

# Smart-Home with Solar

## Monitoring Entry 0NC

Status

/1 Hardware present?

/1 Title Page

/1 Declaration of Joint Authorship

/1 Proposal (500 words)

/1 Executive Summary

Bao Quy Diep, Nam Nguyen, June Patrick Dacaya, Adrian De  
Braga



## Declaration of Joint Authorship

We, June Patrick Dacaya, Adrian De Braga, Bao Quy Diep, and Nam Nguyen confirm that this work submitted is the joint work of our group and is expressed our own words. Any uses made within it of the works of any other author, in any form (ideas, equations, figures, texts, tables, programs), are properly acknowledged at the point of use. A list of the references used is included. The work breakdown is as follows: Each of us provided functioning, documented hardware for a sensor or effector. I and Adrian provided PLC functionalities and a Temperature sensor. Bao provided another temperature sensor. Nam provided a light sensor. In the integration effort Nam is the lead for further development of our mobile application, June and Adrian is the lead for the Hardware, and Bao is the lead for connecting the two via the Database.



## Proposal

We have created a mobile application, worked with databases, completed a software engineering course, and prototyped a small embedded system with a custom PCB as well as an enclosure (3D printed/laser cut). Our Internet of Things (IoT) capstone project uses a distributed computing model of a smart phone application, a database accessible via the internet, an enterprise wireless (capable of storing certificates) connected embedded system prototype with a custom PCB as well as an enclosure (3D printed/laser cut) and are documented via this technical report targeting OACETT certification guidelines.

Intended project key component descriptions and part numbers

Development platforms: Nucleo-F401RE running the STM32 chip, Arduino, Raspberry Pi, and Blue pill with Arduino IDE.

Sensor/Effector 1: PLC functionalities performed by NUCLEO Wi-Fi PLC Stackable components with Temperature sensor, RTD PT100 with a 4-20 ma transmitter, which is one of the most accurate temperature sensor in the market, it is also almost immune to electrical noise which makes it more suitable in an industrial environments. The PLC functions will be performed by three components: X-NUCLEO-PLC01A1 which performs basic PLC input/output functions, X-NUCLEO-OUT01A1 which is another input/output component which supports voltage of up to 24V, and an X-NUCLEO-IDW01M1 which supports internet connectivity through a Wi-Fi connection.

Sensor/Effector 2: MCP9808 is an I2C temperature sensor which will be used to monitor the temperature indoor. It is one of the more precise temperature sensors in the IoT

world.

Sensor/Effector 3: VEML 7700 ambient light sensor which runs in Arduino. A light sensor that will be used to monitor luminosity indoor/outdoor.

Sensor/Effector 4: RPI 8MP CAMERA BOARD is an 8 mega-pixel high resolution camera which will be use to record/snap images of the home's surroundings.

We will continue to develop skills to configure operating systems, networks, and embedded systems using these key components to create a smart house that will have a solar panel activity monitoring as well as to monitor in-house temperature, security system, and a unique user base data available through a cloud base database. The Wi-Fi PLC component will be used to monitor the solar panels in conjunction with the RTD P100 outside a smart house. The temperature sensor will be used to manage the house temperature and the light sensor will be used to monitor and control the lights installed in the house. All the functionalities of the smart house will be monitored through an android mobile application which users can download. The user's mobile application will have the eyes and controls for the smart house.

Our project description/specifications will be reviewed by, Kim Huynh from Alpha Lab, ideally an employer in a position to potentially hire once we graduate. They will also ideally attend the ICT Capstone Expo to see the outcome and be eligible to apply for NSERC funded extension projects. This typically means that they are from a Canadian company that has been revenue generating for a minimum of two years and have a minimum of two full time employees.

The small physical prototypes that we build are to be small and safe enough to be brought to class every week as well as be worked on at home. In alignment with the

space below the tray in the Humber North Campus Electronics Parts kit the overall project maximum dimensions are  $12 \frac{13}{16}'' \times 6'' \times 2 \frac{7}{8}'' = 32.5\text{cm} \times 15.25\text{cm} \times 7.25\text{cm}$ .

Keeping safety and Z462 in mind, the highest AC voltage that will be used is 16Vrms from a wall adapter from which +/- 15V or as high as 45 VDC can be obtained.

Maximum power consumption will not exceed 20 Watts. We are working with prototypes and that prototypes are not to be left powered unattended despite the connectivity that we develop.





## Executive Summary

This document will guide the future investors about our Smart-House Project. The project will be smart house that will be control just by using an android mobile application. Users will have their own personal data securely stored in the cloud while having a complete access to it. Users will be able to monitor their solar panels, doors, indoor temperature, and surroundings with just a tap in the app. The mobile application is their window to their house while being at work or at home. The smart house project will be complete in conjunction with the hardware components such as temperature sensors, light sensors, motors, and camera. The smart house project aims to be a complete, secure, and a mobile project. The main goal for this project is to have a secured, automated, and accessible for the users.



