Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Ingeniería en Ciencias de la Computación

Sistemas Empotrados

Docente: José Luis Hernández Ameca

PROYECTO FINAL

Integrantes

Bryan Arturo Cedillo García, 201934964
Christian Amauri Amador Ortega, 201927821
Isaí López Martínez, 201923778
Iván Morales Morales, 201940635
Juan Pablo Osuna Orduño, 201913278

Puebla, México
Fecha: 27 de noviembre de 2023

Índice

Introducción

En la era actual de la tecnología, la robótica ha emergido como un campo apasionante y en constante evolución que despierta la creatividad y la innovación. En este contexto, el presente reporte da cuenta del proceso de diseño, desarrollo y puesta en marcha del fascinante proyecto: la creación de un robot seguidor de líneas. Este proyecto no solo representa un desafío técnico significativo, sino que también abre las puertas a la exploración de conceptos clave en el ámbito de la robótica y los sistemas empotrados.

A medida que avanzamos en la era de la automatización, la capacidad de crear robots inteligentes capaces de interactuar con su entorno de manera autónoma se vuelve cada vez más relevante. Este proyecto no solo se trata de la creación de un robot seguidor de líneas, sino también de una exploración en profundidad de las tecnologías y conceptos que permiten a las máquinas comprender y reaccionar a su entorno de manera similar a los seres vivos.

Objetivo

El objetivo fundamental de este proyecto es diseñar un robot capaz de seguir una línea no recta trazada en una pista de manera autónoma, respondiendo a las complejidades del entorno de manera eficiente y precisa haciendo uso de las técnicas vistas en la materia de sistemas empotrados, como lo son las practicas realizadas de giros de un motor o las lecturas de un sensor.

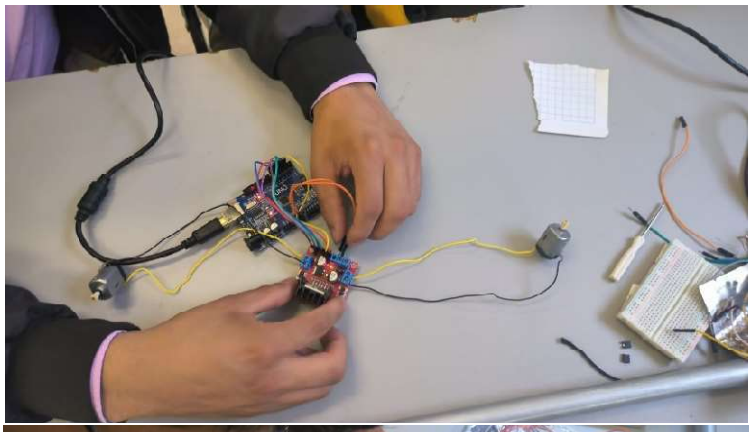
Objetivos específicos:

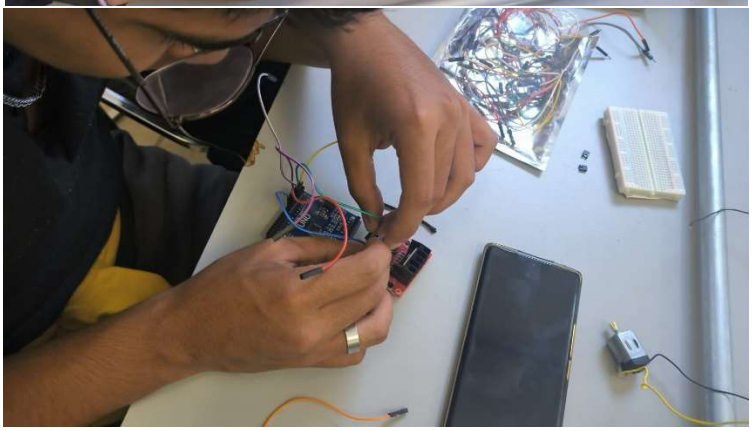
- Realizar un carro capaz de moverse a cualquier dirección cuando se le da la indicación por medio de la programación.
- Que nuestro seguidor de línea pueda terminar uno o más recorridos que se tienen disponibles en la facultad.
- Realizar la tarea de completar el circuito lo más rápido posible.

