Drink Machine Technical Document

November 30th, 2020

Nicholas Nowosad

Project sponsors: Dave McMillan, Fraaz Kamal
ELTR2402: Embedded Systems IV

Abstract

This document contains a comprehensive report of the circuitry, programming, and testing of my ELTR 2402 embedded systems project. For this project, I opted to do a drink machine. The main reason I selected a drink machine was that it could be constructed and designed modularly so that a bare minimum could be constructed then more added on later with extra thought going into access for later expansions. It was also something that could have a fair bit of complexity to it by using motors instead of pumps to drive the movement and selection of drinks which would require motor controls and a decent bit of programming logic. This project was then carried out over 4 months starting in September, The first task of this project was to set clear goals and a tentative timeline. The first decision was that this drink machine was going to have moving components and have no pumps to avoid having a completely stationary project that would be very boring to look at. Next was that there had to be wireless connectivity for controls. To accomplish this a custom touch-pad with wireless TCP/IP capabilities was designed and implemented. With the goal of making a drink machine that has wireless control and moving parts, up next was to decide what is needed to accomplish this task. First was component research and selection. For the microcontroller the MKR-WIFI 1010 was selected for this project as it had a sufficient number of I/O ports and has WIFI / Bluetooth capabilities. Then it had to be decided how a physical mechanism was going to be used to move the drinks around. After some research, a 3 drink rotational shot stand, 3-D printer, and 12-volt NEMA 17 stepper motors/drivers were purchased to build and operate a mechanism to rotate drinks and dispense shots. Then I had to ensure there were enough inputs to meet the criteria of the project. To accomplish this a motion sensor, temperature sensors, touch sensors, and flow rate sensor were sourced although not all were used. For the motion sensor, a Panasonic active low sensor was selected for user detection. The two temperature sensors selected were a waterproof digital sensor and an IR sensor, after testing only one was used. For touch sensors, the CAP1298 capacitive touch sensor was used for its I2C interface and 8 interface pins, This chip was a 14 SOIC whereas the other chips that were available were 24 pin SOICs, and that many interface pins were not needed for this project. The flowrate monitor selected uses a hall effect sensor. The last components selected were the solenoid valves, LED’s, and power supply. Two different types of solenoid were sourced the first was a cheap no-name valve from amazon the second being a 12V ½ inch geerte. The LED’s were also cheap no-name products off amazon that had no part numbers or information but had a 1206 surface mount footprint. Lastly, the power supply was selected which was a Mean Well 100-240 VAC to 12 V 3.5 amp DC supply. The next step after ordering all the components was to test and design circuitry that would later be implemented in Multisim and Ultiboard to have a

PCB fabricated. To test components simple test programs and circuits were implemented. Testing of the components resulted in the flow rate monitor ,amazon solenoid and IR temperature sensor not being used in this project. The solenoid simply did not work, when more than sufficient current to charge the coil was applied nothing happened and no fluid could pass through. The flow rate monitor ended up working but had to operate at 5 volts which worked with pin change interrupts but risked damage to the MKR-WIFI 1010. This was not the only problem with the flow rate monitor as the physical design of the machine progressed it was found that the flow rate monitor would be impractical to implement in a meaningful way which was also true for the IR temperature sensor.The other components were found to be in working order and were implemented into a PCB design, components that had no footprint like the motor drivers and MKR-WIFI 1010 were created. Finally came the 3D design for the cases and mechanisms for driving the machine. After the physical parts for the machine were created a modular firmware was written for the standard operation of the machine. The results of all the testing and design resulted in a functioning but imperfect machine and controller that with more time and resources could be a professional looking project, not just a neat prototype. Improvements to this project in its current state and not completely changing its purpose or the majority of its design include things like case design, shot dispenser design, touchpad button implementation, along with more programming for different modes and user accessibility. Overall the project was a success and met all the goals set out to be completed.

Table of contents

List of Figures

1.0 Introduction

The purpose of this project was to build a microcontroller project that had complexity and a minimum of three inputs to obtain a decent grade for the ELTR2402 final project course. This project never had a specific problem to solve, it is a proof of concept that incorporates different electrical and physical components to drive a machine that makes drinks. This project was inspired by products that can be seen in some bars that make drinks by using the hot drink vending machine formula as well as robot bartenders like the one on the Royal Caribbean's cruise ships which have robot arms that move around and mix drinks. The scope of this project was to make a drink machine that has at least 3 input sensors, has wireless capabilities and has a user interface. This ended up including capacitive touch sensors, a temperature sensor and a motion sensor along with the MKR-WIFI 1010 for its wireless capabilities. All these features were deemed to be important because the criteria for this project included having at least 3 sensor inputs, wireless capabilities, 25 components on a PCB, and physical interaction with the world around it. There are many limitations to this project with the biggest factor being time, money, and access to facilities/tooling. The reasons these were the biggest limitation was that if I had more time the whole project would be something that was custom designed instead of purchasing pre-built products to build off of which would allow for a much more compact and user-friendly design. If there was more money I was willing to spend I'd be able to make a higher quality product that would look much more professional and not like a prototype. The last one is access to facilities which for me was limited to zero as I am living over 3 hours away from where I would have access to NAIT and there is nowhere in my town that has anything remotely similar to a technical facility, This caused me to make do with what I had which most of the time was my 3D printer basic hand tools and a half-broken DMM. With this in mind, I did what I could and planned around what I knew I'd be able to realistically accomplish. In the following sections of this report the process of how I went about creating this project will be covered. It starts with the components and why I selected them, followed by how they were tested and implemented. This is followed by the design and firmare of the final product. By the end of this report you should be able to have a complete understanding of this project from top to bottom.

2.0 Component selection/specifications

This section will be a thorough description of all components, order of selection and the initial specification for the project. Starting this section off will be the initial specifications followed by the central driving component of the project, the arduino MKRWIFI1010.

2.01 Initial Specifications

The initial specifications were set by myself with the rubric for this project in mind to try and check as many boxes as possible to get the highest grade possible with what I thought I could accomplish in the time frame with the restrictions I was working under.

-motor driven movement

-custom touchpad

-wireless operation via web based app and touchpad

-gravity fed valves

-motion sensor for to detect user

-lighting for the touchpad to respond when the motion sensor is tripped

-temperature sensing for temperature control

2.02 Microcontroller

Firstly the MKRWIFI1010 was selected for its programming IDE, 3.3 VDC operating voltage and NINA-W10 radio module that can communicate via bluetooth or wifi. When the project was initiated I did not imagine needing a lot of I/O and wanted to have wireless capabilities and this device was perfect for it.

MKR WIFI 1010 Specifications:

-SAMD21 Cortex®-M0+ 32bit low power ARM MCU, Microcontroller

-u-blox NINA-W102, Radio module

-ATECC508, Secure element

-5V input voltage

- 3.3V operating voltage

-8 Digital I/O pins

-12 PWM pins

-1 UART

-1 SPI

-1 I2C

-7 Analog input pins (8/10/12) bit

-1 Analog output (DAC 10)bit

-8 External interrupts

-7mA DC current per I/O pin

-256 KB CPU flash memory

-32 KB SRAM

-32.768 kHz (RTC), 48MHz

2.03 Stepper Motors

The ECO-WORTHY 12V Nema17 Stepper motors motor was selected next because it was going to the determining factor for selecting a power supply. Depending on the size and current drive of the motors the supply would need different specs and the rest of the discrete components along with the microcontroller do not require much current to operate. I was not sure exactly how much torque I would need to operate the systems I was making as I did not have access to any sort of measurement tools other than a multimeter so I decided that I was going to make these work.

