Refrigerator Food Contamination Detection using Electronic Nose

By

Agnivah Poddar
Siddharth Muralidaran
Simran Patil

Final Report for ECE 445, Senior Design, Spring 2018
TA: Anthony Caton

2nd May 2018
Project No. 30

Abstract

The electronic nose is a system that can detect early food contamination. It consists of a sensing unit and a display unit that communicate with each other over Bluetooth to detect onset of food contamination and notify the user of it. The sensing unit comprises of an array of sensors that detects chemicals like CO2 and alcohol in increased compositions and creates a packet that triggers the display unit to display a warning for the user on the LCD screen. The user can press the ACK button and send a packet to the sensing unit placed inside the fridge to recalibrate the sensors indicating that the spoiled food was cleared. The system is modular and works in cold and humid conditions like inside the refrigerator. It can be used to make any refrigerator smart by enabling it to notify of food contamination. The system has a very simple user interface and is powered with 4 AA batteries for convenience.

Contents

1. Introduction

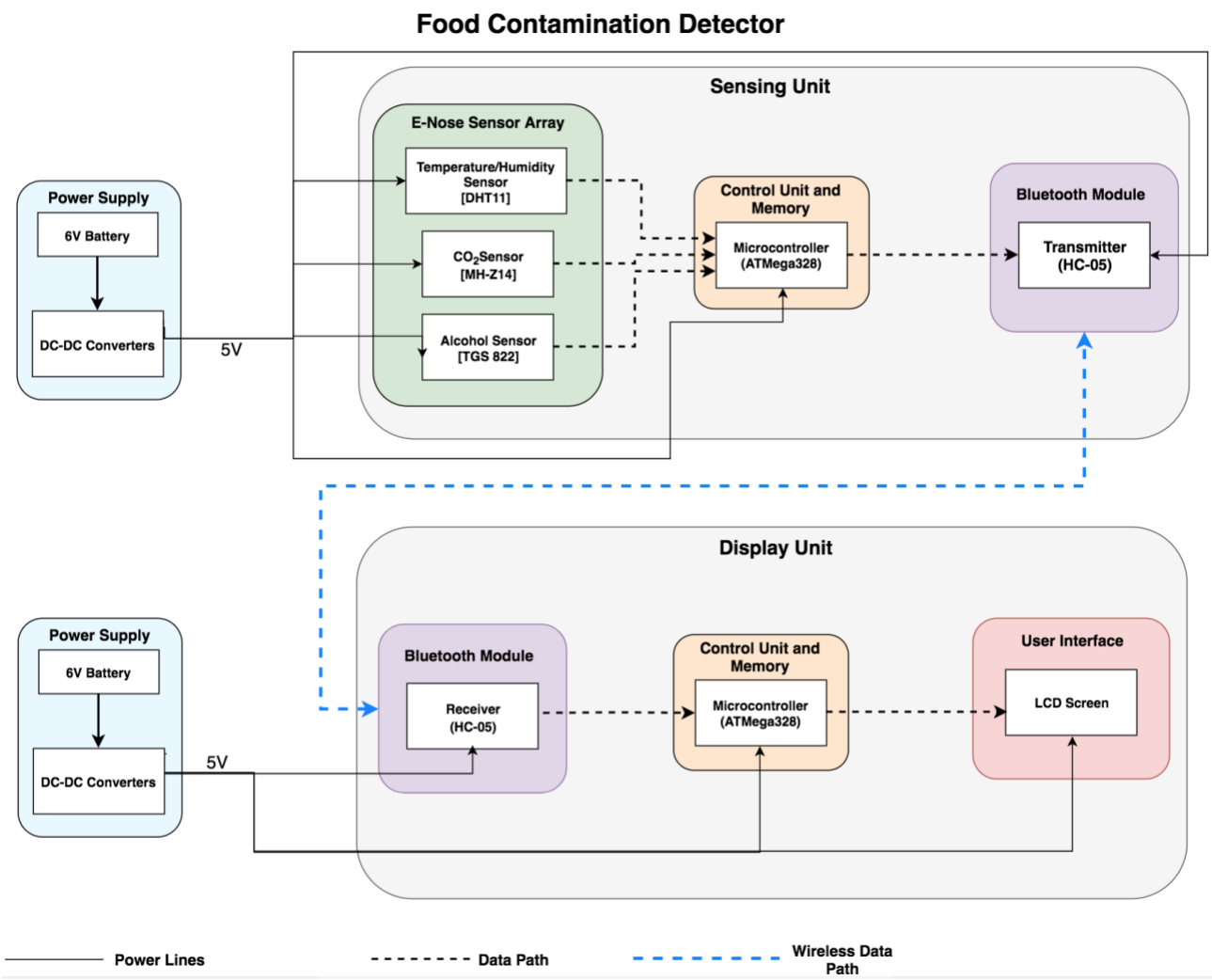
Food poisoning is a serious problem that affects thousands of people every year. Out of approximately 5000 total deaths reported in the United States every year, 1500 are caused by pathogens like Salmonella, Listeria and Toxoplasma. The World Health Organization (WHO) reports that salmonellosis caused by Salmonella spp. is the most frequently reported food borne disease worldwide [2]. Poisoning food must be detected early in order to prevent diseases. A lot of food gets wasted in developing countries or in tropical regions while getting transported from the farmlands to the markets, mainly because of improper intermediary storage techniques. This results in colossal amounts of post-harvest losses that are incurred by the poor farmers or result in raised food prices to make up for the lost food.

An early detection system can help prevent the rotting item from affecting the remaining produce. This will imply that food wastage is reduced and the prices associated with these goods don't skyrocket. Even in our households, we often forget about some food item stowed away in our refrigerator drawer hidden under other items, until it turns smelly and moldy, and needs to be disposed. Having a device that detects this would be of great help in reducing such wastage. Not just in a household setting, an early food contamination detection system can find application in industrial refrigerators and storage units that cater to concentration of a single food item in refrigerators and incur losses due to spoilage of one element resulting in contamination of the entire unit.

Contaminated food is usually detected by odor which is composed of molecules of specific sizes and shapes with a corresponding receptor in the human nose. The brain identifies the smell associated with that particular molecule when signaled by the receptor. Electronic nose is an array of sensors that imitates this biological functionality.

2. Design

To keep the system as modular and the UI as user friendly as possible, our design consists of 2 separate units. There is a power supply that is composed of a 4 AA batteries contributing 6V to power all the circuitry and a DC-DC converter to provide each of the units with the required voltages. The sensor array unit consists of a CO2 sensor, Alcohol/Organic Solvent sensor and Temperature Humidity sensor - all working together to provide the values on detection of the chemicals at the particular temperature and humidity setting to contribute to accurate readings. The inputs from this unit are fed to the microcontroller which interprets the readings and passes a signal to the Bluetooth module to be transmitted to the receiver. The other unit is composed of the Bluetooth receiver module and another MCU that handles the received values, interprets the data and displays appropriate values on the LCD screen to notify the user of potential food contamination. There is a push button on the UI unit that acts as an acknowledgement for the LCD screen to be cleared after the user has seen the warning. The LCD screen will also warn the user if the battery in the refrigerator needs to be changed based on a message from the internal unit after studying any drop in voltage across the battery.
The sensing unit can be placed anywhere inside a refrigerator or a storage unit in proximity to food. The internal subsystems can communicate with the display unit placed outside the refrigerator over Bluetooth modules that are paired with each other. We decided to go ahead with this design because this design was scalable and could be manufactured economically as compared to the currently present electronic nose systems used in laboratories.
Note: There are certain components common to both the subsystems are the design details of those have been provided before diving into the details specific to each of the subsystems.



2.1 Sensing Unit

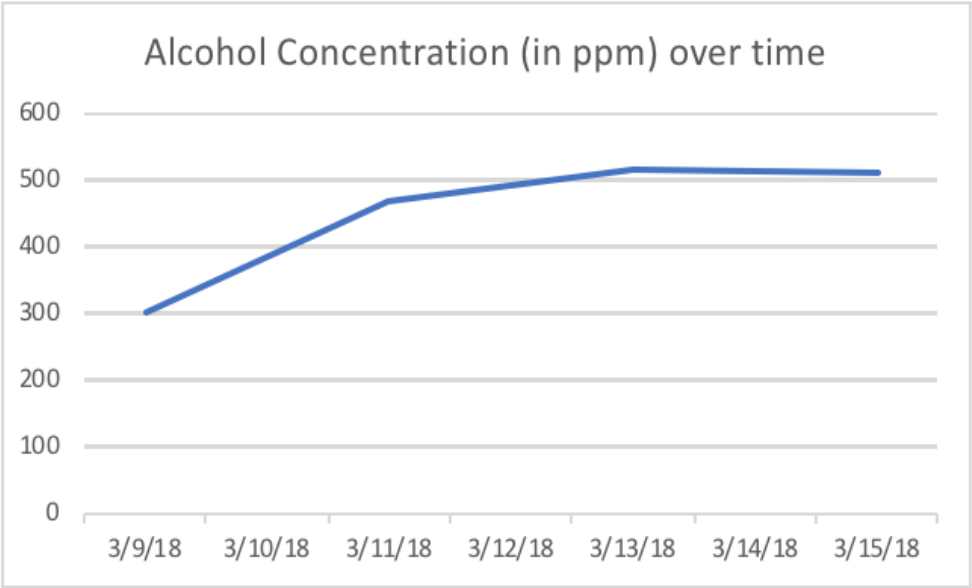
This is the subsystem that is responsible for detecting food detection and communicating with the UI unit to warn the user of any potential spoilage. This unit functions in cold and humid conditions and can be placed in refrigerators and storage units as per the users convenience.

2.1.1 E-Nose Sensor Array

After studying certain white papers and analyzing trends in food spoilage, we realized that the gases that are released most often are CO_2 and certain types of alcohols and organic solvents like ethanol. This is because the bacteria numbers increase on food spoilage. Most papers mentioned studies that stated that when certain foods like apples are about to get spoiled, the concentration of CO_2 is 0-5%VOL. The ranges for alcohol are around 20-100 ppm for rotten meat. We included a temperature sensor to calibrate the readings based on the temperature of

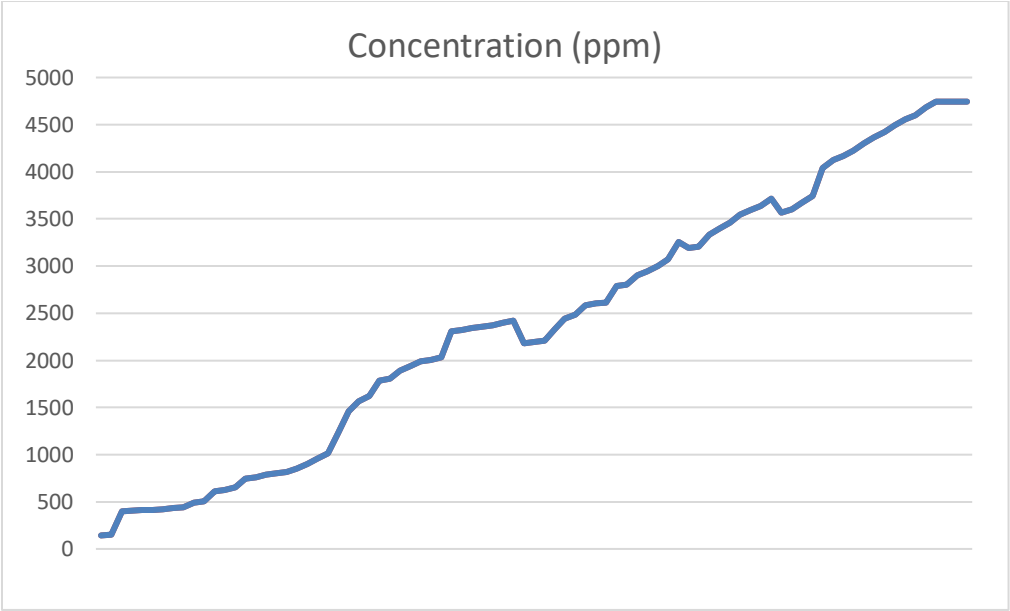
the surrounding air because the sensitivity of the gas sensors as well as the rate at which food gets spoiled varies in different temperature and humidity conditions. Based on studies conducted, the optimal average relative humidity for refrigerators is within a range of 70-80% at 4°C. The sensors that we are using are as follows and the tests have shown that they work as desired in cold and humid conditions. Thus, they are suitable for refrigerator conditions as desired by the use of our product.

Sensor	**Chemical Detected**	**Specifications**	**Protocol Used**
TGS822	Organic Solvents, Alcohol	Heater Voltage : 5.0±0.2V (AC or DC) I = 132 mA (typ) Ps ≤ 15 mW; Ph = 660 mW (typ)	Voltage readings to be inferred – Analog Pins on the MCU
MH-Z14	CO_2	Working voltage: 4.5 V ~ 5.5V DC Average current: < 85 mA Interface level: 3.3 V	UART
DHT22	Temperature and Humidity	Source Voltage 3.3V - 6V Typical: 3.3V Power Consumption: 7.5 mW Current: 1.5 mA (typ)	Single bus serial interface



X-axis: date
Y-axis: Alcohol concentration (in ppm)

The above figure shows incremental readings of the alcohol sensor as the banana rots over a week.

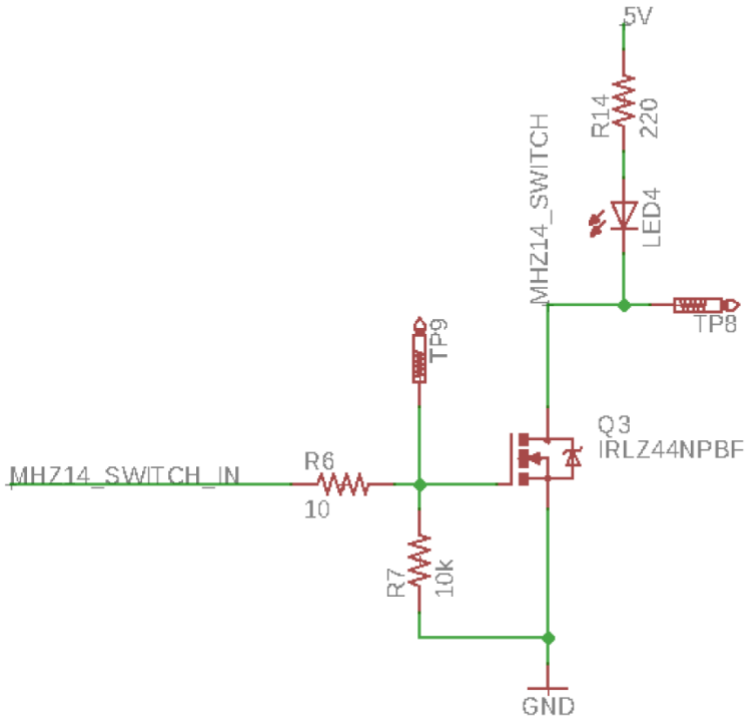


The above figure shows incremental readings of the CO2 sensor as the banana rots over a week.

