Research Design Lab

ARDUINO EXPERIMENTS

ARDUINO EXPERIMENTS

IR OBSTACLE SENSOR

OVERVIEW

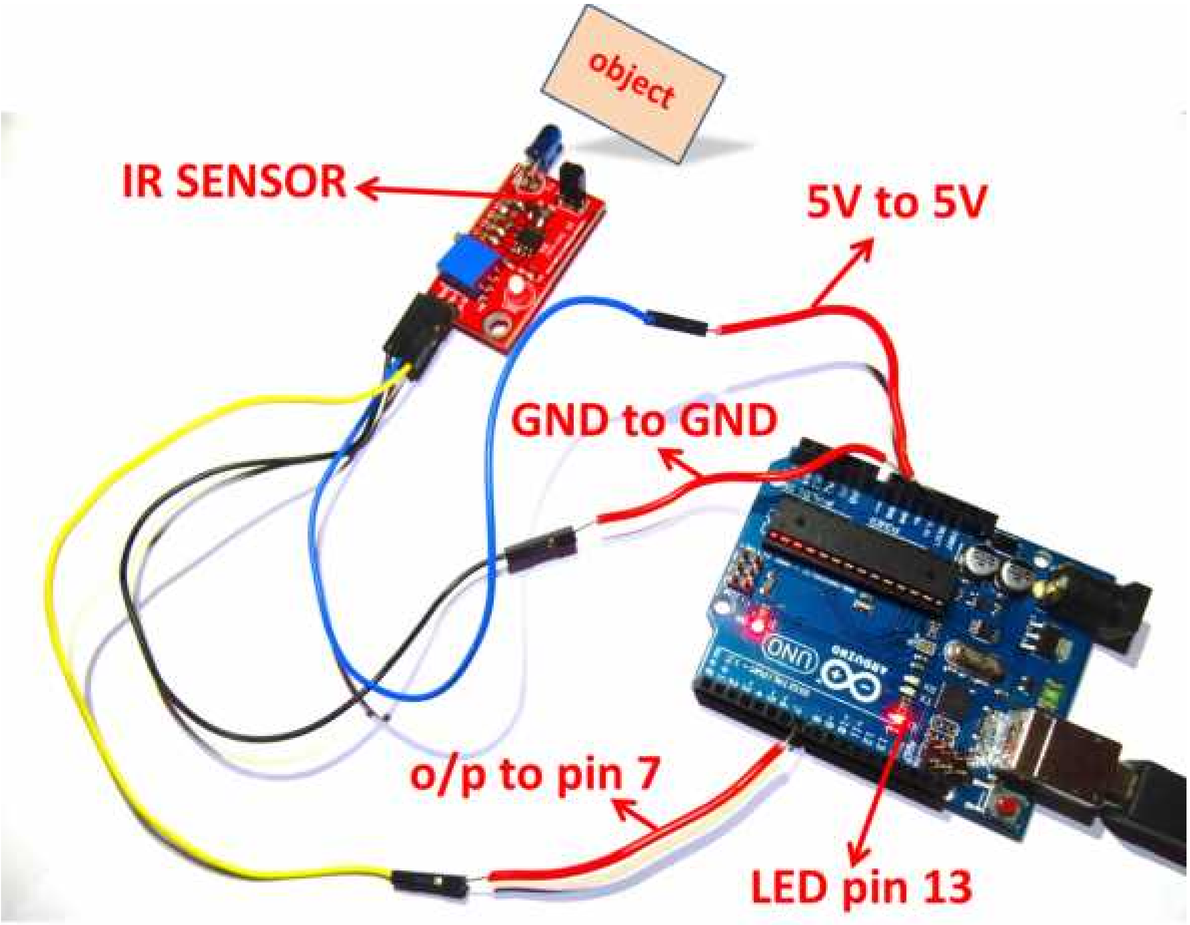
Based on simple basic idea, the sensor is build which is easy to calibrate. This sensor provides a detection range of 10 – 30 cm.This sensor can be used for most of the indoor applications where no important ambient light is present. It follows the same principle as in all Infra – Red proximity sensors. The basic idea is to send infra red light though IR – LED which reflects any object in front of the sensor.

OBJECTIVE OF THE EXPERIMENT

If object is detected pin 13 will go high (onboard LED ON) and "object detected" message will be displayed in serial monitor

If object is not detected pin 13 will go low (onboard LED OFF) and "object not detected" message will be displayed in serial monitor

EXPERIMENTAL SETUP

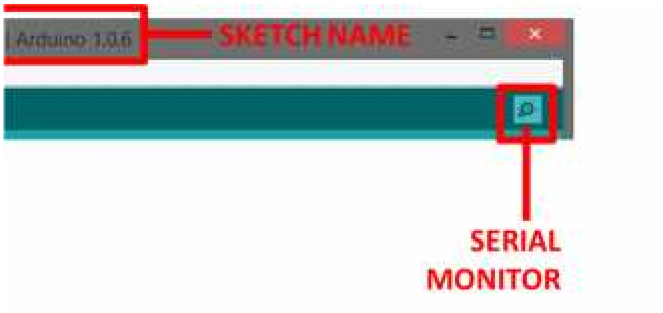


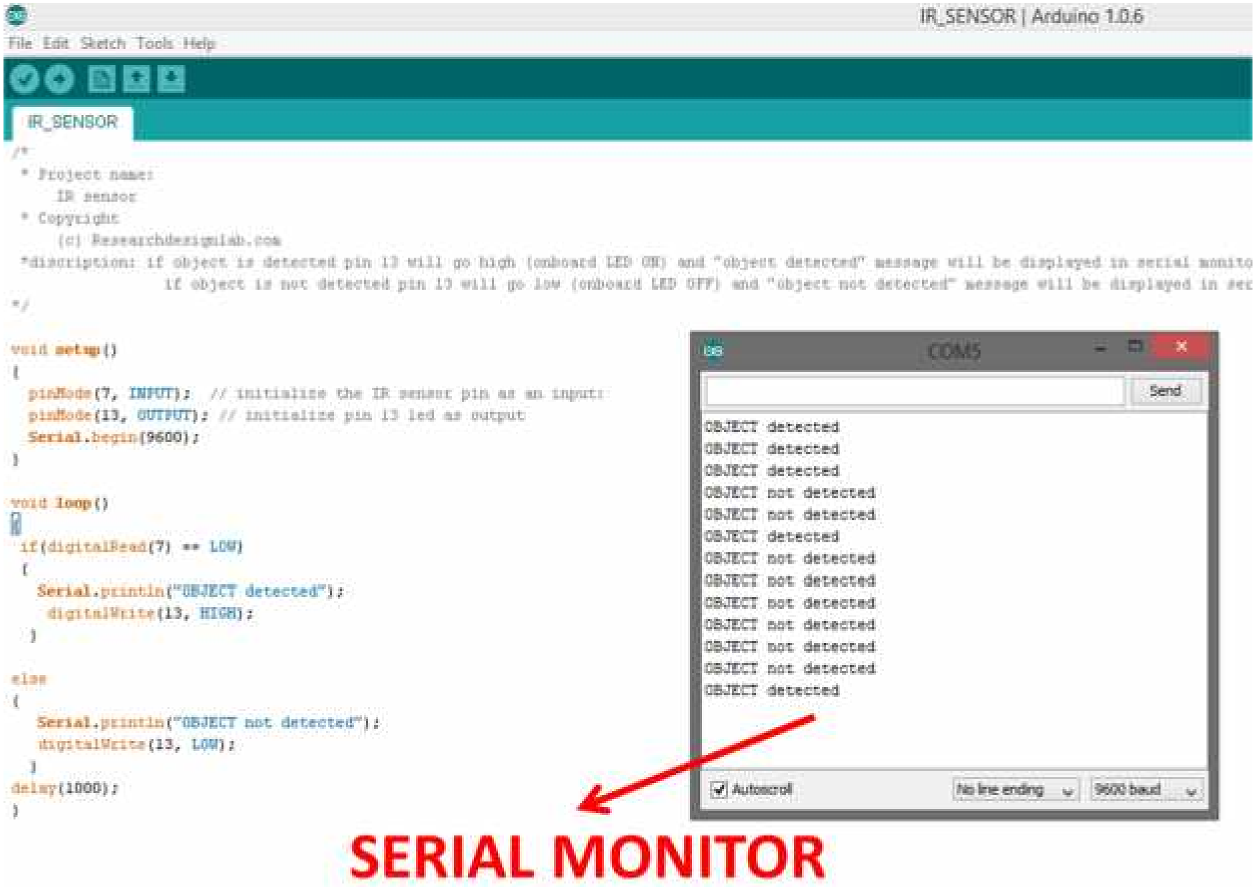
IR SENSOR ARDUINO CODE

```
/*
* Project name:
    IR sensor
* Copyright
    (c) Researchdesignlab.com
*description:   if object is detected pin 13 will go high (onboard LED ON) and "object detected"
                message will be displayed in serial monitor
                if object is not detected pin 13 will go low (onboard LED OFF) and "object not detected"
                message will be displayed in serial monitor
*/
void setup()
{
pinMode(7, INPUT);                      // initialize the IR sensor pin as an input:
pinMode(13, OUTPUT);                    // initialize pin 13 led as output
Serial.begin(9600); //baud rate
}

void loop()
{
if(digitalRead(7) == LOW)               // if object detected IR sensor sends 0 to pin 7
{
Serial.println("OBJECT detected");      //"object detected" message will be displayed in serial monitor
digitalWrite(13, HIGH);                 //led pin 13 will be turned on
}
else
{
Serial.println("OBJECT not detected");  //"object not detected" message will be displayed in serial
                                        monitor
digitalWrite(13, LOW);                  //led pin 13 will be turned off
}
delay(1000);                            //delay of one second
}
```

After compiling and uploading the above code, click on serial monitor in Ardunio ide to observe the output.

