Laser Game Report

Alex Wheelock

Robotics and Communications Systems Engineering Technology, Idaho State University

RCET 3375: Advanced Computer Architecture and Embedded Systems Laboratory

Instructor: Shane Slack

December 11, 2024

Table of Contents

Table of Figures

1: Abstract

The Laser Game project is a game that will be used for recruiting purposes for Idaho State University's Robotics and Communications Systems Engineering Technology program. This project has two components, the blaster and the target system. It has two laser blasters that will allow two players to compete against each other, engaging targets for points. Each blaster's laser pulses at a different frequency, allowing IR receivers to determine who shot the target. Targets will randomly enable, and light up to tell the players which target to shoot. Each target shot gives a point for the player who shot it first, and the player with the most points at the end wins.

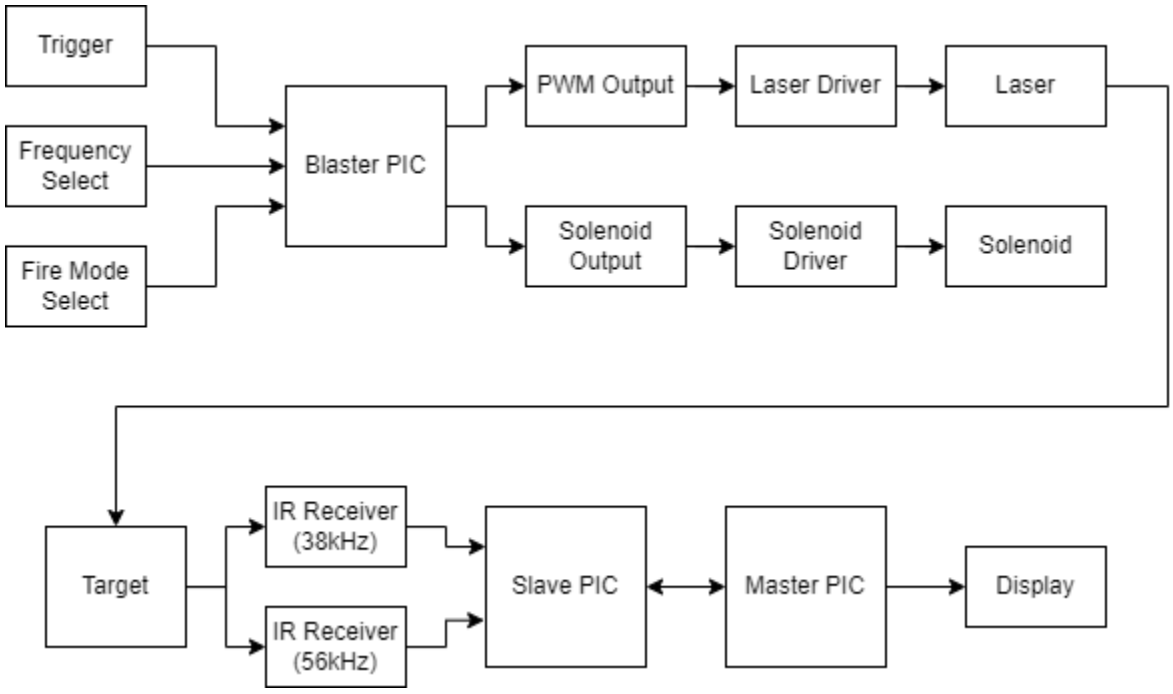

Figure 1.1: Laser Game Block Diagram

2: Background

Though there is no previous work on this project, this project is being designed and built from the ground up in the Fall 2024 semester. However, the Laser Game project is being developed to replace a similar project from the past which was Nerf based, rather than laser based.

3: Proposed Problem

3.1: Overview

In the past, the ISU Robotics program has used motor driven Nerf gun turrets controlled by game controllers (Shown in Figure 3.1.1), however there are some drawbacks to the Laser Game's predecessor. First, the balls that the Nerf guns shoot need to be picked up and the guns need to be reloaded which lead to downtime that the game cannot be played. In response to this, a robot was designed to pick up the shot Nerf balls inside the netted area to reduce the downtime. However, this robot does not currently work. The second issue is that the motors which control the turrets were damaged and also no longer work. The Laser Game is being developed to eliminate the need for cleanup and downtime, and create an improved game for recruiting with better reliability compared to its predecessor.

Image provided by Shane Slack

Figure 3.1.1: Nerf Arcade Project

3.2: Semester Goals

The goals for this semester on this project were to get a base of the project laid out, and develop as much as possible within the given project time of the semester, focusing on the programming. Specific goals included:

1. Design and develop a prototype for the blasters.
2. Get the blasters working so that they are able to trigger different IR receivers.
3. Come up with an idea of how the game will function, and all parts necessary for the project.
4. Design and develop the target circuits, so that the game functions as intended.

4: Blaster

4.1: Blaster Overview

The blaster is the device used by the players to engage targets and score points. The blaster has 4 digital inputs, and 2 digital outputs. The inputs are: a fire switch, a frequency select switch, and a fire-mode select switch. The outputs are: the PWM output which controls the laser, and a solenoid which is energized in tandem with the laser. The laser is driven by the PWM so that it can pulse at the player specific frequencies (38kHz & 56kHz), corresponding to the IR receiver frequencies at the targets on the slave. The solenoid output will be used to simulate the recoil and cycling of the slide as felt on a real pistol.

The software and hardware of the blaster are completed. However, there is currently no PCB designed, nor is there a gun to mount the future PCB and hardware into. The working prototype on a breadboard can be seen below in Figure 4.1.1: Blaster Breadboard Layout. The schematic for the blaster can be found in Figure 10.3: Blaster Circuit Schematic in Appendix A (Schematics). The code for the blaster can be found in Appendix D (Code), and the port bitmaps can be found in Appendix C (Port Bitmaps), tables 1-3.

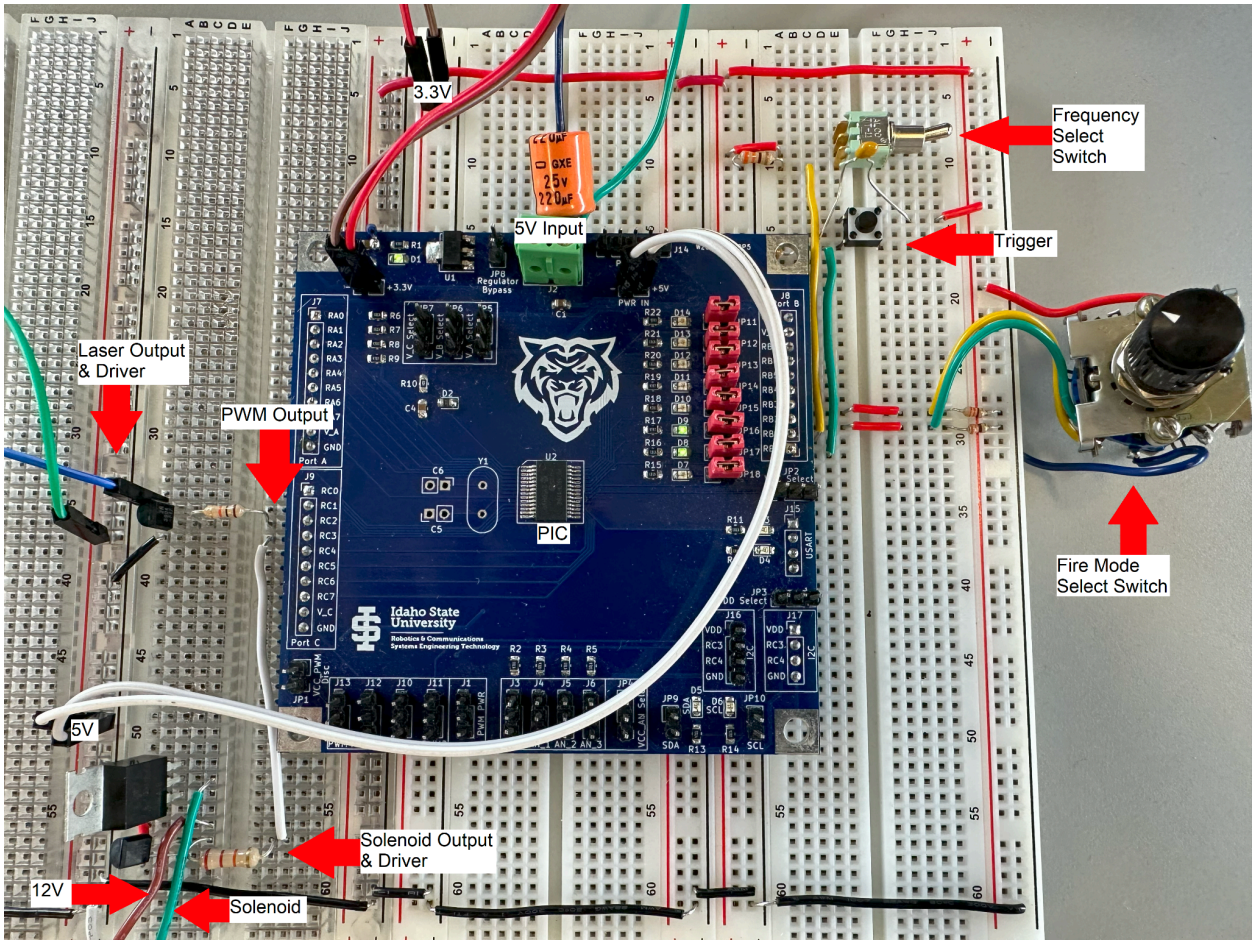

Figure 4.1.1: Blaster Breadboard Layout

4.2: Blaster Circuit Description

4.2.1: Laser Output

The laser used is a simple laser diode with two wires, and a built-in regulator which requires 5V to turn on the laser. The breadboard circuit used a PIC development board which was equipped with a PIC16LF1788 which takes 3.3V, and thus an output driver using a BJT as a switching transistor was used to provide the necessary 5V for the laser. The final product will use the PIC16F1788 to reduce the amount of linear regulators required for the circuit, also reducing the need for an output driver for the laser. The PIC16F1788 is capable of ±50mA, and the laser was measured to pull 26mA, making it safe for the PIC to drive on its own.

The laser output is driven by the PWM output of the PIC's Capture and Compare Module 1 (CCP1) on RC2. CCP1 is used to pulse the laser at the frequencies of 38kHz or 56kHz, which corresponds to the IR receiver frequencies, and are how the target is able to differentiate between the two players. The PWM output is always running but under normal conditions the TRISC bit is set for RC2, disabling the output driver. Once the trigger button (SW1) is pressed on RB0, the laser will turn on, with a frequency of 38kHz or 56kHz as determined by the frequency selector switch (SW2) on RB1, and in 1 of 3 modes (semi-auto, burst-fire, or full-auto) as determined by the fire-mode selector switch (SW3) on RB2 and RB3. Regardless of the laser's PWM frequency, the duty cycle of each frequency has been set to ~95% for maximum brightness of the laser for more power and a longer effective range, while still allowing the receivers to pick up the frequencies.

4.2.2: PWM Frequency Select

The PWM output has two frequency options, 38kHz and 56kHz. The frequency is selected by the frequency selector switch, shown in the top right corner of Figure 4.1.1. The frequency selector switch is a SPST switch connected to RB1 labeled SW2. In the open position RB1 sees a logic low, and the PWM will operate at 38kHz. When closed RB1 sees a logic high, the PWM will operate at 56kHz. See the truth table in Table 4.2.1 below, as well Figures 4.2.2.1 and 4.2.2.2 for waveform captures detailing this relationship.

Table 4.2.1: PWM Frequency Select Truth Table

SW2 Position	RB1	PWM Frequency
OFF	0	38kHz
ON	1	56kHz

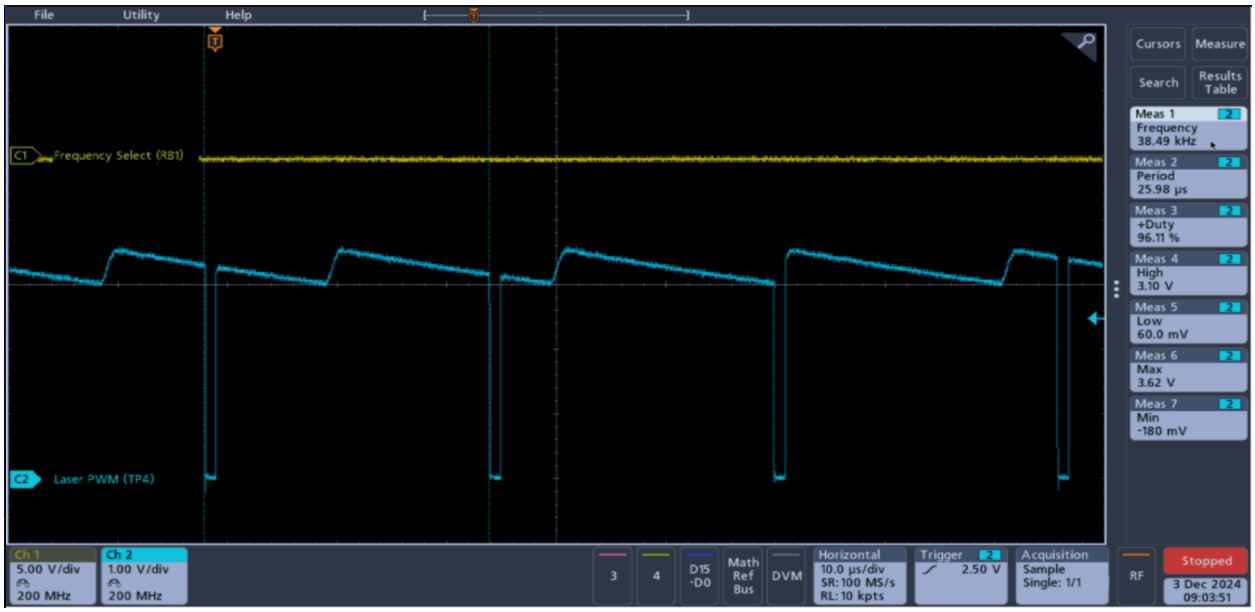

Figure 4.2.1: 38kHz PWM Output Frequency Waveform Capture (RB1 = 0)

