# Infrared transmission experiment

## Introduction the devices

This time, we will introduce the infrared transmitter and receiver modules, which actually play an important role in our daily life. Now such a device is widely used in many home appliances, such as air conditioning, television, DVD, etc. It is based on wireless remote sensing, but also a remote control, it is necessary to study its principle and how to use.

Infrared transmitting tube and infrared receiving tube are devices that convert electric energy into near-infrared light directly. Its structure and principle are similar to ordinary light-emitting diodes, but the semiconductor material is different.

The infrared receiver is a receiving, amplifying and demodulating device. The internal integrated circuit has been demodulated and the output is digital signal.

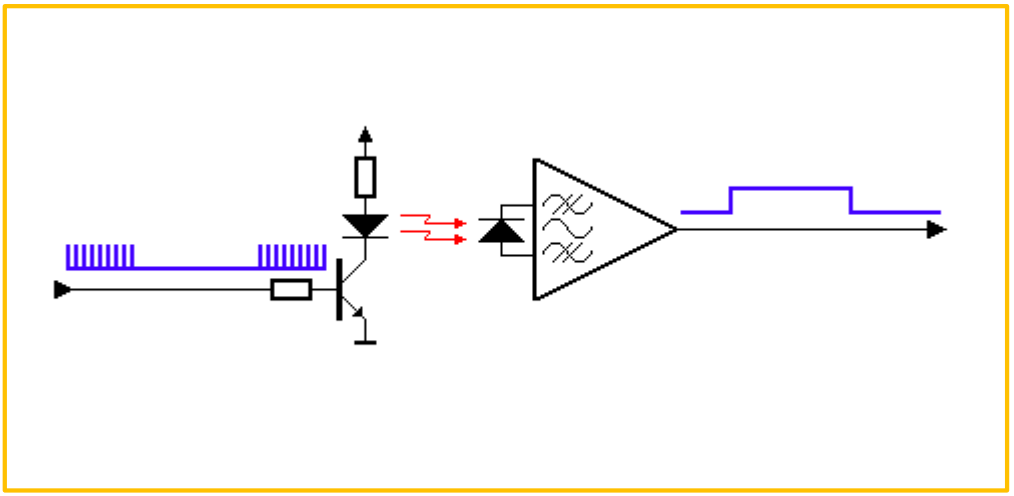
Infrared receiver

Infrared transmitter

## Working Principle

To understand the structure of infrared receiver: infrared receiver is composed of IC and PD. IC is the processing element of the receiver, mainly composed of silicon crystal and circuit. It is a highly integrated device. PD is a photodiode whose main function is to receive optical signals. The infrared emitting diode sends out the modulation signal, and the infrared receiving head recovers the signal after receiving, decoding, filtering and a series of operations.

In this experiment, we wanted to use two Arduino motherboards, a transmitter (Master) and a receiver (Slave). After the circuit is connected, we can do the test.



