GNG1103/2101

Design Project User Manual

[User Manual]

Submitted by:

[Team 10]

[Sandra Michael, 0300142873]

[Bassam Mahdi,0300041252]

[Ella Fossum, 300076269]

[Het Patel, 300143987]

December 4th 2019

University of Ottawa

Abstract

Our team was tasked with creating an automated system to be used in the University of Ottawa's new STEM building using the RossVideo product, Dashboard. The STEM building is host to many different design and entrepreneurial spaces, including MakerSpace. MakerSpace, as well as the other design spaces, are operated by CEED, the Centre for Entrepreneurship and Engineering Design. Based on a meeting and interview with the CEED staff, we determined that they need a system to indicate the availability of 3D printers in MakerSpace.

Table of Contents

List of Figures

List of Tables

I. Introduction

The automated system is composed of three fundamental subsystems: the sensor & sensor support system, a circuit & nodeMCU, as well as the user interface (Dashboard UI). Subsequent to establish explicit design criteria according to the customer's needs, we analyzed and evaluated various types of sensors currently on the market to develop a set of theoretical design concepts which have the potential to be used as our final product. At the very basic level, the user interface is able to clearly indicate whether an ultimaker 3D printer is in use, idle, or out of order. Our secondary objective is to use the data collected by the sensors to collect data to monitor the machine use over time and help CEED staff keep up with maintenance.

The figure 1 below shows the basic organization of user manual:

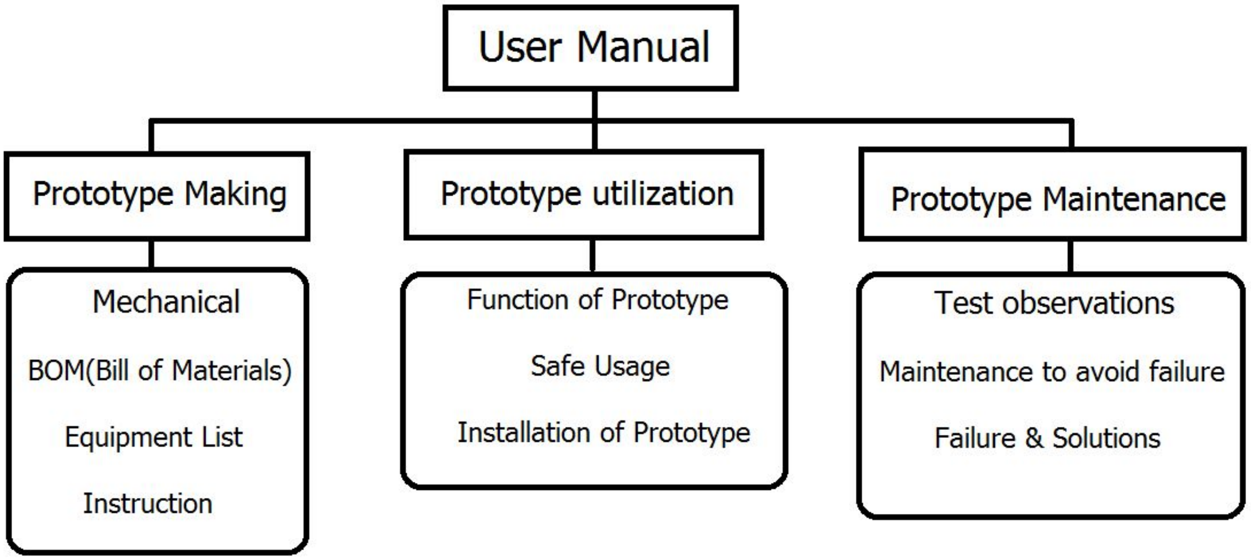


figure 1 organization of User Manual

A. Our Product -- An Overview

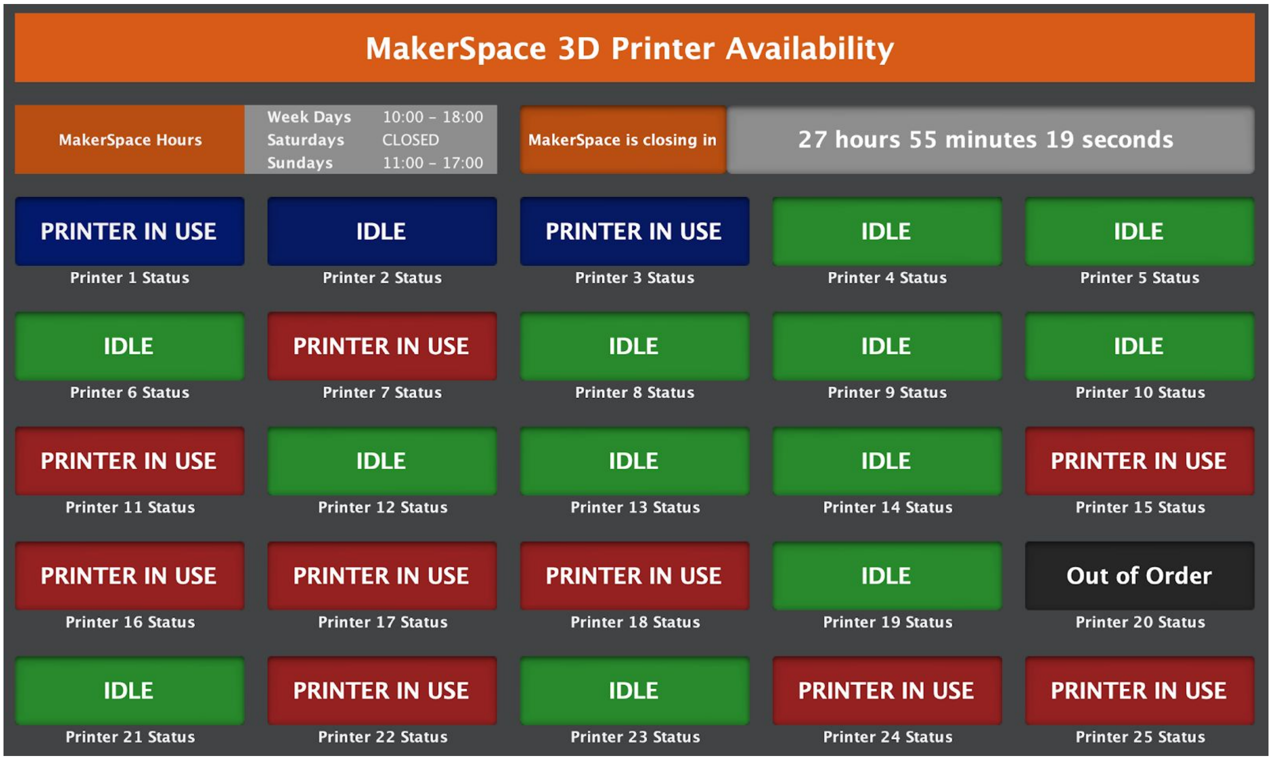
After the first client meeting, our team has established that the CEED staff members need an affordable product which could potentially optimize and monitor the machine use in MakerSpace as well as reduce confusion and ensure the completion of each project during the hours of operation of MakerSpace. These factors are important for the staff members since many students often have trouble identifying which UltiMaker 3D printer is available or do not take the time to calculate and compare the remaining operational hours of MakerSpace with the amount of time their printing will take.