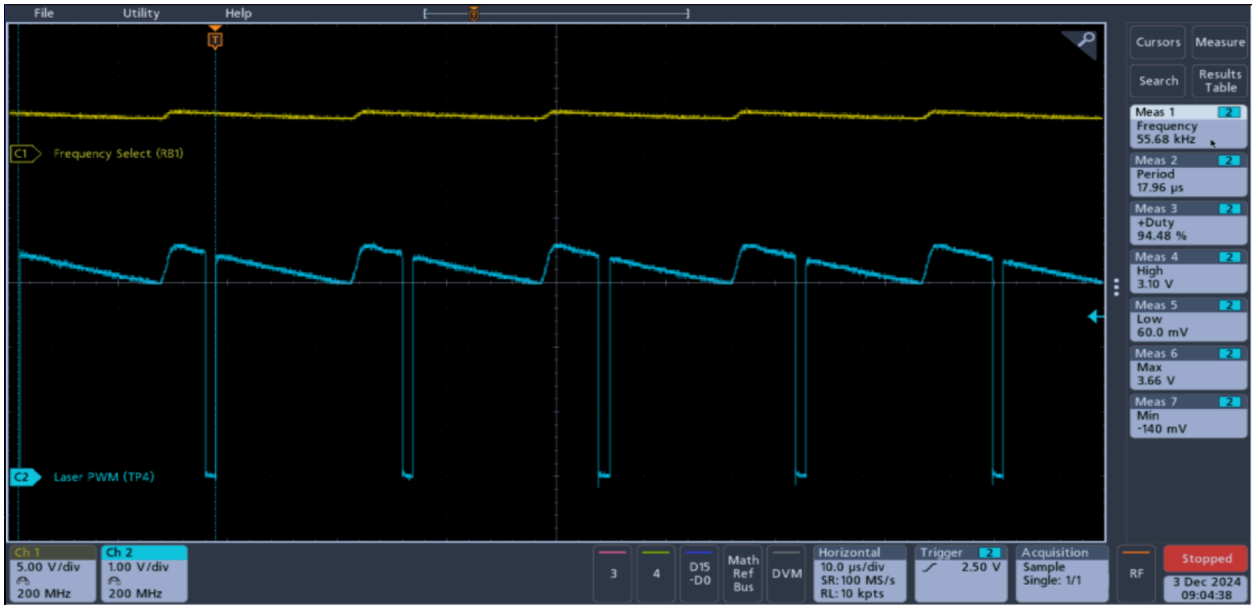

Figure 4.2.2: 56kHz PWM Output Frequency Waveform Capture (RB1 = 1)

4.2.3: Fire Mode Select

There are 3 fire mode options available on the laser blaster: semi-auto, burst-fire, and full-auto.The user can choose between these modes using SW3, which has three positions and will cause a logic high on RB2, RB3, or neither. The truth table for the inputs' correlation with the firing mode can be found in Table 4.2.2 below. The timing for the toggling of the outputs is controlled by Timer 1 of the PIC which takes 65 ms to overflow

The semi-automatic or semi-auto mode will fire one pulse each time that the user presses down the trigger. The semi-auto mode is selected with RB2 and RB3 seeing a logic low. Upon the trigger being pressed, the PWM output driver is enabled to turn the laser on for 65 ms. The timing is handled using the PIC's Timer 1. Upon the trigger being pressed, Timer 1 is reset and the output is enabled. After the 65 ms pulse has ended, as indicated with Timer 1's interrupt flag, the laser will turn off. Regardless of how long the user holds down the trigger, it must be pressed again in order to fire another shot as shown below in Figure 4.2.3.

The burst-fire mode operates the same as the semi-auto mode in principle, with the difference being that it fires 3 shots rather than 1. This mode is selected with RB3 seeing a logic high, with RB2 seeing a logic low. Upon the button being pressed the PWM output driver is enabled for 65 ms, then off for 65 ms, then back on and off until the blaster fires 3 shots. In this mode, a single trigger press will fire 3 shots and no more again until the button is released and then pressed again, as shown in Figure 4.2.4 below.

The third and final firing mode is the fully-automatic, or full-auto mode. This mode is active when RC2 sees a logic high, and RC3 sees a logic low. In this mode, the blaster will continuously toggle the output as was done in the burst-fire mode at 65 ms intervals, but will do

so until the user releases the trigger. So, rather than 3 shots, the user can fire a theoretically infinite amount of times with one trigger press, as shown below in Figure 4.2.5.

Table 4.2.2: Fire Mode Select Truth Table

RB3	RB2	Fire Mode
0	0	Semi-Auto
0	1	Full-Auto
1	0	Burst-Fire
1	1	Unused

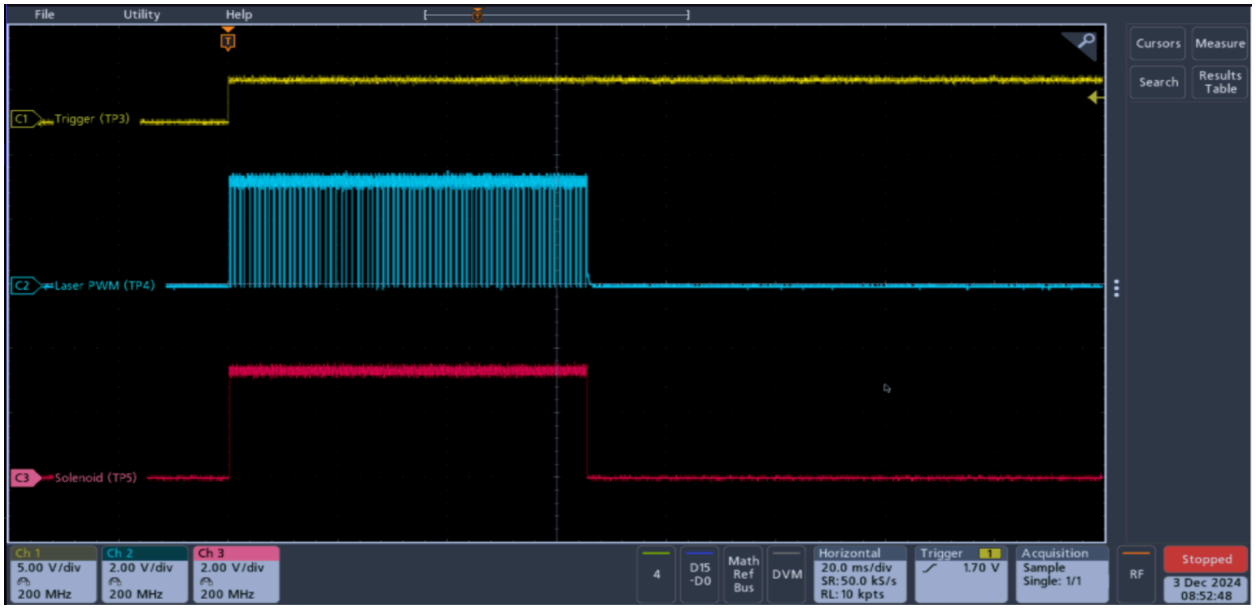

Figure 4.2.3: Semi-Auto Fire Mode Waveform Capture (RB2 = 0, RB3 = 0)

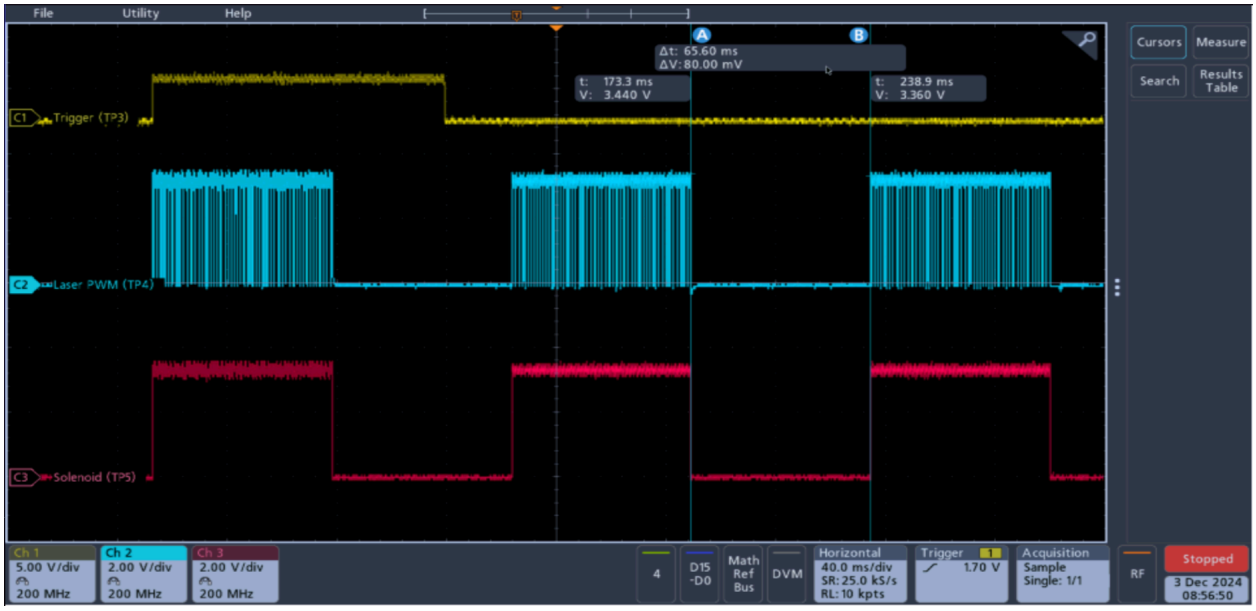

Figure 4.2.4: Burst-Fire Fire Mode Waveform Capture (RB2 = 0, RB3 = 1)

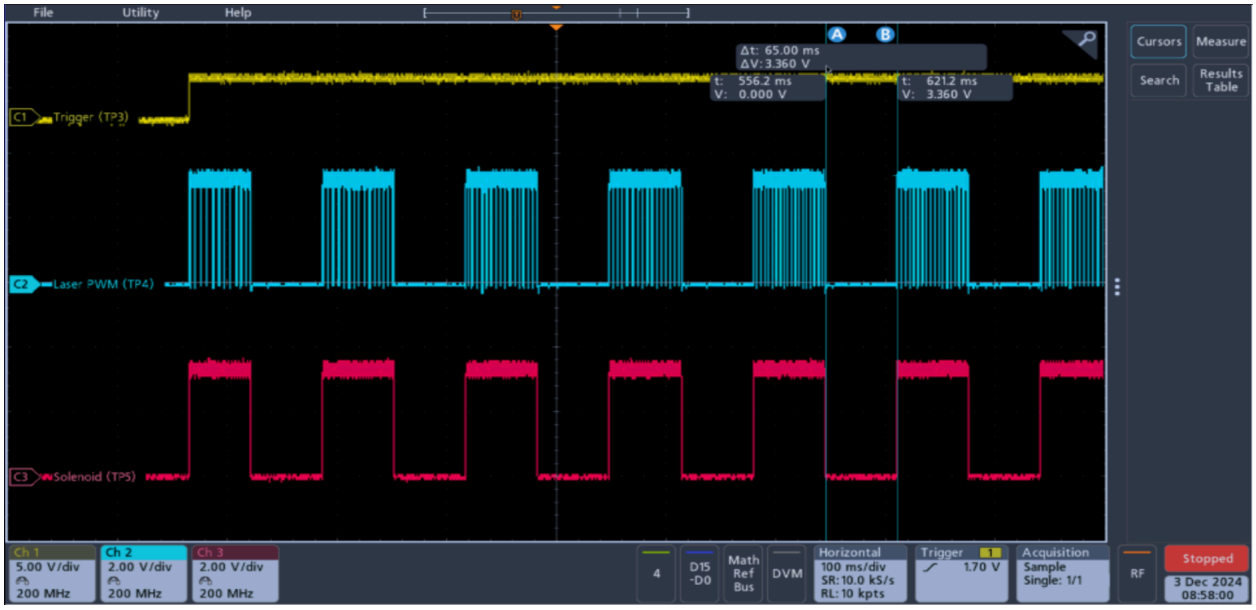

Figure 4.2.5: Full-Auto Fire Mode Waveform Capture (RB2 = 1, RB3 = 0)

4.2.4: Solenoid Output

A solenoid will be used to push a slide back to simulate the slide of a pistol in the final product. In order to control the solenoid, RC3 is being used to energize the coil with a high or low. A logic high being energized and a logic low being de-energized, and the time that the solenoid is energized will correlate directly with the time that the laser is on, as shown above in Figures 4.2.3, 4.2.4, and 4.2.5.

The solenoid being used currently is rated for 12V/300mA. During testing, the solenoid was being driven with 800mA and operating normally without getting hot since it is on 50% of the time in 65 ms intervals. Since the PIC cannot handle 800mA nor can it handle 12V, a switching transistor darlington pair made from a 2N3904 and a TIP41 was used to allow the PIC to control the solenoid with sufficient current and voltage, without damaging itself.

