Technical Report

for

Greenhouse Monitoring System

Version 1.0 approved

Prepared by Colin Blakley, Kenneth Chen and Princess Hernandez

Humber College – School of Applied Technology

April 25, 2019

# Declaration of Joint Authorship

We, Colin Blakley, Kenneth Chen and Princess Hernandez, confirm that this work submitted for assessment is our own and is expressed in our own words. Any uses made within it of the works of any other author, in any form (core concepts, diagrams and figures, previous technologies, programs and source code), are properly acknowledged at the point of use. A list of the references used is included. Colin handled the developing of the GUI and linking with the database. Kenneth handled the developing of the mobile application and the database. Princess managed integrating all hardware components.

|  |  |  |
| --- | --- | --- |
| Co-authors’ signatures | | |
| Date | Name | Signature |
|  | Colin Blakley |  |
|  | Kenneth Chen |  |
|  | Princess Hernandez |  |

# 

# Approved Proposal

January 17, 2019

***Proposal for the development of Greenhouse Monitoring System***

Prepared by Colin Blakley, Kenneth Chen, and Princess Hernandez  
*Computer Engineering Technology Students*

https://github.com/PrincessHernandez/GreenhouseMonitoringSystem

**Executive Summary**

As a student in the Computer Engineering Technology program, I will be integrating the knowledge and skills I have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The internet connected hardware will include a custom PCB with the following sensors and actuators AM2315 Humidity/Temp (0x5C), CCS811 VOC (0x5B), SSD1306 OLED (0x3C). The database will store temperature, humidity and gas level data retrieved from the device. The mobile device functionality will include a series readings of temperature and humidity, as well as a variety of organic compound levels found in the air. There will be warnings when the data retrieved is below or above acceptable ranges. and will be further detailed in the mobile application proposal. I will be collaborating with the following company/department Colin Blakley, Kenneth Chen, and Princess Hernandez. In the winter semester I plan to form a group with the following students, who are also building similar hardware this term and working on the mobile application with me. The hardware will be completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project as a member of a 2 or 3 student group.

**Background**

The problem solved by this project is temperature, humidity and carbon dioxide (CO2) levels are the key factors in monitoring and manage the growth process of plants. Plants become more prone to diseases when temperature is too low or deteriorate when it is too high. Low relative humidity can attract pests like red spider mites because of little to no moisture in plants. The temperature and humidity maintenance play a big impact in contributing to the generation carbon dioxide level. Carbon dioxide is essential for plant growth as it supplies the nutrients.. A bit of background about this topic is greenhouses are closed environments where conditions are optimized for plant growth. Ideally, in greenhouses, temperature should be 25 degrees Celsius, 95% relative humidity and 300 - 500 ppm CO2 level. Measuring and maintaining an ideal temperature, humidity and carbon dioxide (CO2) in greenhouses decreases the dependency on pesticides. Farmers can rely on real-time data of temperature, humidity and CO2 to determine wether their agriculture is at risk. They are able to use the monitoring system to encourage themselves to increase yields. This is an important need for framers in a world where consumers demand more and corporations demand increased profits and cannot afford substantial loses.These sensors will become a staple in the framer community where greenhouse and any structure that need to have a maintained climate.

Existing products on the market include (Jones, 2013). I have searched for prior art via Humber’s IEEE subscription selecting “My Subscribed Content” (IEEE, 2015) and have found and read (Dan, Yang, Jianqiu, 2016) which provides insight into similar efforts.

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.

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

**Methodology**

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall semester. My coursework will focus on the first two of the 3 phases of this project:  
 Phase 1 Hardware build.  
 Phase 2 System integration.  
 Phase 3 Demonstration to future employers.

*Phase 1 Hardware build*

The hardware build will be completed in the fall term. It will fit within the CENG Project maximum dimensions of 12 13/16" x 6" x 2 7/8" (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that will be used is 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC can be obtained. Maximum power consumption will be 20 Watts.

*Phase 2 System integration*

The system integration will be completed in the fall term.

*Phase 3 Demonstration to future employers*

This project will showcase the knowledge and skills that I have learned to potential employers.

The brief description below provides rough effort and non-labour estimates respectively for each phase. A Gantt chart will be added by week 3 to provide more project schedule details and a more complete budget will be added by week 4. It is important to start tasks as soon as possible to be able to meet deadlines.

OLED screen from Ebay, USB powered fan from Amazon

**Concluding remarks**

This proposal presents a plan for providing an IoT solution for the atmospheric factors that can be controlled in greenhouses are temperature, humidity and CO2 levels. High humidity and low temperature allows for plants to grow at an ideal rate. The temperature and humidity sensor will gather data to help manage a greenhouse. If the temperatures were to go out of range the yield of the plants could be effected in a negative way, and in a worst case scenario - kill the plants. By tracking and monitoring the data, adjustments can be made to the greenhouse to ensure maximum yield. Therefore, using humidifiers and circulation fans helps maintain an ideal greenhouse environment where plants can freely grow at the best rate. Better control of temperature and humidity can regulate precise CO2 levels and create a secure environment for growing plants.. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project demonstrating my ability to learn how to support projects such as the initiative described by (Dan, Yang, Jianqiu, 2016). I request approval of this project.

