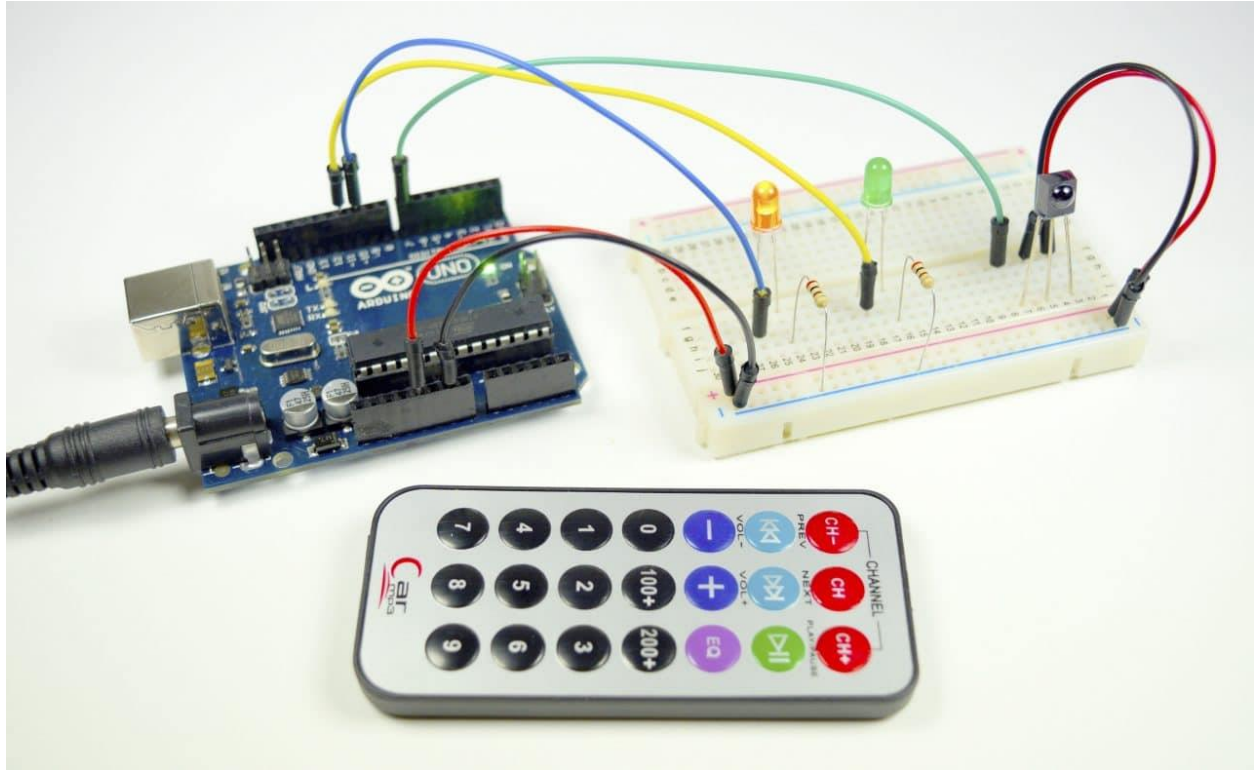


IR RECEIVER



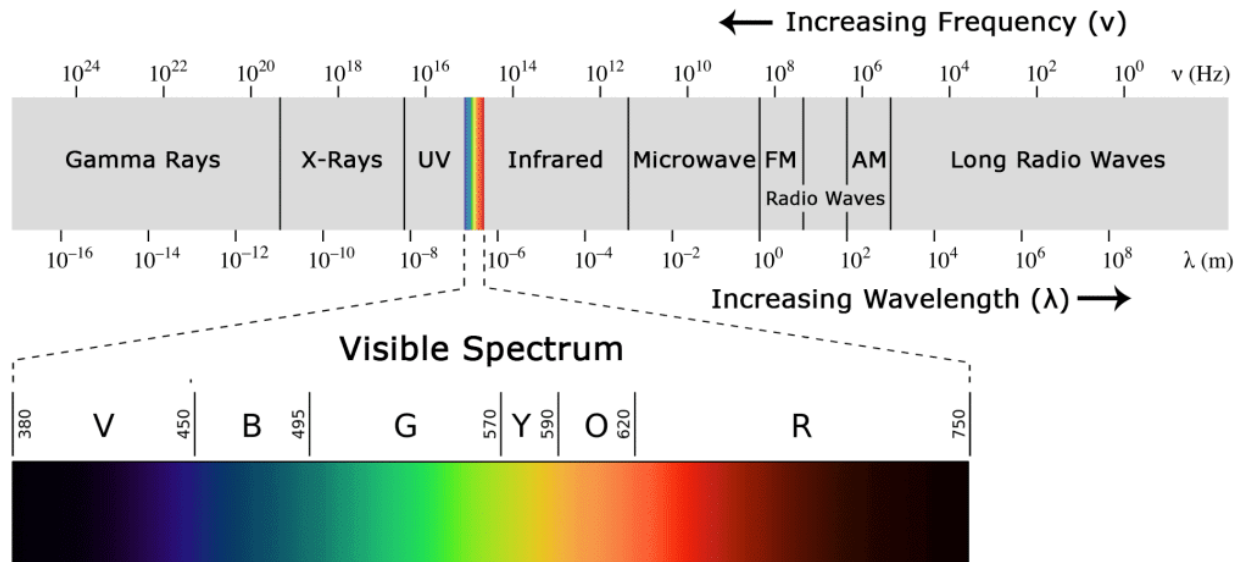
Infrared (IR) communication is a widely used and easy to implement wireless technology that has many useful applications. The most prominent examples in day to day life are TV/video remote controls, motion sensors, and infrared thermometers.

There are plenty of interesting Arduino projects that use IR communication too. With a simple IR transmitter and receiver, you can make remote controlled robots, distance sensors, heart rate monitors, DSLR camera remote controls, TV remote controls, and lots more.

In this tutorial I'll first explain what infrared is and how it works. Then I'll show you how to set up an IR receiver and remote on an Arduino. I'll also show you how to use virtually any IR remote (like the one for your TV) to control things connected to the Arduino.