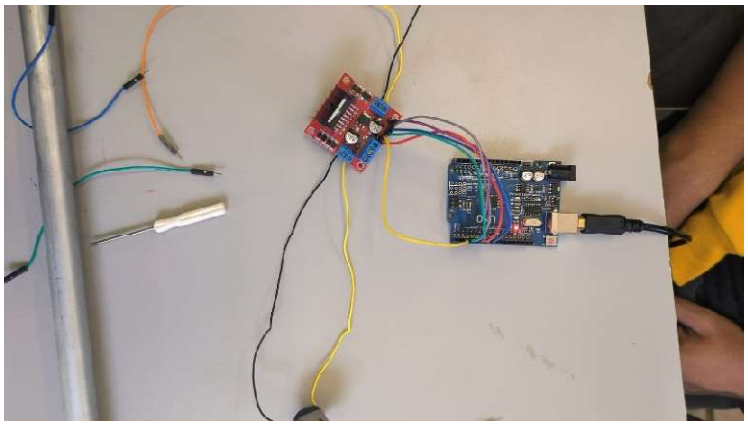
Desarrollo

(Fases de elaboración del robot)

Fase de elaboración de circuito para hacer funcionar los motores

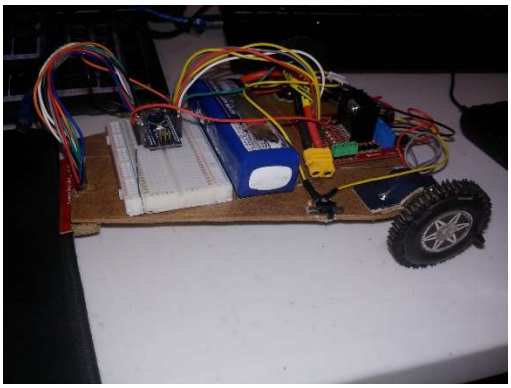




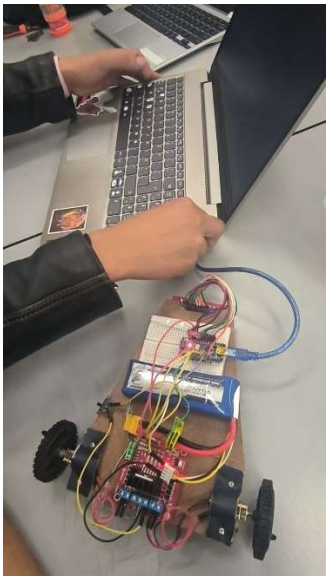


Armado de carcasa de carrito

Prototipo armado



Programacion de la lógica del carrito:



