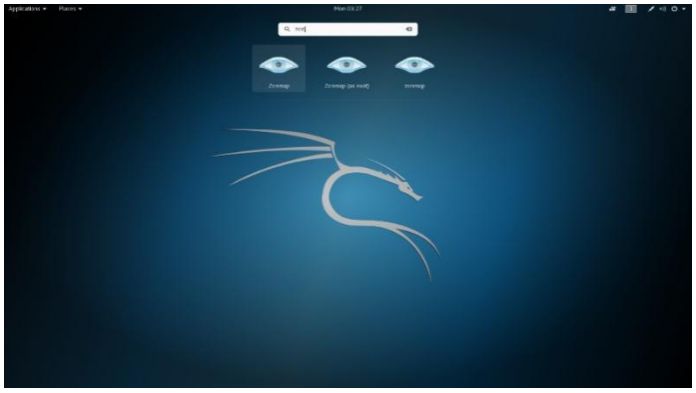
LAB MANUAL -II

ON

(INTERNET OF THINGS)



ESTABLISHMENT OF ADVANCED LABORATORY FOR CYBER SECURITY TRAINING TO TECHNICAL TEACHERS

DEPARTMENT OF INFORMATION MANAGEMENT AND EMERGING ENGINEERING

MINISTRY OF ELECTRONICS AND INFORMATION TECHNOLOGY GOVERNMENT OF INDIA

Principal Investigator: Prof. Maitreyee Dutta

PREPARED BY: Prof. Maitreyee Dutta and Ms. Nitika Khurana (MSA)

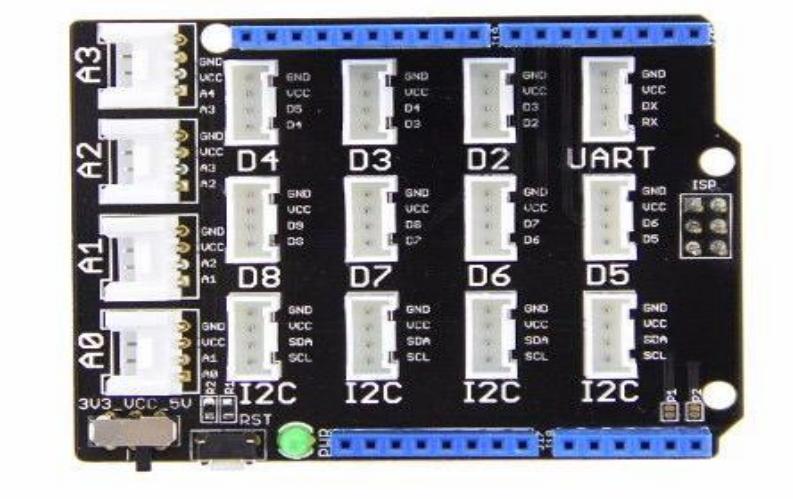
Table of Contents

Practical 11

Study of the Base Shield

Introduction:

Arduino Uno is the most popular Arduino board so far, however it is sometimes frustrating when your project requires a lot of sensors or Leds and your jumper wires are in a mess. The purpose of creating the Base Shield is to help you get rid of bread board and jumper wires. With the rich grove connectors on the base board, you can add all the grove modules to the Arduino Uno conveniently! The pinout of Base Shield V2 is the same as Arduino Uno R3.



Specifications:

Parameter	Value/Range
Operating voltage	3.3⁄5V
Operation Temperature	-25℃ to +85℃
Analog Ports	4

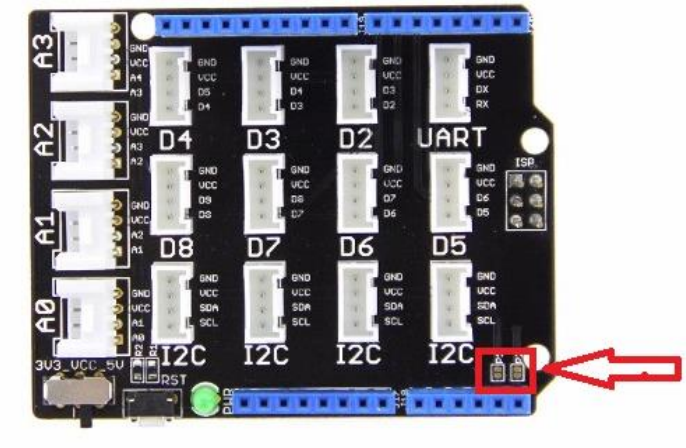
Parameter	Value/Range
Digital Ports	7
UART Ports	1
I2C Ports	1
Size	69mm x53mm

Features:

There are 16 on-board Grove connectors including 4 x Analog, 7 x Digital, 1 x UART, 4 x I2C. Apart from the rich Grove connectors, on the board you can also see an RST button, a green LED to indicating power status, ICSP pin, a toggle switch and four rows of pin-outs.

Power Compatible:

Every Grove connector has four wires, one of which is the VCC. However, not every micro-controller main board needs a supply voltage of 5V, some boards only need 3.3V. That's why we add a power toggle switch to Base Shield V2 so that you can select the suitable voltage of the micro-controller main board you are using via this switch.