WHAT IS INFRARED?

Infrared radiation is a form of light similar to the light we see all around us. The only difference between IR light and visible light is the frequency and wavelength. Infrared radiation lies outside the range of visible light, so humans can't see it:



Because IR is a type of light, IR communication requires a direct line of sight from the receiver to the transmitter. It can't transmit through walls or other materials like WiFi or Bluetooth.

HOW IR REMOTES AND RECEIVERS WORK

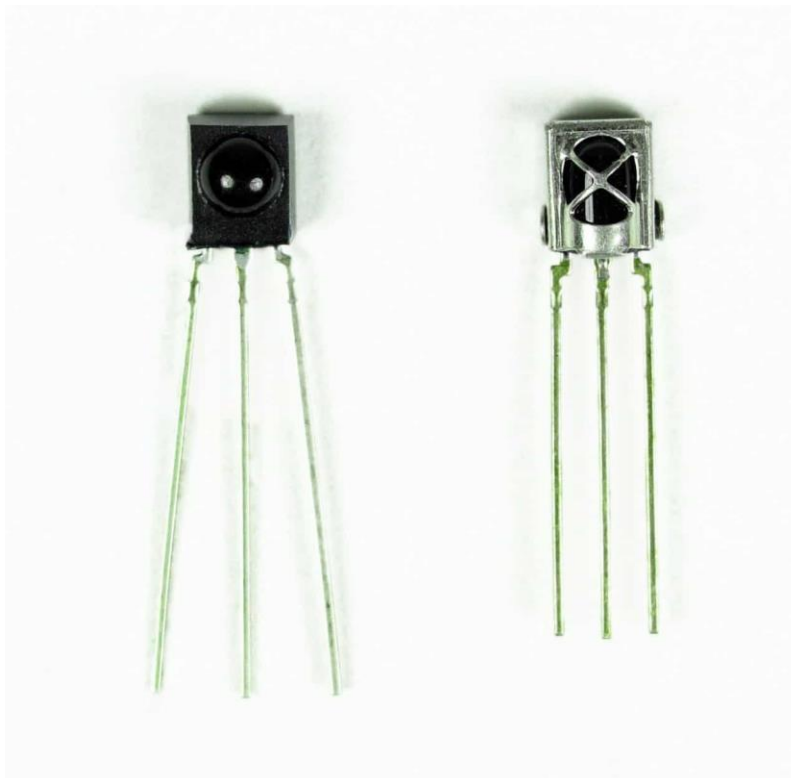
A typical infrared communication system requires an IR transmitter and an IR receiver. The transmitter looks just like a standard LED, except it produces light in the IR spectrum instead of the visible spectrum. If you have a look at the front of a TV remote, you'll see the IR transmitter LED:



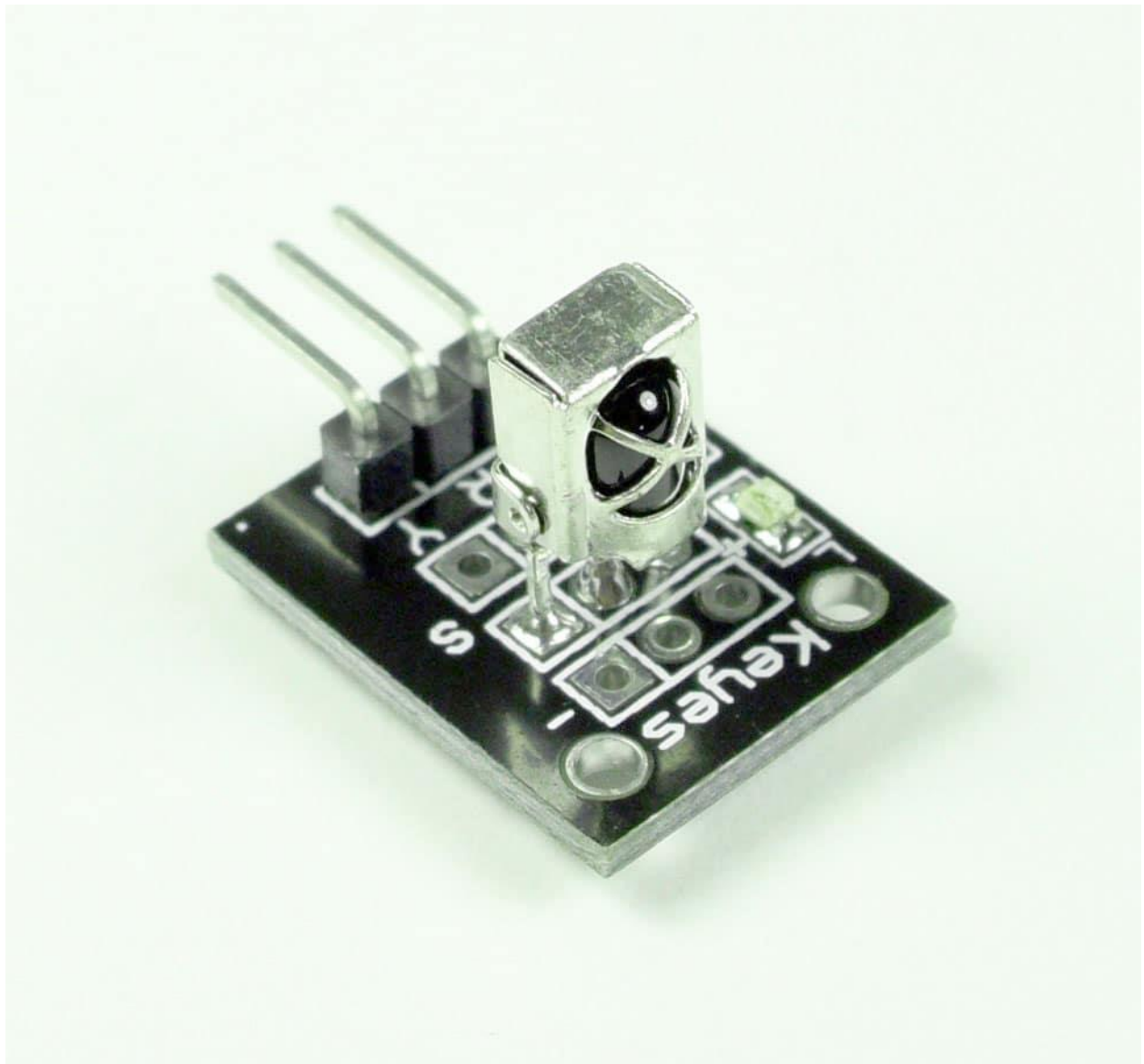
The same type of LED is used in IR transmitter breakout boards for the Arduino. You can see it at the front of this Keyes IR transmitter:



The IR receiver is a photodiode and pre-amplifier that converts the IR light into an electrical signal. IR receiver diodes typically look like this:



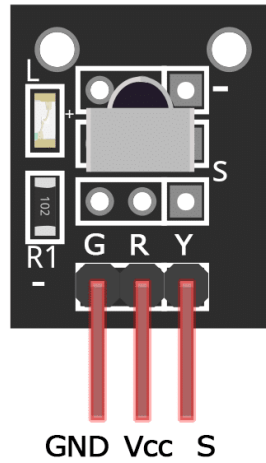
Some may come on a breakout board like this:



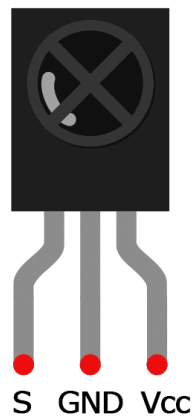
HOW TO CONNECT AN IR RECEIVER TO THE ARDUINO

There are several different types of IR receivers, some are stand-alone, and some are mounted on a breakout board. Check the datasheet for your particular IR receiver since the pins might be arranged differently than the HX1838 IR receiver and remote set I am using here. However, all IR receivers will have three pins: signal, ground, and Vcc.

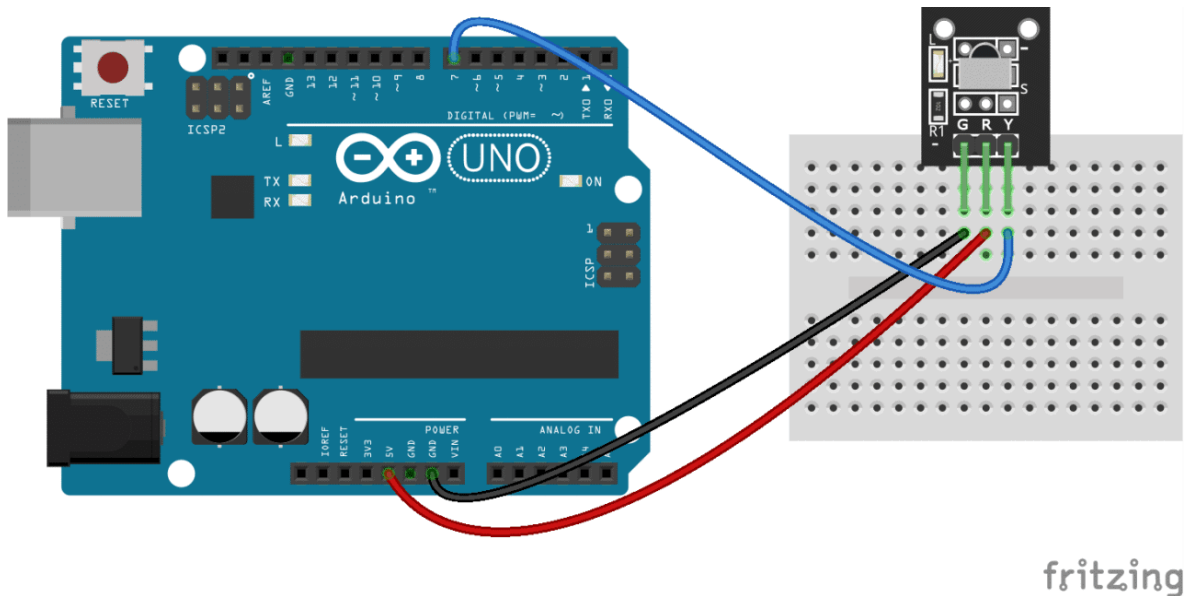
Lets get started with the hardware connections. The pin layout on most breakout boards looks like this:



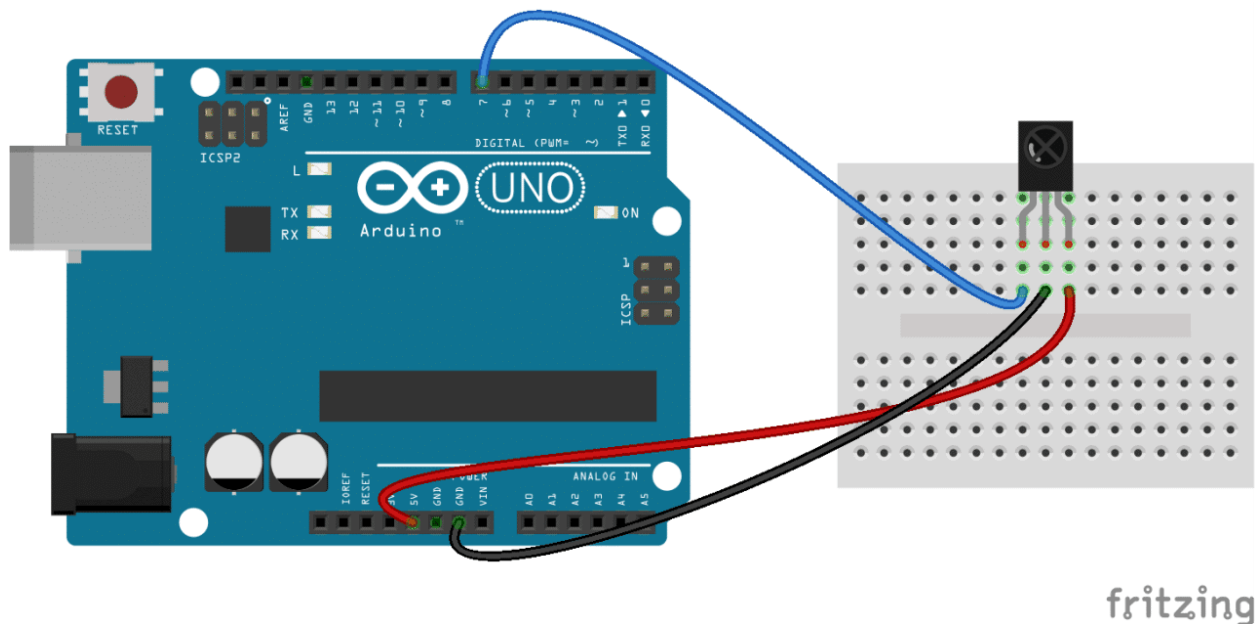
The pinout of most stand-alone diodes is like this:



To connect a breakout board mounted IR receiver, hook it up to the Arduino like this:



To connect a stand-alone receiver diode, wire it like this:



PROGRAMMING THE IR RECEIVER

Once you have the receiver connected, we can install the Arduino library and start programming. In the examples below, I'll show you how to find the codes sent by your remote, how to find the IR protocol used by your remote, how to print key presses to the serial monitor or an LCD, and finally, how to control the Arduino's output pins with a remote.

