Portland State University

Capstone Project Report

Department of Electrical & Computer Engineering

3D Metal Printer

Members:
Cameron Tribe
Branden Driver
Brian Andrews
Ahmad Qazi

Faculty Supervisor:
Dr. Marek Perkowski
Industry Sponsor:
Aram Kasparov

June 11, 2015

Portland State University

Contents

1 Project Overview

The team interfaced a CNC machine with a MIG welder to create a 3D metal printer.

2 Project Proposal

2.1 Sponsor Proposal

The company is in the process of constructing an innovative 3D metal printer controlled by CNC (Computer Numerical Control). The project was a combination of two machines:

- CNC mill – (3, 4 or 5 axis CNC mill)
- MIG/TIG welding machine.

The purpose for the CNC motion control (CNC mill) is to program and control motion of the machine, and in this case, the metal deposition process. The purpose for the MIG welder is to deposit liquid metal. Many kinds of wire can be used by the welder to form the parts; carbon steel, titanium, stainless steel or aluminum. The idea for metal deposition and an example that uses a laser can be found at:
https://www.youtube.com/watch?feature=player_embedded&v=s9IdZ2pI5dA

A problem with laser use is its high cost. In this project, the welding machine used cost $400. Another example can be found here:
http://www.wired.co.uk/magazine/archive/2014/08/play/steel-sketch

AKTechnology's plan is to manufacture parts for pump and compressors, and research and develop parts for all sorts of use. The goal is to fabricate low cost and highly usable machines.

The company has CNC PC based CNC mill-motion controller.
https://www.youtube.com/watch?v=Plf3t7o951U&list=UUlGufPQeEKdN1-50F89Ejig
https://www.youtube.com/watch?v=G-jokU7v92E&list=UUlGufPQeEKdN1-50F89Ejig
https://www.youtube.com/watch?v=bPQ5UNiGA4c&list=UUlGufPQeEKdN1-50F89Ejig

The project was to upgrade this CNC motion controller – mill into 3D metal deposition printer by adding a MIG welder instead of a cutting tool spindle. The CNC motion control was reprogrammed. The MIG welder was operational.

The project will also build control to integrate CNC mill and MIG welding machine. The Welder has 2 adjustments - feed of wire and current. A stepper motor is planned to be used to control those analog data for wire feed and power current. The PLC, programmable logic controller, will join the CNC motion controller and the MIG welding machine.

2.2 The Goal

The end goal of this project is to fully integrate the MIG welder with the LinuxCNC system. Integration will include a way to control all of the functions of the welder, i.e. wire speed, maximum current output, engaging and disengaging the welder at appropriate times. In order for this to be done, electromechanical devices must be used to manipulate the knobs on the MIG welder. At the very least, the machine must be able to deposit material, reproducing a simple single object from a CAD drawing. Our aim is to produce a 1" cube. However, it is desired that the machine will be able to create complex structures on a single base. Precision of the deposition is not the primary concern, however it will be a requirement that the total amount of material deposited is more than the minimum tolerance of the part being created. This will allow for material to be machined away to a more precise tolerance.

2.3 Our Starting Point

The groundwork of this project has been completed by Aram Kasparov, the project sponsor. The project at its current state consists of a PC controlled CNC machine, a MIG welder, an infrared temperature sensor and a current measuring sensor. The PC controlling the CNC machine is running a Linux operating system. LinuxCNC an open-source software is used for programing and interfacing with the physical machine. Additional hardware is installed onto the PC, consisting of Mesa Electronics 5I20 FPGA based PCI Anything I/O card, 7i33 analog servo interface card and two 7i37-COM isolated I/O cards. The LinuxCNC software communicates the control signals and receives feedback through these cards. The CNC machine is a 3-axis machine-that is it can move in the X, Y and Z directions. Each axis is moved by a servo-motor and each servo motor is driven by a driver which receives its control commands from the PC. The machine is functional, though the motors will require some tuning and limit switches need to be programmed in (they are physically installed on the machine but not included in the program). The MIG/Flux cored welder is rated at 180 Amp-DC, 240 Volt with a duty cycle of 20% at 140 amps. The welder has current and wire feed adjustment capabilities for controlling the weld. These two knobs will be controlled by two stepper motors which have been installed onto the welder already. The current sensor has the ability to measure up to 225A. It has been demonstrated to be functional and will be used to monitor the current of the weld. The infrared non-contact temperature sensor is rated to measure temperatures up to 1800 degrees Celsius, though no tests have been performed yet.

2.4 Requirements

- Must use a wire feed welder
- Welder must have a Control System
- Must measure weld temperature
- Must measure weld current
- Must use both previous parameters to estimate current quality of weld
- Must use "G code" as inputs
- Must control when material is being deposited
- Must have user interface
- Should allow for welder thermal shutdown
- Should Measure Wire Speed from welder

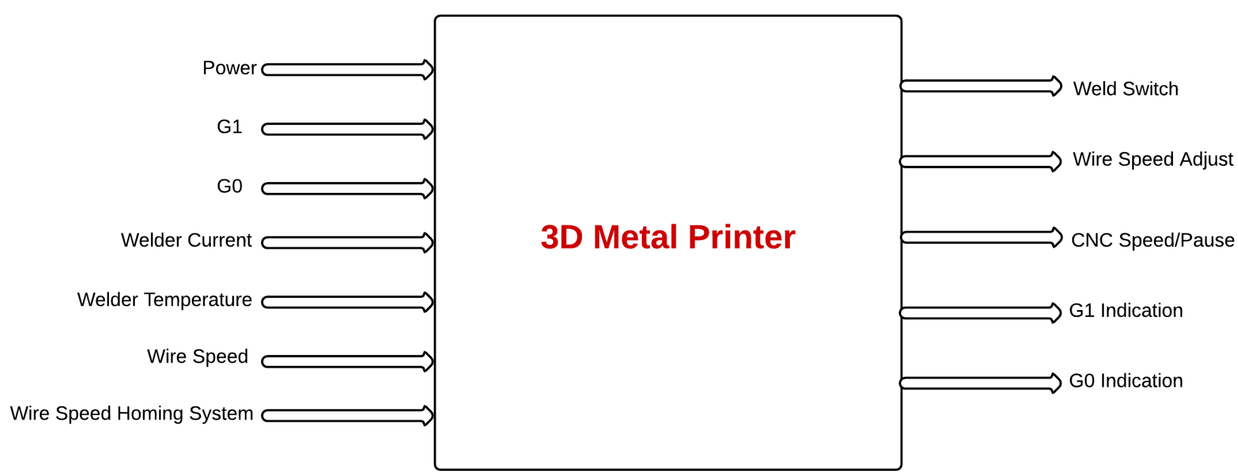


Figure 1: Level-0 Block Diagram of the 3D Metal Printer

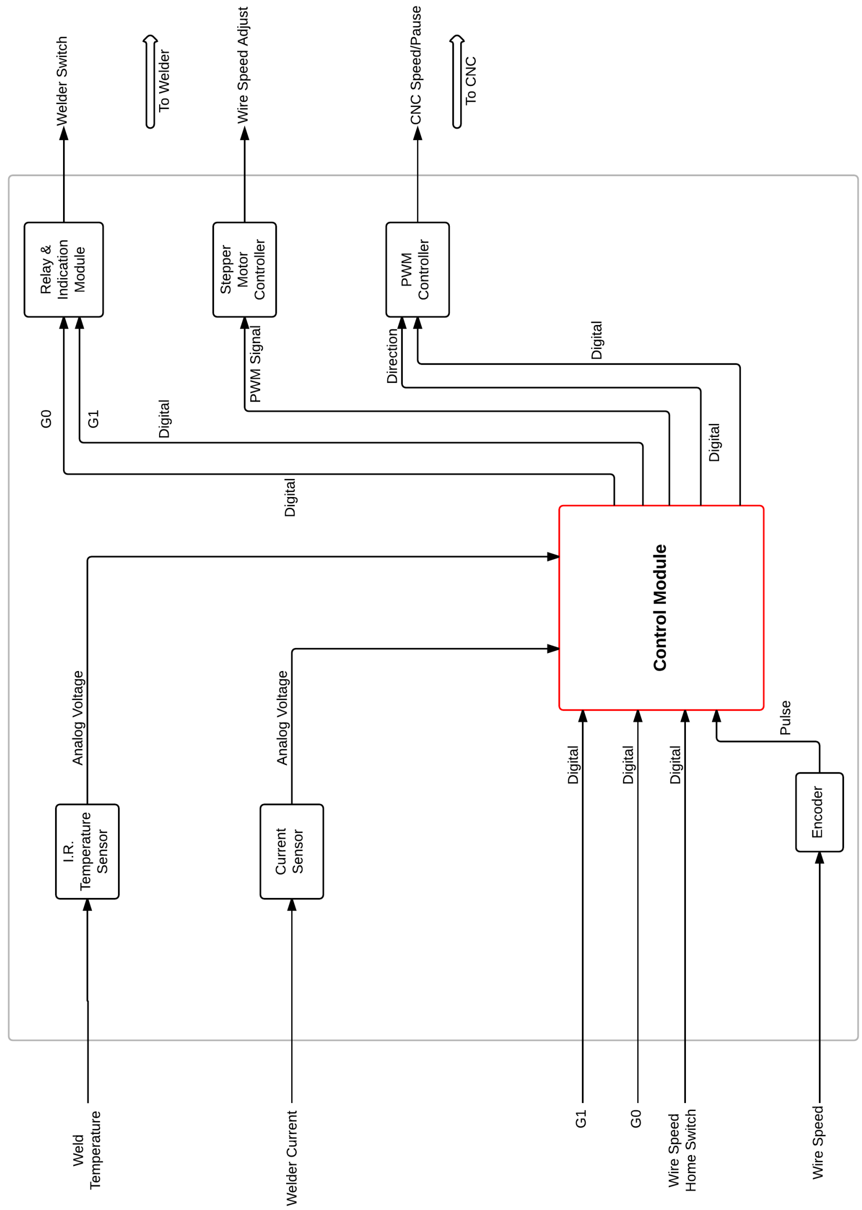
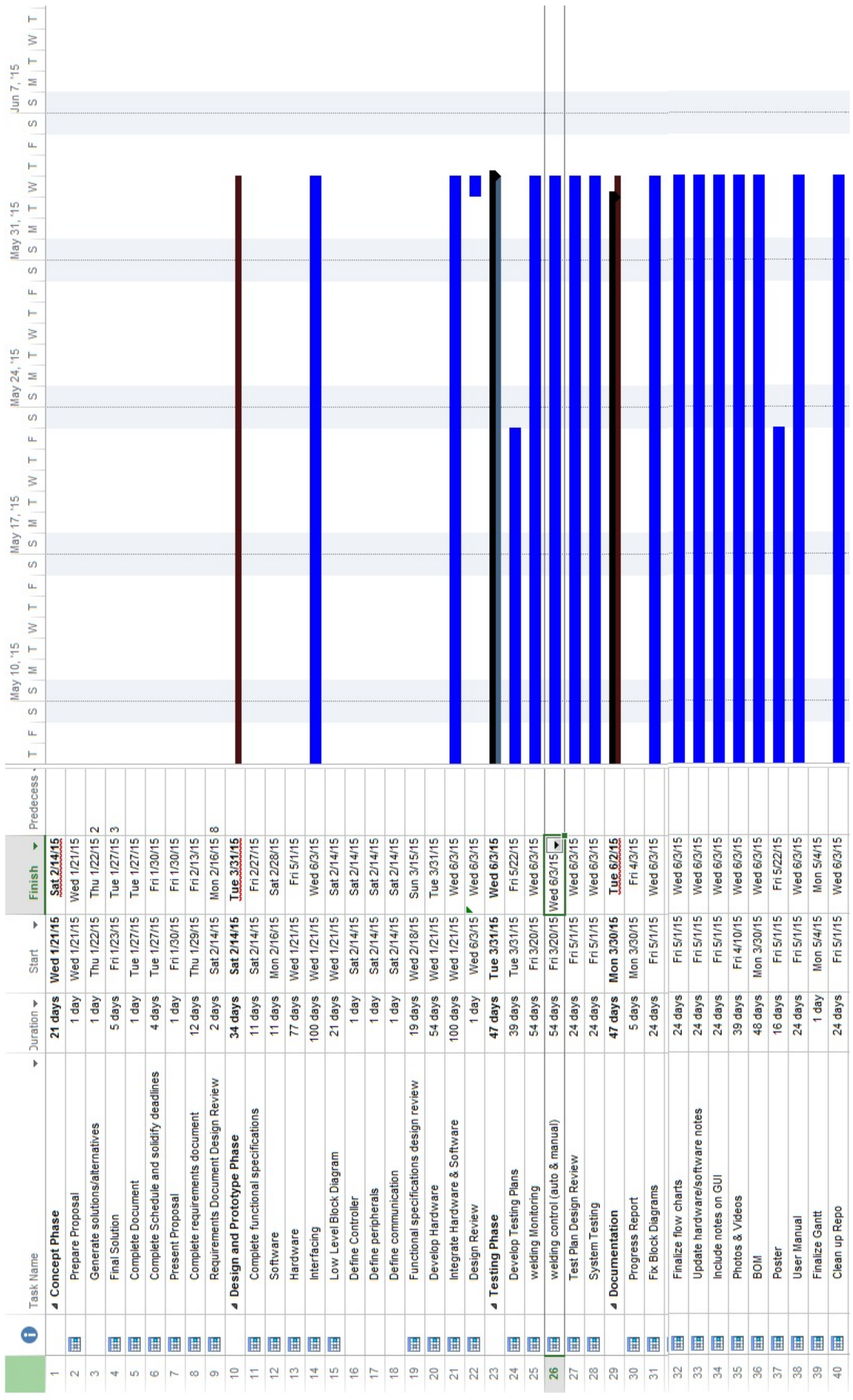


Figure 2: Level-1 Block Diagram of the 3D Metal Printer

3 Schedule



	Task Name	Duration	Start	Finish	Predecessors
1	**Concept Phase**	**21 days**	**Wed 1/21/15**	**Sat 2/14/15**	
2	Prepare Proposal	1 day	Wed 1/21/15	Wed 1/21/15	
3	Generate solutions/alternatives	1 day	Thu 1/22/15	Thu 1/22/15	2
4	Final Solution	5 days	Fri 1/23/15	Tue 1/27/15	3
5	Complete Document	1 day	Tue 1/27/15	Tue 1/27/15	
6	Complete Schedule and solidify deadlines	4 days	Tue 1/27/15	Fri 1/30/15	
7	Present Proposal	1 day	Fri 1/30/15	Fri 1/30/15	
8	Complete requirements document	12 days	Thu 1/29/15	Fri 2/13/15	
9	Requirements Document Design Review	2 days	Sat 2/14/15	Mon 2/16/15	8
10	**Design and Prototype Phase**	**34 days**	**Sat 2/14/15**	**Tue 3/31/15**	
11	Complete functional specifications	11 days	Sat 2/14/15	Fri 2/27/15	
12	Software	11 days	Mon 2/16/15	Sat 2/28/15	
13	Hardware	77 days	Wed 1/21/15	Fri 5/1/15	
14	Interfacing	100 days	Wed 1/21/15	Wed 6/3/15	
15	Low Level Block Diagram	21 days	Wed 1/21/15	Sat 2/14/15	
16	Define Controller	1 day	Sat 2/14/15	Sat 2/14/15	
17	Define peripherals	1 day	Sat 2/14/15	Sat 2/14/15	
18	Define communication	1 day	Sat 2/14/15	Sat 2/14/15	
19	Functional specifications design review	19 days	Wed 2/18/15	Sun 3/15/15	
20	Develop Hardware	54 days	Wed 1/21/15	Tue 3/31/15	
21	Integrate Hardware & Software	100 days	Wed 1/21/15	Wed 6/3/15	
22	Design Review	1 day	Wed 6/3/15	Wed 6/3/15	
23	**Testing Phase**	**47 days**	**Tue 3/31/15**	**Wed 6/3/15**	
24	Develop Testing Plans	39 days	Tue 3/31/15	Fri 5/22/15	
25	welding Monitoring	54 days	Fri 3/20/15	Wed 6/3/15	
26	welding control (auto & manual)	54 days	Fri 3/20/15	Wed 6/3/15	
27	Test Plan Design Review	24 days	Fri 5/1/15	Wed 6/3/15	
28	System Testing	24 days	Fri 5/1/15	Wed 6/3/15	
29	**Documentation**	**47 days**	**Mon 3/30/15**	**Tue 6/2/15**	
30	Progress Report	5 days	Mon 3/30/15	Fri 4/3/15	
31	Fix Block Diagrams	24 days	Fri 5/1/15	Wed 6/3/15	
32	Finalize flow charts	24 days	Fri 5/1/15	Wed 6/3/15	
33	Update hardware/software notes	24 days	Fri 5/1/15	Wed 6/3/15	
34	Include notes on GUI	24 days	Fri 5/1/15	Wed 6/3/15	
35	Photos & Videos	39 days	Fri 4/10/15	Wed 6/3/15	
36	BOM	48 days	Mon 3/30/15	Wed 6/3/15	
37	Poster	16 days	Fri 5/1/15	Fri 5/22/15	
38	User Manual	24 days	Fri 5/1/15	Wed 6/3/15	
39	Finalize Gantt	1 day	Mon 5/4/15	Mon 5/4/15	
40	Clean up Repo	24 days	Fri 5/1/15	Wed 6/3/15	

Figure 3: Gantt Chart showing the schedule

4 Hardware

4.1 Welder Control

To control the welder, a central control module will be used. This was a hot topic of debate for several weeks, as the number of choices available for this project are very high. The sponsor's requirements for the project was that all of the control work was done by a separate computer from the one used by Linux CNC, this only narrowed it down to a choice between a PCI-Express DAQ board and a single board computer. Based on the need for both analog and digital control pins and the need for future expansion, we researched and came up with several options.

Single Board Computers	**DAC**
Raspberry-Pi	Sensoray 826
Intel Galileo	MCC DAS1602/16
SBC 8600B	
Wander Board Solo	
Beagle Bone Black	

In the end we chose the Sensoray 826 board because for the price it outperforms all other boards on the market by having 16 analog inputs, 8 analog outputs, and 48 digital I/O pins. This board was chosen for the high level of future expandability that it has, and because it is a PCI-Express card which can be packaged into its own desktop as per request from the sponsor.
To control the current to the weld and the wire speed of the welder, two stepper motors have been fitted to the manual control knobs, and are connected to a motor driver module. The Sensoray board will be controlling the motor drivers using a sequence of rising and falling edges. To allow the controller board to control at what time the welder is depositing and when it is not depositing, a relay with a transistor driver will be used.

The signal that tells the controller will be coming from the CNC machine's I/O card. It is a switch type signal which means that when the signal is sent, an internal switch will be closed, causing what ever is on the input to be shown on the output. The CNC machine uses G-Code (described below), and Linux CNC allows outputs to be asserted when a particular G-Code instruction is executed. G1 and G0 are going to be used to tell the I/O card to close and open the switch, which will assert 5V DC to the input to the control module. The control module will assert an output high or low which will open or close the welder switch, turning the welder on and off.

The Sensoray 826 I/O card has three 50 pin connectors and two 26 pin connectors. To allow easy access to these pins, a breakout board with screw terminals has been made so that wires can easily be disconnected and switched.

4.2 Breakout Boards

To easily connect to the Sensoray 826 board, several breakout boards with screw terminals were made. There are two types of break out boards, a 50-pin board and a 26-pin board. 50 pin boards are used to connect to the digital and analog pins of the control board, while 26 pin boards are used to connect to the various counter channels. Shown below are images of the boards them selves. With the empty side of the board facing you, the screw terminals are in order from 1 to 50/26 starting on the right hand side.

