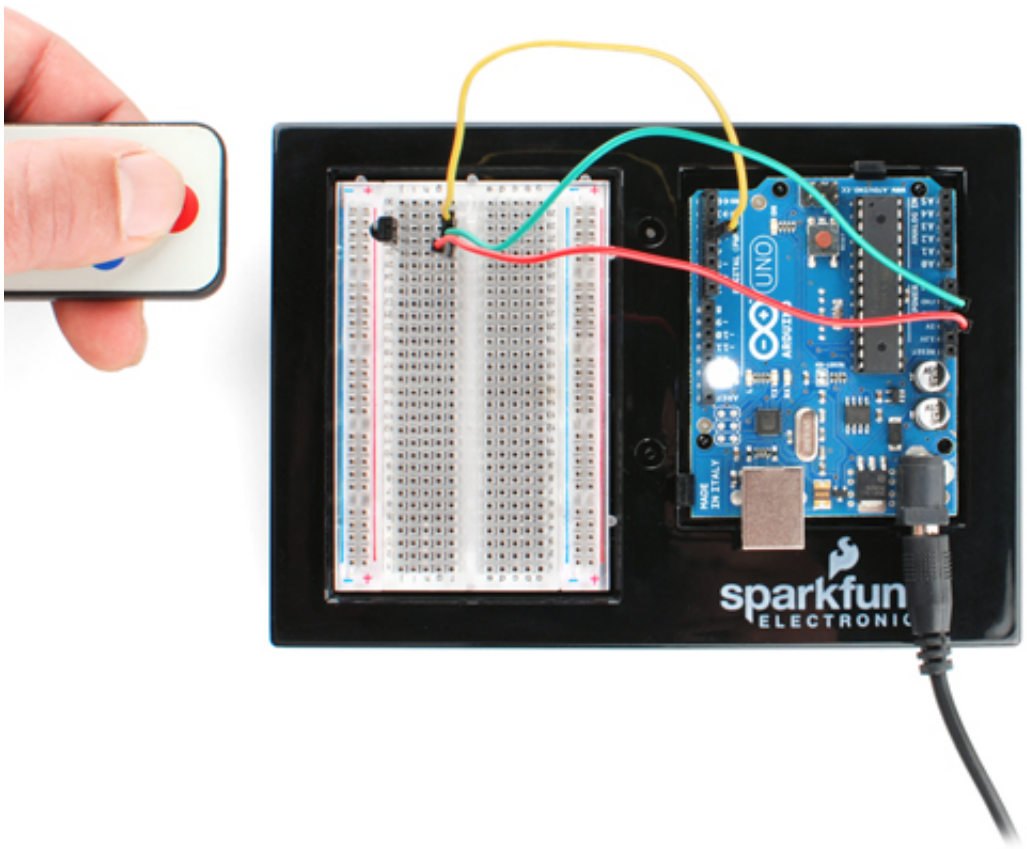
SparkFun Electronics

# IR Control Kit Quickstart Guide

by a1ronzo | June 21, 2011 | *10 comments* Skill Level: Beginner

## Overview:




If you have the IR Control Kit Retail, you probably are wanting to setup some type of communication link between two devices. You might want to control your TV with one of the IR LEDs. Or you might want to control a relay on an Arduino using the Keychain Remote. There are many applications for IR communication. This guide will show you how to transmit data from an IR LED and receive the data using the TSOP38238 IR receiver. In addition, we will show you an example receiving data with the IR Keychain Remote.

