

## Overview

Arduino is prototype platform based on an easy-to-use hardware and software.

It consists of a circuit board, which can be programmed to ready-made software called Arduino IDE, which is used to write and upload the computer code to the physical board.

## Key Features

- Arduino boards able to read analog or digital input signals & produce output such as activating a motor, turning LED on/off.
- Board functions can controlled by sending a set of instructions to all microcontroller.
- Arduino use to USB cable in order to load a new code on the board.

## Board description

Arduino have components like.

- Power USB : Arduino board can be powered by using the USB cable from your computer.

2) Power (Board Jack) : Arduino board can be powered directly from the AC mains power supply by connecting it to the Barrel Jack.

3) Voltage regulator: The function of the voltage regulator is to control the voltage given to Arduino board.

4) Crystal oscillator: The crystal oscillator helps Arduino in dealing with time issues.

5, 11) Arduino Reset: you can reset your Arduino board i.e. start your program from the beginning. You can reset the UNO board in 2 ways.  
First, by using reset button (n)  
Second, you can connect an external reset button to RESET (S)

6, 7, 8, 9) Pins (3, 3, 5, GND, Vin)

- 3.3V (6) - supply 3.3 output volt
- 5V (7) - supply 5 output volt
- GND (8) (Ground) - These are several GND pins, on the Arduino, used to ground your circuit.

- 10) Analog pins : The Arduino UNO board has 6 analog input pins A0 through A5. These pins can read signals from an analog sensor like humidity sensor or temperature sensor.
- 11) Main microcontroller : Each Arduino board has its own microcontroller (ATmega328P). You can assume it as the brain of your board.
- 12) ICSP pin : ICSP (12) is an AVP, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC and GND.
- 13) Power LED indicator : This LED should light up when you plug your Arduino into a power source to indicate that your board is powered by correctly.
- 14) TX and RX LEDs : On your board, you will find 2 labels : TX (transmit) & RX (receive). They appear in two places on the Arduino UNO board.  
First, at the digital pins 0 & 1, to indicate the pins responsible for serial communication.

Second, the TX and RX led(13). The TX called led flashes with different speed while sending the serial data. RX flashes during the receiving process.

(5) Digital I/O: The Arduino UNO board has 14 digital I/O pins (15) of which 6 provide Pulse width modulation output.

(6) Free AREF: This stands for analog reference. It is used to set an external reference voltage as the upper limit per analog input pins.

### Arduino Installation.

Step1: First you must have Arduino board and a USB cable.

Step2: Download Arduino IDE software.

Step3: Power up your board.

Step4: Launch Arduino IDE

Step5: Open your first project

Step6: Select your Arduino board.

Step 7: Select your serial port.

Step 8: Upload the program to your board.

✓ - used to check if there is any compilation error.

→ - used to upload a program to the Arduino board.

.createNewSketch - shortcut used to create a new sketch.

↑ ↓ - shortcut used to create a new sketch.  
Used directly open one of the example sketch and to save your sketch.

serialMonitor - serial monitor used to receive serial data from the board & send the serial data to the board.

### Program Structure:

Three main parts of Arduino Programs:  
structure, values . functions.

Software structure consists of a main functions.

- setup() function.
- loop() function.

## Void setup() 23

- Purpose - The setup() function is called when a sketch starts use it to initiate the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

