GNG1103

Deliverable G: Prototype II

3D Printer Monitoring System

Submitted by

GNG 1103, Section A3, Group 1

Magdalena Richardson, 7231925

Qaiz Mohamed, 300072323

Amesh Roy, 300073404

Midas Chin, 300066761

Lucas Hubert, 8386891

November 7, 2019

University of Ottawa

Introduction

The objective of this deliverable is to develop the second of three prototypes involved in this project.

This second prototype builds on the first prototype and refines the subsystems of our design to incorporate new tests done over the past week. An upgraded SD card holder that fits the SD card sockets as well as the LED lights is created for this prototype. We also refined our card sensing method and ran tests to verify if the NodeMCU was properly reading the cards and registering them as IN or OUT of their socket. We have also changed the way our UI operates and integrates information from the printers, moving away from an external database system to an all in one information management system. We also incorporated feedback from Ross Video and MakerSpace staff in our improved subsystems to ensure that our product reflects the client's needs as accurately as possible.

This prototype is one of the last building blocks leading up to Design Day, and allows our team to further implement client feedback and tests into our design.

Client Feedback

We recently had the chance to pitch our idea to our client, Ross Video. We got a lot of positive feedback about our pitch and our first prototype, as well as recommendations and ideas to improve our design.

The client thought our problem statement was good and addressed an actual need in the Makerspace that can be solved with the help of Ross Dashboard. Their main comments about the design were about the UI. The one we currently have is less about inputs regarding the printers and more about their status at a given time. They recommended adding more inputs to improve the versatility and controllability of our design. Ideally, the client would like us to use Dashboard as a management console with lots of inputs to control things rather than a website tool.

Ross really liked the idea of having a second management-style UI for staff to gather and update information and databases about the printer through Dashboard without having to download that information externally.

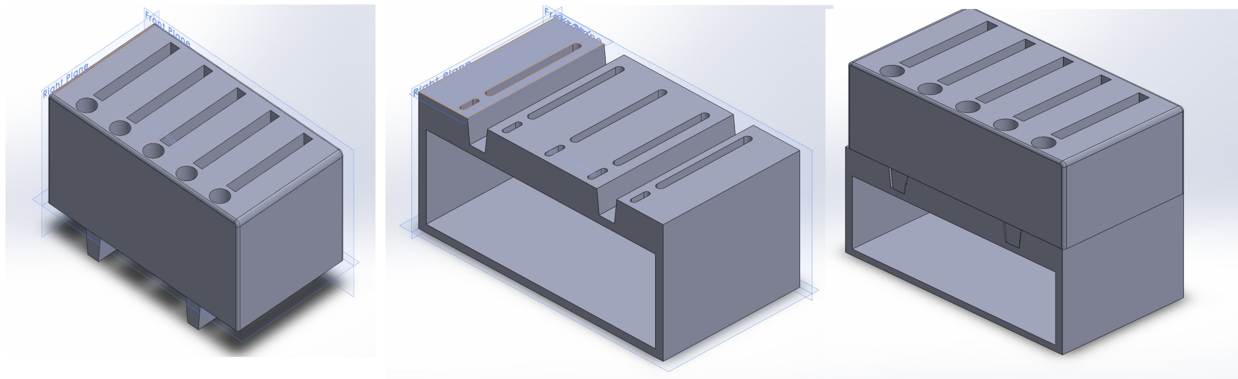
Finally, it is important to the client that we complete our project on time for design day. For that reason, they recommended not adding too many components to our design or looking for solutions that will take a lot of time to implement without a lot of added value to the project. The main goal is to have a working product by Design Day.

Discussion

While our plan had been to create a way to remotely access this information, our client feedback has brought our attention to how difficult this might be and how it might not align with the intended purpose of the software. We are now adapting our expectations for what our final product will look like on Design Day and have decided to focus our efforts on a kiosk-style display with two separate but complementary UIs - one for users to monitor 3D printer usage and one for staff to update printer information.

There are a couple ways we could incorporate remote access. One way could be to incorporate an RSS feed that sends information to a website, however currently this seems beyond the scope of the project and unrealistic to finish in two weeks. Another way could be to upload our app on the university Remote Apps service. We would need to approach the IT department in the university to see if we could have this running for Design Day.