## Contents

|  |    |
|--|----|
| Declaration of Joint Authorship .....    | 3  |
| Proposal .....                           | 5  |
| Executive Summary .....                  | 9  |
| List of Figures.....                     | 13 |
| 1.0 Introduction .....                   | 15 |
| 1.1 Scope and Requirements.....          | 15 |
| 2.0 Background .....                     | 19 |
| 3.0 Methodology .....                    | 23 |
| 3.1 Required Resources .....             | 23 |
| 3.1.1 Parts, Components, Materials ..... | 23 |
| 3.1.2 Manufacturing .....                | 25 |
| 3.1.3 Tools and Facilities .....         | 27 |
| 3.1.4 Shipping, duty, taxes .....        | 29 |
| 3.1.5 Time expenditure .....             | 29 |
| 3.2 Development Platform.....            | 31 |
| 3.2.1 Mobile Application .....           | 31 |
| 3.2.2 Image/firmware .....               | 34 |
| 3.2.3 Breadboard/Independent PCBs .....  | 35 |
| 3.2.4 Printed Circuit Board .....        | 37 |

|   |    |
|---|----|
| 3.2.5 Enclosure .....                       | 38 |
| 3.3 Integration .....                       | 39 |
| 3.3.1 Enterprise Wireless Connectivity..... | 40 |
| 3.3.2 Database Configuration.....           | 40 |
| 3.3.3 Security .....                        | 40 |
| 3.3.4 Testing .....                         | 40 |
| 4.0 Results and Discussions .....           | 41 |
| 5.0 Conclusions.....                        | 43 |
| 6.0 References .....                        | 45 |
| 7.0 Appendix .....                          | 47 |
| 7.1 Firmware code .....                     | 47 |
| 7.2 Application code.....                   | 50 |

## List of Figures

|  |    |
|--|----|
| Figure 1. By Android Studio - <a href="https://developer.android.com/studio/">https://developer.android.com/studio/</a> , CC BY-SA 4.0,<br><a href="https://commons.wikimedia.org/w/index.php?curid=74094999">https://commons.wikimedia.org/w/index.php?curid=74094999</a> ..... | 34 |
| Figure 2. Initial schematic. This work is a derivative of " <a href="http://fritzing.org/parts/">http://fritzing.org/parts/</a> " by<br>Fritzing, used under CC:BY-SA 3.0.....   | 36 |
| Figure 3. This work is a derivative of " <a href="http://fritzing.org/parts/">http://fritzing.org/parts/</a> " by Fritzing, used under<br>CC:BY-SA 3.0. ....   | 36 |
| Figure 4. Breadboard prototype. ....   | 37 |
| Figure 5. PCB design this work is a derivative of " <a href="http://fritzing.org/parts/">http://fritzing.org/parts/</a> " by Fritzing,<br>used under CC:BY-SA 3.0. ....  | 38 |
| Figure 6. Humber Sense Hat Prototype PCB. ....   | 38 |
| Figure 7. Example enclosure.....   | 39 |



## 1.0 Introduction

Nowadays, smart home technology is getting popular. Modern and new technologies are coming out every day in order to support people. Smart-Home application could help people to have an overview and control their house in an efficient way. The structure of Smart-Home application consists of sensor and application layer. The sensor gets the primitive data from the house that monitored by using various sensor and user input. Then the primitive data is stored in the database. The application layer displays the database that users request for. There are some advantages that users have when they use the Smart-Home app. First one is energy efficiency; user can adjust the thermostat of their house on their way home to control the temperature so when they get home, they don't feel cold in the house. That is about energy efficiency. Secondly, Smart-Home helps to save energy and money. When people leave the kitchen without turning the light off, the light will turn itself off. So, no more energy will be wasted. In addition, it helps the home owners have more security about their house. When people walk out of their house and five minutes later, they don't remember if they lock the door so they can open the app and check it then lock it easily. The Smart-Home system will also to monitor solar panel activity and its history.

### 1.1 Scope and Requirements

It is an Internet of Things (IoT) capstone project that uses a distributed computing model of a smart phone application that was developed in the previous months and will be constantly updated, it will support database access via the internet to read and display data as well as control various functionalities for example: turning the lights off. It will incorporate closely with an enterprise wireless (capable of storing certificates)

connected embedded system prototype with a custom PCB for sensors we provide: luminosity sensor, temperature sensor, and camera as well as an enclosure (3D printed/laser cut) for the project. The project will be documented via an OACETT certification acceptable technical report that will have a minimum of 9000 words. We will not be doing a CSA testing for this project because we are only making a prototype of a bigger project.

Here are our prototype specification,

#### Mobile Application Specification:

- Developed using Android Studio
- Supports API version 21 (Lollipop)
- Supports database connectivity
- Internet Connection

#### Database Specification:

- Firebase database
- Real-time database
- NoSQL functionalities

#### Hardware Specification:

- PCB will be developed/organized using Fritzing.
- Custom PCB will be printed in our Prototype Laboratory
- Enclosure will be printed in our Prototype Lab or a 3D printing company.
- Should not be left unattended
- Assembled in our classroom



## Report

/1 Hardware present?

/1 Introduction (500 words)

/1 Scope and Requirements

/1 Background (500 words)

/1 References



## 2.0 Background

We would like to thank our collaborator Kim Huynh from Alpha Lab for supporting this project. Our Smart-Home with Solar Monitor will have an X-NUCLEO-OUT01A1 which is going to be where all the components will be connected to (Beningo, 2018). The RTD pt100 and 4-20 mA transmitter (Administrator, 2016) will be connected to a Wheatstone bridge/differential amplifier to find the voltage/temperature difference of the sensor. We have another temperature sensor, the MCP9808 which is an I2C sensor which can be used to compare the temperature between the RTD (Rudi, 2019), to find a more exact temperature. A VEML 7700 ambient light sensor to monitor the light level (Rembor, 2019). All these sensor/actuators will be connected to the PLC which will have a wireless connection to our app. The mobile app will have a real-time database storing the readings it receives from the sensors. The smart house's main goal is to have a secure and automated system for the home owners.

