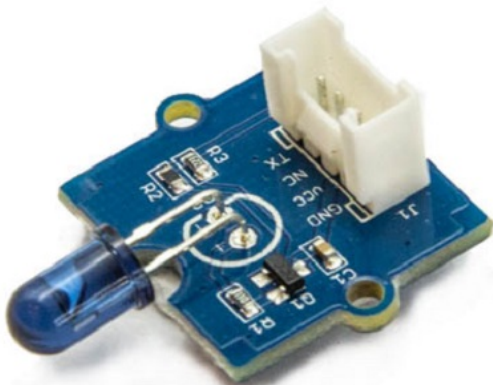# Grove - Infrared Emitter SKU: 101020026

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 and the Grove - Infrared Emitter to work together.

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	3.3/5V
Peak wavelength	940nm
Angle of half intensity	$\phi = \pm\,17°$
Radiant Intensity	72 mW/sr
Distance	10 meter(MAX)
Operation Temperature	-40℃ to +80℃
Size	L:20mm W:20mm H:10mm
Weight	1.9g
Package size	L: 140mm W: 90mm H: 10mm
Gross Weight	9g
Certification	ROHS

!!!Tip More details about Grove modules please refer to Grove System

Platforms Supported

Arduino	Raspberry Pi	BeagleBone	Wio	LinkIt ONE

!!!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.
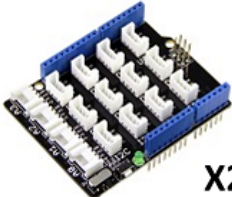
Getting Started

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 before the start.

Hardware

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

Seeeduino V4.2	Base Shield	Grove - Infrared Emitter	Grove - Infrared Receiver
X2	X2		

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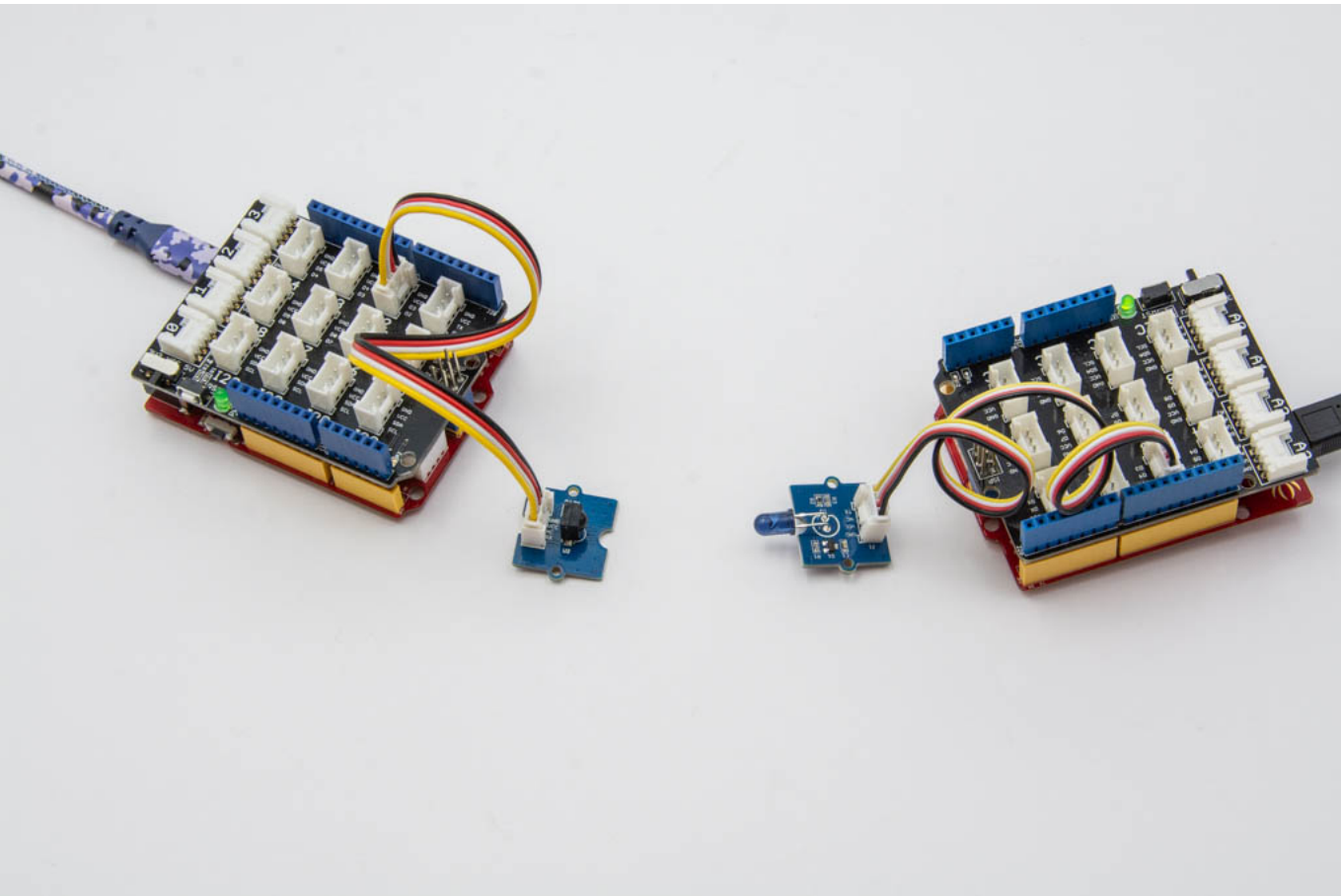
!!!Note If we don't have Grove Base Shield, We also can directly connect this module to Seeeduino as below.