**References**

Jones, J.B. (2013, December 1). *Maintaining Control in the Greenhouse*. Retrieved from

https://www.maximumyield.com/maintaining-control-in-the-greenhouse/2/949

Institute of Electrical and Electronics Engineers. (2015, August 28). IEEE Xplore Digital Library

[Online]. Available: https://ieeexplore.ieee.org/search/advsearch.jsp

Dan, L., Jianmei, S., Yang, Y., & Jianqiu, X. (2016). Precise Agricultural Greenhouses Based on

the IoT and Fuzzy Control. 2016 International Conference on Intelligent Transportation,

Big Data & Smart City (ICITBS). doi:10.1109/icitbs.2016.19

# Abstract

Greenhouses has been an environmental solution to grow crops without involving any form of contaminants. However, due to atmospheric factors it is difficult to maintain plant growth. We have conducted research on integration of 3 different sensors that will carry out a solution that will maintain the growth of plants by monitoring atmospheric factors.

The following technical report will be on the software and hardware requirements for our project Greenhouse Monitoring System. It consists of 3 sensors that will be stationed in a greenhouse that works with our mobile application “Greenhouse Monitor.” This report will provide the details of the device we are going to develop by including features of our system and diagrams for the application.

Table of Contents

[Declaration of Joint Authorship iii](#_Toc4757015)

[Approved Proposal v](#_Toc4757016)

[Abstract xi](#_Toc4757017)

[List of Illustrations xv](#_Toc4757018)

[Introduction 1](#_Toc4757019)

[Intended Audience and Reading Suggestions 1](#_Toc4757020)

[Product Scope 1](#_Toc4757021)

[Overall Description 3](#_Toc4757022)

[Problem 3](#_Toc4757023)

[Solution 3](#_Toc4757024)

[Product Perspective 3](#_Toc4757025)

[Product Functions 4](#_Toc4757026)

[Operating Environment 4](#_Toc4757027)

[Design and Implementation Constraints 4](#_Toc4757028)

[User Documentation 4](#_Toc4757029)

[Assumptions and Dependencies 5](#_Toc4757030)

[Project Requirements 6](#_Toc4757031)

[Database Requirements 6](#_Toc4757032)

[Software Requirements 6](#_Toc4757033)

[Hardware Requirements 6](#_Toc4757034)

[Build Instructions 7](#_Toc4757035)

[Introduction 7](#_Toc4757036)

[System Diagram 8](#_Toc4757037)

[Material Requirements and Budget 9](#_Toc4757038)

[Time Commitment 10](#_Toc4757039)

[Raspberry Pi Configuration 10](#_Toc4757040)

[Mechanical Assembly 10](#_Toc4757041)

[Soldering 10](#_Toc4757042)

[Power Up 14](#_Toc4757043)

[I2C Detection 14](#_Toc4757044)

[Unit Testing 15](#_Toc4757045)

[Production Testing 16](#_Toc4757046)

[Problems Encountered 19](#_Toc4757047)

[VOC Sensor Inconsistent Readings 19](#_Toc4757048)

[Date-Time Python to Timestamp Java 19](#_Toc4757049)

[Approaches 21](#_Toc4757050)

[After Burn-In Period 21](#_Toc4757051)

[Converting Time to Seconds 21](#_Toc4757052)

[Progress Reports 23](#_Toc4757053)

[Report 1 23](#_Toc4757054)

[Report 2 23](#_Toc4757055)

[Report 3 24](#_Toc4757056)

[Report 4 25](#_Toc4757057)

[Conclusion and Recommendations 29](#_Toc4757058)

[Technical References 31](#_Toc4757059)

[Appendix A: Glossary 33](#_Toc4757060)

[Appendix B: Analysis Models 33](#_Toc4757061)

# 

# List of Illustrations

[Figure 1. Diagram for ideal communication between software and hardware 3](file:///C:\Users\n01099524\Downloads\Greenhouse%20Technical%20Report.docx#_Toc5293285)

[Figure 2. Final look of the device with an enclosure. 7](#_Toc5293286)

[Figure 3. System diagram for components of the system. 9](#_Toc5293287)

[Figure 4. Connection between Raspberry Pi and 3 sensors using a breadboard. 11](#_Toc5293288)

[Figure 5. Schematic view of the connection between Raspberry Pi and 3 sensors. 12](#_Toc5293289)

[Figure 6. PCB view of the connection between Raspberry Pi and 3 sensors. 12](#_Toc5293290)

[Figure 7. Vias soldered on PCB as circled in red. 13](#_Toc5293291)

[Figure 8. Necessary pins soldered on the 40 pin header. 14](#_Toc5293292)

[Figure 9. 4-and-5-pin on PCB as circled in red. 14](#_Toc5293293)

[Figure 10. Complete look of the PCB. 15](#_Toc5293294)

[Figure 11. Display output of 2 sensors detected. 16](#_Toc5293295)

[Figure 12. Display output of 3 sensors detected. 16](#_Toc5293296)

[Figure 13. Sample output terminal. 17](#_Toc5293297)

[Figure 14. Sample output on OLED screen. 18](#_Toc5293298)

# Introduction

This technical report presents the methods and processes of how we achieved our project result. Atmospheric factors can affect plant growth and the result of our device can help farmers control the factors that affect them. Our project will be used as a monitoring system that can be accessed from a mobile device. The system is ideally situated in a greenhouse for best performance and outcome. The application will provide temperature, humidity and CO2 levels in the greenhouse. It also allows the user to control the fan connected to the device. This device is meant for agricultural workers that need help to maintain their plant growth if they cannot tend to their plants immediately.