In the Computer Engineering Technology program we have learned about the following topics from the respective relevant courses:

- Java Docs from CENG 212 Programming Techniques In Java,
- Construction of circuits from CENG 215 Digital And Interfacing Systems,
- Rapid application development and Gantt charts from CENG 216 Intro to Software Engineering,
- Micro computing from CENG 252 Embedded Systems,
- SQL from CENG 254 Database With Java,
- Web access of databases from CENG 256 Internet Scripting; and,

- Wireless protocols such as 802.11 from TECH152 Telecom Networks,
- Creating Android Mobile Application with a real-time database from CENG319,
- PCB creation and casing as well as documenting our works from CENG318

This knowledge and skill set will enable us to build the subsystems and integrate them together as our capstone project.

The problem solved by this project is to make life easier by starting a home automation system. Our proposal requests the approval to build a model house that will connect to a database as well as to a mobile device application.

In relation to the project, this semester we plan to complete a prototype of a Smart-Home system.

Potentially, the project will include the following software attributes and design as for the application side of the project in the following sequence from startup:

A login screen which will feature a logo of a house & security related symbol and our team name.

Log in screen will employ the typical login and password functionality tied to the database.

A menu which will contain the following tabs

- Temperature – showing the temperature of the location of the sensor
- Lighting – control and observe the status of lighting
- Camera – showing the video side of the house, optimally control to move the camera
- Door – status and history of the door and optimally control (locked or unlocked)

- Ventilation – status and control of the ventilation
- Help/About – will be a link to a website describing user functionality and contact info

The design will be as simple as possible for the user. We will use menu buttons, text views and various other android studio functionalities which we will learn in this class for this. The UI should provide audible and visual alerts and notifications to the user to any changes in the system of the house which will persist until the user sees them as a measure of security and consistency.

The database we plan to use is Firebase for now as it seems simple and intuitive.

The time estimation for this project is as per the following schedule:

Fall Semester – Software Application, Database Integration and Hardware Practicality

Winter Semester – Software/Hardware Integration and Hardware Modeling

There are a few similar projects on the market as this is currently a popular idea in undertaking by big companies, some of the apps similar to our project are:

SmartHome, Rogers Smart Home Monitoring, Samsung Smart Home

Our goal is to try to make a simpler, faster, friendlier and cheaper model.



## 3.0 Methodology

This project will focus on integrating the Smart-Home hardware component and mobile application into a working system.

### 3.1 Required Resources

There are 3 major components for this project.

- Hardware Component which include the programming platforms, sensors and the Printed Circuit Board (PCB).
- Mobile Application created using Android Studio.
- Real-time database with Firebase.

#### Report

/1 Parts/components/materials (500 words)

/1 PCB, case (500 words)

/1 Tools, facilities (500 words)

/1 Shipping, duty, taxes (250 words)

/1 Working time versus lead time (250 words)

#### 3.1.1 Parts, Components, Materials

The major parts for our hardware components will be sensors/actuators and the platform we will be programming the components.

One of the major components for the project will be the Wi-Fi PLC (Programmable Logic Controller). The PLC compose of 4 components:

Nucleo-401RE which acts as a CPU where it runs a STM32 microcontroller chipset.

Nucleo-401RE will process the input/output of the PLC and decide what to do with it.

X-Nucleo-PLC01A1 is an I/O stackable device. It has 8 conditional inputs which runs through a CLT01-38SQ7 chip and 8 conditional outputs that runs through VNI8200XP monolithic 8-channel driver.

X-Nucleo-OUT01A1 is another stackable component which process digital I/O signals. This chip can operate between 10.5 to 33 volts which is required for one of the temperature sensors we will be using.

X-Nucleo-IDW01M1 is stackable component that will deal with IoT connection. It is an 802.11 b/g/n compliant Wi-Fi expansion module.

For the temperature, we need MCP9808 sensor to measure the temperature. We also need a platform that is Raspberry to run a C program to read the temperature. In addition, to power up the raspberry we use the SD card to load the OS and then insert into the Raspberry Pi. Then using the Ethernet to USB cable and Ethernet cable to display in Remote Desktop Connection through Raspberry IP address. For the PCB, there are two 2x8 pin sockets required to hook up the sensor and raspberry.

We have the RTD PT100 which is another temperature sensor that will be connected to the PLC. For this to work we would need four resistors for a Wheatstone bridge design, a power supply with the capability of at least 24v.(<https://www.divize.com/techinfo/4-20ma-calculator.html>) Also some headers to connect the components too. We will also use an OpAmp set as a subtractor, so we would need two more resistors. If the signal is



really small, we would need to suggest using an instrumentation amplifier which would require two more OpAmps and a few more resistors.

In addition to Light Sensor, we are going to use the VEML7700 Lux sensor to measure the lux sensitivity in the environment. In order to have this sensor fully functional, we also need an Arduino as a platform to a program that read the data gathered from the sensor. Most light sensors just give you a number for brighter/darker ambient lighting. The VEML7700 makes your life easier by calculating the lux, which is an SI unit for light. You'll get more consistent readings between multiple sensors because you aren't dealing with some unit less values.

Another component we are trying to add into the current project is the raspberry pi camera. We will be using a 5MP Raspberry Pi camera which will capture images, store it into the database and display the images into the mobile application.

### 3.1.2 Manufacturing

Although some components doesn't need to have a PCB, challenges can occur during manufacturing. One of the challenges most students make is perfecting a working PCB. Perfecting a PCB can take time, patience, and dedication. Correcting the wire connections can be a hard task because sometimes we may be confused on which connection is correct or not. PCB board is the complicated one to make. We use the software called fritzing to design the circuit. Then we export the Gerber file and send these files to Humber Prototype Lab to make the PCB board. We usually make the PCB board twice because it wouldn't work at the first time. Secondly, the enclosure for the platform we use CorelDRAW to design and we use the laser-cut to print them. Usually,

we have to book an appointment for laser-cut. Each appointment has the maximum length is two hours.

