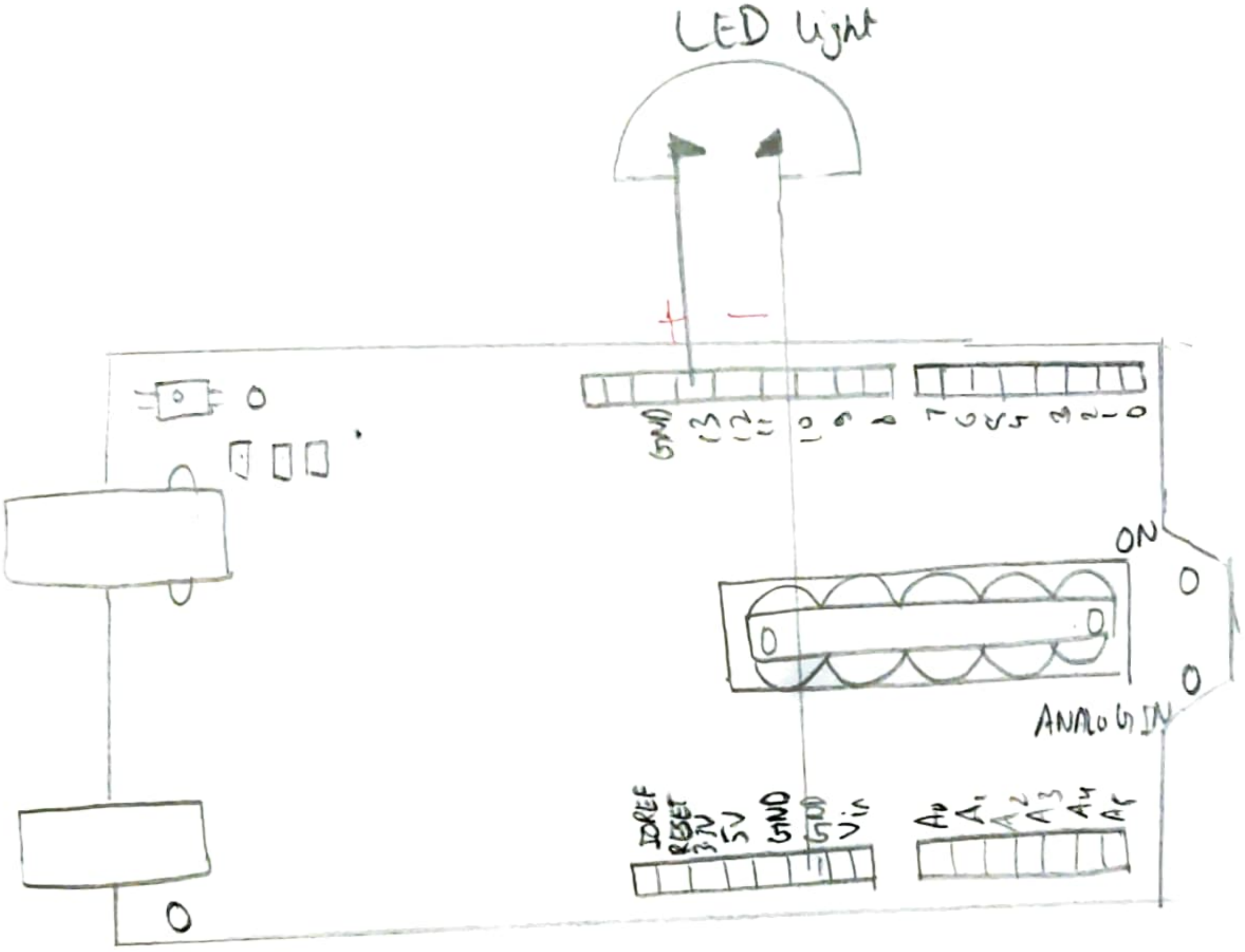


Interfacing diagram

Experiment - 1

☆ **Aim:-** Interfacing LED with Ardiuno Uno

☆ **Component used:-** Ardiuno Uno, LED & Wires

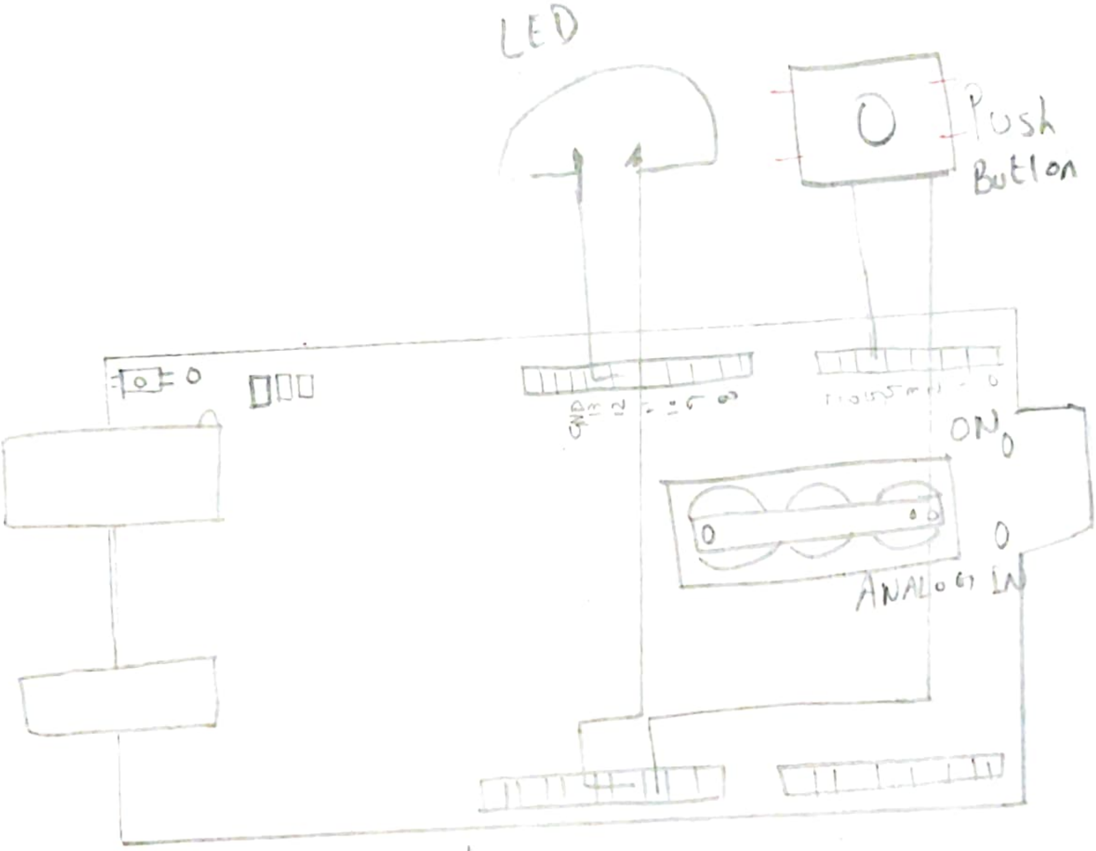
☆ **Softwar used:-** WokWi

☆ **Interfacing code:-**

```
Void setup()
 {
     ~~printer~~ pinMode (13, output); // set LED Pin as output
 }
void Loop()
{
   digitalWrite (13, HIGH);        // Turn on (HIGH) LED
   delay (500);
   digital Write (13, Low);        // Turn off (LOW) LED
   delay(500);
}
```

☆ **Result:**

We have successfully interfaced the LED with Ardiuno Uno and it is showing output succesfully.



Interfacing diagram

Experiment -2

☆ Aim: Interfacing LED with pushbutton with Ardiuno Uno

☆ Component used: Ardiuno Uno, LED, Pushbutton & wires

☆ Software used: WoKwi

☆ Interfacing code:

```
const int button Pin = 4;  //Pin connected to push button