Hardware Subsystem



Figures 1,2,3 - 3D model of the top and bottom compartment of the SD card holder and an assembly

Figure 4 - 3D printed model of the 2 SD card holder components.

Prototype two consists of an upgraded SD card holder that allows for the placement of SD card sockets and LED lights within it as well as a basic interlocking system for an underlying compartment that will be used to store the nodeMCU as well as a multiplexer. The bottom compartment will also be 3d printed and then tested. The goal of this prototype is to investigate if all the components and parts fit together and how well they are integrated with other subcomponents of the project (i.e nodeMCU)

The tests that will be carried out are very similar to the tests used for prototype 1. The two main compartments will be 3D printed and then tested to see if the interlocking system works, if the tolerancing (0.4mm for most features) is appropriate, and how much space is required if necessary. After putting the two components together, the SD sockets and the LED's will be tested to see if they fit suitably in their respective positions, and how much additional space is required if necessary. Finally, the last two tests that will be conducted are to see if the slots on the top and bottom components are aligned, and if the nodeMCU fits correctly within the bottom housing unit.

Using the same formulas used in prototype 1 to calculate the cost required to print the components, the total cost was determined to be $4.80. For the tests that will be conducted, the only dependencies are the size restrictions of the individual components (nodeMCU, LED's, and SD sockets).

After printing the two compartments, it was determined that the top and bottom fit tightly together and that an additional 1mm of tolerancing would be necessary going forward. Therefore, it can be concluded that test one is successful and that only marginal space is required for the two components to fit in the desired way.

Once the necessary electrical components were obtained, they were tested to see if they fit in their respective positions. The SD sockets were determined to fit perfectly within the socket width - wise, and it was decided that the length for each socket slot and can be reduced from 32mm to 28.5 mm and the socket will fit snugly within the slot. The LED lights on the other hand were determined to fit, however, they were extremely difficult to place and remove from their position. Therefore, it was decided that diameter of the hole where the LED's will be placed should be changed from 6.2mm to 6.4mm with tolerancing included in that dimension.

After joining the top and bottom components were joined together, wires were run through each slot to see if the slots on the top compartment were aligned with the slots on the bottom compartment. Each slot allowed for wires to pass from the top compartment to the bottom compartment, therefore making the test a success. The last test was to see if the nodeMCU fit within the bottom compartment. After placing it within the compartment it was concluded that the nodeMCU fit within it but the volume of the bottom compartment may need to be increased in the future to allow for a multiplexer and a breadboard to fit as well.

Electrical Subsystem

Each 3D printer will be represented by two hardware components, a 2-colour indicator LED that requires data wires for red (W_{red} and W_{green}) and a card socket sensor that requires one data wire (W_{sensor}). As we can see from Figs. 1-4, our final prototype will manage five 3D printers (*P*).

Our total number of data pins (*D*) needed can be represented by the following equation:

$$D = P * (W_{red} + W_{green} + W_{sensor}) = 5*3 = 15$$

Our original prototype plan had predicted that by the end of this work week, we were going to have an electrical system that was fully wired to our shift-bit register and could detect several card sockets at once. Unfortunately, due to time constraints caused by NodeMCU connection problems (see Appendix A for a small troubleshooting log) and by unrealistic inexperienced estimates of how long this would take, this step did not live up to our original plan and we do not currently have a prototype with a multiplexed capability. This is going to be crucial for our final prototype because we do not have enough pins on our NodeMCU to accommodate the 15 pins necessary for our final wiring system.

Instead, this week we refined our card-sensing method and ran tests to ensure that the card could be read reliably by the NodeMCU, registering as either IN or OUT of the socket.

