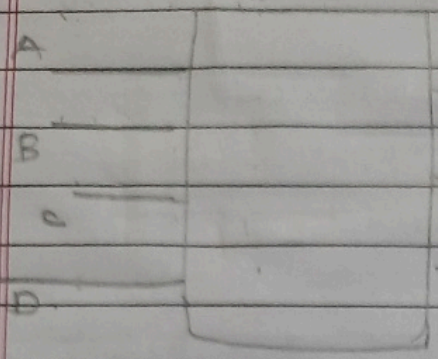
Combinational Circuits

* ~~For~~ sequential there is feedback but in this there is not feedback.



OR
NOR
AND
NOT
NAND XNOT

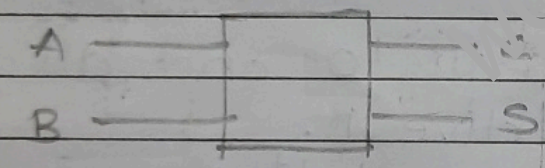
? combination of ~~input~~ u want ? combi. output the combinational circuit used.

* Adder

A
B
0 + 0 = 0 0
0 + 1 = 0 1
1 + 0 = 0 1
1 + 1 = 1 0 ← two bit nor. ...to

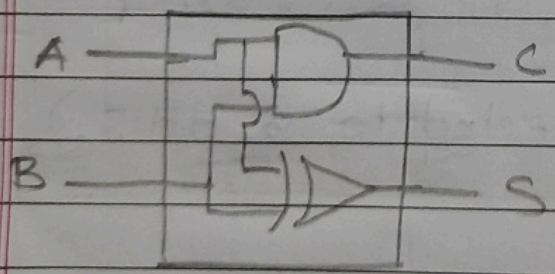
Carry Sum

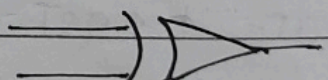
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Truth table OF XOR gat

Design





If both input same o/p
diff. o/p

- It is Half adder.
As it adds 2 bits produces s & c.

```
  1  1 ← A
+ 0  1 ← B
-------
  0
```

$A_1\ A_0$
$+\ B_1\ B_0$
$C_1\ C_0$
$S_1\ S_0$

2 bi
gi
Bdi

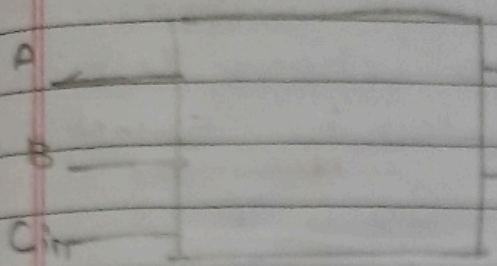
carry coming out from first addition need to taken.

$\frac{1}{2}$ Half Adder OR to

For that,

Truth table FA

- **Full Adder.**



Cin	A	B	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Sum is 1 if odd no^r of bits are 1

