

Fecha: 23/11/2020

Tecnológico de Estudio Superiores Chalco

Ing. Sistemas Computacionales

Practica: 7

Profesor:

Galicia Moysen Alfredo

Equipo: 3

Integrantes:

- **Aguirre Velázquez Luis Raymundo**
- **Espinoza Sánchez Daniel Antonio**
- **Medina García José**
- **Olivares Vargas Luis Alberto**
- **Soriano López Alberto**

Grupo: 4701

Ing. Sistemas Computacionales

Carrera	Plan de estudio	Clave de asignatura	Nombre de asignatura
Ing. Sistemas Computacionales		SCC-1023	Sistemas programables

No. De practica	Laboratorio	Salón de clase	Duración (hora)
7	Banda Transportadora	4701	2:00 HRS

1	Introducción
	Se nos ha solicitado diseñar una banda transportadora que realice el traslado de un laboratorio a otro las muestras para ello tengan en cuenta dos motores que giran en ambos sentidos izquierda y derecha y dos sensores que hacen inicio y fin de carrera, buscar información para la construcción de la mejor banda transportadora.

2	Objetivo
	<ul style="list-style-type: none"> Diseñar procesos de automatización y control por medio de placas de desarrollo de Hardware Libre y uso de software de programación de alto nivel. Utiliza placas de desarrollo de Hardware Libre como Arduino para la automatización y control de procesos propuestos en el proyecto del curso.

3	Código
	<pre>#include <LiquidCrystal.h> LiquidCrystal lcd(12,11,5,4,3,2); const int pinmotor1=7; const int pinmotor2=6; const int pinBtn1=10; const int pinBtn2=9; const int pinSensor1=13; const int pinSensor2=8; int estadoOn=0; int estadoOFF=0; int estadoOn_LAST=0; int estadoOFF_LAST=0; int sensor1=0; int sensor2=0; int llego1=0; int llego2=0; void setup(){ lcd.begin(16,2); lcd.print("Fase 4 Grupo 14"); lcd.setCursor(0,1); lcd.print("Circuito Iniciado"); pinMode(pinBtn1, INPUT); pinMode(pinBtn2, INPUT); pinMode(pinmotor1, OUTPUT); pinMode(pinmotor2, OUTPUT); pinMode(pinSensor1, INPUT);</pre>

```

pinMode(pinSensor2, INPUT);
}

void loop(){
  estadoOn=digitalRead(pinBtn1);
  estadoOFF=digitalRead(pinBtn2);
  sensor1=digitalRead(pinSensor1);
  sensor2=digitalRead(pinSensor2);

  //BOTON LAB 1
  if(estadoOn != estadoOn_LAST){
    if(estadoOn==HIGH){
      if(sensor1==1 && sensor2==0){
        digitalWrite(pinmotor1, HIGH);
        digitalWrite(pinmotor2, LOW);
        lcd.clear();
        lcd.print("Corriendo hacia");
        lcd.setCursor(0,1);
        lcd.print("la Derecha");
        llego1=1;
      }
      else
      if(sensor1==0){
        lcd.clear();
        lcd.print("Poner Muestra");
        lcd.setCursor(0,1);
        lcd.print("en la banda LAB1");
      }
      else{
        digitalWrite(pinmotor1, LOW);
        digitalWrite(pinmotor2, LOW);
        lcd.clear();
        lcd.print("Retirar Muestra");
        lcd.setCursor(0,1);
        lcd.print("de LAB 2");
      }
    }
    estadoOn_LAST = estadoOn;
  }

  if(sensor2 && llego1==1){ //FIN DE CARRERA
    digitalWrite(pinmotor2, LOW);
    digitalWrite(pinmotor1, LOW);
    lcd.clear();
    lcd.print("La Muestra llego");
    lcd.setCursor(0,1);
    lcd.print("al LAB 2");
    llego1=0;
  }

  //BOTON LAB 2
  if(estadoOFF != estadoOFF_LAST){
    if(estadoOFF==HIGH){