## Intended Audience and Reading Suggestions

This document is intended for agricultural farmers, greenhouse technicians, and workplace management as a reference when they are using the device and application. Also, the developers of the system are able to contribute and revise to this document at any time.

## Product Scope

Greenhouse workers must check on their plants frequently and the surrounding factors of plants’ growth. This system will be designed to help raise awareness of the atmosphere factors that affect plants by providing real-time record of temperature, humidity and air quality like CO2.

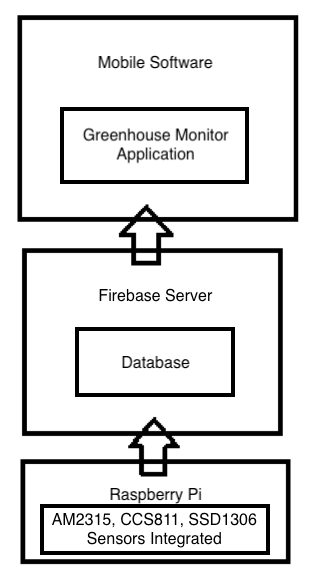
# 

# Overall Description

## Problem

The environment of plants is consistently monitored throughout the day. Today, there are current devices that measure plant environment but it can take hours to measure temperature, humidity and carbon dioxide respectively.

## Solution

**A system that gathers data of all three (temperature, humidity and carbon dioxide) plant environment is the most ideal solution for this problem. An all-in-one monitoring device allows for easier and faster ways to grow plants at a constant rate.

## Product Perspective

The product has 2 main parts, the mobile app and the integrated sensors. The application is used to view data on Temperature and humidity from AM2315 sensor, and CO2 from CCS811. The OLED (SSD1306) sensor displays the data of all three factors. All sensors will be integrated onto the Raspberry Pi.

The mobile application will need to communicate with the database to get the data from the Raspberry Pi.

Figure 1. Diagram for ideal communication between software and hardware

The Raspberry Pi is used to communicate with the database by sending data for it to be stored.

Refer to Figure 1.

## Product Functions

The device and application need an internet connection to function, and will connect to a Firebase database server. The product must be connected to the Internet to do the following:

* Read the temperature, and send it to the server so the application can read it
* Read the humidity, and send it to the server so the application can read it
* Read the CO2 levels, and send it to the server so the application can read it

## Operating Environment

The device operating environment is mostly ideal for indoor, but it can also work outdoor. The mobile application can be used anywhere with an Internet connection, software is android lollipop and up.

## Design and Implementation Constraints

The Internet connection is a constraint for the application. It isn’t a huge constraint but the application fetches data from the database once in a while or the static wouldn’t update, it is crucial that there is an Internet connection for the application to function. The Raspberry Pi itself has the same constraint due to the need of it sending data to the database as well.

## User Documentation

All documentation and source code for our firmware and Android Application are available through GitHub as it is open source.

## Assumptions and Dependencies

One assumption about the product will be use on a mobile phone that meets the required specifications. If the phone does not have the hardware resources required for the application, for example other application has already allocated space, the application may not work as intended.

# 

# Project Requirements

## Database Requirements

The server-side application requires a Google account for the database with Firebase. The database will store the readings of the temperature, humidity, CO2 and time. The application will be connected with the database to show the readings stored on the database to the application. Kenneth will be working on the database, and Colin will link the database to connect to the application to retrieve readings from the Pi.

## Software Requirements

The mobile application shows information of temperature, humidity and CO2 from the database. There will be an added feature where the user has the ability to control the fan to go on or off. Kenneth will be working on developing the application and its features, and Colin will develop the GUI (XML files). This part will be updated as we move towards the completion of the project and document.

## Hardware Requirements

The AM2315 sensor measures the temperature and humidity, and the CCS811 sensor measures the levels of CO2 that are in the air. Both sensors are integrated together with Raspberry Pi. The completed hardware will be able to be connected with the software in order to receive and store data in a remote database. Additionally, OLED will be added as a feature to show readings of temperature, humidity and CO2. Princess will be integrating all these sensors onto her Raspberry Pi.

# Build Instructions

Figure 2. Final look of the device with an enclosure.

## Introduction

The Greenhouse Monitoring System is a device that allows greenhouse technicians to monitor atmospheric factors that may affect plant growth. The temp/humid sensor measures temperature and humidity respectively, while the VOC sensor can measure equivalent CO2. These three factors can have an impact in plants whether they are growing at a normal or slow pace. To increase awareness, our monitoring system serves as a tool for greenhouse technicians to maintain and control the temperature, humidity and CO2 levels in a greenhouse environment.

