



uOttawa

Project 17

.....

Innovation of the CEED Space

Deliverable H - Prototype III and Customer Feedback

CEED

by Alec Edward, Evangelina Schonfeldt, Isabelle Tam, Rehana Lawrence

The staff, students, and community that use the CEED space need an interactive system in order to time and inventory manage more efficiently.

The focus for the project for the past two weeks has been about getting all the components of the project to work and putting them together to achieve the final design. The functionality of the design is being tested, as well as its ability to accomplish the tasks it is supposed to.

.....

Results and Development

In the last prototype, the PIR sensor was connected to the NodeMCU and through the communication port, the serial monitor was able to detect changes in motion. For this prototype, the team's own PIR sensor was delivered. Testing was done to compare the sensing capacity and range of the new sensor. The reason the NodeMCU board was used, was because it can communicate to the serial monitor over an HTTP server. For this prototype, code was written in the Arduino IDE to make this happen. The NodeMCU was connected to a mobile hotspot on a laptop and it would set up an HTTP server for sending messages. The other requirement was that the information of the motion changes could be sent to the dashboard. Once the IP address was taken from the mobile hotspot, it could be plugged into the dashboard and it could send and receive signals from the NodeMCU. In essence this is the biggest breakthrough of the entire project to date. This is the primary function of the device and it is now quote-on-quote operational. Evidently, many things must be fine-tuned and tweaked up after this step to prepare for Design Day, however the model essential works as of the end of that test.

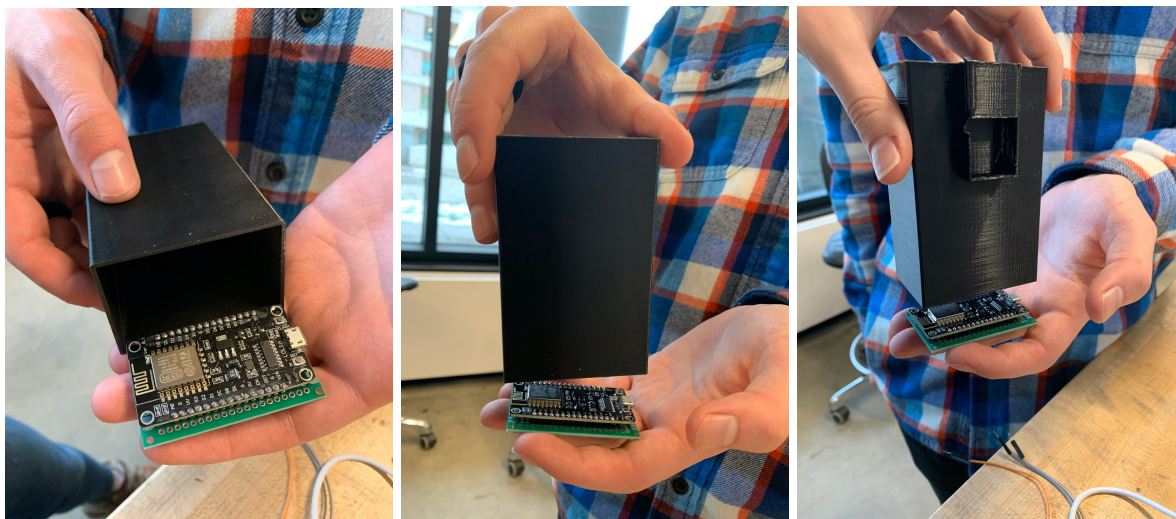
Analysis of Objectives and Testing

PIR Range and Detection:

When placing the sensor in the 3D printer, many things must be taken into consideration. For instance, the sensor should not interfere with the 3D printing and must be able to detect the motion of the arm at all points of the printing process. The first steps are to take a look at the printer, analyse the range of the arm. The goal is to place the sensor in a place that will detect motion of the arm, but also not disturb or interfere with the arm of the printer. By looking at the range of the arm, the sensor will be placed in an area that arm of the printer does not reach, but is also able to "see" the movement of the arm. An important thing to consider, is that the sensor and board will be placed inside a box that will be printed. The spot picked to place the sensor must be large enough to fit the entire box.