```

```

if(sensor1==0 && sensor2==1){
  digitalWrite(pinmotor1, LOW);
  digitalWrite(pinmotor2, HIGH);
  lcd.clear();
  lcd.print("Corriendo hacia");
  lcd.setCursor(0,1);
  lcd.print("la Izquierda");
  llega2=1;
}
else
if(sensor2==0){
  lcd.clear();
  lcd.print("Poner Muestra");
  lcd.setCursor(0,1);
  lcd.print("en la banda LAB2");
}
else{
  digitalWrite(pinmotor1, LOW);
  digitalWrite(pinmotor2, LOW);
  lcd.clear();
  lcd.print("Retirar Muestra");
  lcd.setCursor(0,1);
  lcd.print("de LAB 1");
}
}
estadoOFF_LAST = estadoOFF;
}
if(sensor1 && llega2==1){ //FIN DE CARRERA
  digitalWrite(pinmotor1, LOW);
  digitalWrite(pinmotor2, LOW);
  lcd.clear();
  lcd.print("La Muestra llegó");
  lcd.setCursor(0,1);
  lcd.print("al LAB 1");
  llega2=0;
}
}

```

4 Procedimiento (Descripción)

Equipo Necesario

- Computadora o laptop.
- SO: Windows 8 o superior.
- Software Proteus 8.
- Software Arduino 1.8.

Material de Apoyo

- Navegador de su agrado.
- Video tutorial.

A Desarrollo de la practica

1. Realizar la configuración del código:

- Tendremos que abrir el programa “Arduino”, posteriormente asignaremos los atributos necesarios para el correcto funcionamiento del sensor.

- Una vez declarados dichos atributos procederemos a llamarlos y a su vez agregaremos la declaración de las variables.
- Por ultimo realizamos la configuración y programado sobre las acciones que realizaran dichos motores.

2. Realizar el modelado en Proteus:

- Primero ejecutaremos el software; posteriormente le daremos “click” en el apartado de “esquemas”, ahí procederemos agregar los siguientes competes: dos Logicstate, un L293D, un LM016L, dos resistencias (10k), dos Motores, cinco tierras, cuatro power, dos push button, catorce outputs, catorce inputs y un Simulino Uno.

Nota: Los outputs e inputs deberían contener sus respectivas etiquetas referentes a cada componente y/o acción a realizar.

- A continuación, procederemos a colocar cada uno de los componentes con una cierta distancia entre cada uno, esto con el fin de simplificar los enlaces (conexiones).

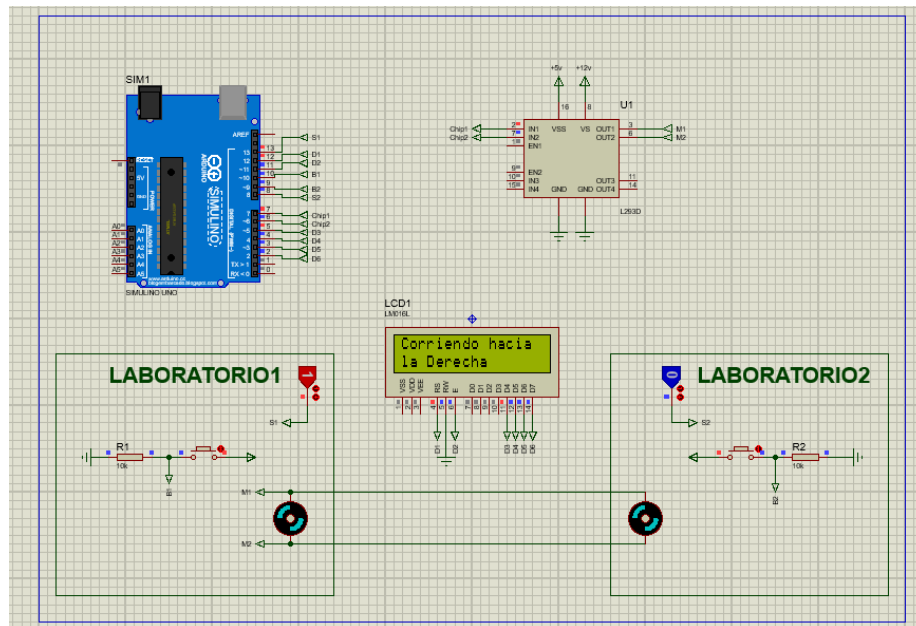
Nota: Se anexará una captura sobre el posible acomodo de cada componente.