12V Nema 17 Stepper motor specifications:

-1.8 degree step angle (200 steps a revolution)

- 0.4A a phase at 12V, holding torque of 26Ncm

2.04 Motor Driver

The Longrunner A4988 Stepstick stepper motor driver module was selected to ensure functionality and save time in the testing process. The pack that was purchased also came with heat sinks and more modules than needed so if there were any failures it would be easy to fix on a moment's notice.

Longrunner A4988 Stepstick stepper motor driver module specifications:

- overcurrent protection/ground fault protection

- five different step resolutions

-adjustable max current potentiometer

2.05 Power Supply

The MEAN WELL USA Inc AC/DC CONVERTER 12V 46W, IRM-45-12ST power supply was selected because of its 85VAC-240VAC input and 12V output at up to 3.8 A which is plenty enough to drive the motors above and any other small circuitry that will be on the board.

AC/DC CONVERTER 12V 46W specifications:

-85VAC-240VAC input

-12V output -3.8 A -46 watts

-87.5% efficiency

2.06 DC-DC step down converter

The LM2596 Step Down Buck Converter Adjustable DC to DC 3.2-40V to 1.25-35V Voltage Regulator Modules were used to supply the MKR WIFI 1010 and the solenoids of the circuit. A module was selected to reduce the chance of failure and testing time to turn in a complete project.

LM2596 module specifications:

-3.2 V to 40V input range

-2.5 A, 3A max current

-Pot controlled output

2.07 Motion Sensor

The EKMC2603112K Panasonic electric works sensor was selected due to its active low output when triggered and its wide sensing angle which would make it easy to implement.

EKMC2603112K panasonic electric works specifications:

-12 detection range

-3.3-5V operating range

-active low output

2.08 Touch Sensor

The Microchip CAP1298-1-SLwas selected due to its minimal sensing pins, many other touch sensors have in excess of 20 pins which was not necessary for this project. This chip supports I2C which is what was used in this project and with the many programmable registers this chip turned out to be very versatile in what it can accomplish from proximity sensing to wide range touch sensitivity options this chip was an easy choice.

CAP1298-1-SL specification:

-3.3-5V operating voltages

-14 pin SOIC, 8 sensing pins

-lots of options for programing via I2C registers

2.09 Digital Temperature Sensor

The DS18B20 temperature sensor was selected for its one wire interface and for its waterproof casing which allowed for fluid temperature measurements. This sensor will be the main temperature control if the project is put in a temperature controlled environment.

DS18B20 specifications:

-125 to -55 degree operational range

2.10 Analog Temperature Sensor

The analog, infrared thermopile,ZTP-148SRC1 was initially being selected for air temperature but after deciding to move away from having the project in a temperature controlled environment this sensor was selected for sensing the temperature of the stepper motors. This turned out not to work either which will be explored in the testing phase of the report.

ZTP-148SRC1 specifications:

-No input voltage

-2 grounds, 2 analog outputs

-1 thermopile,1 thermistor output

-100 to -20 degree operation range

-7.8mV to -2mv Output

2.11 LEDs

These LEDs were selected because I could get a lot of them for very cheap. They came with no distinguishing information, had no part numbers or any sort of product detail, just the footprint which was 1206 surface mount.

2.12 Transistors

The TIP122G transistors were selected because I had them on hand and never needed anything specific out of them asides being able to handle up to an amp and these can handle up to 5 amps.

TIP122G specifications:

- 5 amps collector current, 8 amps peak

2.13 3.6V Solenoid

This ½ inch 3.6V NC solenoid was selected because of how cheap it was. It had no information or model number only said it was a 3.6 V. This valve never ended up working which will be explored in the test section of this report.

2.14 12V Solenoid

This Brass liquid ½ inch 12V solenoid NC valve was selected due to the first valve not working when tested and had plenty of information available which made confident it would work.

Brass liquid ½ inch 12V solenoid NC specifications:

-6V,1.6 ADC to 12V,3ADC operation range

-Normally closed

-No head pressure required

2.15 Flow rate monitor

The DIGITEN G1/2" Water Flow Hall Sensor Switch Flow Meter device was selected to measure the drink flow rate which would determine when to shut off the valves but in the end this was just going to be tacked on and not actually interact with anything and the pin it was using ended up being needed for the motor operation after further complications that will be explored during the testing section of this report.

DIGITEN flow meter specifications:

-3.5-12 V operational range

3.0 Testing

In this section there will be in-depth overviews of how each component was tested. There will be circuit diagrams and explanations of any shortcomings/failures that caused changes to what was initially planned.

3.01 MKR WIFI 1010 Testing

To test the microcontroller all that was done was set up a simple TCP/IP request and response as the wireless capabilities were the only thing that really needed testing as the rest of what this microcontroller needs to do is basic I/O operations and were checked when testing other components that used the I/O pins. The code below in figures one and two show the final code that was used to send and receive information via TCP/IP request, the only difference from this to the test code was that there was no information being sent it was just hello being sent and received as the reply.

```
Serial.println("connected");
 String request = client.readStringUntil('\n');
 client.print("received\n");
 client.stop();
 Serial.println(request);
 return request;
```

Figure 1- Server Side TCP/IP Request

```
if (client.connect(ip, 80)) {//connects to server
  Serial.println("connected");
  client.print(Command);
  String response = client.readStringUntil('\n');
  Serial.println(response);
  client.stop();
```

Figure 2- Client Side TCP/IP Request

3.02 Power Supply Testing

To test this a power cord was cut and the ACL and ACN were connected to the supply and the output was checked with a voltmeter which read 12 VDC. This is tested more thoroughly later with more components and current intensive systems.

3.03 Stepper Motor Testing

To test the Nema 17 12V stepper motor and Longrunner A4988 motor driver, it was connected with minimal connections as seen in figure 3 below. The enable pin is active low so it is grounded along with all the sensitivity pins as no sensitivity adjustment was needed, then the necessary power connections are made with the direction and step pin are connected to the MKR WIFI 1010 with the MD_4 needing a PWM signal with a 10ms 50% duty cycle being used. This configuration worked flawlessly and the current limiting pot was used to actively adjust the current to get the maximum amount of torque for the central motor.

