2015

Application Note on Interfacing Arduino with Universal GPIO board

Author: vivek.g.s

Reviewer:

Version1.0

Interfacing Arduino UNO with universal GPIO board for TSOP

Introduction

In this application note we will be discussing on interfacing universal GPIO board's TSOP with Arduino UNO to display on LCD, which button is being pressed. Here we will be connecting the TSOP output to Arduino to display the received data that is being received from TSOP which is pressed on remote.

Arduino UNO: Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. All this is defined by a set of instructions programmed through the Arduino Software (IDE).

Universal GPIO board: Ready to use Input and Output circuits are always important to experiment with any microcontroller. The Universal GPIO Board is very useful for beginners, hobbyist and students. It is suitable for carrying out quick experiments with any microcontroller and lets you access numerous peripheral devices. It provides access to pins through male connectors for wiring to the microcontroller development board.

This board has below listed interface circuits to work with:

1. 8 LEDs
2. 4 Switches
3. 1 Potentiometer
4. 1 Light Sensor (using Light Dependent Resistor)
5. 1 Temperature Sensor
6. 1 Infrared Receiver
7. 1 Buzzer
8. 1 Seven Segment Display

TSOP: The **TSOP 1738** is a member of **IR remote control receiver** series. This IR sensor module consists of a PIN diode and a pre amplifier which are embedded into a single package. The output of **TSOP** is active low and it gives +5V in off state. When IR waves, from a source, with a Centre frequency of 38 kHz incident on it, its output goes low.

TSOP module has an inbuilt control circuit for amplifying the coded pulses from the IR transmitter. A signal is generated when PIN photodiode receives the signals. This input signal is received by an automatic gain control (AGC). For a range of inputs, the output is fed back to AGC in order to adjust the gain to a suitable level. The signal from AGC is passed to a

band pass filter to filter undesired frequencies. After this, the signal goes to a demodulator and this demodulated output drives an NPN transistor. The collector output of the transistor is obtained at pin 3 of TSOP module.
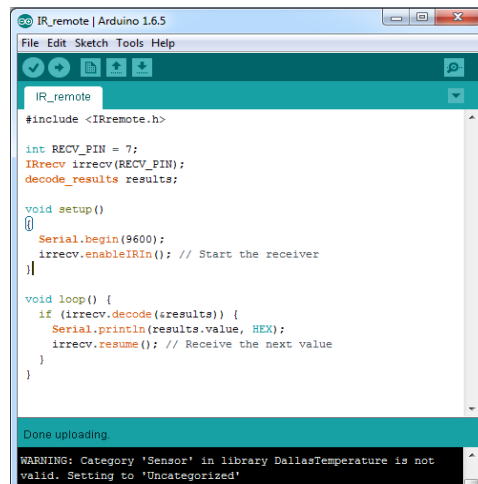
Step1. The Materials required are:

- Arduino UNO
- Universal GPIO board
- 2 channel Relay breakout board
- Power supply breakout board
- Bulb
- 12V, 1A adapter
- Male to Female Jumpers

1. Open Arduino sketch on your PC or Laptop to start the programming.
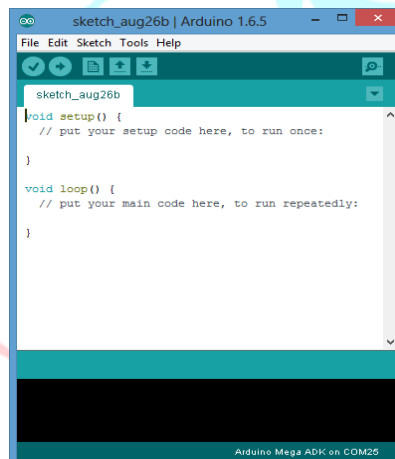
- Include the **IRremote** library in the blank sketch.
- Include the IRremote Library.