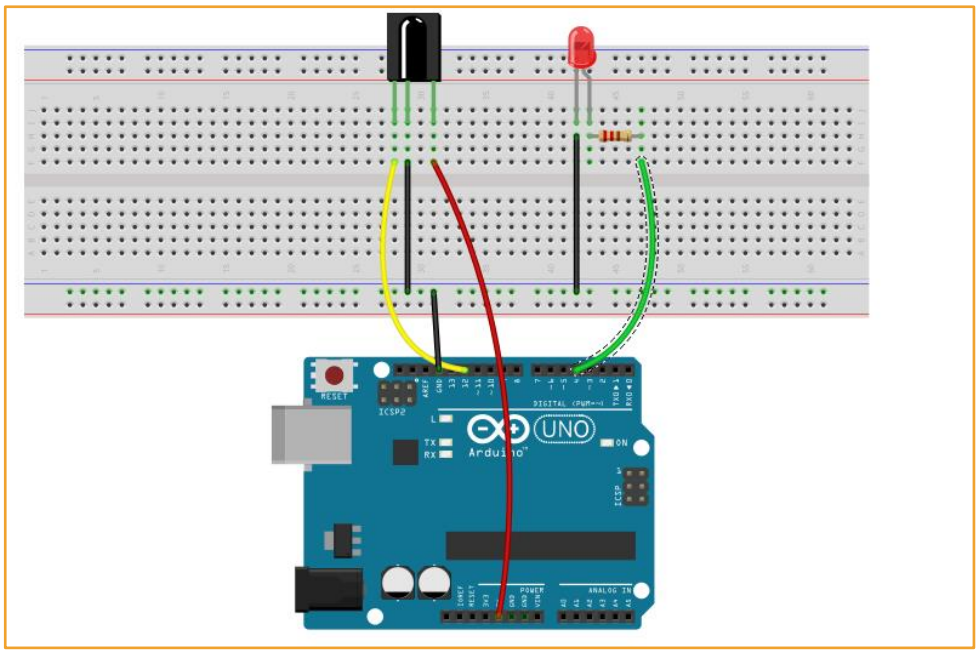
## Component List

- Keywish Arduino Uno R3 mainboard*2
- USB cable*2
- Infrared transmitter module*1
- Infrared receiver module*1
- Jumper wires

## Wiring of Circuit

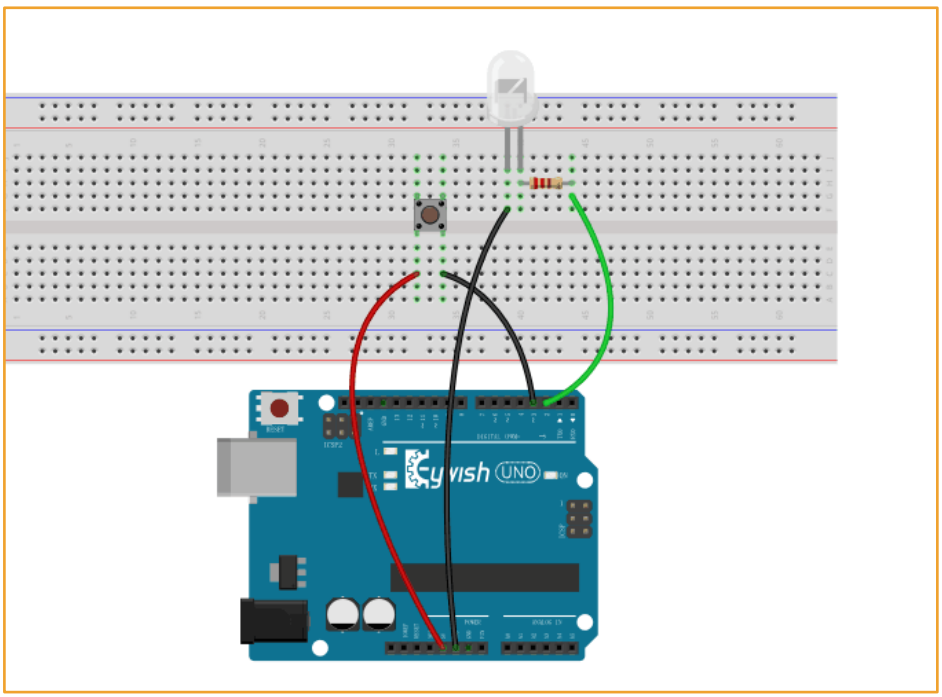
### Infrared receiving connection

| Arduino Uno | Infrared receiver |
|---|---|
| 12 | S |
| GND | - |
| +5V | + |
| Arduino Uno | LED |
| 4 | S |
| GND | GND |



## Infrared transmission experiment

| Arduino Uno | receiver |
|---|---|
| 3 | + |
| GND | G |
| Arduino | button |
| 4 | Pin-1 |
| +5V | Pin-2 |



