

UNIVERSIDAD PÚBLICA DE EL ALTO

INGENIERIA DE SISTEMAS



**CINTA TRANSPORTADORA
PRIMER PARCIAL**

Universitarios:

WILLY MARCOS CHANA TITO

GILDA ALCON MAURICIO

Carrera: INGENIERIA DE SISTEMAS

Docente: ING. ELIAS ALI ALVAREZ

Materia: ROBOTICA INDUSTRIAL

Paralelo: 9° C

Fecha: 29/04/2021

WWW.EDUCA.COM.BO

Estudiantes: Willy Marcos Chana Tito - Gilda Alcón Mauricio

INFORME

INTRODUCCION

En las últimas tres décadas, se ha producido el gran progreso en el transporte, influenciado por las nuevas tecnologías y el desarrollo de diferentes herramientas informáticas y de automatización que aparecen a lo largo de los años.

Las cintas transportadoras fueron uno de los inventos más importantes en 1795, debido a que estos ayudaron a facilitar el transporte de las materias primas y otros objetos que al transportarlos en vehículos o con máquinas resultan poco eficientes y muy costosos en combustible, y esto impulso el crecimiento de la industria.

En este proyecto se aplicará todo el conocimiento adquirido hasta el momento en la materia de Robótica Industrial.

Aplicaremos el uso de Arduino, sensores y motores en el funcionamiento de una pequeña cinta transportadora con contador de objetos y control a distancia por infrarrojo.

MARCO TEORICO

¿Qué es una Cinta Transportadora?

Definición. - Son sistemas de transporte que funcionan de manera continua y que se encuentran formadas de forma general por bandas que son movidas por medio de tambores, dos.

Dentro de estas cintas transportadoras podemos encontrar dos tipos o clasificaciones. La primera de ellas es la banda usada para transportar peso ligero y la segunda para trabajo pesado.

Así mismo debes entender el funcionamiento del equipo el cual es arrastrado, la banda, haciendo uso de la fricción que ejerce un tambor. De los dos tambores solo uno es accionado gracias a un motor mientras que el otro lo que hace es girar de

manera libre. Lo que hace el tambor que no maneja accionamiento es funcionar como pieza de retorno de la banda.

De esta forma y cuando sobre la banda en movimiento ponemos algún material lo que ocurre es que este es llevado en dirección hacia el tambor que se acciona con motor. Al llegar a este punto el material es recogido y la banda gira para regresar en el sentido contrario.

Componentes de las cintas

Conocidos los aspectos anteriores de las cintas transportadoras es hora de conocer sus componentes principales:

Bandas transportadoras: son las que soportan los objetos y las cuales permiten su transporte. Estas se encuentran de varias clases, considerando su tipo de tejido, la disposición del mismo y su aspecto.

Rodillos y los soportes: en las cintas transportadoras los rodillos son fundamentales ya que gracias a ellos es que el sistema se activa, debido al giro de los mismos. Los rodillos además cumplen como función el soporte de la banda, al centrando de la misma y además al soporte del material.

Tensores: estos dispositivos son esenciales también en el equipo ya que son los que permiten que el contacto entre banda y tambor de acción sea el adecuado. Así mismo evita que los objetos se derramen por los costados y mantienen la tensión en el ramal.

Los Bastidores: dan el soporte necesario a la banda y por ende soportan también a los rodillos y el material.

Los reductores suspendidos y clásicos.

Materiales Electrónicos

Listado de materiales que Utilizaremos:

- 1 fototransistor infrarrojo
- 1 control remoto
- Arduino Mega
- PC previa instalación del IDE de ARDUINO.
- Cable USB
- Cables de conexión.
- Pinzas y alicates
- Cables de conexión
- Fuente de poder
- Multímetro digital
- Protoboard
- Controlador L298N
- Motor Paso a Paso
- Sensor Infrarrojo
- Pantalla LCD
- Controlador I2C para LCD

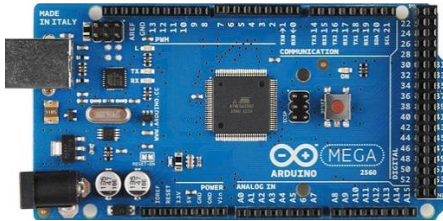
Descripción de los materiales más Importantes que Utilizaremos

Arduino Mega 2560

El Arduino Mega 2560 es una placa de desarrollo basada en el microcontrolador ATmega2560. Tiene 54 entradas/salidas digitales (de las cuales 15 pueden ser usadas como salidas PWM), 16 entradas analógicas, 4 UARTs, un cristal de 16Mhz, conexión USB, jack para alimentación DC, conector ICSP, y un botón de reseteo.

La placa Mega 2560 es compatible con la mayoría de shields compatibles para Arduino UNO.

El Arduino Mega 2560 sera utilizado para controlar la parte lógica de nuestro proyecto siendo el componente más importante.

