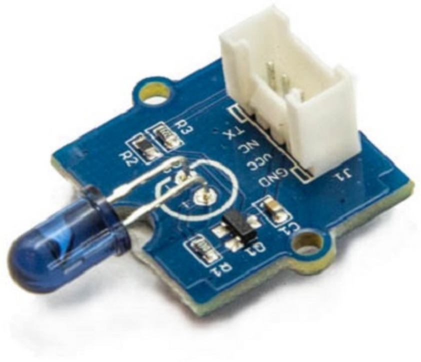
Grove - Infrared Emitter



The Infrared Emitter is used to transmit infrared signals through an infrared LED, while there is an **Infrared receiver** to get the signals on the other side. An infrared LED is like any other LED, with its color centered around 940nm. We can not only use the emitter to transmit data or commands, but also to emulate remotes to control your home appliance using an Arduino. The Infrared Emitter can transmit signals reliable up to 10 meters. Beyond 10 meters, the receiver may not get the signals. We often use the two Groves-the Infrared Receiver [https://wiki.seeedstudio.com/Grove-Infrared_Receiver] and the Grove - Infrared Emitter to work together.

Get One Now 🛒

[https://www.seeedstudio.com/Grove-Infrared-Emitter-p-993.html]

Version

Product Version	Changes	Released Date
Grove - Infrared Emitter v1.0	Initial	Nov. 01 2015
Grove - Infrared Emitter v1.1	Change the Infrared transmitting tube location	Jul. 24 2016
Grove - Infrared Emitter v1.2	Change the valnue of C1 to make the power more stable	Dec. 14 2016

Application

- Infrared remote control units with high power requirements
- Free air transmission systems
- Infrared source for optical counters and card readers

Specification

Parameter	Value/Range
Operating voltage	3.⅗V
Peak wavelength	940nm
Angle of half intensity	ϕ = ± 17°
Radiant Intensity	72 mW/sr
Distance	10 meter(MAX)
Operation Temperature	-40℃ to +80℃
Size	20mmX20mm

Tip

More details about Grove modules please refer to Grove System [https://wiki.seeedstudio.com/Grove_System/]

Platforms Supported

Arduino	Raspberry Pi	
	TO DO	

Caution
The platforms mentioned above as supported is/are an indication of the module's software or theoritical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

Getting Started

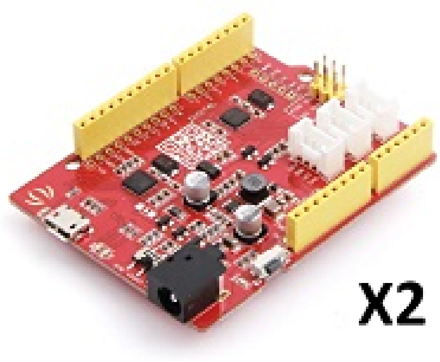
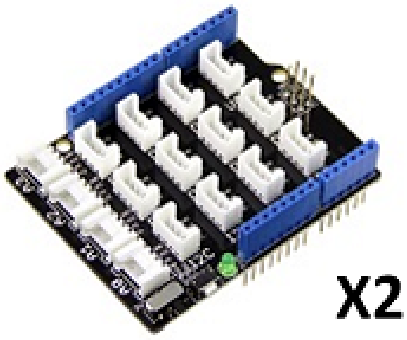
The Grove - Infrared Emitter can send data while Grove - Infrared Receiver will receive them.