Creating a case can be a hard task as well, although most of us can find an example online for our case to base upon. Some projects do not have an example we can base into. Creating a case from scratch can be a hard task but being able to do it properly can also be rewarding. Creating a case needs trial and error for all the parts to fit in perfectly. Most students used the Prototype Lab for printing their case and some students used third party 3D printing shops.

The first thing we need to consider when beginning any custom enclosure design is simply the size of the case. How big—or small—must it be? Start by laying out all components on a flat surface and begin compactly arranging and rearranging them until the configuration makes sense.

Then determine how much clearance we can get away with above and below the PCB to calculate exterior dimensions. Even if we just got a single beefy capacitor sticking out of the board, it's imperative to make any adjustments to the effective volume of the project box now while the design is still flexible.

The next step we need to be concerned about is the internal mounting. While the electrons in the circuit are eager to find that lovely path of least resistance, we need to make sure that all of them need to take a certain route and keep all components from shorting or loosely jumbling about in the enclosure. Any exposed wires or connections that could cause a short circuit should be insulated as well as every internal component will need to be secured in its spot to withstand any motion or hard impact.

Lastly, we will need to ensure the accessibility of our enclosure. For most hardware, it can be assembled once but if there is anything wrong happen to our PCB, we need it to be easily accessed from outside.

We will likely be using both and see what gives us the best result because the project will consist of a number of large components like the PLC and one of the temperature sensors.

### 3.1.3 Tools and Facilities

The tools and facilities we will be using are either provided by the school or are already we possession of. Safety is also an advocate on making this project. Firstly, we have to wear safety glasses in order to enter the lab for our own safety. Then we have the parts kit, so we could use screw-driver, breadboard, wires, etc. We also use the school lab which is equipped with soldering tools and exhaust tubes to suck out the bad chemicals of fumes. Another facility we often use for creating this project is the Prototype Lab. The Prototype Lab is the place where we will be printing our circuit boards, it also has 3D printing machines for creating our case as well as a laser cutting machine. The Prototype Lab also provided us with some of the parts we needed to complete our PCBs like headers, wire, and soldering materials. The Lab also provided us with working environment to develop our PCB. Another facility we have to use to fulfill this project is the school lab computers. We often use laboratory computer or our personal computers to create our mobile application. Android Studio is the tool we use to finish our mobile application in the previous semester and we will continue to work on it for future updates. Another tool we use in creating our app is the Real-time database, Firebase. Firebase offers a real-time data update which is a very powerful tool in the IoT

world. It can automatically update data in the mobile application as data is changed in the database by the sensors. It will help keep track of all the activity of the homeowner and will keep data safe. One tool we use in programming our Wi-Fi PLC is the STM32CubeIDE. It is a tool provided by STMicroelectronics which allow user to code the Wi-Fi PLC or any device running a STM32 chip. STMicroelectronics also provided us with sample codes to flash into the PLC. The SM32CubeIDE is also used to program the RTD PT100.

We also used CorelDraw on making our design for the case. We will use the Prototype Lab or a third party 3D printing Company to make it for us. Another tool we use in creating of PCB Design is the Fritzing. Materials and examples are from [github.fritzing](https://github.com/fritzing). We use the examples provided to make our own PCB using Fritzing. The file created for this work will then be sent into the Prototype Lab where it will be printed.

Another tool we used is the School Library (Online) which provided us with great reference for our project. Other online references we use are from STMicroelectronics which is a build instruction for the Wi-Fi PLC.

For the programming the MCP9808, we used C, Python or C++ in the Linux Operating System from Raspberry Pi. Then we compile our application by using the command **gcc -g -o**. It will build the target file which we can run it to read the temperature readings from the sensor.

For the programming the VEML7700, we used C, Python in the Linux Operating System from Arduino. Then we compile the application by using 3rd application called Arduino to build the target file which will result in the lux readings from the sensor.

#### 3.1.4 Shipping, duty, taxes

Although shipping, duty, and taxes are different from each student, some students have to pay surprising duty fee and some students had an easy experience when it comes to this. On last year's project, creating the Wi-Fi PLC was overall a fast assemble because Digikey where the components came from offered a next day free shipping on orders \$100 CAD and up. The order arrived the next day with all the components, although I missed it delivered. I have to pick it up in a UPS pickup location which is a 15 minute drive, the package arrived quickly and ready. Although shipping and duty free, regular taxes are still applied for the order.

For this project, we will continue to look for opportunities to save on shipping if we have for example a company's promotion free shipping, or amazon prime shipping. Some materials which comes from abroad will have a greater amount of shipping and duty charges. For saving some shipping from other components, we can allocate the saved budget for the components we need that might come from abroad.

#### 3.1.5 Time expenditure

Working time versus lead time.

For this hardware project, it is easy to do and take about 9 hours to complete the project. We just need basic knowledge about circuit connection, soldering skills, Raspberry power up, etc. The longest time is that we have to wait for the parts we order and we also have to wait 2-3 business days for the prototype lab make our PCB board.

While we are waiting for our components, we will be designing/bread boarding circuits for the temperature sensors to read the input. We will also be coding a way to convert the input voltage into temperature and output it. For the output, we will need to find a way to output the results wirelessly via Wi-Fi connection that's supported by our plc. The App will need to receive the reading and display it appropriately.

The PCB design will be worked on as well. But we would need to wait for the parts before creating the PCB, to make sure that the parts would fit and actually provide the proper outcome.

