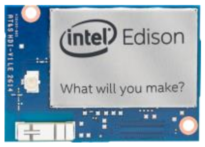


Intel Edison ® IR Receiver Shield Kit

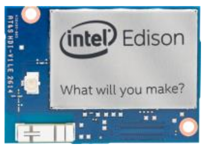
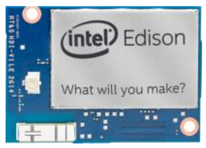


Table of Contents

Revision history		
Version	**Date**	**Comment**
1.0	03/05/2016	Initial release



Introduction

This document provides guidance on assembling and testing the IR Receiver Shield kit for use on the SparkFun GPIO Block. From the Arduino website: “Shields are boards that can be plugged on top of the Arduino PCB extending its capabilities. The different shields follow the same philosophy as the original toolkit: they are easy to mount, and cheap to produce.” This shield for the SparkFun GPIO Block was designed with the same principles in mind.

In this tutorial, you will learn to

1. Assemble the hardware,
2. Test the hardware, and
3. Read an input signal from the IR receiver.

Things Needed

Hardware for Assembly

1. An Intel® Edison
2. A SparkFun® GPIO Block
3. A SparkFun® Battery Block or a Base Block
    a. If you use a base block, you need a micro USB cable supply power.
4. Soldering tools, and
5. A IR Receiver Shield kit
    a. The kit contains:
    b. (2) Header - 10-pin Female
    c. (1) IR Receiver Shield [TSOP38238 or TSOP312__.TSOP314__ version are both fine]
    d. (1) 330 Ω Resistor (R1)
    e. (1) 1 kΩ Ohm Resistor (R2)
    f. (1) 100 µF (microfarad) Capacitor (C1)
    g. (1) TSOP38238 IR Receiver Diode
    h. (1) Heat Shrink Tubing
    i. (2) Arduino Stackable Header - 10 Pin

Hardware for Testing

1. Wires
2. A Digital Multi-meter
3. Either
    a. A television remote control, or
    b. (1) Current Limiting Resistor, 330 Ω or less, and (1) IR LED, 940 nm,
4. (Optional) An oscilloscope (This images in this tutorial are taken using the National Instruments MyDAQ)
5. (Optional) An Arduino Breakout Board
6. (Optional) (1) 50 KΩ Resistor, (1) 10 KΩ Resistor,
7. A PC or a Mac

Hardware Assembly

Part 1: GPIO Block

The SparkFun GPIO Block comes as a breakout board with 20 pins exposed and connected. To use it together with the IR communication add-on board, we need to attach part 4b from the kit – (2) Header - 10-pin Female to the board.

Components

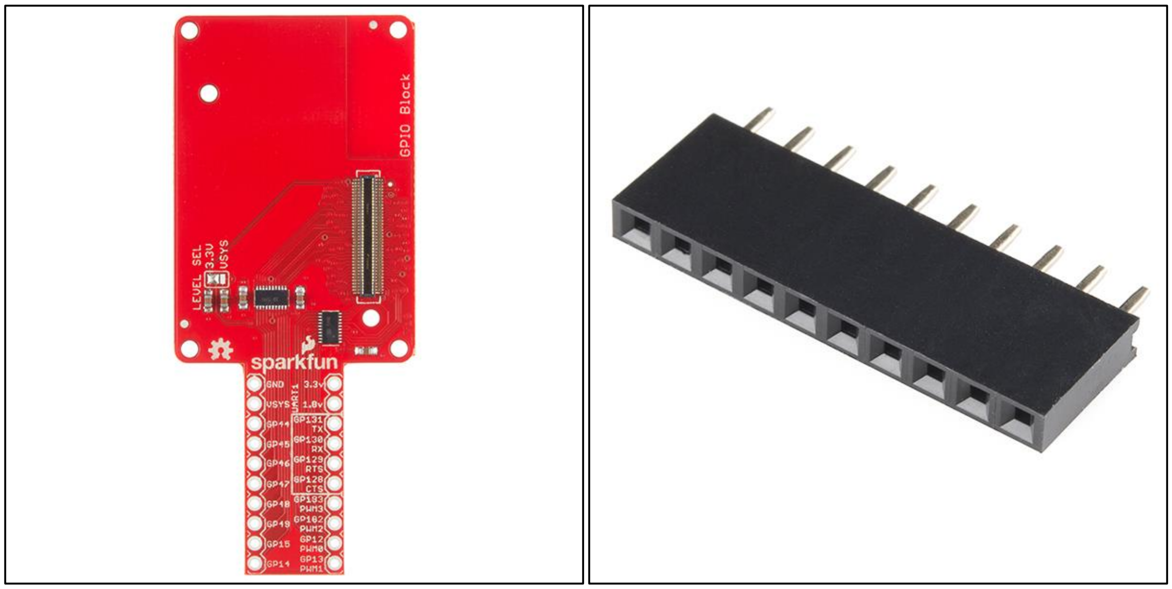


Figure 1 SparkFun GPIO Block and 10-pin Female Headers

Assembly

1. Take the female headers and insert them both into the slots on the GPIO block, taking care that they are on the correct side of the block. Specifically, we want to fix the headers such that they are on the female side of the GPIO block, so that Edison would be on the same side if attached to the block directly.

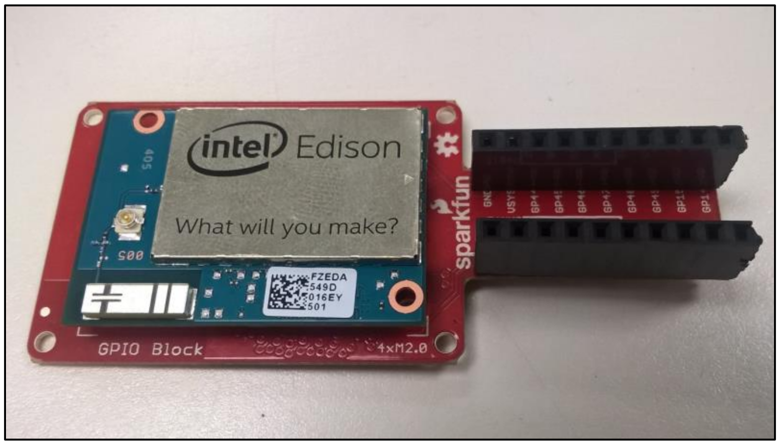


Figure 2 Female Header Pins inserted on the SAME side as Intel Edison

2. Flip the GPIO block and header pins over so that the Edison is face down and the pins are exposed to be soldered.
3. Solder all 20 pins, 10 pins on each side.
   a. Apply the hot tip of the soldering iron to the point where the pin and the GPIO board intersect such that the tip contacts both surfaces and heats them both.

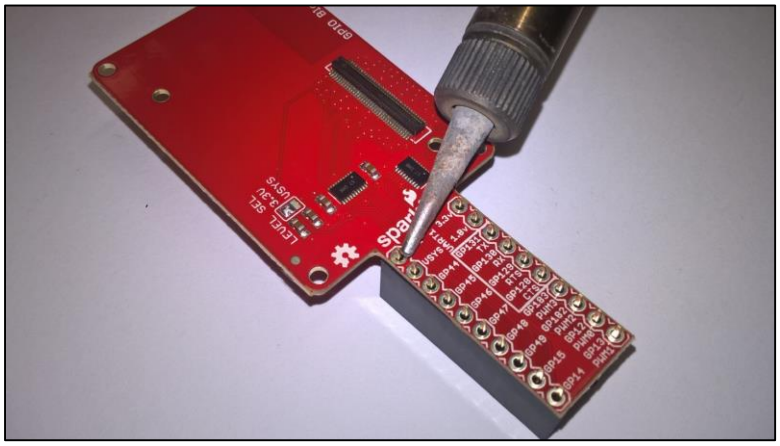


Figure 3 Applying hot soldering iron tip to GPIO board and pin contact point

   b. Simultaneously apply the solder to the intersection point on the heated metal surface and allow it to melt, filling in the gap cleanly as pictured.

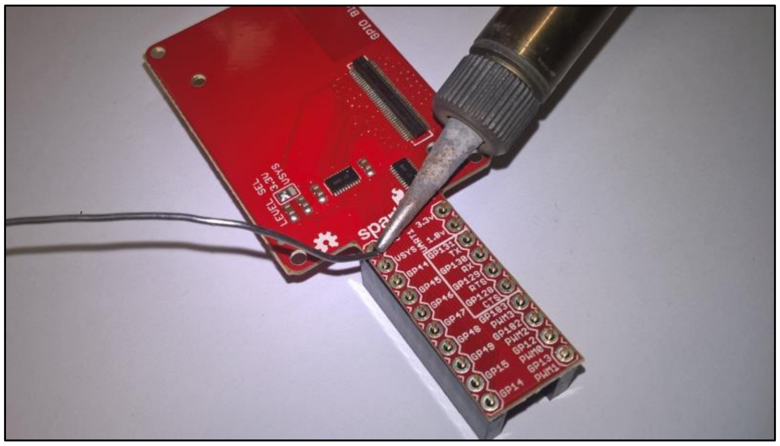


Figure 4 Applying solder to the pin/board contact point

c. To prevent damage to the board from the heat of the soldering iron, it is suggested that when soldering to the board, do not solder adjacent pins sequentially, but instead after soldering each pin, move to a pin on the opposite side of the board which has had time to cool off.

   If at any time the entire board is extremely hot, please let it cool before continuing to solder.

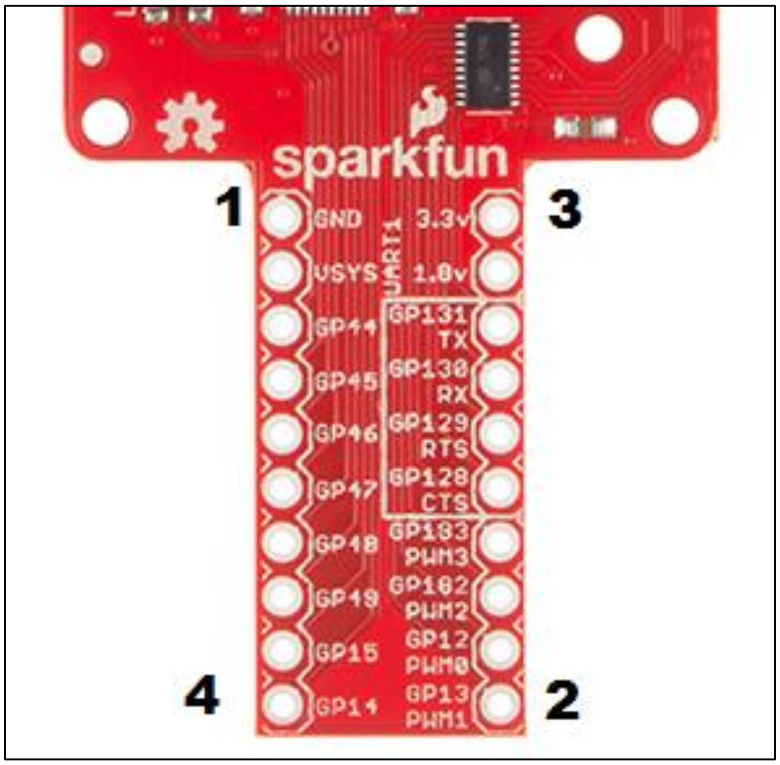


Figure 5 Suggested Pin Soldering Order

4. Afterwards, the GPIO Block will be completely ready to use together with the shield.

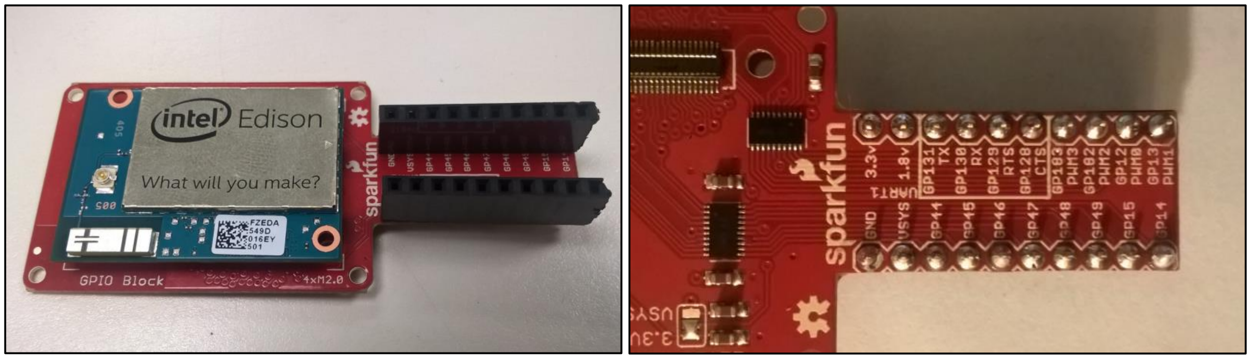


Figure 6 Complete GPIO block with Female Headers

Part 2: IR Receiver Shield

The IR Receiver Shield PCB was custom made to facilitate rapid progress of IR communication development. This PCB was created using the free open source program Fritzing. Pictured here is the circuit diagram of the PCB, which was created using Fritzing.

