Project Report

ECE 3399

Weld Power Supply

April 16th, 2019

Joshua Green, Tyler Hartley, Jason Stephenson,

Table of Contents

Design

As seen in the Appendix of Fig 5, our design consisted of a breadboard circuit in a box, which connected to the high amp circuit components sitting outside. The breadboard was used after abandoning the PCB due to our H Bridge being a different size than the model we used on Eagle. The breadboard contained the H Bridge, along with its standard operating components, and the RC filter for the current sensor. The generated voltages to be used were 14 V to power the H bridge, as specified on the data sheet and 18 V to power the 4 MOSFETs. The 4 MOSFETs sat outside the box inside the mending plates, along with the inductor and the current sensor being situated outside the box.

The mosfets were arranged as 2 pairs of mosfets in parallel to compensate for the amount of current they were taking in. We used mending plates to hold each pair while helping to form the parallel connections. The selected mending plates had holes to increase convection as well as fins to reduce the heat. The pulse width from the arduino was sent to the MOSFETs via a wire soldered to an alligator clip for extra stability when connecting to the MOSFETs. The pins on the current sensor received the current in the wire leading to the load, while the data pins translated this into a voltage which was filtered by the RC circuit and directed to the Analog input pin A0 of the arduino. The arduino then compared using a PI function which would adjust the duty cycle of the PWM accordingly. With the provided capacitor bank and load resistor, we achieved an expected result of success.

Our arduino program begun by prompting the user to enter a weld time on the serial monitor. The serial value was then converted into an int which would be compared using the function milis() to keep count of how much time has gone by to apply the weld duration. It once again prompts the user to “enter 1” to start the weld. Once initialized, we measured a voltage at the load of around 1 V, which varied but stayed between 0.8 and 1.3V for the duration of the weld time, before dropping back to 0 V, as the duty cycle sets itself to 0%. Our PI controller was functional, lowering the duty cycle when the input to the sensor exceeded our setpoint and increasing the duty cycle when under the setpoint. We implemented our PI controller using a function made by the arduino community, which had K constants chosen which we did not change due to the circuit acting properly.

Design Decisions

The decisions that were implemented consist of the following: the resistor and capacitor values of the four resistors and three capacitors, the R and C values of the RC filter for the current sensor, the diameter of the PVC to be used, what type of mending plates, how to connect the whole circuit, how to implement a PI controller. First off, of the other main decisions that needed to be made was the wire that would be used in the design. Calculations using the chart provided below and the time that the wire will experience the 100A current helped come to the conclusion that 12 AWG wire would be the best option for connecting the mosfets to the inductor and the handling of the 100A of current. The other wires used were 22 AWG wire and these were used on the breadboard for the low current connections. Secondly, the 4 resistors and 3 capacitors. This was mainly on the H-bridge portion of the circuit. The other two out of 4 capacitors are the boost capacitors for the H bridge which was calculated to be 7nF and 70nF using the data sheet for the H bridge. Given the values used in actual capacitors the values used were 10nF and 100nF since that was the closest available values. The diode resistor was to be between 1 and 10 ohms which was decided to be 10 ohms as choosing the larger resistor value would be the safer bet. The last two resistors were connected to the H bridge directly and were 1 ohm resistors. Thirdly, the R&C values were determined by the Equation $RC = \frac{1}{2000}$. They were both free variables so 3.9k ohm was chosen for the resistor and the corresponding capacitance was solved by $C = \frac{1}{2000R} = 128nF$. These were chosen due to knowledge of the H bridge provided from lectures.

Fourthly, the diameter of the PVC pipe for the inductor had to be chosen. The inductance was already found previously by calculations in Fig 7 found that using a diameter of 0.48cm created an inductance using relatively low number of turns. This value came by comparing different diameter PVC pipes from home depot to see which one would result in the lowest number of turns for the length we think should be appropriate. Finally it seemed with a length of 8cm, and a diameter of 48.45mm would be adequate. This was done using the equation $L = \frac{N^2 \mu_0 A}{l}$, where $A = 2\pi r^2$, and L = 9.4μH. Then $N = \sqrt{\frac{lL}{2\pi r^2 \mu_0}}$(more in fig(8)). This would make sure the wires are used efficiently to lower the amount of wire that was used to construct the inductor. Another specification to be chosen is the way our circuit was constructed. The way the circuit is constructed not only dictates conductivity, but also structural stability. For this reason, all of the connection to do with the 12 gauge wire were connected using a connector and crimped. Additionally the PI controller had to be designed. The PI was chosen from a Function that was built by the arduino community which had preset values for the constants. This was used in many different applications by different arduino users. The only adjustment to be made was to the setpoint value.

