# HOME AUTOMATION USING EMBEDDED SYSTEMS

Hannes Venter | 44908903

MACQUARIE University
SYDNEY · AUSTRALIA

# Table of Contents

## Document Version

| Version | Date | Description | Author |
|---|---|---|---|
| 0.1 | 18/3/20 | Initial thoughts on CPD project. Began a list of dot-points with ideas for CPD projects. | Hannes Venter |
| 0.2 | 19/3/20 | Created Title Page and TOC. Further filled in ideas for CPD project. | Hannes Venter |
| 0.3 | 20/3/20 | Selected Final Idea. Wrote topic and motivation. | Hannes Venter |
| 0.4 | 22/3/20 | Finalised project timeline and first ideas about how to achieve the project. | Hannes Venter |
| 1.0 | 22/3/20 | Finalised version 1.0 | Hannes Venter |
| 1.1 | 25/3/20 | Taking feedback into consideration and improving formality of plan. | Hannes Venter |
| 1.2 | 25/3/20 | Moving Initial Plans and Further Motivation into Appendices | Hannes Venter |
| 1.3 | 26/3/20 | Further explanation of hardware, software and knowledge | Hannes Venter |
| 1.4 | 27/3/20 | Research and selection of hardware and software | Hannes Venter |
| 1.5 | 28/3/20 | Purchase and description of hardware and software | Hannes Venter |
| 2.0 | 29/3/20 | Finalised version 2.0 | Hannes Venter |

# 1. Introduction

This project will consist of 2 smaller projects, both centred around home-automation. The projects will both be embedded systems and will vary in complexity as they progress. Appendix I provides explanations of additional project considerations.

## 1.1 Ceiling Vents

The first project will create a system which draws hot air out of the ceiling, thus cooling the house. A fan will be placed inside the ceiling space with its exhaust directing outwards through a vent. The fan will be enabled once the temperature inside the ceiling reaches above a certain temperature and will run until the ceiling is sufficiently cooled. Light sensors will be implemented to alter the power settings and ensure that the fan is quiet during the night.

## 1.2 Automatic Blinds

The second project will be more complex. This project will enable external blinds to be gradually lowered in front of windows as the sun sets. An array of light sensors and temperature sensors will be used to track the location of the sun. The system will communicate with existing electrical blinds to be lowered with the setting sun. This will ensure that the room remains cool during summer and reduces the glare of the sun throughout the room. Additional sensors could be used to raise the blinds during strong winds to prevent any damage.

# 2. Motivation

Motivation for this project arose after reflecting on the multitude of software engineers which exist. Through thought and research, I tried to determine the type of software engineer that I want to be. Embedded systems were found to be most interesting. The aspect of problem solving and creating projects with physical interactions have always been particularly enjoyable. Reflection on the work completed in ENGG200 and ENGG300 instigated the initial idea to undertake a project on embedded systems. There is a significant amount of knowledge and experience that can be personally gained in this field. These projects were thus chosen with a goal of self-improvement, above and beyond the endeavours completed in ENGG200 and ENGG300. Appendix II contains further explanation of motivations for this topic.

# 3. Research

## 3.1 Hardware

### 3.1.1 Processor

The computing power for the project will be provided by a microprocessor. The initial two selections were between Arduino and Raspberry Pi.

A Raspberry Pi is a fully functional computer, usually with a Linux operating system, and the ability to run multiple programs. [1] [2] Arduino is a microcontroller, which is just a single component of a computer. [2] A microcontroller is a simple computer that can run one program at a time, over and over again. [1]

### 3.1.2 Sensors

The sensors and other components will be selected once a processor has been chosen. These will include temperature sensors, light sensors and wind speed sensors.

### 3.1.3 Additional Hardware

- Fan/Vent
- Relay which can be triggered to supply power to the fan.
- Resistors
- Electronic Blinds
- Remote Control to control blinds
- Wiring capable of handling the different power requirements.

## 3.2 Software

Raspberry Pi runs on a Linux operating system. [2] Ubuntu is a free Linux operating system which allows additional software and programs to be installed.

Arduino uses its own language and IDE. The Arduino language is in many ways a simplification of C++. [3]

## 3.3 Knowledge

A vast amount of additional knowledge will be required, regardless of which hardware is selected. Initial research will involve determining which processor to select for the control of the system. Additional sensors and components must then be selected to ensure a coherent match with the chosen microprocessor. Once a processor is chosen, a significant amount of research will be conducted into the software which manages it.

This project will require a remarkable amount of hardware knowledge as it is an embedded system. Electrical and computer engineering research will be conducted to maximise safety and minimise any possible damage to the components.

