Infra Red Sensing through Arduino

Alok Ranjan Kesari[†] and G V V Sharma[*]

1 HARDWARE/SOFTWARE SETUP FOR IR SENSING

Problem 1. Connect the 5V pin of the arduino to one extreme of the breadboard in Fig. 1.
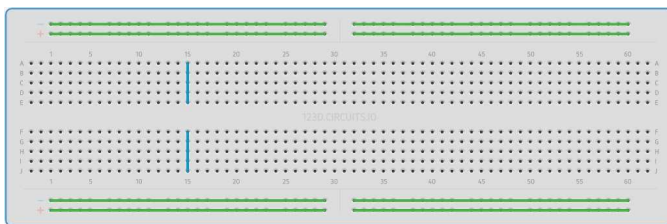

Fig. 1

Problem 2. Connect the GND pin of the arduino to the opposite extreme pin of the breadboard.

Problem 3. SM0038/TSOP1738, IR Sensor is used to detect IR radiations emitted from IR remotes and convert it into a sequence of numbers to be transmitted serially to Arduino or any microcontroller. Every remote follows IR protocols and transmits serially at 38 KHz frequency. The sequence transmitted is unique for any protocol. It is a 48 bit sequence. Plug the IR sensor in Fig. 3 to the breadboard.

Problem 4. Connect the OUT pin of the IR sensor to the D11 pin of the Arduino.

Problem 5. Download the IRremote library from the internet and copy it into libraries in the Arduino folder.

Problem 6. Connect the Arduino to the computer.

2 IDENTIFYING THE REMOTE BUTTONS

Problem 7. Upload the following code into the Arduino.

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. † The author was an intern at IIT Hyderabad in the summer of 2017 e-mail: alok.kesari@yahoo.co.in.
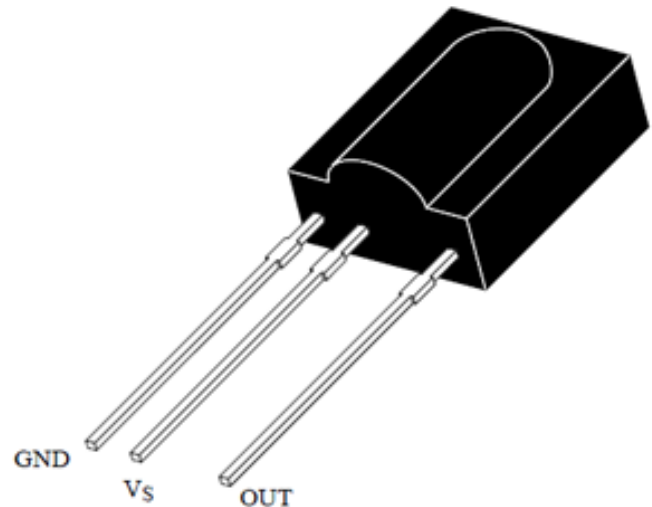
Fig. 3

```
#include "Arduino.h"
#include <IRremote.h> //header
    file for IR receiver
int irpin = 11; //pin number to
    which IR sensor is connected
IRrecv irrecv(irpin); //object of
    class IRrecv included in the
    header file IRremote.h which
    specifies which pin of arduino
    is connected to the sensor
decode_results results; //decoded
    value of the result stored as
    object of the class
    decode_results included in the
    header file IRremote.h
void setup()
{
  Serial.begin(9600); //Setting the
      baud rate of serial
      communication
  irrecv.enableIRIn(); // Start IR
```

Button	Code
2	16758855
4	16756815
6	16767015
8	16754775
Terminating Code	**4294967295**

TABLE 8

```
    receiver
}
void loop() {
 if (irrecv.decode(&results)) { //
    checking whether decoded value
    is not a null value
  {
    Serial.println(results.value);
       // printing the results on the
       serial monitor
    irrecv.resume(); // Receive the
       next value
  }
 }
}
```

Problem 8. Press different remote buttons. This will throw up some numbers. There will be a common terminating code for all the buttons. The desired code for the button is the one that appears before the terminating code.

Each IR remote will give different codes for different buttons. For the remote considered in this manual, the button codes are as in Table 8.

3 IR CONTROL

Problem 9. Plug 4 LEDs and corresponding resistors on the breadboard. Make sure that one end of the resistor is connected to GND.

Problem 10. Connect the LEDs to D2-D5 pins of the Arduino respectively.

Problem 11. Upload the following code to the Arduino and use the IR remote to turn each LED ON/OFF. You will have to modify the code according to the codes for the IR remote buttons.

```
#include "Arduino.h"
```

```
#include <IRremote.h> // header
    file for IR receiver
int irpin = 11; // pin number to
    which IR sensor is connected
IRrecv irrecv(irpin); // object of
    class IRrecv included in the
    header file IRremote.h which
    specifies which pin of arduino
    is connected to the sensor
decode_results results; // object
    of class decode_results included
     in the header file IRremote.h
    which specifies the obtained
    value from the sensor
void setup() {
  Serial.begin(9600); // Setting the
     baud rate of serial
    communication
  irrecv.enableIRIn(); // Start IR
    receiver
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop() {
  if (irrecv.decode(&results)) //
    checking whether decoded value
    is not a null value
  {
    if (results.value==16758855){
       // code for button press 2
    digitalWrite(2, HIGH);
    }
    else if (results.value
       ==16756815){ // code for
       button press 8
    digitalWrite(3, HIGH);
    }
    else if (results.value
       ==16767015){ // code for
       button press 4
    digitalWrite(4, HIGH);
    }
    else if(results.value
       ==16754775){ // code for
       button press 6
    digitalWrite(5, HIGH);
    }
```

```
    else{ //code for stopping
       digitalWrite(2, LOW);
       digitalWrite(3, LOW);
       digitalWrite(4, LOW);
       digitalWrite(5, LOW);
    }

    irrecv.resume(); //for
       checking next value, resume
       () function is included from
        IRrecv class of IRremote.h
       header file
  }
}
```