4.3: Blaster Troubleshooting

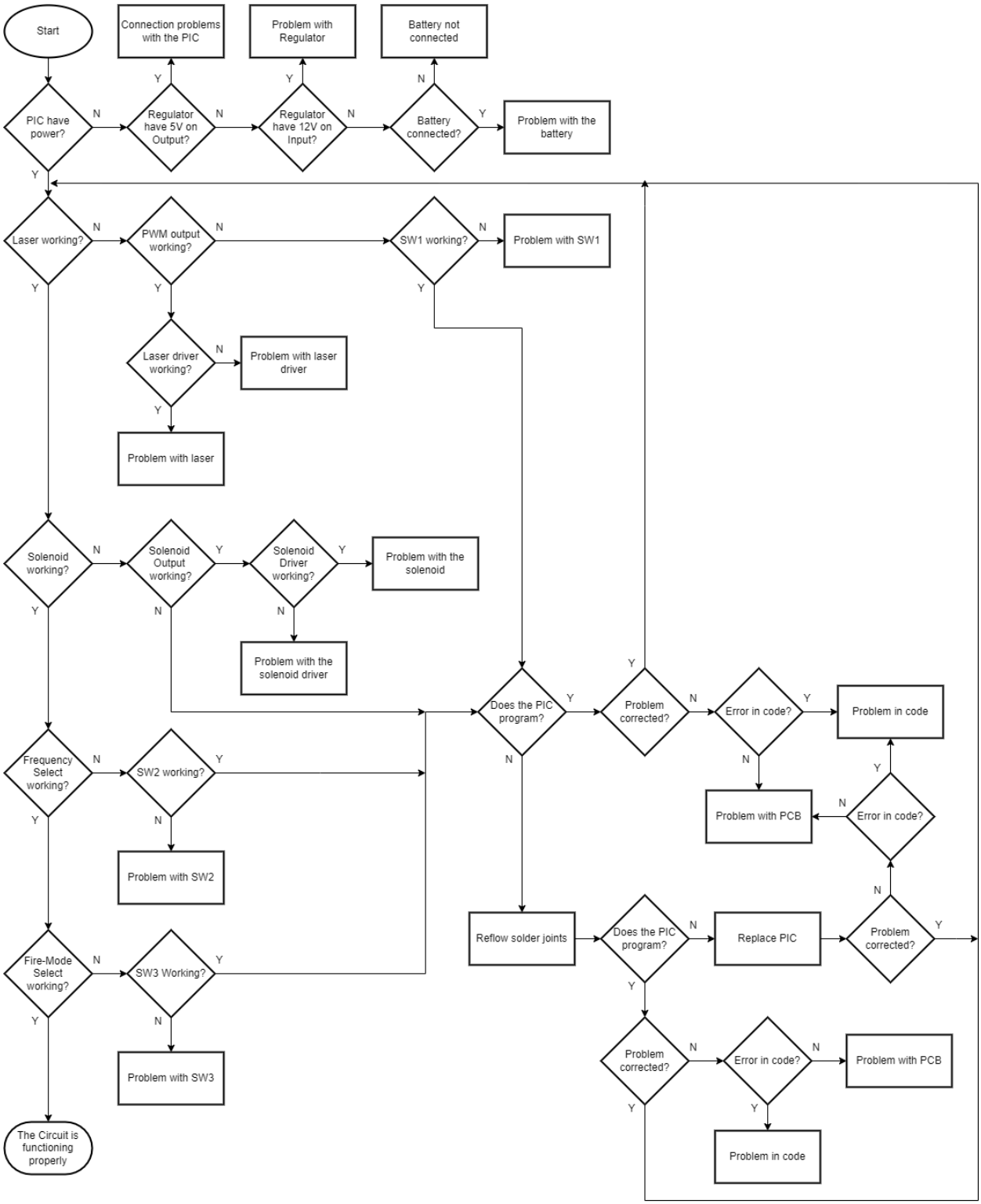

Figure 4.3.1: Troubleshooting Flowchart

To troubleshoot the blaster, follow through the steps below, which follows along with the flowchart shown above in Figure 4.3.1.

1. Is there +5V on pin 20 of the PIC?
    a. If yes, move to step 4.
    b. If not, move to step 2.
2. Is there +5V on TP2?
    a. If yes, there is a problem between the output of the linear regulator and pin 20 of the PIC.
    b. If not, move to step 3.
3. Is there +12V on TP1?
    a. If yes, there is a problem with the linear regulator.
    b. If not, there is a problem with the battery, or its connections to the circuit.
4. When pressing SW1, does the laser turn on?
    a. If yes, move to step 8.
    b. If not, move to step 5.
5. When pressing SW1, does the PWM output on TP7 enable?
    a. If yes, then move to step 6.
    b. If not, then move to step 7.
6. Without pressing SW1, measuring the collector of Q1, is there +5V? Now pressing SW1, does the output change with the PWM enabled?
    a. If yes to both questions, there is a problem with the laser.
    b. If no to either of the questions, there is a problem with Q1.

7. Measuring TP3 without SW1 being pressed, is the input low? Additionally with SW1 pressed, is the input high?
    a. If yes, there is a problem with the PIC, move to step 15.
    b. If no to either of the questions, there is a problem with SW1 and/or the pull-down resistor R1.
8. When pressing SW1, does the solenoid energize as the laser turns on?
    a. If yes, move to step 11.
    b. If not, move to step 9.
9. Measuring TP8, does the output go high as SW1 is pressed?
    a. If yes, move to step 10.
    b. If not, then there is a problem with the PIC, move to step 15
10. Measuring the collector of Q2/Q3, without pressing the SW1 are you measuring 12V? When pressing SW1, does the signal change?
    a. If yes, there is a problem with the solenoid.
    b. If no to either of the questions, there is a problem with the solenoid driver (Q2 and/or Q3).
11. Measuring TP7 and toggling SW2, when pressing SW1 in each position, is the output toggling between 38kHz and 56kHz?
    a. If yes, then move to step 13.
    b. If not, then move to step 12.

12. Measuring TP4, toggle SW2. Is the input a logic low in one position and a logic high in the other position?
    a. If yes, then there is a problem with the PIC, move to step 15
    b. If not, then there is a problem with SW2, and/or its pull-down resistor R2.
13. Does the blaster change between the three firing modes as SW3 is rotated, and SW1 is pressed?
    a. If yes, then the blaster is functioning as intended.
    b. If not, move to step 14.
14. Measuring TP5 and TP6, as you rotate SW3, does it cycle from both low (TP5 = 0, TP6 = 0), one high (TP5 = 0, TP6 = 1), and then the other high (TP5 = 1, TP6 = 0)?
    a. If yes, then there is a problem with the PIC, move to step 15
15. Try reprogramming the PIC, does it program?
    a. If yes, verify functionality starting at step 4. If this still does not work, there is a problem in the program. Fix the code, reprogram, and verify functionality. If it still does not work then there may be something wrong with the board.
    b. If not, move to step 16
16. Reflow the solder to the pins of the PIC to ensure all of the pins are making a good connection to the board, and try reprogramming again. Did it program?
    a. If yes, verify functionality starting at step 4. If this still does not work, there is a problem in the program. Fix the code, reprogram, and verify functionality. If it still does not work then there may be something wrong with the board.
    b. If not, move to step 17.

17. The PIC may be damaged, try replacing the PIC and programming it. Did this work?
    a. If yes, verify functionality starting at step 4. If this still does not work, there is a problem in the program. Fix the code, reprogram, and verify functionality.
    b. If not, then there is something wrong with the board. Check KiCad, fix the issue and try again next semester.

5: Target System

5.1: Target System Overview

The target system will be constructed of two types of boards, a master and a slave board, communicating via I2C. The master board is responsible for enabling targets, keeping track of each player's score, as well as updating the display showing the score to the players. The slave board is the physical target board, which receives the laser from the blaster and determines who scores. The slave board target is disabled until the master enables it. The target is enabled when the indicator LEDs which tell the player which targets are active. Once the enabled target is shot, the slave disables the target as well as the indicator LEDs, and reports which player shot the target on the next master read. There can be up to 128 targets with 7-bit addressing. All slave boards will be daisy-chained together, connected to the master using RJ11 cables which supply +5V, GND, and have SCL and SDA lines for I2C communication. The master constantly polls each slave, occasionally writing to one in order to enable that specific slave's target.

The target system is still in development and has not been tested, however circuits and program flowcharts have been designed to accelerate future development. The schematic for the master can be found in Appendix A: Figure 10.6, the flowchart in Appendix B: Figure 11.2, and the port bitmaps in Appendix C: Tables 12.4-12.6. Additionally the slave schematic for the slave

can be found in Appendix A: Figure 10.7, the flowchart in Appendix B: Figure 11.3, and the port bitmaps in Appendix C: Tables 12.7-12.9.

5.2: Target Master Circuit Description

5.2.1: I2C Communication

The master can communicate with up to 128 slave devices. The slave devices that the master will communicate with is the LK204 LCD display, as well as the target slave PICs. The master will run power, ground, a clock, and data line out to all slave boards via RJ11 connectors. The master will constantly request data back from the slaves while a game is running to determine if a player has scored. If a slave PIC returns no score update, the master will move on to read the next slave device. If a slave does return a score, then the master will then increment that player's score stored in a temporary register, and write to the LCD display to update the displayed player scores.

5.2.1.1: Enabling a Target

Occasionally the Master will enable a target, with the timing determined by Timer 2. Upon TMR2IF being thrown, the master will enable a target by writing the character “E” (0x45) to the slave. Once the master enables a target, it will move on to reading the next slave’s status.

5.2.1.2: Reading a Slave’s Status

When a master reads the status from the slave, it will request one byte of data, to read if a score was reported or not. Currently, only the bits 0 and 1 will be used to indicate if a player scored or not. This can be seen in the truth table below in Table 5.2.1. Upon receiving a byte from a slave that indicates a player has scored, that player’s score is incremented in a general purpose register, and then updates the score on the LCD display, by writing to it through I2C.

Table 5.2.1: Truth Table for Slave Read Byte

7	6	5	4	3	2	1	0	Status
Unused*						0	0	No Player Scored
						0	1	Player 1 Scored
						1	0	Player 2 Scored

*Bits 7:2 of the receive byte from the slave are currently not being used, however there may be an implementation of them in the future to track the time that it takes to hit a target from the time that it turns on, if not then an additional byte may be used for this purpose.

5.3: Target Slave Circuit Description

5.3.1: Overview

The function of the slave is to handle target information, and report the status of the target back to the master using I2C communication. The slave board receives power, GND, and clock and data lines from the master board using RJ11 connectors. Each slave board is equipped with two RJ11 female connectors so that each slave can be connected to the master by daisy-chaining all of the slave boards together, providing each with power and I2C lines from the master.

5.3.2: Target Enable

The target is in standby mode until receiving an “E” (0x45) character from the master via I2C. In standby mode, the player can shoot the target, but will receive no points as the slave ignores all inputs from the receivers until enabled by the master. Upon receiving an “E” from the master, the slave PIC will enable the target in software, and enable RA0 which enables LEDs around the target to indicate to the players that the target is now active. Once a target has been shot, the target and indicator LEDs will be disabled, and a flag will be thrown indicating which of the two players has scored. Then upon the next read from the master, the slave will return a byte telling the master which player scored, and if no player shot the target, it will stay enabled,

and indicate that no player has shot it. The logic for this returned byte to the master can be seen in Table 5.2.1.

5.3.3: Target Reception

The target determines that it has been shot by one of two IR sensors, one that is 38kHz (TSOP32238), and the other that is 56kHz (TSOP53456). Both sensors are active low, and when shot by a laser matching their frequency, their output will drop to 0V which will be picked up using interrupt-on-change on RC0 and RC1 of the slave PIC. The 38kHz is designated for player 1, so that each time the parabolic target is shot by a 38kHz laser, RC0 will see a change on its pin and register it as a hit, and then disable the target. Upon the next read from the master, the slave will then indicate that player 1 has hit the target. The same is true for player 2's 56kHz laser triggering RC1, disabling the target and indicating a player 2 hit on the next master read.

7: Engineering Log

Table 7.1: Alex Wheelock's Project Engineering Log

Date	Category	Time Started	Time Stopped	Duration	Description
11/1/24	Research & Blaster Circuit Schematic Design	10:10 AM	3:00 PM	4:50	Learning how to use CCP for PWM frequency of the laser gun, and designing the circuit in KiCad
11/4/24	Programming Blaster	9:30 AM	3:00 PM	5:30	Trying to get the CCP2 PWM module to work on the PIC16F883
11/5/24	Programming Blaster & Research	11:00 AM	3:00 PM	4:00	Starting to move over to the PIC16F1788, setting it up and doing research
11/7/24	Programming Blaster	11:00 AM	3:00 PM	4:00	Got PWM to work on the 1788, testing the laser
11/7/24	Programming Blaster	11:00 AM	12:00 PM	1:00	Trying to get the laser to fire only when the button is pressed
11/7/24	Programming Blaster	1:00 PM	3:00 PM	2:00	Got the laser to fire only when the button is pressed
11/8/24	Programming Blaster	11:00 AM	3:00 PM	4:00	Got both frequencies for the PWM working, and am now able to toggle the frequency using a switch
11/11/24	Programming Blaster	11:30 AM	3:00 PM	3:30	Working on the different firing modes for the gun
11/12/24	Programming Blaster	11:00 AM	3:30 PM	4:30	Working on different firing modes for the gun, got single shot working
11/13/24	Programming Blaster	11:00 AM	3:00 PM	4:00	Got single shot working better, got burst fire working, got a solenoid hooked up and working
11/14/24	Programming Blaster & Code cleanup	10:00 AM	3:00 PM	5:00	Got the solenoid working with every firing mode, finished controller, worked on cleaning up/commenting code & updating schematic
11/18/24	Documentation	10:30	3:00 PM	4:30	Figuring out how to setup the target

	& Planning target array & scoring	AM			array
12/3/24	Documentation	10:00 AM	3:00 PM	5:00	Getting pictures together and starting on documentation for the project
12/5/24	Documentation	9:00 AM	12:00 PM	3:00	Updating images, designing schematics
12/6/24	Documentation	9:00 AM	3:00 PM	6:00	Doing last flow charts, working on report
12/6/24	Documentation	6:00 PM	7:30 PM	1:30	Working on report
12/7/24	Documentation	12:00 PM	5:30 PM	5:30	Working on report
12/7/24	Documentation	10:00 PM	1:00 AM	3:00	Working on report
12/8/24	Documentation	12:45 PM	3:45 PM	3:00	Working on report
12/8/24	Documentation	6:00 PM	9:00 PM	3:00	Working on report
12/8/24	Documentation	11:15 PM	1:20 AM	2:10	Working on report
12/9/24	Documentation & Presentation	9:45 AM	4:15 PM	6:30	Working on report, starting presentation
12/9/24	Presentation	6:45 PM	11:00 PM	4:15	Working on presentation, practicing presentation
12/10/24	Presentation	9:20 AM	3:00 PM	5:35	Cleaning up presentation, practicing presentation
12/10/24	Presentation	8:00 PM	11:00 PM	3:00	Practicing presentation
12/11/24	Documentation	12:00 PM	5:00 PM	5:00	Cleaning up & finalizing report
				Total Hours Worked:	**103:20:00**

8: Cost Analysis

The cost analysis provides each individual item, item cost, and cost per board for each circuit, as well as the cost of labor. Each part number is a link that will take you to the listing used for pricing. Note that total prices do not include any chassis for the boards, nor does it include PCB pricing itself as the PCBs have yet to be designed. The blaster cost $16.15 to build, see Table 8.1. The target master cost $71.40 to build, with the majority coming from the LCD display, see Table 8.2. The target slave cost $7.09 to build, see Table 8.3. The cost in wages was $5,166.50 with 103.33 hours worked at $50/hr, see Table 8.4.

The total build will require two blasters, one master, and a few dozen slaves for the target array. With 36 slaves/targets, the total build cost would be $330.58 (see Table 8.5). Including wages with this total, the project costs $5,261.14 (see Table 8.6).