Assembly and Testing

During the Assembly phase, all the conceptual and calculated specifications needed to be made physical. This process started at the Inductor. The PVC was purchased from home depot and came with a diameter of about 0.5cm. The turns required were calculated, the turns were secured with electrical tape, and the finished product was an inductor of about 9μH which was around our 9.4μH calculation. Next, the 12 gauge wires and positive end of the inductor were stripped, crimped with ring connectors, and secured on to the mending plates. After the wires going into and out of the mending plates were securely connected, the MOSFETs were added and the second mending plates were added to secured the MOSFETs in place. The inductor still needed the current sensor which was done by stripping the negative side of the inductor, using a female-female connector to connect the two and crimp to secure. The same was done on the other terminal of the current sensor to the load using a wire and female-female connector for the current sensor and a ring connector for the end of wire going to load. Everything was then secured and grounded.

The most challenging steps of the Assembly was the arduino schematic. To begin, the beginning of this started out very negatively. The electronics shop supplied the wrong capacitors, then the PCB had too small connections for the H-bridge, a piece of conductor pulled off the PCB, we scrapped the PCB and moved to the breadboard. After that mess, the code was uploaded, and everything was connected as shown in Fig 2 and Fig 5.

During the Testing phase there were only small changes to the circuit and code. Firstly, code changed. Our code worked which was great, however the circuit stayed active after it should have shut off. This was cured by setting the duty cycle to zero at the end of the loop which turned out to work perfectly. The second change was the multiple changing of MOSFETs. During the assembly stage, we tried soldering the MOSFETs, however, we learnt that they probably died due to heat which is our reasoning for using the mending plates to hold the MOSFETs stable. Lastly, we had a grounding error which made the voltage not get to the correct level. Once the arduino ground and the actual ground were all considered common ground, the circuit acted as it should with the proper voltage etc.

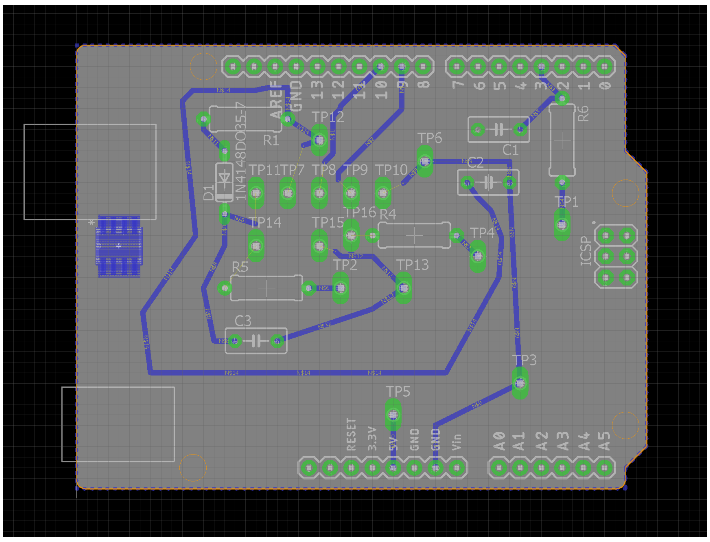
Altogether the majority of getting this circuit to operate properly was due to software issues or figuring out how to implement the code. After many hours of testing and checking all the hardware and perfecting the software, the power supply worked as desired. The end result during the testing was when the arduino received a serial input value for time as well as a start weld signal, the weld would start for the specified time while voltage instantly jumped up around 1V varying between 0.8V and 1.3V across the weld load. After the specified time, the weld stopped and voltage returned to 0V across the load.