## Requirements:

For this guide, we will be using these items:

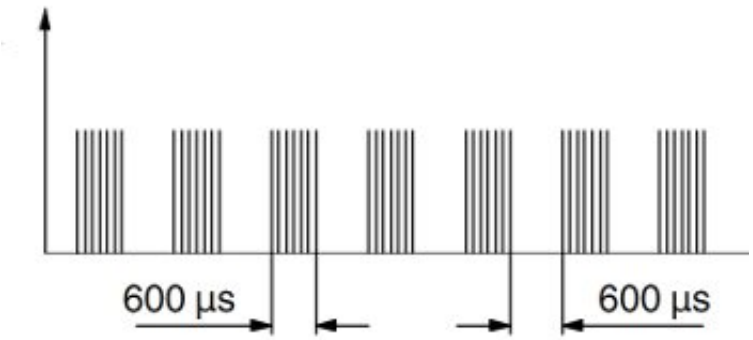
- 1 x Arduino Uno (not included in the kit)
- 1 x IR LED
- 1 x TSOP38238 IR Receiver
- 1 x IR Keychain Remote
- 1 x 330 Ohm resistors

## How it Works:

This is a brief explanation of how IR communication works with common IR systems, like some TV remotes and the TSOP382.

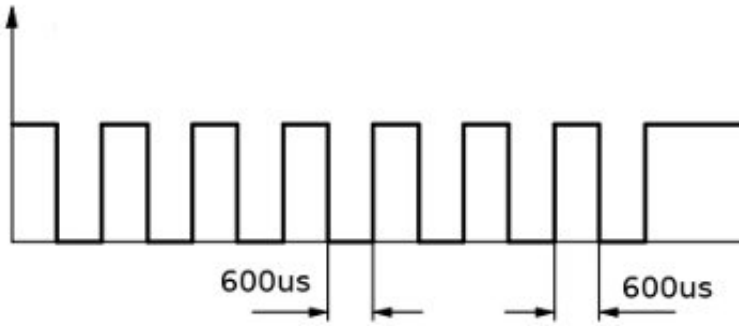
## IR Communication

The TSOP382 only reads modulated 38kHz IR radiation. This means you need to pulse the LED at 38kHz to encode the datastream. This is how most IR remote control systems work. For each bit of information, the LED will either be off for about 600us or on modulating at 38kHz for 600us, as shown below.




If you were to hook an oscilloscope up to your TV remote's IR LED, you would see a signal similar to the one above. The main reason why this is done, is to prevent interference from other sources. Remember IR radiation is all around us and presents itself wherever there is heat. There are not many natural sources that have the regularity of a 38kHz wave, so an LED blinking at that frequency would stand out among the ambient IR radiation.

The modulated signal above is exactly what the RX system sees. However, the point of the RX device (TSOP8382) is to demodulate the signal and output a binary waveform that can be read by a microcontroller. When you read the OUT pin of the TSOP382 with the input wave from above, you will see something like this:




This waveform can be read by an Arduino input pin and encoded as a serial bit stream.

## How to Use it:

Below are two demos. The first one will guide you through setting everything up and providing a simple transmit and receive example. The second demo incorporates the Keychain Remote on the same setup.