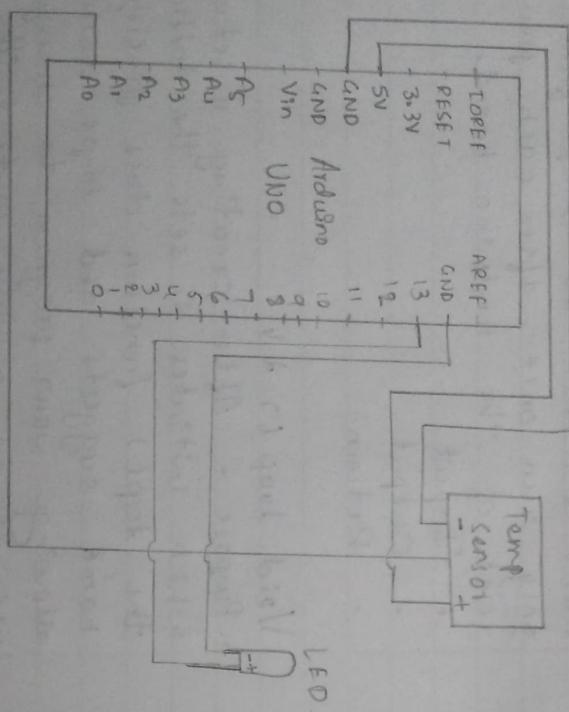
- Input
- Output
- Return

## Void loop() 23

- Purpose - After creating a setup() function which initiates & sets the initial values, the loop() function does precisely what its name suggests and loops consecutively, allowing your program to change & respond use it to actively control the Arduino board.

- Input
- Output
- Return

## Circuit Diagram:



Date / /

Expt. No. 1a)

Page No.

7

## Microcontroller experiments using Arduino and TINMSP430

### 1a) Testing Temperature Sensors.

Aim: To implement temperature sensor

Components required: arduino board, breadboard, wire, temperature sensor, LED bulb.

Theory: Temperature sensor converts the surrounding temperature to voltage. It can sense the temperature & it is put around and transmit it to degrees Celsius

Code:

```
int val;
int tempPin = 13;
void setup() {
  Serial.begin(9600);
}

void loop() {
  val = analogRead(tempPin);
  float mv = val / 1024;
  float cel = mv * 10;
```

Output:

Temperature = 30°C  
Temperature = 31°C  
Temperature = 32°C

```
float flash = (cel * 9) / 5 + 32;  
Serial.print ("Temperature = ");  
Serial.print (val);  
Serial.print (" °C");  
Serial.println ();  
delay (1000);  
}
```

卷之三

Aim : To implement fire sensor

Components required : Arduino board, bread board, jumper wires, fire sensor, buzzer.

```
Code : #include <softwareSerial.h>
```

```
int sensorPin = A0;
int sensorValue = 0;
int led = 29;
int buzzen = 12;

void setup() {
    pinMode(led, OUTPUT);
    pinMode(buzzen, OUTPUT);
}

Serial.begin(9600);

void loop() {
    sensorValue = analogRead(sensorPin);
    if (sensorValue < 100) {
        digitalWrite(led, HIGH);
        digitalWrite(buzzen, HIGH);
        delay(100);
        digitalWrite(led, LOW);
        digitalWrite(buzzen, LOW);
        delay(100);
    }
}
```

Output:-

No fire

fire detected

fire detected

No fire

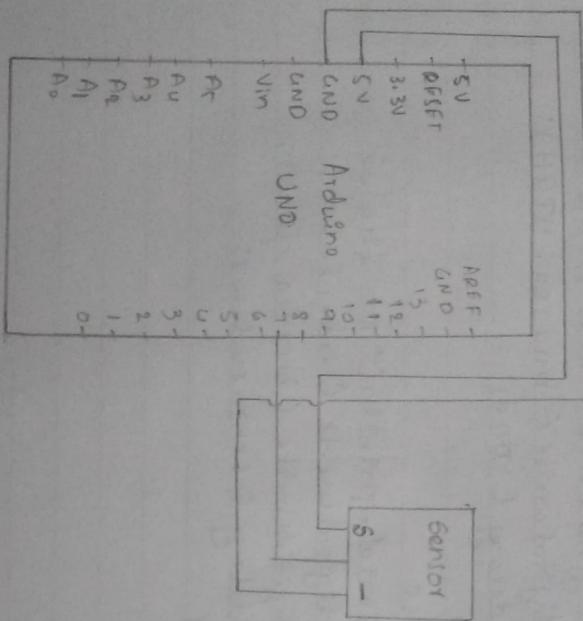
```

Serial.println("Fire Detected");
Serial.println("LED ON");
digitalWrite(led, HIGH);
digitalWrite(buzzer, HIGH);
delay(1000);
}

else
{
Serial.println("No fire");
digitalWrite(led, LOW);
digitalWrite(buzzer, LOW);
delay(sensorValue);
}