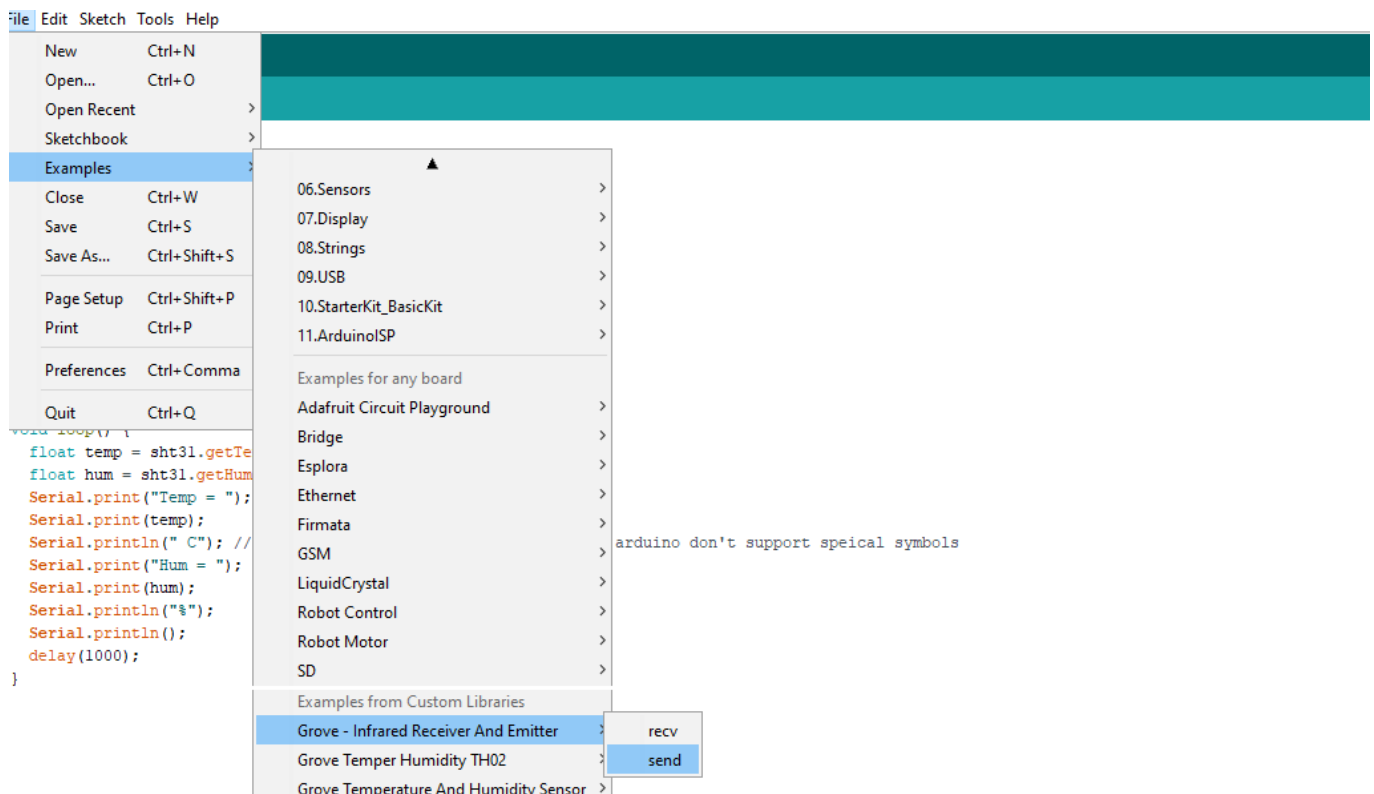| Seeeduino	Grove - Infrared Emitter
5V	Red

Seeeduino	Grove - Infrared Emitter
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 IRSendRev-master library from Github.

