

Fecha: 19/10/2020

Tecnológico de Estudio Superiores Chalco

Ing. Sistemas Computacionales

Practica: 6

Profesor:

Galicia Moysen Alfredo

Equipo: 3

Alumno:

- **Espinoza Sánchez Daniel Antonio**

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)
6	Potenciómetro	4701	2:00 HRS

1	Introducción
	<p>Se nos ha solicitado generar un ensamble de manera virtual en el cual debe ser visible el proceso donde gira un motor en ambos sentidos implementando un potenciómetro y haciendo uso de 3 botones, los cuales controlaran la rotación de dicho motor. Esto nos permitirá dar un punto de vista técnico de cómo funcionan los motores y los potenciómetros con respecto a la materia de sistemas programables y a la vez cómo funcionan en el mundo laboral.</p>

2	Objetivo
	<ul style="list-style-type: none"> Implementar nuevas herramientas para tener más conocimiento acerca de cómo funcionan la programación de dispositivos en la actualidad.

3	Código
	<pre> int enablePin = 11; int reloj = 4; int noreloj = 7; int botonD = 1; int botonI = 2; int botonP = 3; int Poten = A0; int estado1 = 0; int estado2 = 0; int estado3 = 0; void setup() { pinMode(enablePin, OUTPUT); pinMode(botonD, INPUT); pinMode(botonI, INPUT); pinMode(botonP, INPUT); pinMode(reloj, OUTPUT); pinMode(noreloj, OUTPUT); } void Derecha() { int velocidad = analogRead (Poten); velocidad = map (velocidad,0,1024,0,256); digitalWrite (reloj, HIGH); digitalWrite (noreloj, LOW); digitalWrite (enablePin, HIGH); analogWrite (enablePin, velocidad); delay(50); } void Izquierda() { int velocidad = analogRead (Poten); </pre>

```

velocidad = map (velocidad,0,1024,0,256);
digitalWrite (reloj, LOW);
digitalWrite (noreloj, HIGH);
digitalWrite (enablePin, HIGH);
analogWrite (enablePin, velocidad);
delay(50);
}

void Paro() {
digitalWrite (reloj, LOW);
digitalWrite (noreloj, LOW);
digitalWrite (enablePin, LOW);
}
void loop() {
estado1 = digitalRead (botonD);
if (estado1 == HIGH)
{
  Derecha();
}
estado2 = digitalRead (botonI);
if (estado2 == HIGH)
{
  Izquierda();
}
estado3 = digitalRead (botonP);
if (estado3 == HIGH)
{
  Paro();
}
}

```

4	Procedimiento (Descripción)
<p style="text-align: center;">Equipo Necesario</p> <ul style="list-style-type: none"> • Computadora o laptop. • SO: Windows 8 o superior. • Software Proteus 8. • Software Arduino 1.8. 	<p style="text-align: center;">Material de Apoyo</p> <ul style="list-style-type: none"> • Navegador de su agrado.

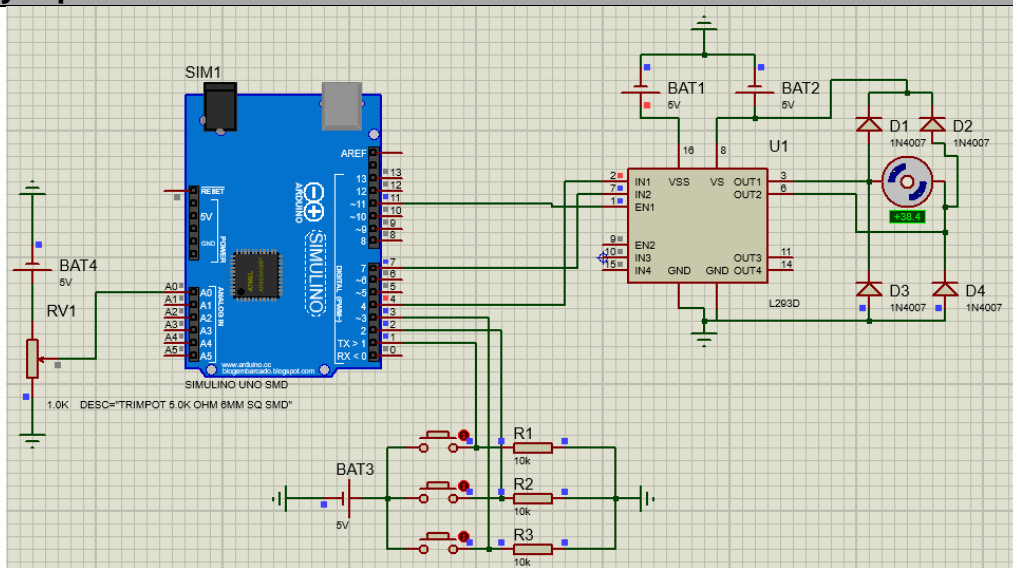
A	Desarrollo de la practica
<p>1. Realizar la configuración del código:</p> <ul style="list-style-type: none"> ➤ Tendremos que abrir el programa “Arduino”, posteriormente asignaremos los atributos necesarios para el correcto funcionamiento del sensor. ➤ Una vez declarados dichos atributos procederemos a llamarlos y a su vez agregaremos la declaración de las variables. ➤ Por ultimo realizamos la configuración y programado sobre las acciones que realizara dicho motor. <p>2. Realizar el modelado en Proteus:</p>	

- Primero ejecutaremos el software; posteriormente le daremos “click” en el apartado de “esquemas”, ahí procederemos agregar los siguientes compotes: cuatro CELL, un L293D, un Motor, seis tierras, tres resistencias, tres botones, un potenciómetro, cuatro diodos y un Simulino SMD.
- 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).
- 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



Respecto a esta práctica, no ha sido necesario realizar algún agregado o modificación al mismo, dado que la misma está bien desarrollada y complementada.

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.

Conclusión:

Dicha práctica nos ayuda a familiarizarnos con el software Arduino, aprendiendo a configurar cada uno de los apartados y permitiéndonos desarrollar nuevas acciones para una misma problemática. Por ultimo cabe agregar que cada vez es más entendible el lenguaje y configuración de los diferentes parámetros y complementos.