Experiment 1

Aim: To blink onboard LED

Procedure:

1. BLink LED is connected to pin 13 of Arduino
2. Initialize pin to I/P or O/P pin
3. If O/P write on pin (LED ON/OFF)

Commands:

1) Pin mode (pin no, output)
Tells arduino whether the mentioned pin is on I/P or O/P.

2) Digital write (pin no, high)

When you are using a pin as an O/P you can command it to high (output 5 volts) or low (output 0 volts).

```
void setup ()
{
// put your setup code here,
 pin mode (13, output); // declare pin 13 as output }

Vold loop() {

// put your main code here, to run digital write (13, high); // Led on

delay (1000);

 digital write (13, low); // Led off

delay (1000);
}
```

Save this and verify code.

Experiment 2

- Arduino interfacing with tri color LED and push button.

Components - 1 Arduino UNO board.
2) Tri color LED
3) Resistors
4) Jumper wires
5) Push buttons

Arduino IDE: Part A.

```
Void setup() {
// put pin mode (10, output); // red Led
pin mode (11,, output); // g.
pin mode (12, output); //
}

void loop () {
// put your main code
digital write (10, HIGH); // red LED ON
delay (500);
digital write (10, LOW), red LED OFF
delay (500);
digital write (11, HIGH); // BLUE ON
delay (500)
digital write (11, LOW); // Blue OFF
delay (500):
Digital write (12, HIGH); // green ON
delay (500);
Digital write (12, LOW); // green OFF
delay (500);
}
```

Add button to pin 4 to board one wire to ground resistor arduino

Add New programme

Experiment 3

Aim: Sensing analog voltage using on-board ADC and printing it on serial monitor

Part A: Program to send data to PC (TX)

Print “Hello World"

1. Set serial commu (serial.begin (baurd rate) bit/sec

2. serial print (......)

```
Void setup() {
serial. begin (9600);
}
void loop() {
serial.print In ("Hello world");
Delay (1000);
```

B. Program Arduino to receive data from PC [RX]

```
  Arduino RX Pin
⬇                   ⬇
Bit O            Bit 1
LED OFF     LED ON
```

1. Set up serial commons.
2. Check availability of data of Rx pin.
   serial available.
3. Read data on Rx serial.read
4. Check received bit 0 ➡ write low on pin 13
   bit 1 ➡ write high on pin 13

Here LED is connected to pin 13

Command

“Serial.begin()" ➡ To set up serial communication

"Serial.available () > 0" ➡ If some data is there.

To check, use the command.

"Serial print In".

Part B:- Sensing analog voltage using onboard ADC and print it on a serial monitor.

- **Some necessary points**:

1. Voltage, current, resistance.
2. Binary - two values - combination of 0 and 1
3. Analog Vs. digital.

⬇

6 pins marked when the valve is as A.

When the value is

⬇ ⬇

0V 5V

bit 0 bit 1

- But when the value is 2.72 V. then this is converted to digital signal.
- This work is done by ADC :- Analog to digital conversion

Eg. 2 bit ADC because levels 2^2 = 4.

5V	11
3.75V	10
2.5V	01
1.25V	00

Step size = (Vmax - Vmin) / level

Analog Value Digital Value
2.3 V ➡ 01
1.75V➡ 00
2.00V➡ 00

2^1	**2^0**		**Decimal**
0	0	➡	0
0	1	➡	1
1	0	➡	2
1	1	➡	3

- Arduino UNO supports ‘10 bit ADC’
- Relating ADC value. to voltage conversion

Resolution of ADC = ADC reading

System Voltage Analog Voltage

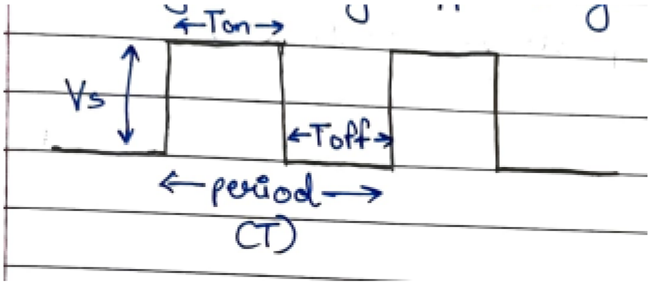
Therefore, 1023/5 = ADC reading/ Analog voltage

Experiment 4

Aim: PWM generation using Arduino.

↳ (Pulse width modulation)

getting analog o/p using digital pins.



