



# **Ceylon German Technical Training Institute**

**Micro Control  
&  
Robotics**

## **Student Portfolio**

- Name - .....
- Admission Number - .....
- Course Name - .....
- Course Term - .....



Date -

Topic -

## Table of Contents

01. Pull-up and Pull-down Circuits .....	4
1. Pull-Up Resistors.....	4
2. Pull-Down Resistors (to GND).....	5
02. Simple Colour Light Circuit .....	7
03. Knight Rider Circuit.....	8
04. Turn LED ON and OFF Using One Push Button .....	9
05. Analog Input Value Check.....	10
05.1. Code for Check Analog Input Value .....	10
06. Volume Level Controller with Potentiometer.....	11
07. Liquid Crystal Display LCD.....	12
07.1. LCD I2C (I <sup>2</sup> C).....	12
07.2. I2C Address Chart .....	12
08. Set Cursor Selection.....	13
08.1. For 16x2 Display .....	13
08.2. For 20x4 Display .....	13
09. Battery Charging Indicator with I2C and Potentiometer .....	14
10. Horizontal Battery Charging Indicator with I2C and Potentiometer.....	16
11. Ultrasonic Sensor.....	18
12. Ultrasonic Distance Count .....	18
13. Ultrasonic Distance Count with LED Indicators .....	19
14. Seven Segment Display.....	20
15. Combining Two Seven Segment Displays .....	22
16. Joystick Module .....	28
17. Control LED's Brightness Using Joy Stick .....	29
18. Pulse Width Modulation (PWM) .....	31
19. Motor Driver L298N.....	32
20. Control Motor with Joy Stick Using L298N Motor Driver .....	34
21. Control Motor Using Potentiometer .....	34
22. Control Speed of the Motor Using L298N Motor Driver .....	34

Signature -

Page No - 1



# Ceylon German Technical Training Institute

## Micro Control & Robotics (MCR)

### Student Portfolio

Date -	Topic -
23. Control Simple Robot Car's Two Motors .....	35
24. Data Types .....	36
24.1 byte.....	36
24.2. int.....	36
24.3. long.....	37
24.4. float .....	37
24.5. bool.....	38
24.6. double.....	38
24.7. char.....	38
24.8. string.....	39
24.9. Array .....	39
25. Servo Motor.....	42
26. Servo Motor.....	43
27. Servo Motor with Potentiometer .....	43
28. Servo Motor with Push Button .....	43
29. Relay Module.....	44
30. LDR Sensor Module .....	46
31. IR Sensor Module .....	47
32. Key Pad .....	49
33. Key Pad with Password .....	50
34. DHT22/11 Temperature and Humidity Sensor .....	52
35. DHT22 Sensor with I2C Display .....	53
36. Stepper Motor .....	54
37. Brushless DC Motor .....	56
38. Arduino Software Serial.....	57
39. Bluetooth Module .....	58
40. Communication of 2 UNO Boards .....	60
40.1. Master Arduino's Code .....	60
40.2. Slave Arduino's Code .....	61
40.3. Control LED Using 2 Arduino Boards as Master and Slave .....	62
41. Encoder Module .....	63
41.1. Encoder and Counter.....	65
41.2. Encoder 1-5 Count.....	66

Signature -

Page No - 2



# Ceylon German Technical Training Institute

## Micro Control & Robotics (MCR)

### Student Portfolio

Date -

Topic -

41.3. Encoder with Counter and LCD .....	68
42. Arduino UNO Board .....	70
42.1. Digital Input/Output Pins .....	71
42.2. Analog Input Pins.....	71
42.3. Microcontroller.....	71
42.4. Arduino Uno Board Specifications.....	72
42.5. Logical Operators.....	72
1. For Loop -.....	73
2. While Loop - .....	74
43. Electrically Erasable Programmable Read-Only Memory (EEPROM) .....	75
44. Gyroscope.....	76
45. Real-Time Clock Module (RTC) .....	78
46. Load Cell Sensor .....	80
47. PCB Designing .....	82
47.1. Rectifier Circuit .....	82
47.2. Arduino Board Design Using ATmega 328p.....	83

Signature -

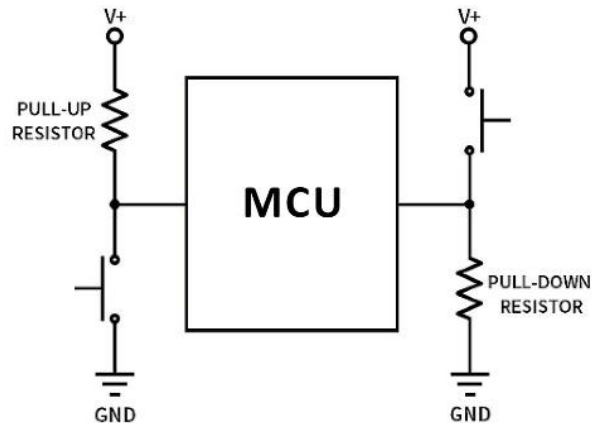
Page No - 3



Date -

Topic -

## 01. Pull-up and Pull-down Circuits



- ❖ Pull-up and pull-down resistors are used in digital circuits to ensure that a pin or signal line is at a defined logic level (HIGH or LOW) when it is not actively driven by another component.

### 1. Pull-Up Resistors

- Connected between the signal line and VCC.
- Ensures the signal remains HIGH when no other device is actively pulling it LOW.

Example: A push button with a pull-up resistor

When the button is not pressed, the resistor pulls the input to HIGH.

When the button is pressed, it connects to GND, making it LOW.

Signature -

Page No - 4



Date -

Topic -

## 2. Pull-Down Resistors (to GND)

- Connected between the signal line and GND.
- Ensures the signal remains LOW when no other device is actively pulling it HIGH.
- Used in circuits where a switch or sensor needs to default in LOW.

Example: A push button with a pull-down resistor:

When the button is not pressed, the resistor pulls the input to LOW.

When the button is pressed, it connects to VCC, making it HIGH.

### With Pull-up resistor

```
3.ino
1 int value;

Output Serial Monitor X
Not connected. Select a board and a port to connect automatically.

11:22:24.150 -> 1
11:22:24.150 -> 1
11:22:24.150 -> 1
11:22:24.150 -> 1
11:22:24.150 -> 1
11:22:24.150 -> 1
11:22:24.150 -> 1
11:22:24.150 -> 1
11:22:24.150 -> 1
11:22:24.150 -> 1
11:22:24.183 -> 1
11:22:24.183 -> 1
11:22:24.183 -> 1
11:22:24.183 -> 1
11:22:24.183 -> 1
11:22:24.183 -> 1
11:22:24.183 -> 1
11:22:24.183 -> 1
11:22:24.183 -> 1
11:22:24.183 -> 1
11:22:24.214 -> 1
11:22:24.214 -> 1
11:22:24.214 -> 1
11:22:24.214 -> 1
11:22:24.214 -> 1
11:22:24.214 -> 1
```

### With Pull-down resistor

```
3.ino
1 int value;

Output Serial Monitor X
Not connected. Select a board and a port to connect automatically.

11:21:27.753 -> 0
11:21:27.753 -> 0
11:21:27.753 -> 0
11:21:27.753 -> 0
11:21:27.753 -> 0
11:21:27.753 -> 0
11:21:27.753 -> 0
11:21:27.753 -> 0
11:21:27.785 -> 0
11:21:27.785 -> 0
11:21:27.785 -> 0
11:21:27.785 -> 0
11:21:27.785 -> 0
11:21:27.785 -> 0
11:21:27.785 -> 0
11:21:27.785 -> 0
11:21:27.785 -> 0
11:21:27.785 -> 0
11:21:27.785 -> 0
11:21:27.817 -> 0
11:21:27.817 -> 0
11:21:27.817 -> 0
11:21:27.817 -> 0
11:21:27.817 -> 0
11:21:27.817 -> 0
```

Signature -

Page No - 5



Date -

Topic -

- ❖ The code use for check current state of push button is showing bellow.

With physical pull-up/down Resistor

```
3 | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Arduino Uno
3.ino
1 int value;
2
3 void setup() {
4   pinMode(4, INPUT);
5   Serial.begin (9600);
6 }
7
8 void loop() {
9   value=digitalRead(4);
10  Serial.println(value);
11 }
12
13
14
```

With enabling internal pull-up Resistor

```
3 | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Arduino Uno
3.ino
1 int value;
2
3 void setup() {
4   pinMode(4, INPUT_PULLUP);
5   Serial.begin (9600);
6 }
7
8 void loop() {
9   value=digitalRead(4);
10  Serial.println(value);
11 }
12
13
14
```



Date -

Topic -

## 02. Simple Colour Light Circuit

```
#define time1 4000
#define time2 2000
#define time3 8000

void setup() {
  pinMode (13, OUTPUT); //Red
  pinMode (12, OUTPUT); //yellow
  pinMode (11, OUTPUT); //Green
  pinMode (10, OUTPUT);
}

void loop() {

  digitalWrite (10,1);

  digitalWrite (13, 1);
  delay (time1);
  digitalWrite (12, 1);
  delay (time2);

  digitalWrite (13, 0);
  digitalWrite (12, 0);

  digitalWrite (11, 1);
  delay (time3);
  digitalWrite (11, 0);

  digitalWrite (12, 1);
  delay (time2);
  digitalWrite (12, 0);

}
```

Signature -

Page No - 7





Date -

Topic -

### 03. Knight Rider Circuit

```
int b = 3;

void setup() {
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  Serial.begin (9600);
}

void loop() {

  while (b < 9) {
    digitalWrite (b, 1);
    digitalWrite (b-1, 0);
    b++;
    delay(50);
  }

  while(b > 2) {
    digitalWrite(b, 0);
    digitalWrite (b-1, 1);
    b--;
    delay(50);
  }

}
```

Signature -

Page No - 8



Date -

Topic -

## 04. Turn LED ON and OFF Using One Push Button

```
//LED ON/OFF using one push button
```

```
int a;
int b=1;

void setup() {
  Serial.begin(9600);
  pinMode(4, INPUT);
  pinMode(13, OUTPUT);
}

void loop() {

  a=digitalRead(4);
  Serial.print("a=");
  Serial.print(a);
  Serial.print("    b=");
  Serial.println(b);

  if ((a==1)&&(b==1)) {
    digitalWrite(13, 1);
    b=0;
    delay(1000);
  }

  else {
    if((a==1)&&(b==0)) {
      digitalWrite(13, 0);
      b=1;
      delay(1000);
    }
  }
}
```

Signature -

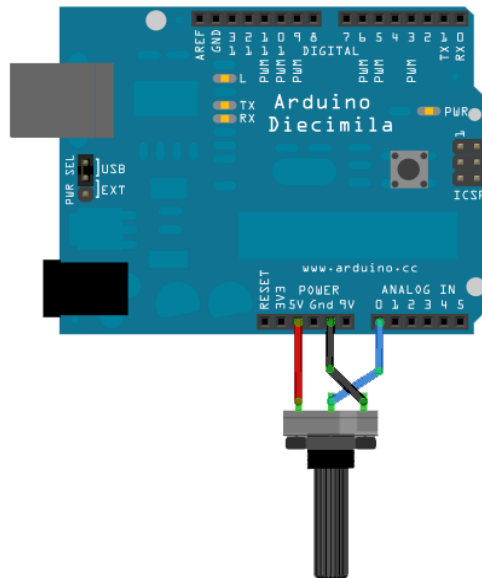
Page No - 9



Date -

Topic -

## 05. Analog Input Value Check



### 05.1. Code for Check Analog Input Value

```
//Analog input value checking

int a;

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

void loop() {

  a = analogRead(A0);
  Serial.println(a);
}
```

Signature -

Page No - 10



Date -

Topic -

## 06. Volume Level Controller with Potentiometer

```
//volume level controller with  
POT  
  
int b;  
int a;  
  
void setup() {  
  pinMode(3, OUTPUT);  
  pinMode(4, OUTPUT);  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
  pinMode(7, OUTPUT);  
  pinMode(8, OUTPUT);  
  Serial.begin(9600);  
}  
  
void loop() {  
  
  b = analogRead(A0);  
  a = map(b,1023,0,9,2);  
  Serial.print(a);  
  Serial.println(b);  
  
  digitalWrite(a, 1);  
  digitalWrite(a-1, 0);  
}
```

Signature -

Page No - 11

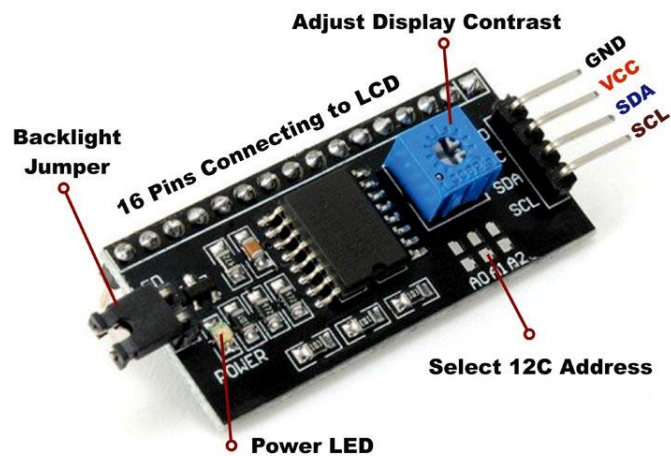


Date -

Topic -

## 07. Liquid Crystal Display LCD


### 07.1. LCD I2C (I<sup>2</sup>C)





### 07.2. I2C Address Chart


 = 0x27  
A0 A1 A2


 = 0x26  
A0 A1 A2


 = 0x25  
A0 A1 A2

 = 0x24  
A0 A1 A2

 = 0x23  
A0 A1 A2

 = 0x22  
A0 A1 A2

 = 0x21  
A0 A1 A2

 = 0x20  
A0 A1 A2



Date -

Topic -

## 08. Set Cursor Selection

### 08.1. For 16x2 Display

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															

Example:

If I Want to select the yellow colour location I can simply write

setCursor (5,1);

### 08.2. For 20x4 Display

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1																			
2																			
3																			

Signature -

Page No - 13



Date -

Topic -

## 09. Battery Charging Indicator with I2C and Potentiometer

```
Battery_charging_Indicator_with_I2C_and_Potentiometer
File Edit Sketch Tools Help
[Icons] Arduino Uno
Battery_charging_Indicator_with_I2C_and_Potentiometer.ino
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3 int value;
4 int a;
5 int b;
6 int c;
7
8
9 byte S0 [8]={
10     B01110,
11     B11011,
12     B10001,
13     B10001,
14     B10001,
15     B10001,
16     B11111,
17     B11111
18 };
19
20
21 byte S1 [8]={
22     B01110,
23     B11011,
24     B10001,
25     B10001,
26     B10001,
27     B10001,
28     B11111,
29     B11111
30 };
31
32 byte S2 [8]={
33     B01110,
34     B11011,
35     B10001,
36     B10001,
37     B10001,
38     B11111,
39     B11111,
40     B11111
41 };
42
```

```
Battery_charging_Indicator_with_I2C_and_Potentiometer
File Edit Sketch Tools Help
[Icons] Arduino Uno
Battery_charging_Indicator_with_I2C_and_Potentiometer.ino
41 };
42
43 byte S3 [8]={
44     B01110,
45     B11011,
46     B10001,
47     B10001,
48     B11111,
49     B11111,
50     B11111,
51     B11111
52 };
53
54 byte S4 [8]={
55     B01110,
56     B11011,
57     B10001,
58     B11111,
59     B11111,
60     B11111,
61     B11111,
62     B11111
63 };
64
65 byte S5 [8]={
66     B11111,
67     B11111,
68     B11111,
69     B11111,
70     B11111,
71     B11111,
72     B11111,
73     B11111
74 };
75
76
77 LiquidCrystal_I2C lcd(0x27,20,4);
78
```

Signature -

Page No - 14



Date -

Topic -

```
78
79 void setup()
80 {
81     lcd.init();
82     lcd.init();
83     lcd.backlight();
84     lcd.createChar(0,S0);
85     lcd.createChar(1,S1);
86     lcd.createChar(2,S2);
87     lcd.createChar(3,S3);
88     lcd.createChar(4,S4);
89     lcd.createChar(5,S5);
90     Serial.begin(9600);
91 }
92
93
94
95 void loop()
96 {
97     lcd.setCursor(0,0);
98     lcd.write(c);
99
100     b=analogRead(A0);
101     a=map(b,966,2,100,0);
102     c=map(a,100,0,1,6);
103
104     lcd.setCursor(3,0);
105     lcd.print(a);
106     lcd.print("%");
107     lcd.print(" ");
108
109     lcd.print("Charging");
110     lcd.print(" ");
111
112 }
113
```

Signature -

Page No - 15





Date -

Topic -

## 10. Horizontal Battery Charging Indicator with I2C and Potentiometer

```
Battery_charging_Indicator_horizontal_with_I2C
File Edit Sketch Tools Help
[Icons] Select Board

Battery_charging_Indicator_horizontal_with_I2C.ino
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3  int value;
4  int a;
5  int b;
6  int c;
7  int d;
8  int e;
9  int i=0;
10
11
12  byte S0 [8]={
13      B00000,
14      B00000,
15      B00000,
16      B00000,
17      B00000,
18      B00000,
19      B00000,
20      B00000
21  };
22
23
24  byte S1 [8]={
25      B10000,
26      B10000,
27      B10000,
28      B10000,
29      B10000,
30      B10000,
31      B10000,
32      B10000
33  };
34
35  byte S2 [8]={
36      B11000,
37      B11000,
38      B11000,
39      B11000,
40      B11000,
41      B11000,
42      B11000,
43      B11000
44  };
45
```

```
Battery_charging_Indicator_horizontal_with_I2C | Ar
File Edit Sketch Tools Help
[Icons] Select Board

Battery_charging_Indicator_horizontal_with_I2C.ino
46  byte S3 [8]={
47      B11100,
48      B11100,
49      B11100,
50      B11100,
51      B11100,
52      B11100,
53      B11100,
54      B11100
55  };
56
57  byte S4 [8]={
58      B11110,
59      B11110,
60      B11110,
61      B11110,
62      B11110,
63      B11110,
64      B11110,
65      B11110
66  };
67
68  byte S5 [8]={
69      B11111,
70      B11111,
71      B11111,
72      B11111,
73      B11111,
74      B11111,
75      B11111,
76      B11111
77  };
78
79
80  LiquidCrystal_I2C lcd(0x27,20,4);
81
82  void setup()
83  {
84      lcd.init();
85      lcd.init();
86      lcd.backlight();
87      lcd.createChar(0,S0);
88      lcd.createChar(1,S1);
89      lcd.createChar(2,S2);
90      lcd.createChar(3,S3);

```

Signature -

Page No - 16



Date -

Topic -

```
Battery_charging_Indicator_horizontal_with_I2C.ino
File Edit Sketch Tools Help
Select Board
Battery_charging_Indicator_horizontal_with_I2C.ino
79
80 LiquidCrystal_I2C lcd(0x27,20,4);
81
82 void setup()
83 {
84     lcd.init();
85     lcd.init();
86     lcd.backlight();
87     lcd.createChar(0,S0);
88     lcd.createChar(1,S1);
89     lcd.createChar(2,S2);
90     lcd.createChar(3,S3);
91     lcd.createChar(4,S4);
92     lcd.createChar(5,S5);
93     Serial.begin(9600);
94
95 }
96
97
98 void loop()
99 {
100
101     b=analogRead(A0);
102     a=map(b,1024,0,100,0);
103     c=map(a,100,0,100,0);
104     d=c/5;
105     e=c-(d*5);
106
107     while (i<d) {
108         lcd.setCursor(i,0);
109         lcd.write(5);
110         i++;
111
112         lcd.setCursor(i,0);
113         lcd.write(e);
114         lcd.print(" ");
115
116         lcd.setCursor(3,3);
117         lcd.write(a);
118         lcd.print("%");
119         lcd.print(" ");
120         lcd.print("Charging");
121         lcd.print(" ");
122     }
123 }
```

Signature -

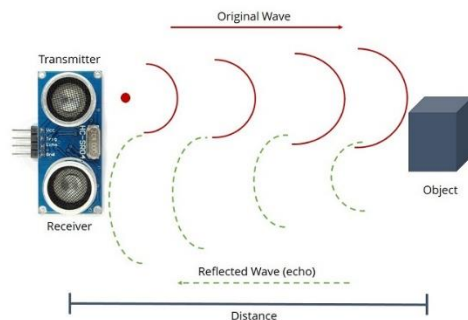
Page No - 17



Date -

Topic -

## 11. Ultrasonic Sensor



- ❖ Ultrasonic sensor calculates the target's distance by emitting ultrasonic sound waves and converting them into electrical signals.