ARDUINO IDE – SERIAL MONITOR





```
/*
 * Project name:
    IR sensor
 * Copyright
    (c) Researchdesignlab.com
 *discription: if object is detected pin 13 will go high (onboard LED ON) and "object detected" message will be displayed in serial monito
              if object is not detected pin 13 will go low (onboard LED OFF) and "object not detected" message will be displayed in ser
*/

void setup()
{
  pinMode(7, INPUT);  // initialize the IR sensor pin as an input:
  pinMode(13, OUTPUT); // initialize pin 13 led as output
  Serial.begin(9600);
}

void loop()
{
 if(digitalRead(7) == LOW)
 {
   Serial.println("OBJECT detected");
    digitalWrite(13, HIGH);
  }

else
{
   Serial.println("OBJECT not detected");
   digitalWrite(13, LOW);
  }
delay(1000);
}
```

```
OBJECT detected
OBJECT detected
OBJECT detected
OBJECT not detected
OBJECT not detected
OBJECT detected
OBJECT not detected
OBJECT not detected
OBJECT not detected
OBJECT not detected
OBJECT not detected
OBJECT not detected
OBJECT detected
```

☑ Autoscroll   No line ending   9600 baud

GAS SENSOR

OVERVIEW

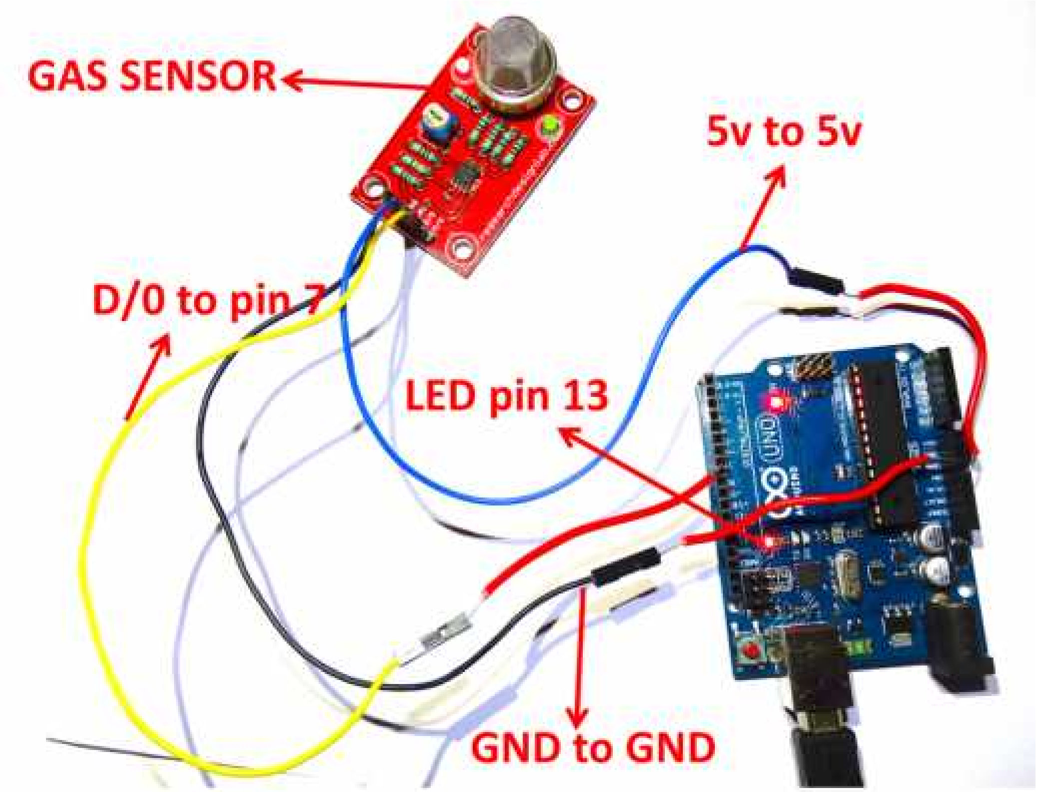
The liquefied Petroleum Gas (LPG) sensor is suitable for sensing LPG (composed of mostly propane and butane) concentration in the air. This can be used in Gas Leakage Detection equipment for detecting the ISO-butane, Propane, LNG combustible Gases. If output goes above the preset range, indication will be shown as high otherwise it will remain in idle condition

OBJECTIVE OF THE EXPERIMENT

If Gas is detected pin 13 will go high (onboard LED ON) and "gas detected" message will be displayed in serial monitor

If Gas is not detected pin 13 will go low (onboard LED OFF) and "gas not detected" message will be displayed in serial monitor

EXPERIMENTAL SETUP



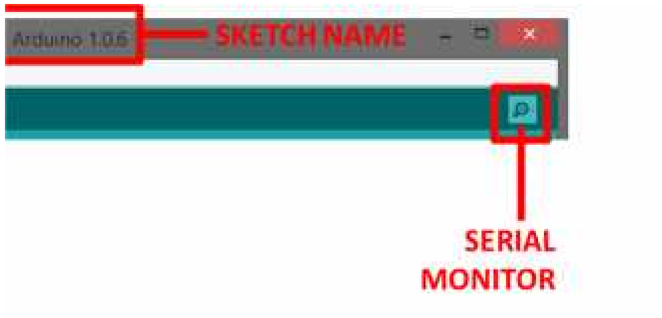
(Note: for testing, Get a cigarette lighter and half press the lighter button to spill out the GAS.)

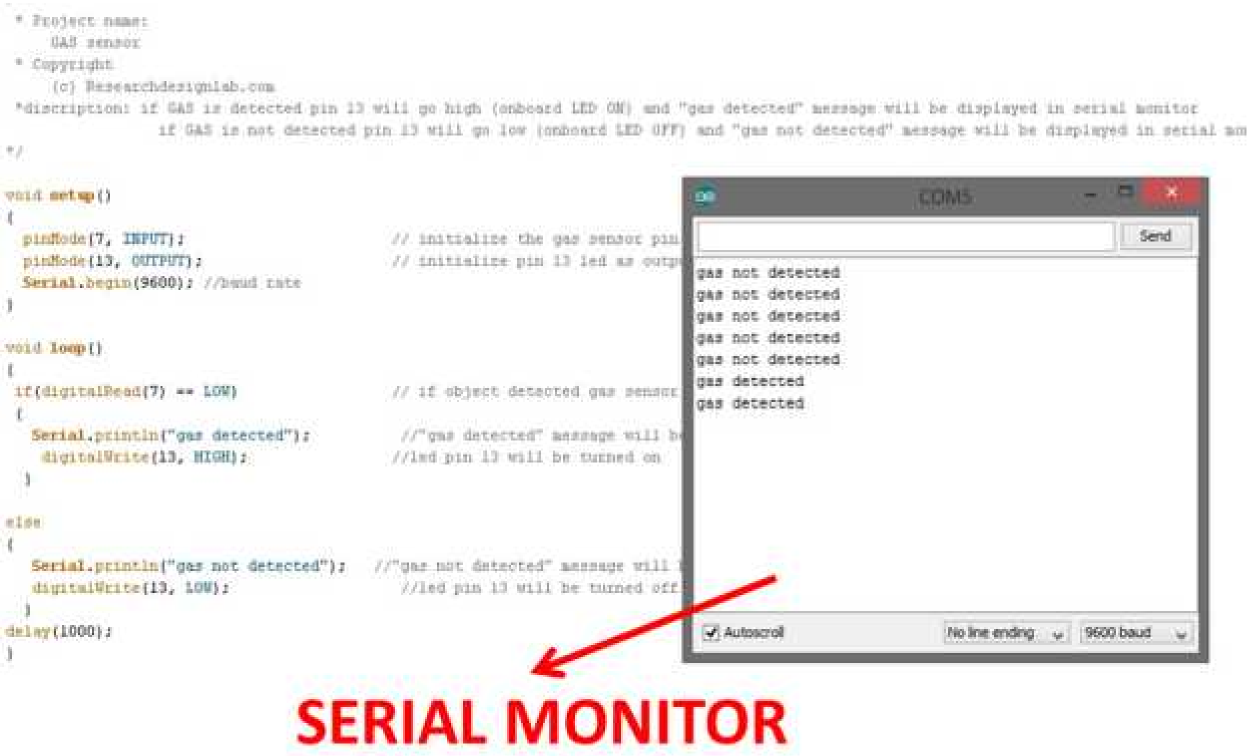
GAS SENSOR ARDUINO CODE

```
/*
* Project name:
    Gas sensor
* Copyright
    (c) Researchdesignlab.com
*description:    if Gas is detected pin 13 will go high (onboard LED ON) and "gas detected" message will
                 be displayed in serial monitor
                 If Gas is not detected pin 13 will go low (onboard LED OFF) and "gas not detected"
                 message will be displayed in serial monitor
*/
void setup()
{
pinMode(7, INPUT);                   // initialize the GAS sensor pin as an input:
pinMode(13, OUTPUT);                 // initialize pin 13 led as output
Serial.begin(9600); //baud rate
}

void loop()
{
if(digitalRead(7) == HIGH)           // if gas detected GAS sensor sends 0 to pin 7
{
Serial.println("gas detected");      //"gas detected" message will be displayed in serial monitor
digitalWrite(13, HIGH);              //led pin 13 will be turned on
}
else
{
Serial.println("gas not detected");  //"gas not detected" message will be displayed in serial monitor
digitalWrite(13, LOW);               //led pin 13 will be turned off
}
delay(1000);                         //delay of one second
}
```

After compiling and uploading the above code, click on serial monitor in Ardunio ide to observe the output.

