This is the Revision A version of the Laser1 RoboBrick. The status of this project is that it has been replaced by the Revision B version.

Laser1 Robobrick (Revision A)

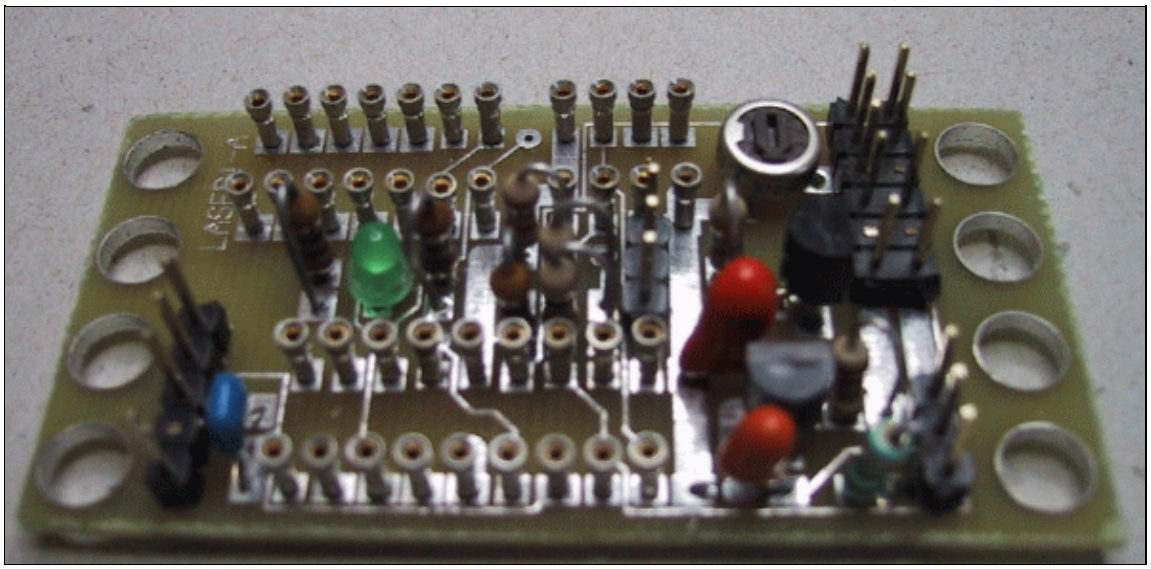
Table of Contents

This document is also available as a PDF document.

1. Introduction

The Laser1 RoboBrick is a module that is designed to allow the use of a slightly modified laser pointer to detect passive refector beacons at a distance of approximately 10 meters (or about 32 feet for the metrically impaired.) It is meant to be used in conjunction with the LaserHead1 RoboBrick. It has a 3.1 volt power supply that can be turned on and off under program control to replace the batteries that come with the laser pointer. In addition, the Laser1 RoboBrick can send a pulses to a servo to cause it to slew back and forth under program control. In conjunction with 3 or more properly placed passive reflectors, the Laser1 RoboBrick can be used to triangulate a robot's position. We will not know how accurately until after it has been built and tested.

The basic idea behind this laser pointer system came from an article written by Jim Ubersetzig [Ubersitzig1999]. We are greatly indebted to Jim for figuring this system out in the first place and for E−mail support as we asked some follow on questions. The equations for doing triangualation can be found in a paper by Clare McGillem and Theodore Rappaport [McGillem1989]. A more recent on−line derivation of the formulas can be found in a paper by Richard Vannoy [Vannoy2001]. There is another paper by M. Bertke and L. Gurvits [Bertke1994] that explains how to deal with more than three beacons; alas, the math is quite a bit over my head. It should be mentioned that the McGillem and Gurvits papers were found by scanning the bibliography of the fairly comprehensive treatment of robot navigation by Borenstein, Everett, and Feng [Borenstein1996].



2. Programming

The initial software for the Laser1 RoboBrick will only be responsible for measuring the angles between the beacons and the robot proper. The next software/hardware reversion should be able to compute the (x, y) position of the robot, given the (x, y) positions of the beacons. The reason for the currently reduced functionality is because the floating point math libraries occupy more than the 2K of available code space on PIC16F628; indeed, all of the code appears to occupy a significant fraction of the 8K of code space available in a PIC16F876.