## 12. Ultrasonic Distance Count

```
Ultrasonic_distance_count | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Select Board
Ultrasonic_distance_count.ino
1 int trig=4;
2 int echo=5;
3 long time=0;
4 long cm=0;
5 long inch=0;
6
7 void setup() {
8   pinMode(trig,OUTPUT);
9   pinMode(echo,INPUT);
10  Serial.begin(9600);
11 }
12
```

```
12
13 void loop() {
14
15   digitalWrite(trig,0);
16   delayMicroseconds(2);
17   digitalWrite(trig,1);
18   delayMicroseconds(10);
19   digitalWrite(trig,0);
20   time=pulseIn(echo,1);
21   cm=time/29/2;
22   inch=time/74/2;
23   Serial.print(inch);
24   Serial.print("inch\t");
25   Serial.print(cm);
26   Serial.println("cm");
27 }
28
```

Signature -

Page No - 18



Date -

Topic -

### 13. Ultrasonic Distance Count with LED Indicators

```
Ultrasonic_distance_count_LED | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Select Board
Ultrasonic_distance_count_LED.ino
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3
4 int trig=4;
5 int echo=5;
6 long time=0;
7 long cm=0;
8 long inch=0;
9
10 LiquidCrystal_I2C lcd(0x27,16,2);
11
12 void setup() {
13   pinMode (8,OUTPUT);
14   pinMode (9,OUTPUT);
15   pinMode (10,OUTPUT);
16   pinMode (11,OUTPUT);
17   pinMode(trig,OUTPUT);
18   pinMode(echo,INPUT);
19   lcd.init();
20   lcd.init();
21   lcd.backlight();
22   Serial.begin(9600);
23 }
24
25 void loop() {
26
27   digitalWrite(trig,0);
28   delayMicroseconds(2);
29   digitalWrite(trig,1);
30   delayMicroseconds(10);
31   digitalWrite(trig,0);
32   time=pulseIn(echo,1);
33   cm=time/29/2;
34   inch=time/74/2;
35   Serial.print(inch);
36   Serial.print("\t");
37   Serial.print(cm);
38   Serial.println("cm");
39   lcd.setCursor(2,0);
40   lcd.print(inch);
41   lcd.print("inch");
42   lcd.print(" ");
43   lcd.setCursor(10,0);
44   lcd.print(cm);
45   lcd.print("cm");
46 }
```

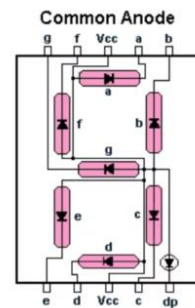
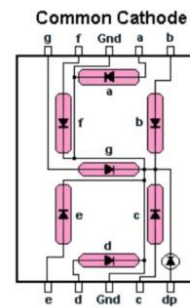
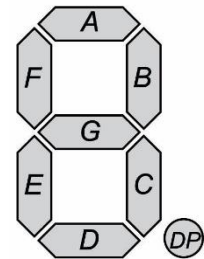
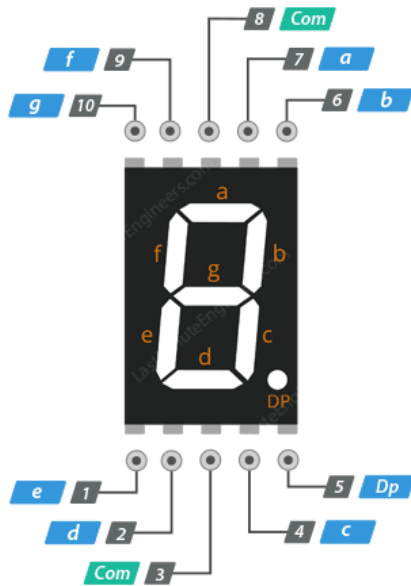
```
44 lcd.print(cm);
45 lcd.print("cm");
46 lcd.print(" ");
47 delay(100);
48
49     if (cm>=10) {
50         digitalWrite(8,HIGH);
51     }
52     else {
53         digitalWrite(8,LOW);
54     }
55     if (cm>=20) {
56         digitalWrite(9,HIGH);
57     }
58     else {
59         digitalWrite(9,LOW);
60     }
61     if (cm>=30) {
62         digitalWrite(10,HIGH);
63     }
64     else {
65         digitalWrite(10,LOW);
66     }
67     if (cm>=40) {
68         digitalWrite(11,HIGH);
69     }
70     else {
71         digitalWrite(11,LOW);
72     }
73 }
74
```



Date -

Topic -

## 14. Seven Segment Display



- ❖ Seven segment display is an electronic display device used to show decimal numbers from 0 to 9. It consists of seven segments.

```
sketch_nov30a-Seven_segment_display | Arduino I
File Edit Sketch Tools Help
Select Board
sketch_nov30a-Seven_segment_display.ino
1 #define time1 1000
2 #define time2 0
3
4 #define a 7
5 #define b 6
6 #define c 5
7 #define d 4
8 #define e 3
9 #define f 2
10 #define g 1
11 #define DP 0
12
13 void setup() {
14
15     pinMode(a,OUTPUT);
16     pinMode(b,OUTPUT);
17     pinMode(c,OUTPUT);
18     pinMode(d,OUTPUT);
19     pinMode(e,OUTPUT);
20     pinMode(f,OUTPUT);
21     pinMode(g,OUTPUT);
22     pinMode(DP,OUTPUT);
23
24 }
25
```

```
24 }
25
26 void loop() {
27
28     zero();
29     delay(time1);
30     one();
31     delay(time1);
32     two();
33     delay(time1);
34     three();
35     delay(time1);
36     four();
37     delay(time1);
38     five();
39     delay(time1);
40     six();
41     delay(time1);
42     seven();
43     delay(time1);
44     eight();
45     delay(time1);
46 }
```

Signature -

Page No - 20



Date -

Topic -

```
sketch_nov30a-Seven_segment_display | Arduino IDE
File Edit Sketch Tools Help
[Icons] Select Board
sketch_nov30a-Seven_segment_display.ino
43 delay(time1);
44 eight();
45 delay(time1);
46 nine();
47 delay(time1);
48
49 eight();
50 delay(time1);
51 seven();
52 delay(time1);
53 six();
54 delay(time1);
55 five();
56 delay(time1);
57 four();
58 delay(time1);
59 three();
60 delay(time1);
61 two();
62 delay(time1);
63 one();
64 delay(time1);
65
66 }
67
68 void zero(){
69     digitalWrite(a,1);
70     digitalWrite(b,1);
71     digitalWrite(c,1);
72     digitalWrite(d,1);
73     digitalWrite(e,1);
74     digitalWrite(f,1);
75     digitalWrite(g,0);
76     digitalWrite(DP,0);
77 }
78
79
80
81
82 void one(){
83     digitalWrite(a,0);
84     digitalWrite(b,1);
85     digitalWrite(c,1);
86     digitalWrite(d,0);
87     digitalWrite(e,0);
88     digitalWrite(f,1);
89     digitalWrite(g,0);
90     digitalWrite(DP,0);
91 }
```

```
sketch_nov30a-Seven_segment_display | Arduino IDE
File Edit Sketch Tools Help
[Icons] Select Board
sketch_nov30a-Seven_segment_display.ino
104 digitalWrite(f,0);
105 digitalWrite(g,1);
106 digitalWrite(DP,1);
107 }
108
109
110 void three(){
111     digitalWrite(a,1);
112     digitalWrite(b,1);
113     digitalWrite(c,1);
114     digitalWrite(d,1);
115     digitalWrite(e,0);
116     digitalWrite(f,0);
117     digitalWrite(g,1);
118     digitalWrite(DP,1);
119 }
120
121
122
123 void four(){
124     digitalWrite(a,0);
125     digitalWrite(b,1);
126     digitalWrite(c,1);
127     digitalWrite(d,0);
128     digitalWrite(e,0);
129     digitalWrite(f,1);
130     digitalWrite(g,1);
131     digitalWrite(DP,1);
132 }
133
134
135
136 void five(){
137     digitalWrite(a,1);
138     digitalWrite(b,0);
139     digitalWrite(c,1);
140     digitalWrite(d,1);
141     digitalWrite(e,0);
142     digitalWrite(f,1);
143     digitalWrite(g,1);
144     digitalWrite(DP,1);
145 }
146
147
148 }
```

Signature -

Page No - 21



Date -

Topic -

```
sketch_nov30a-Seven_segment_display | Arduino
File Edit Sketch Tools Help
[Icons] Select Board
sketch_nov30a-Seven_segment_display.ino
156 digitalWrite(f,1);
157 digitalWrite(g,1);
158 digitalWrite(DP,1);
159 }
160
161
162 void seven(){
163     digitalWrite(a,1);
164     digitalWrite(b,1);
165     digitalWrite(c,1);
166     digitalWrite(d,0);
167     digitalWrite(e,0);
168     digitalWrite(f,0);
169     digitalWrite(g,0);
170     digitalWrite(DP,1);
171 }
172
173
174
```

```
175
176 void eight(){
177     digitalWrite(a,1);
178     digitalWrite(b,1);
179     digitalWrite(c,1);
180     digitalWrite(d,1);
181     digitalWrite(e,1);
182     digitalWrite(f,1);
183     digitalWrite(g,1);
184     digitalWrite(DP,1);
185 }
186
187
188
189 void nine(){
190     digitalWrite(a,1);
191     digitalWrite(b,1);
192     digitalWrite(c,1);
193     digitalWrite(d,1);
194     digitalWrite(e,0);
195     digitalWrite(f,1);
196     digitalWrite(g,1);
197     digitalWrite(DP,1);
198 }
199
200
```

## 15. Combining Two Seven Segment Displays

```
sketch_nov30b-Two_Seven_segment_dis
File Edit Sketch Tools Help
[Icons] Select Board
sketch_nov30b-Two_Seven_segment_display.ino
1 #define time1 1000
2 #define time2 0
3
4 #define a1 7
5 #define b1 6
6 #define c1 5
7 #define d1 4
8 #define e1 3
9 #define f1 2
10 #define g1 1
11 #define DP1 A0
12
13 #define a2 0
14 #define b2 13
15 #define c2 12
16 #define d2 11
17 #define e2 10
18 #define f2 9
19 #define g2 8
20 #define DP2 A1
21
22 void setup() {
23     pinMode(a2,OUTPUT);
24     pinMode(b2,OUTPUT);
25
```

```
25     pinMode(b2,OUTPUT);
26     pinMode(c2,OUTPUT);
27     pinMode(d2,OUTPUT);
28     pinMode(e2,OUTPUT);
29     pinMode(f2,OUTPUT);
30     pinMode(g2,OUTPUT);
31     pinMode(DP2,OUTPUT);
32
33
34     pinMode(a1,OUTPUT);
35     pinMode(b1,OUTPUT);
36     pinMode(c1,OUTPUT);
37     pinMode(d1,OUTPUT);
38     pinMode(e1,OUTPUT);
39     pinMode(f1,OUTPUT);
40     pinMode(g1,OUTPUT);
41     pinMode(DP1,OUTPUT);
42 }
43
44
45 void loop() {
46     zero1();
47     delay(time2);
48
49     zero2();
50     delay(time1);
51     one2();
52     delay(time1);
53     two2();
54     delay(time1);
55
```

Signature -

Page No - 22





Date -

Topic -

```
sketch_nov30b-Two_Seven_segment_disp
File Edit Sketch Tools Help
[Icons] Select Board
sketch_nov30b-Two_Seven_segment_display.ino
54 two2();
55 delay(time1);
56 three2();
57 delay(time1);
58 four2();
59 delay(time1);
60 five2();
61 delay(time1);
62 six2();
63 delay(time1);
64 seven2();
65 delay(time1);
66 eight2();
67 delay(time1);
68 nine2();
69 delay(time1);
70
71 one1();
72 delay(time2);
73
74 zero2();
75 delay(time1);
76 one2();
77 delay(time1);
78 two2();
79 delay(time1);
80 three2();
81 delay(time1);
82 four2();
83 delay(time1);
84 five2();
85 delay(time1);
86 six2();
87 delay(time1);
88 seven2();
89 delay(time1);
90 eight2();
91 delay(time1);
92 nine2();
93 delay(time1);
94
95 two1();
96 delay(time2);
97
98 zero2();
99 delay(time1);
100 one2();
101 delay(time1);
102 two2();
103 delay(time1);
104 three2();
105 delay(time1);
106 four2();
107 delay(time1);
108 five2();
```

```
sketch_nov30b-Two_Seven_segment_d
File Edit Sketch Tools Help
[Icons] Select Board
sketch_nov30b-Two_Seven_segment_display.ino
106 four2();
107 delay(time1);
108 five2();
109 delay(time1);
110 six2();
111 delay(time1);
112 seven2();
113 delay(time1);
114 eight2();
115 delay(time1);
116 nine2();
117 delay(time1);
118
119 three1();
120 delay(time2);
121
122 zero2();
123 delay(time1);
124 one2();
125 delay(time1);
126 two2();
127 delay(time1);
128 three2();
129 delay(time1);
130 four2();
131 delay(time1);
132 five2();
133 delay(time1);
134 six2();
135 delay(time1);
136 seven2();
137 delay(time1);
138 eight2();
139 delay(time1);
140 nine2();
141 delay(time1);
142
143 four1();
144 delay(time2);
145
146 zero2();
147 delay(time1);
148 one2();
149 delay(time1);
150 two2();
151 delay(time1);
152 three2();
153 delay(time1);
154 four2();
155 delay(time1);
156 five2();
157 delay(time1);
158 six2();
159 delay(time1);
160 seven2();
161 delay(time1);
```

Signature -

Page No - 23





Date -

Topic -

```
sketch_nov30b-Two_Seven_segment_display.ino
File Edit Sketch Tools Help
Select Board
sketch_nov30b-Two_Seven_segment_display.ino
159 delay(time1);
160 seven2();
161 delay(time1);
162 eight2();
163 delay(time1);
164 nine2();
165 delay(time1);
166
167 five1();
168 delay(time2);
169
170 zero2();
171 delay(time1);
172 one2();
173 delay(time1);
174 two2();
175 delay(time1);
176 three2();
177 delay(time1);
178 four2();
179 delay(time1);
180 five2();
181 delay(time1);
182 six2();
183 delay(time1);
184 seven2();
185 delay(time1);
186 eight2();
187 delay(time1);
188 nine2();
189 delay(time1);
190
191 six1();
192 delay(time2);
193
194 zero2();
195 delay(time1);
196 one2();
197 delay(time1);
198 two2();
199 delay(time1);
200 three2();
201 delay(time1);
202 four2();
203 delay(time1);
204 five2();
205 delay(time1);
206 six2();
207 delay(time1);
208 seven2();
209 delay(time1);
210 eight2();
211 delay(time1);
212 nine2();
213 delay(time1);
```

```
sketch_nov30b-Two_Seven_segment_display.ino
File Edit Sketch Tools Help
Select Board
sketch_nov30b-Two_Seven_segment_display.ino
211 delay(time1);
212 nine2();
213 delay(time1);
214
215 seven1();
216 delay(time2);
217
218 zero2();
219 delay(time1);
220 one2();
221 delay(time1);
222 two2();
223 delay(time1);
224 three2();
225 delay(time1);
226 four2();
227 delay(time1);
228 five2();
229 delay(time1);
230 six2();
231 delay(time1);
232 seven2();
233 delay(time1);
234 eight2();
235 delay(time1);
236 nine2();
237 delay(time1);
238
239 eight1();
240 delay(time2);
241
242 zero2();
243 delay(time1);
244 one2();
245 delay(time1);
246 two2();
247 delay(time1);
248 three2();
249 delay(time1);
250 four2();
251 delay(time1);
252 five2();
253 delay(time1);
254 six2();
255 delay(time1);
256 seven2();
257 delay(time1);
258 eight2();
259 delay(time1);
260 nine2();
261 delay(time1);
262
263 nine1();
264 delay(time2);
265
266 zero2();
```