2.1.2 Switching Circuitry

When all the devices on the sensing unit are powered, there is a huge power loss due to the fact that these gas sensors have heating elements to help with sensing. This meant we had to come up with a power management solution. We decided to add digital switches using NMOS to power each of the sensors from the MCU. The grounds of all the sensors are connected to the MCU through a switch. The switch is operated with signals from the MCU program so that the power cycles can be operated efficiently. Currently, the switching circuitry gets triggered on every 2 hours with an operational time of 15 minutes. We leave the circuit on for 15 minutes because the sensors take around 5-8 minutes to heat up and stabilize to an impedance level so that they can start getting the readings. The readings are computed over the span of the next 5-7 minutes. Because the sensors are slow to react to the gases, this window provides enough time for the readings to be registered and processed by the MCU before the packet is created and sent out over the Bluetooth channel. Time is kept track of by the system clock since the time the program starts. The overflow window for the clock is approximately 50 days and because the batteries are swapped approximately every 2 weeks depending on the power consumption, this issue should not occur as the system gets instantiated each time it is turned on. NMOS (IRLZ449PBF) is used to achieve the switching functionality.

The circuit is the same for all the sensors and is described in the figure below. The Bluetooth module is never powered off.



The above digital switch circuit is specific to MHZ14 but an identical circuit is implemented for the other sensors as well.

2.2 Display Unit

2.2.1 LCD Display

We have integrated the circuitry with an LCD display that can warn the user of potential food contamination detection or of power warnings. In addition to that, the UI has an acknowledge button that is simply a hardware interrupt to clear the message displayed on the screen indicating that the user has acknowledged the current notification from the system and taken appropriate measures for the current message. Even if the user acknowledges the messages but does not clear the food, the system will generate the messages again after the next cycle of detection.

We are using Seeed Grove 16x2 LCD screen with RGB backlight for this purpose so that in addition to an updated message, the screen can also change colors. The system can display two lines of text and uses the I2C protocol to communicate with the MCU. Additionally, it is powered with 5V and thus the set of AA batteries is sufficient to make the system work. The Bluetooth receiver placed alongside accepts the packets from the internal sender and displays the appropriate message on the LCD screen as programmatically indicated by the MCU. The LCD's I2C connector is connected to the PCB using a universal 4 pin hub.

2.2.2 ACK Button

To make the user interface unit more interactive and as a means to enable the sensors to get zero calibrated from time to time to make sure that the readings are appropriate, we have placed an ACK button near the LCD screen. This is a SPST Tactile Push Button that when pressed raises an interrupt. This triggers the subroutine to clear the screen and send a packet back to the Master Bluetooth module placed on the sensing unit to recalibrate the sensors for future runs. The internal Bluetooth module is always functional so it can receive a packet and process it. When the sensors are turned on, based on the flag set, they are recalibrated if the acknowledgement button was pressed while the sensors were in sleep mode.

2.3 Components common to both the subsystems

2.3.1 Power

The main components of the power supply include 4 AA batteries of 1.5V each providing 6V that will be used to power the sensor unit. The voltage requirements of each of the individual

units are different and lesser than 6V. We changed the requirements and verifications from our previously proposed requirements because we realized that all the components can be powered with 5V or under. Additionally, we moved from rechargeable batteries to using AA batteries because of the convenience factor to the user associated with it. Any off-the-shelf batteries placed in a battery holder with a jack that can connect into a DC barrel power jack is suitable to power the subsystems.

2.3.2 Voltage Regulator

Linear regulators are used to make sure that each of the units get the required voltage. The linear regulators have an enable mechanism to switch off power supply when not required. They are used to step down voltage from 6V to 5V because all the components can be powered using 5V as determined by running tests with the required set of sensors. We are using LDL1117S50R voltage regulator to achieve this functionality.

2.3.3 Microcontroller

The microcontroller is the main control unit of the entire system. The MCU communicates with the different blocks of the product to execute the seamless functional design of the contamination detector. Both the subsystems use the ATMega328 MCU for the project as it provides two UART communication pins, I2C communication pins and several analog pins that can be used for communicating with various devices specific to each of the sub units.

- HC05 Bluetooth modules use the UART interface provided by the MCU to communicate with it. These are RX and TX pins present on the MCU.
- The MHZ14 CO2 sensor uses the analog pins A0 and A1 in conjugation with the Software serial library to run the UART protocol.
- The TGS822 sensor uses analog pins on the MCU.
- The LCD screen uses SDA and SCL to communicate with the MCU using the I2C protocol.
- DHT22 communicates with the MCU over a serial bus.
- The ACK pushbutton is connected to the MCU using a digital pin.
- The switching circuitry for the operation of these sensors utilizes the digital pins.

2.3.4 Bluetooth Module

To aid with the user experience and to enable modularity in the design, we included a Bluetooth module that has the transmitter placed with the sensor array unit and the receiver on the User Interface unit so that both these blocks can be placed separately as per the user's convenience. This helps in situations where the notification screen would be placed on the door of the refrigerator and the electronic noses can be placed at multiple locations inside the refrigerator and can wirelessly communicate with this central node. This enables scalability in

the design. Additionally because of the range of signal needed for communicating in our proposed use-case environments, Bluetooth is a feasible solution for enabling wireless communication. The sensors don't send data very frequently and thus the bitrate can be handled well by Bluetooth as well. The operational bitrate is a few bits per second.

The HC05 Master is placed on the sensor unit and it sends a power or a food contamination warning to the HC05 Slave placed on the UI subsystem. The master creates single character packets when either of the thresholds are crossed. If the sensors detect presence of CO2 or alcohol beyond a certain point, it sends a 'F' packet to the receiver that translates to 'Time to bake banana bread' message on the LCD. The Bluetooth module also receives power input value by checking the voltage on one of the MCU pins. If the value read is below a certain threshold, a packet 'P' is sent to the receiver indicating that the batteries need to be changed. If food warning and power warning have an overlap, both the messages are displayed on the UI and this is handled in software. The pairing of the two modules makes the channel secure and because of this configuration set in the device using AT commands, it is harder to breach into the communication channel.

3. Design Verification

3.1 Sensing Unit

6V Power Supply	
Unit Name	Sensor Unit
Requirement	Provide 6V constant voltage
Process	Connect the Battery pack to the PCB using barrel jack
Reference	Appendix C Fig. 1 shows that the power supply provides a constant 6V supply voltage

LDO output 5V Voltage Level	
Unit Name	Sensor Unit
Requirement	Provide 5V constant voltage for the PCB
Process	Connect the Battery pack to the PCB using barrel jack and probe the output
Reference	Appendix C Fig. 2 shows that the LDO outputs a constant 5V voltage

TGS822 Alcohol Sensor Voltage Level	
Unit Name	Sensor Unit
Requirement	Changing voltage level for changing concentration of alcohol content in air
Process	Switch ON TGS822 and probe the output of the alcohol sensor while changing alcohol content
Reference	Appendix C Fig. 6 shows changing voltage level for changing concentration of alcohol in air

DHT22 Temperature Sensor Data Line	
Unit Name	Sensor Unit
Requirement	Correct timing of the digital serial data line as given by the datasheet
Process	Switch ON DHT22 and probe the data line of the sensor
Reference	Appendix C Fig. 7 shows that the timing diagram is in accordance with the graph obtained

MHZ14 CO2 Sensor UART Lines	
Unit Name	Sensor Unit
Requirement	Correct timing of the UART Tx and Rx lines of the sensor as per the datasheet while a communication is triggered between the MCU and MHZ14

Process	Switch ON MHZ14 and probe UART Tx and Rx lines while establishing a continuous communication between the two
Reference	Appendix C Fig. 8 & 9 show the communication diagram verified for TX and RX lines between the MCU and MHZ14

HC05 Bluetooth UART Lines	
Unit Name	Sensor Unit
Requirement	Correct timing of the UART Tx and Rx lines of the sensor as per the datasheet while a communication is triggered between the MCU and HC05
Process	Switch ON HC05 and probe UART SCL and SDA lines while establishing a continuous communication between the two
Reference	Appendix C Fig. 10 & 11 show the timing diagram verified for the UART TX and RX of the Bluetooth sensor with the MCU

Digital Switch	
Unit Name	Sensor Unit
Requirement	Turn ON the device and associated LED when the digital switch goes high(5V)
Process	Probe the drain of the MOSFET to while the digital switch turns ON
Reference	Appendix C Fig. 12 shows the change in voltage as the switch goes high when the device is turned ON

MHZ14 CO2 Concentration Reading	
Unit Name	Sensor Unit
Requirement	A slowly increasing concentration of CO2 throughout the process of contamination of banana
Process	Allow a banana to rot while taking periodic readings of CO2 gas concentration emitted during the process
Reference	Appendix C Fig. 17 shows the increase in concentration of CO2 throughout the process of banana contamination

TGS822 Alcohol Concentration Reading	
Unit Name	Sensor Unit
Requirement	A slowly increasing concentration of alcohol throughout the process of contamination of banana
Process	Allow a banana to rot while taking periodic readings of alcohol concentration emitted during the process
Reference	Appendix C Fig. 18 shows the increase in concentration of alcohol throughout the banana contamination

3.2 Display Unit

6V Power Supply	
Unit Name	Display Unit
Requirement	Provide 6V constant voltage
Process	Connect the Battery pack to the PCB using barrel jack
Reference	Appendix C Fig. 3 & 4 show that the power supply provides a constant 6V supply voltage

LDO 5V Voltage Level	
Unit Name	Display Unit
Requirement	Provide 5V constant voltage for the PCB
Process	Connect the Battery pack to the PCB using barrel jack and probe the output
Reference	Appendix C Fig. 5 shows that the LDO outputs a constant 5V voltage

HC05 Bluetooth UART Lines	
Unit Name	Display Unit
Requirement	Correct timing of the UART Tx and Rx lines of the sensor as per the datasheet while a communication is triggered between the MCU and HC05
Process	Switch ON HC05 and probe UART Tx and Rx lines while establishing a continuous communication between the two
Reference	Appendix C Fig. 13 & 14 show the timing diagram verified for the TX and RX lines between the MCU and HC05

LCD Display I2C Lines	
Unit Name	Display Unit
Requirement	Correct timing of the I2C SCL and SDA lines of the sensor as per the datasheet while a communication is triggered between the MCU and LCD
Process	Probe I2C SCL and SDA lines while establishing a continuous communication between the two
Reference	Appendix C Fig. 15 & 16 show the timing verified for I2C communication between the LCD and the MCU

4. Costs

4.1 Parts

S. No.	Manufacturer	Part Number	Description	Qty	SMD/DIP	Price/unit	Total Cost
1	Microchip	ATmega328P-PU	Low-power 8-bit MCU	2	PDIP	2.08	4.16
2		HC-05	Bluetooth Module	2	Through-hole	11.11	22.22
3	TXC, Mouser	DIP HC-49S 9B SERIES	16MHz Crystal Oscillator	2	Through-hole	0.95	1.9
4			DC Barrel Jack Power Connector	2	Through-hole	0.95	1.9
5	ST Microelectronics	LDL1117S50R	6V to 5V LDO Regulator	2	SMD	0.4	0.8
6	Wurth Electronics	156120VS75000	Standard LEDs SMD 1206	4	SMD	0.18	0.72
7			6mm SPST Tactile Push Button	1	SMD	0.5	
8	Seeed		Universal 4 pin I2C connector	2	Through-hole	1.5	
9	Figaro	TGS822	Organic Solvent Vapor Sensor	2	Through-hole	3.045	6.09
10	Winsen Electronics	MH-Z14	Infrared CO2 Sensor	1	Through-hole	19.78	19.78
11	Adafruit	DHT22	Temperature and Humidity Sensor	2	Through-hole	4.14	8.28
12			4xAA battery holder	2	-		
13		IRLZ449PBF	NMOS	8			

14		CD1206-S01575	Diode	4	SMD		0.15
15	General		Passive components	several	SMD		

4.2 Labor

(For each member of group)
Salary: $40 /hr
Time per week: 8 hrs/week
Work Completion : 75%
Total = 40 $hr 8hrsweek 16 weeks.75 2.5= $17,066
Total labor costs = 3 17066 = $51,200

4.3 Schedule

Week	Siddharth	Simran	Agnivah
2/26	Build prototype circuit	Write testing code for communicating with sensors	Write testing code for communicating with sensors
3/5	Schematics, PCB layout	Data Collection	Data Collection
3/12	PCB v1 ordered	Testing of sensor readings	Creating signatures of different concentrations
3/26	PCB testing. PCB v2 redesign and order	PCB testing. Designing Display unit PCB	PCB testing. Final UI design start
4/2	Check power monitoring circuitry	Sensing unit processing and Bluetooth transmission	Bluetooth reception and processing for LCD display
4/9	Ordering parts	Bluetooth Pairing and UART communication debugging	Bluetooth Pairing and UART communication debugging
4/16	New PCB debugging	Code integration	Code Integration
4/23	Soldering on new PCBs	Final tests	Creating the casing

5. Conclusion

The Honey Nose is a modular food contamination detection system that can aid the user to eat healthier and prevent food wastage by warning him/her of spoilage well in advance. The easy UI helps abstract away the complexities of the biochemical system underneath. The units can be configured as per the user's food detection needs and can be placed at different locations inside refrigerators or storage units. The system is powered with AA batteries and facilitates convenience.