## System Diagram

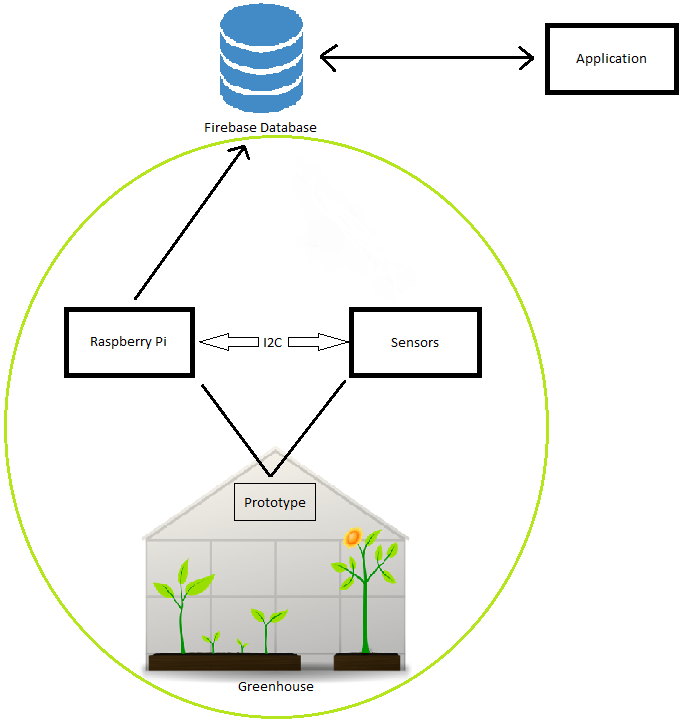


Figure 3. System diagram for components of the system.

## Material Requirements and Budget

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Source | CAD Price | USD Price | Link |
| CanaKit Raspberry Pi 3 B+ Starter Kit | Amazon | $98.99 |  | https://www.amazon.ca/CanaKit-Raspberry-Starter-Premium-Black/dp/B07BCC8PK7/ref=sr\_1\_1\_sspa?crid=3FN3IC415XF1B&keywords=raspberry+pi+3&qid=1551450371&s=electronics&sprefix=raspberry+%2Celectronics%2C153&sr=1-1-spons&psc=1 |
| OLED Display for Arduino 128x64 Pixel I2C, 0.96 inch, SSD1306, Library, 3-5V | Amazon | $6.29 |  | https://www.amazon.ca/Display-Arduino-128x64-SSD1306-Library/dp/B077D4RQG1/ref=sr\_1\_1\_sspa?hvadid=324955435621&hvdev=c&hvlocphy=9000993&hvnetw=g&hvpos=1t1&hvqmt=b&hvrand=7609421822030419687&hvtargid=kwd-45652301614&keywords=oled+ssd1306&qid=1551449756&s=electronics&sr=1-1-spons&tag=googcana-20&psc=1 |
| 20x2-pin Header (Female) | Amazon | $9.99 |  | https://www.amazon.ca/2x20-pin-Female-Stacking-Header-Raspberry/dp/B071FT161B/ref=sr\_1\_3?crid=SKA71RFLSHC5&keywords=20+pin+header+female&qid=1551450588&s=electronics&sprefix=20+pin+header%2Celectronics%2C149&sr=1-3 |
| 5-pin Header (Female) | Creatron | $0.43 |  | https://www.creatroninc.com/product/5-pin-receptacle-socket/?search\_query=4+pin+stackable+header&results=48 |
| 4-pin Header (Female) | Creatron | $0.42 |  | https://www.creatroninc.com/product/4-pin-receptacle-socket/ |
| AM2315 - Encased I2C Temperature/Humidity Sensor | Adafruit |  | $29.95 | https://www.adafruit.com/product/1293 |
| SparkFun Air Quality Breakout - CCS811 | Sparkfun |  | $20.95 | https://www.sparkfun.com/products/14193?\_ga=2.97662492.2095878335.1537831851 |

Table 1. List of materials and their respective prices needed to build the system.

\*Prices does not include tax and shipping.

## Time Commitment

This project can be completed in approximately 7 days if you followed the mechanical assembly and diagrams. Desired to do so, you must order the materials 2 - 3 weeks before working on this project as shipping takes about 1 - 2 weeks. Once you receive the materials, you can start configuring your Raspberry Pi which will take around 3 hours. Connecting the parts to the Raspberry Pi will take 30 minutes. And then to set up the code on the Raspberry Pi and testing the code will take around a 1 hour. It is recommended that you take 3 - 4 hours working on this project daily.

## Raspberry Pi Configuration

1. Create your Raspberry Pi's [image](https://github.com/six0four/StudentSenseHat/blob/master/cribpisdcard.md) for your project.
2. Then test your LED to [blink](https://github.com/six0four/StudentSenseHat/blob/master/README.md#student-raspberry-pi-image-creation-and-test-code).

## Mechanical Assembly

1. Set your sensors onto your breadboard and connect your sensors to the appropriate GPIO pinout of the Raspberry Pi. The wiring should look something like this:
2. Boot your Raspberry Pi and open terminal.
3. To check if your sensors are functioning, click [here](#i2c-detection).

## Soldering

1. You can design your own PCB using [Fritzing software](http://fritzing.org/download/) for free or our version of the [fritzing file](https://github.com/PrincessHernandez/GreenhouseMonitoringSystem/blob/master/documentation/Fritzing/Greenhouse.fzz). You can refer to the images below:

Breadboard view

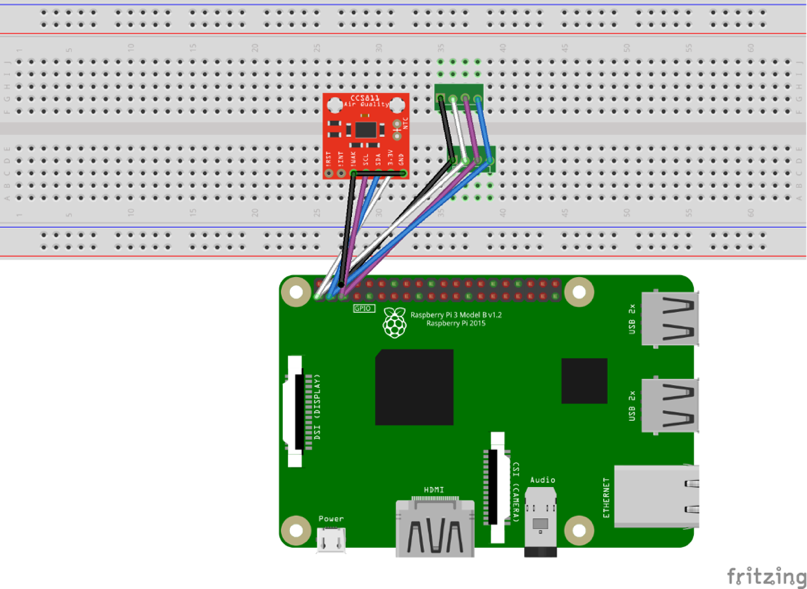


Figure 4. Connection between Raspberry Pi and 3 sensors using a breadboard.

Schematic view

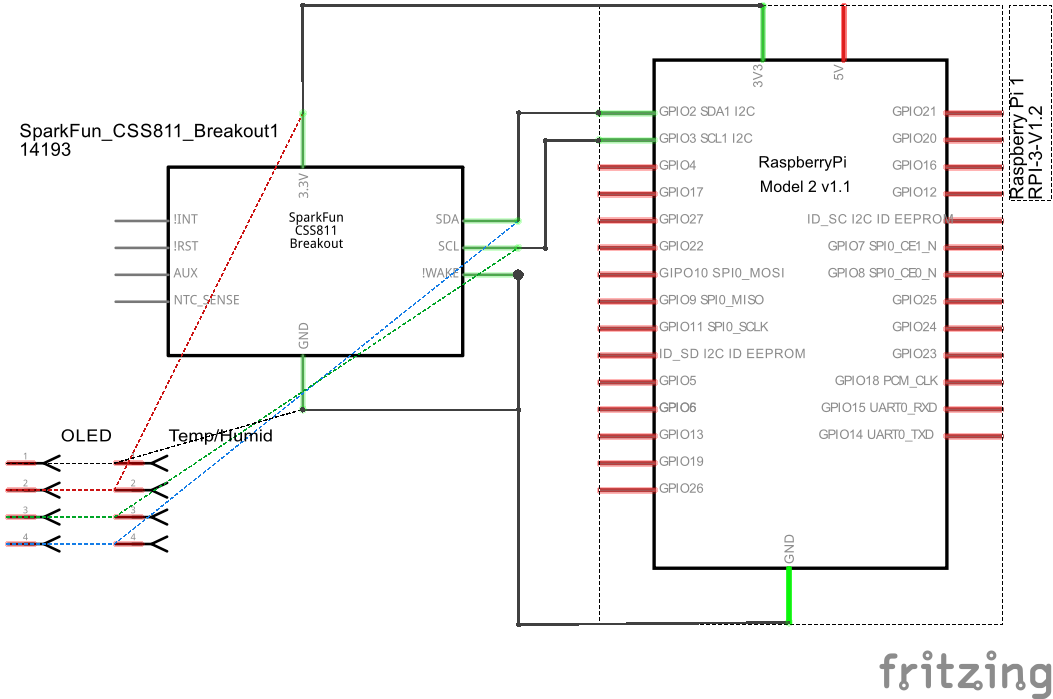


Figure 5. Schematic view of the connection between Raspberry Pi and 3 sensors.

PCB view

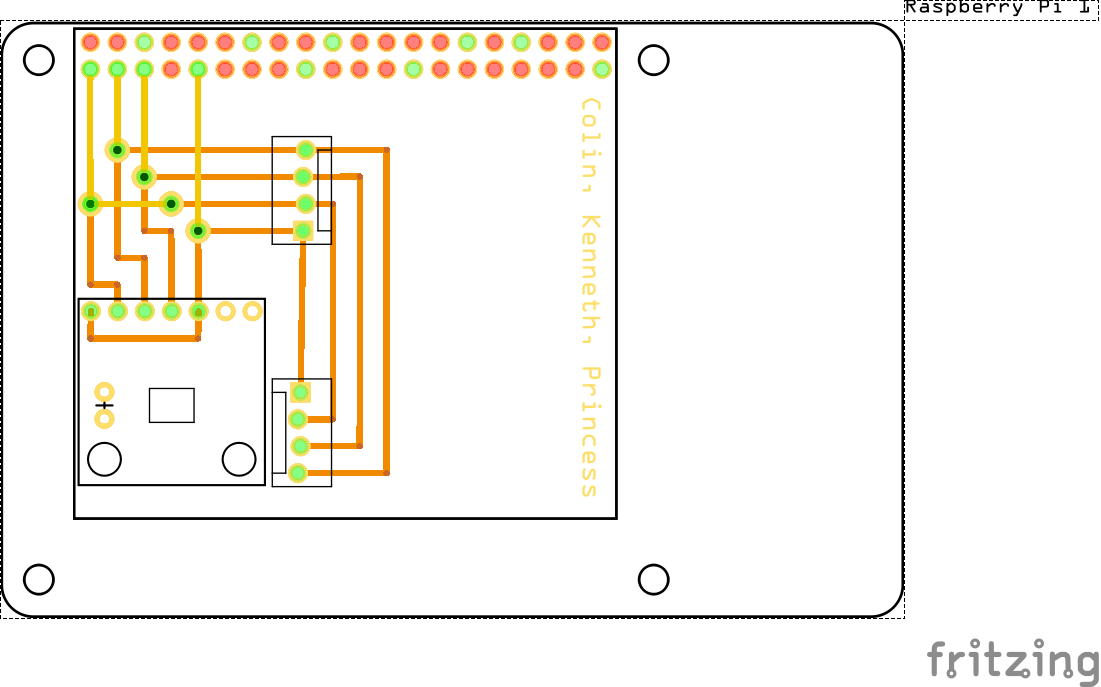


Figure 6. PCB view of the connection between Raspberry Pi and 3 sensors.