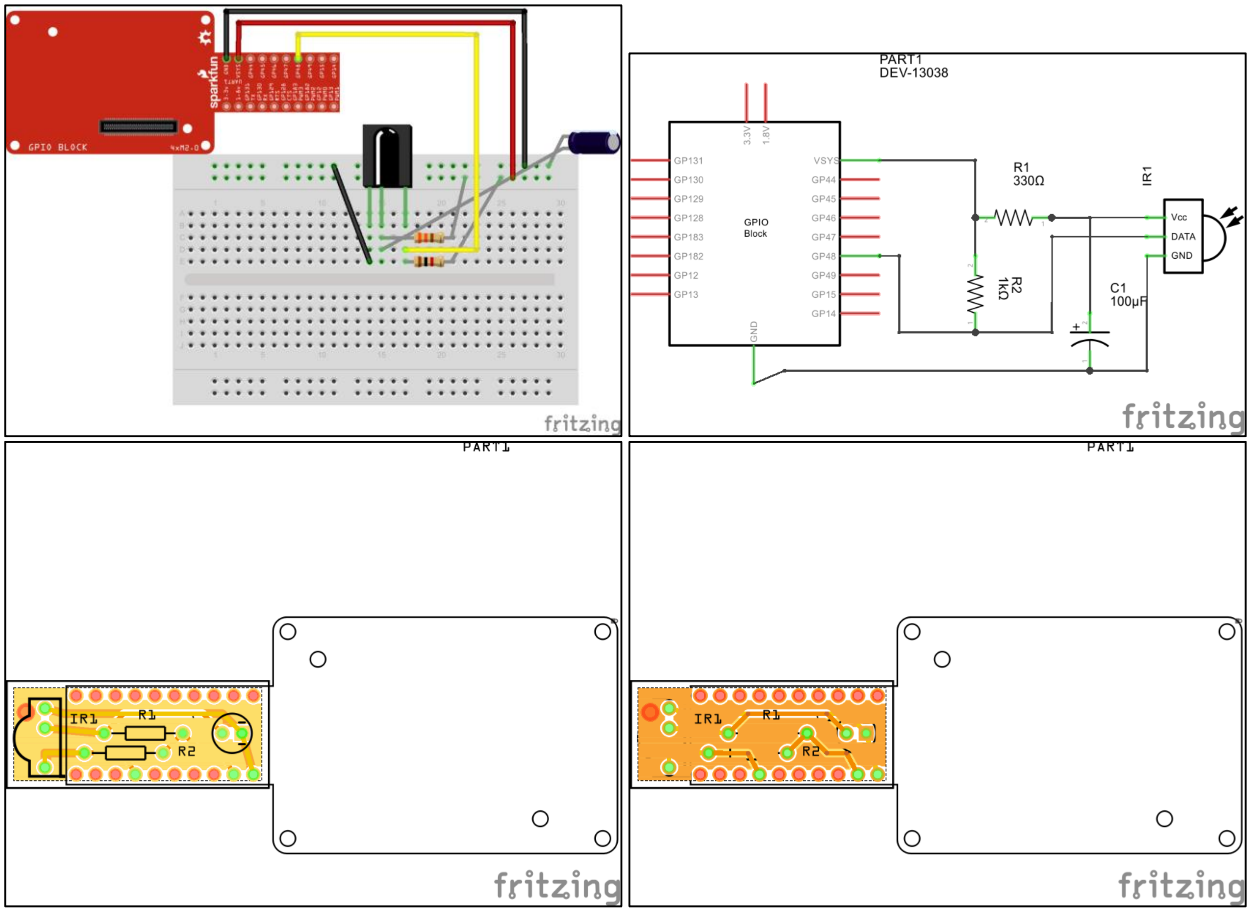


Figure 7 Circuit Layout, Diagram, and PCB Design, Top View and Bottom View

You may notice that this circuit is different than the circuit presented in the datasheet for the TSOP38238; it has an extra resistor. This resistor is acting as a pull-up resistor on DATA pin of the IR receiver, which is required for the receiver to operate on the SparkFun GPIO board. A more detailed investigation about the circuit can be found in Appendix A. If you want to perform the investigation, you may want to do it before completing this section, because you need to use these components on a breadboard.

The investigation is completely optional and your ability to use the IR receiver will not be hindered in any way if you choose to skip it.

The IR Receiver Shield as well as other shields are available on GitHub at the URL: https://github.com/resolutedreamer/IR-Shields

Note that there are two versions of the shield; the different versions are because the different TSOP IR Receiver diodes have different pin-outs. Otherwise, the shields function identically.

Components

We need all of the remaining components from our IR communication add-on kit, parts 4c-4i.

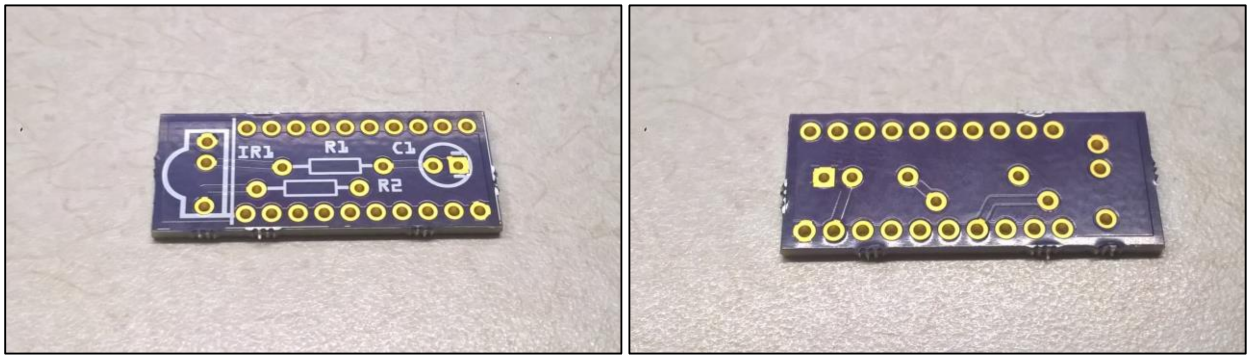


Figure 8 Part 4c, IR Receiver Shield, Top View and Bottom View

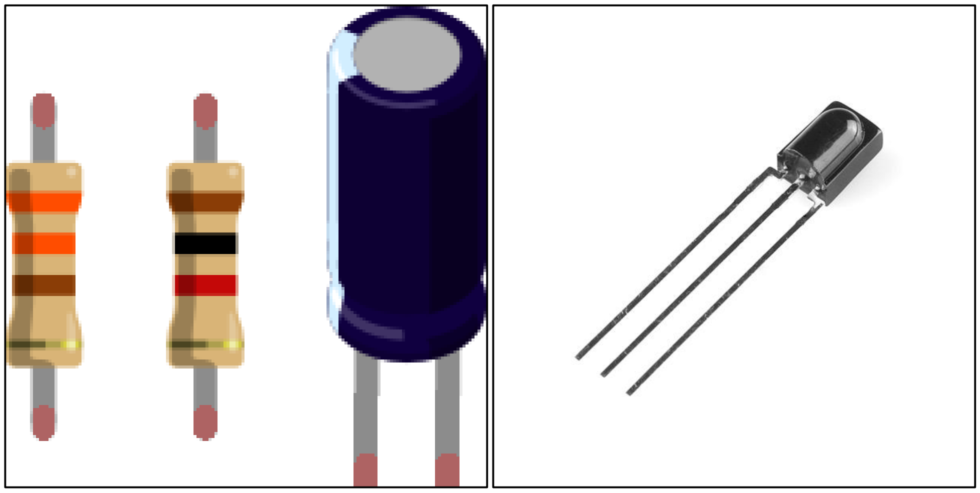


Figure 9a Parts 4d – 4g, pictured from left to right

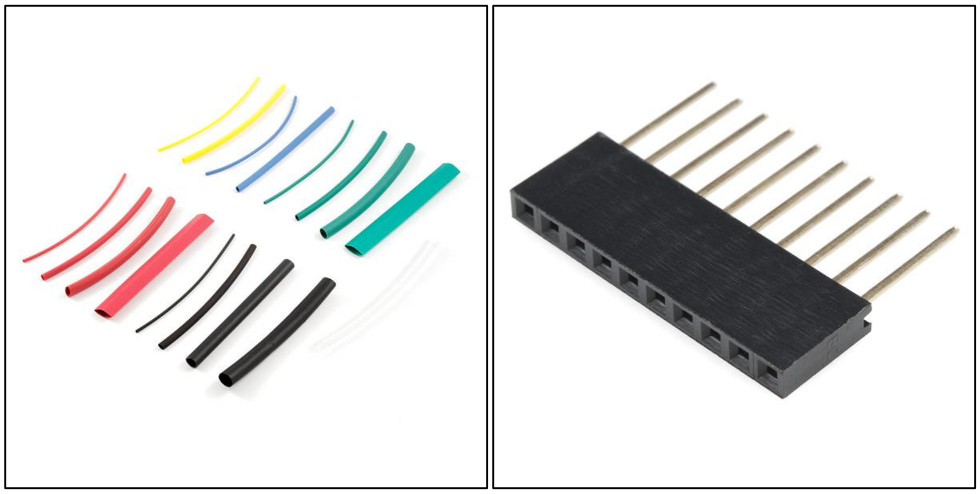


Figure 9b Parts 4h – 4i, pictured from left to right

Several pieces of heat shrink tubing are pictured but only one small piece will be needed.

Assembly

Part 1: Passive Components: 4d/4e/4f

1. Take the 330 Ω resistor, R1, and pass the leads through from the top of the PCB to the bottom of the PCB, and then bend the leads 90 degrees to firmly hold the components in place. The direction of the resistor does not matter.