5.1 Accomplishments

- Our system can detect food contamination and based on signatures fed in specific to different foods after running experiments.
- The system can work in cold and humid conditions as well.
- Because the system can be customized to one food type, this system finds a great use case for industrial food storage units and refrigerators.
- It has a modular design and is can scale to larger storage units.
- The power mechanism is convenient and the UI is simple.
- It notifies the user both of power requirements and food contamination status.
- It is affordable and compact with a neat design that can be customized to fit any décor.

5.2 Uncertainties

- The system isn't robust enough to work with different types of foods and hence isn't the best fit for households.
- The sensors take time to heat up and calibrate to an impedance value to be ready to take readings.
- The processing time of the unit isn't very power efficient although our switching mechanism helps with the power aspect of the system.
- To switch to a different food item and to configure the system to adopt to it, a period of weeklong signature generation is required before the system can be used by the customer.

5.3 Ethical considerations

Potential applications of this device could be businesses using it to monitor the quality of their produce or food items they are selling. This device could be tampered to show wrong results and deceive the consumers. Since the device is not connected to the network, hacking into it

remotely would not be possible. However, someone could reverse engineer and change the threshold of contamination detected by the sensors thereby declaring something as fresh to consume, despite being of inferior quality. This is in direct violation to #3 of the IEEE Code of Ethics which is “to be honest and realistic in stating claims or estimates based on available data”. Moreover, since the Bluetooth modules have been paired with each other, interception and modification of packets is harder.

Our project doesn’t have serious potential hazards as such. One particular potential hazard could be the short circuit of the sensors. Since the sensor array is placed inside the refrigerator, a potential liquid spill could damage the e-nose sensors and cause a short-circuit. Since it is composed of gas sensors, complete waterproofing of the sensor array is not possible. The casing would adhere only IP61 guidelines, which protects it from condensation.

5.4 Future work

Our system currently detects spoilage of one particular food item, namely bananas. This includes collecting threshold signatures of multiple different food items and creating a signature database. After having the signature database, we plan on improving our algorithm to be able to handle the overlapping signatures and detect food spoilage correctly. By detecting spoilage of a wide range of food types, we plan on making our system more robust.

Another feature that we plan on adding after creating the threshold database is 'mode selection'. A separate 'mode' button on the display unit will allow the user to select the type of food being monitored and calibrate the sensors accordingly. This feature will particularly be helpful for industry applications and prevent the need to disassemble and reprogram the MCU whenever the type of food item being monitored changes.

Lastly, we plan on improving the sensor array with more sophisticated sensors that have a lower preheating time and are faster in their detection. This will further save on power usage and improve battery life of our device.

References

[1] N. Benabdellah, M. Bourhaleb, N. Benazzi, M. B. Nasri, and S. Dahbi, “The Detection of Smell in Spoiled Meat by TGS822 Gas Sensor for an Electronic Nose Used in Rotten Food,” Advances in Intelligent Systems and Computing Europe and MENA Cooperation Advances in Information and Communication Technologies, pp. 279–286, 2016.

[2]S. Balasubramanian, S. Panigrahi, C. Logue, C. Doetkott, M. Marchello, and J. Sherwood, “Independent component analysis-processed electronic nose data for predicting Salmonella typhimurium populations in contaminated beef,” Food Control, vol. 19, no. 3, pp. 236–246, 2008.

[3]N. Benabdellah, K. Hachami, M. Bourhaleb, and N. Benazzi, “Identification Of Two Types Of Rotten Meat Using An Electronic Nose For Food Quality Control,” International Journal on Smart Sensing and Intelligent Systems, vol. 10, no. 3, pp. 673–695, 2017.

[4]S. N.divekar and S. S.n.pawar, “PIC Microcontroller and PC based Multi Sensors Artificial Intelligent Technique for Gas Identification,” International Journal of Computer Applications, vol. 121, no. 14, pp. 34–38, 2015.

[5] A. Berna, “Metal Oxide Sensors for Electronic Noses and Their Application to Food Analysis,” Sensors, vol. 10, no. 12, pp. 3882–3910, 2010.

[6] “eNose Technology,” The eNose Company. [Online]. Available: http://www.enose.nl/rd/technology/. [Accessed: 08-Feb-2018].

[7] “Ketosense – An Arduino based ketosis detector,” jenslabs, 16-Feb-2015. [Online]. Available: https://jenslabs.com/2013/06/06/ketosense-an-arduino-based-ketosis-detector/. [Accessed: 19-Feb-2018].

[8] “Calibrating a Figaro TGS822 sensor, by drawing...,” jenslabs, 21-Mar-2013. [Online]. Available: https://jenslabs.com/2013/03/21/calibrating-a-figaro-tgs822-sensor-by-drawing/. [Accessed: 19-Feb-2018].

[9] Wills, R. and Golding, J. (2016). Postharvest. Wallingford: CABI, pp.63-67.

[10] HC-05 Bluetooth Module User Manual. (2018). 1st ed. [ebook] Available at: https://www.gme.cz/data/attachments/dsh.772-148.1.pdf [Accessed 26 Feb. 2018].

[11] Grove - LCD RGB Backlight. (2018). 1st ed. [ebook] Available at: http://www.west-l.com/uploads/tdpdf/104030001_eng_tds.pdf [Accessed 26 Feb. 2018].

[12] Digital-output relative humidity & temperature sensor/module. (2018). [ebook] Available at: https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf [Accessed 26 Feb. 2018]

[13] Intelligent Infrared Carbon Dioxide Module. (2014). 2nd ed. [ebook] Available at: http://www.winsen-sensor.com/d/files/PDF/Infrared%20Gas%20Sensor/NDIR%20CO2%20SENSOR/MH-Z14%20CO2%20V2.4.pdf [Accessed 26 Feb. 2018]

[14] TGS 822 - for the detection of Organic Solvent Vapors. (2002). [ebook] Available at: http://www.figarosensor.com/products/822pdf.pdf [Accessed 26 Feb. 2018]

Appendix A Requirement and Verification Table

Power:

Requirements	Verification
A. The battery should be able to provide 5 V	A. Connect the probes of battery to a voltmeter and verify it is greater than 5V when fully charged

Observations: We are using four 1.5V batteries to power the system. We connected the probes of a multimeter to measure the voltage across the battery. Reading obtained = 5.84 V

Voltage Regulator:

Requirements	Verification
A. The voltage regulator provides 5V from a 6 V source	A. Measure the output voltage using an oscilloscope, ensuring that the output voltage stays within 5% of 5V for a simple load circuit.

Observations: Plug in the battery, probe the test point for input and the output of voltage regulator. Readings - approximately 6V and 5V respectively.

Note: Because the requirements for the power system changed, the voltage regulator requirements changed as well. We now need the output voltage from the regulator to be 5V because all components work with 5V.

E-Nose Sensor Array:

Requirements	Verification
A. Alcohol sensor should be able to detect in the range 20-100 ppm B. CO2 sensor should be able to detect in the range 0-5% VOL	• Verify working of the sensor using substances that are proven to produce the target gas • We set up a closed chamber with a food specimen and the sensor powered to record readings over a span of a few days at a

C. The sensors should be able to operate between 0°C - 10°C D. The sensors should be able to detect and function in humid conditions with average relative humidity of 70-80% at 4°C	temperature of around 4°C - and at an average relative humidity of 70-80%. • We create a signature correlating the concentration of the gas in air due to spoilage to the voltage change. • Using the data sheet of the sensors and doing necessary calculations, we verify that the concentration falls in the required range. *The above steps will be used to verify all the requirements.

Observations:

- The sensors worked in cold and humid conditions in the fridge.
- We noted a change in the voltage and incremental change in readings as food got spoiled over time.
- The datasheet of TGS822 is used with DHT22 to get the formulation for the value calibration with humidity and temperature.
- The computation for scaling factor is → scalingFactor = (((currentTemperature * -0.02573)+1.898)+((currentHumidity*-0.011)+0.3966));

<u>Micro-Controller:</u>

Requirements	Verification
A. The MCU should have at least two UART to communicate with the CO_2 sensor and Bluetooth module at at least 9600 baud B. The MCU should be able to work with I2C protocol to communicate with the LCD screen of the UI C. The MCU should be able to communicate via a serial	A. Connect microcontroller to USB UART bridge and to a terminal. Set up terminal at 9600 baud. Send and echo back 100 characters. Ensure that all characters match those sent B. I2C protocol can be verified by requesting the device ID from the display. Ensure that the data received matches the specification in the data sheet C. The MCUs digital pins will be used to design the protocol that is followed by the sensor. Perform tests on the design to retrieve the

interface with the DHT22 sensor D. The MCU should have at least 2 pins to connect to the push-button circuit for the acknowledgement button	correct device ID to validate the serial interface D. Probe two pins with a multimeter to check if it has the voltage required by the push button switch at the two pins where it will be placed. Attach the button and notice a change in voltage value in software to see if the button gives the desired signal.

Note: We switched to Atmega328P from Atmega2560 because the above requirements could be met by this MCU.

Observations:

- The Atmega328P comes with a serial UART and we use this for communicating with HC05 Bluetooth module. For the other serial port requirement, we use the software serial library to make the MHZ14 CO2 module work with the MCU. It receives the data as desired and transmits packets as checked in the serial monitor. The baudrate can be changed as per the requirement too.
- I2C working is verified by printing messages on the LCD screen.
- Validation of digital pins through probing.
- Validation of button checked with multimeter and an LED switching circuit.

<u>User Interface:</u>

Requirements	Verification
A. A 16x2 LCD screen capable of displaying two lines of text in English with RGB backlight for clearer visibility. B. The working voltage of the display should be 5 V as that meets the power supply limits and operating current should be <100mA. C. The LCD screen should use I2C communication to interface with the microcontroller.	A. We connect the LCD to an arduino and print out two lines of text. If it displays them as desired, we know that the component is acceptable. B. We connect the LCD to a 5V supply and monitor its functioning. Additionally, we use an ammeter to measure the current passing through the circuit with the LCD being the only component. C. I2C protocol can be verified by requesting the device ID from the

	display. Ensure that the data received matches the specification in the data sheet

Observations:
We tested the LCD for all the above requirements both with the Arduino and PCB. We also included test pads in the previous PCB iterations to make sure that voltages were correct at the connector edges. Verified the functioning of the protocol by sending packets.

<u>Bluetooth Module:</u>

Requirements	Verification
A. The signal from the Bluetooth modules can pass through the refrigerator door and cover the range of a refrigerator. B. The Bluetooth module should be able to work with a serial communication interface to make sure that the MCU can communicate with it C. Operating voltage of the Bluetooth module should be around 5V.	A. Place a Bluetooth receiver attached to the CoolTerm Serial interface outside the refrigerator. Place the Bluetooth sender inside the refrigerator. If the MCU can generate values on CoolTerm, it implies that the serial interface of the module was functional. B. Upload a executable to the Bluetooth transmitter module to send packets with incrementing serial numbers and RSSI values as payload. Capture the packets on the receiver's end and print out the readings using the CoolTerm serial interface. If the packets are received, we know that the communication channel is functioning well. C. We use a voltmeter to measure the voltage across the bluetooth module and the ammeter to measure the current through it to make sure that those requirements are met. *CoolTerm is a simple serial port terminal application (no terminal emulation) that is geared towards hobbyists and professionals with a need to exchange data with hardware connected to serial ports such as

	servo controllers, robotic kits, GPS receivers, microcontrollers, etc.

Observations: We tested the transmission across the doors and checked the packets received with a two way button trigger code. The video can help check the validity and functionality.

The unit outside sends packets when a button is pressed.

```
//this is slave

#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11); // RX | TX
#include <Wire.h>
#include "rgb_lcd.h"

rgb_lcd lcd;
int r = 20;
int g = 90;
int b = 55;
int button = 2;

char state = '0';
String readString;
char c;
int val = 0;

void setup()
{
  BTSerial.begin(38400);
  Serial.begin(9600);
  Serial.write("This is slave");
  lcd.begin(16, 2);
  lcd.setCursor(0,0);
  lcd.setRGB(r,g,b);
  lcd.print("hello");
  pinMode(button, INPUT);
}

void loop()
{

  val = digitalRead(button);
  if(val == HIGH){
    lcd.print("ACK");
    BTSerial.write('R');
    delay(3000);
    lcd.clear();
```

```
    val = 0;
    c = 'A';
  }
  else{
    lcd.setCursor(0,0);
    while (BTSerial.available()) {
    //small delay to allow input buffer to fill
      c = BTSerial.read();  //gets one byte from serial buffer
    }

    if ( c == 'F' ) {
      lcd.setCursor(0,0);
      lcd.print("Food is bad");
      Serial.println("Food is bad");
      delay(2000);
      lcd.clear();
    }
    else{
      lcd.setCursor(0,0);
      lcd.print("all good");
      Serial.println("all good");
      delay(1000);
      lcd.clear();
    }
  }
}
```

The following is the code for the sensing unit and an LED blinks when the packet is received.

```
//this is master

#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11);

char state = '0';

void setup()
{
  BTSerial.begin(38400);
  Serial.begin(9600);
  Serial.write("this is master");
}

void loop()
{
 if(BTSerial.available() > 0)
 {
    // Checks whether data is coming from the serial port
    state = BTSerial.read(); // Reads the data from the serial port
```

```
 }

 if (state == 'R')
 {
  analogWrite(A0, 225);
  BTSerial.write('F');
  delay(2000);
  analogWrite(A0, LOW);
  delay(2000);
  state = 'P';
 }
 else{
  analogWrite(A0, LOW);
  BTSerial.write('P');
  delay(2000);
 }
}
```

B. We tested the HC05 Module with both the MCU's Serial TX and RX and the software serial enabled pins. The readings were checked on a serial monitor.