The CEED staff is busy enough maintaining the daily operations of the MakerSpace and needed a system to filter the foot traffic through the space and decrease the amount of questions being asked of them. The students using MakerSpace also needed a system that would save them time and reduce the amount of needless trips to check on their current 3D printer job or check on the availability of the printers.

The main component of our product that makes us stand out the most from others is our sensor support structure. As a matter of fact, this subsystem is very easily removable without causing any damage to the printer and is cost-effective. The first part of the subsystem, Part A, is mounted onto the side of the printer and hence, does not require an additional sticking mechanism to attach it to the printer. The second part of the subsystem, Part B, has two holes drilled underneath the piece which are placed at an equivalent distance as the distance between both thumb screws on the arm of the printer. Since the holes are also of the same diameter as the thumb screws, they fit perfectly on the printer while still being easily removable. In order to construct our prototype, we essentially needed one nodeMCU unit (12$/unit), three magnetic reed switches (8.00$/sensor) and wires to connect the sensors to the circuit. Considering that each nodeMCU unit could support approximately eleven sensors and that there are twenty-five UltiMaker 3D printers in MakerSpace, the main expensive element which be required to manufacture our product is the magnetic reed switch for each product. On a bigger scale, the total cost of production is fairly inexpensive, if it were to be manufactured.

The main purpose of this product is to display on a screen the availability and condition of the UltiMaker 3D printers in MakerSpace. It consists of one pair of a magnetic reed switch which is placed on the top corner of the printer, more precisely where the arm of the printer rests whenever it is not operating. Once the printer is given a printing job, the nozzle automatically starts moving around the printing space. Considering one half of the sensor is attached to the arm of the printer and the other remains immobile in its initial location, the magnetic reed switches disengage and could not detect each other. Consequently, this input information is sent through the nodeMCU and then towards the user interface, DashBoard, which displays that the printer is no longer idle. After the printer is done printing, the arm nozzle returns to its initial position and the magnetic reed switches re-engage. Once again, this input information is sent to DashBoard through the nodeMCU, and the screen displays that the printer is currently idle.

The final Prototype



MakerSpace 3D Printer Availability

MakerSpace Hours

Week Days	10:00 – 18:00
Saturdays	CLOSED
Sundays	11:00 – 17:00

MakerSpace is closing in 27 hours 55 minutes 19 seconds

PRINTER IN USE Printer 1 Status	IDLE Printer 2 Status	PRINTER IN USE Printer 3 Status	IDLE Printer 4 Status	IDLE Printer 5 Status
IDLE Printer 6 Status	PRINTER IN USE Printer 7 Status	IDLE Printer 8 Status	IDLE Printer 9 Status	IDLE Printer 10 Status
PRINTER IN USE Printer 11 Status	IDLE Printer 12 Status	IDLE Printer 13 Status	IDLE Printer 14 Status	PRINTER IN USE Printer 15 Status
PRINTER IN USE Printer 16 Status	PRINTER IN USE Printer 17 Status	PRINTER IN USE Printer 18 Status	IDLE Printer 19 Status	Out of Order Printer 20 Status
IDLE Printer 21 Status	PRINTER IN USE Printer 22 Status	IDLE Printer 23 Status	PRINTER IN USE Printer 24 Status	PRINTER IN USE Printer 25 Status

Figure 2: Final prototype of our Dashboard UI

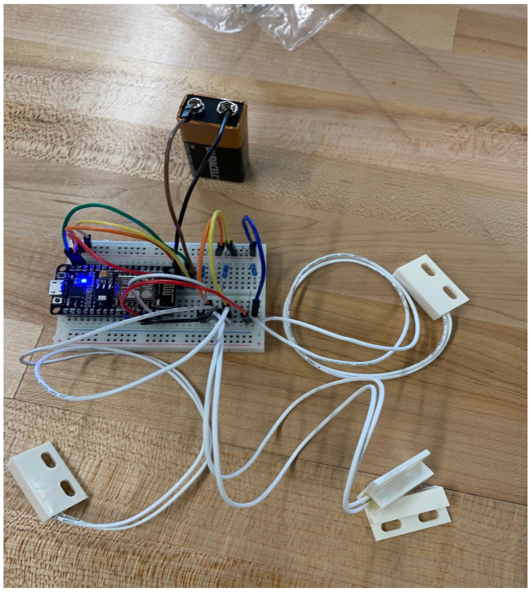


Figure 3: Prototype 3 including the three sensors, the nodeMCU, and an external battery.

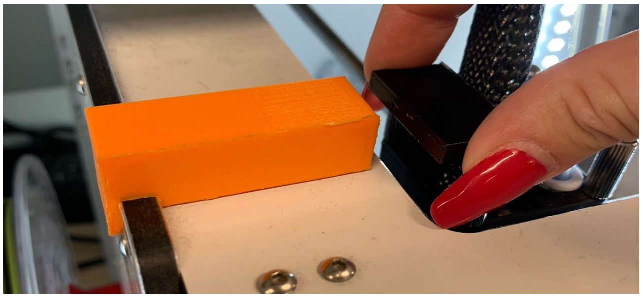


Figure 4 : Part A (in orange) and B (black) of the second prototype of the sensor support structure subsystem.

II. How the Prototype is Made

A. Mechanical: Sensor Support Structure

❖ Designing and building the first sensor support structure prototype

The first step of designing the sensor support structure was to take the proper measurements of the Ultimaker 3D printers in MakerSpace. Using a ruler, the length of the top left corner of the 3D printer was 5.2 cm and the measured distance between the two thumb screws, located on the arm of the printer, was 3.6 cm. Once the necessary measurements were taken, Part A and Part B of the support structure were designed on Solidworks, accordingly. The two components were 3D printed using the UltiMaker printers in MakerSpace and the printing process required approximately 1 hour to print. However, one major technical difficulty encountered was exporting the files from SolidWorks into Cura, the 3D printing slicing application. The files which we had saved our parts on were not in the proper format which consequently lead to complications when it was time to export them. They needed to be saved under the ".STL" file format. Another issue encountered while building the prototype, was the dimensions of Part A. Once both components were printed, they were placed on an UltiMaker printer to test them. The length of Part A, illustrated in *figure 5*, was too long and could have possibly interfered with the movement of the arm. Also, the width of the gap, which consists of a hook system, was not large enough to fit onto the side of the printer and hence, defeats its whole purpose. Lastly, the issue encountered with Part B of the structure was its material. Considering that this part was 3D printed, the plastic material consisting as the "ink" of the printer, lacks friction as well as the top surface of the thumb screws on the printer, where Part B would have typically been placed. Therefore, the element would slip off the thumb screws. The length of this part was also too short which could also have been another factor contributing to its lack of stability.