The current design requires a servo that has been modified for continuous rotation [AcronameServo, SandbergServo, Buse2000]. This allows the LaserHead1 to sweep a full 360 degrees. An additional sensor is used to detect each time the LaserHead1 sweeps past a known location.

Since the Laser1 uses a PIC16F628 with a 16 bit Timer1 module and the PWM/Capture/Compare module, it is capable of resolving the time between the beacon reflection returns to the instruction rate of 5 MHz. Since it takes our current laser head 2−3 seconds to slew 360 degrees, we need a counter resolution of $3 \times 5 \times 10^6 = 15$ million. Since 15 million is greater than 64 thousand, a 24−bit counter is needed.

{to be continued...}

The BIROD2 RoboBrick supports both the standard shared commands and the shared interrupt commands in addition to the following commands:

Command	Send/ Receive	Byte Value								Discussion
		7	6	5	4	3	2	1	0	
Set Laser Enable	Send	0	0	0	0	0	0	0	*z*	Set laser enable bit to *z*
Read Laser	Send	0	0	0	0	0	0	1	0	Read and return the sweeping bit *s*, high bit *h*, low bit *l*, and laser bit *z*, and capture count *nnnn*.
	Receive	*s*	*h*	*l*	*z*	*n*	*n*	*n*	*n*	
Disable Servo	Send	0	0	0	0	0	1	0	0	Disable servo
Enable Servo Low	Send	0	0	0	0	0	1	0	1	Enable Servo Low
Enable Servo High	Send	0	0	0	0	0	1	1	0	Enable Servo High

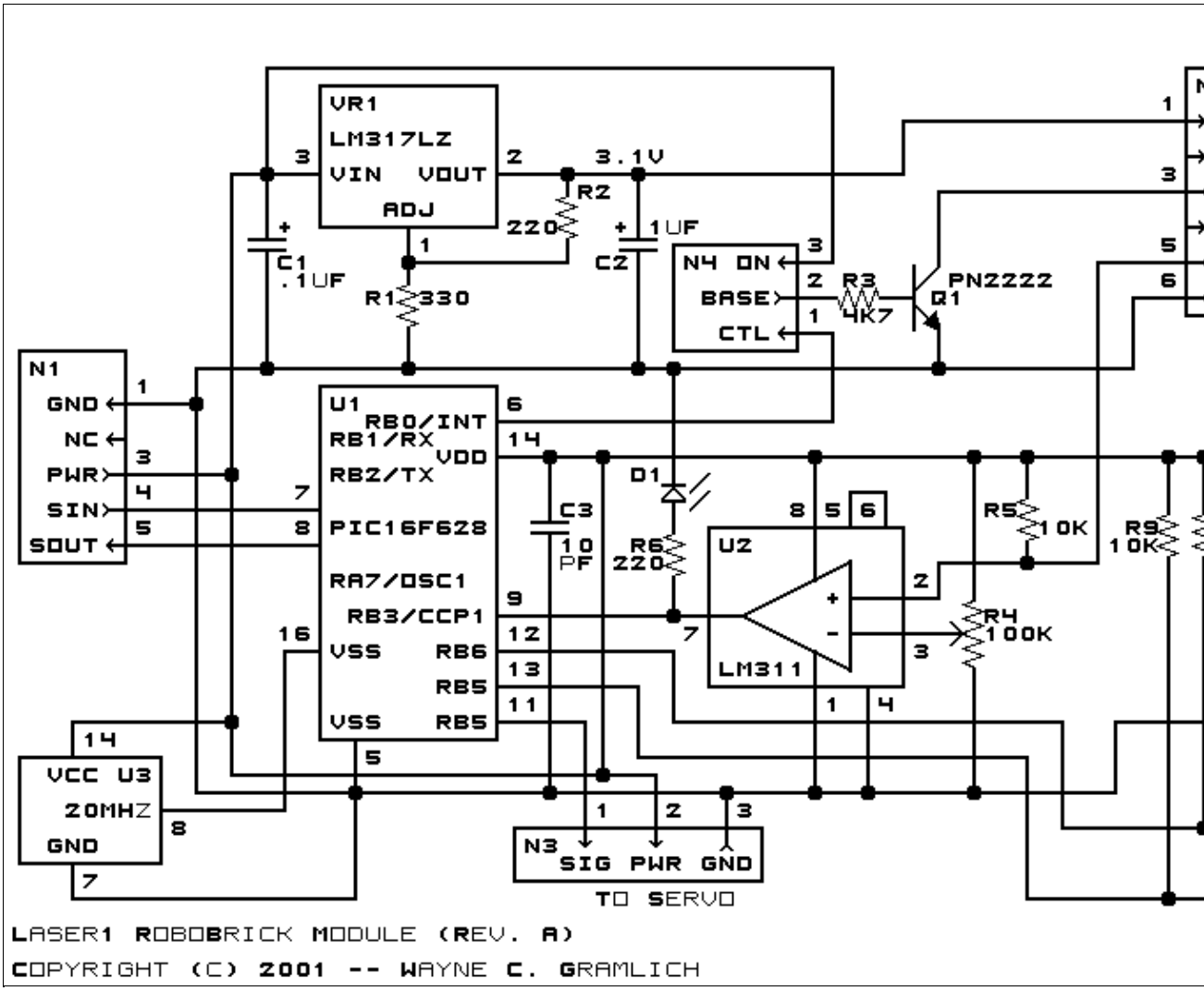
Sweep	Send	0	0	0	0	0	1	1	1	Sweep
Set Servo Low	Send	0	0	0	0	1	0	0	0	Set servo low register to *lllllll*
	Send	*l*	*l*	*l*	*l*	*l*	*l*	*l*	*l*	
Set Servo High	Send	0	0	0	0	1	0	0	1	Set servo high register to *hhhhhhhh*
	Send	*h*	*h*	*h*	*h*	*h*	*h*	*h*	*h*	
Read Servo Low	Send	0	0	0	0	1	0	1	0	Read and return servo low register of *lllllll*
	Receive	*l*	*l*	*l*	*l*	*l*	*l*	*l*	*l*	
Read Servo High	Send	0	0	0	0	1	0	1	1	Read and return servo high register of *hhhhhhhh*
	Send	*h*	*h*	*h*	*h*	*h*	*h*	*h*	*h*	
Read Interrupt Bits	Send	1	1	1	0	1	1	1	1	Return the interrupt pending bit *p* and the interrupt enable bit *e*.
	Receive	0	0	0	0	0	0	*e*	*p*	
Set Interrupt Bit Commands	Send	1	1	1	1	0	*c*	*c*	*c*	Execute shared set interrupt command *ccc*.
Shared Commands	Send	1	1	1	1	1	*c*	*c*	*c*	Execute shared command *ccc*.