const int ~~butto~~ledPin =12;  //Pin connected to LED
int buttonState;         // Give pushbutton a value
Void setup()
{
  pinMode (ledPin, OUTPUT);      //set LED pin as output
  pinMode(button Pin, INPUT);    //set pushbutton pin as input

 }
 void loop()
 { buttonState= digitalRead(buttonPin); //Read input from pin 4
   if (buttonState = = low)
     { digitalState(ledPin, HIGH); //Turn on LED
       delay(500);
       digitalWrite(ledPin, LOW);
       delay(500);
     }
   else {
     digitalWrite(ledPin, LOW); // Turn off LED
     }
  }
```

☆ Result: We have successfully interfaced LED with pushbutton with Ardiuno Uno and it is showing output successfully.

Experiment-3

☆ Aim: Interfacing seven-segement display with Arduino Uno.

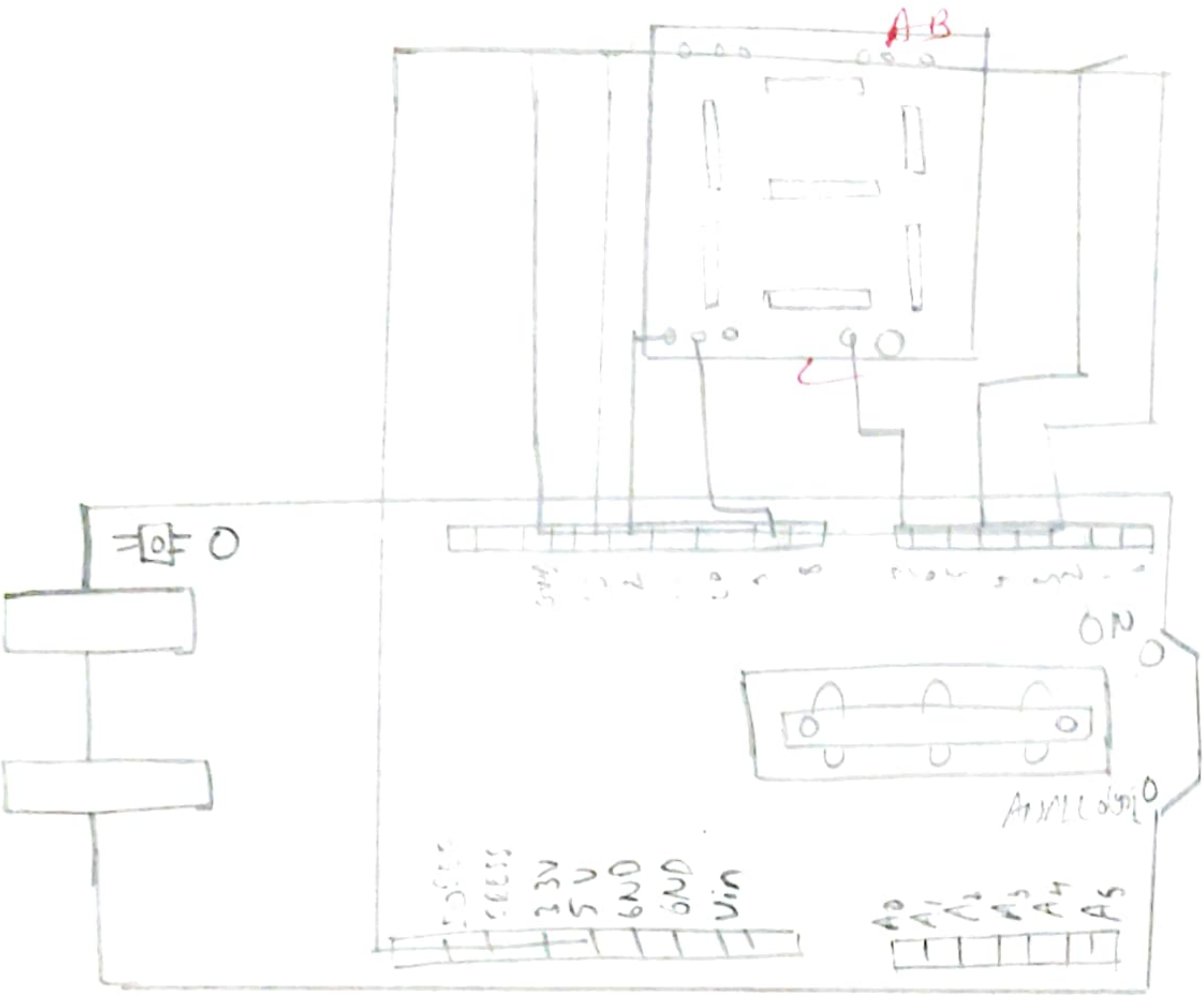
☆ Component used: Arduino Uno, seven-segement display and wires