```

### Circuit Diagram:



### 1c) Heart beat sensor.

Aim: To implement heart beat sensor

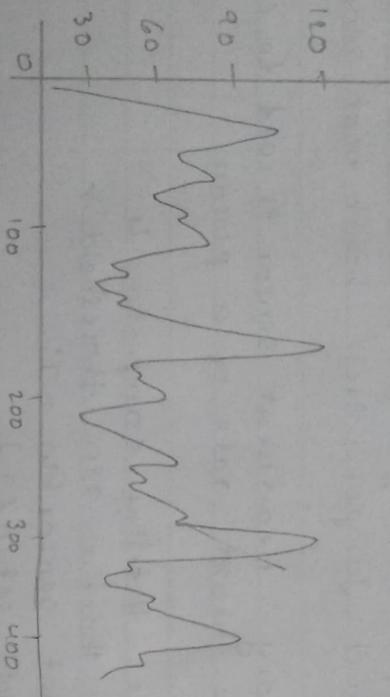
Components required: Arduino board, bread board, jumper wires, heart beat sensor.

Theory : Heartbeat sensor is used to calculate the heart rate of a person

```
code: #define samp_size 1
#define rise_threshold 4
int sensorPin = 7;
void setup() {
    Serial.begin(9600);
}

void loop() {
    float reads[Samp_size], sum;
    long int now, ptr;
    float last, reader, slent;
    float first, second, third, before, print_value;
    bool rising;
    int rise_count;
    int n;
```

output: (serial plotter)



```

long int last-beat;
for( int i=0; i< samp-size ; i++) {
    reads[i] = 0, y
    sum = 0;
    ptr = 0;
    while (1) {
        n = 0;
        Start = millis();
        readon = 0;
        do {
            reader += analogRead(sensorPin);
            now = millis();
            y
            while (now < start + 20);
            readen / = n;
            sum -= reads[ptr];
            lastsum += reader;
            reads[ptr] = reader;
            last = sum / samp-size;
            if (last > before)
            {
                rise_count++;
                if (rising && subsequent_rise_threshold)
            }
        }
    }
}

```

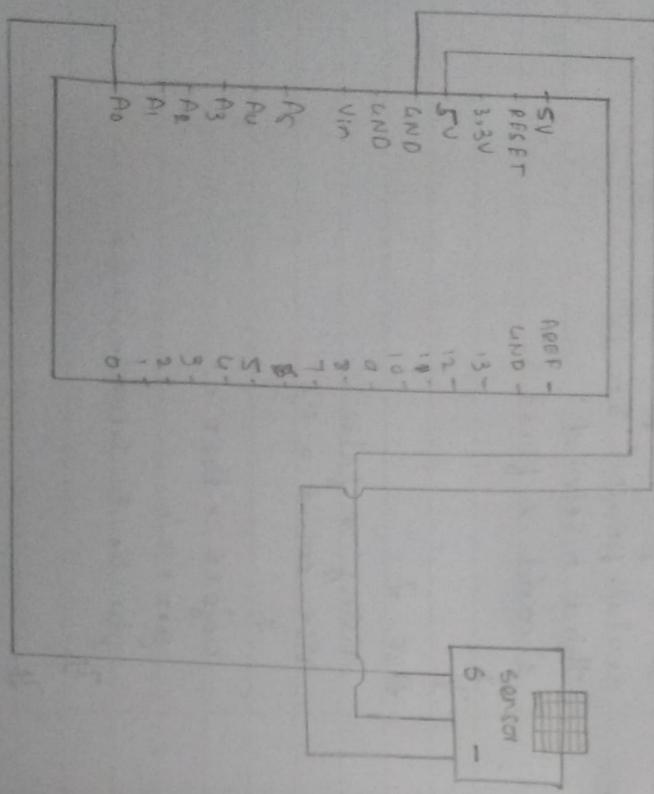
output: (serial port)

```

rising = true;
first = millis() - last-beat;
last-beat = millis();
print-value = 60000 / (0.1 * float + 0.3 * second
                        + 0.3 * third);
serial.print (print-value);
serial.print ("\\n");
third = second;
second = first;
}
}
else {
    rising = false;
    rise-count = 0;
}
before = last;
ptr++;
ptr % Samp-size;
}
}

```

Circuit diagram:



## 2b) Humidity sensor

Aim : To implement humidity sensor

Components required : Arduino board, bread board, jumper wires, humidity sensor.