ARDUINO IDE – SERIAL MONITOR





```
 * Project name:
     GAS sensor
 * Copyright
     (c) Researchdesignlab.com
 *discription: if GAS is detected pin 13 will go high (onboard LED ON) and "gas detected" message will be displayed in serial monitor
                 if GAS is not detected pin 13 will go low (onboard LED OFF) and "gas not detected" message will be displayed in serial mo
*/

void setup()
{
  pinMode(7, INPUT);                          // initialize the gas sensor pin
  pinMode(13, OUTPUT);                        // initialize pin 13 led as outp
  Serial.begin(9600); //baud rate
}

void loop()
{
 if(digitalRead(7) == LOW)                    // if object detected gas sensor
 {
   Serial.println("gas detected");             //"gas detected" message will be
    digitalWrite(13, HIGH);                   //led pin 13 will be turned on
  }

else
{
   Serial.println("gas not detected");      //"gas not detected" message will
   digitalWrite(13, LOW);                      //led pin 13 will be turned off
  }
delay(1000);
}
```

```
gas not detected
gas not detected
gas not detected
gas not detected
gas not detected
gas detected
gas detected
```

FIRE SENSOR

OVERVIEW

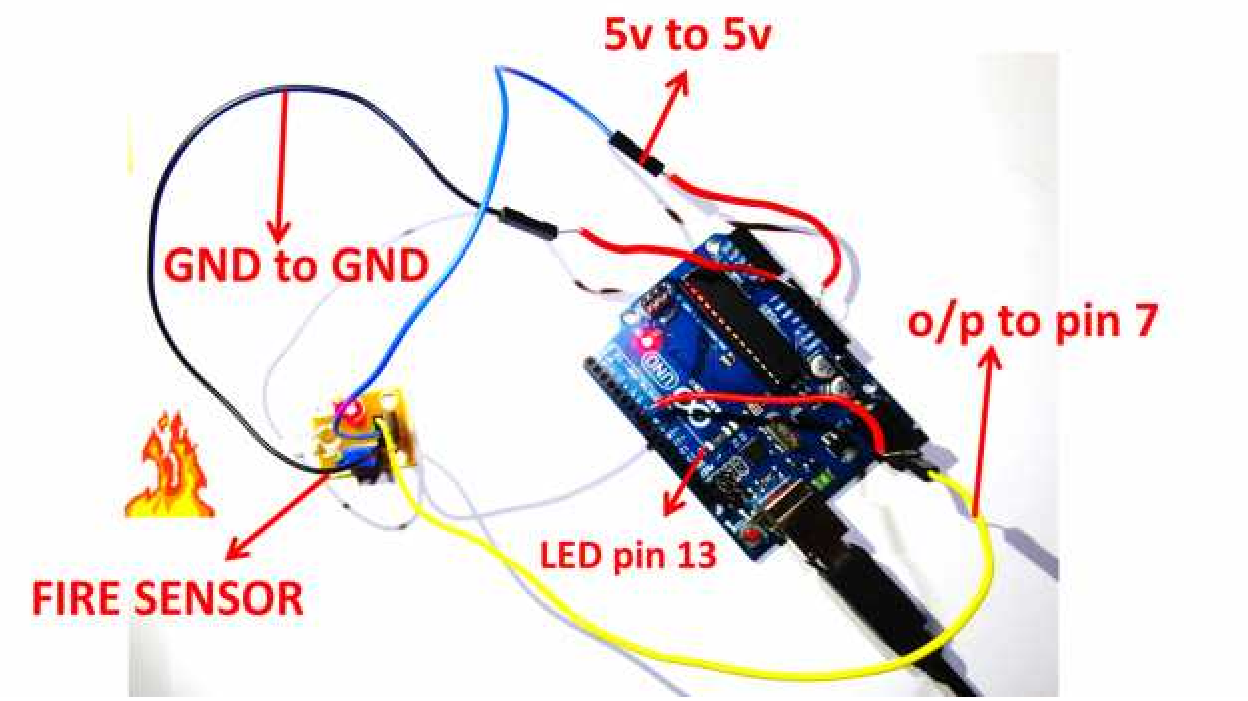
The Fire sensor, as the name suggests, is used as a simple and compact device for protection against fire. The module makes use of IR sensor and comparator to detect fire up to a range of 1 - 2 meters depending on fire density.

OBJECTIVE OF THE EXPERIMENT

If FIRE is detected pin 13 will go high (onboard LED ON) and "FIRE detected" message will be displayed in serial monitor

If FIRE is not detected pin 13 will go low (onboard LED OFF) and "FIRE not detected" message will be displayed in serial monitor

EXPERIMENTAL SETUP

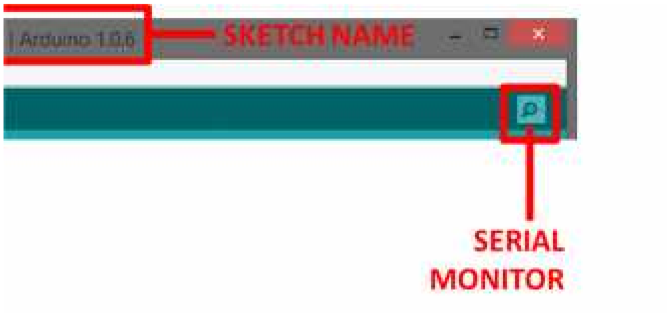


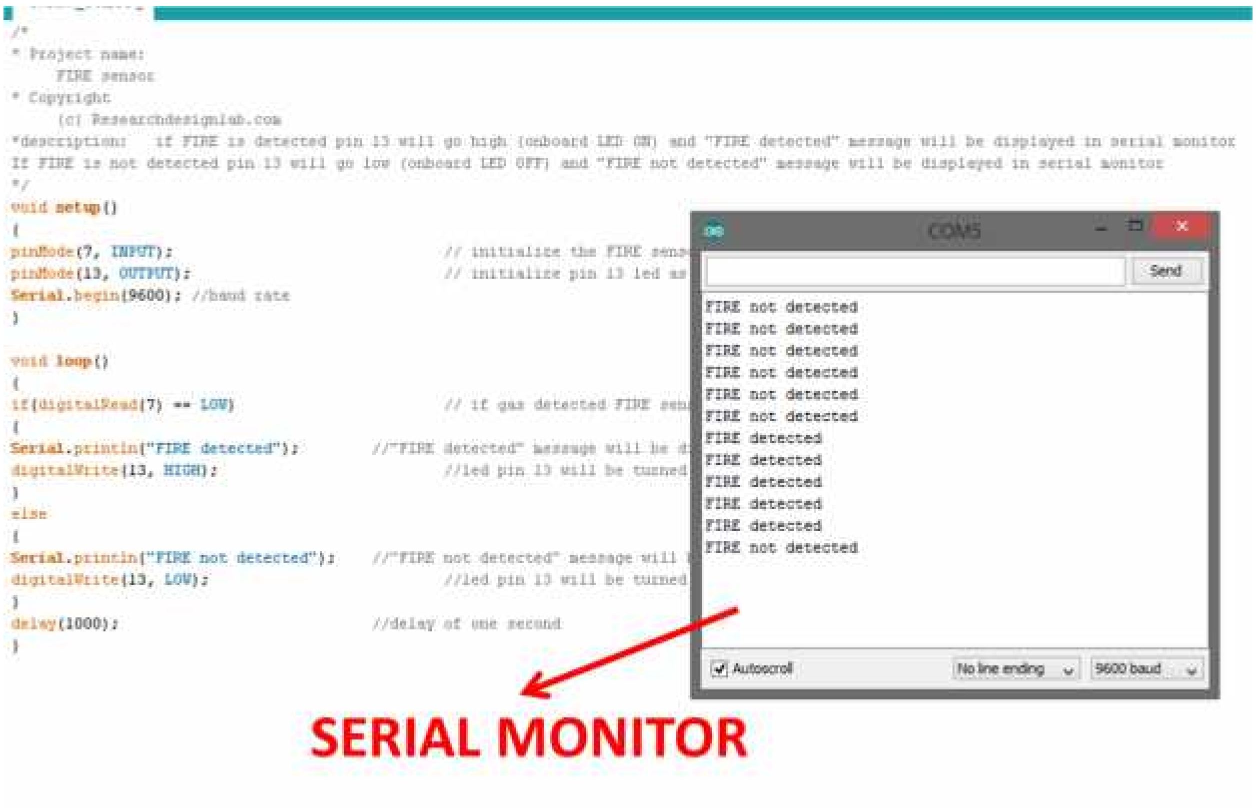
FIRE SENSOR ARDUINO CODE

```
/*
* Project name:
   FIRE sensor
* Copyright
   (c) Researchdesignlab.com
*description:   if FIRE is detected pin 13 will go high (onboard LED ON) and "FIRE detected" message
               will be displayed in serial monitor
               If FIRE is not detected pin 13 will go low (onboard LED OFF) and "FIRE not detected"
               message will be displayed in serial monitor
*/
void setup()
{
pinMode(7, INPUT);                    // initialize the FIRE sensor pin as an input:
pinMode(13, OUTPUT);                  // initialize pin 13 led as output
Serial.begin(9600); //baud rate
}

void loop()
{
if(digitalRead(7) == HIGH)            // if gas detected FIRE sensor sends 0 to pin 7
{
Serial.println("FIRE detected");      //"FIRE detected" message will be displayed in serial monitor
digitalWrite(13, HIGH);               //led pin 13 will be turned on
}
else
{
Serial.println("FIRE not detected");  //"FIRE not detected" message will be displayed in serial
                                      monitor
digitalWrite(13, LOW);                //led pin 13 will be turned off
}
delay(1000);                          //delay of one second
}
```

After compiling and uploading the above code, click on serial monitor in Ardunio ide to observe the output.