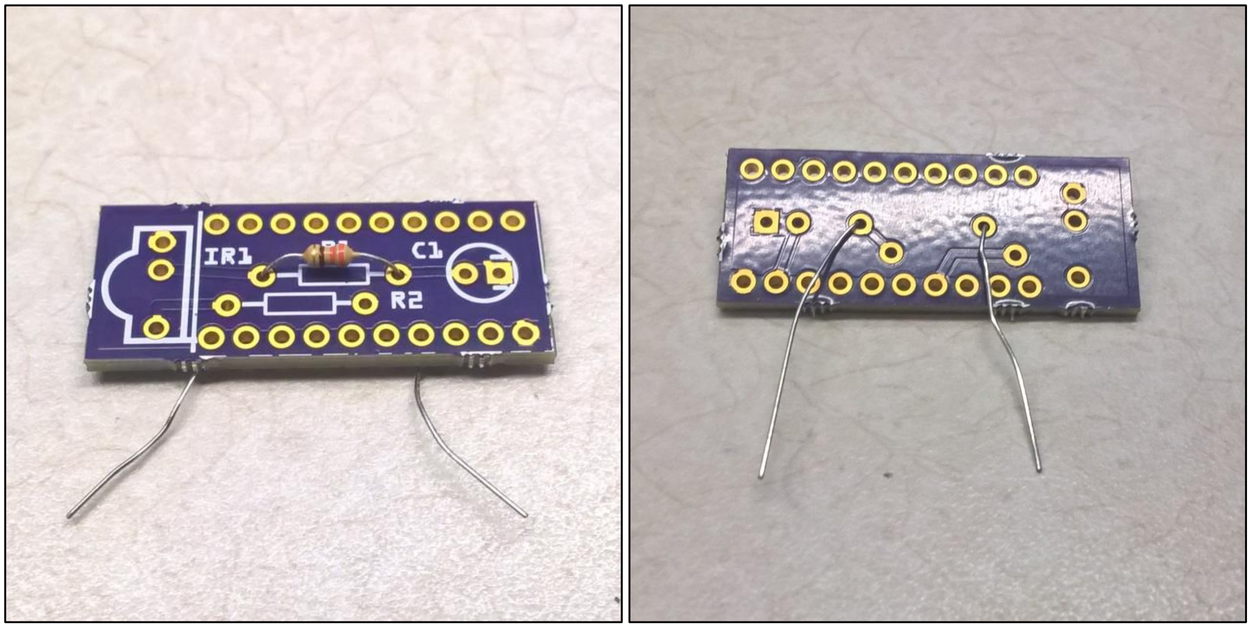


Figure 10 330 Ohm Resistor fixed to PCB, Top and Bottom Views

2. Solder the two connections from the bottom and then cut the excess leads.

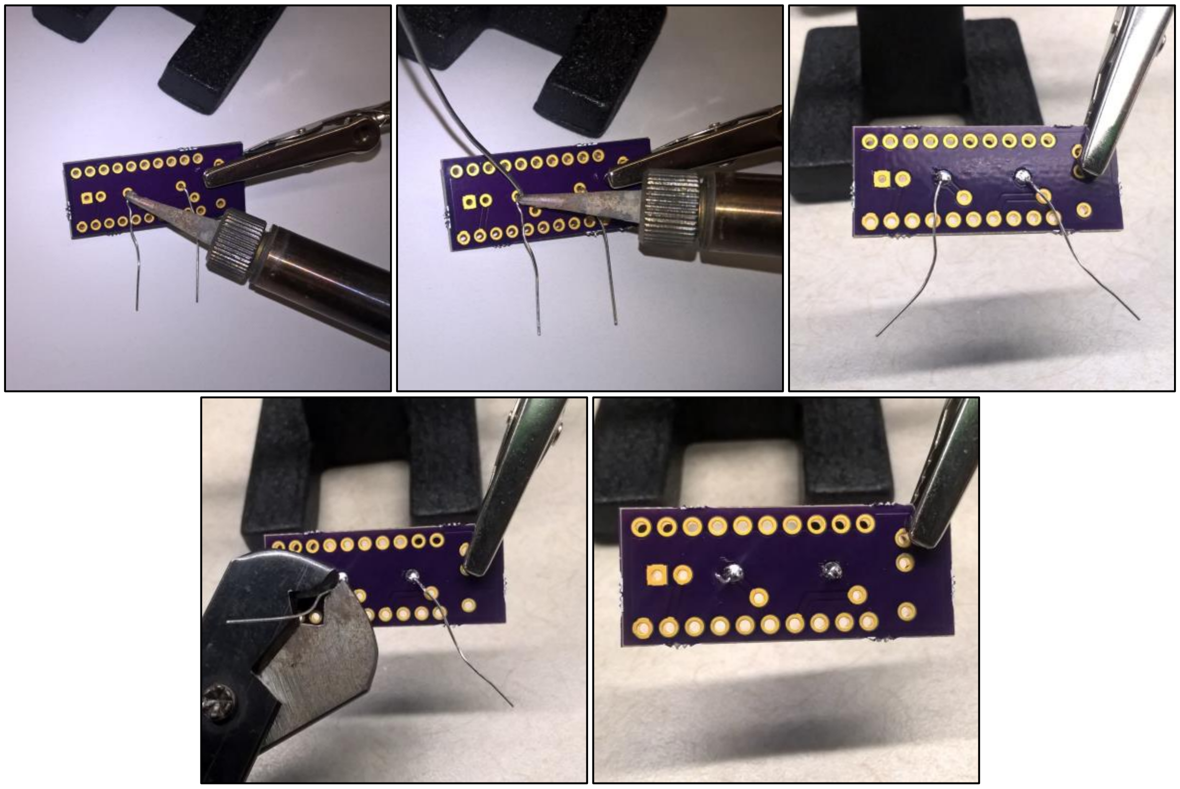


Figure 11 Sodering the 330 Ohm Resistor

3. Repeat for the 1 KΩ resistor R1.
4. For the capacitor C1, this time the direction of the capacitor does matter. Take care that the negative pin of the electrolytic capacitor (the pin by the gold strip, with the shorter lead) matches up with the negative markings on the PCB.

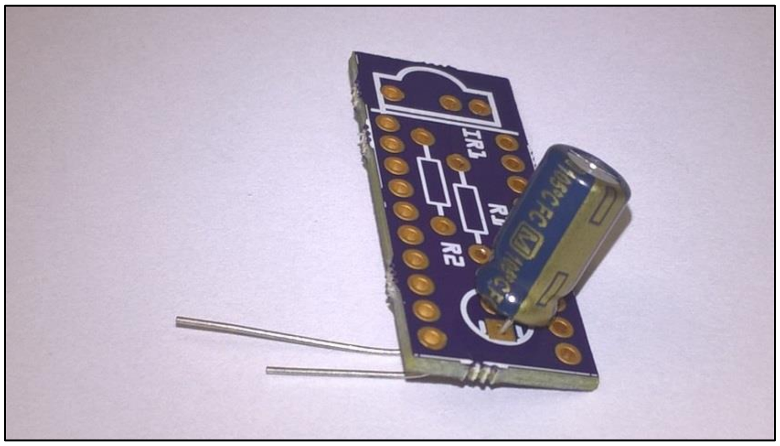


Figure 12 Capacitor, oriented correctly

Part 2: IR Receiver Diode

We could put the IR receiver in directly and it would work fine, however the robustness of the circuit is greatly improved by adding the heat shrink tubing around each of the 3 leads of the receiver.

1. Using the wire cutter, cut the heat shrink tubing into 3 equal length pieces and then slot each piece of tubing onto a lead of the receiver.

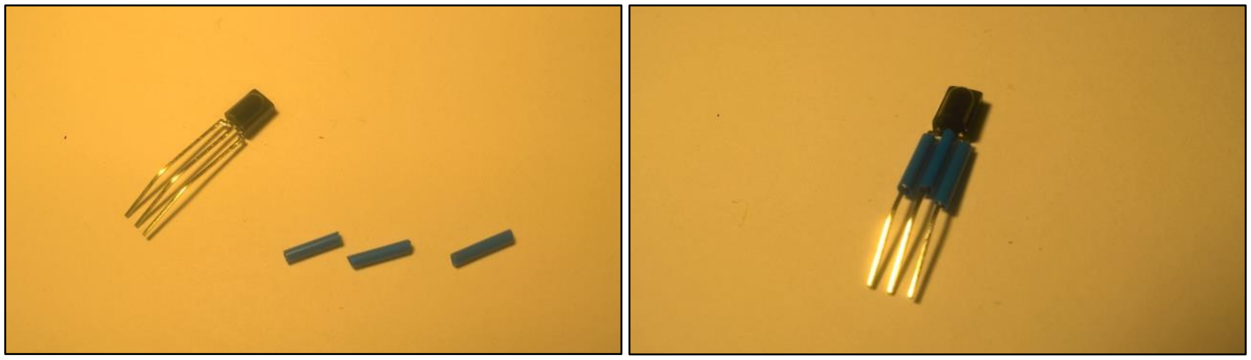


Figure 13 Safely covering IR Receiver leads with heat shrink tubing

2. Connect the three leads of the IR receiver through the appropriately labeled pins

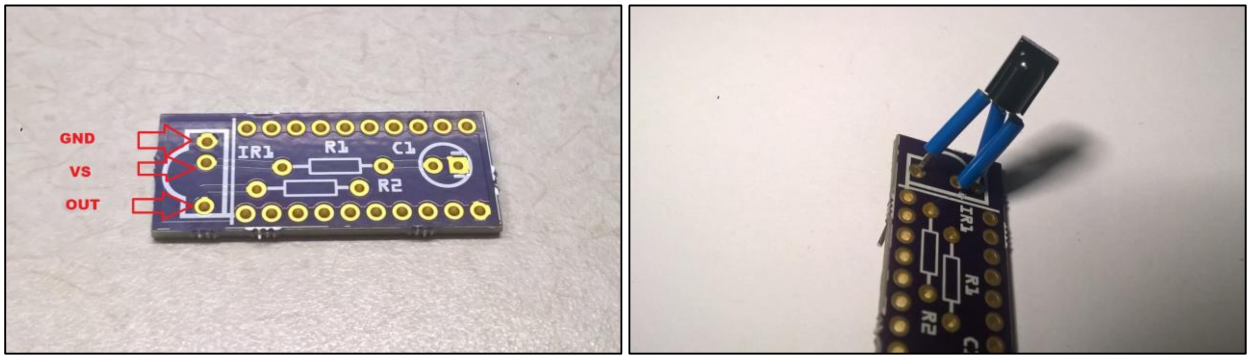


Figure 14 Connecting the IR receiver

3. Solder in the leads and cut the excess length.

Part 3: Arduino Stacking Headers

1. Once the other components are on, the headers can be added last.
    a. When soldering them, the soldering pattern can be used as for the female headers on the GPIO block. Take care not to use too much solder when attaching the headers otherwise it may be more difficult to stack them.

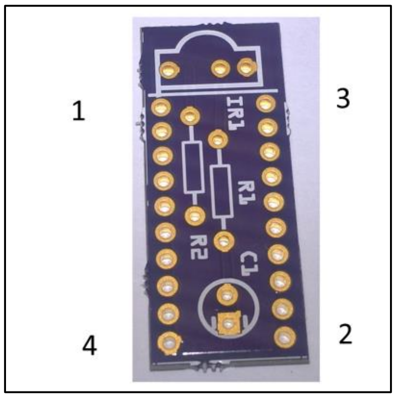


Figure 15 Suggested Pin Soldering Order

2. After soldering the pins in, the assembly of the IR Receiver Kit onto the PCB will be completed.

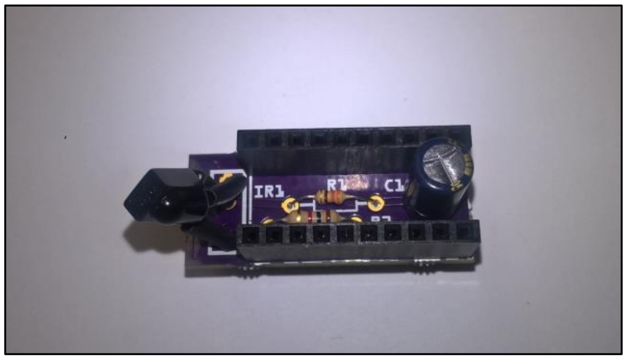


Figure 16 Completed PCB8 Part 4c, IR add-on PCB

Part 3: Final Assembly

At this point, you have finished all of the soldering and can now assemble the final product.

First, assemble the Intel Edison onto the GPIO block. The following three steps are repeated from the SparkFun GPIO Block Programming Guide and do not show the female headers.

It is important to note that you must assemble the hardware BEFORE you supply power. Otherwise, you may damage the devices.