☆ Software used: wokwi

☆ Interfacing code:

```
int a = 13;        //Pin connected to seven segements
int b = 12;
int c = 11;
int d = 10;
int e = 9;
int f = 8;
int g = 7;
void setup()
{     pinMode (a, output);  // set the seven segament as output
      pinMode(b, outPut);
      pinMode (c, OUTPUT);
      pinMode (d, OUTPUT);
      pinMode(e, OUTPUT);
      pinMode(f, OUTPUT);
      pinMode(g, OUTPUT);
}
Void loop()        //To display 0
{     digitalWrite(a,0);
      digitalWrite (b,0);
      digitalWrite (c,0);
      digitalWrite (d,0);
      digitalWrite (e,0);
      digitalWrite (f,0);
      digitalWrite (g,1);
```

```
delay(1000);
digitalWrite(a,1);          // To display 1
digitalWrite(b,0);
digitalWrite(c,0);
digitalWrite(d,1);
digitalWrite(e,1);
digitalWrite(f,1);
digitalWrite(g,1);
delay(1000);

digitalWrite(a,0);          // To display 2
digitalWrite(b,0);
digitalWrite(c,1);
digitalWrite(d,0);
digitalWrite(e,0);
digitalWrite(f,1);
digitalWrite(g,0);
delay(1000);

digitalWrite(a,0);          // To display 3
digitalWrite(b,0);
digitalWrite(c,0);
digitalWrite(d,0);
digitalWrite(e,1);
digitalWrite(f,1);
digitalWrite(g,0);
delay(1000);

digitalWrite(a,1);          // To display 4
digitalWrite(b,0);
digitalWrite(c,0);
digitalWrite(d,1);
digitalWrite(e,1);
digitalWrite(f,0);
digitalWrite(g,0);
delay(1000);
```