- Type the program for the IR_Remote to produce output whenever the key is pressed in remote.
- Click on verify and check for any errors in the program. If no errors are present select the Arduino UNO in IDE. Go to tools> Board> Select Arduino UNO.
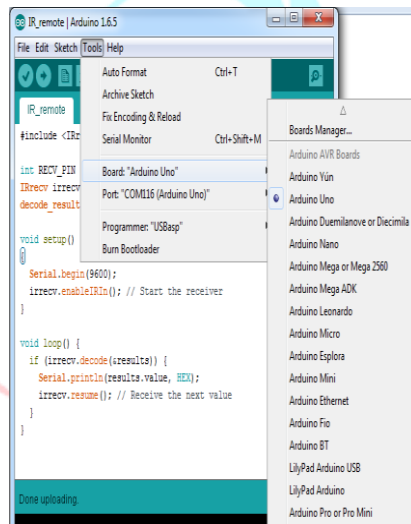
- Select port of programming by Tools> Port> Select the port for programming
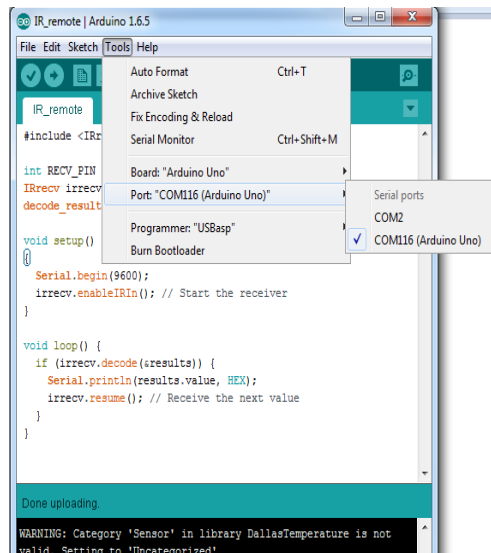
- Now Upload the program to the arduino
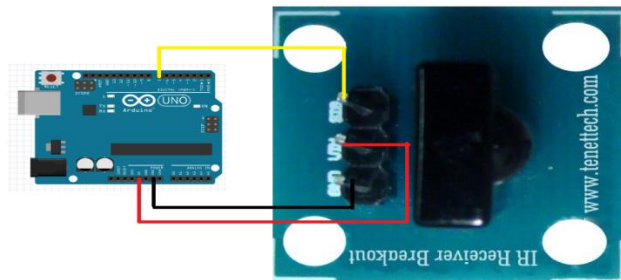
CODE:

#include <IRremote.h>

int RECV_PIN = 7;

IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()

{

```
  Serial.begin(9600);

  irrecv.enableIRIn(); // Start the receiver

}


void loop() {

  if (irrecv.decode(&results)) {

    Serial.println(results.value, HEX);

    irrecv.resume(); // Receive the next value

  }

}
```
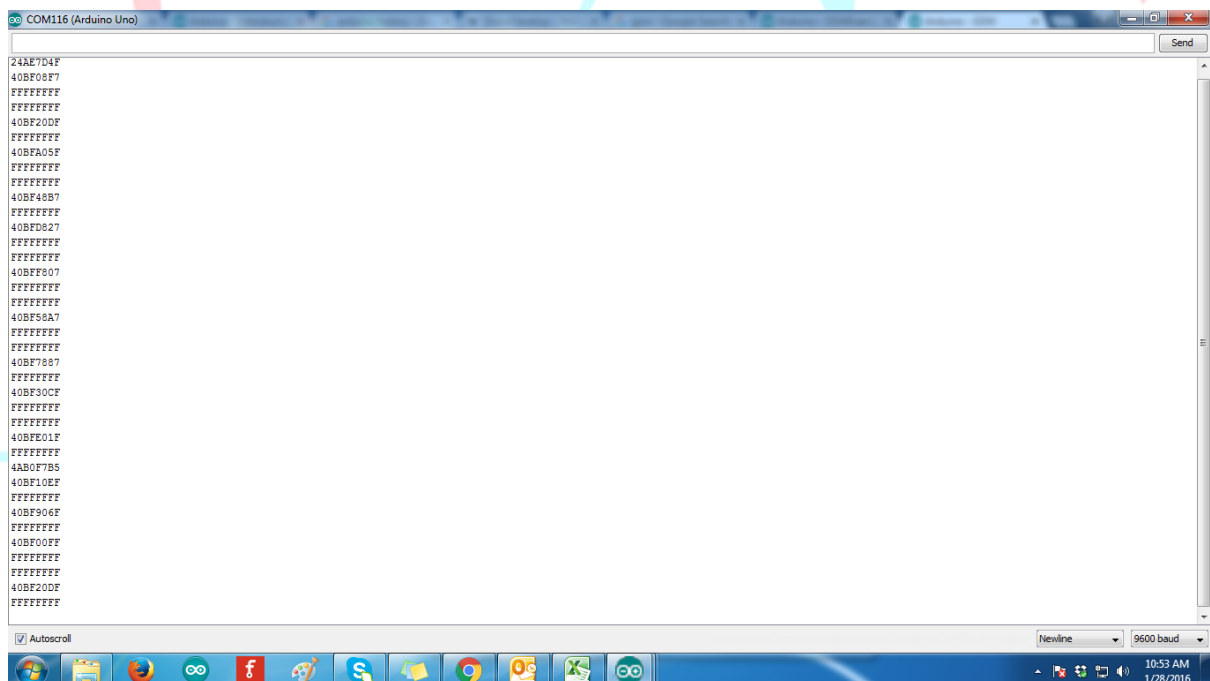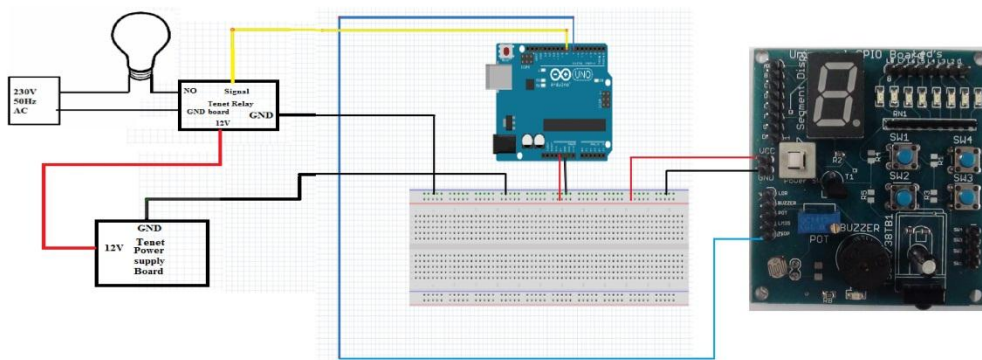
OUTPUT:

Interfacing TSOP with bulb:

Code:

#include <IRremote.h> // use the library for IR

int receiver = 7; // pin 1 of IR receiver to Arduino digital pin 11

int BULB = 8;

IRrecv irrecv(receiver); // create instance of 'irrecv'

decode_results results;

void setup()

{

pinMode(BULB, OUTPUT);

** irrecv.enableIRIn(); // Start the receiver**

** Serial.begin(9600);**

}

void translateIR() // takes action based on IR code received

// describing Sony IR codes on LCD module

```
{
  switch(results.value)
  {
    case 0x40BF08F7: Serial.println(" MUTE          ");
    digitalWrite(BULB, HIGH);
                                        break;
    case 0x40BF807F: Serial.println(" ONE           "); break;
    case 0x40BF40BF: Serial.println(" TWO           "); break;
    case 0x40BFC03F: Serial.println(" THREE         "); break;
    case 0x40BF20DF: Serial.println(" FOUR          "); break;
    case 0x40BFA05F: Serial.println(" FIVE          "); break;
    case 0x40BF609F: Serial.println(" SIX           "); break;
    case 0x40BFE01F: Serial.println(" SEVEN         "); break;
    case 0x40BF10EF: Serial.println(" EIGHT         "); break;
    case 0x40BF906F: Serial.println(" NINE          "); break;
