2020BTECS00110
Mayur Kamble

Practical No : 1

Aim - Write a program to blink onboard LED

Components requires - Arduino board, LED & resistor, wires etc.

Procedure :-

Mention pin no to be used globally. Initialize pin to I/P or O/P.

If output write on pin (LED ON/OFF)

setup :-

```
pinMode (13, OUTPUT) ; // 13 as an OUTPUT
```

LOOP :-

```
digitalWrite (13, HIGH) ; // led ON
delay (1000) ;
digitalWrite (13, LOW) ; // led OFF
```

Conclusion - After running the programme all conditions are working properly & the LED blinks as given time interval ON and OFF resp.

Practical No. 2

Aim :- Arduino interfacing with tricolor LED and push button.

Components - Arduino uno board, Tricolor LED, resistors, Jumpet wires, push button etc.

Procedure :-

i) Tricolor LED ~~are~~ contains Red, Blue & Green colours.

ii) Different LEDs are connected to arduino unoboard via resistors.

iii) Also connect push button to arduino uno board via resistor.

iv) I/O pins used here are 10, 11, 12.

Set up :-

```
pinMode (10, OUTPUT) ; // red led.
pinMode (11, OUTPUT) ; // blue led.
pinMode (12, OUTPUT) ; // green led
```

LOOP -

```
digitalWrite (10, HIGH) ; // red led ON
delay (500);
digitalWrite (10, LOW) ; // red led OFF
delay (500);
digitalWrite (11, HIGH) ; // blue led ON
delay (500);
digitalWrite (11, LOW) ; // blue led OFF
delay (500);
```

Push button :-

```
pinMode (4, INPUT); // pin 4 as led

digitalWrite (12, HIGH); // green led ON
delay (500);
digital Write (12, LOW); // green led OFF
delay (500);
```

Conclusion - The tricolour LED works as programme setted & done and blinks all colours i.e. Red, Blue & green at equal time interval as seted & starts blink after pushing push button.

Practicale No-3

Aim :- Sensing analog voltage using onboard ADC & printing it on serial monitor

Procedure :-

Part A- Use of Serial pins on arduino uno

A1) Program to ~~sed~~ send data to PC(TX)

~~Print~~("Print "Hellow world"

i) Setup serial command
serial begin(baudrate)

ii) serial.print can print a string.
serial print can print a string
Serial print ("Hello World")

setup-
serial begin (9600); // setup serial comn with baud rate of 9600.

loop-
Serial println ("Hello World");
delay (1000);

A2) Program to recieve data from Pc (Rx)

Arduino Rx Pin → Bit 0 → led off
→ Bit 1 → led ON

Procedure :-

i) Setup serial communication
ii) checks availability of data on Rx pin
iii) Read the data on Rx pin serial read bit 0
bit 1
iv) Check the bit received.
Bit 0 → LOW on pin 13
Bit 1 → High on pin 13

Set up :-

```
int led = 13 ; int value = 0;
serial begin (9600); // set up serial communicating
pinMode (led, OUTPUT);
```

LOOP :-

```
Value = Serial.read ();
delay (5);
if (value == '1')
digitalWrite (led, HIGH); // led is ON
serial print IN ("LED is ON");

If (value == "0")
digitalWrite (led, LOW); // led is off
serial.print IN ("LOW is off")
```

Part B - Sensing analong voltage using onboard ADC & print it on serial monitor.

Relating ADC value to voltage conversion

$$\frac{\text{Resolution of ADC}}{\text{system voltage}} = \frac{\text{ADC reading}}{\text{Analog voltage measured}}$$

$$\frac{1023}{5V} = \frac{\text{ADC reading (varies)}}{\text{Analog voltage measured.}}$$

Analog Voltage measured = 2.12 V

$$\therefore \text{ADC reading} = \frac{1023 \times 2.12}{5}$$

ADC reading = 434 is a digital value

Setup

```
serial begin (9600);
```

loop:-

```
analogRead (A3) = adc reading
serial.print ln("adc-reading");
delay (1000);
Float voltage = adc-reading x (50/1023); //(0-1023) to (0-5)V
delay (1000);
```

