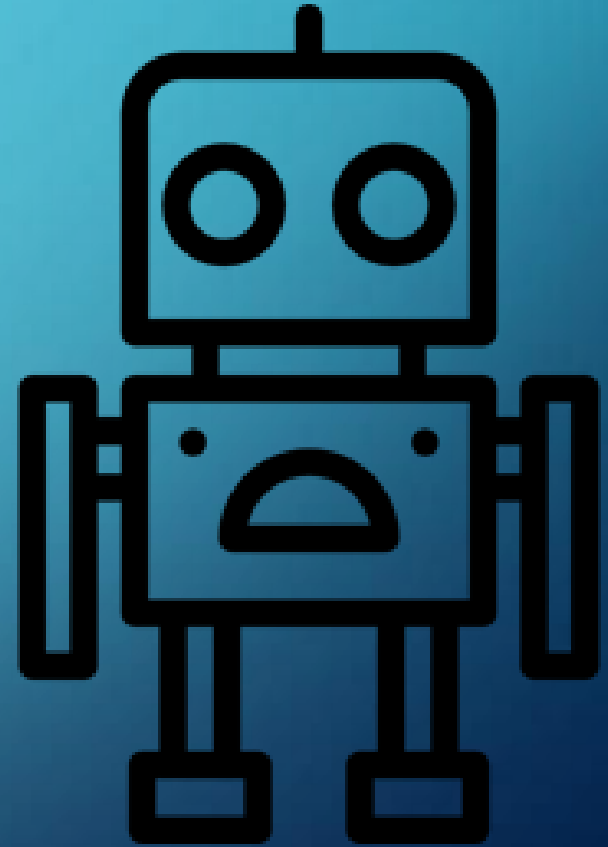




HELLO, WORLD!

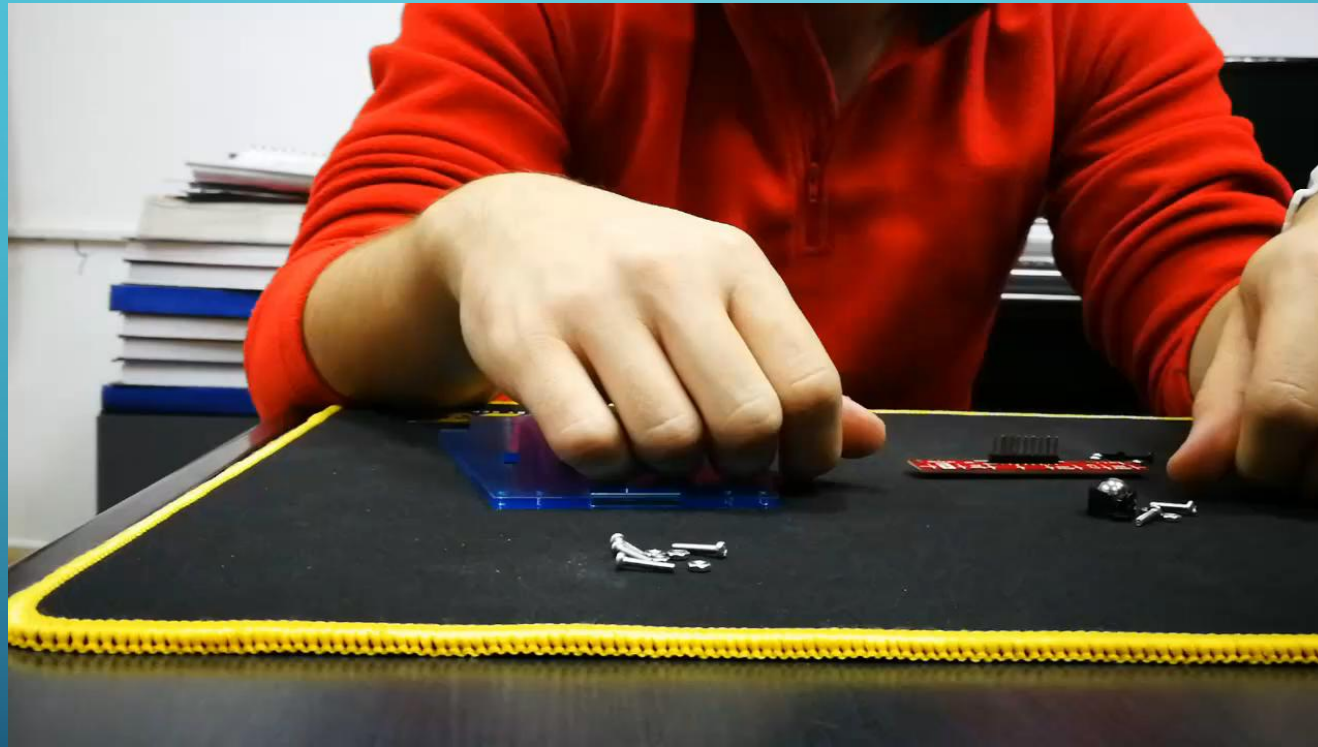
1. HOW TO CONSTRUIRE ROBOT?!
2. HOW TO PROGRAMARE ROBOT?!
3. EXAMPLE



COMPONENTE (ROBOT) ;

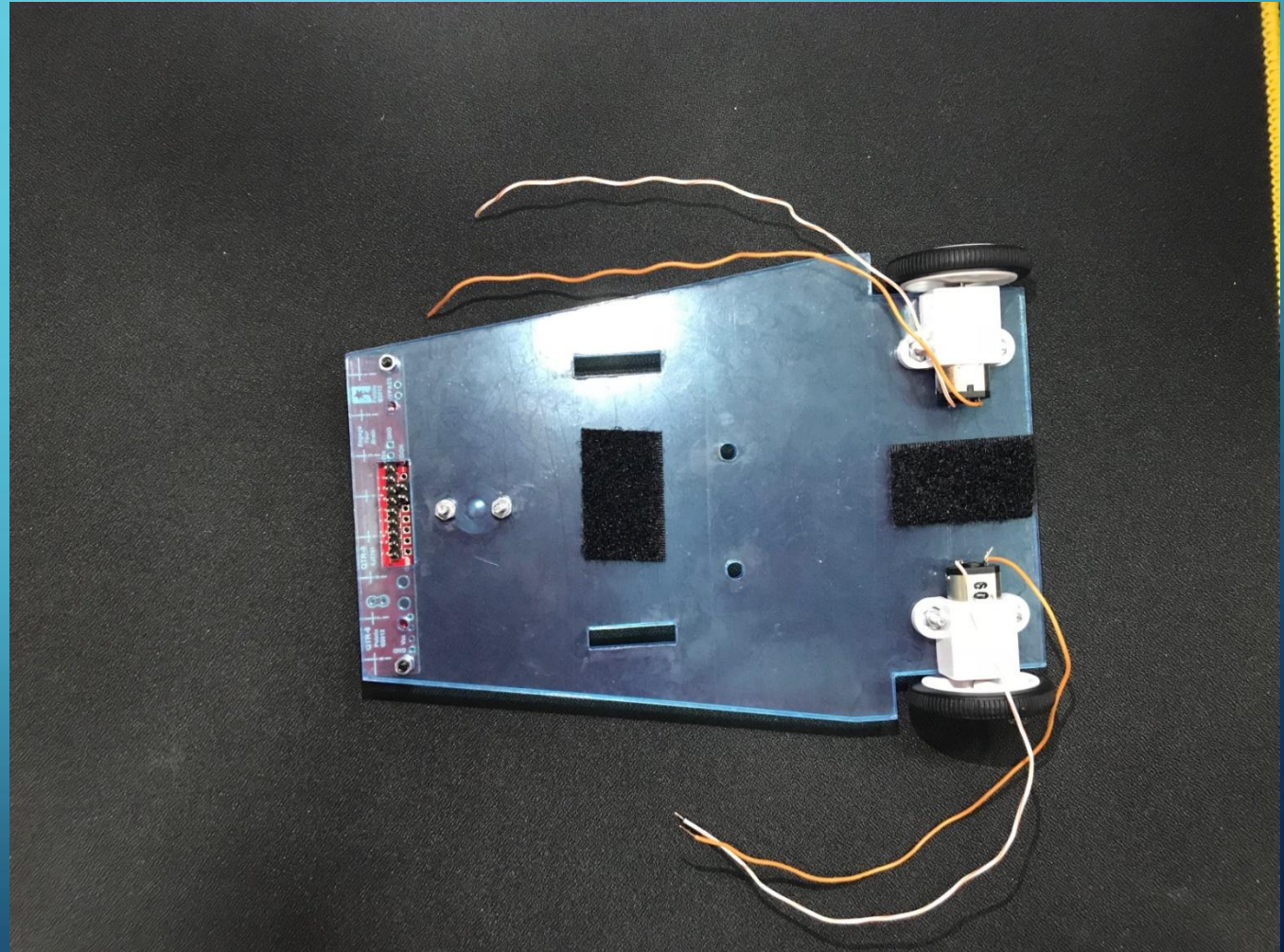
- 1 × sasiu
- 2 × motor cu reductie (1000 rotatii pe secunda)
- 2 × suport motor
- 2 × roata
- 1 × sensor infrarosu
- 1 × Arduino Uno
- 1 × driver motoare (L298)
- 1 × suport baterii
- 1 × bila control robot
- 1 × fire: mama – mama, tata – tata, mama – tata

CONSTRUIESTE (ROBOT);

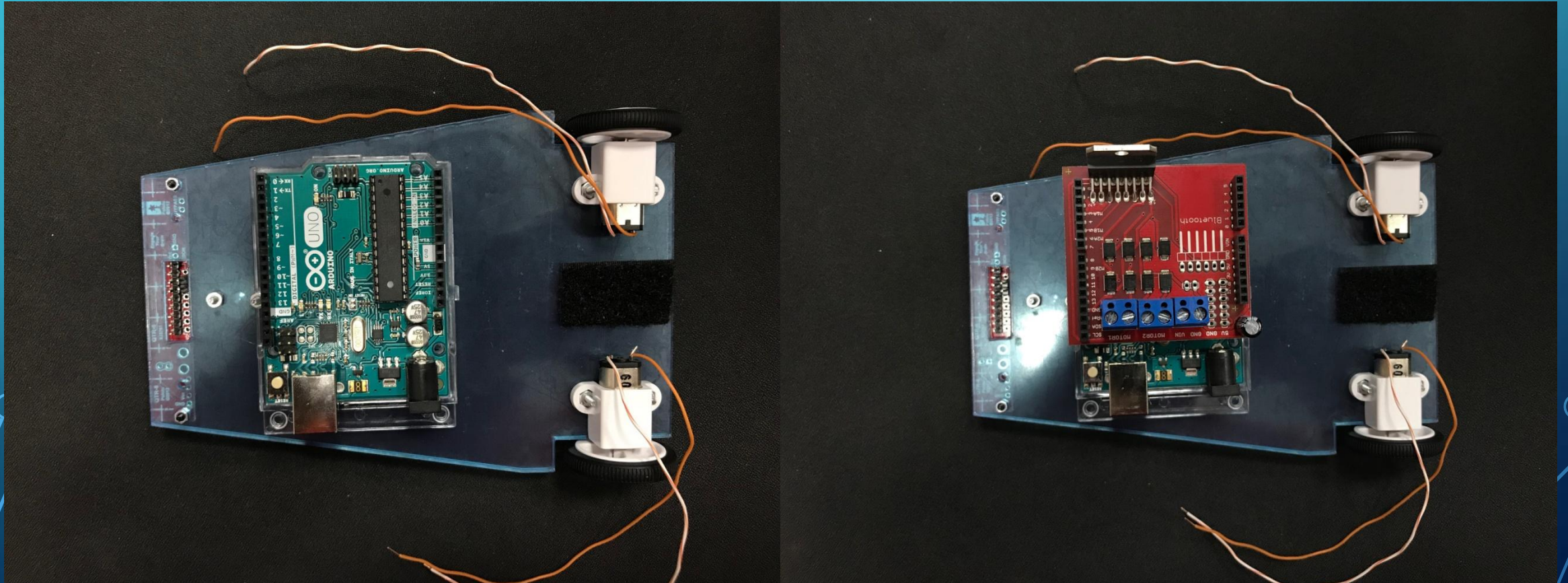


URMATOR(PAS1);

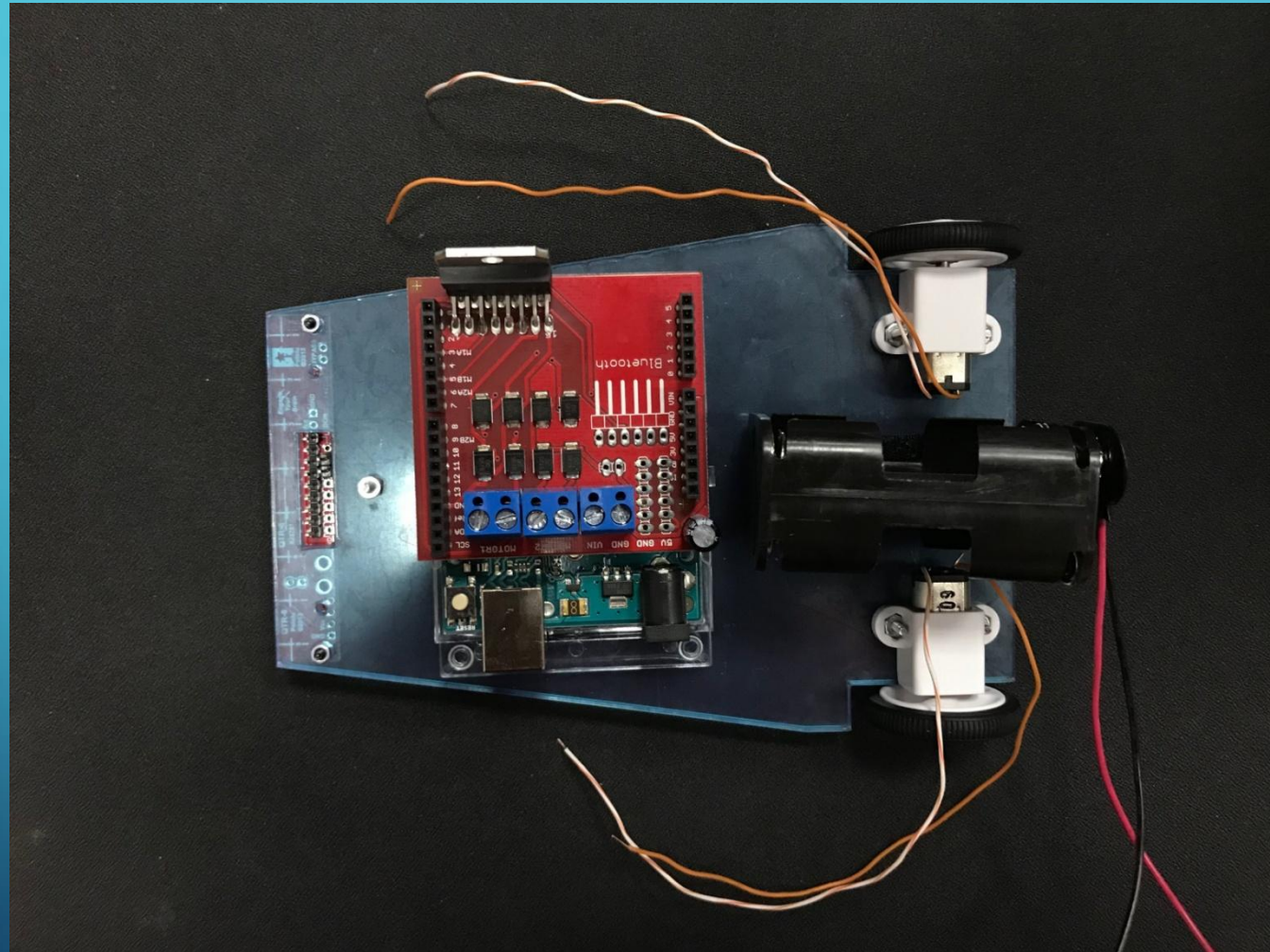
- Lipituri:
 - Pinii de + si - ai motoarelor
 - Pinii senzorului (de lipit inainte de a pune senzorul pe sasiu)



URMATOR(PAS2);

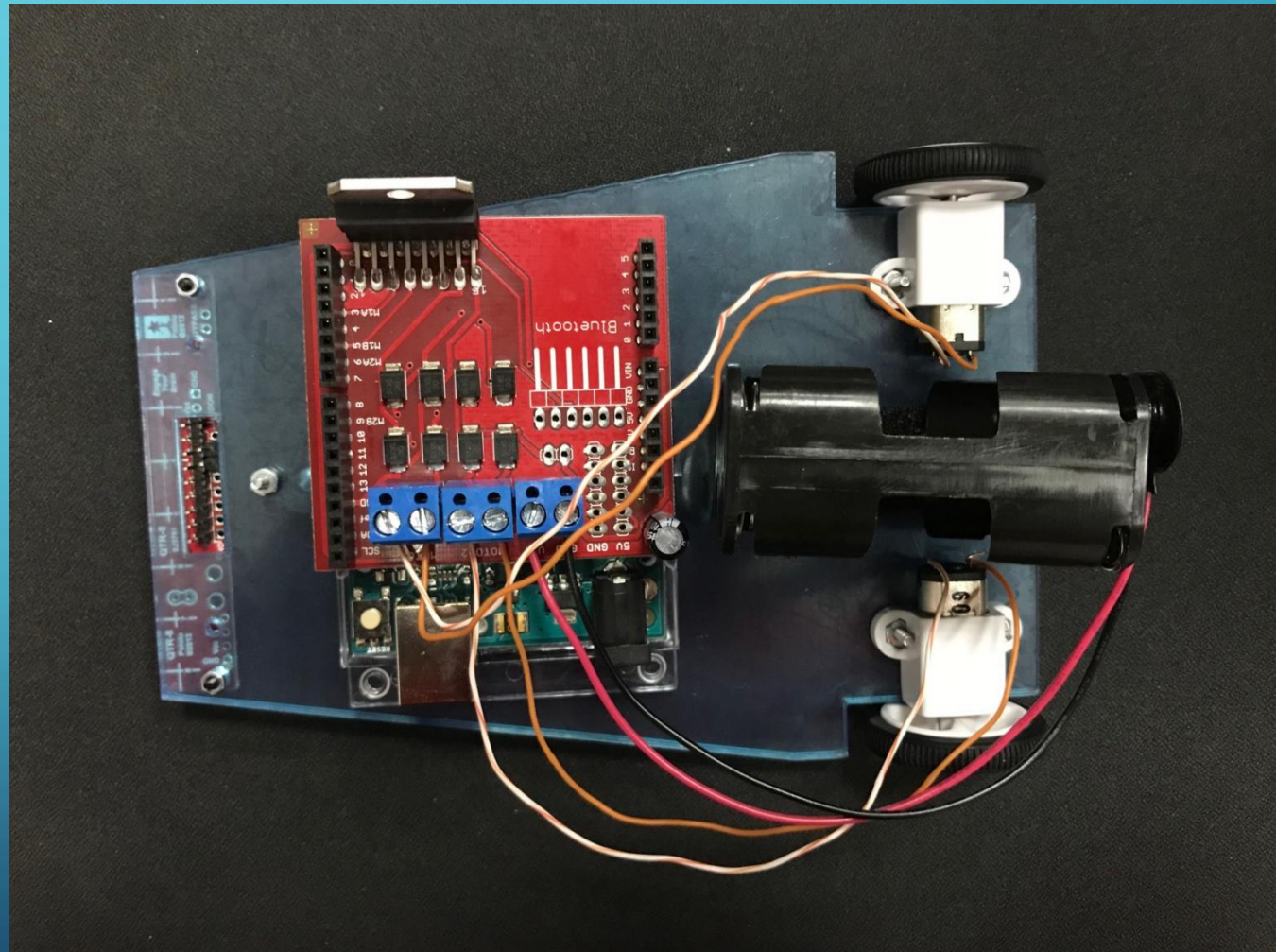


URMATOR(PAS3);



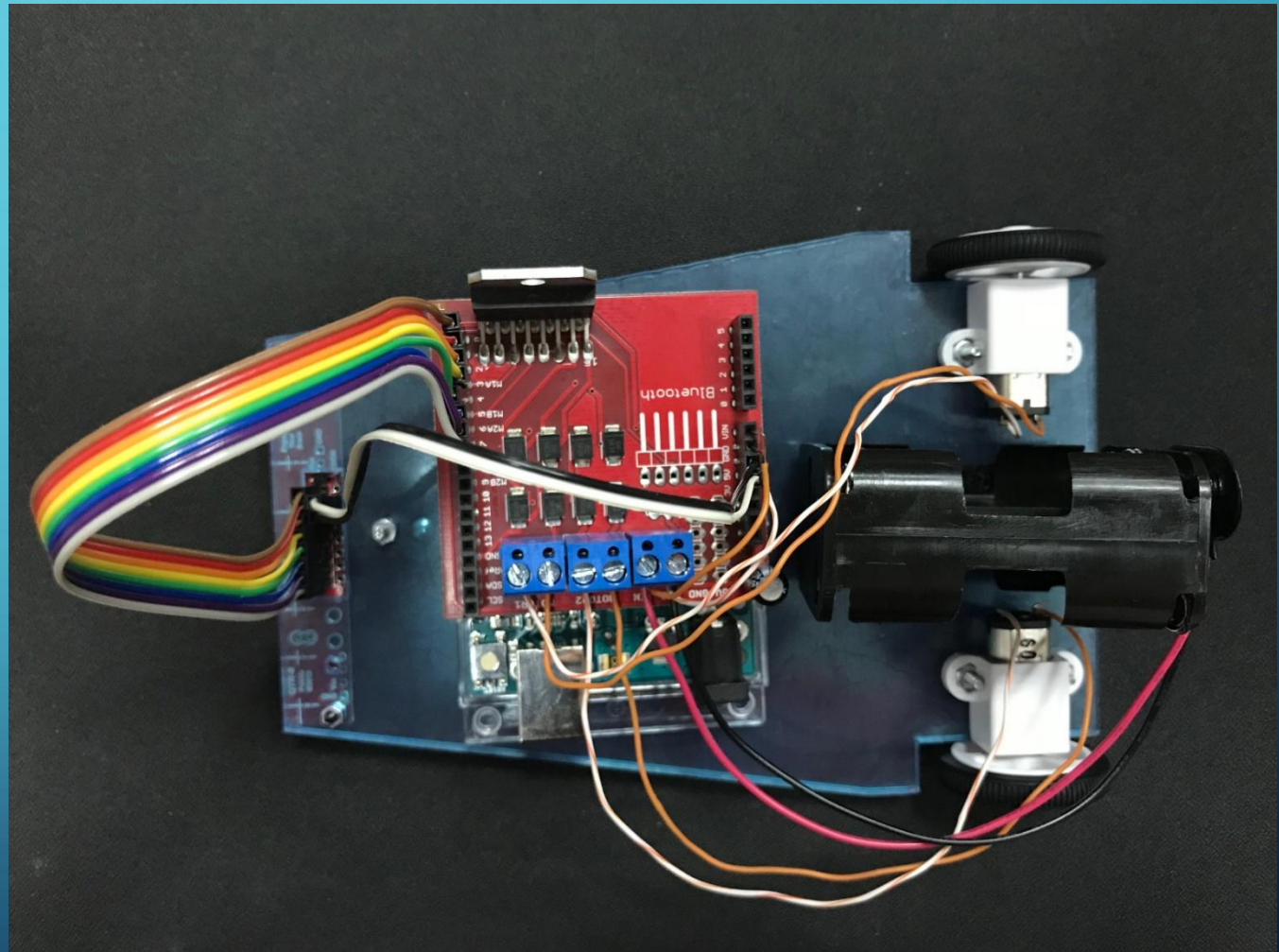
URMATOR(PAS4);

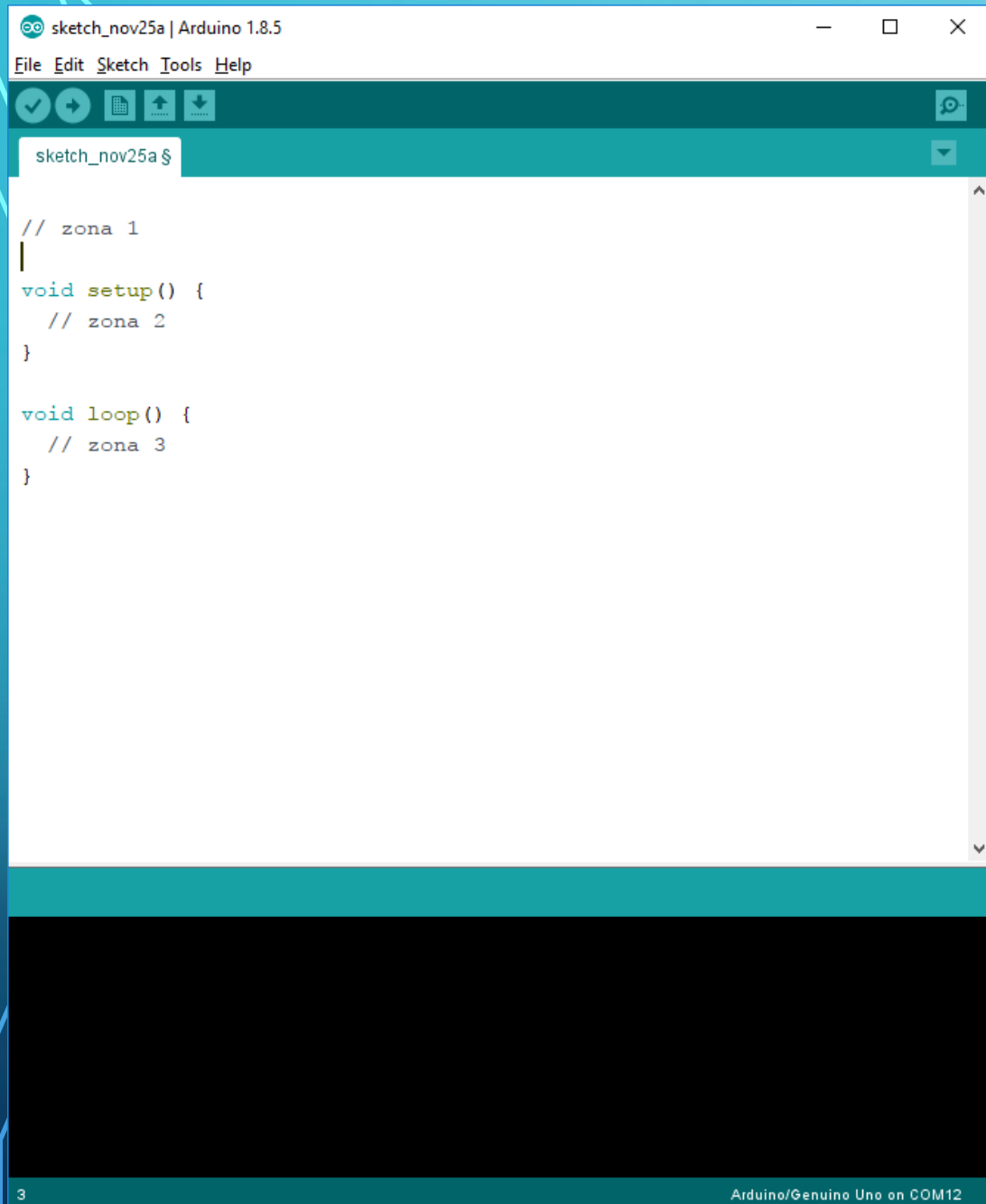
- Se conecteaza:
 - Pinii de + si - de la fiecare motor in soclul asociat (MOTOR1 si MOTOR2)
 - Firul negru (GND) al bateriilor se conecteaza la intrarea GND a L298
 - Firul rosu (VCC) al bateriilor se conecteaza la intrarea VIN a L298



URMATOR(PAS5);

- Se conecteaza:
 - De la soclul de alimentare a L298 (GND si VIN) la pinii GND si VIN a L298 (pentru a alimenta microcontroller-ul Arduino)
 - VCC si GND de la senzor la 5V si GND a L298 (firul alb si firul negru din imagine)
 - Pinii senzorului la pinii digitali/analogici a L298 (in imagine sunt conectati la pini digitali)
 - DE RETINUT! Exceptie fac pinii digitali 3, 5, 6 si 9 deoarece sunt folositi implicit la controlul motoarelor





```
sketch_nov25a | Arduino 1.8.5
File Edit Sketch Tools Help

sketch_nov25a $

// zona 1
|
void setup() {
  // zona 2
}

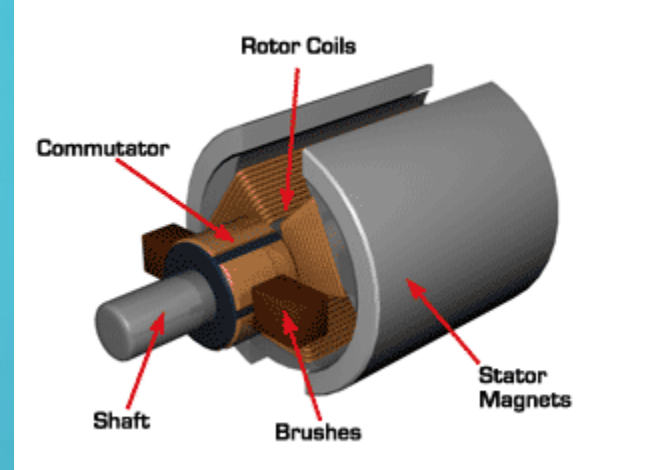
void loop() {
  // zona 3
}

3 Arduino/Genuino Uno on COM12
```

ARDUINO_IDE;

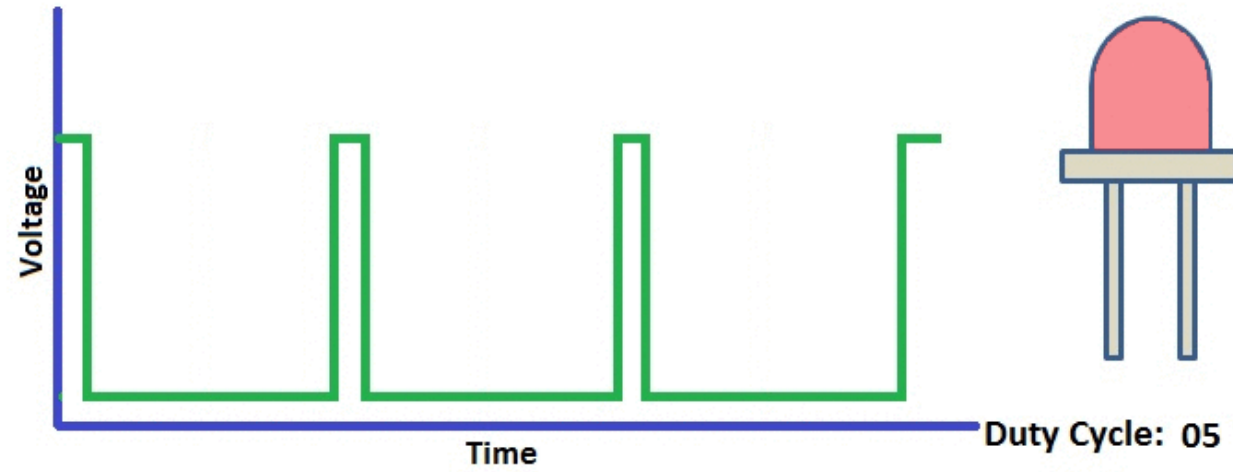
- Zona 1:
 - Declararea variabilelor globale si constantelor care corespund pinilor:
 - `int analogInput = A0; //corespunzator pinului A0`
 - `int digitalOutput = 0; //corespunzator pinului D0`
 - `#define digitalOutput 0; //corespunzator pinului D0`
 - Declararea si initailizarea variabilelor globale si constantelor care ne pot ajuta la dezvoltarea programului
- Zona 2:
 - Structura de program care se executa o singura data
 - Initializarea tipului variabilelor:
 - INPUT: `pinMode(analogInput, INPUT);`
 - OUTPUT: `pinMode(digitalOutput, OUTPUT);`
 - `Serial.begin(9600); //pentru transmitere seriala (folosit pentru debugging)`
- Zona 3:
 - Structura de program care se repeta la infinit
 - Programul care va rula pe microcontroller-ul Arduino

PUTERE (MOTOR);



- Pinii impliciti pentru controlul motoarelor: 3, 5, 6, 9
- Puterea motorului (viteza) este cuprinsa intre -255 si 255 (PWM)
- Rularea inainte a fiecarui motor:
 - Primul pin al motorului primeste valoarea vitezei dorite
 - Al doilea pin al motorului primeste valoarea 0
- Rularea inapoi a fiecarui motor:
 - Primul pin al motorului primeste valoarea 0
 - Al doilea pin al motorului primeste valoarea vitezei dorite
- Se foloseste metoda `analogWrite(pin, viteza)` pentru comanda motoarelor:
 - `analogWrite(motor1_pin1, 255);`
 - `analogWrite(motor1_pin2, 0);`

PWM;



```
int MOTOR1_PIN1 = 3;
int MOTOR1_PIN2 = 5;
int MOTOR2_PIN1 = 6;
int MOTOR2_PIN2 = 9;
```

```
void setup() {
    Serial.begin(9600);
    pinMode(MOTOR1_PIN1, OUTPUT);
    pinMode(MOTOR1_PIN2, OUTPUT);
    pinMode(MOTOR2_PIN1, OUTPUT);
    pinMode(MOTOR2_PIN2, OUTPUT);
}
```

```
void loop() {
    mergi(50,50);
}
```

```
void mergi(int vitezaStanga, int vitezaDreapta) {
    if (vitezaStanga > 0) {
        analogWrite(MOTOR1_PIN1, vitezaStanga);
        analogWrite(MOTOR1_PIN2, 0);
    } else {
        analogWrite(MOTOR1_PIN1, 0);
        analogWrite(MOTOR1_PIN2, -vitezaStanga);
    }

    if (vitezaDreapta > 0) {
        analogWrite(MOTOR2_PIN1, vitezaDreapta);
        analogWrite(MOTOR2_PIN2, 0);
    } else {
        analogWrite(MOTOR2_PIN1, 0);
        analogWrite(MOTOR2_PIN2, -vitezaDreapta);
    }
}
```



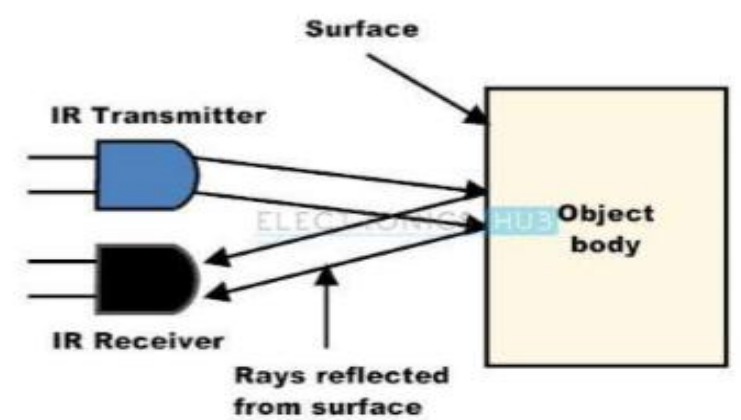
```
MERGI(MOTOR1, INAINTE);  
MERGI(MOTOR2, INAPOI);
```

- Controlul motoarelor pentru miscarea robotului catre dreapta:

```
analogWrite(MOTOR1_PIN1, vitezaStanga);  
analogWrite(MOTOR1_PIN2, 0);  
  
analogWrite(MOTOR2_PIN1, 0);  
analogWrite(MOTOR2_PIN2, -vitezaDreapta);
```

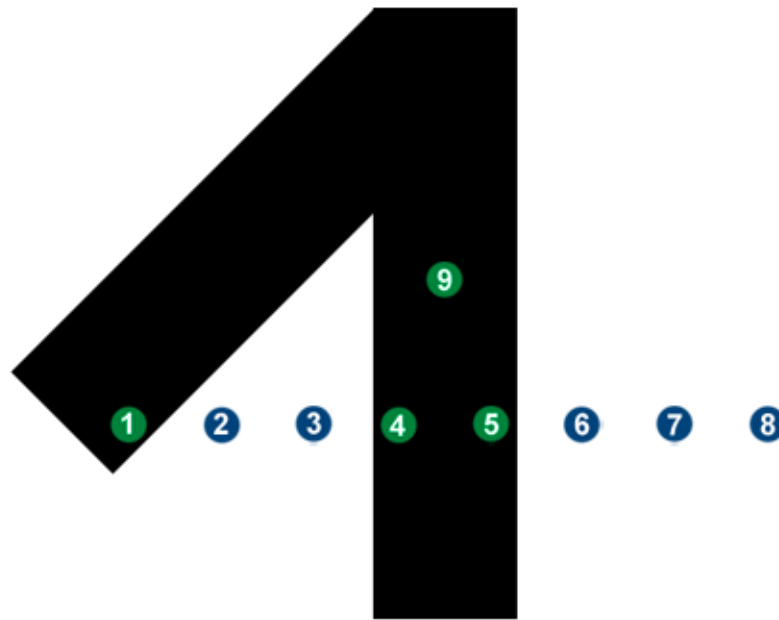


SENZORI (LINIE);

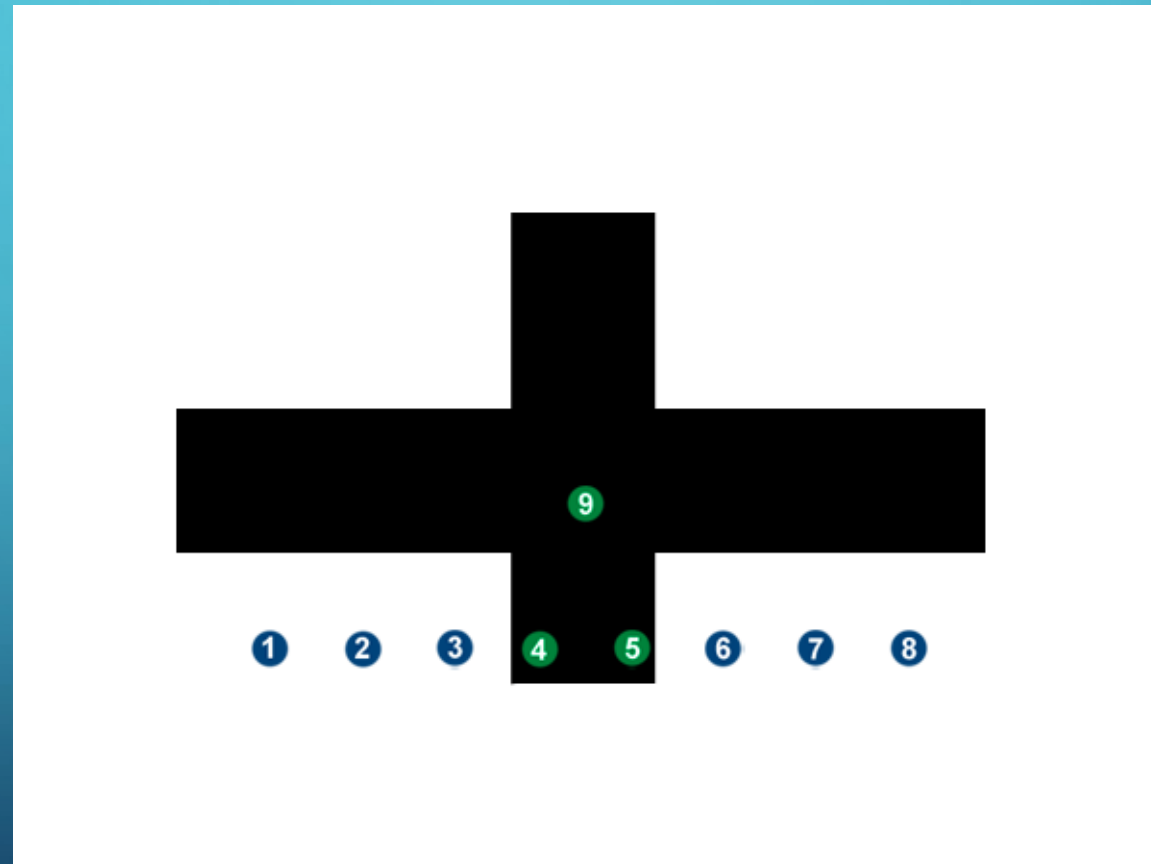


- Bareta de senzori prezintă 8 pini, fiecare provenind de la ieșirea unui fototranzistor; pe bareta se află 8 perechi LED – fototranzistor ce operează în IR (infraroșu).
- Radiația în infraroșu emisă de led lovește suprafața și este reflectată înapoi către fototranzistor care generează o tensiune invers proporțională cu cantitatea de radiație IR reflectată (pentru suprafețe reflectante vom avea o tensiune mică, iar pentru suprafețe nereflectante o tensiune mare)
- Pini de la bareta de senzori se pot conecta la pini de tip analogici sau de tip digitali de la placa Arduino, în acest ultim caz microcontrolerul citind valori digitale 0 sau 1 pentru tensiunile generate de fototranzistori.

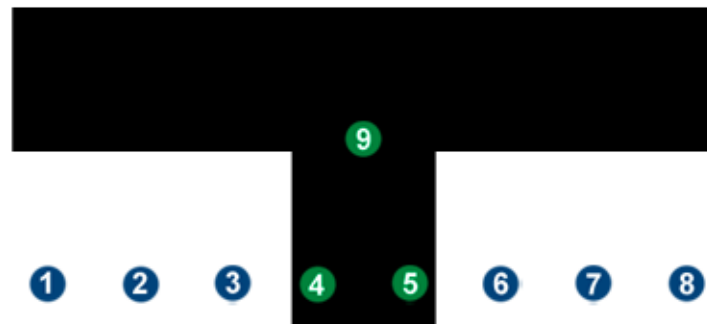
EXEMPLU(FUNCTIONARE1);



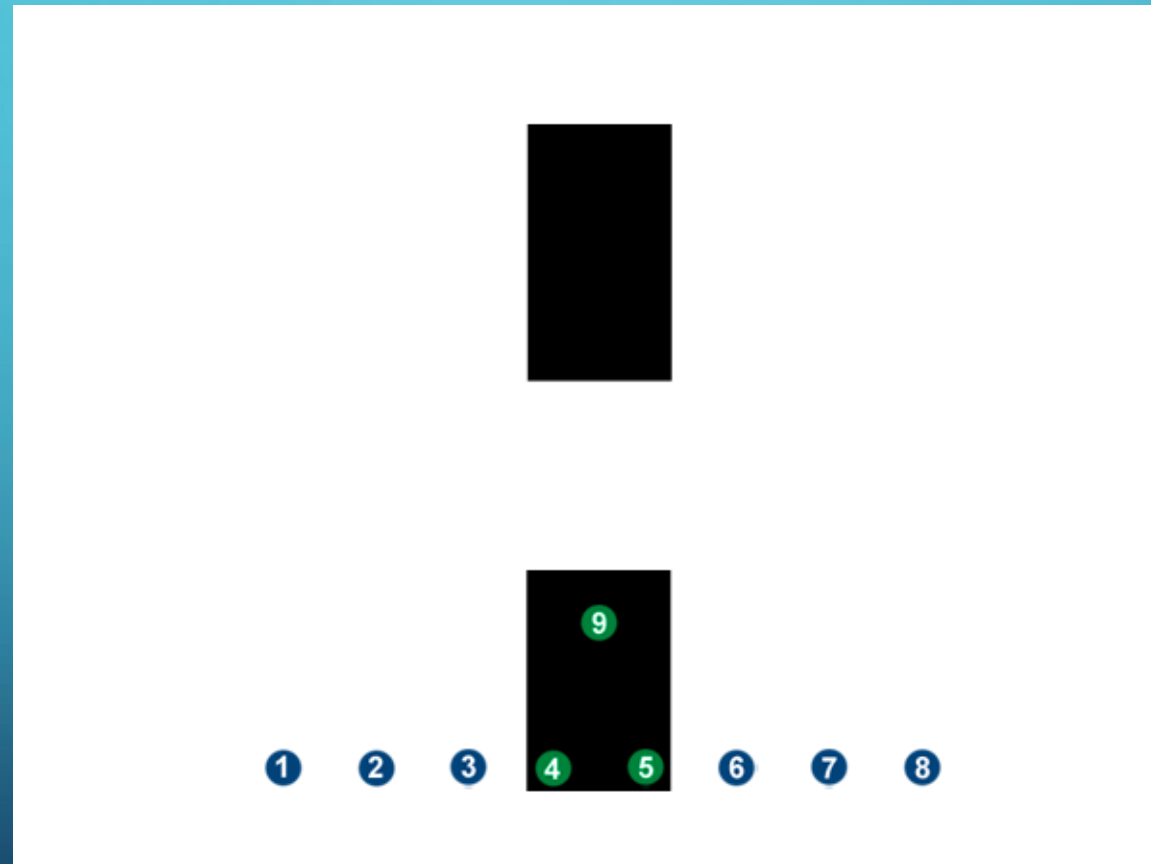
EXEMPLU(FUNCTIONARE2);



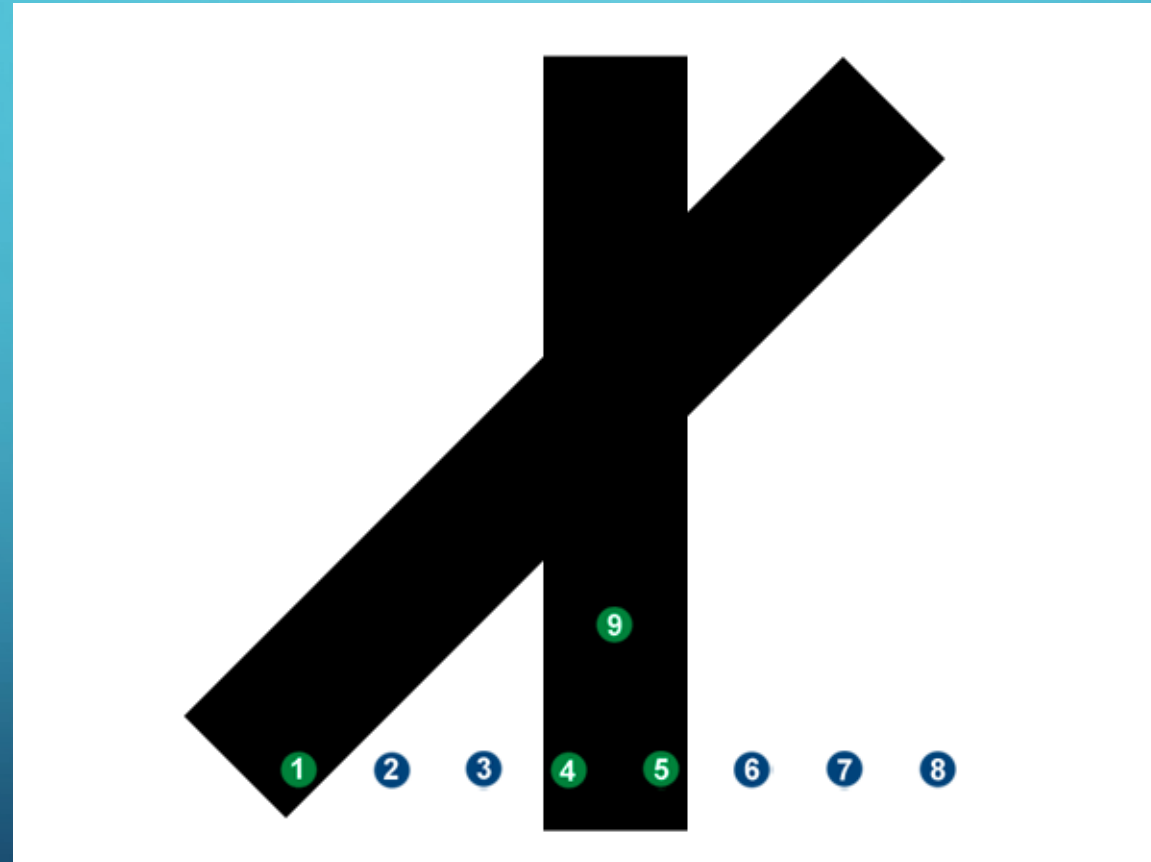
EXEMPLU(FUNCTIONARE3);



EXEMPLU(FUNCTIONARE4);



EXEMPLU(FUNCTIONARE5);



```
int IN2 = A0;  
int IN3 = A1;  
int IN4 = A2;  
int IN5 = A3;  
int IN6 = A4;  
int IN7 = A5;
```

```
int semnalIN2;  
int semnalIN3;  
int semnalIN4;  
int semnalIN5;  
int semnalIN6;  
int semnalIN7;
```

```
void setup() {  
    Serial.begin(9600);  
  
    pinMode(IN2, INPUT);  
    pinMode(IN3, INPUT);  
    pinMode(IN4, INPUT);  
    pinMode(IN5, INPUT);  
    pinMode(IN6, INPUT);  
    pinMode(IN7, INPUT);  
}
```

```
void loop() {  
    semnalIN2 = analogRead(IN2);  
    semnalIN3 = analogRead(IN3);  
    semnalIN4 = analogRead(IN4);  
    semnalIN5 = analogRead(IN5);  
    semnalIN6 = analogRead(IN6);  
    semnalIN7 = analogRead(IN7);  
  
    if ((semnalIN4 > 250) && (semnalIN5 > 250))  
        //mergi inainte  
}
```


EXEMPLU();

- Aprinderea sau stingerea a 3 LED-uri in functie de valoarea de intrare a senzorului de linie. Se exclude primul si ultimul pin, iar restul se vor grupa cate 2 si vor aprinde/stinge cate un LED din cele 3.
- $IN2 \ \&\& \ IN3 \Rightarrow LED0$
- $IN4 \ \&\& \ IN5 \Rightarrow LED1$
- $IN6 \ \&\& \ IN7 \Rightarrow LED2$



Line Following Robot