1. Insert your Edison module into the GPIO block.

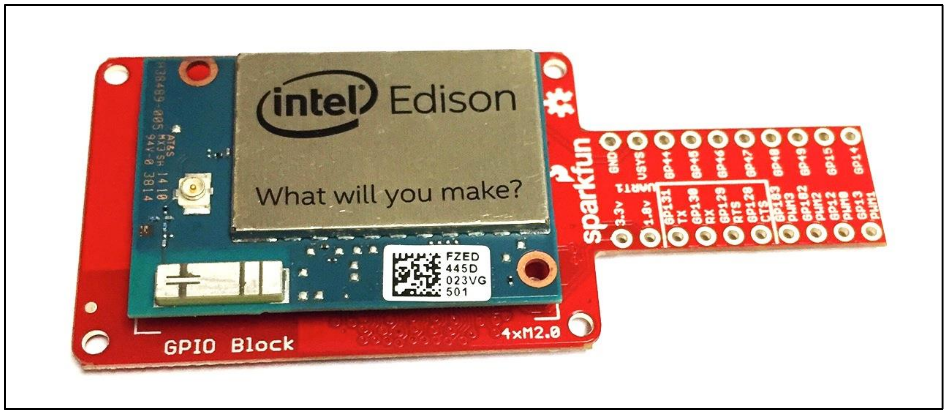


Figure 17 Edison and GPIO Block

2. Connect a battery block or a base block.

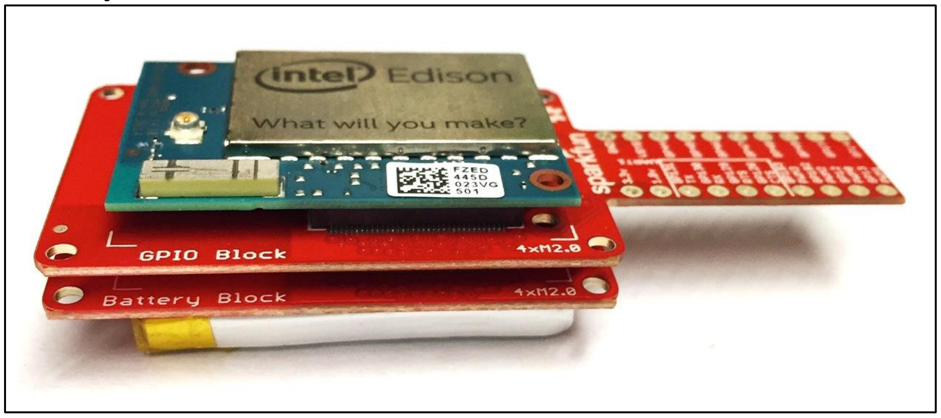


Figure 18 Edison, GPIO Block, and Battery Block

3. This is the general hardware installation. You can add more SparkFun® blocks. If you have a hardware pack (screws and nuts), you can use them to secure the connection.

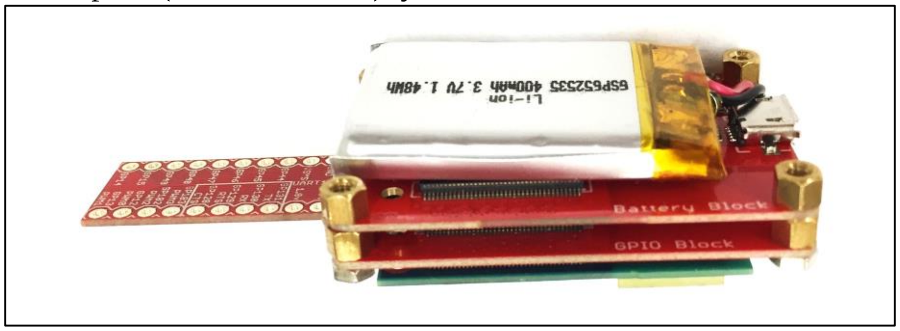


Figure 19 Hardware Pack

4. With three SparkFun blocks and the Edison put together, the final result will resemble the left figure. On the right is the completed PCB, ready to stack on top of the GPIO block.

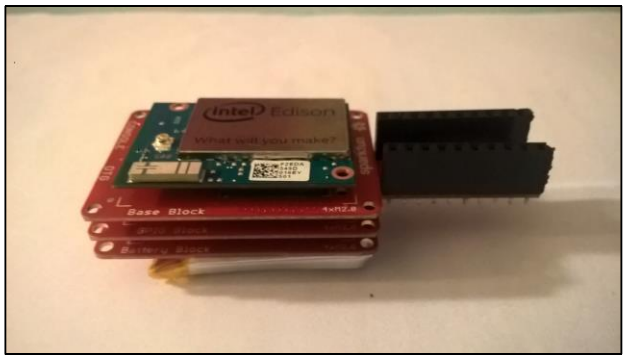
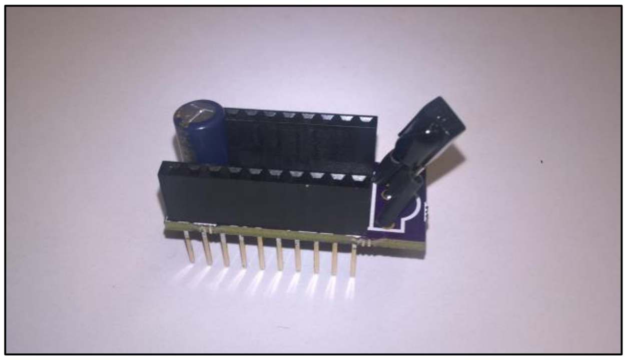



Figure 20 Fully Stacked SparkFun Blocks and IR Receiver Shield

5. Place the completely assembled PCB on top of the female headers on the GPIO block. Take care to make sure that the end with the IR receiver attached is facing away from the Intel Edison, and not towards it.

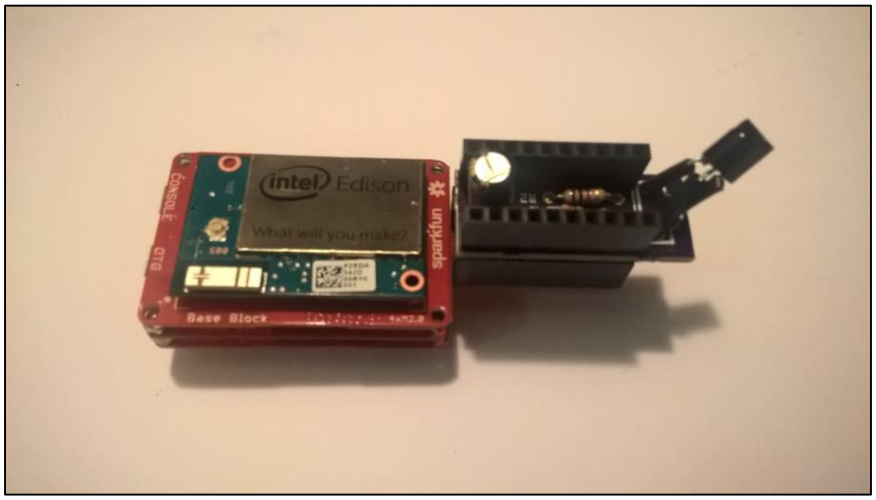
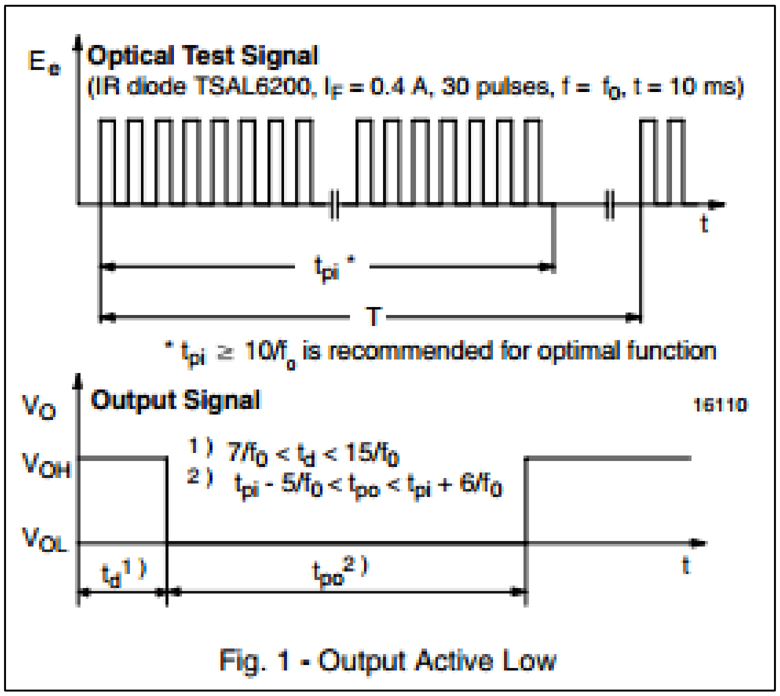


Figure 21 IR Receiver Shield attached to SparkFun GPIO Block – Proper Orientation

Hardware Testing

After assembling IR Receiver Shield, the next step is to test it to verify proper operation. By investigating the datasheet for the TSOP38238 IR Receiver Diode, the figure in the bottom left corner of page 2 shows that the TSOP38238 is an active low device. Therefore, the voltage at the OUT pin (also called the DATA pin) of the TSOP38238 should be **HIGH** if there is no IR signal present or **LOW** if there is IR signal present.

The **LOW** value will be approximately **0 V**, and the **HIGH** value will be approximately equal to the input voltage $\mathbf{V_S = 4\ V}$. Therefore, to test to the board, we want to measure the voltage at the output and verify that the voltage is $\mathbf{V_S = 4\ V}$ when there is no IR signal being sent at it, and that it goes to **0 V** when there is an IR signal present.



E_e
Optical Test Signal
(IR diode TSAL6200, I_F = 0.4 A, 30 pulses, f = f_0, t = 10 ms)
t
t_{pi} *
T
* $t_{pi} \geq 10/f_o$ is recommended for optimal function
V_O
Output Signal
16110
V_{OH}
V_{OL}
1) $7/f_0 < t_d < 15/f_0$
2) $t_{pi} - 5/f_0 < t_{po} < t_{pi} + 6/f_0$
t_d 1)
t_{po} 2)
t
Fig. 1 - Output Active Low

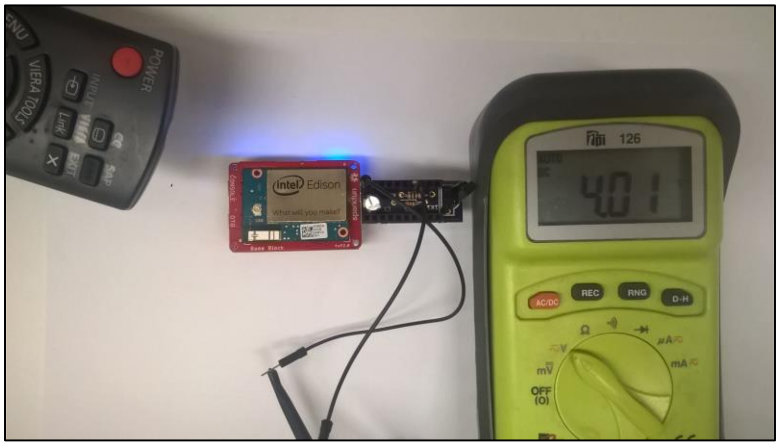
Figure 22 Sample output signal of IR Receiver

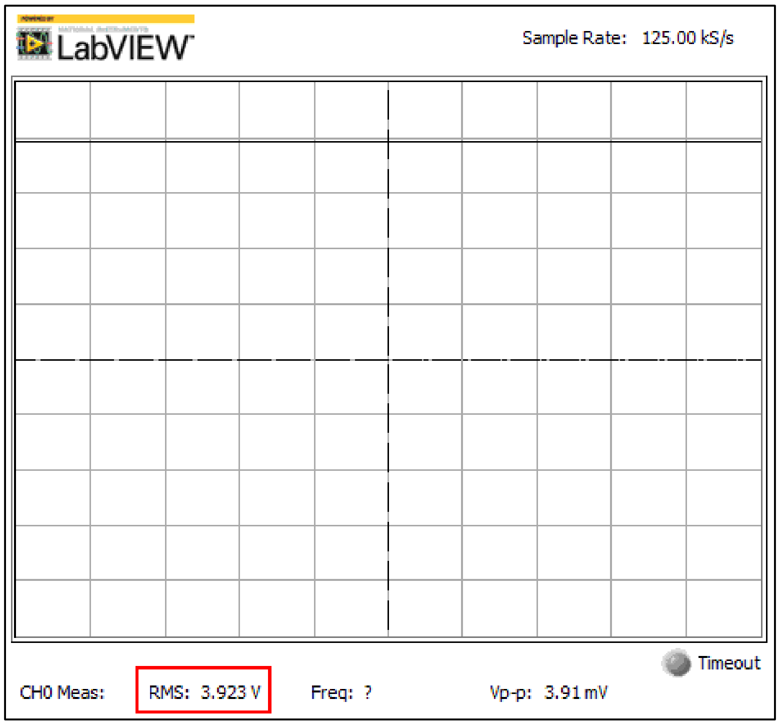
Television Remote Control

The easiest way to test the functionality of the board is to use a TV remote control.