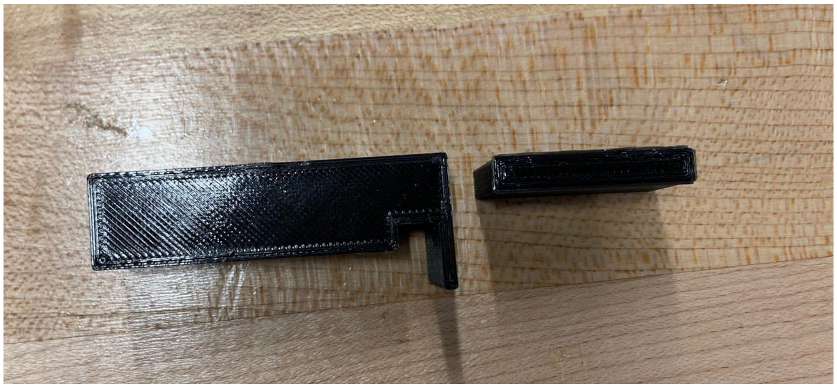


Figure 5 : First 3D printed prototype of the sensor support structure. The component located on the left of the image consists of "Part A" and the component located on the right consists of "Part B".

- **Designing and building the second sensor support structure prototype**

According to the observations and testing done on the first prototype, the second prototype was built. In general, the major difference between the two prototypes are the dimensions of Part A. The length of Part A was shortened to 4.7 cm, originally being 5.2 cm, and its gap was widened to 0.9 cm instead of 0.7 cm. As demonstrated in Figure 6 , the second prototype was not extruding from the side of the printer into the printing space and was hooked properly onto the side, without interfering with the screw placed behind the printer either. The issue encountered in this prototype was the method of attachment of Part B onto the thumb screws. The initial plan was to stick an elastic onto the part using hot glue, and then wrap the elastic around the thumb screws, which is illustrated in Figure 7 . The first issue encountered was sticking the elastic onto the part. The type of glue used was hot glue, which was available in MakerSpace. Unfortunately, the glue would peel off the structure component once it had cooled down and hence, would not stick the elastic onto it. The alternative method was to wrap another elastic in the middle of Part B which would hold it and the main elastic in place. However, since it has a short in length and lacked static friction, this method was unsuccessful. The piece would slip off on the side of the thumb screws when it was released from our hands.

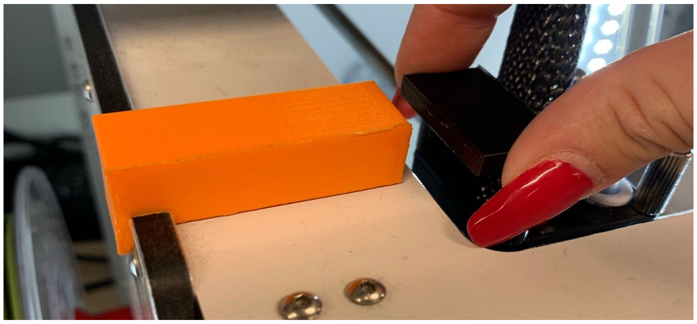


Figure 6 : The second prototype of the sensor support structure. The component located on the left of the image consists of the re-printed version of Part A and the component located on the right consists of the same Part B as in prototype 1.

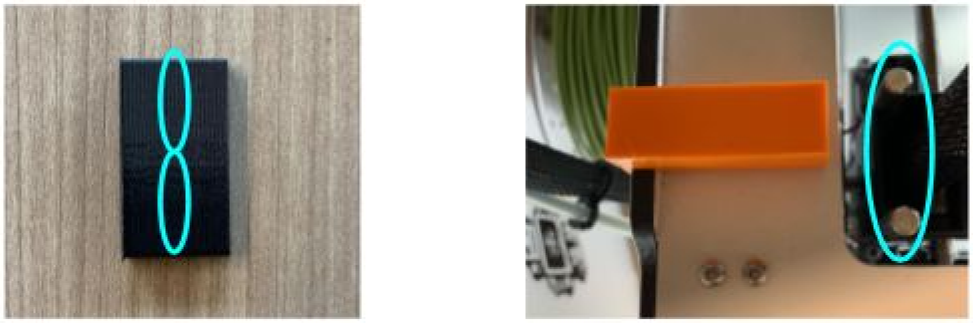


Figure 7 : Initial method of attachment of Part B onto the thumb screws.

❖ Designing and building the third sensor support structure prototype

The last prototype was once again designed based on the failures of second prototypes, in order to improve the effectiveness of the overall product. The first issue of the second prototype was the length and choice of material of its Part B component. Since our team was time constricted, we decided to select a recycled piece of wood and saw it into a more appropriate dimension. Its new length consisted of 4.2 cm, which was re-measured once again with a ruler. The second issue consisted of the method of attachment. Since our team was not given the permission to glue anything onto the 3D printers permanently, we decided to drill two holes ¾ of the way through the piece, which we could then placed on the thumb screws. The holes had a diameter of 0.6 cm and were placed 3.6 cm apart. These measurements were taken according to the diameter and the distance between both thumb screws of the 3D printers. Fortunately, once Part B was built and sanded, it fit immediately and did not cause any potential interference with its surroundings, as shown in *Figure 8* . The next step was to figure out how our team will stick the magnetic reed switches onto the prototypes. Since both components of the structure consist of different materials, we had to use different sticking methods. After considering our given options, we concluded that we should drill a small hole in Part A of a very minimal depth in order to screw the magnetic reed switch to Part A. A staff member of MakerSpace informed our team that if we were to directly screw both parts together, without drilling a small hole first, our piece might crack and hence, damage our prototype. Lastly, since Part B consisted of a different material and had a smaller width, we decided to use a double sided tape to stick the sensor onto the component. This method is less invasive and decreases the chances of permanent damage on the part.