Table 8.1: Blaster Bill of Materials

Component Type	Value	Quantity	Part Number	Cost/Part (based on 100 QTY)	Cost/PCB
Capacitor	10uF	2	187-CL21A106MQFNNNE	$0.022	**0.044**
Capacitor	1uF	3	581-KAF21KR71E105KL	$0.12	**0.36**
Capacitor	50pF	1	791-0805N500J250CT	$0.023	**0.023**
LED	Green	1	710-150080GS75000	$0.151	**0.151**
Diode	1N4148	1	637-1N4148W	$0.031	**0.031**
Pin Header	1x5	1	M20-9990546	$0.133	**0.133**
Pin Header	1x2	2	M20-9990246	$0.062	**0.124**
Laser	Laser Diode	1	N/A	$3.001	**3.001**
Transistor	2N3904	2	512-2N3904TA	$0.064	**0.128**
Transistor	TIP41	1	511-TIP41C	$0.449	**0.449**
Resistor	2.2KΩ	1	71-CRCW080522K0JNEAC	$0.01	**0.01**

Resistor	330Ω	4	603-RC0805JR-13330RL	$0.009	**0.036**
Resistor	1kΩ	1	667-ERA-6APB102V	$0.266	**0.266**
Resistor	40kΩ	1	71-CRCW080540K0FKEA	$0.018	**0.018**
Resistor	10kΩ	1	791-RMC1/10K1002FTP	$0.008	**0.008**
Solenoid	12V	1	BM-0530B	$7.49	**7.49**
Switch	Momentary	1	TS-D001	$0.05	**0.05**
Switch	Toggle	1	612-EG1215AA	$0.683	**0.683**
Switch	SP3T	1	2236244	$0.35	**0.35**
Voltage Regulator	AMS1117-5.0	1	926-LM1117MPX5.0NOPB	$0.543	**0.543**
Microcontroller	PIC16F1788	1	579-PIC16F1788T-I/SS	$2.25	**2.25**
				Total Blaster Board Cost:	**$16.15**

Table 8.2: Target Master Bill of Materials

Component Type	Value	Quantity	Part Number	Cost/Part (based on 100 QTY)	Cost/PCB
Capacitor	10uF	2	187-CL21A106MQFNNNE	$0.022	$0.04
Capacitor	1uF	3	581-KAF21KR71E105KL	$0.12	$0.36
LED	Green	1	710-150080GS75000	$0.151	$0.15
Diode	1N4148	1	637-1N4148W	$0.031	$0.03
LCD Display	LK204-25	1	891-LK204-25-E	$67.72	$67.72
Connector	RJ11 Header	1	649-54601-906001WPLF	$0.104	$0.10
Pin Header	1x5	1	M20-9990546	$0.133	$0.13
Resistor	2.2KΩ	1	71-CRCW080522K0JNEAC	$0.01	$0.01
Resistor	10kΩ	1	791-RMC1/10K1002FTP	$0.008	$0.01
Resistor	4.7kΩ	2	603-AC0805DR-074K7L	$0.023	$0.05
Voltage Regulator	AMS1117-5.0	1	926-LM1117MPX5.0NOPB	$0.543	$0.54
Microcontroller	PIC16F1788	1	579-PIC16F1788T-I/SS	$2.25	$2.25
				Total Target Master Board Cost:	**$71.40**

Table 8.3: Target Slave Bill of Materials

Component Type	Value	Quantity	Part Number	Cost/Part (based on 100 QTY)	Cost/PCB
Capacitor	1uF	4	581-KAF21KR71E105KL	$0.12	$0.48
Diode	1N4148	1	637-1N4148W	$0.031	$0.03
LED	Red	16	710-150080RS75000	$0.151	$2.42
Pin Header	1x5	1	M20-9990546	$0.133	$0.13
Connector	RJ11 Header	2	649-54601-906001WPLF	$0.104	$0.21
Transistor	STN851	1	511-STN851-A	$0.453	$0.45
Resistor	10kΩ	1	791-RMC1/10K1002FTP	$0.008	$0.01
Resistor	330Ω	16	603-RC0805JR-13330RL	$0.009	$0.14
Resistor	4.7kΩ	1	603-AC0805DR-074K7L	$0.023	$0.02
IR Receiver	TSOP32238	1	782-TSOP32238	$0.549	$0.55
IR Receiver	TSOP53456	1	78-TSOP53456	$0.39	$0.39
Microcontroller	PIC16F1788	1	579-PIC16F1788T-I/SS	$2.25	$2.25
Total Target Slave Board Cost:					**$7.09**

Table 8.4: Labor Bill of Materials

Employee	Hours Worked	Hourly Pay	Total Cost/Employee
Alex Wheelock	103.33	$50	$5,166.50
Total Labor Cost:			**$5,160.50**

Table 8.5: Project Build Cost

Item	Quantity	Cost	Total Item Cost
Blaster	2	$16.15	$32.30
Target Master	1	$71.40	$71.40
Target Slave	36	$7.09	$255.24
Total Build Cost:	**$330.58**		

Table 8.6: Project Total Cost

Item	Cost
Project Build	$330.58
Labor	$5,160.50
Total Project Cost:	**$5,261.14**

9: Conclusion

This semester, the focus was to get the coding completed for the blaster and target, as well as understand how all of the project's pieces will go together. Unfortunately, within the given project time those goals were not met completely. All features for the blaster, hardware and software were designed and tested, and the blaster is in a functional state. The blaster is able to trigger both receivers which will be used as the targets using only the correct frequency for each receiver. However, the target development did not make it past a schematic and flowchart, meaning that the primary focus next semester will be on building a functional target.

9.1: Recommendations

Recommendations for future development focus are on the target circuitry to develop a functional prototype, design the circuit boards for all circuits, and design and build the physical components of the project.

The primary focus should be on working on designing a functional target, so that the project can start to come together. A base to start from for the target has been laid out, with the schematics for both the master and slave circuits (Figure 10.6 and Figure 10.7), as well as designed flowcharts for each program (Figure 11.2 and Figure 11.3). No code has been written for either the master or slave, so the master and slave programs will need to be written from scratch.

Once a working prototype for all circuits has been created, the rest of the project needs to be completed. a PCB needs to be designed for the blaster, and target master and slave circuits. Then, once the PCBs have been designed, there needs to be a place to mount the PCBs, so the physical components need to be designed. This includes a pistol design for the blaster, whether this be a modified design such as a Nerf gun, or a new design from scratch. A parabolic target

that can mount onto the PCB so that the player does not have to directly shoot the specific IR receiver to score also needs to be created or found. Additionally, there will need to be a place to mount the target PCBs, so a board will need to be cut and have holes cut out, so that the parabolic target and indicator LEDs can poke through and be visible.

10: Appendix A (Schematics)