Codigo implementado:

```
#include <QTRSensors.h>
QTRSensors qtr;

const uint8_t SensorCount = 8;
uint16_t sensorValues[SensorCount];

  int MotorDer1=2; //El pin 2 de arduino se conecta con el pin In1 del L298N
  int MotorDer2=3; //El pin 3 de arduino se conecta con el pin In2 del L298N
  int MotorIzq1=7; //El pin 7 de arduino se conecta con el pin In3 del L298N
  int MotorIzq2=4; //El pin 4 de arduino se conecta con el pin In4 del L298N
  int PWM_Derecho=5; //El pin 5 de arduino se conecta con el pin EnA del
L298N
```

```
  int PWM_Izquierdo=6; //El pin 6 de arduino se conecta con el pin EnB del
L298N
  int velocidad=60; //Declaramos una variable para guardar la velocidad
  int girar=25;
  int extra=50;
  int paro=12;
  int flancoDerecho =0;
  int flancoIzquierdo =0;
  int sum;


void setup()
{


  // configure the sensors
  qtr.setTypeAnalog();
  qtr.setSensorPins((const uint8_t[]){A0, A1, A2, A3, A4, A5, A6, A7},
SensorCount);
  qtr.setEmitterPin(9);
  pinMode(MotorDer1, OUTPUT); pinMode(MotorDer2, OUTPUT);
  pinMode(MotorIzq1, OUTPUT); pinMode(MotorIzq2, OUTPUT);
  pinMode(PWM_Derecho, OUTPUT); pinMode(PWM_Izquierdo, OUTPUT);
  pinMode(paro, INPUT);

  delay(500);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, HIGH); // turn on Arduino's LED to indicate we
are in calibration mode

  // analogRead() takes about 0.1 ms on an AVR.
  // 0.1 ms per sensor * 4 samples per sensor read (default) * 6 sensors
  // * 10 reads per calibrate() call = ~24 ms per calibrate() call.
  // Call calibrate() 200 times to make calibration take about 5 seconds.
  for (uint16_t i = 0; i < 200; i++)
  {
    qtr.calibrate();
  }
  digitalWrite(LED_BUILTIN, LOW); // turn off Arduino's LED to indicate we
are through with calibration

  // print the calibration minimum values measured when emitters were on
  Serial.begin(9600);
  for (uint8_t i = 0; i < SensorCount; i++)
  {
```

```
    Serial.print(qtr.calibrationOn.minimum[i]);
    Serial.print(' ');
  }
  Serial.println();

  // print the calibration maximum values measured when emitters were on
  for (uint8_t i = 0; i < SensorCount; i++)
  {
    Serial.print(qtr.calibrationOn.maximum[i]);
    Serial.print(' ');
  }
  Serial.println();
  Serial.println();
  delay(1000);
}


void izquierda()
{
 //En esta fución la rueda derecha girará en sentido antihorario y la izquierda en sentido horario. En este caso, si las ruedas estuvieran en el chasis de un robot, el robot retrocederia./
 digitalWrite(MotorDer1,HIGH);
 digitalWrite(MotorDer2,LOW);
 digitalWrite(MotorIzq1,LOW);
 digitalWrite(MotorIzq2,HIGH);
 analogWrite(PWM_Derecho,10);//Velocidad motor derecho 200
 analogWrite(PWM_Izquierdo,velocidad+35);//Velocidad motor izquierdo 200
}
void derecha(){
 //En esta fución la rueda derecha girará en sentido horario y la izquierda en sentido antihorario. En este caso, si las ruedas estuvieran en el chasis de un robot, el robot avanzaría./
 digitalWrite(MotorDer1,HIGH);
 digitalWrite(MotorDer2,LOW);
 digitalWrite(MotorIzq1,LOW);
 digitalWrite(MotorIzq2,HIGH);
 analogWrite(PWM_Derecho,velocidad+35);//Velocidad motor derecho 200
 analogWrite(PWM_Izquierdo,10);//Velocidad motor izquierdo 200
}

void avanzar(){
 //En esta fución ambas ruedas girarán en sentido horario. En este caso, si las ruedas estuvieran en el chasis de un robot, el robot giraria a la derecha./
```

```
 digitalWrite(MotorDer1,HIGH);
 digitalWrite(MotorDer2,LOW);
 digitalWrite(MotorIzq1,LOW);
 digitalWrite(MotorIzq2,HIGH);
 analogWrite(PWM_Derecho,velocidad);//Velocidad motor derecho 200
 analogWrite(PWM_Izquierdo,velocidad);//Velocidad motor izquierdo 200
}
void ajustarDer(int dist){
 //En esta fución ambas ruedas girarán en sentido horario. En este caso, si
las ruedas estuvieran en el chasis de un robot, el robot giraria a la
derecha./
 int vel=20;
 digitalWrite(MotorDer1,HIGH);
 digitalWrite(MotorDer2,LOW);
 digitalWrite(MotorIzq1,LOW);
 digitalWrite(MotorIzq2,HIGH);
 analogWrite(PWM_Derecho,velocidad+vel*dist);//Velocidad motor derecho 200
 analogWrite(PWM_Izquierdo,velocidad-vel*dist);//Velocidad motor izquierdo
200
}
void ajustarIzq(int dist){
 //En esta fución ambas ruedas girarán en sentido horario. En este caso, si
las ruedas estuvieran en el chasis de un robot, el robot giraria a la
derecha./
 int vel=20;
 digitalWrite(MotorDer1,HIGH);
 digitalWrite(MotorDer2,LOW);
 digitalWrite(MotorIzq1,LOW);
 digitalWrite(MotorIzq2,HIGH);
 analogWrite(PWM_Derecho,velocidad-vel*dist);//Velocidad motor derecho 200
 analogWrite(PWM_Izquierdo,velocidad+vel*dist);//Velocidad motor izquierdo
200
}
void retroceder(){
 //En esta fución ambas ruedas girarán en sentido antihorario. En este caso,
si las ruedas estuvieran en el chasis de un robot, el robot giraria a la
izquierda./
 digitalWrite(MotorDer1,LOW);
 digitalWrite(MotorDer2,HIGH);
 digitalWrite(MotorIzq1,HIGH);
 digitalWrite(MotorIzq2,LOW);
 analogWrite(PWM_Derecho,velocidad);//Velocidad motor derecho 200
 analogWrite(PWM_Izquierdo,velocidad);//Velocidad motor izquierdo 200
}
```

