Universidad del País Vasco Euskal Herriko Unibertsitatea

BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TRABAJO FIN DE GRADO

AUTONIVELACIÓN DE MESA CON SENSORES ULTRASÓNICOS. DISEÑO Y SIMULACIÓN

ANEXO II - *PROGRAMACIÓN*

Alumna: Vidaurre, Gutiérrez, Famara

Directora: Otaegi, Aizpeolea, Aloña

Curso: 2020-2021

Fecha: 04, noviembre, 2020

ÍNDICE DE CONTENIDO

1. CÓDIGO PRINCIPAL

```
//VARIABLES
int Lectura=0, Hmin=0, Distancia1=0, Distancia2=0, Distancia3=0, Distancia4=0;
const float Velocidad=20.03028125;
int time = 0; // Tiempo de espera
const int speed = 206;      //velocidad de giro
long tiempo=0; //Se va a guardar el tiempo de duración del pulso generado por el pin Echo,
               //el que utilizaremos para la lectura de sensor.
float SensorVoltaje=0; // Sensor voltaje batería
int Pulsador=0; // Pulsador. Deja una variable encendida en el momento que se activa el pulsador.


//PINES MOTOR 1
const int pinPWMMotor1 = 8;
const int pinMotorDer1 = 31;
const int pinMotorIzq1 = 32;

//PINES MOTOR 2
const int pinPWMMotor2 = 9;
const int pinMotorDer2 = 33;
const int pinMotorIzq2 = 34;

//PINES MOTOR 3
const int pinPWMMotor3 = 10;
const int pinMotorDer3 = 35;
const int pinMotorIzq3 = 36;

//PINES MOTOR 4
const int pinPWMMotor4 = 11;
const int pinMotorDer4 = 37;
const int pinMotorIzq4 = 38;


//PINES SENSOR 1
const int pinTriggerS1=23;
const int pinEchoS1=24;
int LecturaS1;


//PINES SENSOR 2
const int pinTriggerS2=25;
const int pinEchoS2=26;
int LecturaS2;

//PINES SENSOR 3
const int pinTriggerS3=27;
const int pinEchoS3=28;
int LecturaS3;

//PINES SENSOR 4
const int pinTriggerS4=29;
const int pinEchoS4=30;
int LecturaS4;



const int pinMotor1[3] = { pinPWMMotor1, pinMotorDer1, pinMotorIzq1 }; // Variables motor 1
const int pinMotor2[3] = { pinPWMMotor2, pinMotorDer2, pinMotorIzq2 }; // Variables motor 2
const int pinMotor3[3] = { pinPWMMotor3, pinMotorDer3, pinMotorIzq3 }; // Variables motor 3
const int pinMotor4[3] = { pinPWMMotor4, pinMotorDer4, pinMotorIzq4 }; // Variables motor 4

const int pinSensor1[2] = { pinTriggerS1, pinEchoS1 }; // Variables sensor 1
const int pinSensor2[2] = { pinTriggerS2, pinEchoS2 }; // Variables sensor 2
const int pinSensor3[2] = { pinTriggerS3, pinEchoS3 }; // Variables sensor 3
const int pinSensor4[2] = { pinTriggerS4, pinEchoS4 }; // Variables sensor 4
```

```
void setup()
{
  Serial.begin(9600);
  // Variables Motor1
   pinMode(pinPWMMotor1, OUTPUT); //Salida pin PWM
   pinMode(pinMotorDer1, OUTPUT); //Salida pin movimiento derecha. Avance
   pinMode(pinMotorIsq1, OUTPUT); //Salida pin movimiento isquierda. Retroceso
   // Variables Motor2
   pinMode(pinPWMMotor2, OUTPUT); //Salida pin PWM
   pinMode(pinMotorDer2, OUTPUT); //Salida pin movimiento derecha. Avance
   pinMode(pinMotorIsq2, OUTPUT); //Salida pin movimiento isquierda. Retroceso
   // Variables Motor3
   pinMode(pinPWMMotor3, OUTPUT); //Salida pin PWM
   pinMode(pinMotorDer3, OUTPUT); //Salida pin movimiento derecha. Avance
   pinMode(pinMotorIsq3, OUTPUT); //Salida pin movimiento isquierda. Retroceso
   // Variables Motor4
   pinMode(pinPWMMotor4, OUTPUT); //Salida pin PWM
   pinMode(pinMotorDer4, OUTPUT); //Salida pin movimiento derecha. Avance
   pinMode(pinMotorIsq4, OUTPUT); //Salida pin movimiento isquierda. Retroceso

   // Variables Sensor1
   pinMode (pinTriggerS1, OUTPUT);//Trigger S1 como salida
   pinMode (pinEchoS1, INPUT); //Echo S1 como entrada
  //Variables Sensor2