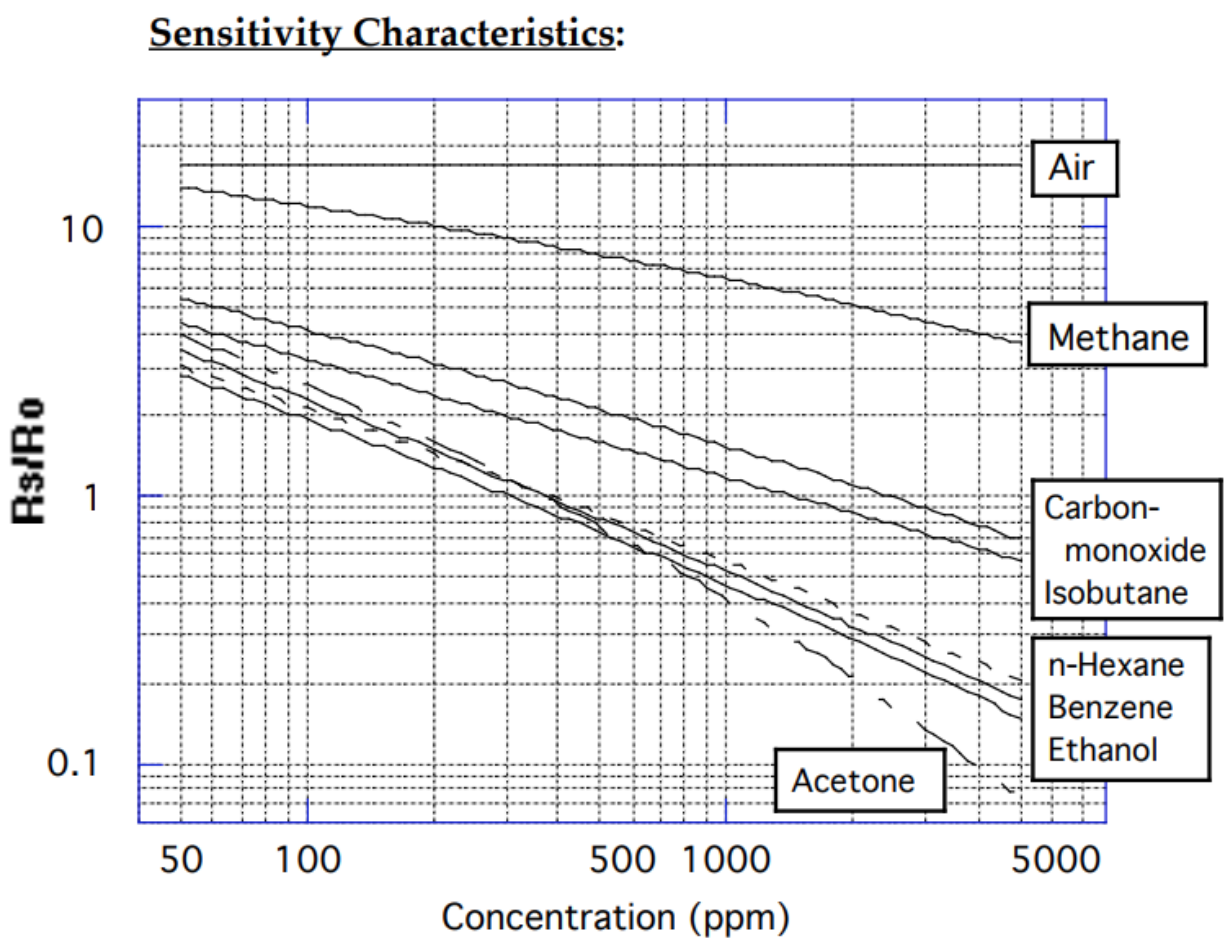
C. Probed the pins at the HC05 with a multimeter. Readings for Vin - 4.99 V as expected.

Appendix B Calculations

TGS822 sensor:

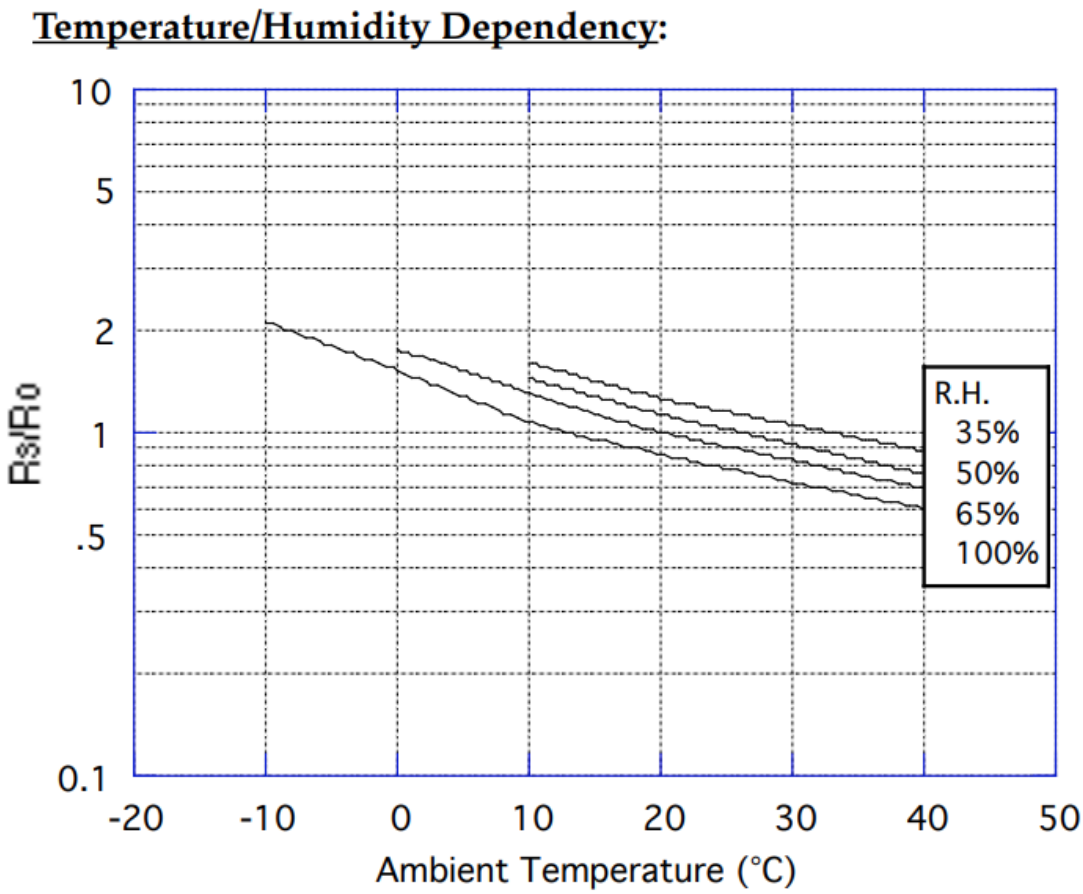
When it is connected as shown in fig.1, depending on the concentration of the gases, output across the Load Resistor (V_{RL}) increases as the sensor's resistance (Rs) decreases. Sensor Resistance (Rs) is calculated by the following formula:

$$Rs = (V_c / V_{RL} - 1)R_L$$



Rs = Sensor resistance of displayed gases at various concentrations

Ro = Sensor resistance in 300 ppm ethanol



Rs = Sensor resistance at 300 ppm of ethanol at various temperatures/humidities
Ro = Sensor resistance at 300 ppm of ethanol at 20°C and 65% R.H.

<u>MH-Z14 sensor module:</u>

This particular sensor module has multiple output formats:

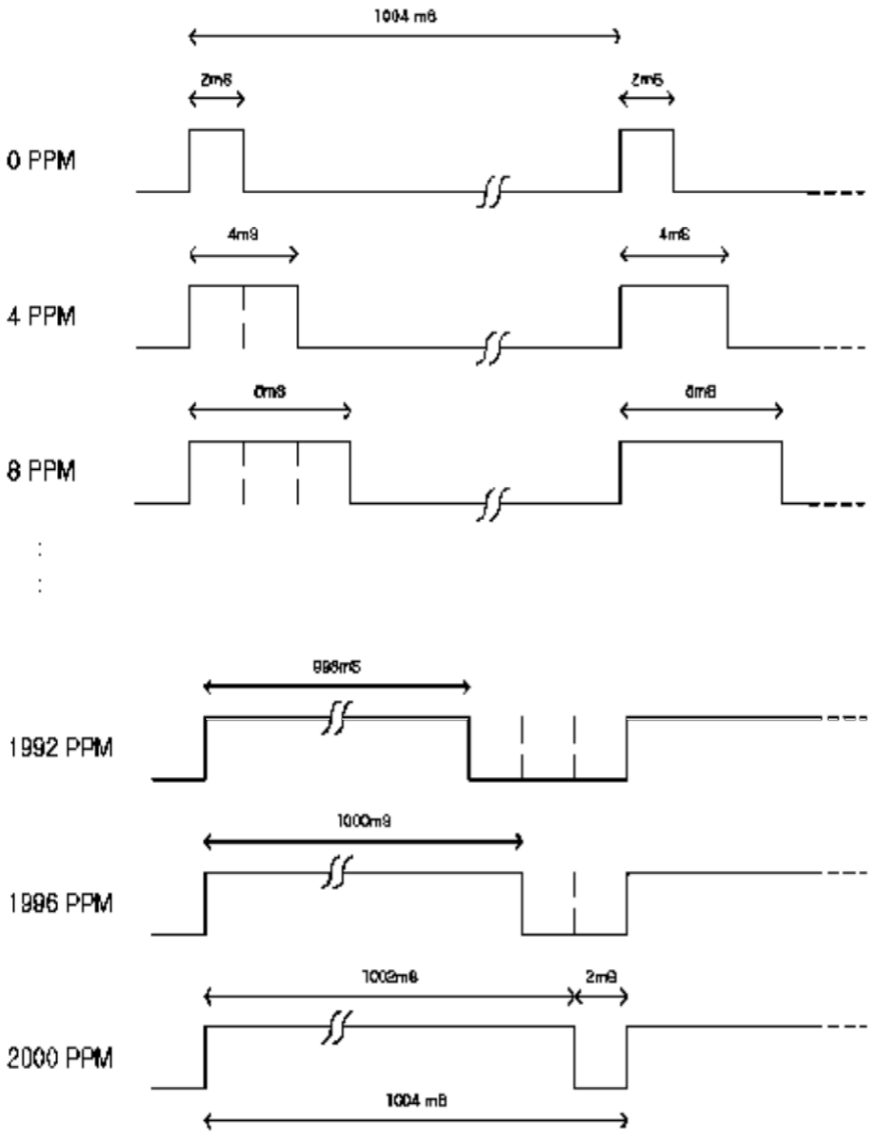
- PWM output - Formula for concentration of CO_2 from PWM output is as follows:

$$C_{ppm}=2000(TH-2ms) / (TH+ TL-4ms)$$

Where,

C_{ppm} ≡ calculated CO2 concentration, unit is ppm

$T_H \equiv$ time for high level during an output cycle
$T_L \equiv$ time for low level during an output cycle