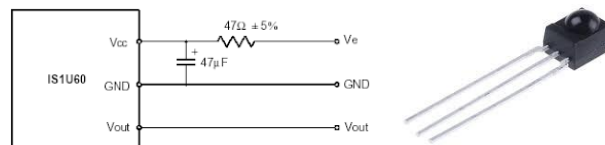


Fototransistor Infrarrojo (IS1U60)

Un fototransistor es un dispositivo semiconductor capaz de detectar los niveles de luz y alterar la corriente que fluye entre el emisor y el colector según el nivel de luz que recibe.

Tanto los fototransistores como los fotodiodos pueden utilizarse para detectar la luz, pero el fototransistor es más sensible en vista de la ganancia que proporciona el hecho de que es un transistor bipolar. Esto hace que los fototransistores sean más adecuados en varias aplicaciones.

El fototransistor IS1U60 funciona con 5V y este es sensible a la luz infrarroja que emiten los controles remotos, como los de los televisores, reproductores DVD, controles de radios ,etc.



Fuente de poder

Una fuente de poder o fuente de alimentación es componente electrónico que sirve para abastecer de electricidad al computador. Un nombre más adecuado sería el de

transformador, porque convierte o transforma corriente alterna (AC) en corriente directa (DC), y baja el voltaje de 120 voltios AC a 12,5 voltios DC, necesarios para la PC y sus componentes. También asegura que esta no opere a menos que la corriente que se suministre sea suficiente para que funcione de forma adecuada

Usaremos una fuente de poder de 25V, 2A para alimentar a nuestro motor paso a paso.



Multímetro digital

Un multímetro, también denominado polímetro¹ o tester, es un instrumento eléctrico portátil para medir directamente magnitudes eléctricas activas, como corrientes y potenciales (tensiones), o pasivas, como resistencias, capacidades y otras.

Las medidas pueden realizarse para corriente continua o alterna y en varios márgenes de medida cada una. Los hay analógicos y posteriormente se han introducido los digitales cuya función es la misma, con alguna variante añadida.

Este sera utilizado para comprobar el voltaje que alimentara nuestro motor paso a paso.



Controlador L298N

El L298N es un controlador (driver) de motores, que permite encender y controlar dos motores de corriente continua desde Arduino, variando tanto la dirección como la velocidad de giro.

El L298N también puede controlar un único motor de paso a paso aunque, en general, preferiremos usar dispositivos específicamente diseñados para motores paso a paso.

La corriente máxima que el L298N puede suministrar a los motores es, en teoría, 2A por salida (hasta 3A de pico) y una tensión de alimentación de 3V a 35V.

Básicamente un L298N consiste en dos puentes-H, uno para la salida A y otro para la salida B.

Un puente-H es un componente ampliamente utilizado en electrónica para alimentar una carga de forma que podemos invertir el sentido de la corriente que le atraviesa.

Internamente un puente-H es una formación de 4 transistores, conectados entre Vcc y GND, con la carga a alimentar entre ellos. Dibujado en esquema el conjunto tiene forma de "H", de la que recibe su nombre su nombre.



Motor Paso a Paso (18° 25V 1,2A)

Un motor paso a paso es un dispositivo electromecánico que convierte una serie de pulsos eléctricos en desplazamientos angulares, lo que significa que es capaz de

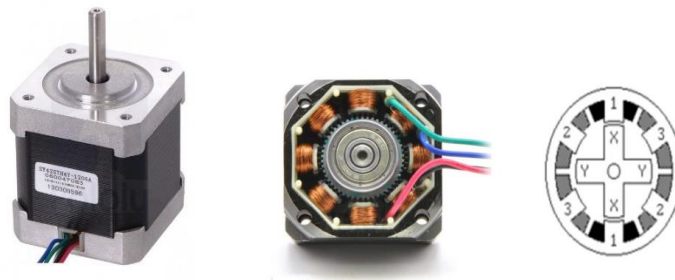
Estudiantes: Willy Marcos Chana Tito - Gilda Alcón Mauricio

girar una cantidad de grados (paso o medio paso) dependiendo de sus entradas de control.

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos. La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde 90° hasta pequeños movimientos de 1.8° , Es por eso que ese tipo de motores son muy utilizados, ya que pueden moverse a deseo del usuario según la secuencia que se les indique a través de un microcontrolador.

El motor paso a paso utilizado en este proyecto sera uno unipolar.

Estos motores suelen tener 5 o 6 cables de salida dependiendo de su conexionado interno, suelen ser 4 cables por los cuales se recibe los pulsos que indican la secuencia y duración de los pasos y los restantes sirven como alimentación del motor. Este tipo se caracteriza por ser más simple de controlar.



Sensor Infrarrojo (IR 5V)

El detector de obstáculos infrarrojo es un dispositivo que detecta la presencia de un objeto mediante la reflexión que produce en la luz. El uso de luz infrarroja (IR) es simplemente para que esta no sea visible para los humanos.

Constitutivamente son sensores sencillos. Se dispone de un LED emisor de luz infrarroja y de un fotodiodo (tipo BPV10NF o similar) que recibe la luz reflejada por un posible obstáculo.