   pinMode (pinTriggerS2, OUTPUT);//Trigger S2 como salida
   pinMode (pinEchoS2, INPUT); //Echo S2 como entrada
  //Variables Sensor3
   pinMode (pinTriggerS3, OUTPUT);//Trigger S3 como salida
   pinMode (pinEchoS3, INPUT); //Echo S3 como entrada
  //Variables Sensor4
   pinMode (pinTriggerS4, OUTPUT);//Trigger S4 como salida
   pinMode (pinEchoS4, INPUT); //Echo S4 como entrada

   //Led rojo error sensor
   pinMode (33,OUTPUT); //Led rojo error sensor

   //Led verde batería baja
   pinMode (34,OUTPUT); //Led verde batería

   //Pulsador
   pinMode (22,INPUT); //Pulsador activa nivelación
}

void loop()
{
//COMPROBAR TENSIÓN BATERÍA.
CargaBateria();

if (digitalRead(22)==HIGH); //Resistencia Pull-down. Sistema Anti-rebote
{
  Pulsador=1; // El pulsador está activado
}

if (digitalRead(22)==LOW && Pulsador==1)
  {
       time = 0; // Ponto el tiempo a "0"

       //PUESTA A 0 de los cuatro actuadores.

       time = 1947; // Tiempo determinado porque como máximo se tendrá que mover 39 mm. 39/20,03028125*1000=1947ms.
       MovRetroceso (pinMotor1, time);
       Parada (pinMotor1);
       MovRetroceso (pinMotor2, time);
       Parada (pinMotor2);
       MovRetroceso (pinMotor3, time);
       Parada (pinMotor3);
       MovRetroceso (pinMotor4, time);
       Parada (pinMotor4);
```

```
        //PUESTA A PUNTO DE INICIO 75 mm de los cuatro actuadores.
        time = 3744; // Tiempo determinado para que se mueva 75mm será de 75/20,03028125*1000=3744ms.
        MovAvance (pinMotor1, time);
        Parada (pinMotor1);
        MovAvance (pinMotor2, time);
        Parada (pinMotor2);
        MovAvance (pinMotor3, time);
        Parada (pinMotor3);
        MovAvance (pinMotor4, time);
        Parada (pinMotor4);

        //Lectura sensores
        do
        {
        LecturaS1 = LecturaSensor(pinSensor1);
        LecturaS2 = LecturaSensor(pinSensor2);
        LecturaS3 = LecturaSensor(pinSensor3);
        LecturaS4 = LecturaSensor(pinSensor4);
        }
        while (LecturaS1==0||LecturaS2==0||LecturaS3==0||LecturaS4==0); //Hay problema en la lectura de algún sensor.

        // Ha realizado bien la lectura, sigue con la programación.

        //Hacer comparación de lecturas sensor y sacar altura mínima.
        int LecturaSensoresArray[4] = { LecturaS1, LecturaS2, LecturaS3, LecturaS4}; //Array lectura sensores
        Hmin = ComparacionSensores(LecturaSensoresArray);

        //Calculo de las distancias a mover
        Distancia1= LecturaS1 - Hmin;
        Distancia2= LecturaS2 - Hmin;
        Distancia3= LecturaS3 - Hmin;
        Distancia4= LecturaS4 - Hmin;


        //Movimiento motor según distancias
        if (Distancia1!=0) // Si es distinta, el motor1 se mueve porque implica que hay una distancia.
        {
        time = round (Distancia1/Velocidad);
        MovAvance (pinMotor1, time);
        Parada (pinMotor1);
        }
        if (Distancia2!=0)//Se mueve Motor2
        {
        time = round (Distancia2/Velocidad);
        MovAvance (pinMotor2, time);
        Parada (pinMotor2);
        }
        if(Distancia3!=0) // Se mueve Motor3
        {
        time = round (Distancia3/Velocidad);
        MovAvance (pinMotor3, time);
        Parada (pinMotor3);
        }
        if(Distancia4!=0)// Se mueve Motor4
        {
        time = round (Distancia4/Velocidad);
        MovAvance (pinMotor4, time);
        Parada (pinMotor4);
        }
        Pulsador = 0;
}
}
```