Theory : The humidity sensor detects the amount of water vapour in air.

```
Code: #include "DHT.h"
#define dht - pin A0
DHT dht;
void setup() {
  Serial.begin (9600);
  delay (500);
  Serial.println ("Humidity & Temp. Sensor");
  delay (1000);
}
void loop() {
  DHT.read (dht - pin);
  Serial.print ("Current humidity ");
  Serial.print (DHT.humidity);
  Serial.print ("% ");
}
```

Date / /

Expt. No. □

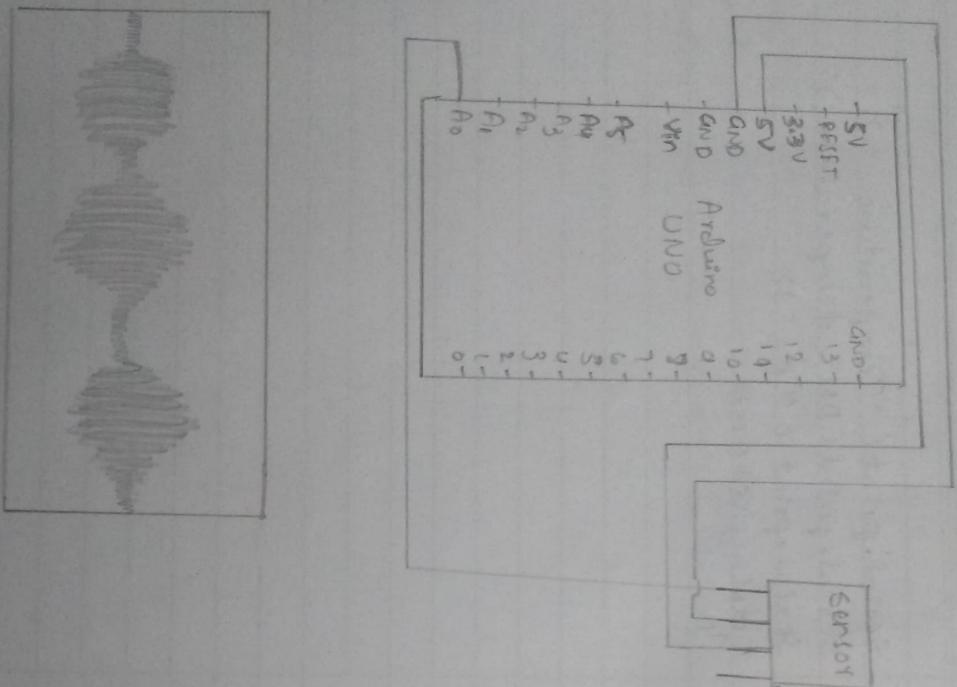
Page No. □ 15

Output:

Current humidity = 56.00% Temperature = 19.00°C  
Current humidity = 27.00% Temperature = 37.00°C  
Current humidity = 26.00% Temperature = 36.00°C

```
Serial.print("Temperature = ");  
Serial.print(DHT.temperature);  
Serial.print(" °C");  
delay(5000);  
y
```

## Circuit diagram:



Output :-

```

2d) Big sound / small sound sensor

Aim : To implement big/small sensor

Components required : Arduino board, bread board
jumper wires, round sensor.

Theory : This sensor detects sound & gives
a measurement of how broad/loud sound.

code : void setup() {
    sensor.begin(9600);
}

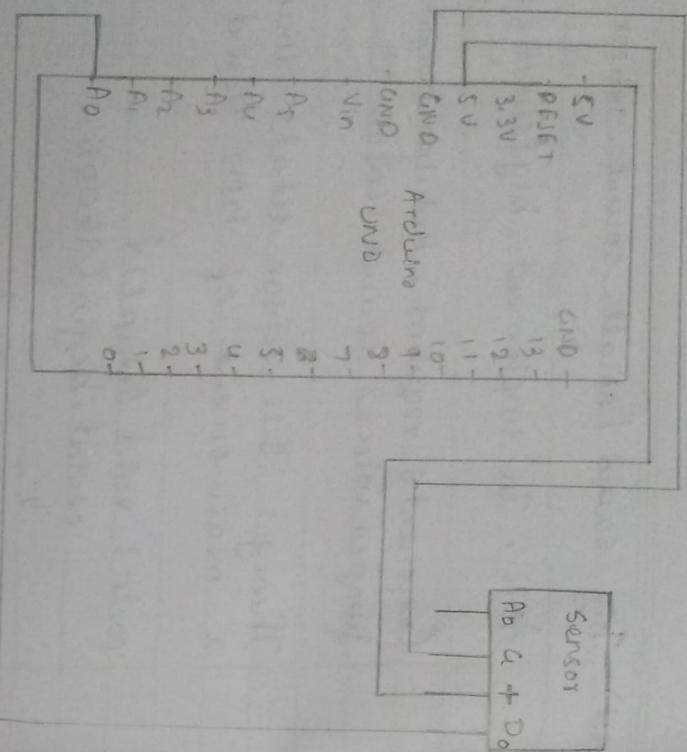
void loop() {
    int sensorValue = analogRead(A0);
    serial.println(sensorValue);
}

