these emissions related to domestic space heating (Liao, Swainson and Dexter, 2005). In an effort to reduce its overall impact on the global carbon footprint, the UK has set a goal to reduce national carbon emissions by 80% by 2050 and has identified energy consumption due to domestic heating as a priority area for action in order to achieve this goal (Department of Energy and Climate Change, 2015).

1.1 Carbon emissions and domestic heating systems

There are a number of UK and European Union (EU) initiatives already set in place to reduce carbon emissions related to domestic space heating. Among these are the 2013 UK housing regulation requiring all dwellings greater than 150m^2 to have at least two independently controlled heating zones (UK Communities and Local Government, 2008) as well as the European Commission 'Smart Grids Task Force', which aims to replace traditional electricity and gas meters with smart meters that can better inform users about their energy consumption. By 2020, it is expected that almost 72% of European consumers will have a smart meter for electricity and 40% will have one for gas (European Commission, 2017). In terms of the role of new technology in addressing energy concerns, the UK Department of Energy and Climate Change has recently launched the Smarter Heating Controls Research Programme to develop and improve technology to reduce energy consumption due to domestic heating systems, which has been identified as a crucial area of interest in plans to eliminate carbon emissions in line with the 2050 carbon emissions target (Department of Energy and Climate Change, 2015).

1.2 The role of automated domestic heating systems in reduction of energy consumption

As indicated above, domestic heating systems are currently being targeted in campaigns to reduce energy consumption and the associated carbon footprint. Automated heating systems are key to improving the efficiency of domestic energy use (Liao, Swainson and Dexter, 2005). Currently, as many as 90% of homes in the United Kingdom (UK) use central heating systems, though as many as 70% of homes do not have basic automatic controls installed, suggesting significant room for improvement in energy consumption (Utley & Shorrock, 2008; Department of Energy and Climate Change, 2015). Automatic controls for heating systems can range from basic measures, such as timers paired with thermostats to turn heating on and off, to more advanced measures such as zoned internet-connected systems (Nagale, Kasper and Girod, 2017). These complex internet-connected systems, often termed 'smart' systems, allow for more efficient energy usage and show great potential for reducing domestic energy consumption. Smart heating control systems work to reduce the carbon footprint of heating control systems by maximizing efficiency in home heating by identifying which rooms are, or will be, in use at particular times and the temperature which is preferred within those rooms (Kleiminger, Mattern and Santini, 2014). This is thought to be an improvement over more basic models of automated control systems, which may heat entire zones based on a timer without regard to whether residents are home or where in the house they are. Following this, smart heating control systems, particularly those which can be controlled remotely, are often marketed as being user-, cost- and environmentally-friendly (Nest, 2017; Honeywell, 2017; Heatgenius, 2017).

However, current smart heating control systems available for purchase have substantial room for improvement, both in terms of cost and capability. Many popular lower-cost models, including Honeywells Lyric T6 thermostat and the Nest Learning Thermostat, available at approximately £130 and £249 respectively, do not allow for zonal heating in the base package, meaning that basic models control the heating systems of an entire property at once based on past user data (Honeywell UK, 2017; Nest Learning Thermostat, 2017). While these smart systems allow for greater user control than traditional heating control systems, it is arguable that the lack of control by zone in these basic packages could limit their impact on energy consumption, as the entire property is being heated regardless of which areas users are in. Models capable of zonal heating control, such as the Honeywell Evohome and Heatgenius are considerably more expensive: for a starter pack and sensors for five rooms, the cost is £424.95 for the Heatgenius model and approximately £500 for the Honeywell Evohome (Heatgenius, 2017; Honeywell, 2017). It is possible that the high cost of these systems may limit their uptake in the consumer market. Additionally, no smart heating control systems currently on the market account for different temperature preferences among individual users. Personalising smart heating control systems to individual user preferences was identified as a key area in need of improvement in a review of optimized control systems in 2014 (Shaikh et al, 2014) and offers opportunities in terms of reducing energy consumption as well as improving the comfort and experience of the user in their home.

This project aims to improve upon current smart heating control systems by 1) designing and fabricating a cost-effective device which is capable of collecting user data for analysis and 2) testing whether such a device would be a viable option for predicting user heating patterns as would be done in a smart home heating system. This is intended to provide preliminary evidence supporting the concept of a low-cost personalised smart home heating control system with the intention of challenging expensive and less capable models currently on the market. Evidence supporting the potential for such a device could suggest an opportunity for more widespread adoption of smart home control systems in line with national priorities for reducing domestic energy consumption and the associated carbon footprint.

My personal contributions were:

- Designing the devices
- Sourcing components
- Making the devices
- Testing the devices
- Programming the devices
- Installing the devices
- Designing the server side software
- Collecting outside temperature data from an API (wunderground)
- Analysing and visualising the data

Chapter 2

Designing the hardware

2.1 Requirements

A major criticism of current smart home heating control systems is that they are very expensive and thus appeal to only a limited portion of the consumer market. In order to address this criticism, this device was intentionally designed to be low-cost. The device designed and fabricated in this project is similar to sensors sold as part of smart heating control systems such as Honeywell's Lyric T6 (£130) or Heatgenius' sensors (£34.99 per sensor) (Heatgenius, 2017; Honeywell, 2017). In comparison to the price of similar sensors on the consumer market, the cost target this project was set at £10 per device in parts.

In order to achieve a low start cost the sensors should be able to work individually without supporting equipment (such as a hub or controller). As these sensor devices were to be retrofitted wireless communication was essential to avoid the need for running cables throughout the property inconveniencing users and adding a cost to fitting the device. The devices needed to be able to measure temperature, motion and light. It was desirable to be able to measure humidity as this is thought to have an effect on thermal comfort though this was not possible, as discussed later (Jing, S., Li, B., Tan, M., & Liu, H, 2013). There also needed to be some method of tracking individual users within a property.

2.1.1 Wireless communication

Many home automation devices use 433MHz radios for communication. One advantage of these is that they can be very low power. These radios were considered for this project, however due to the desired aim of having no central hub or controller, they were deemed unsuitable.

WiFi was the preferred choice of communication as it removes the need for any central hub or controller and utilises existing hardware (the router). This reduces cost and means a user could buy a single sensor and have it work with no supporting equipment.

A further advantage of WiFi is that it can be used to track other WiFi devices, and thus the people who carry them. One disadvantage of using WiFi is that it tends to be higher power than other wireless communication. Within the scope of this project power consumption was not a large concern as the device was plugged in and power was negligible in comparison to heating (<500mW). However devices such as the Amazon Dash button have shown that transmitting a small amount of information over WiFi periodically can be done for long periods of time before having to replace batteries, even using relatively small batteries. As the WiFi transmission hardware only needed to be active periodically it would not use excessive power on WiFi.

2.1.2 Microcontroller

The microcontroller needed to have WiFi capabilities and enough RAM and flash memory to hold and run a program capable of running a WiFi stack and reading from several sensors. There were 2 candidate choices: The ESP8266 and the ESP32.

The ESP32 had the advantage of both more SRAM (512kb vs 160kb for the ESP8266), and having dual cores. However it was significantly more expensive than the ESP8266, costing £9.35 for a development module vs £3 for an ESP8266.

The ESP8266 was chosen as it could be obtained for £3 in small quantities, it had WiFi built in and had good community support. It also had hotspot capability, which was deemed to be useful for configuration and an analogue to digital converter (ADC), which was useful for measuring analogue sensors.

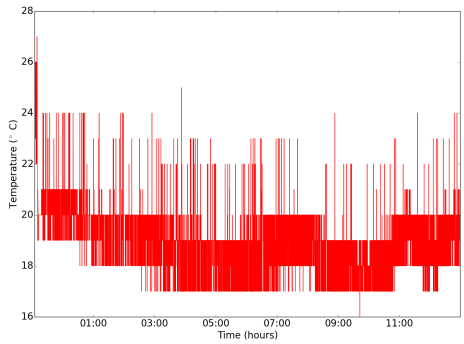
The ESP8266 with its default firmware is a WiFi-to-serial converter, allowing a microcontroller with a serial port to communicate over WiFi. There are some advantages to this configuration, such as the ability to use a very low power microcontroller and only wake up the ESP8266 when tranmission is needed, giving very low overall power usage. This configuration was not chosen as it adds an additional cost in parts, larger board area and a higher complexity. Instead the firmware of the ESP8266 was programmed directly with the aid of the espressif sdk.