1. Configure the digital multimeter to measure voltage, and connect the positive terminal to GP48 and the negative terminal to GND. If the Edison is off the digital multimeter should measure **0 V**.
    a. (Optional) If there is an oscilloscope available, it can also be used to visualize the voltage vs time, which will be very valuable to see the shape of IR signals. Configure the window so that the time-axis is on the order of tens of milliseconds, and the voltage-axis will show a **4 V** signal.

2. Turn on the Edison and the voltage measured by the multimeter should be roughly **4 V**, the same voltage at the VSYS pin. Point the TV remote so that it is pointed toward the curved side of the IR Receiver as in Figure 22.
    a. (Optional) If the signal is viewed on the oscilloscope, it should also display a constant value of approximately **4 V**.



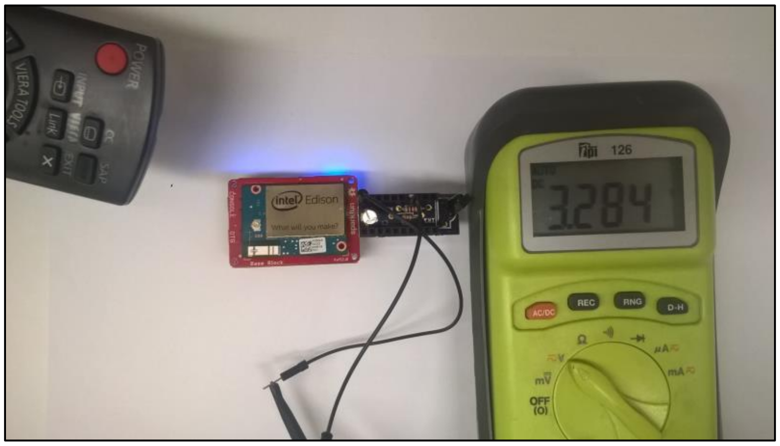


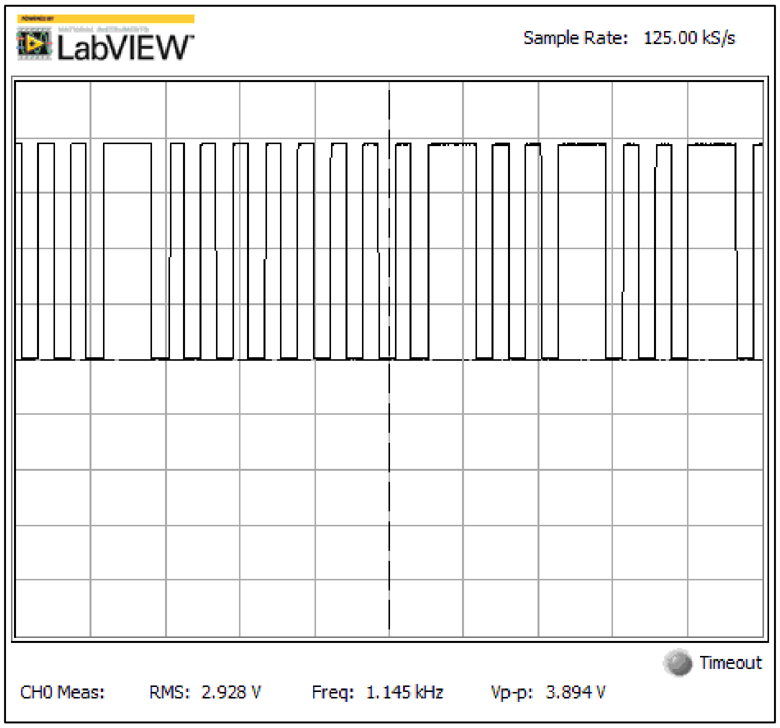
Sample Rate: 125.00 kS/s

CH0 Meas: RMS: 3.923 V Freq: ? Vp-p: 3.91 mV

Figure 23 IR Receiver output in the absence of signal

3. Press any button on the TV remote and the voltage measured by multimeter will decrease. It will not decrease all the way to zero because the signal is alternating between high and low values.
    a. (Optional) If viewed using the oscilloscope, you will be able to see sequences of high and low values. The observed signal will likely not be identical to the one in the screenshot provided because different TV remote controls will output different signals. This will be explored in more depth in the tutorial Intel Edison® IR Communication Fundamentals.





Sample Rate: 125.00 kS/s

CH0 Meas: RMS: 2.928 V Freq: 1.145 kHz Vp-p: 3.894 V

Figure 24 TV Remote Signal

4. The testing is complete. The IR Receiver Shield is now ready for immediate use. To learn more about IR communication, please consult the Intel Edison® IR Communication Fundamentals tutorial.

IR LED

If you do not have a TV remote control handy, you can use an IR LED and current limiting resistor instead. Please prepare the following circuit:

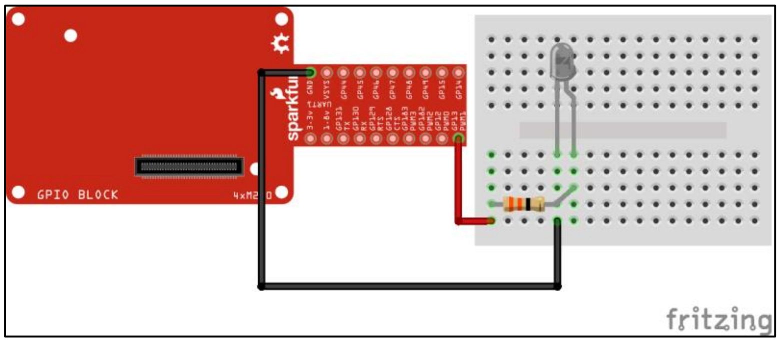


Figure 25 IR LED Circuit

This can be connected to the same Edison connected to the IR Receiver or it can be done on a different Edison.

5. Make sure the IR LED is pointed at the curved end of the IR receiver. This image depicts the setup using two Edisons:

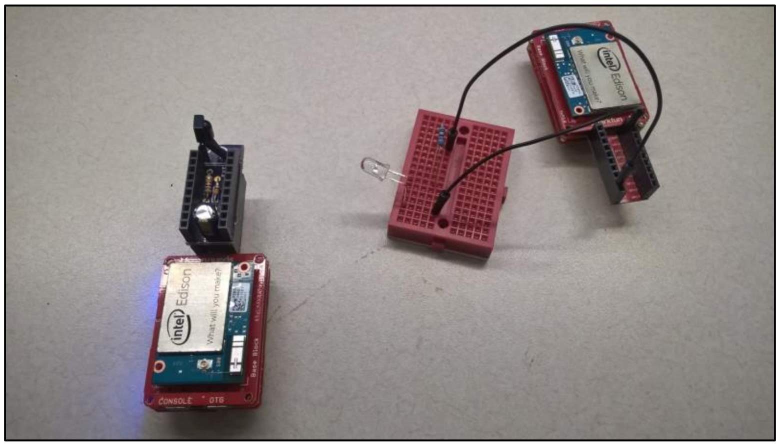
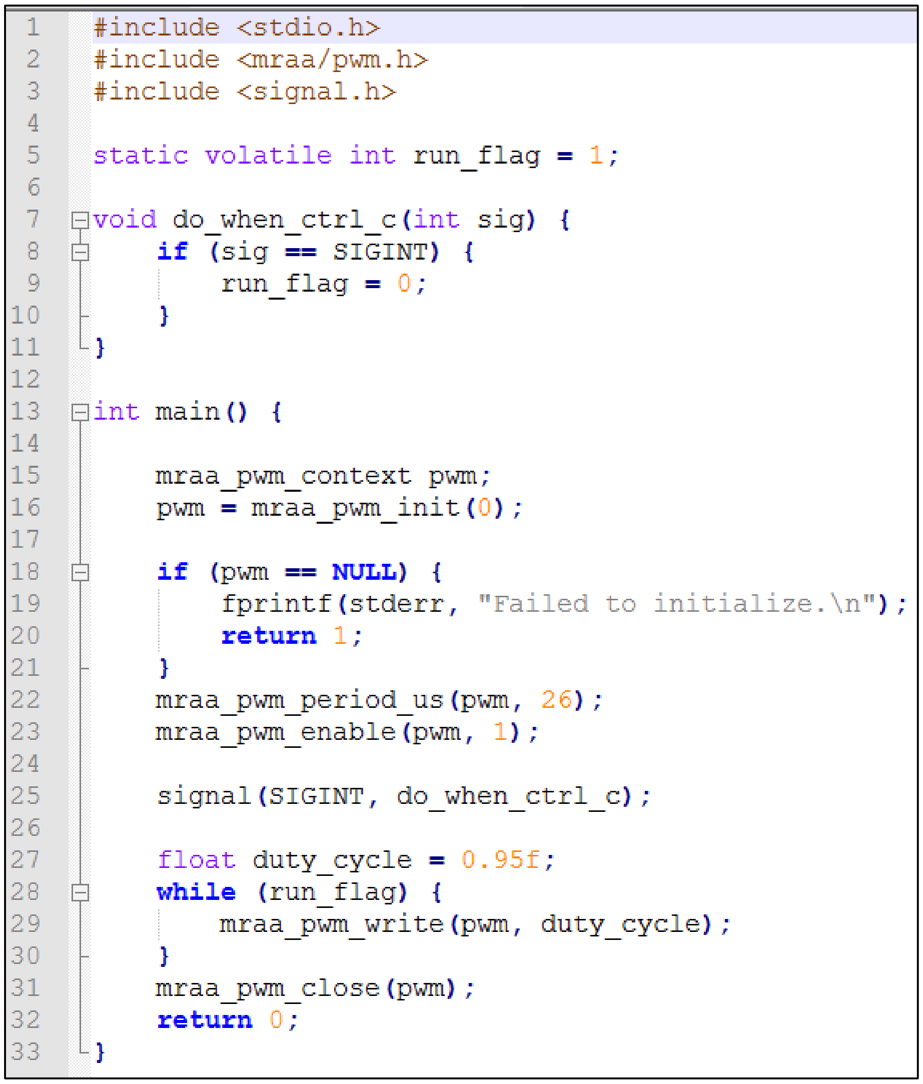


Figure 26 IR LED pointed toward IR Receiver

6. The hardware setup for testing is complete
7. Log into the Edison connected to the IR LED via Serial or SSH connection.
8. **$ vi ir_emitter_test.c**