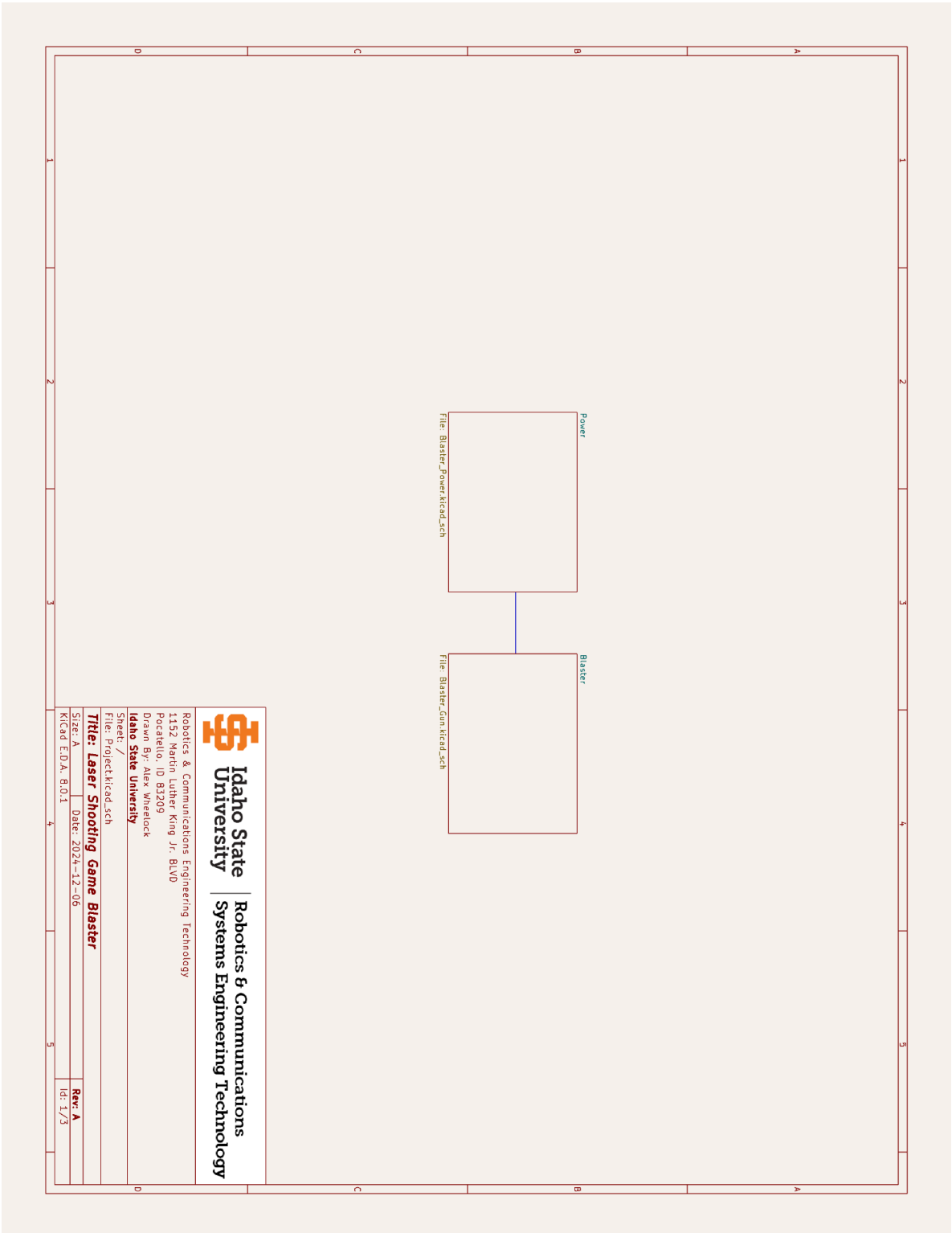

Figure 10.1: Blaster Main Schematic Sheet

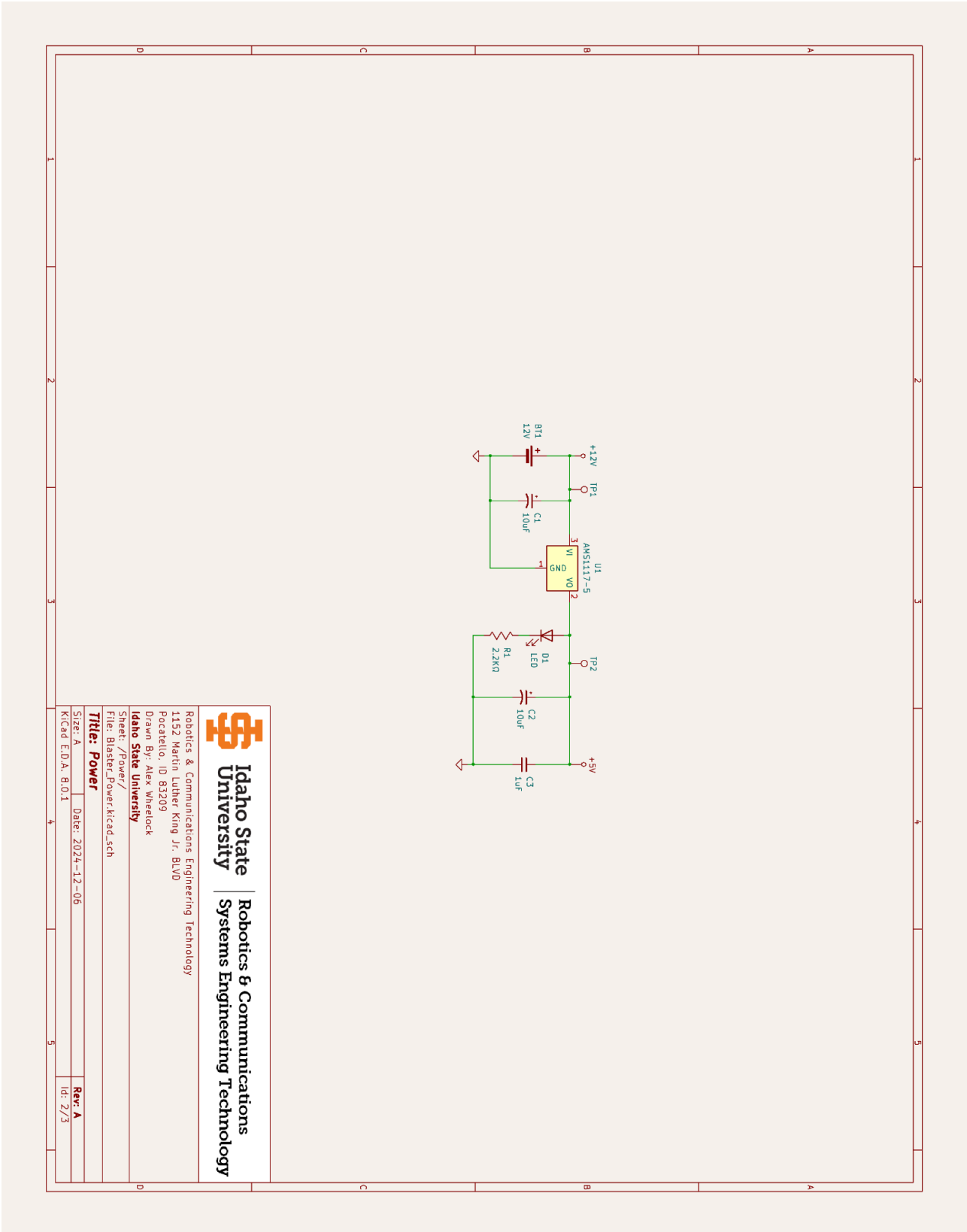

Figure 10.2: Blaster Power Schematic

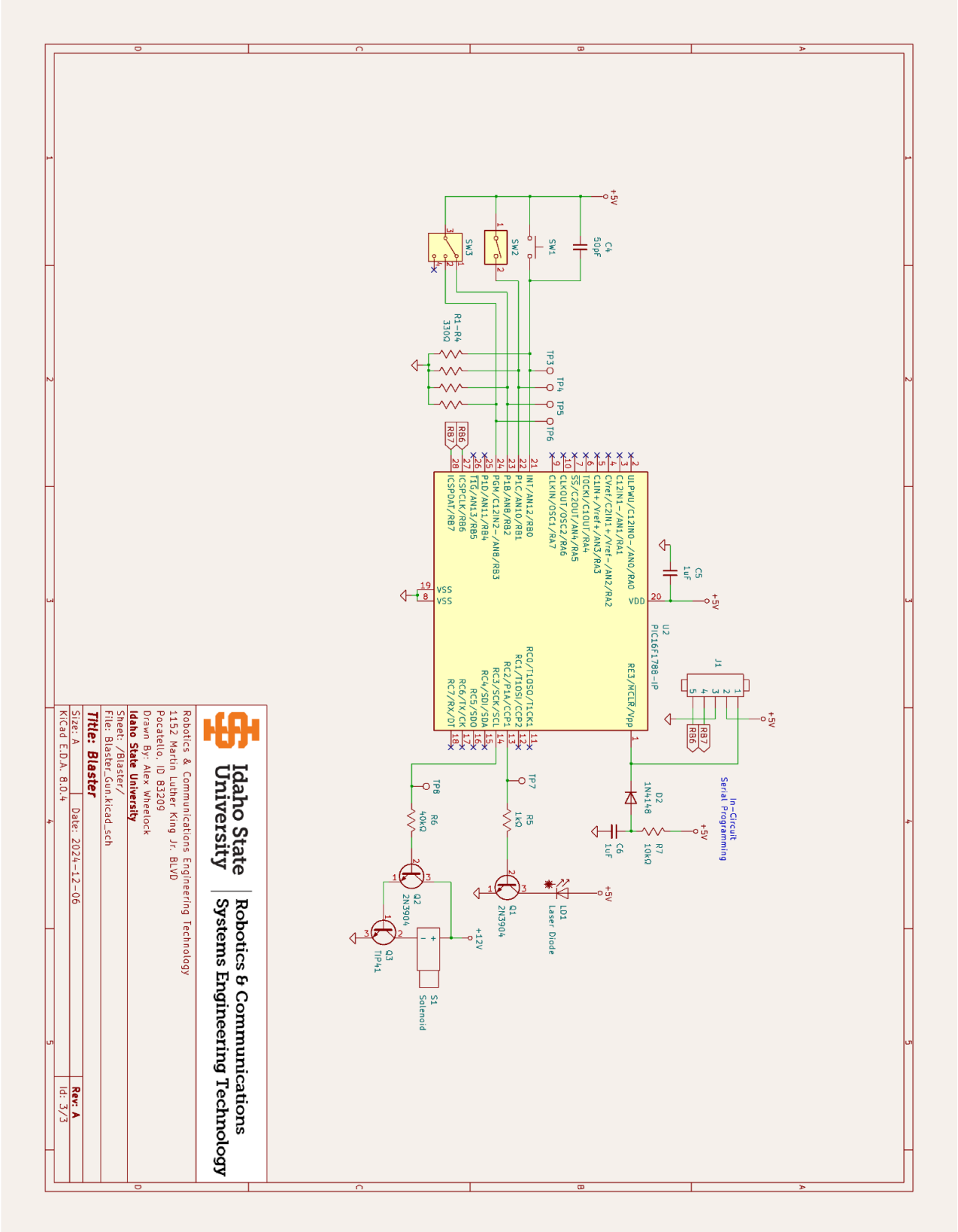

Figure 10.3: Blaster Circuit Schematic

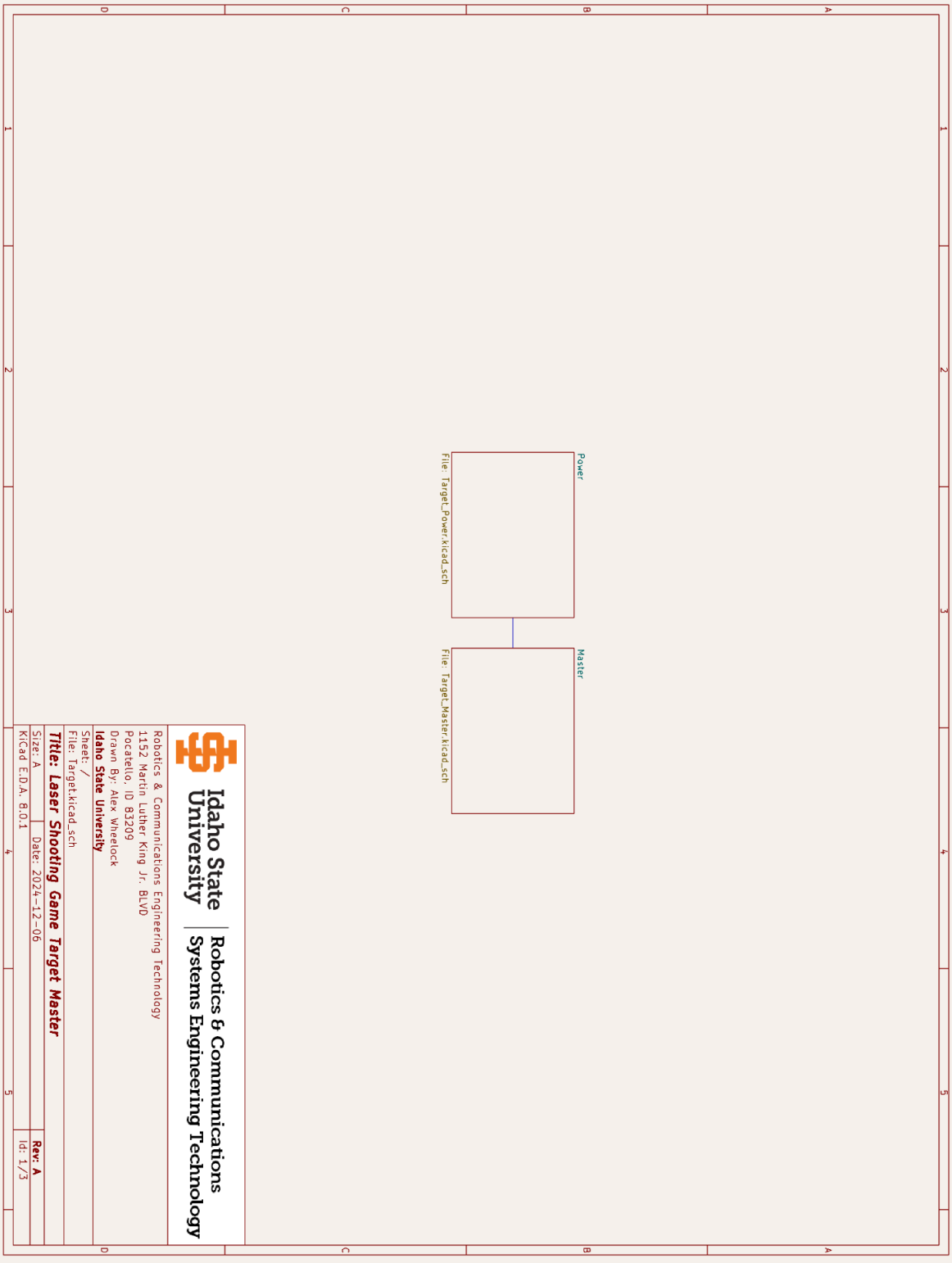

Figure 10.4: Target Master Main Schematic Sheet

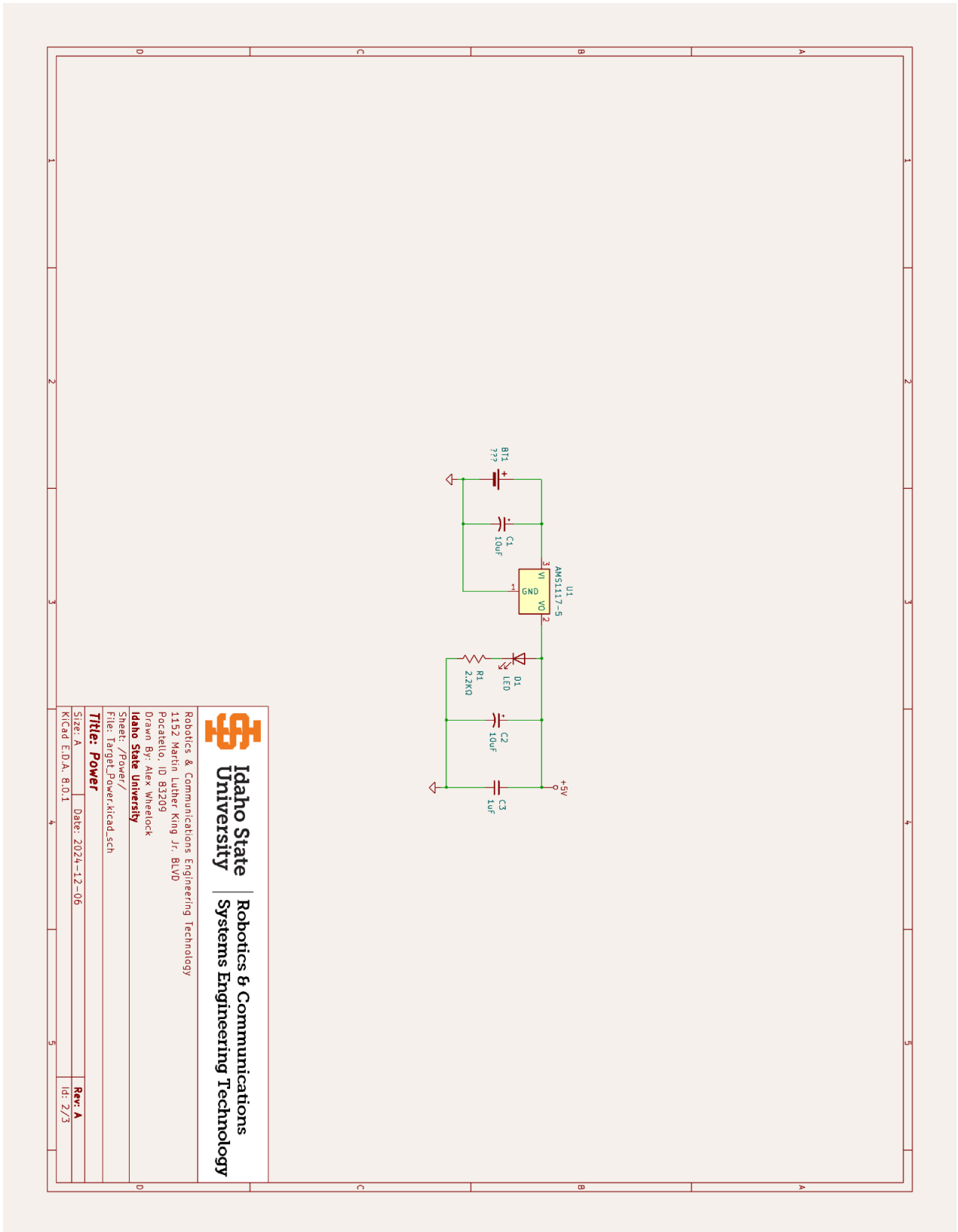

Figure 10.5: Target Master Power Schematic

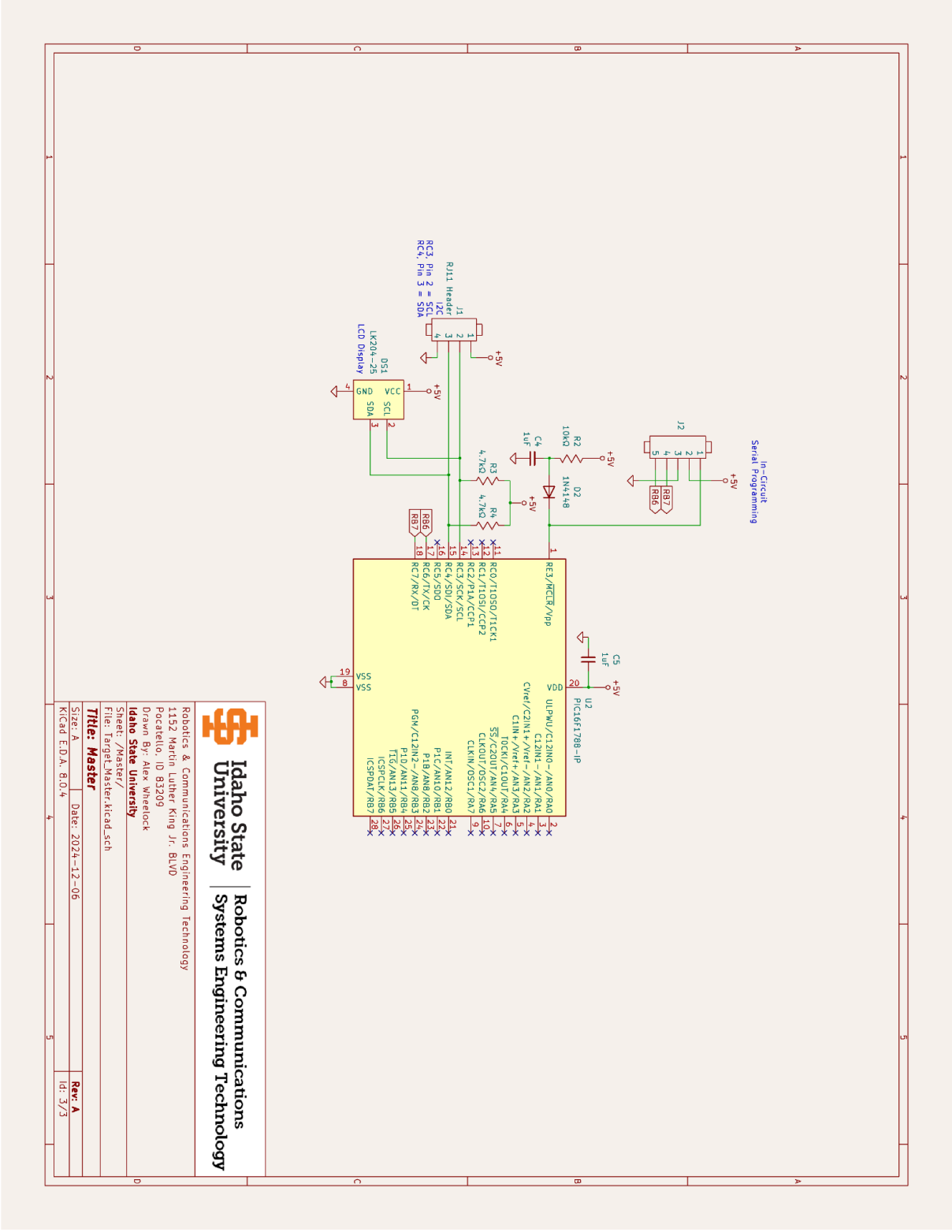

Figure 10.6: Target Master Circuit Schematic*

*Target master circuit has not been tested.