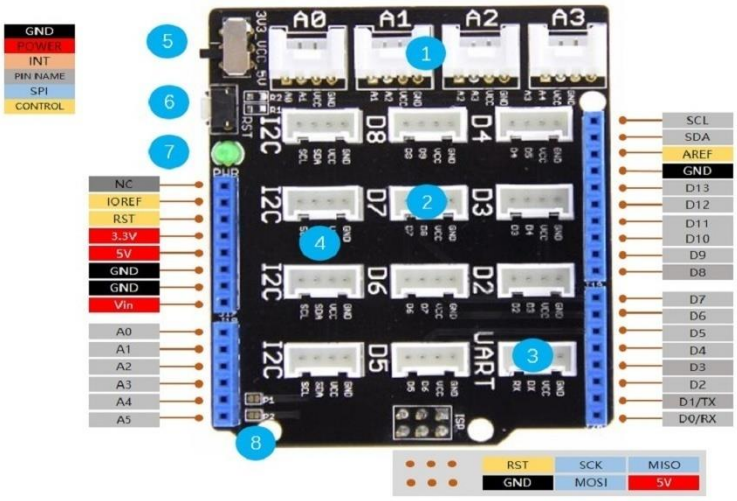
Board Compatible:

The pin-out of Base Shield V2 is the same as Arduino Uno R3, however Arduino Uno is not the board one that the Base Shield V2 is compatible with, here we listed the boards that we have confirmed that can be used with Base Shield V2.

The Base Shield is tested and fully compatible with following boards:

1. Arduino Uno(all revisions)/Seeeduino(V4&V4.2)
2. Arduino Mega/Seeeduino Mega
3. Arduino Zero(M0)/Seeeduino Lorawan
4. Arduino Leonardo/Seeeduino Lite
5. Arduino 101
6. Arduino Due 3.3V
7. Intel Edison 5V
8. Linkit One

Hardware Overview:



1. Analog Ports: include 4 analog ports, A0, A1, A2 and A3.
2. Digital Ports: include 7 digital ports, D2, D3, D4, D5, D6, D7 and D8.
3. UART Port: 1 UART port.
4. I2C Ports: 4 I2C ports.
5. Power Switch: when using Arduino UNO with Base Shield v2, please turn the switch to 5v position; while using Arduino Arch with Base Shield v2, please turn the switch to 3.3v.
6. Reset Button: Reset the Arduino board.
7. PWR LED: The Green LED turns on when power on.
8. Dimension: 2.1 * 2.7 inch

Practical 12

Interfacing the Rotary Angle sensor (Rotary Potentiometer) with the Arduino to control the brightness of LED.

Introduction:

A rotary encoder is a type of position sensor that converts the angular position (rotation) of a knob into an output signal that is used to determine what direction the knob is being rotated. There are two types of rotary encoder – absolute and incremental. The absolute encoder gives us the exact position of the knob in degrees while the incremental encoder reports how many increments the shaft has moved. The rotary angle sensor produces analog output between 0 and Vcc on its D1 connector. The D2 connector is not used. The angular range is 300 degrees with a linear change in value. The resistance value is 10k ohms, perfect for Arduino use. This may also be known as a "potentiometer"LED Specifications.

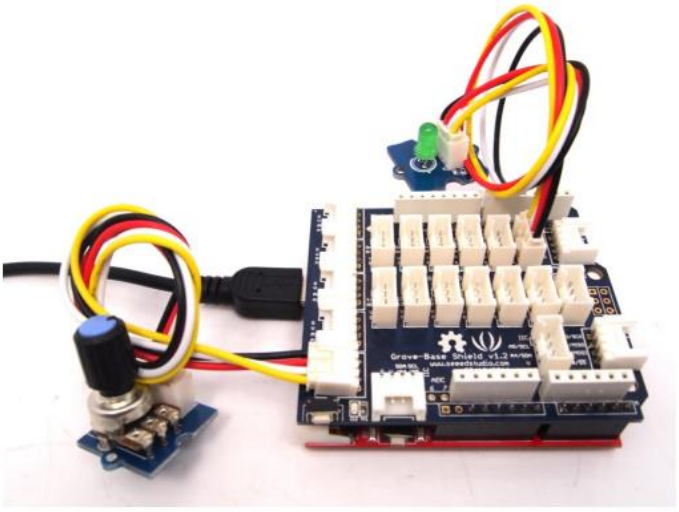
Specifications:

Voltage	0-5 VCC
Rotary Angle	0- 300 degrees

Hardware Required:

Component Name	Quantity
Arduino UNO	1
Grove Light Sensor	1
Grove Rotary Angle sensor	1
Base shield	1
USB Cable	1
Connectors	2

Connection Diagram:



Steps of working

1. Connect a Grove - Rotary Angle Sensor to port A0.
2. Connect a Grove - LED to port D3 of a Base Shield.
3. Plug the Base Shield to the Arduino.
4. Connect Arduino to your PC via an USB cable.
5. Upload the code
6. Open the serial monitor.
7. Rotate the rotary angle sensor and control the brightness of the LED.
8. Observe the change in brightness of LED with change in rotary angle