9. Type the following code:



```c
#include <stdio.h>
#include <mraa/pwm.h>
#include <signal.h>

static volatile int run_flag = 1;

void do_when_ctrl_c(int sig) {
    if (sig == SIGINT) {
        run_flag = 0;
    }
}

int main() {

    mraa_pwm_context pwm;
    pwm = mraa_pwm_init(0);

    if (pwm == NULL) {
        fprintf(stderr, "Failed to initialize.\n");
        return 1;
    }
    mraa_pwm_period_us(pwm, 26);
    mraa_pwm_enable(pwm, 1);

    signal(SIGINT, do_when_ctrl_c);

    float duty_cycle = 0.95f;
    while (run_flag) {
        mraa_pwm_write(pwm, duty_cycle);
    }
    mraa_pwm_close(pwm);
    return 0;
}
```

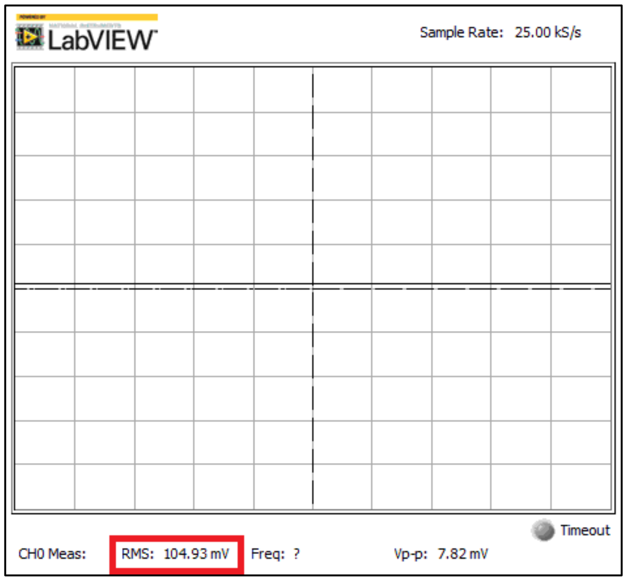
Figure 27 ir_emitter_test.c

Note: Closing the program without closing the PWM pin will result in undefined behavior. Typically, the pin will continue outputting the previous value. Attempting to make use of the PWM pins again will not automatically reset the pin; the new value output by the pin will likely not be as expected.

10. $ gcc -o ir_emitter_test ir_emitter_test.c

11. $ ./ir_emitter_test

12. Now, the IR LED will be emitting a constant signal. The multimeter will display a reading of approximately **0 V** because it will be receiving a constant signal.
    a. You can also see the same reading on the oscilloscope:



Sample Rate: 25.00 kS/s

CH0 Meas: RMS: 104.93 mV Freq: ? Vp-p: 7.82 mV

Figure 28 Low Voltage Level indicating constant received signal

13. The testing is complete. The IR Receiver Shield is now ready for use. To learn more about IR communication, please consult the Intel Edison® IR Communication Fundamentals tutorial.

Appendix A: SparkFun GPIO Block and Pull Up Resistors

This appendix provides a short lab investigation about pull up resistors and why they are needed to properly use the SparkFun GPIO block.

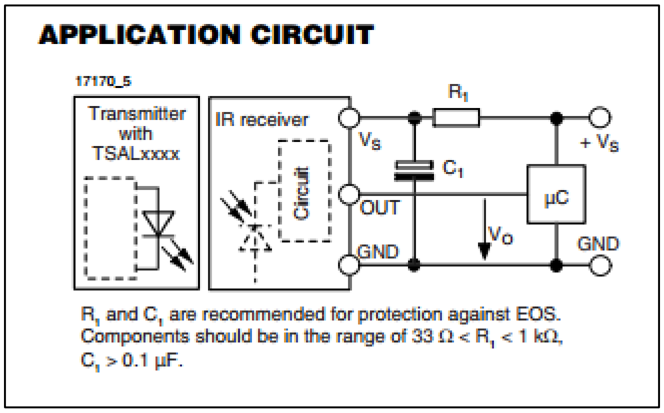

APPLICATION CIRCUIT
17170_5
Transmitter with TSALxxxx
IR receiver
Circuit
R1
C1
VS
OUT
GND
VO
μC
+ VS
GND
R1 and C1 are recommended for protection against EOS. Components should be in the range of 33 Ω < R1 < 1 kΩ, C1 > 0.1 μF.

Figure 28 Recommended Application Circuit for TSOP38238. Taken from official datasheet

As discussed in the main tutorial, the TSOP38238 is an active low device. This tutorial will demonstrate that without adding an additional pull up resistor to the circuit, the TSOP38238 will not behave correctly.

1. Please begin by connecting the IR Receiver using the circuit suggested in the datasheet, using the Arduino Breakout Board:

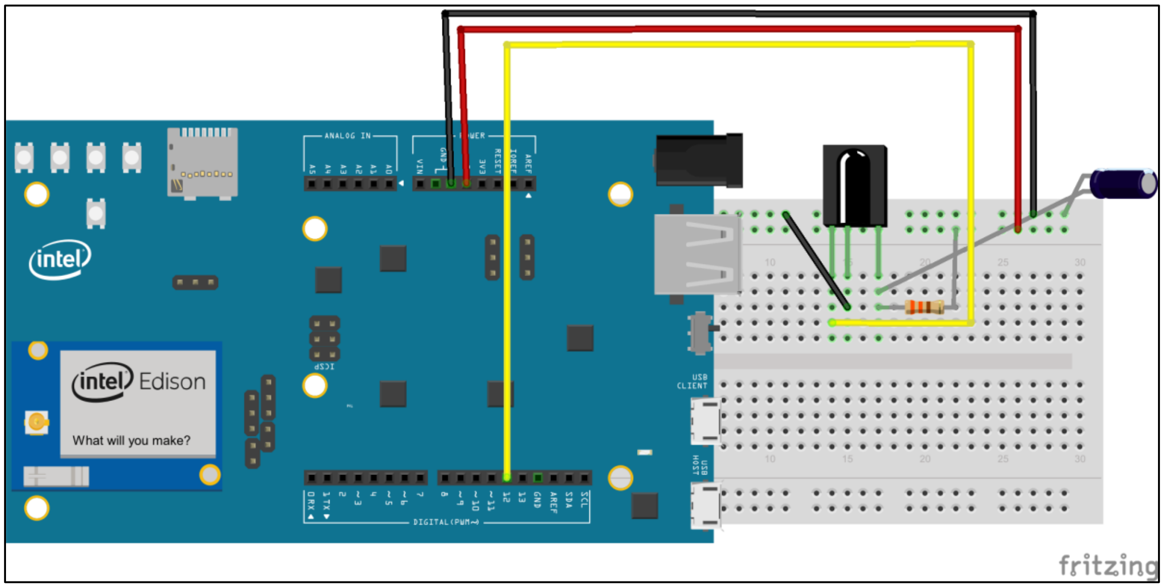


Figure 29 IR Receiver Circuit, No Pull Up Resistor

Notice that there is no pull up resistor in this circuit, whereas in the circuit of the IR Receiver Shield, a 1 KΩ resistor is included.

2. Follow the procedure provided in the “Hardware Testing” section above. This test should proceed smoothly; the **HIGH** voltage level will be **5 V** instead of **4 V**.
3. After confirming proper operation of this circuit, replace the Arduino Breakout Board with the SparkFun GPIO Block.

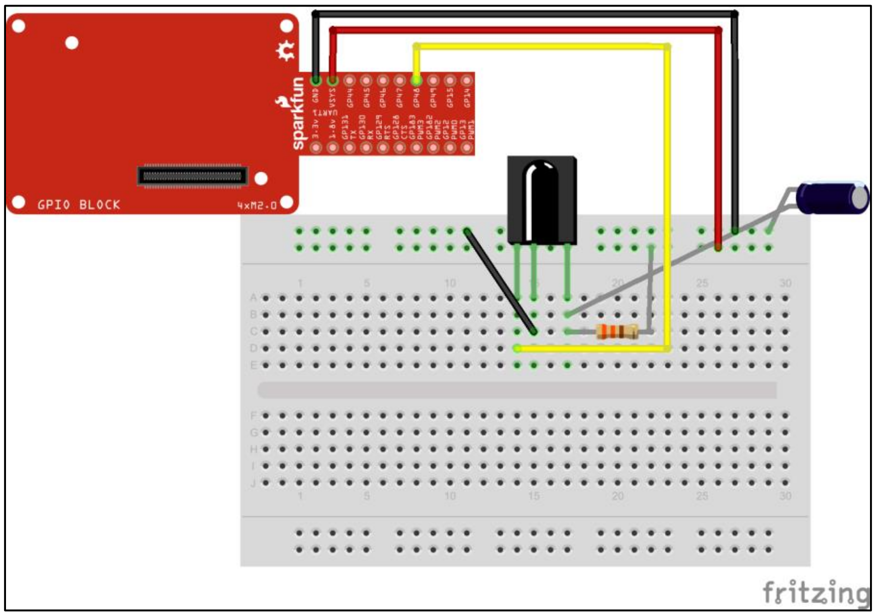
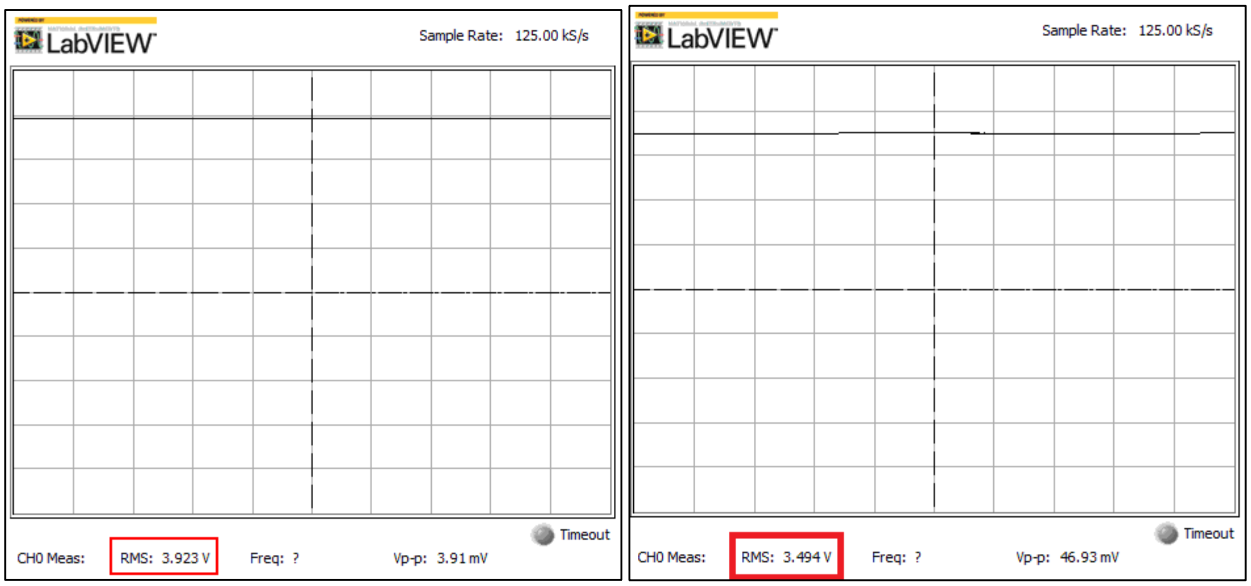


Figure 30 IR Receiver Circuit, No Pull Up Resistor

4. Compared to the voltage observed on the Arduino Breakout board, the HIGH voltage level should drop as well as becoming less stable.