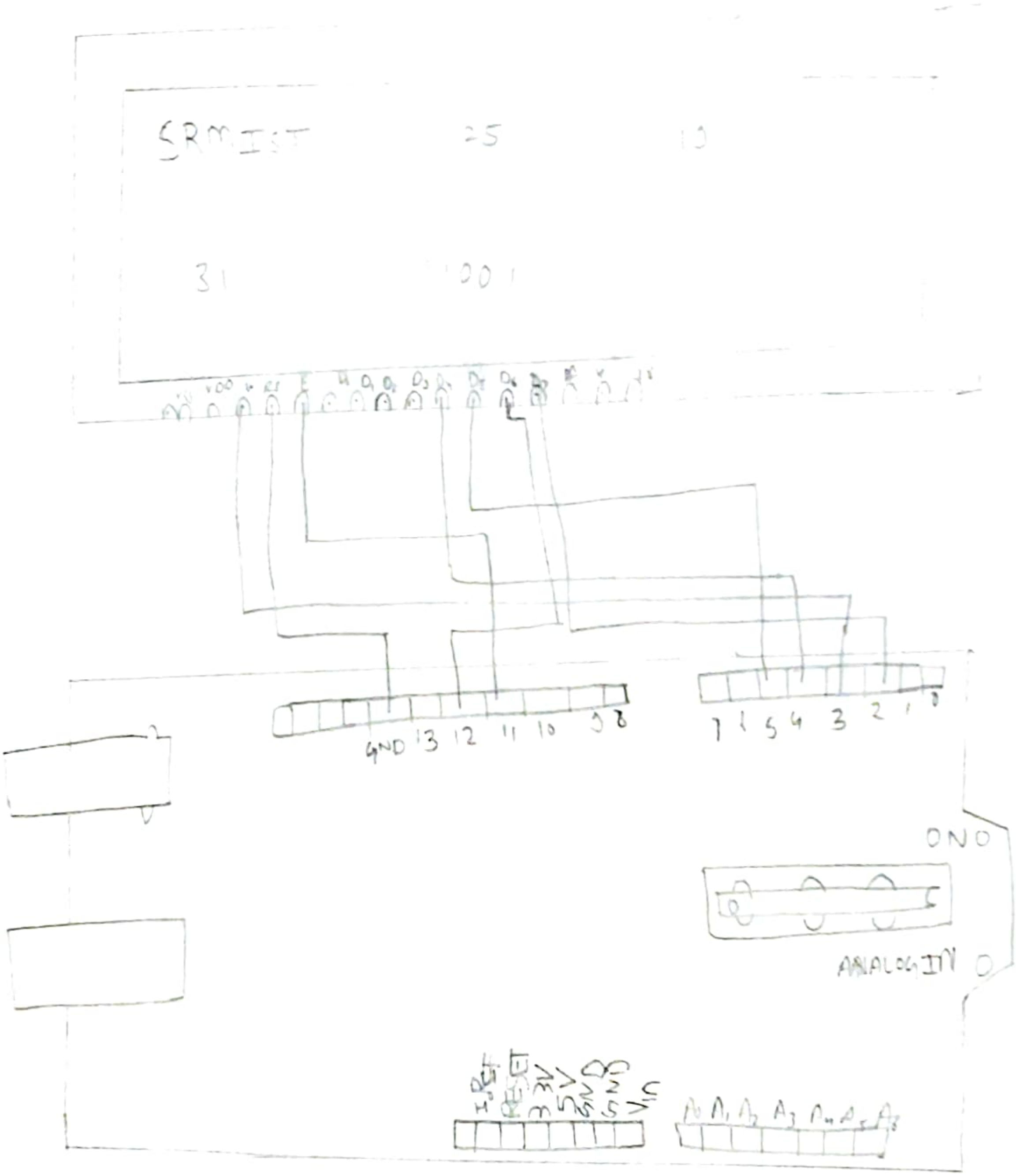
Interfacing diagram

```
digitalWrite(a,0);          // To display 5
digitalWrite(b,1);
digitalWrite(c,0);
digitalWrite(d,0);
digitalWrite(e,1);
digitalWrite(f,0);
digitalWrite(g,0);
delay(1000);

digitalWrite(a,0);          // To display 6
digitalWrite(b,1);
digitalWrite(c,0);
digitalWrite(d,0);
digitalWrite(e,0);
digitalWrite(f,0);
digitalWrite(g,0);
delay(1000);

digitalWrite(a,0);          // To display 7
digitalWrite(b,0);
digitalWrite(c,0);
digitalWrite(d,1);
digitalWrite(e,1);
digitalWrite(f,1);
digitalWrite(g,1);
delay(1000);

digitalWrite(a,0);          // To display 8
digitalWrite(b,0);
digitalWrite(c,0);
digitalWrite(d,0);
digitalWrite(e,0);
digitalWrite(f,0);
digitalWrite(g,0);
delay(1000);
```

```
digitalWrite(a,0);          // To display 9
digitalWrite(b,0);
dijital Write(c,0);
dijitalWrite(d,1);
dijitalWribe(e,1);
dijital Write (f,0);
dijitalWrite (g,0);
delay (1000);
}
```

☆Result:

We have successfully interfaced the sevensegment display with ArdiunoUno and it showing successfully from 0 to 9.


SRMIST
25
10
31
1001
VDD
V
RS
E
D1
D2
D3
D4
D5
D6
D7
GND
13
12
11
10
9
8
7
6
5
4
3
2
1
0
UNO
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Expriment - 3

☆ **Aim:** Interfacing liquid crystal display with arduino Uno.