2.1.3 Temperature/humidity sensor

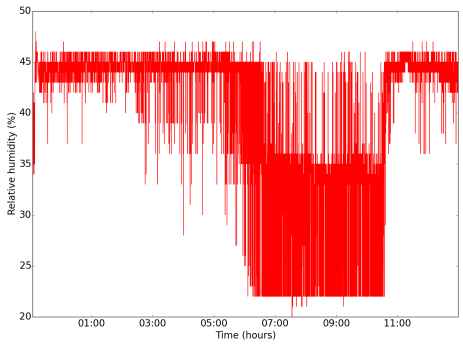
The DHT22 was initially selected as it was available at low cost (£2.69 per item) and provided both temperature and humidity readings on a combined sensor, which was seen as an opportunity to get humidity data at low cost. A prototype device using the DHT22 sensor was tested by leaving it overnight in a room at a test property so that it transmitted to a server while the room was in normal use as seen in Figures 2.1a and 2.1b. This test indicated two problems regarding the DHT22 sensor: 1) readings were being treated as integers, so that values were reported only as whole number values, limiting precision, and 2) the sensor readings would vary by $\pm 5°$C, and up to 7°C, for temperature and $\pm 5\%$ for humidity within a few seconds of each other, far out of the specification given in the provided datasheet (Liu, T.,2008). The datapoints appearing as whole number values was found to be due to a float being cast to an

integer in the data collection process, but this did not account for the rapid oscillation in readings. The temperature sensor was replaced by another DHT22 but the inaccuracy and instability persisted.

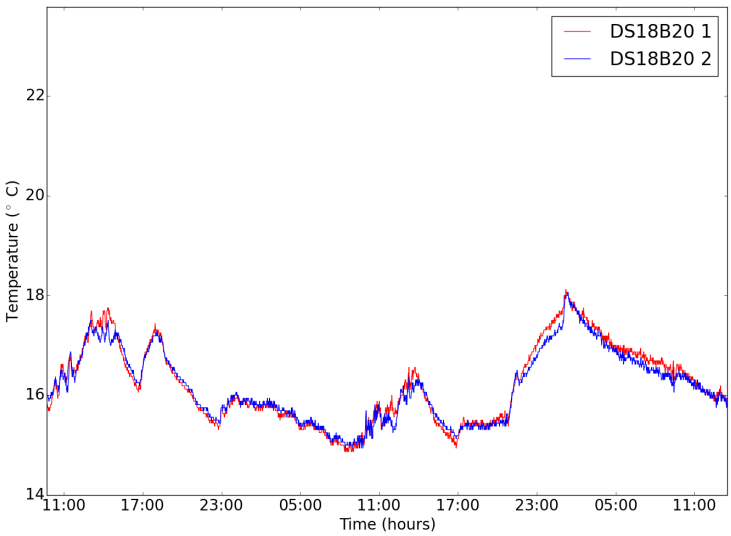
As the DHT22 was found to be unsuitable for the purposes of this project, the DS18B20 was sourced. The DS18B20 was chosen because it occupied the same footprint as the DHT22 (both use 3 pins, VDD, data and GND in the same order) and so could occupy the same place on the board, requiring minimal redesign of the device. It also had a stated accuracy of ±0.5°C (Dallas Semiconductor, 2008). Unfortunately, the DS18B20 was a slightly higher price (£3.68 per item). Some software changes were required to replace the DHT22 with the DS18B20, as it uses a different protocol (onewire), and a 1k pull-up resistor was added in accordance with documentation for the DS18B20 (Dallas Semiconductor, 2008). Two DS18B20 were tested by leaving two prototype devices next to each other overnight to be sure the DS18B20 had accurate, stable readings in agreement with one another. As seen in Figure 2.1c, the new temperature sensors were in agreement to within 0.5°C at all times.



(a) Unstable temperature readings from a DHT22



(b) Unstable humidity readings from a DHT22



(c) Temperature readings from 2 DS18B20

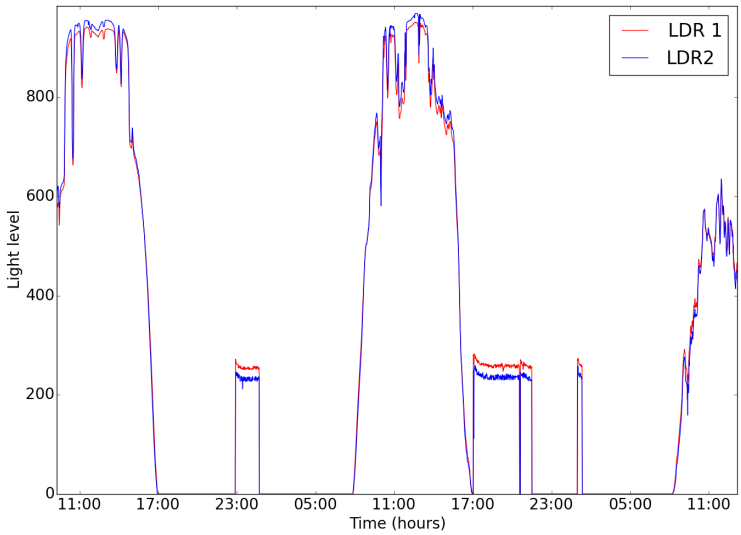


Figure 2.2: Light sensor test data

2.1.4 Light sensor

The light sensor needed to be low cost, and of all the options surveyed the lowest cost option was an analogue light-dependent resistor (LDR). As the ESP8266 has an an analogue-to-digital-converter (ADC) analogue sensors were possible.

The light sensors were tested in a similar manner to the temperature sensors by placing them in prototype devices next to each other and checking they were in agreement. The light sensor consisted of an LDR and a resistor as seen in Figure 2.4. The analogue voltage from the light sensors was converted into a 10 bit integer by the ADC giving them a range from 0 to 1024. As the LDRs were on the low voltage side of a voltage divider (as seen in Figure 2.4) and as an increase in light incident on the LDR decreases its resistance, it was necessary to invert the values by using the formula $inverted = (1024 - value)$. This gives a value where 0 represents no light detected and 1024 is the maximum measurable brightness. A value of 1024 was only be achieved in testing by shining a bright torch directly onto the LDR and a value of 0 was obtained by covering the sensor completely with black tape.

The light sensors were tested in the same way as the temperature sensors and can be seen to have been in agreement from the plot in Figure 2.2. Step changes are seen where lights are turned on and off as well as cyclic daylight patterns. The sensors can be seen to have a slight offset of around 10 units/1024. This was attributed to tolerances in the manufacturing of the light sensors and the resistors used in the voltage divider and was within the tolerances specified by the (non light-dependent) resistors (1%).

2.2 Circuit design

The circuit was prototyped on a solderless breadboard, to test the software and hardware. This circuit was used during testing of components and during writing the software. The prototype circuit can be seen in Figure 2.3 with the wiring diagram in Figure 2.4.