3. Hardware

The hardware consists of a circuit schematic and a printed circuit board.

3.1 Circuit Schematic

The schematic for the Laser1 RoboBrick is shown below:



The parts list kept in a separate file −− laser1.ptl. `

3.2 Printed Circuit Board

The printed circuit board files are listed below:

laser1_back.png
: The solder side layer.

laser1_front.png
: The component side layer.

laser1_artwork.png
: The artwork layer.

laser1.gbl
: The RS−274X "Gerber" back (solder side) layer.

laser1.gtl
: The RS−274X "Gerber" top (component side) layer.

laser1.gal
: The RS−274X "Gerber" artwork layer.

laser1.drl

The "Excellon" NC drill file.

laser1.tol
The "Excellon" tool rack file.

4. Software

The Laser1 software is available as one of the followin:

laser1.ucl
The µCL source code.

laser1.asm
The resulting human readable PIC assembly listing

laser1.lst
The resulting human readable PIC listing file.

laser1.hex
The resulting Intel® Hex file that can be fed into a PIC16F628 programmer.

5. Issues

The following fabrication issues came up:

- Pull MCLR (pin 4) high.
- For some reason, the current system only works at 10 MHz. When I plug in one of my 20 MHz crystals, it does not work. Perhaps a bad batch of crystals? Perhaps a bad F628? Maybe the F628 does not like the voltage levels coming out of the crystal. Who knows?
- The LM311 has an open collector output. Thus the LED needs to tied to Vcc to pull the line up.
- Think about adding a bias trim pot to adjust the sensor sensitivity.
- There is no evidence that the comparator is doing any good. The signal the comes back is so clean that it can be read directly as binary signal. There is certainly no evidence that the trim pot is providing any useful descrimination.
- The holes for N1−N5 are too large (size 3) and should be made smaller (size 2.)
- Put plus signs next to postive terminals of C1 and C2.
- Move R5 .05 inches to the left to avoid interfering with the corner of the crystal in U3.
- Change R3 from 4.7K to 1K Ohm.
- Fix artwork labels for N2 and N5 are swapped.
- Maybe make the crystal half height to get a little more space.
- It may be necessary to swap the input pins to the LM311 to invert the polarity of the light sensor.
- We need to upgrade to a PIC16F876 to have enough program space to do all of the floating point calculations needed to to actually triangulate.
- Make sure that the sensor wire goes to an interrupt pin.
- Change the pin polarity on N5 to be pin 3 instead of pin 2.

References

[AcronameServo]
Modifying Servos for Continuous Rotation. Published online by Acroname at URL: http://www.acroname.com/robotics/info/ideas/continuous/continuous.html.

[Bertke1994]

Mobile Robot Localization Using Landmarks by Margrit Bertke and Leonid Gurvits in the proceedings of the *1994 International Conference on Intelligent Robots and Systems* (IROS '94), pages 135−142. URL: http://www.umiacs.umd.edu/~betke/#landmarkpaper

[Borenstein1996]

"Where am I?" −− Systems and Methods for Mobile Robot Positioning by J. Borenstein, H. R. Everett, and L. Feng published by A. K. Peters, Ltd. ISBN: 156881058X. An on−line version of the paper is available at URL: http://www.eecs.umich.edu/~johannb/pos96rep.pdf. Note: There is a slightly different 1995 edition entitled *Navigating Mobile Robots: Systems and Techniques* that is out of print ASIN: 1568810660 .

[Buse2000]

Variable Speed Control Modification to the Futaba S3003 RC Servo by Lee Buse published in the *September 2000 issue of the Seattle Robotics Encoder*. URL: http://www.seattlerobotics.org/encoder/200009/S3003C.html.

[McGillem1989]

A Beacon Navigation Method for Autonomous Vehicles by Clare D. McGillem and Theodore S. Rappaport published in *IEEE Transactions on Vehicular Technology* Vol. 38 No. 3 August 1989, pages 132−139. Note: The algorithm proper can be found in equations (7) through (16). The derivation of the algorithm can be found in the appendix. In additin, I have written a C program that implements the described algorithm.

[SandbergServo]

The Sandberg Servo Modification Method published by the Portland Area Robotics Society (PARTS) at URL: http://www.portlandrobotics.org/servo/ds−hack.html.

[Ubersitzig1999]

A Circular Navigation System written by Jim Ubersetzig, published in three parts in the September, October, and November 1999 issues of *The Robot Builder* newsletter of the Robotics Society of Southern California. URL: http://www.csulb.edu/~wmartinz/rssc/newsletters/. Note: At the end of the third part of the series, there is mention of a continuation of the article (part 4) in the next month of "The Robot Builder"; to the best of our knowledge this continuation article has not actually been published.