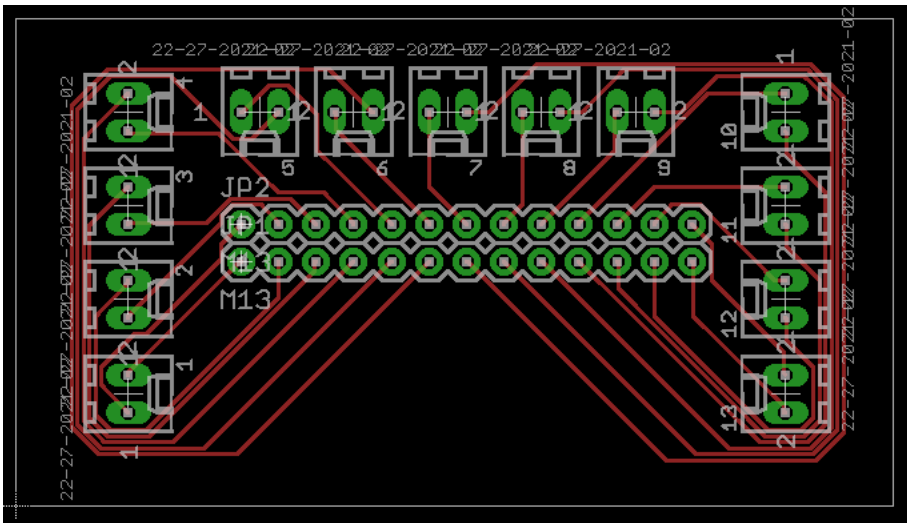


Figure 4: Board Layout of the 26 pin Breakout Board for the Sensoray 826 I/O card

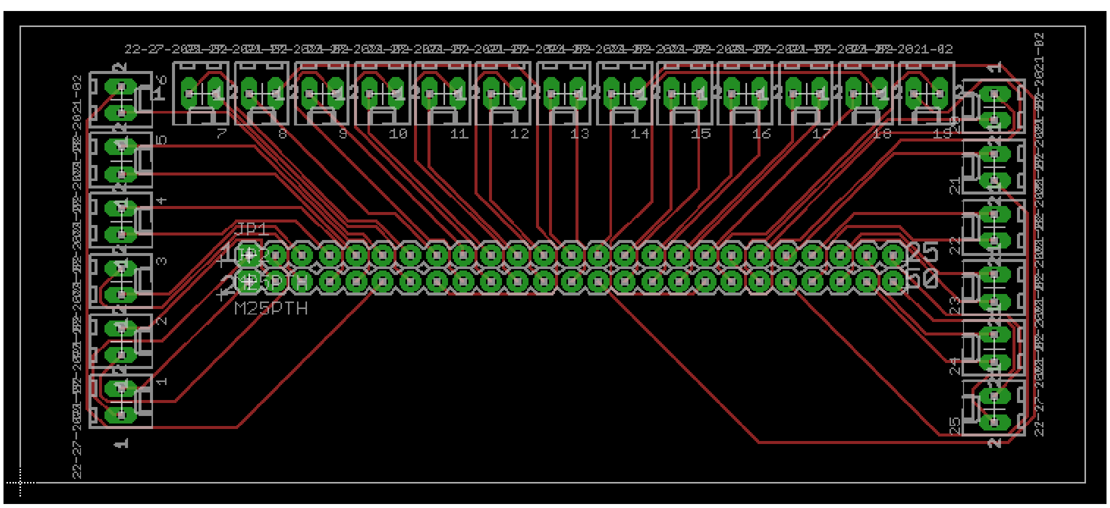


Figure 5: Board Layout of the 50 pin Breakout Board for the Sensoray 826 I/O card

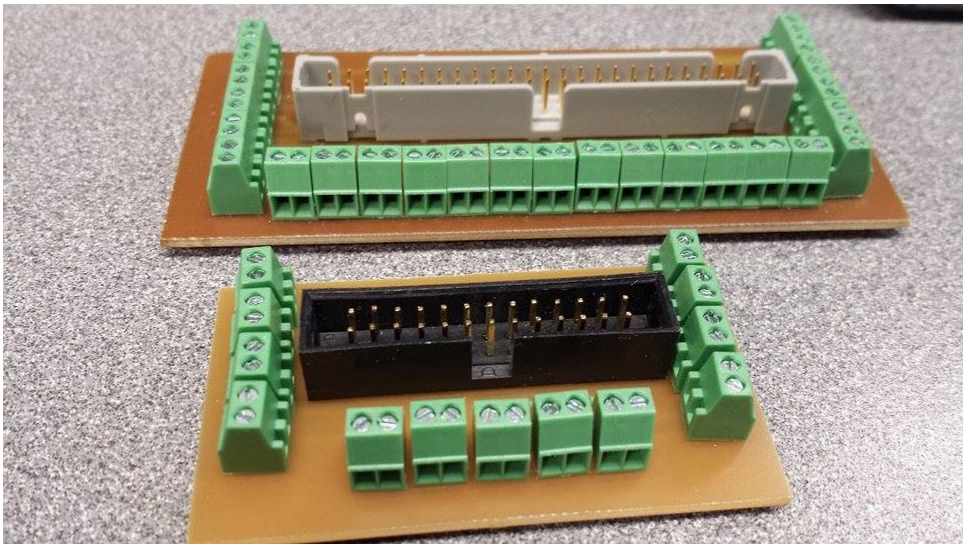


Figure 6: The 26 and 50 pin Breakout Boards with connectors for The Sensoray 826 I/O card

4.3 Temperature Sensor

An infrared temperature sensor will be used to measure the temperature surrounding the weld area. These temperature sensors typically have a higher operating range than other types of temperature sensors. The chosen sensor is the CTLM-1M-1H1-CTL-CF4 from Micro-Epsilon. This sensor was chosen of it's operating range was 800C to 2200C. It has multiple configurable output types, including current output, voltage output, and alarm outputs. This sensor also has a focus point at 450mm ($\tilde{1}8$ in) which gives considerable distance from the weld, and it is not impractically far away. Further documentation can be found (Appendix temp sens). The Sponsor has agreed to take care of mounting the sensor on the machine.

The chosen output type of the temperature sensor is chosen to be a voltage output with a full-scale range of 0V to 10V. This range was chosen because the ADC on the Sensoray 826 are configurable to accept up to +10V and another Analog input needed the input configured to +10V. Setting the output range to 0V-10V on the current sensor removed the need to write additional software to solve an issue that was solved in other means.

Figure 7: The Micro-Epsilon 1MH1-CF4 Temperature Sensor

4.4 Incremental Encoder

The Incremental Encoder is used to measure the actual wire speed of the welder wire. The encoder needed to be incremental because of the need to know speed, and that current position did not matter. The chosen encoder is the U.S. Digital S5-5000-250-IE-D-B. Initially a pulley on a shaft type encoder was going to be used. The pulley be mounted to the frame of the welder and the encoder would be placed under tension underneath the wire that is being fed to the weld. However this was not successful as there could not be any external tension placed on the wire inside the welder.

Fortunately the main drive pulley has a square drive shaft and stuck out past the edge of the pulley. A small plastic "coupler" piece was 3D printed that adapted the round end of the encoder, to the square drive shaft of the drive pulley. The CAD drawing of this part can be seen to the left. This piece allowed an accurate measurement of the actual wire speed of the welder to be interpreted by the Sensoray 826. Here, the calculations for the wire speed can be found.

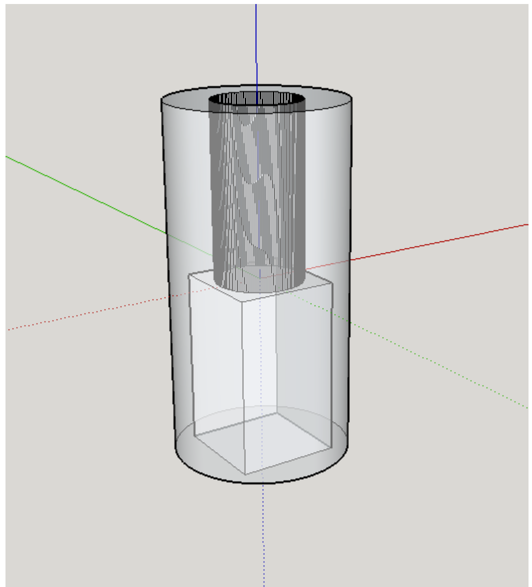


Figure 8: Schematic of the Incremental Encoder

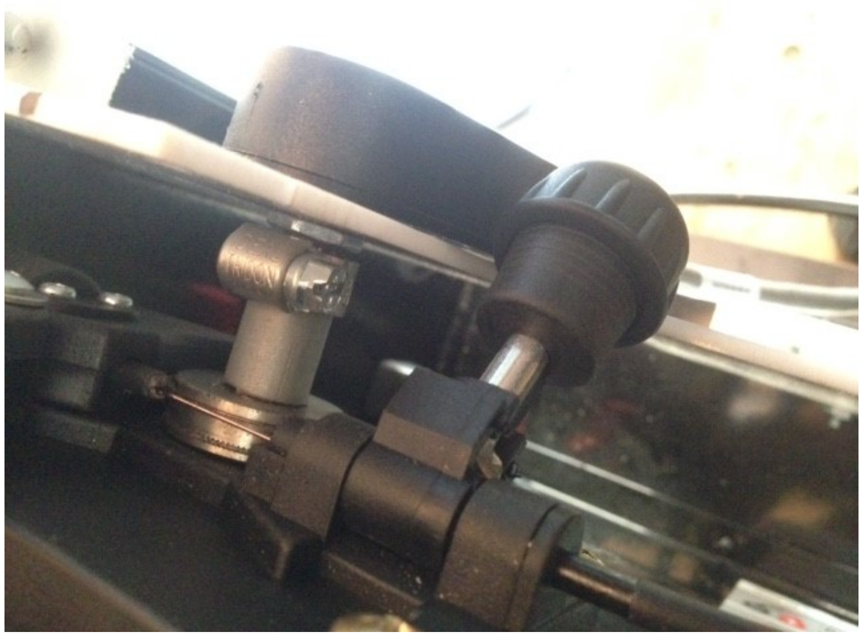


Figure 9: The US Digital S5-5000-250-IE-D-B Incremental Encoder

4.5 Relay and Indication Module

A relay was used to interface the Sensoray 826 with the welder. A relay provides isolation from any harmful voltage spike that occurs on the welder's circuit. It also acts as a mechanical switch, which is the same as the trigger on the welder gun itself. It is unknown what type of signal is passed through the welder switch weather its DC, or AC the relay will on act differently, as a transistor might. The interface circuit includes a transistor drive circuit that switches an 8V supply (which comes from a wall wart power supply) across the coil of the relay on and off. This module also includes LEDs to indicate what state the CNC machine is currently in. A schematic and an image of the final module can be seen below.

There were some issues with the relay module, where were that the Sensoray 826 has internal 10k pull up resistors, and when the machine powers up, the active low DIO pins are initialized to a 0-state, which means that the voltage is high on the pin. When connected to the welder, this meant that when there was not any code being executed the welder would be activated. To remedy this issue, an inverter was placed on the input.

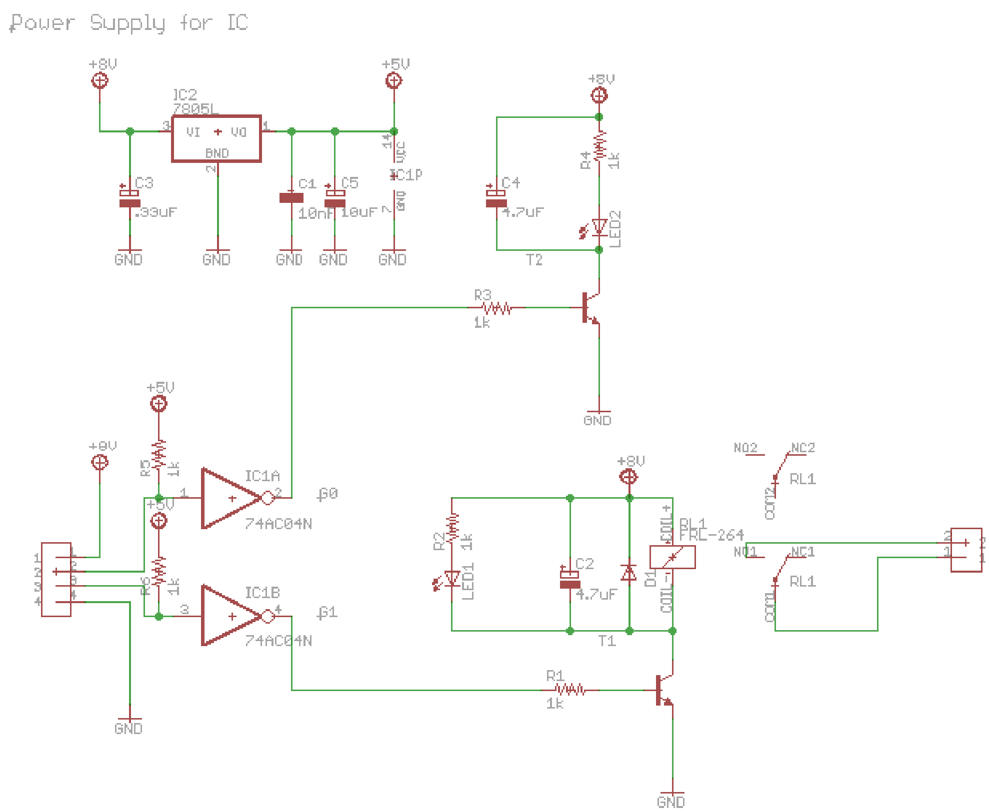


Figure 10: Schematic of the Relay & Indication Module

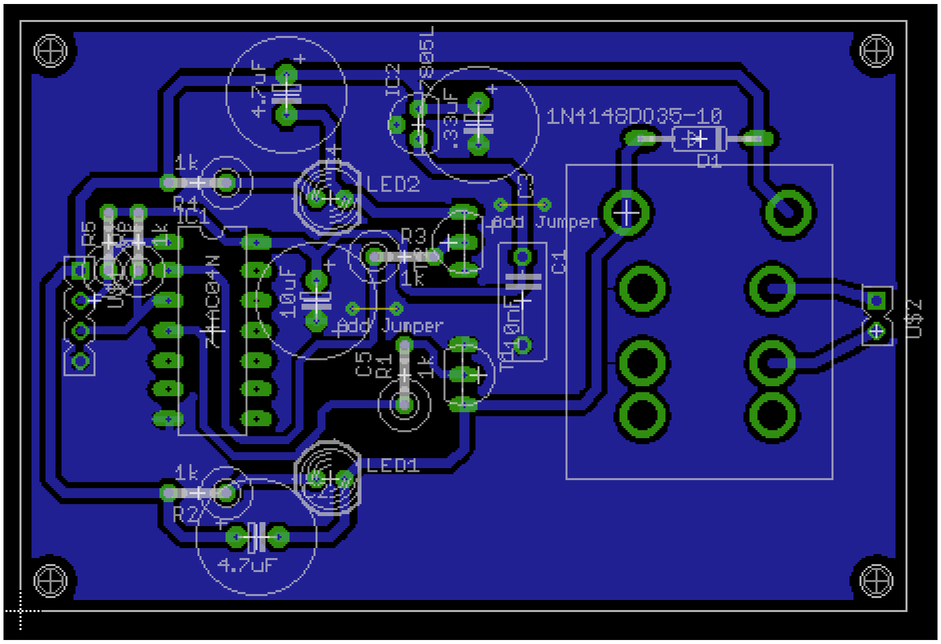


Figure 11: Board Layout of the Relay & Indication Module

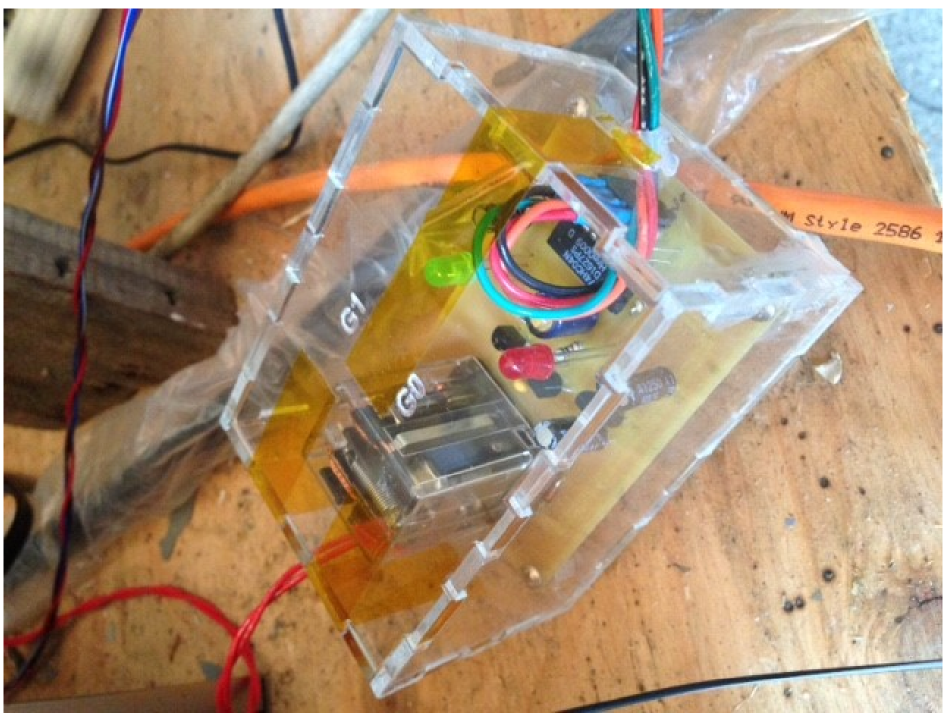


Figure 12: The Relay & Indication Module

4.6 Stepper Motor Controller

P/N:
Controller: KL-5056
Motor: KL23H2100-35-4B

The stepper motor controller is used to control a motor that adjusts the wire speed knob on the welder. From the Sensoray 826, there are two digital signals that control how much and in which direction the motor turns. To tell the motor two turn, a PWM signal with a 50% duty cycle is used. The frequency determines the speed of rotation. The direction signal is either a high or low 5V signal that will turn the motor clockwise or counterclockwise. The controller also has a programmable step resolution, which is either defined in software, or by using the DIP switched located on the side of the controller. Further documentation on the controller can be found here.

The motor is coupled to the wire speed adjustment knob via a PVC cylinder, which have elevated surfaces where the limits of the knob are. Shown below, are switches that get triggered if the motor turns too far. The limit switches are connected to the enable pin of the motor controller, so that if the motor accidentally turns too far, it will disable its self and it will not damage the knob on the welder. An additional switch is added to the PVC cylinder, which is used as a reference position of the knob. Shown below is the wiring diagram of the motor controller.

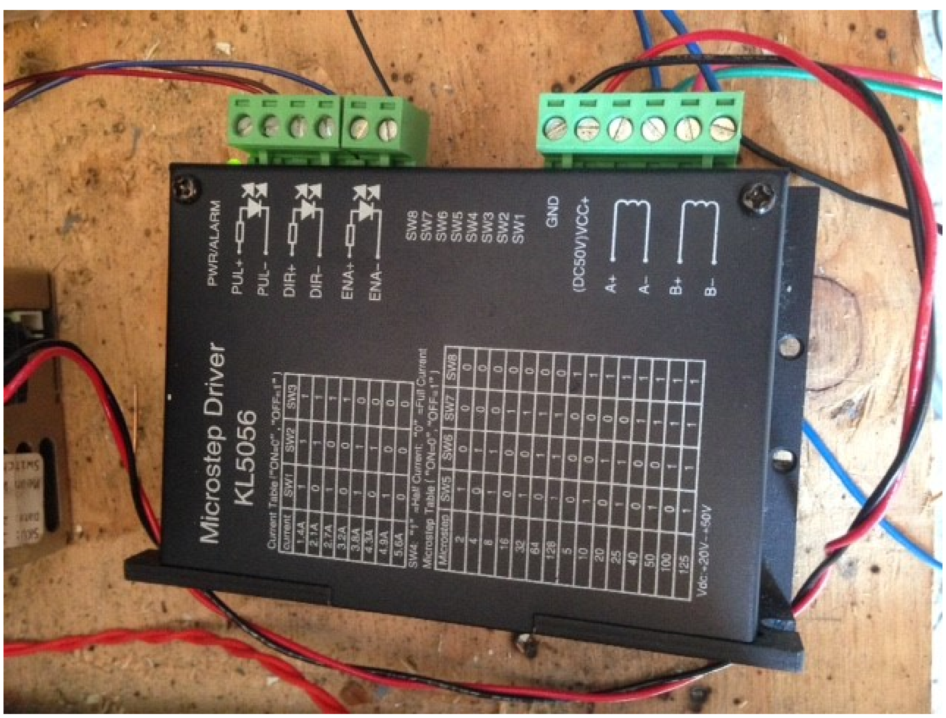


Figure 13: The Stepper Motor Controller

4.7 PWM Controller

P/N: Mesa THC A-D

The PWM controller is a voltage to frequency converter. It is used to externally start and stop the CNC machine. The voltage input of the PWM controller is connected to a digital output of the Sensoray 826. To stop the CNC Machine, a 5V signal is send and to stop it is 0V. The CNC machine is initialized to use this 5V signal in the .HAL file associated with that machine. In fact the voltage on that pin can be anywhere between 0V and 5V for further external control of the welder, however in the scope of this project, only a high or low signal needs to be sent.