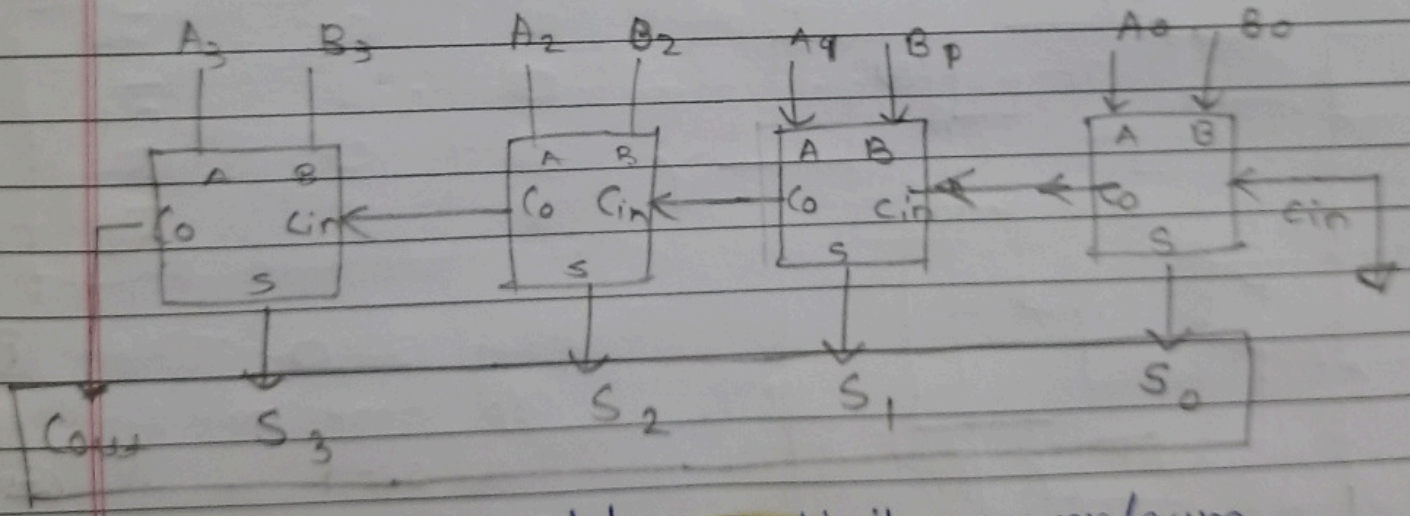
Even no^r of bits 1

Sum 0

If two or more no^r s are 1 then C_{out} = 1

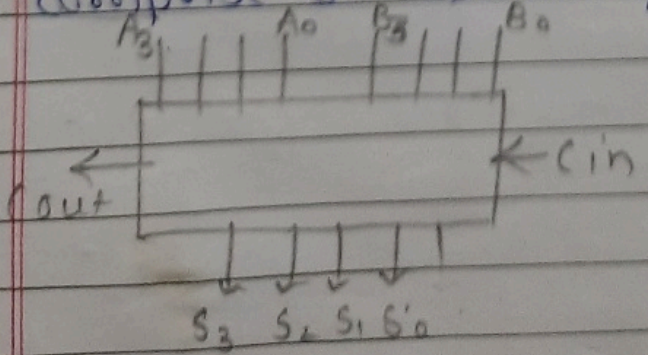
* For 4 bit adder total 4 F.A

```
  0 1 0 1  <- A
+ 0 1 0 1  <- B
-------
  1 0 1 0
1 0 0 1 0
```



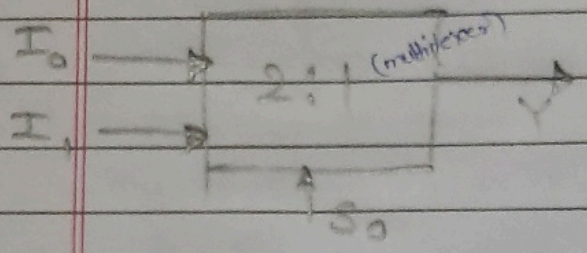
4 bit binary adder. 5 digit answer/sum.

comprise of (2) 4 bit i/p 1 4bit o/p sum Cin, Cout.

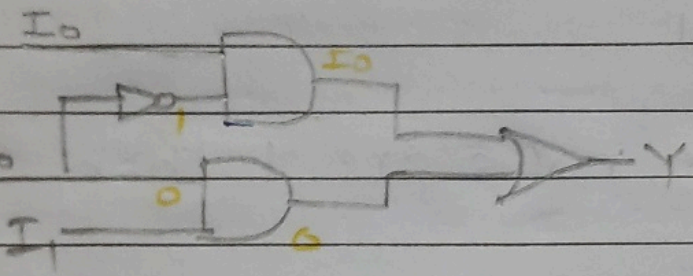


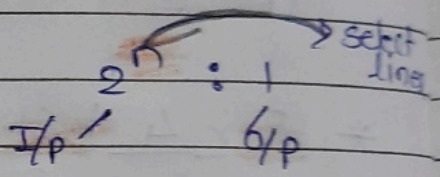
* Multiplexer. (Mutiple i/p)

- More that one inputs & one output.
- ↳ can be set into scenario where which i/p occur on output is selected by Select line