Signature -

Page No - 24



Date -

Topic -

```
sketch_nov30b-Two_Seven_segment_d
File Edit Sketch Tools Help
Select Board
sketch_nov30b-Two_Seven_segment_display.ino
267 delay(time1);
268 one2();
269 delay(time1);
270 two2();
271 delay(time1);
272 three2();
273 delay(time1);
274 four2();
275 delay(time1);
276 five2();
277 delay(time1);
278 six2();
279 delay(time1);
280 seven2();
281 delay(time1);
282 eight2();
283 delay(time1);
284 nine2();
285 delay(time1);
286
287 }
288
289
290 //display 02
291
292 void zero2(){
293
294     digitalWrite(a2,1);
295     digitalWrite(b2,1);
296     digitalWrite(c2,1);
297     digitalWrite(d2,1);
298     digitalWrite(e2,1);
299     digitalWrite(f2,1);
300     digitalWrite(g2,0);
301     digitalWrite(DP2,0);
302 }
303
304 void one2(){
305
306     digitalWrite(a2,0);
307     digitalWrite(b2,1);
308     digitalWrite(c2,1);
309     digitalWrite(d2,0);
310     digitalWrite(e2,0);
311     digitalWrite(f2,0);
312     digitalWrite(g2,0);
313     digitalWrite(DP2,0);
314 }
315
316
317
318 void two2(){
319
320     digitalWrite(a2,1);
321     digitalWrite(b2,1);
```

```
sketch_nov30b-Two_Seven_segment_d
File Edit Sketch Tools Help
Select Board
sketch_nov30b-Two_Seven_segment_display.ino
320 digitalWrite(a2,1);
321 digitalWrite(b2,1);
322 digitalWrite(c2,0);
323 digitalWrite(d2,1);
324 digitalWrite(e2,1);
325 digitalWrite(f2,0);
326 digitalWrite(g2,1);
327 digitalWrite(DP2,0);
328 }
329
330
331 void three2(){
332
333     digitalWrite(a2,1);
334     digitalWrite(b2,1);
335     digitalWrite(c2,1);
336     digitalWrite(d2,1);
337     digitalWrite(e2,0);
338     digitalWrite(f2,0);
339     digitalWrite(g2,1);
340     digitalWrite(DP2,0);
341 }
342
343
344 void four2(){
345
346     digitalWrite(a2,0);
347     digitalWrite(b2,1);
348     digitalWrite(c2,1);
349     digitalWrite(d2,0);
350     digitalWrite(e2,0);
351     digitalWrite(f2,1);
352     digitalWrite(g2,1);
353     digitalWrite(DP2,0);
354 }
355
356
357 void five2(){
358
359     digitalWrite(a2,1);
360     digitalWrite(b2,0);
361     digitalWrite(c2,1);
362     digitalWrite(d2,1);
363     digitalWrite(e2,0);
364     digitalWrite(f2,1);
365     digitalWrite(g2,1);
366     digitalWrite(DP2,0);
367 }
368
369
370 void six2(){
371
372     digitalWrite(a2,1);
373     digitalWrite(b2,0);
374     digitalWrite(c2,1);
```

Signature -

Page No - 25



Date -

Topic -

```
sketch_nov30b-Two_Seven_segment_d
File Edit Sketch Tools Help
Select Board
sketch_nov30b-Two_Seven_segment_display.ino
375 digitalWrite(d2,1);
376 digitalWrite(e2,1);
377 digitalWrite(f2,1);
378 digitalWrite(g2,1);
379 digitalWrite(DP2,0);
380 }
381
382
383 void seven2(){
384
385     digitalWrite(a2,1);
386     digitalWrite(b2,1);
387     digitalWrite(c2,1);
388     digitalWrite(d2,0);
389     digitalWrite(e2,0);
390     digitalWrite(f2,0);
391     digitalWrite(g2,0);
392     digitalWrite(DP2,0);
393 }
394
395
396
397 void eight2(){
398
399     digitalWrite(a2,1);
400     digitalWrite(b2,1);
401     digitalWrite(c2,1);
402     digitalWrite(d2,1);
403     digitalWrite(e2,1);
404     digitalWrite(f2,1);
405     digitalWrite(g2,1);
406     digitalWrite(DP2,0);
407 }
408
409
410 void nine2(){
411
412     digitalWrite(a2,1);
413     digitalWrite(b2,1);
414     digitalWrite(c2,1);
415     digitalWrite(d2,1);
416     digitalWrite(e2,0);
417     digitalWrite(f2,1);
418     digitalWrite(g2,1);
419     digitalWrite(DP2,0);
420 }
421
422
423
424 // display 01
425 void zero1(){
426
427     digitalWrite(a1,1);
428     digitalWrite(b1,1);
429     digitalWrite(c1,1);
```

```
sketch_nov30b-Two_Seven_segment_d
File Edit Sketch Tools Help
Select Board
sketch_nov30b-Two_Seven_segment_display.ino
430 digitalWrite(d1,1);
431 digitalWrite(e1,1);
432 digitalWrite(f1,1);
433 digitalWrite(g1,0);
434 digitalWrite(DP1,0);
435 }
436
437
438 void one1(){
439
440     digitalWrite(a1,0);
441     digitalWrite(b1,1);
442     digitalWrite(c1,1);
443     digitalWrite(d1,0);
444     digitalWrite(e1,0);
445     digitalWrite(f1,0);
446     digitalWrite(g1,0);
447     digitalWrite(DP1,0);
448 }
449
450 void two1(){
451
452     digitalWrite(a1,1);
453     digitalWrite(b1,1);
454     digitalWrite(c1,0);
455     digitalWrite(d1,1);
456     digitalWrite(e1,1);
457     digitalWrite(f1,0);
458     digitalWrite(g1,1);
459     digitalWrite(DP1,0);
460 }
461
462
463 void three1(){
464
465     digitalWrite(a1,1);
466     digitalWrite(b1,1);
467     digitalWrite(c1,1);
468     digitalWrite(d1,1);
469     digitalWrite(e1,0);
470     digitalWrite(f1,0);
471     digitalWrite(g1,1);
472     digitalWrite(DP1,0);
473 }
474
475
476 void four1(){
477
478     digitalWrite(a1,0);
479     digitalWrite(b1,1);
480     digitalWrite(c1,1);
481     digitalWrite(d1,0);
482     digitalWrite(e1,0);
483     digitalWrite(f1,1);
484     digitalWrite(g1,1);
```

Signature -

Page No - 26



Date -

Topic -

```
sketch_nov30b-Two_Seven_segment_d
File Edit Sketch Tools Help
Select Board
sketch_nov30b-Two_Seven_segment_display.ino
485 digitalWrite(DP1,0);
486 }
487
488
489 void five1(){
490
491     digitalWrite(a1,1);
492     digitalWrite(b1,0);
493     digitalWrite(c1,1);
494     digitalWrite(d1,1);
495     digitalWrite(e1,0);
496     digitalWrite(f1,1);
497     digitalWrite(g1,1);
498     digitalWrite(DP1,0);
499 }
500
501
502 void six1(){
503
504     digitalWrite(a1,1);
505     digitalWrite(b1,0);
506     digitalWrite(c1,1);
507     digitalWrite(d1,1);
508     digitalWrite(e1,1);
509     digitalWrite(f1,1);
510     digitalWrite(g1,1);
511     digitalWrite(DP1,0);
512 }
513
514
515 void seven1(){
516
517     digitalWrite(a1,1);
518     digitalWrite(b1,1);
519     digitalWrite(c1,1);
520     digitalWrite(d1,0);
521     digitalWrite(e1,0);
522     digitalWrite(f1,0);
523     digitalWrite(g1,0);
524     digitalWrite(DP1,0);
525 }
526
527
528
529 void eight1(){
530
531     digitalWrite(a1,1);
532     digitalWrite(b1,1);
533     digitalWrite(c1,1);
534     digitalWrite(d1,1);
535     digitalWrite(e1,1);
536     digitalWrite(f1,1);
537     digitalWrite(g1,1);
538     digitalWrite(DP1,0);
539 }
```

```
540
541
542 void nine1(){
543
544     digitalWrite(a1,1);
545     digitalWrite(b1,1);
546     digitalWrite(c1,1);
547     digitalWrite(d1,1);
548     digitalWrite(e1,0);
549     digitalWrite(f1,1);
550     digitalWrite(g1,1);
551     digitalWrite(DP1,0);
552 }
553
```

Signature -

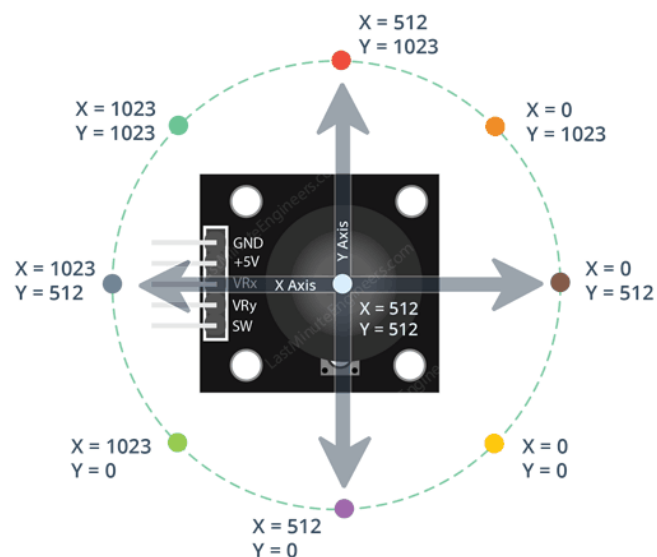
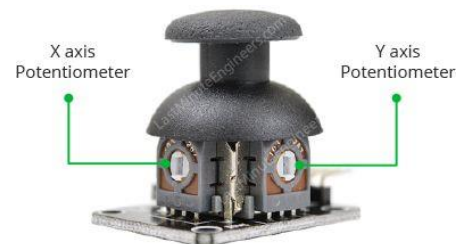
Page No - 27



Date -

Topic -

## 16. Joystick Module



- ❖ The pinout for a Dual Axis XY Joystick Module typically includes five pins, arranged in a row or a column.

Signature -

Page No - 28



Date -

Topic -

## 17. Control LED's Brightness Using Joy Stick

```
Controlling_LED_brightness_using_joy_stick_copy_20241207161301.ino
File Edit Sketch Tools Help
Select Board

Controlling_LED_brightness_using_joy_stick_copy_20241207161301.ino
1 int X_Pin = A0;
2 int Y_Pin = A1;
3 int X1;
4 int X2;
5 int Y1;
6 int Y2;
7
8
9 void setup() {
10
11     Serial.begin(9600);
12     pinMode(A0, INPUT);
13     pinMode(A1, INPUT);
14     pinMode(3, INPUT_PULLUP);
15     pinMode(5, OUTPUT);
16     pinMode(6, OUTPUT);
17     pinMode(9, OUTPUT);
18     pinMode(10, OUTPUT);
19 }
20
21 void loop() {
22
23     int X_data = analogRead(A0);
24     int Y_data = analogRead(A1);
25     int SW = digitalRead(3);
26
27
28     Serial.print("X Axis: ");
29     Serial.print(X_data);
30     Serial.print("\t");
31     Serial.print("Y Axis: ");
32     Serial.print(Y_data);
33     Serial.print("\t");
34     Serial.print("SW: ");
35     Serial.println(SW);
36     delay(100);
37
38     X1=map(X_data,508,0,0,255);
39     X2=map(X_data,508,1021,0,255);
40
41     Y1=map(Y_data,518,0,0,255);
42     Y2=map(Y_data,518,1022,0,255);
43
44
45     Serial.print("X1");
```

```
Controlling_LED_brightness_using_joy_stick_copy_20241207161301.ino
File Edit Sketch Tools Help
Select Board

Controlling_LED_brightness_using_joy_stick_copy_20241207161301.ino
36     delay(100);
37
38     X1=map(X_data,508,0,0,255);
39     X2=map(X_data,508,1021,0,255);
40
41     Y1=map(Y_data,518,0,0,255);
42     Y2=map(Y_data,518,1022,0,255);
43
44
45     Serial.print("X1");
46     Serial.print(X1);
47     Serial.print("\t");
48     Serial.print("X2");
49     Serial.print(X2);
50     Serial.print("\t");
51     Serial.print("Y1");
52     Serial.print(Y1);
53     Serial.print("\t");
54     Serial.print("Y2");
55     Serial.print(Y2);
56     Serial.print("\t");
57
58
59     if (X1>0) {
60         X1=0;
61     }
62
63     if (X2>0) {
64         X2=0;
65     }
66
67     if (Y1>0) {
68         Y1=0;
69     }
70
71     if (Y2>0) {
72         Y2=0;
73     }
74
75     analogWrite(5,X1);
76     analogWrite(6,X2);
77     analogWrite(9,Y1);
78     analogWrite(10,Y2);
79 }
80
```

Signature -

Page No - 29





Date -

Topic -

```
Controlling_8_LED_using_joy_stick | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Select Board
Controlling_8_LED_using_joy_stick.ino
1 int X_Pin = A0;
2 int Y_Pin = A1;
3 int X1;
4 int X2;
5 int Y1;
6 int Y2;
7
8
9 void setup() {
10
11     Serial.begin (9600);
12     pinMode (A0, INPUT);
13     pinMode (A1, INPUT);
14     pinMode (3, INPUT_PULLUP);
15     pinMode (5,OUTPUT);
16     pinMode (6,OUTPUT);
17     pinMode (9,OUTPUT);
18     pinMode (10,OUTPUT);
19 }
20
21 void loop() {
22
23     int X_data = analogRead (A0);
24     int Y_data = analogRead (A1);
25     int SW = digitalRead(3);
26
27
28     Serial.print("X Axis: ");
29     Serial.print(X_data);
30     Serial.print("\t");
31     Serial.print("Y Axis: ");
32     Serial.print(Y_data);
33     Serial.print("\t");
34     Serial.print("SW: ");
35     Serial.println(SW);
36     delay (100);
37
38     X1=map(X_data,508,0,0,255);
39     X2=map(X_data,508,1021,0,255);
40
41     Y1=map(Y_data,518,0,0,255);
42     Y2=map(Y_data,518,1022,0,255);
43
44
45     Serial.print("X1");
```

```
Controlling_8_LED_using_joy_stick | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Select Board
Controlling_8_LED_using_joy_stick.ino
36     delay (100);
37
38     X1=map(X_data,508,0,0,255);
39     X2=map(X_data,508,1021,0,255);
40
41     Y1=map(Y_data,518,0,0,255);
42     Y2=map(Y_data,518,1022,0,255);
43
44
45     Serial.print("X1");
46     Serial.print(X1);
47     Serial.print("\t");
48     Serial.print("X2");
49     Serial.print(X2);
50     Serial.print("\t");
51     Serial.print("Y1");
52     Serial.print(Y1);
53     Serial.print("\t");
54     Serial.print("Y2");
55     Serial.print(Y2);
56     Serial.print("\t");
57
58
59     if (X1>0) {
60         X1=0;
61     }
62
63     if (X2>0) {
64         X2=0;
65     }
66
67     if (Y1>0) {
68         Y1=0;
69     }
70
71     if (Y2>0) {
72         Y2=0;
73     }
74
75     analogWrite(5,X1);
76     analogWrite(6,X2);
77     analogWrite(9,Y1);
78     analogWrite(10,Y2);
79 }
80
```

Signature -

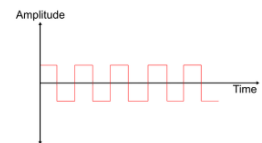
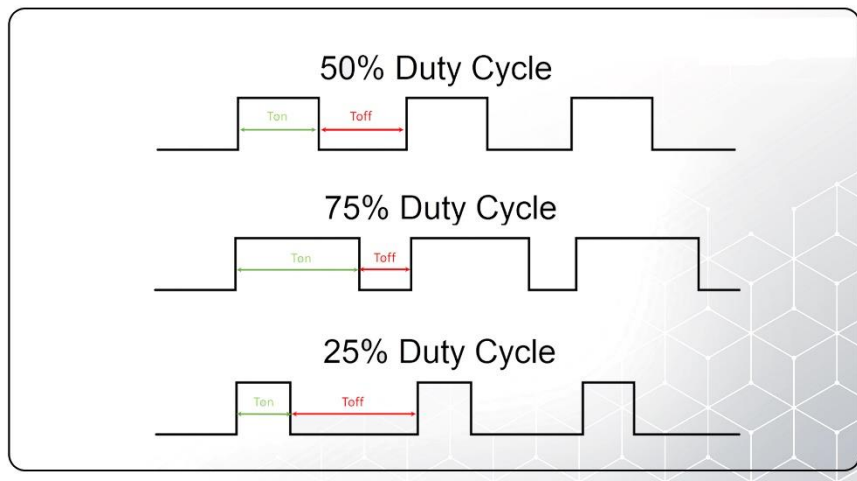
Page No - 30



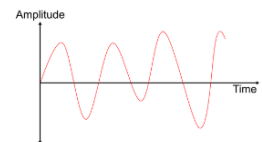
Date -

Topic -

## 18. Pulse Width Modulation (PWM)



Digital Signal



Analog Signal

- ❖ PWM (Pulse Width Modulation) is a technique used to control the power delivered to electronic components by varying the duty cycle of a digital signal.
- ❖ It works by rapidly switching a signal ON and OFF at a fixed frequency, where the duty cycle (percentage of time the signal is HIGH) determines the effective output power.
- **Common Uses of PWM**
  - Motor speed control (e.g., DC motors, servo motors)
  - LED brightness control
  - Voltage regulation (e.g., switching power supplies)

For example, in a PWM signal with a 50% duty cycle, the output is HIGH for half the time and LOW for the other half, resulting in half the power output.

Signature -

Page No - 31

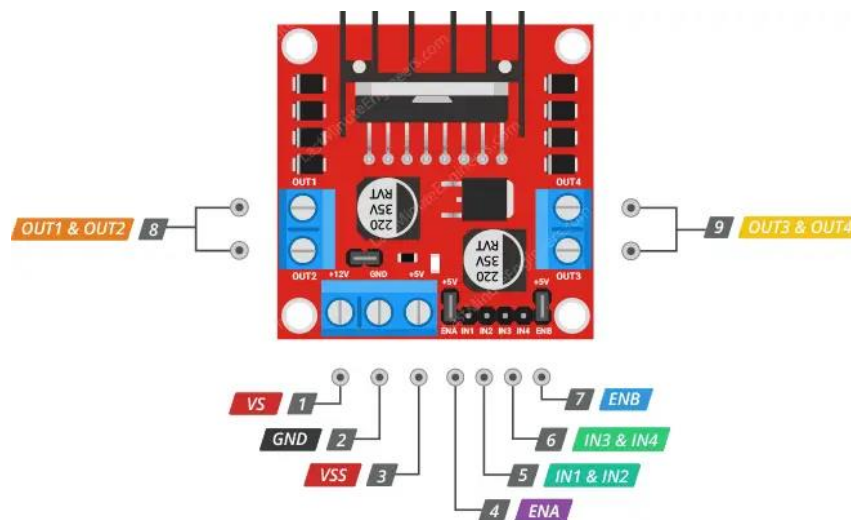




Date -

Topic -

## 19. Motor Driver L298N



- ❖ The L298N is a motor driver IC designed to control the speed and direction of DC motors and stepper motors.
- ❖ It operates on the H-bridge principle, which allows the motor to rotate in both clockwise and counterclockwise directions by changing the polarity of the voltage applied to it.
- ❖ The L298N motor driver has a supply range of 5V to 35V and is capable of 2A continuous current per channel, so it works very well with most of our DC motors.

Motor output voltage	5V – 35V
Motor output voltage (Recommended)	7V – 12V
Logic input voltage	5V – 7V
Continuous current per channel	2A
Max Power Dissipation	25W

Signature -

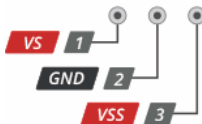
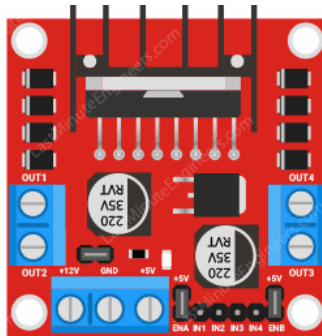
Page No - 32



Date -

Topic -

- The L298N motor driver has two input power pins: VS and VSS.