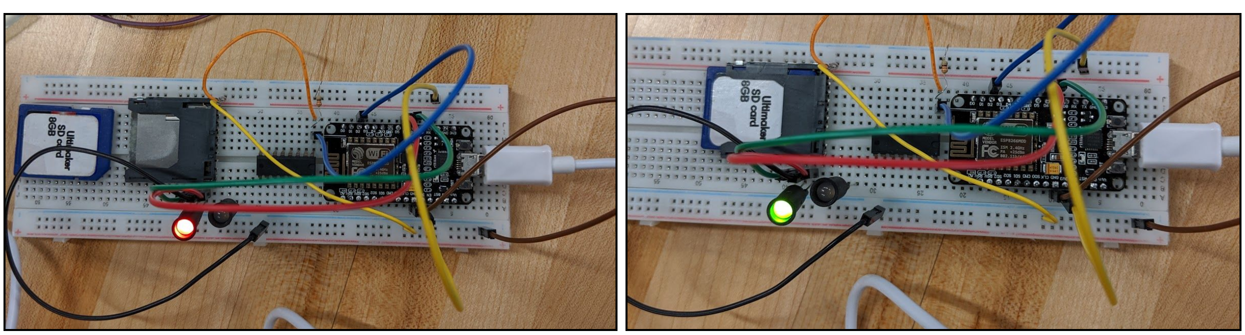


Figure 5 *- With the LED as a status indicator (red means OUT and green means IN), the NodeMCU now detects the presence of the card.*

This was accomplished by using including a pull-up resistor on the card socket according to the following diagram.

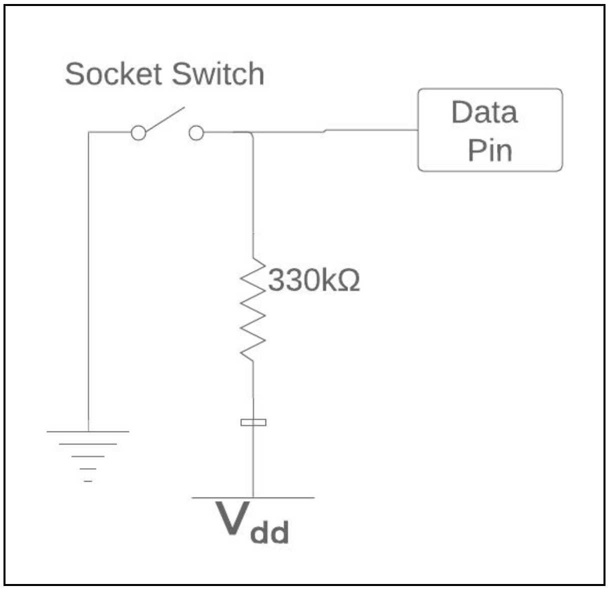


Figure 6 *- With the LED as a status indicator (red means OUT and green means IN), the NodeMCU now detects the presence of the card.*

We tested this process using a modified version of the button tutorial given on Arduino, where the LED would be red when the mechanical switch was open (ie when the card was out) and would be green when the switch was closed (ie when the card was in).

```
const int buttonPin = D2;     // the number of the pushbutton pin
const int led_r =  D6;      // the number of the LED pin
const int led_g = D7;

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(led_r, OUTPUT);
  pinMode(led_g, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(led_r, HIGH);
```

```
    digitalWrite(led_g, LOW);
  } else {
    // turn LED off:
    digitalWrite(led_r, LOW);
    digitalWrite(led_g, HIGH);
  }
}
```

***Figure** 7 - Testing was performed using an adapted version of Arduino's example program 'Button' (Adapted from code found at https://www.arduino.cc/en/Tutorial/Button.)*

Our next steps will involve wiring the shift-bit register, developing an algorithm for transmitting and receiving information, and incorporating multiple sockets into the electrical system. We may find that the shift-bit register does not easily facilitate branching off into many different data pins, either because we run out of pins on the register or because the learning curve on how to use them is too high. In this case, we might have to switch to our back-up multiplexer, which is originally intended to solder to a PCB but can be adapted to fit our needs.

Frontend Software Subsystem

We have begun integrating tasks into the Dashboard UI and started using varying parameters to modify the style of buttons to incorporate the feedback we received from Ross video. To do this, we have changed the information portion of our UI to be editable in the staff version of the UI so that these fields can be updated. It is worth mentioning that this also removes the need for us to rely on externally-stored databases like we had anticipated when we were planning Prototype 1.

The results of this change can be seen below in the following figures. Whereas before the colour of the buttons was fixed depending on the value of databases, we have now incorporated the values in the tables into editable arrays that will trigger the buttons' colour change.