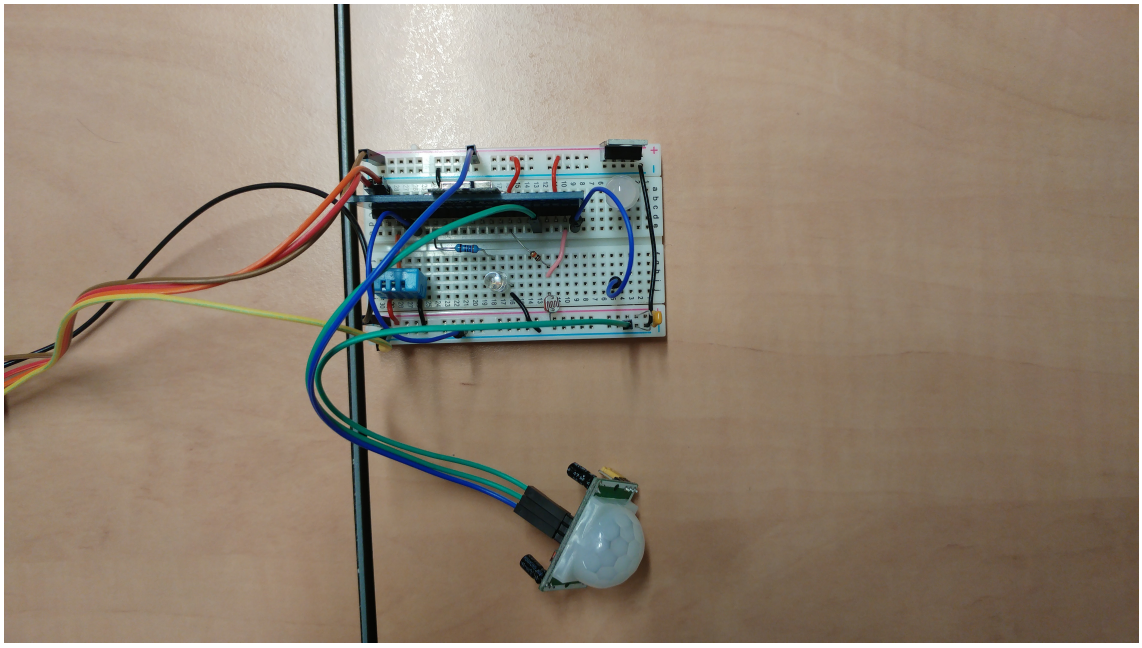


Figure 2.3: Prototype Circuit

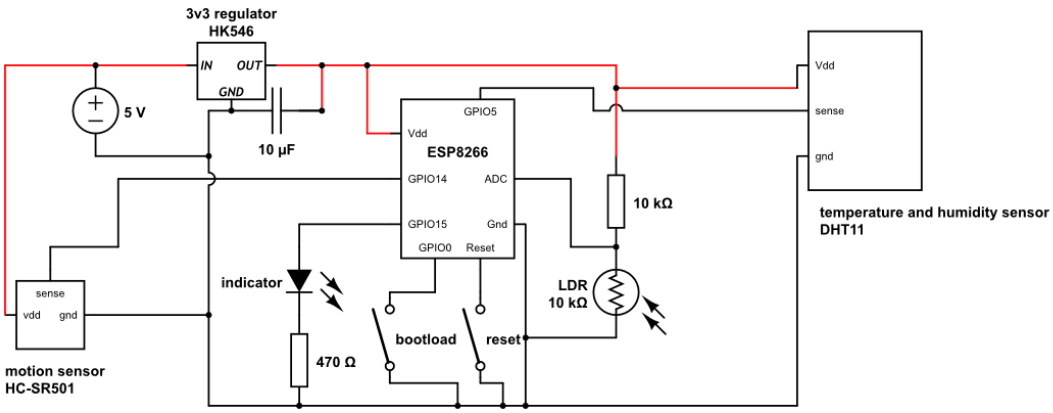


Figure 2.4: Prototype Circuit Diagram

During prototyping the ESP8266 module was switched for an ESP8266 based dev board - the Wemos D1 mini - instead of using just the bare module. This integrated the voltage regulator, bootload button and reset button. It also cost the same as the bare ESP8266 module making it a cost saving when the voltage regulator and buttons are

considered. The D1 also had a micro USB connector for convenient power supply and programming compared to soldering directly to the board or to header pins.

After finalising the circuit it was moved onto veroboard. Circuits on veroboard rely on through hole components being placed through the perforations and soldered to the copper cladding on the underside. The copper cladding is arranged in 'rails' that run the length of the board. Where 2 components are connected onto the same rail but need to not be electrically connected, the rail must be broken.

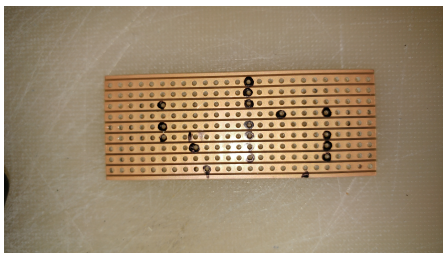
The layout was planned out using an online routing tool before being copied onto the top and bottom of the veroboard. The board breaks on the veroboard, along with the component placement on the top of the board can be seen in Figure 2.5.

Considerations for the layout included:

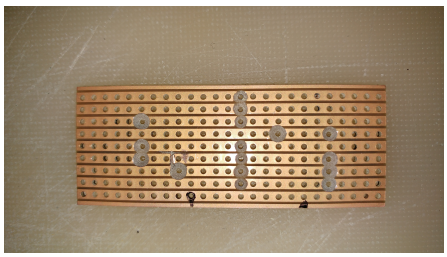
- Having the ESP8266 trace antenna overhanging the board so that capacitance with the copper cladding on the veroboard does not interfere and attenuate the signal
- Keeping the temperature sensor far away from sources of heat such as the Wemos D1 mini so it does not affect temperature readings
- Keeping the button accessible to users
- Keeping the LED and light sensor apart so the led does not affect the light readings
- Reducing the number of wires needed by using the rails effectively
- Reducing the number of breaks needed between adjacent holes as these are laborious during construction
- Keeping the area of the board down to keep the sensors small and reduce cost

The final circuit required only one break between adjacent holes, kept the board very compact, kept the button accessible, kept the LED away from the light sensor and kept the temperature sensor away from the Wemos D1 mini.

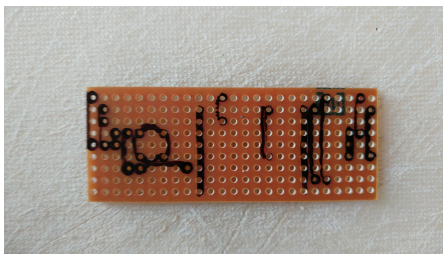
Testing showed the prototype board performed as expected and the rest of the boards were made to the same design.



(a) Underside with copper cladding



(b) Underside with breaks drilled



(c) Upperside with Components marked out

(d) Upperside with Components placed

Figure 2.5: Veroboard circuit

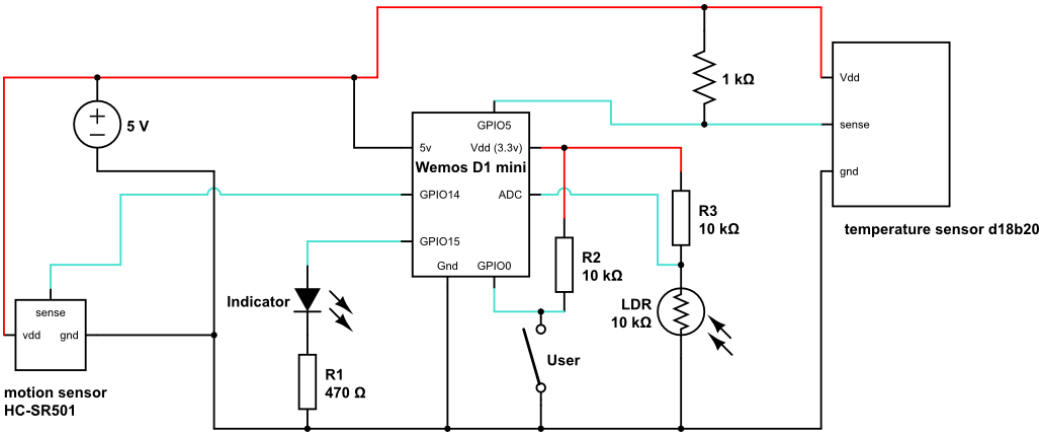


Figure 2.6: Revised circuit with colour coding

2.3 Final bill of materials

Item	Cost	Source
veroboard	£0.20	amazon.co.uk
light-dependent resistor	£0.33	amazon.co.uk
HC-SR501 motion detector	£1.85	amazon.co.uk
pushbutton	£0.10	existing parts
WeMos D1 mini ESP8266 board	£3.68	banggood.com
1% resistors * 5	£0.10	existing parts
header pins	£0.20	existing parts
USB power supply	£1.92	aliexpress.com
micro USB cable	£4.43	aliexpress.com
DS18B20 temperature sensor	£3.17	radiospares.co.uk
total	**£15.98**	

Table 2.1: Final Bill of Materials

The final cost was above the £10/unit target part cost but it is possible that with larger scale production the cost could be brought down, however on a large scale additional costs would have to be considered as will be discussed.

2.4 Fabricating and testing the devices

Once the prototype was tested the remaining 14 sensors were made in stages, labelling all the boards from 2-15, marking the undersides of the boards where the rails needed to be broken, marking the upper sides of the boards where components needed to be placed, drilling out the breaks in the copper rails with an electric drill (a hand tool is normally used but due to the relatively high volume a drill was faster), testing all the traces had been cut with a multimeter, placing components, then soldering components. As 15 boards needed to be tested a test program was written to test each feature - first blinking the led, then giving light readings and requesting the user to cover the light sensor, giving motion readings and requesting the user to move in front of the motion sensor, finally it gave temperature readings and requested that the user pinch the temperature sensor between their fingers to see a temperature change. The tests which could not be passed were recorded for each sensor in a spreadsheet along with the solution when it was fixed. Some common mistakes included misaligned parts, missed connections and short circuits, these were identified and fixed leaving all 15 boards functional.