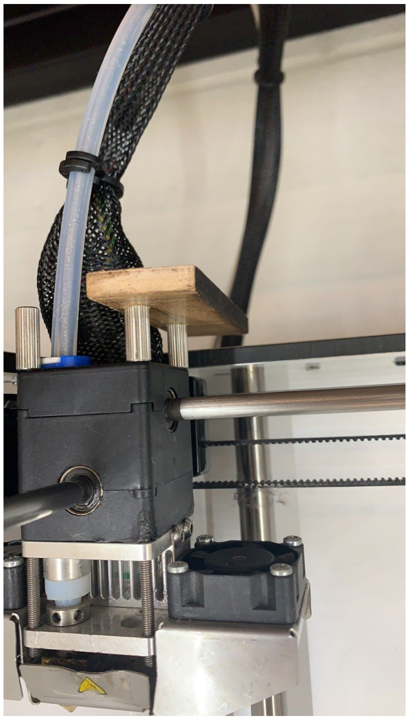


Figure 8 : Part B of the third prototype attachment method.

B. Electrical

❖ Designing the first prototype of the circuit.

The first step was to get all of the necessary components to construct a fully functional circuit. After benchmarking three different types of sensors, our team has decided to proceed designing our product using the magnetic reed switch. Once we had decided which sensor we are going to use, we ordered the sensor and the microcontroller (nodeMCU) from the Makerstore website. Also, we borrowed the other required components which consist of the Breadboard, 10k Ohm resistors and jumper wires from the lab. The next step was to determine where each component should be connected onto the nodeMCU. Accordingly, we had to figure out what each pin does in the nodeMCU in order to make the proper wire connection with the sensor, as demonstrated in Figure. Furthermore, we started to test the circuit by connecting the wires and the sensor to the microcontroller. Unfortunately, our team had not yet received the magnetic reed switch and had to use another type of sensor, a tact switch, as an alternative. Ultimately, as shown on *Figure9 and 10*, the first pin goes from one leg of the pushbutton through a pull-up resistor (here 10K Ohms) to the 5**v** supply. The second pin goes from the corresponding leg of the pushbutton to Ground (GND) pin. The third pin connects to a Digital I/O pin (here pin D0) which reads the button's state.

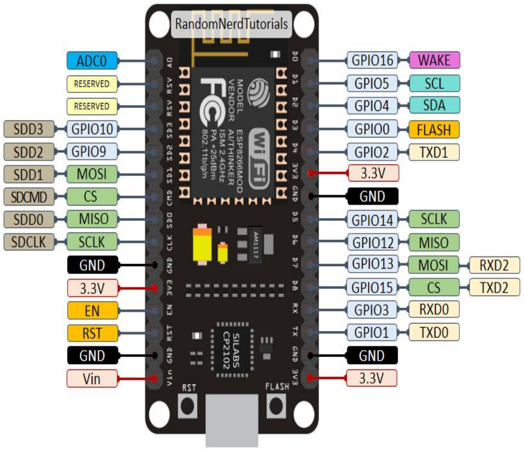


Figure 9: Descriptive figure indicating the identification of each pin.

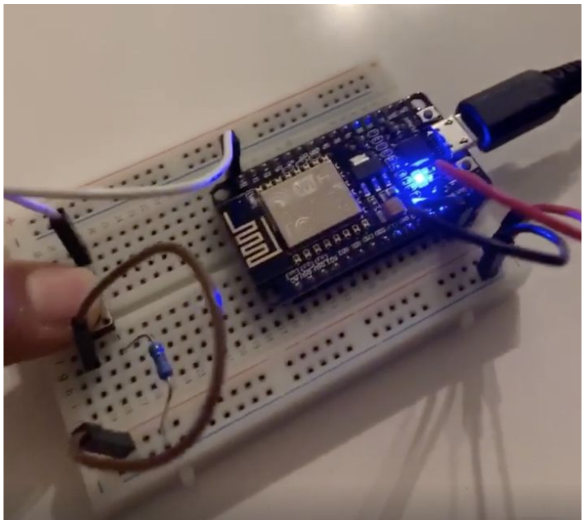


Figure 10 : When the pushbutton is open (unpressed) there is no connection and the opposite when the pushbutton is closed (pressed).

- **Designing the second prototype of the circuit.**

The major difference between the first and second prototype is the type of sensor used in the circuit. In fact the tact switch, which served as a temporary alternative to test our circuit, was replaced with a magnetic reed switch, as shown on Figure 11 . Moreover, we verified if our circuit was functional when the nodeMCU was connected to Dashboard. Also, our team had ordered two additional magnetic reed switch sensors in order to show that our product could handle more than one input simultaneously. Lastly, the magnetic reed switch works the same as the tact switch sensor. When the two parts of the sensors are attached there is a connection and there is no connection when they are not attached.

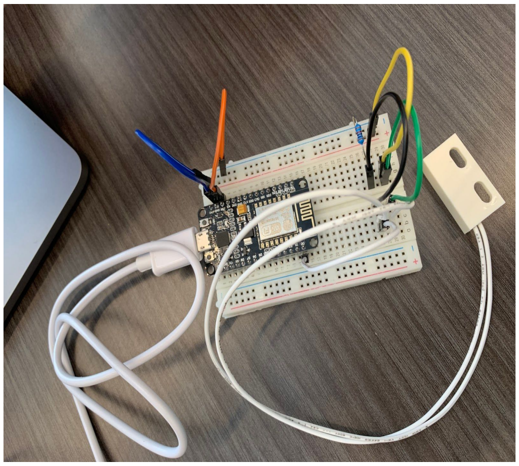


Figure 11: The second circuit prototype with a magnetic reed switch sensor as a replacement of the initial tact switch.

- **Designing the third prototype of the circuit.**

For the last prototype, we decided to power the microcontroller by using an external battery (9V battery), which is illustrated in Figure 12. We chose to use an external battery to power the nodeMCU because it would be much easier for us to place the nodeMCU on the wall of the printer. Also, we could show that our microcontroller can be connected through Wi-Fi. Moreover, we connected the positive side of the battery with the Vin pin (VIN), and the negative side with the ground pin (GND). In addition, once we received the two other magnetic reed switch sensors, we added them into our circuit to verify that our system could support more than one input at a time, as shown on Figure 12 .For the other two sensors, the first pin goes from one of the wires of the magnetic reed switch to the Vin pin (VIN). The second pin goes from the other wire of the sensor through a pull-up resistor (10K) to the ground pin (GND). The third pin connects to a digital pin(D1) which reads the sensor’s state. Similarly, the last pin connects to a digital pin(D2). Furthermore, when the sensors are attached, it will indicate on the dashboard that the printer is idle. On the other hand, when the sensors are disengaged, the printer will be in use. Lastly, after we tested the circuit on a breadboard, we placed all the components into a perf board and then we soldered them.