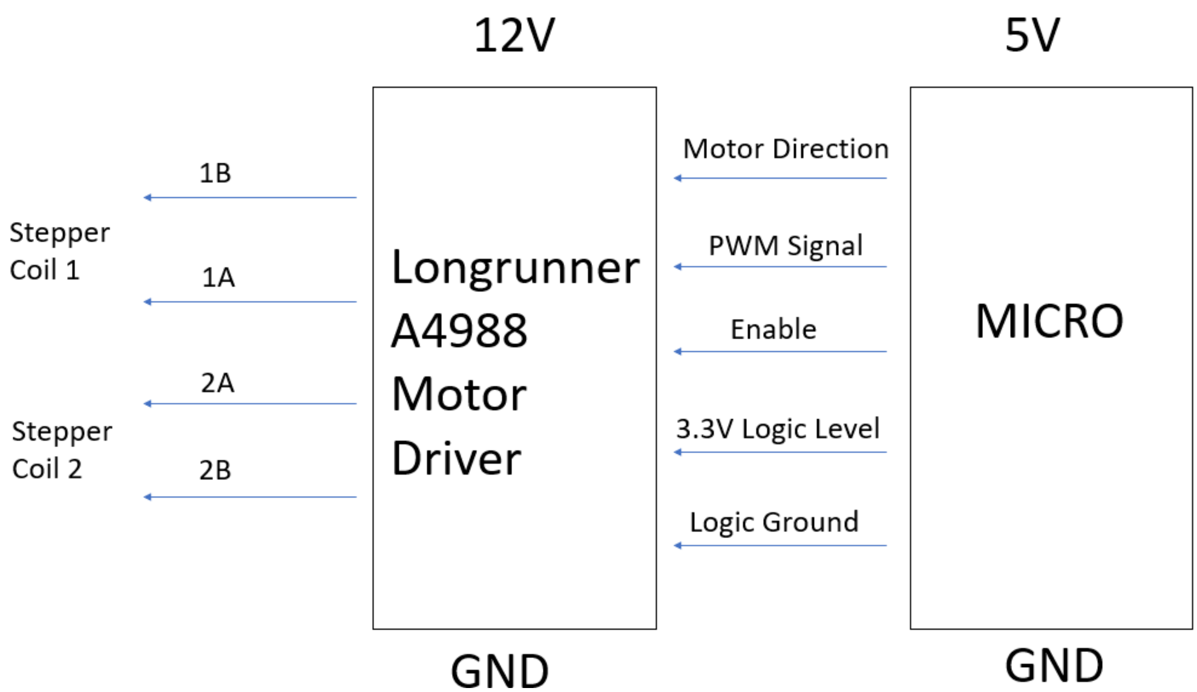

Figure 3- Motor Driver Circuit

This configuration was implemented in the final circuit on the PCB. When doing final testing it was found that the stepper motors were getting too hot too fast. With the current limit needed to have enough torque to spin the central mechanism the motors would get hot enough to melt the 3D print in a very short amount of time. To remedy this the enable pins were desoldered and a wire was soldered into the pin hole and connected to the one digital I/O pin left and the other was connected to the pin that was initially going to be used for the flow rate monitor. The flow rate monitor would have also been impractical due to physical design limitation that will be explained in the design section.

3.04 Voltage Regulator Testing

Testing this was very similar to the AC/DC power supply, to test the 12VDC output of the input of the regulator module and the output was measured while the output control pot was adjusted until 5 VDC and 6VDC were reached as those are the outputs needed to drive the solenoids and the MKR WIFI 1010.

3.05 Motion Sensor Testing

Testing the motion sensor required no circuitry, the power and ground pin were wired to the MKR WIFI 1010 and the signal pin to an analog input. From this it was found that when in close proximity the signal pin outputs zero and with nothing moving in front of it an analog value of roughly 600. If something moving in front of it was a couple feet away the analog value would go down to around 100. For this project it will check for a zero value to ensure what is moving in front of it is close by to conserve power instead of turning on light when someones walking by six feet away.

3.06 Touch Sensor Testing

To test this sensor it was connected as shown in figure 4 below with the 8 capacitive touch pins just having wires directly tied to the pin for testing and going to two throughhole pads in the final PCB. After it was connected and the data sheet was consulted to see what register to read in order to view the sensor status register (0x03) it was read while the pins were connected to wires and touched. After testing this it was found that it was good practice to write the interrupt bit in the general status register (0x00) to zero which forces the status register (0x03) low unless a pin is constantly receiving a signal. In final testing with the assembled touch pad it was found that the sensitivity register(0x1F) had to be adjusted from the nominal value of 32x sensitivity to 8x sensitivity to compensate for the parasitic capacitance from the ground plane and power lines for the LEDs.

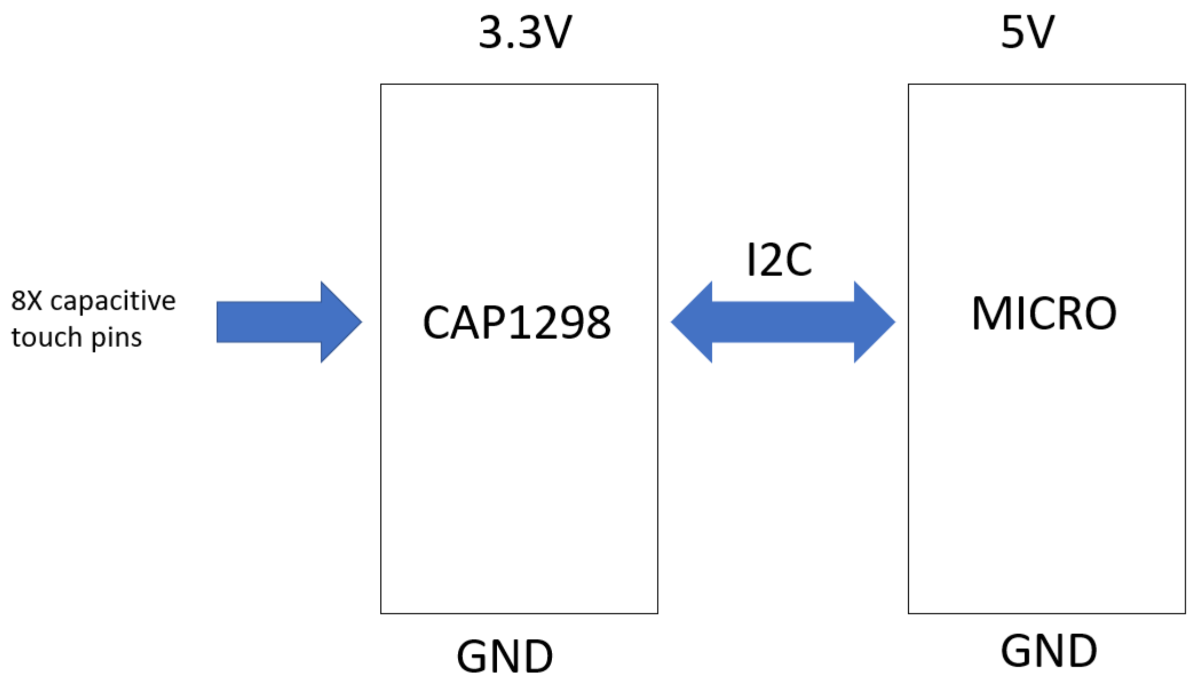

Figure 4- Touch Sensor Circuit

3.07 Digital Temperature Sensor Testing

To test this temperature sensor it was just connected to 3.3V and ground with the signal pin connected to a digital I/O pin with a 4.7kohm pull up resistor connected to the VCC of the MKR WIFI 1010.Next the one wire library was included in a sketch and a test code was implemented with a few adjustments and it started outputting accurate temperature readings.

3.08 Infrared Temperature Sensor Testing

With this sensor it was as easy as connecting the grounds and the two analog outputs to analog input pins but that was as far as I went with this sensor. I only went this far with this sensor as I was going to need to design extra circuitry to operate it looking at something that was less than 25 degrees and I had initially intended it to be viewing one of the bottles that were going to be mounted in a cooled case. I ended up not using a cooled case and opted for a more open design. This left me in a position where I would just be implementing it in a gimmicky way just taped somewhere on my 3D print and did not want to have anything that was just a gimmick and I already had a waterproof sensor for fluid temperatures.

3.09 1206 LED Testing

Since these LEDs came in ziplocks and plastic envelopes with no markings I just connected one side to 6 VDC through a voltage divider from my 12VDC supply and the other to ground and the light turned on. I left it on for around an hour and the LED never blew. With this in mind I made a transistor driven circuit with 1k ohm resistors.

3.10 3.6V Solenoid Testing

Initially this was being tested with a common collector transistor circuit seen in figure 5 being turned on via a digital I/O pin from the micro but it would not turn on. The next attempt to charge this solenoid was connecting it directly to a USB cable that was cut and striped to have the wires exposed. When connecting the solenoid to the 5VDC from the cable it never charged, this was not a complete surprise and I was ready to purchase another set of solenoids that were a little bit more expensive but had good documentation and were brass not plastic which made me fairly confident that they would work.