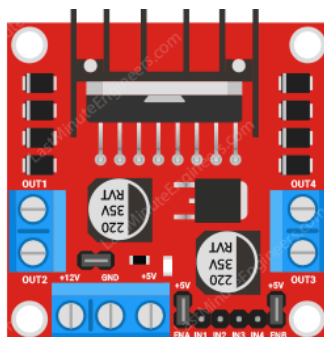
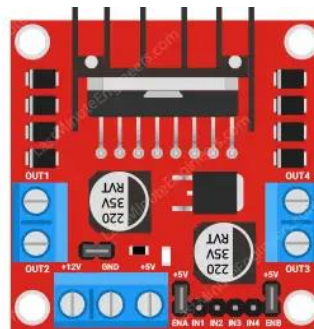


**VS** - pin powers the IC's internal H-Bridge, which drives the motors. This pin accepts input voltages ranging from 5 to 12V.

**VSS** - is used to power the logic circuitry within the L298N IC, and can range between 5V and 7V.

**GND** - is the common ground pin.

OUT1 and OUT2 for motor A  
and OUT3 and OUT4 for motor B.



The module has two direction control pins. The IN1 and IN2 pins control the spinning direction of motor A; While IN3 and IN4 control the spinning direction of motor B.

Signature -

Page No - 33



Date -

Topic -

## 20. Control Motor with Joy Stick Using L298N Motor Driver

```
1 int A
2 int B
3 int C
4
5 void setup() {
6   pinMode(3,OUTPUT);
7   pinMode(5,OUTPUT);
8   pinMode(6,OUTPUT);
9   pinMode(A0,INPUT);
10 }
11
```

```
11
12 void loop() {
13
14   A= analogRead(A0);
15   B= map(A,0,512,255,0);
16   C= map(A,512,1024,0,255);
17
18   if (A<512){
19     analogWrite(3,B);
20     digitalWrite(6,HIGH);
21     digitalWrite(5,LOW);
22   }
23
24   else {
25     analogWrite(3,C);
26     digitalWrite(6,LOW);
27     digitalWrite(5,HIGH);
28   }
29
30 }
31
```

## 21. Control Motor Using Potentiometer

```
1 int A;
2 int B;
3
4 void setup() {
5   pinMode (6,OUTPUT);
6   pinMode (5,OUTPUT);
7   pinMode (3,OUTPUT);
8   pinMode (A0,INPUT);
9
10 }
11
12 void loop() {
13   A= analogRead(A0);
14   B= map(A,0,512,0,255);
15   analogWrite(3,B);
16   digitalWrite(6,HIGH);
17   digitalWrite(5,LOW);
18
19 }
20
```

## 22. Control Speed of the Motor Using L298N Motor Driver

```
1 void setup() {
2   pinMode (6,OUTPUT);
3   pinMode (5,OUTPUT);
4   pinMode (3,OUTPUT);
5
6 }
7
8 void loop() {
9   analogWrite(3,100);
10  digitalWrite(6,HIGH);
11  digitalWrite(5,LOW);
12  delay(1500);
13  digitalWrite(6,LOW);
14  digitalWrite(5,HIGH);
15  delay(1500);
16 }
17
```

Signature -

Page No - 34



Date -

Topic -

## 23. Control Simple Robot Car's Two Motors

```
sketch_jan4a_L298N_Motor_driver_controll_With_ultrasonic_sensor.ino
File Edit Sketch Tools Help
[Icons] Select Board

sketch_jan4a_L298N_Motor_driver_controll_With_ultrasonic_sensor.ino
1 int A;
2 int B;
3 int C;
4 int trig=A1;
5 int echo=A0;
6 long time=0;
7 long cm=0;
8 long inch=0;
9
10 void setup() {
11     pinMode(3,OUTPUT);
12     pinMode(5,OUTPUT);
13     pinMode(6,OUTPUT);
14     pinMode(9,OUTPUT);
15     pinMode(10,OUTPUT);
16     pinMode(11,OUTPUT);
17
18     pinMode(trig,OUTPUT);
19     pinMode(echo,INPUT);
20     Serial.begin(9600);
21 }
22
23 void loop() {
24
25     digitalWrite(trig,0);
26     delayMicroseconds(2);
27     digitalWrite(trig,1);
28     delayMicroseconds(10);
29     digitalWrite(trig,0);
30     time=pulseIn(echo,1);
31     cm=time/29/2;
32     inch=time/74/2;
33     Serial.print(inch);
34     Serial.print("\t");
35     Serial.print(cm);
36     Serial.println("cm");
37
38     A= analogRead(A0);
39     B= map(A,0,1024,0,255);
40     C= map(A,0,1024,0,255);
41
42     analogWrite(3,B);
43     analogWrite(11,C);
44
45     ...
```

```
sketch_jan4a_L298N_Motor_driver_controll_With_ultrasonic_sensor.ino
File Edit Sketch Tools Help
[Icons] Select Board

sketch_jan4a_L298N_Motor_driver_controll_With_ultrasonic_sensor.ino
52 }
53 if(cm>35) {
54     turnleft();
55     delay(5000);
56 }
57 }
58
59 else {
60     forward();
61 }
62 }
63
64 void forward() {
65     digitalWrite(6,HIGH);
66     digitalWrite(5,LOW);
67     digitalWrite(9,HIGH);
68     digitalWrite(10,LOW);
69 }
70
71 void backward() {
72     digitalWrite(6,LOW);
73     digitalWrite(5,HIGH);
74     digitalWrite(9,LOW);
75     digitalWrite(10,HIGH);
76 }
77
78 void turnright() {
79     digitalWrite(6,LOW);
80     digitalWrite(5,HIGH);
81     digitalWrite(9,LOW);
82     digitalWrite(10,LOW);
83 }
84
85 void turnleft() {
86     digitalWrite(6,LOW);
87     digitalWrite(5,LOW);
88     digitalWrite(9,LOW);
89     digitalWrite(10,HIGH);
90 }
91
92 void stop() {
93     digitalWrite(6,LOW);
94     digitalWrite(5,LOW);
95     digitalWrite(9,LOW);
96     digitalWrite(10,LOW);
97 }
```



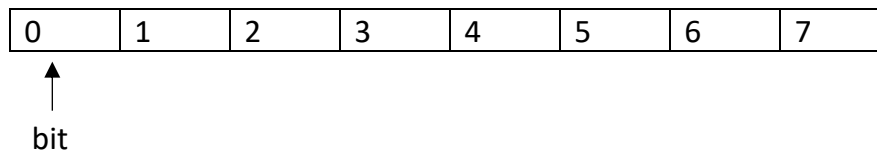
Date -

Topic -

## 24. Data Types

### 24.1 byte

- ❖ A byte stores an 8-bit unsigned number, from 0 to 255.



### 24.2. int

- ❖ Integers are primary data-type for number storage.
- ❖ On the Arduino UNO (and other ATmega based boards) an int stores a 16-bit (2-byte) value. This yields a range of -32,768 to 32,767 (minimum value of -215 and a maximum value of (215) - 1).

```
sketch_jan11d_integer | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Checkmark] [Next] [Previous] [Select Board]
sketch_jan11d_integer.ino
1 void setup() {
2   Serial.begin(9600);
3
4   int num = 12345; //Declare an integer variable
5   Serial.println(num); //Print a integer
6 }
7
8 void loop() {
9
10
11 }
```

Signature -

Page No - 36



Date -

Topic -

### 24.3. long

- ❖ Long variables are extended size variables for number storage, and store 32 bits (4 bytes), from -2,147,483,648 to 2,147,483,647.
- ❖ If doing math with integers at least one of the values must be of type long.

### 24.4. float

- ❖ Datatype for floating-point numbers, a number that has a decimal point.
- ❖ Floating-point numbers are often used to approximate analog and continuous values because they have greater resolution than integers.
- ❖ Floating-point numbers can be as large as 3.4028235E+38 and as low as -3.4028235E+38.
- ❖ They are stored as 32 bits (4 bytes) of information.

```
sketch_jan11i_float | Arduino IDE 2.3.4
File Edit Sketch Tools Help

[Checkmark] [Next] [Previous] [Select Board]

sketch_jan11i_float.ino
1 float SensorReading[4]={1.23,3.44,5.66,111.0};
2
3 void setup() {
4   Serial.begin(9600);
5
6 }
7
8 void loop() {
9
10  delay(1000);
11  Serial.println(SensorReading[0]);
12  Serial.println('\t');
13
14  Serial.println(SensorReading[2]);
15  Serial.println('\t');
16 }
17
```

Signature -

Page No - 37



Date -

Topic -

#### 24.5. bool

- ❖ A bool holds one of two values, true or false. (Each bool variable occupies one byte of memory.)

#### 24.6. double

- ❖ Double precision floating point number.
- ❖ On the UNO and other ATMEGA based boards, this occupies 4 bytes.
- ❖ That is, the double implementation is exactly the same as the float, with no gain in precision.
- ❖ On the Arduino Due, doubles have 8-byte (64 bit) precision.

#### 24.7. char

- ❖ A data type used to store a character value. Character literals are written in single quotes, like this: 'A' (for multiple characters use - strings -).
- ❖ The size of the char datatype is at least 8 bits. It's recommended to only use char for storing characters.
- ❖ For an unsigned, one-byte (8 bit) data type, use the byte data type.
- ❖ Stores a single character or a small number (1 byte )

```
sketch_jan11a_char | Arduino IDE 2.3.4
File Edit Sketch Tools Help

[Checkmark] [Next] [Run] [Select Board]

sketch_jan11a_char.ino
1 void setup() {
2   Serial.begin(9600);
3
4   char letter = 'A'; //Declare a char variable
5   Serial.println(letter); //Print the character
6 }
7
8 void loop() {
9
10 }
11
```

Signature -

Page No - 38



Date -

Topic -

## 24.8. string

- ❖ Text strings can be represented in two ways. Can use the String data type, or can make a string out of an array of type char and null-terminate it.
- ❖ Strings are always defined inside double quotes ("Abc") and characters are always defined inside single quotes(A).

```
sketch_jan11b_string | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Checkmark] [Next] [Upload] [USB] DOIT ESP32 DEVKIT V1
sketch_jan11b_string.ino
1 void setup() {
2   Serial.begin(9600);
3
4   String greeting = "Hello,Arduino!"; //Declare a string
5   Serial.println(greeting); //Print a string
6 }
7
8 void loop() {
9
10 }
11
```

## 24.9. Array

- ❖ An array is a collection of variables that are accessed with an index number.
- ❖ Arrays in the C++ programming language Arduino sketches are written in can be complicated, but using simple arrays is relatively straightforward.
- ❖ All of the methods below are valid ways to create (declare) an array.

```
sketch_mar13e | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Checkmark] [Next] [Upload] Select Board
sketch_mar13e.ino
1 // Declare an array of a given length without initializing the values:
2 int myInts[6];
3
4 // Declare an array without explicitly choosing a size (the compiler
5 // counts the elements and creates an array of the appropriate size):
6 int myPins[] = {2, 4, 8, 3, 6, 4};
7
8 // Declare an array of a given length and initialize its values:
9 int mySensVals[5] = {2, 4, -8, 3, 2};
10
11 // When declaring an array of type char, you'll need to make it longer
12 // by one element to hold the required the null termination character:
13 char message[6] = "hello";
```

Signature -

Page No - 39





Date -

Topic -

### 24.10. Addition

```
sketch_jan11e_adition | Arduino IDE 2.3.4
File Edit Sketch Tools Help

[Icons] Select Board

sketch_jan11e_adition.ino
1  int x=5;
2  int y=10;
3
4  void setup() {
5      Serial.begin(9600);
6  }
7
8
9  void loop() {
10     int i = x+y;
11     delay(1000);
12     Serial.println(i);
13 }
14
```

### 24.11. Subtraction

```
sketch_jan11f_sutraction | Arduino IDE 2.3.4
File Edit Sketch Tools Help

[Icons] Select Board

sketch_jan11f_sutraction.ino
1  int x=10;
2  int y=5;
3
4  void setup() {
5      Serial.begin(9600);
6  }
7
8
9  void loop() {
10     int i = x/y;
11     delay(1000);
12     Serial.println(i);
13 }
14
```

### 24.12. Multiplication

```
sketch_jan11h_multiplecaction | Arduino IDE 2.3.4
File Edit Sketch Tools Help

[Icons] Select Board

sketch_jan11h_multiplecaction.ino
1  int x=10;
2  int y=5;
3
4  void setup() {
5      Serial.begin(9600);
6  }
7
8
9  void loop() {
10     int i = x*y;
11     delay(1000);
12     Serial.println(i);
13 }
14
```

### 24.13. Division

```
sketch_jan11g_divition | Arduino IDE 2.3.4
File Edit Sketch Tools Help

[Icons] Select Board

sketch_jan11g_divition.ino
1  int x=10;
2  int y=5;
3
4  void setup() {
5      Serial.begin(9600);
6  }
7
8
9  void loop() {
10     int i = x-y;
11     delay(1000);
12     Serial.println(i);
13 }
14
```

Signature -

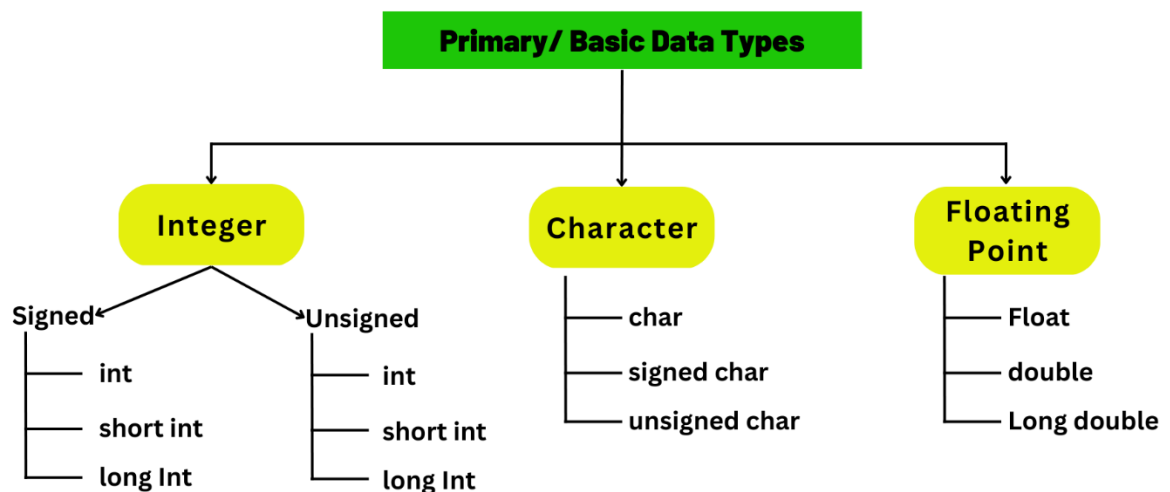
Page No - 40



Date -

Topic -

Arduino Data Types	Value Assigned	Value Ranges
boolean	8 Bit	True or False
byte	8 Bit	0 to 255
char	8 Bit	-127 to 128
unsigned char	8 Bit	0 to 255
word	16 Bit	0 to 65535
unsigned int	16 Bit	0 to 65535
int	16 Bit	-32768 to 32767
long	32 Bit	-2,147,483,648 to 2,147,483,647
float	32 Bit	-3.4028235E38 to 3.4028235E38





Date -

Topic -

## 25. Servo Motor

**SG90**



**MG995**



**MG90**



**MG996R**



- ❖ A servo motor is a type of electric motor that can rotate or move to a specific position, speed, or torque based on an input signal from a controller.

Signature -

Page No - 42



Date -

Topic -

## 26. Servo Motor

```
sketch_jan11a_sover_Moter | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Select Board
sketch_jan11a_sover_Moter.ino
1  #include <Servo.h>
2
3  Servo Myservo;
4
5  void setup() {
6      Myservo.attach(3);
7  }
8
9  void loop() {
10     Myservo.write(90);
11     delay(1000);
12
13     Myservo.write(0);
14     delay(1000);
15 }
16
```

## 27. Servo Motor with Potentiometer

```
sketch_jan11b_servo_with_protenciometer | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Select Board
sketch_jan11b_servo_with_protenciometer.ino
1  int A;
2  int B;
3
4  #include <Servo.h>
5
6  Servo Baseservo;
7
8  void setup() {
9      Serial.begin(9600);
10     Baseservo.attach(3);
11     pinMode(A0, INPUT);
12 }
13
14 void loop() {
15     A = analogRead(A0);
16     B = map(A, 0, 1024, 0, 180);
17
18     Baseservo.write(B);
19     Serial.println(A);
20 }
21
```

## 28. Servo Motor with Push Button

```
sketch_jan11c_servo_with_pushbuttn | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Select Board
sketch_jan11c_servo_with_pushbuttn.ino
1  #include <Servo.h>
2  Servo Baseservo;
3
4  void setup() {
5      Serial.begin(9600);
6      Baseservo.attach(6);
7      pinMode(2, INPUT);
8      pinMode(3, INPUT);
9      pinMode(4, INPUT);
10 }
11
12 void loop() {
13
14     int A = digitalRead(2);
15     int B = digitalRead(3);
16     int C = digitalRead(4);
17
18     if (A == HIGH) {
19         Baseservo.write(30);
20     }
21
22     if (B == HIGH) {
23         Baseservo.write(120);
24     }
25
26     if (C == HIGH) {
27         Baseservo.write(180);
28     }
29     Serial.print(A);
30     Serial.print(B);
31     Serial.println(C);
32 }
33
```

```
17
18     if (A == HIGH) {
19         Baseservo.write(30);
20     }
21
22     if (B == HIGH) {
23         Baseservo.write(120);
24     }
25
26     if (C == HIGH) {
27         Baseservo.write(180);
28     }
29     Serial.print(A);
30     Serial.print(B);
31     Serial.println(C);
32 }
33
```

Signature -

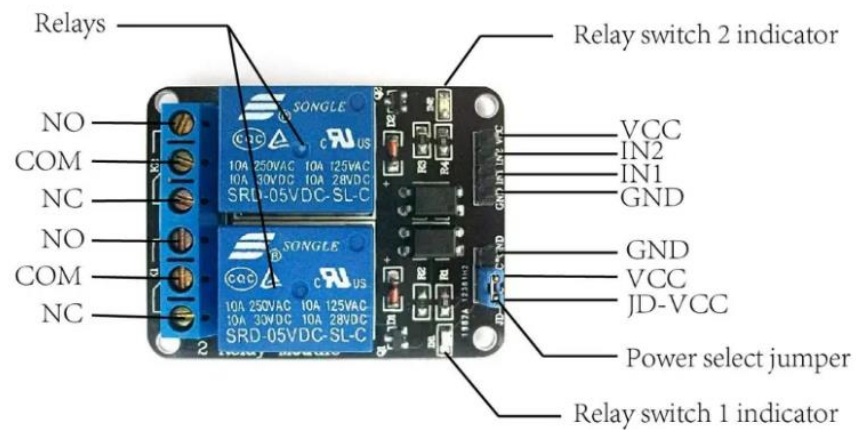
Page No - 43



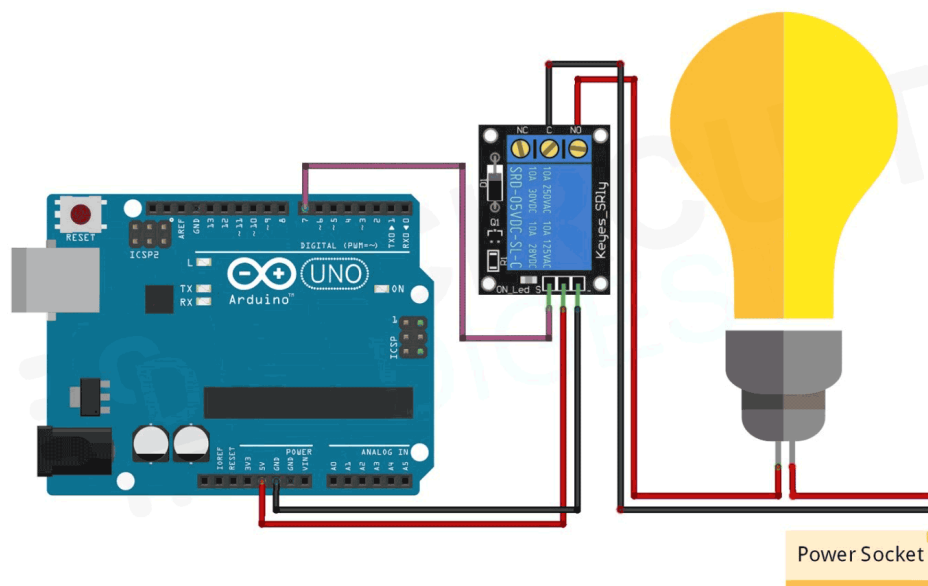
Date -

Topic -

## 29. Relay Module



- ❖ A relay module is an electrically operated switch that allows a low-voltage signal (like from an Arduino or ESP32) to control a high-voltage circuit (AC or DC devices).