- UART output - Concentration can be read without any calculations

Appendix C – RV graphs

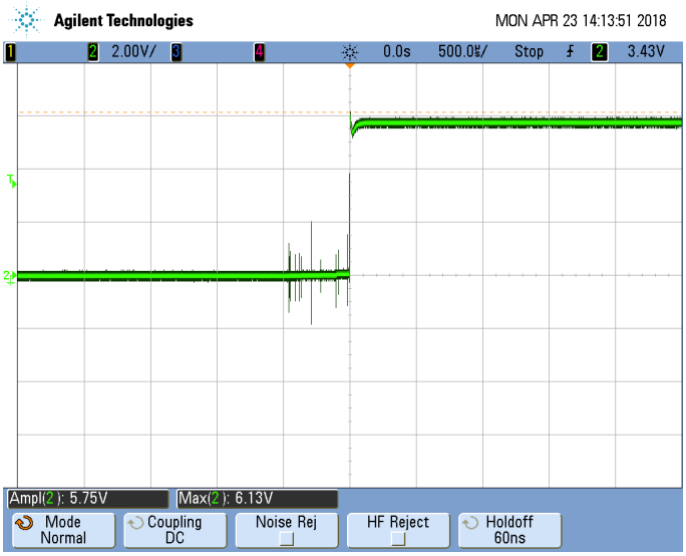


Fig. 1: Sensing Unit 6V Power

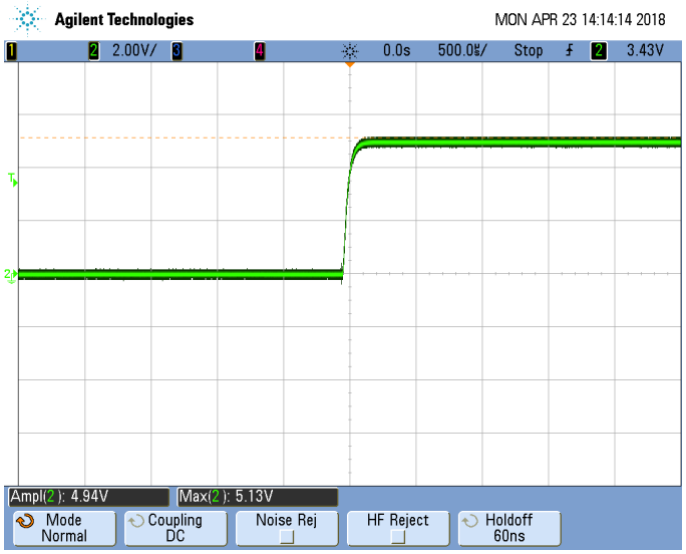


Fig. 2: Sensing Unit – LDO output

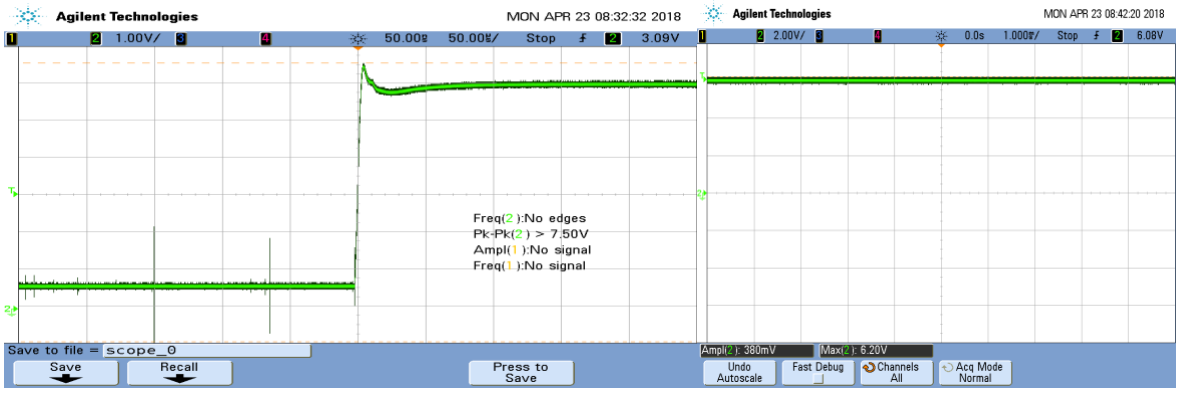


Fig 3. and Fig 4: Display Unit – 6V Power

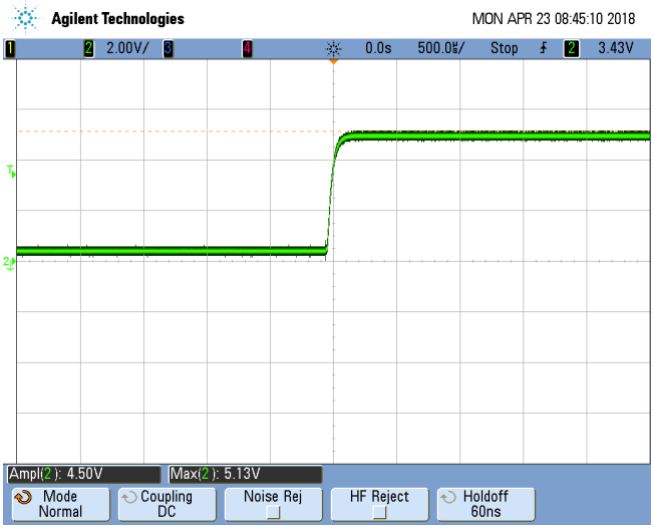


Fig. 5: Display Unit – LDO output

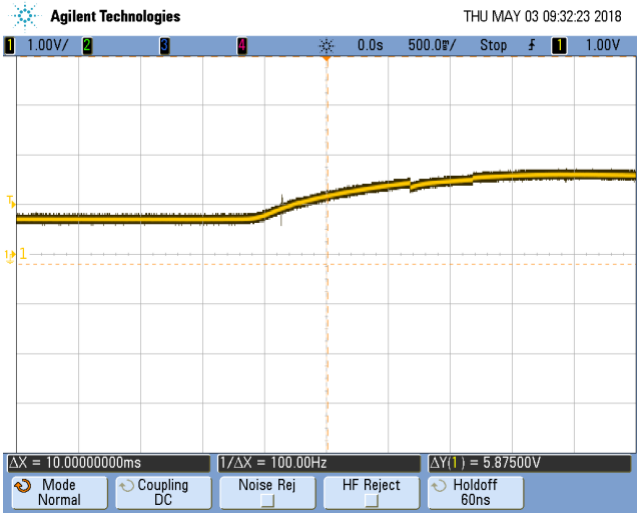


Fig. 6: TGS822 – Change in voltage with changing alcohol concentrations

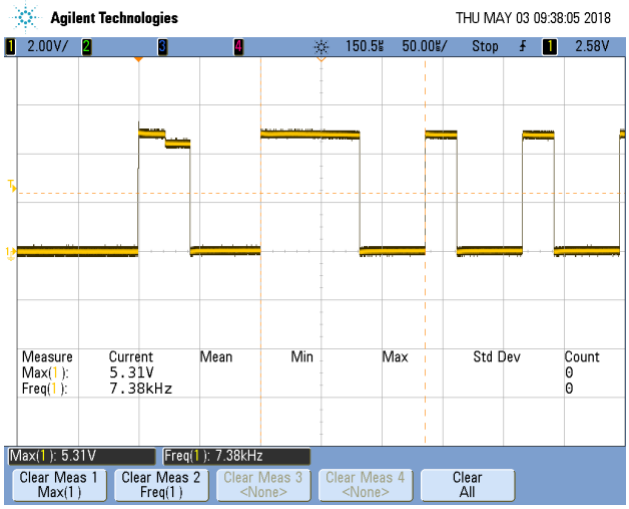


Fig. 7: Timing diagram of DHT22

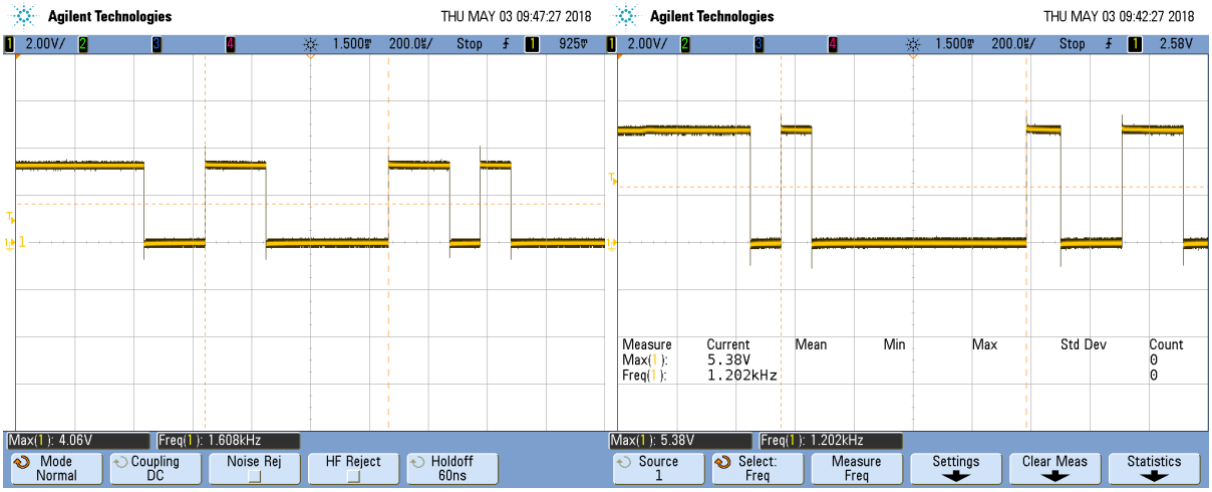


Fig. 8 and Fig. 9 : MHZ14 UART communication – TX , RX

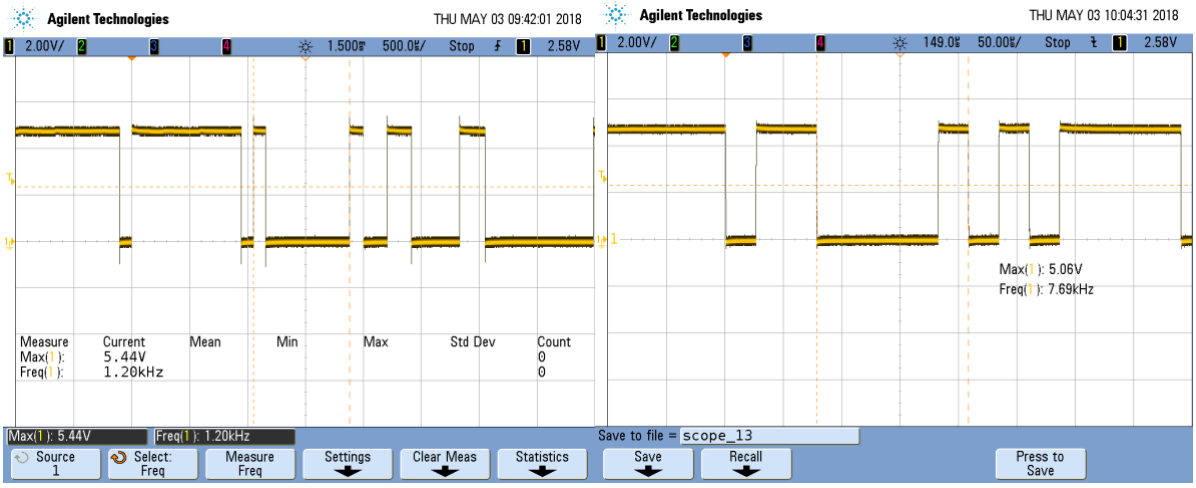


Fig. 10 and Fig. 11: Sensing Unit Bluetooth UART lines

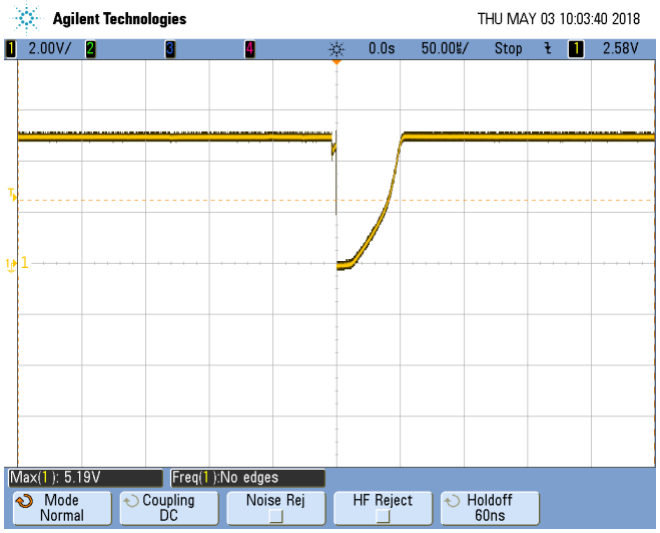


Fig. 12: Digital Switch

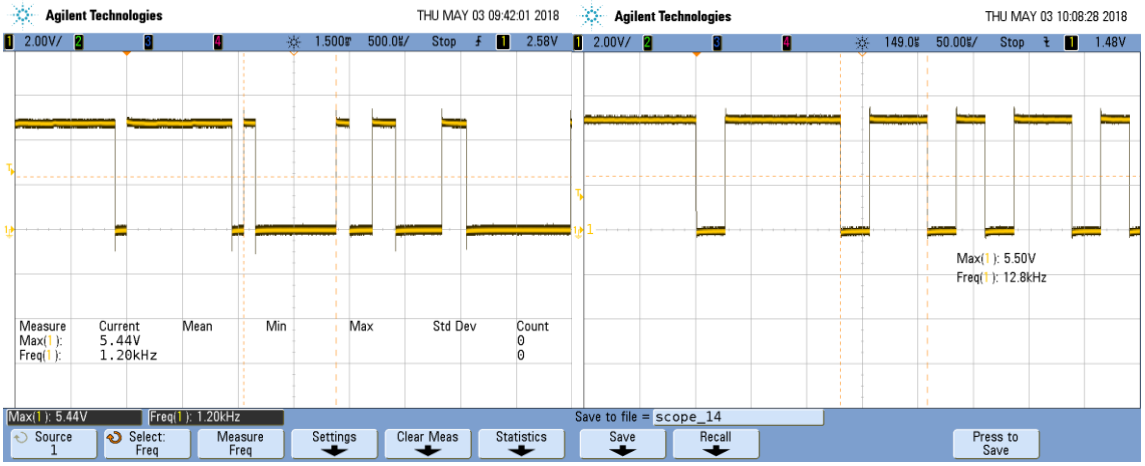


Fig. 13 and Fig. 14: Display Unit Bluetooth UART lines

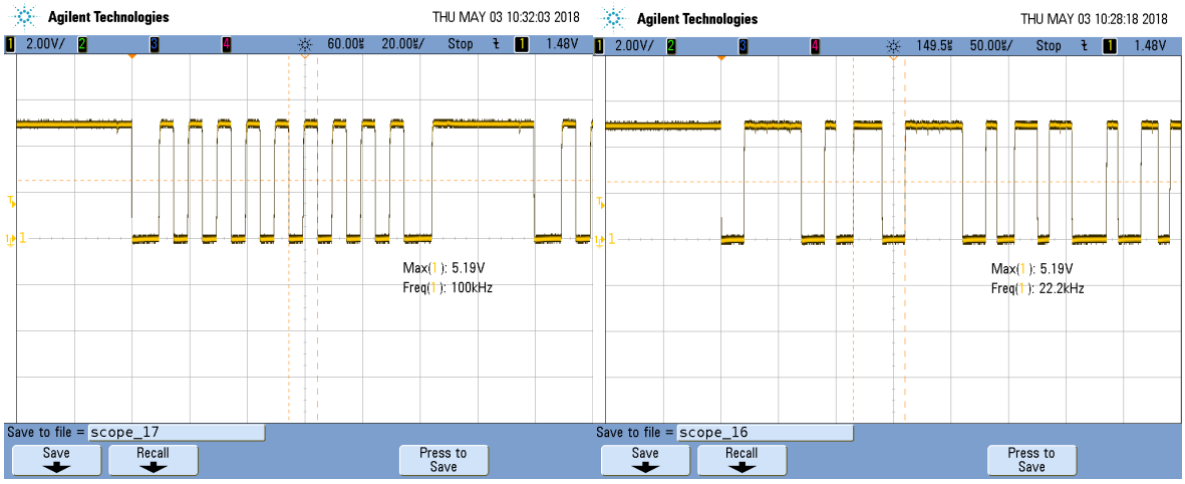


Fig. 15 and Fig. 16: LCD display SCL and SDA lines (I2C)

