INFRARED PROXIMITY SENSOR

Kevin C. Fronczak

***Abstract*—Using a mix of analog and digital circuitry, an Infrared Proximity Sensor was designed, prototyped and tested at a board level. The sensor operates off of ±2.5 V rails and displays changes in distance in three inch intervals and displays the range on one of four LEDs. A multivibrator is used to transmit the Infrared light which is then reflected off of a surface, detected by a photodiode. This signal then passes through a trans-impedance amplifier and rectified in order to detect distance via a DC voltage. Due to the non-linearity of Infrared Sensors, the voltage leveled was linearized in both hardware and software in order to obtain better precision when determining distance. At large distances the peak-to-peak voltage of the rectified signal becomes more of a problem so both an analog and digital low-pass filter was implemented to reduce this ripple. The level detection is accuracy can be fine-tuned by simply changing level parameters in the software. This gives the user the ability to fine-tune the values to be accurate with any reflecting surface.**

I. Introduction

INFRARED Proximity Sensors at the simplest level consist of two component: an infrared LED and a Photodiode. However, in order to achieve a signal that can actually be used for detecting distances, various control circuitry must be created. First, circuitry must be designed that will turn the LED on and off at a certain frequency. This has dual purposes: the first is that it cuts power consumption in half as opposed to perpetually being on and the second is that it makes detection easier.

On the receiver side, the photodiode is going to output in current which is not ideal for level detection. Thus this current must be converted to a voltage. To make distance detection easier, the signal can then be rectified to achieve a constant DC level which can be sent to distance detection circuitry.

One large problem in detecting distances with a infrared is that with a fully reflective surface the current received by the photodiode will be related by $1/r^2$. On a semi-reflective surface the performance is even worse at $1/r^4$. This makes detecting large distances very difficult as the variance between voltage levels will become very small. As such, the receiver needs to be selectively linearized so that the voltage levels at short distance maintain their large variance while increasing the variance at large distances.

Infrared Proximity Sensor
Author: Kevin C. Fronczak Written on 2/23/2012
High resolution figures can be found on:
http://kevinfronczak.com/documents/IR_Sensor/figures