Duty cycle: % of time when signal is high

Eg. 50% Duty cycle with 1 Hz

Pins 3, 5, 6, 9, 10, 11 (for PWM) (~)

- Program to change brightness of LDC analog write (generate PWM on given points)

➡ PWM output in arduino is 8-bit value 2^8 = 256

Eg. 100% ➡ 255

30% ➡ 76

Now, $V_{eff} = V_s (T_{on}/T)$

Relations:

PWM output level = 255 x V_{eff}/V_s (:: V_s = 5)

Code:

```
int led - pin = 9;
void setup () {
pin mode (led-pin, OUTPUT); }
void loop () {
analog write (9.0);
0% DC
delay (2000);
```

```
analog write (9, 76); 30%
delay (2000)
analog write (9, 153);
detay (2000)
anolog write (9, 255);
delay (2000);
analog write (9,0); 100%
delay(2000)
```

Put code on proteus by verifying it. (resistor, UNO, LED, oscilloscope)

Dimming from 0 ➡ 100%

for (used for repeating condition)

Code:

```
Int Led pin= 9;
Void setup () {
Pin mode (led - pin, OUTPUT);
}
Void loop () { //fading LED
For (int i= 0; i<255 ; i--) {
Analog write (led pin, i);
delay(100);
}
```

Application: DC motor (Speed control), Surro motor.

Experiment 5

Aim: Interfacing ultrasonic sensor with Ardurio for distance measurement.

Institution of measurements.

(Humans detect light, sound, chems, Temp & Physical)

- **Ultrasonic sensors:**

Using ultrasonic waves (refle, transm).

D = s.t.

(HCSR 04) → ultrasonic sensors

4 pins [V_{cc}, Ground, trigger (for trans), Echo (receiver)]

eg:- D=10 cm speed 340 m/s = 0.034 cm/microsecond

D=s.t   t = D/S = 29.4 micro-second

Therefore, D = s.t 1/2 = 5 cm.

Code:

```
const. int trig Pin = 9;
const. int echo Pin = 10; // define
long duration;
int distance
void setup() {
pin mode (trig fin, OUTPUT);
pin mode (echo Pin, INPUT);
serial. begin (9600); // serial commu. }
void loop () {
digital write (trig Pin, Low);
delay microseconds (2);
digital write (trig Pin, HIGH);
delay microseconds (10);
 // reach echo pin
serial print (" Distance:" ;
delay (5000);
```

Applications: Proximity, people detection, self parking technology (obstacle avoidance), levels of liquid.

Experiment 6

Aim: To control servo control using Arduino.

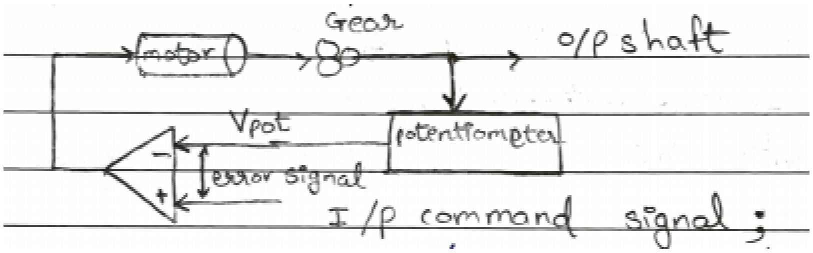
- **Control System**

Eg. Temperature control of room

Design-1	Design-2
Manual control (open loop systems)	Actual temperature= designated temperature; Automated (closed loop) error signal minimized.

- **Servo mechanism**

It has a motor, gears, potentiometer, Intelligent circuits. Here, x degree angle ➡ converts to Vx



$V_{pot} = V_x$ $E_v = 0$ V_{stops} gears stop rotating

Eg. SG90 servo motor.

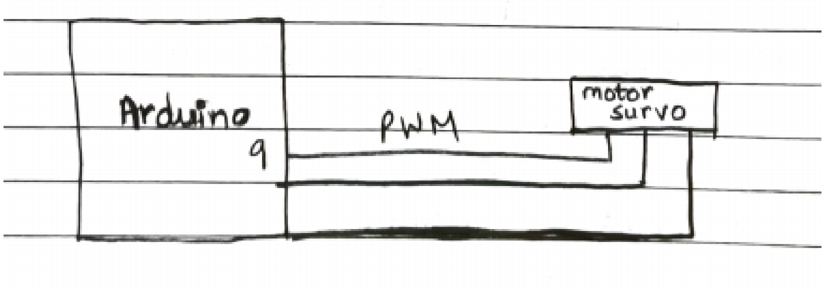
- Consume 10mA (I)[idle] 100mA - 250 mA [rotating]

Here, provide a pulse after every 20ms.