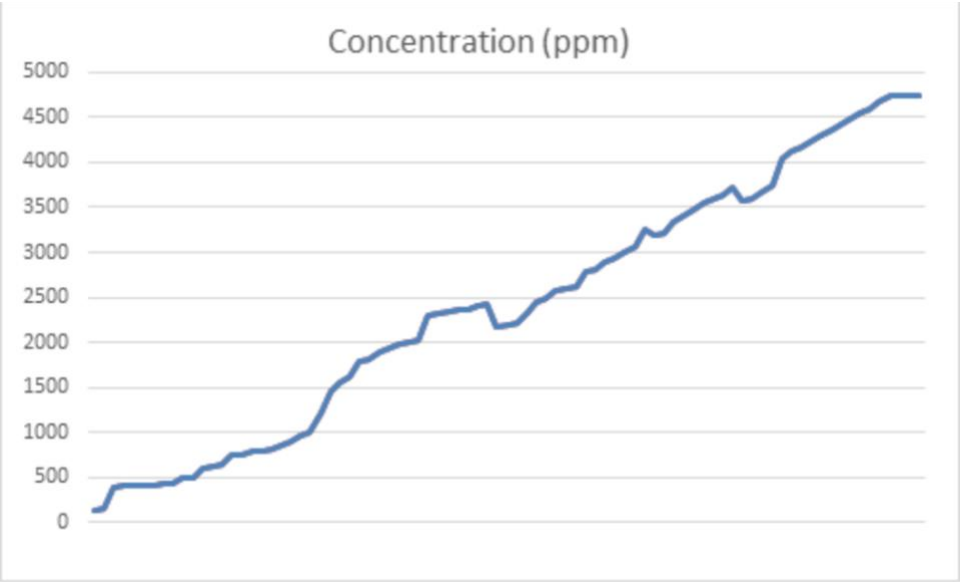
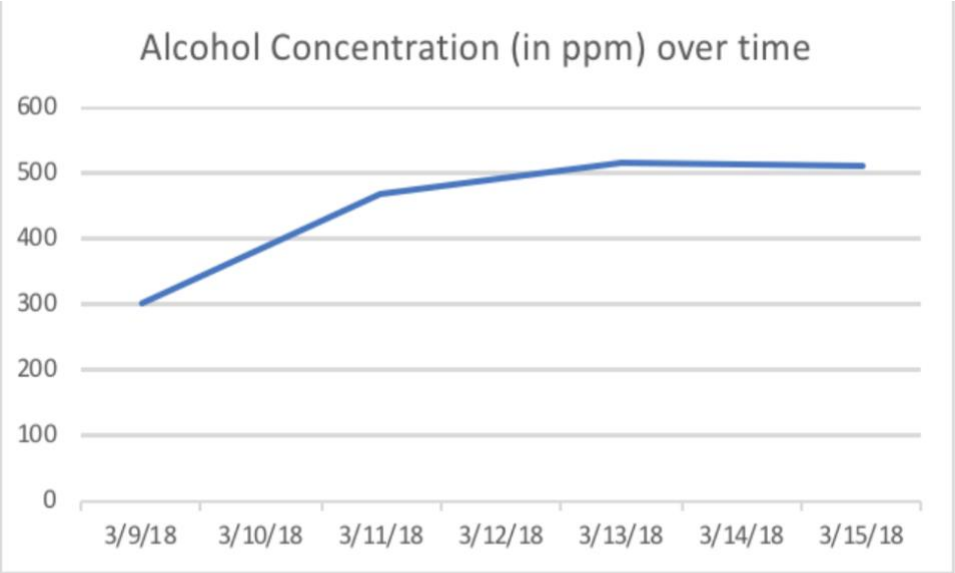


Fig. 17: CO_2 concentration signature (Concentration vs. Time)



X-axis: date
Y-axis: Alcohol concentration (in ppm)

Fig. 18: Alcohol signature (TGS 822)

Appendix D - Final Prototype

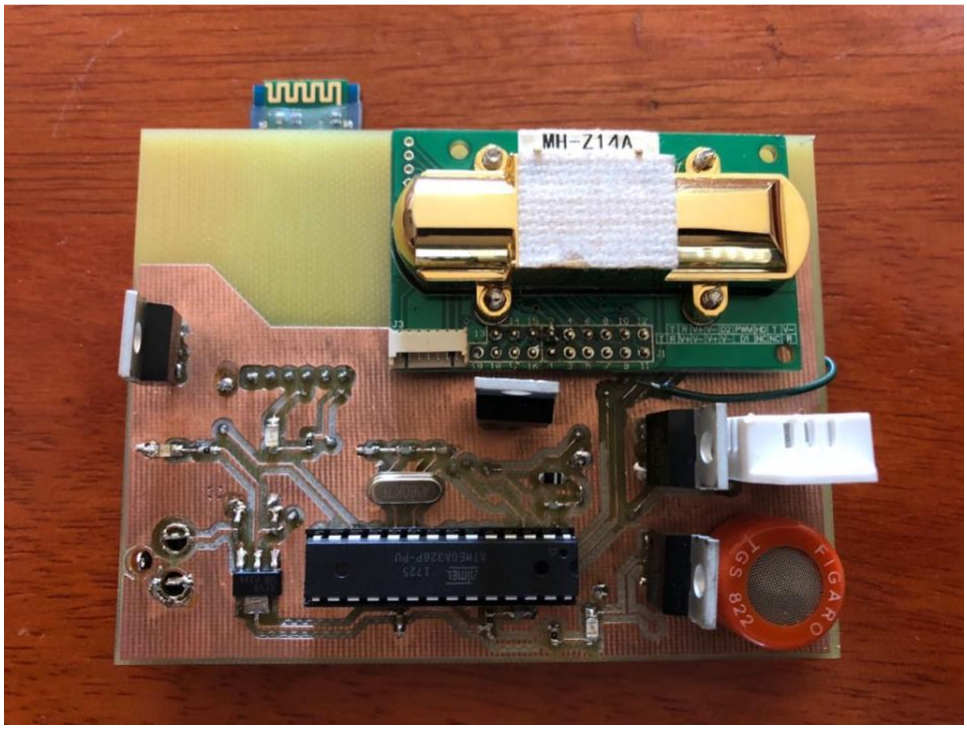


Fig 1: Sensing Unit – PCB

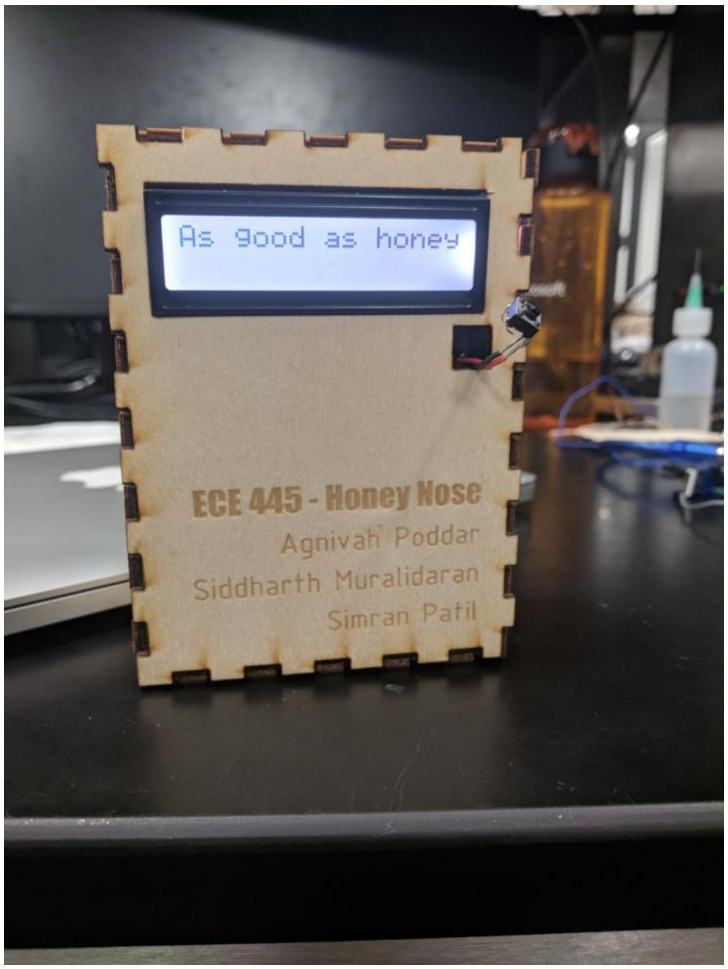


Fig 2: Display Unit with 'No spoilage' message

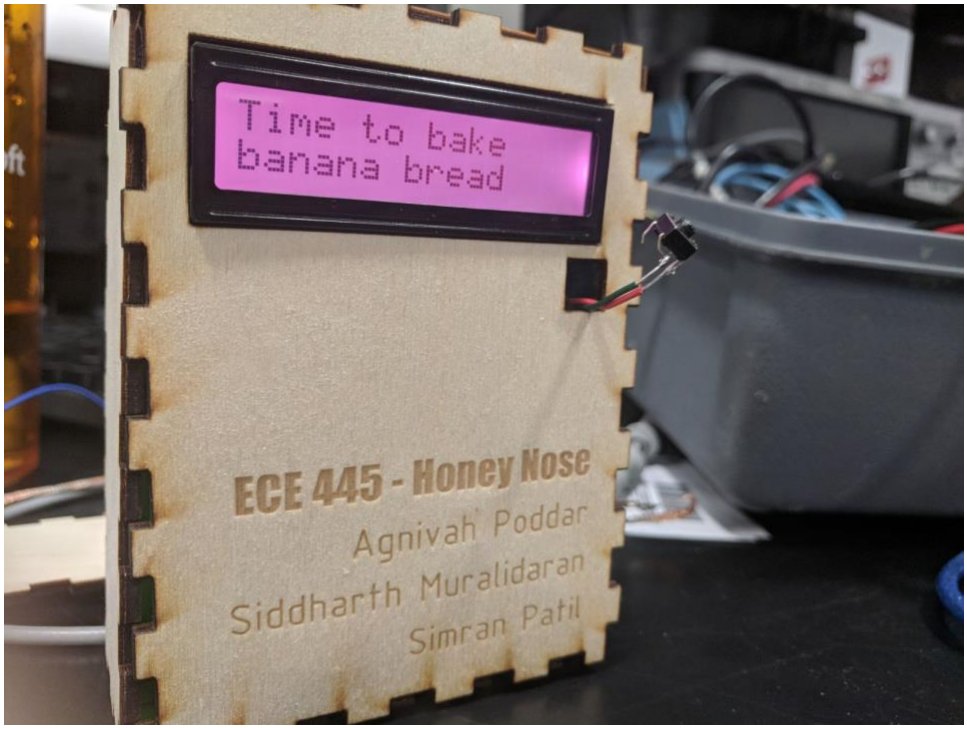
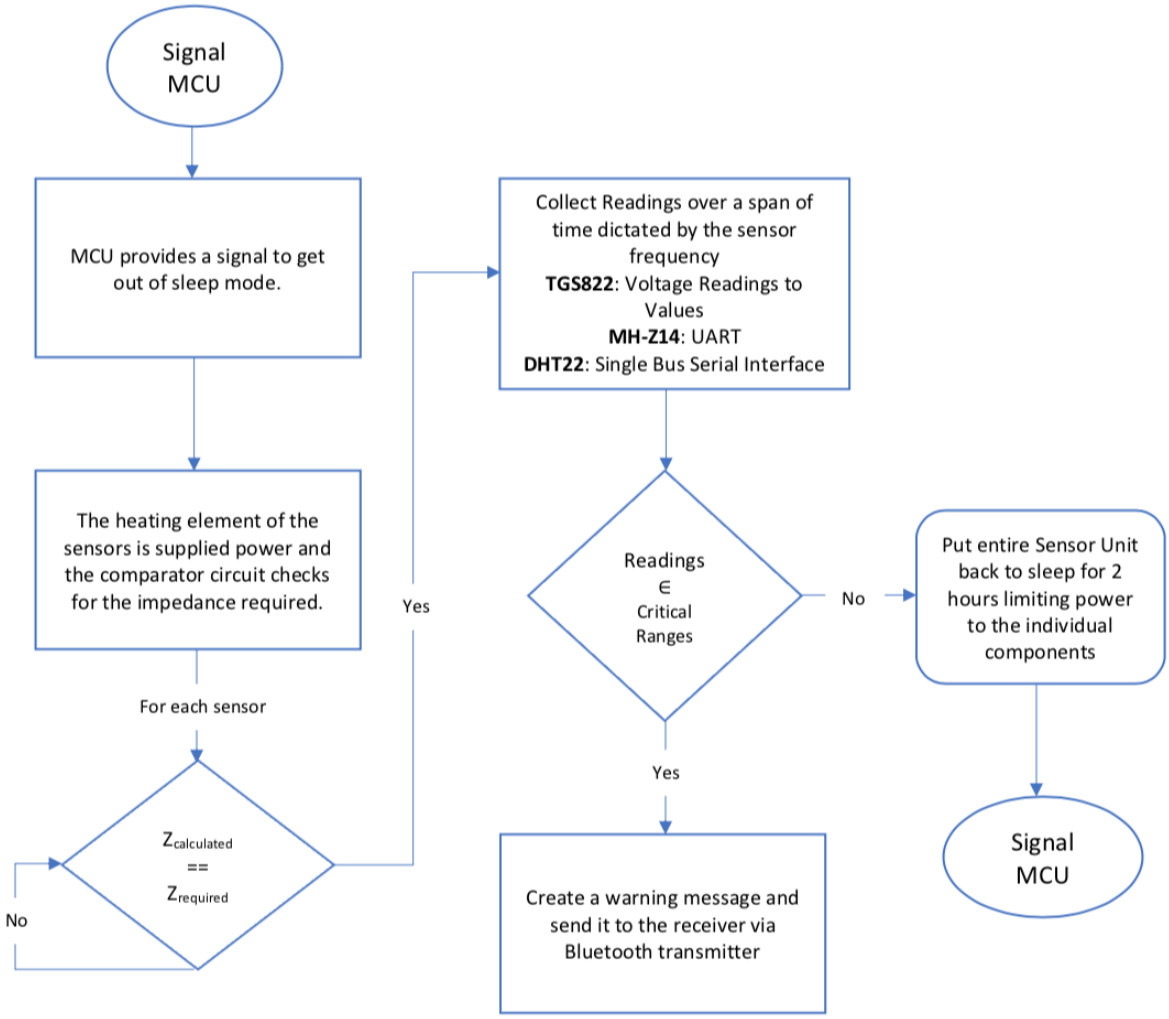