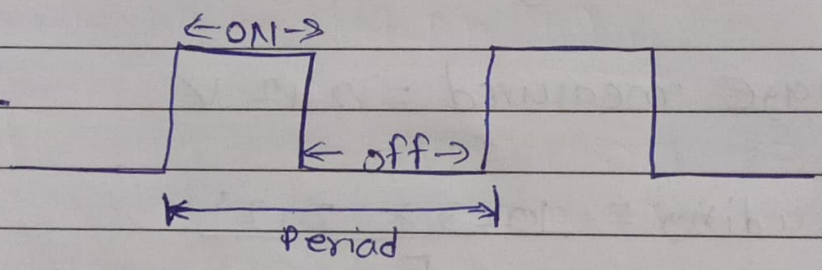
Conclusion - sensing analog voltage using onboard ADC and printing it on serial monitor is done.

Practical No :- 4

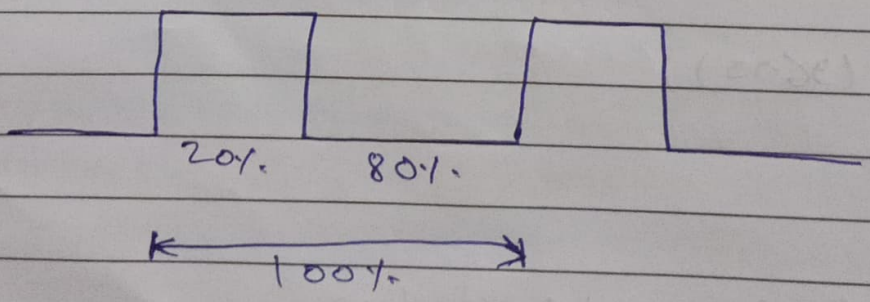
Aim :- To develope PWM generation using aurdino uno.

PWM - Pulse with modulation.

Procedure :-



Above 50% duty cycle with frequency of Hz 20% duty cycle we have.



Pins required - 3, 5, 6, 9, 10, 11.
PWM O/P in arduino is,
8-bit value - $2^8 = 256$
0 → 255
0v → 5v

100% = 255

70% = ? $= \frac{255 \times 70}{100} = 179$ 70% duty cycle

$V_{eff} = V_s \left(\frac{T_{ON}}{T}\right)$ ← Duty cycle

$V_{eff} = V_s$ [Duty Cycle]

~~Veff~~

PWM o/p level $= \frac{255 \times V_{eff}}{V_s}$

$= 255 \times \frac{V_{eff}}{5} = 51 \times V_{eff}$

```
int led_pin = 9 ;

setup -
    pinMode (led_Pin, OUTPUT) ;

loop -
analogWrite (9, 0) ; // 0% duty cycle
delay (2000);
analogWrite (9, 76); // 30% Duty cycle.
delay (2000);
analogWrite (9, 153); // 60% Duty cycle.
delay (2000);
analogWrite (9, 255); // 100% duty cycle.
delay (2000);
analogWrite (9, 0); // 0% Duty cycle
delay (2000);
```

```
int led-Pin = 9;

Setup -

  PinMode (led-Pin, OUTPUT);

loop -

  For int i=0, i<255, i++) {
    analogWrite (led-Pin, i);
    delay (100);
    For (int i = 255; 170; i = -1 {
      analogWrite (led-pin, i);
      delay (100);
```

Conclusion - PWM generation is done by using arduino uno.