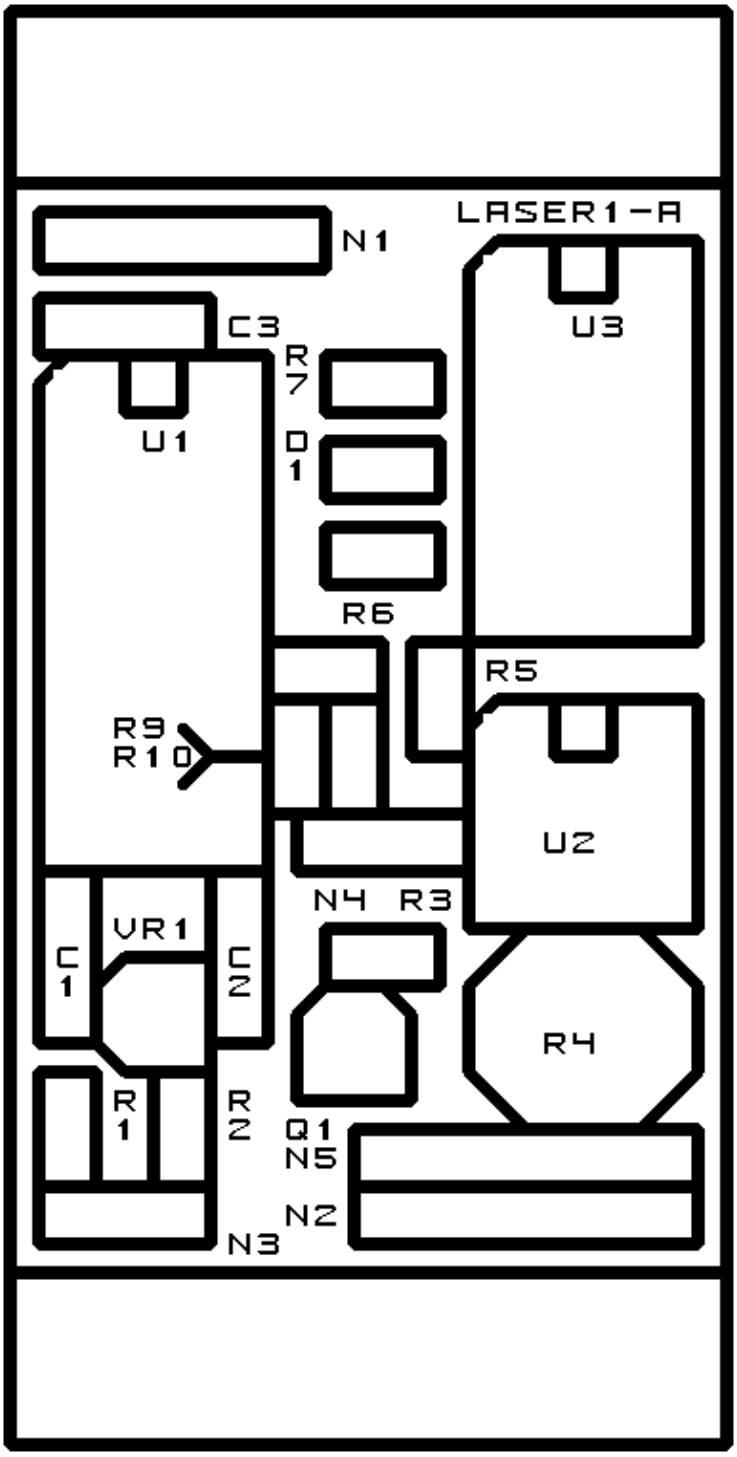
[Vannoy2001]

Accurate Autonomous Robot Laser Navigation Using Only Passive Reflectors by Richard T. Vannoy II published in the *September 2001 issue of the Seattle Robotics Encoder*. URL: http://www.seattlerobotics.org/encoder/200109/lasernav.htm.