Signature -

Page No - 44



Date -

Topic -

```
sketch_jan18a_Relay_module | Arduino IDE 2.3.4
File Edit Sketch Tools Help

sketch_jan18a_Relay_module.ino
1 void setup() {
2   pinMode(3,OUTPUT);
3   pinMode(5,OUTPUT);
4
5 }
6
7 void loop() {
8
9   digitalWrite(3,HIGH);
10  delay(1000);
11  digitalWrite(3,LOW);
12  delay(1000);
13  digitalWrite(5,HIGH);
14  delay(2000);
15  digitalWrite(5,LOW);
16  delay(2000);
17 }
18
```

Signature -

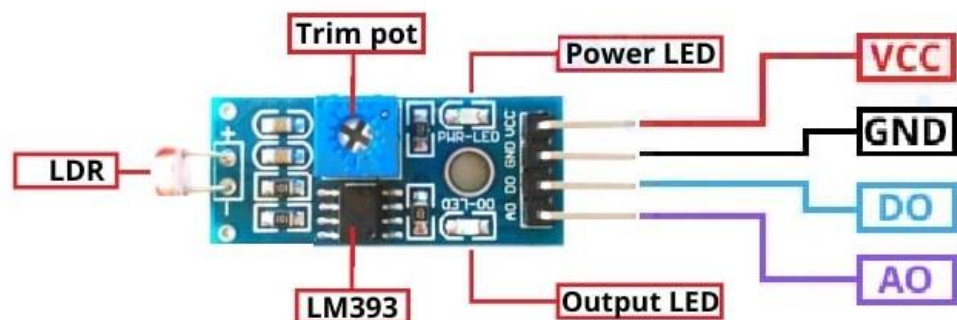
Page No - 45



Date -

Topic -

### 30. LDR Sensor Module



- ❖ An LDR (Light Dependent Resistor) module is a sensor that detects light intensity. The resistance of the LDR changes based on light levels:
  - Bright light → Low resistance (high voltage output)
  - Darkness → High resistance (low voltage output)
- ❖ Used in automatic lighting systems, solar trackers, and alarms.
- ❖ Outputs analog or digital signals depending on the module.

```
sketch_jan18b_LDR | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Select Board
sketch_jan18b_LDR.ino
1 int X;
2 int Y;
3
4 void setup() {
5   pinMode(2,INPUT); //DO Pin
6   pinMode(3,OUTPUT); //LED Pin
7   pinMode(A0,INPUT); //AO Pin
8   Serial.begin(9600);
9 }
10
11 void loop() {
12   //Analog Value (serial Print)
13   X= analogRead(A0);
14   Serial.print("Value. ");
15   Serial.print(X);
```

```
16
17 //Digital Value (Serial Print)
18 Y= digitalRead(2);
19 Serial.print(". Value2. ");
20 Serial.println(Y);
21
22 //LED Lit
23 if(Y==1) {
24   digitalWrite(3, 1);
25 }
26 else{
27   digitalWrite(3, 0);
28 }
29
```

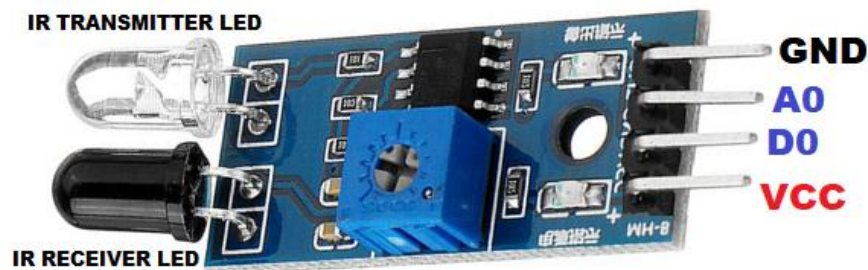




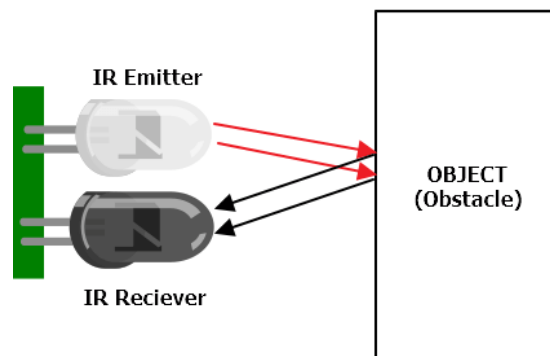
Date -

Topic -

## 31. IR Sensor Module



- ❖ An IR (Infrared) sensor module detects objects, motion, or heat using infrared light. It consists of an IR LED (emitter) and an IR receiver (photodiode or phototransistor).
- Types of IR Sensor Modules
  - Obstacle Avoidance Sensor – Reflects IR light off objects to detect proximity.
  - PIR (Passive Infrared) Sensor – Detects body heat/motion (used in security systems).
  - 
  - IR Remote Receiver – Decodes IR signals from remotes (e.g., TVs, ACs)



Signature -

Page No - 47





Date -

Topic -

```
sketch_jan18d_IR_Sensor | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Select Board
sketch_jan18d_IR_Sensor.ino
1  int X;
2
3  void setup() {
4      pinMode(2,INPUT); //DO Pin
5      pinMode(3,OUTPUT); //LED Pin
6      pinMode(4,OUTPUT); //LED Pin
7      pinMode(A0,INPUT); //AO Pin
8      Serial.begin(9600);
9  }
10
11 void loop() {
12     //Digital Value (serial Print)
13     X= digitalRead(2);
14     Serial.print("IR - ");
15     Serial.println(X);
16
17     //LED Lit
18     if(X==1) {
19         digitalWrite(3, 1);
20         digitalWrite(4, 0);
21     }
22     else{
23         digitalWrite(3, 0);
24         digitalWrite(4, 1);
25     }
26
27 }
```

Signature -

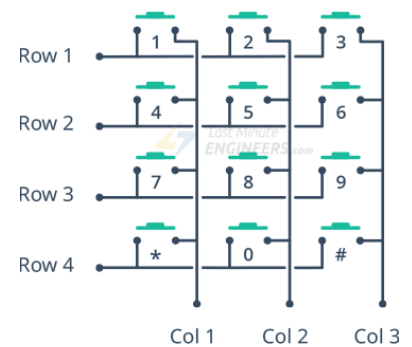
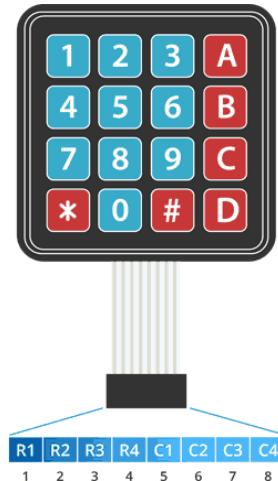
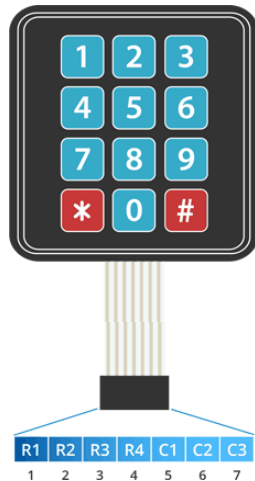
Page No - 48



Date -

Topic -

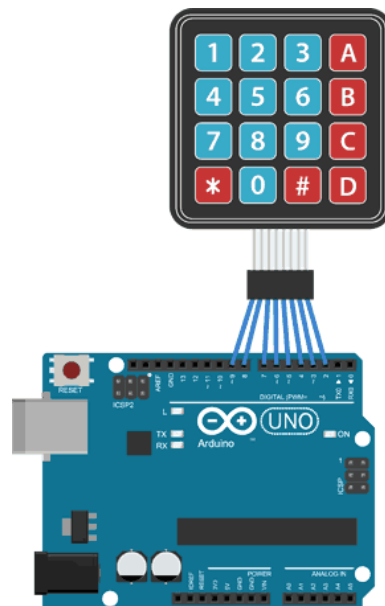
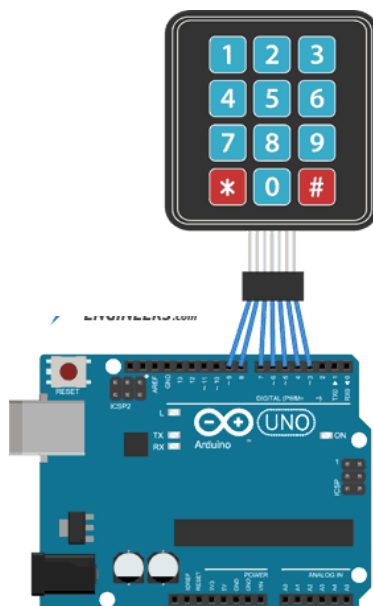
## 32. Key Pad



➤ R-> Rows

➤ C-> Columns

- ❖ A keypad for Arduino is an input device used to capture numerical or alphanumeric inputs. It consists of a matrix of buttons arranged in rows and columns, typically 4x3 (12 keys) or 4x4 (16 keys).
- ❖ Each button press connects a row and a column. The Arduino reads the key press by scanning row-column connections.
- ❖ Requires the Keypad library for easy interfacing.



Signature -

Page No - 49



Date -

Topic -

### 33. Key Pad with Password

```
sketch_jan18g_KeyPad_with_I2C | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Checkmark] [Next] [Previous] [Run] Arduino Uno
sketch_jan18g_KeyPad_with_I2C.ino
1  #include <Wire.h>
2  #include <Keypad.h>
3  #include <LiquidCrystal_I2C.h>
4
5  #define Password_Length 8
6
7  int SignalPin=12;
8
9  char Data[Password_Length];
10 char Master [Password_Length] = "Password";
11 byte data_count=0, master_count=0;
12 bool Pass_is_good;
13 char customKey;
14
15 const byte ROWS = 4; //four rows
16 const byte COLS = 4; //four columns
17 //define the symbols on the buttons of the keypads
18 char hexaKeys[ROWS][COLS] = {
19   {'1','2','3','A'},
20   {'4','5','6','B'},
21   {'7','8','9','C'},
22   {'*','0','#','D'}
23 };
24
25 byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
26 byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts of the keypad
27
28 //initialize an instance of class NewKeypad
29 Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
30
31
32 LiquidCrystal_I2C lcd(0x27, 16, 2);
33
34 void setup(){
35   lcd.init();
36   lcd.init();
37   lcd.backlight();
38   pinMode(SignalPin,OUTPUT);
39 }
40
41 void loop(){
42   lcd.setCursor(0,0);
43   lcd.print("Enter Password");
44
45   while (true) {
46     customKey = customKeypad.getKey();
47     if (customKey) {
48       if (data_count < Password_Length) {
49         Data[data_count] = customKey;
50         data_count++;
51       }
52       if (data_count == Password_Length) {
53         if (strcmp(Data, Master) == 0) {
54           Pass_is_good = true;
55           lcd.print("Access Granted");
56         } else {
57           Pass_is_good = false;
58           lcd.print("Access Denied");
59         }
60         data_count = 0;
61       }
62     }
63     delay(100);
64   }
65 }
```

Output

sketch uses 4906 bytes (15%) of program storage space. Maximum is 32256 bytes.

Signature -

Page No - 50



Date -

Topic -

```
sketch_jan18g_KeyPad_with_I2C | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Checkmark] [Next] [Previous] [Dropdown: Arduino Uno]

sketch_jan18g_KeyPad_with_I2C.ino
35  lcd.init();
36  lcd.init();
37  lcd.backlight();
38  pinMode(SignalPin,OUTPUT);
39  }
40  |
41  void loop(){
42  lcd.setCursor(0,0);
43  lcd.print("Enter Password");
44  customKey = customKeypad.getKey();
45
46  if (customKey){
47  Data[data_count]=customKey;
48  lcd.setCursor(data_count,1);
49  lcd.print(Data[data_count]);
50  data_count++;
51  }
52
53  if(data_count==Password_Length-1){
54  lcd.clear();
55  if(!strcmp(Data,Master)){
56  lcd.print("Correct");
57  digitalWrite(SignalPin,HIGH);
58  delay(5000);
59  digitalWrite(SignalPin,LOW);
60  }
61  else{
62  lcd.print("Incorrect");
63  delay(1000);
64  }
65  lcd.clear();
66  ClearData();
67  }
68  }
69
70  void ClearData(){
71  while(data_count!=0){
72  Data[data_count--]=0;
73  }
74  return;
75  }
76

Output
Sketch uses 4906 bytes (15%) of program storage space
```

Signature -

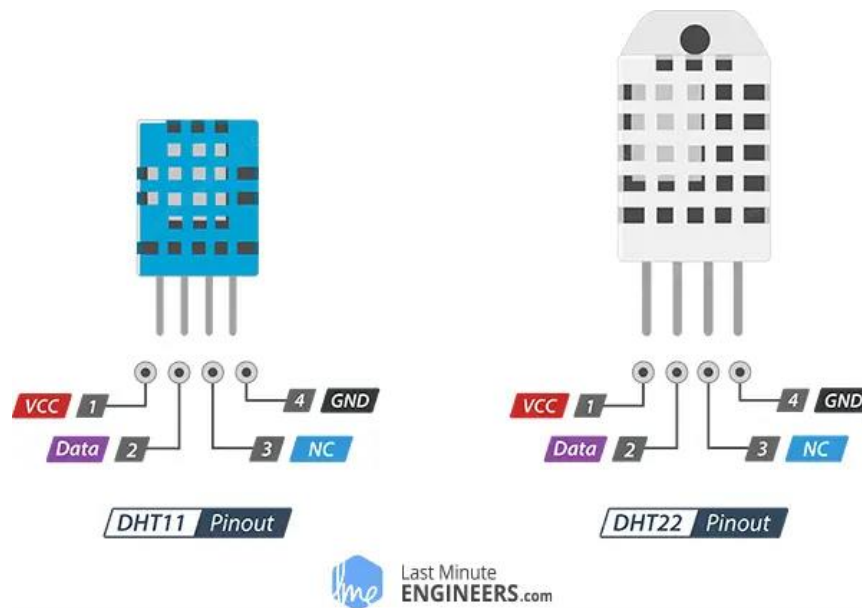
Page No - 51



Date -

Topic -

### 34. DHT22/11 Temperature and Humidity Sensor



The **DHT11** and **DHT22** are digital sensors used for measuring **temperature and humidity**. They communicate using a **single-wire protocol** and are commonly used in weather stations, home automation, and IoT projects.