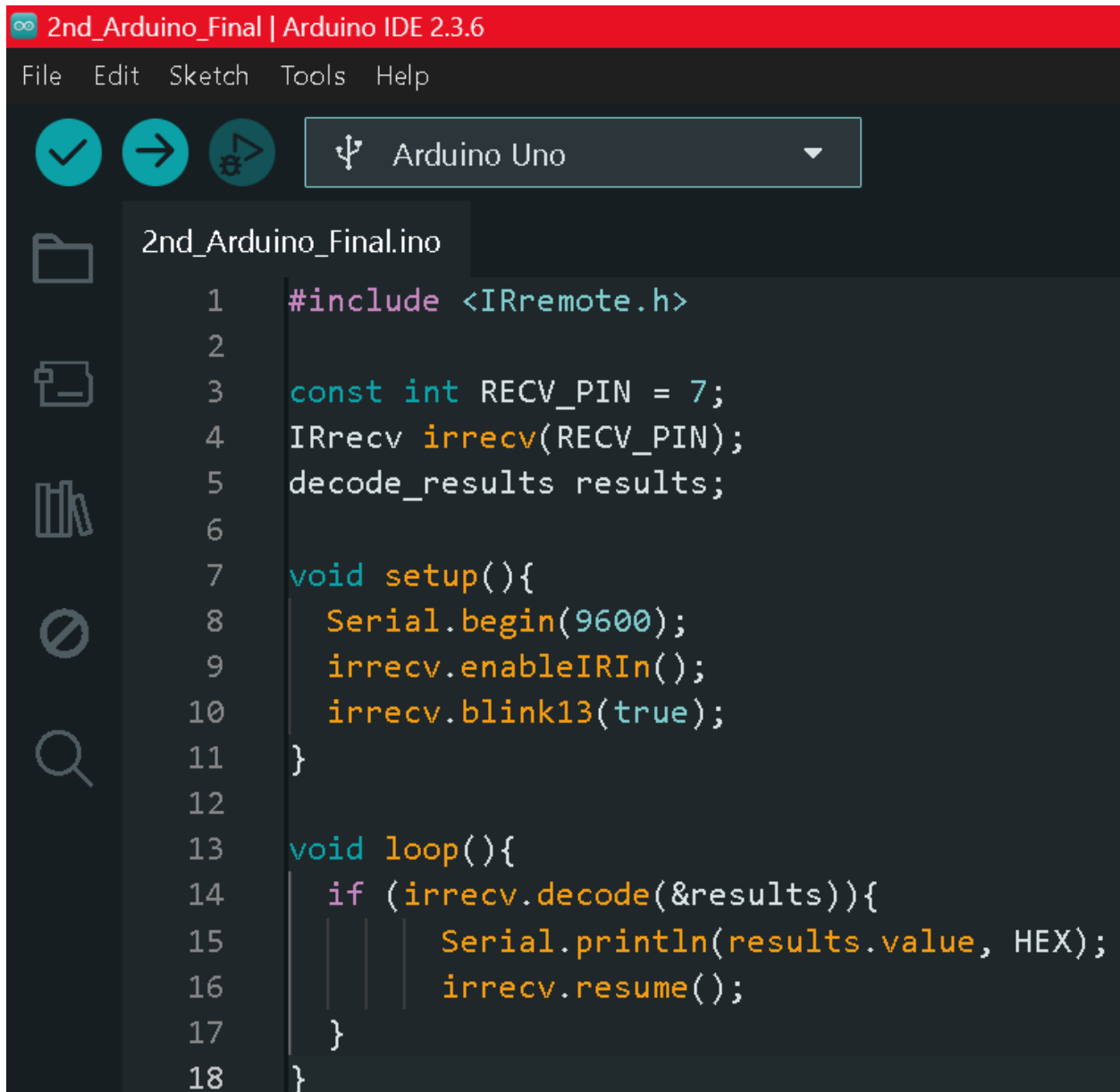
INSTALL THE IRREMOTE LIBRARY

We'll be using the IRremote library for all of the code examples below. You can download a ZIP file of the library from [here](#).

To install the library from the ZIP file, open up the Arduino IDE, then go to Sketch > Include Library > Add .ZIP Library, then select the IRremote ZIP file that you downloaded from the link above.