# Code

## Transmitter:

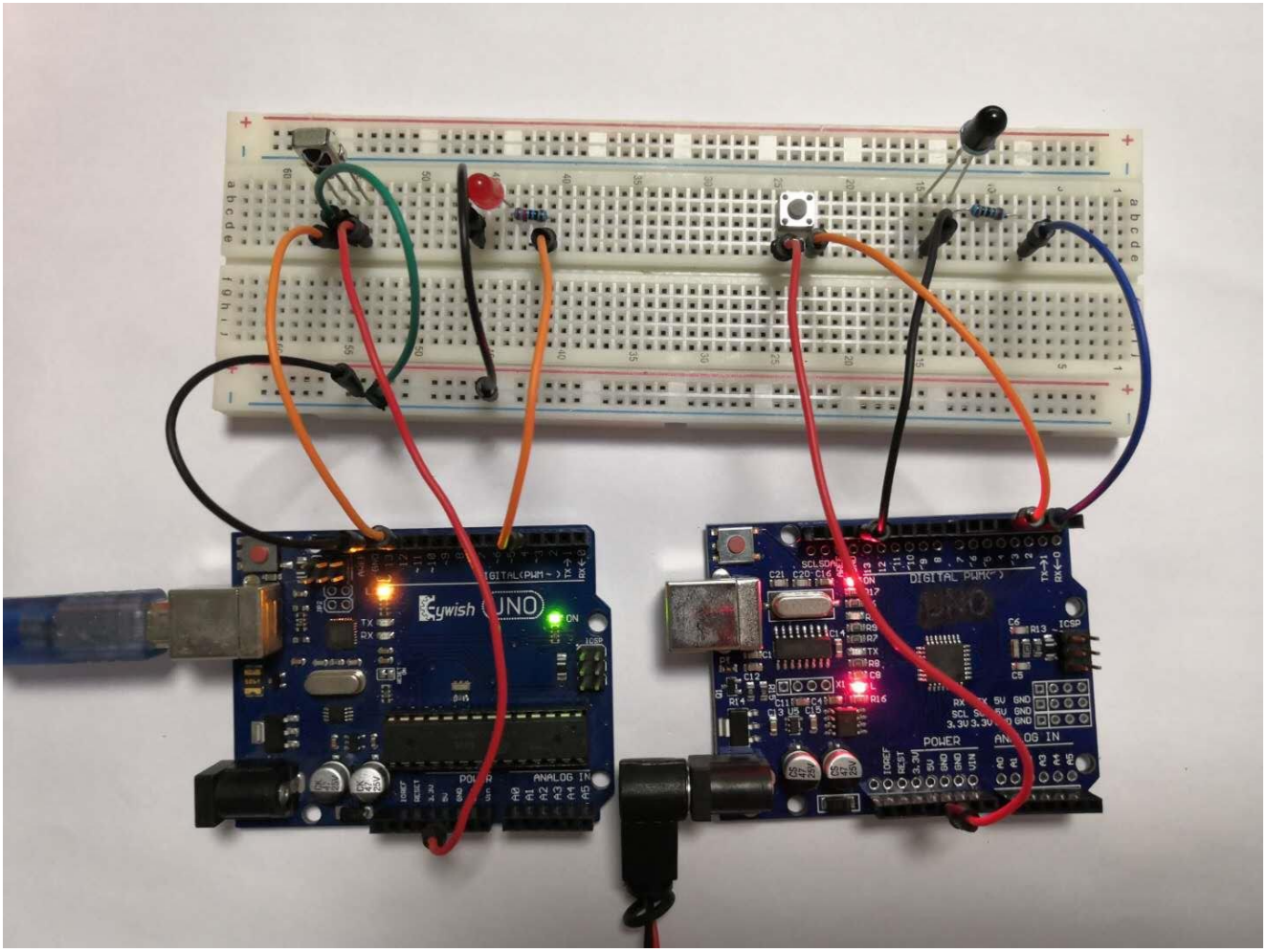
```
#include "IRremote.h"                    // Reference IRRemote library
const int buttonPin = 4;                 // Push button is connected to pin 4
int buttonState = 0;                     // Button status
IRsend irsend;                           // Define IRsend object to emit infrared signal
void setup()
{
    Serial.begin(9600);
    pinMode(buttonPin, INPUT);             // Set buttonPin to input
}

void loop()
{
    // read key status
    buttonState = digitalRead(buttonPin);
  // Check if the button is presse
    // If there is, buttonState will be high level output
    if (buttonState == HIGH)
    {
        // emit infrared signa
        Serial.println("button");
        irsend.sendNEC(0x4CB3817E, 32);   // This code is the code of the key you pressed
(connected to pin4), you can change it at will
    }
    delay(200);
}
```