Play With Arduino

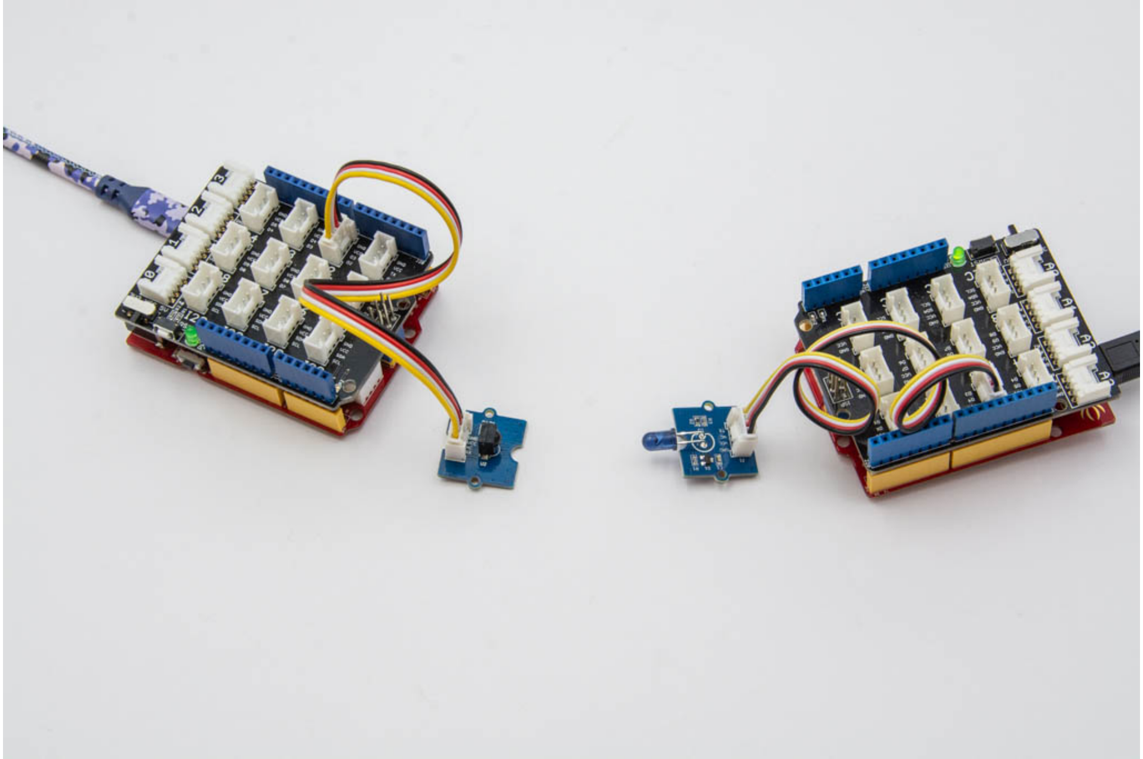
Note
If this is the first time you work with Arduino, we firmly recommend you to see Getting Started with Arduino [https://wiki.seeedstudio.com/Getting_Started_with_Arduino/] before the start.

Hardware

- **Step 1.** Prepare the below stuffs:

Seeeduino V4.2	Base Shield
	
Get One Now [https://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html]	Get One Now [https://www.seeedstudio.com/Base Shield-V2-p-1378.html]

- **Step 2.** Connect Grove - Infrared Emitter to port **D3** of one Grove-Base Shield.
- **Step 3.** Connect Grove - Infrared Receiver to port **D2** of the other Grove-Base Shield.
- **Step 4.** Plug Grove - Base Shield into Seeeduino.
- **Step 5.** Connect Seeeduino to PC via a USB cable.



Note

If we don't have Grove Base Shield, We also can directly connect this module to Seeeduino as below.

Seeeduino	Grove - Infrared Emitter
5V	Red
GND	Black
Not Conencted	White
D3	Yellow

Seeeduino	Grove - Infrared Receiver
5V	Red
GND	Black
Not Conencted	White
D2	Yellow

Software

- **Step 1.** Download the Seeed_Arduino_IR [https://github.com/Seeed-Studio/Seeed_Arduino_IR] from Github.
- **Step 2.** Refer How to install library [https://wiki.seeedstudio.com/How_to_install_Arduino_Library] to install library for Arduino.