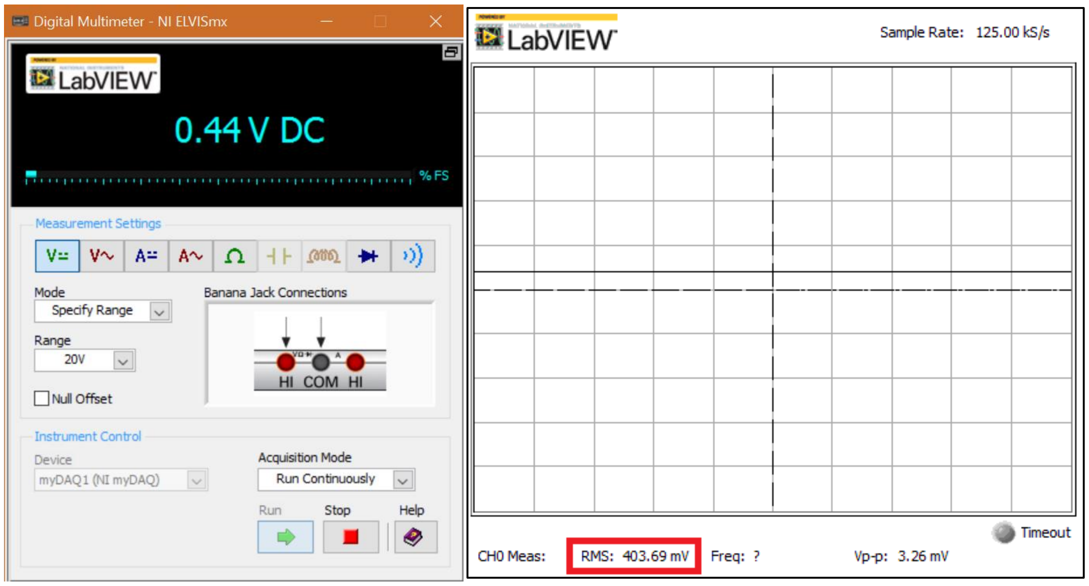


Sample Rate: 125.00 kS/s
CH0 Meas: RMS: 3.923 V Freq: ? Vp-p: 3.91 mV

Sample Rate: 125.00 kS/s
CH0 Meas: RMS: 3.494 V Freq: ? Vp-p: 46.93 mV

Figure 31 Voltage Before and After Connecting to IR Receiver OUT Pin

5. Use either the TV Remote or IR LED to transmit a signal to the IR Receiver. Observe the voltage level on the multimeter or on the oscilloscope, and notice that the voltage level drops significantly to approximately **0.3 – 0.4 V**.



0.44 V DC

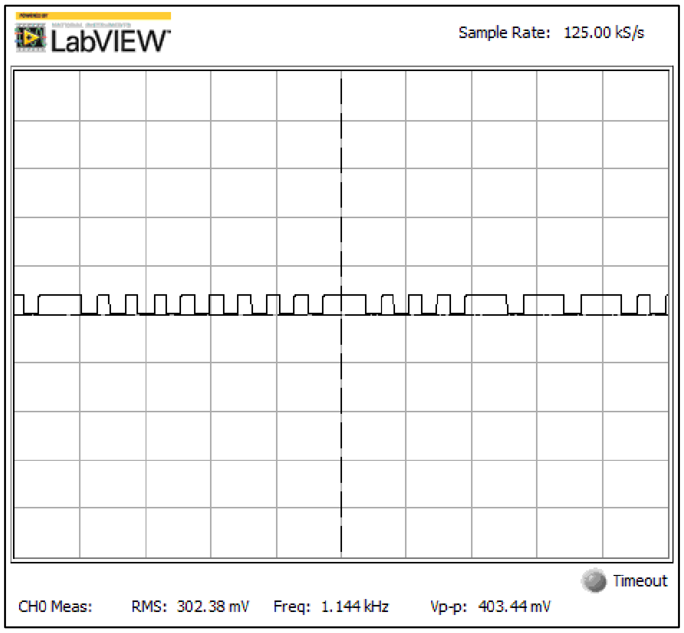
Mode: Specify Range
Range: 20V
Null Offset
Device: myDAQ1 (NI myDAQ)
Acquisition Mode: Run Continuously

Sample Rate: 125.00 kS/s

CH0 Meas: RMS: 403.69 mV Freq: ? Vp-p: 3.26 mV

Figure 31 Voltage at IR Receiver OUT Pin after receiving a signal

a. If both the oscilloscope and the TV remote control are available, take a close look at the oscilloscope when the TV remote is transmitting:



Sample Rate: 125.00 kS/s

CH0 Meas: RMS: 302.38 mV Freq: 1.144 kHz Vp-p: 403.44 mV

Figure 33 Signal at IR Receiver OUT Pin

The IR Receiver is actually still working, and the signal decoding is still functioning properly. The problem is that the **HIGH** value is now **0.3-0.4 V**, while the **LOW** value remains at **0 V**. Because the GPIO block operates at **3.3 V** logic, the Intel Edison will constantly report a LOW value.

By exchanging the Arduino Breakout Board for the SparkFun GPIO board, the IR Receiver circuit did not appear to change, however, the voltage behavior is drastically different. What is the reason for this?

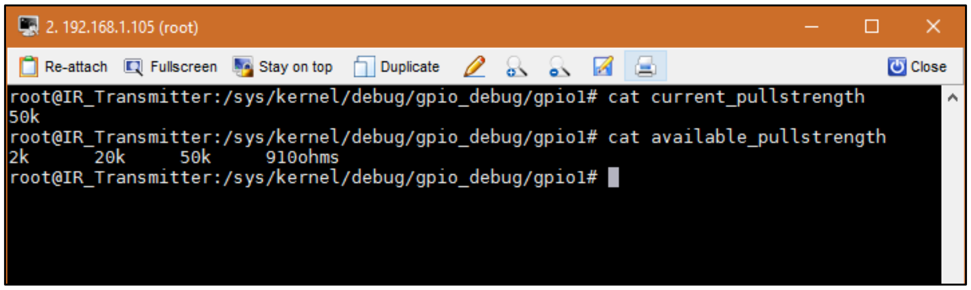
The Arduino Breakout Board comes preconfigured with pull up resistors.

What is a Pull Up Resistor?

Let's say you have an MCU with one pin configured as an input. If there is nothing connected to the pin and your program reads the state of the pin, will it be high (pulled to VCC) or low (pulled to ground)? It is difficult to tell. This phenomena is referred to as floating. To prevent this unknown state, a pull-up or pull-down resistor will ensure that the pin is in either a high or low state, while also using a low amount of current.

- SparkFun Pull Up Resistor Tutorial

According to the datasheet for the Intel Edison Arduino Breakout Board [5], the pull up resistors are configured by default with a value of 50 KΩ, but 910 Ω, 2 KΩ and 20 KΩ are also selectable. So when using the Arduino Breakout board, the user does not need to worry about floating values at the pin.



```
root@IR_Transmitter:/sys/kernel/debug/gpio_debug/gpio1# cat current_pullstrength
50k
root@IR_Transmitter:/sys/kernel/debug/gpio_debug/gpio1# cat available_pullstrength
2k      20k     50k     910ohms
root@IR_Transmitter:/sys/kernel/debug/gpio_debug/gpio1#
```

Figure 34 Pull Up Strength

Given this information, one might hypothesize that if the 50 KΩ pull up works on the Arduino Breakout Board, it may also work on the SparkFun GPIO Block.

6. Repeat the "Hardware Testing" instructions, with the IR Receiver connected to the SparkFun GPIO Board, and an additional 50 KΩ pull up resistor connected between the OUT pin and the VS pin.

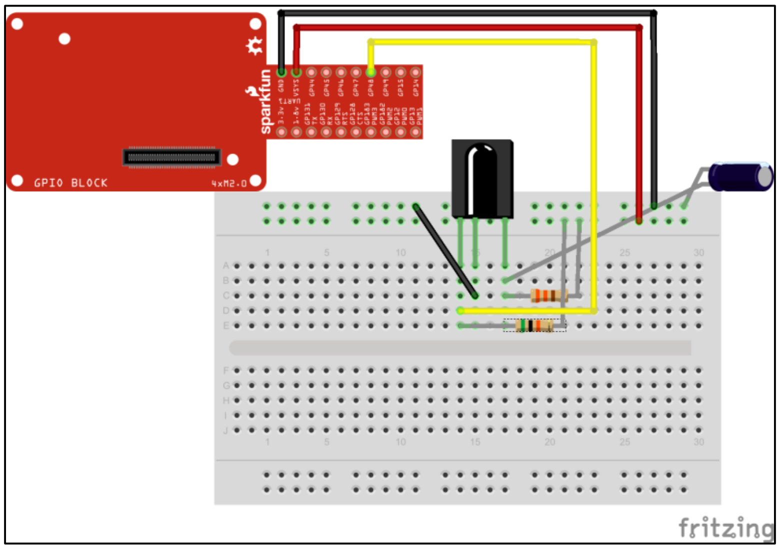
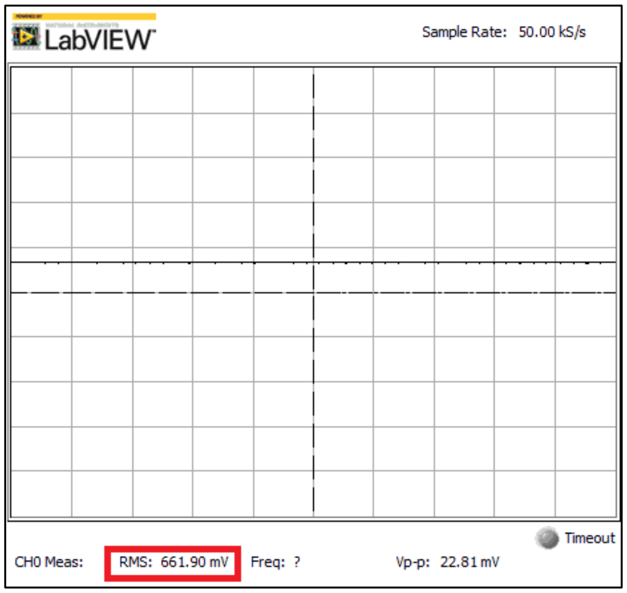
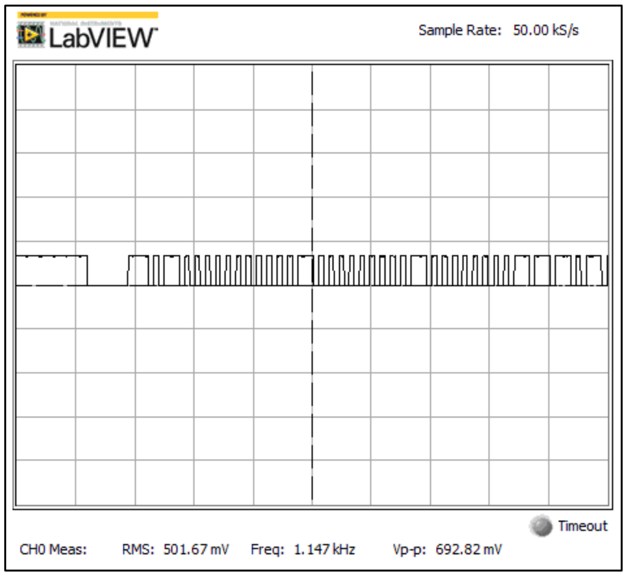


Figure 35 50K Pull Up Resistor Circuit



Sample Rate: 50.00 kS/s

CH0 Meas: RMS: 661.90 mV Freq: ? Vp-p: 22.81 mV



Sample Rate: 50.00 kS/s

CH0 Meas: RMS: 501.67 mV Freq: 1.147 kHz Vp-p: 692.82 mV

Figure 36 50K Pull Up Resistor Circuit Signal Levels

The voltage level is now about 0.6 V, which is higher than 0.4 V, but this voltage level is still not a valid logic level for the Intel Edison.

7. If you have past experience using pull up resistors, you may know that a very typically used value is 10 KΩ. Indeed, the website for the Arduino [6] suggests that 10 KΩ is a good value. Try using the 10 KΩ resistor with the SparkFun GPIO Block.