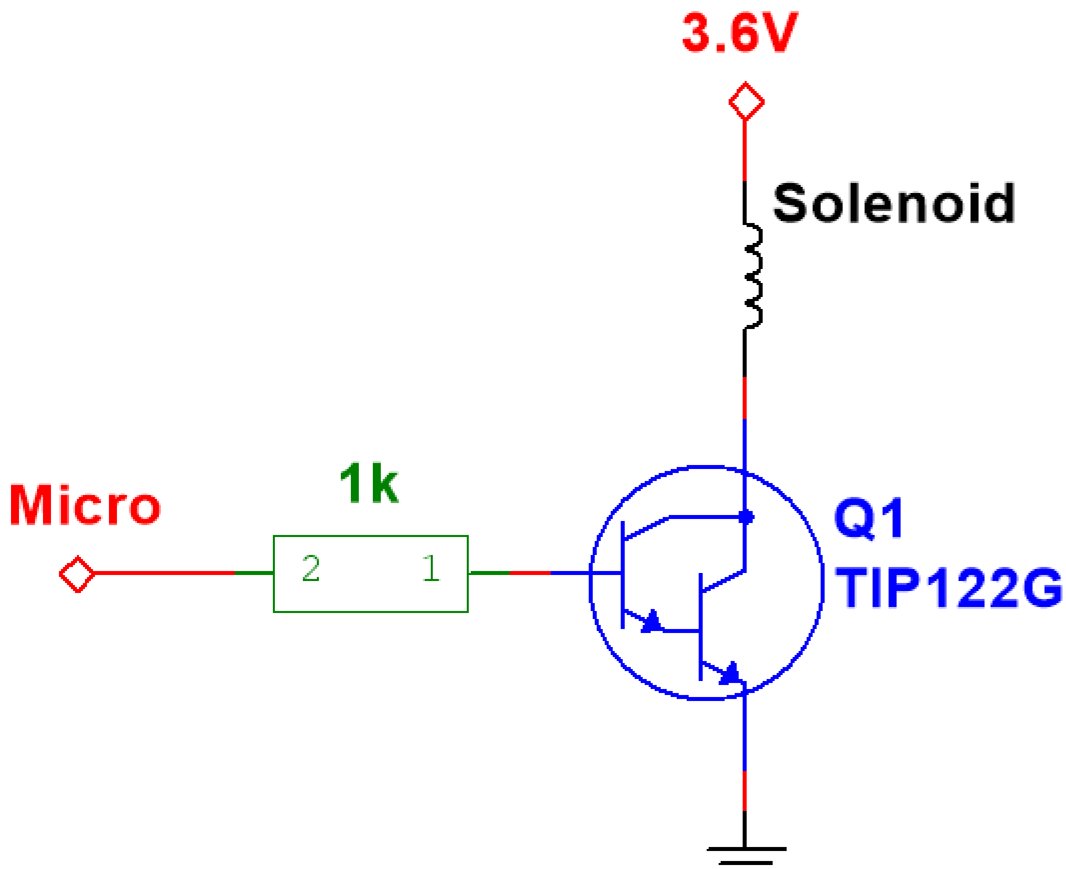

Figure 5- 3.6VDC Solenoid Test Circuit

3.11 12V Solenoid Testing

This solenoid was tested the same way as the first with a common collector transistor circuit but the input to the collector was 6VDC from the DC-DC step down converter seen in the figure 6 below. This circuit turned on when a voltage was applied with no problems.

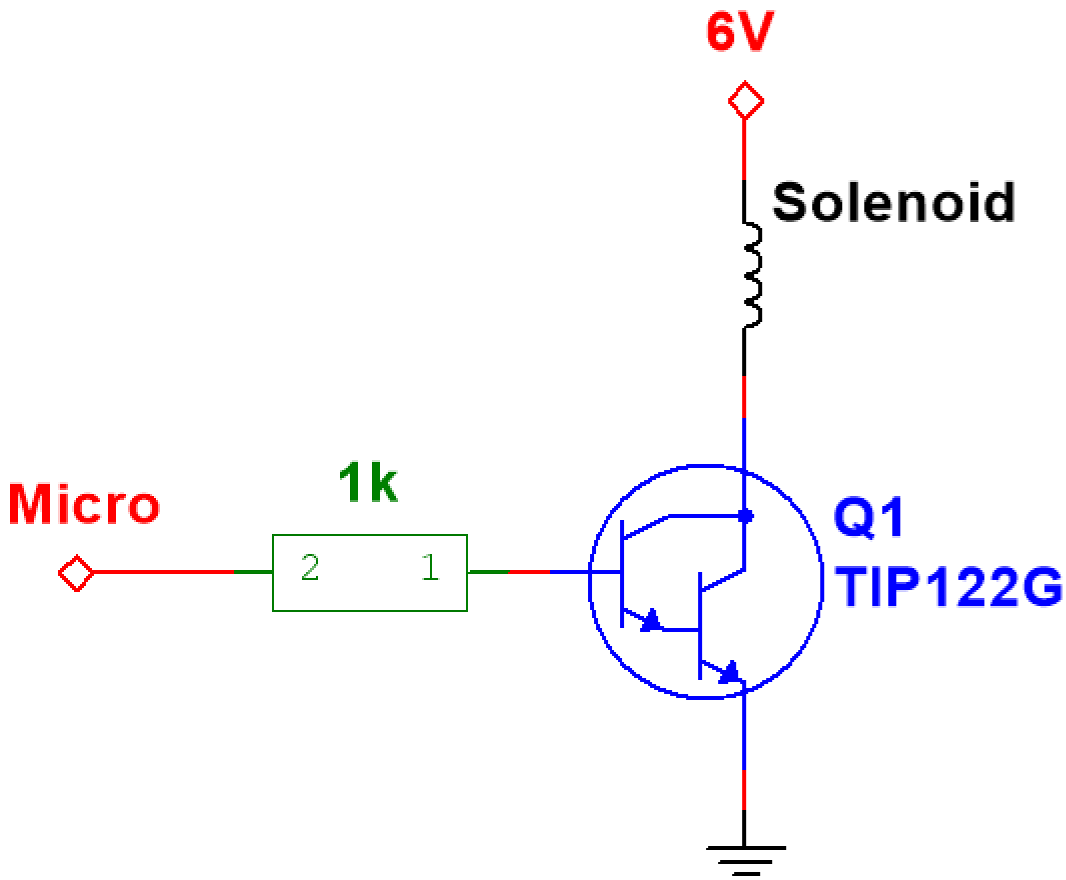

Figure 6-12VDC Solenoid Test Circuit

3.12 Flow rate monitor

Testing the flow rate monitor found that the monitor could not operate at a 3.3VDC level and needed to be connected to the 5VDC out from the MKR WIFI 1010. The signal pin also needed a pullup resistor to operate. This device worked but ended up being removed from the project for two reasons. The first being that the initial design would have had the flow rate monitor less than a foot away from the cup which was later changed to a few feet so it was pointless to have the monitor controlling the valves when there would be so much fluid in between the valve and the flow rate monitor. The second reason that came up later was that the two motor drivers needed 1 extra digital I/O pin each for the driver enable pins, so the motors would not melt the print when the coils were charged in the holding state.

4.0 Design

In this section the overall design of the project will be covered. This will include the final multisim and ultiboard design along with the case design and the problems that arose from not building in an enclosure and building a modular unit. The end of this section will conclude with an overview of the firmware followed by the conclusion of this report with improvements and recommendations that came from this project.