Here are the following pins that you should know for this project:

* *Power Pins*
* Vin - power pin
  + Since the sensor uses 3.3V, give it the same power as the logic level of you Raspberry Pi.
* GND - common ground for power and logic
* *Logic Pins*
* SCL - i2c clock pin
  + Connect to your Raspberry Pi i2c clock line.
* SDA - i2c data pin
  + Connect to your Raspberry Pi i2c data line.
* WAKE\* - wakeup pin for the sensor
  + Please make sure that the WAKE pin is connected to GND. Otherwise, you will not get the address for CCS811 sensor.

1. Once you have obtained your PCB board solder the following:

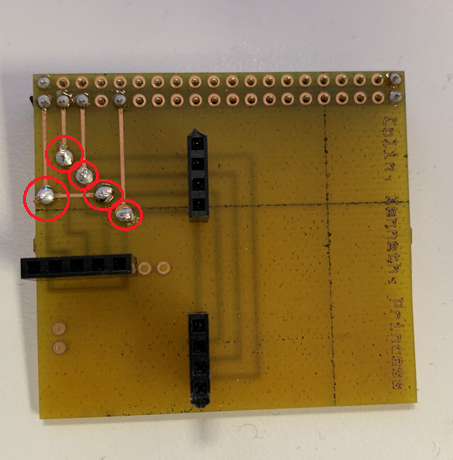
* Vias\*
* 

Figure 7. Vias soldered on PCB as circled in red.

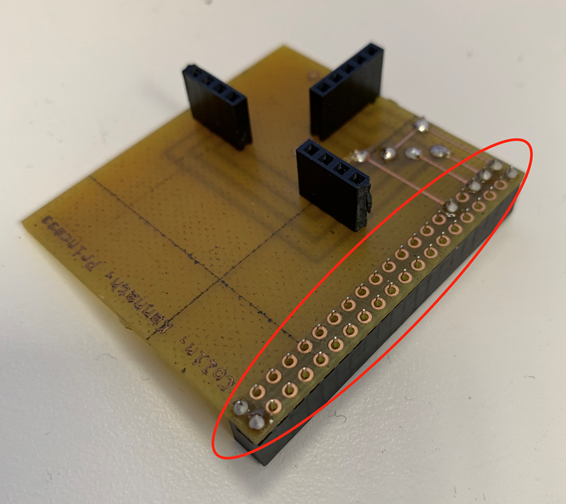
* \*Note: You must thread a single strand of wire through the holes, solder it, and then cut the remaining wires off.
* 20x2-pin socket
* 

Figure 8. Necessary pins soldered on the 40 pin header.

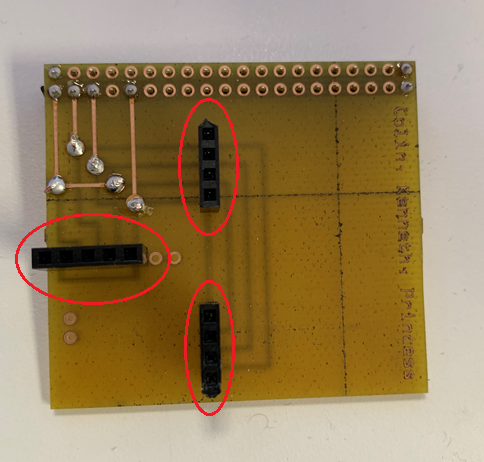
* 4 and 5-pin sockets
* 

Figure 9. 4-and-5-pin on PCB as circled in red.

Once you have finished soldering, your board should look like this along with the sensor:

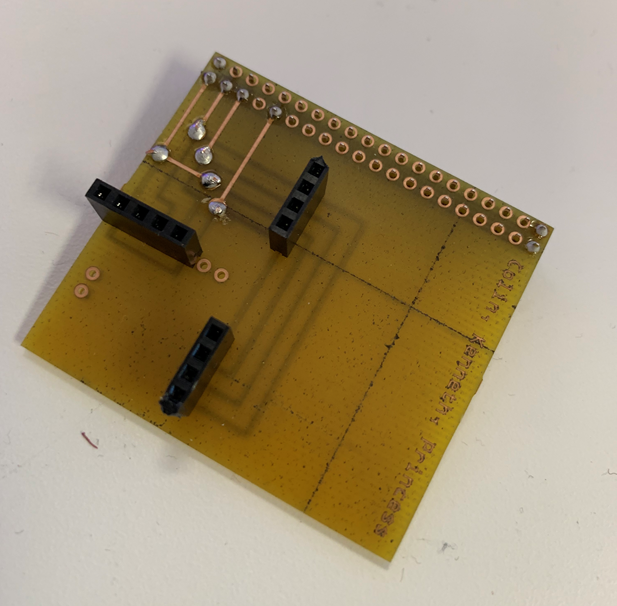


Figure 10. Complete look of the PCB.