II. PSpice Circuit Schematic

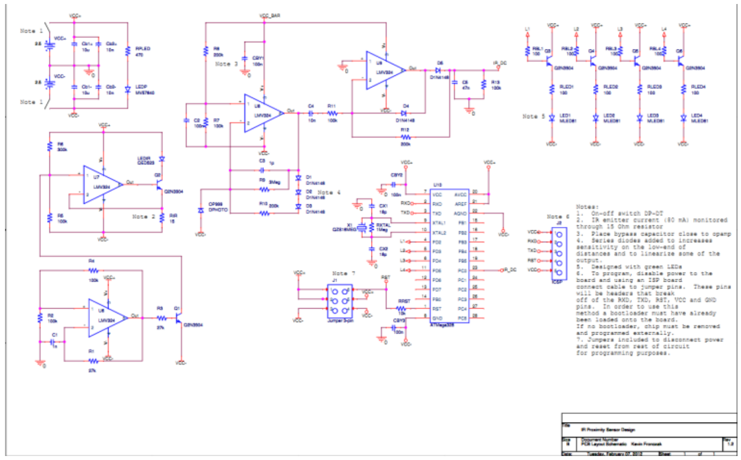

Fig. 1: PSpice Circuit Schematic

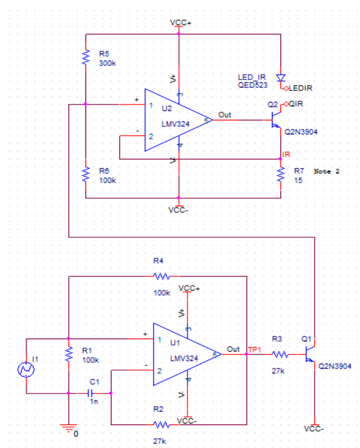

Fig. 2: Transmitter Design and Simulation Schematic

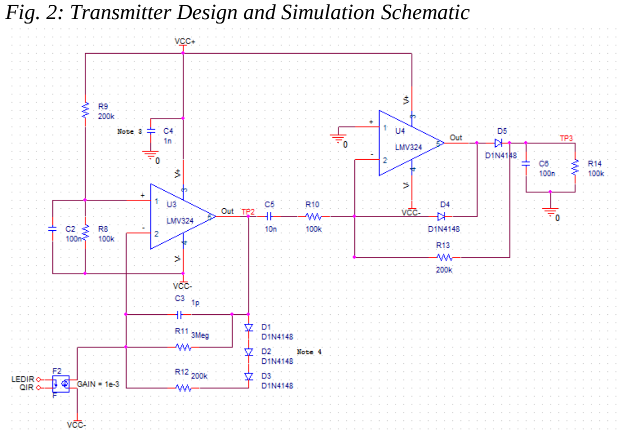

Fig. 3: Receiver Design

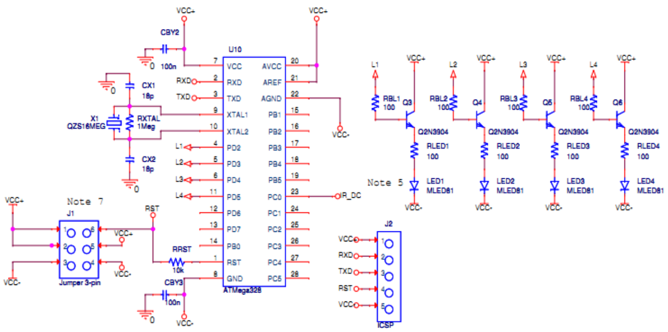

Fig. 4: Detector Design Schematic

III. PCB Description

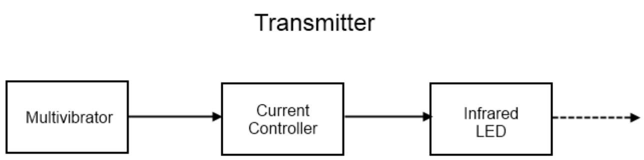

Fig. 5: Transmitter Block Diagram

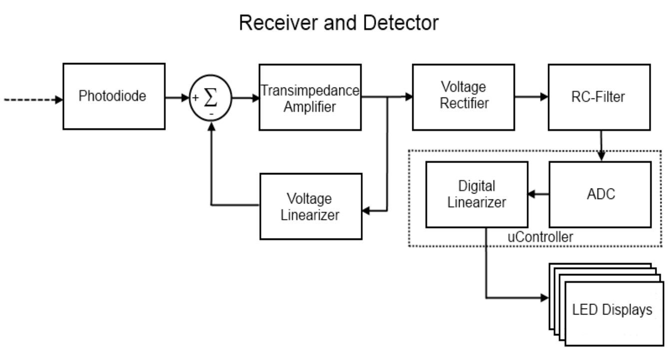

Fig. 6: Receiver and Detector Block Diagram

The transmitter consists of three simple blocks, a multivibrator, current controller and Infrared LED. The multivibrator is simply a square-wave oscillator that turns Q1 on and off. This alternating switch is part of the current controller, along with a voltage divider, that sets the voltage drop across RIR which determines the current through the infrared resistor. The reason the voltage drop needs to be set is to limit any error induced by turning the IR LED completely off as that would introduce a delay in the infrared transmission.

The Receiver circuit is more complicated. It involves the use of a transimpedance amplifier that simultaneously converts the current induced by the photodiode into a voltage but also shifts this voltage up as determined by the voltage divider network R7 and R8. A linearizing circuit block was added in to decrease the gain of the amplifier at close distances. This was accomplished by placing three diodes in series with a 200 kΩ resistor which would become active once the voltage at the output was sufficiently high. This signal then passes to a voltage rectifier which, as the name might suggest, rectifies the waveform. This produces a DC voltage which is smoothed at the output by an RC-filter.

The Detection circuit takes this DC voltage and passes it to an ADC. The ADC value is then smoothed in software to eliminate any error caused by rippling at large distances and then illuminates the appropriate LED to indicate the detected object distance.

IV. Theory of Operation

The operation of the transmitter is fairly trivial when compared with the other blocks of the circuit. The first block is the multivibrator whose frequency is determined by the following equation: $T = 2R_1C \ln(1 + 2R_2 / R_4)$ which results in a frequency of roughly 16 kHz using the components found in the design schematic.