Los detectores de obstáculo suelen proporcionarse con una placa de medición estándar con el comparador LM393, que permite obtener la lectura como un valor digital cuando se supera un cierto umbral, que se regula a través de un potenciómetro ubicado en la placa.

Este tipo de sensores actúan a distancias cortas, típicamente de 5 a 20mm. Además la cantidad de luz infrarroja recibida depende del color, material, forma y posición del obstáculo, por lo que no disponen de una precisión suficiente para proporcionar una estimación de la distancia al obstáculo.



Pantalla LCD (16x2 5V)

El LCD(Liquid Crystal Display) o pantalla de cristal líquido es un dispositivo empleado para la visualización de contenidos o información de una forma gráfica, mediante caracteres, símbolos o pequeños dibujos dependiendo del modelo. Está gobernado por un microcontrolador el cual dirige todo su funcionamiento.

En este caso vamos a emplear un LCD de 16x2, esto quiere decir que dispone de 2 filas de 16 caracteres cada una. Los píxeles de cada símbolo o carácter, varían en función de cada modelo.

Este será utilizado para mostrar el conteo de objetos

Estudiantes: Willy Marcos Chana Tito - Gilda Alcón Mauricio



Controlador I2C para LCD

Abreviatura de Inter-IC (inter integrated circuits), un tipo de bus diseñado por Philips Semiconductors a principios de los 80s, que se utiliza para conectar circuitos integrados (ICs). El I2C es un bus con múltiples maestros, lo que significa que se pueden conectar varios chips al mismo bus y que todos ellos pueden actuar como maestro, sólo con iniciar la transferencia de datos. Este bus se utiliza dentro de una misma placa de un dispositivo.

El bus I2C, un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos con cierto nivel de «inteligencia», sólo requiere de dos líneas de señal y un común o masa. Fue diseñado a este efecto por Philips y permite el intercambio de información entre muchos dispositivos a una velocidad aceptable, de unos 100 Kbits por segundo, aunque hay casos especiales en los que el reloj llega hasta los 3,4 MHz.

La metodología de comunicación de datos del bus I2C es en serie y sincrónica. Una de las señales del bus marca el tiempo (pulsos de reloj) y la otra se utiliza para intercambiar datos.



MARCO APLICATIVO

Procedimiento Experimental

En primer lugar, comenzamos codificando el sensor Infrarrojo para el conteo de objetos, como también La pantalla LCD,

Luego comenzamos el armado del circuito para leer los datos del control remoto infrarrojo, en el cual fuimos tabulando los códigos en formato decimal que nos mandaban los botones del control remoto al fototransistor, y que capturamos del monitor serial de IDE de Arduino

En nuestro caso tabulamos la siguiente tabla para nuestro control remoto

Botón de control que Utilizaremos	Código Hexadecimal	Código Decimal	Uso
Power	7070D02F	1886441519	Encender el motor
Mute	707028D7	1886398679	Cambiar la dirección de rotación del motor
1	707000FF	1886388479	Velocidad de motor lenta
2	7070807F	1886421119	Velocidad de motor media
3	707040BF	1886404799	Velocidad de motor rápida

En nuestro caso utilizaremos los códigos decimales para facilitar la comparación con la estructura de control SWITCH de C++, ya que esta solo acepta Datos Enteros a comparar.

Una vez comprendido el funcionamiento del control con el fototransistor, y el sensor infrarrojo con el Display LCD podemos proceder a simular el circuito.

Estudiantes: Willy Marcos Chana Tito - Gilda Alcón Mauricio

En nuestro caso haremos dos Simulaciones debido a que en Protheus es difícil simular un control remoto para el fototransistor y que también en TinkerCad no se encuentra el sensor infrarrojo pero el funcionamiento será normal ya que estos circuitos son independientes y solo comparten el Arduino

Código Implementado

```
1. #include <IRremote.h>
2. #include <Stepper.h> //Importamos la librería para controlar motores paso a paso
3. #include <Wire.h>
4. #include <LiquidCrystal_I2C.h>
5. #include <LiquidCrystal.h> // Incluye la Libreria LiquidCrystal
6.
7. #define STEPS 200 //definimos el número de pasos que necesita para dar una vuelta. 200
   en nuestro caso
8. LiquidCrystal_I2C lcd(0x27, 16, 2); // importa libreria LiquidCrystal
9.
10. int pinIN = 8; // Crea Variable pinIN y la asocia al pin digital 8
11. int valorIN = 0; // Crea Variables con un valor inicial asociado
12. int contenido = 0;
13.
14. // Ponemos nombre al motor, el número de pasos y los pins de control
15. Stepper stepper(STEPS, 8, 9, 10, 11);
16.
17. int RECV_PIN = 2;
18. // declaramos receptor
19. IRrecv irrecv(RECV_PIN);
20.
21. long valor;
22. int pasos = 200; //variable de pasos de motor
23. boolean power = false;
24. //decodificamos resultados
25. decode_results results;
26.
27. void setup()
28. {
29. //***** Motor *****
30. // pasos del motor en RPM
31. stepper.setSpeed(10); // pasos 0 para apagado
32. Serial.begin(9600);
33. irrecv.enableIRIn(); // Iniciar el receptor
34.
35. //***** LCD *****
36. lcd.backlight();
37. lcd.init(); // inicializa lcd en 16 columnas por 2 filas
```

Estudiantes: Willy Marcos Chana Tito - Gilda Alcón Mauricio

```

38. // entradas analógicas no requieren inicialización
39. lcd.print("Robotica industrial"); // Imprime "RSLICING3D" sobre el LCD
40. delay(500); // Espera 1 segundo
41. lcd.setCursor(0, 1); // Seteamos la ubicacion texto 0 linea 1 que sera escrita sobre el LCD
42. delay(1500);
43. lcd.print("PRIMER PARCIAL"); // Imprime sobre el LCD
44. delay(500); // Espera 1 segundo
45. lcd.setCursor(0, 2);
46. lcd.setCursor(1, 0); // Seteamos la ubicacion del texto 1, linea 0 que sera escrita
    sobre el LCD
47. lcd.print("CUENTA OBJETOS "); // Imprime "CUENTA OBJETOS" sobre el LCD
48. lcd.setCursor(2, 1);
49. lcd.print("CANTIDAD: ");
50. }
51.
52. void loop()
53. {
54. //***** LCD *****
55. valorIN = digitalRead(pinIN); // Realiza la lectura Digital del pin 8
56. if (valorIN != 1) // Si la lectura es 1 incremente el valor de contenido
57. {
58.     contenido++;
59.     while (digitalRead(pinIN) != 1)
60.     {
61.         delay(100); // Realiza un Delay para estabilizar
62.     }
63.     lcd.setCursor(12, 1); // Ubicamos el cursor en la posicion 12 sobre la linea 1
64.     lcd.print(contenido); // Imprimimos el valor del contenido sobre dicha ubicacion
65. }
66.
67.
68. //***** Motor *****
69. if (irrecv.decode(&results))
70. {
71.     valor = results.value;
72.     //imprimimos el valor en decimal
73.     Serial.println(results.value, DEC);
74.     // Recibe el siguiente valor
75.     irrecv.resume();
76. }
77. switch (valor)
78. {
79. case 1886437439: //boton power
80.
81.     power = !power;
82.     if (power)
83.     {

```

Estudiantes: Willy Marcos Chana Tito - Gilda Alcón Mauricio

```
84.     stepper.setSpeed(30);
85. }
86. else
87. {
88.     stepper.setSpeed(0);
89. }
90.
91. break;
92. case 1886398679:
93.
94.     pasos = pasos * (-1); //invertimos la direccion
95.
96.     break;
97. case 1886388479: //velocidad del motor lenta
98.
99.     stepper.setSpeed(30);
100.
101.     break;
102. case 1886421119: //velocidad del motor media
103.     stepper.setSpeed(65);
104.
105.     break;
106. case 1886404799: //velocidad del motor rapida
107.     stepper.setSpeed(100);
108.
109.     break;
110. }
111.
112.     valor = 0; //limpiamos el valor del transitor y control remoto
113.     stepper.step(pasos); //asigna los pasos del motor
114.     delay(50); //Pequeña pausa
115.
116. } //fin de loop
117.
```

Codigo en IDE Arduino

```
primer_parcial

#include <IRremote.h>
#include <Stepper.h> //Importamos la libreria para controlar motores paso a paso
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <LiquidCrystal.h> // Incluye la Libreria LiquidCrystal

#define STEPS 200 //Ponemos el número de pasos que necesita para dar una vuelta. 200 en nuestro caso
LiquidCrystal_I2C lcd(0x27, 16, 2); // importa libreria LiquidCrystal

int pinIN = 8; // Crea Variable pinIN y la asocia al pin digital 8
int valorIN = 0; // Crea Variables con un valor inicial asociado
int contenido = 0;

// Ponemos nombre al motor, el número de pasos y los pins de control
Stepper stepper(STEPS, 8, 9, 10, 11);

int RECV_PIN = 2;
// declaramos receptor
IRrecv irrecv(RECV_PIN);

long valor;
int pasos = 200; //variable de pasos de motor
boolean power = false;
//decodificamos resultados
decode_results results;

void setup()
{
  //***** Motor *****
  // pasos del motor en RPM
  stepper.setSpeed(10); // pasos 0 para apagado
  Serial.begin(9600);
  irrecv.enableIRIn(); // Iniciar el receptor

  //***** LCD *****
  lcd.backlight();
  lcd.init(); // inicializa lcd en 16 columnas por 2 filas
  // entradas analógicas no requieren inicialización
  lcd.print("Robotica industrial"); // Imprime "RSLICING3D" sobre el LCD
  delay(500); // Espera 1 segundo
  lcd.setCursor(0, 1); // Seteamos la ubicacion texto 0 linea 1 que sera escrita sobre el LCD
  delay(1500);
  lcd.print("PRIMER PARCIAL"); // Imprime sobre el LCD
  delay(500); // Espera 1 segundo
  lcd.setCursor(0, 2);
  lcd.setCursor(1, 0); // Seteamos la ubicacion del texto 1, linea 0 que sera escrita sobre el LCD
  lcd.print("CUENTA OBJETOS "); // Imprime "CUENTA OBJETOS" sobre el LCD
  lcd.setCursor(2, 1);
  lcd.print("CANTIDAD: ");
}
```

Estudiantes: Willy Marcos Chana Tito - Gilda Alcón Mauricio

primer_parcial

```
void loop()
{
    //***** LCD *****
    valorIN = digitalRead(pinIN); // Realiza la lectura Digital del pin 8
    if (valorIN != 1)              // Si la lectura es 1 incremente el valor de contenido
    {
        contenido++;
        while (digitalRead(pinIN) != 1)
        {
            delay(100); // Realiza un Delay para estabilizar
        }
        lcd.setCursor(12, 1); // Ubicamos el cursor en la posicion 12 sobre la linea 1
        lcd.print(contenido); // Imprimimos el valor del contenido sobre dicha ubicacion
    }

    //***** Motor *****
    if (irrecv.decode(&results))
    {
        valor = results.value;
        //imprimimos el valor en decimal
        Serial.println(results.value, DEC);
        // Recibe el siguiente valor
        irrecv.resume();
    }
    switch (valor)
    {
        case 1886437439: //boton power

            power = !power;
            if (power)
            {
                stepper.setSpeed(30);
            }
            else
            {
                stepper.setSpeed(0);
            }

            break;
        case 1886398679:

            pasos = pasos * (-1); //invertimos la direccion

            break;
        case 1886388479: //velocidad del motor lenta

            stepper.setSpeed(30);

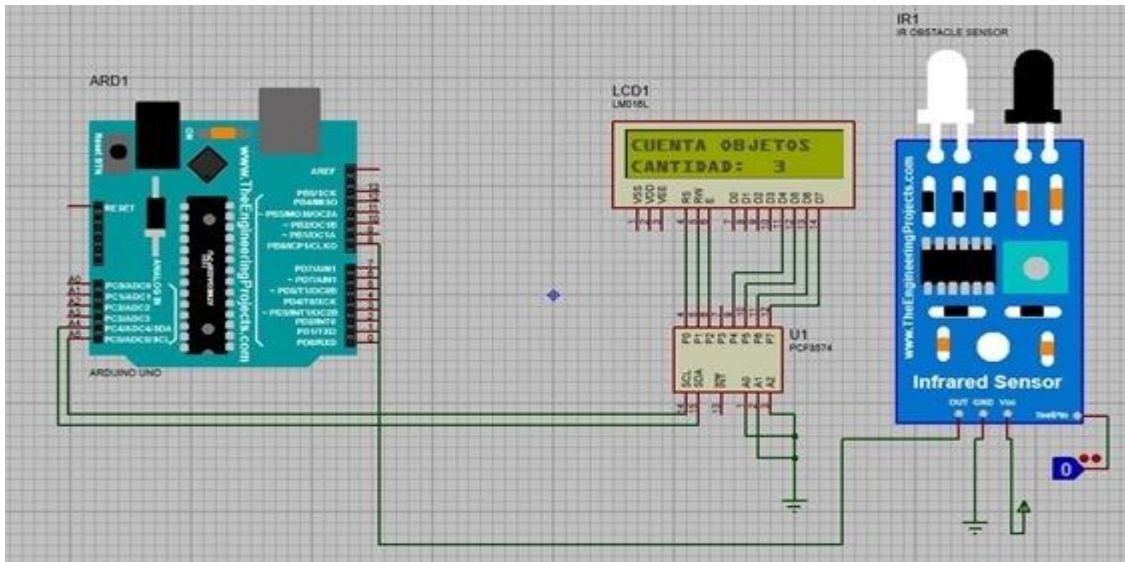
            break;
        case 1886421119: //velocidad del motor media
            stepper.setSpeed(65);

            break;
        case 1886404799: //velocidad del motor rapida
            stepper.setSpeed(100);

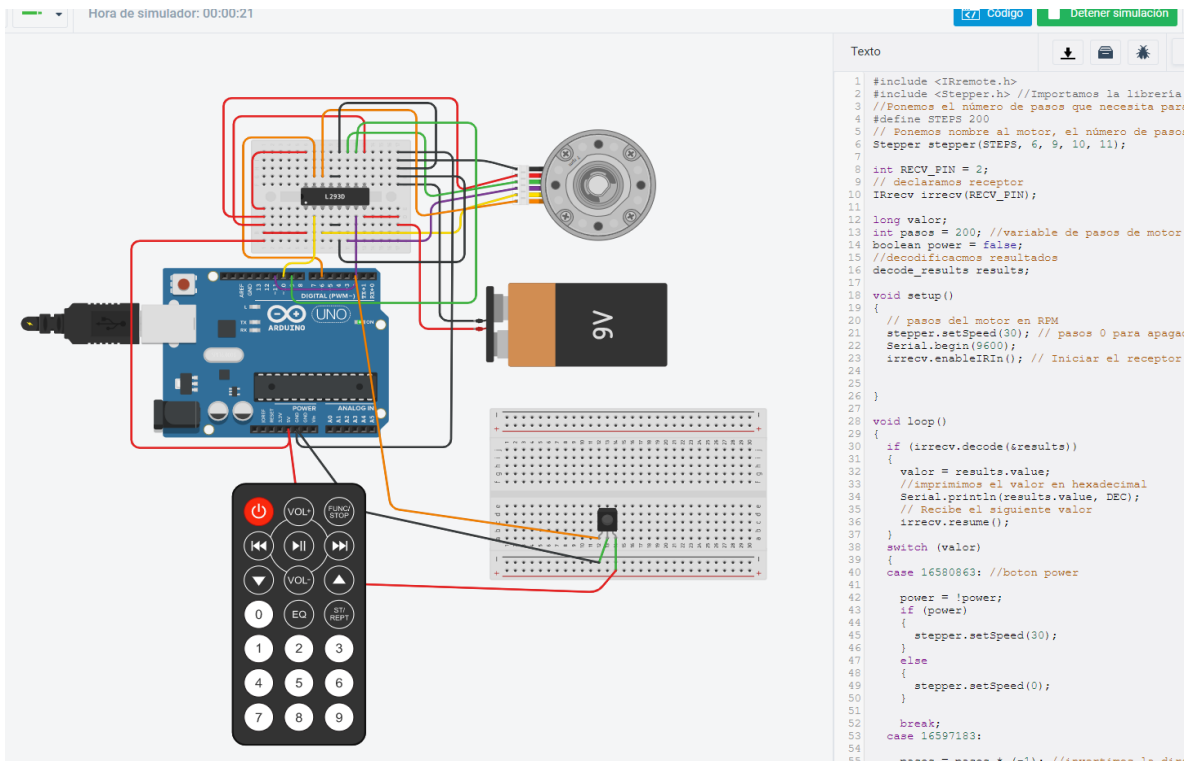
            break;
    }

    valor = 0; //limpiamos el valor del transitor y control remoto
    stepper.step(pasos); //asigna la pasos del motor
    delay(50); //Pequeña pausa
}
```