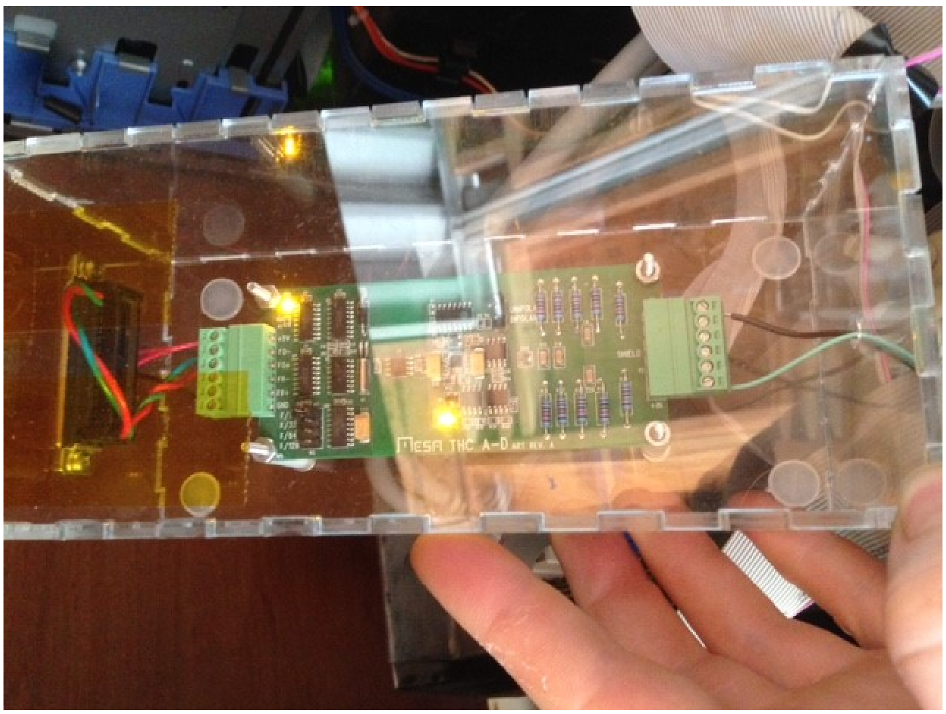


Figure 14: The Mesa THC A-D PWM controller

4.8 Current Sensor

P/N: CSLA1DJ

This current sensor has an operational range of 0 to 225A, which is well over the maximum current of the welder. It is placed in a small plastic housing, which a jumper cable is passed though. The sensor is a Hall effect sensor that is placed inside a ferrite toroid. The output of the sensor is a voltage that sits at half of the supply voltage, and will deviate above or below that level based on he magnitude and direction of the current. There is a metal lug, which the ground connection of the welder connects to. The other end of the module is a large alligator clamp that connects to the plate being welded to.

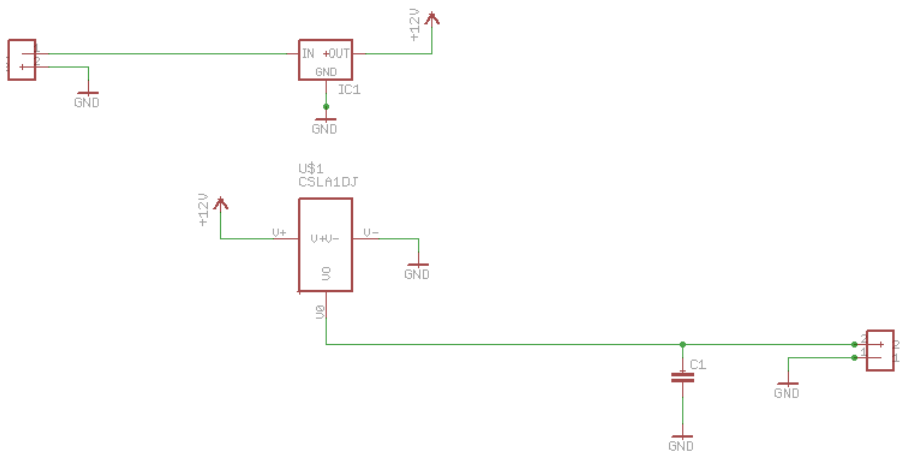


Figure 15: Schematic of the Current Sensor

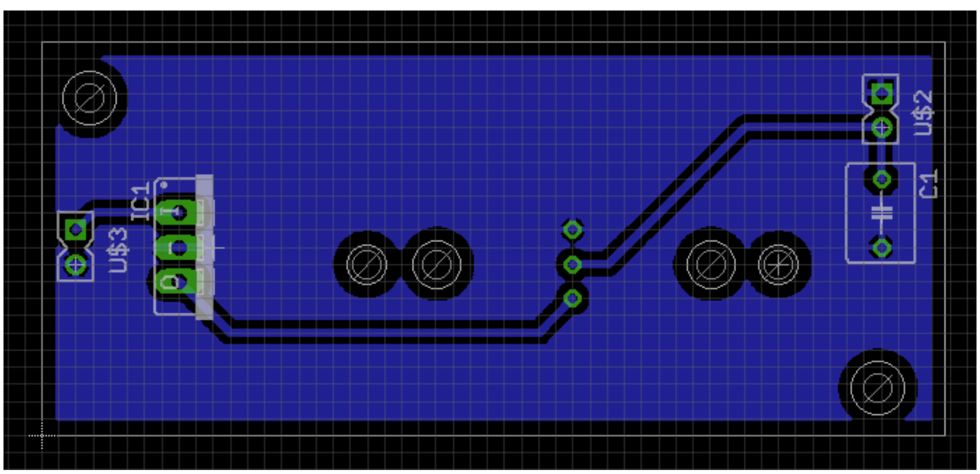


Figure 16: Board Layout of the Current Sensor

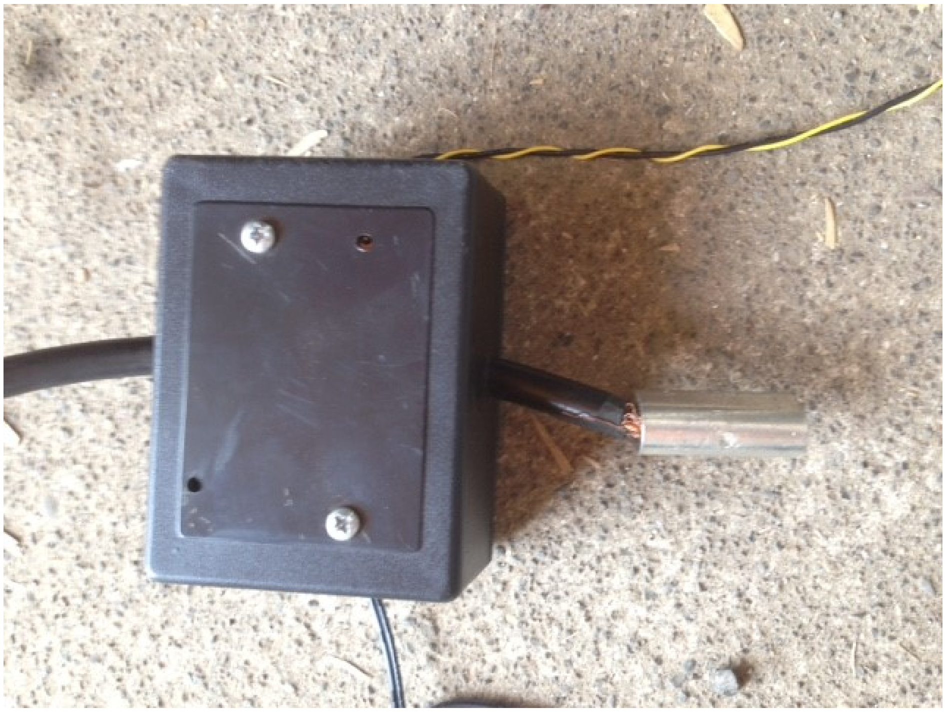
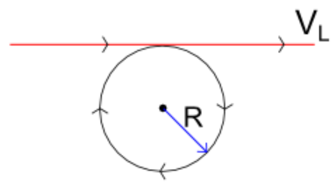


Figure 17: The CSLA1DJ Current Sensor

4.9 Wire Speed

For calculating Wire Speed with Rotary Encoder



So Angular Velocity is,

$$\boldsymbol{f = \frac{n}{Nt}}$$

n = number of up/down counts, N = number of up/down counts/rev, T = sampling time.

And Linear Velocity is,

$$v_L = \omega r$$
$$\omega = 2\pi f$$

So,

$$\boldsymbol{V = \frac{\pi n d}{NT}}$$

Connections to 826

J_4	Pin 1	+A0
	Pin 2	-A0
	Pin 3	GND
	Pin 4	+B0
	Pin 5	-B0

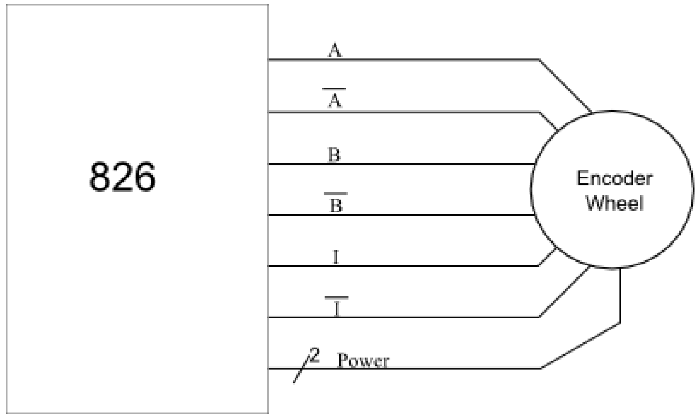


Figure 18: Encoder Connection

4.10 Pin Assignments

DIO PINS	Wire Color	Channel Number	Function
1	Teal	23	CNC Start/Stop
2	Black		GND for CNC Start/Stop
3	Blue with White Stripe	22	Direction of Stepper Motor
4	Black with White Stripe		Return path for Limit Switch
5	Brown	21	Pulse on Motor Controller
6	N/C		GND
7	Purple	20	Wire Speed Home Switch
8	Black		GND for Wire Speed Home Switch
9	Green with White Stripe	19	G1 Input (From CNC Machine)
10	N/C		GND
11	Orange with White Stripe	18	G0 Input (From CNC Machine)
12	N/C		GND
13	Green	17	G1 Output (To Relay and Ind. Module)
14	Black		GND for Relay Module
15	Orange	16	G0 Output (To Relay and Ind. Module)
16	Black		Power Supply GND for Relay Module
17		15	
18			GND
19		14	
20			GND
21		13	
22			GND
23		12	
24			GND
25		11	

DIO PINS	Wire Color	Channel Number	Function
26			GND
27		10	
28			GND
29		9	
30			GND
31		8	
32			GND
33		7	
34			GND
35		6	
36			GND
37		5	
38			GND
39		4	
40			GND
41		3	
42			GND
43		2	
44			GND
45		1	
46			GND
47		0	
48			GND
49			5V to the CNC Switching system and Motor Controller
50			GND

ANALOG PINS	Wire Color	Channel Number	Function
1			
2			
3	Black	0	GND for Current Sensor
4	Yellow	0	Current Sensor Signal
5	Black	1	GND for Temperature Sensor
6	Grey	1	Temperature Sensor Signal
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			

ANALOG PINS	Wire Color	Channel Number	Function
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			

COUNTER PINS	Wire Color	Counter Channel Number	Function
1	Blue with White dots	0	A+ on Encoder
2	White with Blue dots	0	A- on Encoder
3	Green with White dots (note cable shield also connected)	0	GND on Encoder
4	Brown with White dots	0	B+ on Encoder
5	White with Brown dots	0	B- on Encoder
6	White with Green dots	0	POWER on Encoder
7	Orange with White dots	0	I+ on Encoder
8	White with Orange dots	0	I- on Encoder
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			

5 Software

5.1 Software Description

The main program for our control system was written entirely in the programming language "C". This was chosen so as to ensure compatibility with the Sensoray 826 DAQ board used for control. The program begins by asking the user to disconnect the grounding cable of the welder and waiting for an input from the user confirming this task is completed. Upon receiving that input the machine then runs a calibration procedure by running a PWM procedure to turn the stepper motor until the system sees the homing switch is triggered. The program then turns on the welder's gun to feed wire while measuring the wire speed and setting the average of those wire speeds as the new homing offset.

The next step of the program asks the user to input the speed the CNC machine will be running at for the duration of the print in in/sec. It uses this value to find a corresponding nominal wire speed in an array of nominal wire speeds found via testing, and sets the welding machine to that wire speed. The program then waits for a signal from the CNC machine that it has switched from Relocation mode to Deposition mode. Once the system sees that we have reached Deposition mode, it stops the CNC movement and makes sure the welder is not triggered, in order to check the temperature of the base plate being welded to. If the base plate's temperature is too low the system will pause and wait for a torch to heat the base plate up to above a given threshold value.

On completion of the torch routine the system enters the main loop, this begins by reading a timestamp from the 826's onboard timestamp generator, this value in microseconds will be used to keep track of how long the machine has been in deposition mode later. The system then starts the welder and runs a check for spikes in the current seen by the Current Sensor, if it sees none the system will end the program and return an error report, otherwise the program moves on and starts the CNC's movement back up so that we can check if the machine is still in Deposition mode. If the CNC machine is still in Deposition mode at that time it moves on to take measurements of the number of "peaks" seen by the current sensor, the temperature of the weld's base plate, and the incremental encoder used to measure wire speed. Amongst the measurements it checks and updates the mode of movement seen by the CNC in order to avoid issues with our system trying to deposit while in Relocation mode.

At the end of checking all sensor measurements, the system begins comparing the observed value with the pre-programmed threshold values. The first check is to make sure that the current plate temperature is at an acceptable value. If the temperature has fallen or risen too far, the whole system stops, and runs the torch routine before starting the system back at the initial timestamp read. If the temperature is at an acceptable value, the average droplet spacing is checked against a nominal value found through testing. If the error between the two values is greater or less than 20%, the system terminates with an error, asking the user to double check that the entire system is working. The last check is to see if the droplet spacing is greater or less than a 5% tolerance, and if so the system makes an appropriate proportional adjustment to the wire speed before continuing on.

5.2 G-Code

G-code is the commonly used name to refer to a numerical programming language. As is the case with all languages, G-code has its own syntax and semantics. A line of code is referred to as a block, and a program is defined as multiple blocks. All programs start and end with the percent symbol (%). While writing G-code, the backslash (/) can be used to comment out an entire line, whereas if you want to make comments about a block, parentheses are to be used. Anything inside of a set of parentheses will be ignored by the compiler. As is the case with most other languages, G-code ignores white space, so spacing is used to clarify the code for the writer as well as future users.

All points of G code are comprised of words and numbers. A word is simply a letter. For example, the block "X0" is simply the word X and the value 0. The words X, Y, and Z refer to the three axes, while the G words refer to the movement, motion, and location. When first started up, any blocks of code will use the point (0,0,0) as home, however G54 establishes a new temporary "home" point and G52 establishes the point where the temporary reference point is to be set. In other words, the block of code "G54 G52 X100 Y100 Z0" changes the reference point from (0,0,0) to (100,100,0) and the remainder of the code will run from this point, unless a new reference point is defined.

Similarly, other G-code words can be used to define how the space between two points should be interpolated. In some instances you may want two points to be connected via a straight line, while at other times it would be better to have them connected via a circular pattern. When dealing with circular interpolation, you can set it to be done in either a clockwise or counterclockwise manner. Beyond just linear and circular interpolation, there are dozens of G-code words that determine how the code is to be run (Appendix C).