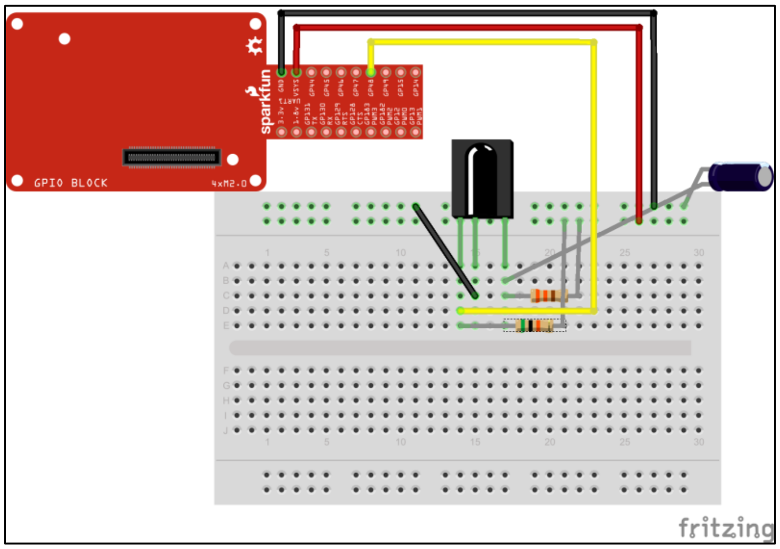
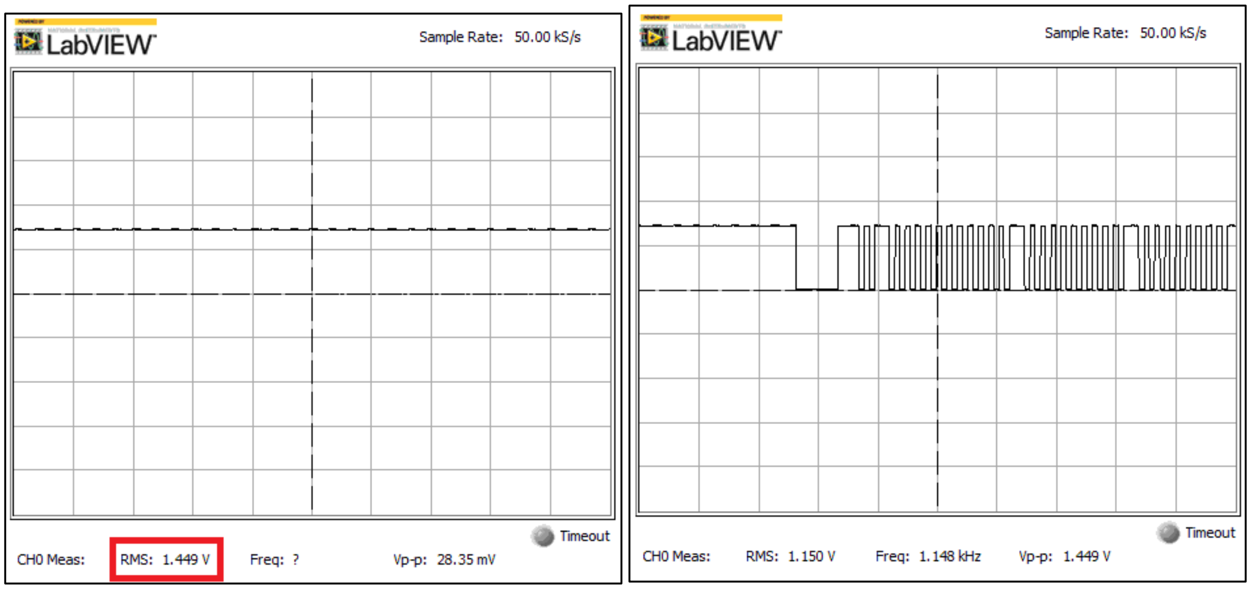


Figure 37 10K Pull Up Resistor Circuit


LabVIEW
Sample Rate: 50.00 kS/s
Timeout
CH0 Meas: RMS: 1.449 V Freq: ? Vp-p: 28.35 mV
LabVIEW
Sample Rate: 50.00 kS/s
Timeout
CH0 Meas: RMS: 1.150 V Freq: 1.148 kHz Vp-p: 1.449 V
Figure 38 10K Pull Up Resistor Circuit Signal Levels

Compared to the 50 KΩ resistor, the 10 KΩ resistor has a higher **HIGH** voltage level of **1.5 V**, which is an improvement, but this value will still not be a valid logic level for the Intel Edison.

At this point, one may realize that there is a trend here – as the resistance of the pull up decreases, the **HIGH** voltage level increases. As an exercise, consider trying out different pull up resistors and discovering the relationship between different resistances and the pulled up voltage. On the next page is a chart with experimental results.

It turns out values lower than about 4 kΩ function properly as a pull up resistors:

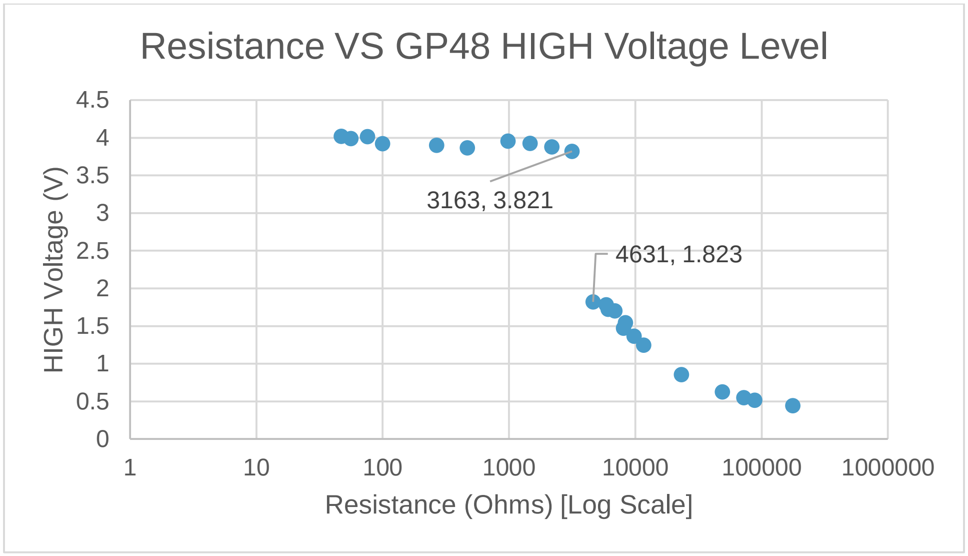


Figure 39 Experimental Pull Up Voltages

After the experimental testing was completed, a pull up resistor with value 1 KΩ was included in the basic kit.

However, these results are a little bit unsatisfying. What is the cause of the behavior? The answer is in basic circuit theory. Do the voltage values corresponding to each pull up resistor make sense? Consider the following circuit:

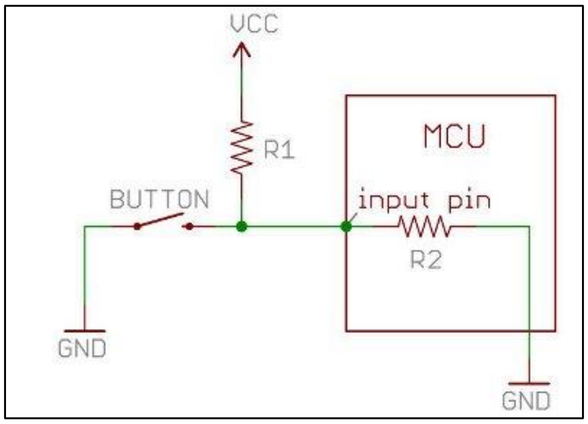
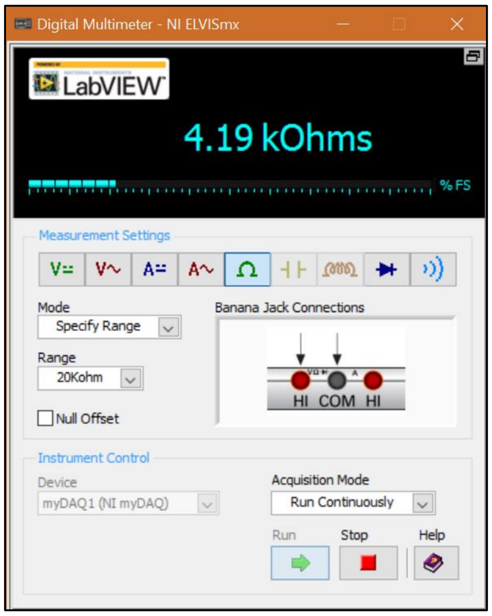


Figure 40 Pull Up Circuit

Instead of a button, the left half of the circuit is connected to the OUT pin of the IR Receiver Diode, but like a button, the IR Receiver will either be connected to GND or to VCC. Thus, at the connection to the input pin, there is a simple voltage divider between R1 and R2. It is possible to measure the resistance between the input pin, GP48, and GND, as in Figure 40:


Digital Multimeter - NI ELVISmx

4.19 kOhms

Measurement Settings

Mode: Specify Range

Range: 20Kohm

Null Offset

Banana Jack Connections

Instrument Control

Device: myDAQ1 (NI myDAQ)

Acquisition Mode: Run Continuously

Figure 41 Pull Up Circuit

Given that VCC has a value of 4 V, using the voltage divider it is possible to calculate the expected voltage at the input pin. Consider the 10 KΩ case:

$$V_{pin} = 4\ V\left(\frac{4.19\ k\Omega}{4.19 + (10\ k\Omega)}\right) = 1.181\ V$$

But this is not quite right, because a voltage of about 1.44 V was measured experimentally for the 10 KΩ resistor. To get a more accurate result, one must also include the resistor that is hiding inside of the IR Receiver Diode.

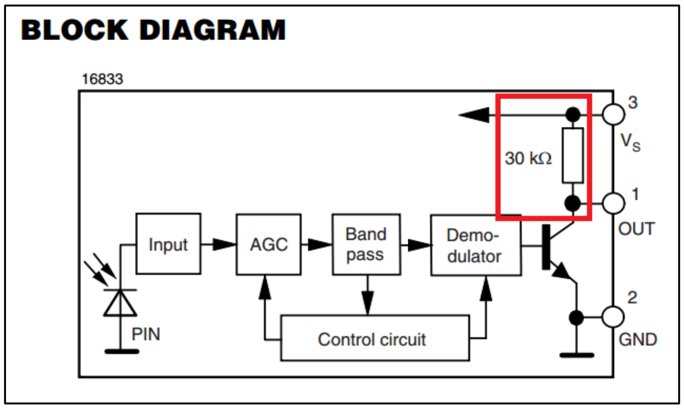


Figure 42 IR Receiver Internal Pull Up Resistor

This resistor is connected in parallel with the 10 K resistor. So the correct expression should be:

$$V_{pin} = 4\ V\left(\frac{4.19\ k\Omega}{4.19 + (10\ k\Omega\ \|\ \ 30k\Omega)}\right) = 1.434\ V$$

And this is much closer to the experimentally measured value. The calculation can also be checked for when there is no extra pull up resistor connected, and only the internal pull up resistor of the IR Receiver Diode:

$$V_{pin} = 4\,V\left(\frac{4.19\;k\Omega}{4.19 + (30k\Omega)}\right) = 0.4902\,V$$

This is slightly higher than the actual voltage observed, but it is close.

References

1. https://learn.sparkfun.com/tutorials/sparkfun-blocks-for-intel-edison---gpio-block/all
2. http://www.emutexlabs.com/project/215-intel-edison-gpio-pin-multiplexing-guide
3. https://www.sparkfun.com/products/10266
4. https://learn.sparkfun.com/tutorials/pull-up-resistors
5. https://communities.intel.com/thread/59604?start=0&tstart=0
6. https://www.arduino.cc/en/Tutorial/DigitalPins
7. http://download.intel.com/support/edison/sb/edisonarduino_hg_331191005.pdf