The Sketch

This sketch works by setting pin A0 as for the Rotary Angle sensor as INPUT and pin D3 as an OUTPUT to power the LED. The initial state of the sensor is set to off and continually reads the angle and position from the rotary angle sensor and sends that value as voltage to the LED. The brightness of the LED will be changed accordingly.

```
/***/macro definitions of Rotary angle sensor and LED pin***/

#define ROTARY_ANGLE_SENSOR A0

#define LED 2//the Grove - LED is connected to D3 of Arduino

#define ADC_REF 5//reference voltage of ADC is 5v.

#define GROVE_VCC 5//VCC of the grove interface is normally 5v

#define FULL_ANGLE 300//full value of the rotary angle is 300 degrees

void setup()

{

Serial.begin(9600);

pinsInit();
```

```
}
void loop( )
{
int degrees;
degrees = getDegree();
Serial.println("The angle between the mark and the starting position:");
Serial.println(degrees);
int brightness;
/*The degrees is 0~300, should be converted to be 0~255 to control the*/
/*brightness of LED */
brightness = map(degrees, 0, FULL_ANGLE, 0, 255);
controlBrightness(brightness);
delay(500);
}
void pinsInit()
{
pinMode(ROTARY_ANGLE_SENSOR, INPUT);
pinMode(LED,OUTPUT);
}
/*PWM control brightness */
/*If brightness is 0,the LED is off. */
```

```
/*The Greater the brightness, the brighter the LED.*/
/*The range of brightness is 0~255 */
void controlBrightness(int brightness)
{
analogWrite (LED,brightness);
}
/*Function: Get the angle between the mark and the starting position */
/*Parameter:-void */
/*Return: -int,the range of degrees is 0~300 */
int getDegree()
{
int sensor_value = analogRead(ROTARY_ANGLE_SENSOR);
float voltage;
voltage = (float)sensor_value*ADC_REF/1023;
float degrees = (voltage*FULL_ANGLE)/GROVE_VCC;
return degrees;}
```

Observation Table:

Sr no.	Angle	LED Brightness
1		
2		

Practical 13

To study and Interface the Sound Sensor with the Arduino

Introduction:

The sound sensor is a small board that combines a microphone (50Hz-10 kHz) and some processing circuitry to convert sound waves into electrical signals. This electrical signal is fed to on-board LM393 High Precision comparator to digitize it and is made available at OUT pin. The module has a built-in potentiometer for sensitivity adjustment of the OUT signal. The threshold can be set by using a potentiometer; So that when the amplitude of the sound exceeds the threshold value, the module will output LOW otherwise HIGH.

Grove - Sound Sensor can detect the sound intensity of the environment. The main component of the module is a simple microphone, which is based on the L358 amplifier and an electret microphone. This module's output is analog and can be easily sampled and tested by an Arduino.

Apart from this, the module has two LED's. The Power LED will light up when the module is powered. The Status LED will light up when the digital output goes LOW.

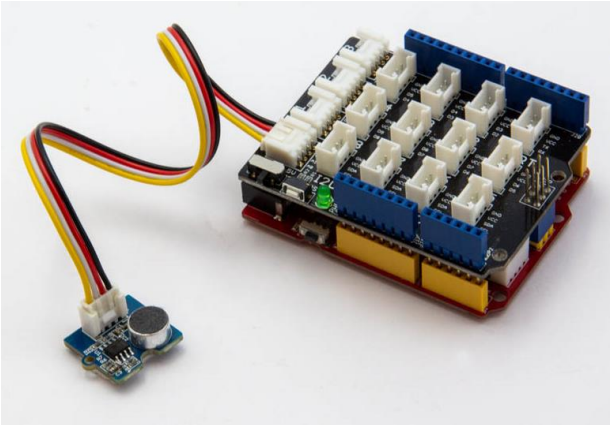
Specifications:

Pin	Description
VCC	3.3V – 5V
GND	Ground
OUT	Outputs HIGH when conditions are quiet and goes LOW when sound is detected

Hardware Required:

Component Name	Quantity
Arduino UNO	1
Sound Sensor	1
USB Cable	1
Base shield	1
Jumper wires	several

Connection Diagram:



Steps of working

1. Connect Grove-Sound Sensor to port A0 of Grove-Base Shield.
2. Plug Grove - Base Shield into Arduino
3. Connect Arduino to PC via a USB cable.
4. To get accurate readings out of your sound sensor, it is recommended that you first calibrate it.
5. The module has a built-in potentiometer for calibrating the digital output (OUT).
6. By turning the knob of the potentiometer, you can set a threshold. So that when the sound level exceeds the threshold value, the Status LED will light up and the digital output (OUT) will output LOW.
7. Now to calibrate the sensor, start clapping near the microphone and adjust the potentiometer until you see the Status LED on the module blink in response to your claps.
8. That's it your sensor is now calibrated and ready for use.
9. Upload the sketch

10. Click on Serial > Plotter to get the changing curve of the sensor with the noise to view the change of the value.

The Sketch

```
// ********Sound Sensor *******//

const int pinAdc = A0;

void setup()

{

   Serial.begin(115200);

   //Serial.println("Grove - Sound Sensor Test...");

}

void loop()

{

   long sum = 0;

   for(int i=0; i<32; i++)

   {

      sum += analogRead(pinAdc);

   }

   sum >>= 5;

   Serial.println(sum);

   delay(10);}
```

Observations:

Sr. No.	Noise	voltage
1	-	-
2	-	-

Important Points:

- ✓ Rotate the knob counterclockwise to increase sensitivity and clockwise to decrease it.
- ✓ Double check that the powers supply is clean. Because the sound sensor is an analog circuit, it's more sensitive to noise on the power supply.
- ✓ The electret microphone on the sound sensor is also sensitive to mechanical vibration and wind noise. Mounting it with a resilient material can help absorb vibration.
- ✓ The sensing range of this sound sensor is very small, probably 10 inches, so you have to make a noise much closer to get a good response.