Case design on the other hand, can be designed before and created based on what the PCB size will be beforehand.

Because we are only updating our Smart-Home mobile application it will not need the same amount of time as starting from scratch. It already has a set up database connection which was done on the previous semester, we only need to incorporate a data readings from a solar panel and store it into the database. We would also need time to code and test the sensors are able to communicate with the database and update data into the mobile application.

After we have all the parts, we would then set a time to assemble the parts and test codes into it. Performing some test cases and see if the mobile application is working properly with the real-time database and updates data in reference with the sensor readings.

## 3.2 Development Platform

### 3.2.1 Mobile Application

#### Status

/1 Hardware present?

/1 Memo by student A + How did you make your Mobile Application? (500 words)

/1 Login activity

/1 Data visualization activity

/1 Action control activity

Include screenshots such as Figure 1. Testing. Progress.

The mobile application we are building now is the Smart-Home application; it will connect sensors and interactive functional devices of a home with an app and a database. This application will give the user control and overview of their home, various sensors and devices are planned such as ones for the temperature, lighting, and ventilation/ac of the house. These will in turn feedback data to the user through a database, in effect the user will have full control and overview of these sensors and devices and in turn control of their home.

Features of our project:

- Control up to four home appliances wirelessly (expandable based on free IO pins)
- As the android application is password protected, it automatically adds security to your home as it can be controlled by the user only

- Databases will be store on Firebase

Potentially, the project will include the following attributes and design as for the application side of the project in the following sequence from startup:

A login screen which will feature a logo of a house & security related symbol and our team name

Log in screen will employ the typical login and password functionality tied to the database

A menu which will contain the following tabs

- Temperature – showing the temperature of the location of the sensors
- Lighting – control and observe the status of lighting
- Door – status and history of the door and optimally control (locked or unlocked)
- Help/About – will be a link to a website describing user functionality and contact info

The design will be as simple as possible for the user. We use menu buttons, text views and various other android studio functionalities. The UI should provide audible and visual alerts and notifications to the user to any changes in the system of the house which will persist until the user sees them as a measure of security and consistency.

When a user registers for a new account in the app, the user information will be stored and sync in the cloud immediately. The users then can access their information through any devices or mobiles. Each user has different data in their database and, they have different types of access.



The Smart-Home mobile app is made using Android Studio. We used Java as programming language on developing the app as well as used Object-Oriented Programming techniques. We then incorporate the application with a real-time database, Firebase where the data, user information, and variable values are stored. The mobile app will be updated to be able to have solar panel monitoring display in the following weeks. The application is functioning properly, and we hope to incorporate the hardware's/sensors in the following weeks as well.

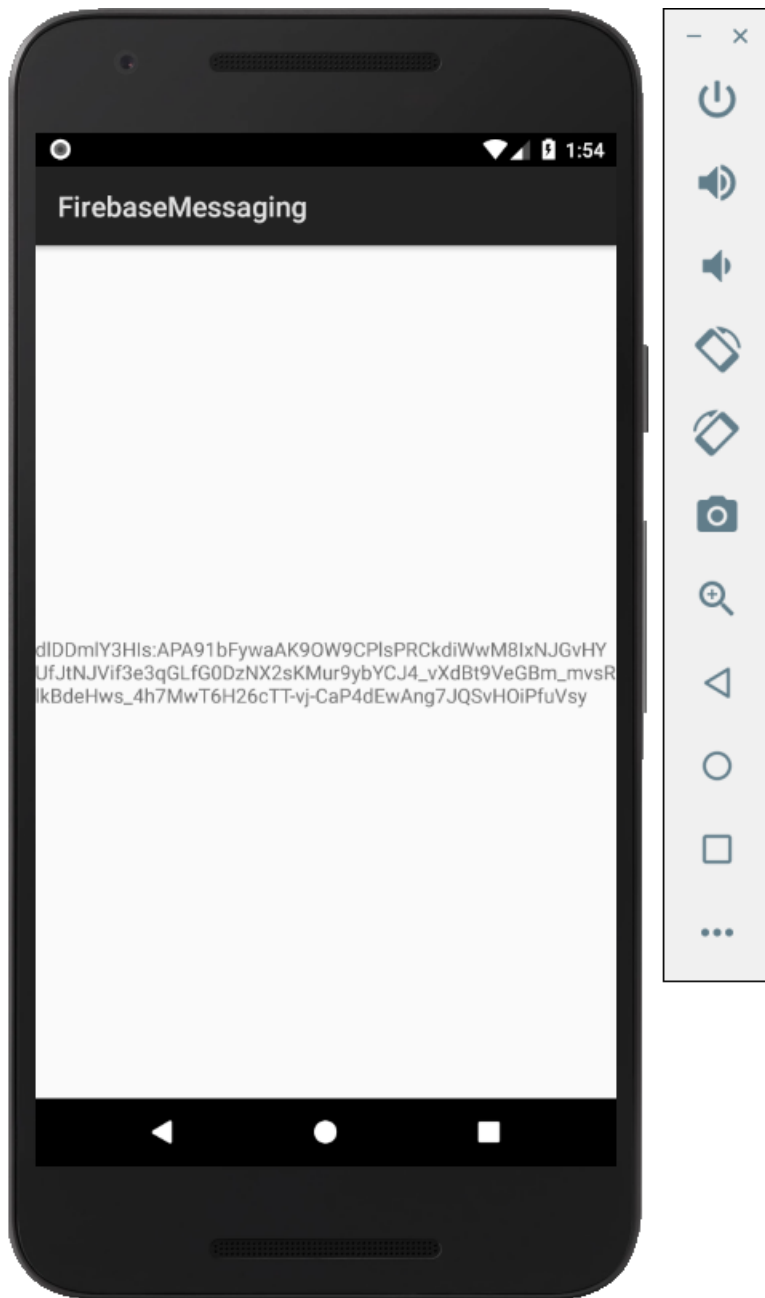


Figure 1. By Android Studio - <https://developer.android.com/studio/>, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=74094999>

### 3.2.2 Image/firmware

#### Status

/1 Hardware present?