- Por último, daremos doble click en simulino SDM, buscaremos el apartado “program file” y ahí pegaremos la dirección. HEX que copiamos del software Arduino.

3. Ejecución del programa:

- Por ultimo nos posicionaremos en la parte inferior derecha del programa, ahí nos encontraremos con un icono de “Play” lo presionamos; con eso ya estaríamos ejecutando el programa donde podremos observar su funcionamiento.

B Cálculos y reporte



Cuando la luminosidad es menor a 50% las bandas giran a la izquierda y cuando la luminosidad es igual o mayor a 50% las bandas giran a la derecha.

5 Resultado y conclusiones

Como se puede observar en la captura de pantalla, dicho programa cumple con los parámetros establecidos al inicio de la práctica, demostrando su correcto funcionamiento a través del simulador "Proteus". A si mismo se puede observar los componentes necesarios que dicho software solicita para poder ejecutar las acciones programadas.

Conclusiones:

- Aguirre Velázquez Luis Raymundo: Hay mil maneras de poner a trabajar Proteus para simulaciones, en este caso la simulación nos permite visualizar como sería el comportamiento en el traslado de una banda transportadora de un área de trabajo a otra y esto lo establecimos por tiempos dentro del código, los materiales si bien, no son idénticos a los físicos, bien funcionan para darnos una perspectiva real del comportamiento del código y el ensamble de los mismos.
- Espinosa Sánchez Daniel Antonio: La actividad permitió conocer más a fondo el microcontrolador Arduino, así como su funcionamiento y los periféricos que lo complementan, dándole solución a través de la utilización de softwares importantes como Arduino IDE donde se configura y compila el microcontrolador y Proteus como simulador de los circuitos, brindándole al usuario una perspectiva más exacta y grafica sobre el funcionamiento del sistema.
- Medina García José: Una de las mejores herramientas para la simulación de proyectos de menor a mayor escala es Proteus ya que con esta herramienta podemos desarrollar de manera virtual una gran variedad de proyectos desde el más fácil hasta el más complejo y con ello nos podemos dar cuenta en los posibles errores que se pueden presentar a corto o a mayor plazo para evitar fallos al momento de hacer el ensamblamiento en físico.
- Olivares Vargas Luis Alberto: Este sistema de transporte por su eficiencia es uno de los pilares de la industria por muchos factores. Las bandas y rodillos transportadores han aportado una gran parte en el desarrollo de la industria de mediana y gran escala reduciendo los costos de producción. Otro aspecto importante a mencionar es que este tipo de maquinaria ha hecho que la industria cree productos con una mayor calidad y a un corto período de tiempo.
- Soriano López Alberto: En las prácticas que uno puede realizar en Arduino y para ver si están correctas hacerlas en Proteus nos pueden ayudar a darnos cuenta cómo es en realidad que está funcionando nuestro circuito en caso de que no tengamos los materiales en físico para poder simular lo esto también nos puede ayudar que Proteus es un software en el cual se puede simular la función de un código en diferentes microcontroladores y con el poder darnos cuenta enciende algún momento en el que empezamos a realizar nuestro código no llegase a compilar y tenga la función que nosotros necesitamos.