The next block is the current controller. This block simply controls the amount of current through RIR which controls the current through the infrared LED. Using a buffer transistor, the voltage at the plus input of the opamp will go to the low voltage state when the multivibrator output is high and the high voltage state when the multivibrator output is low. This voltage is then mirrored to the minus input of the opamp which is directly connected to RIR. When the multivibrator is high, V_+ of the opamp is just controlled by a voltage divider and sits at -1.25 V. When the multivibrator output is low, however, the current is drawn down through Q1which subsequently sets V_+ to V_{CE} which, for a 2N3904 transistor, is around 0.1 V above the emitter voltage. For this circuit, that would be -2.4 V. Using these two values, the high and low currents through the infrared LED can be determined to be 83.3 mA and 7 mA respectively. These values were verified by the waveforms in [Fig. 7] in the Simulation section.

The receiver involves the use of a transimpedance amplifier and a rectification circuit in order to manipulate the received infrared signal into a usable DC voltage level. The transimpedance amplifier works by taking the input current from the photodiode and converting it into a voltage at the output. This output's gain is controlled by a parallel combination of a resistor and capacitor. The capacitor is there to lower the amplifier gain at high frequencies to prevent any potential (and unwanted) oscillation. The voltage divider located on the input of V_+ is there to shift the output to a certain DC level. This level, as determined by the voltage divider, is about one-third of V_{CC}- or roughly -0.83 V.

The next stage in the receiver design is the rectification circuit. This stage takes the level shifted voltage from the transimpedance amplifier and converts it to a DC voltage. The RC Filter at the output is used to smooth the ripples in the signal. The values of 100 kΩ and 100 nF were chosen by trial and error. The feedback resistor, R12, was increased from the original design by 100 kΩ to 200 kΩ. The reason for this decision was that it increased the gain (from a gain of 1 to a gain of 2) which allowed for a larger voltage range at the output of the rectifier. This ultimately helped to detect larger distances by differentiating them from a case where there was no object in front of the board (which would generally pick up noise in the 200 mV to 400 mV range).

The chosen technology for the detector circuit was an Atmel ATMega328P due to familiarity with the platform and ease of access to programming utilities. The initial design utilized a TI MSP430G2231 microcontroller but due to the lack of a proper Spy-Bi-Wire programming interface was quickly discarded.

The detector works by utilizing the on-board ADC to convert to the rectified DC voltage from the Receiver into a 10-bit integer. During testing three threshold levels were set in memory: three inch voltage level, six inch voltage level and nine inch voltage level. The ADC voltage would then be compared to these thresholds and select the proper LED to turn on. The digital outputs were routed to the base of NPN transistors which then would act as switches to supply the indicator LEDs the correct amount of current.

V. Simulations

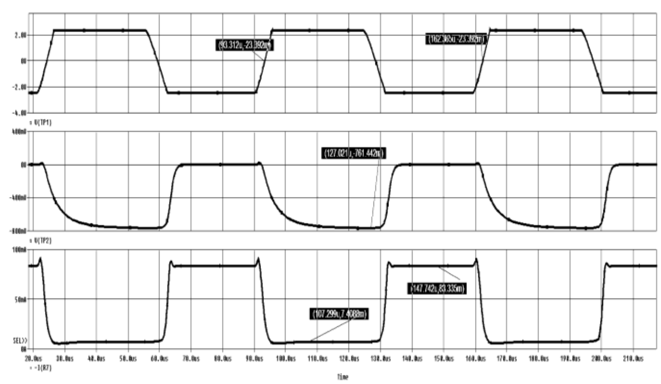

Fig. 7: (Top to Bottom) Multivibrator output, Receiver TP2, IR LED Current

[Fig. 8] and [Fig. 9] below shows the rectifier output for a simulated gain of $1x10^{-6}$ and $1x10^{-3}$ respectively. Notice how at low gain (which correlated to large distances) that the response time is around 3 mS as opposed to almost instant for higher gains (close distances). Another important note is that the ripple voltage is far greater (nearly one order of magnitude greater) in the low gain case as opposed to the high gain case. As will be seen in the board testing phase, this will produce large enough voltage variances to make it difficult to differentiate between certain thresholds (6 inches, 9 inches, etc).

