

Infrared communication

Outline

1. Introduction

- a. What is the Infrared?
- b. How do the infrared receiver and emitter work?
- c. Infrared communication protocols
- d. IR communication advantages and disadvantages
- e. IRremote.h library

2. Tutorials

- a. Tutorial 1: Receiving IR signals
- b. Tutorial 2: Emitter
- c. Tutorial 3: IR Chat

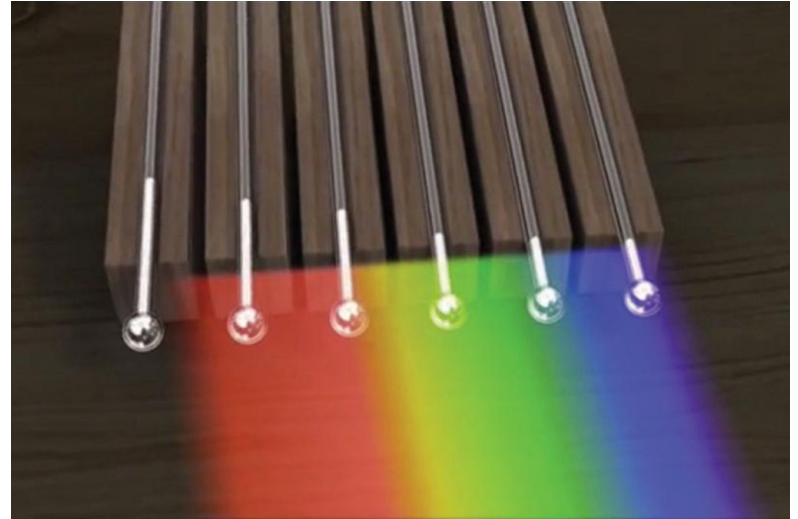
3. Conclusions

Introduction

In 1800, [William Herschel](#) conducted an experiment measuring the difference in temperature between the colors.

Results showed an increase in temperature from blue to red.

Herschel discovered infrared light noticing an even warmer temperature measurement just beyond the red end of the visible spectrum.



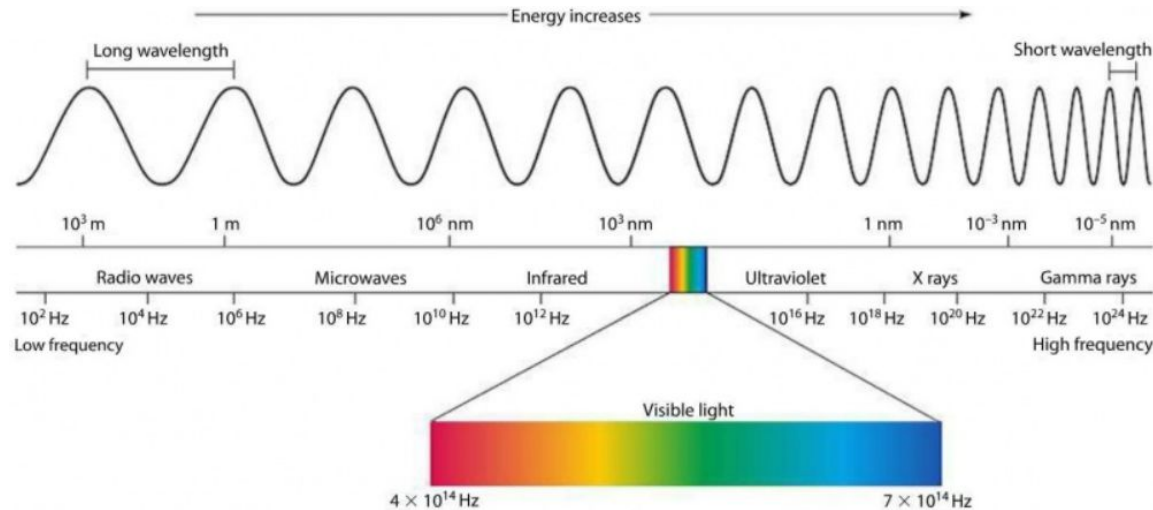
shorturl.at/ilS57

IR light corresponds to a range of the electromagnetic spectrum between 300 GHz to 400 THz.

Common natural sources are solar radiation and fire.