Copy the following **Send Example Code** to the Arduino IDE:

Send Example Code:

```
/* send.ino Example sketch for IRLib2
 *  Illustrates how to send a code.
 */
#include <IRLibSendBase.h>    // First include the send
//Now include only the protocols you wish to actually us
//The lowest numbered protocol should be first but remai
//can be any order.
#include <IRLib_P01_NEC.h>
#include <IRLib_P02_Sony.h>
#include <IRLibCombo.h>     // After all protocols, incl
// All of the above automatically creates a universal se
// class called "IRsend" containing only the protocols y
// Now declare an instance of that sender.

IRsend mySender;

#define IR_SEND_PWM_PIN D3

void setup() {
  Serial.begin(9600);
  delay(2000); while (!Serial); //delay for Leonardo
  Serial.println(F("Every time you press a key is a seri
}

void loop() {
  if (Serial.read() != -1) {
    //send a code every time a character is received fro
    // serial port. You could modify this sketch to send
    // push a button connected to an digital input pin.
    //Substitute values and protocols in the following s
    // for device you have available.
    mySender.send(SONY,0xa8bca, 20);//Sony DVD power A8B
    //mySender.send(NEC,0x61a0f00f,0);//NEC TV power but
    Serial.println(F("Sent signal."));
  }
}
```

Copy the following **Receive Example Code** to the Arduino IDE:

Receive Example Code:

```
/* rawR&cv.ino Example sketch for IRLib2
 *  Illustrate how to capture raw timing values for an u
 *  You will capture a signal using this sketch. It will
 *  serial monitor that you can cut and paste into the "
 *  sketch.
 */
// Recommend only use IRLibRecvPCI or IRLibRecvLoop for
#include <IRLibRecvPCI.h>

```

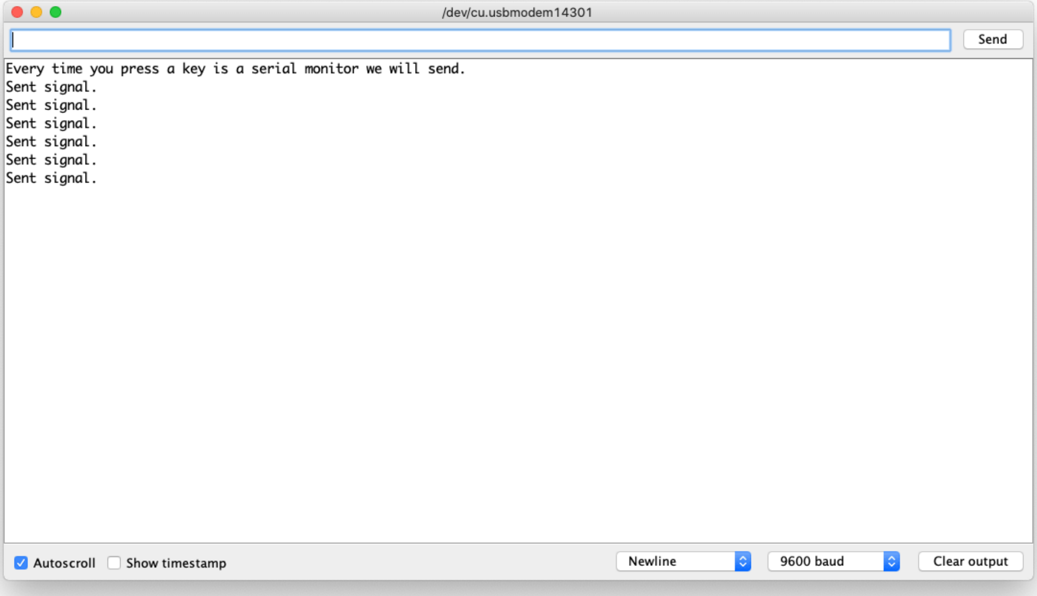
```
IRrecvPCI myReceiver(2);//pin number for the receiver

void setup() {
  Serial.begin(9600);
  delay(2000); while (!Serial); //delay for Leonardo
  myReceiver.enableIRIn(); // Start the receiver
  Serial.println(F("Ready to receive IR signals"));
}

void loop() {
  //Continue looping until you get a complete signal rec
  if (myReceiver.getResults()) {
    Serial.println(F("Do a cut-and-paste of the followin
    Serial.println(F("designated location in rawSend.ino
    Serial.print(F("\n#define RAW_DATA_LEN "));
    Serial.println(recvGlobal.recvLength,DEC);
    Serial.print(F("uint16_t rawData[RAW_DATA_LEN]={\n\t
    for(bufIndex_t i=1;i<recvGlobal.recvLength;i++) {
      Serial.print(recvGlobal.recvBuffer[i],DEC);
      Serial.print(F(", "));
      if( (i % 8)==0) Serial.print(F("\n\t"));
    }
    Serial.println(F("1000};"));//Add arbitrary trailing
    myReceiver.enableIRIn();      //Restart receiver
  }
}
```