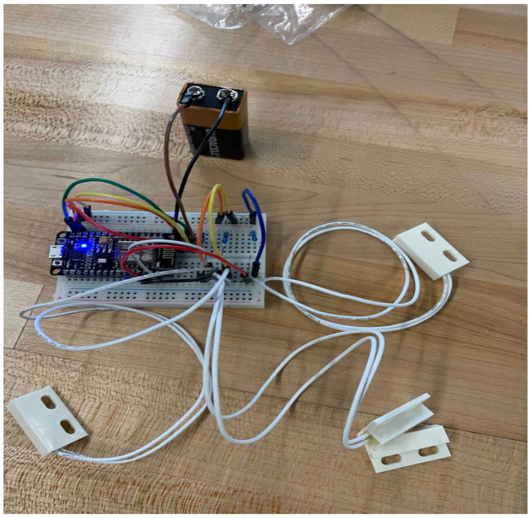
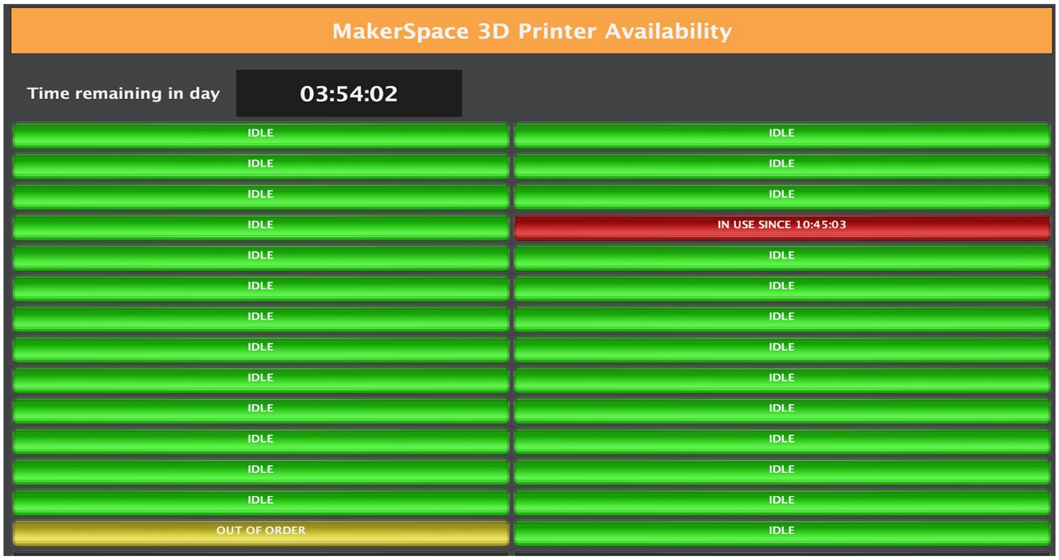


Figure 12: The third circuit prototype with three function sensors and an external battery.

C. Software

❖ Creating the first Prototype



MakerSpace 3D Printer Availability

Time remaining in day 03:54:02

IDLE	IDLE
IDLE	IDLE
IDLE	IDLE
IDLE	IN USE SINCE 10:45:03
IDLE	IDLE
IDLE	IDLE
IDLE	IDLE
IDLE	IDLE
IDLE	IDLE
IDLE	IDLE
IDLE	IDLE
IDLE	IDLE
IDLE	IDLE
OUT OF ORDER	IDLE

Figure 13 **:** An initial concept for our user interface

For our initial concept, we wanted coloured labels that would display the current status of the printers with a timestamp next to each of the "in use" printers indicating how long it had been active. The display was also to have a countdown timer which displays the time remaining until the MakerSpace closed for the day. We also wanted a feature to indicate which printers were out of order. We soon discovered that this last feature was out of the scope of our project and decided to cut it in order to focus on the more critical aspects of the UI design and back end coding.

The first prototype consisted of an initial concept of our user Dashboard custom panel interface, pictured in Figure 13, as well as a separate custom panel (not pictured) which connected to the nodeMCU. This separate panel, call it panel X, was created by following the "*Basic+Dashboard+Tutorial+with+NodeMCU+(Arduino).pdf*" on brightspace. This was an important step because it familiarized us with the ins and outs of networks, microcontrollers, ports, and IP addresses. All of these were new concepts for our group members, so this first prototype was mainly just a learning tool for our group to be able to use and apply to future prototypes. In addition, the visual display was designed based on our own ideas without any input from users. This was important for us to do so that we could provide a basic proof of concept while also getting used to using Dashboard and understanding its basic capabilities and limitations.