4.1 Multisim/Ultiboard

Seen in figures 7,8,9 and 10 on the following pages are the final circuits that are being used to drive this project. The motor driver circuits (U2,U4,U5,U6) seen and figure 8 is the second version because in the first version that was ordered the motor driver circuits had the pin ordering inverted on pins 9-16. This was due to lack of due diligence when exporting the custom foot print over from Ultiboard to Multisim. The rest of the circuits described from the testing section were operational. For this project the MKR WIFI 1010, motor driver boards and DC-DC step down converter were custom foot prints created in ultiboard. The touch sensor and LEDs were not in Multisims component database so the component wizard was used to get the footprints. The size of these boards are drastically different from one another with the motor controller board having more on it but being significantly smaller and the touchpad being much larger but having small components that do not take up much space. This was because the motor controller was designed to be as small as possible but still having access to multiple power pins for routing of extra devices as well as a header strip for all the unused pins in case more access was needed later. On the other hand the touchpad was designed to have 3cm by 3cm touch pads around the LEDs with a couple centimeters of spacing between the pads which took up the bulk of the space. It is also being powered by a battery pack made up of 4 AA batteries that was going to need a decent bit of room for mounting on the back.

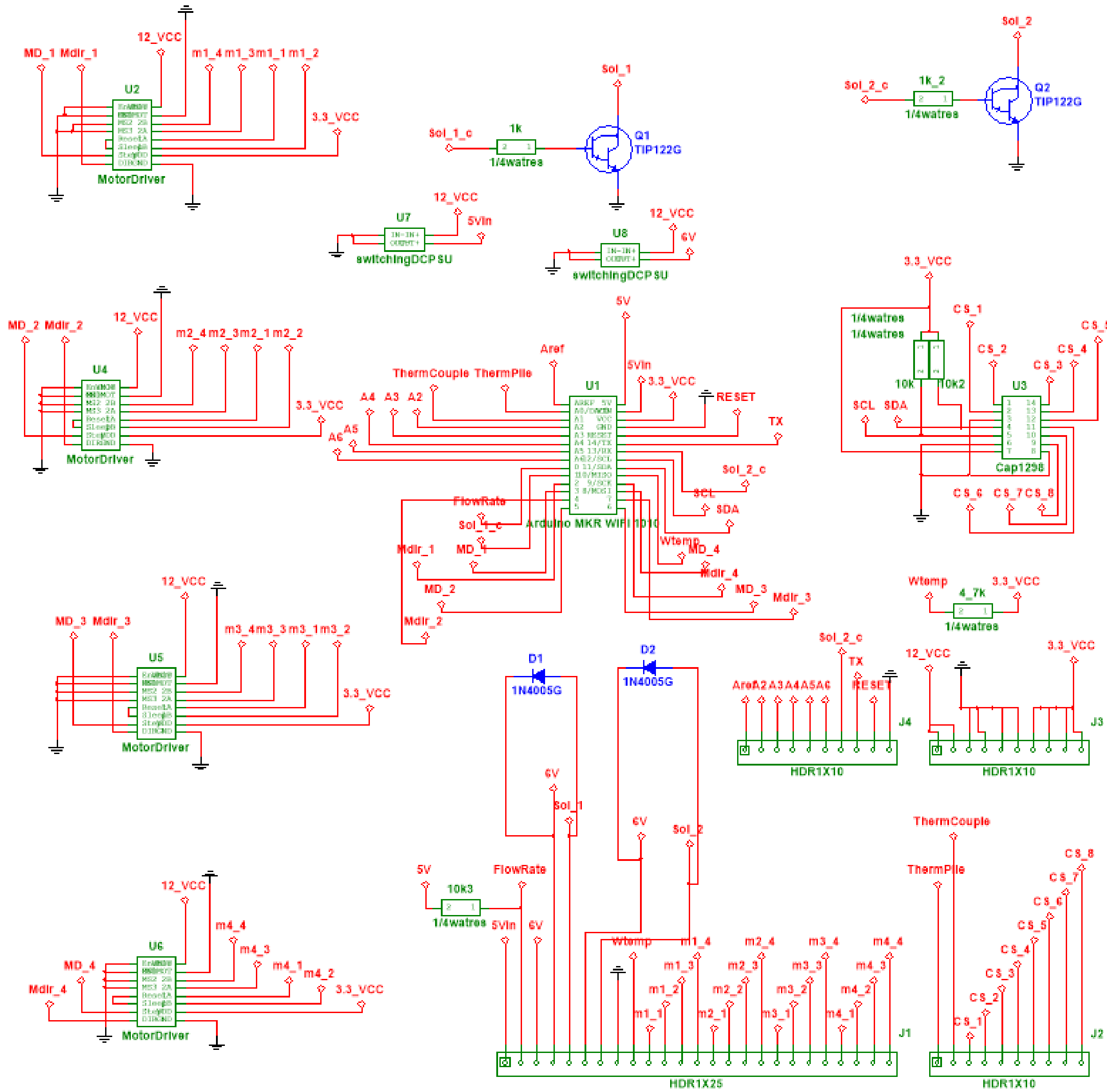

Figure 7- Main Control Board Multisim

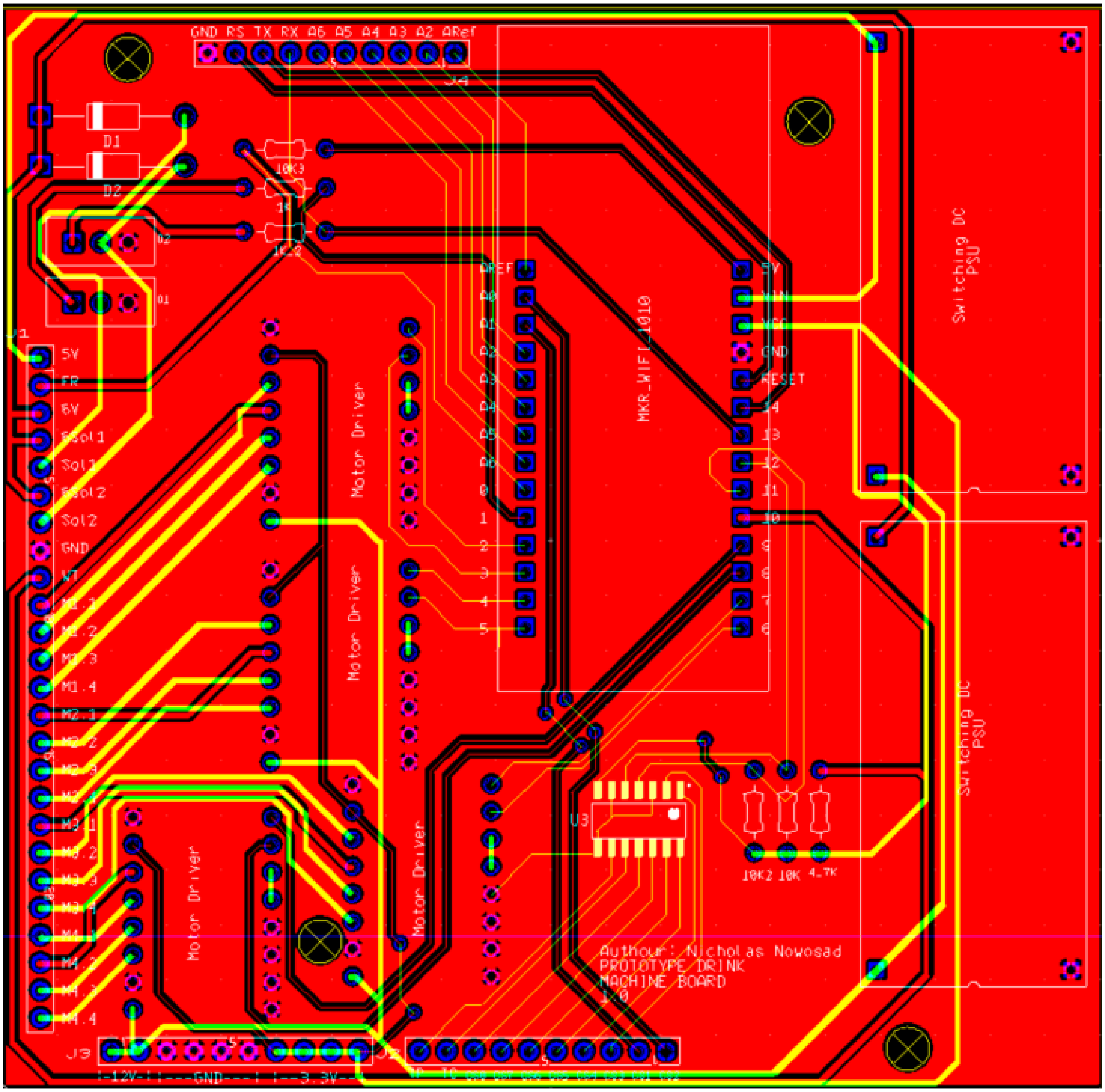

Figure 8- Main Control Board PCB

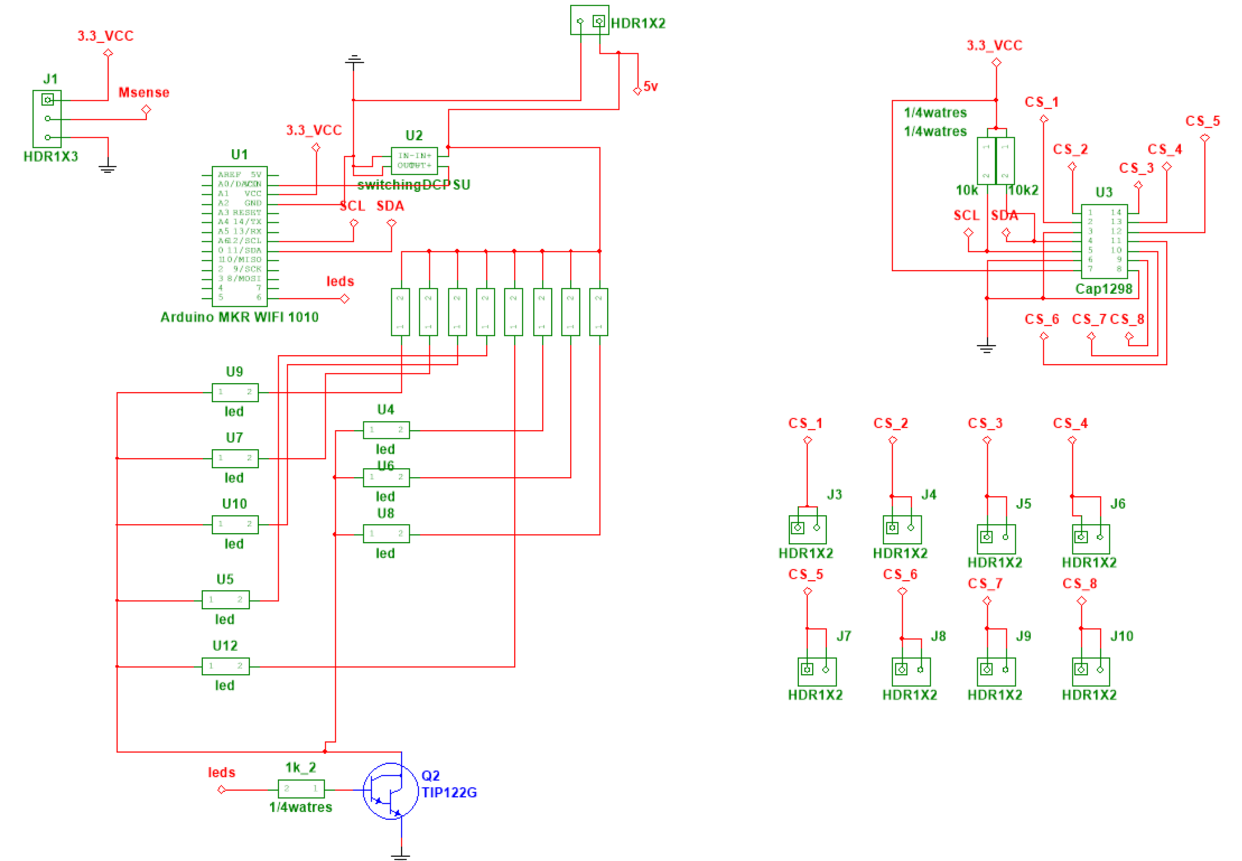