    case 0x40BF00FF: Serial.println(" ZERO          "); break;
    case 0x40BF48B7: Serial.println(" POWER         ");
    digitalWrite(BULB, LOW);                    break;
    case 0x40BF30CF: Serial.println(" MENU          "); break;
    case 0x40BF58A7: Serial.println(" volume up     "); break;
    case 0x40BF7887: Serial.println(" volume down   "); break;
    case 0x40BFD827: Serial.println(" channel up    "); break;
    case 0x40BFF807: Serial.println(" channel down  "); break;
    //default: lcd.println(" other button   ");
  }
```
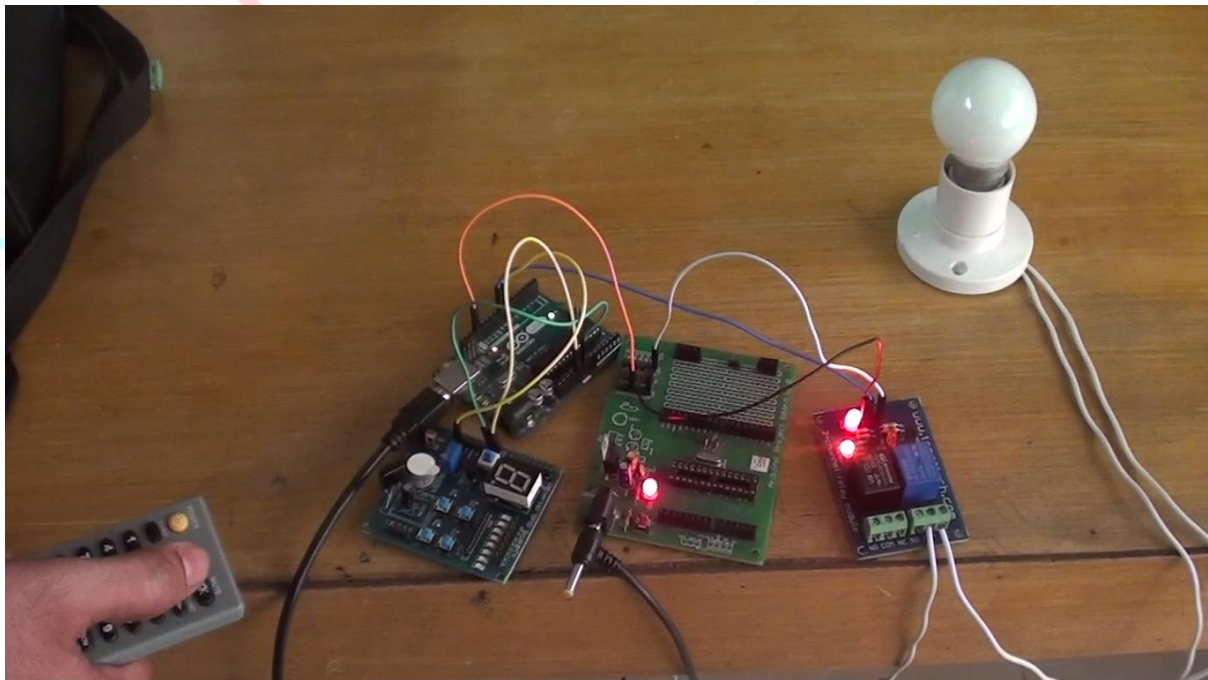
```
  delay(500);


}

void loop()

{

  if (irrecv.decode(&results)) // have we received an IR signal?

  {

    translateIR();

    for (int z=0; z<2; z++) // ignore 2nd and 3rd signal repeat

    {

      irrecv.resume(); // receive the next value

    }

  }

}
```
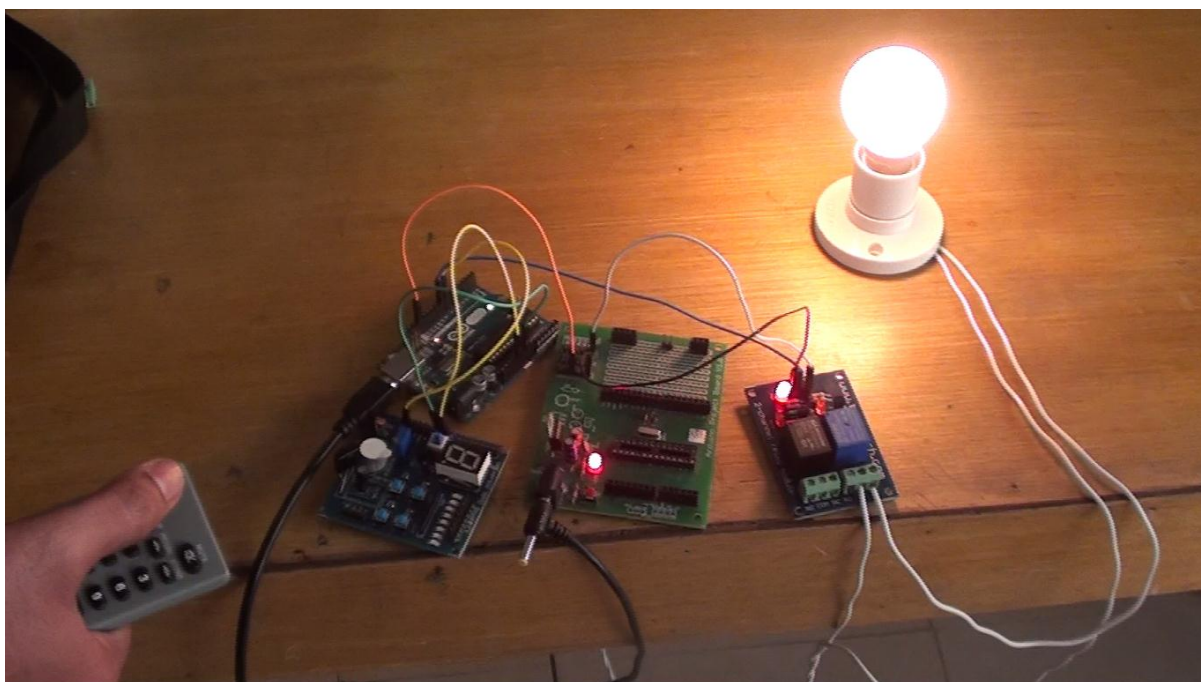
OUTPUT:

For more information please visit: www.tenettech.com

For technical query please send an e-mail: info@tenettech.com

For product info:

1. http://www.tenettech.com/search?q=arduino+uno&r1=default

2. http://tenettech.com/product/6655/universal-gpio-board#contentGuestbook-tab

3. http://tenettech.com/product/6068/power-supply-breakout-board

4.