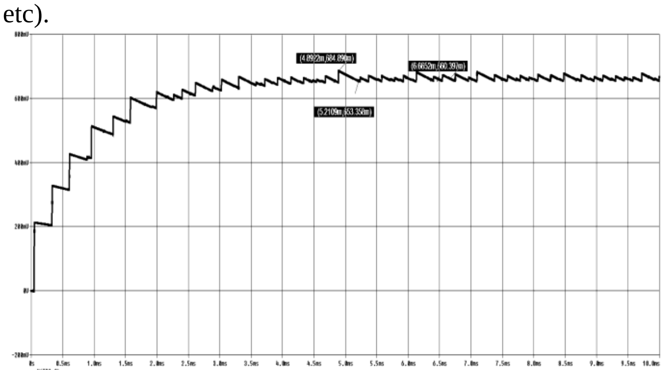

Fig. 8: Rectifier output with receiver gain of $1x10^{-6}$

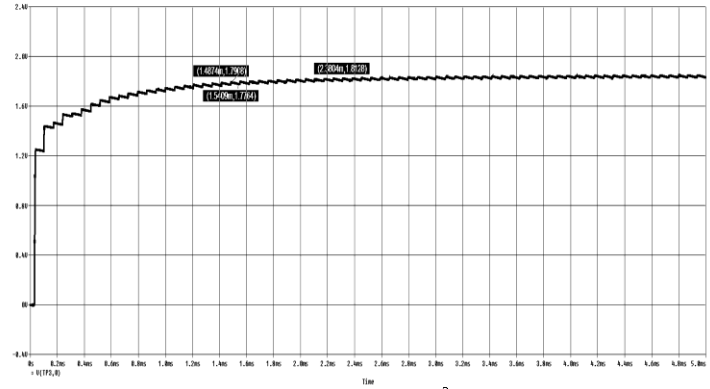

Fig. 9: Rectifier output with receiver gain of $1x10^{-3}$

The original design for the receiver did not include the linearization portion in the feedback loop of the transimpedance amplifier. As [Fig. 10] shows below, small changes in current on the photodiode result in large variances in voltage which is problematic; it causes the rectified output to saturate at close distances making it nearly impossible to differentiate between 1 inch and 3 inches. A fix for this is by adding in non-linear elements to help linearize the curve. Three diodes were used in series with a 200 kΩ resistor so that when the voltage at the output reached a certain threshold, the 200 kΩ would essentially replace the 3 MΩ resistor in the feedback loop thus decreasing the gain. [Fig. 11] shows the result of this change.