ARDUINO IDE – SERIAL MONITOR





```
/*
* Project name:
     FIRE sensor
* Copyright
     (c) Researchdesignlab.com
*description:   if FIRE is detected pin 13 will go high (onboard LED ON) and "FIRE detected" message will be displayed in serial monitor
If FIRE is not detected pin 13 will go low (onboard LED OFF) and "FIRE not detected" message will be displayed in serial monitor
*/
void setup()
{
pinMode(7, INPUT);                          // initialize the FIRE sens
pinMode(13, OUTPUT);                        // initialize pin 13 led as
Serial.begin(9600); //baud rate
}

void loop()
{
if(digitalRead(7) == LOW)                   // if gas detected FIRE sen
{
Serial.println("FIRE detected");        //"FIRE detected" message will be d
digitalWrite(13, HIGH);                     //led pin 13 will be turned
}
else
{
Serial.println("FIRE not detected");    //"FIRE not detected" message will
digitalWrite(13, LOW);                      //led pin 13 will be turned
}
delay(1000);                            //delay of one second
}
```

```
FIRE not detected
FIRE not detected
FIRE not detected
FIRE not detected
FIRE not detected
FIRE not detected
FIRE detected
FIRE detected
FIRE detected
FIRE detected
FIRE detected
FIRE not detected
```

RELAY SHIELD

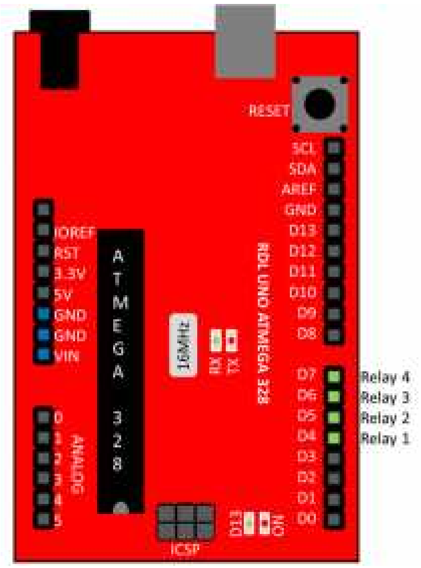
OVERVIEW

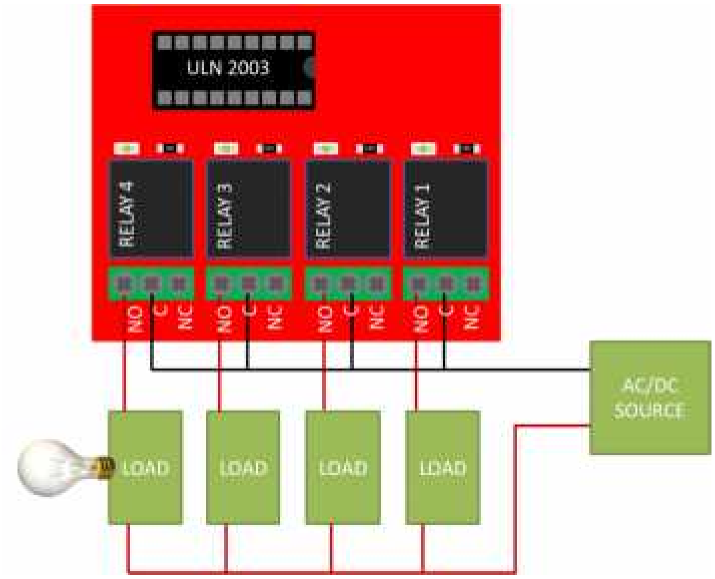
The Relay shield is capable of controlling 4 relays. The max switching power could be 12A/250VAC or 15A/24VDC. It could be directly controlled by Arduino through digital IOs.

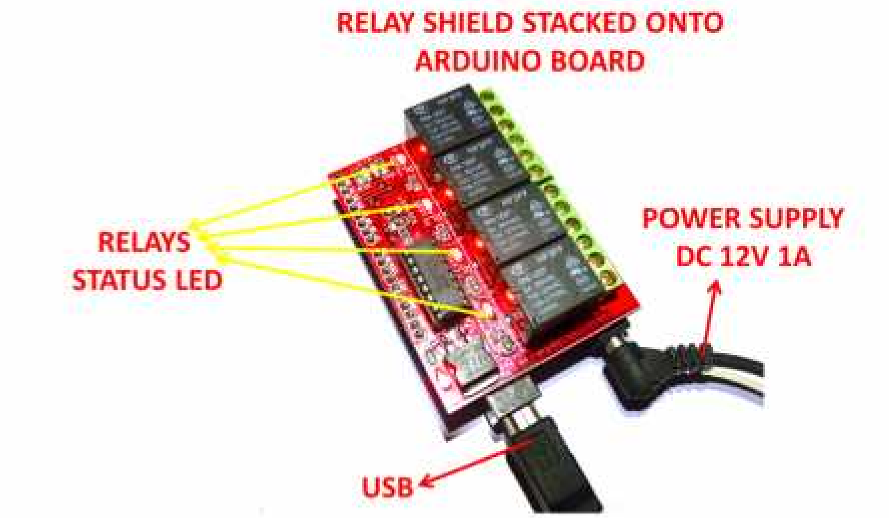
OBJECTIVE OF THE EXPERIMENT

Controlling relay shield from serial monitor (Arduino IDE)

EXPERIMENTAL SETUP





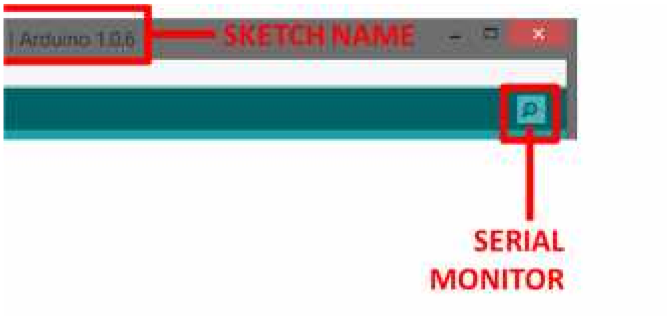


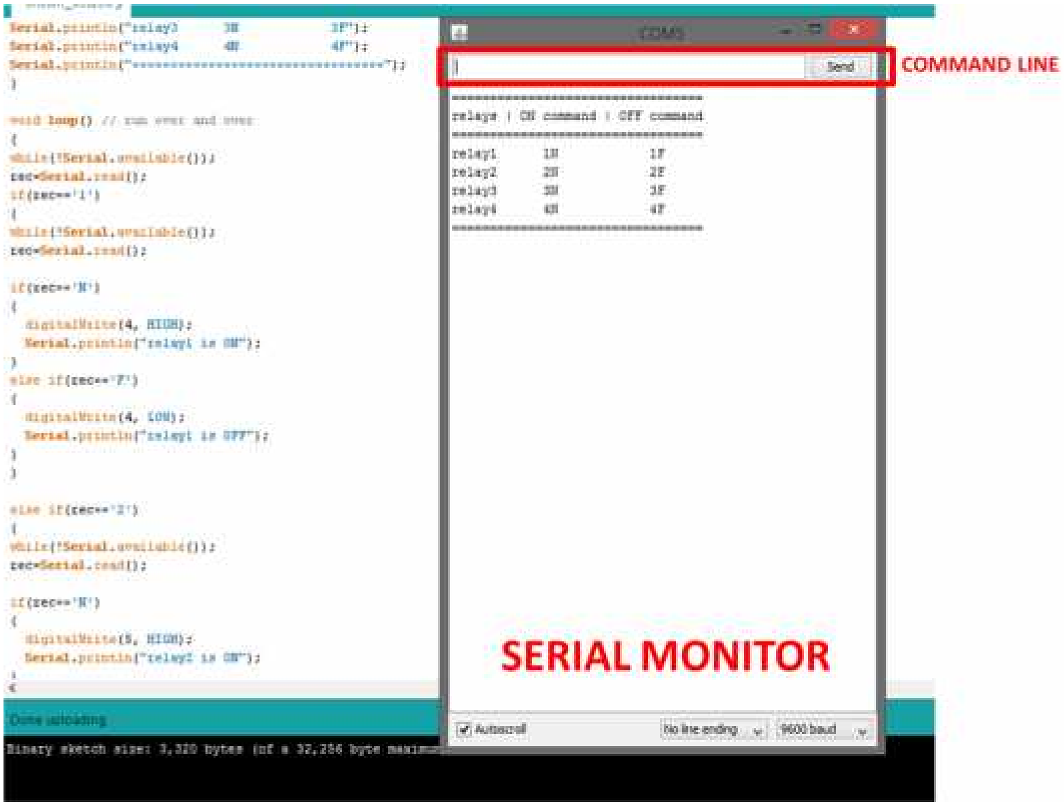
Note: Both USB and DC power supply must be plugged in.

RELAY SHIELD ARDUINO CODE

```
/*
* Project name:
    Relay shield - arduino
* Copyright
    (c) Researchdesignlab.com
*description:
      Controlling relay shield from serial monitor
*/

char rec;
void setup()
{
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
Serial.begin(9600);
delay(1000);

Serial.println("=================================");
Serial.println("relays | ON command | OFF command");
Serial.println("=================================");
Serial.println("relay1     1N          1F");
Serial.println("relay2     2N          2F");
Serial.println("relay3     3N          3F");
Serial.println("relay4     4N          4F");
Serial.println("=================================");
}

void loop() // run over and over
{
while(!Serial.available());
rec=Serial.read();
if(rec=='1')
{
while(!Serial.available());
rec=Serial.read();

if(rec=='N')
{
  digitalWrite(4, HIGH);
  Serial.println("relay1 is ON");
}
else if(rec=='F')
{
  digitalWrite(4, LOW);
  Serial.println("relay1 is OFF");
}
}

else if(rec=='2')
{
while(!Serial.available());
rec=Serial.read();

if(rec=='N')
{
  digitalWrite(5, HIGH);
  Serial.println("relay2 is ON");
}
else if(rec=='F')
{
  digitalWrite(5, LOW);
  Serial.println("relay2 is OFF");
}
}

else if(rec=='3')
{
while(!Serial.available());
rec=Serial.read();

if(rec=='N')
{
  digitalWrite(6, HIGH);
  Serial.println("relay3 is ON");
}
else if(rec=='F')
{
  digitalWrite(6, LOW);
  Serial.println("relay3 is OFF");
}
}

else if(rec=='4')
{
while(!Serial.available());
rec=Serial.read();

if(rec=='N')
{
  digitalWrite(7, HIGH);
  Serial.println("relay4 is ON");
}
else if(rec=='F')
{
  digitalWrite(7, LOW);
  Serial.println("relay4 is OFF");
}
}

}
```