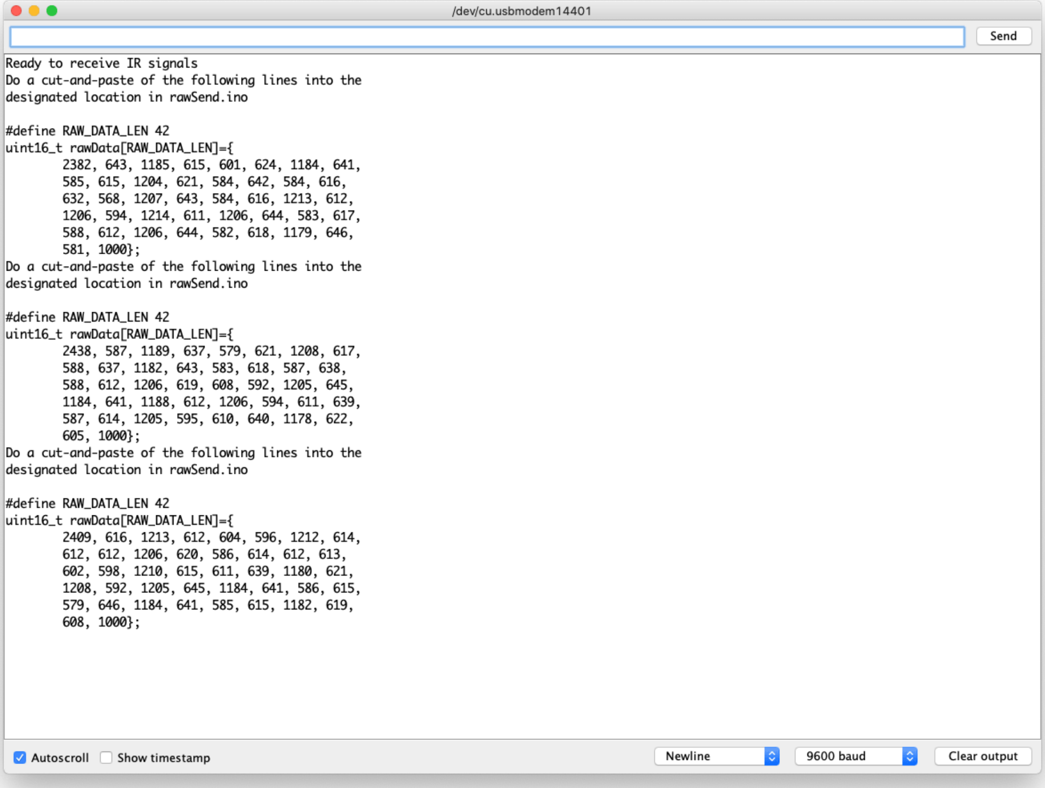
- **Step 7.** Open the **Serial Monitor** of Arduino IDE by click **Tool-> Serial Monitor**. Or tap the `Ctrl`+`Shift`+`M` key at the same time.