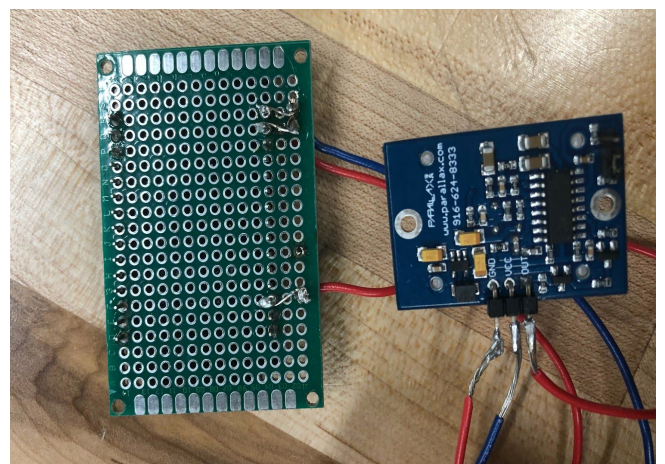


When 3D printing the box, there were some obstacles that were faced. For instance, the hole that the sensor is supposed to peak out of, was not extruded in the print. Also, the lip of the sensor hole is too small and needs to be made bigger for the final design. Lastly, the dimension of the box are not a good fit. The length and width of the box needs to be a bit bigger, and the height of the box can be smaller in order to reduce the amount of space it takes up in the printer.



Soldering the Protoboard:

For the final design, breadboards are not allowed to be used, as they are very temperamental and not viable as a permanent solution. Therefore the team got a protoboard and soldered all the components to the board. Once the NodeMCU and sensor are connected and are compatible with the dashboard, the pieces must be put together for the final design. The NodeMCU was soldered into the protoboard and connections were made between the board pins and the input/output for the PIR sensor.



As seen in the photo, connections have been connected and soldered. Additionally, a multimeter was used to ensure that the connections were made properly. The final test of the soldered protoboard was done simultaneously with the troubleshooting of the Dashboard Compatibility. We used the soldered board in this test and it worked as expected, thus completing our testing of the soldered connections.

HTTP Server Functionality:

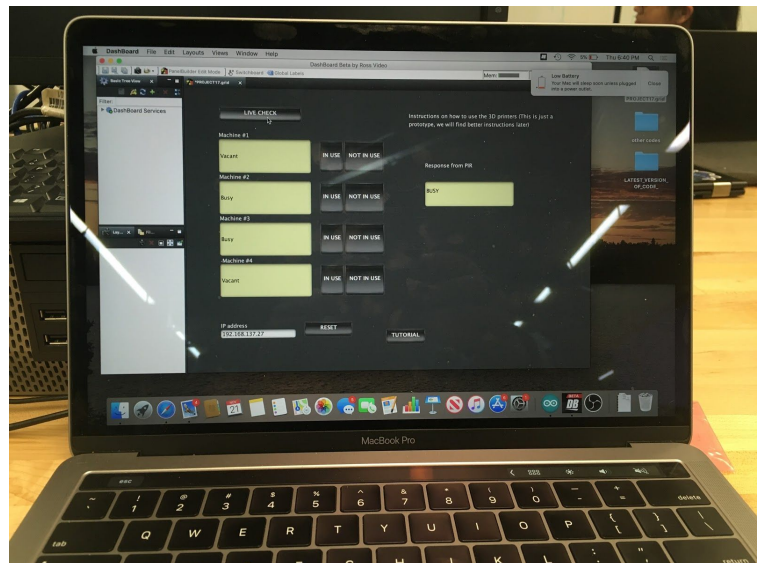
The HTTP Server Functionality is a fancy way of saying how the NodeMCU communicates and broadcasts it's information over the internet. The final goal is to get the NodeMCU to read the changes in motion of the PIR sensor and send these changes over the internet, the end receiver being the dashboard. This was tested many times, and many times it was unsuccessful. This resulted in multiple overhauls of the arduino code. Eventually we were able to get the code to run and to have the NodeMCU connect to the HTTP Server. It notified the team of this in the serial monitor with a simple line of `Serial.println`. It was possible to print out the Network it was connected to, it's IP address, and the moment the HTTP server started.