## Receiver:

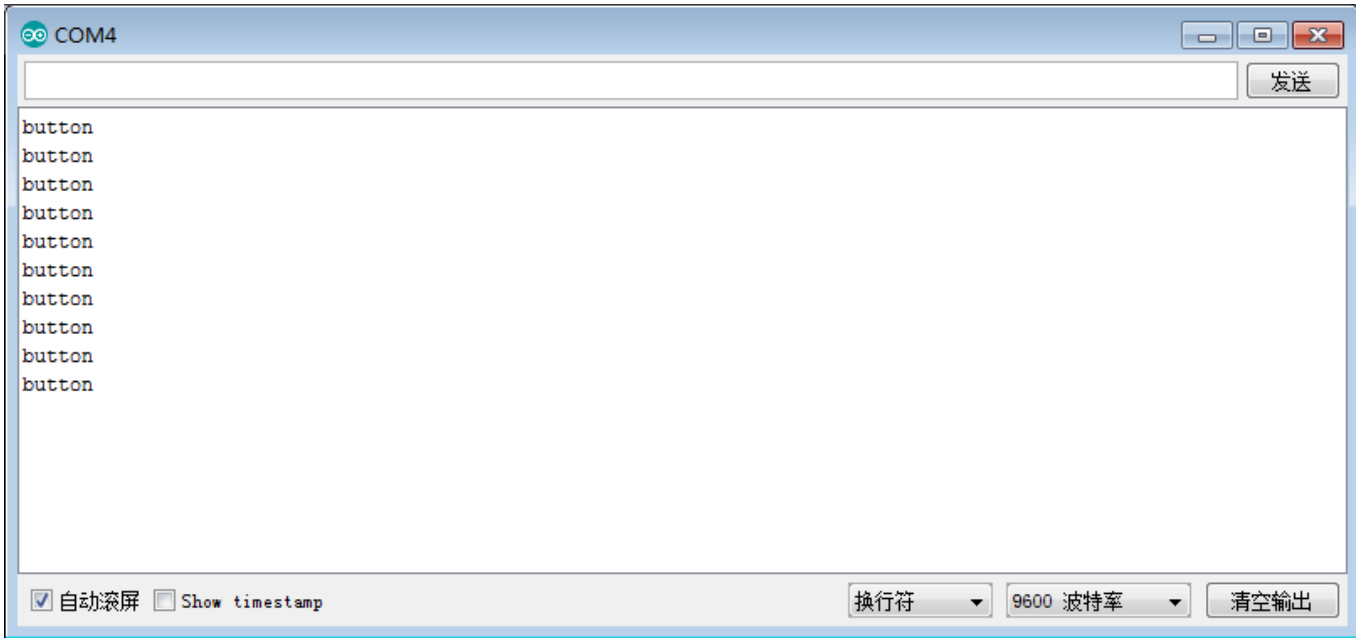
```
#include "IRremote.h"

int RECV_PIN = 12; // Define the pin of infrared receiver as 12
int LED_PIN = 4;  // Define the light-emitting LED pin number 4
int a = 0;
IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
    Serial.begin(9600);
    irrecv.enableIRIn();     // Initialize the infrared receiver
    pinMode(LED_PIN, OUTPUT); // Set the luminous LED pin number 4
}

void loop() {
    if (irrecv.decode(&results))
    {
        Serial.println(results.value, HEX);
        if(results.value == 0x4CB3817E & a == 0) // The 32-bit value here must be consistent
with the value transmitted by the transmitting part
        {
            digitalWrite(LED_PIN,HIGH);// LED lights up
            a=1;
        }
        else if(results.value == 0x4CB3817E & a == 1)
        {
            digitalWrite(LED_PIN,LOW);// LED off
            a=0;
        }
        irrecv.resume(); // Receive the next value
    }
    delay(120);
}
```