Creating the second Prototype

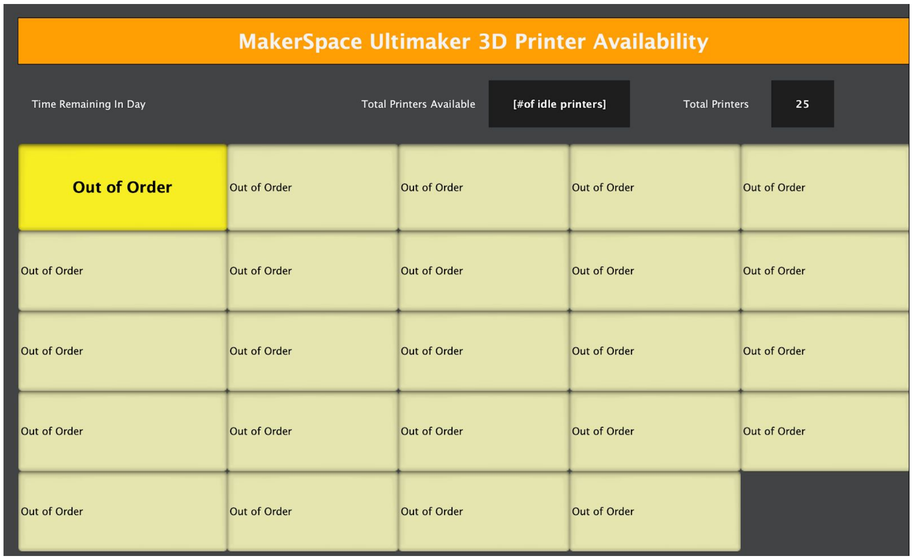


Figure 14 **:** An updated concept with one functional button

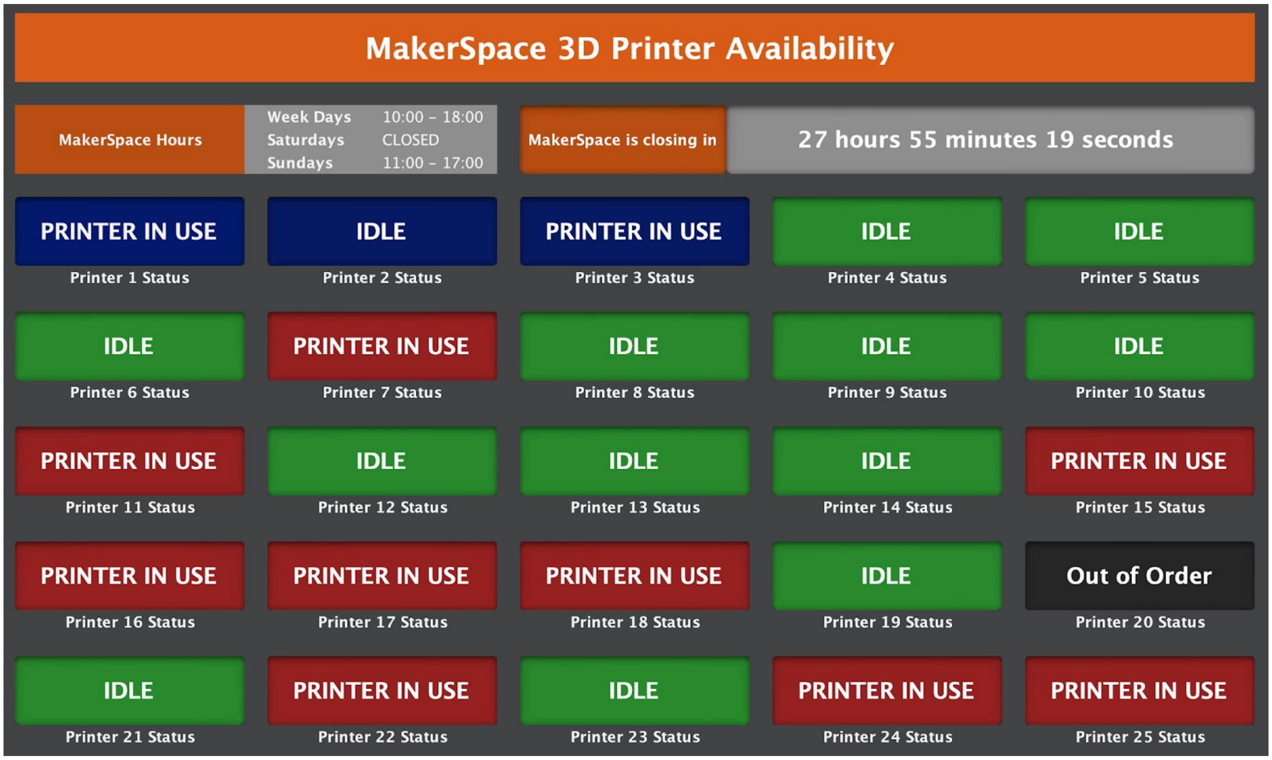
Our second prototype focused more on the functional side of the panel. It consisted of one functional buttons and a bunch of place holders that future printers would occupy. Prototype two, pictured in Figure 14 provided the shell of a new concept that we built off of in later prototypes.

For this prototype, we followed the "*Listener Tutorial with NodeMCU (Arduino).pdf*" file to build off of our first back end prototype and integrated it with the visual component to create a label had changing text based on the input. This was done by creating a program in arduino that read the information from one pin on the node MCU, formatted the information into a 0 or a 1 and sent it to Dashboard.

Dashboard then took this digit and evaluated it using a switch statement, and then used the data to update a "read-only" label that we had created on our panel. When the dashboard received the information, it set the colour and text to the appropriate values based on the value. The colour command was not working, but we discovered our error in time to submit our third prototype. At this point, only the text of one button changed.

At this point, we gathered feedback from potential users and used that to inform the visuals of our next prototype. The feedback that we got was to add some colour, space things out a bit, and increase the font size for an easier reading experience. It was also pointed out to us that our printers did not have any labels so it was hard to tell which information box was referring to which printer in the MakerSpace.

Creating the third Prototype



MakerSpace 3D Printer Availability

MakerSpace Hours	Week Days	10:00 – 18:00
	Saturdays	CLOSED
	Sundays	11:00 – 17:00

MakerSpace is closing in 27 hours 55 minutes 19 seconds

PRINTER IN USE Printer 1 Status	IDLE Printer 2 Status	PRINTER IN USE Printer 3 Status	IDLE Printer 4 Status	IDLE Printer 5 Status
IDLE Printer 6 Status	PRINTER IN USE Printer 7 Status	IDLE Printer 8 Status	IDLE Printer 9 Status	IDLE Printer 10 Status
PRINTER IN USE Printer 11 Status	IDLE Printer 12 Status	IDLE Printer 13 Status	IDLE Printer 14 Status	PRINTER IN USE Printer 15 Status
PRINTER IN USE Printer 16 Status	PRINTER IN USE Printer 17 Status	PRINTER IN USE Printer 18 Status	IDLE Printer 19 Status	Out of Order Printer 20 Status
IDLE Printer 21 Status	PRINTER IN USE Printer 22 Status	IDLE Printer 23 Status	PRINTER IN USE Printer 24 Status	PRINTER IN USE Printer 25 Status

Figure 15 **:** Final product with three functional buttons and functional timer

The third prototype required lots of work on the visual side customizing the panel and testing various shades of green and red. As it stands, our third prototype has three functional buttons (shown in *Figure 15* in blue, printers 1, 2 and 3) which are able to display “IDLE” or “PRINTER IN USE”. In the figure, they are in blue because that is the default colour when there is no WIFI connection. The panel also has a countdown timer and a bunch of sample printer displays (Printers 4-25). We were unable to figure out a way to determine and display a printer which was “Out of Order”, but we kept a sample display to show that we intended for this to be a future feature added to our product.

The back end of the coding involved further formatting of the data being sent from arduino, reading from two additional pins, formatting two additional labels, and finally solving the colour changing issue. These first tasks were accomplished through a bit of trial and error, with a bit of brute force as well. All of the results and instructions are included in the comments made in the code. The biggest and most difficult new development was the introduction of the functional countdown timer.

The countdown timer consists of two labels: The text label and the time label. Both of these labels change based on the time of day. The timer determines the hours of operation for the MakerSpace given the day of the week, and then displays a countdown to either the end of the day or to the time of opening. The timer works as following. The program gets the current day of the week and references a predefined array to access the MakerSpace hours of operation for that given day. It now has information about the opening time and closing time of the MakerSpace. After this, the program determines the current time of day. It then takes this value and compares it to the MakerSpace hours of operation to determine if the MakerSpace is currently “not yet open”, “open”, or “closed for the day”. Depending on this, the text label displays “MakerSpace is Opening in”, “MakerSpace is closing in”, or “MakerSpace is Closed for the day”. The time label determines the difference between the current time and the next opening or closing hours, and displays a time showing the hh/mm/ss until the MakerSpace next closes or opens (whichever is appropriate for the current time of day). This code runs once a second and updates the label with the new time each second. This means that it is not really a countdown timer, but a code that displays a new value every second to simulate the appearance and function of a countdown timer.