ARDUINO IDE – SERIAL MONITOR

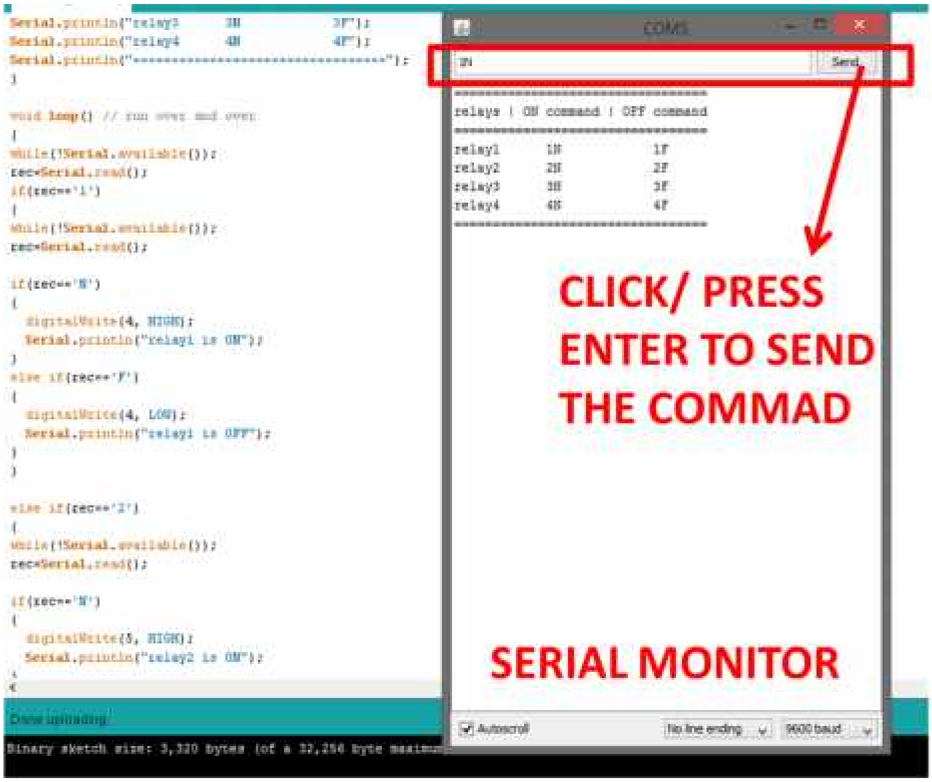




```
Serial.println("relay3       3N             3F");
Serial.println("relay4       4N             4F");
Serial.println("===================================");
}

void loop() // run over and over
{
while(!Serial.available());
rec=Serial.read();
if(rec=='1')
{
while(!Serial.available());
rec=Serial.read();

if(rec=='N')
{
  digitalWrite(4, HIGH);
  Serial.println("relay1 is ON");
}
else if(rec=='F')
{
  digitalWrite(4, LOW);
  Serial.println("relay1 is OFF");
}
}

else if(rec=='2')
{
while(!Serial.available());
rec=Serial.read();

if(rec=='N')
{
  digitalWrite(5, HIGH);
  Serial.println("relay2 is ON");
}
```

```
===================================
relays | ON command | OFF command
===================================
relay1       1N             1F
relay2       2N             2F
relay3       3N             3F
relay4       4N             4F
===================================
```

```
Binary sketch size: 3,320 bytes (of a 32,256 byte maxim
```



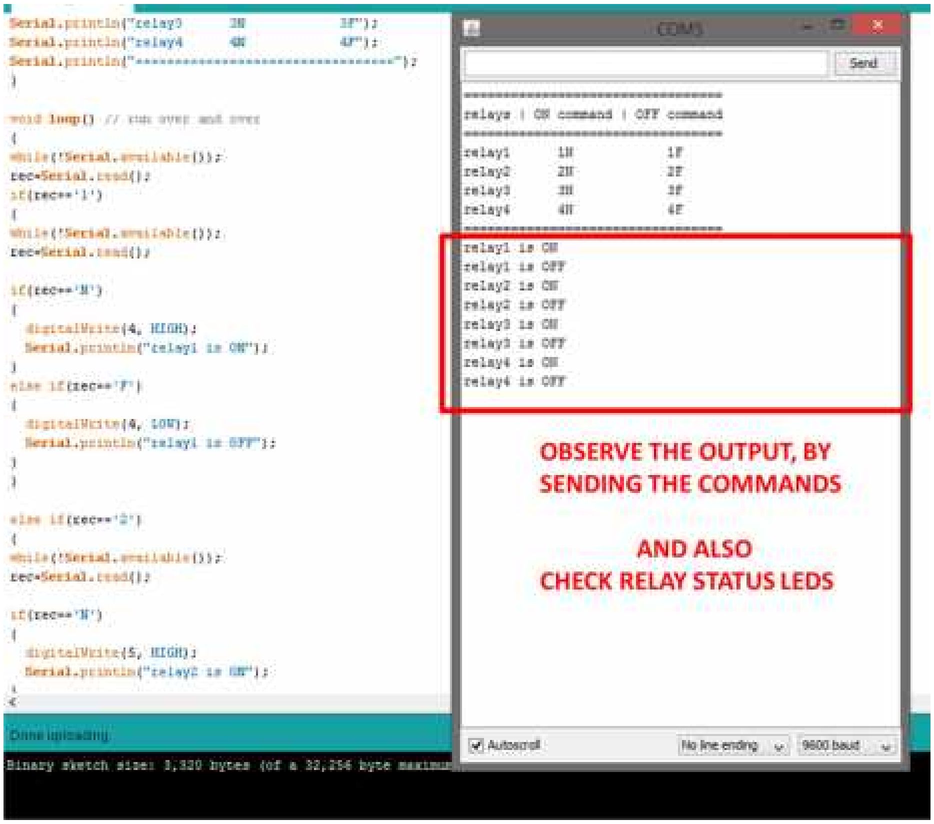
```
Serial.println("relay3      3N          3F");
Serial.println("relay4      4N          4F");
Serial.println("==================================");
}

void loop() // run over and over
{
while(!Serial.available());
rec=Serial.read();
if(rec=='1')
{
while(!Serial.available());
rec=Serial.read();

if(rec=='N')
{
  digitalWrite(4, HIGH);
  Serial.println("relay1 is ON");
}
else if(rec=='F')
{
  digitalWrite(4, LOW);
  Serial.println("relay1 is OFF");
}
}

else if(rec=='2')
{
while(!Serial.available());
rec=Serial.read();

if(rec=='N')
{
  digitalWrite(5, HIGH);
  Serial.println("relay2 is ON");
```

```
Done uploading.
Binary sketch size: 3,320 bytes (of a 32,256 byte maximum
```

```
1N

==================================
relays | ON command | OFF command
==================================
relay1      1N          1F
relay2      2N          2F
relay3      3N          3F
relay4      4N          4F
==================================
```

CLICK/ PRESS ENTER TO SEND THE COMMAD

SERIAL MONITOR



```
Serial.println("relay3      3N          3F");
Serial.println("relay4      4N          4F");
Serial.println("==================================");
}

void loop() // run over and over
{
while(!Serial.available());
rec=Serial.read();
if(rec=='1')
{
while(!Serial.available());
rec=Serial.read();

if(rec=='N')
{
  digitalWrite(4, HIGH);
  Serial.println("relay1 is ON");
}
else if(rec=='F')
{
  digitalWrite(4, LOW);
  Serial.println("relay1 is OFF");
}
}

else if(rec=='2')
{
while(!Serial.available());
rec=Serial.read();

if(rec=='N')
{
  digitalWrite(5, HIGH);
  Serial.println("relay2 is ON");
```

```
Done uploading.
Binary sketch size: 3,320 bytes (of a 32,256 byte maximum
```

```
==================================
relays | ON command | OFF command
==================================
relay1      1N          1F
relay2      2N          2F
relay3      3N          3F
relay4      4N          4F
==================================
relay1 is ON
relay1 is OFF
relay2 is ON
relay2 is OFF
relay3 is ON
relay3 is OFF
relay4 is ON
relay4 is OFF
```

OBSERVE THE OUTPUT, BY SENDING THE COMMANDS

AND ALSO CHECK RELAY STATUS LEDS

GSM SHIELD

OVERVIEW

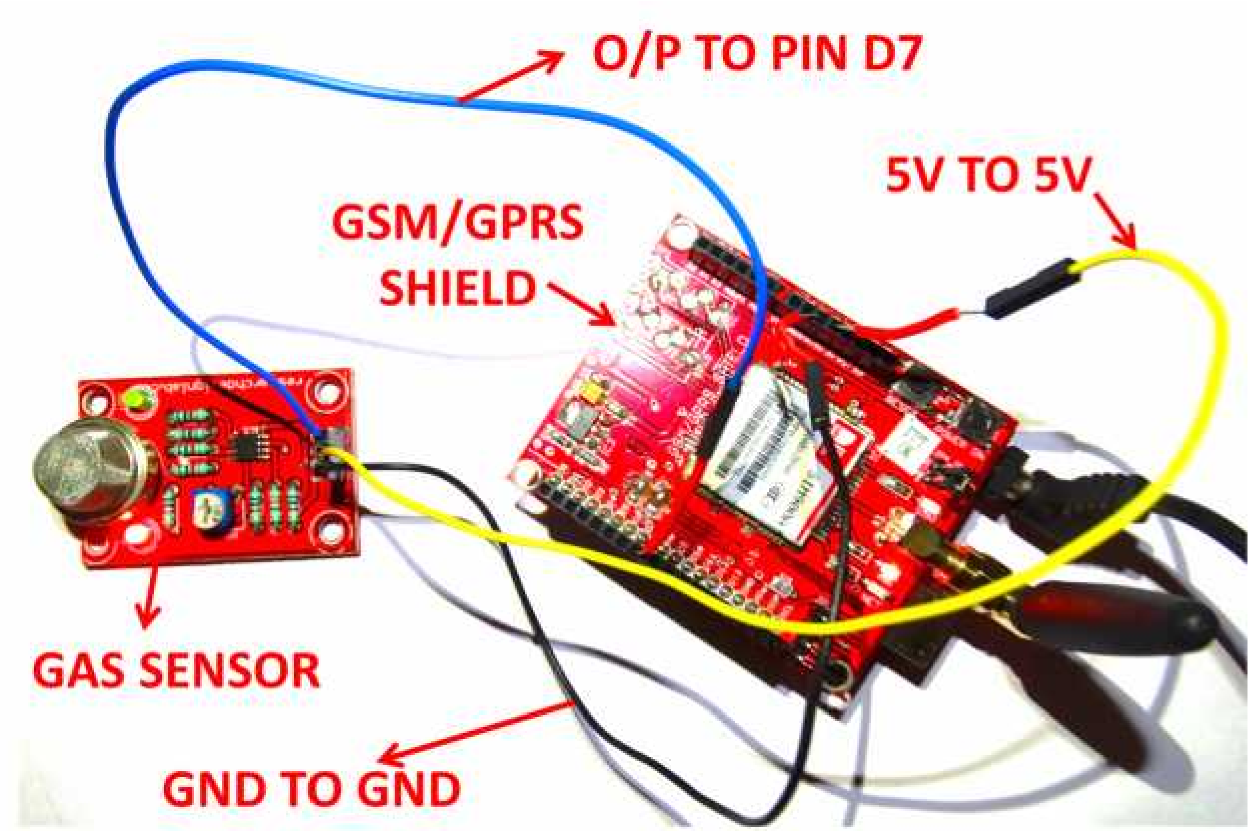
This is a very low cost and simple Arduino GSM and GPRS shield. We use the module SIMCom SIM900A.