☆ **Components Used:** Arduino Uno, LCD(16x2) and wires

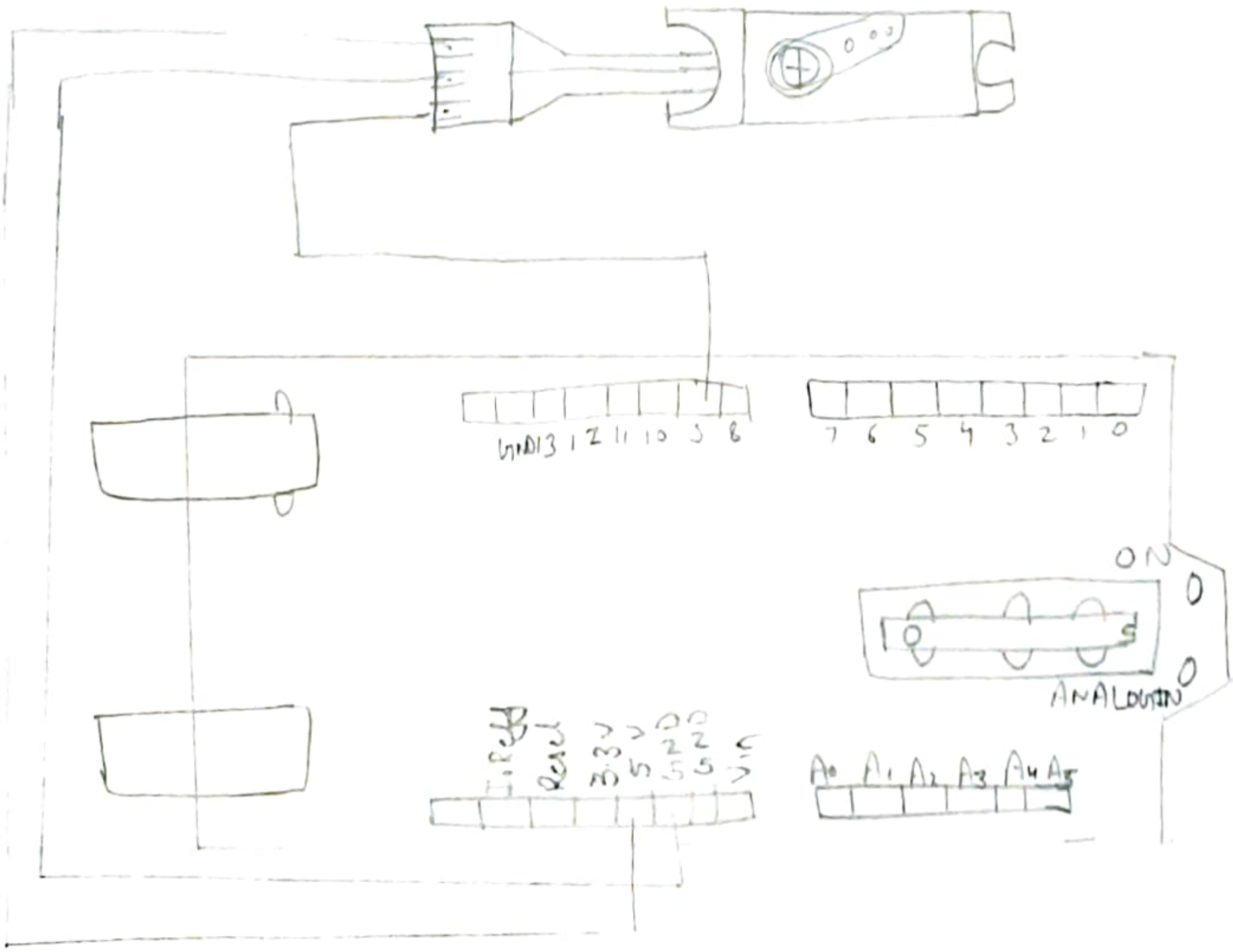
☆ **Software Used:** Wokwi

☆ **Interfacing code:**

```
#include <liquid Crystal.h>
 int rs=12, en=11, d4=5, d5=4, d6=3, d7=2;
 liquid Crystal lcd (rs, en, d4,d5, d6, d7);
 void setup()
 {
    lcd.begin(16,2);
  }
  void loop()
  {
     lcd.home();
     lcd.println("SRMIST");
     delay(1000);
     lcd.println(25,DEC);
     delay(1000);
     lcd.println(25,HEX);
     delay(1000);
     lcd.setcursor(12);
     lcd.println(25,OCT);
     delay(1000);
     lcd.println(25,BIN);
     delay(1000);
  }
```

☆ **Result:**

We have successfully interfaced the liquid crystal display with Arduino Uno and it is showing the desired output.

GND 13 12 11 10 9 8
7 6 5 4 3 2 1 0
ON
ANALOGIN
I.Ref
Reset
3.3V
5V
GND
GND
Vin
A0 A1 A2 A3 A4 A5

Experiment - 4

☆ **Aim:** Interfacing servo motor with Arduino Uno.

☆ **Components used:** Arduino Uno, servo, motor and wires.

☆ **Software used:** Wokwi

☆ **Interfacing code:**

```
#include <Servo.h>
Servo myServo; // create servo object to control servo
// 12 servo objects can be created on most boards
int pos; // To store servo position
void setup()
{  myservo.attach(9); // attaches servo on pin 9 to the servo object
}
void loop()
 { for (pos=0; pos< =180; pos+=1) // goes from 0 to 180 degree
    {      //tell servo to go position in variable pos
     myservo.write(pos);
     delay(15); // wait 15 ms for servo to reach position
    }
   for (pos=180; pos>=0; pos-=1) // goes from 180 to 0 degree
     { my servo.write(pos);
       delay(15);
     }
   }
}
```

☆ **Result:**

We have successfully interfaced servo motor with Arduino Uno and it is showing the desired movement from 0 to 180 degree and 180 to 0 degrees.