## Power Up

1. Boot your Raspberry Pi and open terminal.
2. To check if your sensors are functioning, click [here](#i2c-detection). If they are not being detected, there might be a problem with the solder or the PCB itself.
3. Download the following codes here to setup your sensors: [CCS811](https://github.com/PrincessHernandez/GreenhouseMonitoringSystem/blob/master/python%20script/classes/CCS811_RPi.py) and [SSD1306](https://github.com/PrincessHernandez/GreenhouseMonitoringSystem/blob/master/python%20script/classes/SSD1306.py).

## I2C Detection

To check if the board is functioning and detecting the sensor, open terminal and type

sudo i2cdetect -y 1

This will display an output of the sensor's address - 0x3C, 0x5B. See sample output below:

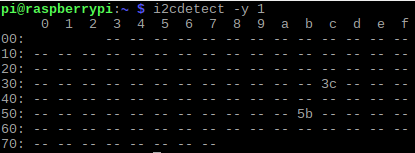


Figure 11. Display output of 2 sensors detected.

To wake up AM2315 Temp/Humid sensor, run the command(quickly) again. It will show all addresses - 0x3C, 0x5B, 0x5C.

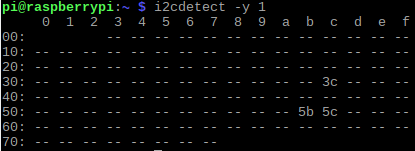


Figure 12. Display output of 3 sensors detected.

## Unit Testing

Download the program [greenhouse.py](https://github.com/PrincessHernandez/GreenhouseMonitoringSystem/blob/master/python%20script/greenhouse.py). Now that you've downloaded the codes and detected your sensors, go to terminal and run

sudo python greenhouse.py

This command should run the program and display on the terminal. An example output is shown below:

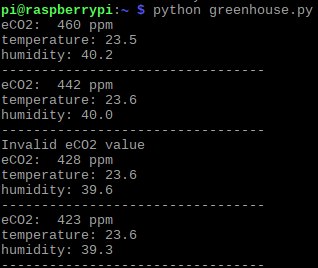


Figure 13. Sample output terminal.

## Production Testing

Now that you've seen what the sensors' output will look like, download the program [greenhouse2.py](https://github.com/PrincessHernandez/GreenhouseMonitoringSystem/blob/master/python%20script/greenhouse2.py) and run

sudo python greenhouse2.py

This command should run the program and display the output on the OLED screen. An example output is shown below:

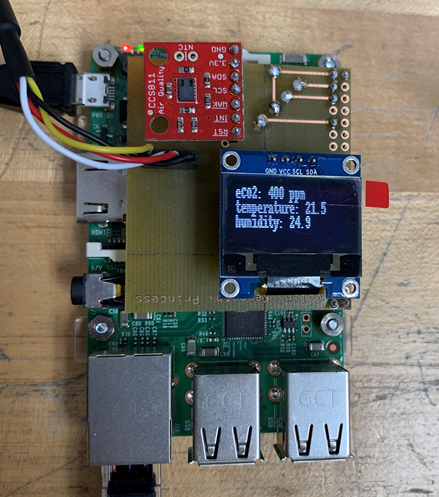


Figure 14. Sample output on OLED screen.

# 

# Problems Encountered

## VOC Sensor Inconsistent Readings

According to the hookup guide on the SparkFun website, the VOC sensor needs a 48-hour burn in time and a 20-minute warmup. However, the readings go above 8192 ppm each time we run the program. Also, there are moments the readings jump spontaneously, e.g. 50 000 ppm, which is inconsistent with previous readings that start from 400 ppm. Many users of the sensor are also having the same issue and it is yet to be solved.

## Date-Time Python to Timestamp Java

We are having trouble converting the date and time from the database to the application. The date and time for Python is formatted in string, while the timestamp for Java is in integer. This is the very reason we cannot plot the time on the graph in the application.

# Approaches

## After Burn-In Period

In order to ensure that the VOC sensor performance is stable, we must run the sensor for 20 minutes before we start testing and/or troubleshooting. We noticed more accurate readings are generated when doing this step and therefore, we must do this each time we are testing the device.

## Converting Time to Seconds

To plot the time in the graph, we worked on sending the time as seconds by converting epoch to readable date and time. The application grabs the time from the database as a string in seconds and parse the time into a date format. The date is plotted into the graph and is multiplied by 1000 to be displayed. The date uses a SimpleDateFormat function to display the date in any format we want. In this case, the format is MMM HH:MM.

# Progress Reports

## Report 1

|  |  |
| --- | --- |
|  | Thu, Jan 31, 2019 at 6:06 PM |
|  | |
| |  | | --- | | Dear Kristian,    **Group Update**  As of today we now have all of the hardware for our project. We have decide to create a new PCB to integrate all three of our sensors. The fritzing files has been complete and sent to the prototype lab for creation. Colin has begun working on the database. Kenneth hasstarted work on our application.    **Financial**  Financially we needed to purchase an OLED screen at the cost of $14.40 and a fan for $13.55. The project will also require us to purchase 2 four pin headers and a 5 pin header along with a 24 pin GPIO header with a total cost $15.09.    **Problems**  Some issues we have encountered are trying to get all of parts together and into a case and how they will all fit together because two sensors have to be exposed outside of the case.  Sincerely,  Princess, Colin, Kenneth | | |