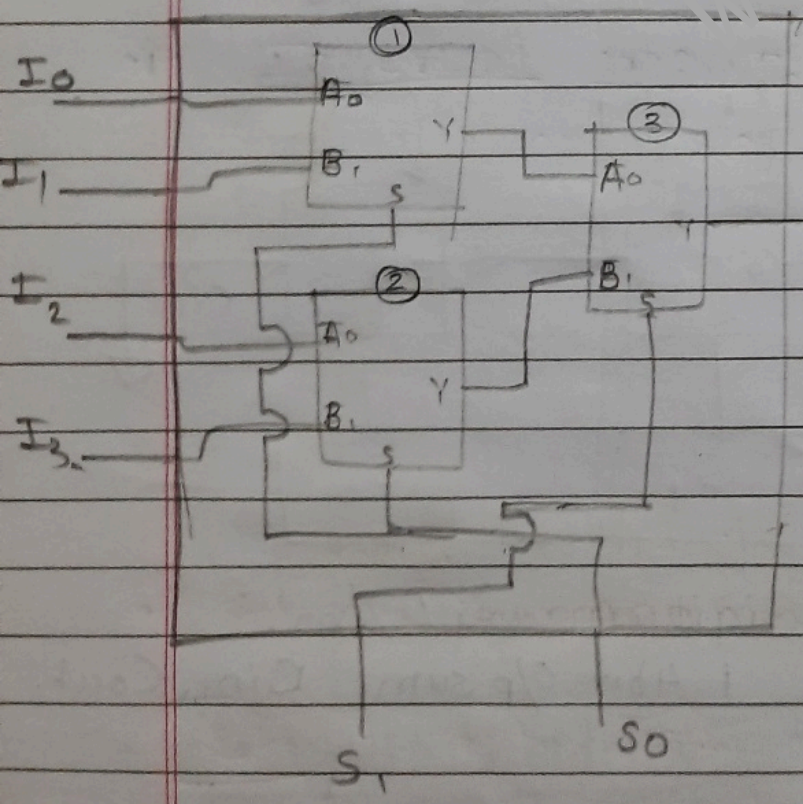


S_0	Y
0	I_0
1	I_1





* Design 4 : 1 multiplexer.



S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

When $S_1 = 0$ Multiplex. (1) st
O/p comes one Final O/p
$S_1 = 1$ (2nd)

* Aurduino Board.

(1) Power USB ⇒ connect Aurdino to PC.
(2) Power (Barrel Jack) ⇒ Ext. power adapter.
(3) Voltage regulator
(4) Crystal Osc. → det. speed of micro.
(5) 17 Reset pins ⇒ onbord reset
(6) 7, 8, 9 supply pins.
(10) Analog pins ⇒ 6 analog inputs.
(11) Microcontroller ⇒
(13) Power LED indicator. ⇒ Blink when connected.
(14) Tx and Rx pins. ⇒ set up LED → the code given.
(15) Digital I/O

* ARDUINO PROGRAMIN LANGUAGE.

- Aurduino IDE

Control statements	Operators.	Functions.
if		PinMode()
if..else.	+, -, *, /,	digitalWrite()
For	<, >, <=,	digitalRead()
while	>=, <>,	delay()
do..while	&&, ||, !	analogRead()
switch..case		analogWrite()

Coding (Aurduino File = called sketch)

```
void setup () {
    // Put your setup code here, to run once:

void loop () {
    // Put your main code here; to run repeatedly:
}.
```

IDE = Integrated development Environment.

* Practical 3 Part A2

• (A)

① — Program to send data to PC [Tx]

Print "Helloo World"

i) Set serial comm. Serial.begin (baurdrate)

ii) serial.print (.....)

```
Void setup () {
serial.begin (9600);
}

Void loop () {
Serial.println ("Hello World");
Delaw (1000);
```

Verify code (compiling)

Then use (Sketch) to see virtual UNO board

② — Programm to Reciere data from PC. [Rx]

Arduino Rx pin

Bit (0) Bit (1)

LED (OFF) LED (ON)

i) Set up sereal commni

ii) check availibilty of data on Rx pin.

serial.available

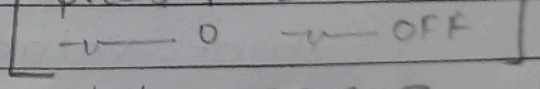
iii) Read data on Rx Serial.read.

iv) check recieved bit 0 → write low on pin 13

bit 1 → high

```
void setup
int led = 13;
int value = 0;
void setup() {
serial.begin(9600);
pinMode(led, OUTPUT);
}
void loop() {
if (serial.available() > 0)
{
  value = serial.read();
  delay(5);
  if (value == '1') {
  digitalWrite(led, HIGH);
  Serial.println("LED is ON");
}
  if (value == '0') {
  digitalWrite(led, LOW)
  Serial.println("LED is OFF");
}
}
```

verify (compile)



Use virtual ... use resister 220Ω

Take path.