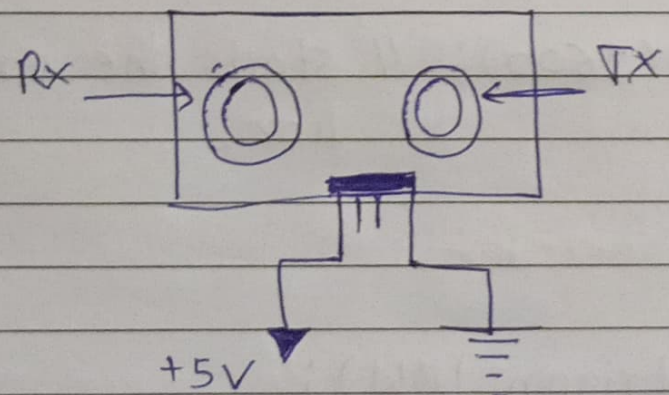
Practical No: 5

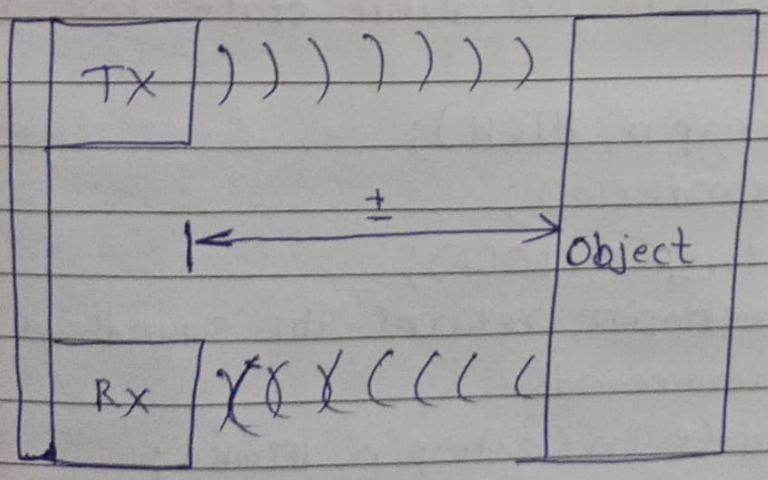
Aim - To interface ultrasonic sensor with arduino for distance measurement.

Procedure :-

Ultrasonic Sensor - HCSR04

- Trig - High state for 10µs
- Echo - O/P the time in µsec the sound wave has travelled.





D = s.t.

```
Const int trigpin = 9;
const int echoPin = 10;

long duration;
int distance;

setup -

pinMode(trigpin, OUTPUT); // sets the trigpin as
                             an output.

pinMode(echopin, INPUT); // sets the echopin as
                            an input.

Serial.begin(9600); // starts the serial communica-
                       tion.

loop -

digitalWrite(trigpin, LOW);
delayMicroseconds(2);
// sets the trigopin on HIGH state for 10microse

digitalWrite(trigpin, HIGH);
delayMicroseconds(10);
digitalWrite(trigpin, LOW);
//Reads the echopin, returns the soundwave travel
 time in microseconds.
duration = pulseIn(echopin, HIGH);
//calculating the distance
 distance = duration x 0.034/2;
```

```
// Prints the distance on a serial monitor
serial.Print ("Distance");
serial.Println(distance);
delay(5000);
```

Applications :-

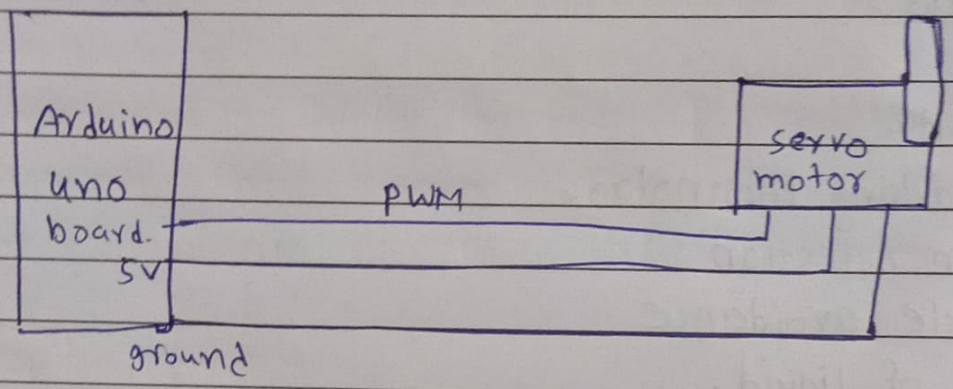
i) Proximity
ii) self parking technology
iii) collision detection
iv) obstacle avoidance
v) levels of liquid.