Chapter 3

Software running on the devices

3.1 Configuration

The devices require some data from the user, in the form of the WiFi SSID and password, and the MAC addresses of the residents. The former is needed to connect to the WiFi and the latter to know which MAC addresses should be tracked, this is to avoid capturing private data beyond the scope of the project.

The initial design was to include a web server for configuration, where the data could be validated in Javascript and HTML and a friendly user interface could be presented. This only relies on WiFi and a web browser. Although this uses more program memory and RAM it allows the user to easily install the device themselves. There was not enough RAM for both the webpage and the rest of the program however, so a more lightweight solution had to be found.

The final configuration design was for the user to hold the button on the device to until the led turned on to put it into configuration mode. The device would then set up a hotspot with the SSID as the device's MAC address and wait for formatted UDP packets to give one of a number of commands:clear EEPROM (persistent) memory, add WiFi credentials or add a MAC address to track. On pressing the button again the device would reboot and try to use the supplied credentials.

3.2 Response from server

Most homes internet systems have restrictive firewalls by default and network address translation (NAT). Stateful firewalls allow a connection started by someone behind a firewall to be used for both incoming and outgoing traffic. Network Address translation maps network ports of devices on the local network to external ports of the router. As the devices only communicate with the server periodically, and are not connected to WiFi the rest of the time, initial attempts to configure the devices remotely were

unsuccessful as a connection was not maintained and the NAT port changed for every transmission.

By adding a waiting period where the devices would stay connected to WiFi for 5 seconds after transmitting, the server was able to respond through the opened port. Incoming messages from the device have a ‘destination’ port and a ‘source’ port, the source port is translated by the router into a randomly assigned port and sent on to the server. By setting the destination port of the response to the received source port it is translated back to the original source port by the router and allowed through the stateful firewall as part of a maintained connection.

This allowed some information, namely which MAC addresses to track, to be sent to the devices from the central server after installation, making the setup process less laborious as only the WiFi credentials needed to be added to the device at installation time.

3.3 Sniffing WiFi

In order to detect nearby WiFi devices the device was put into ‘promiscuous mode’ this allows packets to be intercepted. Following the 802.11 spec seen in Figure 3.1 there should be up to 4 MAC addresses in an 802.11 frame. Using the wifi_set_promiscuous libraries provided by the espressif sdk frames could be captured and sent to a pc. Looking in the expected location for the MAC addresses did not show the presence of known devices in the vicinity. Known devices were not in the data and there were unusual sequences such as many 0x00 or many 0xFF which did not appear to be valid MAC addresses.

It appeared, the packets did not give the entire frame. In order to find the length of the preamble and the location of the MAC addresses the captured data was compared to known data. According to the 802.11 standard routers send out ‘beacons’ in a standard format with the SSID in plaintext. Using this a known ssid ‘eduroam’ (hex 0x656475726F616D) was searched for in the capture data. This gave the location of the frame body (example in hex below), working back from this it was possible to tell where the addresses should be.

```
AB1BE75000000000000000000080000000FFFFFFFFFFFFFEEFEFFEFEEFF4
CFE26321C1B00648887DA9E6030000660031140007656475726F616D
```

To verify the location of the addresses, traffic was generated from a WiFi device of known MAC address: `C48E8FF2E8B9`, then captured packets from the prototype device were dumped to a pc, and examined using regular expressions to find matching packets. By examining packets such as the one below the following locations were found: ADDR1:bytes 16-22, ADDR2: bytes 22-28, ADDR3 bytes 28-34, ADDR4 bytes 36-42. The bold area below shows ADDR1 in a captured frame.

```
C40B205000000000000000094000000C48E8FF2E8B9000000000000400A0E7
FF07000000000000300000064002104010020000000400A0E7FF07
```

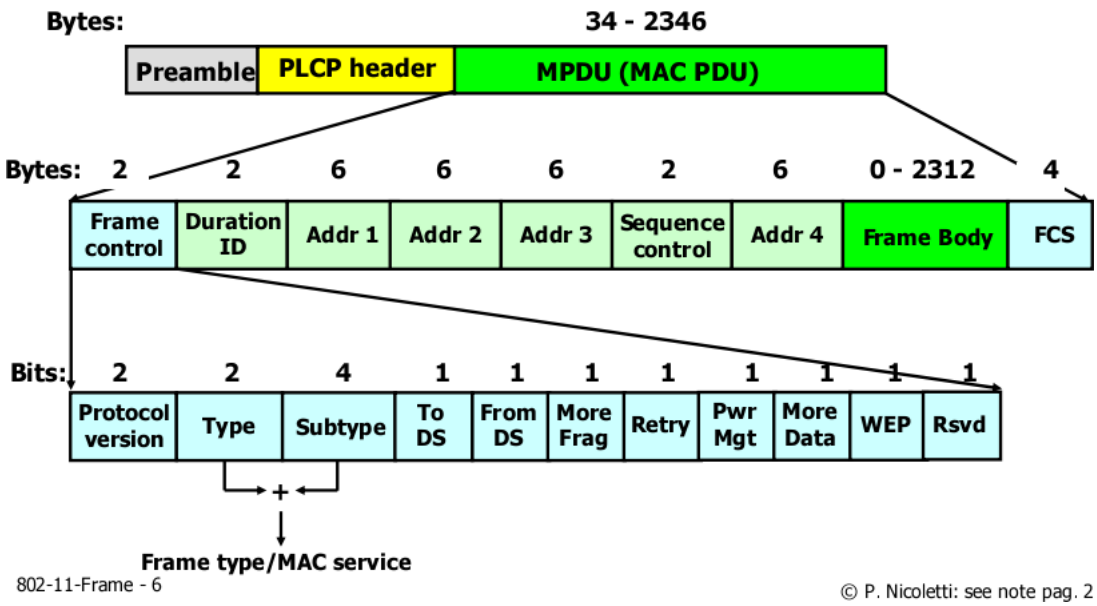


Figure 3.1: 802.11 frame

Testing found that some of the MAC addresses captured from these locations corresponded to those of known devices, but there were still some spurious MAC addresses.

Research found that the cause of these was that some devices, including iOS and Android devices, randomise their MAC addresses. In order to prevent tracking these devices randomise the MAC address sent in WiFi probes, (packets sent out to determine what WiFi hotspots are nearby) (Schauer L., 2014). This generates many spurious MAC addresses as it is changed for each probe. As the devices only needed to track residents of the home, all MAC addresses not in the list of known devices were ignored. The limitation of this is that these WiFi devices can only be detected when they are connected to a WiFi network (not necessarily the same network as the device) however it is not limited to known WiFi devices, as the MAC address is fixed once connected to WiFi and probes can be identified and ignored by looking at the type and subtype bits seen in Figure 3.1.

Finally, the espressif (manufacturers of the ESP8266) sdk does not allow switching from promiscuous mode to station mode (normal WiFi operation) during operation. This problem was found during the development of the sniffing software when the device would crash when WiFi connection functions were called after having sniffed WiFi. In order to bypass this restriction and switch between modes, a flag had to be added to the EEPROM before calling the reset function, to allow the device to start in station mode, check the configuration for WiFi credentials, and if they are present attempt to connect to WiFi and send measurements to the server. All captured WiFi device information also had to be stored in the EEPROM, so it could be retreived and sent when the device restarted.

Chapter 4

The Server

4.1 The database

An SQLite3 database was created to store the information from the sensors. The database was designed to reflect that every sensor belonged to a property, and every user had a sensor in their bedroom. The initial design is as shown in Figure 4.1a and was used during prototyping. Maintaining the data in an SQL database allowed data changes to be rolled out easily, by reducing replication of data as well as maintaining a strict structure which made managing the data easier during analysis.

After implementing user feedback, device registration and fetching outside temperature, additional tables were added to the database as seen in Figure 4.1b.