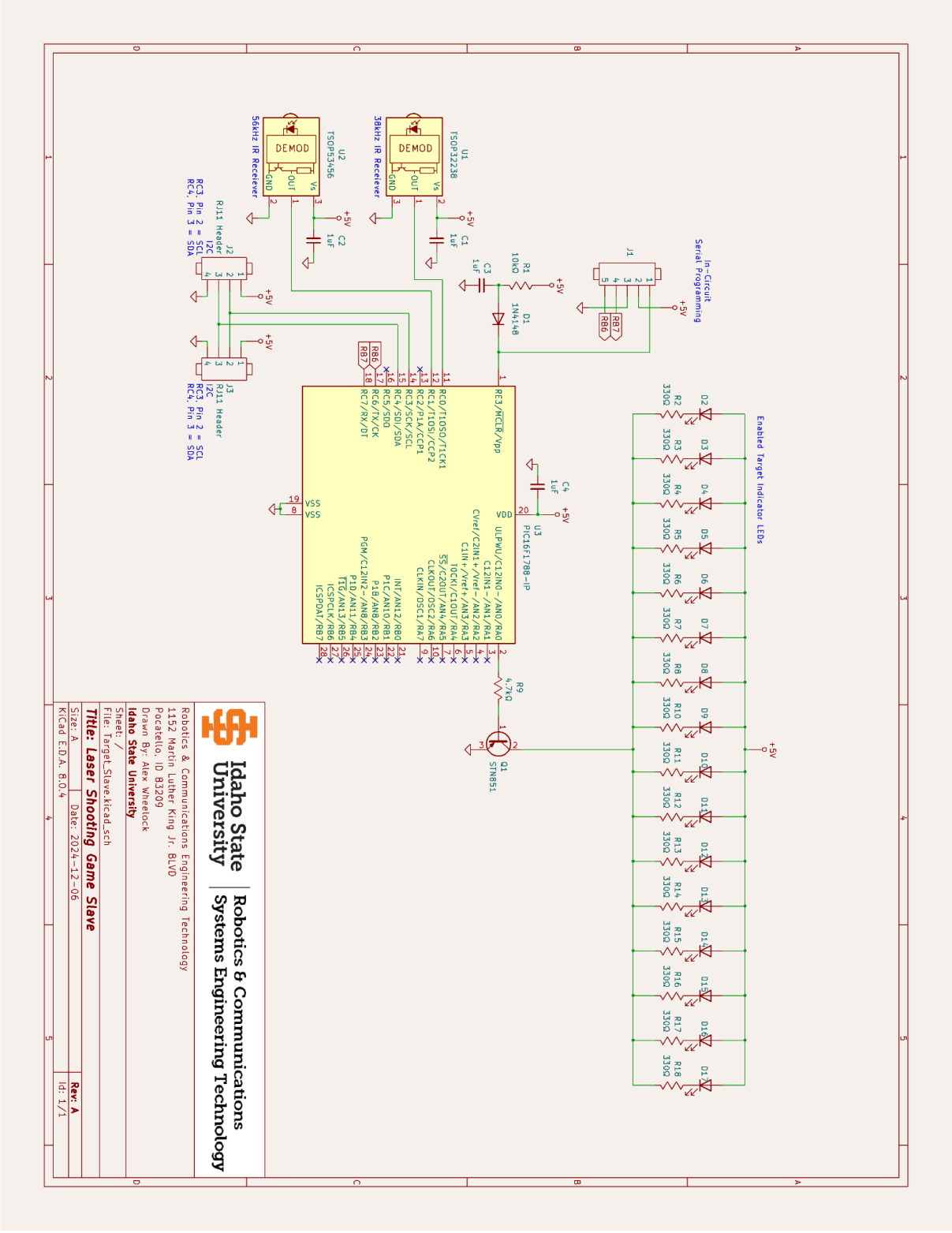

Figure 10.7: Target Slave Circuit Schematic*

*Target slave circuit has not been tested.

11: Appendix B (Flowcharts)

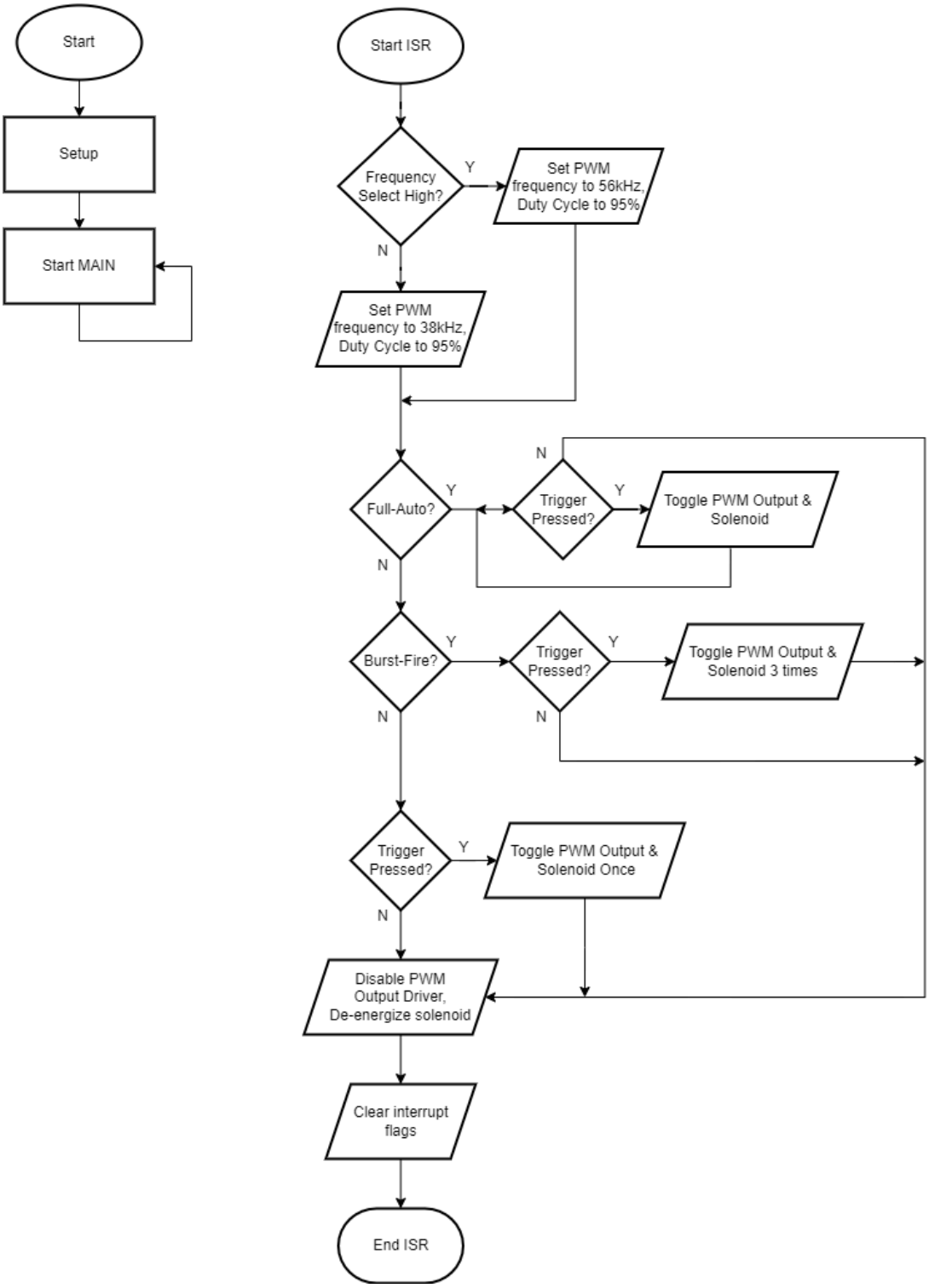

Figure 11.1: Blaster Program Flowchart

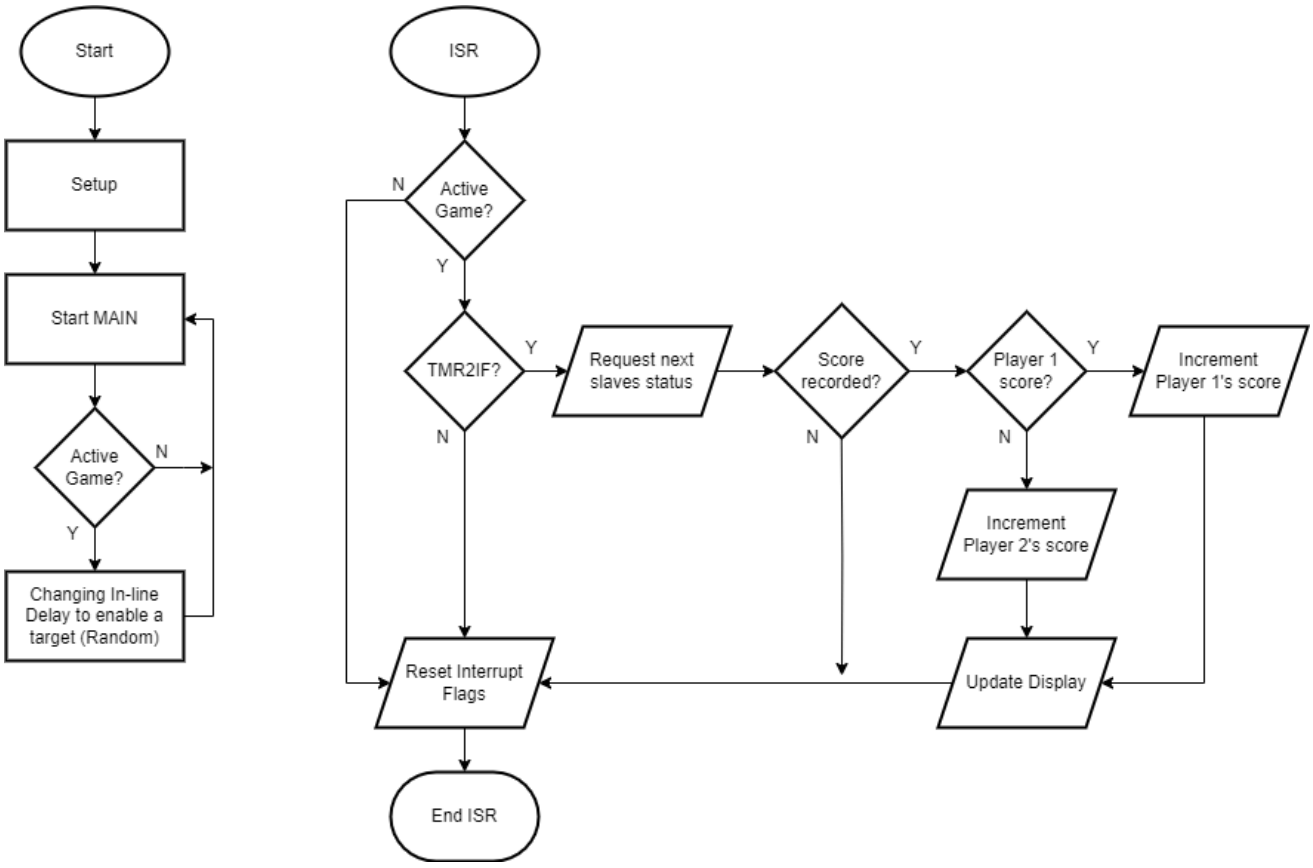

Figure 11.2: Target Master Program Flowchart*

*Target master program has not been written and/or tested.

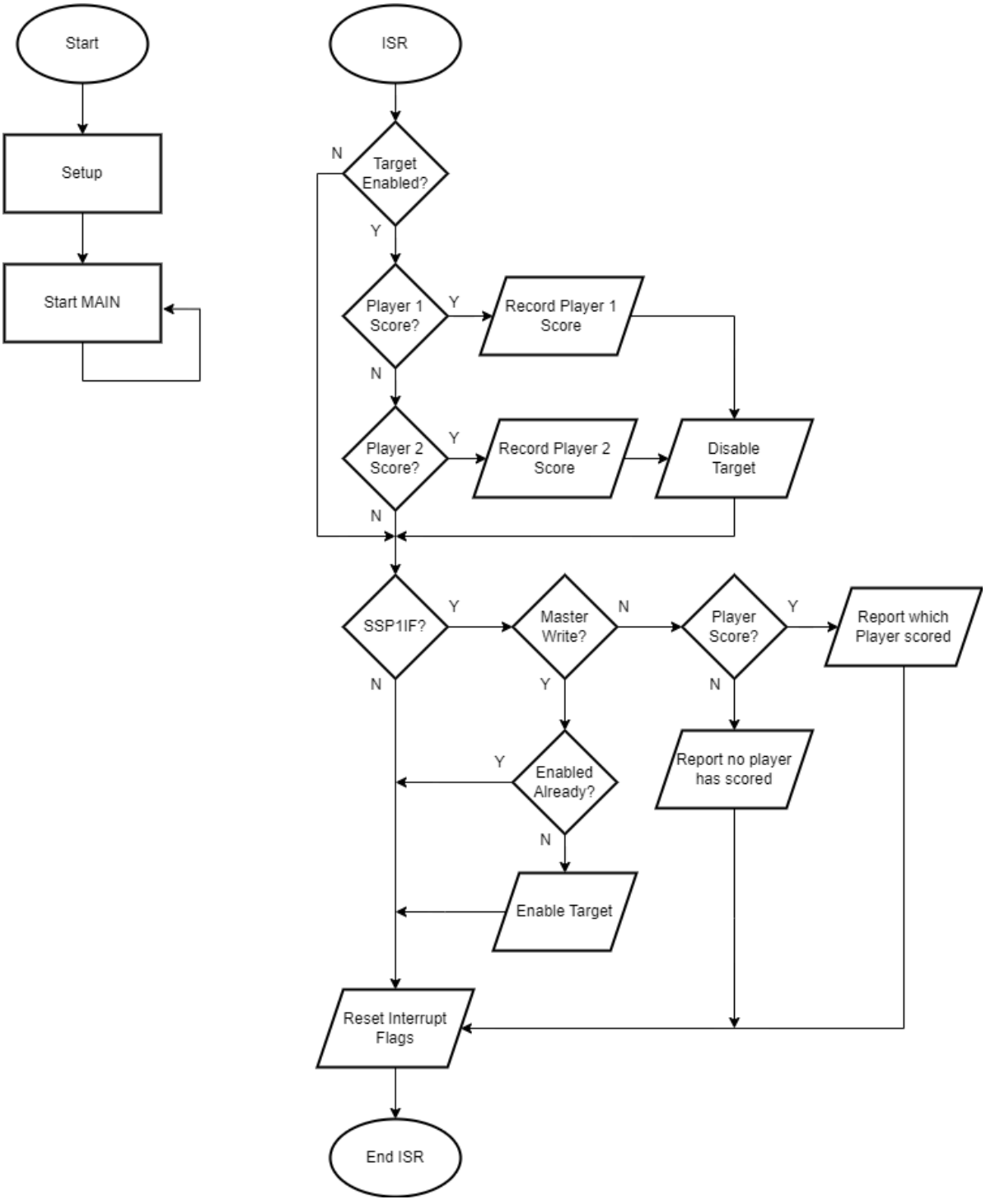

Figure 11.3: Target Slave Program Flowchart*

*Target slave program has not been written and/or tested.