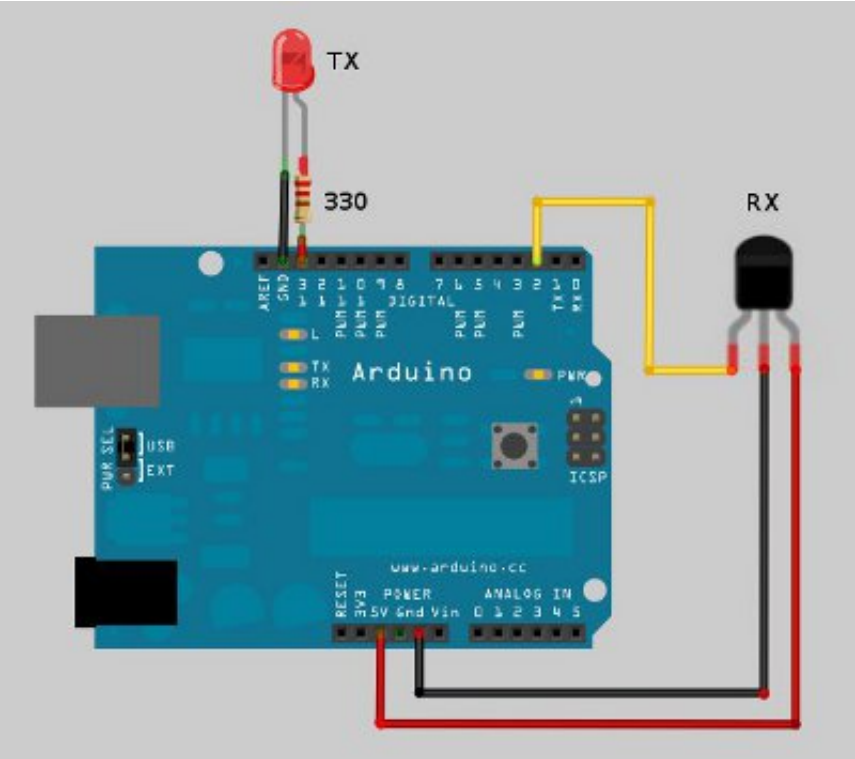
## TX and RX Setup

The TSOP38238 is an IR receiver (RX) that will receive the signals from the IR LED (TX). In this example, I am only going to use one Arduino and the communication will only be a couple of inches. In a realistic situation, you will probably want each unit to have its own microcontroller and power supply.

The TX setup only involves an IR LED and a 330 Ohm resistor being driven by an output pin on the Arduino. The output pin on the Arduino can only source 40mA of current. This means the LED cannot be driven to its full potential of 50mA. This will result in a loss of transmission distance. If you want to drive an IR LED properly, you can look at the Max Power IR LED Kit.

The RX setup just plugs right in to the Arduino. You can find the pinout of the TSOP382 in the datasheet.

Here is the setup for the complete connection to an Arduino:




## TX and RX Arduino Code

All we need is one sketch to be written to test the transmission and reception. Here is the code.

```
  /*
  SparkFun Electronics 2011
  OSHW License http://freedomdefined.org/OSHW

  IR TX and RX demo at 38k Hz
  -Outputs a 38KHz wave on pin 11
  -takes input from TSOP382 on pin 2

  The IR LED carrier wave of 38kHz is turned on and off
  to blink and LED.

*/

//define your square wave frequency
#define IR_CLOCK_RATE    38000L

int ledPin = 13;        // the pin that the LED is attached to

void setup() {
```

The code simply imitates a periodic modulated wave at 38kHz on pin 11 of the Arduino. The IR receiver is attached to an interrupt pin (digital pin 2), so when the demodulated wave has a falling edge, the LED blinks and prints 'hit'.

You could replicate this setup with two individual systems (two sets of code) separated at a distance, then you would have a simple IR trigger system!

## Keychain Remote Example

Try leaving the TSOP328 connected to your Arduino and disconnect the IR TX LED. Load this code onto your Arduino, open a terminal window and hit a button on the keychain remote.

```
  /*
SparkFun Electronics 2010
Playing with IR remote control

IR Receiver TSOP382: Supply voltage of 2.5V to 5.5V
With the curved front facing you, pin 1 is on the left.
Attach
  Pin 1: To pin 2 on Arduino
  Pin 2: GND
  Pin 3: 5V

This is based on pmalmsten's code found on the Arduino forum
http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1176098434/0

This code works with super cheapo remotes. If you want to loo
of the bits, use this code:
http://www.arduino.cc/playground/Code/InfraredReceivers

This code clips a lot of the incoming IR blips, but what is l
```

The sketch will identify which button on the keychain is pressed, then it will send a message out to a serial port indicating which button was pressed.

## Resources:

- Datasheet (TSOP38238)
- Illumitune Project (tutorial) (video)
- Be sure to check out our IR Sensor category for more transmitter and receiver options.

## Conclusion:

You should now have a good idea how to utilize IR receivers in your next project. If you have questions, problems or want to show off your project, don't hesitate to contact us at techsupport@sparkfun.com.

SparkFun Electronics ® | Boulder, Colorado | Customer Service