5.3 Control Program

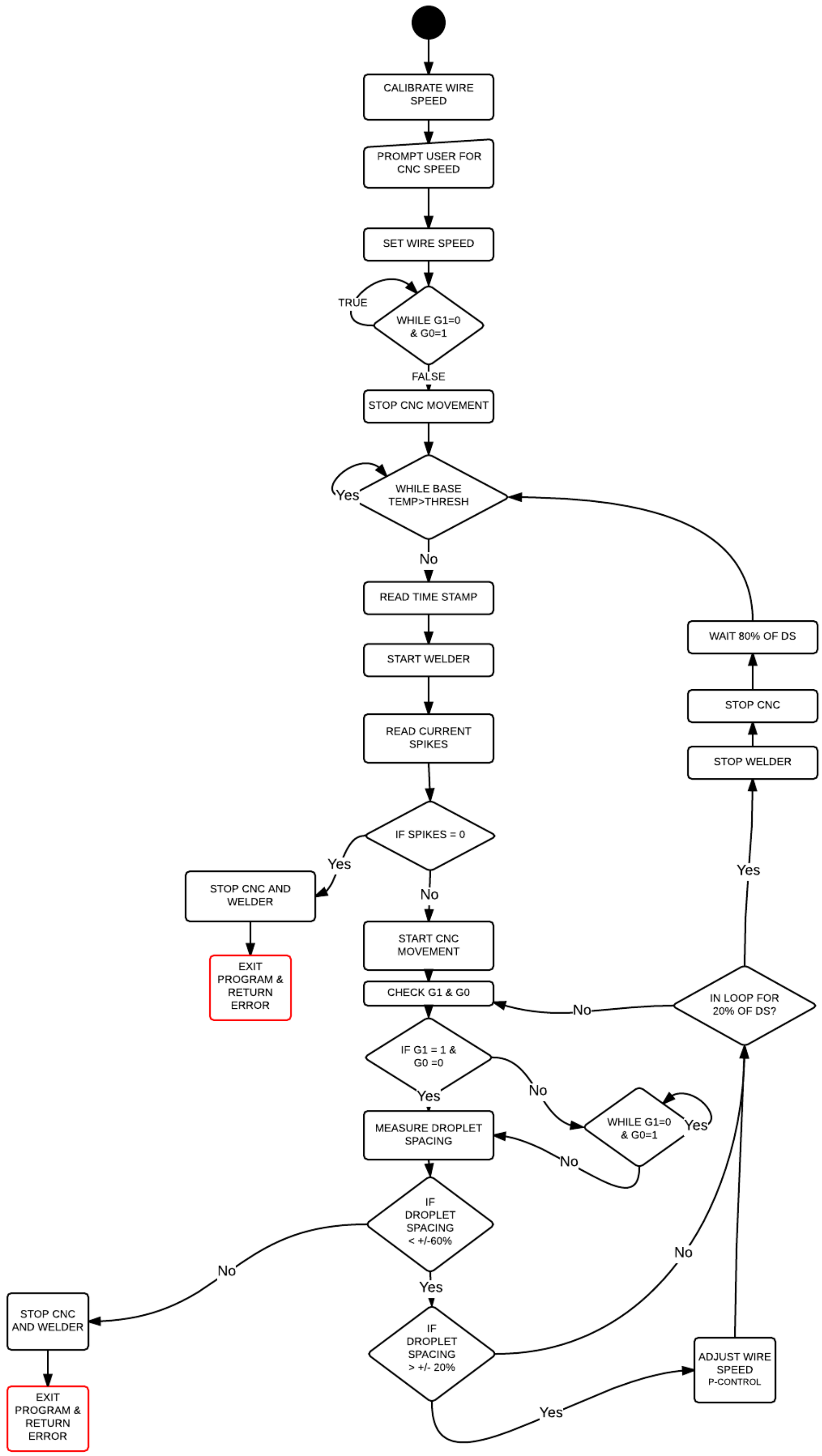


Figure 19: Control Program Flowchart

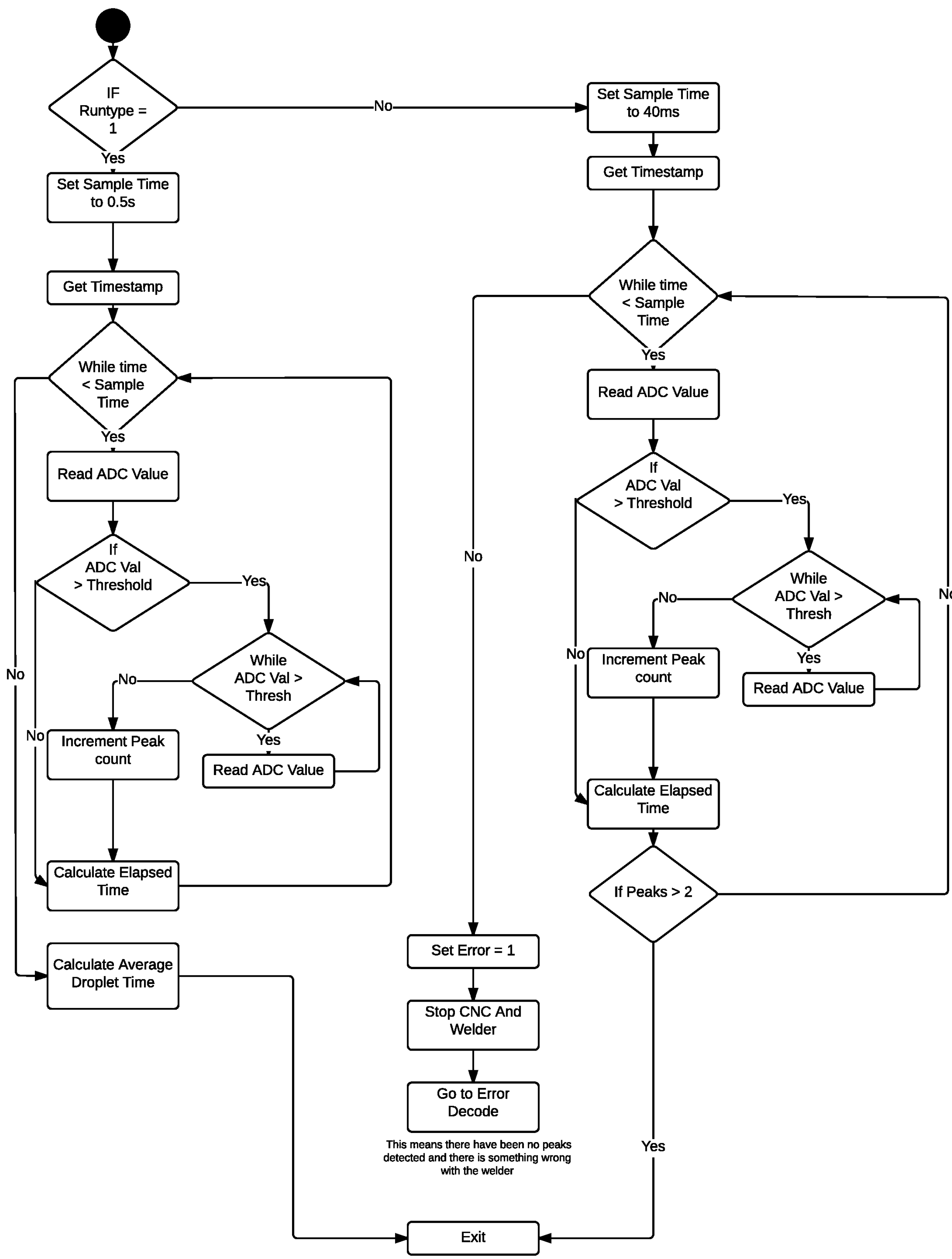


Figure 20: Droplet Spacing Flowchart

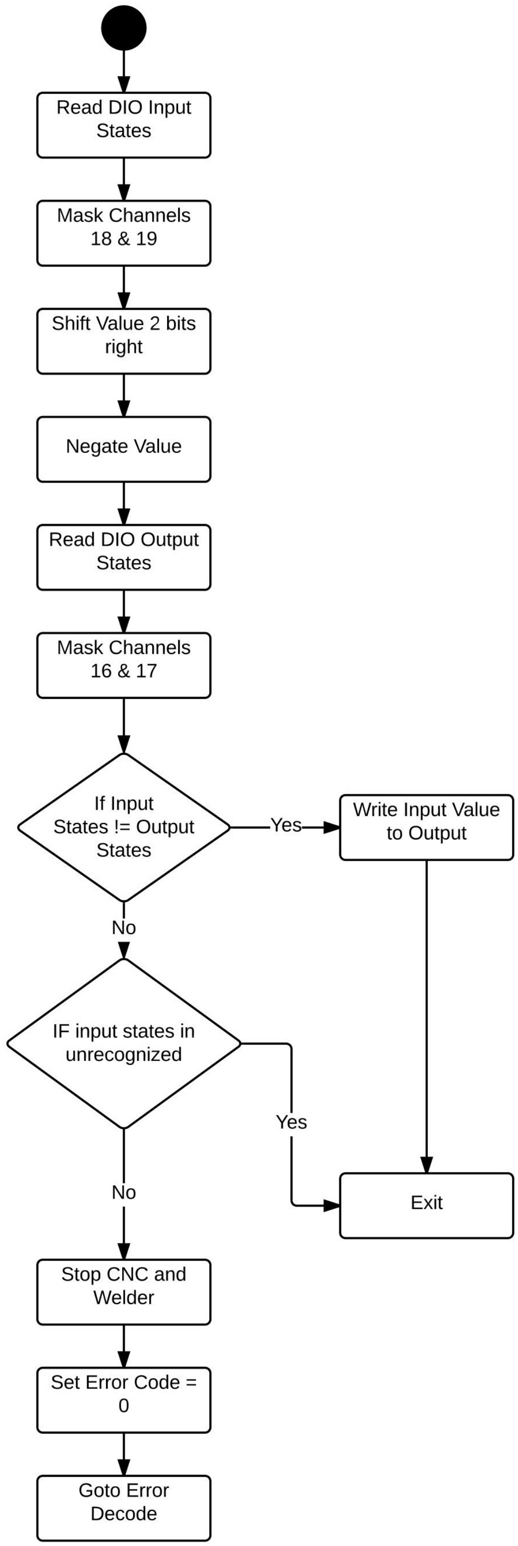


Figure 21: Getting G0 & G1 Flowchart

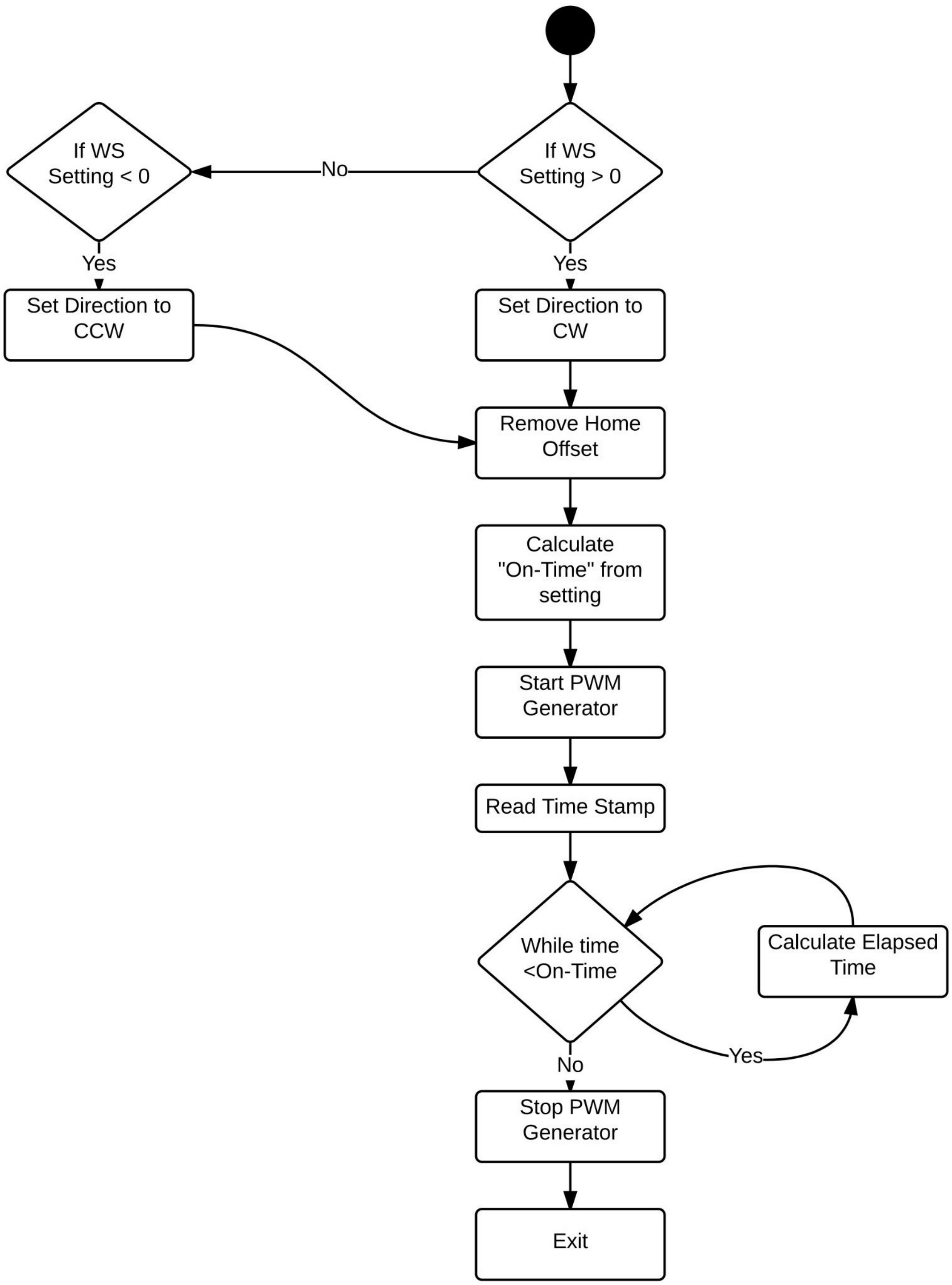


Figure 22: Set Wire Speed Flowchart

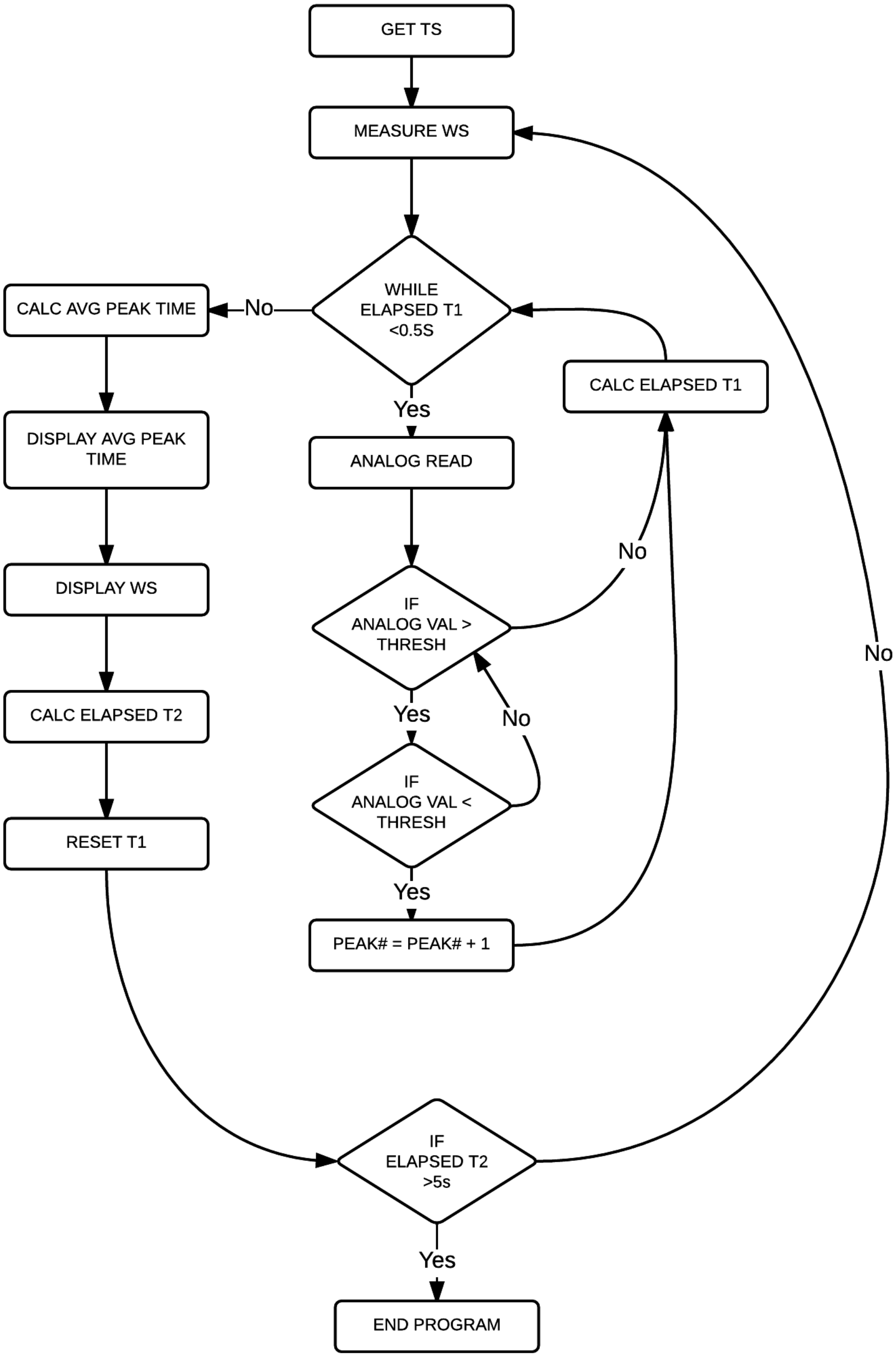


Figure 23: Wire Feed and Droplet Spacing Test Flowchart

5.4 Register Descriptions

The following list shows the definitions for the rows in Table 1:

- Row 3 - BIN VAL For PWM Mode
- Row 4 - HEX VAL
- Row 5 - BIN VAL For Frequency Measurement Mode
- Row 6 - HEX VAL

The following list shows the definitions for the rows in Table 2:

- Row 1 - Bit
- Row 2 - Channel
- Row 3 - Pin
- Row 4 - G1/G0 Read Mask
- Row 5 - G1/G0 Write Mask
- Row 6 - G1 INPUT ACTIVE
- Row 7 - G0 INPUT ACTIVE
- Row 8 - Motor CCW
- Row 9 - Motor CW
- Row 10 - Wire Speed Home Switch
- Row 11 - Weld and CNC stop
- Row 12 - CNC start

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	IP	IM		0	0	0	TP	NR	UD	BP	OM			OP	TP						TE		TD		K			XS			
0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0		0	1	0	0	0	0	0
0				1				6				8				2				0				2				0			

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
0				0				0				0				8				0				0				A			

Table 1: Counter Mode Register Description

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47
								1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47
								×	×	×	×	×	×	1	1	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
								×	×	×	×	0	1	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
								×	×	×	×	1	0	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
								×	1	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
								×	0	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
								×	×	0	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
								1	×	×	×	×	×	1	1	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
								×	×	×	×	1	0	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×

Table 2: DIO[0] Register Description

5.5 GUI Description

The initial plan was to have a GUI for our software and but was later not pursued due to time constraints. In GUI the user could see real-time graphed data coming from the current and temperature sensors as well as see the current wire speed. The system would use this feedback to automatically adjust the weld and keep it in a state that can be considered a "good weld". The plan also included manual overrides for the user to adjust the current and wire speed to their own desired result. Fig. 10 shows an initial GUI layout that we were aiming for.

5.6 The Graphical User Interface (GUI)

We were planning on using GTK+ in C programming language to generate the Graphical User Interface in order to view the different data and also control different settings.

The GTK+ is a library for creating graphical user interfaces. The library is created in C programming language. The GTK+ library is also called the GIMP toolkit. Originally, the library was created while developing the GIMP image manipulation program. Since then, the GTK+ became one of the most popular toolkits under Linux and BSD Unix. Today, most of the GUI software in the open source world is created in Qt or in GTK+. The GTK+ is an object oriented application programming interface. The object oriented system is created with the Glib object system, which is a base for the GTK+ library. The GObject also enables to create language bindings for various other programming languages. Language bindings exist for C++, Python, Perl, Java, C#, and other programming languages.

The GTK+ itself depends on the following libraries.

- Glib
- Pango
- ATK
- GDK
- GdkPixbuf
- Cairo

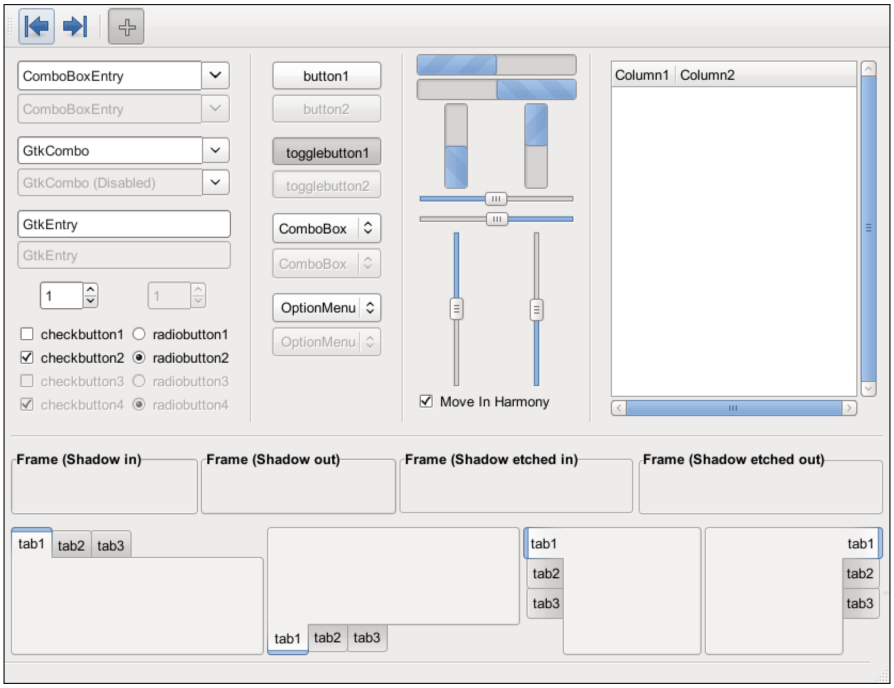
*Note: For detailed functions of each library refer to Appendix A

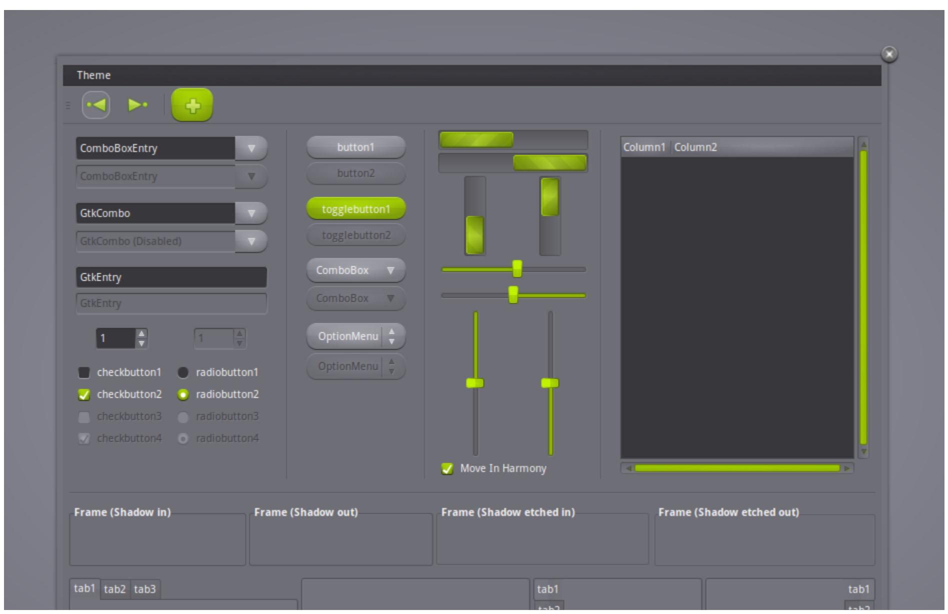
5.7 Reasons for using GTK+

- Language Bindings
  GTK+ is available in many other programming languages thanks to the language bindings available. This makes GTK+ quite an attractive toolkit for application development.

- Interfaces
  GTK+ has a comprehensive collection of core widgets and interfaces for use in your application.
  - Windows (normal window or dialog, about and assistant dialogs)
  - Displays (label, image, progress bar, status bar)
  - Buttons and toggles (check buttons, radio buttons, toggle buttons and link buttons)
  - Numerical (horizontal or vertical scales and spin buttons) and text data entry (with or without completion)
  - Multi-line text editor
  - Tree, list and icon grid viewer (with customizable renderers and model/view separation)
  - Combo box (with or without an entry)
  - Menus (with images, radio buttons and check items)
  - Toolbars (with radio buttons, toggle buttons and menu buttons)
  - GtkBuilder (creates your user interface from XML)
  - Selectors (color selection, file chooser, font selection)
  - Layouts (tabulated widget, table widget, expander widget, frames, separators and more)
  - Status icon (notification area on Linux, tray icon on Windows)
  - Printing widgets
  - Recently used documents (menu, dialog and manager)

5.8 Examples of GUIs created using GTK+





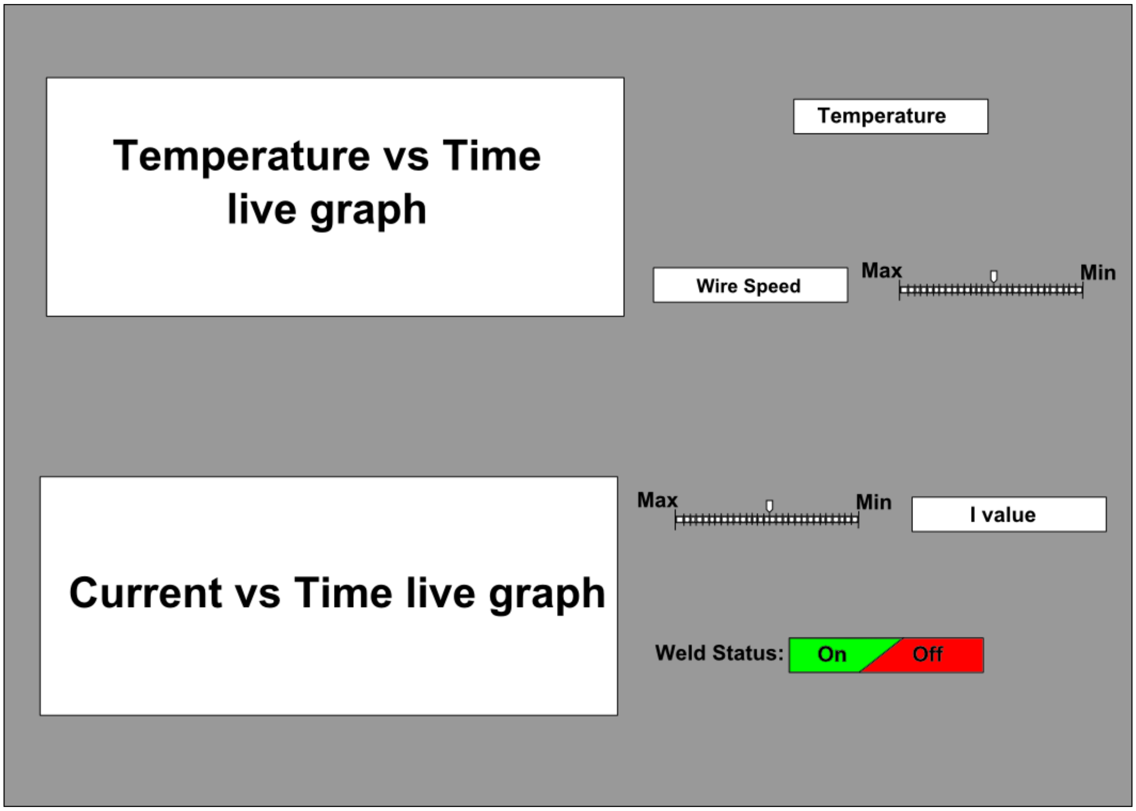


Figure 24: Proposed Rough GUI Layout

6 Testing

Testing was a constant part of this project. Many of the tests were small trouble shooting sessions where we would work on getting a specific piece of hardware or software to work. There were also several tests that were done more methodically, due to their importance of the final outcome. Some of these tests, with their test plans and results, can be seen below.