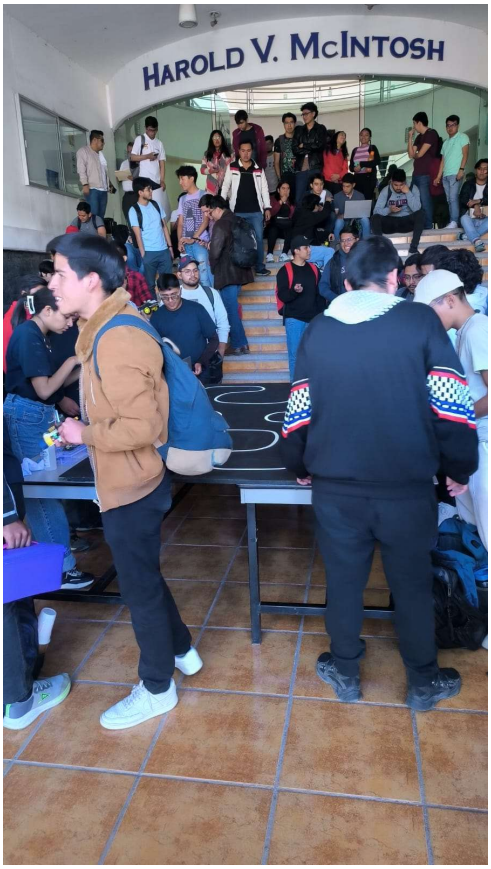
```
void loop()
{
    // read calibrated sensor values and obtain a measure of the line
position
    // from 0 to 5000 (for a white line, use readLineWhite() instead)
    uint16_t position = qtr.readLineWhite(sensorValues);
    sum = 0;
    for (uint8_t i = 0; i < SensorCount; i++)
    {
      sum+=sensorValues[i];
    }

    if(sensorValues[3]+sensorValues[4]<200)
      avanzar();
    else if(sensorValues[2]<200)
      ajustarIzq(2);
    else if(sensorValues[1]<200)
      ajustarIzq(3);
    else if(sensorValues[0]<200){
      izquierda();
      delay(100);
    }
    else if(sensorValues[5]<200)
      ajustarDer(2);
    else if(sensorValues[6]<200)
      ajustarDer(3);
    else if(sensorValues[7]<200)
    {
      derecha();
      delay(100);
    }
    else if(sum==8000)
      retroceder();
  }
```

Video de prueba de funcionamiento:

https://youtu.be/qncr8kAJ1u0

Fotos del funcionamiento

Dia del evento:



Conclusiones

El desarrollo del robot seguidor de líneas ha sido un proceso enriquecedor que nos ha permitido adentrarnos en los fundamentos de la robótica y los sistemas empotrados. A través de este proyecto, hemos logrado alcanzar varios hitos significativos y obtener valiosas lecciones que resumen nuestro semestre y conocimientos previos acerca del mismo tema.

Logramos ver las capacidades de los demás equipos y su perspectiva para la solución del problema dado, con varias formas y tamaños, aunque una forma en particular si era muy popular y veloz gracias a la potencia de sus motores, su tamaño y su forma compacta.

Pudimos aprender sobre la conexión de nuestro circuito ya que batallábamos un poco en lograr que nuestro carro lograra moverse y utilizar el Arduino, pero se pudo solucionar gracias a la conexión que realizamos, conectando el driver al Arduino y el sensor a este.

En especial pudimos aprender sobre los sensores de qtr y su configuración, el uso de drivers de motores para realizar funciones de movimiento y poder utilizarlos simultáneamente para que se ayuden entre si.