## Experiment Result

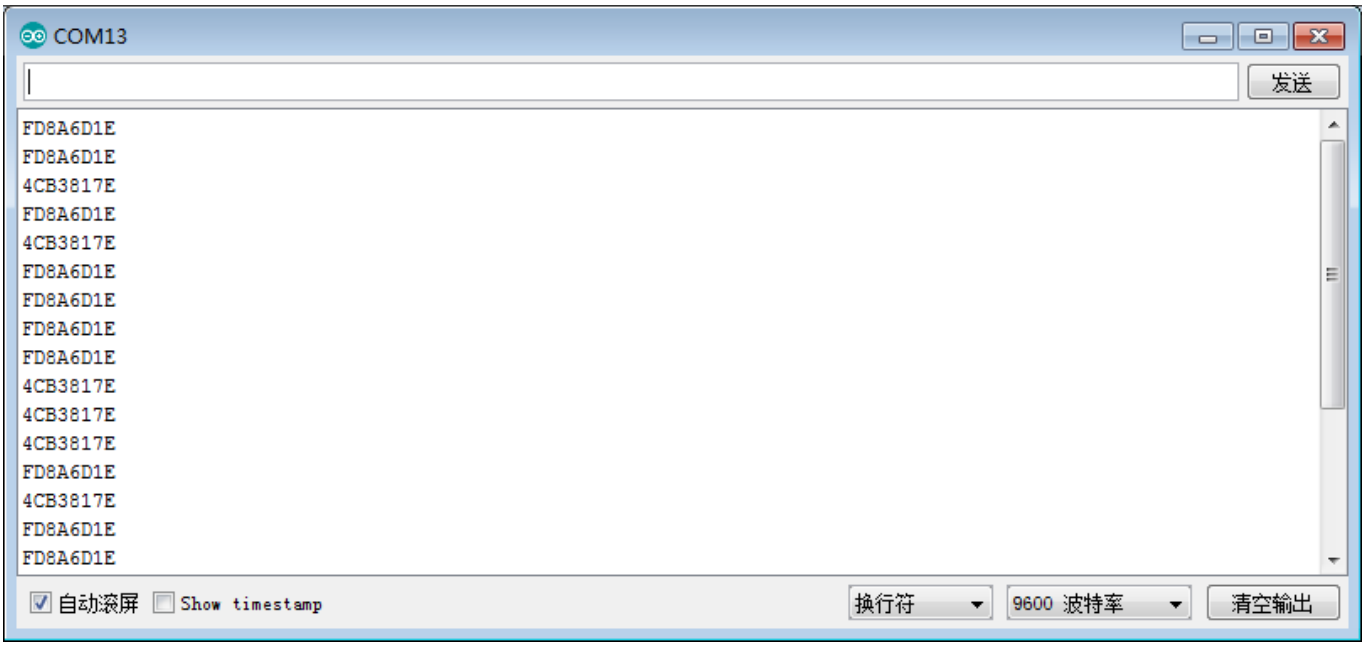
This time we use two Arduino mainboards, when downloading the codes, you have to know which is the transmitter code and which is receiver code. If the download is wrong, there will be no result. After downloading the codes, we open the Serial Monitor window, if you can see the following displayed data, it shows that you succeed.