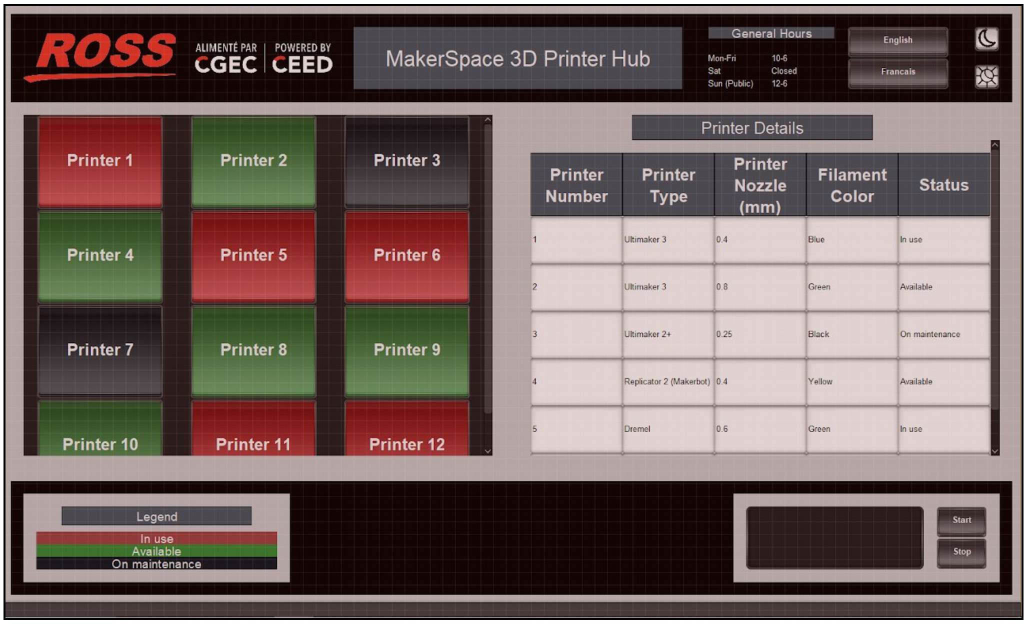


Figure # - Printer 1 has state "in use" and according to the legend, the printer 1 button turns Red.

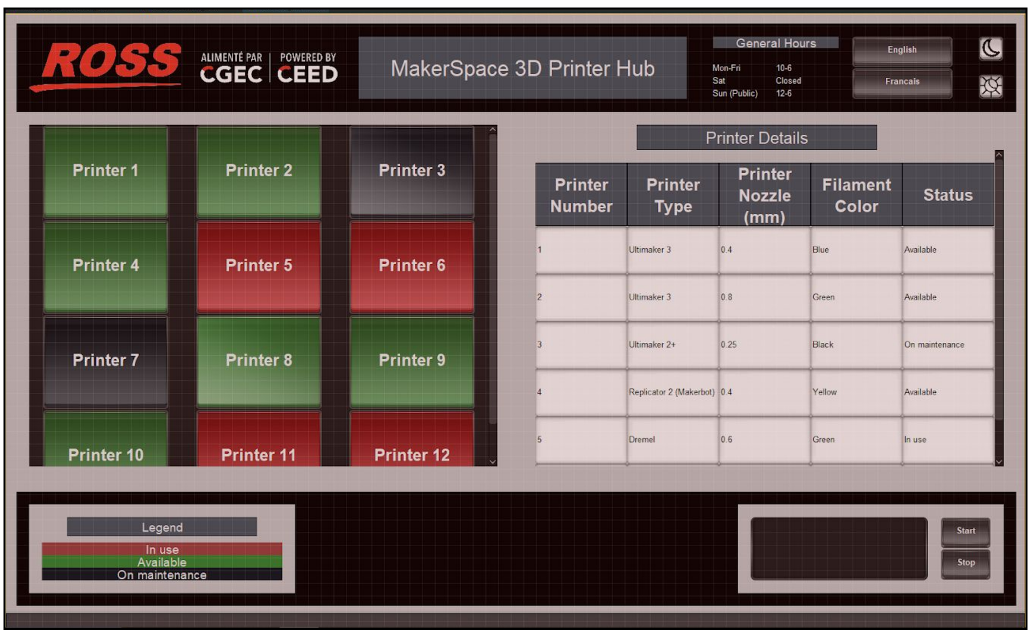


Figure # - Printer 1 has state "Available" and according to the legend, the printer 1 button turns green.