12: Appendix C (Port Bitmaps)

Table 12.1: Blaster PORTA Bitmap

Bit	7	6	5	4	3	2	1	0
Function	Digital Input							
Description	Unused							

Table 12.2: Blaster PORTB Bitmap

Bit	7	6	5	4	3	2	1	0
Function	Digital Input							
Description	Unused				Used For Fire-Mode Select Inputs		Used for Frequency Select Input	Used for Trigger Input

Table 12.3: Blaster PORTC Bitmap

Bit	7	6	5	4	3	2	1	0
Function	Digital Output					*	Digital Output	
Description	Unused				Solenoid Output	PWM Output	Unused	

***** PWM output is constantly changing as the trigger button is pressed, RC2 default is digital input, then upon SW1 being pressed, RC2's output driver is enabled making it a digital output.

Table 12.4: Target Master PORTA Bitmap

Bit	7	6	5	4	3	2	1	0
Function	Digital Input							
Description	Unused							

Table 12.5: Target Master PORTB Bitmap

Bit	7	6	5	4	3	2	1	0
Function	Digital Input							
Description	Unused							

Table 12.6: Target Master PORTC Bitmap

Bit	7	6	5	4	3	2	1	0
Function	Digital Output							
Description	Unused			I2C SDA, for sending/ receiving data from slave devices	I2C SCL, for communi-cating with slave devices	Unused		

Table 12.7: Target Slave PORTA Bitmap

Bit	7	6	5	4	3	2	1	0
Function	Digital Output							
Description	Unused							Controls Target Enabled Indicator LEDs

Table 12.8: Target Slave PORTB Bitmap

Bit	7	6	5	4	3	2	1	0
Function	Digital Input							
Description	Unused							

Table 12.9: Target Slave PORTC Bitmap

Bit	7	6	5	4	3	2	1	0
Function	Digital Output							
Description	Unused			I2C SDA, for sending/ receiving data from the master device	I2C SCL, for communi-cating with the master device	Unused	Used for receiving pulses from 56kHz IR receiver	Used for receiving pulses from 38kHz IR receiver

13: Appendix D (Code)

13.1: Blaster Assembly Code

```
;*******************************************************************************
;                                                                              *
;  Filename:            Project_Gun.asm                                        *
;  Date:        November 5, 2024                                               *
;  File Version:   1                                                           *
;  Author:      Alex Wheelock                                                  *
;  Company:             Idaho State University                                 *
;  Description:    Assembly file for Laser Shooting Game gun. A gun that can fire  *
;                       a laser at two frequencies (38kHz & 56kHz), with firing mode    *
;                       select (single-shot, three round burst, continuous). Different  *
;                       frequencies will be used to determine between player 1 (38kHz)  *
;                       and 2 (56kHz) for scoring. Gun will use 12V (with solenoid), or *
;                       5V (without solenoid), can also use 3.3V if using the LF PIC,   *
;                       with the adjustment of some resistors for the solenoid current. *
;                                                                              *
;                       Uses CCP1 (RC2) for laser PWM, RC3 for solenoid control. RB0 for*
;                       trigger, RB1 for frequency select and RB3:2 used to select from *
;                       the firing modes.                                      *
;                                                                              *
;*******************************************************************************

;*******************************************************************************
;                                                                              *
;  Revision History:                                                           *
;                                                                              *
;  1:          Got everything for the gun working the way that I think it should with    *
;              base features.
;                                                                              *
;*******************************************************************************

                    #include "p16lf1788.inc"            ; processor specific variable definitions
                    #INCLUDE <16LF1788_SETUP.inc>               ; Custom setup file for the PIC16F883 micro-controller
                    #INCLUDE <SUBROUTINES.inc>                  ; File containing all used subroutines
                    LIST      p=16lf1788                ; list directive to define processor
                    errorlevel -302,-207,-305,-206,-203         ; suppress "not in bank 0" message,  Found label after column 1,
                                                                ; Using default destination of 1 (file),  Found call to macro in
column 1

                    ; CONFIG1
; __config 0xC9A4
 __CONFIG _CONFIG1, _FOSC_INTOSC & _WDTE_OFF & _PWRTE_OFF & _MCLRE_OFF & _CP_OFF & _CPD_OFF & _BOREN_OFF
& _CLKOUTEN_OFF & _IESO_OFF & _FCMEN_OFF
; CONFIG2
; __config 0xDFFF
 __CONFIG _CONFIG2, _WRT_OFF & _PLLEN_ON & _STVREN_ON & _BORV_LO & _LPBOR_OFF & _LVP_OFF



;*****************************************
;Interrupt Vectors
;*****************************************
                    ORG         H'000'                  ;BEGINNING OF CODE
                    GOTO        SETUP                   ;
                    ORG         H'004'                  ;INTERRUPT LOCATION
                    GOTO        INTERRUPT                       ;INTERRUPT OCCURRED, RUN THROUGH ISR
```

```
;*****************************************
;SETUP ROUTINE
;*****************************************
SETUP
                CALL    INITIALIZE                      ;CALL SETUP INCLUDE FILE
                GOTO    MAIN                    ;START MAIN CODE

;*****************************************
;INTERRUPT SERVICE ROUTINE
;*****************************************
INTERRUPT
                BANKSEL         PORTB
                BTFSC   PORTB,1                         ;TEST FREQUENCY SELECT SWITCH
                GOTO    SET_56KHZ                       ;RB1 IS HIGH, SET PWM FREQUENCY TO 56kHz
                GOTO    SET_38KHZ                       ;RB1 IS LOW, SET PWM FREQUENCY TO 38kHz

        TEST_MODE
                BANKSEL         PORTB
                BTFSC   PORTB,2                         ;DETERMINE IF IN CONTINUOUS/FULL-AUTO MODE
                GOTO    FIRE_CONTINUOUS                 ;GUN IS IN CONTINUOUS/FULL-AUTO MODE, CHECK IF
READY TO FIRE
                BTFSC   PORTB,3                         ;GUN IS NOT IN CONTINUOUS/FULL-AUTO MODE, TEST
IF IT IS IN BURST-FIRE MODE
                GOTO    FIRE_BURST                      ;GUN IS IN BURST-FIRE MODE, CHECK IF READY TO FIRE
                GOTO    FIRE_SEMI                       ;GUN IS NOT IN CONTINUOUS OR BURST-FIRE MODE,
DEFAULT TO SEMI-AUTO MODE, TEST IF READY TO FIRE
        GOBACK
                BANKSEL         IOCBF
                CLRF    IOCBF                   ;CLEAR PORTB IOC FLAGS
                BCF     INTCON,IOCIF            ;CLEAR IOCIF
                BANKSEL         PIR1
                BCF     PIR1,0                  ;CLEAR TMR1IF
                RETFIE                                  ;RETURN TO MAIN, RE-ENABLE GIE

;*****************************************
;Main Code
;*****************************************
MAIN
                GOTO    MAIN                    ;ENDLESS LOOP, NOTHING HAPPENS (WAITING FOR INTERRUPT)
END

;*****************************************
;END PROGRAM DIRECTIVE
;*****************************************
```

13.2: Blaster Subroutine Assembly Code

```
;*******************************************************************************
;                                                                              *
; Filename:             SUBROUTINES.inc                                        *
; Date:       November 5, 2024                                                 *
; File Version:  1                                                             *
; Author:    Alex Wheelock                                                     *
; Company:              Idaho State University                                 *
; Description:   Contains all subroutines needed for the Laser Shooting Game  *
;                       project. Subroutines include the ability to swap between the  *
;                       two player PWM frequencies of 56kHz and 38kHz, and the ability *
;                       to shoot in three different modes (single-shot, burst,      *
;                       & continuous), to move a solenoid and shoot           the laser with the *
;                       press of a button.                                     *
;                                                                              *
;*******************************************************************************

;*******************************************************************************
;                                                                              *
; Revision History:                                                            *
;                                                                              *
; 1:        Initialize file, get everything working the way that I thought it should   *
;           work.                                                              *
                                                    *
;                                                                              *
;*******************************************************************************

SUBROUTINES_CODE CODE

  FIRE_CONTINUOUS
                BANKSEL         PORTB
                BTFSS   PORTB,0                         ;TEST IF THE TRIGGER IS BEING HELD DOWN
                GOTO    STOP_FIRING                     ;TRIGGER IS NOT BEING HELD DOWN, STOP FIRING
                BANKSEL         TRISC
                BTFSC   TRISC,2                         ;TEST IF OUTPUT DRIVER FOR PWM IS SET OR NOT (FOR
ALTERNATING BETWEEN SET/RESET)
                GOTO    FIRE                    ;OUTPUT DRIVER WAS NOT ENABLED, ENABLE IT
                GOTO    _RESET                          ;OUTPUT DRIVER WAS ENABLED, DISABLE IT

        FIRE
                BCF     TRISC,2                         ;ENABLE OUTPUT DRIVER FOR PWM
                BANKSEL         PORTC
                BSF     PORTC,3                         ;ENERGIZE THE SOLENOID
                GOTO    CONTINUOUS_CHECK_TIMER          ;POLL TIMER 1 FOR TIMING
        _RESET
                BSF     TRISC,2                         ;DISABLE OUTPUT DRIVER FOR PWM
                BANKSEL         PORTC
                BCF     PORTC,3                         ;DE-ENERGIZE THE SOLENOID
                GOTO    CONTINUOUS_CHECK_TIMER
        CONTINUOUS_CHECK_TIMER
                BCF     PIR1,0                  ;CLEAR TMR1IF
        POLL_TMR1_CONTINUOUS
                BTFSS   PIR1,0                  ;POLL TMR1IF UNTIL SET, THEN CONTINUE THE CYCLE
                GOTO    POLL_TMR1_CONTINUOUS            ;TMR1 NOT DONE
                GOTO    FIRE_CONTINUOUS                 ;TMR1 DONE, REPEAT CYCLE

  STOP_FIRING
                BANKSEL         TRISC
```

```
                BTFSS   TRISC,2                 ;DETERMINE IF PWM OUTPUT DRIVER IS ALREADY
DISABLED
                BSF     TRISC,2                 ;OUTPUT DRIVER NOT DISABLED, DISABLE OUTPUT
DRIVER
                BANKSEL         PORTC
                BCF     PORTC,3                 ;NO MATTER WHAT, VERIFY THAT THE SOLENOID IS
DE-ENERGIZED
                GOTO    GOBACK                  ;LEAVE ISR

  FIRE_SEMI
                BANKSEL         IOCBF
                BTFSS   IOCBF,0                 ;DETERMINE IF A TRIGGER PRESS WAS RECORDED
                GOTO    GOBACK                  ;DO NOTHING IF NO TRIGGER PRESS
                BANKSEL         PORTB
                BTFSS   PORTB,0                 ;VERIFY THAT THE BUTTON WAS PRESSED, AND THAT
THE INTERRUPT WAS NOT FROM A TRIGGER RELEASE
                GOTO    STOP_FIRING             ;TRIGGER WAS BEING RELEASED FOR THE INTERRUPT,
CANCEL EVERYTHING
                BANKSEL         TRISC
                BCF     TRISC,2                 ;ENABLE OUTPUT DRIVER
                BANKSEL         PORTC
                BSF     PORTC,3                 ;ENERGIZE SOLENOID
                CLRF    TMR1L           ;/RESET TMR1 FOR FULL TIMING
                CLRF    TMR1H           ;\
                BCF     PIR1,0          ;CLEAR TMR1IF
        POLL_TMR1_SEMI
                BTFSS   PIR1,0          ;POLL TMR1IF UNTIL COMPLETE
                GOTO    POLL_TMR1_SEMI          ;TMR1 NOT DONE
                BANKSEL         PORTC
                BCF     PORTC,3                 ;TMR1 DONE, DE-ENERGIZE SOLENOID
                GOTO    STOP_FIRING             ;STOP FIRING

  FIRE_BURST
                BANKSEL         IOCBF
                BTFSS   IOCBF,0                 ;DETERMINE IF A TRIGGER PRESS WAS RECORDED
                GOTO    GOBACK                  ;DO NOTHING IF NO TRIGGER PRESS
                BANKSEL         PORTB
                BTFSS   PORTB,0                 ;VERIFY THAT THE BUTTON WAS PRESSED, AND THAT
THE INTERRUPT WAS NOT FROM A TRIGGER RELEASE
                GOTO    STOP_FIRING             ;TRIGGER WAS BEING RELEASED FOR THE INTERRUPT,
CANCEL EVERYTHING
                BANKSEL         TRISC
                BCF     TRISC,2                 ;ENABLE OUTPUT DRIVER FOR PWM
                BANKSEL         PORTC
                BSF     PORTC,3                 ;ENERGIZE THE SOLENOID
                CLRF    TMR1L           ;/RESET TMR1 FOR FULL TIMING
                CLRF    TMR1H           ;\
                BCF     PIR1,0          ;CLEAR TMR1IF
        POLL_TMR1_BURST1
                BTFSS   PIR1,0          ;POLL TMR1
                GOTO    POLL_TMR1_BURST1        ;TMR1 NOT DONE
                BANKSEL         TRISC
                BSF     TRISC,2                 ;TMR1 IS DONE, DISABLE PWM OUTPUT DRIVER FOR
RESET
                BANKSEL         PORTC
                BCF     PORTC,3                 ;DE-ENERGIZE SOLENOID
                BCF     PIR1,0          ;CLEAR TMR1IF
        POLL_TMR1_BURST2
                BTFSS   PIR1,0          ;POLL TMR1
                GOTO    POLL_TMR1_BURST2
                BANKSEL         TRISC
```

```
                BCF     TRISC,2                 ;ENABLE OUTPUT DRIVER FOR PWM (SECOND SHOT)
                BANKSEL         PORTC
                BSF     PORTC,3                 ;ENERGIZE THE SOLENOID
                BCF     PIR1,0                  ;CLEAR TMR1IF
        POLL_TMR1_BURST3
                BTFSS   PIR1,0                  ;POLL TMR1
                GOTO    POLL_TMR1_BURST3
                BANKSEL         TRISC
                BSF     TRISC,2                 ;TMR1 IS DONE, DISABLE PWM OUTPUT DRIVER FOR
RESET
                BANKSEL         PORTC
                BCF     PORTC,3                 ;DE-ENERGIZE SOLENOID
                BCF     PIR1,0                  ;CLEAR TMR1IF
        POLL_TMR1_BURST4
                BTFSS   PIR1,0                  ;POLL TMR1
                GOTO    POLL_TMR1_BURST4
                BANKSEL         TRISC
                BCF     TRISC,2                 ;ENABLE OUTPUT DRIVER FOR PWM (THIRD/FINAL
SHOT)
                BANKSEL         PORTC
                BSF     PORTC,3                 ;ENERGIZE THE SOLENOID
                BCF     PIR1,0                  ;CLEAR TMR1IF
        POLL_TMR1_BURST5
                BTFSS   PIR1,0                  ;POLL TMR1
                GOTO    POLL_TMR1_BURST5
                BCF     PORTC,3                 ;DE-ENERGIZE SOLENOID
                GOTO    STOP_FIRING


  SET_38KHZ
                BANKSEL         CCPR1L
                MOVLW   0x19                    ;SET PW TIME FOR 95%DC
                MOVWF   CCPR1L
                BANKSEL         PR2
                MOVLW   0x19                    ;SET PR2 TO 25 FOR 38kHz PERIOD
                MOVWF   PR2
                BANKSEL         TMR2
                CLRF    TMR2                    ;RESET TMR2
                GOTO    TEST_MODE
  SET_56KHZ
                BANKSEL         CCPR1L
                MOVLW   0x11                    ;SET PW TIME FOR 95%DC
                MOVWF   CCPR1L
                BANKSEL         PR2
                MOVLW   0x11                    ;SET PR2 TO 17 FOR 56kHz PERIOD
                MOVWF   PR2
                BANKSEL         TMR2
                CLRF    TMR2                    ;RESET TMR2
                GOTO    TEST_MODE
```