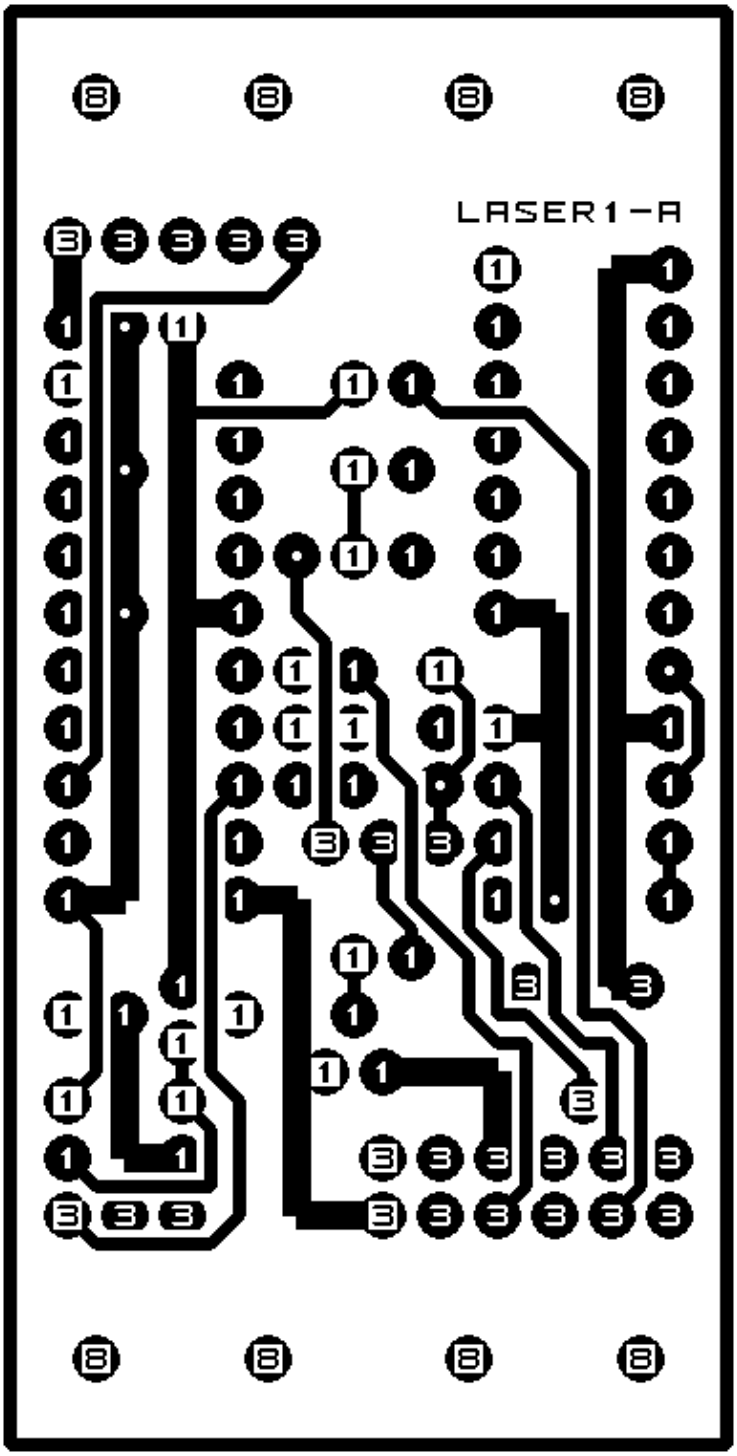
A. Appendix A: Parts List

```
# Parts list for Laser1 RoboBrick (Rev. A)
#
C1: Capacitor100nF – .1uF Tantalum Capacitor [Jameco: 33486]
C2: Capacitor1uF – 1uF Tantalum Capacitor [Jameco: 33662]
C3: Capacitor10pF – 10 pF Ceramic Capacitor [Jameco: 15333]
D1: LEDGreen – Green LED [Jameco: 34606]
N1: Header1x5.RBSlave – 1x5 Male Header [5/40 Jameco: 160881]
N2: Header1x6.Laser1_Head – 1x6 Male Header [6/40 Jameco: 160881]
N3: Header1x3.Laser1_Servo – 1x3 Male Header [3/40 Jameco: 160881]
N4: Header1x3.Laser1 – 1x3 Male  Header [3/40 Jameco: 160881]
N5: Header1x6.Laser1_Position – Laser 1x6 Header [6/40 Jameco: 160881]
Q1: PN2222.EBC – Plastic NPN Silicon Switching Transistor [Jameco: 28628]
Q2–5: IRD500 – Infrared Detector [Jameco: 112168]
R1: Resistor330.Vertical – 330 Ohm 1/4 Watt Resistor [Jameco: 30867]
R2: Resistor220.Vertical – 220 Ohm 1/4 Watt Resistor [Jameco: 30470]
R3: Resistor4K7.Vertical – 4.7K Ohm 1/4 Watt Resistor [Jameco: 31026]
R4: ResistorTrimPot100K.Laser1 – 100K Ohm Potentiometer [Jameco: 95484]
R5: Resistor10K.Vertical – 10K Ohm 1/4 Watt Resistor [Jameco: 29911]
R6–8: Resistor220.Vertical – 220 Ohm 1/4 Watt Resistor [Jameco: 30470]
R9–10: Resistor10K.Vertical – 10K Ohm 1/4 Watt Resistor [Jameco: 29911]
U1: PIC16F628.Laser1 – Microchip PIC16F628 [Digikey: PIC16F628–20/P–ND]
U2: LM311 – Voltage Comparator [Jameco: 23528]
U3: Oscillator20MHz – 20MHz Oscillator[Jameco: 27932]
VR1: LM317LZ – Adjustable Positive Voltage Regulator [Jameco: 23552]
```

B. Appendix B: Artwork Layer



C. Appendix C: Back (Solder Side) Layer



D. Appendix D: Front (Component Side) Layer