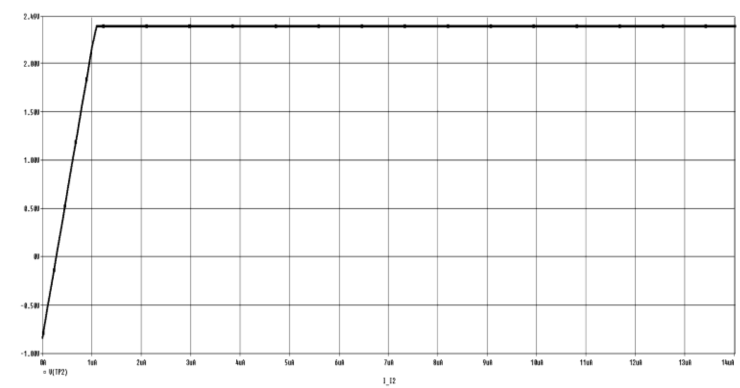

Fig. 10: Default output of transimpedance amplifier

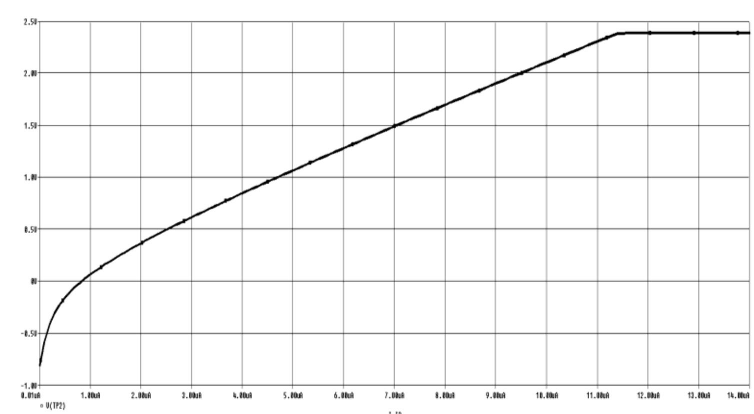

Fig. 11: Modified output of transimpedance amplifier

VI. PCB Physical Layout

Fig. 12: ExpressPCB board layout schematic

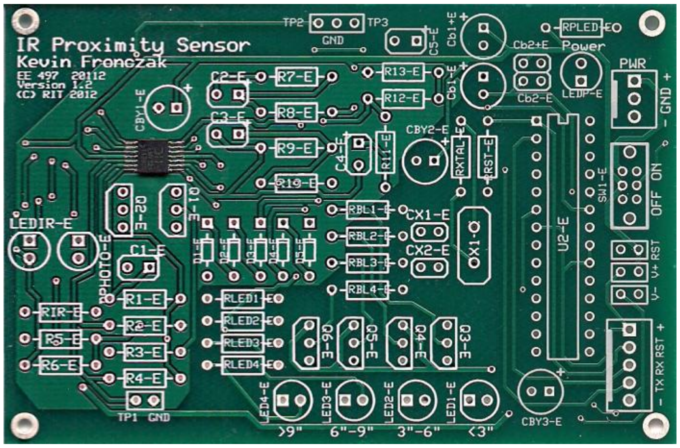

Fig. 13: Front of PCB

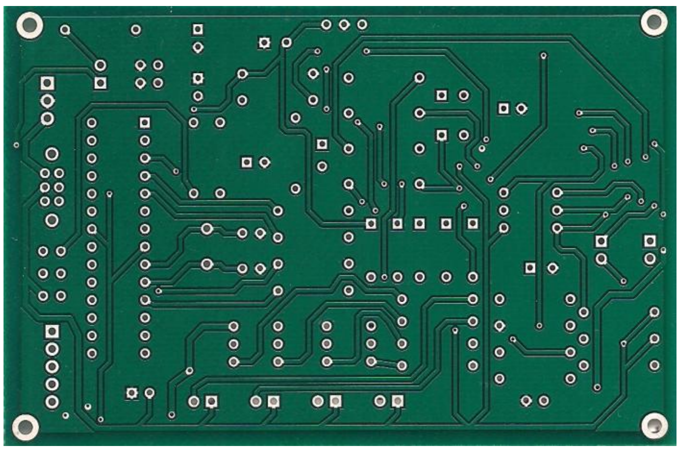

Fig. 14: Back of PCB

Upon receiving the board, each power and ground trace was tested to verify continuity and verify that no shorts were present. Next, each connecting trace was test to verify continuity. This test proved to be useful as it indicated two errors with the board. First, a via was placed too closely to a trace which ended up shorting two pins on the opamp. This problem was quickly remedied by cutting a portion of the via off with a sharp blade. Another problem occurred with two overlapping traces between the RST pin of the microcontroller's programming headers and V_{CC}. This actually wouldn't have cause any operational errors but would make the on-board programming useless. Thus the trace was cut and re-routed with a wire. These were the only two board issues.

