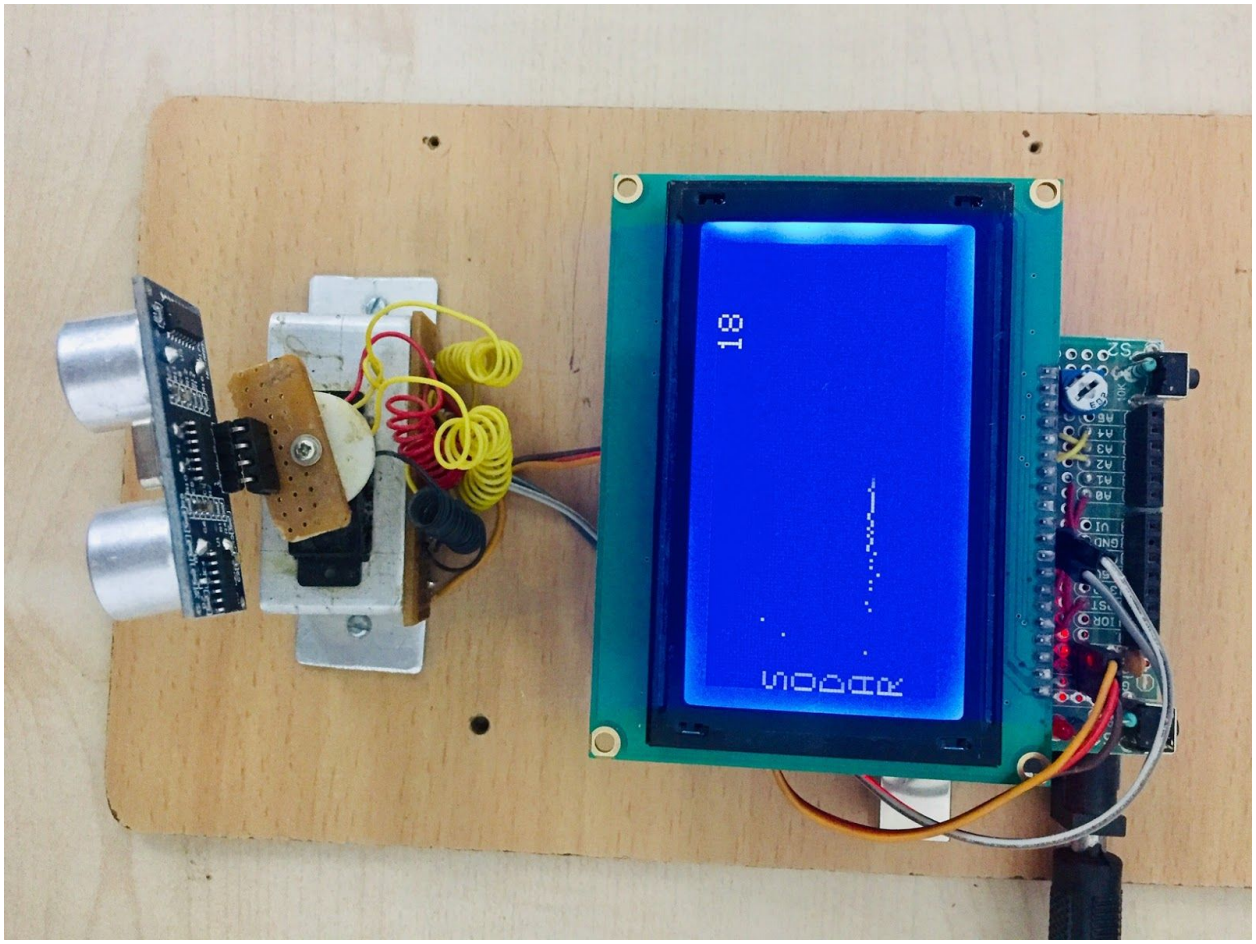
# SODAR



**Arduino Sketch:**

```
#include <openGLCD.h> // include the graphics LCD library

#include <fonts/SystemFont5x7.h> // include the standard character fonts for it

#include <Servo.h>;

int tdist = 0;

float distance = 0;
```

```
int duration = 0;
int angle = 0;
int prev_time;
int new_time;
int lapsed_time;
Servo Scan;//initialize a servo object for the connected servo
const int trigPin = 2; // define pins numbers for HC-SR04
const int echoPin = 12;//
void setup()
{
GLCD.Init(NON_INVERTED); // configure GLCD
GLCD.ClearScreen(); // turn off all GLCD pixels
GLCD.SelectFont(System5x7);
Scan.attach(3); // attach the signal pin of servo to pin3 of arduino
pinMode(trigPin, OUTPUT); // Sets the Pin2 as an Output
pinMode(echoPin, INPUT); // Sets the Pin12 as an Input
}
void loop() {
  drawScreen();
   for(angle = 10; angle < 128; angle++) // move servo from 10 degrees to 128 degrees
{
  Scan.write(127-angle); // to synch servo direction with display direction
delay(15);
getDistance();
```

```
  GLCD.CursorTo(16, 0);
if (tdist == 0)
{
 GLCD.PrintNumber(00);
}
else if ((tdist > 63)|| (tdist < 5))
{
  GLCD.Puts("    ");
}
else
{
GLCD.PrintNumber(tdist);
tdist = (63 - tdist);
GLCD.SetDot(angle, tdist, BLACK);
delay(70);
}}
drawScreen();
for(angle = 128; angle > 10; angle--) // command to move from 128 degrees to 10 degrees
 {
   Scan.write(127 - angle); //command to rotate the servo to the specified angle
delay(15);
getDistance();
GLCD.CursorTo(16, 0);
```

```
if (tdist == 0)
{
  GLCD.PrintNumber(00);
}
 else if ((tdist > 63) || (tdist < 5))
{
  GLCD.Puts("    ");
}
else
{
  GLCD.PrintNumber(tdist);
tdist = (63 - tdist);
GLCD.SetDot(angle, tdist, BLACK);
delay(70);
} }}
void getDistance() // function to get distance from SR-04
{
      digitalWrite(trigPin, LOW);
          delayMicroseconds(2);
          digitalWrite(trigPin, HIGH);                        // send signal
      delayMicroseconds(10);                        // wait 10 microseconds
          digitalWrite(trigPin, LOW); // close signal
          duration = pulseIn(echoPin, HIGH);// calculate time for signal to return
          distance = duration*0.01657;  // Calculate the distance in cm
```

```
        tdist = int(distance);
        }
void drawScreen() // generate GLCD display effects
{
GLCD.ClearScreen();
//GLCD.CursorTo(0,0);
//GLCD.Puts("  CHITKARA");
GLCD.CursorTo(0,2);
GLCD.Puts("S");
GLCD.CursorTo(0,3);
GLCD.Puts("O");
GLCD.CursorTo(0,4);
GLCD.Puts("D");
GLCD.CursorTo(0,5);
GLCD.Puts("A");
GLCD.CursorTo(0,6);
GLCD.Puts("R");
//GLCD.CursorTo(0, 6);
//GLCD.PrintNumber(0);
}
```