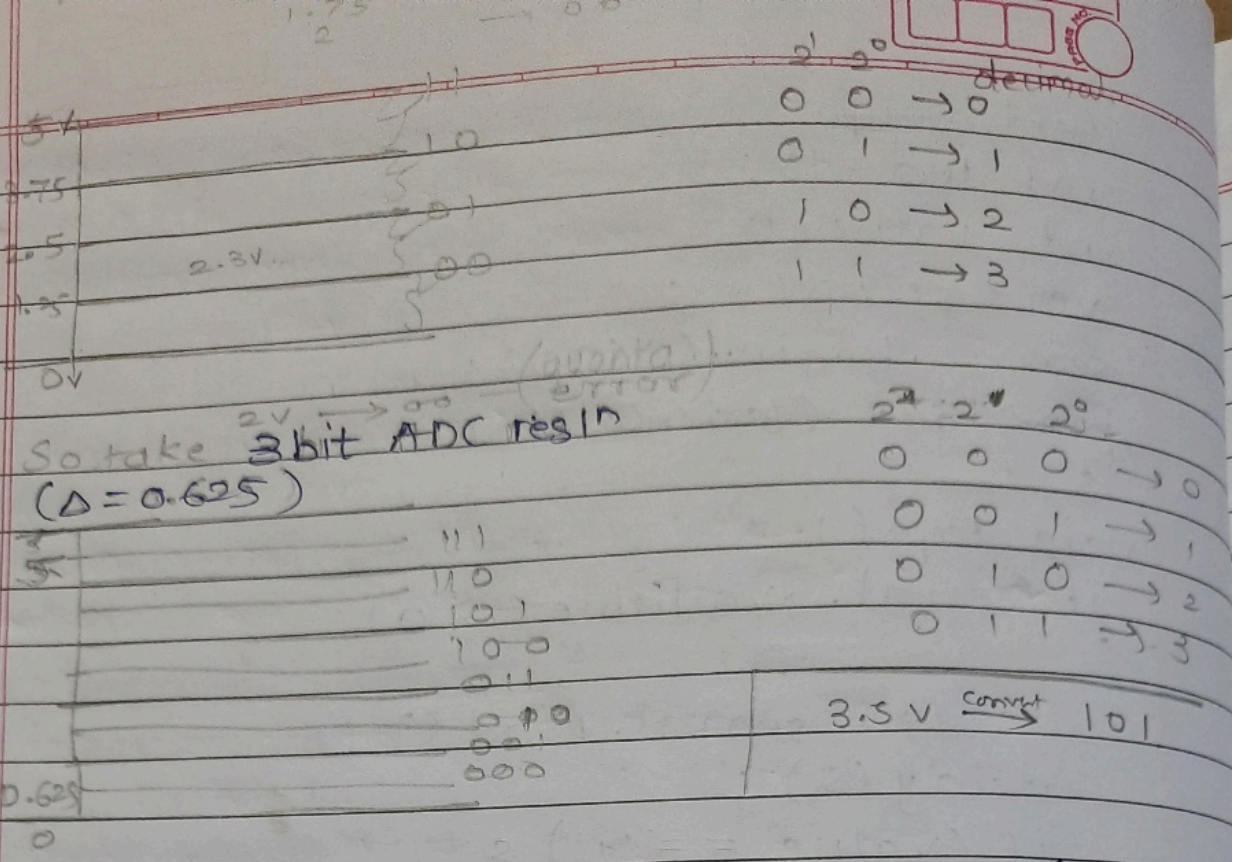
* **Part B**

③. Sensing analog voltage using onboard ADC & print it on serial monitor. (analog to digital conv)