Code

```
//Power supply code
double kp = 2;                    //PID constants
double ki = 5;
unsigned long currentTime, previousTime;
double elapsedTime;
double error;
double lastError;
double input, output;
double setPoint = 100;
double cumError, rateError;
int out1 = 9;
int out2 = 10;
int weldTime = 7;
int startWeld = 4;
int Vcomp = A0;                   //Initializing pins
void setup() {                    //Declaring output and input
        pinMode(out1, OUTPUT);
        pinMode(out2, OUTPUT);
        pinMode(startWeld, INPUT);
        pinMode(weldTime, INPUT);
        pinMode(Vcomp, INPUT);           //Define Variables we'll be connecting to
        Serial.begin(9600);              // opens serial port, sets data rate to 9600 bps
                }

double computePI(double inp)
        {
      currentTime = millis();            //get current time
      elapsedTime = (double)(currentTime - previousTime);       //compute time elapsed from
previous computation

      error = setPoint - inp;                    // determine error
      cumError += error * elapsedTime;           // compute integral
      double out = kp*error + ki*cumError;       //PI output
      lastError = error;                         //remember current error
      previousTime = currentTime;                //remember current time
      return out;                                //have function return the PI output
        }

void invertAnalogWrite(int pin, int value)
        {
          analogWrite(pin, value);
```

```
      TCCR1A = TCCR1A & ~B00110000;           //switch off output B
      TCCR1A |= B00110000;                    //switch on the B output with inverted output
     }

void loop() {

     int startWeld = 0;
     int weldTime = 0;
     Serial.println("Enter a weld time");
     while(!Serial.available()){}
     weldTime = Serial.parseInt();
     Serial.println("Enter 1 to start welding");
     while(!Serial.available()){}
     startWeld = Serial.parseInt();

 while (startWeld == 1) {      //while the start value is found, start weld
       int period = millis();
       while(millis() - period <= weldTime)
{              //while the timer is less than the weld time

       Serial.println("lol");
       analogWrite(out1, 25.5);             //make out1 pulse with a 10% Duty
       invertAnalogWrite(out2,25.5); //make out2 invert the signal being sent with 90%
duty cycle (1-D)
       input = analogRead(A0);              //read from rotary encoder connected to A0
       output = computePI(input);
       analogWrite(out1, output);             //make out1 pulse with a 10% Duty
       invertAnalogWrite(out2,output); //make out2 invert the signal being sent with 90%
duty cycle (1-D)
       startWeld = 0;
       }

    weldTime = 0;
     analogWrite(out1, 0);             //make out1 pulse with a 10% Duty
       invertAnalogWrite(out2,0);

     }
}
```

Appendix



Fig(1)

