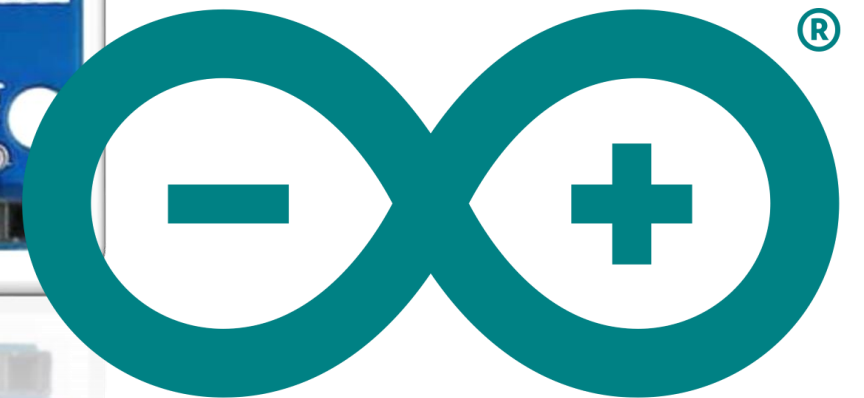
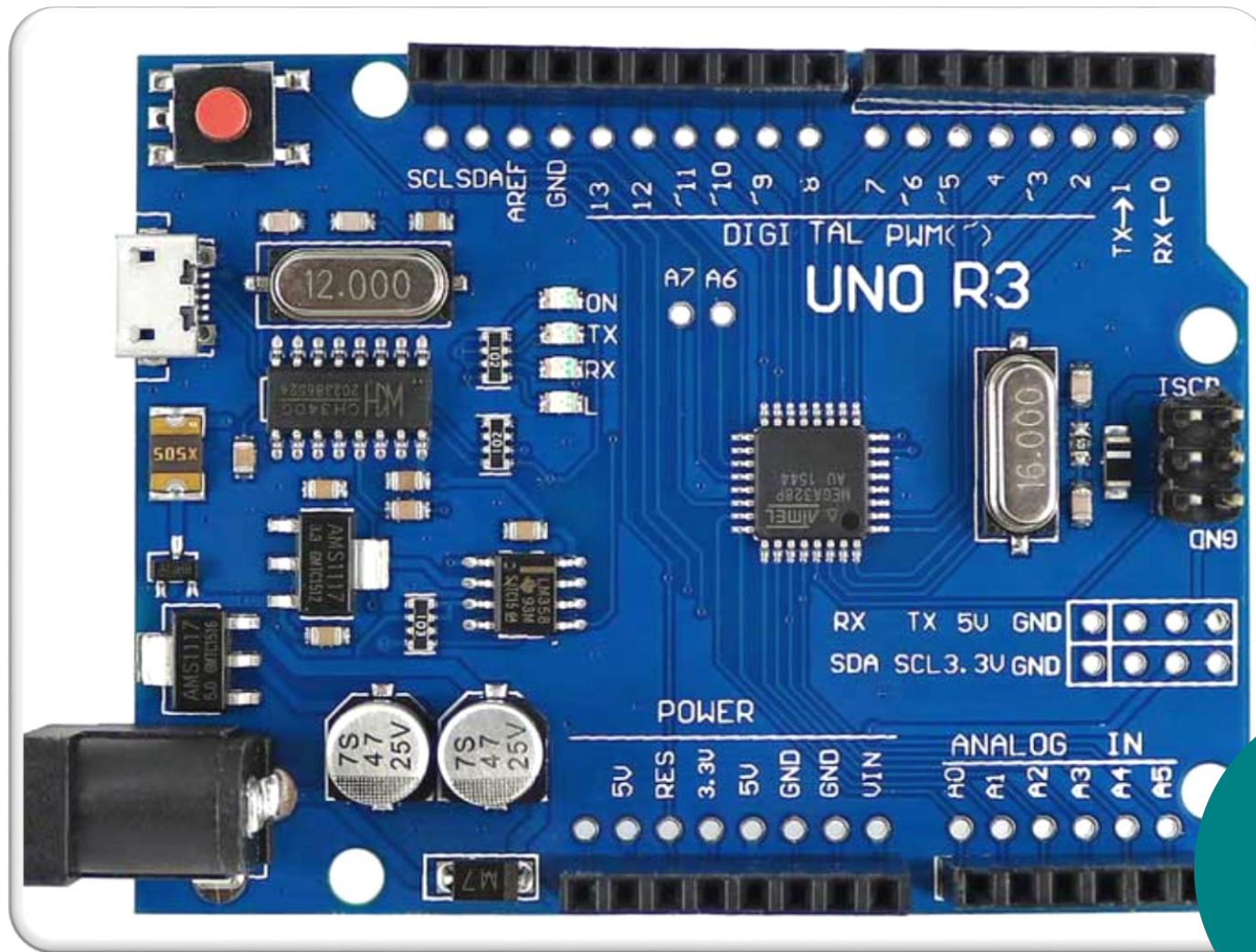
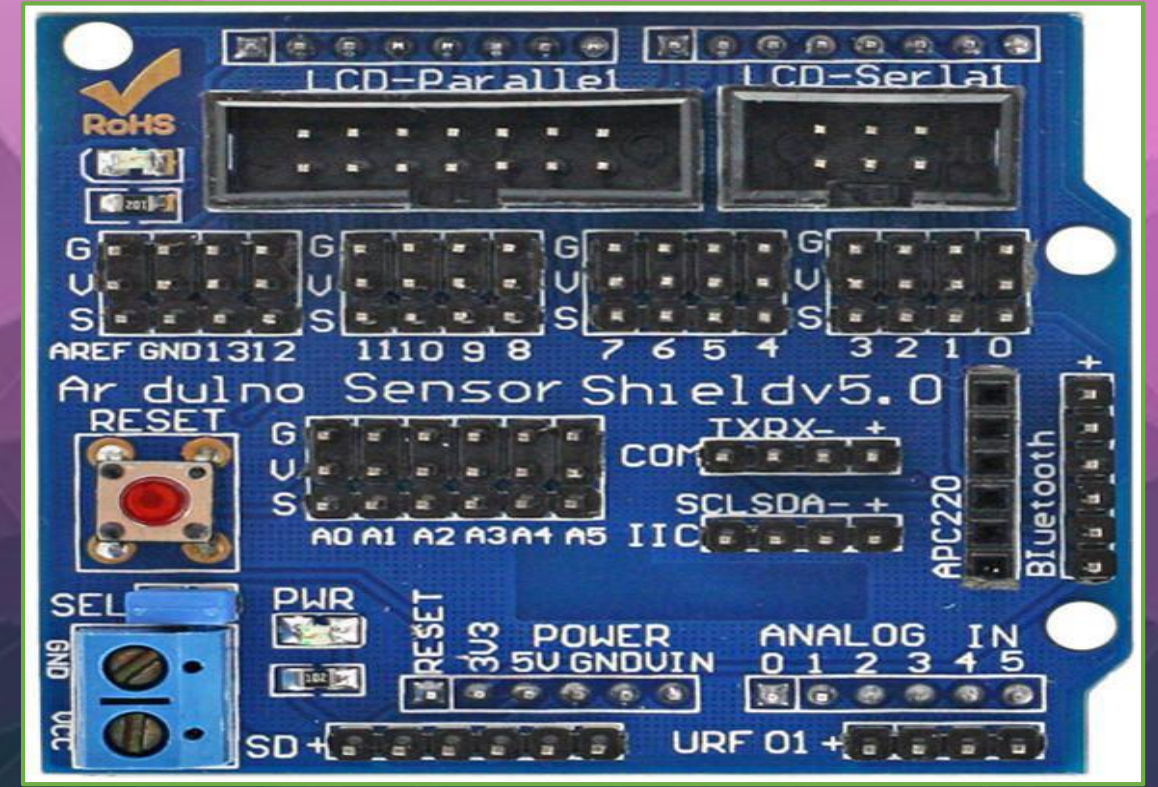
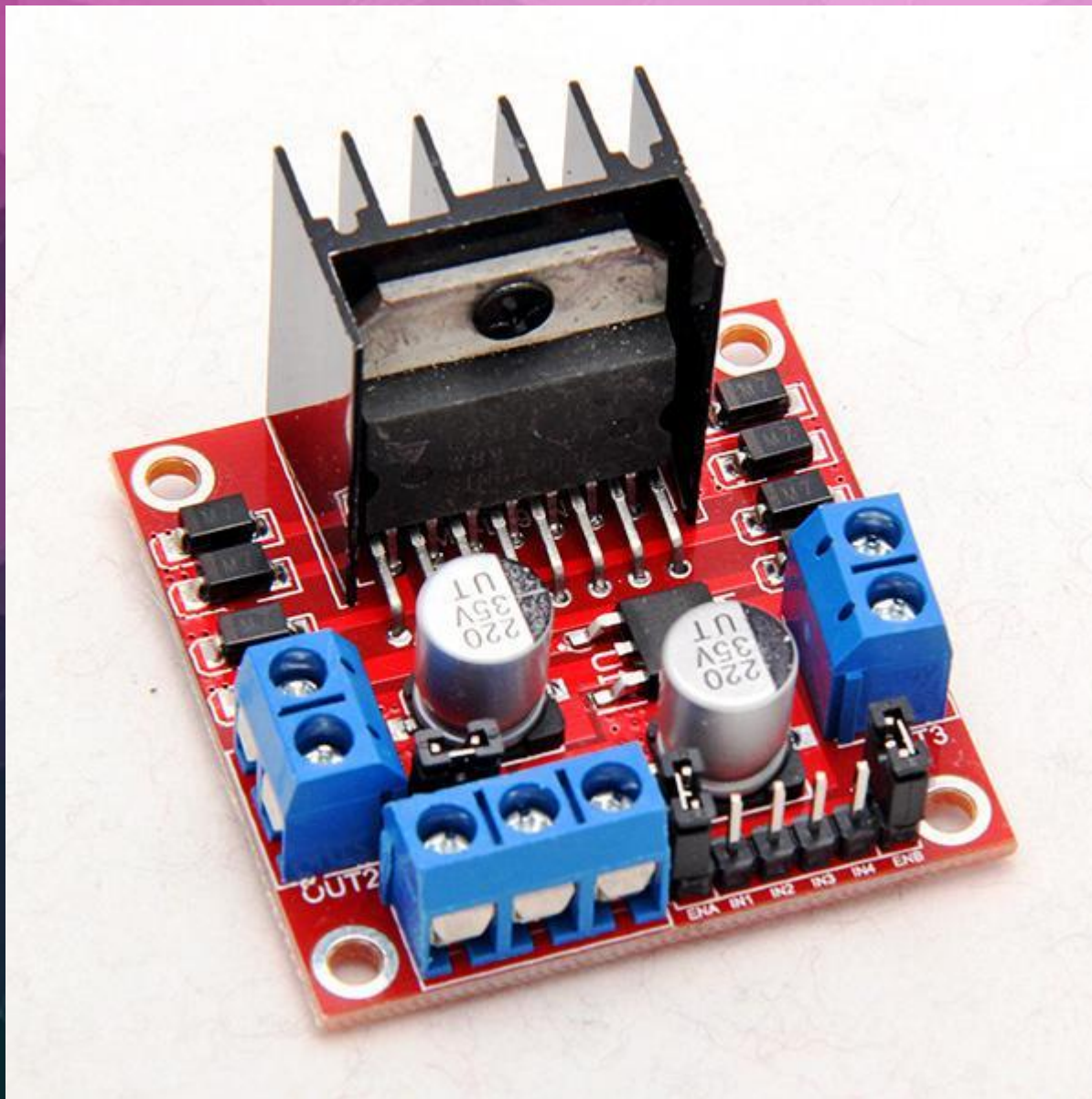




ARDUINO





ENA ความเร็ว

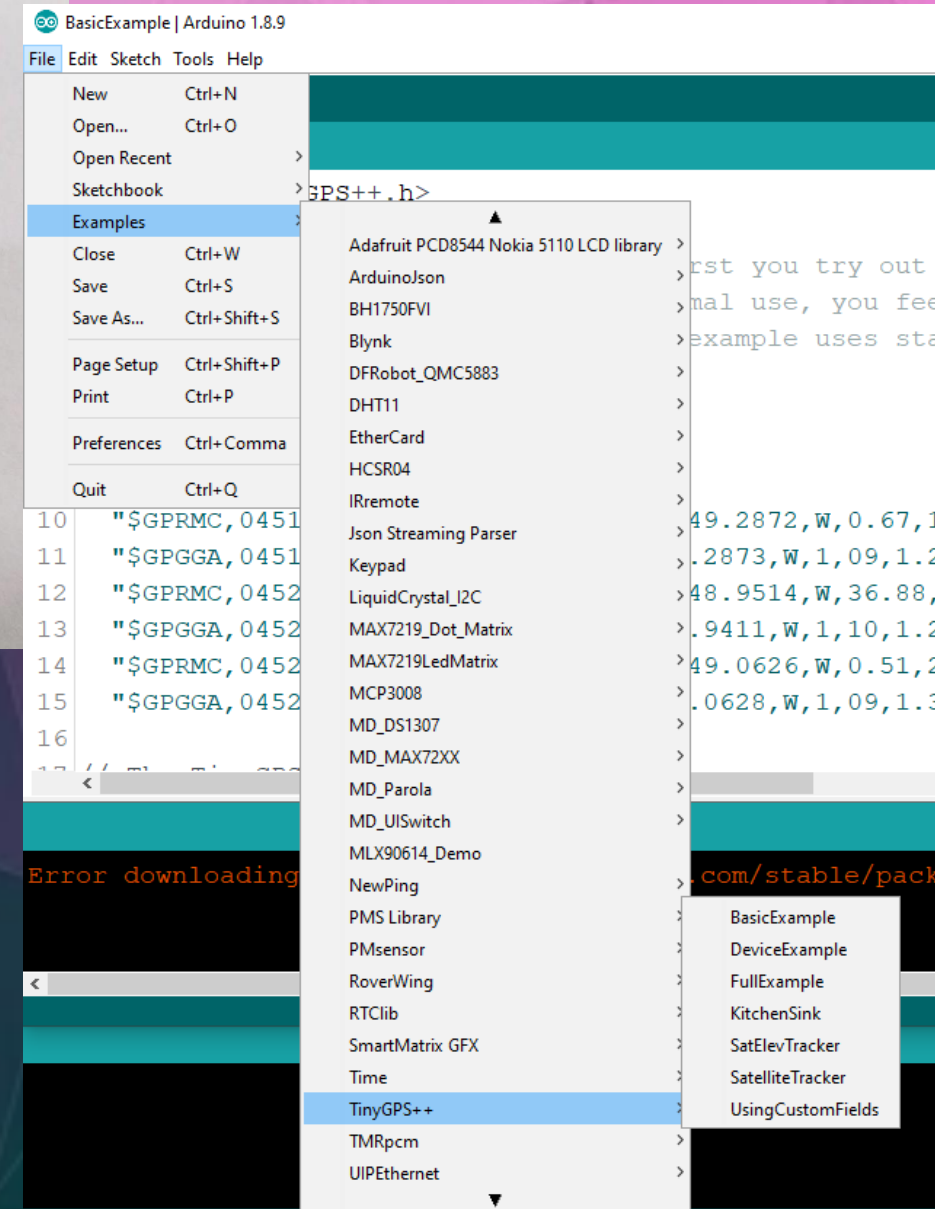
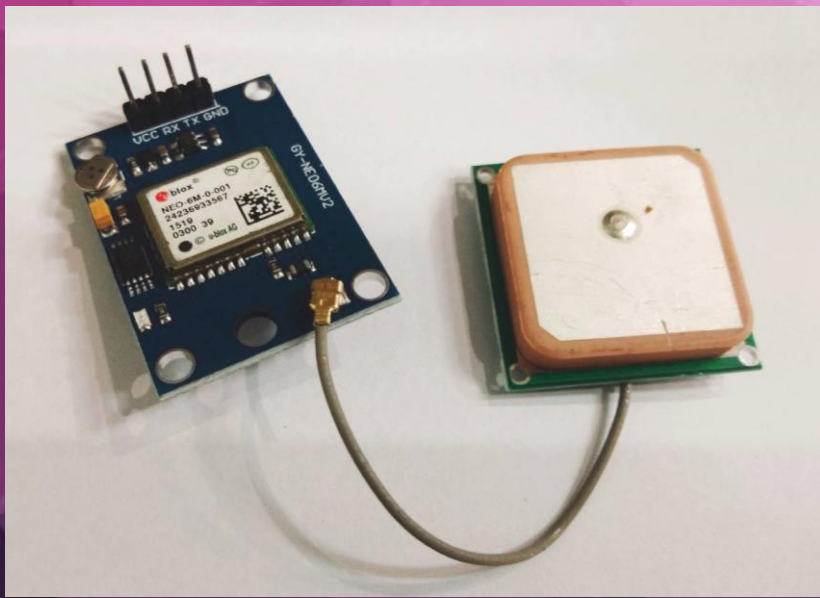


```
1 int tringPin = 5;
2 int echoPin = 3;
3
4 long duration;
5 int distance;
6
7 void setup() {
8     Serial.begin(9600);
9     pinMode(tringPin, OUTPUT);
10    pinMode(echoPin, INPUT);
11 }
12 void loop() {
13     digitalWrite(tringPin, LOW);
14     delayMicroseconds(2);
15     digitalWrite(tringPin, HIGH);
16     delayMicroseconds(10);
17     digitalWrite(tringPin, LOW);
18     duration = pulseIn(echoPin, HIGH);
19     distance = duration * 0.034/2;
20     Serial.print("Distance: ");
21     Serial.println(distance);
22     delay(1000);
23 }
```




```
1 #include <Servo.h>
2 #define pinServo 4
3
4 Servo myServo;
5
6 void setup() {
7     myServo.attach(pinServo);
8 }
9
10
11 void loop() {
12     myServo.write(180);
13     delay(5000);
14     myServo.write(95);
15     delay(3000);
16 }
17 }
```

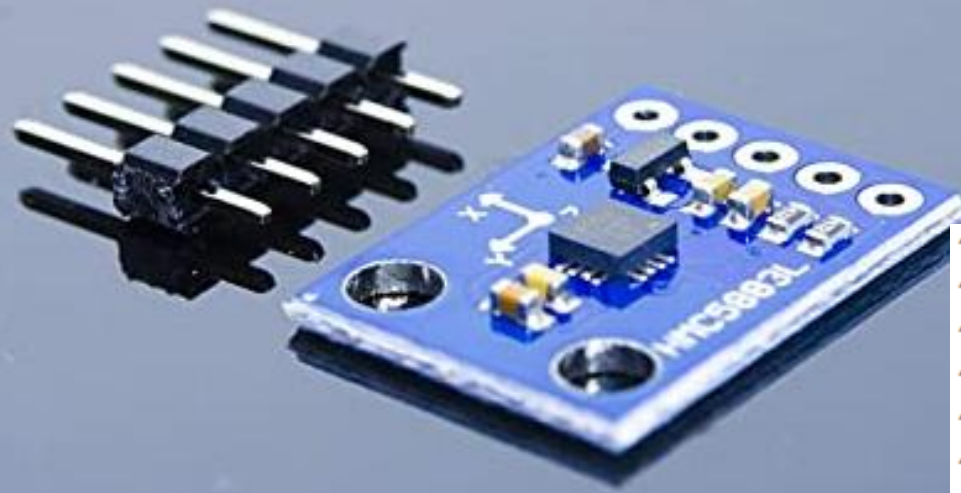




```

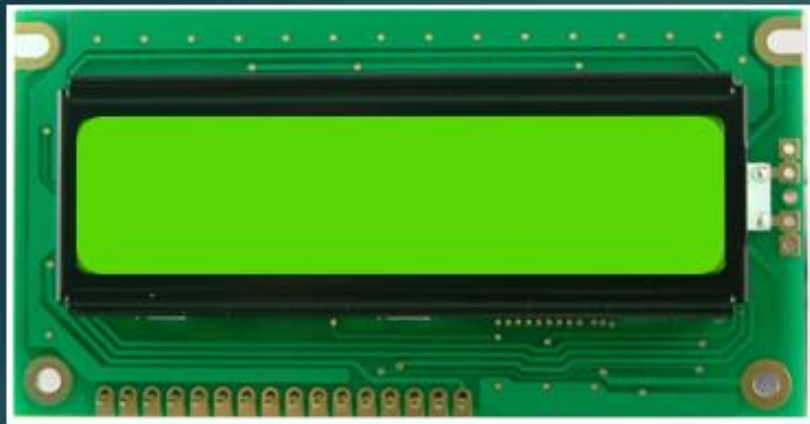
8 #include <TinyGPS++.h>
9 // A sample NMEA stream.
10 const char *gpsStream =
11 "$GPRMC,045103.000,A,3014.1984,N,09749.2872,W,0.67,161.46,030913,,A*7C\r\n"
12 "$GPGLL,045104.000,3014.1985,N,09749.2873,W,1.09,1.2,211.6,M,-22.5,M,,0000*62\r\n"
13 "$GPRMC,045200.000,A,3014.3820,N,09748.9514,W,36.88,65.02,030913,,A*77\r\n"
14 "$GPGLL,045201.000,3014.3864,N,09748.9411,W,1.10,1.2,200.8,M,-22.5,M,,0000*6C\r\n"
15 "$GPRMC,045251.000,A,3014.4275,N,09749.0626,W,0.51,217.94,030913,,A*7D\r\n"
16 "$GPGLL,045252.000,3014.4273,N,09749.0628,W,1.09,1.3,206.9,M,-22.5,M,,0000*6F\r\n";
17
18 // The TinyGPS++ object
19 TinyGPSPlus gps;

```

```
14 #include <Wire.h>
15 #include <DFRobot_QMC5883.h>
16
17 DFRobot_QMC5883 compass;
```

```
43 void loop()
44 {
45     Vector norm = compass.readNormalize();
46
47     // Calculate heading
48     float heading = atan2(norm.YAxis, norm.XAxis);
49
50     // Set declination angle on your location and fix heading
51     // You can find your declination on: http://magnetic-declination.com/
52     // (+) Positive or (-) for negative
53     // For Bytom / Poland declination angle is 4'26E (positive)
54     // Formula: (deg + (min / 60.0)) / (180 / M_PI);
55     float declinationAngle = (4.0 + (26.0 / 60.0)) / (180 / PI);
56     heading += declinationAngle;
57
58     // Correct for heading < 0deg and heading > 360deg
59     if (heading < 0){
60         heading += 2 * PI;
61     }
62
63     if (heading > 2 * PI){
64         heading -= 2 * PI;
65     }
```

GND
5V
SDA
SCL

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

ตัวอักษร 16 ตัว x 2 เส้น

LiquidCrystal_I2C

```

2 #include <Wire.h>
3 #include<LiquidCrystal_I2C.h>
4 #include<SoftwareSerial.h>
5
6 // -- function LCD 20_4  --
7   LiquidCrystal_I2C lcd(0x27,16,2);  // pin D2=SDA, D1=SCL
8
9 PMS pms (Serial);
10 PMS::DATA data;
  
```

```

else if (data.PM_AE_UG_2_5 >= 38 )
{
    Serial.println ("  ฝุ่นกลาง");
    lcd.setCursor(15 ,1);lcd.print("3");
    delay(500);
    digitalWrite(led_s5,LOW);
    digitalWrite(led_s4,LOW);
    digitalWrite(led_s3,HIGH);  //YELLOW LED ON
    digitalWrite(led_s2,LOW);
    digitalWrite(led_s1,LOW);
}
  
```