- Binary (0, 1) (0v, 5v)
- [ADC]

e.g 2-bit ADC. → Resol^n $(2)^2 = 4$ levels. Δ = stepsize $= \frac{V_{max} - V_{min}}{levls.}$

$\Delta = \frac{5}{4} = 1.25$



So take 3bit ADC resln
($\Delta = 0.625$)

* Aurduino UNO supports 10-bit ADC ∴ (1024 level
- Relative ADC value to Voltag conversion.

$$\frac{\text{Resl}^n \text{ of ADC}}{\text{System V}} = \frac{\text{ADC Reading}}{\text{Analog V measured.}}$$

e.g. $\frac{1023}{5} = \frac{\quad}{\quad}$

(consider Ana V = 2.12v)

$\therefore$ ADC (x) = 434

Anar. Votage of 2.12V
Dig. value → 434

Code (IN IDE)

~~Void setup () {~~

Steps (i) set se.comm (ii) Read V at pin 3 (A3)
(iii) Print. (serial commu)

Commands.

~~Void setup (){~~

* Analog Read (Ao)

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

```
void loop () {
~~analogRead (A3);~~
int adc_reading = analogRead (A3);
Serial.println ('adc:_reading);
delay (1000)
```

verify

Now to in schematic (POT HG) required setup it and (use code path)

(Here analog converted to digital.)

To get again analog value change code. usin
Formulae. (analog voltage

```
Float voltage = adc_reding * (5.0/1023);
Serial.println (~~float~~ voltage);
}
```

~~Appl~~ ~~sensors~~.

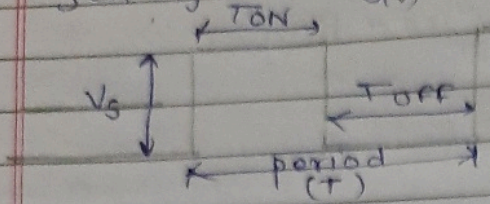
$\frac{1024}{5}$. $\frac{1023}{5} = 100$

DATE 10/5/21

• Practical No: 4. •

PWM Generation Using Arduino.

- PWM (Pulse width modulation) getting analog o/p (v) using digital pins.



Duty cycle - % of time when sig is high

e.g. 50% Duty cycle with 1 Hz.

→ Pins 3, 5, 6, 9, 10, 11 (For PWM) ('~')

① • Program to change brightness of LED analog Write (generate PWM on given pins)

— PWM o/p in Arduino is 8-bit value $2^8 = 255$ (0 → 255)

e.g 100% → 255

30% → ? = 76

Now, $V_{eff} = V_s \times \frac{T_{ON}}{T}$

Relations:

PWM o/p level = $255 \times \frac{V_{eff}}{V_s}$ (∵ $V_s = 5$)

(0 → 5v, 0 → 255)

Code.

```
int led_pin = 9;
void setup() {
pinMode (led_pin, OUTPUT);
}
void loop () {
analogWrite (9,0);   0% D.C.
delay (2000);
analogWrite (9,76);  30%
delay (2000)
analogWrite (9,153);
(9,255);   100%
(9,0);   0
delay (2000)
```

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

Part C Dimming from 0 → 100%

— For (used for repeating condition)

B) Code

```
int led_pin = 9;
void setup () {
pinMode (led_pin, OUTPUT);
}
void loop () {   // Fading LED
for (int i=0; i<255; i++) {
analogWrite (led_pin, i);
delay (100);
}
for(int i=255; i>0; i--) {
}
```

(verify code and use in proteus)

Applic: DC motor (speed control), surva motor

100 → 1024
40

1023 → 5
512

Practical No. 05

- Interfacing ultrasonic sensor with Arduino For Disance Measurement.

① Intituition of measurments.
(Humans Detect ligh, sound, chems, Temp & P.
physical qun.] Sensors
— temp ↓ Distance.
— pressure
— Position
— Motion

Measurable qus
— V
— I
— P

* Ultrasonic Sensors.
→ Using ultrasonic waves (refle, transm) Rx Tx
D = S·t
(HCSR 04) – Ultrasonic sensors.
4 pins [Vcc, ⏚, Triger (for trans), Echo (receiver)
Triger - High For 10us. Echo ⇒ o/p time in usec.