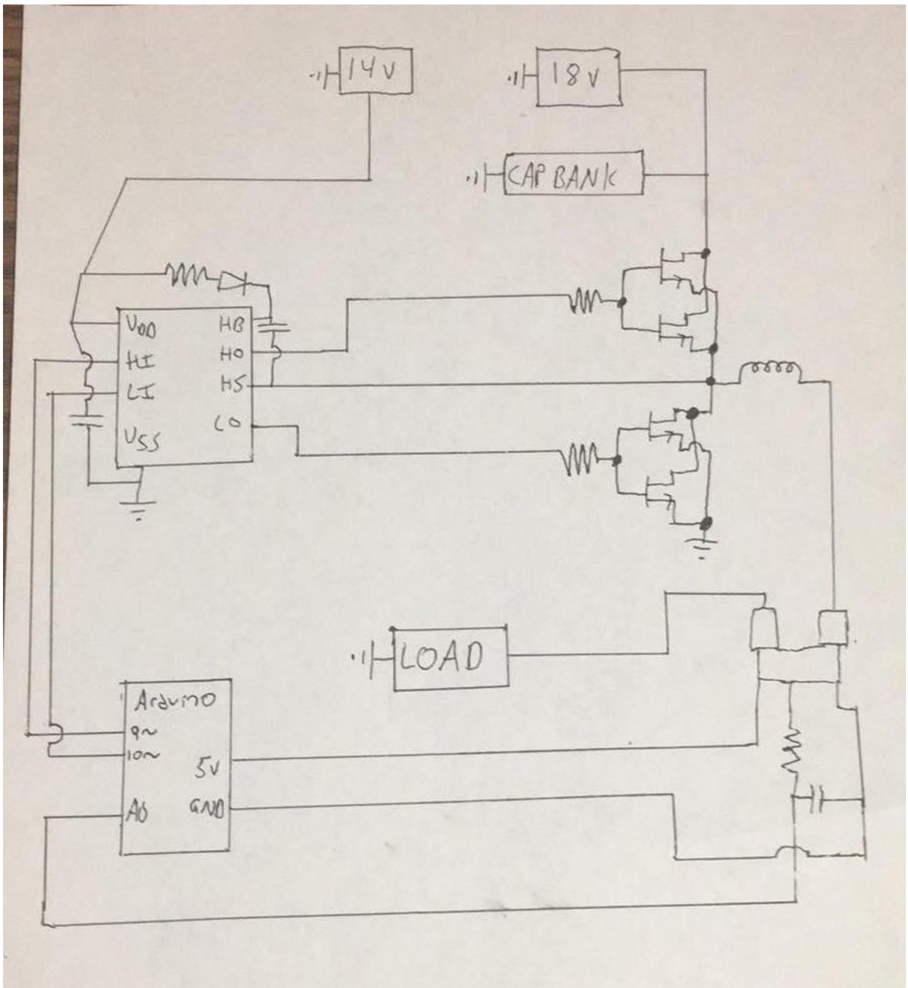
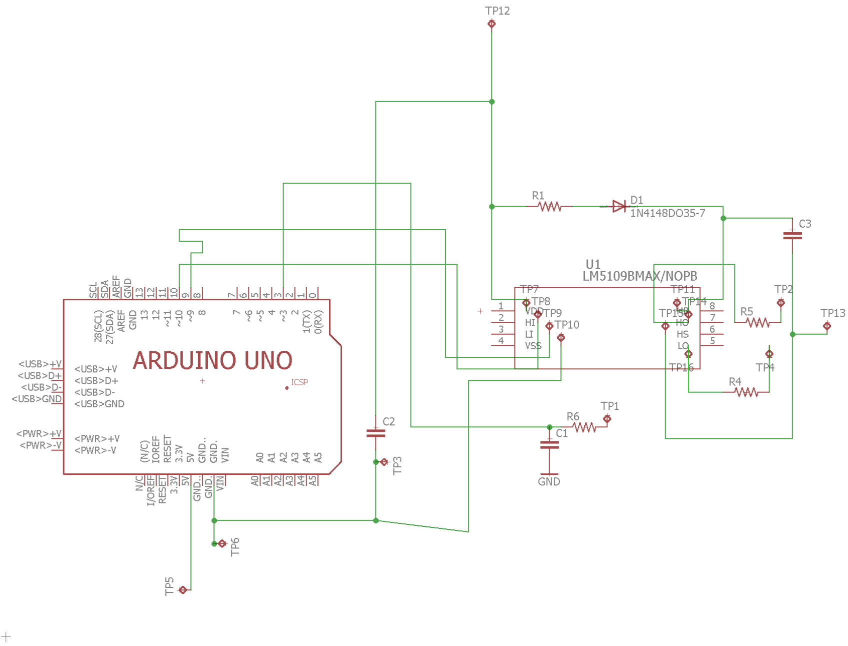
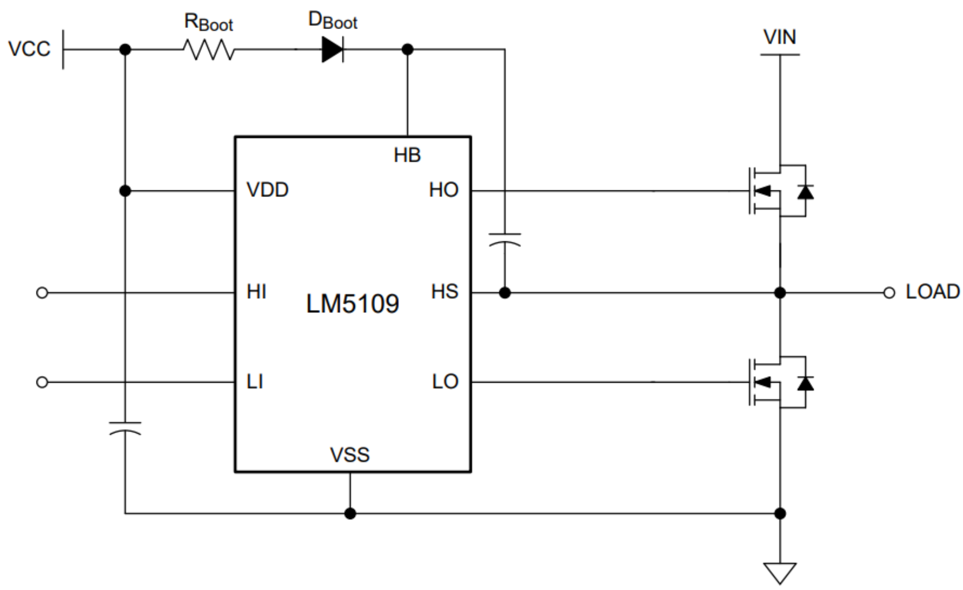