Experiment -5

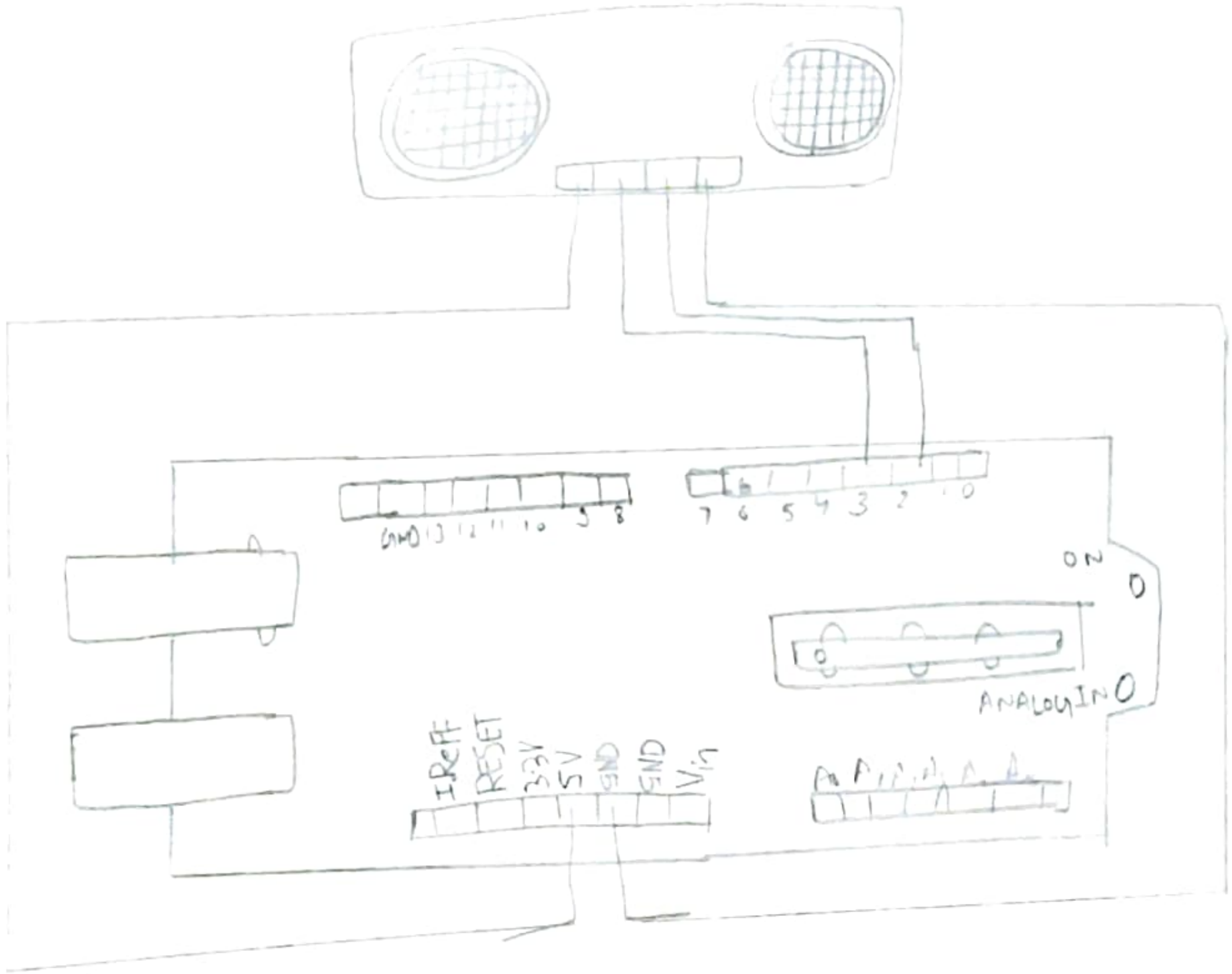
☆ **Aim:** Interfacing ultrasonic sensor with Ardiuno Uno.

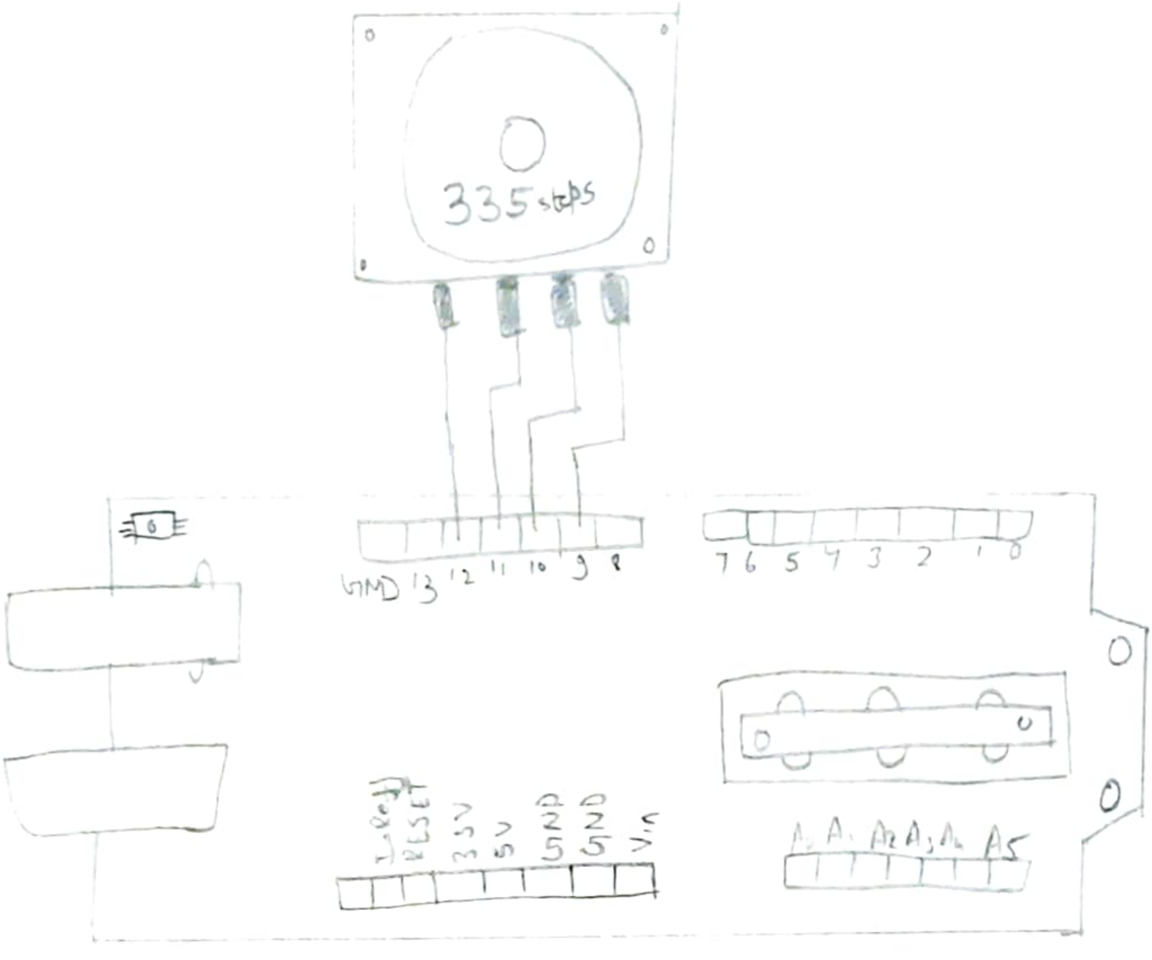
☆ **Components Used** Ardiuno Uno, Ultrasonic, wire.