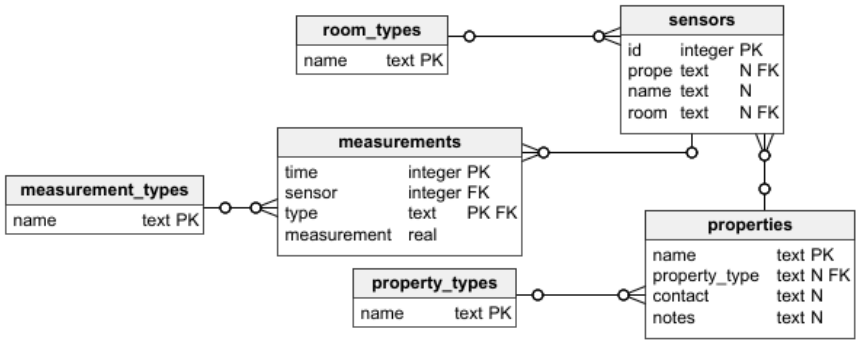
4.2 The daemon

A daemon was written to receive packets from the sensors and put it into an SQLite3 database. Packets received from unregistered sensors are ignored and a record is placed into the unregistered sensors table seen if Figure 4.1b. A management program allows registration of sensors into the database as described below.

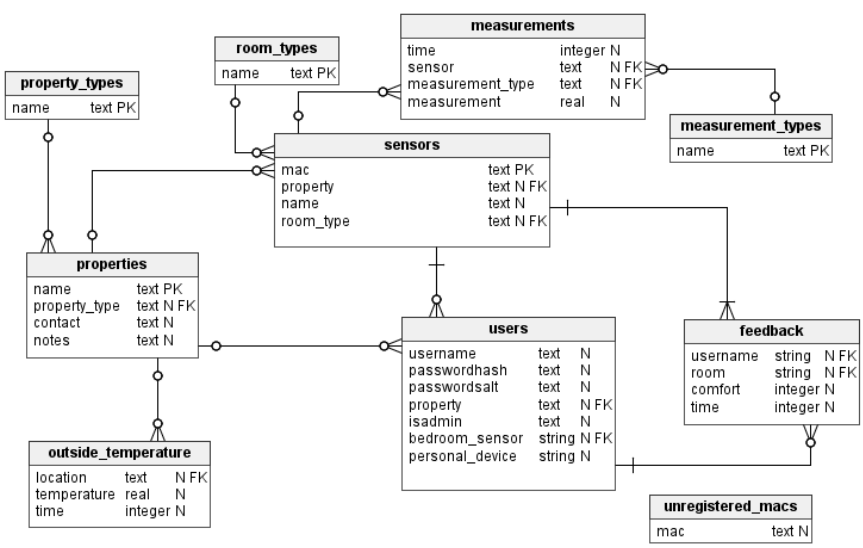
4.3 Adding a device to the database

In order to minimise the amount of configuration for each device, the device sent its MAC address with each packet and information about the device was stored on the server.

In order to add a device to the system it had to be given WiFi credentials using the process detailed in section 3.1, then added registered in the database. Devices were registered in the database using a management script which presented options for what



(a) Initial database design



(b) Final database design

property it should belong to, what type of room it was in and what MAC address to use (from the unregistered MACs table).

4.4 The web server

An Apache web server was used with Python cgi scripts to design a simple user system. Passwords were salted and hashed using sha256 and stored in the database. A webpage allowed users to insert feedback into the database detailing the room they were in and if they were at a comfortable temperature. A seperate web interface allowed administrator users to add new users.

Chapter 5

The data

5.1 Selection of candidate properties

Three properties were purposefully selected. The properties, known as A,B and C, were comprised of one large shared flat with 5 residents, one smaller shared flat with 3 residents and one large shared house with 2 residents respectively. All the properties were shared properties as tracking individual users was important. Consideration was given to include small and large properties with different numbers of residents and both flats and houses in order to ensure the results were generalisable.

Although a larger number of properties may have given more reliable and accurate data it was considered to be beyond the scope of the project due to cost and time constraints.

5.2 Interruptions in the data

The data was largely interrupted for the duration of the experiment, with the exception of one period at Property C where the internet connection was disrupted for approximately 10 hours on day 13. This affected all sensors at the property. This can be seen as a straight line connecting the data points in Figure 5.2. The graphs of light and temperature data from all properties showing continuous collection of data throughout the trial period can be found in the appendix.

5.3 Graphing the data

A Python script was written to generate graphs for any sensor device for a variety of attributes including light, internal temperature, motion, network device presence and external temperature using SQL queries. The option to display as a daily average was given, to allow daily patterns to be seen. All the graphs were generated in this manner.

5.4 Motion data

By graphing the motion data it was seen that values were reliably transmitted for the duration of the experiment. An example plot is shown in Figure 5.1 of the daily average hourly motion values. The data can be seen to be cyclic with the room being used more at certain times of day, notably 18:00 for this property. It is clear that the device's motion sensor is not being triggered at night, so it appears the motion data did correspond to the residents' activity.

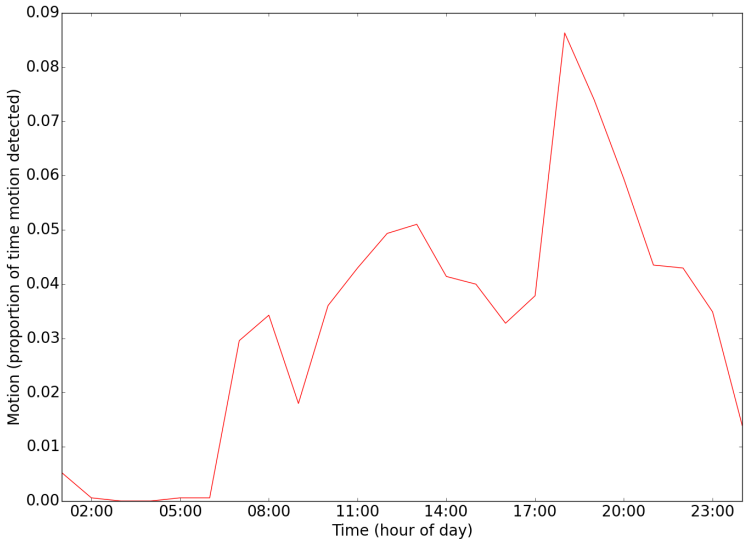


Figure 5.1: Property C shared space averaged daily motion data

5.5 The temperature data

The temperature data was also successfully captured for the duration of the experiment. The high spikes in temperature were initially though to be erroneous values, but on closer inspection formed a rising and falling curve more consistent with a thermostat being set very high. The temperature for the primary shared spaces of all 3 properties can be seen in Figure 5.2. The individual graphs can be seen in the appendix.

5.6 The light data

The light data was also successfully captured for the duration of the experiment. The light data showed both gradual inclines cycling daily, and step changes indicating that both daylight and artificial light were being captured. It was hoped that by including both light and time of day as features in the data, light level at a given time of day may

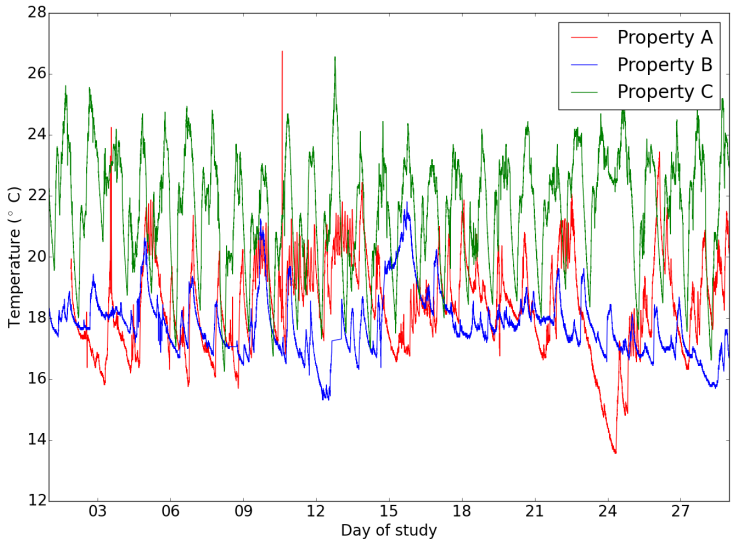


Figure 5.2: Temperature data for shared spaces in all 3 properties

indicate if the lights were on in a given room, which may correlate to user activity in that room.