```

Date / /

Expt. No. \_\_\_\_\_

Page No. 16



### 3a) Touch sensor

Aim :- To implement touch sensor

Components required : Arduino board, bread board, jumper wires, touch sensor.

Theory :- Detects the sensor is touched or released - Toggle LED when the sensor is touched.

```
Code: int touchPin = A0;
```

```
int sensorValue = 0;
```

```
void setup()
```

```
Serial.begin(9600);
```

```
}
```

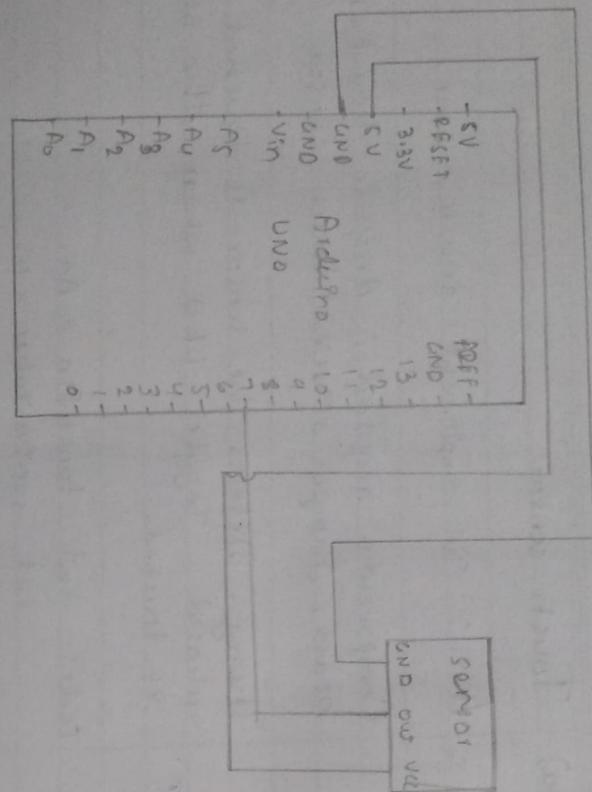
```
void loop()
```

```
sensorValue = analogRead(touchPin);
```

```
Serial.println(sensorValue, DEC);
```

```
delay(1000);
```

```
}
```



### 3b) Tracking sensor.

Aim: To implement tracking sensor

Components required : Arduino board, bread board, tracking sensor, jumper wires.

Theory : This sensor senses black lines against white surface or white lines against black surface.

```
Code: Void setup() {pinMode(13, OUTPUT);  
printCode(7, INPUT); serial.begin(9600);}
```

```
void loop() {if (digitalRead(7) == HIGH){  
    serial.println ("Light surface");  
    digitalWrite(13, HIGH);  
}  
else {  
    serial.println ("Dark surface");  
    digitalWrite(13, LOW);  
}  
delay (1000);  
}
```

Output:  
 light surface  
 Dark surface  
 Dark surface  
 Light surface.

### Circuit diagram

Date / /

Expt. No. [ ]

Page No. 19

4a) Mercury tilt switch.