One of the biggest issues in this part of the program was using getTime functions in JavaScript, because we had to code around timezones since JavaScript would get us the time in GMT time when were in GMT-5. This not only gave us errors with the countdown timer displaying the wrong time, but would often give us the wrong day of the week if we were programming late at night. We ultimately got it to work, but the code is not very elegant and needs to be tested to work through a few logic bugs in the code.

1. BOM (Bill of Materials)

Item number	Part name	Description	Quantity	Unit cost	Total cost
1.	Magnetic reed switch	an electrical switch that switches when a magnetic field affects it	3	$8.00	$24.00
2.	nodeMCU	Microcontroller featuring Wifi chip	1	$11.99	$11.99
3.	Dashboard	an open platform that enables users to use custom panels that makes complex operations simple.	1	Free	Free
4.	Breadboard	solderless device for temporary prototype with electronics and test circuit designs.	1	Free (borrowed from MakerSpace)	Free

5.	Wires	Electrical wire with pin at each end	3		
total cost					$35.99

Table 1: Cost Approximation of our product's production.List all the parts and materials in this category (if available, include links to each item).

2. Equipment list

Screw	Screw driver
3D printer	Ruler
Magnetic reed sensor	Saw
Drill press	Wood
Drill bit (0.6 cm diameter)	2 sided tape

Table 2: Items required to build the magnetic reed sensor support structure. All of the items listed above are available in MakerSpace, on campus.

Solder iron	Soldering
Jumper wires	Breadboard
Battery (9V)	nodeMCU
microcontroller	

Table 3: Items required to build the circuit of our product.

microUSB cable	nodeMCU
Computer	Dashboard by RossVideo (software)
Arduino (software)	ESP8266WiFi library (library download)

ESP8266WiFiMulti (library download)	

Table 3: Items and software required to build the User interface and program the nodeMCU

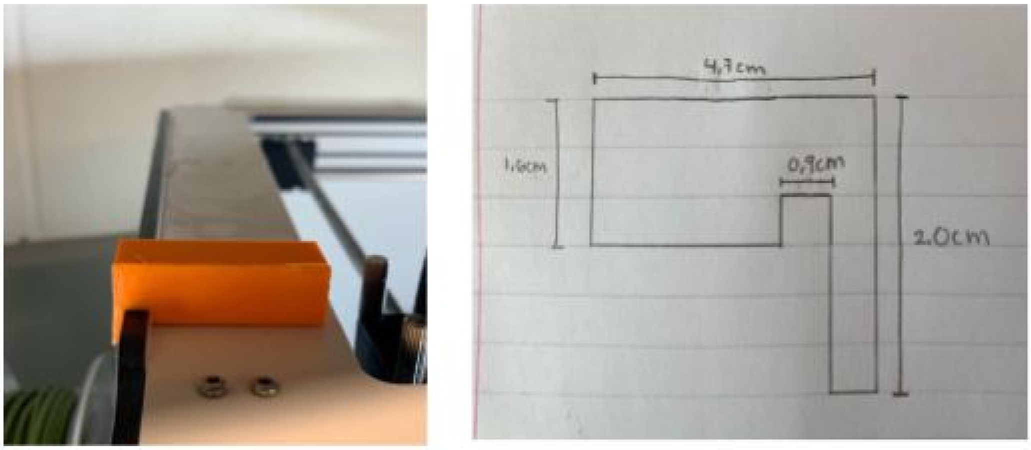
3. Instructions

Explain step by step instructions on how to build this specific part. Include as many pictures and diagrams for clear understanding of the process. Make sure to attach all files you are referencing.

1. Sensor Support Structure

1.1. Part A:

- Use SolidWorks to design and build Part A of the sensor support structure, illustrated on *Figure 16*.



- *Figure 16:* The third prototype of the sensor support structure on the left and an oblique sketch with its dimensions on the right.

-

- Once Part A is built on SolidWorks, save the file under the type ".STL".
- Upload the STL file into Cura and convert it to a "g-code" file.
- Save the print file on the SD card.
- Insert the SD card into the printer and start printing the part.

1.2. Part B:

- On a piece of wood with a width of 0.5cm, draw the dimensions of Part B (length: 4.2 cm, height: 1.6 cm) using a ruler.
- Draw a faint line in the center of the piece (2.1 cm away from the left or right edge of the structure). Then, draw a dot 1.8 cm to the right of the line drawn previously and a dot 1.8 cm to the left of the line.
- Using a Drill press and a drill bit (0.6cm diameter), drill two holes ¾ through the piece where the two dots were drawn.
- Once the holes are done, use a saw to cut Part B at the construction lines drawn at step I.
- Use sander paper to sand the external sides of the piece.
- Stick the sensor without wires onto Part B using a double sided tape of length shorter than the piece (its length should not matter if it's less than the length of Part B).

2. Circuit

- Get all the required materials, which are 10K resistors,microcontroller(nodeMCU), jumper wires, an external battery, and magnetic reed switch sensor.
- place the microcontroller on the breadboard, as shown on Figure17.
- place the sensor on the breadboard, as shown on figure 18 .
- connect the vin pin with the positive side of the breadboard and connect the ground pin with the negative side, as demonstrated in figure19.
- Place one leg of the 10K resistor with the negative side of the breadboard, and the other leg with the sensor, as shown on figure20.

- connects the positive side of the breadboard with the sensor, as shown on figure20.
- connects a wire with the pin (D1) and with the sensor but make sure that the wire is connected vertically with the 10k resistor, as shown on figure21 .
- add the other two sensors similarly as the first one.
- connect the other two sensors, one sensor with pin (D2) and the other one with pin (D3). Also, do not forget to place the wires vertically with the 10k resistor.