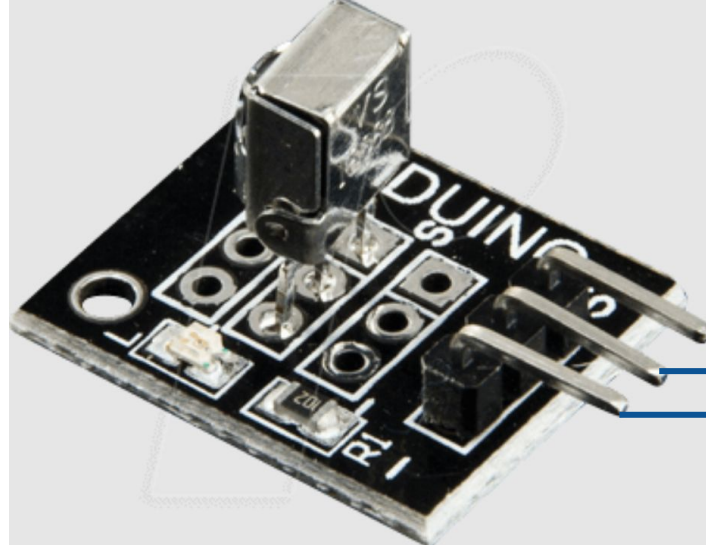
Common artificial sources are lamps and heating devices.

Which means that direct use of the IR receiving sensor may prove faulty if used directly outdoors without special filters.



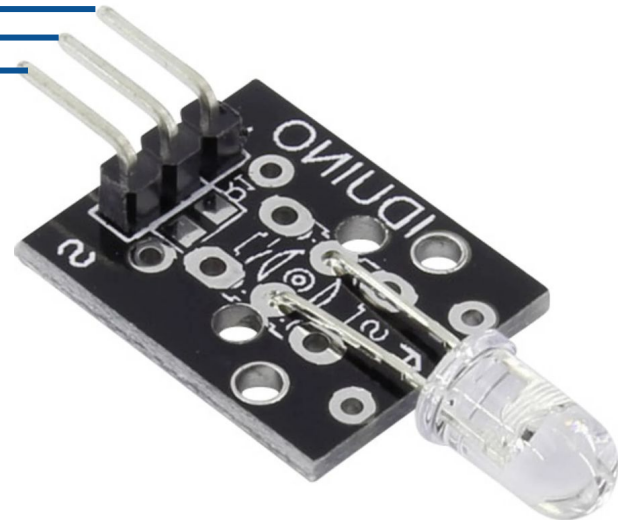
Observations

While working on how to combine the sender and the receiver we found that if both the sensors on the same breadboard are very close in proximity to each other, there is a danger of the signal to be intercepted by the same receiver.



