GNG1103

Deliverable K - User Manual
Submitted by

Team 4
Palmer Insull, 300069587
Jackson Williams, 300144715
Mahmoud Othman, 8544186
Rafik Salib, 300146161

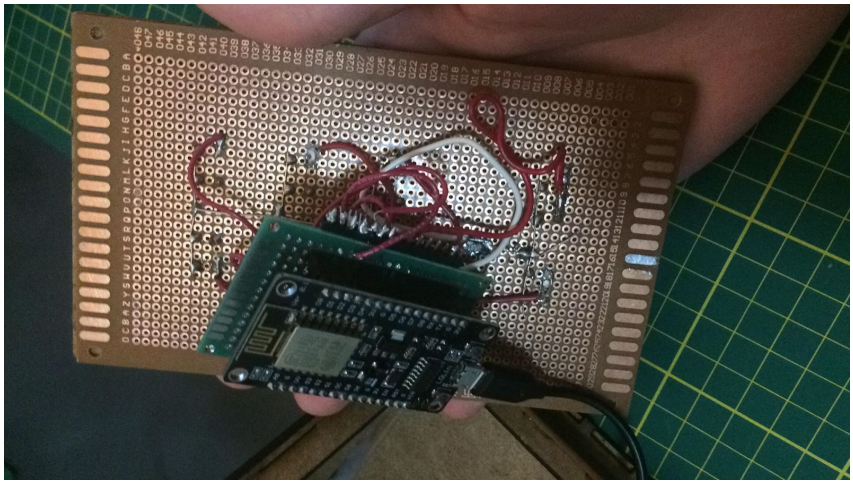
December 8th, 2019

Introduction

This manual will cover the basic steps required to operate and replicate our final product, a fully functional CEED feedback Kiosk and Dashboard interface. Our manual will be split into three brief sections, each one explaining how we designed, constructed and operated the various aspects of our final project.

Circuitry

The circuitry we designed for our kiosk was incredibly simple, it consisted of a break-out shield with solder pins for our Node MCU module, as well as a button-plate on perf board (perforated circuit board) that consisted of 8 tactile switches with a common ground wire and hook up wire between the two components.



Button Plate and Shield

The actual soldering performed in the creation of our final product was very simple and we relied on the training provided to us during the GNG 1103 labs and practice in the Makerspace.

We relied on pull-up resistors in order to have single-press tactile switches that work reliably. The digital pins we defined, 1-10 in the Arduino code are set to work as inputs on pull-up resistors so that they will register an input/button/pin is grounded with the grounding wire along the button-board. d

Arduino

Our Arduino code is well commented, self explanatory and very simple. It has all of the regular components of an Arduino IO program.

The main/important sections are described as such:

Our Code Was Adapted From The Example NodeMCU Code

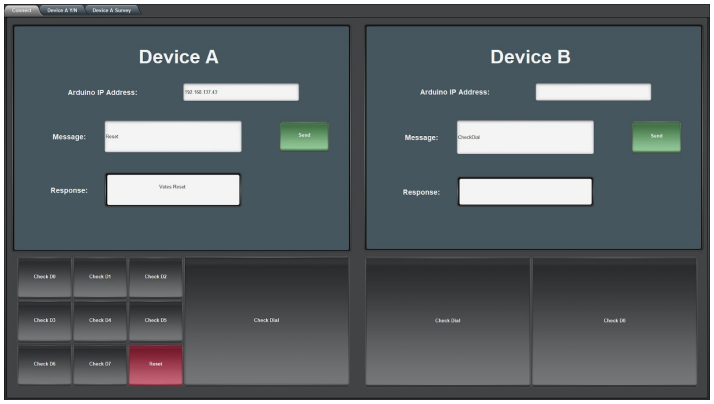
- Initial (For Global Actions)
    - Pin Setup
    - WIFI Password, ID setup
    - Global Variable Setup (Vote counters variables, pin Booleans)
- Setup
    - Pin Initialization
    - **Node MCU Setup Code**
    - Debug Messages to test startup
- Loop
    - Checking Button States
        - Very typical boolean logic for 2-state switches, boils down to when pressed (Increment and set boolean true - unless boolean true), else if boolean true and button not pressed (set boolean false)
    - Commented Out Pin-Testing
- Listener Method (**Adapted/Extended From Example)**
    - Consists of if statements to see which command is being received from the Dashboard
        - Reset
        - Individual Pin Tests
        - Send Bulk Data, using construct method data returned below
            - All in an if-else format