Figure 9- Touchpad Multisim

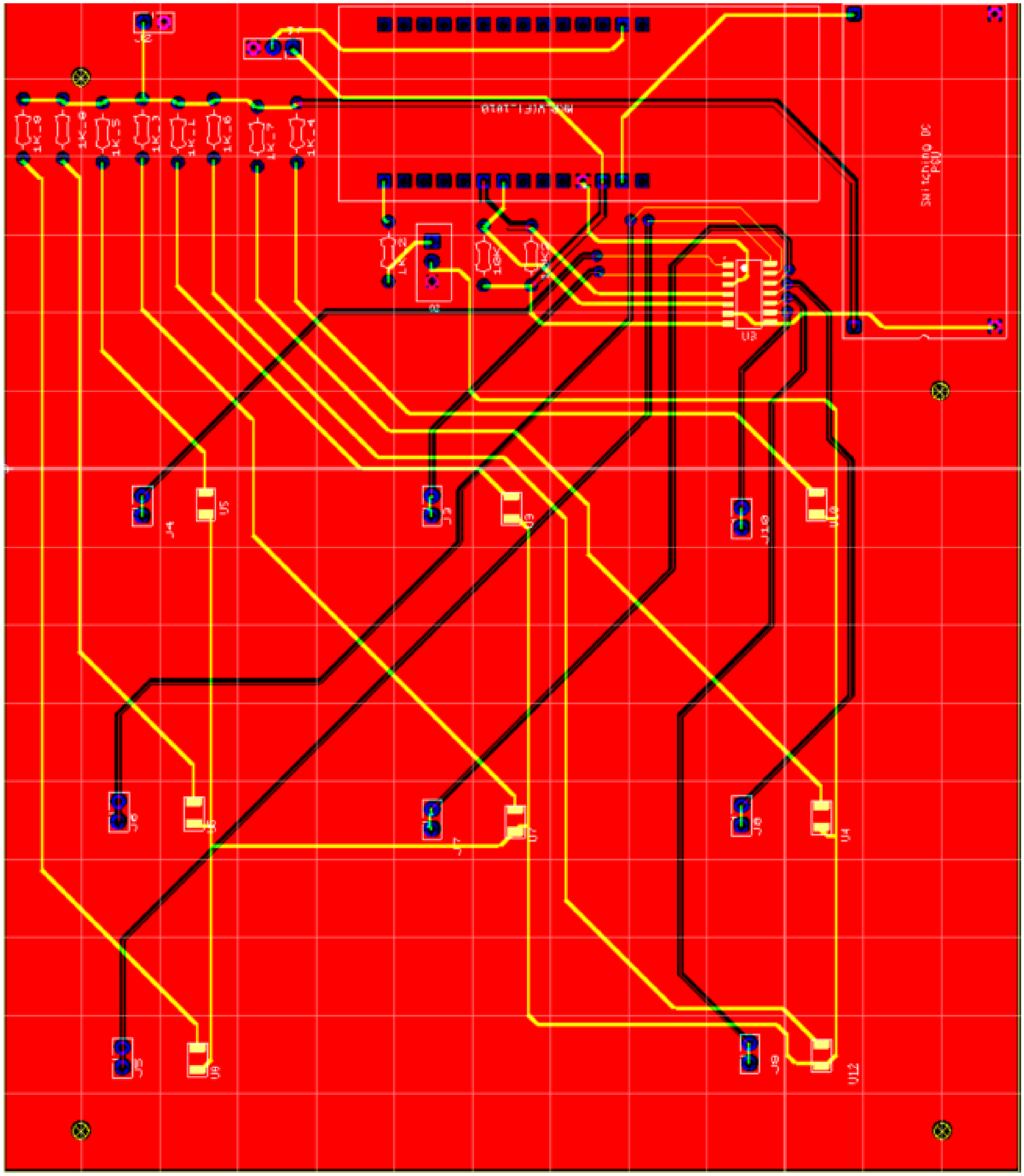

Figure 10- Touchpad PCB

4.2 Case design

The initial thought for the case was to have all the components and PCB for the controller circuit mounted in a mini fridge. After pricing out the up front cost of doing this project from home I decided that a more modular unit would be much more cost efficient but not as aesthetically pleasing as cables will have to be out in the open. When designing the 3D components for dispensing shots and spinning the drinks everything worked as intended except for the gear system that was set up for spinning the drinks. In order to spin the drinks with a gear the motor would have required more torque than could be output. This system was changed to a timing belt instead of a gear which worked but since I had already spent in excess of 600 dollars on this project including the up front cost I did not want to spend anymore money so I ended up using electrical tape as my belt. I had tried to print a belt but they were only good for one or two rotations as they were not flexible enough and either warped or broke. For the touchpad it was initially planned to sandwich a piece of tinoxied glass between two prints slightly elevated above the pcb and have a wire soldered to the board and touching the glass. This turned out to be unrealistic for my budget and tooling. Instead I soldered the pads with a little glob of soldered and then covered them with a piece of tinfoil which was then covered with a green tape to signify the touchpoint. I also created a circuit housing for the motor driver PCB that consists of a square with a few points to elevate the board for the threwhole components. This along with a small mount for the AC-DC converter is covered by a larger box with holes for cable management. The touchpad and other housing can be seen below in figures 11 ,12 and 13.