/1 Memo by student B + How did you make your Image/firmware? (500 words)

/1 Code can be run via serial or remote desktop

/1 Wireless connectivity

/1 Sensor/effector code on repository

### 3.2.3 Breadboard/Independent PCBs

#### Status

/1 Hardware present?

/1 Memo by student C + How did you make your hardware? (500 words)

/1 Sensor/effector 1 functional

/1 Sensor/effector 2 functional

/1 Sensor/effector 3 functional

The initial schematic design, Figure 2, based on datasheets led to a breadboard layout Figure 3 that was realized Figure 4.

How did you build your Prototype: Breadboard?

Then a PCB was designed, Figure 5, and populated (Figure 6). Bill of Materials, Case, Time commitment. Testing. Progress.

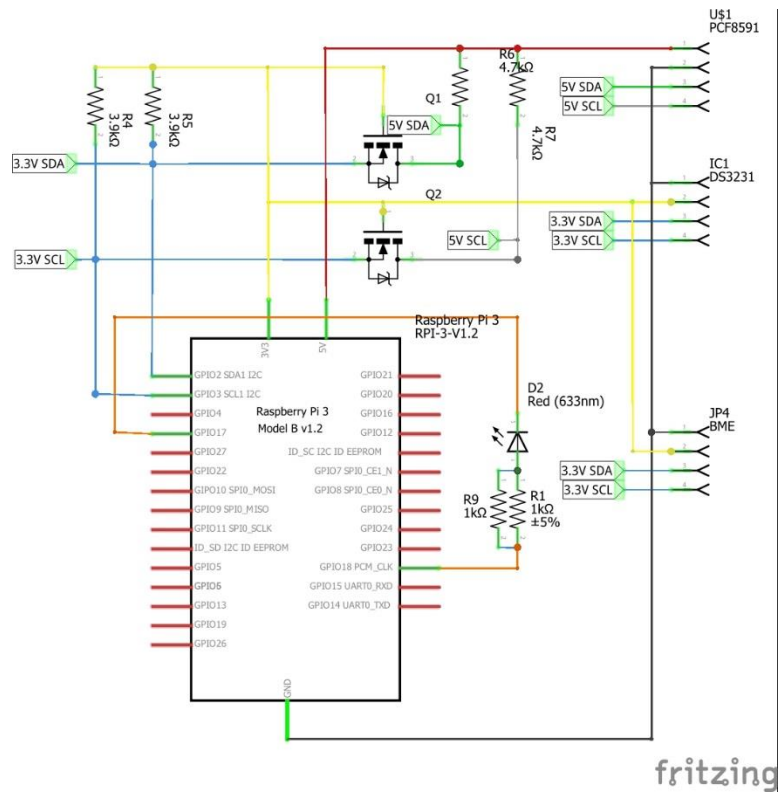


Figure 2. Initial schematic. This work is a derivative of "<http://fritzing.org/parts/>" by Fritzing, used under CC:BY-SA 3.0.

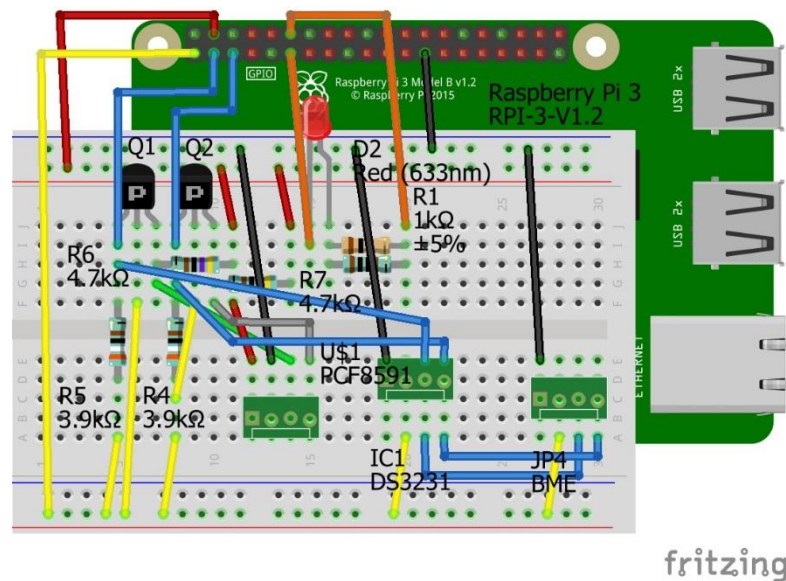
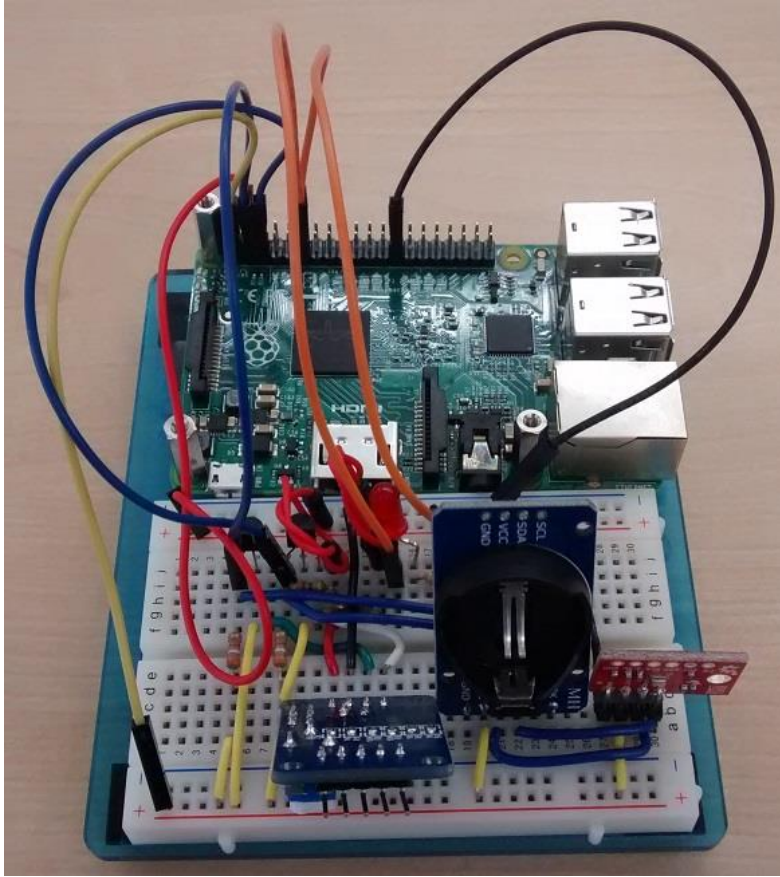