Practical 14

To study and Interface the Temperature and Humidity with the Arduino

Introduction:

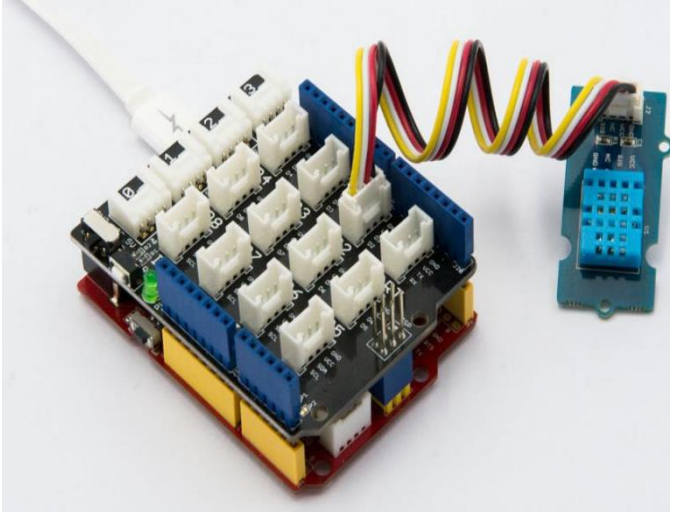
The Temperature Humidity sensor provides a pre-calibrated digital output. A unique capacitive sensor element measures relative humidity and the temperature is measured by a negative temperature coefficient (NTC) thermistor. It has excellent reliability and long term stability. Please note that this sensor will not work for temperatures below 0 degree.

Hardware Required:

Component Name	Quantity
Arduino UNO	1
Temperature & Humidity sensor	1
USB Cable	1

Base shield	1
Jumper wires	1

Connection diagram:



Steps of Working:

1. Connect Grove – Temperature & Humidity Sensor to port D2 of Grove-Base Shield.
2. Plug Grove - Base Shield into Arduino.
3. Connect Arduino to PC via a USB cable.
4. Open the Serial Monitor of Arduino IDE by click Tool-> Serial Monitor and get the temperature.

The Sketch:

```
#include "DHT.h"
//#define DHTTYPE DHT11                 // DHT 11
#define DHTTYPE DHT22                   // DHT 22  (AM2302)
//#define DHTTYPE DHT21                 // DHT 21 (AM2301)
//#define DHTTYPE DHT10                  // DHT 10
//#define DHTTYPE DHT20                 // DHT 20
#define DHTPIN 2                         // what pin we're connected to
                                        （DHT10 and DHT20 don't need define it）
DHT dht(DHTPIN, DHTTYPE);               //   DHT11 DHT21 DHT22
//DHT dht(DHTTYPE);                     //   DHT10 DHT20 don't need to define Pin
// Connect pin 1 (on the left) of the sensor to +5V
// Connect pin 2 of the sensor to whatever your DHTPIN is
// Connect pin 4 (on the right) of the sensor to GROUND
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor
#if defined(ARDUINO_ARCH_AVR)
   #define debug  Serial
#elif defined(ARDUINO_ARCH_SAMD) ||  defined(ARDUINO_ARCH_SAM)
   #define debug  SerialUSB
#else
```

```
    #define debug  Serial
#endif
void setup() {
    debug.begin(115200);
    debug.println("DHTxx test!");
    Wire.begin();
    /*if using WIO link,must pull up the power pin.*/
    // pinMode(PIN_GROVE_POWER, OUTPUT);
    // digitalWrite(PIN_GROVE_POWER, 1);
    dht.begin();
}
void loop() {
    float temp_hum_val[2] = {0};
    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
    if (!dht.readTempAndHumidity(temp_hum_val)) {
        debug.print("Humidity: ");
        debug.print(temp_hum_val[0]);
        debug.print(" %\t");
        debug.print("Temperature: ");
        debug.print(temp_hum_val[1]);
```

```
        debug.println(" *C");
    } else {
        debug.println("Failed to get temperature and humidity value.");
    }

    delay(1500);
}
```

Observations:

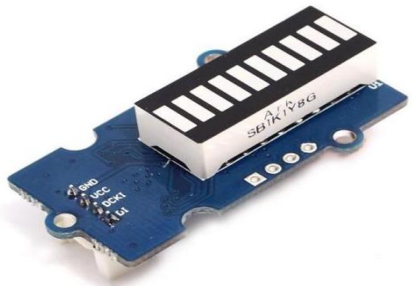
Feature	Value
Temperature	-
Humidity	-

Practical 15

Interfacing LED Bar with Arduino

Introduction:

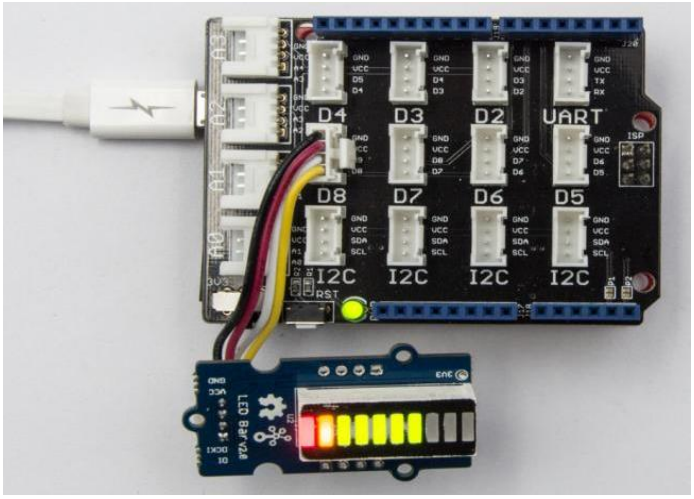
Grove – LED Bar is comprised of a 10 segment LED gauge bar and an MY9221 LED controlling chip. It can be used as an indicator for remaining battery life, voltage, water level, music volume or other values that require a gradient display. There are 10 LED bars in the LED bar graph: one red, one yellow, one light green, and seven green bars. Demo code is available to get you up and running quickly. It lights up the LED's sequentially from red to green, so the entire bar graph is lit up in the end.