SIGNAL
VCC (5 or 3.3 V)
GND

GND
VCC (5 or 3.3 V)
SIGNAL



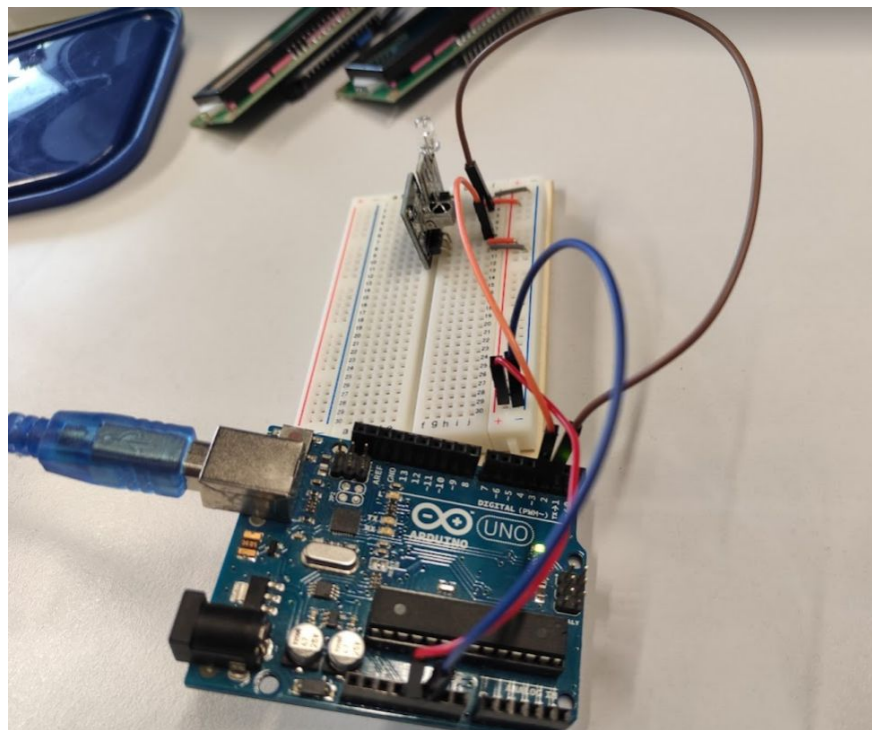
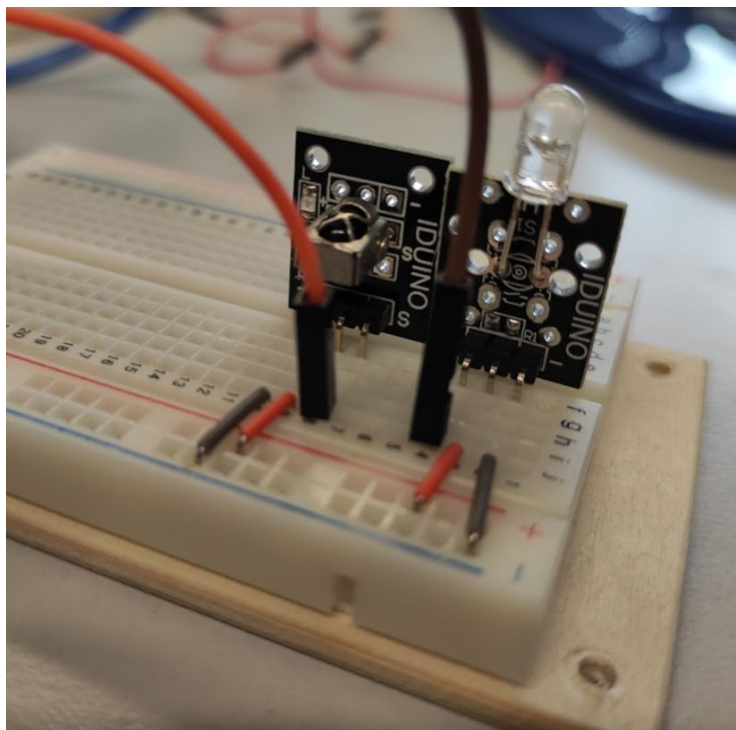
Tutorials followed

<https://mschoeffler.com/2021/05/01/arduino-tutorial-ir-transmitter-and-ir-receiver-hx-m121-hx-53-ky-005-ky-022-keyes-iduino-open-smart/>

https://kogyikaungtet.com/ir_text/

IRremote library

<https://arduino-irremote.github.io/Arduino-IRremote/classIRsend.html>



Receiver

```
1  #include <IRremote.h> // >v3.0.0
2
3  #define PIN_RECV 5 // Pin connected to the IR receiver
4
5  void setup()
6  {
7      Serial.begin(9600); //initialize serial connection to print on the Serial Monitor of the Arduino IDE
8      IrReceiver.begin(PIN_RECV); // Initializes the IR receiver object
9  }
10
11 void loop()
12 {
13     if (IrReceiver.decode()) {
14         Serial.println("Received something...");
15         IrReceiver.printIRResultShort(&Serial); // Prints a summary of the received data. The command is received as HEX.
16         Serial.println();
17         Serial.println((char) IrReceiver.decodedIRData.command); // Converts the ASCII value received as command into a char and prints it.
18         Serial.println();
19         IrReceiver.resume(); // Important, enables to receive the next IR signal
20     }
21 }
22
```

```
#include <IRremote.h> // >v3.0.0
```

```
#define PIN_RECV 5 // Pin connected to the IR receiver
```

```
void setup()
{
  Serial.begin(9600); //initialize serial connection to print on the Serial Monitor of the Arduino IDE
  IrReceiver.begin(PIN_RECV); // Initializes the IR receiver object
}
```

```
void loop()
{
  if (IrReceiver.decode()) {
    Serial.println("Received something...");
    IrReceiver.printIRResultShort(&Serial); // Prints a summary of the received data. The command is received as HEX.
    Serial.println();
    Serial.println((char) IrReceiver.decodedIRData.command); // Converts the ASCII value received as command into a char and prints it
    Serial.println();
    IrReceiver.resume(); // Important, enables to receive the next IR signal
  }
}
```