13.3: Blaster Setup Assembly Code

```
;*******************************************************************************
;                                                                              *
;  Filename:            16LF1788_SETUP.inc                                     *
;  Date:        November 5, 2024                                               *
;  File Version:  1                                                            *
;  Author:    Alex Wheelock                                                    *
;  Company:             Idaho State University                                 *
;  Description:   Firmware for setting up a PIC16LF1788 for the laser gun of the  *
;                       Laser Shooting Game project.                           *
;                                                                              *
;*******************************************************************************

;*******************************************************************************
;                                                                              *
;  Revision History:                                                           *
;                                                                              *
;  1:       Initialize file, setup how I think it should be setup for the project.      *
;                                                                              *
;*******************************************************************************

;================================================================================================================
==========================*
;                                                                PORTA BITMAP
                                *
;================================================================================================================
==========================*
;               |               |               |               |               |               |
        |               *
;       7       |       6       |       5       |       4       |       3       |       2       |
1       |       0       *
;---------------|---------------|---------------|---------------|---------------|---------------|---------------*
;DIGITAL |DIGITAL |DIGITAL |DIGITAL |DIGITAL |DIGITAL |DIGITAL |DIGITAL *
;INPUT          |INPUT          |INPUT          |INPUT          |INPUT          |INPUT          |INPUT
        |INPUT          *
;---------------|---------------|---------------|---------------|---------------|---------------|---------------*
;UNUSED         |UNUSED         |UNUSED         |UNUSED         |UNUSED         |UNUSED         |UNUSED
        |UNUSED         *
;               |               |               |               |               |               |
        |               *
;               |               |               |               |               |               |
        |               *
;               |               |               |               |               |               |
        |               *
;----------------------------------------------------------------------------------------------------------------*

;================================================================================================================
==========================*
;                                                                PORTB BITMAP
                                *
;================================================================================================================
==========================*
;               |               |               |               |               |               |
        |               *
;       7       |       6       |       5       |       4       |       3       |       2       |
1       |       0       *
;---------------|---------------|---------------|---------------|---------------|---------------|---------------*
;DIGITAL |DIGITAL |DIGITAL |DIGITAL |DIGITAL |DIGITAL |DIGITAL |DIGITAL *
```

```
;INPUT          |INPUT          |INPUT          |INPUT          |INPUT          |INPUT          |INPUT
        |INPUT          *
;--------------|--------------|--------------|--------------|--------------|--------------|--------------*
;UNUSED         |UNUSED         |UNUSED         |UNUSED         |INPUT USED FOR |INPUT USED FOR |INPUT
USED FOR        |INPUT USED FOR *
;               |               |               |               |BURST-FIRE     |CONTINUOUS/    |THE
FREQUENCY       |TRIGGER BUTTON *
;               |               |               |               |MODE SELECT    |FULL-AUTO MODE|SELECT
SWITCH  |               *
;               |               |               |               |               |SELECT         |
        |               *
;-------------------------------------------------------------------------------------------------------------*

;=============================================================================================================
=======================*
;                                                               PORTC BITMAP
                        *
;=============================================================================================================
=======================*
;               |               |               |               |               |               |
        |               *
;       7       |       6       |       5       |       4       |       3       |       2       |
1       |       0       *
;--------------|--------------|--------------|--------------|--------------|--------------|--------------*
;DIGITAL |DIGITAL |DIGITAL |DIGITAL |DIGITAL |DIGITAL |DIGITAL |DIGITAL *
;OUTPUT         |OUTPUT         |OUTPUT         |OUTPUT         |OUTPUT         |OUTPUT         |OUTPUT
        |OUTPUT         *
;--------------|--------------|--------------|--------------|--------------|--------------|--------------*
;UNUSED         |UNUSED         |UNUSED         |UNUSED         |OUTPUT USED TO |OUTPUT USED TO |UNUSED
        |UNUSED         *
  ;             |               |               |               |CONTROL THE    |CONTROL THE    |
        |               *
;               |               |               |               |SOLENOID       |LASER          |
        |               *
;               |               |               |               |               |               |
        |               *
;-------------------------------------------------------------------------------------------------------------*

INITIALIZE_CODE  CODE
INITIALIZE

;*** INTOSC SETUP *******

                BANKSEL         OSCCON
                MOVLW   0xEA                    ;ENABLE INTOSC, SET TO 4MHz
                MOVWF   OSCCON
                BANKSEL         CLKRCON
                CLRF    CLKRCON                         ;DISABLE CLOCK REFERENCE


;*** SFR SETUP **********


;*** SET OPTION_REG: ****

                BANKSEL         OPTION_REG
                BCF     OPTION_REG, PS0                 ;\\
                BCF     OPTION_REG, PS1                 ; >>PRESCALER RATIO SET 1:1
                BCF     OPTION_REG, PS2                 ;//
                BSF     OPTION_REG, PSA                 ;PRESCALER ASSIGNED TO WDT
                BCF     OPTION_REG, TMR0SE              ;TMR0 EDGE SET RISING
```

```
                BCF     OPTION_REG, TMR0CS              ;TMR0 CLOCK SOURCE SET TO INTERNAL
                BSF     OPTION_REG, INTEDG              ;RB0/INT SET TO RISING EDGE
                BSF     OPTION_REG, NOT_WPUEN           ;WEAK PULL UP ENABLED

;*** SET INTCON REG: ****

                BANKSEL INTCON
                BSF     INTCON, IOCIE           ;ENABLE IOC INTERRUPT
                BCF     INTCON, INTE            ;DISABLE INT EXTERNAL INTERRUPT
                BSF     INTCON, PEIE            ;ENABLE PERIPHERAL INTERRUPTS
                BSF     INTCON, GIE                     ;ENABLE ALL UNMASKED INTERRUPTS

;*** SET PIE1-4 REG: *****

                BANKSEL         PIE1
                MOVLW   0x01
                MOVWF   PIE1                    ;DISABLE ALL PIE1 INTERRUPTS, EXCEPT TMR1IE
                BANKSEL         PIE2
                CLRF    PIE2                    ;DISABLE ALL PIE2 INTERRUPTS
                BANKSEL         PIE3
                CLRF    PIE3                    ;DISABLE ALL PIE3 INTERRUPTS
                BANKSEL         PIE4
                CLRF    PIE4                    ;DISABLE ALL PIE4 INTERRUPTS

;*** CLEAR INTERRUPT FLAGS ***

                BANKSEL         PIR1
                CLRF    PIR1                    ;CLEAR ALL PIR1 INTERRUPT FLAGS
                BANKSEL         PIR2
                CLRF    PIR2                    ;CLEAR ALL PIR2 INTERRUPT FLAGS
                BANKSEL         PIR3
                CLRF    PIR3                    ;CLEAR ALL PIR3 INTERRUPT FLAGS
                BANKSEL         PIR4
                CLRF    PIR4                    ;CLEAR ALL PIR4 INTERRUPT FLAGS

;*** SET CCP1CON REG: **

                BANKSEL         TRISC
                BSF     TRISC,2                         ;DISABLE PWM OUTPUT DRIVER FOR CCP1 SETUP
                BANKSEL         CCP1CON
                MOVLW   0x0C
                MOVWF   CCP1CON                         ;SELECT PWM MODE FOR CCP1
                BANKSEL         CCPR1L
                MOVLW   0x19                    ;PW FOR INITIALIZING 95%DC AT 38kHz
                ;MOVLW  0x11                    ;PW FOR INITIALIZING 95%DC AT 56kHz
                MOVWF   CCPR1L

;*** TIMER 1 SETUP *****

                BANKSEL         T1CON
                MOVLW   0xA8                    ;ENABLE & SELECT TMR1 LP OSCILLATOR, SET PRESCALE TO 1:4,
DISABLE TMR1 AT THIS POINT
                BANKSEL         T1GCON
                CLRF    T1GCON                          ;DISABLE TMR1 GATE

;*** TIMER 2 SETUP *****

                BANKSEL         T2CON
                CLRF    T2CON
                BANKSEL         PR2
                MOVLW   0x19                    ;SET PR2 TO 25 FOR 38kHz PWM FREQUENCY
```

```
                ;MOVLW   0x11                ;SET PR2 TO 17 FOR 56kHz PWM FREQUENCY
                MOVWF   PR2

;*** ALTERNATE PIN FUNCTION *****

                BANKSEL         APFCON1
                CLRF    APFCON1                 ;SELECTS PORT MAPPING
                CLRF    APFCON2                 ;SEE DATASHEET (pg 132-133) FOR MAPPING WITH BOTH
CLEARED

;*** PORT A SETUP **** PORT A SET AS LOGIC OUTPUT (NOT USED)

                BANKSEL         PORTA
                CLRF    PORTA               ;CLEAR PORTA
                BANKSEL         LATA
                CLRF    LATA                ;CLEAR LATA
                BANKSEL         TRISA
                MOVLW   0xFF
                MOVWF   TRISA               ;SET PORTA ALL AS INPUT
                BANKSEL         ODCONA
                CLRF    ODCONA                  ;DISABLE PORTA OPEN-DRAIN
                BANKSEL         ANSELA
                CLRF    ANSELA                  ;SET PORTA ALL AS DIGITAL INPUT
                BANKSEL         SLRCONA
                CLRF    SLRCONA                 ;SET THE SLEW RATE FOR PORT A TO MAXIMUM

;*** PORT B SETUP **** PORT B USED AS LOGIC INPUT

                BANKSEL         PORTB
                CLRF    PORTB               ;CLEAR PORTB
                BANKSEL         LATB
                CLRF    LATB                ;CLEAR LATB
                BANKSEL         TRISB
                MOVLW   0xFF
                MOVWF   TRISB               ;SET PORTB ALL AS INPUTS
                BANKSEL         ODCONB
                CLRF    ODCONB                  ;DISABLE PORTB OPEN-DRAIN
                BANKSEL         ANSELB
                CLRF    ANSELB                  ;MAKE PORTB DIGITAL INPUTS
                BANKSEL         SLRCONB
                CLRF    SLRCONB                 ;SET THE SLEW RATE FOR PORT B TO MAXIMUM
                BANKSEL         IOCBP
                MOVLW   0x0F                ;ENABLE PORTB IOC FOR POSITIVE EDGES ON RB3:0
                MOVWF   IOCBP
                BANKSEL         IOCBN
                MOVLW   0x0E                ;ENABLE PORTB IOC FOR NEGATIVE EDGES ON RB3:1
                MOVWF   IOCBN
                BANKSEL         INLVLB
                CLRF    INLVLB                  ;SET PORTB INPUT THRESHOLD TO TTL (IGNORED NOISE
BETTER FOR TRIGGER INPUT)

;*** PORT C SETUP ****

                BANKSEL         PORTC
                CLRF    PORTC               ;CLEAR PORTC
                BANKSEL         LATC
                CLRF    LATC                ;CLEAR LATC
                BANKSEL         TRISC
                MOVLW   0x00
                MOVWF   TRISC               ;SET PORTC ALL AS OUTPUTS
                BANKSEL         ODCONC
```

```
                CLRF    ODCONC                  ;DISABLE PORTC OPEN-DRAIN
                BANKSEL         ANSELC
                CLRF    ANSELC                  ;MAKE PORTC ALL DIGITAL I/O
                BANKSEL         SLRCONC
                CLRF    SLRCONC                 ;SET THE SLEW RATE FOR PORT C TO MAXIMUM

;*** BOOT CODE ****

                BCF     INTCON,0
                BANKSEL         T1CON
                BSF     T1CON,0                 ;ENABLE TIMER 1
                BANKSEL         T2CON
                BSF     T2CON,2                 ;ENABLE TIMER 2
                BANKSEL         TRISC
                BSF     TRISC,2                 ;DISABLE OUTPUT DRIVER ON STARTUP (WAS ENABLED
ON STARTUP WITHOUT THIS)


                RETURN                          ;RETURN TO .asm FILE
```

14: Appendix E (GitHub)

The QR code below (Figure 14.1) will take you to the project repository, where you can download all files for the project, including schematics, PCBs, code, etc.

Figure 14.1: GitHub Laser Game Project Repository