Fig (2)

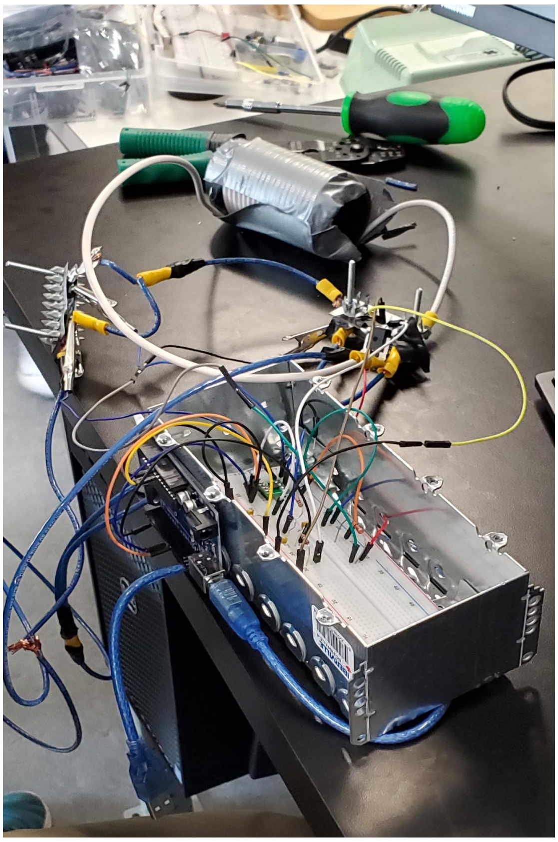


Fig(3)

Simplified Application Diagram



Fig(4)



Fig(5)

AWG	Dia Inch	Cir Mil	Dia mm	Area Inch^2	lb/kft	ohms /kft	Ohms /km	CU Max free-air Amps	CU Max enclosed Amps
32	0.008	63.2	0.20	4.964E-05	0.19	164.1	538.4	.53	0.32
30	0.010	100.5	0.25	7.894E-05	0.30	103.2	338.6	.86	0.52
28	0.013	159.8	0.32	1.255E-04	0.48	64.9	212.9	1.4	0.83
26	0.016	254.1	0.40	1.996E-04	0.77	40.81	133.9	2.2	1.3
24	0.020	404.0	0.51	3.173E-04	1.22	25.67	84.22	3.5	2.1
22	0.025	642.4	0.64	5.046E-04	1.94	16.14	52.95	7.0	5.0
20	0.032	1,021.5	0.81	8.023E-04	3.09	10.15	33.30	11.0	7.5
18	0.040	1,624.3	1.02	1.276E-03	4.92	6.385	20.95	16	10
16	0.051	2,582.7	1.29	2.028E-03	7.82	4.016	13.18	22	13
14	0.064	4,106.7	1.63	3.225E-03	12.43	2.525	8.284	32	17
12	0.081	6,529.9	2.05	5.129E-03	19.77	1.588	5.210	41	23
10	0.102	10,383.0	2.59	8.155E-03	31.43	0.999	3.985	55	33
8	0.128	16,509.7	3.26	1.297E-02	49.98	0.628	3.274	73	46
6	0.162	26,251.4	4.12	2.062E-02	79.46	0.395	1.296	101	60
4	0.204	41,741.3	5.19	3.278E-02	126.35	0.248	0.8136	135	80
2	0.258	66,371.3	6.54	5.213E-02	200.91	0.156	0.5118	181	100
1	0.289	83,692.7	7.35	6.573E-02	253.34	0.123	0.4035	211	125
0	0.325	105,534.5	8.25	8.289E-02	319.46	0.0983	0.3225	245	150
00	0.365	133,076.5	9.27	1.045E-01	402.83	0.0779	0.2556	283	175
000	0.410	167,806.4	10.40	1.318E-01	507.96	0.0618	0.2028	328	200
0000	0.460	211,600.0	11.68	1.662E-01	640.53	0.04901	0.1608	380	225