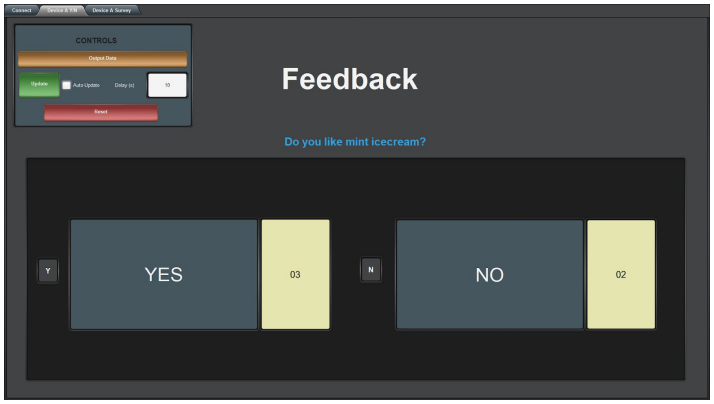
- Construct Method
    - Simply formats the counter variables as a string to send as one message

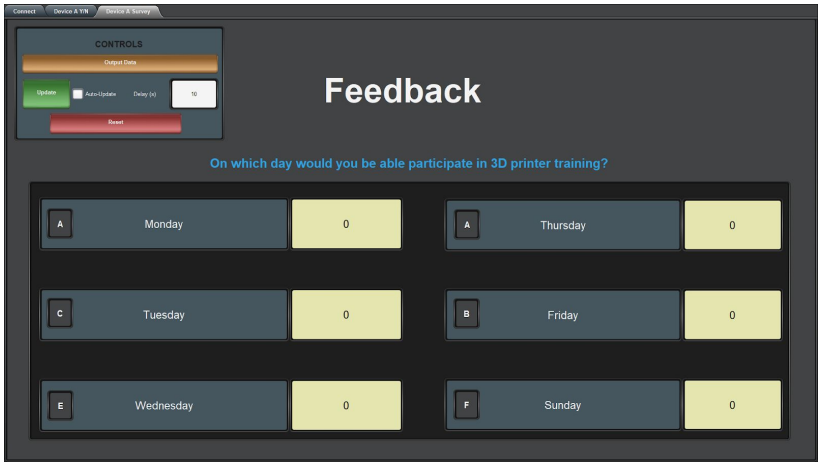
All code is fairly well commented, and formatted with sensible naming conventions so understanding and extending the code should be a simple task.

Our Dashboard design is very simple and it consists of 3 Dashboard tabs, 1 for connectivity and testing, as well as two tabs for prompted survey questions. The Dashboard code was written entirely using their OgScript editor. Most of the work done on dashboard followed the simple principles and guidelines provided in the documentation we received from ROSS video. The

connectivity and send and receive functions were adapted directly from provided visual script editor code.







1) The first panel was created exactly as described in the ROSS video NodeMCU connectivity tutorial included in the project files, in addition some buttons were added to fill in the message box to make it easier to send certain commands. When buttons are pressed, they simply set the message parameter to a different String, representing the appropriate command.

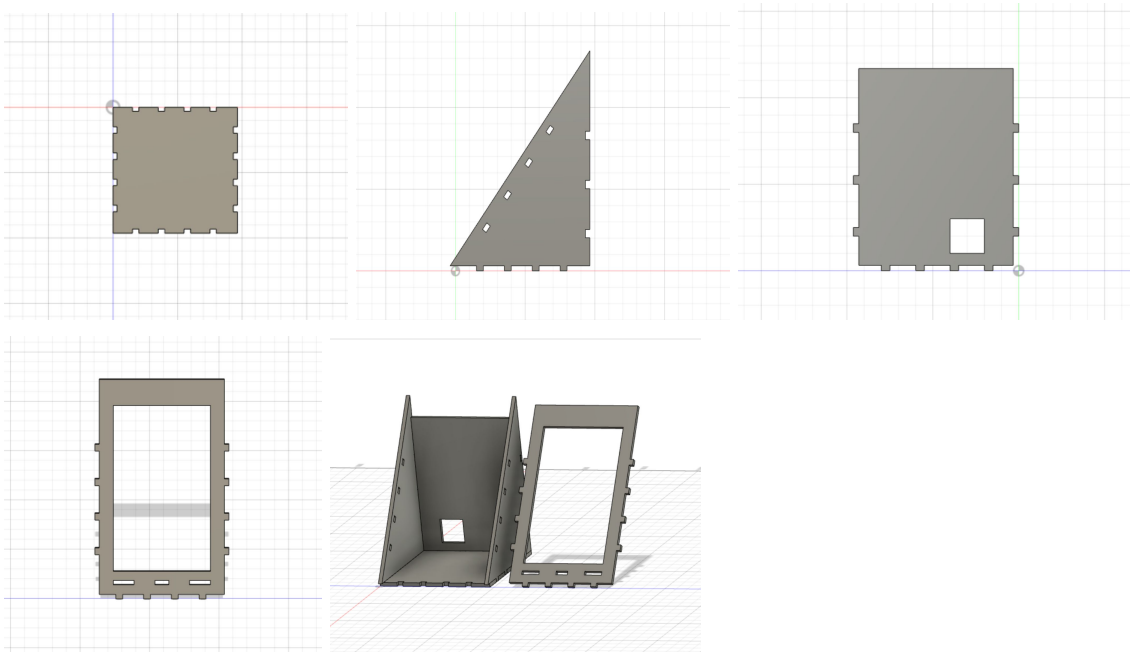
2) The second and third panels consist of the survey the survey boxes, questions, titles, and vote counters. Each of these elements is a very simple displayed parameter, all text boxes with different chosen borders/backgrounds and colors. Each one has an associated parameter than can be changed and all of the vote counter parameters are set as read-only so only the dashboard can make changes to them.