	DHT11	DHT22
Operating Voltage	3 to 5V	3 to 5V
Max Operating Current	2.5mA max	2.5mA max
Humidity Range	20-80% / 5%	0-100% / 2-5%
Temperature Range	0-50°C / $\pm 2^{\circ}\text{C}$	-40 to 80°C / $\pm 0.5^{\circ}\text{C}$
Sampling Rate	1 Hz (reading every second)	0.5 Hz (reading every 2 seconds)
Advantage	Ultra-low cost	More Accurate

Signature -

Page No - 52



Date -

Topic -

## 35. DHT22 Sensor with I2C Display

```
sketch_jan25b_DHT22_with_I2C | Arduino IDE 2.3.4
File Edit Sketch Tools Help

sketch_jan25b_DHT22_with_I2C.ino
1 #include "DHT.h"
2 #include <Wire.h>
3 #include <LiquidCrystal_I2C.h>
4
5 #define DHTPIN 2
6 #define potPin A0
7 #define buzzerPin 11
8
9 int x;
10 int y;
11
12 DHT dht(DHTPIN, 22);
13
14 LiquidCrystal_I2C lcd(0x27, 16, 2);
15
16 byte S0 [8] {
17     B00011,
18     B00011,
19     B00000,
20     B00000,
21     B00000,
22     B00000,
23     B00000,
24     B00000
25 };
26
27 byte S1 [8] {
28     B01110,
29     B10001,
30     B10000,
31     B10000,
32     B10000,
33     B10000,
34     B10001,
35     B01110
36 };
37
38
```

```
sketch_jan25b_DHT22_with_I2C | Arduino IDE 2.3.4
File Edit Sketch Tools Help

sketch_jan25b_DHT22_with_I2C.ino
38
39 void setup() {
40     Serial.begin(9600);
41     lcd.init();
42     lcd.init();
43     lcd.backlight();
44     pinMode(buzzerPin, OUTPUT);
45     dht.begin();
46     lcd.createChar (1, S0);
47     lcd.createChar (2, S1);
48 }
49
50 void loop() {
51
52     x=analogRead(potPin);
53     y=map(x,1024,0,25,35);
54
55     delay(2000);
56
57     int h = dht.readHumidity();
58     int t = dht.readTemperature();
59
60     Serial.print(F("Humidity: "));
61     Serial.print(h);
62     Serial.print(F("% Temperature: "));
63     Serial.print(t);
64     Serial.println(F("°C "));
65
66     lcd.setCursor(0,0);
67     lcd.print(F("Humidity: "));
68     lcd.print(h);
69     lcd.print(F("%"));
70     lcd.setCursor(0,1);
71     lcd.print(F("Temperature:"));
72     lcd.print(t);
73     custom_chr();
74
75 }
76
77
78 void custom_chr () {
79     lcd.write(1);
80     lcd.write(2);
81 }
82
```

Signature -

Page No - 53

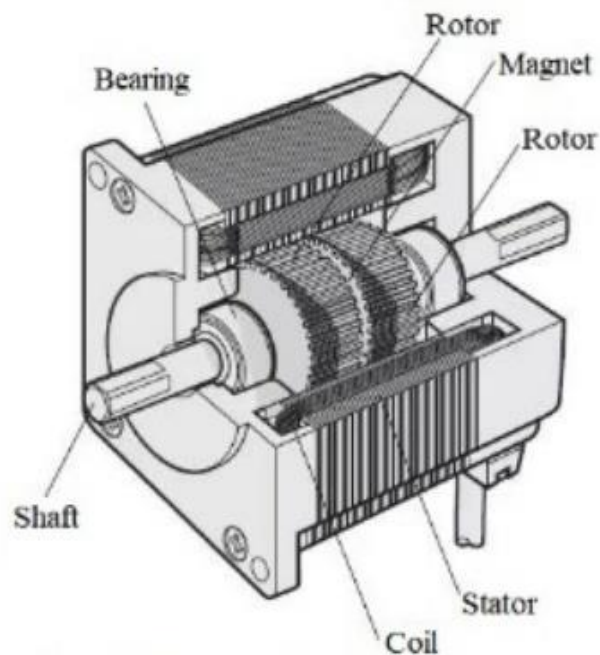
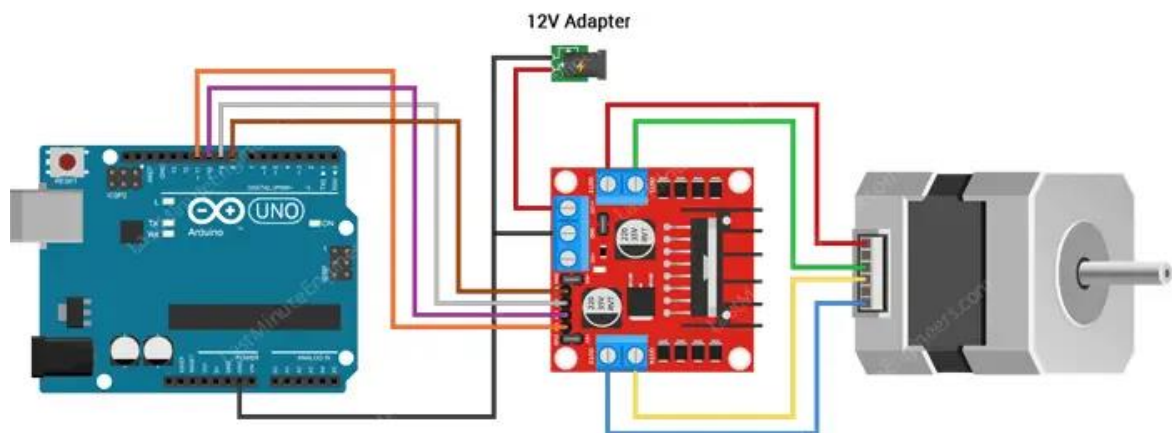


Date -

Topic -

## 36. Stepper Motor

- ❖ stepper motor is a brushless DC electric motor that rotates in a series of small and discrete angular steps. Stepper motors can be set to any given step position without needing a position sensor for feedback.



Signature -

Page No - 54



Date -

Topic -

```
stepper_oneRevolution | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Select Board

stepper_oneRevolution.ino
1  #include <Stepper.h>
2
3  const int stepsPerRevolution = 100; // change this to f
4
5  // initialize the stepper library on pins 8 through 11:
6  Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);
7
8  void setup() {
9      // set the speed at rpm:
10     myStepper.setSpeed(200);
11     // initialize the serial port:
12     Serial.begin(9600);
13 }
14
15 void loop() {
16     // step one revolution in one direction:
17     Serial.println("clockwise");
18     myStepper.step(stepsPerRevolution);
19     delay(500);
20
21     // step one revolution in the other direction:
22     Serial.println("counterclockwise");
23     myStepper.step(-stepsPerRevolution);
24     delay(500);
25 }
26
27
```

Signature -

Page No - 55

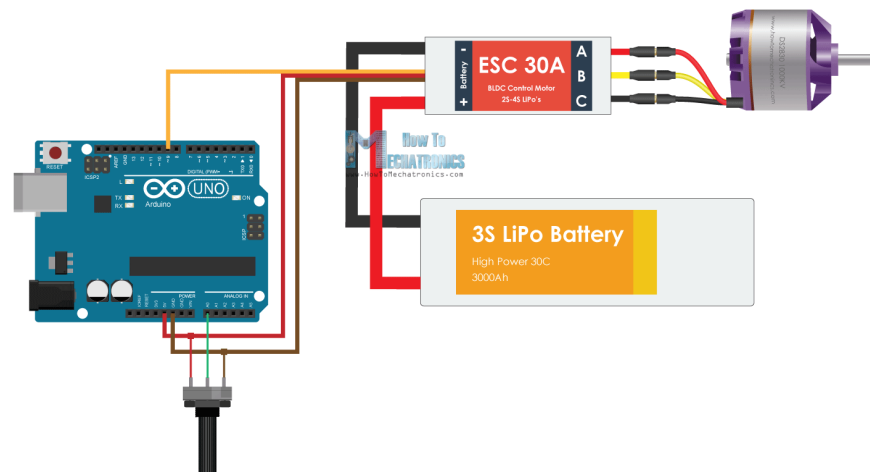




Date -

Topic -

## 37. Brushless DC Motor



```
sketch_feb1a_brushless_dc_motor_drive | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Select Board
sketch_feb1a_brushless_dc_motor_drive.ino
1 //drone motor
2
3 #include <Stepper.h>
4 #include <Servo.h>
5
6 Servo Baseservo;
7
8 void setup() {
9   Baseservo.attach(3);
10  pinMode(3,OUTPUT); //motor drive signal pin
11  pinMode(A0, INPUT); //pot pin
12 }
13
14 void loop() {
15
16   int a = map(analogRead(A0), 0, 1023, 0, 180);
17
18   Baseservo.write(a); // Send PWM signal to control motor speed
19
20   delay(100); // delay for holding the motor
21 }
22
```

Signature -

Page No - 56



Date -

Topic -

## 38. Arduino Software Serial

- ❖ Arduino Software Serial library allows users to create additional serial communication on the Arduino board using software. This is especially used when more serial ports needed than the hardware provides. With Software Serial any digital pins can be defined as RX (receive) and TX (transmit) pins for communication.

```
SoftwareSerialExample | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Select Board
SoftwareSerialExample.ino
1  #include <SoftwareSerial.h>
2
3  char temp; // this can be any name
4
5  SoftwareSerial mySerial(10, 11); // RX, TX.
6
7  void setup() {
8      // Open serial communications and wait for port to open:
9      Serial.begin(9600);
10     while (!Serial) {
11         ; // wait for serial port to connect. Needed for native USB port only
12     }
13
14
15     Serial.println("Goodnight moon!");
16
17     // set the data rate for the SoftwareSerial port
18     mySerial.begin(9600);
19     mySerial.println("Hello, world?");
20 }
21
22 void loop() { // run over and over
23     if (mySerial.available()) {
24         Serial.write(mySerial.read());
25     }
26     if (Serial.available()) {
27         mySerial.write(Serial.read());
28     }
29 }
30
```

Signature -

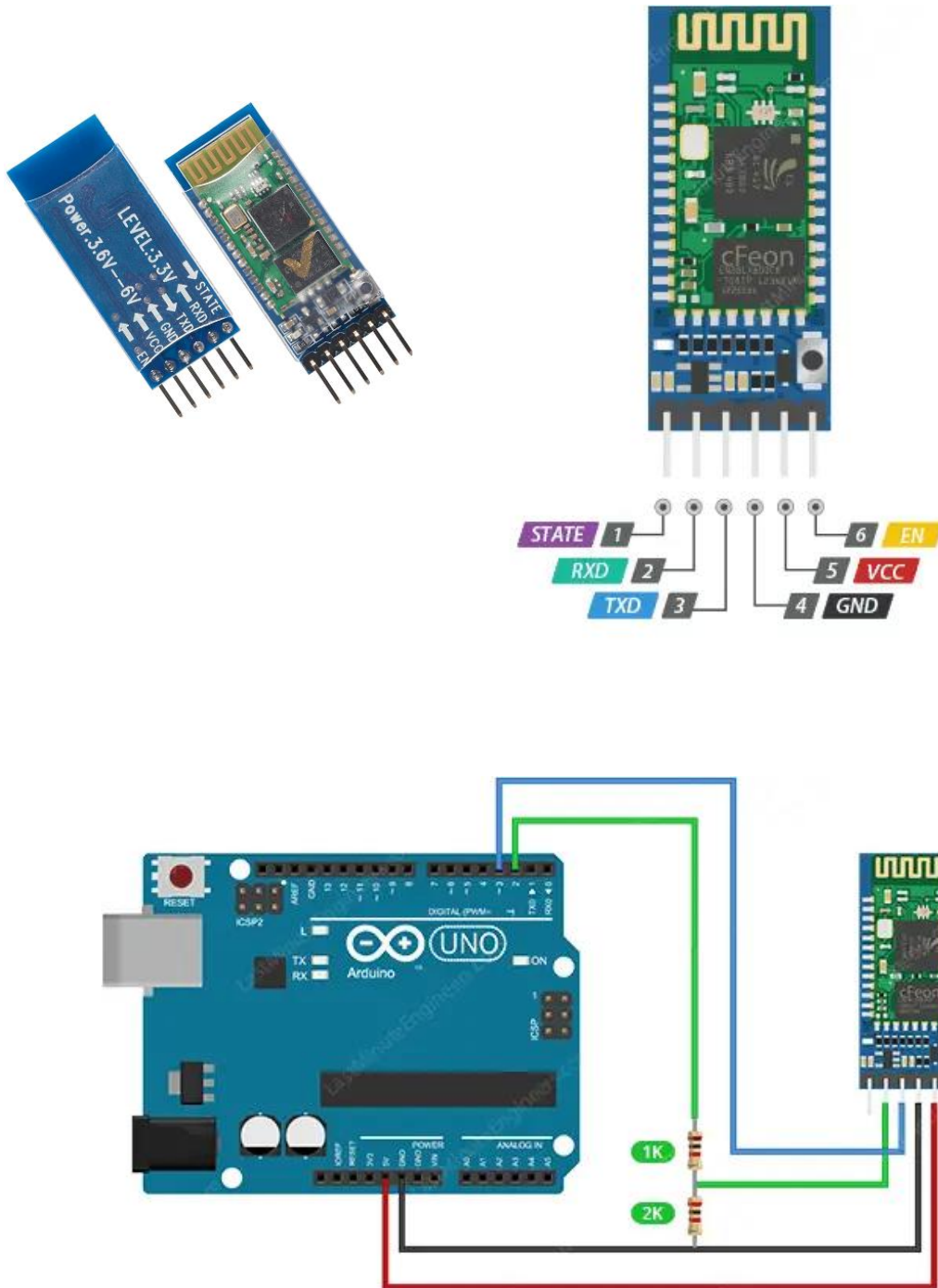
Page No - 57



Date -

Topic -

### 39. Bluetooth Module



Signature -

Page No - 58



Date -

Topic -

- ❖ Bluetooth module is a specialized chip that wirelessly connects two compatible devices for communication using low energy waves. These modules act as the interface between the microcontroller and the devices.

```
etch_feb1d_LED_ONOFF_With_bluetooth | Arduino IDE 2.3.4
Edit Sketch Tools Help
Select Board
sketch_feb1d_LED_ONOFF_With_bluetooth.ino
1  #include <SoftwareSerial.h>
2
3  char temp;
4
5  SoftwareSerial mySerial(10, 11); // RX,
6
7  void setup() {
8    // Open serial communications and wait
9    pinMode(5,OUTPUT);
10   Serial.begin(9600);
11   while (!Serial) {
12     ; // wait for serial port to connect
13   }
14
15
16   Serial.println("Goodnight moon!");
17
18   // set the data rate for the SoftwareS
19   mySerial.begin(9600);
20   mySerial.println("Hello, world?");
21 }
22
23 void loop() {
24   if (mySerial.available()) {
25     temp = mySerial.read();
26     Serial.print(temp);
27     if(temp=="R"); {
28       digitalWrite(5, HIGH);
29     }
30     if (temp=="Y") {
31       digitalWrite(5,LOW);
32     }
33
34     if(temp=="G"); {
35       digitalWrite(6, HIGH);
36     }
37     if (temp=="B") {
```

```
37     if (temp=="B") {
38       digitalWrite(6,LOW);
39     }
40   }
41   if (Serial.available()) {
42     mySerial.write(Serial.read());
43   }
44 }
45
```



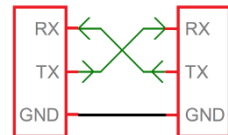
Date -

Topic -

## 40. Communication of 2 UNO Boards



- ❖ This communication typically occurs via Serial UART Communication, utilizing three pins: Rx (receive), Tx (transmit), and GND (ground). In this particular project, the **master** Arduino sends commands to the **slave** Arduino through the Serial Tx Pin and receives responses via the Serial Rx Pin.



### 40.1. Master Arduino's Code

```
sketch_feb08b_communication_of_2_uno_boards_Master_ | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Select Board
sketch_feb08b_communication_of_2_uno_boards_Master_.ino
1  #include <SoftwareSerial.h>
2
3  SoftwareSerial mySerial(10, 11); // RX, TX
4
5  void setup() {
6      Serial.begin(9600);
7      mySerial.begin(9600);
8  }
9
10 void loop() { // run over and over
11     mySerial.println("Hello world from Master");
12     delay(1000);
13     if (mySerial.available()) {
14         String recivedData = mySerial.readStringUntil('\n');
15         Serial.println ("Recived: " + recivedData);
16     }
17 }
18
```



Date -

Topic -

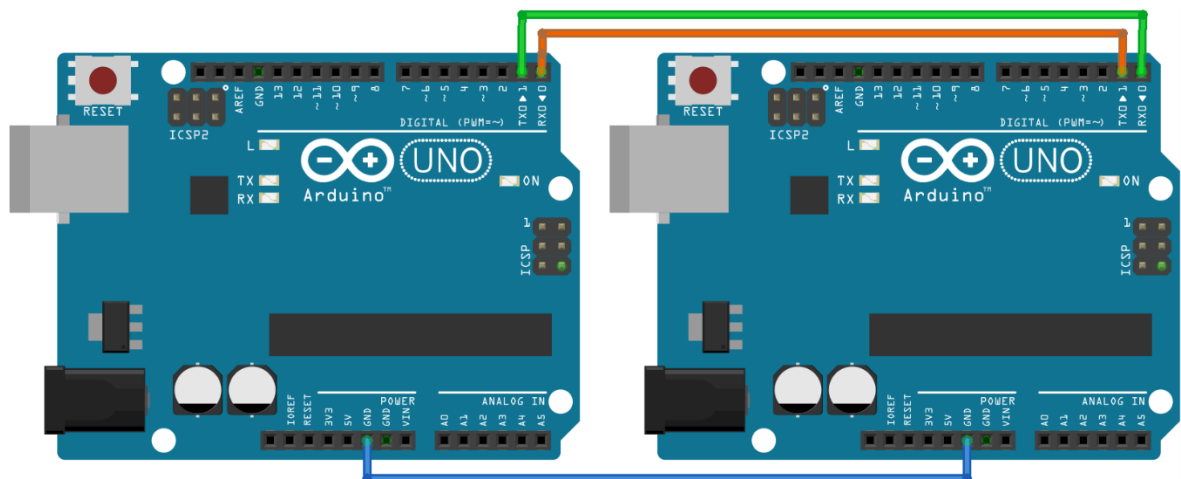
## 40.2. Slave Arduino's Code

```
sketch_feb08c_communication_of_2_uno_boards_Slave_ | Arduino IDE 2.3.4
File Edit Sketch Tools Help

[Checkmark] [Next] [Upload] [Select Board]

sketch_feb08c_communication_of_2_uno_boards_Slave_.ino

1  #include <SoftwareSerial.h>
2
3  SoftwareSerial mySerial(10, 11); // RX, TX
4
5  void setup() {
6    Serial.begin(9600);
7    mySerial.begin(9600);
8  }
9
10 void loop() { // run over and over
11   mySerial.println("Hello world from Slave");
12   delay(1000);
13   if (mySerial.available()) {
14     String recivedData = mySerial.readStringUntil('\n');
15     Serial.println ("Recived: " + recivedData);
16   }
17 }
18
```



Signature -

Page No - 61



Date -

Topic -

### 40.3. Control LED Using 2 Arduino Boards as Master and Slave

```
sketch_feb08d_comunication_led_on_off | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Checkmark] [Next] [Run] [Select Board]
sketch_feb08d_comunication_led_on_off.ino
1 #include <SoftwareSerial.h>
2
3 SoftwareSerial mySerial(10, 11); // RX, TX
4
5 void setup() {
6   Serial.begin(9600); // Serial monitor
7   mySerial.begin(9600); // Software Serial
8 }
9
10 void loop() {
11   // put your main code here, to run repeatedly;
12   mySerial.available("1");
13   delay(1000);
14
15   if (mySerial.available()) {
16     String receivedData = mySerial.readStringUntil("\n");
17     Serial.print("Received: " + receivedData);
18
19     num = receivedData.toInt();
20
21     if(num==1) {
22       digitalWrite(13, HIGH);
23     }
24     else {
25       digitalWrite(13, LOW);
26     }
27
28     if(num==2) {
29       digitalWrite(13, HIGH);
30     }
31     else {
32       digitalWrite(13, LOW);
33     }
34   }
35   if (Serial.available()) {
36     mySerial.write(Serial.read());
37   }
38 }
39
```

