

Fecha: 09/12/2020

Tecnológico de Estudio Superiores Chalco

Ing. Sistemas Computacionales

Practica: 8

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)
8	Maquina Centrifugadora	4701	2:00 HRS

1 | Introducción

Se nos ha solicitado diseñar una maquina centrifuga que incluya una perilla para ajustar el tiempo de funcionamiento del rotor, el tiempo mínimo es de 10 segundos y el tiempo máximo es de 1 minuto, por lo que esta perilla se puede ajustar con un potenciómetro.

2 | Objetivo

- 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

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12,11,5,4,3,2);

//Katherin Valega Martinez

const int pinBtn=13; //PULSADOR

const int pinSpeed=A1; //potenciometros
const int pinTime=A0;

const int pinmotor7=10; // PWM
const int pinmotor2=9;

const int LedOn=7; // LED
const int LedW=6;

int estado=0; // Variables
int contador=0;
int tiempo_escogido=10; //10seg.
int velocidad_escog=0;
int pwm;
int velocidad_nueva;
int fin = 0;

void setup() {

// Mensajes de Inicio
```

```

lcd.begin(16,2);
lcd.print("Fase 6. Grupo 14");
lcd.setCursor(0,1);
lcd.print("Proyecto Final");
delay(1000);

lcd.clear();
lcd.print("Confi. perillas");
lcd.setCursor(0,1);
lcd.print("y pulse Encender");
delay(1000);

pinMode(pinBtn, INPUT); // Default
pinMode(pinmotor7, OUTPUT);
pinMode(pinmotor2, OUTPUT);
pinMode(LedOn, OUTPUT);
pinMode(LedW, OUTPUT);

}

void loop() {

estado=digitalRead(pinBtn); // pulsador
digitalWrite(LedOn, HIGH); //led Conectado

// DISPLAY visualizar las perillas

tiempo_escogido =
(analogRead(pinTime)/20.2)+10; // formula de
proporcion
velocidad_escog =
(analogRead(pinSpeed)/25.57);
lcd.clear();
lcd.print("Time:
"+String(tiempo_escogido)+"seg.");
lcd.setCursor(0,1);
lcd.print("Speed:"+String(velocidad_escog)+"x100
RPM");
delay(100);

//ENCENDIDO
if(estado == HIGH ){

digitalWrite(LedW, HIGH); //led funcionando

while (contador <= tiempo_escogido){

    velocidad_escog =
(analogRead(pinSpeed)/25.57);
    velocidad_nueva = analogRead(pinSpeed);

```

```

    pwm = map(velocidad_nueva, 0, 1023, 0,
255); // invertir rango
    analogWrite(pinmotor7, pwm); // envia la
velocidad del motor

    lcd.clear();

    lcd.print("Time:"+String(tiempo_escogido)+"seg. -
>"+String(contador));
    lcd.setCursor(0,1);

    lcd.print("Speed:"+String(velocidad_escog)+"x100
RPM");

    delay(200); // delay(1000);

    contador++;
    fin = 1;
}
}

// AL FINALIZAR
if(fin == 1){
    digitalWrite(LedW, LOW);
    analogWrite(pinmotor2, LOW);
    analogWrite(pinmotor7, LOW);
    lcd.clear();
    lcd.print("Fin del Proceso");
    lcd.setCursor(0,1);
    lcd.print("Tome la muestra");
    delay(2000);
    fin = 0;
}
}

```

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

A	Desarrollo de la practica
	<ol style="list-style-type: none"> 1. Realizar la configuración del código: <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 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 compotes: un L293D, un LM016L, tres resistencias (una de 1k y dos de 10k), un Motor, tres tierras, cuatro power, un push button, veintiún outputs, nueve inputs, un led red, un led Green, un pot-hg 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).

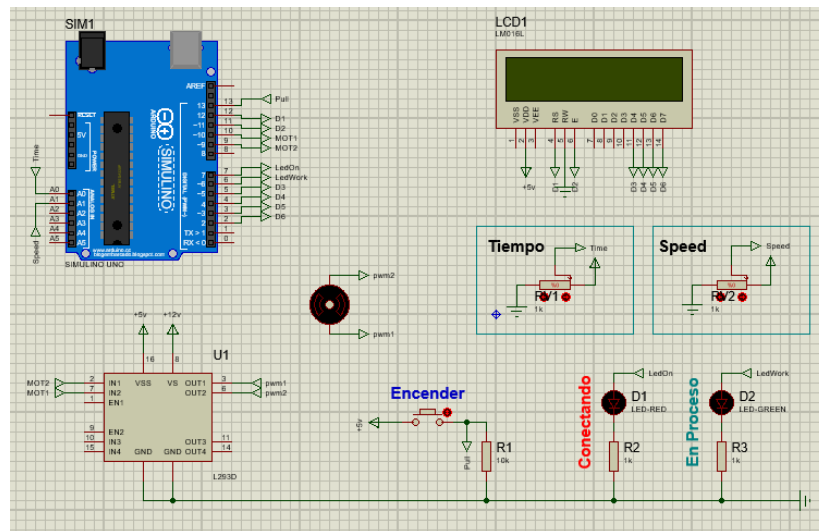
Nota: Se anexará una captura sobre el posible acomodo de cada componente.

- 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



Puedes hacer uso de los apartados “Tiempo” y “Speed” respectivamente, para tener el control de la velocidad por segundo del rotor de la centrifugadora.

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: En esta practica se puede apreciar como una maquina centrifugadora pone en rotación una muestra para separar aplicando fuerza sus componentes en función a su densidad, lo cual le permite realizar su tarea principal dando una buena experiencia y satisfacción al usuario que la utilice.
- 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.