## Transmitter Exeripment Result

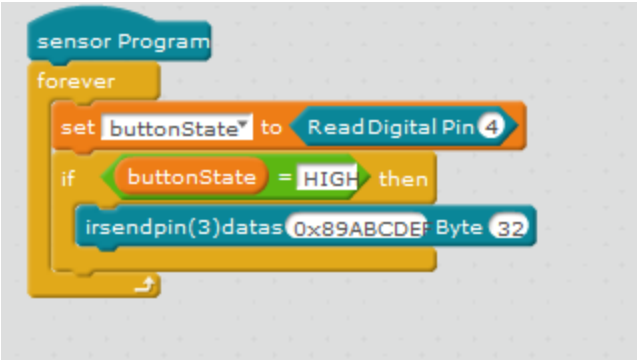



## Receiver Exeripment Result

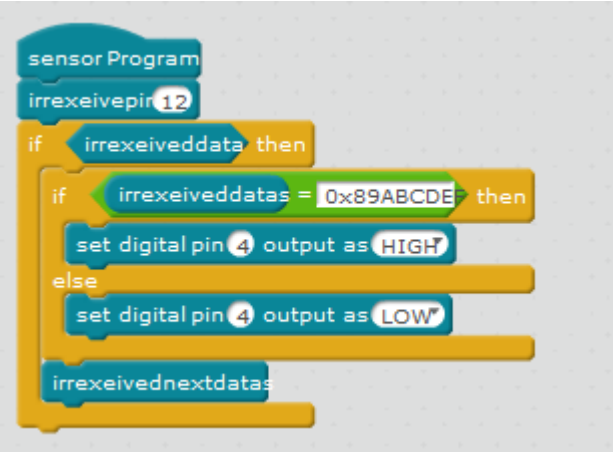


# MBlock graphical programming program

## Transmitter:

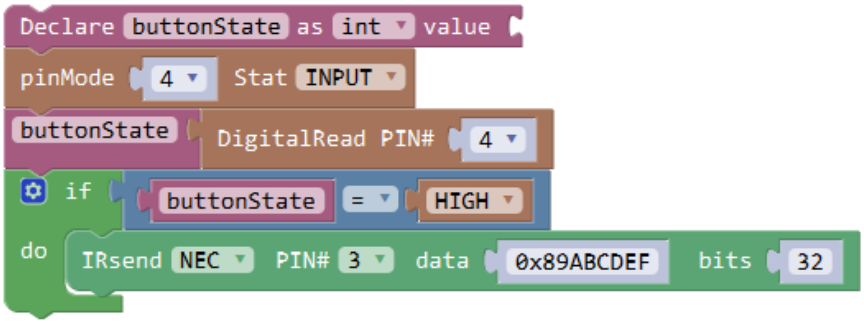



## Receiver:

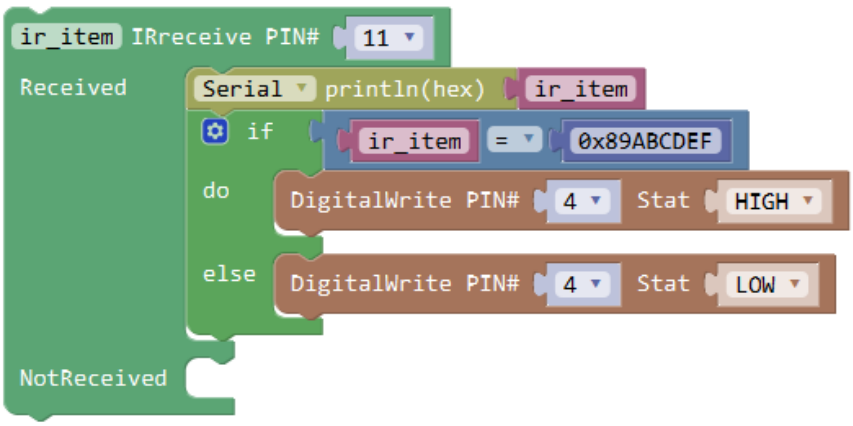


## Mixly graphical programming program

Transmitter:

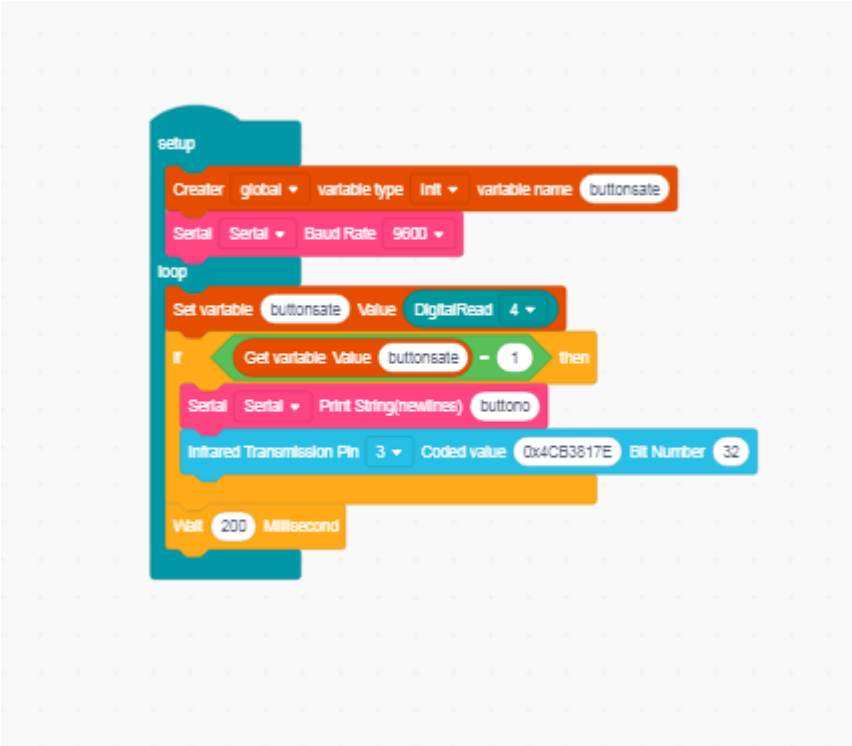



Receiver:



# MagicBlock graphical programming program

Transmitter:




Receiver