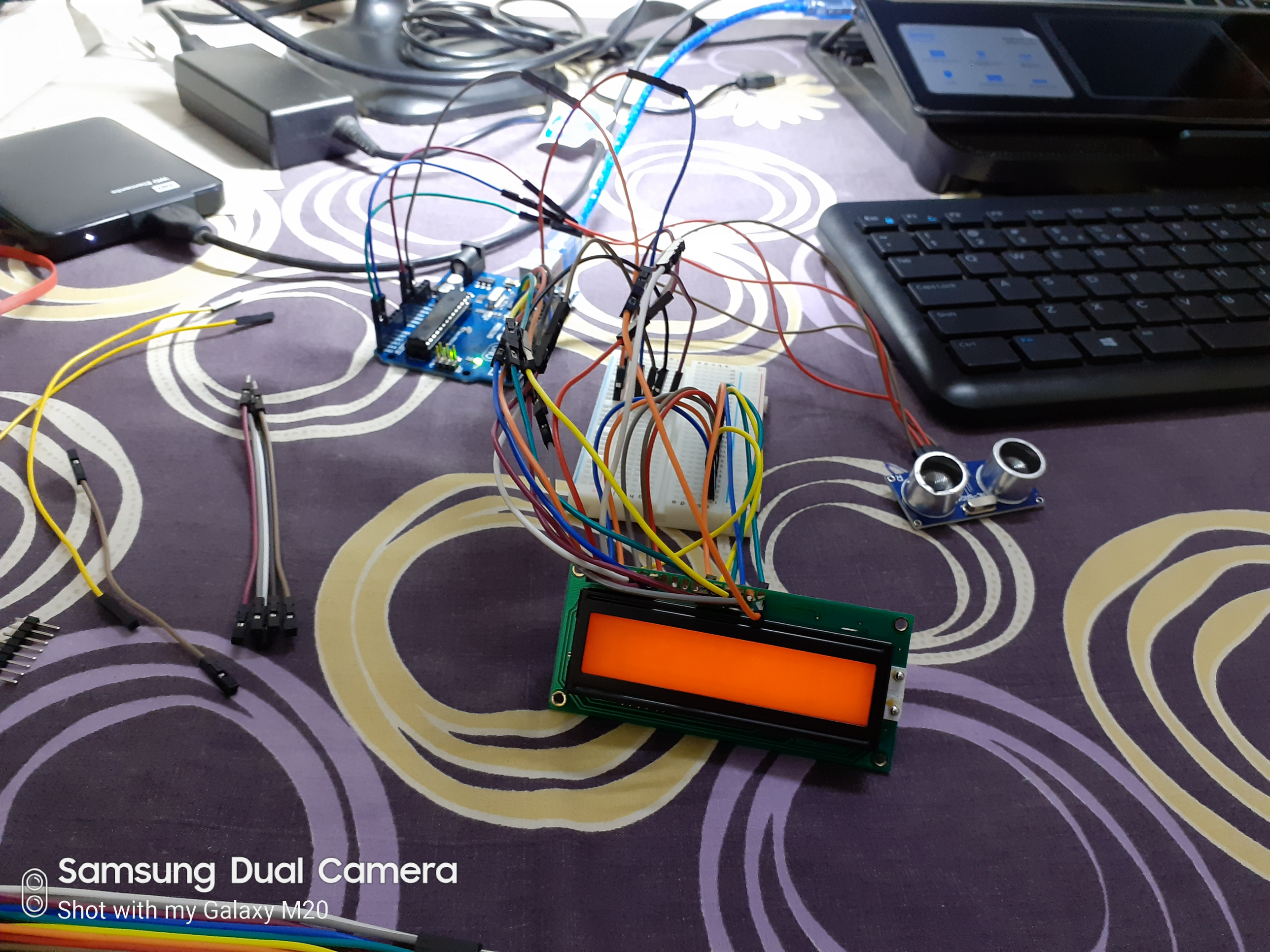
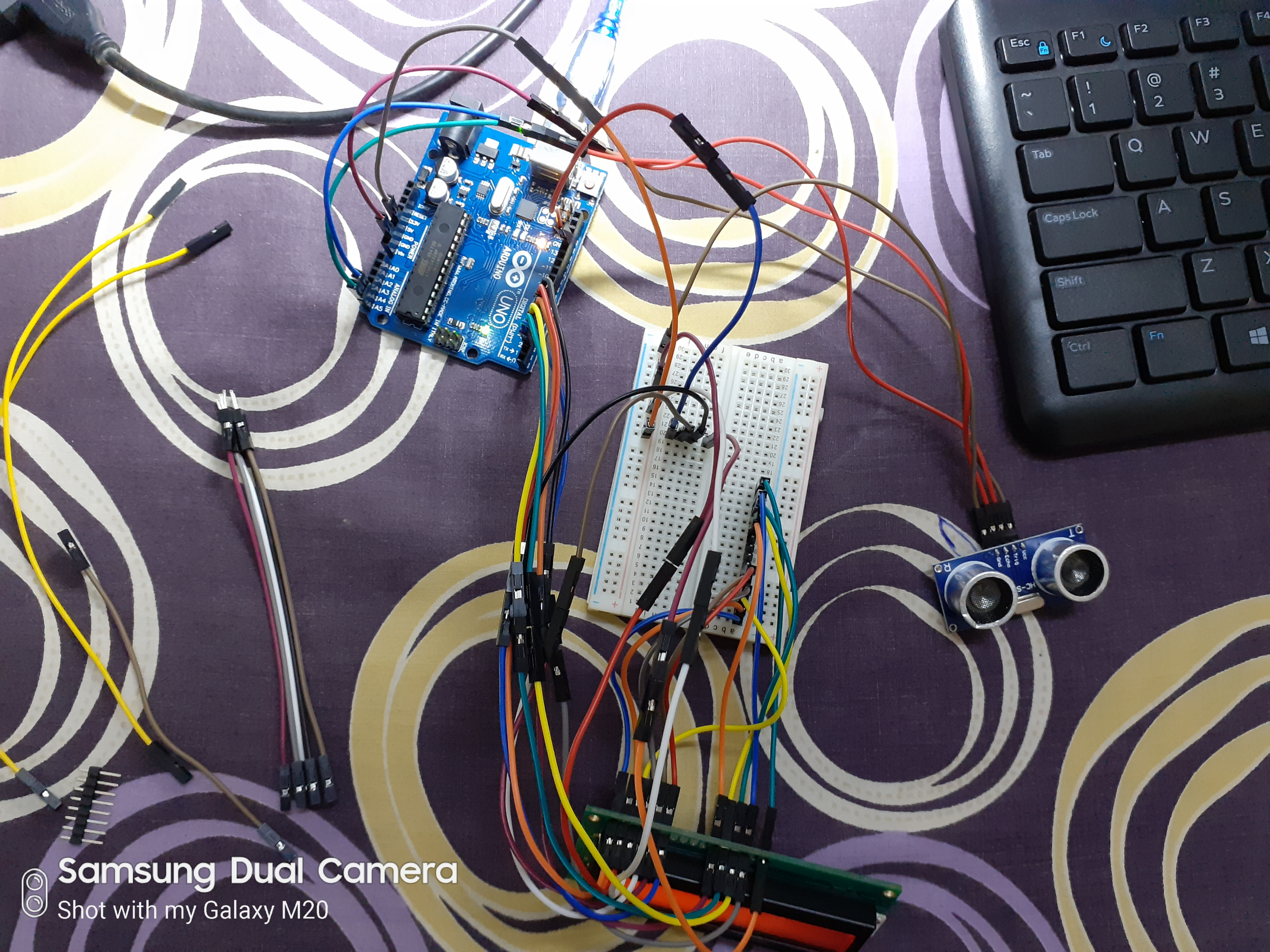