6.1 CNC Confirmation Test

Test Case # 1: CNC Confirmation Test

Test Writers: Cameron Tribe						
Test Case Name:		CNC Confirmation Test #1			**Test ID #:**	3MP-CNC-01
Description:		This test is to simply confirm that the CNC is operational in all three dimensions.			**Type:**	
Tester Information						
	Name of Tester:				**Date:**	05/30/2015
Hardware Ver:					**Time:**	
Setup:		Only Welder will be used, not CNC. Remove Cover of welder to ensure wire feed is properly set up. Welding will not be taking place, so be sure ground wire is not connected.				

Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Load G-Code into Linux CNC and run initial homing procedure					
2	Place paper on base					
3	Fix felt marker onto CNC machine at an appropriate height so that it will draw on the paper.					
4	Run CNC Machine	This test verified that the CNC machine was working properly. Fig 14 shows the reproduced image that was created by the CNC Machine.				
Overall Test Result:						

6.2 Wire Feed Test

Test Case # 4: Wire Speed Test

Test Writers: Branden Driver			
Test Case Name:	Wire Speed Encoder Test #1	**Test ID #:**	3MP-Wire-01
Description:	Measures the actual wire speed in correlation with the wire speed encoder. This test will allow for the construction of the lookup table to be used with the main program.	**Type:**	
Tester Information			
Name of Tester:		**Date:**	
Hardware Ver:	Display V1.0, Main Board V1.5, Sensor V1.0	**Time:**	
Setup:	Only Welder will be used, not CNC. Remove Cover of welder to ensure wire feed is properly set up. Welding will not be taking place, so be sure ground wire is not connected.		

Step	Action	Expected Result	Pass	Fail	N/A	Comments
1	Open wire feed program	Program should open				
2	Set program to run for one second	program should only run for one second				
3	Cut wire extruded from nozzle as short as possible	Very little wire should be showing				
4	Set wire feed speed nozzle to home setting	Wire feed should be at lowest setting				
5	Turn on welder	Welder should turn on				
6	Run test program	Welder should feed wire for exactly one second				
7	Turn off welder	Welder should turn off				
8	Cut wire extruded from nozzle as short as possible	Very little wire should be showing				
9	Measure length of cut wire and record value	Value determined for specified wire feed setting				
10	Repeat steps 1-9 for various values of wire speed	Determine if wire feed is linear. If so, interpolate for all wire feed speeds				
Overall Test Result:						

Wire Speed (in/s)	Time (sec)	Expected Length (in)	Measured Length (in)
0.88	3	2.64	2.56
0.90	3	2.70	2.46
2.00	3	6.00	6.06
3.20	3	9.60	10.10
5.50	3	16.50	16.80
7.60	3	22.80	23.20
9.50	3	28.50	28.06

Table 3: Linear Velocity Program Test

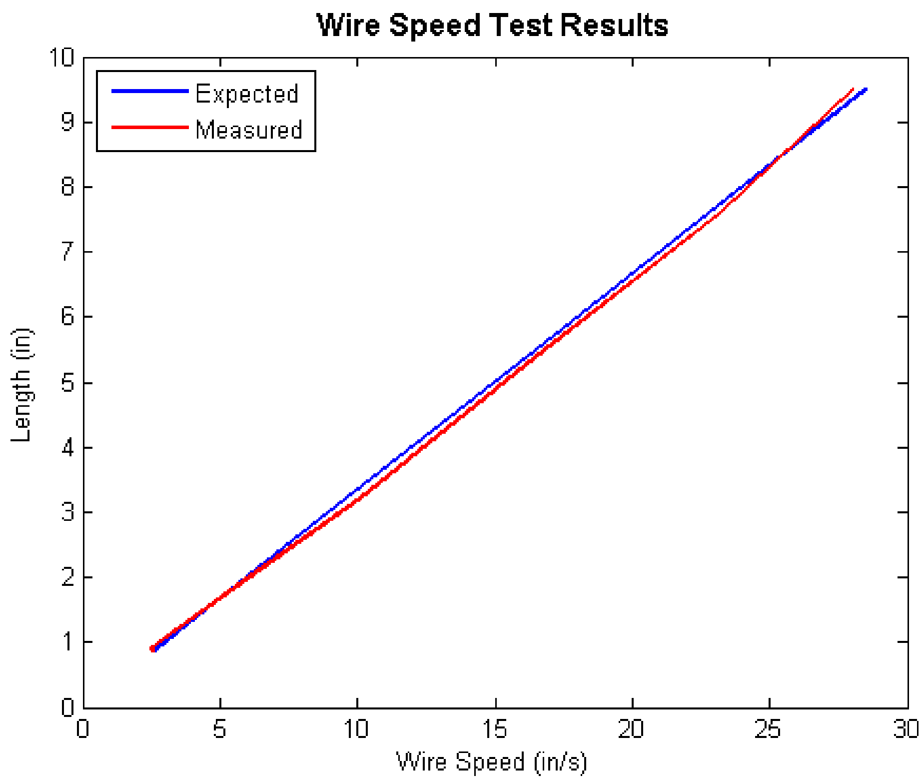


Figure 25: Wire Speed Test Results

6.3 Weld Quality Test

Test Case # 5: Weld Quality Test

Test Writers: Branden Driver						
Test Case Name:		Weld Quality Test #1			**Test ID #:**	3MP-Weld-01
Description:		This test will check the quality of the weld across a variety of currents, wire speeds, and CNC movement speeds			**Type:**	
Tester Information						
	Name of Tester:				**Date:**	
Hardware Ver:		Display V1.0, Main Board V1.5, Sensor V1.0			**Time:**	
Setup:		Ensure both welder and CNC are ready to use. Have at least one 1/8" baseplate on hand for each version of test you wish to run. In LinuxCNC, open file "m100". This file is a G-code program that will print seven 1-inch lines, each at a different CNC movement speed.				

Step	Action	Expected Result	Pass	Fail	N/A	Comments
1	Set current level	Current level selected				
2	Set wire speed around 2	Wire speed selected				
3	Run program m100	Printing will begin				
4	During weld, run droplet spacing program	Program will output wire speed from encoder and average droplet spacing				
5	Quickly record both wire speed and average droplet spacing	Data acquired				
6	Repeat steps 4-5 for each of the seven welds of the program	First run is complete				
7	Adjust wire speed to 4	Welder ready for next run				
8	Run program m200	Print will continue on same plate, next to previous run				
9	Repeat steps 4-6	Data acquired for all 7 welds, 2nd run is complete				
10	Adjust wire speed to 6	Welder ready for next run				
11	Run program m300	Print will continue on same plate, next to previous run				
12	Repeat steps 4-6	Data acquired for all 7 welds, 3rd run is complete				
13	Adjust wire speed to 8	Welder ready for next run				
13	Run program m400	Print will continue on				

		same plate, next to previous run				
14	Repeat steps 4-6	Data acquired for all 7 welds, 4th run complete				
	Adjust wire speed to 8	Welder ready for next run				
	Run program m400	Print will continue on same plate, next to previous run				
	Run program m400	Data acquired for all 7 welds, 5th run complete				
	Overall Test Result:					

Wire Speed (in/s)	CNC Speed (in/min)	Avg Droplet Spacing (μs)
4.21	2	19047
4.04	3	20619
3.88	4	22857
3.98	5	21978
4.14	6	20833
3.88	7	20943
3.95	8	18872
6.02	2	13119
5.81	3	15037
6.19	4	14084
6.06	5	12539
5.71	6	15504
5.6	7	10257
6.54	8	17621
7.5	2	12820
7.2	3	14035
7.61	4	14760
7.67	5	15267
7.61	6	17167
7.45	7	11594
6.95	8	10790
8.99	2	11364
8.82	3	13114
9.58	4	11940
10	5	11765
8.5	6	10106
9.47	7	12751
9.84	8	n/a

Wire Speed (in/s)	CNC Speed (in/min)	Avg Droplet Spacing (μs)
11	2	10582
10.86	3	9557
10.65	4	10811
10.03	5	12270
11.05	6	11007
11.49	7	12012
11.85	8	11695

Table 4: Plate 1

Wire Speed (in/s)	CNC Speed (in/min)	Avg Droplet Spacing (μs)
3.48	2	20014
3.47	3	23121
3.3	4	23530
3.3	5	27398
2.96	6	26845
3.39	7	29646
3.42	8	30534
4.07	2	24342
4.19	3	19802
3.95	4	17467
3.86	5	18868
4.14	6	14545
4.09	7	13333
3.99	8	19900
6.55	2	11267
6.8	3	11173
7.23	4	12012
6.49	5	9788
6.38	6	9389
6.3	7	9216
6.68	8	9009
9.8	2	9442
9.57	3	8928
10.28	4	9456
10.89	5	9280
9.89	6	9204
9.83	7	8050
10.01	8	9456

Wire Speed (in/s)	CNC Speed (in/min)	Avg Droplet Spacing (us)
11.49	2	8677
11.46	3	8340
11.43	4	8752
11.75	5	8798
11.6	6	9302
11.53	7	9029
11.69	8	9132

Table 5: Plate 2

Wire Speed (in/s)	**CNC Speed (in/min)**	**Avg Droplet Spacing (us)**
5.34	2	12658
4.9	3	14053
4.97	4	12317
5.18	5	16461
4.53	6	11019
4.83	7	79756
n/a	8	n/a
7.12	2	9876
6.93	3	9959
7.22	4	10315
6.82	5	9527
6.74	6	9195
6.8	7	9227
6.87	8	9852
9.3	2	8948
9.19	3	9091
9.42	4	9599
9.26	5	9466
9.31	6	9204
n/a	7	n/a
9.16	8	8180
12.58	2	8421
12.41	3	8445
12.77	4	8465
12.74	5	8792
12.56	6	8620
12.66	7	8602
12.82	8	8714
14.67	2	8477
15.19	3	8585
14.87	4	8756
14.69	5	8547
14.89	6	8677
14.32	7	8639
14.65	8	8403

Table 6: Plate 3

Wire Speed (in/s)	**CNC Speed (in/min)**	**Avg Droplet Spacing (us)**
3.83	4	9013
3.85	4.5	9828
3.89	5	9780
4	5.5	11662
3.96	6	9552
3.97	6.5	10447
3.8	7	10371
4.66	4	9860
5.07	4.5	9780
4.61	5	9569
4.64	5.5	9238
4.55	6	9501
4.59	6.5	9099
4.75	7	9287
4.77	4	9909
4.63	4.5	10032
4.6	5	9029
5.19	5.5	9784
4.56	6	n/a
4.67	6.5	9756
4.65	7	9434
5.35	4	9645
5.37	4.5	10025
5.34	5	9758
5.35	5.5	10554
5.61	6	9546
5.17	6.5	9961
5.36	7	9307
6.1	4	8877
5.98	4.5	9412
6.09	5	9615
6.03	5.5	9350
6.76	6	9863
6.35	6.5	10474
6.15	7	9198

Table 7: Plate 4

Wire Speed (in/s)	CNC Speed (in/min)	Avg Droplet Spacing (us)
4.09	4.8	9195
4.47	5	9737
4.28	5.2	10126
3.39	5.4	9623
4.25	5.6	9740
4.09	5.8	9876
3.98	6	9262
4.61	4.8	9953
4.38	5	9761
4.56	5.2	9389
4.77	5.4	10582
4.56	5.6	9661
4.53	5.8	9625
4.2	6	9183
4.71	4.8	9266
4.61	5	9569
4.94	5.2	9479
4.62	5.4	9376
4.92	5.6	9184
4.89	5.8	9324
4.86	6	9546
5.21	4.8	8994
4.97	5	9111
5.31	5.2	9595
5.21	5.4	9324
5.02	5.6	9117
5.1	5.8	9153
5.26	6	9456
5.37	4.8	9111
5.56	5	9376
5.51	5.2	9456
	5.4	
	5.6	
	5.8	
	6	

Table 8: Plate 5

Wire Speed (in/s)	**CNC Speed (in/min)**	**Avg Droplet Spacing (us)**
2.57	2	12903
2.55	3	12195
2.59	4	13559
2.48	5	15873
2.79	6	15269
2.62	7	16953
2.63	8	14760
3.38	2	10392
3.27	3	10309
3.72	4	11396
3.7	5	12232
3.39	6	11950
3.45	7	11954
3.52	8	14100
4.12	2	9464
4	3	9662
3.98	4	10000
4.26	5	10256
4.54	6	10816
4.41	7	10816
3.91	8	11665
4.47	2	9117
4.48	3	8960
4.47	4	9284
4.84	5	9756
4.39	6	9112
4.98	7	11628
4.37	8	11338
n/a	2	n/a
4.64	3	8658
4.67	4	8928
4.71	5	8565
5.07	6	11111
4.99	7	10638
5	8	10340

Table 9: Plate 6

6.4 Degrees to Wire Speed Test

Determining the amount to turn the wire speed knob to increase the overall wire speed by exactly one inch was another test that was completed. This was done by figuring out the amount of time to leave the motor on to turn it 30° and measuring the difference in wire speeds. Multiple iterations of this test gave us a good average value, which came out to be about 40° per inch per second. This value was then used to determine how long to turn the stepper motor on, so as to reach a final desired wire speed setting.

Run #	**Initial Wire Speed (in/s)**	**Final Wire Speed (in/s)**	**Turn Amount (deg)**	**Ratio (deg/(in/s))**	**Time/deg (us)**	**Pulse On Time (us)**
Run 1	0.3	1.1	30	37.5	22222	666660
Run 2	1.1	1.8	30	42.85714286	22222	666660
Run 3	1.8	2.5	30	42.85714286	22222	666660
Run 4	2.5	3.3	30	37.5	22222	666660
Run 5	1.9	3	45	40.90909091	22222	999990
Run 6	3	4.2	45	37.5	22222	999990
Run 7	1.2	2.3	45	40.90909091	22222	999990
Run 8	2.3	3.4	45	40.90909091	22222	999990
Run 9	0.9	2.4	60	40	22222	1333320
Run 10	2.4	3.9	60	40	22222	1333320
Run 11	3.9	5.5	60	37.5	22222	1333320
Run 12	3.4	5	60	37.5	22222	1333320
Run 13	1	7	235	39.16666667	22222	5222170
			Average	**39.66179654**		

Table 10: Test Results to Determine Degrees to Turn for 1-inch Increase in Wire Speed

7 Problems Faced

There were several problems faced throughout this project. From the very beginning to the end, there were problems with the CNC machine. At the beginning of the project, the CNC was only semi-functional. The industry sponsor built the CNC in 2006 as part of his Masters Degree. It had not been used much sense then, so there were a lot of kinks and various wiring issues that needed to be resolved. At the beginning of the project, a fair amount of time was used to just get the CNC working correctly.

Due to using time to get the CNC working properly, as well as other issues faced, time management was another problem faced in this project. Because not enough time was put in during the first half of the project, the second half required a lot more work. Meetings increased from one a week to several times a week, often with the group meeting until after midnight to work on different aspects of the project.

Another issue that caused a problem was the temperature sensor. Early on in the project, the sponsor purchased an IR temperature sensor. At the time, it seemed like the purchased sensor would be able to accomplish the necessary tasks. However, after looking into the documentation a little more closely, it was discovered that the distance of measurement was about 30 inches. In other words, the sensor needed to be placed 30 inches from what needed to be measured. This was an issue in that the printer enclosure didn't allow for distance. A new temperature sensor was ordered, this time one that only needed 18 inches. This was still a bit too long, but was much more acceptable.

At the end of the project when the temperature sensor was finally being implemented, it was discovered the the temperature ranges of the sensor were unacceptable. Measurements were needed to ensure the area of deposition was within a set threshold: not too hot nor too cold. The lowest readable range of the sensor was 800 °C, whereas the required reading was only about 250 °C. Once again, the sensor would not adequately accomplish it's task. Due to the limited time left, the temperature sensor was abandoned at this point.

The last major issue faced was the wire speed in accordance with the current setting. In testing the wire feed settings, it was discovered that for certain settings the wire speed was being hindered by the low current. With a lower current and a higher wire speed, the wire would hit the baseplate, but due to the low current setting, the wire wouldn't melt fast enough. This caused the wire to get jammed up and feed at a rate slower than what was specified. For example, testing showed that for a current setting of 'B', any wire feed speed above about three inches per minute would be slowed down by the low current setting. Granted, the movement speed of the CNC machine would also play a role, but for a large portion of the testing, the CNC speed was constant.

8 Future Work

This project was in principle of proof-of-concept. That being said, there's a lot of future work to be done to improve on the final output of the project. The first, and perhaps most important of these would be the temperature control. Because of the issues faced with the temperature sensors, temperature control ended up not being a part of this project.

For the intial layer of deposition, it's important to heat up the baseplate. If the baseplate is too cold, the output will be a cold weld, which is not an ideal circumstance. If following layers are deposited onto a surface that is too hot, the deposition will re-melt the previous layer and any manner of precision will be lost. To control both of these problems, a heating and measurement system should be put in place.

The first step would be to install a sensor that will get appropriate readings. Once this is in place, the control program can be slightly altered to start and stop the CNC if the weld is too hot or too cold. After the sensor is working properly, the following step would be to develop a torch routine. The sponsor would like to use an acetylene torch to heat up the system when it is too cold. A control program would need to be developed for this, as well as the physical mechanism to control the heating process. Once the sensor and torch are in place, the system would then be able to run by itself and make deposits on layers of the ideal temperature. In addition, too speed up the overall process, a cooling system could also be installed. Currently, there is a manual waiting period of about two minutes in between layers of deposition. This time could be dramatically decreased if a system was in place to force the temperature to drop, rather than just waiting for the previous layer to cool naturally. It should be noted the the results of forcing the temperature to drop are unknown. Doing so may cause unexpected results to the deposition.

The next major step would be to incorporate a GUI with live feedback of the properties of the weld. This was a request of the sponsor at the beginning of the project, but due to time constraints was abandoned. Included in this GUI, the sponsor would like to be able to adjust the control components of the project, such as wire feed speed or the temperature settings.

Another improvement to this project would be to allow the control program to control the CNC speed. The speed of the CNC is controlled by an analog signal that can be adjusted to any value between the set min and max. Currently, this controller is hooked up to a digital signal and was used only to start and stop the CNC. Further advances could be made to put this on one of the analog pins and allow the control program to entirely control the CNC movement speed. There are some considerations that must be met for this to work properly, such as the fact that the max CNC speed would be set only in LinuxCNC.

Other additions that could be made to increase the overall precision of deposition would be automated control of the current setting on the welder and better wire speed control. In testing, it appeared that a higher current setting worked better for the initial layer of deposition, while a lower setting worked better for following layers. As of the end of this project, this control was done manually. However, this could easily be accomplished using a stepper motor. Regarding the wire feed speed, it would be beneficial to find a way to overcome the limitations of the wire speed set by the current setting.

9 Photos and Videos of Progress

Demo of the Printer
https://www.youtube.com/watch?v=Ypetogtn1Iw

One of the fist things done in this project was to confirm the operation of the CNC machine. To do this, G-Code of a 2D image was uploaded to the machine. A felt marker was used to draw the image below.

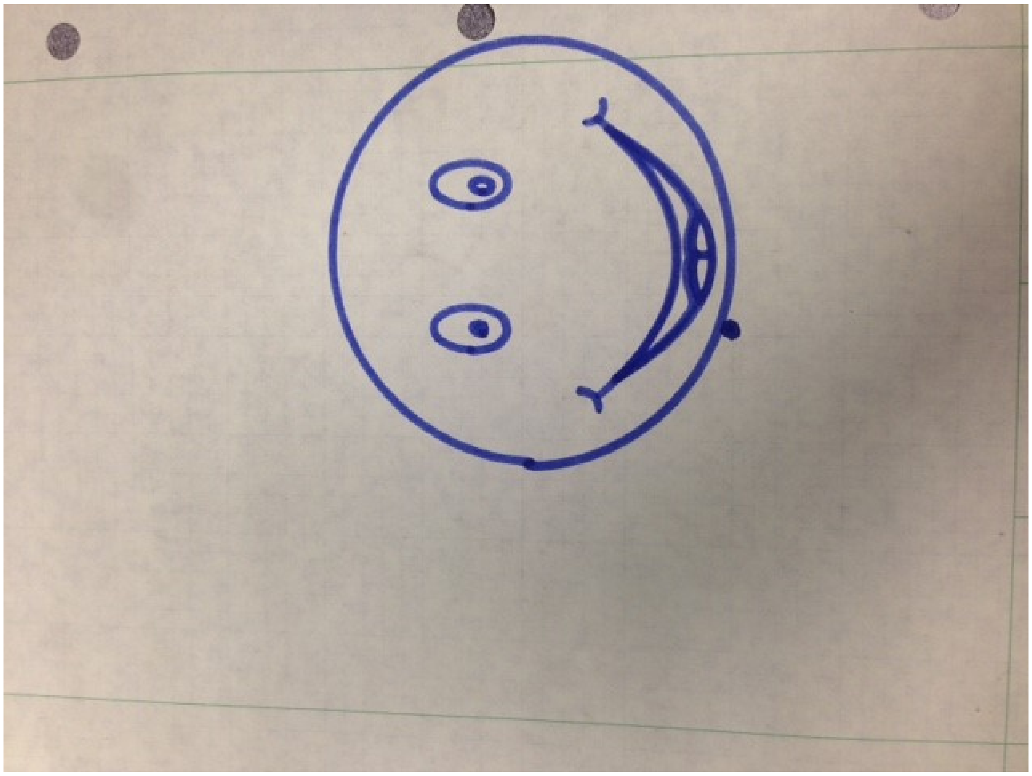


Figure 26: CNC Test Results

Next, we fitted the welder to the machine, and placed a metal base plate to weld on. To control the welder we just used our hand to active the weld while the machine was moving.