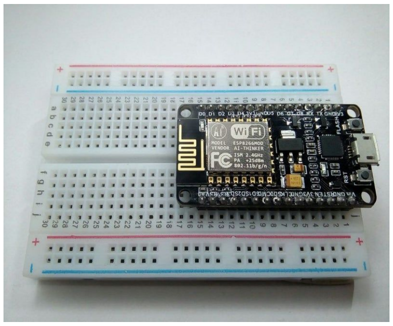


Figure17

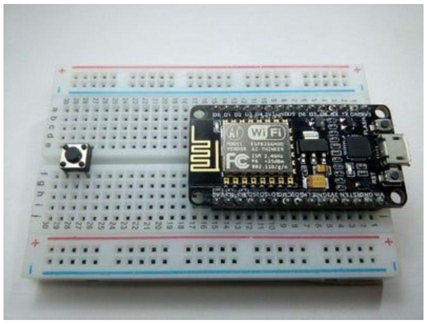


Figure 18

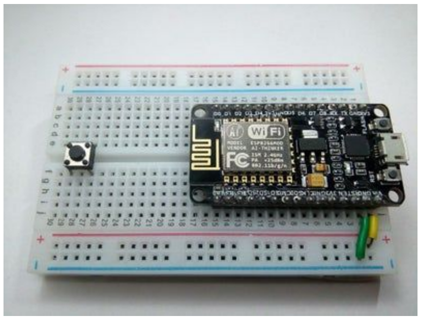


Figure 19

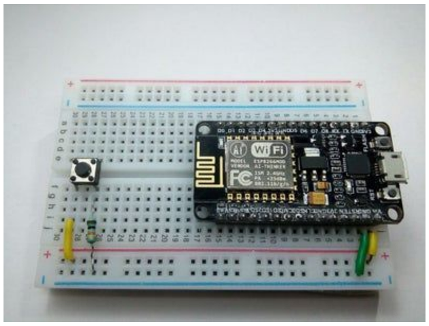


Figure 20

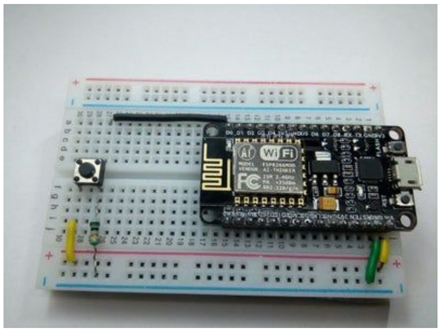


Figure 21

3. Arduino

3.1. Arduino code

- Download Arduino software, install ESP8266WiFi and ESP8266WiFiMulti libraries

- Create function to read from pins where the nodeMCU is connected
- Create function to send pin state to Dashboard in an array

4. User Interface

4.1. Back end

- Open Dashboard and create a custom panel
- Follow the tutorial outlined in "*Basic+Dashboard+Tutorial+with+NodeMCU+(Arduino).pdf*" to connect Dashboard to your nodeMCU
- Follow the tutorial outlined in "*Listener Tutorial with NodeMCU (Arduino).pdf*" to create a listener server
- Customize panel to fit your needs and display proper information from the nodeMCU
- Create a timer function in the listener task to compute the time and calculate time until end of day

4.2. Front end

- Using "read-only" labels and parameters, create interface to display the data from the nodeMCU
- Create visually pleasing labels and information boxes
- Using "read-only" labels and parameters, create a label that displays the time until MakerSpace closes or opens.

III. How to Use the Prototype

To use the prototype, first ensure you have arduino and Dashboard installed. Now download and open team10DashboardPanel.grid and projectNov8.ino. Edit the arduino code and replace “#define STASSID "iPhone"; #define STAPSK "12345678";” with your own wifi network and password. Check your computer’s IP address and update the arduino code. Connect nodeMCU to your computer through the micro USB cable and reupload the arduino file.

The prototype has two main functions: to display the current status of the printers and to display the time until the MakerSpace closes. The first function works by reading a pin on the nodeMCU and transmitting that information to Dashboard to be displayed in a readable way. The pin is connected to a sensor which gathers information from the Ultimaker 3D printer it is connected to. The second timer function is a little more complicated. For this part, the Dashboard function finds the current day of the week and determines the MakerSpace hours for this day. From here, it finds the current time of day and determines if the MakerSpace is (a) not open yet, (b) open, or (c) closed for the day. From here, it computes either the time until the MakerSpace opens, time until close, or time until the MakerSpace is next open, whichever applies.

IV. How to Maintain the Prototype

Explain the tests that were done on the prototype for validation of the final design. Present all of the applicable results that you obtained (i.e. data collected; performance graphs, etc.). List any issues or special requirements for sustained usage.

Describe regular maintenance that should be performed on the prototype to avoid failure.

Explain what parts may be prone to break and need to be replaced.

To keep our prototype efficient, we should maintain our prototype regularly. In order to avoid failure in the future, we should test it and see what could go wrong or might not work for a long time. For example, we have tested our prototype many times, however, in our design day we encountered a problem which was we ran out of battery. what we could have done in terms of powering the nodeMCU is to use a wall wart to power the nodeMCU instead of a rechargeable battery.

Considering that the sensor support structure consists of two components, Part A and Part B, which should remain fixed or attached to their appropriate section on the printer at all times, there should not be any maintenance required. However, since Part B is made out of a thin piece of wood, the holes made for the thumb screws may slightly become damaged if they are constantly being placed and removed from the arm of the printer. Due to time constraint, our team has failed to perform an iterative test on the prototype. We cannot conclude that Part B will remain in excellent condition if it’s constantly being moved around. Moreover, since Part A was 3D printed and its prolonged part, which is mounted onto the printer and which serves as a hook system, is relatively thin compared to the rest of the piece. This part may be prone to break if a force is exerted on it. In that case, a new piece may need to be 3D printed.

V. Conclusions et Recommendations for Future Work

The Dashboard and arduino code need to be updated to be able to process larger amounts of data more efficiently. As it currently stands, they can effectively receive and transmit accurate data, but the format of the code would be cumbersome to write for any large number of printers. For the scope of this project, it was only necessary to transmit information from three sensors, but as we add sensors and nodeMCUs this becomes difficult and clunky to code. As more printers are added, this code need to be refined. We recommend creating a function to read the pin state, one to format the data, and one to send it to dashboard. As it stands, these three tasks are being done manually by each sensor over and over which makes for an inefficient code. In the back end of the user interface, similar functions should be introduced to process data and format the display. The code currently is formatted as a nested set of switch functions which is, once again, feasible for the current prototype but will quickly become unmanageable and difficult to edit.

More testing needs to be done for exception cases to work out any hidden bugs in the programming or code. Default functions and options should be added to prevent malfunction, bugging, or error. For example, test the project during different times of the day to see if the timer is working.

The front end of the User Interface should be tested on more potential users. We recommend further studying effective user interfaces to continue to update and refine this aspect of the product. Seeing as this is the only part of the product the student user interacts with, it is important for it to look professional and communicate information effectively.

The material used to construct Part A of the sensor support structure may not be convenient in the long-run since it may easily be damaged if used constantly. If this product were to be used for future work, another alternative material should be considered for this specific piece of the sensor support structure in order to enhance its performance.

VI. Bibliography

MakerStore. “Electronics, Materials, and Merch.” *MakerStore*, makerstore.ca/shop?olsPage=t%2Felectronics&page=2.

Instructables. (2017, July 26). Control LED Using PushButton With NodeMCU. Retrieved from https://www.instructables.com/id/Control-LED-Using-PushButton-With-NodeMCU/.

Powering the ESP-12E NodeMCU Development Board. (2018, February 10). Retrieved from http://henrysbench.capnfatz.com/henrys-bench/arduino-projects-tips-and-more/powering-the-esp-12e-nodemcu-development-board/.