Samsung Dual Camera
Shot with my Galaxy M20


Esc
F1
F2
F3
Tab
Caps Lock
Shift
Ctrl
Fn
UNO
ARDUINO
Samsung Dual Camera
Shot with my Galaxy M20

Sonar_Code_Mine_

```
  Serial.print("V");
  Serial.println(average);                    // preceeding character to separate values
   lcd.clear();                               // average of sensor readings
       lcd.print("Distance:");
       lcd.print(average);
       lcd.print("cm");
       lcd.setCursor(0,1);
       lcd.print("Angle: ");
       lcd.print(leftRightPos);
       lcd.print("deg");
}
/*

start going right to left after we got to 180 degrees
same code as above
*/

for(leftRightPos = 180; leftRightPos > 0; leftRightPos--) {  // going right to left
  leftRightServo.write(leftRightPos);
  for (index = 0; index<=numReadings;index++) {
    digitalWrite(initPin, LOW);
    delayMicroseconds(50);
    digitalWrite(initPin, HIGH);
    delayMicroseconds(50);
    digitalWrite(initPin, LOW);
    pulseTime = pulseIn(echoPin, HIGH);
    distance = pulseTime/58;
    total = total + distance;
```

Compiling sketch...

The sketch name had to be modified. Sketch names can only consist of ASCII characters and numbers and be less than 64 characters long.

Sonar_Code_Mine_

```
  Serial.print("V");
  Serial.println(average);                                  // preceeding character to separate values
   lcd.clear();                                             // average of sensor readings
       lcd.print("Distance:");
       lcd.print(average);
       lcd.print("cm");
       lcd.setCursor(0,1);
       lcd.print("Angle: ");
       lcd.print(leftRightPos);
       lcd.print("deg");
}
 /*
 start going right to left after we got to 180 degrees
 same code as above
 */
 for(leftRightPos = 180; leftRightPos > 0; leftRightPos--) {  // going right to left
    leftRightServo.write(leftRightPos);
    for (index = 0; index<=numReadings;index++) {
      digitalWrite(initPin, LOW);
      delayMicroseconds(50);
      digitalWrite(initPin, HIGH);
      delayMicroseconds(50);
      digitalWrite(initPin, LOW);
      pulseTime = pulseIn(echoPin, HIGH);
      distance = pulseTime/58;
      total = total + distance;
```

Done uploading

```
of ASCII characters and numbers and be less than 64 characters long.
Sketch uses 5470 bytes (16%) of program storage space. Maximum is 32256 bytes.
Global variables use 309 bytes (15%) of dynamic memory, leaving 1739 bytes for local variables. Maximum is 2048 bytes.
```

rasonic_SONAR_display

```
ground (0); // 0 = black
ont = createFont("verdana", 12);
tFont(myFont);
 setup the serial port and buffer
duinoport = new Serial(this, Serial.list()[0], 9600);

 draw the screen */
oid draw(){
ill(0);                                   // set the following shapes to be black
oStroke();                                // set the following shapes to have no outline
llipse(radius, radius, 750, 750);         // draw a circle with a width/ height = 750 with its center position (x and y) set by the radius
rectMode(CENTER);                         // set the following rectangle to be drawn around its center
rect(350,402,800,100);                    // draw rectangle (x, y, width, height)
if (degree <= 1) {                        // if at the far right then set motion = 1/ true we're about to go right to left
motion = 0;                               // this changes the animation to run right to left
}
 (degree >= 179) {                        // if servo at 0 degrees then we're about to go left to right
ion = 1;                                  // this sets the animation to run left to right
}
/* setup the radar sweep */
/*
We use trigonometry to create points around a circle.
So the radius plus the cosine of the servo position converted to radians
Since radians 0 start at 90 degrees we add 180 to make it start from the left
Adding +1 (i) each time through the loops to move 1 degree matching the one degree of servo movement
cos is for the x left to right value and sin calculates the y value
since its a circle we plot our lines and vertices around the start point for everything will always be the center.
*/
strokeWeight(7);                          // set the thickness of the lines
if (motion == 0) {                        // if going left to right
for (int i = 0; i <= 20; i++) {           // draw 20 lines with fading colour each 1 degree further round than the last
stroke(0, (10*i), 0);                     // set the stroke colour (Red, Green, Blue) base it on the the value of i
line(radius, radius, radius + cos(radians(degree+(180+i)))*w, radius + sin(radians(degree+(180+i)))*w); // line(start x, start y, end x, end y)
}
```

Console Errors


File Edit Sketch Debug Tools Help
ultrasonic_SONAR_display
background (0); // 0 = black
myFont = createFont("verdana", 12);
textFont(myFont);
// setup the serial port and buffer
arduinoport = new Serial(this, Serial.list()[0], 9600);
}
/* draw the screen */
void draw(){
fill(0);
noStroke();
ellipse(radius, radius, 750, 750);
rectMode(CENTER);
rect(350,402,800,100);
if (degree <= 1) {
motion = 0;
}
if (degree >= 179) {
motion = 1;
}
// set the following
// draw a circle
// if at the far right
// this changes the
// if servo at 0
// this sets the
ultrasonic_SONAR_display
Screen Key:
Second sweep
Average
Motion
Degrees: 177
Distance: 156
Radar screen code
DELL
Samsung Dual Camera
Shot with my Galaxy M20