The Shield connects your Arduino to the internet using the GPRS wireless network. Just plug this module onto your Arduino board, plug in a SIM card from an operator offering GPRS coverage and follow a few simple instructions to start controlling your world through the internet. You can also make/receive voice calls (you will need an external speaker and microphone circuit) and send/receive SMS messages

OBJECTIVE OF THE EXPERIMENT

If GAS is detected pin 7 will go LOW and "GAS detected" message will be sent to destination number.

EXPERIMENTAL SETUP

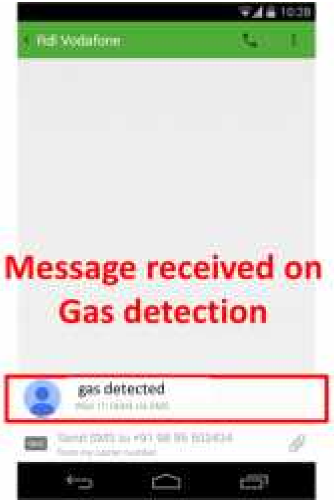


(Note: for testing, Get a cigarette lighter and half press the lighter button to spill out GAS.)

GSM SHIELD

```
/*
* Project name:
    GSM Shield
* Copyright
    (c) Researchdesignlab.com
*description:   If GAS is detected pin 7 will go LOW and "GAS detected" message will be sent to
                destination number
*/
void setup()
{
Serial.begin(9600);                         // SERIAL COMMUNICATION BAUD RATE
pinMode(7, INPUT);                          //INITIALIZE PIN 7 FOR GAS SENSOR OUTPUT
delay(5000);
}
void loop()
{
if(digitalRead(7)== LOW)
{
Serial.println("AT");                               //TO CHECK MODEM
delay(1000);
Serial.println("AT+CMGF=1");                        //TO CHANGE MESSAGE SENDING MODE
delay(1000);
Serial.println("AT+CMGS=\"0123456789\"");           //CHANGE TO DESTINATION NUMBER
delay(1000);
Serial.print("Gas detected");                       //MESSAGE WILL SENT ONCE GAS IS DETECTED
Serial.write(26);
delay(1000);
}
}
```

Compile and upload the above code to arduino, then mount the GSM Shield onto arduino board (place jumper on JP3) and plugin power supply DC 12V 1A.(remove USB cable).



Message received on Gas detection

gas detected

BLUETOOTH RELAY SHIELD

OVERVIEW

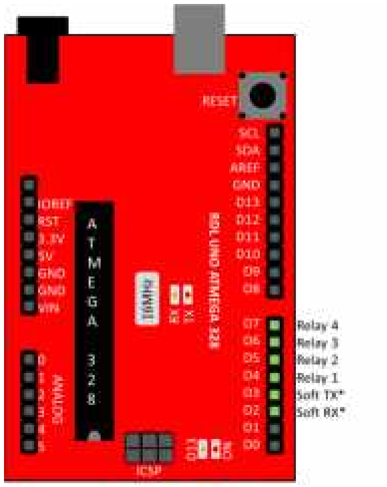
Bluetooth technology is a short distance communication technology used by almost all phones including smart phones and all laptops. This technology find very wide uses including that of Home & Industrial automation.

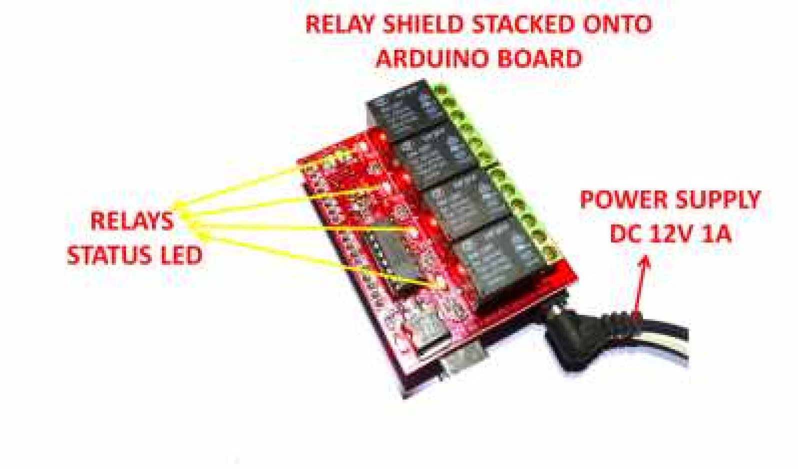
The Relay shield is capable of controlling 4 relays. The max switching power could be 12A/250VAC or 15A/24VDC. It could be directly controlled by Arduino through digital IOs.

OBJECTIVE OF THE EXPERIMENT

Controlling relay shield from Bluetooth enabled device (Android APK)

EXPERIMENTAL SETUP



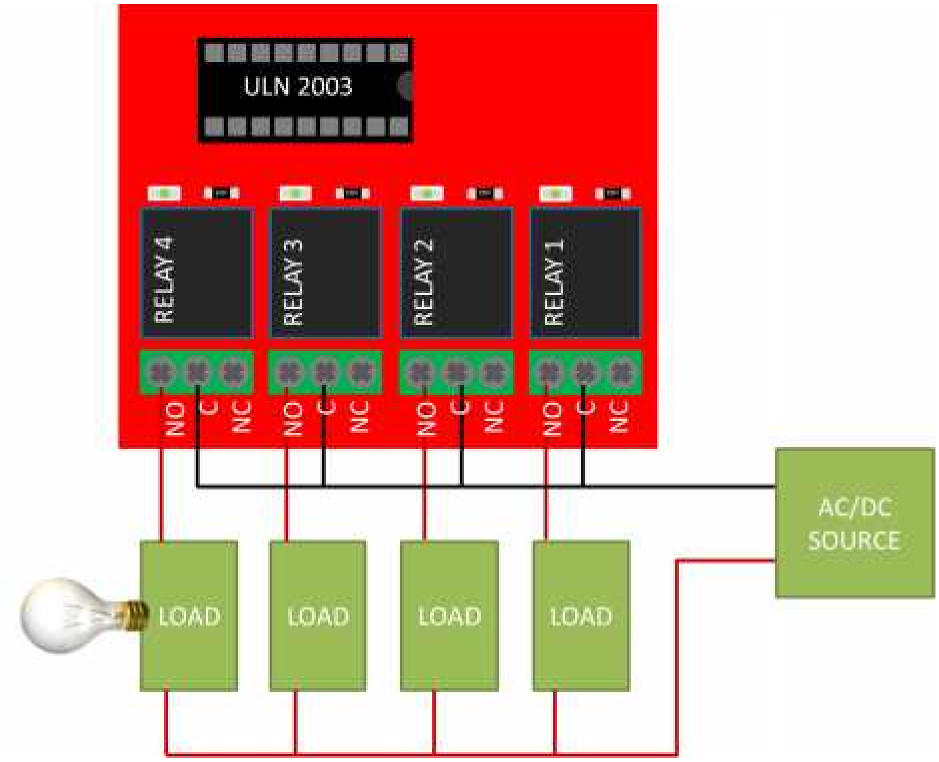


Note: remove USB after uploading the code, DC 12V 1A must be plugged in.

RELAY SHIELD ARDUINO CODE

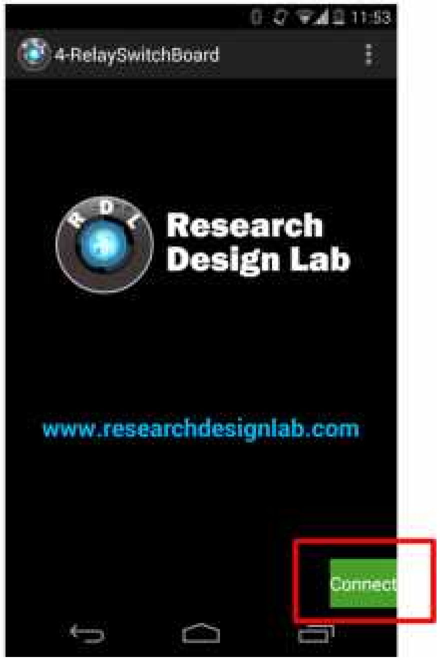
```
/*
Software serial multple serial test

Receives from the hardware serial, sends to software serial.
Receives from software serial, sends to hardware serial.

The circuit:
* RX is digital pin 10 (connect to TX of other device)
* TX is digital pin 11 (connect to RX of other device)

Note:
Not all pins on the Mega and Mega 2560 support change interrupts,
so only the following can be used for RX:
10, 11, 12, 13, 50, 51, 52, 53, 62, 63, 64, 65, 66, 67, 68, 69

Not all pins on the Leonardo support change interrupts,
so only the following can be used for RX:
8, 9, 10, 11, 14 (MISO), 15 (SCK), 16 (MOSI).

 Software serial multple serial test

Receives from the hardware serial, sends to software serial.
Receives from software serial, sends to hardware serial.

The circuit:
* RX is digital pin 2 (connect to TX of other device)
* TX is digital pin 3 (connect to RX of other device)

SENDING DATA FORMAT

1N TO ON RELAY1
1F TO OFF RELAY1

2N TO ON RELAY2
2F TO OFF RELAY2

3N TO ON RELAY3
3F TO OFF RELAY3

4N TO ON RELAY4
4F TO OFF RELAY4

This example code is in the public domain.

*/
```