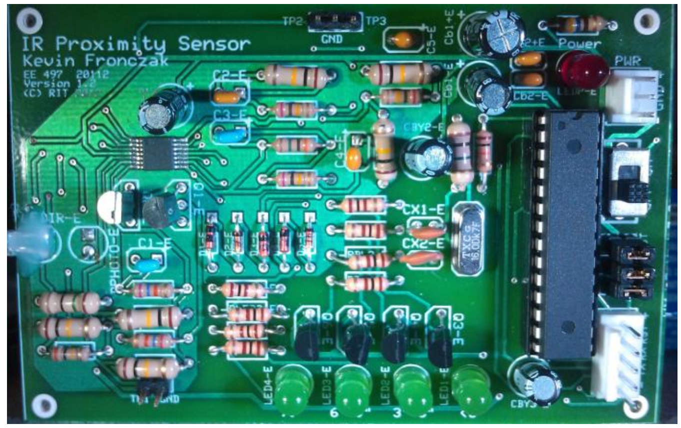

Fig. 15: Assembled PCB Front

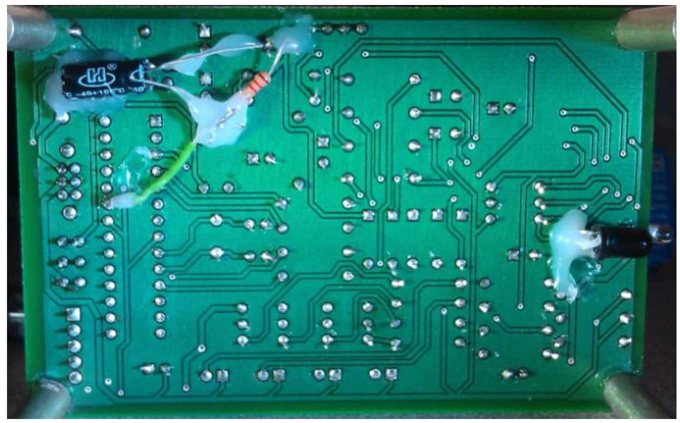

Fig. 16: Assembled PCB Back

No issues occurred during board assembly and the PCB functioned as designed. As both [Fig. 15] and [Fig. 16] show, hot glue was used to tack down loose components or components that could possibly cause shorts if moved. The rear of the PCB shows a slight modification made to stabilize the rectifier signal going to the ADC pin of the microcontroller. A trace was cut leading from the RC filter to the ADC pin and a Low-Pass Filter with cutoff frequency set to 8 kHz. The reason for this modification is outlined in the discussion section.

VII. Measured Results Comparison

The output of the rectifier was measured and recorded for different distances using cardboard box with foil tape acting as a reflector. [Table 1] shows the recorded values over four tests to verify accuracy.

	Simulation	**Prototype**
Multivibrator Frequency	14.48 kHz	13.76 kHz
IR LED 'on' current	83.3 mA	87.2 mA
IR LED 'off' current	7.4 mA	8.3 mA

Table 1: Comparison of simulation and prototype data.

Distance [in]	VDC (V)				Mean
1	2.10	2.09	2.12	2.11	2.105
2	2.00	2.12	2.16	2.07	2.0875
3	1.50	1.31	1.65	1.35	1.452
4	0.96	1.26	1.28	1.21	1.178
5	0.93	0.96	0.91	0.93	0.93075
6	0.83	0.85	0.85	0.86	0.84925
7	0.79	0.78	0.81	0.82	0.79775
8	0.76	0.75	0.75	0.77	0.75575
9	0.72	0.71	0.71	0.72	0.7135
10	0.70	0.70	0.69	0.70	0.70075
11	0.68	0.69	0.69	0.69	0.68575
12	0.67	0.68	0.67	0.68	0.67825

Table 2: Rectifier output data for varying distances during prototype stage

Distance [in]	VDC Prototype	VDC Board Level	% Error
1	2.1 V	2.03 V	3.3%
3	1.45 V	1.78 V	22.8%
6	0.849 V	1.14 V	34.3%
9	0.715 V	0.875 V	22.4%

Table 3: Comparison of board level and prototype data.

As [Table 3] shows, there is a rather large level of error between the readings during the prototype and the board level testing. The reason for this can be attributed mostly to a changed value of the capacitance on the RC filter after the rectifier. During the prototype stage a capacitance of 50 nF was used. However, during PCB testing the ripple from the output of the rectifier was very large at distances greater than 6 inches which caused instability in the ADC reading. On top of including the Low-Pass Filter mentioned in the PCB Physical Layout section, the filter capacitance was increased to 100 nF. These two changes can most certainly account for the roughly 30% difference in recorded prototype values.