Ex. D = 10 cm. Speed = 340 m/s. = 0.034 cm/us
D = S·t t = D/S = 294 us
Now to get dist.
D = St 1/2 ∴ D = 5 cm.

Code

```
const int trigPin = 9;
const int echoPin = 10;     // define
long duration;
int distance;
void setup() {
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
Serial.begin(9600);  // serial commu.
}
void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
```

```
digitalWrite (trigPin, HIGH);
delayMicroseconds (10);          //start us
digitalWrite (trigPin, LOW);
// Reach echopin.
duration = pulseIn (echoPin, HIGH);
distance = duration * 0.034/2;
Serial.print ("Distance:");
Serial.println (distance);
dely (5000);
```

verify. (use in proteus)

// (use in Resistance Dista use)

Appli : proximity,

- people detection
- self parking tech
- collision detection, - Obstable avoidance.
- levels of liquid.

Practical No. 6.

* Servo motor control using Arduino.

* Control system.

e.g Temp controller of room.

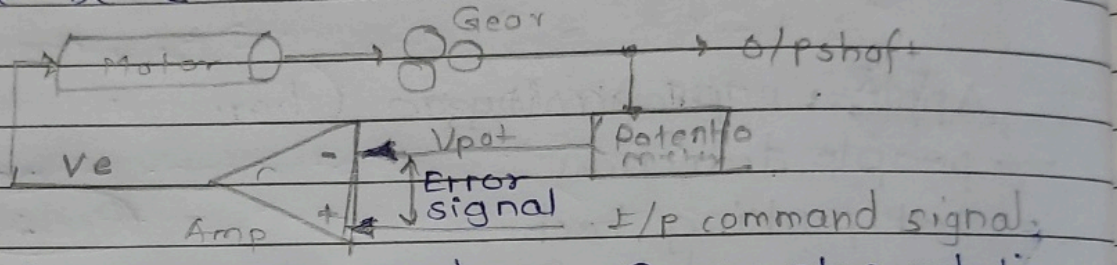
Design I
- Manual control (open loop systems)

Design II
Actual temp = Designed temp.
→ power off
Automated. (closed loop)
error signal minimized

* Servomechanism

It has motor, Gears, Potentiometer, Intelli. circuits, (Amp)

Here x^0 ∠ie $\xrightarrow{\text{conv}}$ V_x



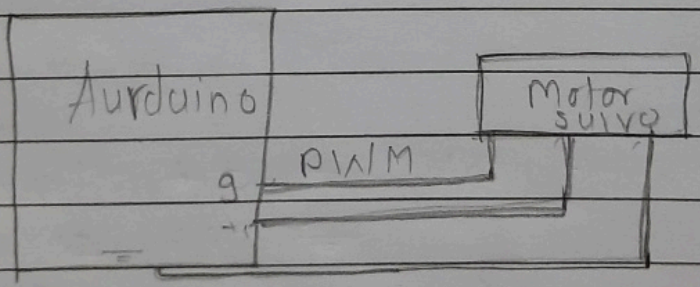
$V_{pot} = V_x$ ∴ Err. S. = 0 V stops. Gears stop rotating.

e.g. SG90 servo motor.

- consume 10 mA (I) [idle] 100 mA - 250 mA (Rotating)

Here, provide pulse after every 20 ms.

Pulse high for	Servo is
1 ms	0°
1.5 ms	90°
2 ms.	180°



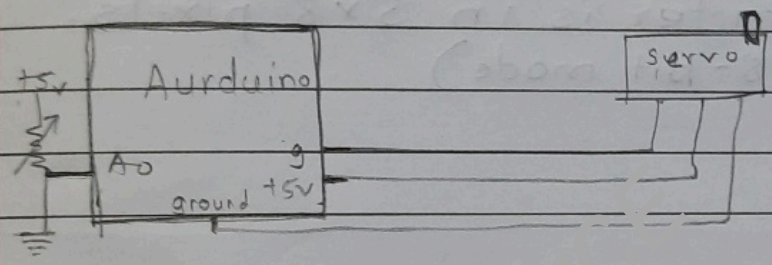
Code

```
# include < Servo.h >
Servo myservo;  // create servo object.
int pos = 0;    // variable.
void setup () {
myservo.attach (9);
}
```

```
void loop () {
  For (pos = 0 ; pos < 180 ; pos+ = 1)
  // in steps of degree.
  myservo.write (pos);
  delay (15)
}
  For (pos = 180 ; pos >= 0 ; pos - = 1)
  myservo.write (pos);
  delay (15)
}
```

verify and use in proteus.