Hardware Required:

Component Name	Quantity
Arduino UNO	1
LED Bar	1
USB Cable	1
Base shield	1
Jumper wires	1

Connection diagram:



Steps of working:

1. Connect Grove-LED Bar to port D8 of Grove-Base Shield.
2. Plug Grove - Base Shield into Arduino.
3. Connect Arduino to PC via a USB cable.
4. Upload the code
5. Observe the changes in the state of the LED Bar.

The Sketch:

This sketch works by connecting portD8 as for the LED Bar . After that the run a loop that continually reads the LED Bar variance.

//*** LED Bar****//

#include <Grove_LED_Bar.h>

Grove_LED_Bar bar(6, 7, 0, LED_CIRCULAR_24); // Clock pin, Data pin, Orientation

```
void setup()
 {
   // nothing to initialize
   bar.begin();
}
void loop() {
   // Walk through the levels
   for (int i = 0; i <= 10; i++) {
      bar.setLevel(i);
      delay(100);
   }
}
```

Observation Table:

Sr. no.	Light detected	LED state
1		
2		

Practical 16

Interfacing of light sensor with Arduino

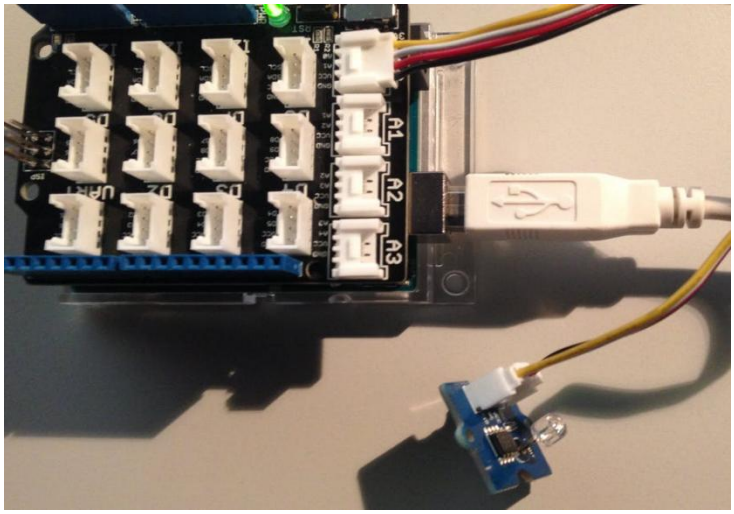
Introduction:

The Grove - Light sensor integrates a photo-resistor (light dependent resistor) to detect the intensity of light. The resistance of photo-resistor decreases when the intensity of light increases. A dual OpAmp chip LM358 on board produces voltage corresponding to intensity of light (i.e. based on resistance value). The output signal is analog value, the brighter the light is, the larger the value.

Hardware Required:

Component Name	Quantity
Arduino UNO	1
Base Shield	1
USB Cable	1
Grove Light Sensor	1
connecting wires	1

Connection Diagram:



Steps of working:

1) Connect Grove-Light Sensor to port A0 of Grove-Base Shield.
2) Plug Grove - Base Shield into Arduino.
3) Connect Arduino to PC through a USB cable.
4) Observe the light intensity readings on the serial monitor.

The Sketch:

This sketch works by setting pin A0 as for the Light sensor.

```
#include <math.h>

#define LIGHT_SENSOR A0   //Grove - Light Sensor is connected to A0 of
                          Arduino

const int ledPin=12;      //Connect the LED Grove module to Pin12, Digital12

const int thresholdvalue=10;
```

```
float Rsensor;
void setup()
{
   Serial.begin(9600);                  //Start the Serial connection
   pinMode(ledPin,OUTPUT);              //Set the LED on Digital 12 as an OUTPUT
}
void loop()
{
int sensorValue = analogRead(LIGHT_SENSOR);
   Rsensor = (float)(1023-sensorValue)*10/sensorValue;
   Serial.println("the analog read data is ");
   Serial.println(sensorValue);
   Serial.println("the sensor resistance is ");
   Serial.println(Rsensor,DEC); //show the light intensity on the serial monitor;}
```

Observation Table:

Sr. no.	Analog value	Resistance value
1		
2		
3		

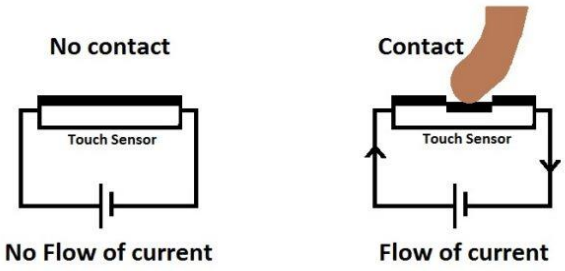
Practical 17

Interfacing Touch sensor with the Arduino to operate the Buzzer

Introduction:

A touch sensor is an electronic sensor used in detecting and recording physical touch. Also known as tactile sensors, it's a small, simple, low-cost sensor made to replace old mechanical switches.

Grove - Touch Sensor enables you to replace press with touch. It can detect the change in capacitance when a finger is nearby. That means no matter your finger directly touches the pad or just stays close to the pad, Grove - Touch Sensor would outputs HIGH also.



A touch sensor works like a switch, where when there's contact, touch, or pressure on the surface of a touch sensor, it opens up an electrical circuit and allows currents to flow through it.