Simulaciones



Simulación del circuito del sensor infrarrojo en Protheus



Simulación de motor paso a paso en Tinkercad

Url: <https://www.tinkercad.com/things/cDiuPQFS4DG>

Estudiantes: Willy Marcos Chana Tito - Gilda Alcón Mauricio

Una vez simulado y que no nos mande errores procedemos a armar la maqueta para probar su funcionamiento e ir ajustando los sensores.

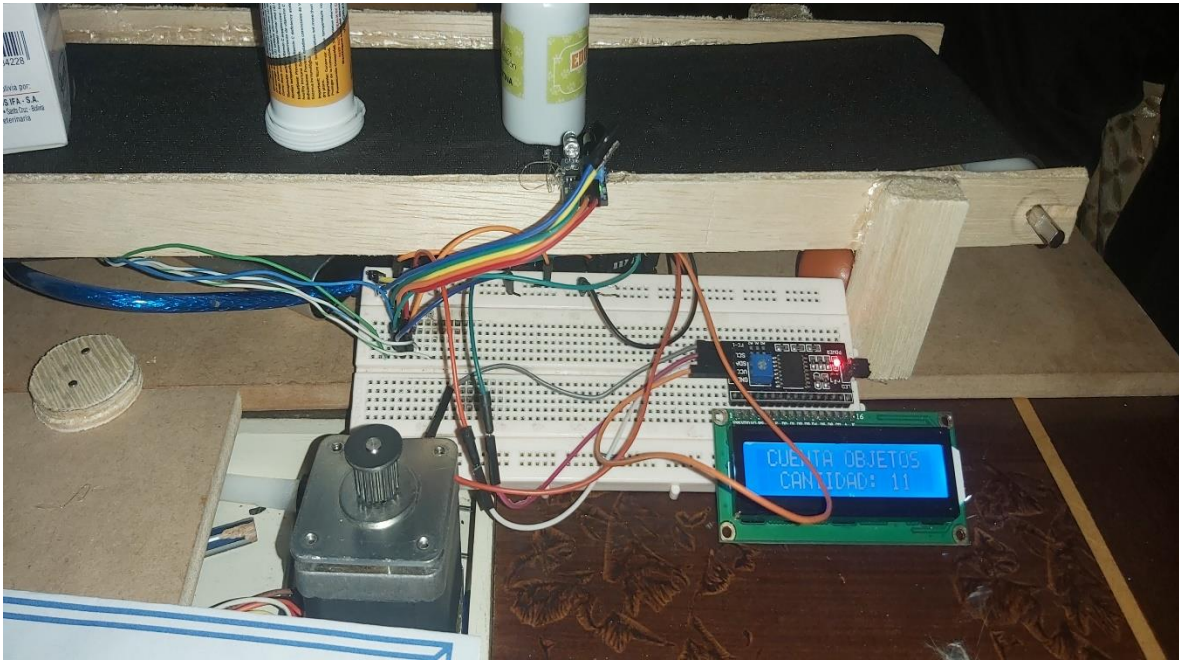
Para el funcionamiento usaremos una pequeña correa que unirá al eje del motor con el eje del rodillo de a su vez hará que la cinta transportadora se deslice transportando así los objetos sobre ella, como vemos en la imagen a continuación.