5.7 The personal device presence data

The data for when users' personal WiFi devices were detected was graphed as number of times detected in total for a given hour of the day, an example is shown in Figure 5.4 for Property C. The data in this graph shows a clear daily pattern. Device 1 was owned by a participant who was self employed, so spent most of their time at home and had no particular times where they were regularly out. Device 2 was owned by someone with a full time job away from the property, and this can be seen by the fact that between the hours of 09:00 and 17:00 their device was detected around a quarter as many times as around 07:00. This shows that personal device presence can be used to determine when users are usually home.

5.8 Finding a comfortable temperature for each user

When trying to analyse how important users preferences were, the most important step was measuring the temperature at which the participants were most comfortable. The participants were given access to a web interface where they could self report if they were comfortable. The user was asked to give the room they were in, and whether they were at a comfortable temperature.

To find the comfortable temperature for each user, all the self reports where they were comfortable were found, then cross-referenced with the temperature readings from that

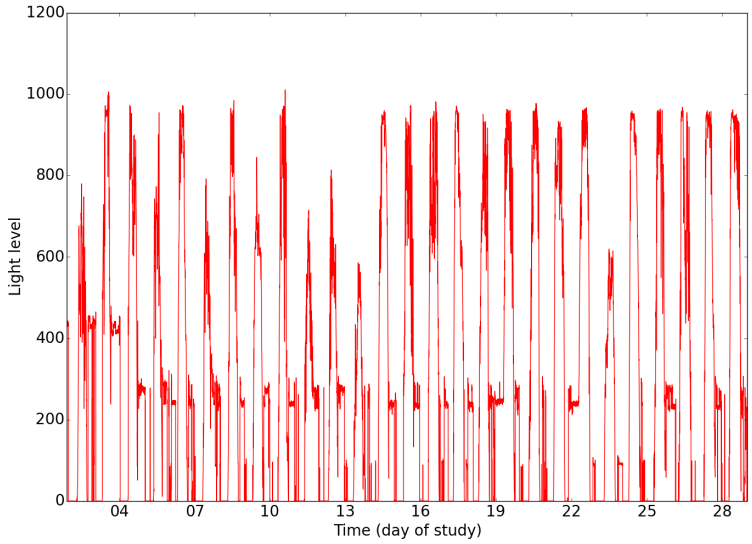


Figure 5.3: Light data for Property A shared space

room around that time. An average is taken from all the readings from 5 minutes before and after they submitted data on comfort level, in order to get an accurate idea of the temperature in the room.

The comfortable temperatures varied from 15.64-21.13°C, a difference of 3.89°C as seen in Tables 5.1-5.3. Such a large temperature difference has potential for a difference in energy consumption and comfort. As different participants within the same properties were comfortable at different temperatures an ideal smart heating control system would change the temperature of the property depending on who was inside.

The outside temperatures at the residents homes were then examined, using data collected through the Wunderground API with a Python script. An average was taken throughout the trial period (month of February) during active hours (08:00-23:59). In the tables below proportional difference of two temperatures 'a' and 'b' with an outside temperature 'c' is $(a-c)/(b-c)$ where $b > a$ so represents a percentage reduction in temperature difference. Heat dissipation is proportional to temperature difference, so a reduction in the temperature difference reduces the energy required to maintain that temperature.

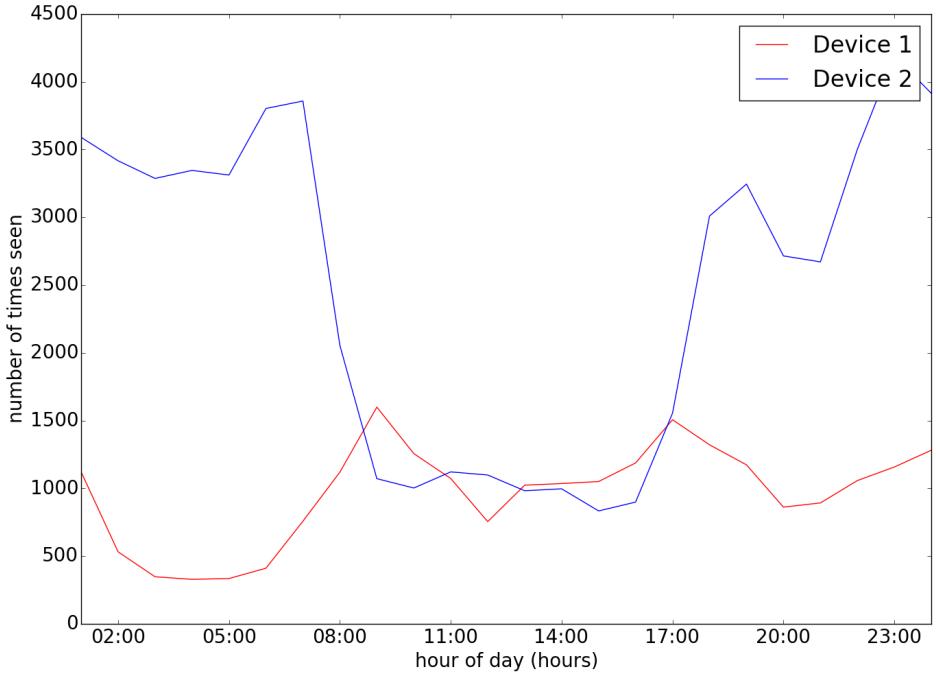


Figure 5.4: WiFi device presence by hour in property C as an average of all days in trial period

Participant 1	18.88°C
Participant 2	20.43°C
Participant 3	17.58°C
Participant 4	18.35°C
Participant 5	17.44°C
Average daytime outside temperature	6.19°C
Biggest proportional difference	21.6%

Table 5.1: Comfortable temperatures for property A

Participant 1	21.06°C
Participant 2	18.79°C
Participant 3	15.64°C
Average daytime outside temperature	6.19°C
Biggest proportional difference	37.4%

Table 5.2: Comfortable temperatures for property B

Participant 1	18.74°C
Participant 2	21.13°C
Average daytime outside temperature	6.59°C
Biggest proportional difference	16.4%

Table 5.3: Comfortable temperatures for property C

5.9 Where the participants are

5.9.1 K nearest neighbours

In order to use KNN, it was essential to transform the data first. As the lighting and temperature data (continuous) were on separate scales from the WiFi device presence data(binary), and motion data (binary) the lighting and temperature data were normalised by subtracting the minimum value and dividing by the range. This is to prevent features with higher variance having a greater effect on the classification and was necessary for the Euclidean distance measure used.

In order to validate the data 5 fold cross validation was used. The data was divided into 5 equal sections for each user, and 4 sections were used for the training data, with the remaining section for the validation data. This was done for all permutations of the 5 sections.

KNN was used for two problems:

- Deciding if a user was in or out
- Deciding which room a user was in

5.9.1.1 Features

The features used for each datapoint are:time of day, day of week, motion (for each room), light level(for each room) and network device presence (of the user being classified). Temperature was not used as it may create a feedback loop if it is used to detect presence in a system that controls temperature. For example if higher temperature rooms are classified as occupied they would be heated, then continued to be classified as occupied and continually heated.

5.9.1.2 Choosing K

For both classifying a user as in or out and for identifying their exact location k values from 1-20 were used. The accuracy was plotted as shown below in Figure 5.5. The k value that maximised accuracy was chosen for each. In the case of determining whether a user was in or out the k value was 14. In the case of determining which room a user was in the k value was 15.

5.9.2 Deciding if a user was in or out

Using KNN, the system was able to tell if each user was in or out with accuracies ranging from 61.81% to 90.16%, with an average accuracy of 79.3% (see table 5.4). The differing accuracies in classifying users may be due to some users more often having their phone connected to WiFi and on their person, giving a higher correlation

	In	Out
In	304	043
Out	056	082
Accuracy	**79.31%**	

Table 5.4: Confusion Matrix for All Participants

between device presence and presence in the property. However, it appears it is not noticeably more difficult to track users in larger properties or properties with more residents, as can be seen in Table 5.5.