- **Step 2.** Refer How to install library to install library for Arduino.

- **Step 3.** Restart the Arduino IDE. Open recv example via the path: **File->Examples->Grove - Infrared Receiver And Emitter->recv**.

Or you can open a new sketch and copy the belowing code into your Arduino IDE.

```
#include <IRSendRev.h>
```

```cpp
#define BIT_LEN          0
#define BIT_START_H      1
#define BIT_START_L      2
#define BIT_DATA_H       3
#define BIT_DATA_L       4
#define BIT_DATA_LEN     5
#define BIT_DATA         6

const int pinRecv = 2;              // ir receiver connect to D2

void setup()
{
    Serial.begin(115200);
    IR.Init(pinRecv);
    Serial.println("init over");
}

unsigned char dta[20];

void loop()
{
    if(IR.IsDta())                  // get IR data
    {
        IR.Recv(dta);               // receive data to dta

        Serial.println("+----------------------------------------------------
+");
                Serial.print("LEN = ");
        Serial.println(dta[BIT_LEN]);
        Serial.print("START_H: ");
        Serial.print(dta[BIT_START_H]);
        Serial.print("\tSTART_L: ");
        Serial.println(dta[BIT_START_L]);

        Serial.print("DATA_H: ");
        Serial.print(dta[BIT_DATA_H]);
        Serial.print("\tDATA_L: ");
        Serial.println(dta[BIT_DATA_L]);

        Serial.print("\r\nDATA_LEN = ");
        Serial.println(dta[BIT_DATA_LEN]);

                Serial.print("DATA: ");
        for(int i=0; i<dta[BIT_DATA_LEN]; i++)
        {
            Serial.print("0x");
            Serial.print(dta[i+BIT_DATA], HEX);
            Serial.print("\t");
        }
        Serial.println();

                Serial.print("DATA: ");
        for(int i=0; i<dta[BIT_DATA_LEN]; i++)
        {
```

```
            Serial.print(dta[i+BIT_DATA], DEC);
            Serial.print("\t");
        }
        Serial.println();
        Serial.println("+-------------------------------------------------
+\r\n\r\n");
    }
}
```

- **Step 4.** Upload the recv demo to the seeeduino with Grove - Infrared Receiver. If you do not know how to upload the code, please check how to upload code.

- **Step 5.** Open send example via the path: **File->Examples->Grove - Infrared Receiver And Emitter->send**.

Or you can open a new sketch and copy the belowing code into your Arduino IDE.

```
#include <IRSendRev.h>

#define BIT_LEN        0
#define BIT_START_H    1
#define BIT_START_L    2
#define BIT_DATA_H     3
#define BIT_DATA_L     4
#define BIT_DATA_LEN   5
#define BIT_DATA       6

const int ir_freq = 38;                // 38k

unsigned char dtaSend[20];

void dtaInit()
{
    dtaSend[BIT_LEN]       = 11;                    // all data that needs to
be sent
    dtaSend[BIT_START_H]   = 179;                   // the logic high duration
of "Start"
    dtaSend[BIT_START_L]   = 90;                    // the logic low duration
of "Start"
    dtaSend[BIT_DATA_H]    = 11;                    // the logic "long"
duration in the communication
    dtaSend[BIT_DATA_L]    = 33;                    // the logic "short"
duration in the communication

    dtaSend[BIT_DATA_LEN]  = 6;                     // Number of data which
will sent. If the number is other, you should increase or reduce
dtaSend[BIT_DATA+x].

    dtaSend[BIT_DATA+0]    = 128;                   // data that will sent
    dtaSend[BIT_DATA+1]    = 127;
```

```
    dtaSend[BIT_DATA+2]      = 192;
    dtaSend[BIT_DATA+3]      = 63;
        dtaSend[BIT_DATA+4]     = 192;
    dtaSend[BIT_DATA+5]      = 63;
}

void setup()
{
    dtaInit();
}

void loop()
{
    IR.Send(dtaSend, 38);

    delay(2000);
}
```

- **Step 6.** Upload the send demo to the seeeduino with Grove - Infrared Emitter.

- **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.

If every thing goes well, The result should be like:

Resources

- **[Zip]** Grove-Infrared Emitter eagle files

- **[Lib]** IR Send and Receiver Library
- **[Pdf]** TSAL6200 Datasheet

Tech Support

Please submit any technical issue into our forum or drop mail to techsupport@seeed.cc.