Signature -

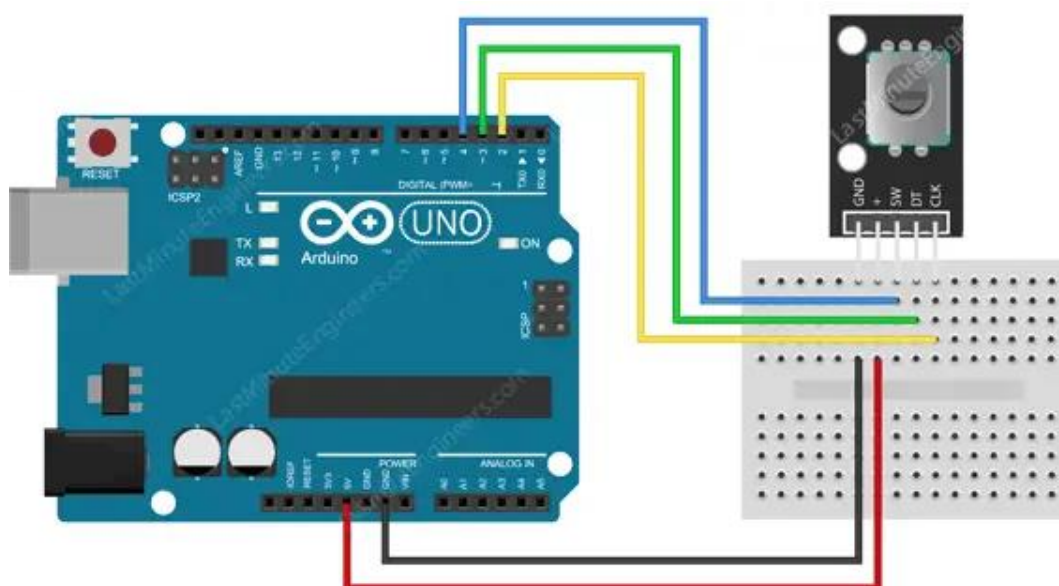
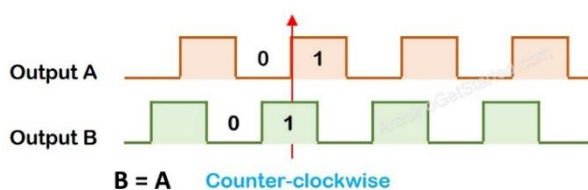
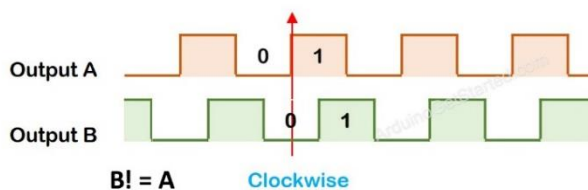
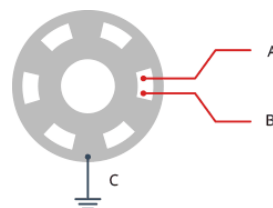
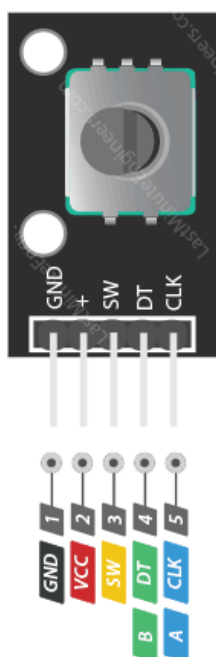
Page No - 62



Date -

Topic -

## 41. Encoder Module



Signature -

Page No - 63





Date -

Topic -

- ❖ Encoder modules constitute an integral part of various digital signal processing systems and artificial intelligence models. These modules encode the input data into a format that is easier for the system or model to interpret and process.

```
sketch_feb15a_Encorder | Arduino IDE 2.3.4
File Edit Sketch Tools Help

sketch_feb15a_Encorder.ino
1  #define OutA 5
2  #define OutB 6
3
4  void setup() {
5      Serial.begin(9600);
6      pinMode(OutA, INPUT_PULLUP);
7      pinMode(OutB, INPUT_PULLUP);
8  }
9
10
11 void loop() {
12     int A=digitalRead(OutA);
13     int B=digitalRead(OutB);
14
15     if((A==HIGH) && (B==LOW)) {
16         Serial.print("Clockwise");
17         Serial.println("\t");
18         delay(1);
19     }
20
21     if((A==LOW) && (B==LOW)) {
22         Serial.print(" Counterclockwise");
23         Serial.println("\t");
24         delay(1);
25     }
26
27 }
28
```

Signature -

Page No - 64



Date -

Topic -

### 41.1. Encoder and Counter

```
sketch_feb15a_Encoeder_and_counter | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Select Board
sketch_feb15a_Encoeder_and_counter.ino
1  #define CLK 5
2  #define DT 6
3  int counter=0;
4  int lastStateCLK;
5  int currentStateCLK;
6  int currentDir;
7
8  void setup() {
9      Serial.begin(9600);
10     pinMode(CLK,INPUT_PULLUP);
11     pinMode(DT,INPUT_PULLUP);
12     lastStateCLK = digitalRead(CLK);
13 }
14
15 void loop() {
16     currentStateCLK = digitalRead(CLK);
17     if(currentStateCLK != lastStateCLK && currentStateCLK == 1) {
18         if (digitalRead(DT) != currentStateCLK) {
19             counter --;
20             currentDir = "CCW";
21         } else {
22             counter ++;
23             currentDir ="CW";
24         }
25
26         Serial.print("direction: ");
27         Serial.print(currentDir);
28         Serial.print(". Counter: ")
29         Serial.println(counter);
30     }
31     lastStateCLK = currentStateCLK;
32     delay(1)
33 }
34
35
```

Signature -

Page No - 65



Date -

Topic -

## 41.2. Encoder 1-5 Count

```
sketch_feb15c_Encorder_1-5_Count | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Select Board
sketch_feb15c_Encorder_1-5_Count.ino
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3
4  LiquidCrystal_I2C lcd(0x27, 16,2);
5
6  #define CLK 5
7  #define DT 6
8
9  int counter=0;
10 int lastStateCLK;
11 int currentStateCLK;
12 String currentDir = "";
13
14 void setup() {
15     Serial.begin(9600);
16     pinMode(CLK,INPUT_PULLUP);
17     pinMode(DT,INPUT_PULLUP);
18     lastStateCLK = digitalRead(CLK);
19     lcd.init();
20     lcd.init();
21     lcd.backlight();
22
23 }
24
25 void loop() {
26     currentStateCLK = digitalRead(CLK);
27     if(currentStateCLK != lastStateCLK && currentStateCLK == 1) {
28         if (digitalRead(DT) != currentStateCLK) {
29             counter --;
30             if (counter==--1) {
31                 counter = 5;
32             }
33             currentDir = "CCW ";
34
35         } else {
36             counter ++;
37             if (counter==6) {
```

Signature -

Page No - 66



Date -

Topic -

```
36         counter ++;
37         if (counter==6) {
38             counter = 0;
39         }
40         currentDir ="CW ";
41     }
42
43     Serial.print("direction: ");
44     Serial.print(currentDir);
45     Serial.print(". Counter: ");
46     Serial.println(counter);
47 }
48 lastStateCLK = currentStateCLK;
49 delay(1);
50
51 lcd.setCursor(0,0);
52 lcd.print("Count: ");
53 lcd.print(counter);
54 lcd.setCursor(0,1);
55 lcd.print("Direction: ");
56 lcd.print(currentDir);
57
58 }
59
```

Signature -

Page No - 67



Date -

Topic -

### 41.3. Encoder with Counter and LCD

```
sketch_feb15b_Encoder_with_conter_and_I2C | Arduino IDE 2.3.4
File Edit Sketch Tools Help

[Checkmark] [Next] [Upload] [Select Board]

sketch_feb15b_Encoder_with_conter_and_I2C.ino
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3
4  LiquidCrystal_I2C lcd(0x27, 16,2);
5
6  #define CLK 5
7  #define DT 6
8
9  int counter=0;
10 int lastStateCLK;
11 int currentStateCLK;
12 String currentDir = "";
13
14 void setup() {
15     Serial.begin(9600);
16     pinMode(CLK, INPUT_PULLUP);
17     pinMode(DT, INPUT_PULLUP);
18     lastStateCLK = digitalRead(CLK);
19     lcd.init();
20     lcd.init();
21     lcd.backlight();
22 }
23
24 void loop() {
25     currentStateCLK = digitalRead(CLK);
26     if(currentStateCLK != lastStateCLK && currentStateCLK == 1) {
27         if (digitalRead(DT) != currentStateCLK) {
28             counter --;
29             currentDir = "CCW ";
30         }
31     }
32 }
```

Signature -

Page No - 68



Date -

Topic -

```
29         counter --;
30         currentDir = "CCW ";
31     } else {
32         counter ++;
33         currentDir ="CW ";
34     }
35
36     Serial.print("direction: ");
37     Serial.print(currentDir);
38     Serial.print(". Counter: ");
39     Serial.println(counter);
40 }
41 lastStateCLK = currentStateCLK;
42 delay(1);
43
44 lcd.setCursor(0,0);
45 lcd.print("Count: ");
46 lcd.print(counter);
47 lcd.setCursor(0,1);
48 lcd.print("Direction: ");
49 lcd.print(currentDir);
50
51 }
52
```

Signature -

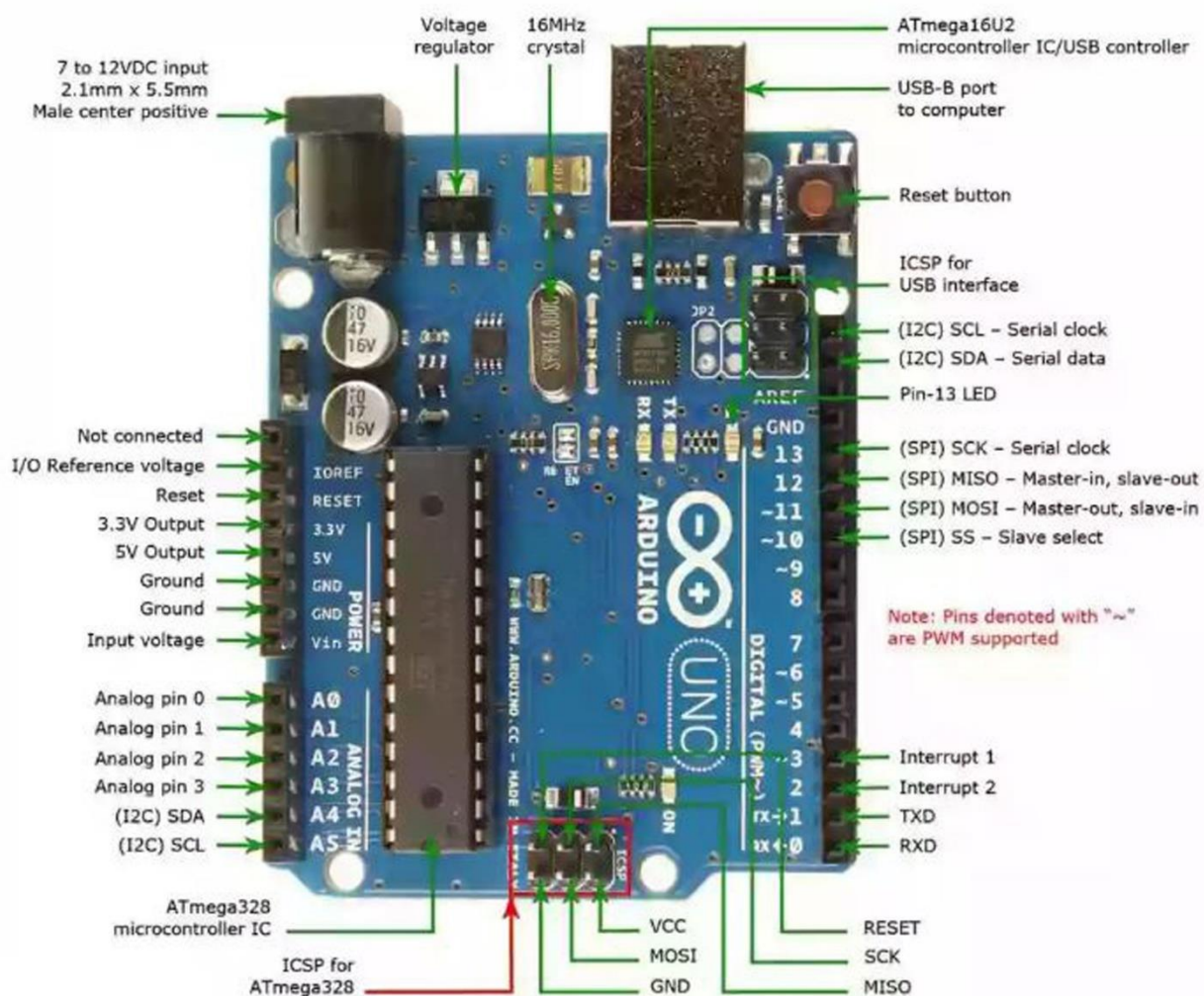
Page No - 69



Date -

Topic -

## 42. Arduino UNO Board





Date -

Topic -

PWM output pins → 3, 5, 6, 9, 10, 11

PWM input pins → 2, 3

#### **42.1. Digital Input/Output Pins**

The Arduino Uno board has 14 digital input/output (I/O) pins labeled from 0 to 13. These 14 pins can be configured as either digital inputs or digital outputs. Out of 14 pins, 6 pins can be used as PWM (Pulse Width Modulation) output pins and 2 can be used as PWM (Pulse Width Modulation) input pins.

PWM out pins → 3, 5, 6, 9, 10, and 11

PWM in pins → 2 and 3.

#### **42.2. Analog Input Pins**

The Arduino Uno board provides 6 analogue input pins labelled from A1 to A5. These pins are useful for reading values from sensors, such as temperature sensors, light sensors, and potentiometers.

Voltage Regulator It controls the voltage provided to the microcontroller and other onboard components.

#### **42.3. Microcontroller**

This is the main controller IC on which Arduino Code is stored. It is **Atmel ATMEGA328P** microcontroller. It is the Brain of the Arduino Board, When Arduino board is powered up it executes the instructions stored in its 32KB flash memory.

USB Connector A USB connector allows users to connect it directly to a computer or other devices for programming and communication. It serves multiple purposes. Firstly, it provides a means to power the board during development and testing phases. Secondly, the USB connection allows users to upload code to the board and facilitates serial communication between the Arduino Uno and the computer, allowing data exchange and debugging.

Signature -

Page No - 71





Date -

Topic -

#### 42.4. Arduino Uno Board Specifications

Microcontroller	ATmega328P
Operating voltage	5V
Input voltage	7-12V
Flash memory	32KB
SRAM	2KB
On chip EEPROM	1KB
Number of analog inputs	6
Number of digital I/O	14 (6 of them Pulse Width Modulation output pins)
Clock speed	16 MHz

Arduino R3 has three types of memory

1. **Flash Memory (ROM)** - Stores the program/code (non-volatile)
  2. **SRAM (RAM)** - Stores temporary data (volatile)
  3. **EEPROM** - Stores data permanently, even after power is off
- Flash Memory (ROM): Stores your program code permanently.
  - SRAM (RAM): Stores variables during program execution (limited!).
  - EEPROM: Stores user data/settings permanently.

#### 42.5. Logical Operators

Logical Operator	Operator Symbol	Example
OR		a    b
AND	&&	a && b
NOT	!	! a

Signature -

Page No - 72



Date -

Topic -

## 1. For Loop -

A for loop is a loop that is used to perform an action a specified number of times.

```
FOR_loop | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Checkmark] [Next] [Upload] [USB] DOIT ESP32 DEVKIT V1
FOR_loop.ino
1  int ledPin = 13;
2
3  void setup() {
4      pinMode(ledPin, OUTPUT);
5  }
6
7  void loop() {
8      for (int i = 0; i < 5; i++) {
9          digitalWrite(ledPin, HIGH);
10         delay(500);
11         digitalWrite(ledPin, LOW);
12         delay(500);
13     }
14 }
15
```

Signature -

Page No - 73



Date -

Topic -

## 2. While Loop -

A while loop repeats a block of code while a specific condition remains true.

```
While_loop | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Checkmark] [Next] [Upload] [USB] DOIT ESP32 DEVKIT V1
While_loop.ino
1  int ledPin = 13;
2
3  void setup() {
4      pinMode(ledPin, OUTPUT);
5  }
6
7  void loop() {
8      while (true) {
9          digitalWrite(ledPin, HIGH);
10         delay(500);
11         digitalWrite(ledPin, LOW);
12         delay(500);
13     }
14 }
15
```

Signature -

Page No - 74



Date -

Topic -

## 43. Electrically Erasable Programmable Read-Only Memory (EEPROM)

- ❖ In Arduino, EEPROM (Electrically Erasable Programmable Read-Only Memory) is a non-volatile memory that retains data even when the board is powered off. Unlike RAM (Random Access Memory) which loses its contents when power is removed, EEPROM retains its data even without power. EEPROM memory, typically around 1024 bytes.

```
eeprom_write | Arduino IDE 2.3.4
File Edit Sketch Tools Help

eeprom_write.ino
1  #include <EEPROM.h>
2
3  int addr = 0;
4  int val = 100;
5
6  void setup() {
7      Serial.begin(9600);
8
9      EEPROM.write(addr, val);
10     Serial.print("Wrote value");
11     Serial.print(value);
12
13     int readvalue = EEPROM.read(addr);
14     Serial.print("Read value");
15     Serial.print(readvalue);
16
17 }
18
19 void loop() {
20
21 }
22
```

Signature -

Page No - 75

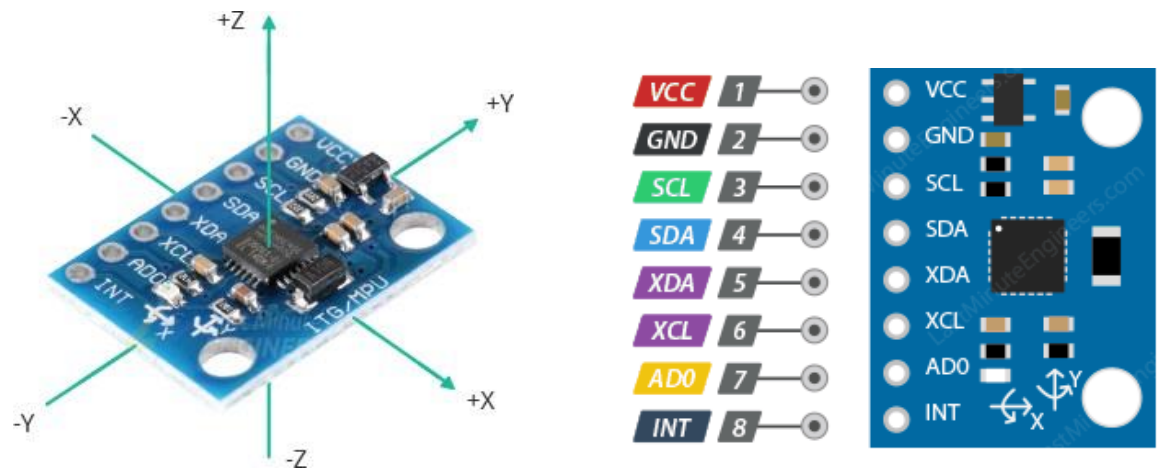


Date -

Topic -

## 44. Gyroscope

Gyroscope module is a device that can measure and maintain the orientation and angular velocity of an object. It is more advanced than an accelerometer, which can only measure linear motion. 3-axis gyroscopes measure angular rate and are usually combined with an accelerometer in a common package to allow advanced algorithms like sensor fusion.



```
sketch_mar1c_Gyroscope | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] Select Board
sketch_mar1c_Gyroscope.ino
1  #include <Wire.h>
2
3  // QMC5883L Register Addresses
4  #define QMC5883L_ADDR 0x0D
5  #define REG_XOUT_LSB 0x00
6  #define REG_CONTROL1 0x09
7  #define REG_CONTROL2 0x0A
8  #define REG_PERIOD 0x0B
9
10 void setup() {
11     Wire.begin();
12     Serial.begin(9600);
13 }
```