Figure 3. This work is a derivative of "<http://fritzing.org/parts/>" by Fritzing, used under CC:BY-SA 3.0.



*Figure 4. Breadboard prototype.*

### 3.2.4 Printed Circuit Board

#### Demo

/1 Hardware present?

/1 PCB Complete and correct

/1 PCB Soldered wire visible but trim, no holes or vacancies

/1 PCB Tested with multimeter

/1 PCB Powered up

How did you build your Prototype: PCB?

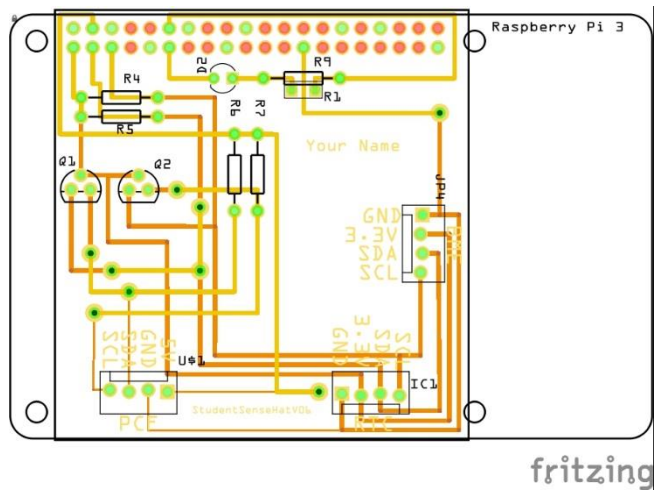


Figure 5. PCB design this work is a derivative of "<http://fritzing.org/parts/>" by Fritzing, used under CC:BY-SA 3.0.

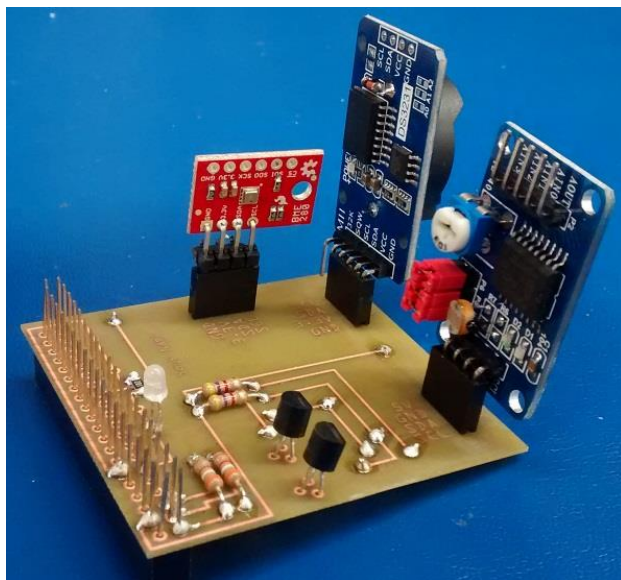


Figure 6. Humber Sense Hat Prototype PCB.

### 3.2.5 Enclosure

#### Demo

/1 Hardware present?

/1 Case encloses development platform and custom PCB.

/1 Appropriate parts securely attached.

/1 Appropriate parts accessible.

/1 Design file in repository, photo in report.

How did you build your Prototype: Case?



*Figure 7. Example enclosure.*

### 3.3 Integration

#### Demo

/1 Hardware present?

/1 Data sent by hardware

/1 Data retrieved by mobile application

/1 Action initiated by mobile application

/1 Action received by hardware

#### Report

/1 Enterprise wireless connectivity (250)

/1 Database configuration (250 words)

/1 Security considerations (500 words)

/1 Unit testing (900 words)

/1 Production testing (100 words)

3.3.1 Enterprise Wireless Connectivity

How did you make a Database accessible by both your Prototype and Mobile Application?

3.3.2 Database Configuration

3.3.3 Security

3.3.4 Testing

Unit testing and Production testing.