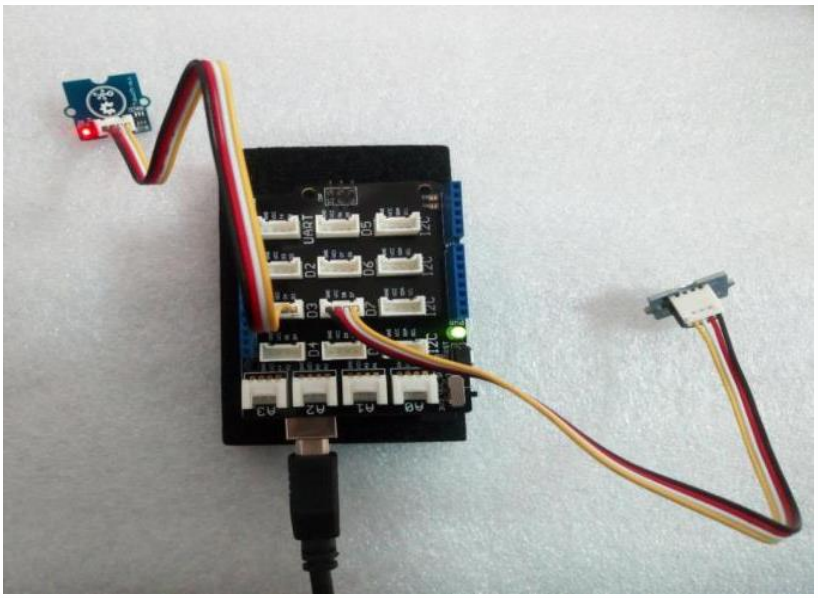
Specifications:

Parameter	Values
Operating Voltage	2.0 - 5.5V
Operating Current(Vcc=3V)	1.5 - 3.0μA
Operating Current(VDD=3V)	3.5 - 7.0μA
Output Response Time	60 - 220mS
Used Chipset	TTP223-BA6

Hardware Required:

Component Name	Quantity
Arduino UNO	1
Touch sensor	1
Buzzer	1
USB Cable	1
connectors	2

Connection Diagram:



Steps of working:

1. Connect a Grove - Rotary Angle Sensor to port A0.
2. Connect a Grove - Red LED to port D3 of a Base Shield.
3. Plug the Base Shield to the Arduino.
4. Connect Arduino to your PC via an USB cable.
5. Upload the code
6. Open the serial monitor
7. Rotate the rotary angle sensor and control the brightness of the LED.

The Sketch

```
const int pinTouch = 3;                    // pin of button define here
const int pinBuzzer    = 7;                    // pin of led define here
void setup()
{
   pinMode(pinTouch, INPUT);                 // set button INPUT
   pinMode(pinBuzzer, OUTPUT);                    // set buzzer OUTPUT}
void loop()
{  if(digitalRead(pinTouch))                  // when button is pressed
   {
      digitalWrite(pinBuzzer, HIGH);            // buzzer on    }
   else
   { digitalWrite(pinBuzzer, LOW);
   }
   delay(1000);
}
```

Observations:

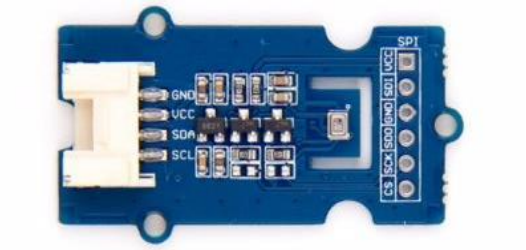
Sr. no.	Touch sensor	Buzzer
1		
2		

Practical 18

Interfacing Barometer Sensor (BMP 280) with Arduino

Introduction:

Grove - Barometer Sensor (BMP280) is a breakout board for Bosch BMP280 high-precision and low-power digital barometer. This module can be used to measure temperature and atmospheric pressure accurately. As the atmospheric pressure changes with altitude, it can also measure approximate altitude of a place. It can be connected to a micro-controller with I2C (integrated with Grove socket) or through SPI bus.



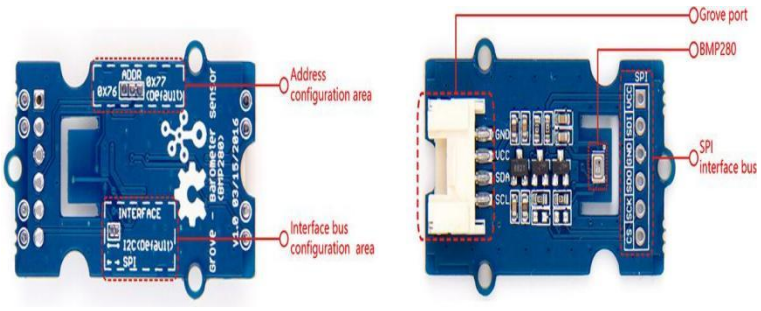
Hardware Overview:

The Grove Barometer sensor consists of:

i. SPI soldering pads, a voltage monitoring circuit.

ii. Interface bus selection pads, to select I2C bus, connect the two pads by soldering (this is connected by default); to select SPI bus, cut the two pads with a sharp knife or a soldering iron.

iii. Slave board address selection pads, to select slave board address to avoid address collision.