Property	**Participant**	**Accuracy**
A	1	74.0%
A	2	75.4%
A	3	83.0%
A	4	78.2%
A	5	84.4%
B	6	73.4%
B	7	89.6%
B	8	72.6%
C	9	75%
C	10	80.3%

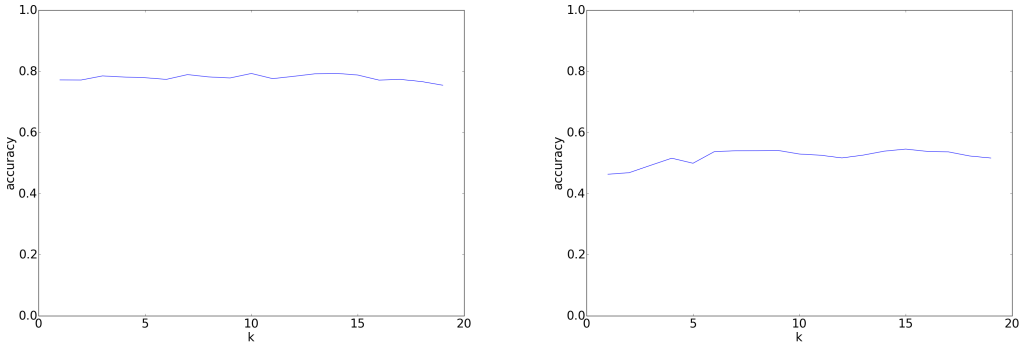
Table 5.5: Accuracies of classifying all users

5.9.3 Identifying user location within a property

Identifying the users' location between multiple rooms or out of flat was significantly less accurate. In table 5.6 it can be seen that it could do equally well by always classifying as bedroom. This may be due to having to little validation data for each room, and may be improved by a longer running experiment.

	Classed as bedroom	Classed as living room	Classed as out
Bedroom	27	0	3
Living room	12	0	1
Out	11	1	0
Accuracy	**50%**		

Table 5.6: Confusion Matrix for a randomly selected Participants



(a) Choosing k for classifying if a user is in or out

(b) Choosing k for classifying which room a user is in

Figure 5.5: k vs classification accuracy

Chapter 6

Discussion

This project aimed to design, fabricate and test a device to collect user data in relation to home heating patterns. The findings of this project, outlined in the previous section, suggest several areas for discussion, relating to success in the design, fabrication and cost of the device, successes and limitations in data collection, and the potential for such a device to be developed into a commercial product.

6.1 Design and fabrication of the device

The device used in this project was designed to collect data on temperature, motion detection, lighting level, and WiFi device presence, which would then be sent to a central server for storage and later analysed with regard to heating patterns. The design and fabrication of the device was largely successful and the device operated as intended to, as evidenced by the largely uninterrupted data collection, agreement of sensors in the same vicinity and tests of sensor components as described in the methods section. Although the data was largely uninterrupted, the device could be improved by having its own power source (for example, a rechargeable battery) and by caching readings it is unable to send and then sending the readings when a connection becomes available. A version of the system using a server that operates on a local network could also be offered which would remove the dependence on an internet connection and give users more control of their data. However, the use of an external power supply and a WiFi connection seemed practically appropriate for the scope of this project.

6.2 Cost of the device

The device fabricated in this project was intentionally designed to be low-cost, particularly in relation to smart home heating control systems currently marketed for domestic use. The project aimed at its outset for a cost of £10 or less per device and the actual cost of the each device was £15.98, more than 50% over the target price. The increase in cost was partially due to the low quality of cheaper products which were initially

thought to be suitable for the project but which did not provide reliable and accurate temperature readings. Thus slightly more expensive temperature sensor components were sourced in order to provide a fully functional device at a higher cost. However, as the initial price per device was modest in comparison to similar options now on the market (for example, £58.99 per sensor for the Evohome by Honeywell; Evohome Shop, 2017), the final price of £15.98 per item does not seem unreasonable. It should also be noted that devices sold to consumers likely benefit from lower manufacturing costs due to being produced in greater quantities.

However, following this, it should be noted that additional costs would also need to be considered if the device were to be produced at scale for a consumer market. The prototype developed in this project did not include any protective casing around the circuit board and other implements. In order for the device to be marketable to consumers, an injection moulded case would be required to protect the product and make the device attractive for placement within the home. A printed circuit board would be also required for manufacturing at scale. If the system was used to control the a heating system, valve actuators and associated software to connect to the central heating system would be needed. Although the current system used SMS technology with relative success to assess whether participants were comfortable with the current room temperature, a more polished system would be required if marketed to consumers, such as an app on which users could record their comfort with the current temperature as well as change the heating controls on an on-demand basis. A wall-mounted control panel may also be required to supplement the app for temperature controls, though this would not necessarily be essential to the overall functioning of the system. It is difficult to assess what the overall costs of these additional components would be in order to develop a marketable product.

6.3 Data analysis and implications for a smart heating control system

The analysis showed some success in determining whether users were in or out of the property, averaging 77.1% accuracy but going as high as 90.16% for one user. This is possibly due to some users carrying their phone more often, making them easier to track, turning the lights off when they leave a room, and the motion sensor in their room being better placed. The data showed that users were most often in their bedrooms, a finding which was validated by the SMS data submitted by users. This suggests implementation potential in homes which have multiple zones, where individual bedrooms could be automatically heated according to the preference of the user typically occupying that space. The system was less successful at predicting when shared spaces were in use; this is likely because users were most often in their bedrooms and there was not a lot of validation data for the shared spaces. The system could be refined to more accurately guess when users would be in shared spaces if more data was available, however it was not practical to prompt users more than 3 times per day to avoid putting an unreasonable burden on them.

The data analysis also showed some success in determining users' daily usage patterns using presence of their network device. However, if an associated app could be developed to supplement the device which could remind users at more frequent intervals to log their location in the property, or if more frequent SMS data was collected on the same topic, then it may have been possible to further refine prediction for when the users would be home and where they would be within the property. This would be necessary in order to control a heating system in such a way that heating turns on and off at the appropriate times in the appropriate places; for example, if it could be predicted that a user who prefers a temperature of 17°C typically comes home at 6pm, the heating system could be automatically switched on at 5:45pm to account for this. As it stands, the device would need further refinement in its prediction ability before it is reliable enough to connect to a home heating system for domestic use; otherwise, there stands the risk that heating energy could be wasted and users could be uncomfortable should the heating be turned on and off at inappropriate times or set to inappropriate temperatures.

Finally, the analysis showed significantly different preferred temperatures for each user by taking the average of temperatures in the rooms users were in at the times when users reported that they felt comfortable. This suggests that a personalised heating control system could be used to save on energy, as zones could be heated to lower temperatures when people who prefer cooler temperatures are the only people in the zone. When more than one person is in a room (for example, in shared spaces), the prediction algorithm suggests an overall preferred temperature by taking the average of each of the individual preferred temperatures for all users currently in the room. While this seems an appropriate way to compromise between the preferences of individual users, it is possible that the suggested temperature for shared spaces may not be ideal for all users, but some compromise is necessary.

6.4 Limitations of the project

Although the device functioned as intended with regard to data collection, there were some challenges in this project that may have hindered the accuracy or generalisability of the results. In particular, the use of SMS messages to extract data from users around comfort level and room occupancy were not always successful as users only responded to messages 67.3% of the time. SMS messages were employed for practical reasons as cell phones are widely available (and used by all participants), though the limitations of soliciting user feedback in this way means that there were some missing data points when participants did not respond, whether due to personal choice or technological limitations (for example, an uncharged phone). Although the data could still be analysed by omitting points where there was no response, this could be improved upon in future iterations of the device by using an app which could prompt and remind participants to respond, perhaps limiting gaps in user response data. It should also be noted that user feedback was not solicited by SMS at the same time of day for every user. (sent three times randomly between 8am and midnight every day in order to get validation different times of day without posing significant irritation to participants by

asking them too many times per day). This means there was no validation data for multiple users in a room at the same time.

Additionally, due to practical limitations around the length of the dissertation project, data was only collected for one month. Data was collected during the month of February, which seemed appropriate for the use of the device (to predict changes in home heating patterns), although interesting and informative data could have been collected around seasonal heating patterns had data collection been extended to include the spring, summer or fall months. It may have also been useful to collect home energy consumption data in order to correlate this with user data, such as preferred temperature and room occupancy. This could provide useful information around the impact of a smart heating control device in relation to energy usage and, by extrapolation, carbon emissions, to provide clearer evidence around the impact of smart home heating control systems on the environment, as discussed in the introduction.