## 4.0 Results and Discussions

Is your prototype perfect? What did you learn?



## 5.0 Conclusions

If you were making 1000 of these.

Report

/1 Hardware present?

/1 Checklist truthful

/1 Valid Comments

/1 Results and Discussion (500 words)

/1 Conclusion



## 6.0 References

- Administrator. (2016). *Temperature sensor from RTD PT100 4-20mA transmitter and Arduino*. Retrieved from <http://www.absolutelyautomation.com/articles/2016/02/09/temperature-measurement-rtd-pt100-4-20ma-transmitter-and-arduino>.
- Beningo, J. (2018). *Creating a Custom Wireless Programmable Logic Controller (PLC)*. Retrieved from <https://www.digikey.ca/en/articles/techzone/2018/jun/creating-a-custom-wireless-programmable-logic-controller>
- Rembor, K. (2019). *Adafruit VEML7700 Ambient Light Sensor*. Retrieved from <https://learn.adafruit.com/adafruit-veml7700/overview>
- Rudi, M. (2019). *Reading Temperature Data from a MCP9808 using a Raspberry Pi*. Retrieved from <https://forum.digikey.com/t/reading-temperature-data-from-a-mcp9808-using-a-raspberry-pi/4962>



## 7.0 Appendix

### 7.1 Firmware code

#### Demo

/1 Hardware present?

/3 Code runs concurrently for all sensors/actuators

/1 Project repository contains integrated code

#### Status

/1 Memo including updates

/1 Financial update

/1 Progress update

/1 Modified Code Files in Appendix

/1 Link to Complete Code in Repository

Here is the report regarding how we will be making our images/firmware. This is summarizing on how we did it on the previous semester to be able to read from our sensors and display it using either a serial connection or a console of an IDE.

First, we used a Raspberry pi for our temperature sensor (MCP9808). To do this image, here are the steps. To power up the Raspberry, we need an SD card to load the OS inside it. We recommend loading the version name Raspbian Buster with desktop and recommended software because it has full functionality. After you download the OS, leave it as a zip file. Then download the software call Etcher, you can find it [here](<https://www.balena.io/etcher/>). This software is to flash the Image OS into the SD card. After you load the OS into the SD card, insert it into the raspberry. Plugin the power supply. The one time set up is easy, so just follow the step on the OS. After setting everything up, the C file that use to read the temperature could be found[here](<https://github.com/namnguyen20999/SmartHome-Entry-CapstoneProject/blob/master/Firmware/MCP9808.c>). Use the command: `gcc -g -o MCP9808 MCP9808.c` to compile. Then the execution file will be MCP9808.

We also found a reference on how we could connect the Raspberry pi into the firebase to enable hardware integration and do a HTTP request. Here is the [reference](<https://www.hackster.io/varuldcube100/send-sensor-data-to-firebase-real-time-database-4d6b83>).

Another flatlorm we will be the Nucleo-F401RE which will be used to connect a temperature sensor, but we are still trying to incorporate the temperature sensor into the Nucleo-F401RE. The Nucleo-F401RE will also have a Wi-Fi expansion module for IoT connectivity that we will be using to connect to the Firebase. We will use STM32CUBEIDE to create and flash a HTTP request program and enable the device to make an Http-Request into the Firebase for the software and hardware integration. Here is the link of [main.c](<https://github.com/namnguyen20999/SmartHome-Entry->



CapstoneProject/blob/master/Firmware/main\_http\_request\_f401re.c) for the sample Http-Request that we will be updated to support our hardware/software integration. We were able to connect it to a local Wi-Fi/hotspot. Here is a capture confirming the connection was successful.![Image](https://github.com/namnguyen20999/SmarthHome-Entry-CapstoneProject/blob/master/Images/wifi%20connection%20capture.PNG?raw=true).

To perform this action, first, we need to compile the HTTP request code and flash the binary file generated into the Nucleo-F401RE. Next press the reset button of the Nucleo-F401RE to run the fresh image. Then open a serial connection of the Nucleo-F401RE to set up the Wi-Fi peripherals. To read the temperature from the RTDPT100 attached to the Nucleo-F401RE we tried an assembly code written by Adrian on the previous semester that converts the voltage readings from the RTDPT100 into a temperature. Here is the sample [code](https://github.com/namnguyen20999/SmarthHome-Entry-CapstoneProject/blob/master/Firmware/RTDPT100READ.ino) for it. For this code, we use the Arduino blue pill to program and read the temperature from the RTDPT100 and display the readings using a serial connection.

Another platform we use to create an image was the Arduino with the VEML7700 light sensor that will help configure/identify the lights. The installation for the image of the

Arduino can be found in this [link](<https://learn.adafruit.com/adafruit-veml7700?view=all#python-computer-wiring-6-5>). The first step is to load the OS into the Adafruit, then connecting and setting the pin configuration for the VEML 7700 and run the code into the Adafruit to read the light intensity.

We hope to be able to read/write into the database Firebase for the hardware integration of our mobile application. As of now, we were able to connect into a Wi-Fi or have an Internet connection with the Nucleo-IDW01M1 Wi-Fi module and the raspberry pi. We will continue to try to implement the reading/writing of data into the database.

Here is our Final Report link (<https://github.com/namnguyen20999/SmartHome-Entry-CapstoneProject/tree/master/Documentation/Reports>).

## 7.2 Application code

### Demo

/1 Hardware present?

/1 Memo by student A

/1 Login activity

/1 Data visualization activity

/1 Action control activity

### Report

/1 Login activity

/1 Data visualization activity

/1 Action control activity

/1 Modified Code Files in Appendix

/1 Link to Complete Code in Repository