Conclusion - by using ultrasonic sensor we can measure distance easily with help of arduino.

Practical No :- 6

Aim - To control serve motor using arduino.

Connection diagram :-



Procedure :-

```
#include <servo.h>

Servomyservo ; // create servo object to
control to servo twelve serve objects can be
credited on most boards.

int pos = 0; // Variable to store the servo
position.
```

setup :-

myservo attach (9); // attaches the servo on pin 9 to the servo.

Loop -

for (Pos = 0; Pos <= 180; Pos += 1)

myservo.write (POS;) // tell servo to go to
delay (15); // wait 15 ms for ~~next~~ servo

for (POS = 180; Pos >= 0; POS == 1)

my servo .write (POS); // tell servo to goto position
delay (15); // wait 15 ms for next servo.

int potpin = 0; // analog pin used to connect the potentiometer.

int val; // variable to read the value from the analog pin.

Setup -

myservo.attach (9); // attaches the servo ping to the servo.

loop -

```
Val = analogRead (Potpin); // read the value
val = map (val, 0, 1023, 0, 180); // scale it to use
myservo.write (val); // sets the servo.
delay (15); // waits for 15 ms.
```

Conclusion :- With the use of resistance ~~variable~~ varying the servo motor starts and stops rotating with the use of arduino.

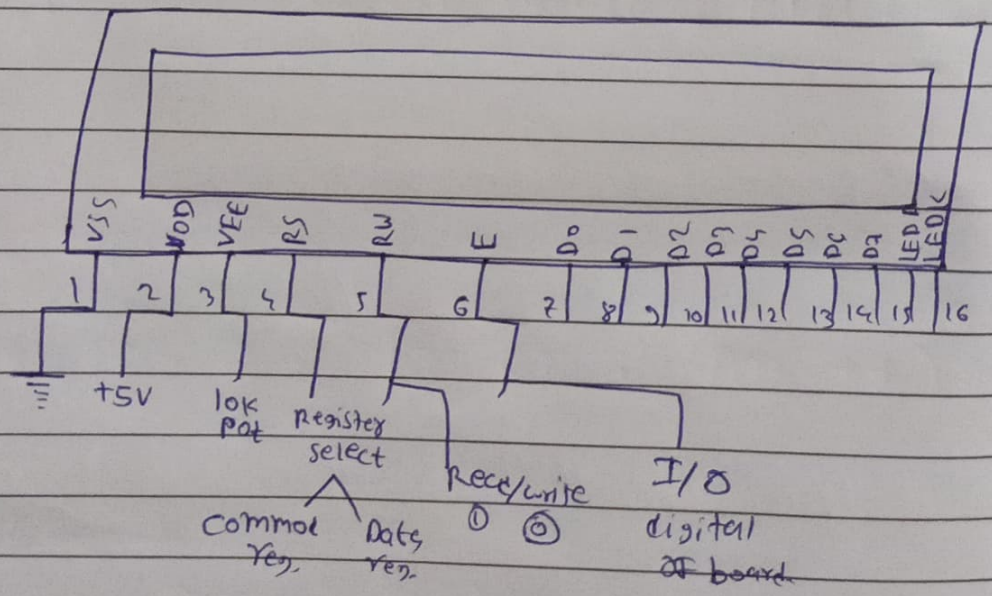
Applications :-

i) Cameras, antenna, Robotic, Textile industry
ii) Robotics - Robotic arms.
iii) Conveyer belts.
iv) Robotic vehicles.
v) Solar tracking system.