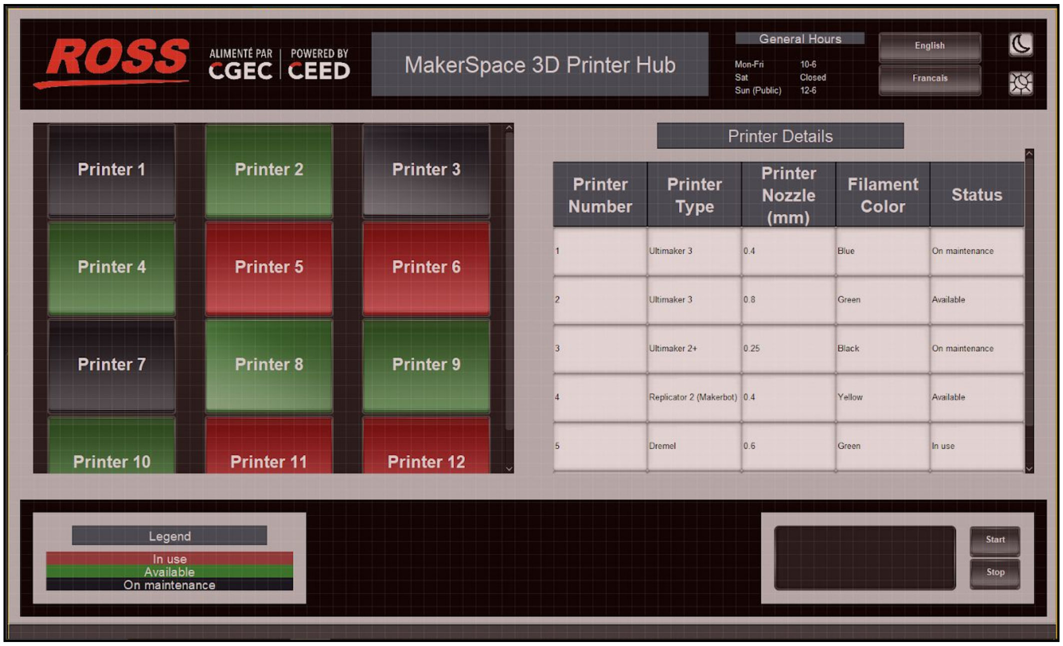


Figure # - *Printer 1 has state "on maintenance" and according to the legend, the printer 1 button turns black.*

Backend Software Subsystem

As previously stated, we have now changed the way our UI integrates information from the printers, moving away from an externally-based database system to an all-in-one information management system. For this reason, this week we did not need as much work on backend software as it has now shifted to the UI.

The next step will be integrating and testing data transmission from our device to Dashboard over WiFi. We will need to create an array that can be read using Dashboard and easily translated into values in our UI arrays.

Benchmarking

Listening to the pitch presentations, we have noticed that many teams seem to be trying to solve the 3D printing monitoring problem. Most teams plan on using a PIR motion sensor in some way in order to detect the SD card and classify the printers as available or in use. For our own design, we have decided to use sockets instead as they require less parts and are cheaper: a PIR motion sensor costs about $6-10 on Amazon while our sockets can be found at a price as low as $1.70 on some websites.

Most teams, including us, did not have a solid idea of how to make the solution remote and accessible through a webpage. This will be the biggest hurdle to overcome before design day, since the potential solutions will be long and complicated to achieve.

One other thing other teams had on us is the creation of a physical prototype that is fully connected, although our UI seems to be the most comprehensive out of all the groups working on the 3D printers monitoring problem. Ours only tested the various subsystems of our design without connecting them all together to Dashboard. This will be our next step.

Conclusion

We have had a successful second round of prototyping, even though our prototypes are not as far along as we had planned. All of our proof-of-concepts have fit within our expectations and we are on schedule for our third round of prototyping and eventually design day.

Moving forward, we will need to do more collaborative work to bring our individual prototyping work together and connect all the subsystems together with Dashboard. This will likely involve increasing the amount of times we meet out of our regular lab sessions and our weekly team meetings as we combine the electrical and software subsystems, the electrical and hardware subsystems, and the two software subsystems.