Signature -

Page No - 76



Date -

Topic -

```
13
14 // Initialize QMC5883L
15 // Reset the sensor
16 Wire.beginTransmission(QMC5883L_ADDR);
17 Wire.write(REG_CONTROL2);
18 Wire.write(0x80); // Soft Reset
19 Wire.endTransmission();
20 delay(100);
21
22 // Configure Control Register 1
23 Wire.beginTransmission(QMC5883L_ADDR);
24 Wire.write(REG_CONTROL1);
25 Wire.write(0x1D); // Mode: Continuous, ODR: 200Hz, RNG: 8G, OSR: 512
26 Wire.endTransmission();
27
28 // Set Period Register
29 Wire.beginTransmission(QMC5883L_ADDR);
30 Wire.write(REG_PERIOD);
31 Wire.write(0x01); // Recommended value
32 Wire.endTransmission();
33
34 Serial.println("QMC5883L Test");
35 Serial.println("X\tY\tZ");
36 }
37
38 void loop() {
39     int16_t x, y, z;
40     uint8_t status;
41
42     // Read 6 bytes of data
43     Wire.beginTransmission(QMC5883L_ADDR);
44     Wire.write(REG_XOUT_LSB);
45     Wire.endTransmission(false);
46     Wire.requestFrom(QMC5883L_ADDR, 6);
47
48     if(Wire.available() == 6) {
49         x = Wire.read() | Wire.read() << 8;
50         y = Wire.read() | Wire.read() << 8;
51         z = Wire.read() | Wire.read() << 8;
52
53         // Print raw values
54         Serial.print(x);
55         Serial.print("\t");
56         Serial.print(y);
57         Serial.print("\t");
58         Serial.println(z);
59     }
60
61     delay(100); // Delay between readings
62 }
```

Signature -

Page No - 77



Date -

Topic -

## 45. Real-Time Clock Module (RTC)

```
RTC_module | Arduino IDE 2.3.4
File Edit Sketch Tools Help

✓ → ⚙ DOIT ESP32 DEVKIT V1

RTC_module.ino
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3
4  LiquidCrystal_I2C lcd2(0x27, 20, 4);
5  #include <MyRealTimeClock.h>
6  MyRealTimeClock RTC1(6, 7, 8);
7  void setup() {
8      lcd2.init();
9      lcd2.init();
10     lcd2.backlight();
11     RTC1.setDS1302Time(00, 52, 14, 7, 28, 01, 2023);
12     Serial.begin(9600);
13     // put your setup code here, to run once:
14
15 }
16
17 void loop() {
18     RTC1.updateTime();
19     Serial.print("Date: ");
20     Serial.print(RTC1.dayofmonth);
21     Serial.print("/");
22     Serial.print(RTC1.month);
23     Serial.print("/");
24     Serial.print(RTC1.year);
25
26     Serial.print("  Time: ");
27     Serial.print(RTC1.hours);
28     Serial.print(":");
29     Serial.print(RTC1.minutes);
30     Serial.print(":");
```

Signature -

Page No - 78



Date -

Topic -

```
30 Serial.print(":");
31 Serial.println(RTC1.seconds);
32 delay(1000);
33 lcd2.setCursor(0, 0);
34 lcd2.print("Date: ");
35 lcd2.print(RTC1.dayofmonth);
36 lcd2.print("/");
37 lcd2.print(RTC1.month);
38 lcd2.print("/");
39 lcd2.print(RTC1.year);
40 lcd2.setCursor(0, 1);
41 lcd2.print("Time: ");
42 lcd2.print(RTC1.hours);
43 lcd2.print(":");
44 lcd2.print(RTC1.minutes);
45 lcd2.print(":");
46 lcd2.print(RTC1.seconds);
47
48 }
```

Signature -

Page No - 79



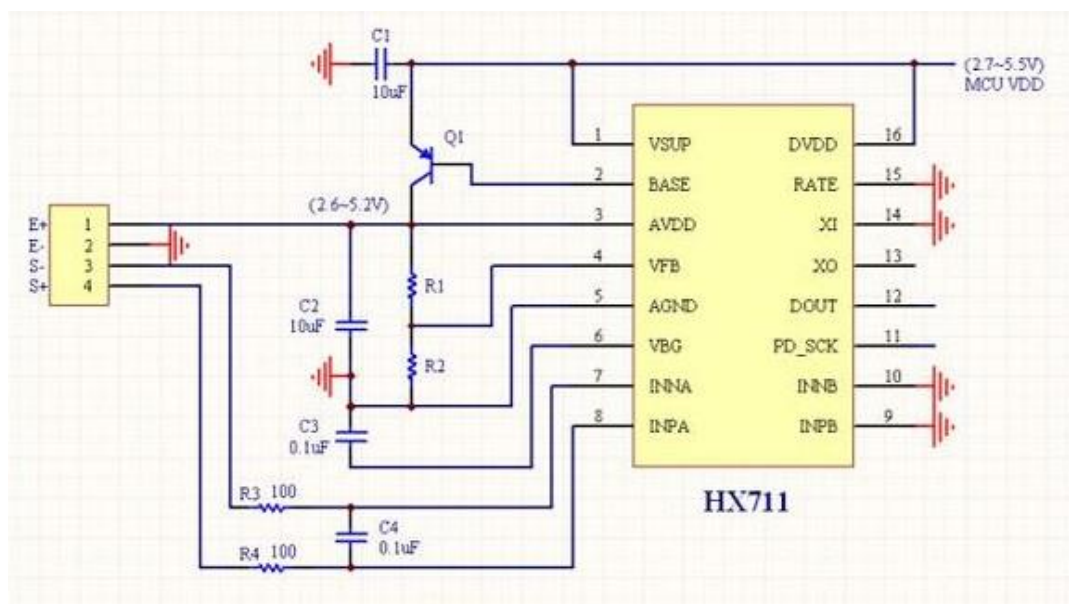
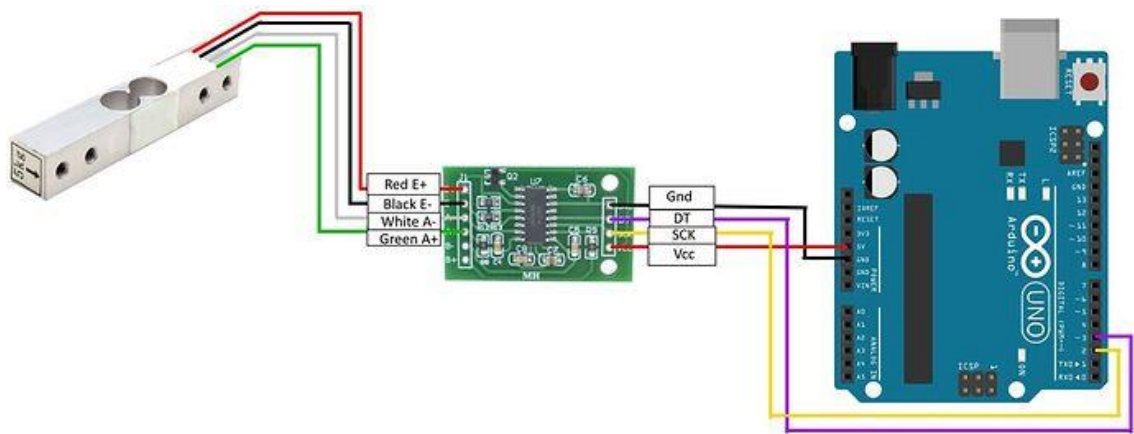


Date -

Topic -

## 46. Load Cell Sensor

A load cell is a type of force transducer that measures force or weight. A load cell converts a force into an electrical signal that can be measured. The electrical signal changes proportionally to the force applied.



Signature -

Page No - 80



Date -

Topic -

```
Load_cell | Arduino IDE 2.3.4
File Edit Sketch Tools Help
[Icons] DOIT ESP32 DEVKIT V1
Load_cell.ino
1  #include "HX711.h"
2  // HX711 circuit wiring
3  const int LOADCELL_DOUT_PIN = 3;
4  const int LOADCELL_SCK_PIN = 2;
5
6  HX711 scale;
7  void setup() {
8      Serial.begin(9600);
9      scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
10 }
11 void loop() {
12     if (scale.is_ready()) {
13         long reading = scale.read();
14         Serial.print("HX711 reading: ");
15         Serial.println(reading);
16     } else {
17         Serial.println("HX711 not found.");
18     }
19     delay(1000);
20 }
```

Signature -

Page No - 81

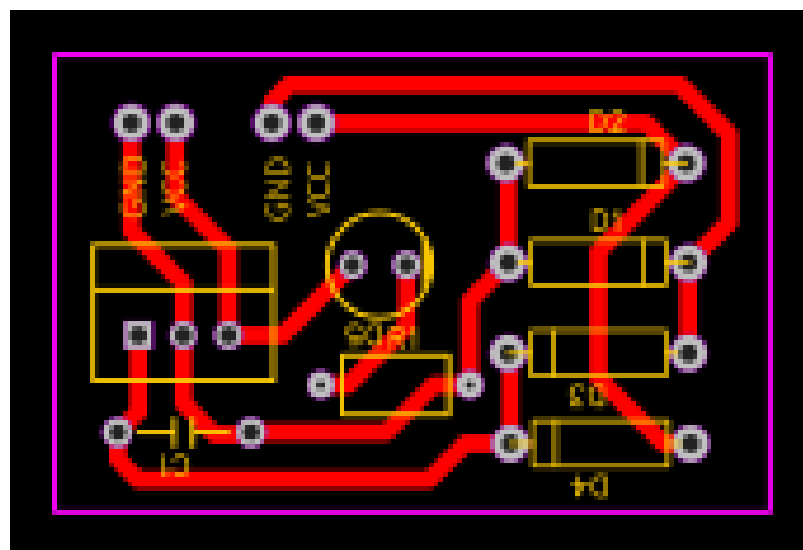
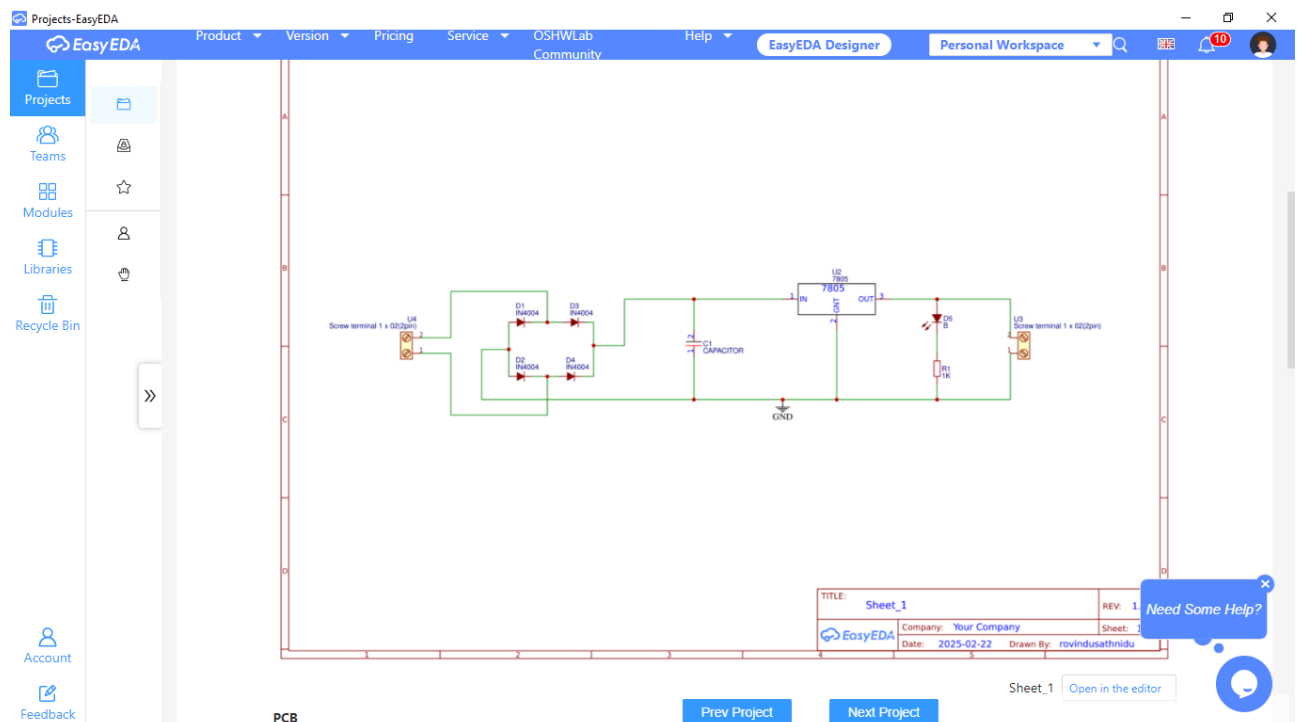


Date -

Topic -

## 47. PCB Designing

### 47.1. Rectifier Circuit



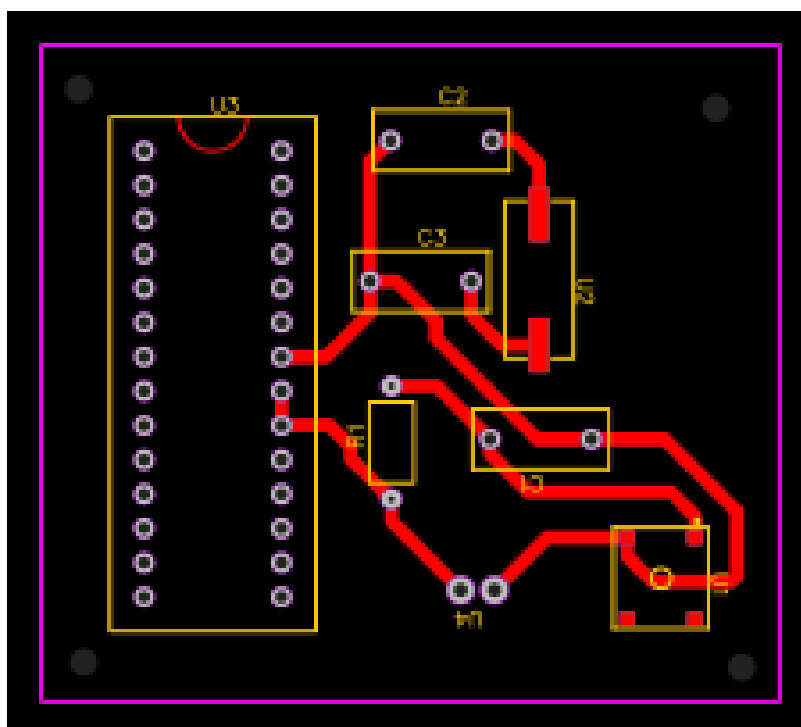
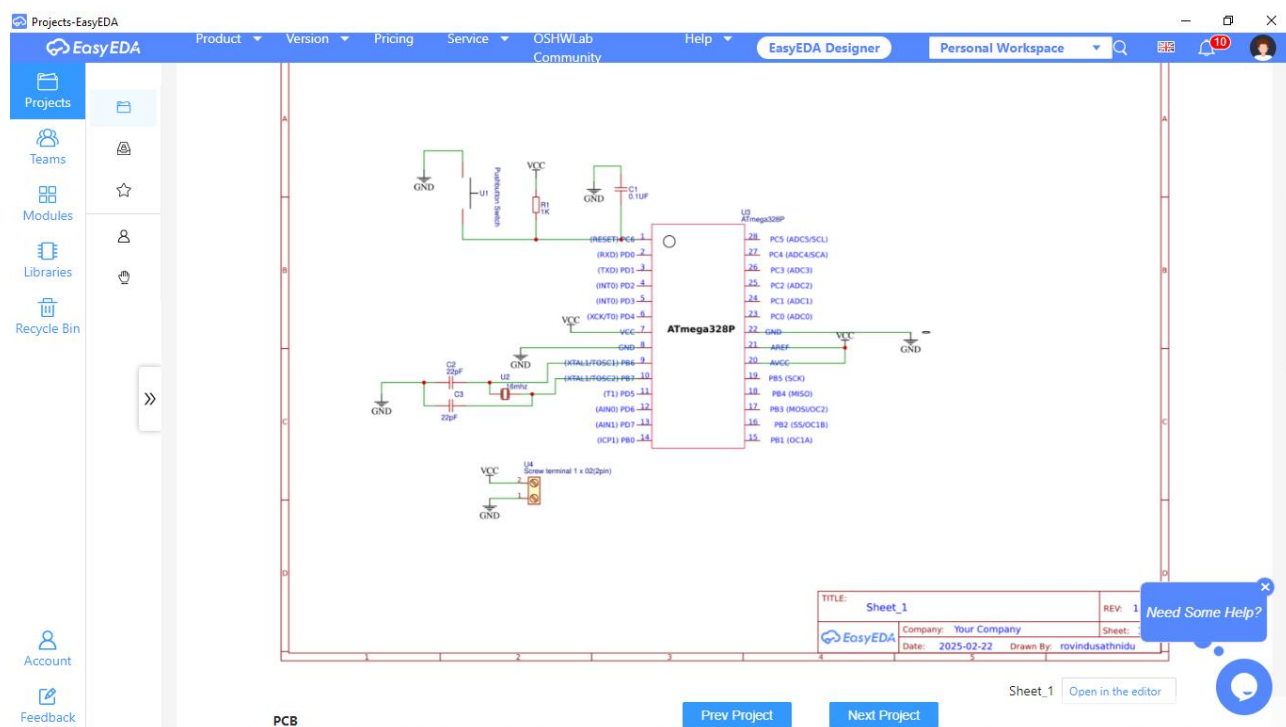
Signature -

Page No - 82

Date -

Topic -

## 47.2. Arduino Board Design Using ATmega 328p



Signature -

Page No - 83



**Ceylon German Technical Training Institute**

**Micro Control & Robotics (MCR)**

**Student Portfolio**

Date -

Topic -

Signature -

Page No - 84



**Ceylon German Technical Training Institute**

**Micro Control & Robotics (MCR)**

**Student Portfolio**

Date -

Topic -

Signature -

Page No - 85