☆ **Software Used** wokwi

☆ **Interfacing code:**

```
#define echoPin 2
# define trigg Pin3

float duration;
float distance;
void setup()
{
   serial.begin(9600);
   Pin mode (trig Pin, OUTPUT);
   Pin mode (echoPin, INPUT);
}
Void loop()
{
      digital Write (trigg Pin, Low);
      delay Microsecods (2);
      digital Write (trigPin HIGH);
      delay Microseconds (10);
      digital Write (trig Pin, Low);

      duration = pulse In (echo Pin, HIGH);
      distance = (duration * 0.034312);
      serial.print ("Distance : ");
      serial.print (distance);
      serial.println ( "cm");
      delay (1000);

}
```


GND
13
12
11
10
9
8
7
6
5
4
3
2
1
0
ON
ANALOGIN
IReFF
RESET
3.3V
5V
GND
GND
Vin



clockwise
counter clockwise

Experiment-6

★★ **Aim:** Interfacing bipolar stepper motor with Aridiuno Uno.

★ **Components Used:** Arduino Uno, Bipolar, stepper motor and wires.

★ **Software Used:** Wokwi

★ **Interfacing code:**

```
#include <stepper.h>
 const int stepsPerRevolution = 200; //change this to fit the number of steps per
                                        revolution for your motor.

 stepper mystepper (stepsPerRevolution, 8, 9, 10, 11);
 void setup() //initialize stepper library on pin 8 through
 {
      myStepper.setSpeed(60); //set the speed at 60 rpm
      Serial.begin(9600); //initialize serial port.
  }
  void loop()
  {  //step one revolution in one direction
     Serial.println("clockwise");
     mystepper.step(400);
     delay(500);

    //step one revolution in another direction
    Serial.println("counter clockwise");
    mystepper.step(-400);
    delay(500);
 }
```

★ **Result:** We have successfully interfaced bipolar stepper motor with Aridiuno Uno and it is showing the desired output.

Experiment-7

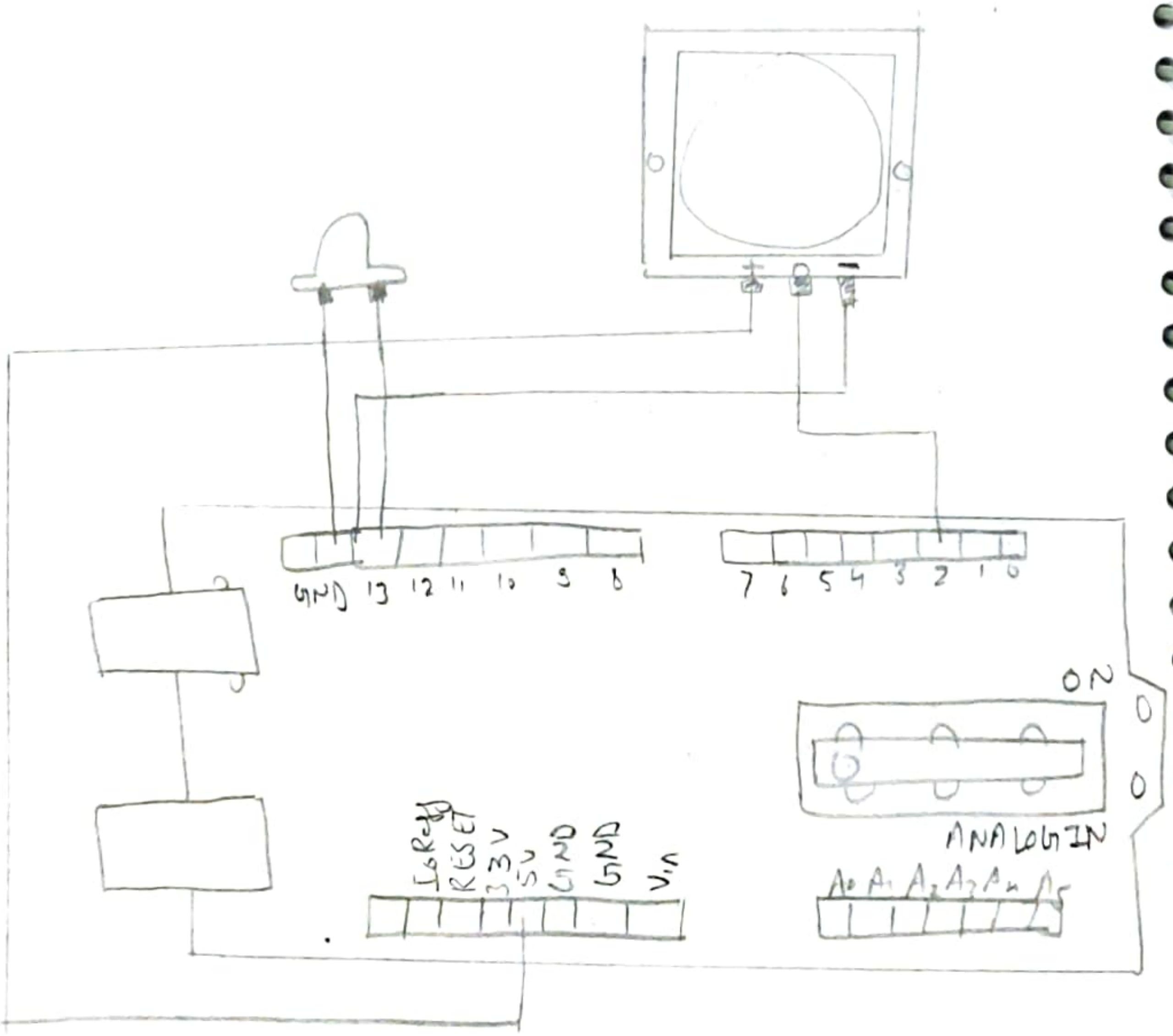
☆ **Aim**: Interfacing PIR motion sensor, with Arduino Uno.