Figure 27: Initial Test Results - No Control

After this, we connected the relay switch circuit in parallel with the manual welder switch, using G-Code to activate the switch. Shown Below is the result of letting the CNC Machine control the weld.

Figure 28: Initial Test Results - No Control

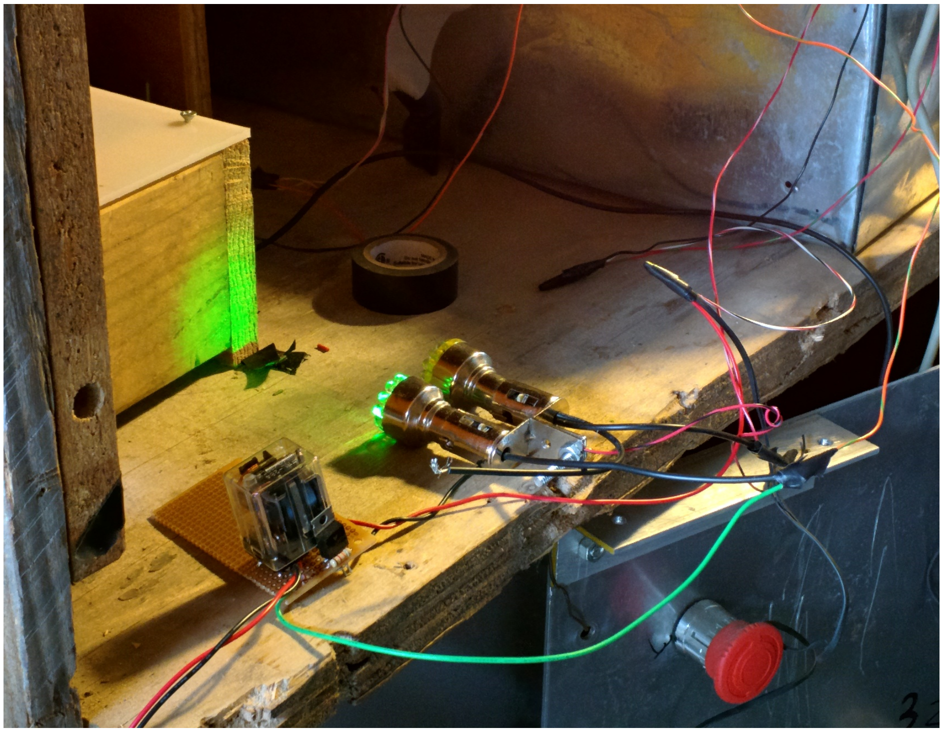


Figure 29: Prototype Relay and Indication Module

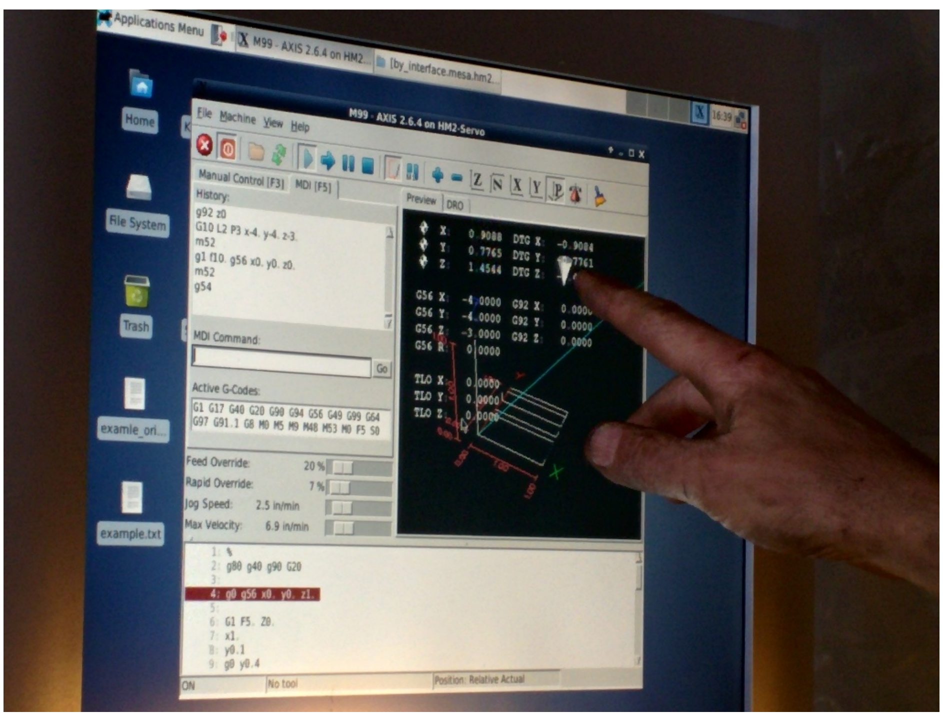


Figure 30: Linux CNC Interface

Figure 31: Baseplate Weld Setup

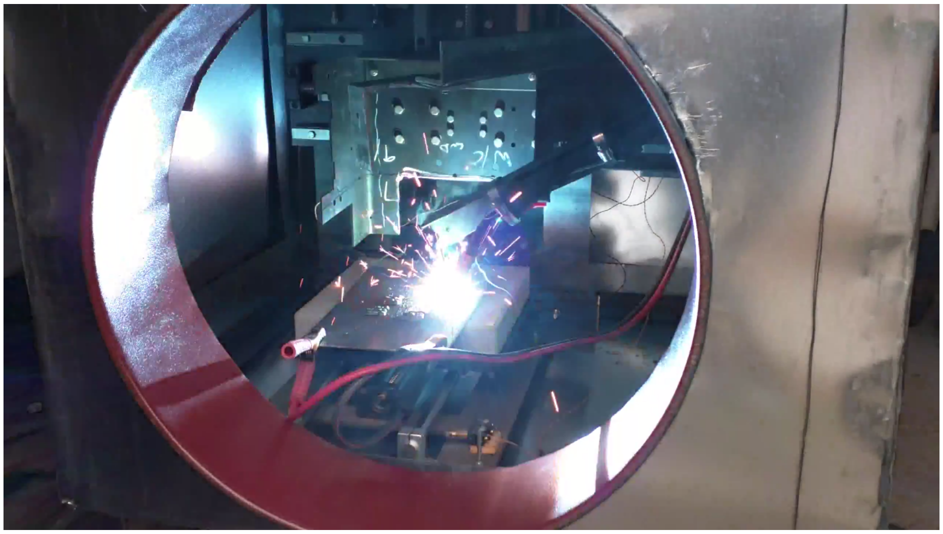


Figure 32: Printer in Deposition Mode

10 Bill of Materials (BOM)

Item	Quantity	Part Number	Mfg	Price	Description
CNC Machine	1	NPN	Aram Kasparov	~$500	
- X-Stepper Motor	1	IG34CK-32-IE8192-SB213	Servo Dynamics	$400	Stepper Motor for the X axis
- Y-Stepper Motor	1	HJ130E8-1308	Servo Dynamics	$800	Stepper Motor for the Y axis
- Z-Stepper Motor	1	IG34CK-32-IE8192-SB213	Servo Dynamics	$450	Stepper Motor for the Z axis
- Motor Driver	3	SD94	Servo Dynamics	$550	Motor driver for XYZ axis motors.
Limit and Home Switches	11	Z15G1308	HIGHLY	$10	Any limit and home switch used on the machine
PCI I/O Card	1	5I20	Mesa Electronics	$250	Part of the CNC system
Servo Interface Card	1	7i33	Mesa Electronics	$100	Interface Card for communicating with servo motors
Isolated I/O Card	2	7i37	Mesa Electronics	$100	I/O card used for limit and home switches
50 Pin Breakout Board	3	NPN		~$10	To connect to analog & digital channels of the Sensoray
Screw Terminals		NPN	Capstone 2015 Team		
50 Pin Connector		NPN	Capstone 2015 Team		

Item	Quantity	Part Number	Mfg	Price	Description
26 Pin Breakout Board	2	NPN		~$10	To connect to the counter channels of the Sensoray
Screw Terminals		NPN	Capstone 2015 Team		
26 Pin Connector		NPN	Capstone 2015 Team		
Incremental Encoder	1	S5-5000-250-IE-D-B	US Digital	$140	Used for measuring the wire speed of the welder
PCIe DAQ	1	826	Sensoray	$677	PCI I/O Card with Digital and Analog I/O
Temperature Sensor	1	1MH1-CF4	Micro-Epsilon	$1400	Temp range x-x
Current Sensor	1	c20058	Honeywell	$30	Outputs a single Voltage
MIG Welder	1	MIG 180	Chicago Electric	$270	Wire Feed Welder that uses inert shielding gas
Motor Controller	2	KL-5056D	Keling Technology Inc	$80	To control the motors on the welder control knobs
Stepper Motors	1	KL23H2100-35-4B	Keling Technology Inc	$50	To control the knobs on the welder
PWM Module	1	THC-AD	Mesa Electronics	$80	To externally set CNC speed
Controller PC	1	NPN	N/A	$80	Computer used to control welder
Relay Module	1	NPN	Capstone 2015 Team	~$20	
Current Sensor Module	1	NPN	Capstone 2015 Team	~$5	
			Total	**~$6,012**	

Appendix A: Sensoray 826i Manual

Sensoray 826 User Manual Download

Appendix B: Linux CNC

LinuxCNC User Manual Download

Appendix C: Visual Studio 2013 on the 826i

How to Create a New Project in Visual Studio 2013 on the 826 board

Creating a Program in C

Our group is doing all of our coding is C, and since VS doesn't have C as one of its default languages, special care should be taken when setting up a new project to be sure that it will compile in C.

1. Open VS
2. Select File -> New -> Project
3. When the New Project dialog box appears, select Visual C++ in the left pane.
4. In the Project window, select Win32 Console Application.
5. Name to Project.
6. When the Win32 Application Wizard box appears, click Next on the Welcome Page.
7. On the Application Settings, make sure the following are selected:
   a. Application Type: Console Application
   b. Additional Options: Empty Project
8. Click Finish

You now have a C project. Now we need to make the C files:

1. If Solution Explorer is not visible, go to View -> Solution Explorer.
2. Right click the Source Files folder in the Solution Explorer and select Add -> New Item
3. The New Item dialog box should appear.
4. Select C++ File(.cpp) and give it a name, but be sure to add the .c extension (for example, your file name might be 826controller.c)
5. Your source file is now in C, and you can start programming.

Compiling the Program

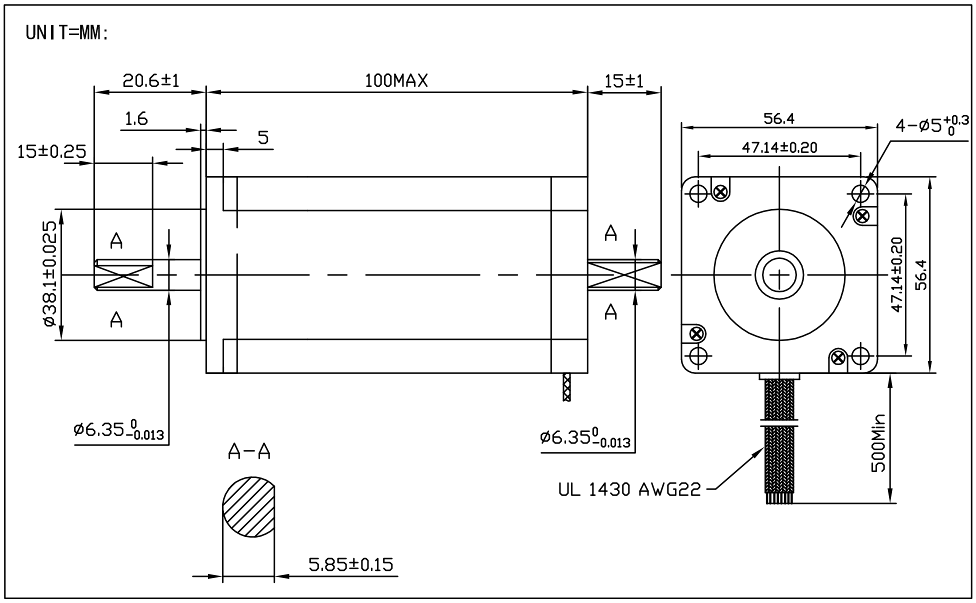
The only thing that needs to be done differently that compiling a normal program in VS is you need to add the DLL .lib file so VS can call on all the board's functions.

1. Click on Debug from the menubar and select "Project Name" Properties
2. When the dialog box opens, expand Linker on the left pane
3. Click on General
   a. On the right hand side find Additional File Directories and click the down arrow on the far right to edit.
   b. Type (or paste) in the address to the file where the .lib and .dll files are located.
4. Click on Input on the left pane
   a. Select Additional Dependencies and click the down arrow to edit
   b. Type (or paste) in the address of the .lib file
5. Close the Properties dialog box.
6. Your program is now ready to compile and talk to the 826 board

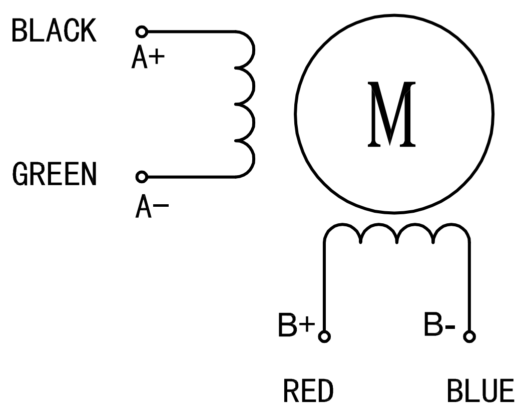
Appendix D: Stepper Motor and Driver

Hybrid Stepper Motor

KL23H2100-35-4B



PHASE	STEP ANGLE	RATED VOLTAGE	CURRENT	RESISTANCE	INDUCTANCE	HOLDING TORQUE	WEIGHT
	DEG/STEP	V	A	ohms	mH	OZ-IN	Kg
2	1.8	2.55	3.5	0.73	2.8	381	1.5



BLACK TO A+

GREEN TO A-

RED TO B+

BLUE TO B-

1430-AWG18#LENGTH310mm

KL-5056D

Fully Digital Stepping Driver

Attention: Please read this manual carefully before using the driver!

Table of Contents

1. Introduction, Features and Applications

Introduction

The KL-5056D is a versatility fully digital stepping driver based on a DSP with advanced control algorithm. The KL-5056D is the next generation of digital stepping motor controls. It brings a unique level of system smoothness, providing optimum torque and nulls mid-range instability. Motor self-test and parameter auto-setup technology offers optimum responses with different motors and easy-to-use. The driven motors can run with much smaller noise, lower heating, smoother movement than most of the drivers in the markets. Its unique features make the KL-5056D an ideal solution for applications that require low-speed smoothness.

Features

- Anti-Resonance, provides optimum torque and nulls mid-range instability
- Motor self-test and parameter auto-setup technology, offers optimum responses with different motors
- Multi-Stepping allows a low resolution step input to produce a higher microstep output for smooth system performance
- Microstep resolutions programmable, from full-step to 102,400 steps/rev
- Supply voltage up to +50 VDC
- Output current programmable, from 0.5A to 5.6A
- Pulse input frequency up to 200 KHz
- TTL compatible and optically isolated input
- Automatic idle-current reduction
- Suitable for 2-phase and 4-phase motors
- Support PUL/DIR and CW/CCW modes
- Over-voltage, over-current, phase-error protections

Applications

Suitable for a wide range of stepping motors, from NEMA frame size 17 to 34. It can be used in various kinds of machines, such as laser cutters, laser markers, high precision X-Y tables, labeling machines, and so on. Its unique features make the KL-5056D an ideal solution for applications that require both low-speed smoothness and high speed performances.

2. Specifications

Electrical Specifications (T_j = 25℃/77℉)

Parameters	KL-5056D			
	Min	Typical	Max	Unit
Output current	0.5	-	5.6 (4.0 RMS)	A
Supply voltage	+20	+36	+50	VDC
Logic signal current	7	10	16	mA
Pulse input frequency	0	-	200	kHz
Isolation resistance	500			MΩ

Mechanical Specifications (unit: mm [inch])

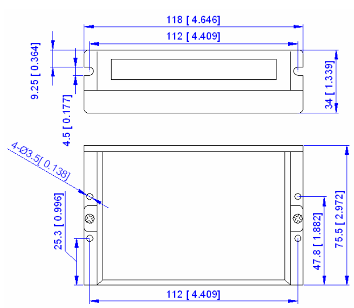


Figure 1: Mechanical specifications

Elimination of Heat

- Driver's reliable working temperature should be <70℃(158℉), and motor working temperature should be <80℃(176℉);
- It is recommended to use automatic idle-current mode, namely current automatically reduce to 60% when motor stops, so as to reduce driver heating and motor heating;
- It is recommended to mount the driver vertically to maximize heat sink area. Use forced cooling method to cool the system if necessary.

Operating Environment and other Specifications

Cooling	Natural Cooling or Forced cooling	
Operating Environment	Environment	Avoid dust, oil fog and corrosive gases
	Ambient Temperature	0℃ － 50℃ (32℉ － 122℉)
	Humidity	40%RH － 90%RH
	Operating Temperature	70℃ (158℉) Max
	Vibration	5.9m/s^2 Max
Storage Temperature	-20℃ － 65℃ (-4℉ － 149℉)	
Weight	Approx. 280g (10 oz)	

3. Pin Assignment and Description

The KL-5056D has two connectors, connector P1 for control signals connections, and connector P2 for power and motor connections. The following tables are brief descriptions of the two connectors. More detailed descriptions of the pins and related issues are presented in section 4, 5, 9.

Connector P1 Configurations

Pin Function	Details
PUL+	Pulse signal: In single pulse (pulse/direction) mode, this input represents pulse signal, each rising or falling edge active (software configurable); 4-5V when PUL-HIGH, 0-0.5V when PUL-LOW. In double pulse mode (pulse/pulse) , this input represents clockwise (CW) pulse，active both at high level and low level (software configurable). For reliable response, pulse width should be longer than 2.5μs. Series connect resistors for current-limiting when +12V or +24V used. The same as DIR and ENA signals.
PUL-	
DIR+	DIR signal: In single-pulse mode, this signal has low/high voltage levels, representing two directions of motor rotation; in double-pulse mode (software configurable), this signal is counter-clock (CCW) pulse，active both at high level and low level (software configurable). For reliable motion response, DIR signal should be ahead of PUL signal by 5μs at least. 4-5V when DIR-HIGH, 0-0.5V when DIR-LOW. Please note that rotation direction is also related to motor-driver wiring match. Exchanging the connection of two wires for a coil to the driver will reverse motion direction.
DIR-	
ENA+	Enable signal: This signal is used for enabling/disabling the driver. High level (NPN control signal, PNP and Differential control signals are on the contrary, namely Low level for enabling.) for enabling the driver and low level for disabling the driver. Usually left **UNCONNECTED (ENABLED)**.
ENA-	

Selecting Active Pulse Edge and Control Signal Mode

The KL-5056D supports PUL/DIR and CW/CCW modes and pulse actives at rising or falling edge. See more information about these settings in Section 13. Default setting is PUL/DIR mode and rising edge active (NPN, and PNP control signal is on the contrary).

Connector P2 Configurations

Pin Function	Details
+Vdc	Power supply, 20~50 VDC, Including voltage fluctuation and EMF voltage.
GND	Power Ground.
A+, A-	Motor Phase A
B+, B-	Motor Phase B

4. Control Signal Connector (P1) Interface

The KL-5056D can accept differential and single-ended inputs (including open-collector and PNP output). The KL-5056D has 3 optically isolated logic inputs which are located on connector P1 to accept line driver control signals. These inputs are isolated to minimize or eliminate electrical noises coupled onto the drive control signals. Recommend use line driver control signals to increase noise immunity of the driver in interference environments. In the following figures, connections to open-collector and PNP signals are illustrated.