Specifications:

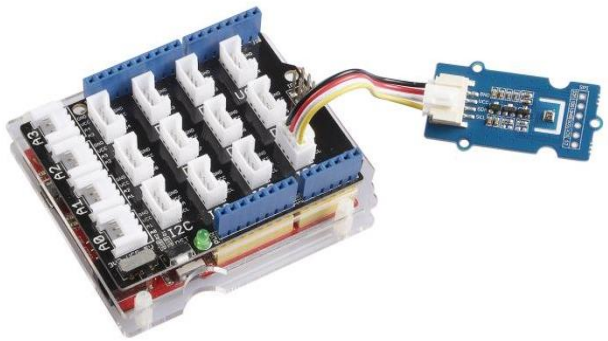
Parameter	Value
Input voltage	3.3V or 5V
I/O voltage	3.3V or 5V
Operating current	0.6mA
Operating temperature	-40 - 85 ℃
Effective pressure measurement range	300 - 1100 hPa (1 hPa= one hundred Pa) with ±1.0 hPa accuracy
Temperature measurement accuracy	±1.0°C
Measurement modes	Piezo & Temperature, forced or periodic
Chip	BMP280 (datasheet)

Parameter	Value
Possible sampling rate	182 Hz (typical)
Interface Bus	SPI, I2C (use either one of them)
Weight	3 g (for breakout board)
Dimensions	40 (width) × 20 (depth) mm

Hardware Required:

Component Name	Quantity
Arduino UNO	1
Barometer sensor BMP 280	1
Base shield	1
USB Cable	1
wires	1

Connection diagram:



Steps of working:

1. Connect Grove-Barometer_Sensor-BMP280 to port I2C of Grove-Base Shield.
2. Plug Grove - Base Shield into Arduino and connect Arduino to PC via a USB cable.
3. Create a Arduino sketch
4. Open the serial monitor to receive the sensor's data including temperature, barometric pressure valueand altitude.

The Sketch:

```
//****** Barometer*****//

#include "Seeed_BMP280.h"

#include <Wire.h>

void setup()

{       Serial.begin(9600);

 if(!bmp280.init()){

   Serial.println("Device error!"); }}

void loop()

{  float pressure;

 //get and print temperatures

 Serial.print("Temp: ");

 Serial.print(bmp280.getTemperature());

 Serial.println("C"); // The unit for  Celsius because original arduino don't support speical symbols
```

```
//get and print atmospheric pressure data
Serial.print("Pressure: ");
Serial.print(pressure = bmp280.getPressure());
Serial.println("Pa");
//get and print altitude data
Serial.print("Altitude: ");
Serial.print(bmp280.calcAltitude(pressure));
Serial.println("m");
Serial.println("\n");//add a line between output of different times.
delay(1000);}
```

Observation Table:

Sr no.	Measured	Value
1	Temperature	
2	Pressure	
3	Altitude	

Practical 19

Interfacing of the Gas Sensor with Arduino

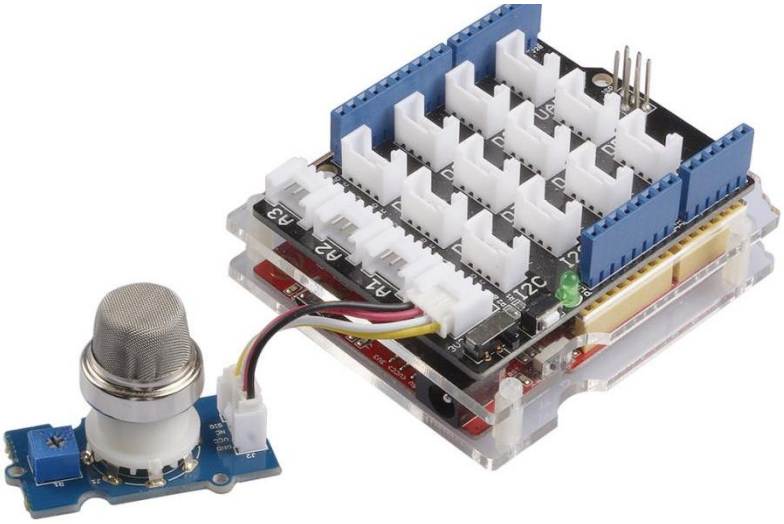
Introduction:

The Grove - Gas Sensor (MQ2) module is useful for gas leakage detection (home and industry). It is suitable for detecting H2, LPG, CH4, CO, Alcohol, Smoke or Propane. Due to its high sensitivity and fast response time, measurement can be taken as soon as possible. The sensitivity of the sensor can be adjusted by potentiometer.

Hardware Required:

Component Name	Quantity
Arduino UNO	1
Gas Sensor	1
USB Cable	1
Breadboard	1
Jumper wires	several

Connection Diagram:



Steps of working:

1. Connect Grove-Gas_Sensor-MQ2 to port A0 of Grove-Base Shield.
2. Plug Grove - Base Shield into Arduino.
3. Connect Arduino to PC via a USB cable.
4. Upload the code.
5. Observe the clicking sound of the relay that states the ON and OFF constantly.

Sketch:

This sketch works by setting 5V supply pin of Arduino as for the control of relay module. After that the run a loop that continually sends that value as voltage to the D6 with the delay given.

```
void setup()
{
   Serial.begin(9600);
```

```
}
void loop()
{
   float sensor_volt;
   float sensorValue;
   sensorValue = analogRead(A0);
   sensor_volt = sensorValue/1024*5.0;
   Serial.print("sensor_volt = ");
   Serial.print(sensor_volt);
   Serial.println("V");
   delay(1000);
}
```

Observations:

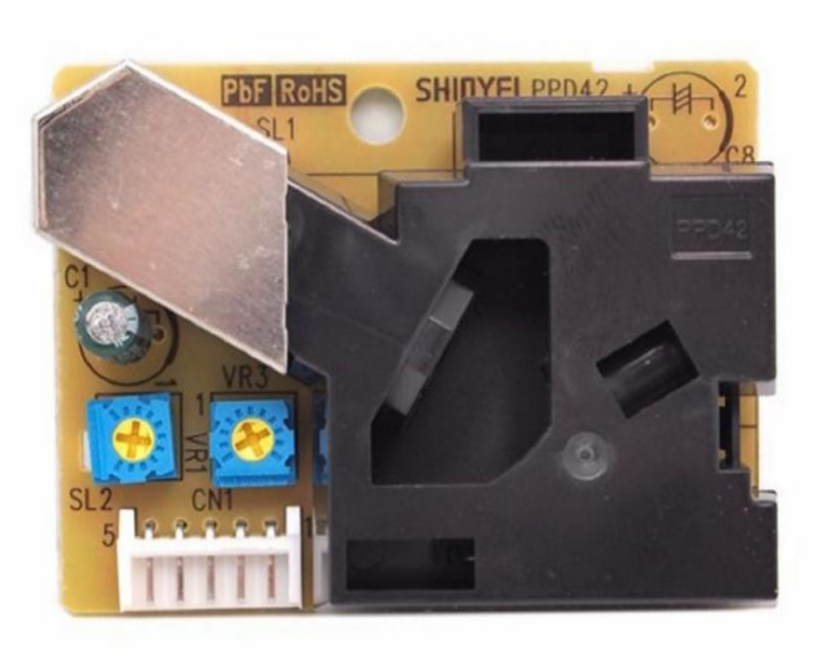
Sensor Voltage	

Practical 20

Air Quality Monitoring using the Particle Sensor with Arduino

Introduction:

The Dust Sensor gives a good indication of the air quality in an environment by measuring the dust concentration. The Particulate Matter level (PM level) in the air is measured by counting the Low Pulse Occupancy time (LPO time) in a given time unit. LPO time is proportional to PM concentration. This sensor can provide reliable data for air purifier systems; it is responsive to PM of diameter 1μm.This sensor uses a counting method to measure dust concentration, not weighing method, and the unit is pcs/L or pcs/0.01cf.



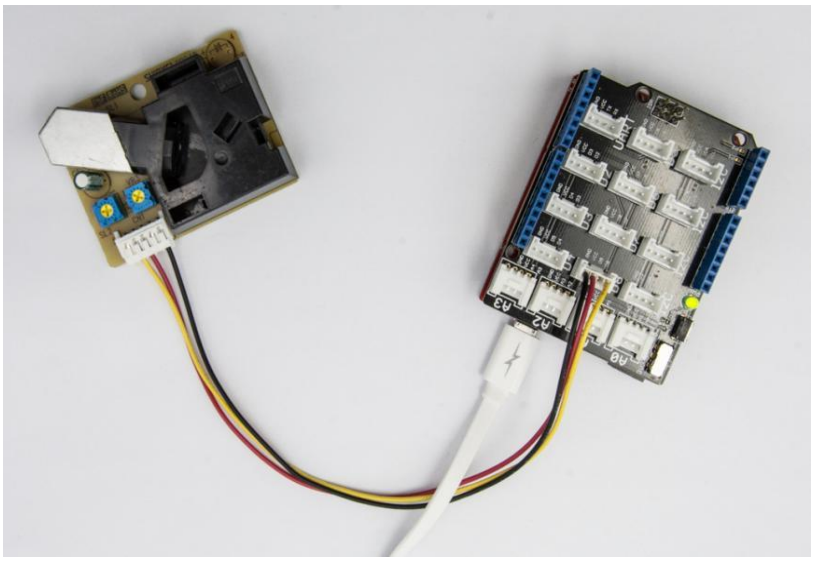
Specifications:

Parameter	Value
Supply voltage range	5V
Minimum detect particle	1um
PWM	Output
Dimensions	59(W)x45(H)x22(D) [mm]

Hardware Required:

Component Name	Quantity
Arduino UNO	1
Grove - Dust Sensor (PPD42NS)	1
Jumper Wires (Male to Male)	3
USB Cable	1
Breadboard	1

Connection Diagram:



Steps of working

1) Connect Grove-Dust Sensor to D8 port of Grove-Base Shield.
2) Plug Grove - Base Shield into Arduino.
3) Connect Arduino to PC via a USB cable.
4) Upload the code
5) Open Serial Monitor, and get air quality's value detected by sensor from PC's serial port.

Sketch:

In this program, the Arduino samples the total duration of "logic low" in 30s, and this duration illustrates the dust density of environment.The air quality's value detected by sensor from PC's serial port.

```
/**********Dust Detection******/
int pin = 8;
unsigned long duration;
unsigned long starttime;
```

```
unsigned long sampletime_ms = 2000;
unsigned long lowpulseoccupancy = 0;
float ratio = 0;
float concentration = 0;
void setup() {
  Serial.begin(9600);
  pinMode(8,INPUT);
  starttime = millis(); }
void loop() {
duration = pulseIn(pin, LOW);
  lowpulseoccupancy = lowpulseoccupancy+duration;
  if ((millis()-starttime) >= sampletime_ms) //if the sampel time = = 30s
  {  ratio = lowpulseoccupancy/(sampletime_ms*10.0);
    concentration = 1.1*pow(ratio,3)-3.8*pow(ratio,2)+520*ratio+0.62;
    Serial.print("Concentration = ");
    Serial.print(concentration);
    Serial.println(" pcs/0.01cf");
    Serial.println("\n");
    lowpulseoccupancy = 0;
    starttime = millis();  }
```

Observations:

Air Quality Value	concentration