Fig(6)

$$D = \frac{V_L}{V_S} = \frac{1}{18} = 0.0556 \quad \therefore D = 0.0556$$

$$L = \frac{(1-D)T}{\Delta I} = \frac{(1-0.0556)\left(\frac{1}{20KHz}\right)}{5A} = 9.4\,\mu H$$

$L = 9.4\,\mu H$

$$C = \frac{1}{\Delta V}\int I\,dt = \frac{I_S T(1-D)}{\Delta V} = \frac{10\left(\frac{1}{20KHz}\right)(1-0.0556)}{0.05}$$

$C = 9.4\,mF$

$$n = \sqrt{\frac{L\ell}{A(1.26\times10^{-6})}}$$

For $14 \le n \le 62$ depending on what is used as area and length

ℓ	A	n
0.2	0.00785	14
0.5	0.00785	22
1	0.00785	31
0.1	2×10^{-4}	62

Fig(7)

$$L = \frac{N^2 A \mu_r}{\ell} = 9.4\,\mu H$$

$$N = \sqrt{\frac{L\ell}{A\mu_r}}$$

$\ell \approx 8cm$

choose PVC $d = 48.45mm$

$$N = \sqrt{\frac{9.4\mu H(0.08)}{\pi(0.024225)^2(1.26\times10^{-6})}}$$

$N = 17.996$

$N = 18$ turns for 8cm in length

Fig(8)

Citations

[1]"Wire-Gauge Ampacity", *Xtronics.com*, 2019. [Online]. Available: https://xtronics.com/wiki/Wire-Gauge_Ampacity.html. [Accessed: 04- Mar- 2019].

[2]"DC-DC Converter Tutorial - Tutorial - Maxim", *Maximintegrated.com*, 2019. [Online]. Available: https://www.maximintegrated.com/en/app-notes/index.mvp/id/2031. [Accessed: 04- Mar- 2019].

[3]"Buck converter", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Buck_converter. [Accessed: 03- Mar- 2019].

[4]M. Salem, "Control and Power Supply for Resistance Spot Welding (RSW)", Ph.D, Western University, 2011.

[5]*How to Design a Coil for Specific Inductance*. 2016.

[6]T. Hirzel, "Arduino - PWM", *Arduino.cc*. [Online]. Available: https://www.arduino.cc/en/Tutorial/PWM. [Accessed: 04- Mar- 2019].

[7]"200 V 100 A MOSFET | Mouser Canada", *Mouser.ca*. [Online]. Available: https://www.mouser.ca/Semiconductors/Discrete-Semiconductors/Transistors/MOSFET/_/N-ax1sf?P=1yw76e0Z1yw78i4. [Accessed: 04- Mar- 2019].

[8]*Instructables.com*, 2017. [Online]. Available: https://www.instructables.com/id/Girino-Fast-Arduino-Oscilloscope/. [Accessed: 04- Mar- 2019].

[9]Ming-Xiang Lu, Bo-Han Hwang, Jiann-Jong Chen and Yuh-Shyan Hwang, "A sub-1V voltage-mode DC-DC buck converter using PWM control technique," *2010 IEEE International Conference of Electron Devices and Solid-State Circuits (EDSSC)*, Hong Kong, 2010, pp. 1-4.
doi: 10.1109/EDSSC.2010.5713791

[10]M. Q. Duong, V. T. Nguyen, G. N. Sava, M. Scripcariu and M. Mussetta, "Design and simulation of PI-type control for the Buck Boost converter," *2017 International Conference on ENERGY and ENVIRONMENT (CIEM)*, Bucharest, 2017, pp. 79-82.
doi: 10.1109/CIEM.2017.8120769