* // There are libraries in IDE //



To get diff •° using variable resistor. (potpi-

~~pot~~ 0 — 180°

Map (map the value)

Code :

```
int potpin = 0;
int val;

void setup () {
  myservo.attach (9);
}
Void loop () {
  val = analogRead (potpin);
  val = map (val, 0, 1023, 0, 18); // scale i
                                   to use i-

  myservo.write (val);  // sets servo position
  delay (15)
}
```

Verify use it in proteus.

Appli
- Camera, antenna, robotics (arms)
- conveyor belts, Robotic vehicles.
- solar tracking system.

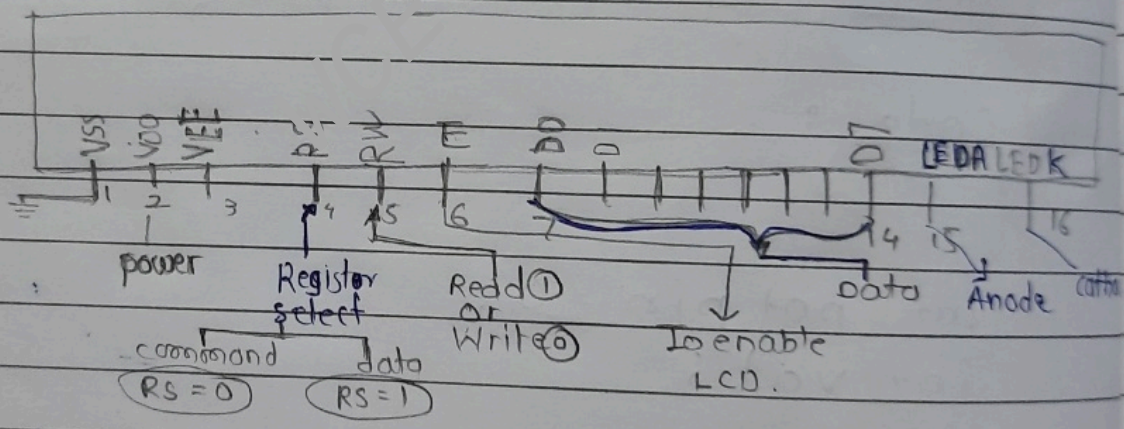
* Practical 7.

- LCD interfacing with Aurduino.

* LCD module.

16 columns & 2 rows. (i.e 16×2 size)
(Datasheet required) operating range 4.7V ~ 5.3V
LCD each character is in 5×8 pixels.
(e.g (4bit) or 8-bit mode)

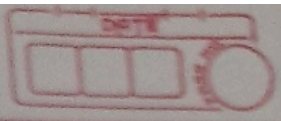
* LCD pinout



* Display + text "Hello World" on LCD. (4bit)
RS = 12 RW = ⏚ E = 11 D4 → 5 to D7 → 2

* CODE (use libraries)

```
# include <Liquid Crystal.h>
Liquid crystal lcd (12, 11, 5, 4, 3, 2);
~~voi~~.
void setup () {
 lcd.begin (16, 12); // set LCD' No. of C & R
 ~~lc~~lcd.print ("hello, World"); //prin msg)
}
```



```
voidloop () {
// set cursor to column '0', line 1
// (not: line 1 is 2nd row, counting begins at 0)
lcd.setCursor(0,1);
//lcd.print (millis()/1000); // No. Sec. : reset.
}
```

verify use

* To scroll display. (code)

```
void setup () {
  lcd.begin (16,2));
  lcd.print ("16x2 LCD MODULE & ARUINO UNO");
}
void loop ()
{ For(:
   For (pos = 0; pos<2; pos++)
   {
lcd.scrollDisplayLeft ();  // scroll display left
 }
 delay (500);
}
```