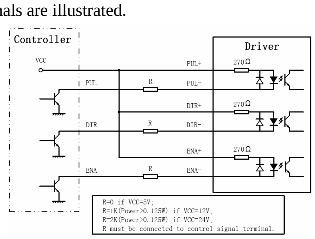


Figure 2: Connections to open-collector signal (common-anode)

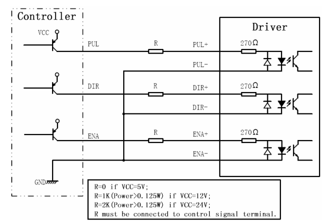


Figure 3: Connection to PNP signal (common-cathode)

5. Connecting the Motor

The KL-5056D can drive any 2-pahse and 4-pahse hybrid stepping motors.

Connections to 4-lead Motors

4 lead motors are the least flexible but easiest to wire. Speed and torque will depend on winding inductance. In setting the driver output current, multiply the specified phase current by 1.4 to determine the peak output current.

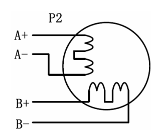


Figure 4: 4-lead Motor Connections

Connections to 6-lead Motors

Like 8 lead stepping motors, 6 lead motors have two configurations available for high speed or high torque operation. The higher speed configuration, or half coil, is so described because it uses one half of the motor's inductor windings. The higher torque configuration, or full coil, uses the full windings of the phases.

Half Coil Configurations

As previously stated, the half coil configuration uses 50% of the motor phase windings. This gives lower inductance, hence, lower torque output. Like the parallel connection of 8 lead motor, the torque output will be more stable at higher speeds. This configuration is also referred to as half chopper. In setting the driver output current multiply the specified per phase (or unipolar) current rating by 1.4 to determine the peak output current.

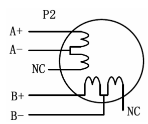


Figure 5: 6-lead motor half coil (higher speed) connections

Full Coil Configurations

The full coil configuration on a six lead motor should be used in applications where higher torque at lower speeds is desired. This configuration is also referred to as full copper. In full coil mode, the motors should be run at only 70% of their rated current to prevent over heating.

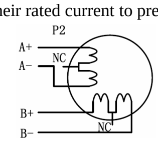


Figure 6: 6-lead motor full coil (higher torque) connections

Connections to 8-lead Motors

8 lead motors offer a high degree of flexibility to the system designer in that they may be connected in series or parallel, thus satisfying a wide range of applications.

Series Connections

A series motor configuration would typically be used in applications where a higher torque at lower speeds is required. Because this configuration has the most inductance, the performance will start to degrade at higher speeds. In series mode, the motors should also be run at only 70% of their rated current to prevent over heating.

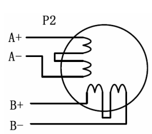


Figure 7: 8-lead motor series connections

Parallel Connections

An 8 lead motor in a parallel configuration offers a more stable, but lower torque at lower speeds. But because of the lower inductance, there will be higher torque at higher speeds. Multiply the per phase (or unipolar) current rating by 1.96, or the bipolar current rating by 1.4, to determine the peak output current.

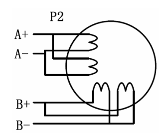


Figure 8: 8-lead motor parallel connections

NEVER disconnect or connect the motor while the power source is energized.

6. Power Supply Selection

The KL-5056D can match medium and small size stepping motors (from NEMA frame size 14 to 34) made by Keling or other motor manufactures around the world. To achieve good driving performances, it is important to select supply voltage and output current properly. Generally speaking, supply voltage determines the high speed performance of the motor, while output current determines the output torque of the driven motor (particularly at lower speed). Higher supply voltage will allow higher motor speed to be achieved, at the price of more noise and heating. If the motion speed requirement is low, it’s better to use lower supply voltage to decrease noise, heating and improve reliability.

Regulated or Unregulated Power Supply

Both regulated and unregulated power supplies can be used to supply the driver. However, unregulated power supplies are preferred due to their ability to withstand current surge. If regulated power supplies (such as most switching supplies.) are indeed used, it is important to have large current output rating to avoid problems like current clamp, for example using 4A supply for 3A motor-driver operation. On the other hand, if unregulated supply is used, one may use a power supply of lower current rating than that of motor (typically 50%～70% of motor current). The reason is that the driver draws current from the power supply capacitor of the unregulated supply only during the ON duration of the PWM cycle, but not during the OFF duration. Therefore, the average current withdrawn from power supply is considerably less than motor current. For example, two 3A motors can be well supplied by one power supply of 4A rating.

Multiple Drivers

It is recommended to have multiple drivers to share one power supply to reduce cost, if the supply has enough capacity. To avoid cross interference, **DO NOT** daisy-chain the power supply input pins of the drivers. Instead, please connect them to power supply separately.

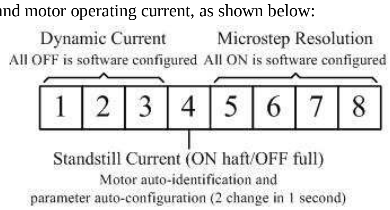
Selecting Supply Voltage

The power MOSFETS inside the KL-5056D can actually operate within +20 ~ +50VDC, including power input fluctuation and back EMF voltage generated by motor coils during motor shaft deceleration. Higher supply voltage can increase motor torque at higher speeds, thus helpful for avoiding losing steps. However, higher voltage may cause bigger motor vibration at lower speed, and it may also cause over-voltage protection or even driver damage. Therefore, it is suggested to choose only sufficiently high supply voltage for intended applications, and it is suggested to use power supplies with theoretical output voltage of +20 ~ +46VDC, leaving room for power fluctuation and back-EMF.

7. Selecting Microstep Resolution and Driver Output Current

Microstep resolutions and output current are programmable, the former can be set from full-step to 102,400 steps/rev and the latter can be set from 0.5A to 5.6A. See more information about **Microstep and Output Current Setting** in Section 13.

However, when it's not in software configured mode, this driver uses an 8-bit DIP switch to set microstep resolution, and motor operating current, as shown below:



Microstep Resolution Selection

When it's not in software configured mode, microstep resolution is set by SW5, 6, 7, 8 of the DIP switch as shown in the following table:

Microstep	Steps/rev.(for 1.8°motor)	SW5	SW6	SW7	SW8
1 to 512	Default/Software configured	ON	ON	ON	ON
2	400	OFF	ON	ON	ON
4	800	ON	OFF	ON	ON
8	1600	OFF	OFF	ON	ON
16	3200	ON	ON	OFF	ON
32	6400	OFF	ON	OFF	ON
64	12800	ON	OFF	OFF	ON
128	25600	OFF	OFF	OFF	ON
5	1000	ON	ON	ON	OFF
10	2000	OFF	ON	ON	OFF
20	4000	ON	OFF	ON	OFF
25	5000	OFF	OFF	ON	OFF
40	8000	ON	ON	OFF	OFF
50	10000	OFF	ON	OFF	OFF
100	20000	ON	OFF	OFF	OFF
125	25000	OFF	OFF	OFF	OFF

Current Settings

For a given motor, higher driver current will make the motor to output more torque, but at the same time causes more heating in the motor and driver. Therefore, output current is generally set to be such that the motor will not overheat for long time operation. Since parallel and serial connections of motor coils will significantly change resulting inductance and resistance, it is therefore important to set driver output current depending on motor phase current, motor leads and connection methods. Phase current rating supplied by motor manufacturer is important in selecting driver current, however the selection also depends on leads and connections.

When it's not in software configured mode, the first three bits (SW1, 2, 3) of the DIP switch are used to set the dynamic current. Select a setting closest to your motor's required current.

Dynamic current setting

Peak Current	RMS Current	SW1	SW2	SW3
Default/Software configured (0.5 to 5.6A)		**OFF**	**OFF**	**OFF**
2.1A	**1.5A**	ON	OFF	OFF
2.7A	**1.9A**	OFF	ON	OFF
3.2A	**2.3A**	ON	ON	OFF
3.8A	**2.7A**	OFF	OFF	ON
4.3A	**3.1A**	ON	OFF	ON
4.9A	**3.5A**	OFF	ON	ON
5.6A	**4.0A**	ON	ON	ON

Notes: Due to motor inductance, the actual current in the coil may be smaller than the dynamic current setting, particularly under high speed condition.

Standstill current setting

SW4 is used for this purpose. OFF meaning that the standstill current is set to be half of the selected dynamic current, and ON meaning that standstill current is set to be the same as the selected dynamic current.

The current automatically reduced to 60% of the selected dynamic current one second after the last pulse. Theoretically, this will reduce motor heating to 36% (due to $P=I^2*R$) of the original value. If the application needs a different standstill current, please contact Keling.

8. Wiring Notes

- In order to improve anti-interference performance of the driver, it is recommended to use twisted pair shield cable.
- To prevent noise incurred in PUL/DIR signal, pulse/direction signal wires and motor wires should not be tied up together. It is better to separate them by at least 10 cm, otherwise the disturbing signals generated by motor will easily disturb pulse direction signals, causing motor position error, system instability and other failures.
- If a power supply serves several drivers, separately connecting the drivers is recommended instead of daisy-chaining.
- It is prohibited to pull and plug connector P2 while the driver is powered ON, because there is high current flowing through motor coils (even when motor is at standstill). Pulling or plugging

connector P2 with power on will cause extremely high back-EMF voltage surge, which may damage the driver.

9. Typical Connection

A complete stepping system should include stepping motor, stepping driver, power supply and controller (pulse generator). A typical connection is shown as figure 9.

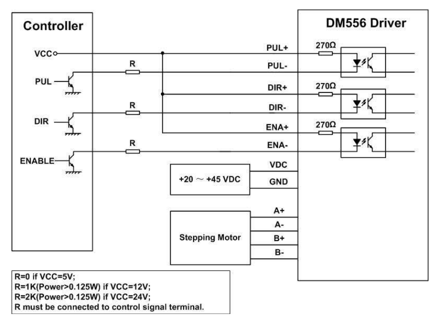


Figure 9: Typical connection

10. Sequence Chart of Control Signals

In order to avoid some fault operations and deviations, PUL, DIR and ENA should abide by some rules, shown as following diagram:

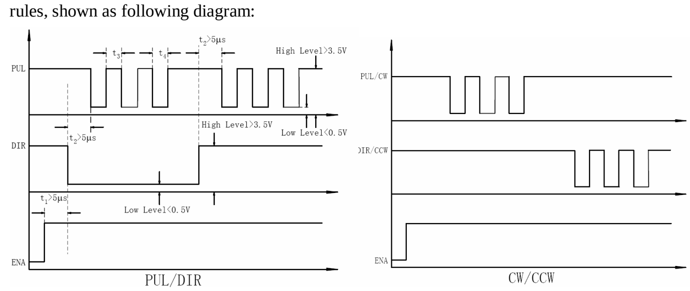


Figure 10: Sequence chart of control signals

Remark:

a) t1: ENA must be ahead of DIR by at least 5µs. Usually, ENA+ and ENA- are NC (not connected). See "Connector P1 Configurations" for more information.
b) t2: DIR must be ahead of PUL active edge by 5µs to ensure correct direction;
c) t3: Pulse width not less than 2.5µs;
d) t4: Low level width not less than 2.5µs.

11. Protection Functions

To improve reliability, the driver incorporates some built-in protection functions. The KL-5056D uses one RED LED to indicate what protection has been activated. The periodic time of RED is 3 s (seconds), and how many times the RED turns on indicates what protection has been activated. Because only one protection can be displayed by RED LED, so the driver will decide what error to display according to their priorities. See the following **Protection Indications** table for displaying priorities.

Over-current Protection

Over-current protection will be activated when continuous current exceeds 16A or in case of short circuit between motor coils or between motor coil and ground, and RED LED will turn on once within each periodic time (3 s).

Over-voltage Protection

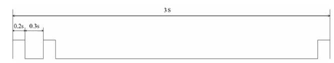
When power supply voltage exceeds 52±1 VDC, protection will be activated and RED LED will turn on twice within each periodic time (3 s).

Phase Error Protection

Motor power lines wrong & not connected will activate this protection. RED LED will turn on four times within each periodic time (3 s).

Attention: When above protections are active, the motor shaft will be free or the LED will turn red. Reset the driver by repowering it to make it function properly after removing above problems. Since there is no protection against power leads (+, -) reversal, it is critical to make sure that power supply leads correctly connected to driver. Otherwise, the driver will be damaged instantly.

Protection Indications

Priority	Time(s) of ON	Sequence wave of RED LED	Description
1st	1		Over-current protection
2nd	2		Over-voltage protection
3rd	4		Phase error protection

12. Frequently Asked Questions

In the event that your driver doesn't operate properly, the first step is to identify whether the problem is electrical or mechanical in nature. The next step is to isolate the system component that is causing the problem. As part of this process you may have to disconnect the individual components that make up your system and verify that they operate independently. It is important to document each step in the troubleshooting process. You may need this documentation to refer back to at a later date, and these details will greatly assist our Technical Support staff in determining the problem should you need assistance.

Many of the problems that affect motion control systems can be traced to electrical noise, controller software errors, or mistake in wiring.

Problem Symptoms and Possible Causes

Symptoms	Possible Problems
Motor is not rotating	No power
	Microstep resolution setting is wrong
	DIP switch current setting is wrong
	Fault condition exists
	The driver is disabled
Motor rotates in the wrong direction	Motor phases may be connected in reverse
The driver in fault	DIP switch current setting is wrong
	Something wrong with motor coil

Erratic motor motion	Control signal is too weak
	Control signal is interfered
	Wrong motor connection
	Something wrong with motor coil
	Current setting is too small, losing steps
Motor stalls during acceleration	Current setting is too small
	Motor is undersized for the application
	Acceleration is set too high
	Power supply voltage too low
Excessive motor and driver heating	Inadequate heat sinking / cooling
	Automatic current reduction function not being utilized
	Current is set too high

13. Professional Tuning Software ProTuner

Introduction

This section will provide an overview of connection and basic setup instructions for Keling’s digital stepping driver KL-5056D using the **ProTuner** software. These instructions will walk you through the following steps necessary to start up your driver and motor. This section is intended for setting up the driver with the **ProTuner**.

Software Installation

The **ProTuner** is windows based setup software for tuning Keling’s digital stepper driver KL-5056D. It can run in windows systems, including Win95/Win98/WindowsNT/ Windows 2000/Windows XP. And the selected PC should have 1 serial port at least for communicating with the driver.

Double click “**ProTuner_All_Setup_V1.0.exe**” to begin installing the **ProTuner**. See Figure 11. Click **Next** to enter the “License Agreement” window. See Figure 12.

Figure 11: Begin to install the ProTuner

Figure 12: License agreement

Choose "I agree to the terms of this license agreement" and click **Next** to continue installation. The user can enter user's information in the following window. See Figure 13. After entering the user's information, click **Next** to select installation folder, where you would like to install the **ProTuner**. See Figure 14.

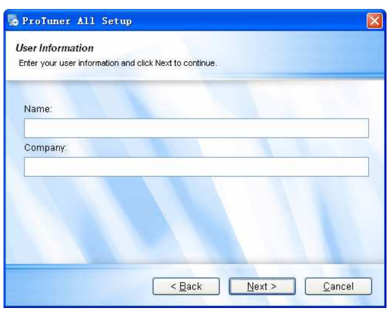


Figure 13: User's information settings

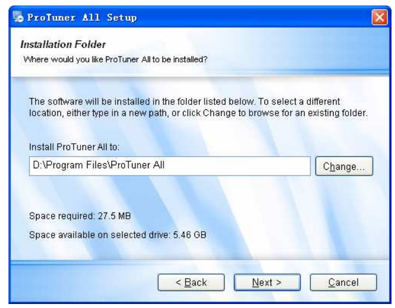


Figure 14: Installation folder settings

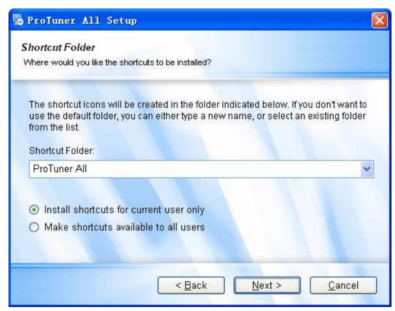


Figure 15: Shortcut folder setting

Set the "Shortcut Folder" in Figure 15 and continue to install the **ProTuner** by following Figure 16 and Figure 17. An **Installation Successful** window will appear if the **ProTuner** is installed successfully. See Figure 18.

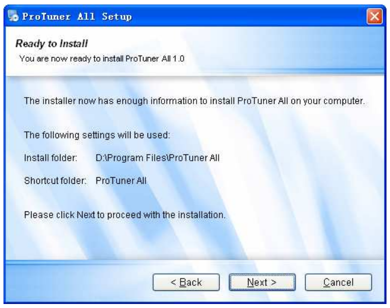


Figure 16: Installation information summarization

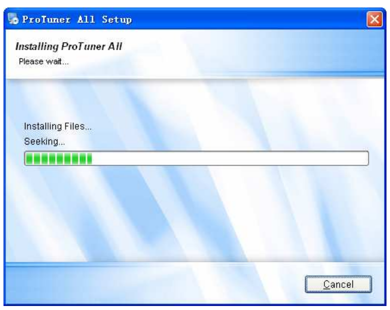


Figure 17: Installing the ProTuner

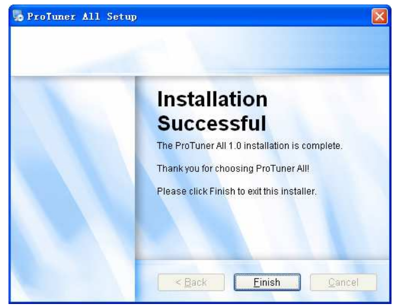


Figure 18: Finish installation

Connections and Testing

Connect the stepping system according to the contents in previous sections and connect the PC to the driver as the following figure.

RS232 Interface Connection

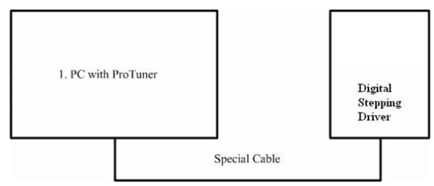


Figure 19: RS232 interface connection

Testing the Stepping System

Turn on the power supply, the green (Power) LED will light. The KL-5056D has default parameters stored in the driver. If the system has no hardware and wirings problem, the motor should be locked and the driver should be ready.

If the red LED immediately turns on (flickers), then check power supply, the motor, motor wirings and try again. Open the tuning software **ProTuner** and check driver status by clicking **Err_check**. If it's **Phase Error**, check the motor, motor wirings and try again. If it still doesn't work after you followed all of the previous steps, please contact us at **kelinginc@kelinginc.net**

If the RED LED is off and the motor is normal, then you can start to tune the servo with **ProTuner**. However, we recommend you see the following contents before starting tuning.

Software Introduction

ProTuner Main Window

➢ **Option**

The user can choose three drop-down menus by clicking "**Option**", including **Com Config**, **SaveToDriver** and **Exit**.

- **Com Config**: Configure Com communication interface.
- **SaveToDriver:** Download the current parameter settings to the driver.
- **Exit:** Exit the **ProTuner**.

Com Config Window

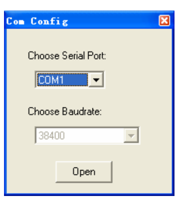


Figure 21: RS232 communication configuration window

Serial Port: Select the serial communication port to which the driver is connected. The factory default setting is COM1.

Baud Rate: Select the communication baud rate. The factory default setting is 38400.

Click **Open** button to establish a connection with the specified settings. When connecting, you can

choose **SaveToDrive** to download the current parameter settings to the driver, or to upload the stored driver settings into the **ProTuner** by clicking **Tuning** > **Position Loop** on the menu bar.

Tuning

The user can choose one or two drop-down menu(s) by clicking **Tuning**, including **CurrentLoop** and **SystemConfig**.

- **CurrentLoop**: In Current Tuning window, the user can tune the **Kp** (**Proportional Gain**) and **Ki** (**Integral Gain)** of driver's current loop to optimize responses with different motors. Start/Restart a Step Response test to get an optimum response.

Kp: Proportional Gain. Proportional Gain determines the response of the driver to current setting command. Low Proportional Gain provides a stable system (doesn't oscillate), has low stiffness, and large current error, causing poor performances in tracking current setting command in each step like Figure 23. Too large Proportional Gain values will cause oscillations and unstable systems.

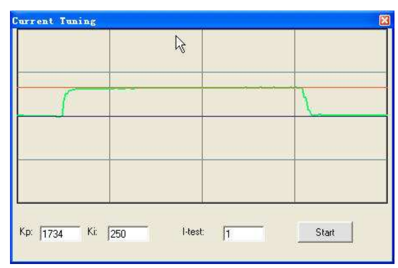


Figure 22: Current Tuning window

Ki: Integral Gain. Integral Gain helps the driver to overcome static current errors. A low or zero value for the Integral Gain may have current errors at rest. Increasing the Integral Gain can reduce the error. If the Integral Gain is too large, the systems may "hunt" (oscillate) about the desired position.

Start button: The user can start a Step Response test by clicking this button. Start/Restart a Step

Response test to get an optimum response like Figure 22, and remember to save the settings to the driver when finish tuning. See Figure 24.