# 4. Project Plan

## 4.1 Microcontroller/Processor

This project will not require tremendous amounts of processing power. Arduino was selected due to its ease of use, lower price and reduced processing power. The choice of Arduino hardware was narrowed to the Arduino Uno, Arduino Leonardo and Arduino Mega 2560.

### 4.1.1 Arduino Uno

The Arduino Uno "has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analogue inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button." [4]

### 4.1.2 Arduino Leonardo

The Arduino Leonardo "has 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, a power jack, an ICSP header, and a reset button." [5]

### 4.1.3 Arduino Mega 2560

The Arduino Mega 2560 "has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button." [6]

### 4.1.4 Selected Microcontroller

The Arduino Uno will be the microcontroller board used in this project. It has sufficient processing power and memory and has enough input/output pins for the requirements of this project. It is also the most inexpensive and compact out of the three Arduino boards.

## 4.2 Software

Thus, C++ will be used as the primary language for this project. Using this language will overcome the limitations of the Arduino language, and enable the functionality of custom-made libraries.

The goal of this project is to program as professionally and efficiently as possible. Libraries will reduce the amount of code required in the main files. It will also allow for OO Programming which will enable multiple sensor objects to be created and controlled using a single library.

## 4.3 Sensors and Additional Hardware

### 4.3.1 Ultraviolet Sensor Module

The XC-4518 Ultraviolet Sensor was selected to detect the location of the sun. An ultraviolet sensor was chosen over an ordinary light sensor as its readings would be significantly more accurate in detecting sunlight. "This module will measure the UV light and adjust the output voltage depending on the UV intensity." [7] It is "capable of measuring over a wide spectral range of 200nm-370nm", has an output voltage 0-1200mV and a current 0.06-0.1mA. [7]

### 4.3.2 Temperature Module

The XC-4520 Temperature and Humidity Sensor Module was selected to measure the temperature in the different locations. It was the only temperature sensor available, however its additional humidity sensor will be valuable. It has a temperature range of 0 ºC - 50 ºC (+/- 2 ºC) and a humidity range of 20 – 80% (+/- 5%). [8]

### 4.3.3 Wireless Transmitter

The ZW-3100 Transmitter Module was selected to communicate with the existing electrical blinds. This is a pre-built 433MHz wireless transmitter which features ASK encoding. [9] It has an output power of 3dBm, requires 3V and 10mA max inputs, and has a data rate of 300bps to 10kbps. [9]

## 4.4 Timeline

This project has a timeline of six weeks. Initial work will begin with research as to the most appropriate components for the project, which will be followed by research into the selected hardware and software. The next will be prototyping and testing, before building and integrating the components into the larger system. The final stage will be the presentation. Figure 1 provides an overview of the project over the dedicated six weeks.

| | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 |
|---|---|---|---|---|---|---|
| Research | X | X | X | X | X | |
| Project Planning | X | | | | | |
| Acquiring Components | X | X | | | | |
| Small Scale Prototyping | | X | | | | |
| System Protoyping | | X | X | | | |
| Testing | | | X | X | X | |
| Building | | | | X | X | |
| Completion | | | | | | X |

*Figure 1 - Gantt Chart for 6 weeks of the project*

### 4.4.1 Week 1

Week 1 will consist of initial research into the project and any required components. The project plan will be developed, and designs will be created from preliminary ideas. Week 1 will also see the purchase of elementary components, once sufficient research has been conducted.

### 4.4.2 Week 2

Week 2 will consist of continued research and purchasing of necessary equipment. Initial prototyping will be undertaken once the essential components have been purchased.

### 4.4.3 Week 3

Week 3 will consist of continued research and further prototyping. Prototypes will become more advanced as they progress. Testing will be introduced to ensure that the various components in the prototypes are performing as intended.

### 4.4.4 Week 4

Week 4 will consist of continued research and testing. The building of the final system will begin after successful testing and performance of the prototypes.

### 4.4.5 Week 5

Week 5 will see significant progress on the build of the project. Continued research and testing will be conducted in conjunction with the build.

### 4.4.6 Week 6

Week 6 will see the conclusion of the project. Any final adjustments will be made to ensure the system is performing optimally. The system will be finalised and ready for presentation.

# 5. References


[1] P. Di Justo, “Raspberry Pi or Arduino Uno? One Simple Rule to Choose the Right Board,” 4 December 2015. [Online]. Available: https://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/. [Accessed 26 March 2020].