For the **Send Example**, the Serial should be like this:



```
Every time you press a key is a serial monitor we will send.
Sent signal.
Sent signal.
Sent signal.
Sent signal.
Sent signal.
Sent signal.
```

For the **Receive Example**, the Serial Monitor should be like this:



```
Ready to receive IR signals
Do a cut-and-paste of the following lines into the
designated location in rawSend.ino

#define RAW_DATA_LEN 42
uint16_t rawData[RAW_DATA_LEN]={
        2382, 643, 1185, 615, 601, 624, 1184, 641,
        585, 615, 1204, 621, 584, 642, 584, 616,
        632, 568, 1207, 643, 584, 616, 1213, 612,
        1206, 594, 1214, 611, 1206, 644, 583, 617,
        588, 612, 1206, 644, 582, 618, 1179, 646,
        581, 1000};
Do a cut-and-paste of the following lines into the
designated location in rawSend.ino

#define RAW_DATA_LEN 42
uint16_t rawData[RAW_DATA_LEN]={
        2438, 587, 1189, 637, 579, 621, 1208, 617,
        588, 637, 1182, 643, 583, 618, 587, 638,
        588, 612, 1206, 619, 608, 592, 1205, 645,
        1184, 641, 1188, 612, 1206, 594, 611, 639,
        587, 614, 1205, 595, 610, 640, 1178, 622,
        605, 1000};
Do a cut-and-paste of the following lines into the
designated location in rawSend.ino

#define RAW_DATA_LEN 42
uint16_t rawData[RAW_DATA_LEN]={
        2409, 616, 1213, 612, 604, 596, 1212, 614,
        612, 612, 1206, 620, 586, 614, 612, 613,
        602, 598, 1210, 615, 611, 639, 1180, 621,
        1208, 592, 1205, 645, 1184, 641, 586, 615,
        579, 646, 1184, 641, 585, 615, 1182, 619,
        608, 1000};
```

> For more advanced usage of the library, please check Seeed_Arduino_IR [https://github.com/Seeed-Studio/Seeed_Arduino_IR].

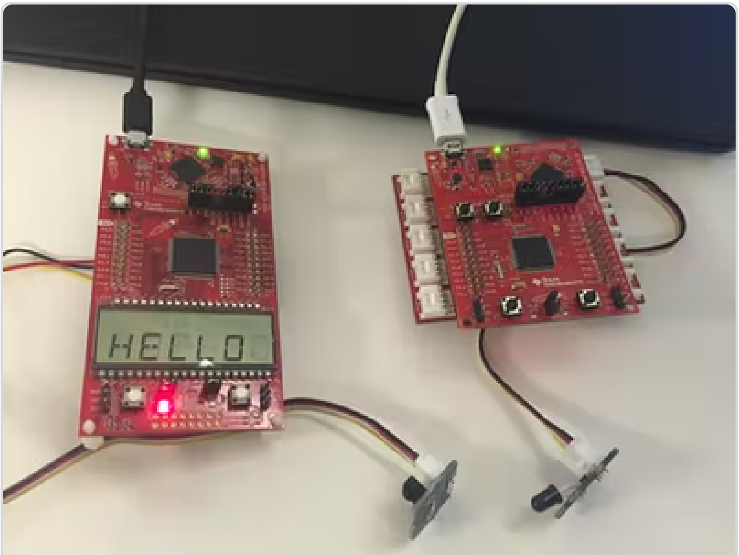
Schematic Online Viewer

Resources

- **[Zip]** Grove-Infrared Emitter eagle files [https://files.seeedstudio.com/wiki/Grove-Infrared_Emitter/res/Grove-Infrared_Emitter_eagle_files.zip]
- **[Lib]** IR Send and Receiver Library [https://github.com/Seeed-Studio/Seeed_Arduino_IRSendRev]
- **[Pdf]** TSAL6200 Datasheet [http://www.vishay.com/docs/81010/tsal6200.pdf]

Projects

IR LaunchPad to LaunchPad Communication: Send text from one LaunchPad to another using the Grove IR emitter and receiver!



(https://www.hackster.io/ctroberts/ir-launchpad-to-launchpad-communication-0dd109)

IR LaunchPad to LaunchPad

Tech Support

Please submit any technical issue into our forum [https://forum.seeedstudio.com/].

[https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts]