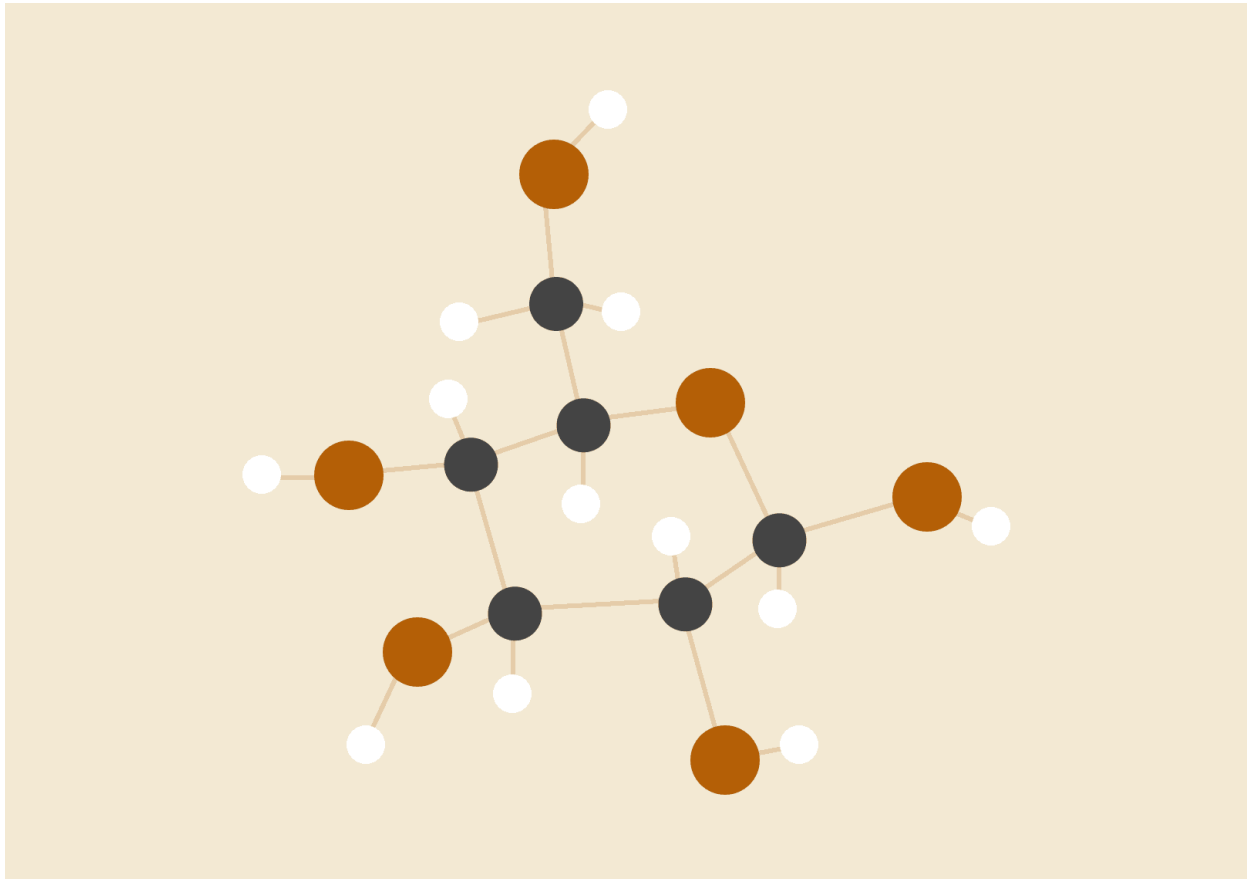


TRABAJO DE LABORATORIO DE ELECTRÓNICA

Trabajo Práctico N°3
Electrónica Digital IV.7



Valentín Fuentes

25/10/2021
7ºmo Año de Electrónica

Actividad;	2
Captura de pantalla del dispenser de alcohol;	3
Primer código con goto;	3
Código Final;	6
Codigo que me paso Laureano;	9

Actividad;

Dispenser de alcohol para manos.

Realizar el programa para automatizar el funcionamiento de un dispositivo que dispensa alcohol en

forma de spray para la sanitización de las manos del público en grandes superficies.

El equipo deberá aceptar recipientes de alcohol de diversos tamaños. El encargado deberá introducir

por teclado la cantidad de dosis que contiene cada recipiente.

Cada vez que una persona aproxima las manos a menos de 10 cm del pico rociador, las mismas serán

detectadas por un sensor ultrasónico; esta situación hará que el sistema accione un servomotor que

mediante un giro de 180 grados producirá el gatillado del spray.

La cantidad de dosis se deberá decrementar en una unidad con cada disparo.

Cuando la cantidad de dosis baje al 10 % de valor de dosis introducidas por el teclado, el sistema deberá

dar alarma encendiendo un led que se mantendrá en ese estado mientras se cumpla la condición

mencionada.

En la primera energización del sistema, arrancará por defecto con un recipiente de 10 dosis.

Si las dosis llegan a cero, el sistema no debe gatillar más.

Debe mostrar toda información útil o necesaria en un display LCD

Usar librería keypad.h

Respetar esta configuración de pines SIN EXCEPCIÓN:

Pins_Filas = 6, 5, 4, 3 Pines del Arduino para las filas del teclado matricial

Pins_Cols = 9, 8, 7 Pines del Arduino para las columnas del teclado matricial

#define trigPin 12 Pin para la orden de medición del ultrasónico

#define echoPin 13 Pin para la recepción del pulso proporcional a la distancia enviado por el ultrasónico

#define LedAlarma 11 Pin para el led de aviso de reposición de alcohol

Servo al pin 10 Pin para accionamiento del servomotor

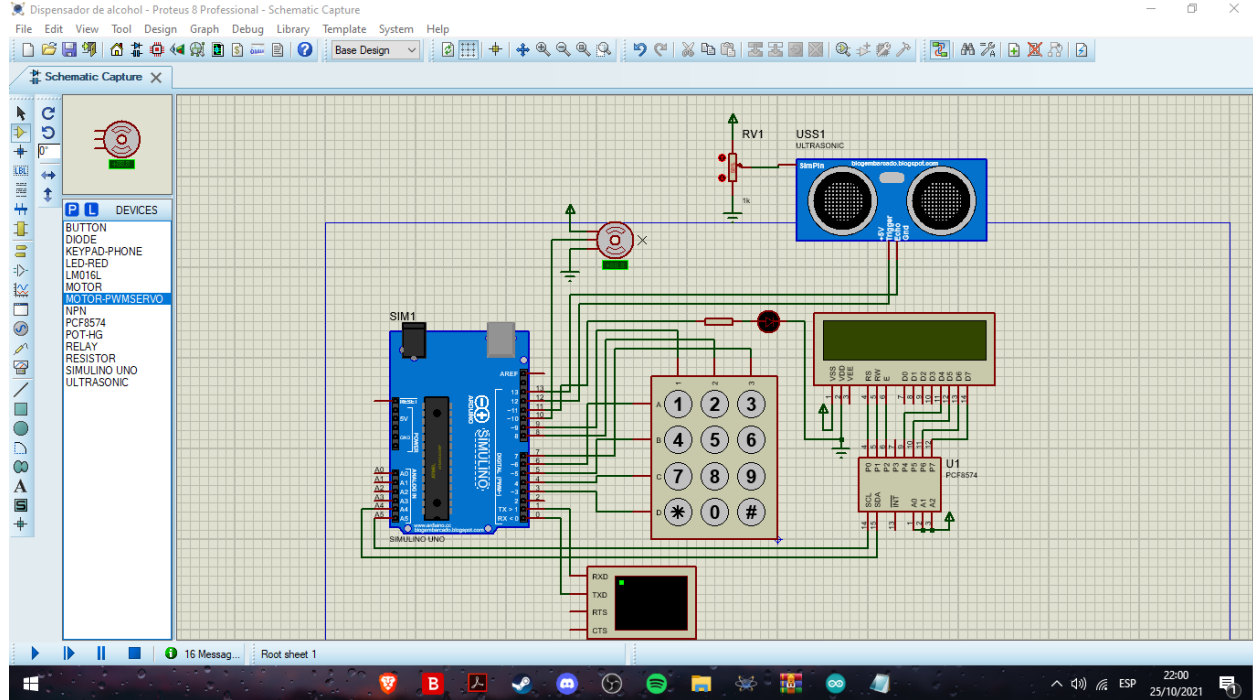
Desafío opcional:

Analizar, razonar y comentar qué mejoras se deberían hacer a este proyecto para que pueda pensarse

como algo posible de comercializar

*/

Captura de pantalla del dispenser de alcohol;



Primer código con go to;

```
#include <Servo.h> // Se incluye la librería para control de servo
#include <Keypad.h> // Librería para uso de teclado matricial
#include <LiquidCrystal_I2C.h>
#define I2C_ADDR 0x27
#define trigPin 12 // Pin para la orden de medición del ultrasónico
#define echoPin 13 // Pin para la recepción del pulso proporcional a la distancia
// enviado por el ultrasónico
#define Alarma 11
int num=0; // Variable donde se irá conformando el número introducido por teclado de a
una cifra
int dosis = 10;
int a = 0;
int Porcentaje = 0;
int Paracent = 10;
const byte Filas = 4; //Cuatro filas
const byte Cols = 3; //Cuatro columnas
byte Pins_Filas[] = {6, 5, 4, 3}; // Pines del Arduino para las filas
byte Pins_Cols[] = {9, 8, 7}; // Pines del Arduino para las columnas
char Teclas [ Filas ][ Cols ] =
{
  {'1','2','3'},
  {'4','5','6'},
```

```

    {'7','8','9'},
    {'*','0','#'}
  };
Keypad Teclado1 = Keypad(makeKeymap(Teclas), Pins_Filas, Pins_Cols, Filas, Cols);
Servo servoMotor; // Declaramos la variable para controlar el servo
LiquidCrystal_I2C lcd(I2C_ADDR,2,1,0,4,5,6,7);
void setup()
{
  lcd.begin(16,2); // Inicializar el display con 16 caracteres 2 lineas en este ejemplo
  lcd.setBacklightPin(3,POSITIVE); //Habilita la luz de fondo
  //lcd.setBacklight(1); // Enciende luz de fondo (Proteus no lo simula, en la placa real funciona)
  pinMode(Alarma, OUTPUT);
  Serial.begin(9600); // Iniciamos el monitor serie para mostrar el resultado
  servoMotor.attach(10); // Iniciamos el servo para que trabaje con el pin 10
  servoMotor.write(0); // Inicializamos en ángulo 0 el servomotor
  delay(1000); // Hacemos una pausa de 1000ms
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  digitalWrite(trigPin, LOW); // Nos aseguramos de que el trigger está desactivado
  delayMicroseconds(2); // Para estar seguros de que el trigger esta LOW
}
void loop()
{
  loop_2:
  long duracion, distancia ; // Almacena el tiempo del escalón que envía el HC en microsegundos
  digitalWrite(trigPin, HIGH); // Activamos el pulso de salida
  delayMicroseconds(10); // Esperamos 10µs. El pulso sigue activo este tiempo
  digitalWrite(trigPin, LOW); // Cortamos el pulso y a esperar el echo
  duracion = pulseIn(echoPin, HIGH) ; // la sentencia pulseIn mide el tiempo del escalón enviado por el HC en microsegundos
  distancia = duracion * 10 / 292 / 2 ; // Ver el porqué de esta fórmula en el material para alumnos
  Serial.println((String)"Distancia: "+distancia+"cm");

  lcd.setCursor(0,0);
  lcd.print((String)"Distancia:"+distancia+" "+"cm ");
  lcd.setCursor(0,1);
  lcd.print((String)"Cantidaddosis:"+dosis+" ");

  Porcentaje = Paracent*10/100;
  Serial.println (Porcentaje);
  Serial.println (dosis);
  if (dosis<=Porcentaje){
    digitalWrite(Alarma, HIGH);
  }
  if(dosis != 0 && distancia <=10){
    servoMotor.write(0); // Inicializamos en ángulo 0 el servomotor
  }
}

```

```

Serial.println ("Servo a 0 grados");
delay (2000); //Demora importante, debe ser suficientemente larga para que el servo llegue a
girar los 180 grados, en un servo real, pueden ser unos 2 segundos
servoMotor.write(180); // Desplazamos 180 grados
Serial.println ("Servo en 180 grados");
delay (2000); //Demora importante, debe ser suficientemente larga para que el servo llegue a
girar los 180 grados, en un servo real, pueden ser unos 2 segundos
dosis = dosis - 1;
}
Serial.println((String)"Dosis: "+ dosis);
delay (100);

```

```

char a = Teclado1.getKey() ; // En la pulsación se guardará el carácter oprimido
if (a != 0){
    digitalWrite(Alarma, LOW);
    goto loop_1;}
else {goto loop_2;}
loop_1:
char pulsacion = Teclado1.getKey() ; // La pulsación se guardará el carácter oprimido
if (pulsacion != 0){

    if (pulsacion == '1'){
        num=num*10+1 ;}
    if (pulsacion == '2'){
        num=num*10+2 ;}
    if (pulsacion == '3'){
        num=num*10+3 ;}
    if (pulsacion == '4'){
        num=num*10+4 ;}
    if (pulsacion == '5'){
        num=num*10+5 ;}
    if (pulsacion == '6'){
        num=num*10+6 ;}
    if (pulsacion == '7'){
        num=num*10+7 ;}
    if (pulsacion == '8'){
        num=num*10+8 ;}
    if (pulsacion == '9'){
        num=num*10+9 ;}
    if (pulsacion == '0'){
        num=num*10+0 ;}

```

```

Serial.println(num); //Muestra el número que se va conformando
Serial.print("Numero final: "); //Muestra el número final
Serial.println(num);
lcd.setCursor(0,0);
lcd.print((String)"Modo Seteo:"+" ");
lcd.setCursor(0,1);
lcd.print((String)"Nuevas dosis:"+num+" ");

```

```

if (pulsacion == '#'){ // Si se pulsa # pone a cero num y queda listo pasando al loop principal
  dosis = num;
  Paracent = num;
  num = 0;
  goto loop_2;
}
goto loop_1;
}
goto loop_1;
}

```

Código Final;

```

#include <Servo.h> // Se incluye la librería para control de servo, que viene por defecto con
el arduino.
#include <Keypad.h> // Librería para uso de teclado matricial
#include <LiquidCrystal_I2C.h>
#define I2C_ADDR 0x27 //Pin para el led de aviso de reposición de alcohol
#define trigPin 12 // Pin para la orden de medición del ultrasónico
#define echoPin 13 // Pin para la recepción del pulso proporcional a la distancia
enviado por el ultrasónico
#define Alarma 11
int num=0; // Variable donde se irá conformando el número introducido por teclado de a
una cifra
int dosis = 10;
int a = 0;
int s = 0;
int Porcentaje = 0;
int Paracent = 10;
const byte Filas = 4; //Cuatro filas
const byte Cols = 3; //Cuatro columnas
byte Pins_Filas[] = {6, 5, 4, 3}; // Pines del Arduino para las filas
byte Pins_Cols[] = { 9, 8, 7}; // Pines del Arduino para las columnas
char Teclas [ Filas ][ Cols ] =
{
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
Keypad Teclado1 = Keypad(makeKeymap(Teclas), Pins_Filas, Pins_Cols, Filas, Cols);
Servo servoMotor; // Declaramos la variable para controlar el servo
LiquidCrystal_I2C lcd(I2C_ADDR,2,1,0,4,5,6,7);
void setup()
{
  lcd.begin(16,2); // Inicializar el display con 16 caracteres 2 lineas en este ejemplo
  lcd.setBacklightPin(3,POSITIVE); //Habilita la luz de fondo
  //lcd.setBacklight(1); // Enciende luz de fondo (Proteus no lo simula, en la placa real
funciona)