Figures 11- Central Mechanism

Figure 12- Controller Board And Case

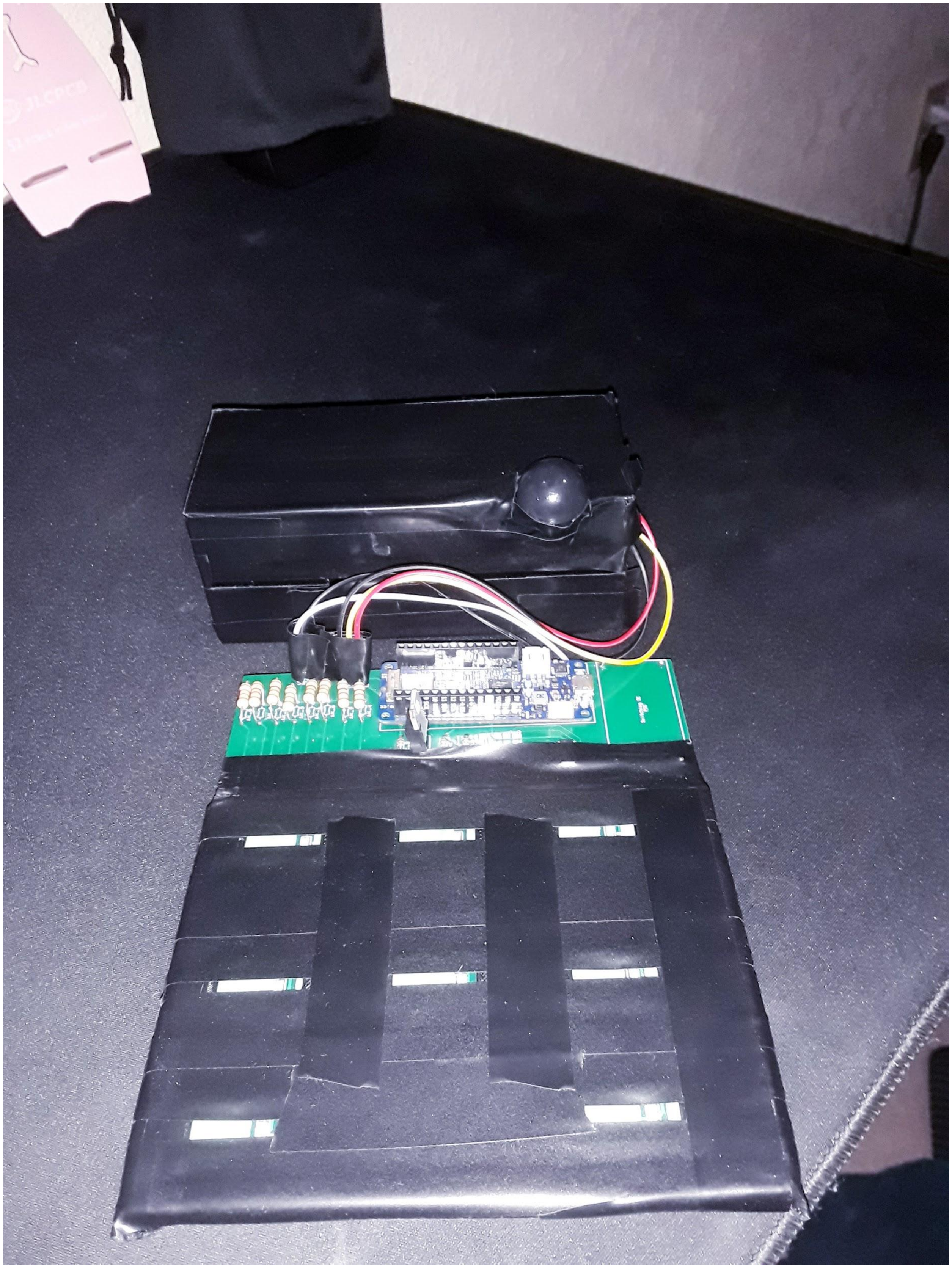

Figure 13- Touchpad With Removed Micro Housing

5.0 Firmware

In this section the firmware of both the touchpad and the motor driver board are going to be explained.The firmware for both the touchpad and the motor driver board are based on case statements and TCP/IP requests with the motor driver board being a server and the touchpad acting as a client. I also have a third MKR WIFI 1010 working with a raspberry pi and codesys to have a web based interface with a login that acts exactly as touchpads TCP/IP transfer of information. So going forward when talking about how the touchpads firmware works I am also talking about how the codesys web app relays information. The only difference is that the web app gets inputs from the user clicking buttons on the screen, not the touch pad.

5.1 Motor Driver Firmware

The motor driver firmware sets up the MKR WiFI 1010 as a server so when it is pinged it can set up connection and can receive information from the clients which are the touchpad and codesys micros. This was achieved with the WIFININA library. The main loop of this program only has a single case statement that checks for incoming client connections and gets commands that call functions that either drive the motors,valves or get the temperature. The temperature does not control anything at the moment, it only tells you if you drink is cold or not which is nice to know but I later plan to move this project into a temperature controlled environment where this temperature will control the cooling system.