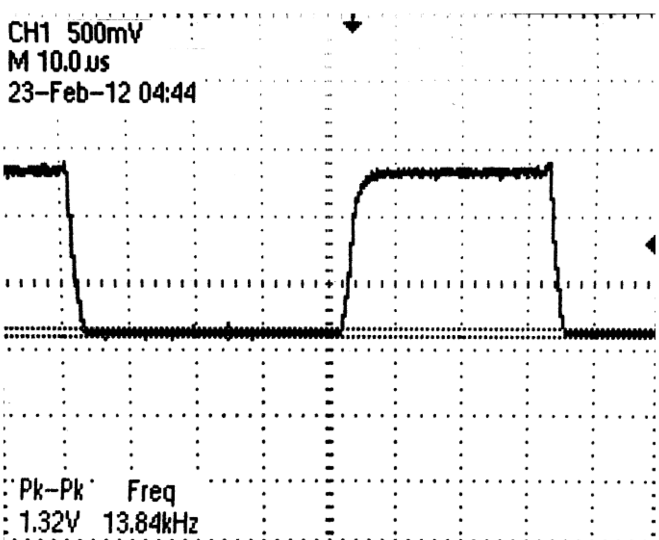

Fig. 17: Board level IR Voltage

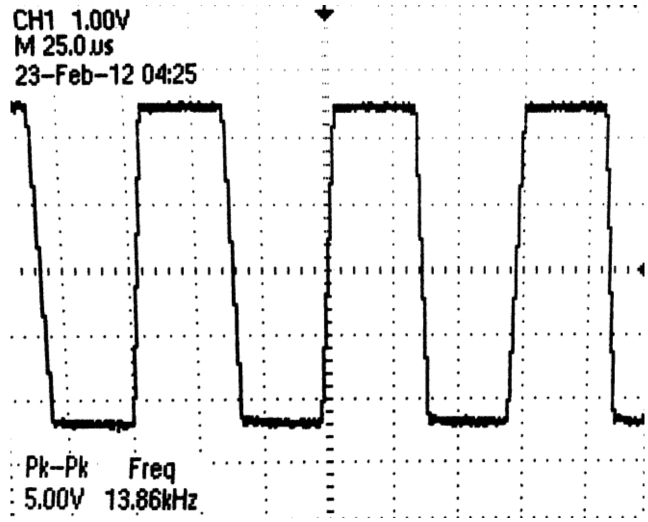

Fig. 18: Board level Multivibrator scope capture

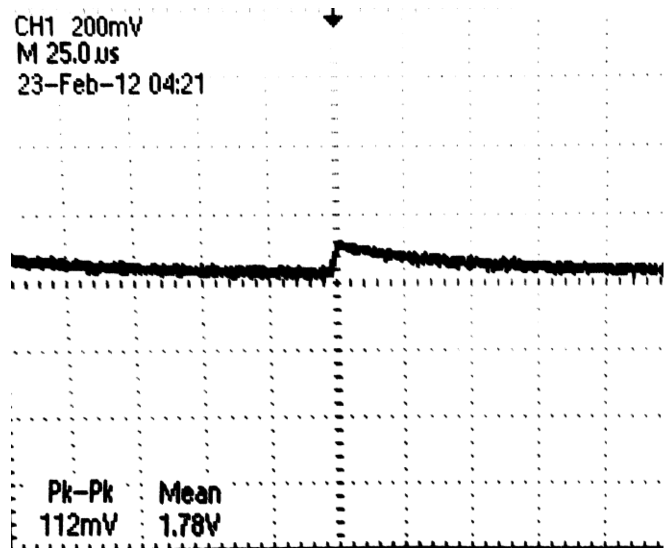

Fig. 19: Board level Rectifier voltage (TP3) at 3 inches

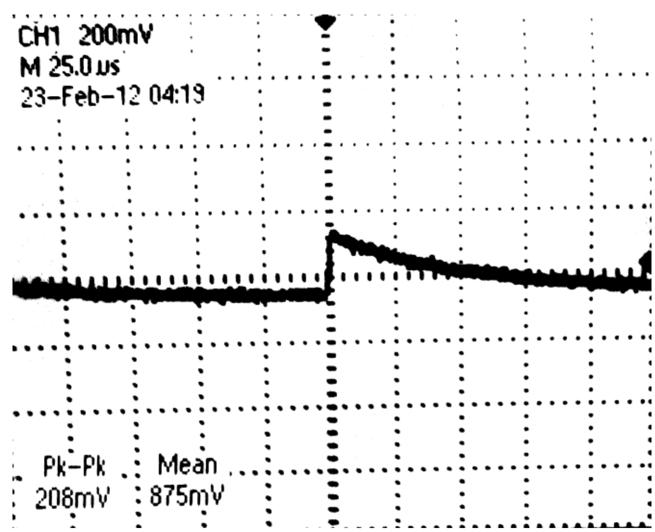

Fig. 20: Board level Rectifier voltage (TP3) at 9 inches

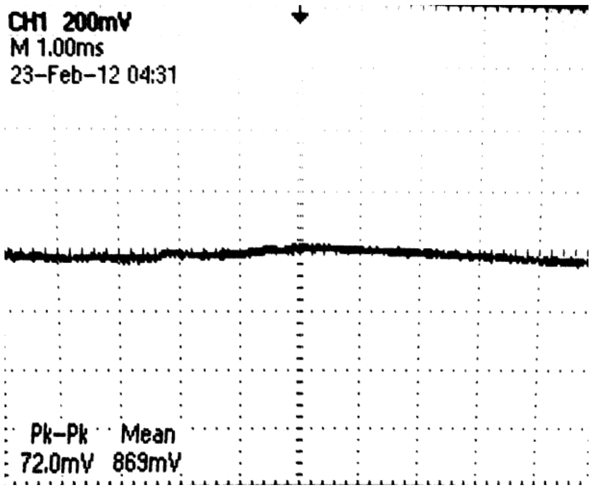

Fig. 21: Board level Rectifier voltage 9 inches after Low Pass Filter

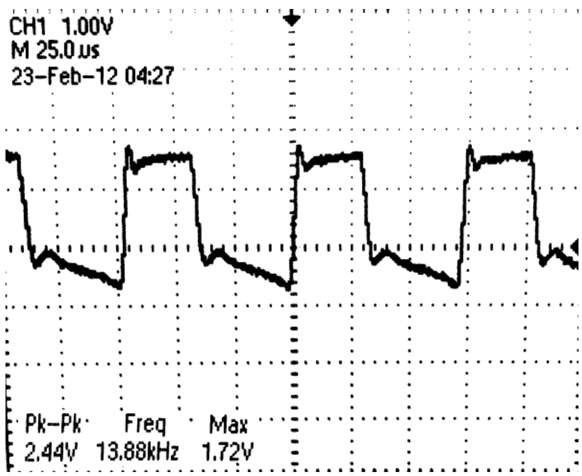

Fig. 22: Board level Transimpedance amplifier output voltage (TP2) at 3 inches

VIII. Discussion

Very few modifications had to be performed on the PCB which resulted in very quick testing and full verification of the circuit. As mentioned earlier, a Low-Pass Filter was added between the output of the rectifier and the input of the ADC to the microcontroller. This helped in suppressing most of the ripple voltage as can be seen by comparing [Fig. 20] with [Fig. 21]. Once this addition was added there were no problems with the microcontroller being able to detect the object distance. An alternative fix would be to lower the ADC resolution from 10-bits to 8-bits. A 10-bit ADC on a $\pm$2.5 V supply gives an effective range of just under 5 mV per integer. With large ripple voltages in the 100 mV range that gives 20 different possible values for a single distance which is difficult to parse. Compare this to the resolution of the 8-bit ADC where each integer represents just less than 20 mV and the same reading will only vary by 5 values. This provides a little more consistency. However, it was decided that an analog LPF mixed with a software smoothing algorithm would provide the most consistent readings as well as the most accurate. Since neither were difficult to implement, it was an easy choice.