## Report 2

|  |  |
| --- | --- |
|  | Thu, Feb 28, 2019 at 6:53 PM |
|  | |
| |  | | --- | | Dear Kristian,    **Summary**  As of today, we are on track with the project. Kenneth finished the structure of the database, but it may change over the next few weeks (we may rearrange the structure because the tested data and user is separate - tied by User ID). The application is functional - reads the data from the database and displays data in a graph or the most recent value. The new PCB we created has been soldered. Princess and Colin are able to get all sensors to work together.    **Financial**  The Prototype Lab does not have 20-pin headers stocked, and Kelly and Vlad are not sure when they will be back in stock. So, we used a 25-pin header to make a 20-pin header. During reading week, we were able to finish soldering the PCB.  **Plans**   We need to work on getting the data from the Raspberry Pi to the Firebase. Also, we are planning to omit the fan in our hardware because it is outside of the scope of our project. We are also planning on creating a new case for the Pi and sensors.  Sincerely,  Colin, Kenneth, Princess | | |

## Report 3

|  |  |
| --- | --- |
|  | Thu, Mar 14, 2019 at 5:51 PM |
|  | |
| |  | | --- | | Dear Kristian,    **Group Update**  We are following the schedule and we believe we can get everything done on time before the Project Demonstration on April 11. Colin was able to send data from the sensors to the database.    **Current Progress**  Princess is working on designing the enclosure. We expect to have it created by next week. Kenneth is making progress on the mobile app, and is currently working on retrieving the live sensor data from the database to be displayed on the mobile application. The technical report is constantly being updated.  **Financial**  We are planning to buy the presentation board for the upcoming CTI presentations. We are still looking for a cheaper option than from Staples.  Sincerely,  Colin, Kenneth, Princess | | |

## Report 4

|  |  |
| --- | --- |
|  | Thu, Mar 28, 2019 at 6:02 PM |
|  | |
| |  | | --- | | Dear Kristian,    **Current Progress**  Currently, the Raspberry Pi is able to send data to the database (Firebase) dynamically with timestamp. A few changes in the firmware will be made in a couple of days to accommodate with the software part of the application. The Android application is complete (able to read data dynamically), but Kenneth is working on getting the time to show on the graph (Python sends the date in a different format), which is our main priority. Enclosure is complete and no modifications are required.  **Problems**  According to the hookup guide on the SparkFun website, the VOC sensor needs a 48-hour burn in time and a 20-minute warmup. However, the readings go above 8192 ppm each time we run the program. Also, there are moments the readings jump spontaneously, e.g. 50 000 ppm, which is inconsistent with previous readings that start from 400 ppm. We’ve searched on SparkFun Forum (<https://forum.sparkfun.com/viewtopic.php?t=47161>) and many users of the sensor are also having the same issue and it is yet to be solved.  **Solution(s)**  In order to ensure that the VOC sensor’s performance is stable, we must run the sensor for 20 minutes before we start testing (every time). We noticed more accurate readings are generated when doing this step and therefore, we must do this each time we are testing the device. This is the only solution we have come up with so far. The solution for the time issue is to send the time as seconds since the epoch to the database and parse the time into a date format on the application.  **Other**  We are hoping to get the application to plot the time on the graph by the end of this week as we are trying to prepare for the upcoming project demonstration. Overall, there are no other problems we have encountered.  Sincerely,  Colin, Kenneth, Princess | | |

# Conclusion and Recommendations

After completing the implementation (Hardware and Software) of the monitoring system, the resulting prototype is advantageous as it employs three components that measure plants’ surrounding environment (Temperature, Humidity and VOC sensors). To assist those who are farmers and technicians in greenhouses, the monitoring system was created to eliminate the extra required work to measure the surrounding environment of plants and enables multitasking by measuring temperature, humidity and carbon dioxide at the same time. The system’s main feature allows greenhouse workers to maintain the growth of plants by checking the monitoring system’s display and/or by checking the mobile application. The system has the potential to be used in greenhouses to simplify work for greenhouse farmers and technicians, and to monitor the atmospheric factors anytime and anywhere on the mobile application for a more efficient use, so long as they are connected to the internet.

# Technical References

Brander, Matthew. (2012, August). *Greenhouse Gases, CO2, CO2e, and Carbon: What*

*Do All These Terms Mean?* [PDF File]. Retrieved from https://ecometrica.com/assets/GHGs-CO2-CO2e-and-Carbon-What-Do-These-Mean-v2.1.pdf

DeVito, J. & DiCola, T. (2017). Adafruit Circuit Python SSD1306. GitHub repository,

https://github.com/adafruit/Adafruit\_CircuitPython\_SSD1306

Jones, J.B. (2013, December 1). *Maintaining Control in the Greenhouse*. Retrieved from

https://www.maximumyield.com/maintaining-control-in-the-greenhouse/2/949

Lukas, P. (2017, July 11). CCS811\_RPi. GitHubGist repository,

https://gist.github.com/xxlukas42/60ae08f75e68a0cfcdb7c9dd60145d34

Taylor, M. (2017, April 27). CCS811 Air Quality Breakout Hookup Guide. SpakFun

Electronics. Retrieved from https://learn.sparkfun.com/tutorials/ccs811-air-quality-

breakout-hookup-guide?\_ga=2.265865676.210527324.1553258399-

1038958827.1553258399

# 

# Appendix A: Glossary

CO2 – Carbon Dioxide

VOC – Volatile Organic Compound

# Appendix B: Analysis Models