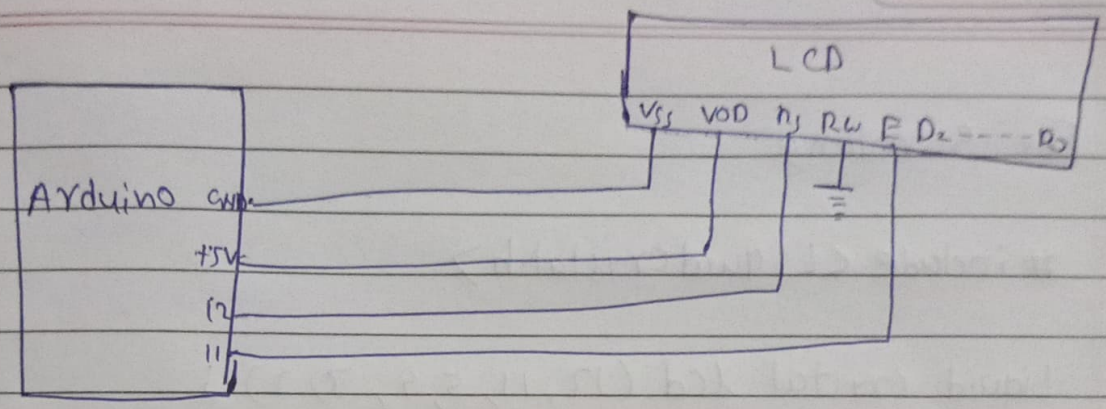
Practical No. 7

To interface LCD module with Arduino UNO.

LCD 16 X 2 (16 column, 2 rows)
5×8 pixel block each
Operating voltage = 4.7 V to 5.3 V
1mA current without any backlight



Display text msg. "Hello World" on LCD screen.



data		Ard. pin
D4	-	5
D5	-	4
D6	-	3
D7	-	2

CODE

```
#include <Liquid Crystal.h>

Liquid Crystal lcd (12, 11, 5, 4, 3, 2);

Void setup() {

  lcd.begin(16,2);
  lcd.print("hello world!");

  }

Void loop {

  lcd.set cursor (0,1);
  lcd.print (millis()/1000);

}
```

For Scrolling

```
#include <LiquidCrystal.h>

Liquid crystal lcd (12, 11, 5, 4, 3, 2);;

Void setup() {

  lcd.begin (16, 2);
  lcd.print ("16 x2 LCD MODULE & ARDUENO UNO")

}

Void loop() {

  for (pos=0 ; POS <2 ; POS +1)

{
  lcd.scroll Disp
  lcd.scrollDisplayLeft();
} delay (500); }
```

Practical No :- 8

Aim :- To study wifi connectivity of arduino.