Sender

```
1  #include <IRremote.h> // >v3.0.0
2
3  #define PIN_SEND 6 // Pin connected to the IR emitter
4  char input;
5
6  void setup()
7  {
8      IrSender.begin(PIN_SEND); // Initializes IR sender
9      Serial.begin(9600);
10     delay(1000);
11     Serial.println("Type something!");
12 }
13
14 void loop()
15 {
16
17     if(Serial.available()){ // Checks if something is written in the serial monitor
18         input = Serial.read();
19         Serial.print("You typed: ");
20         Serial.println(input);
21         IrSender.sendNEC(0xFE01D6DE, "abcd", 0); // the address 0xFE01D6DE with the command input and 0 repetitions is sent.
22     }
23
24     delay(1000); // wait for one second
25 }
26
```

Sender

```
#include <IRremote.h> // >v3.0.0

#define PIN_SEND 6 // Pin connected to the IR emitter
char input;
```

```
void setup()
{
  IrSender.begin(PIN_SEND); // Initializes IR sender
  Serial.begin(9600);
  delay(1000);
  Serial.println("Type something!");
}
```

```
void loop()
{
  if(Serial.available()){ // Checks if something is written in the serial monitor
    input = Serial.read();
    Serial.print("You typed: ");
    Serial.println(input);
    IrSender.sendNEC(0xFE01D6DE, "abcd", 0); // the address 0xFE01D6DE with the command input and 0 repetitions is sent.
  }

  delay(1000); // wait for one second
}
```

Receiver + Sender

```
1  #include <IRremote.h> // >v3.0.0 | tested on v3.9.0
2
3  #define PIN_RECV 3 // Pin connected to the IR receiver
4  #define PIN_SEND 2 // Pin connected to the IR emitter
5
6  char input;          // Related to the emitter
7
8  const uint16_t MY_ADDRESS = 0xAAAA; // Addresses must be different from the other emitter-receiver
9
10 void setup() {
11     // START CODE FOR THE RECEIVER
12     Serial.begin(9600); //initialize serial connection to print on the Serial Monitor of the Arduino IDE
13     IrReceiver.begin(PIN_RECV); // Initializes the IR receiver object
14     // END CODE OF THE RECEIVER
15
16     // START CODE FOR THE SENDER
17     IrSender.begin(PIN_SEND); // Initializes IR sender
18     delay(1000);
19     Serial.println("Type something!");
20     // END CODE FOR THE SENDER
21 }
```

```

23 void loop() {
24     // START CODE FOR THE RECEIVER
25     if (IrReceiver.decode()) {
26
27         if(!IrReceiver.decodedIRData.address == MY_ADDRESS){
28             Serial.println("Received something...");
29             IrReceiver.printIRResultShort(&Serial); // Prints a summary of the received data. The command is received as HEX.
30             Serial.println();
31             Serial.println((char) IrReceiver.decodedIRData.command); // Converts the ASCII value received as command into a char and prints it.
32         }
33
34         Serial.println();
35         IrReceiver.resume(); // Important, enables to receive the next IR signal
36     }
37     // END CODE OF THE RECEIVER
38
39     // START CODE FOR THE SENDER
40     if(Serial.available()){ // Checks if something is written in the serial monitor
41         input = Serial.read();
42         Serial.print("You typed: ");
43         Serial.println(input);
44         IrSender.sendNEC(MY_ADDRESS, input, 0); // the address 0xFE01D6DE with the command input and 0 repetitions is sent.
45     }
46
47     delay(50); // wait for one second
48     // END CODE FOR THE SENDER
49 }
50

```

LCD without I2C

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 7, en = 6, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

The circuit:

- * LCD RS pin to digital pin 7
- * LCD Enable pin to digital pin 6
- * LCD D4 pin to digital pin 5
- * LCD D5 pin to digital pin 4
- * LCD D6 pin to digital pin 3
- * LCD D7 pin to digital pin 2
- * LCD R/W pin to ground
- * LCD VSS pin to ground
- * LCD VCC pin to 5V
- * 10K resistor:
 - * ends to +5V and ground
 - * wiper to LCD VO pin (pin 3)

LCD with I2C

```
void loop() {  
  // Print 'Hello World!' on the first line of the LCD:  
  lcd.setCursor(0, 0); // Set the cursor on the first column and first row.  
  lcd.print("Hello Sandro!"); // Print the string "Hello World!"  
  
  lcd.setCursor(0, 1); //Set the cursor on the first column and the second row (counting starts at 0!).  
  lcd.print("This works!");  
}
```