**ΕΡΓΑΣΙΑ 3-2024 (ΠΡΟΑΙΡΕΤΙΚΗ)**

Σας δίνεται κώδικας σε περιβάλλον ARDUINO ο οποίος ελέγχει την λειτουργία ενός βηματικού κινητήρα και ενός σερβοκινητήρα. Θα τροποποιήσετε τον κώδικα έτσι ώστε χρησιμοποιώντας δύο εξωτερικά Interrupts INT0 και ΙΝΤ1 συνδεδεμένα στα pin 2 και 3:

1. Θα αλλάζει η φορά περιστροφής του βηματικού κινητήρα αν προκληθεί το ΙΝΤ0
2. Θα αλλάζει η γωνία περιστροφής του σερβοκινητήρα (10 μοίρες ή 170 μοίρες.)

Οδηγίες:

Α. Δεν θα αλλάξετε τον κώδικα στην loop();

B. Θα προσθέσετε τις απαραίτητες εντολές στην setup για την ρύθμιση και ενεργοποίηση των interrupts.

C. Θα προσθέσετε δύο Interrupt Service Routines οι οποίες θα μεταβάλουν κατάλληλα τις μεταβλητές που ελέγχουν την φορά περιστροφής του βηματικού και την θέση του σερβοκινητήρα.

Θα παραδώσετε τον πηγαίο κώδικα , το εκτελέσιμο (ΗΕΧ) και δυο στιγμιότυπα που δείχνουν τον σερβοκινητήρα στις δύο διαφορετικές θέσεις και τον βιωματικό κινητήρα σε κίνηση.

Θα αξιολογηθεί θετικά η οργανωμένη δομή και ο αναλυτικός σχολιασμός του κώδικα σας.

Καλή Επιτυχία

//specify stepper connecting pins

int STEPPER0 = 8;

int STEPPER1 = 9;

int STEPPER2 = 10;

int STEPPER3 = 11;

int SERVO = 5;

int INTPIN1 = 2;

int INTPIN2 = 3;

volatile int NewServoPosition =  250;

volatile int OldServoPosition = 0;

volatile int stepperSpeed=10;

volatile int direction = 0;

void setup() {

  //set pins as outputs

  pinMode(STEPPER0, OUTPUT);

  pinMode(STEPPER1, OUTPUT);

  pinMode(STEPPER2, OUTPUT);

  pinMode(STEPPER3, OUTPUT);

  pinMode(SERVO,OUTPUT);

}

void stepper\_turn(int direction, int speed,  int steps)

{

  int count;

  if (direction == 0){

    count = 0;

    do {

      digitalWrite(STEPPER0,0);

      digitalWrite(STEPPER1,1);

      digitalWrite(STEPPER2,0);

      digitalWrite(STEPPER3,1);

      delay(speed);

      digitalWrite(STEPPER0,1);

      digitalWrite(STEPPER1,0);

      digitalWrite(STEPPER2,0);

      digitalWrite(STEPPER3,1);

      delay(speed);

      digitalWrite(STEPPER0,1);

      digitalWrite(STEPPER1,0);

      digitalWrite(STEPPER2,1);

      digitalWrite(STEPPER3,0);

      delay(speed);

      digitalWrite(STEPPER0,0);

      digitalWrite(STEPPER1,1);

      digitalWrite(STEPPER2,1);

      digitalWrite(STEPPER3,0);

      delay(speed);

      count++;

    }while ( count < steps);

  }

  else{

    count = 0;

    do {

      digitalWrite(STEPPER3,0);

      digitalWrite(STEPPER2,1);

      digitalWrite(STEPPER1,0);

      digitalWrite(STEPPER0,1);

      delay(speed);

      digitalWrite(STEPPER3,1);

      digitalWrite(STEPPER2,0);

      digitalWrite(STEPPER1,0);

      digitalWrite(STEPPER0,1);

      delay(speed);

      digitalWrite(STEPPER3,1);

      digitalWrite(STEPPER2,0);

      digitalWrite(STEPPER1,1);

      digitalWrite(STEPPER0,0);

      delay(speed);

      digitalWrite(STEPPER3,0);

      digitalWrite(STEPPER2,1);

      digitalWrite(STEPPER1,1);

      digitalWrite(STEPPER0,0);

      delay(speed);

      count++;

    }while (count < steps);

  }

}

void loop() {

  stepper\_turn(direction, stepperSpeed, 5);

  if (NewServoPosition != OldServoPosition){

    analogWrite(SERVO,NewServoPosition);

    OldServoPosition = NewServoPosition;

  }

}