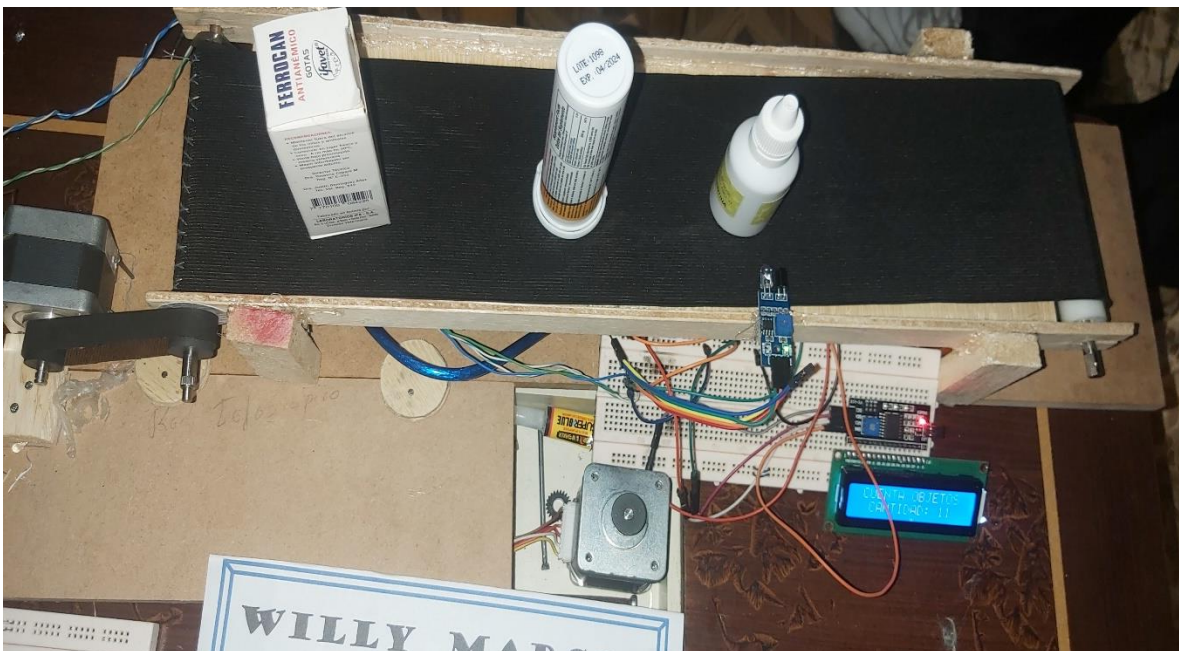
Fotografías de Funcionamiento e implementación



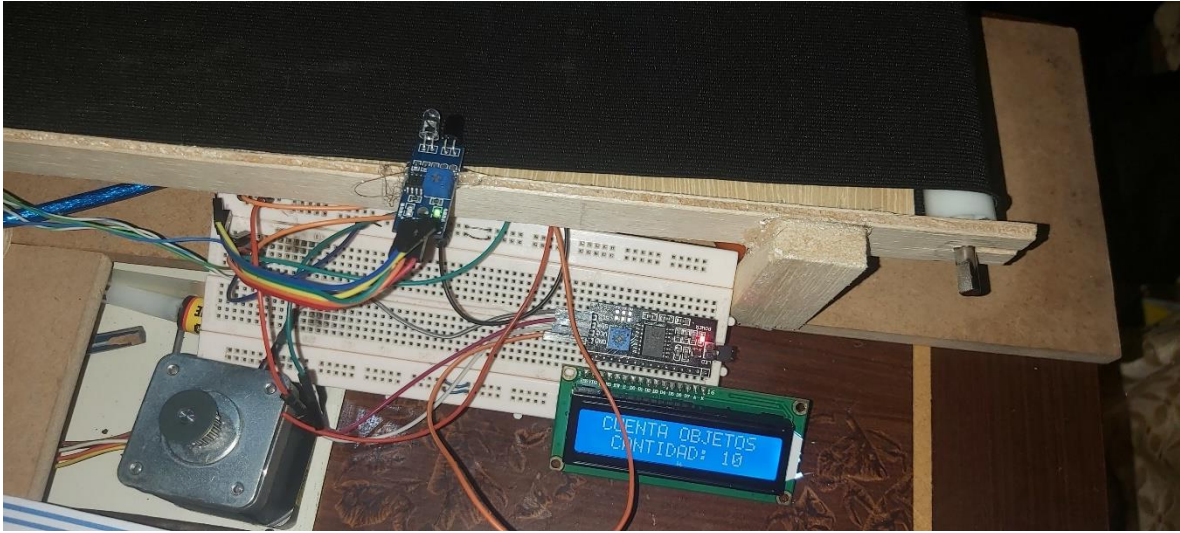
Estudiantes: Willy Marcos Chana Tito - Gilda Alcón Mauricio