Pulse high for 1 ms servo is 0 degrees

1.5 ms servo is 90 degrees

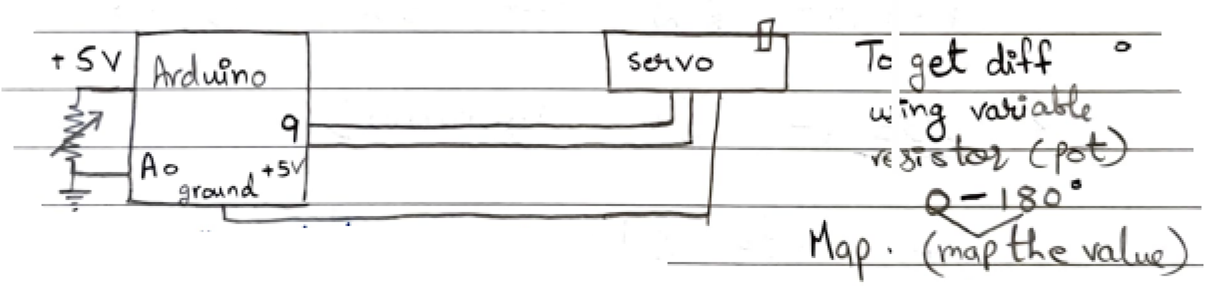
2 ms servo is 180 degrees



Code:

```
#  include <servo.h>
servo int myservo; // create servo object
Int Pos = 0; // variable
void setup () {
myservo. attach (g);
}
void loop() {
for (pos = 0; Pos <180; post =1)
// in steps of degree
myservo.write (POS);
delay (15)
}
for (POS = 180: Pos> = 0; Pos- 1).
myservo.write (POS);
Delay (15)
}
verify & use in proteus.
```

- **<u>There are libraries in IDE</u>**



To get diff °
using variable
resistor (pot)
0-180°
Map. (map the value)

Code:

```
# include <servo.h>
   servo myservo; //create servo object.
   int pot pin = 0;
   int val;
Void setup () {
```

```
my.servo attach (9); }
void loop () {
val = analog Read (pot pin);
val = map ( val, 0, 1023, 0 ,18); //scale it to use it
myservo; write (val); //sets servo.position
}
```

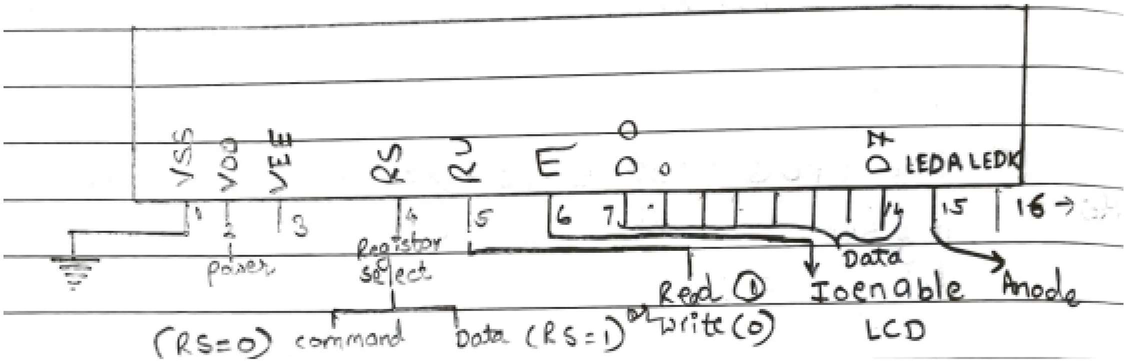
Verify it & use it in proteus.

Experiment 7

Aim: LCD interfacing with Arduino.

- LCD module
  16 columns & 2 rows (ie 16x2 size)
  (Datasheet required) operating range (4.7-5.3)
  LCD each character is in 5 x 8 pixels.
  (eg:- 4 bit or 8 bit mode).

- LCD pinout.



Display + Text "Hello world" on LCD(4 bit) RS=12, RW = ground, E= 11,D4➡5 to D7➡ 2

Code: (use libraries)

```
#include < liquid Crystal.h>

liquid crystal LCD (12, 11, 5, 4, 3, 2);
void setup() {
lcd. begin (16, 12); // set LCD No. of e & R
lcd.print ("Hello, World "); (Print msg)
}
```

```
Void loop () {
// set cursor to column ‘0’; line 1
// (not : line 1 is 2nd row, because counting begins at 0)
Lcd.set cursor (0.1);
// lcd.print (millisecond()/1000); // number sec because reset
}
```

Verify use

- **To scroll display (code)**

```
# include <liquid Crystal .h>
liquid crystal LCD (12, 11, 5, 4, 3, 2);
void setup() {
lcd begin (16, 2);
lcd.print (" 16x 2 LCD module & ARDUINO UNO");

Void loop()
{
        For (pos=0; pos<2; pos++)
        {
        lcd.scroll Display left (); // scroll display
        left
        }
delay (500);
}
```