```

```

pinMode (Alarma, OUTPUT);
Serial.begin(9600); // Iniciamos el monitor serie para mostrar el resultado
servoMotor.attach(10); // Iniciamos el servo para que trabaje con el pin 10
servoMotor.write(0); // Inicializamos en ángulo 0 el servomotor
delay(1000); // Hacemos una pausa de 1000ms
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
digitalWrite(trigPin, LOW); // Nos aseguramos de que el trigger está desactivado
delayMicroseconds(2); // Para estar seguros de que el trigger esta LOW
}
void loop()
{
  if (s == 0){
    //loop_2:
    long duracion, distancia ; // Almacena el tiempo del escalón que envía el HC en
    microsegundos
    digitalWrite(trigPin, HIGH); // Activamos el pulso de salida
    delayMicroseconds(10); // Esperamos 10µs. El pulso sigue active este tiempo
    digitalWrite(trigPin, LOW); // Cortamos el pulso y a esperar el echo
    duracion = pulseIn(echoPin, HIGH) ; // La sentencia pulseIn mide el tiempo del escalón
    enviado por el HC en microsegundos
    distancia = duracion * 10 / 292 / 2 ; // Ver el porqué de esta fórmula en el material para
    alumnos
    Serial.println((String)"Distancia: "+distancia+"cm");

    lcd.setCursor(0,0);
    lcd.print((String)"Distancia:"+distancia+" "+"cm  ");
    lcd.setCursor(0,1);
    lcd.print((String)"Cantidaddosis:"+dosis+" ");

    Porcentaje = Paracent*10/100;
    if (dosis<=Porcentaje){digitalWrite(Alarma, HIGH);}
    if(dosis != 0 && distancia <=10){
      servoMotor.write(180); // Desplazamos 180 grados
      Serial.println ("Servo en 180 grados");
      delay (2000); //Demora importante, debe ser suficientemente larga para qu el servo llegue a
      girar los 180 grados, en un servo real,pueden ser unos 2 segundos
      servoMotor.write(0); // Inicializamos en ángulo 0 el servomotor
      Serial.println ("Servo a 0 grados");
      delay (2000); //Demora importante, debe ser suficientemente larga para qu el servo llegue a
      girar los 180 grados, en un servo real,pueden ser unos 2 segundos
      dosis = dosis - 1;
    }
    Serial.println((String)"Dosis: "+ dosis);
    delay (100);
  }
  char a = Teclado1.getKey() ; // en pulsacion se guardará el caracter oprimido
  if (a != 0 && s == 0){
    s = 1;
  }
}

```



```

digitalWrite(Alarma, LOW);
lcd.setCursor(0,0);
lcd.print((String)"Modo Seteo:"+"  ");
lcd.setCursor(0,1);
lcd.print((String)"Activado..."+"  ");
}
if( s == 1){
char pulsacion = Teclado1.getKey() ; // En la pulsacion se guardará el caracter oprimido
if (pulsacion != 0){
  delay (100); // Coloco un delay para que los if al ir rapido no se a salten ninguna pulsacion.
  if (pulsacion == '1'){
    num=num*10+1 ;}
  if (pulsacion == '2'){
    num=num*10+2 ;}
  if (pulsacion == '3'){
    num=num*10+3 ;}
  if (pulsacion == '4'){
    num=num*10+4 ;}
  if (pulsacion == '5'){
    num=num*10+5 ;}
  if (pulsacion == '6'){
    num=num*10+6 ;}
  if (pulsacion == '7'){
    num=num*10+7 ;}
  if (pulsacion == '8'){
    num=num*10+8 ;}
  if (pulsacion == '9'){
    num=num*10+9 ;}
  if (pulsacion == '0'){
    num=num*10+0 ;}

Serial.print("Numero final: "); //Muestra el número final
Serial.println(num);
lcd.setCursor(0,0);
lcd.print((String)"Modo Seteo:"+"  ");
lcd.setCursor(0,1);
lcd.print((String)"Nuevasdosis:"+num+"  ");
if (pulsacion == '#'){ // Si se pulsa # pone a cero num y queda listo pasar a la parte principal
  dosis = num;
  Paracent = num;
  num = 0;
  s = 0;
}
}
}
}
}

```