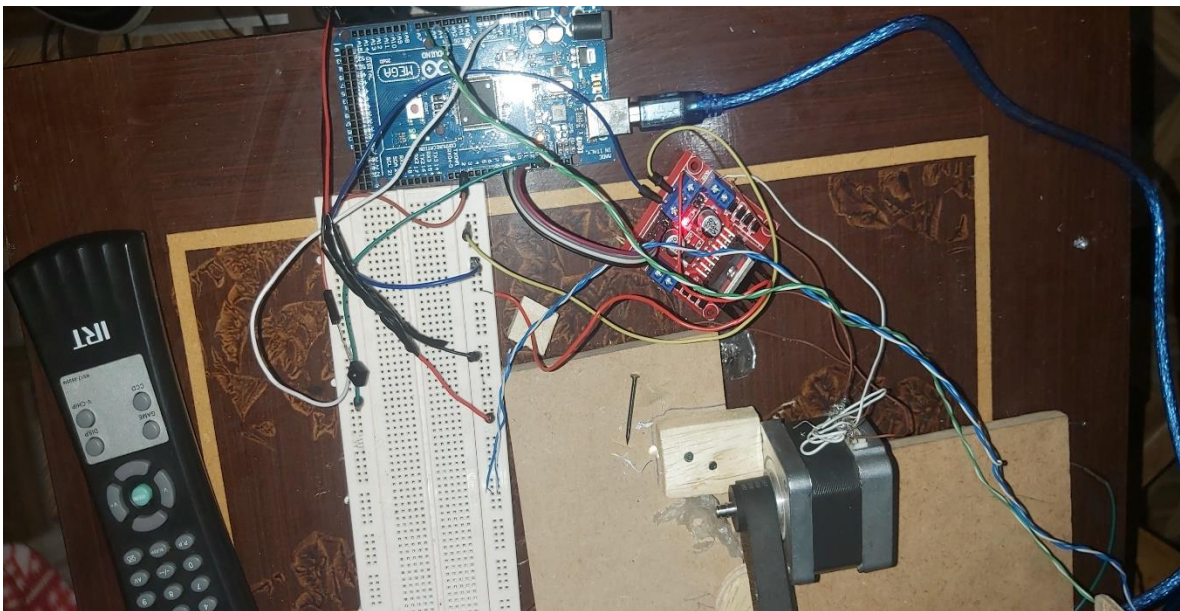
Componentes y materiales Utilizados



Estudiantes: Willy Marcos Chana Tito - Gilda Alcón Mauricio



Sensor infrarrojo, I2C y LCD utilizados



Motor paso a paso, L298N, Arduino, fototransistor y control remoto Utilizados

Estudiantes: Willy Marcos Chana Tito - Gilda Alcón Mauricio

APLICACIONES DE LAS CINTAS TRANSPORTADORAS

Las cintas transportadoras se usan principalmente para:

- Se utilizan en cualquier industria que requiera mover productos o materiales de manera masiva y rápida, como en las fabricas de botellas, enlatados, productos envasados y encajonados, etc.
- Transportar materiales granulados,
- Transportar materiales agrícolas e industriales, tales como cereales, carbón, minerales, etcétera,
- También se pueden usar para transportar personas en recintos cerrados (por ejemplo, en grandes hospitales y ciudades sanitarias).
- A menudo para cargar o descargar buques cargueros o camiones. Para transportar material por terreno inclinado se usan unas secciones llamadas cintas elevadoras.



CONCLUSIONES Y OBSERVACIONES

Conclusión:

El desarrollo de este proyecto nos permitió aplicar todo lo aprendido hasta el momento en el curso de Robótica Industrial, lo cual nos permite establecer criterios de diseño, simulación e implementación de proyectos con Arduino y las diferentes funcionalidades que nos ofrece, ya que cuenta con una multitud de librerías que facilitan la prototipación de proyectos que pueden ser implementados a gran escala en las industrias y otros.

Observaciones:

Una de las Dificultades que tuvimos fue que para la simulación en Protheus no existe una opción para agregar la simulación del fototransistor infrarrojo con el control remoto, por este motivo terminamos por optar simularlo en Tinkercad, pero en Tinkercad no pudimos encontrar una manera de simular el sensor infrarrojo que estuvimos utilizando, ya que este no se encuentra en este simulador, por lo cual optamos por simularlo de manera separada en ambos simuladores, ya que esto no afectaría las simulaciones porque son prácticamente independientes.