6.5 Potential for development into a commercial product

Despite limitations in the data collected in this project, the results still suggest some commercial potential for this device once additional measures have been taken to improve the reliability and specificity of prediction for home heating patterns. As discussed above, the device was more successful at predicting the occupancy of the property as a whole but less successful at predicting the occupancy of specific rooms. A viable commercial product would need to improve upon this discrepancy in order to provide a reliable, high-quality service which is able to compete with existing models of smart home heating control systems. Further user testing, including the development and testing of an associated app for user temperature feedback, would be needed to be sure that prediction patterns are reliable and to tease out any issues with the prediction algorithms which have not been obvious in this project. Additional modifications to the device and the installation of valve actuators, as discussed in the cost section previously, would be required to create a user-friendly and fully functional smart home heating control system product which is financially competitive with current consumers models. This would also include the development of software to connect the device to valve actuators and user testing to ensure that valve actuators work correctly. However, as it stands, the device in this project is able to, at a relatively low cost, sense and predict with some reliability the time at which individual users are home, their preferred temperature and location, which is a novel feature compared to current market models which are unable to differentiate between individual user preferences.

Chapter 7

References

1. Dounis, A. I., & Caraiscos, C. (2009). Advanced control systems engineering for energy and comfort management in a building environmentA review. *Renewable and Sustainable Energy Reviews*, *13*(6), 1246-1261.

2. Liao, Z., Swainson, M., & Dexter, A. L. (2005). On the control of heating systems in the UK. *Building and Environment, 40*(3), 343-351.

3. Department of Energy and Climate Change. 2010 to 2015 government policy: household energy. (2015, May 5). Retrieved March 31, 2017, from `https://www.gov.uk/government/publications/2010-to-2015-government-policy-household-energy/2010-to-2015-government-policy-household-energy`

4. UK Communities and Local Government.*Domestic Heating Compliance Guide (Rep.)* (2008, December). Retrieved March 31, 2017, from UK Communities and Local Government website: `http://webarchive.nationalarchives.gov.uk/20151113141044/http://www.planningportal.gov.uk/uploads/br/br_pdf_ptl_domheat.pdf`

5. European Commission. Smart grids and meters. (2017, March 31). Retrieved March 31, 2017, from `https://ec.europa.eu/energy/en/topics/markets-and-consumers/smart-grids-and-meters`

6. Ngele, F., Kasper, T., & Girod, B. (2017). Turning up the heat on obsolete thermostats: A simulation-based comparison of intelligent control approaches for residential heating systems. *Renewable and Sustainable Energy Reviews.*

7. Utley, J. I., & Shorrock, L. D. (2008). Domestic energy fact file (2007): England, Scotland, Wales and Northern Ireland. BRE Report. `http://www.bre.co.uk/filelibrary/pdf/rpts/countryfactfile2007.pdf`

8.Lyric T6R 7 Day Wireless Programmable Thermostat - Honeywell UK Heating Controls. (n.d.). Retrieved March 31, 2017, from http://www.honeywelluk.com/products/Connected-Thermostats/Wireless-Connected-Thermostats/Lyric-T6R-7-Day-Wireless-Programmable-Thermostat/

9. Nest Learning Thermostat. (n.d.). Retrieved March 31, 2017, from https://store.nest.com/uk/product/thermostat?selectedVariantId=T3010GBBI

10. Heatgenius Products. (n.d.). Retrieved March 31, 2017, from https://www.geniushub.co.uk/products/

11. Shaikh, P. H., Nor, N. B. M., Nallagownden, P., Elamvazuthi, I., & Ibrahim, T. (2014). A review on optimized control systems for building energy and comfort management of smart sustainable buildings. *Renewable and Sustainable Energy Reviews, 34,* 409-429.

12. Honeywell Connected Single Zone Thermostat... (n.d.). Retrieved March 31, 2017, from https://theevohomeshop.co.uk/en/38-honeywell-wireless-thermostats

13. Nicoletti, P. (2005, June). IEEE 802.11 frame format. Retrieved March, 2017, from http://www.studioreti.it/slide/802-11-Frame_E_C.pdf

14. Jing, S., Li, B., Tan, M., & Liu, H. (2013). Impact of relative humidity on thermal comfort in a warm environment. Indoor and Built Environment, 22(4), 598-607.

15. Liu, T. (2008, January). Digital-output relative humidity & temperature sensor/module . Retrieved February, 2017, from https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf

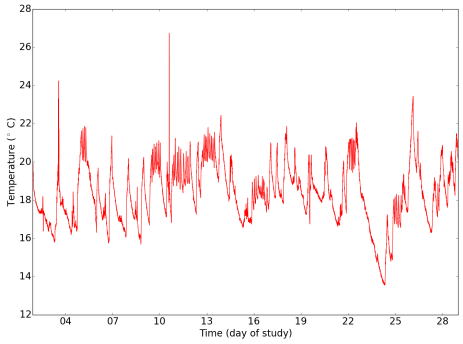
16. Dallas Semiconductor. (2008, January). DS18B20-PAR 1-Wire Parasite-Power Digital Thermomete r. Retrieved February, 2017, from D. (2008, January). DS18B20-PAR 1-Wire Parasite-Power Digital Thermometer. Retrieved February, 2017, from https://datasheets.maximintegrated.com/en/ds/DS18B20-PAR.pdf

17. Schauer, L., Werner, M., & Marcus, P. (2014, December). Estimating crowd densities and pedestrian flows using wi-fi and bluetooth. In Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (pp. 171-177). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

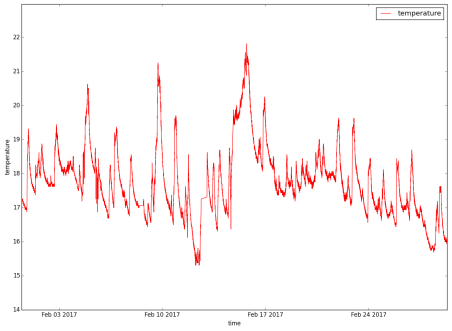
Chapter 8

Appendix

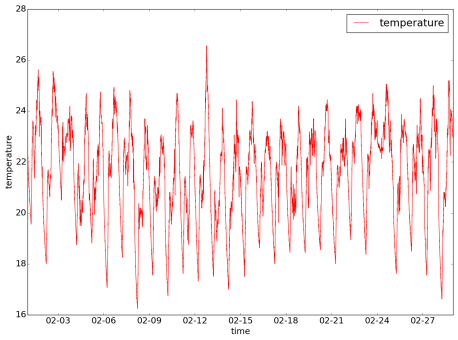
Some additional graphs to show the data was uninterrupted for the length of the trial period have been provided. Only data for the primary shared space is included per property, as including 13 graphs showing data for each individual sensor for each of the four variables tested seemed impractical, as there would be 52 graphs required. Shared spaces were thought to the most comparable space between properties. Additional graphs can be generated using plot.py in the analysis/graphing folder of the submitted repository.



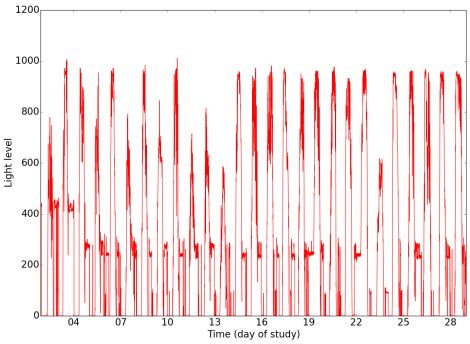
(a) Property A temperature



(b) Property B temperature

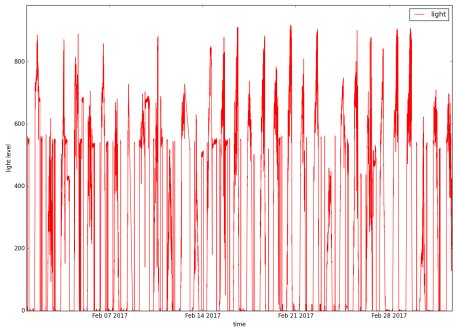


(c) Property C temperature

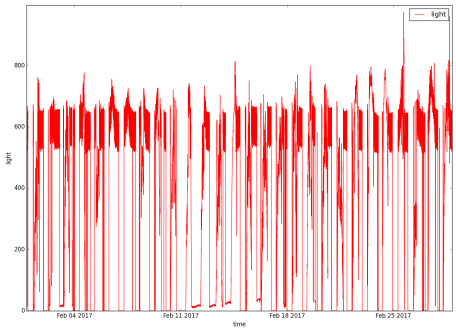


(a) Property A light

Figure 8.2: Property A light



(a) Property B light



(b) Property C light