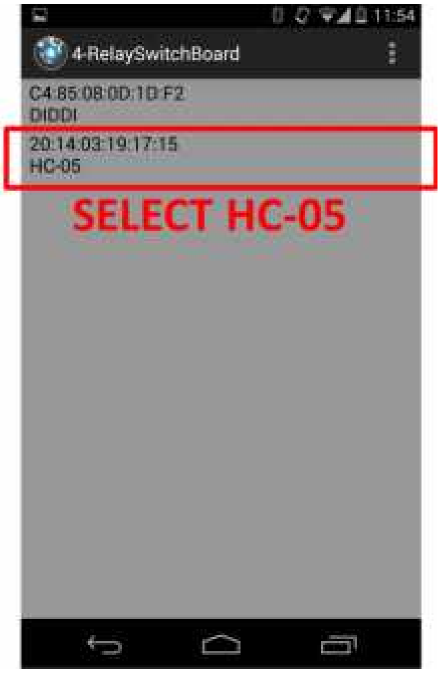
```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); // RX, TX
int rec;
void setup()
{
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
mySerial.begin(9600);

}

void loop() // run over and over
{
while(!mySerial.available());
rec=mySerial.read();


if(rec=='1')
{
while(!mySerial.available());
rec=mySerial.read();

if(rec=='N')
digitalWrite(4, HIGH);
else if(rec=='F')
digitalWrite(4, LOW);
}

else if(rec=='2')
{
while(!mySerial.available());
rec=mySerial.read();
if(rec=='N')
digitalWrite(5, HIGH);
else if(rec=='F')
digitalWrite(5, LOW);
}
else if(rec=='3')
{
while(!mySerial.available());
rec=mySerial.read();
if(rec=='N')
digitalWrite(6, HIGH);
else if(rec=='F')
digitalWrite(6, LOW);

}

else if(rec=='4')
{
while(!mySerial.available());
rec=mySerial.read();
if(rec=='N')
digitalWrite(7, HIGH);
else if(rec=='F')
digitalWrite(7, LOW);
}
}
```



4-RELAY SWITCH BOARD ANDROID APPLICATION

TURN ON BLUETOOTH

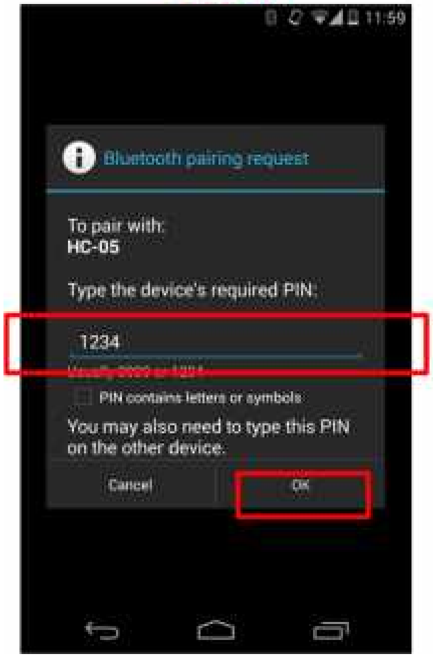




C4:85:08:0D:1D:F2
DIDDI
20:14:03:19:17:15
HC-05

SELECT HC-05

ENTER PASSWORD "1234"



Bluetooth pairing request

To pair with:
HC-05

Type the device's required PIN:

1234

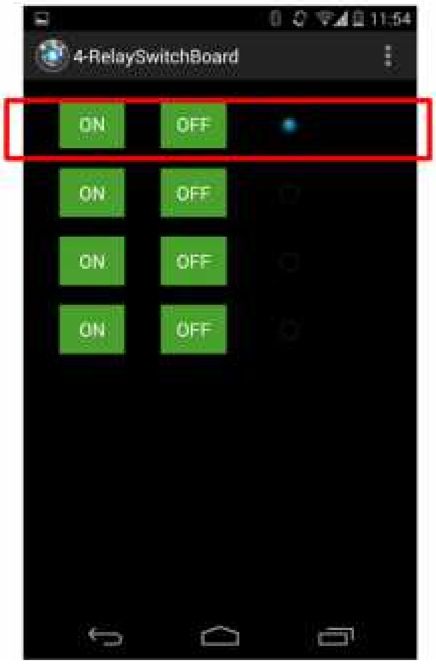
Usually 0000 or 1234

□ PIN contains letters or symbols

You may also need to type this PIN on the other device.

Cancel | OK

TURN ON /OFF RELAYS



LCD-SCREW SHIELD

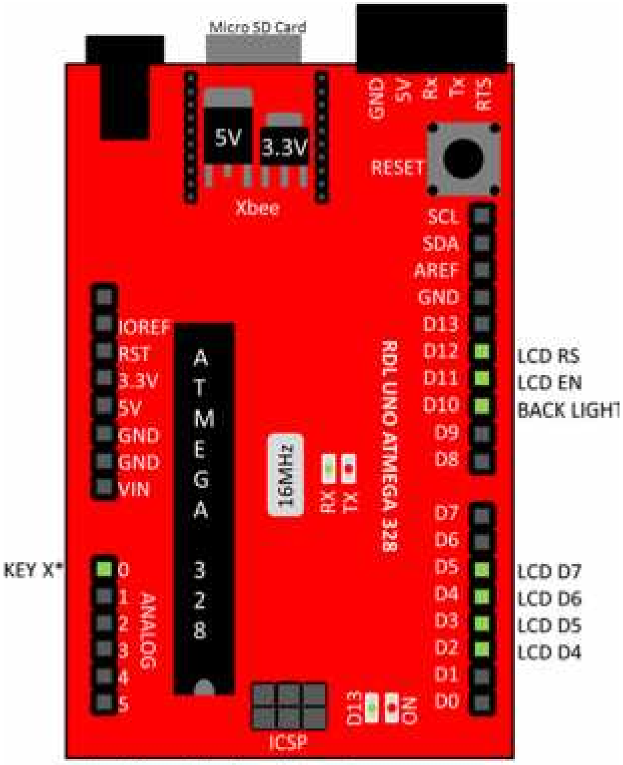
OVERVIEW

One of the basic interfacing requirements for the hobbyists or electronics enthusiasts is I/P (keypad) and O/P (LCD display) for prototype applications. This shield uses minimum number I/O's that is 2 bits(D0 and D1) for LCD data . 5 input key switches (Navigation keys), when it's activated serial data will be sent to pin D0 by internal 2 line LCD controller. Each key has been pulled up to a different voltage level, so a different voltage will be generated every time a user selects a key. This voltage could be read by the analog pin of internal 2 line LCD controller on the board. Hence saves the no of I/O pins. The backlight of the LCD could be controlled by setting PWM (Pin D10) by adding a few lines of code.

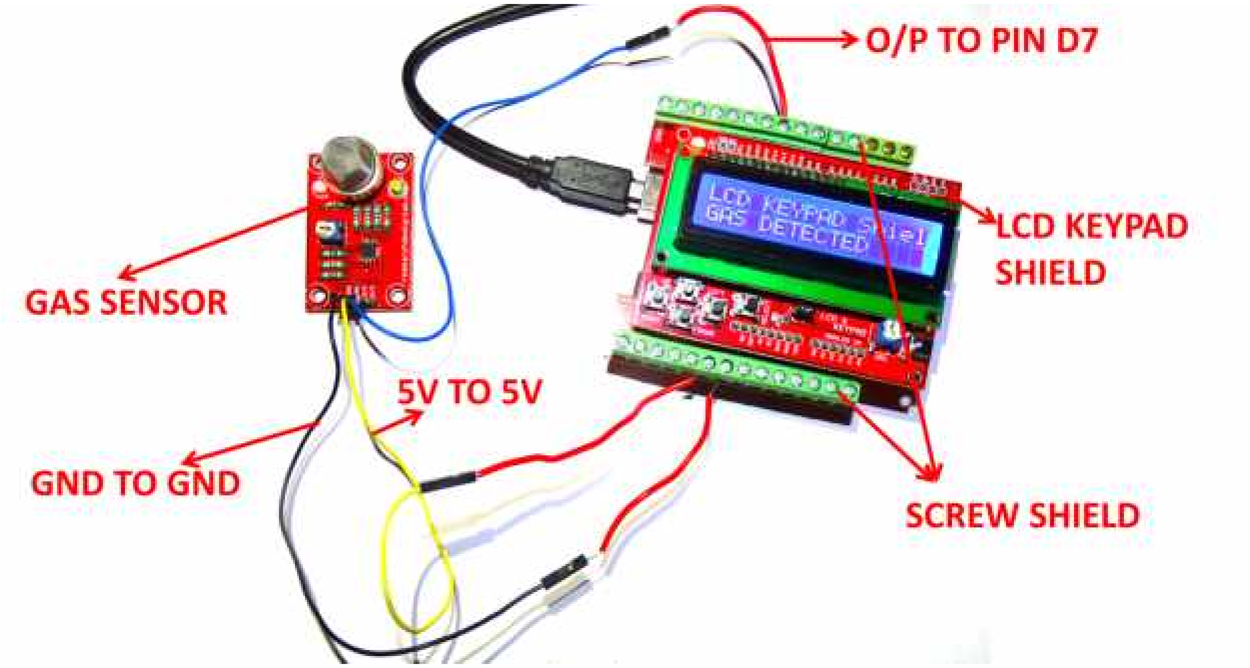
OBJECTIVE OF THE EXPERIMENT

If Gas is detected by sensor ,"gas detected" message will be displayed in LCD else "gas not detected" message will be displayed in LCD.