One oversight was certainly the lack of a calibration stage. The addition of two tactile push-switches would've allowed the end-user to select different distance thresholds and latch in the value for each of the three thresholds. This would have made the design far more robust as the microcontroller would not need to be reprogrammed to detect different objects and could be set on-the-fly to detect the distance of almost any object. Since the ATMega328 has 1KB of EEPROM these thresholds could've also been stored and recalled allowing for one calibration per object and the calibration data could simply by re-loaded by the microcontroller. Of course some sort of display would need to be implemented to allow the user to see what object they were selecting but it would certainly have made for an interesting addition.

Another improvement that was thought about was adding in a 7-segment display to show the estimated distance in terms of inches. This would've been fairly trivial to implement as a single SIPO shift register could've been used to drive the display while the microcontroller compared the ADC value to different distance thresholds to determine the estimated distance.

The microcontroller was fairly trivial to program. Using an Arduino UNO as a programming platform, a bootloader was burned to an ATMega328P chip and then programmed through the USB interface on the UNO to the microcontroller itself via the programming headers on the PCB. A custom cable was made to make this process easier. Using the UNO allowed for serial debugging over USB as well as the use of the Arduino programming environment which simplifies many AVR functions[3].

IX. References

[1] *Op Amp Applications Handbook*, Walter Chu, Analog Devices, Inc. 2003
[2] http://kevinfronczak.com/documents/IR_Sensor/datasheets/
[3] Microcontroller Code: http://kevinfronczak.com/documents/IR_Sensor/IR_Software_v_1/IR_Software_v_1.c