The last time the code was run, a picture was not taken of the serial monitor so here is what it looks like when the HTTP Server is being connected:

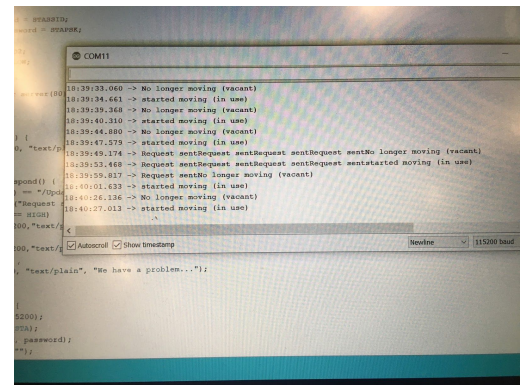
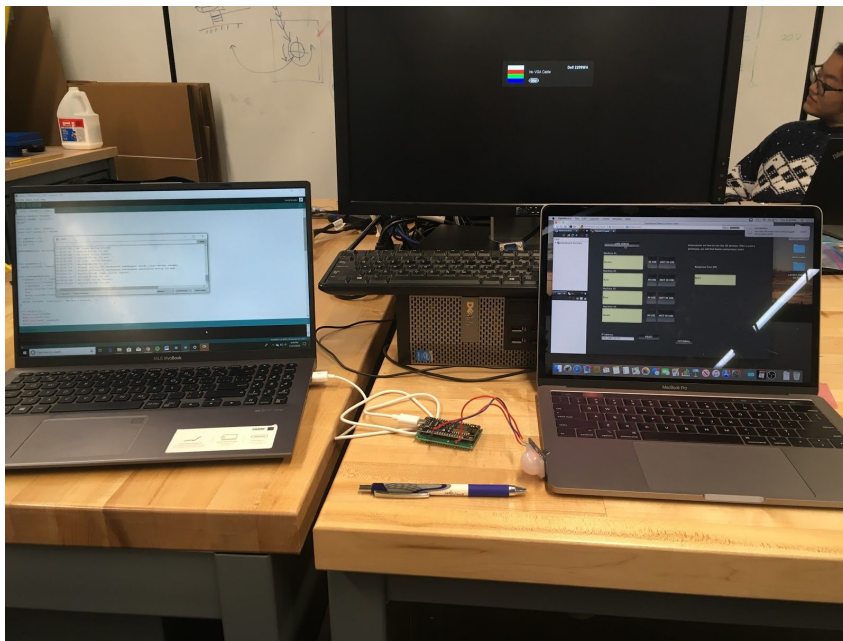
```
-> . . . . .
-> Connected to NodeMCU_connection
-> IP address: 192.168.137.27
-> MDNS responder started
-> HTTP server started
-> started moving (in use)
```

Dashboard Compatibility:

This was the one element of the project that the team had the most trouble with. The Dashboard did not want to connect with the HTTP Server. By proof of our testing above, it was confirmed that the NodeMCU was communicating over the Server, however the dashboard signal was not being received. The Dashboard code was completely rewritten, to eliminate syntax errors etc. but still the signals were not being received. While the computer running the dashboard was connected to the mobile hotspot (to host the HTTP server) an IP lookup was done to see if the computer could locate the board. The page crashed, signifying the IP address was not being located. The team tried running the dashboard on another computer but still no luck. The parameters in dashboard were deleted



and new ones were created as suspicion of cursed parameters were decently high that day. One last thing was tested. In previous tests, the IP address from the computer's mobile hotspot was being used for the dashboard. Instead, the IP address was printed in the serial monitor of the arduino IDE with a Serial.println function. For some unknown reason, the IP address of the board in the serial monitor was different from that of the IP address from the computer's mobile hotspot. The IP lookup was done and the computer was able to locate the ESP8266. When this address was plugged into the dashboard, it worked as expected which was greatly encouraging for the team. This test was completed twice, and was successful both times, therefore ending our testing in this section. The next step is to create an active listener in the dashboard so it would not need to be manually updated. This feature may be functional on Design Day.



Website Compatibility (with Dashboard):

A main component of the design is to upload the availability of 3D printers onto the website. The dashboard receives signals when motion is detected using the PIR sensor, and then information sent to the dashboard will be uploaded onto the website. To test the capability of uploading the dashboard onto the website, a computer should be able to go onto the website and see a clear image of the dashboard. Additionally, the information shown on the website should be up to date and show a live feed of the vacancy of 3D printers.

There was some trouble with uploading a dashboard panel to the website, which prompted up to seek for help. By emailing Ross Video, some information on uploading dashboard panels to a website was made available. With this assistance, we needed to assign an HTTP Trigger server port and assign the buttons we want to access to trigger IDs so that they can be seen from this page. Then, to access this via a web browser, type in the IP followed by the port we assigned under Trigger server port.

Feedback and Comments

In these past two weeks, the team has gone into the CEED space and consulted people about their opinions on the user perceptions of our product. This is a way to be notified of the shortcomings of our user interface and find solutions on how to improve it. We talked with many potential users of the software, mostly consisting of students. The most important results of the feedback sessions are summed up in a pros & cons table below:

<u>Pros</u>	<u>Cons</u>
The potential users liked how all the information as well as the vacancy monitoring system was all in one convenient place	Users were confused by the fact that they could not see the machine vacancy real-time. They wanted the “power” to click the live check button themselves rather than someone else doing it for them.
CEED users liked the design of the black box that contains the sensor. Paraphrasing some of the comments, “Looks modern and professional”.	Users did not like the fact that they had to open up a video to view the machine vacancy. (live stream solution proposed in last deliverable). They said this is inconvenient to the user and they want to see it directly on the screen when they open up the page.
Potential users liked the colour scheme of the website and the timers. They liked how there was various information about the space.	They wished that the interactive map wasn’t so busy on the main page. They also wished there was larger font (making it accessible for many types of users). Latent Data collected: it took them closer to 3min to find the contact information page which was located at the bottom of the page.

Plan for Action:

Facing the first noted downside of our project, the fact that the dashboard has to be manually updated. This is obviously an inconvenience to the user, however, the team believes they can fix this by a simple listener code in arduino, this would update the state of the machine automatically at a certain time increment. This would essentially get rid of the “live check” button and fully automate the updating process. This puts the latest up-to-date information in the users hands without having any other interference, which is what they want to see based on their comments from the interviews.

Moving on to the second downside on the list, the live stream of the dashboard. Users did not like this idea because it takes time to load videos and it would be much more convenient if the

output was displayed on the webpage itself in a simple clean manner. To accomplish this, the live stream idea was scrapped and Ross Video employees were consulted. They noted it would be possible to display the dashboard with an HTTP trigger server port. This would fix the problem the users saw about having to start a live video. The Dashboard output would be directly visible on the main page of the website.

Relating to the website itself, colour schemes will be changed to a slightly less distracting tones. The interactive map will be made to be less busy and the font of the website will be larger in order to accommodate more types of users. We will also be adding more directives and prompts in order to allow for easier navigation of the website for new users. These updates to the website will benefit the project immensely resulting in a more professional style.