EXPERIMENTAL SETUP

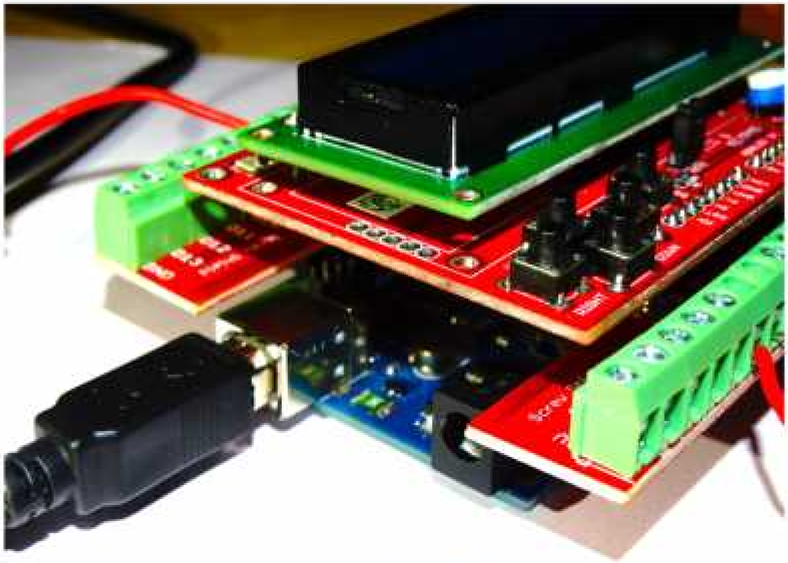


*Key X is enabled by placing a jumper on 'KEYS' tag mentioned on the PCB



(Note: for testing, Get a cigarette lighter and half press the lighter button to spill out the GAS.)

ARDUINO LCD SCREW SHIELD STACKED



LCD AND KEYPAD-SCREW SHIELD ARDUINO CODE

```
/*
* Project name:
LCD KEYPAD Shield
* Copyright
(c) Researchdesignlab.com
* Description:
If Gas is detected by sensor ,”gas detected” message will be displayed in LCD
Else ”gas not detected” message will be displayed in LCD.

The circuit:
* LCD RS pin to digital pin 12
* LCD Enable pin to digital pin 11
* 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
* 10K resistor:
* ends to +5V and ground
* wiper to LCD VO pin (pin 3)
*/
#include <LiquidCrystal.h>          // include the library code:
int sensorValue = 0;                // value read from the keypad
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  // initialize the library with the numbers of the interface pins
int sensorPin = A0;

void setup()
{
lcd.begin(16, 2);                   // set up the LCD's number of columns and rows:
pinMode(7, INPUT );
delay(2000);
}

void loop() {
lcd.clear();                        // clear lcd display
lcd.setCursor(0, 0);                // set the cursor to column 0, line 0
lcd.print("LCD KEYPAD Shield");
lcd.setCursor(0, 1);                // set the cursor to column 0, line 1

if(digitalRead(7) == HIGH)
lcd.print("GAS DETECTED");
else
lcd.print("GAS NOT DETECTED");
delay(500);
}
```

HEART BEAT SENSOR

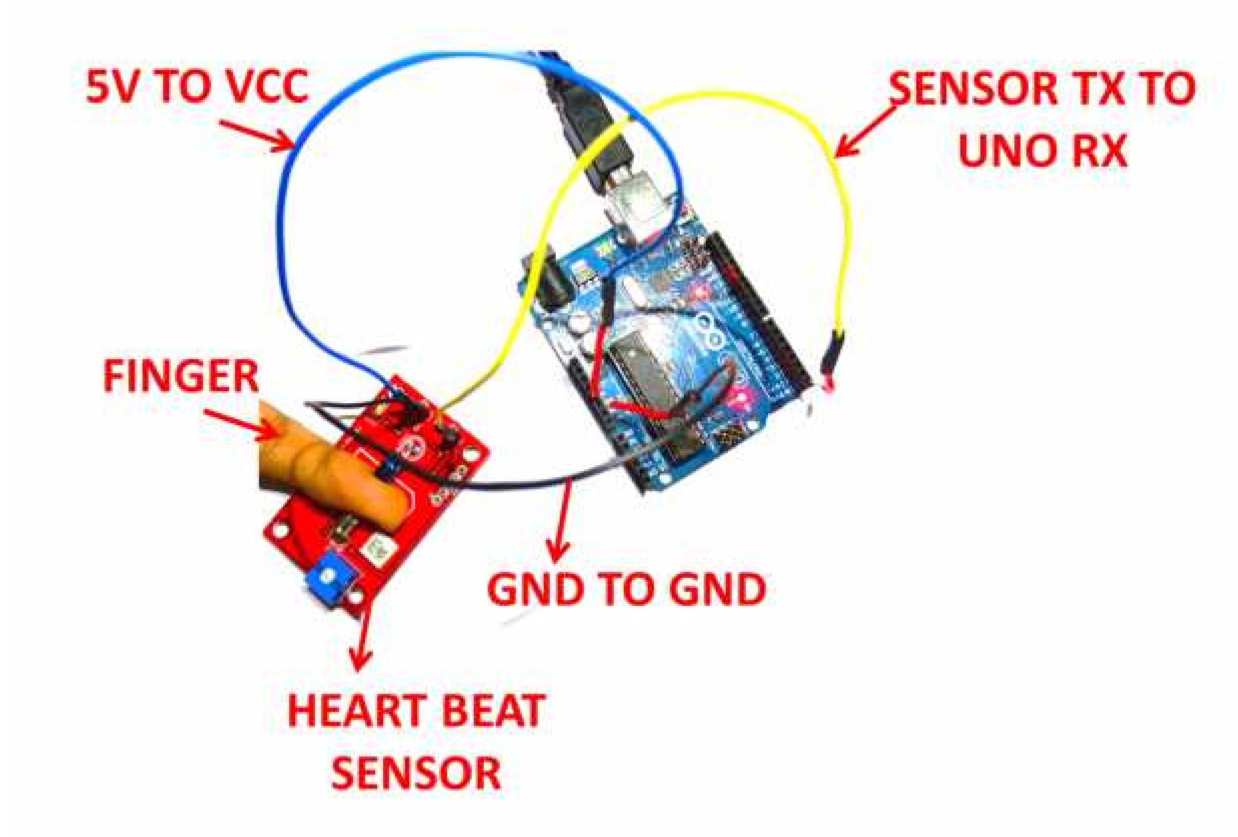
OVERVIEW

The Heart Beat Sensor is designed to provide digital output of heart beat when a finger is placed on it. When the Heart detector starts working, the top most LED will starts flashing with every heartbeat. The output of this sensor can be connected to Micro Controller directly to measure the heart beat per minute (BPM) rate. It functions on the principle of light modulation by blood flow through the nerves of the finger at every pulse. The module output mode, Digital output mode is simple, Serial Output is with exact readings

OBJECTIVE

To measure heart beats per minute.

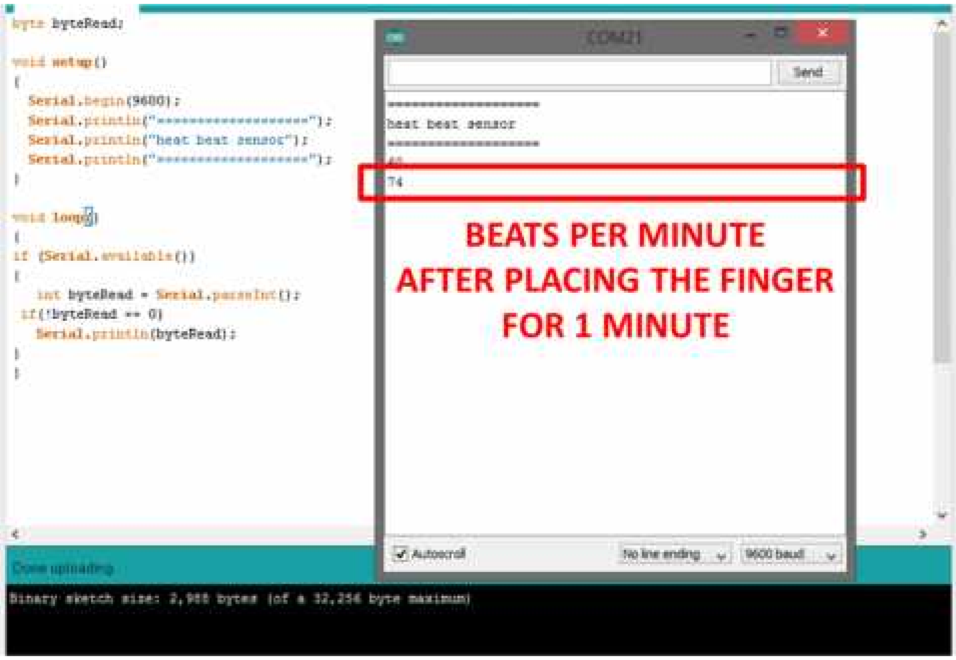
EXPERIMENTAL SETUP



HEART BEAT SENSOR ARDUINO CODE

```
byte byteRead;

void setup()
{
  Serial.begin(9600);                              //BAUD RATE
  Serial.println("==================");
  Serial.println("heat beat sensor");
  Serial.println("==================");
}

void loop()
{
if (Serial.available())
{
 int byteRead = Serial.parseInt();              // CONVERTING ASCII TO INT
 if(!byteRead == 0)
 Serial.println(byteRead);
}
}
```

ARDUINO IDE-SERIAL MONTOR



```
byte byteRead;

void setup()
{
  Serial.begin(9600);
  Serial.println("==================");
  Serial.println("heat beat sensor");
  Serial.println("==================");
}

void loop()
{
if (Serial.available())
{
  int byteRead = Serial.parseInt();
 if(!byteRead == 0)
  Serial.println(byteRead);
}
}
```

```
==================
heat beat sensor
==================
74
```

BEATS PER MINUTE
AFTER PLACING THE FINGER
FOR 1 MINUTE

Done uploading.

```
Binary sketch size: 2,988 bytes (of a 32,256 byte maximum)
```