

Manual Técnico



24 FEBRERO

Grupo # 09

Sergio Sebastián Sandoval Ruiz 202010298

Karen Elisa López Pinto 201313996

Karina Nohemi Ramirez Orellana 201900957

Introducción

Este manual proporciona una guía detallada para la construcción, conexión y programación de una banda transportadora y clasificadora controlada por Arduino. El sistema está diseñado para transportar objetos de un extremo a otro, mientras los clasifica en diferentes categorías según ciertos criterios predefinidos (tamaño y color).

Componentes Requeridos

- Placas de Arduino Uno
- Motor Steppers
- Sensor ultrasónico HC – SR04
- Protoboards
- Jumpers
- Sensor de color
- Pantalla lcd 16x2 con modulo I2C
- Sensor de color TCS230

Arquitectura del Sistema

Construcción del Sistema

- **Montaje Mecánico:**
 - Construcción la estructura para sostener la banda transportadora y los componentes de clasificación.
 - Instalación de motor stepper en un extremo de la banda transportadora.
 - Instalación de que la banda esté correctamente alineada y pueda moverse libremente.
- **Instalación de Sensores:**
 - Colocación del sensor de proximidad o ultrasónico **en** un lugar donde pueda detectar los objetos que pasan por la banda.
 - Ajuste de la posición y el ángulo del sensor tanto de tamaño como de color según sea necesario para una detección precisa.
- **Conexión de Componentes:**
 - Conexión del motor Stepper al Arduino para controlar la velocidad y dirección de la banda.
 - Conexión del sensor ultrasónico al Arduino para detectar la presencia de objetos.
 - Conexión de los módulos de clasificación al Arduino para activarlos según sea necesario, por medio de un motor stepper que realiza una fracción de su recorrido para separar en diferentes caminos el tipo de objeto identificado.
- **Montaje Eléctrico:**
 - Conecta todos los componentes a una fuente de alimentación adecuada, asegurándote de respetar las especificaciones de voltaje y corriente de cada componente.

Programación del Arduino

Definición de rol en placa arduino

- Para esta sistema se requirió de la función denomina “Maestro y esclavo” donde dos Arduinos trabajan simultáneamente, donde uno asume el rol de controlador y principal y el complementario el rol de ejecutar funciones y devolverle la información requerida al principal.

Bibliotecas

- **#include "Wire.h"**
La biblioteca "Wire.h" es comúnmente utilizada en proyectos que involucran la comunicación por I2C (Inter-Integrated Circuit), un bus serial de comunicación utilizado para conectar microcontroladores, sensores, actuadores y otros dispositivos en un sistema embebido.
- **#include <LiquidCrystal_I2C.h>**
La biblioteca "LiquidCrystal_I2C.h" se utiliza comúnmente en proyectos que involucran pantallas LCD controladas mediante el protocolo I2C. Esta biblioteca simplifica la interfaz con la pantalla LCD y permite controlarla fácilmente con un microcontrolador, como Arduino, a través del bus I2C.
- **#include <EEPROM.h>**
La biblioteca "EEPROM.h" se utiliza para interactuar con la memoria EEPROM (Electrically Erasable Programmable Read-Only Memory) en placas Arduino u otros microcontroladores compatibles. La memoria EEPROM es una forma de memoria no volátil que permite almacenar datos de forma permanente incluso cuando se apaga la alimentación.
- **#include <Stepper.h>**
La biblioteca "Stepper.h" es una biblioteca estándar de Arduino que se utiliza para controlar motores paso a paso (stepper motors). Los motores paso a paso son dispositivos electromecánicos que convierten pulsos eléctricos en movimientos mecánicos discretos y precisos.

Algoritmos

- **Menú**
Algoritmo desarrollado para la impresión de las distintas opciones que a su vez cada una de las opciones se subdividen en otras opciones, para formar el Menú que permite realizar las diversas acciones del sistema. Todo controlado por medio de estados que controla y dirige la dirección del usuario entre las opciones.

```

if (estado == 1 && digitalRead(btnDownPin) == LOW) {
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("Histrorial");
    lcd.setCursor(0, 1);
    lcd.print(">Filtro");
    estado = 2;
    delay(300);
} else if (estado == 2 && digitalRead(btnDownPin) == LOW) {
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("Filtro");
    lcd.setCursor(0, 1);
    lcd.print(">Reset");
    estado = 3;
    delay(300);
} else if (estado == 3 && digitalRead(btnUpPin) == LOW) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(">Filtro");
    lcd.setCursor(1, 1);
    lcd.print("Reset");
    estado = 2;
    delay(300);
} else if (estado == 2 && digitalRead(btnUpPin) == LOW) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(">Histrorial");
    lcd.setCursor(1, 1);
    lcd.print("Filtro");
    estado = 1;
    delay(300);
}

```

- **Sensor**

Algoritmo dedicado a la detección de objeto presente de la banda y el cual accionara el funcionami de traslado de la banda.

```

int Sensor() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    pingTravelTime = pulseIn(echoPin, HIGH);
    Serial.println(pingTravelTime);

    if (pingTravelTime < 520 && pingTravelTime != 0) {
        Serial.println("Objeto detectado");
        return true;
    } else {
        Serial.println("Ningún objeto detectado");
        return false;
    }
}

void motorder1() {
    motorder.step(-260);
    delay(1000);
}

void motorder2() {
    motorder.step(260);
    delay(1000);
}

```

- **Clasificación de objetos**

- **Por Color:**

Clasificación realizada por el sensor que identifica la terminología RGB del objeto del cual esta tomando los datos y seguidamente son analizados para la identificación de su color.

```

if (rojo >= 180 && rojo <= 245 && verde >= 45 && verde <= 80 && azul >= 65 && azul <= 75) {
    Serial.println("VERDE");
    CODE = 1;
} else if (rojo >= 370 && rojo <= 470 && verde >= 85 && verde <= 130 && azul >= 100 && azul <= 120) {
    Serial.println("CELESTE");
    CODE = 2;
} else if (rojo >= 150 && rojo <= 179 && verde >= 50 && verde <= 80 && azul >= 60 && azul <= 80) {
    Serial.println("AMARILLO");
    CODE = 3;
}

```

- **Por tamaño:**

Clasificación realizada por el sensor ultrasónico que devuelve una variedad de dígitos que definen las diferentes distancias que reconoció cuando sus ondas rebotan contra el objeto presente en su recorrido. Donde se definio que entre mas cercano el objeto menor será el dígito devuelto por el sensor.

```
if(pingTravelTime >= 150 && pingTravelTime <= 210){  
    Serial.println("GRANDE");  
    CODE = 4;  
}else if (pingTravelTime >= 240 && pingTravelTime<= 290){  
    Serial.println("MEDIANO");  
    CODE = 5;  
}else if (pingTravelTime>=420 && pingTravelTime<= 500){  
    Serial.println("PEQUENO");  
    CODE = 6;
```