FIND THE CODES FOR YOUR REMOTE

To find the key codes for your remote control, upload this code to your Arduino and open the serial monitor:

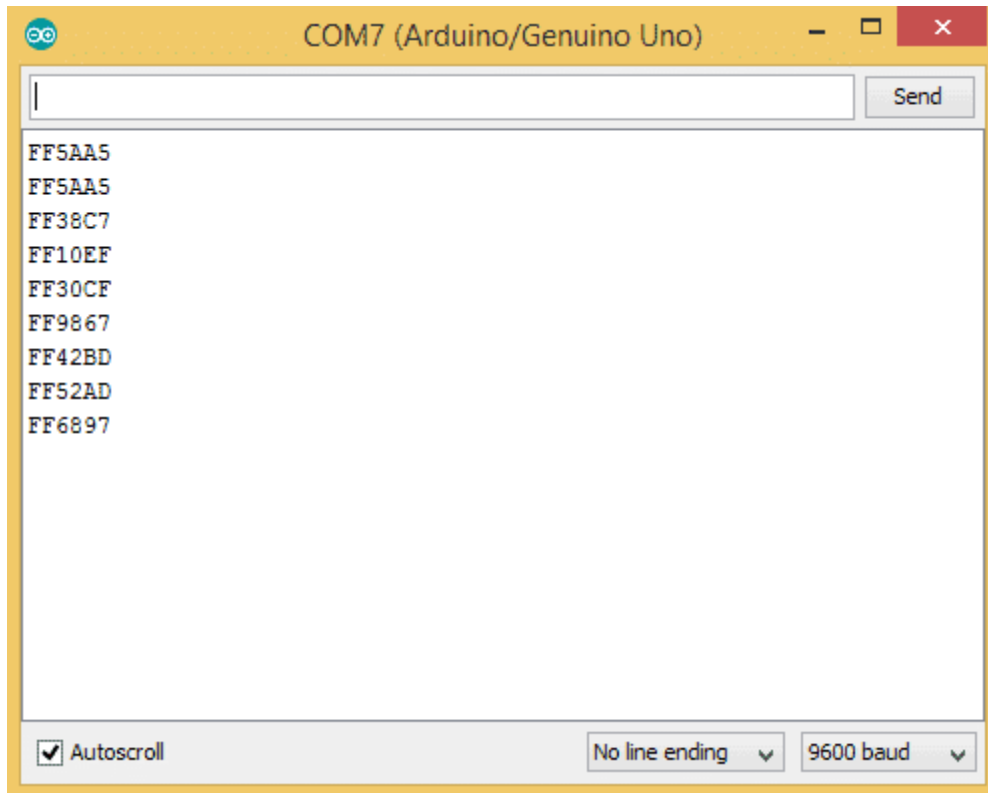


```
2nd_Arduino_Final | Arduino IDE 2.3.6
File Edit Sketch Tools Help

[Checkmark] [Next] [Upload] [USB] Arduino Uno ▼

2nd_Arduino_Final.ino
1  #include <IRremote.h>
2
3  const int RECV_PIN = 7;
4  IRrecv irrecv(RECV_PIN);
5  decode_results results;
6
7  void setup(){
8      Serial.begin(9600);
9      irrecv.enableIRIn();
10     irrecv.blink13(true);
11 }
12
13 void loop(){
14     if (irrecv.decode(&results)){
15         Serial.println(results.value, HEX);
16         irrecv.resume();
17     }
18 }
```

Now press each key on your remote and record the hexadecimal code printed for each key press.



Using the program above, I derived a table of keys and their corresponding codes from the remote that came with my HX1838 IR receiver and remote set. Note that you will receive a 0xFFFFFFFF code when you press a key continuously.

Key	Code
CH-	0xFFA25D
CH	0xFF629D
CH+	0xFFE21D
<<	0xFF22DD
>>	0xFF02FD

>	0xFFC23D
–	0xFFE01F
+	0xFFA857
EQ	0xFF906F
100+	0xFF9867
200+	0xFFB04F
0	0xFF6897
1	0xFF30CF
2	0xFF18E7
3	0xFF7A85
4	0xFF10EF
5	0xFF38C7
6	0xFF5AA5
7	0xFF42BD
8	0xFF4AB5
9	0xFF52AD

REFERENCE:

<https://www.circuitbasics.com/arduino-ir-remote-receiver-tutorial/>