[2] L. Orsini, “Arduino Vs. Raspberry Pi: Which Is The Right DIY Platform For You?,” 7 May 2014. [Online]. Available: https://readwrite.com/2014/05/07/arduino-vs-raspberry-pi-projects-diy-platform/. [Accessed 26 March 2020].

[3] Arduino, “Language Reference,” 2020. [Online]. Available: https://www.arduino.cc/reference/en/. [Accessed 26 March 2020].

[4] Arduino, “ARDUINO UNO REV3,” 2020. [Online]. Available: https://store.arduino.cc/usa/arduino-uno-rev3. [Accessed 27 March 2020].

[5] Arduino, “ARDUINO LEONARDO WITH HEADERS,” 2020. [Online]. Available: https://store.arduino.cc/usa/leonardo. [Accessed 27 March 2020].

[6] Arduino, “ARDUINO MEGA 2560 REV3,” 2020. [Online]. Available: https://store.arduino.cc/usa/mega-2560-r3. [Accessed 27 March 2020].

[7] JayCar, “Arduino Compatible Ultraviolet Sensor Module,” 2020. [Online]. Available: https://www.jaycar.com.au/arduino-compatible-ultraviolet-sensor-module/p/XC4518. [Accessed 28 March 2020].

[8] JayCar, “Arduino Compatible Temperature and Humidity Sensor Module,” 2020. [Online]. Available: https://www.jaycar.com.au/arduino-compatible-temperature-and-humidity-sensor-module/p/XC4520. [Accessed 28 March 2020].

[9] JayCar, “Wireless Modules (Transmitter) - 433MHz,” 2020. [Online]. Available: https://www.jaycar.com.au/wireless-modules-transmitter-433mhz/p/ZW3100. [Accessed 28 March 2020].

# Appendix I: Initial Project Ideas

After reflection and research of what kind of SE I want to be, I decided to complete a project on embedded systems.

## NFC Door lock

Initials thoughts were to create an NFC enabled door lock that unlocks when a registered phone is tapped. An adverse realisation was that there are already too many systems that offer this and would perform significantly better. Another unfavourable insight was the extreme security requirements of this project, since it will control access to the house.

## Automatic Lights

Another thought was to create a system that controls the lights in the garden. The lights would turn on when the sun sets and it becomes dark, and then turn off when it becomes light again. This idea was disregarded as it seemed too elementary.

## Automatic Blinds

Our family room has windows on three of its walls, meaning that the afternoon sun is very prominent. Existing external blinds can be lowered and raised using a remote control. We often lower it in stages as the sun sets in the afternoon. Once the sun has set, we raise it again.

This produced the idea to create a system that uses light sensors at certain increments to lower the blinds gradually as the sun sets and raises it again once the sun has set. In winter, the sun naturally warms the room reducing the need for heaters. The light sensors could thus be combined with a thermometer, which only lowers the blinds once the room reaches above a certain temperature.

## Ceiling Vents

The final idea was to place a vent/fan in the upstairs ceiling which turns on once the temperature in the ceiling space rises above a certain threshold.

## Appendix II: Project Motivation

I have always enjoyed problem solving. Most of software engineering has some form of problem solving in it.

This past week, I have intensely thought about the type of software engineer that I want to be. I thought back about the subjects I did and which ones I enjoyed most. COMP247 stood out, however this was more because it was directly relatable my work as an IT consultant at a school. Through my work there, I have often had to change the VLANs on switch-ports, change the IP address of various things and interact with servers, routers and switches via command line interface.

Something that I've admired greatly is the new metro. I am in awe of the software and hardware that makes that system work. It would be nice to be able to know that I had a role in something like that.

To support my reflections, I decided to do some research on the different types of software engineers and what they all actually do.

I noticed the difference between a high-level developer and a low-level developer. Low-level seems to interact more with hardware and seemed interesting.

*"This is a general term for a developer who writes code that is very close to the hardware, in low-level languages such as assembly and C. Embedded developers are often low-level developers, but not always."*

The main one that really stood out to me was an embedded developer.

*"These developers work with hardware that isn't commonly classified as computers. For example, microcontrollers, real-time systems, electronic interfaces, set-top boxes, consumer devices, iOT devices, hardware drivers, and serial data transmission fall into this category.*

*Embedded developers often work with languages such as C, C++, Assembly, Java or proprietary technologies, frameworks, and toolkits."*

Both ENGG200 and ENGG300 have shown me how enjoyable it is to work with embedded systems. I really enjoy seeing physical actions from my code. It is also heavily centred around problem solving and actually working with real things.

https://www.coderhood.com/19-types-of-developers-explained/