Fig 3: Display Unit with 'Bad Food' message

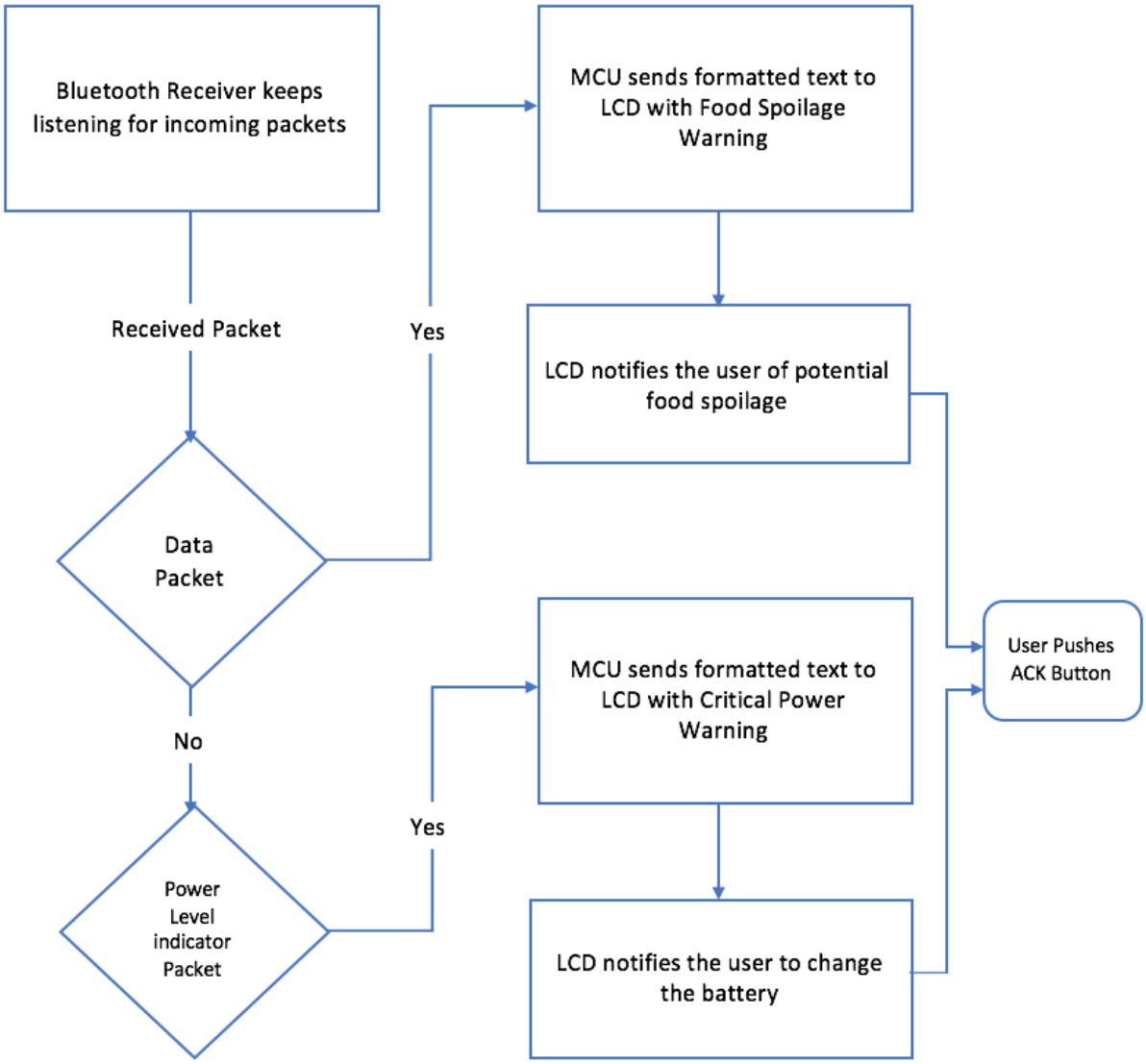
Appendix E – Software Flowchart

Our software handles communication between the different sensors and the microcontroller on one module. The data received from the sensors is processed by the MCU and appropriate data is sent through the bluetooth transmitter to the receiver on the display unit module. The data received is processed by the microcontroller on this module, and based on the type of data, relevant information is displayed on the LCD screen for the user to take necessary action. We will be implementing the UART, I2C communication protocols to be able to transmit and understand data between the MCU and other components. Also other than the data from the sensors, another type of data that is collected and processed is for power management. We have probes across the battery that measure the battery voltage from time to time. When the battery voltage falls below a certain threshold, we create a “Critical Battery Power” message packet and send it to the receiver to notify the user of the need to change the battery. This will function because the voltage required by the bluetooth module to function is under 6V as indicated by our requirements.

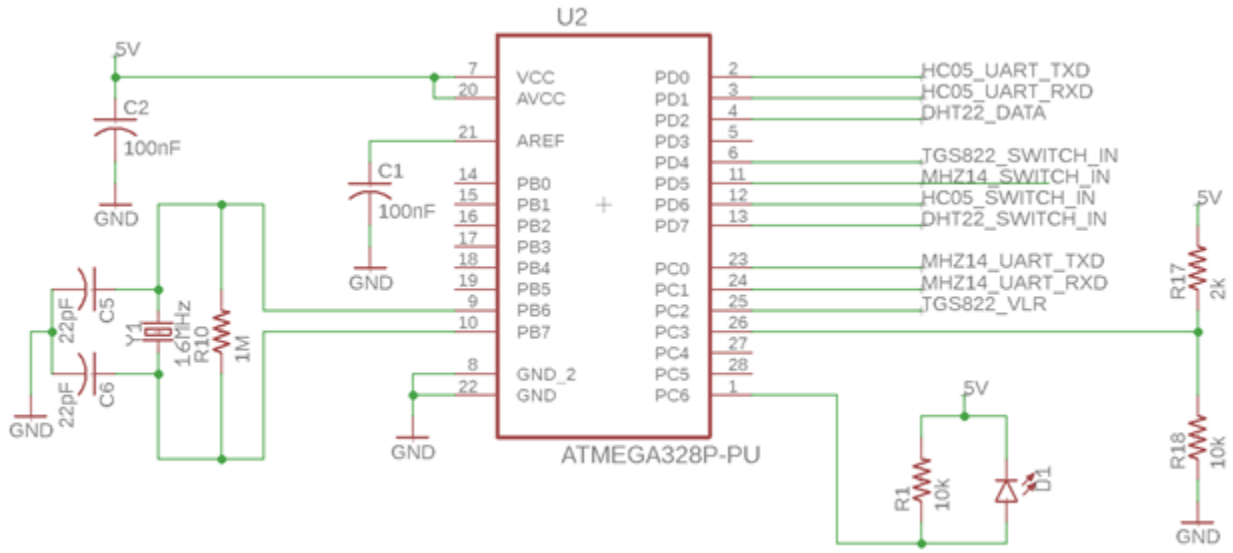
Sensing Unit:



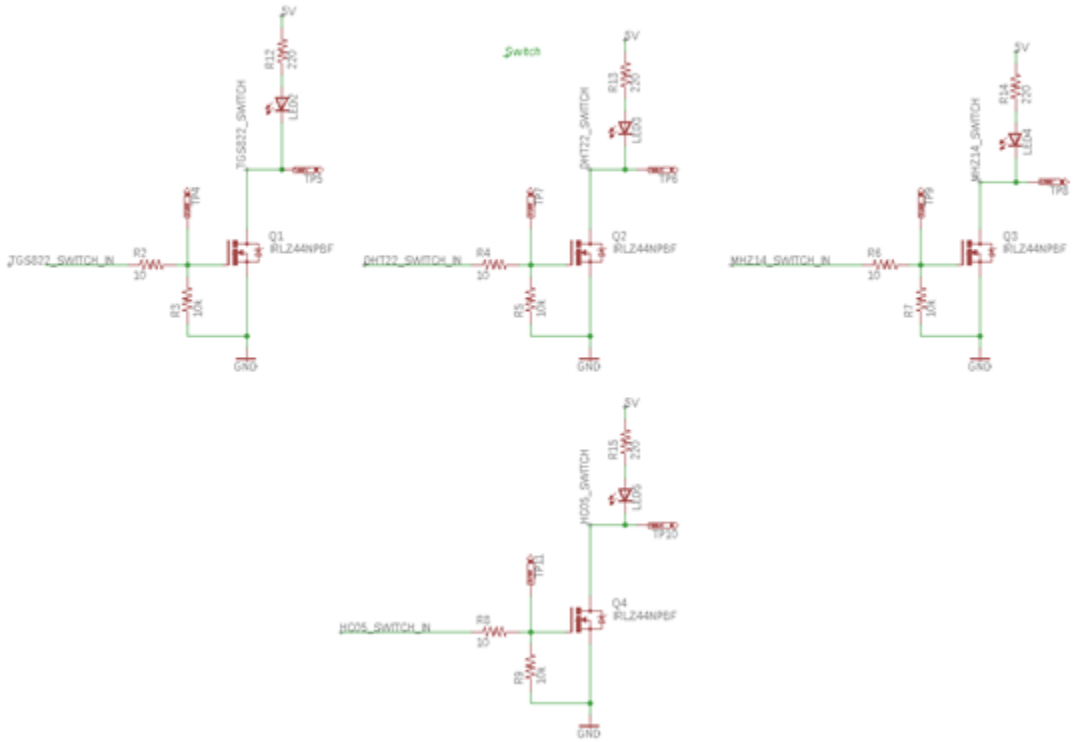
Display Unit:



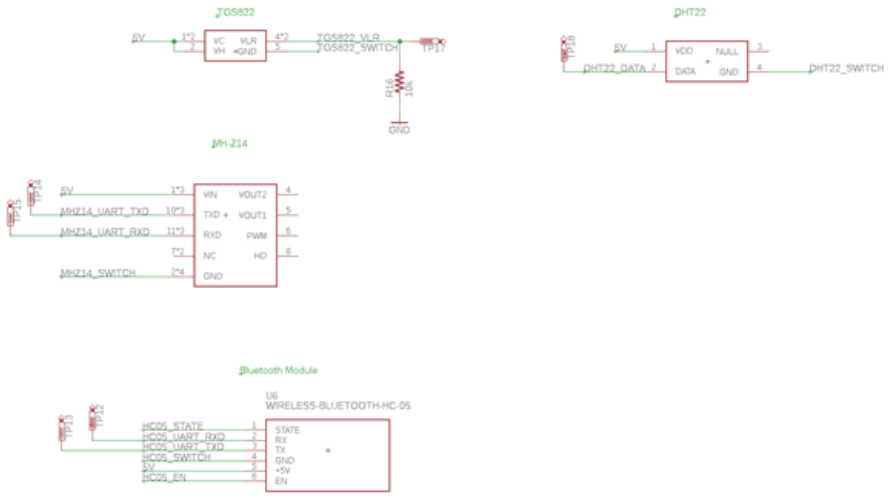
Appendix F – PCB Schematics and Layout



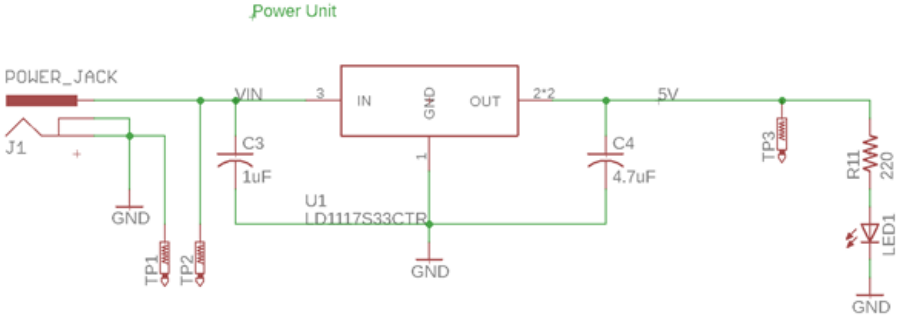
Sensing Unit: MCU



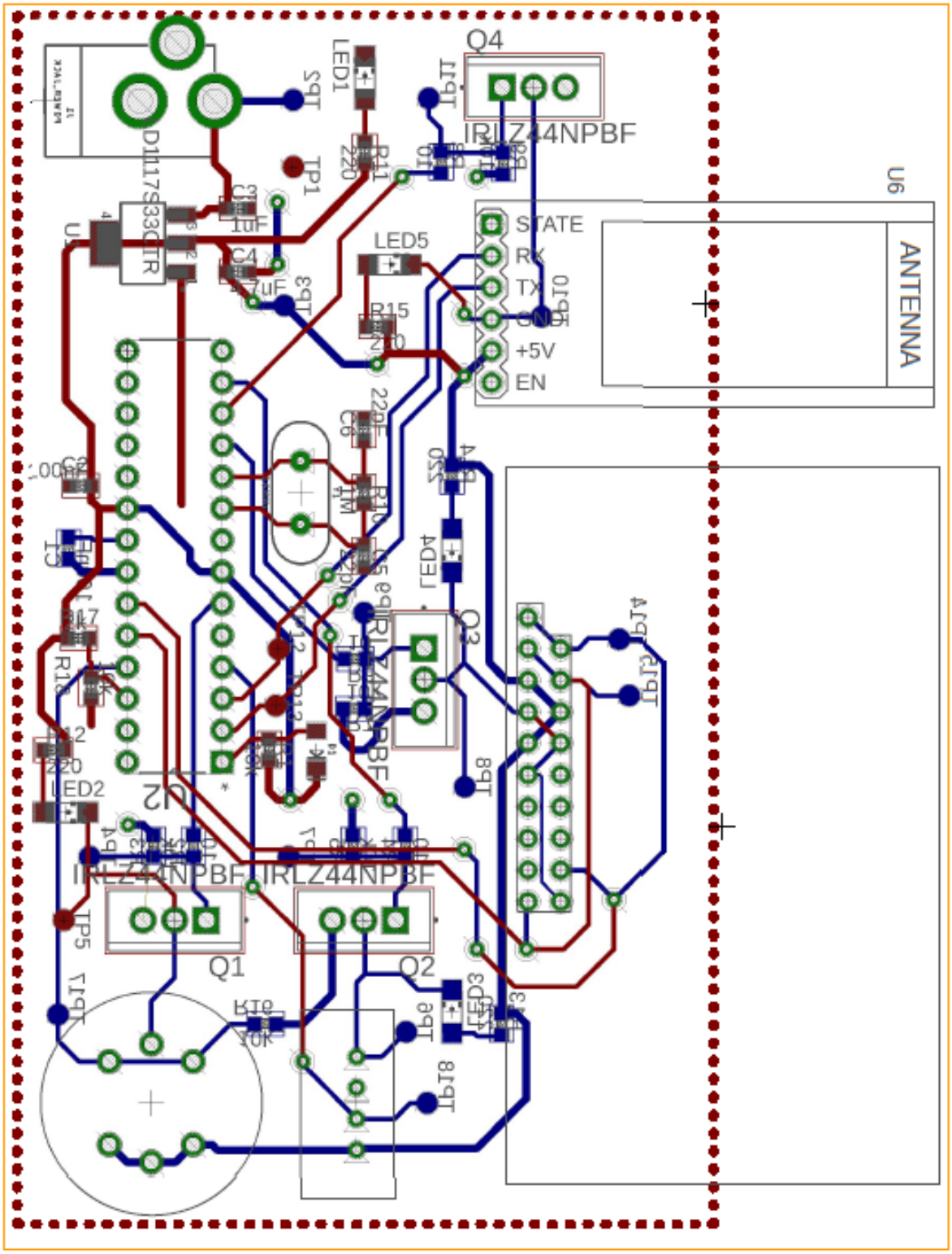
Sensing Unit: Digital Switches



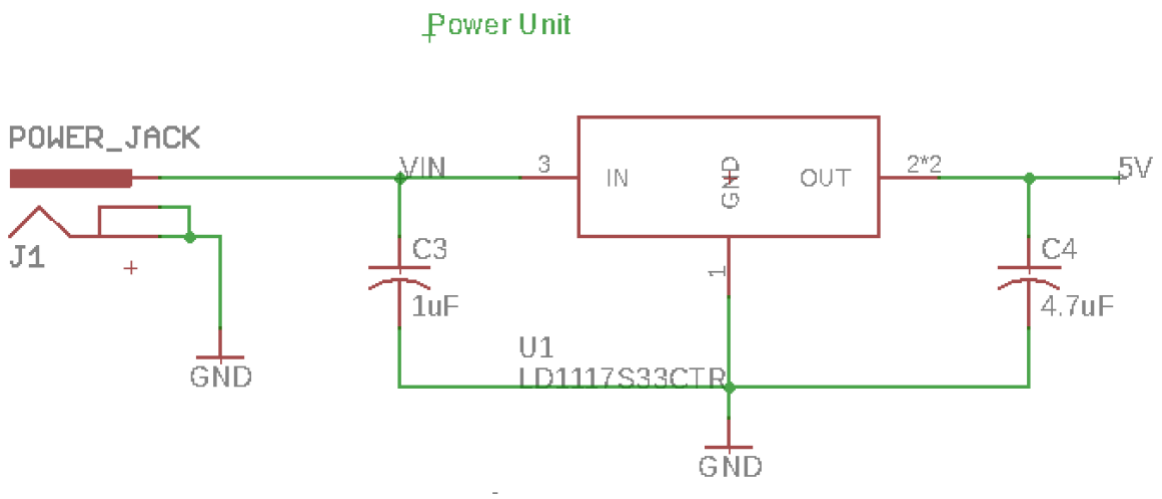
Sensing Unit: Sensors



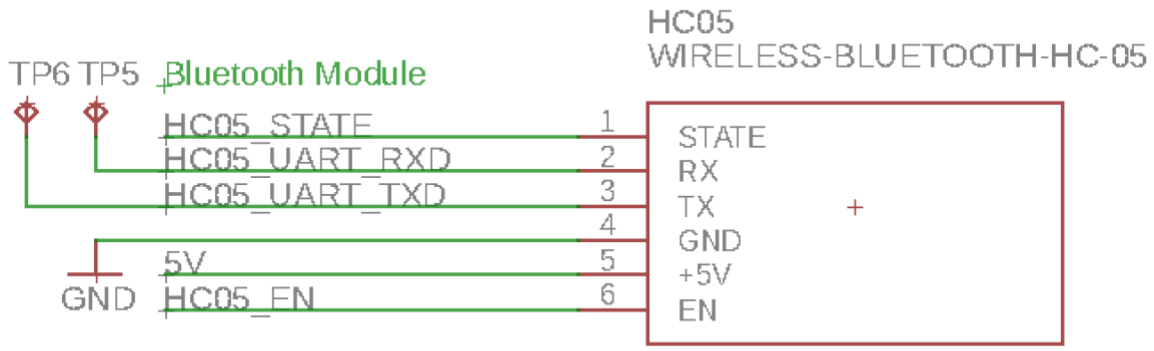
Sensing Unit: Power Unit



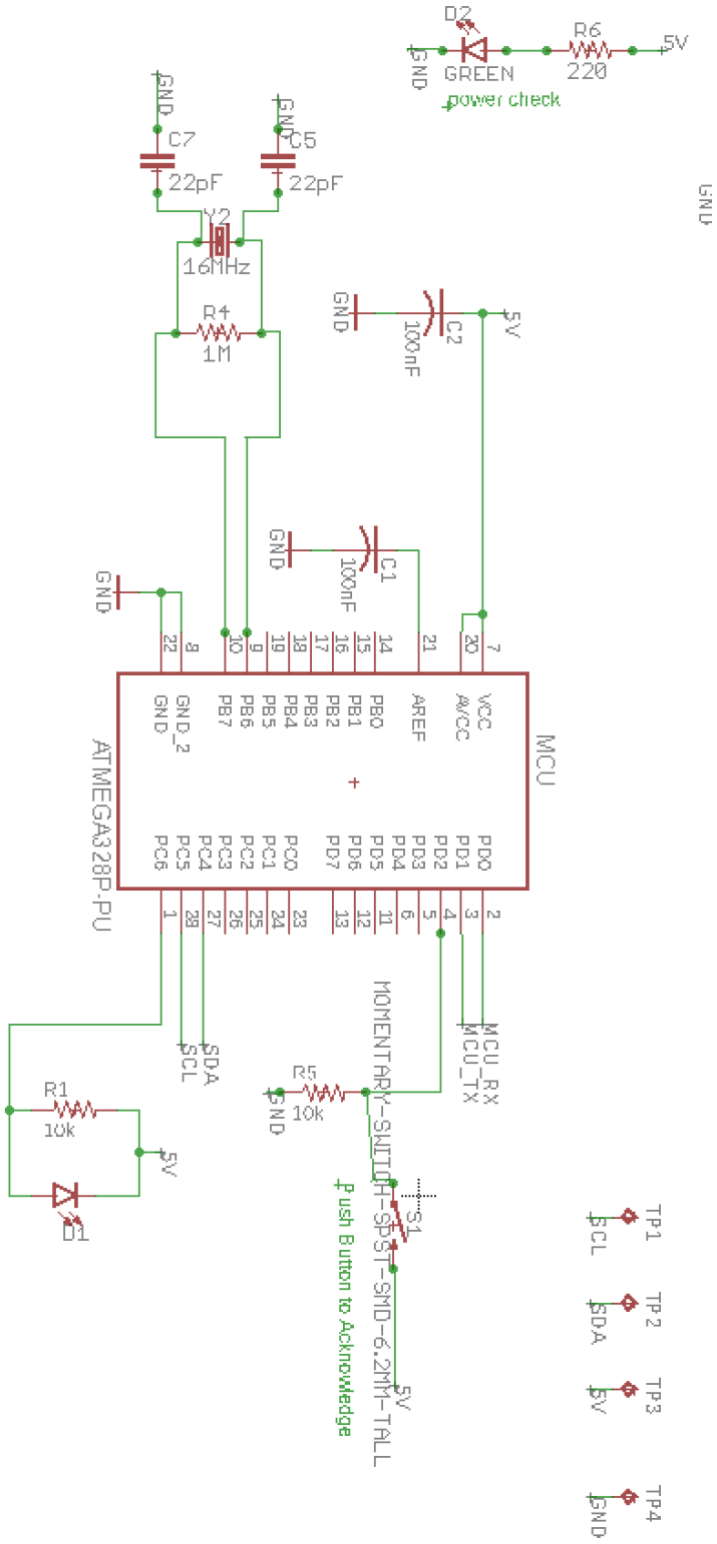
Sensing Unit: PCB Layout



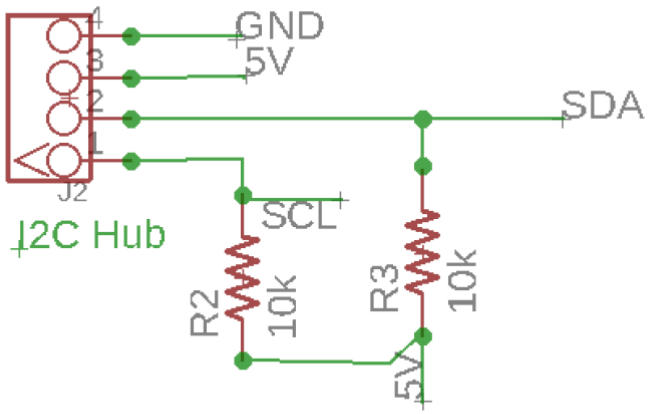
Display Unit : Power



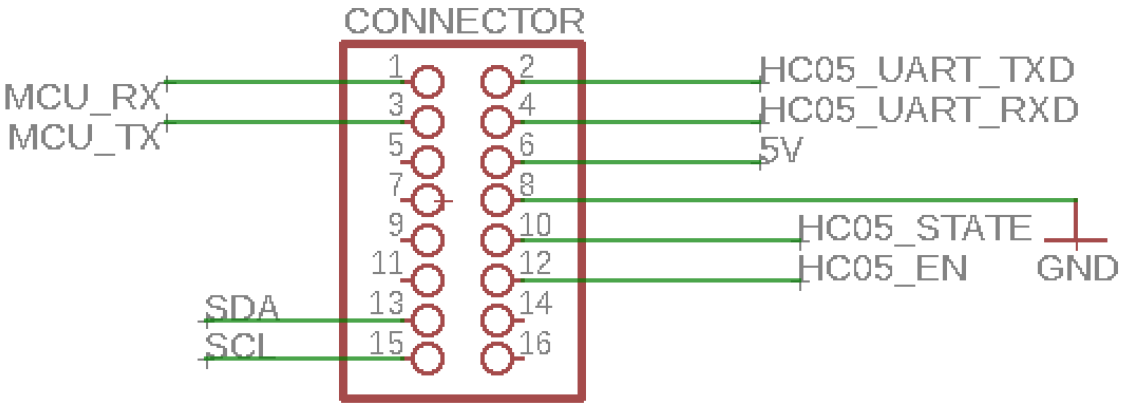
Display Unit : HC05



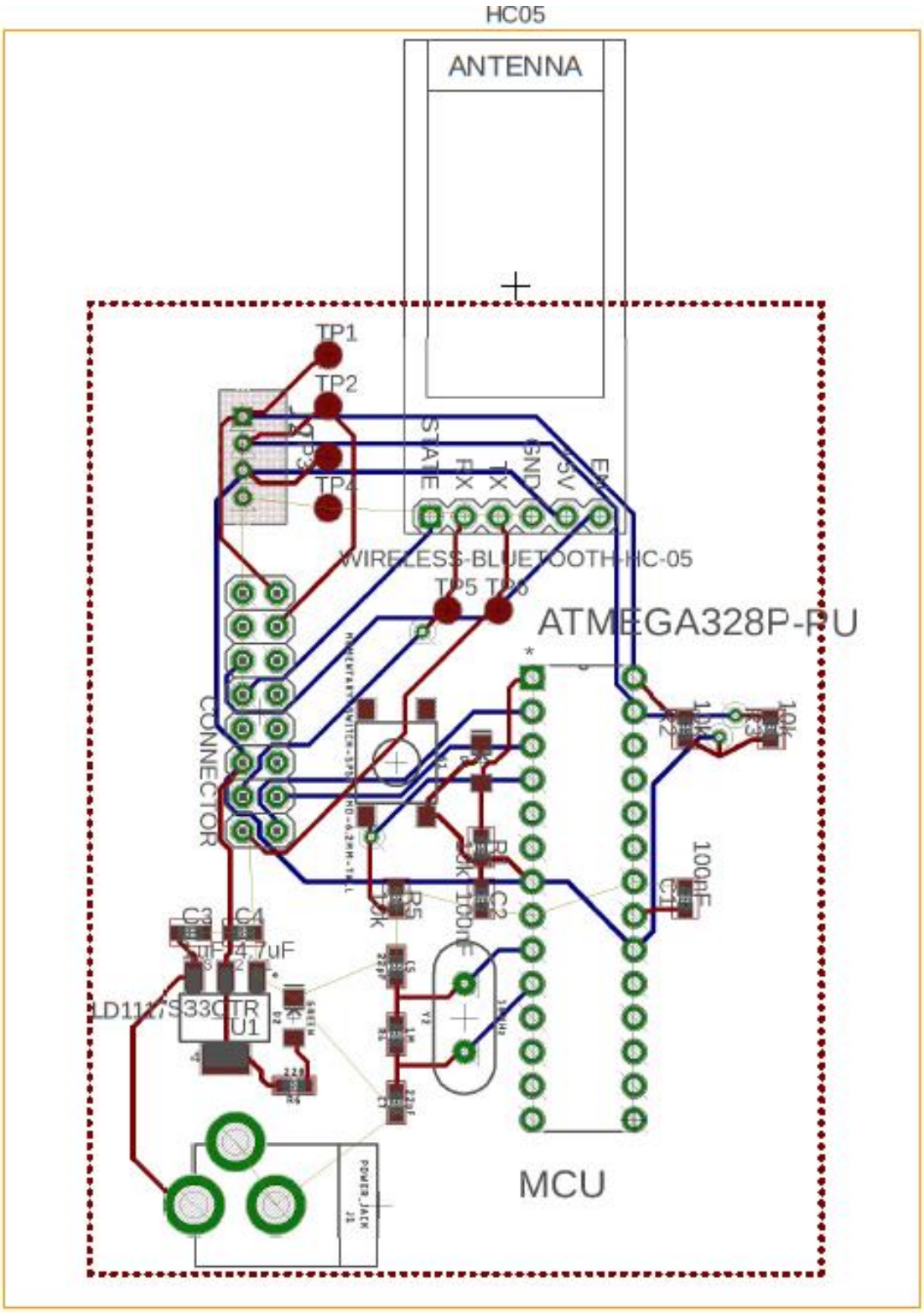
Display Unit : MCU



Display Unit : I2C



Display Unit : Connector



Display Unit: PCB Layout