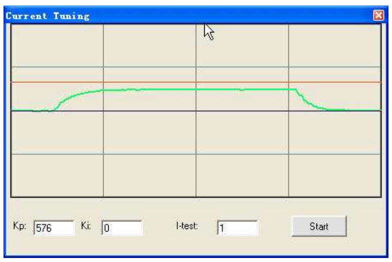


Figure 23: Kp=2604, Ki=0 (poor performances)

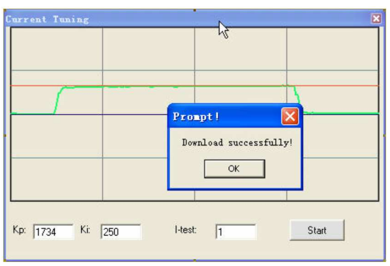


Figure 24: Finish tuning and save setting to the driver

Notes:

However, if the user does not want to tune the current loop after changing a different stepping motor, then **Motor auto-identification and parameter auto-configuration** technology of the KL-5056D can replace manual tuning the driver with **ProTuner**. Just changes SW4 two times in 1 second, and then the driver will auto-identify the new motor and auto-configure related control parameters for optimum responses. **Recommend** use this function after changing the driven motor.

● **SystemConfig**:

In **SystemConfig** window, the user can configure Peak Current, Microstep, Command Type, Active Edge, and eliminate motor resonance. A built-in pulse generator can be used for test during tuning. See Picture 25.

PeakCur: Peak Current. The value is the peak current to the selected motor and can be set from 0.5 to 5.6 A. The user can set the peak current with **ProTuner** or DIP switches, see more information about setting output current of the driver in section 5 "**Connecting the Motor**" and section 7 "**Selecting Microstep Resolution and Driver Output Current**".

MicroStep: Microstep Resolution. The value is driver's microstep resolution setting and can be set from 1 to 512. The user can set the microstep with **ProTuner** or DIP switches, See more information about setting output current of the driver in section 7 "**Selecting Microstep Resolution and Driver Output Current**".

ElecDamp: Electronic Damping Coefficient. The electronic damping restrain resonance of the system and prevent amplitude of the oscillation from increasing to the extend that it makes the motor out of control. The optimal value depends on the system, and the default value is 3000.

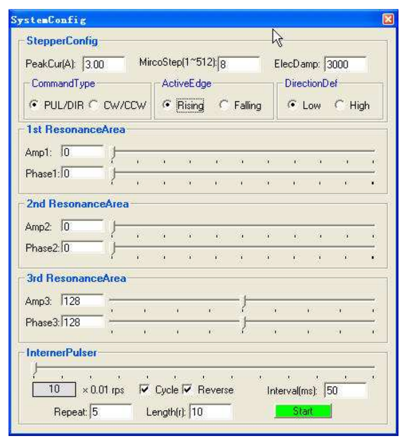


Figure 25: SystemConfig window

CommandType: Command Type of control signal, including PUL/DIR and CW/CCW. Set this parameter according to **Command Type** of motion controller.

ActiveEdge: Active Edge. The user can set the triggered edge of pulse command signal in this panel. When the driver works in CW/CCW mode, no matter what level is at fixed level terminal, the driver can works properly.

DirectionDef: Direction Definition. Relate the default running direction to **a HIGH** level input in DIR or **Low** level input in DIR. This panel is used for PUL/DIR command type only. Please note that the default direction is also related to motor coil connections.

Anti-Resonance Introduction

Step motors are highly resonant, which results in vibration and ringing. The ringing utilizes a large fraction of the motor's available torque – thereby wasting performance. Furthermore, at mid-range velocities, the resonance can become so severe that the motor looses synchronization and stalls. The KL-5056D driver provides robust anti-resonance control to stop the vibrations and maintain equilibrium. This feature requires that the driver be configured with respect to the total inertia in the system. If set improperly, the effectiveness of the feature may be diminished.

The user can invoke or disable the feature by setting **Amp** and **Phase** values in **SystemConfig** window. **Amp** and **Phase** values all zero is to disable the feature, otherwise is to invoke the feature. It should be enabled unless the system configuration either does not need it or cannot tolerate it. A system with loose couplings or viscous loading generally does not need this feature. If a system has compliant (springy) coupling and is absent appreciably viscosity, it may not respond well to the active, anti-resonant loop in the drive. The anti-resonant feature is not designed to damp such a 4th order system. If the application of anti-resonance results in degradation or instability, it should be disabled.

1st ResonanceArea: Parameters for 1st resonance area. Usually between 0.6rps and 1.2rps.

Amp1 is Amplitude adjustment for 1st resonance area.
Phase1 is Phase adjustment for 1st resonance area. The user can enter a value directly in the text box or move the slider bar back and forth to get an optimum value.

2nd ResonanceArea: Parameters for 2nd resonance area. Usually between 1.2rps and 2.4rps. Default **Amp2** and **Phase2** values are zero.

3rd ResonanceArea: Parameters for 3rd resonance area. Usually between 2.4rps and 4.8rps. Default **Amp3** and **Phase3** values are 128.

InternerPulser: There is an internal pulse generator designed for driver self-testing and anti-resonance tuning. You can issue a motion by this simple controller.
Cycle check box: The motion will repeat if this box is checked.
Reverse check box: The motor shaft will reverse direction if this box is checked.
Interval edit box: The stop time between each cycle, unit is **millisecond**.
Repeat edit box: Total motion cycles.
Length edit box: Move distance of each cycle, unit is **revolution**.
Start/Stop button: The user can Start/Stop a motion test by clicking this button.

Procedure for Achieving Optimum Performance

Step 1: Start the motion test by clicking **Start/Stop** button. Find a resonance speed by slightly moving the slider bar of internal pulse generator back and forth. See Figure 26.
Step 2: Run the motor at the resonance speed and verify the motor smoothness. You may find a better smoothing value by slightly moving the slider bars of **AMP**(s) and **Phase**(s) back and forth.
It is very important to make the **AMP**(s) and **Phase**(s) adjustments at the proper test speeds with an unloaded motor. Running at an incorrect test speed will not excite the motor at its peak resonance, making it more difficult to find proper adjustment values. Optimum **AMP**(s) and **Phase**(s) values may be a little different between running the tests with an unloaded motor and a load motor.
Please remember to click **SavetoDrive** to download the final parameter settings to the driver when finish tuning. See Figure 27.

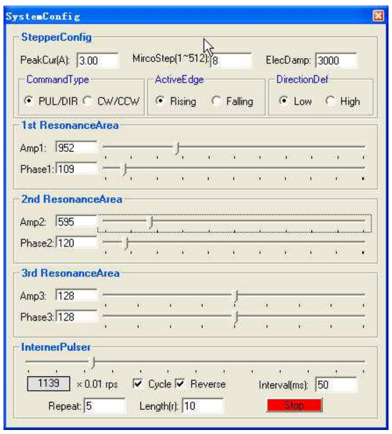


Figure 26: Anti-resonance tuning

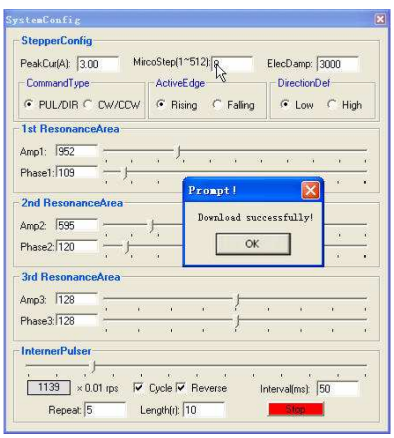


Figure 27: Finishing tuning and download parameter settings to the driver

➢ **Err_check**

- **Error Check**: This window shows both the present status of each error event and their history. Current error event(s) can be reset by clicking **Erase Current Err!** button, and all error events can be reset by clicking **Erase All!** button. List of the last ten drive faults. #0 being the most recent, #9 is the oldest. See Figure 28.

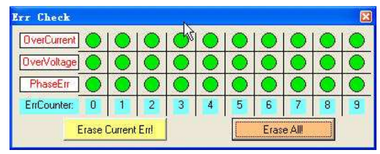


Figure 28: Error check window

OverCurrent: Over-current Protection. Protection will be activated when continuous current exceeds 16A.

OverVoltage: Over-voltage Protection. When power supply voltage exceeds 52±1 VDC, protection will be activated.

PhaseErr: Phase Error Protection. Motor power lines wrong & not connected will activate this protection.

ErrCounter: Displays current error(s) and current error history.

Erase Current Err!: Erase Current Err button. The user can clear current error(s) by clicking this button.

Erase All!: Erase All! button. The user can clear all error(s) including error history by clicking this button.

➢ **About**

The user can choose two drop-down menus by clicking "**About**", including **Product Information** and

Contact Us.

- **Product Information** window: Shows some product information about ProTuner.

.

Appendix E: Optical Shaft Encoder

RoHS COMPLIANT

Description

The **S5** series optical shaft encoder is a non-contacting rotary to digital converter. Useful for position feedback or manual interface, the encoder converts real-time shaft angle, speed, and direction into TTL-compatible quadrature outputs with or without index. The encoder utilizes a mylar disk, metal shaft and bushing, LED light source, and monolithic electronics. It operates from a single +5VDC supply.

Three shaft torque versions are available. The standard torque version has a sleeve bushing lubricated with a viscous motion control gel to provide torque and feel that is ideal for front panel human interface applications.

The no torque added option has a sleeve bushing and a low viscosity lubricant (that does not intentionally add torque) for low RPM applications where a small amount of torque is acceptable.

The ball bearing version uses miniature precision ball bearings that are suitable for high speed and ultra low torque applications.

A secure connection to the **S5** series encoder is made through a 5-pin (single-ended version) or 10-pin (differential version) finger-latching connector (sold separately). The mating connectors are available from US Digital with several cable options and lengths.

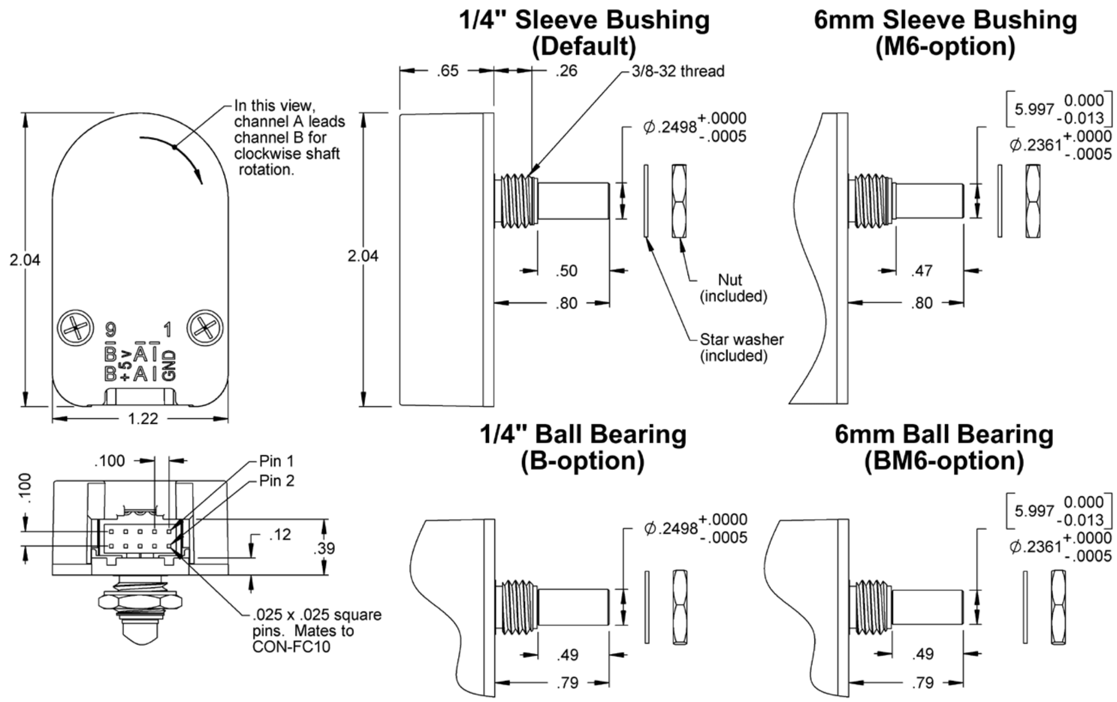
For differential version: the internal differential line driver (26C31) can source and sink 20mA at TTL levels. The recommended receiver is industry standard 26C32. Maximum noise immunity is achieved when the differential receiver is terminated with a 150 Ω resistor in series with a .0047 μF capacitor placed across each differential pair. The capacitor simply conserves power; otherwise power consumption would increase by approximately 20mA per pair, or 60mA for 3 pairs.

Features

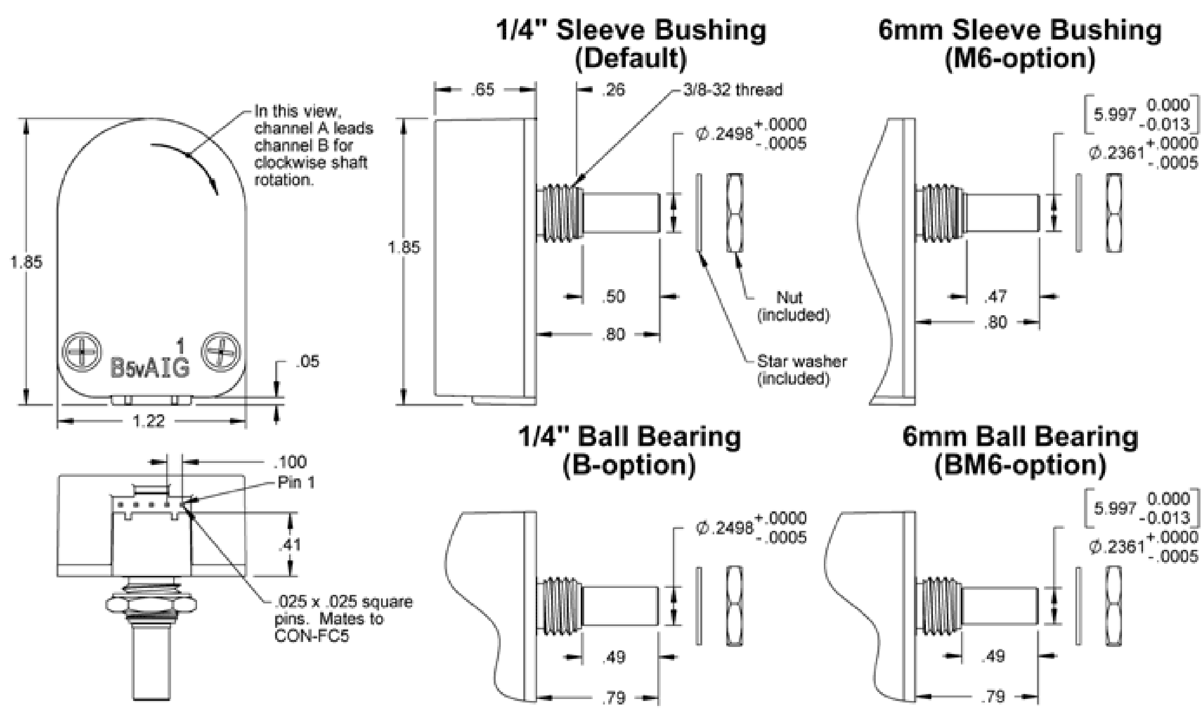
- Small size
- Low cost
- Optional differential / line-driver output
- Positive finger-latching connector
- 2-channel quadrature, TTL squarewave outputs
- 3rd channel index option
- Ball bearing option tracks to 10,000 RPM
- -25 to +100C operating temperature
- Single +5VDC supply

US DIGITAL
1400 NE 136th Avenue
Vancouver, Washington 98684, USA
info@usdigital.com
www.usdigital.com
Local: 360.260.2468
Toll-free: 800.736.0194

Differential



Single-Ended



Environmental

Parameter	Value	Units
Operating Temperature, CPR < 2000	-40 to 100	C
Operating Temperature, CPR ≥ 2000	-25 to 100	C
Vibration (5Hz to 2kHz)	20	G
Electrostatic Discharge, Human Body Model	± 4	kV

Mechanical

Parameter	Sleeve Bushing	Ball Bearing
Max. Acceleration	250000 rad/sec²	250000 rad/sec²
Max. Shaft Speed	100 rpm	10000 rpm
Max. Shaft Torque	0.5 ± 0.2 in-oz 0.3 in-oz (**N**-option)	0.05 in-oz
Max. Shaft Loading	2 lbs. dynamic 20 lbs. static	1 lb.
Bearing Life	> 1000000 revolutions	$L_{10} = (19.3/F_r)^3$ * Where L_{10} = bearing life in millions of revs, and F_r = radial shaft loading in pounds
Weight Single-ended Differential	 1.01 oz. 1.28 oz.	 1.15 oz. 1.42 oz.
Max. Shaft Total Indicated Runout	0.0015 in.	0.0015 in.
Max. Panel Nut Tightening Torque	20 in-lbs	20 in-lbs
Technical Bulletin TB1001 - Shaft and Bore Tolerances		Download

* only valid with negligible axial shaft loading.

Phase Relationship

B leads A for clockwise shaft rotation, and A leads B for counterclockwise rotation viewed from the shaft side of the encoder (*see the EM1 page*).

Single-ended Electrical

- Specifications apply over entire operating temperature range.
- Typical values are specified at Vcc = 5.0Vdc and 25 ° C.
- For complete details, see the EM1 or EM2 product pages.

RoHS COMPLIANT

Parameter	Min.	Typ.	Max.	Units	Conditions
Supply Voltage	4.5	5.0	5.5	V	
Supply Current		27	33	mA	CPR < 500, no load
		54	62	mA	CPR ≥ 500 and <2000, no load
		72	85	mA	CPR ≥ 2000, no load
Low-level Output			0.5	V	I_{OL} = 8mA max., CPR < 2000
			0.5	V	I_{OL} = 5mA max., CPR ≥ 2000
		0.25		V	no load, CPR ≥ 2000
High-level Output	2.0			V	I_{OH} = -8mA max. and CPR < 2000
	2.0			V	I_{OH} = -5mA max. and CPR ≥ 2000
		4.8		V	no load and CPR < 2000
		3.5		V	no load and CPR ≥ 2000
Output Current Per Channel	-8		8	mA	CPR < 2000
	-5		5	mA	CPR ≥ 2000
Output Rise Time		110		nS	CPR < 2000
		50		nS	CPR ≥ 2000, ± 5mA load
Output Fall Time		100		nS	CPR < 2000
		50		nS	CPR ≥ 2000, ± 5mA load

Differential Electrical

- Specifications apply over entire operating temperature range.
- Typical values are specified at Vcc = 5.0Vdc and 25 ° C.
- For complete details, see the EM1 product page.

Parameter	Min.	Typ.	Max.	Units	Conditions
Supply Voltage	4.5	5.0	5.5	V	
Supply Current		29	36	mA	CPR < 500, no load
		57	65	mA	CPR ≥ 500 and < 2000, no load
		73	88	mA	CPR ≥ 2000, no load
Low-level Output		0.2	0.4	V	I_{OL} = 20mA max.
High-level Output	2.4	3.4		V	I_{OH} = -20mA max.
Differential Output Rise/Fall Time			15	nS	

Pin-outs

5-pin Single-ended: (1)

RoHS COMPLIANT

Pin	Description
1	Ground
2	Index
3	A channel
4	+5VDC power
5	B channel

10-pin Differential Standard: (2)

Pin	Description
1	Ground
2	Ground
3	Index-
4	Index+
5	A- channel
6	A+ channel
7	+5VDC power
8	+5VDC power
9	B- channel
10	B+ channel

(1) 5-pin single-ended mating connector isCON-FC5.
(2) 10-pin differential mating connector is CON-FC10.

RoHS COMPLIANT

Ordering Information

S5 - [ ] - [ ] - [ ] - [ ] - [ ]

CPR	Shaft	Index	Output	Torque
32 =	236 = *Metric 6mm diameter shaft*	NE = *No Index*	S = *Single-ended*	D = *Default*
50 =	250 = *1/4" diameter*	IE = *Index*	D = *Differential*	B = *Ball Bearing*
96 =				N = *No torque added*
100 =				
192 =				
200 =				
250 =				
256 =				
360 =				
400 =				
500 =				
512 =				
540 =				
720 =				
900 =				
1000 =				
1024 =				
1250 =				
2000 =				
2048 =				
2500 =				
4000 =				
4096 =				
5000 =				

Notes

- Cables and connectors are not included and must be ordered separately.
- For ordering information please see the Compatible Cables / Connectors section above.
- US Digital warrants its products against defects in materials and workmanship for two years. See complete warranty for details.

US DIGITAL
1400 NE 136th Avenue
Vancouver, Washington 98684, USA
info@usdigital.com
www.usdigital.com
Local: 360.260.2468
Toll-free: 800.736.0194

RoHS COMPLIANT

Base Pricing

Quantity	Price
1	$87.50
5	$64.85
10	$55.87

For volume discounts, please contact us at sales@usdigital.com or 800.736.0194.

- Add 11% per unit for **CPR** of , , , , or
- Add $1.00 per unit for **Shaft** of Metric 6mm diameter shaft
- Add 23% per unit for **Output** of Differential
- Add $5.80 per unit for **Torque** of Ball Bearing
- Add 17% per unit for **Index** of IE or **CPR** greater than or equal to 1000.

US DIGITAL
1400 NE 136th Avenue
Vancouver, Washington 98684, USA
info@usdigital.com
www.usdigital.com
Local: 360.260.2468
Toll-free: 800.736.0194

Rev. 150223122200