Procedure :- IEF 802 develope following strandard wifi standard
-802.11

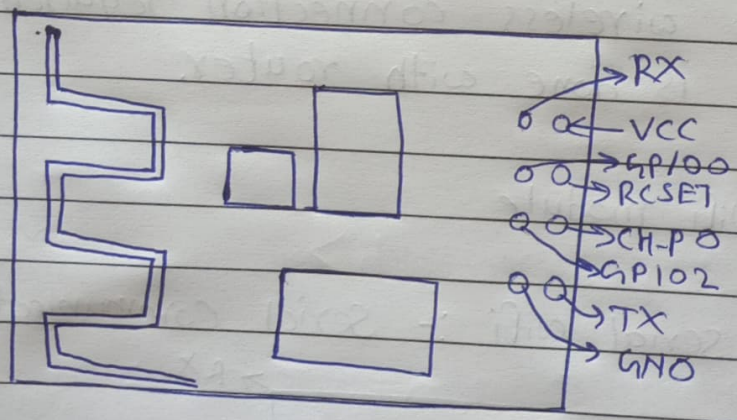
Ethernet - 802.3

Bluetboth - 802.15

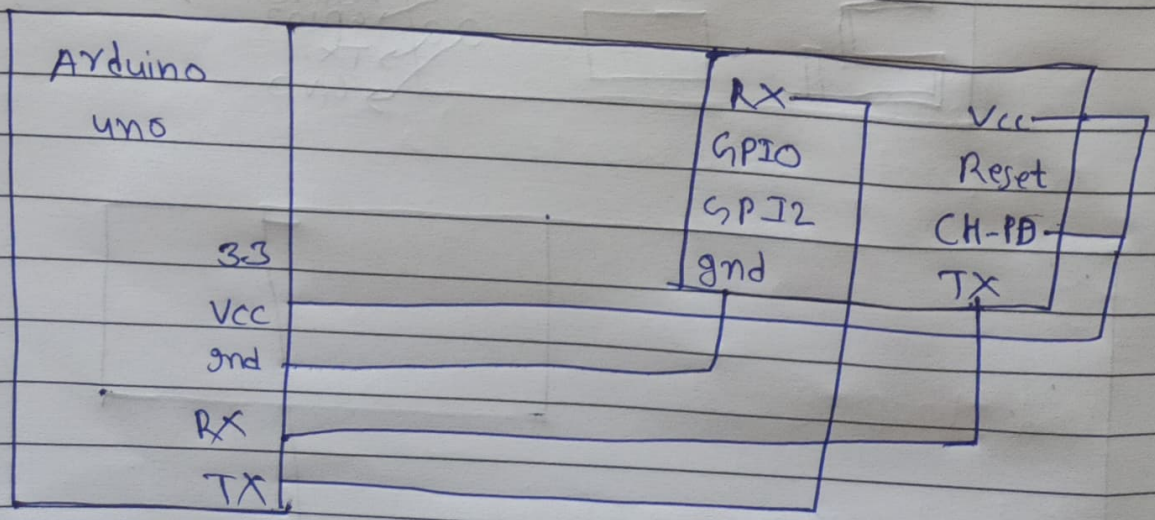
Wifi is a wireless connection required to operate internet which is done with router.

ESP 8266 - Wifi module

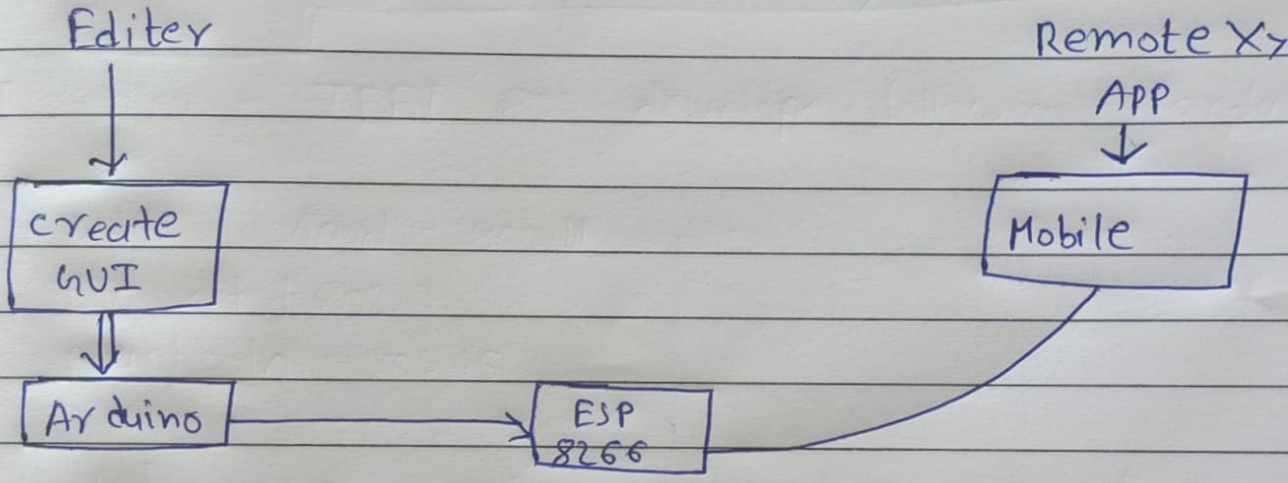
Low cost serial wifi :- serial communication.



ESPP266



Remote XY Arduino control is app to control all devices / appliances present in home like fan, TV, AC etc.



Procedure :-

i) Create GUI using remote XY
ii) Getting source code for GUI
iii) Upload it to arduino ⟺ ESP8266
iv) Get RemoteXY APP (mobile app)
v) Establish connection between APP & wifi.

Conclusion :- We can connect arduino to wifi and done our project by RemoteXY app. Also control appliances of home using this method of connecting wifi to arduino.