2. FUNCIONES

```
//Función avance motor
void MovAvance(const int pinMotor[3], int time)
{
   digitalWrite(pinMotor[1], HIGH); //Activas a derecha. Avance
   digitalWrite(pinMotor[2], LOW); //Desactivas a derecha
   analogWrite(pinMotor[0], speed); //Pwm velocidad
   delay (time);
}

//Función retroceso motor
void MovRetroceso(const int pinMotor[3], int time)
{
   digitalWrite(pinMotor[1], LOW); //Desactivas a derecha
   digitalWrite(pinMotor[2], HIGH);//Activas a izquierda. Retroceso
   analogWrite(pinMotor[0], speed);
   delay (time);
}

//Función parada motor
void Parada(const int pinMotor[3])
{
   digitalWrite(pinMotor[1], LOW);
   digitalWrite(pinMotor[2], LOW);
   analogWrite(pinMotor[0], 0);
   delay (time);
}

//Función carga batería
void CargaBateria()
{
SensorVoltaje=(analogRead(A0)*5)/1023; //Paso de 0-1023 a 0-5V
if (SensorVoltaje<=2.3)
  {
    digitalWrite(40,HIGH); // Enciende led verde. Batería baja
  }
else
  {
    digitalWrite(40,LOW); // Apaga led verde. Batería no está baja
  }
}


//Función lectura sensor
int LecturaSensor (const int pinSensor[2])
{
  digitalWrite (pinSensor[1], LOW); // Se asegura un cero en el pin que se usa como Trigger
  delayMicroseconds(5); // Retardo de 5 microsegundos
  digitalWrite(pinSensor[1],HIGH); //Pone en alto el pin trigger para comenzar el pulso de inicio del sensor
  delayMicroseconds(10); //Retardo de 10 microsegundos (tiempo mínimo para inicializar el trigger del sensor
  digitalWrite(pinSensor[1],LOW);// Se pone en bajo el pin Trigger tras generar el disparo de 10 microsegundos
  tiempo=pulseIn(pinSensor[2],HIGH);  // Se inicia la función de pulseIn para que mida el tiempo del pulso generado por Echo
  Lectura = tiempo*0.343/2; //Calcula de la distancia a la que se encuentra el objeto en mm
  if (681 <= Lectura <= 720)
  {
    digitalWrite(39,LOW); // Se apaga led rojo. No hay error en lectura
    return(Lectura);
  }

  else
  {
    digitalWrite(39,HIGH); //Se enciende led rojo. Hay error en lectura
    return(0);
  }
}
```

```
//Función comparación de lecturas sensor y sacar altura mínima.
int ComparacionSensores (int LecturaSensoresArray[4])
{
      if (LecturaS1 <= LecturaS2)
      {
         if (LecturaS1 <= LecturaS3)
         {
            if (LecturaS1 <= LecturaS4)
            {
              Hmin = LecturaS1;
              return(Hmin);
            }
            else
            {
              Hmin = LecturaS4;
              return(Hmin);
            }
         }
         else
         {
          if (LecturaS3 <= LecturaS4)
          {
            Hmin = LecturaS3;
            return(Hmin);
          }
          else
          {
            Hmin = LecturaS4;
            return(Hmin);
          }
         }
      }

      else
      {
         if (LecturaS2 <= LecturaS3)
         {
           if(LecturaS2 <= LecturaS4)
           {
             Hmin = LecturaS2;
             return(Hmin);
           }
           else
           {
             Hmin = LecturaS4;
             return(Hmin);
           }
         }
         else
         {
           if (LecturaS3 <= LecturaS4)
           {
             Hmin = LecturaS3;
             return(Hmin);
           }
           else
           {
             Hmin = LecturaS4;
             return(Hmin);
           }
         }
      }
```