☆ **Components Used**: Arduino Uno, PIR motion sensor and wires.

☆ **Software Used**: Wokwi

☆ **Interfacing code**:

```
 int ledPin = 13; // Pin for LED
int inputPin = 2; //input Pin for PIR
int pinState = low; // we start assuming no motion detected
int val = 0;        // for reading pin status
void setup()
  { pinMode (ledPin, OUTPUT); //declared LED as output
    pinMode (ledPin, INPUT); // declared sensor as input
    Serial. begin (9600);
  }

void loop()
  { val = digitalRead (inputPin); // read input value
    if (val == HIGH) // check if input is high
       { digitalWrite (ledPin, HIGH); // turn LED on
         if (Pinstate == LOW)
            {   // we have just turned on
              serial.println("Motion detected");
              // we only want to print on output change
              pinstate = HIGH
            }
    }
    else
    {
```



Motion detected'

Motion ended

```
      digital Write (ledPin, Low);
      if (pinstate == HIGH)
       {
           serial.println("motion ended!');

           pinstate = Low;

       }
    }
 }
```

☆ Result:

We have successfully interfaced PIR motion sensor with Arduino Uno and it is giving desired output.

Experiment - 8

☆ **Aim:** • PWM or Pulse width modulation using Ariduino Uno.

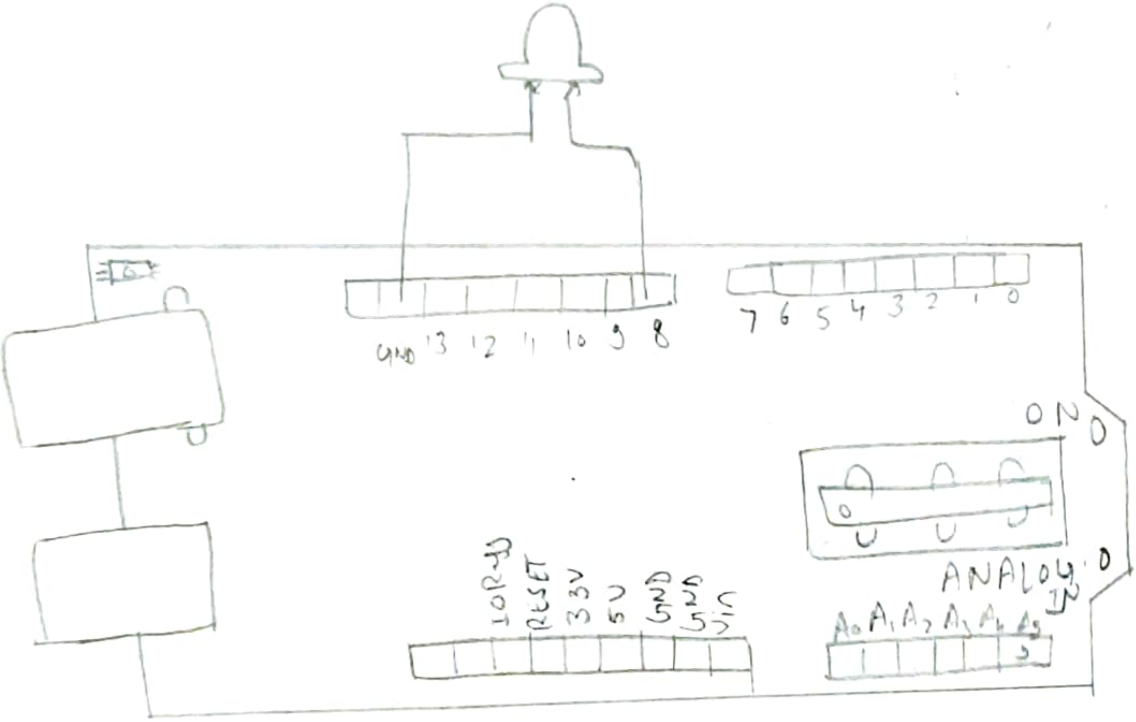
☆ **Components Used:** Arduino Uno , LED and wires

☆ **Software used:** wokwi

☆ **Interfacing code:**

```
void setup()
{
}
 Void loop()
 {
  analogWrite(9,64);
  delay(1000);
  analogWrite(9,127);
  delay(1000);
  analogWrite(9,191);
  delay(1000);
  analogWrite(9,255);
  delay(1000);
  analogWrite(9,0);
  delay(1000);
```

☆ **Result:** we have successfully interfaced LED with Ardiuno Uno to show PWM and it is giving desired result.



GND
13
12
11
10
9
8
7
6
5
4
3
2
1
0
ON
O
IOREF
RESET
3.3V
5V
GND
GND
Vin
ANALOG IN
A0
A1
A2
A3
A4
A5