Γυμνάσιο Πυργετού

Σχ. Έτος 2023-2024

Όμιλος Υπολογιστικής Σκέψης,
Αυτοματισμών και Ρομποτικής

I2

Σχεδιάζω Εντολές Ελέγχου

1. Στον κώδικα που ακολουθεί συμπληρώνω τα σημεία που είναι απαραίτητα για να λειτουργήσουν οι κινητήρες και το servo σύμφωνα με τις προδιαγραφές που βρίσκονται στα σχόλια (//). Σε αυτή τη φάση δοκιμάζουμε το σύνολο της λειτουργίας χρησιμοποιώντας **Εντολές** από την κονσόλα.

```
#include <Servo.h>
const int LMotor1 = 2;      // Αριστερός κινητήρας 1
const int LMotor2 = 4;      // Αριστερός κινητήρας 2
const int LMotorEn = 3;     // Ακίδα PWM Αριστερού κινητήρα
const int RMotor1 = 6;      // Δεξιός κινητήρας 1
const int RMotor2 = 7;      // Δεξιός κινητήρας 2
const int RMotorEn = 5;     // Ακίδα PWM Δεξιού κινητήρα
const int UsonicTrig = 9;   // Ακίδα σκανδαλισμού
const int UsonicEcho = 8;   // Ακίδα απόκρισης
const int HookSrv = 10;     // Ακίδα Σερβομηχανισμού
const int FlashLED = 13;    // Ενσωματωμένο LED

Servo myservo;        // Ορίζω το myservo ως σερβομηχανισμό
String stateFlag="H";  /* stateFlag: Κατάσταση
                                  λειτουργίας οχήματος
                           F > Forward (Εμπρός)
                           S > Slow (Εμπρός-αργά)
                           B > Backward (Όπισθεν)
                           L > Left (Αριστερά)
                           R > Right (Δεξιά)
                           H > Halt (Σταμάτα)
                           C > Close (Κλείσε δαγκάνα)
                           O > Open (Άνοιξε δαγκάνα)   */
String grabFlag="0";   /* grabFlag: Κατάσταση θέσης δαγκάνας
                           180 > Ανοιχτή
                            0  > Κλειστή              */
```

```
void setup()
{
  pinMode(LMotor1, OUTPUT);
  pinMode(LMotor2, OUTPUT);
  pinMode(LMotorEn, OUTPUT);
  pinMode(RMotor1, OUTPUT);
  pinMode(RMotor2, OUTPUT);
  pinMode(RMotorEn, OUTPUT);
  pinMode(UsonicTrig, OUTPUT);
  pinMode(UsonicEcho, INPUT);
  pinMode(FlashLED, OUTPUT);
  myservo.attach(HookSrv, 500, 2500);
  myservo.write(0);

  Serial.begin(9600);    // Εκκίνηση της επικοινωνίας
}

void loop()
{
  // Ανάγνωση κονσόλας
  if (Serial.available() != 0) {
    // Υπάρχει εντολή για εκτέλεση
    String commCommand = Serial.readString();

    if (commCommand == "F") {     // Εντολή Forward
      motorMotion (1, 255, 1, 255);
      stateFlag="F";
    }
    else if (commCommand=="S") {  // Εντολή Slow
      motorMotion (1, 100, ______, ______);
      stateFlag="S";
    }
    else if (commCommand=="B") {  // Εντολή Backward
      motorMotion (______, ______, ______, ______);
      stateFlag=______;
    }
    else if (commCommand=="L") {  // Εντολή Left
      motorMotion (______, ______, ______, ______);
      stateFlag=______;
    }
```

```
    else if (commCommand=="R") {  // Εντολή Right
      motorMotion (______, ______, ______, ______);
      stateFlag=______;
    }
    else if (commCommand=="H") {  // Εντολή Halt
      motorMotion (______, ______, ______, ______);
      stateFlag=______;
    }
    // Εντολή Close (μόνο όταν είναι σταματημένο)
    else if (commCommand=="C" && stateFlag==______) {
      myservo.write(______);
      grabFlag=______;
      delay(100);
    }
    // Εντολή Open (μόνο όταν είναι σταματημένο)
    else if (commCommand==______ && stateFlag==______) {
      myservo.write(______);
      grabFlag=______;
      delay(100);
    }
  }
  else {
    // Δεν υπάρχει νέα Εντολή - Έλεγχος εμποδίου

    // Καταμέτρηση της απόστασης σε εκατοστά
    float cm = distance();

    // Όταν η απόσταση είναι κάτω από 20 εκατοστά
    // σταματάνε οι κινητήρες
    if (cm < ______) {
      motorMotion (______, ______, ______, ______);
      stateFlag=______;
    }
  }
  delay(30);
}
```

```
// ΛΕΙΤΟΥΡΓΙΕΣ ΠΟΥ ΕΧΟΥΝ ΓΡΑΦΕΙ ΣΕ ΠΡΟΗΓΟΥΜΕΝΕΣ
// ΕΦΑΡΜΟΓΕΣ - ΑΝΤΙΓΡΑΨΤΕ ΤΙΣ ΑΥΤΟΥΣΙΕΣ

// Λειτουργία μέτρησης απόστασης
float distance()
{
   ...
}

void motorMotion (int LMotorM, int LMotorV, int RMotorM, int
RMotorV) {
   ...
}
```