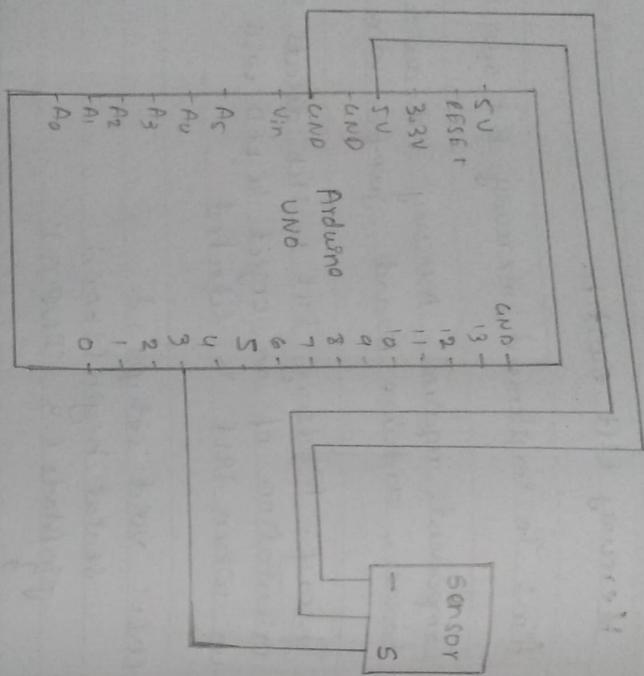
Aim: To implement mercury tilt switch

Components required: Mercury tilt switch, bread board, arduino board, jumper wires

```
code: void setup() {  
    serial.begin(9600);  
    pinMode(3, INPUT);  
}
```

```
void loop() {
    if (digitalRead(3) == 21)
        serial.println("Tilted");
    else
        serial.println("Not tilted");
    delay(1000);
}
```

output:  
Not tilted  
Not tilted  
Tilted  
Tilted



**H.C) Ball switch.**

**Aim :** To implement ball switch

**Components required :** Ball switch, arduino board, bread board, jumper wires.

**Theory :** This sensor detect basic motion, orientation or inclination of object.

Code : void setup() {

  Serial.begin(9600);

  pinMode(3, INPUT);

}

void loop() {

  if (digitalRead(3) == 1)

    Serial.println("Tilted");

  else

    Serial.println("Not tilted");

    delay(1000);

}

**Output:**

Not tilted  
tilted  
tilted  
Not tilted.

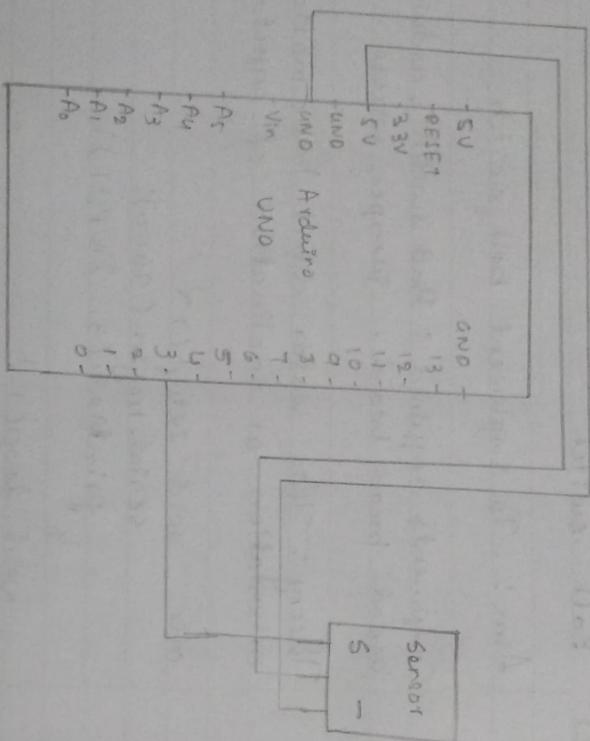
5a)

### Button sensor

Aim: To implement button sensor

Components required: Button sensor, Arduino board, bread board, jumper wires.

### Theory:



Output:-

Not pressed

Not pressed  
pressed  
pressed  
pressed

```

void setup() {
  serial.begin(9600);
  pinMode(3, INPUT);
}

void loop() {
  if (digitalRead(3) == 1)
    serial.println("Not yet pressed");
  else
    serial.println("pressed");
  delay(1000);
}

```