3) The final feature to our Dashboard was the file export and survey control box that is present on our two survey slides. This box is a simple ROSS video canvas with a few buttons. The update button simply sends a command to the Arduino and sets each of the relevant parameters to their associated value. The feedback for the update is sent as a long string that is parsed through to get the specific vote counts. Each of the question and answer boxes are very simple text-parameters than can be edited very easily just by typing in them. Finally the output button is responsible for exported the data to a text file located on the desktop.

- **To change the path of the update button, simple edit the path that is found within the only script under update button.**

Box Design

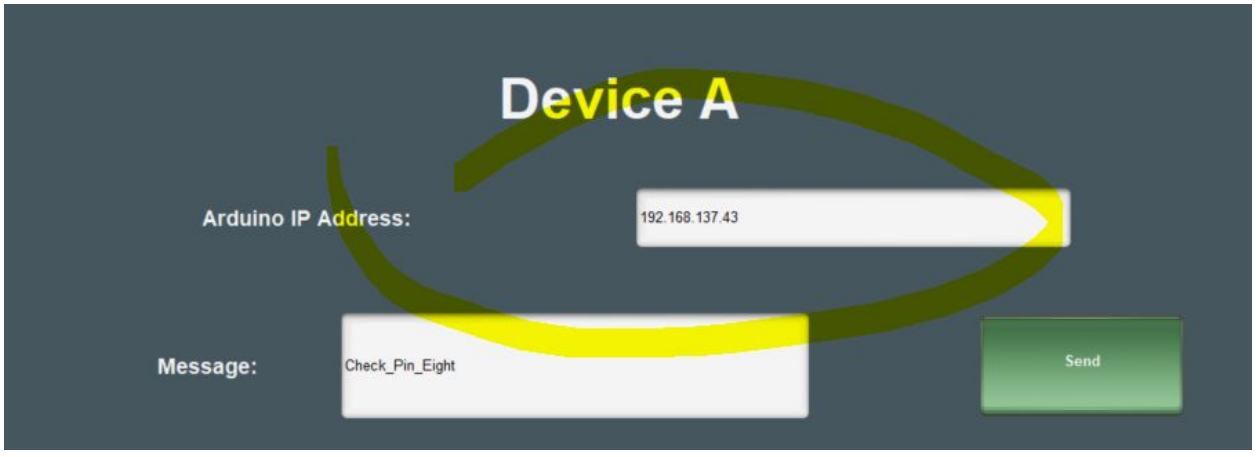
The box's first design was sketched on paper. The measurements were based off the button plate and the components that had to be held inside the box. Fusion360 was then used to make the actual design. The box had to be held together by using dovetail joints. It started with a square base that had to hold all the sides together. Then the sides were designed in a way that had to maintain the front face of the box in place by using biscuit joints. The back of the box was made for the components to be plugged in to an electric outlet. Finally, the front of the box was designed for it to be easy to access the components that had to be held inside the box and the product was finalized.



Setup Steps

To start using our product, there are just a few simple steps.

1) Plug in device into a power device or laptop
2) Setup a wireless hotspot on a laptop or computer, set the password and ID to what is specified in the Arduino code for the device, at the top.
3) Enable wireless hotspot, wait for the IP address of the device to show up
4) Copy the IP address of the device into dashboard



Repo File Descriptions

- **Fusion 360 CAD File**
- **Inkscape Vector Graphics File For Box**
- **SVG For BOX (Ready For Cutting)**
- **Arduino Code**
- **Dashboard File**
- **User Manual**
- **Node MCU ROSS Tutorial**