5.2 Touchpad/Codesys Firmware

This firmware works in a simple manner where it has a central case statement and if a button is pressed a command is sent out to the motor driver board and the action is taken. For the touchpad there is also a motion sensor that when tripped activates the lighting and then waits a certain amount of time before checking that someone is still in front of it. On the codesys side the user acknowledgement is done with the login screen where a user has to login to access the controls.

5.3 Overview

This project uses three MKR WIFI 1010 microcontrollers to obtain user information and operate the drink machine. It does this with a custom touch pad that has foil pads. A codesys HMI on a raspberry pi with a MKR WIFI 1010 communicating to the other MKR WIFI 1010s and the motor driver board. The controller board acts as a server receiving client information and translating it into physical movement of the two motors and opening of the solenoids or reads the temperature to be indicated on the web based HMI from codesys with changes of the background colour. All the firmware mentioned above is written in a way so that checks are always being made but nothing is done until one of the checks is confirmed. This was also made in a modular fashion where all the functions used come from custom libraries made for each device and the wireless connectivity. This was done for easy recycling of code and readability. The final operation of the machine acts as follows. The clients which are the touchpad and the micro connected to a Raspberry Pi hosting a webpage interface send TCP/IP requests that initially connect then send a request. The request is a string command for the controller board micro to carry out, the response to this request is the temperature reading. This was done so the Micro connected to the web interface would not have to send as many requests due to the temperature being returned everytime the interface is used. Shown in Figure 14 below is the Firmware communication Flow chart which visually describes what was explained above.

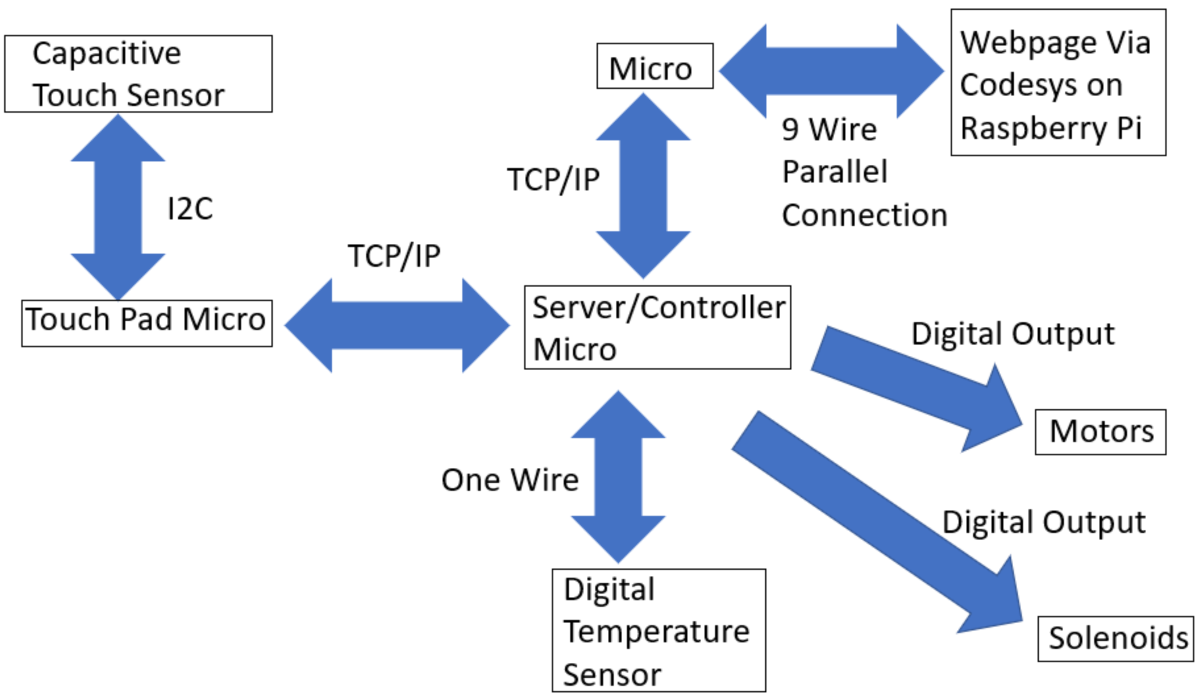

Figure 14- Firmware Communication Flow Chart

6.0 Conclusion

In conclusion this project was initiated to obtain a passing grade in ELTR 2402 embedded systems 4 and never had a problem to solve unless being sober in the 2020 pandemic could be considered a problem in which case consider it solved. This report has covered the selection of parts which mostly came down to reducing costs as much as possible. The creation of test circuits to confirm functionality of components and programming. Along with the pcb design and the custom 3D printed components which facilitates the motor movement. This report also covered the firmware which tied the whole project together into a fully operational machine that can operate indefinitely as long as batteries are replaced in the touchpad and power is maintained to the main control board. This project never hit objectives I wanted it to and never hit the ones it did in the way I would have liked to. This mostly came down to cost saving measures. An objective I never hit was having multiple temperature sensors including the IR sensor that was a medical grade sensor along with the DS18B20 temperature sensor controlling the ambient temperature instead of just making the user aware if the drink is Hot cold or room temperature. The rest of the project does what I wanted to but It just does it in a less than ideal way, this once again came down to how much money I was willing to spend on this. The ways I have the motors operating are not the most precise and this would be solved by more engineering of parts and building it in an enclosure that would have made it much easier to rotate the drinks by mounting the rotational part directly on the motor instead of using a belt system. In the end it did exactly what I expected it to. It's less than Ideal but for the amount of money I threw at it I'm pretty happy. It does what I wanted it to. The only major part that I never completed which would have cost a significant amount of time and money was the temperature control. If I were to do this again with the same budget and time frame I am certain it would come out very similarly as there is not anything that I would change unless I was willing to spend more money on this. Something for the future that I will always double check now is components created in ulitobard when transferring them to multisim, it was not a problem when I was actively working with the part but when I made one and came back to import it 3 weeks later I just assumed they were correct and never double checked which cost me a week and some money to replace the PCBs with the faulty footprint. Another thing that will be taken away from this project is how hot stepper motors get in idle state while fully charged, I just expected my 3D print to hold up and assumed they were going to warm up but they got hot enough to melt my print which caused me to have to add a little bit of code and rewire two pins to turn off the motors when they were not being used. To summarize this project was a success and I accomplished everything that I wanted after changing my design to something more modular and not in an enclosed case. This did cause me not to use everything that I purchased and tested but the project still meets the criteria and the change in design afforded me the opportunity to show some imaginative engineering to achieve the final functioning project.

References

[Online]. Available: https://www.arduino.cc/. [Accessed: 15-Dec-2020].

[Online]. Available: https://b2b-api.panasonic.eu/file_stream/pids/fileversion/4541. [Accessed: 15-Dec-2020].

[Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0198_Web.pdf. [Accessed: 15-Dec-2020].

[Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Thermometrics%20Global%20Business%20PDFs/AAS-920-773A-Thermometrics-ZTP-148SRC1-070820-web.pdf. [Accessed: 15-Dec-2020].

[Online]. Available: https://microchipdeveloper.com/32arm:sam-bare-metal-c-programming. [Accessed: 15-Dec-2020].

[Online]. Available: https://www.digikey.ca/htmldatasheets/production/1361269/0/0/1/cap1298.html. [Accessed: 15-Dec-2020].

[Online]. Available: https://www.digikey.ca/htmldatasheets/production/1801146/0/0/1/996.html. [Accessed: 15-Dec-2020].

[Online]. Available: https://www.meanwell.com/Upload/PDF/IRM-45/IRM-45-SPEC.PDF. [Accessed: 15-Dec-2020].