

7mo7ma

Electrónica Digital IV.

Trabajo Práctico N°4

Actividad

Control de una pecera

/*Realizar un sistema que automatice el funcionamiento de una pecera para peces de agua caliente.

El sistema debe controlar y mantener la temperatura en 24 y 26 grados, accionando para ello el calefactor sumergido de la pecera, mediante un relé.

El equipo deberá también mantener el aireador funcionando durante 12 horas al día.

A los fines de la simulación, realizar el cambio cada 6 segundos

Con la misma frecuencia de 12 horas, (6 segundos), también deberá accionar el sistema de alimentación de los peces.

El mismo funciona con un motor paso a paso que descarga la cantidad de alimento adecuada, de acuerdo a la cantidad de peces que contenga la pecera.

La cantidad de peces es un dato que es programado por el usuario a través de un pulsador. Este dato deberá ser guardado en la memoria eeprom cada vez que el usuario lo modifique, y leído cada vez que el sistema se energiza. De esta manera no será necesario configurar este valor cada vez que se interrumpe la energía eléctrica.

El sistema admite hasta 10 peces.

El motor paso a paso es unipolar y funcionará de a medio paso; con una vuelta completa dispensa alimento para un solo pez.

Toda la información de lo que va haciendo el sistema se deberá observar en un display LCD serial

Respetar las siguientes conexiones:

```
#define Pin1Motor 2 //Conexión para el motor PAP
#define Pin2Motor 3 //Conexión para el motor PAP
#define Pin3Motor 4 //Conexión para el motor PAP
#define Pin4Motor 5 //Conexión para el motor PAP
#define PulsadorSeteoCantidadPeces 6
#define SalidaCalefactor 7
#define SalidaAireador 8
*/
```

Respuesta

Código para Proteus;

```
#include <EEPROM.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define I2C_ADDR 0x27
LiquidCrystal_I2C lcd(I2C_ADDR,2,1,0,4,5,6,7);
#define LM35 A0
#define Pin1Motor 2 //Conexión para el motor PAP
#define Pin2Motor 3 //Conexión para el motor PAP
#define Pin3Motor 4 //Conexión para el motor PAP
#define Pin4Motor 5 //Conexión para el motor PAP
#define Pulsador 6
#define Calefactor 7
#define Aireador 8
unsigned long TiempoInicioCuenta = 0;
unsigned long ChequeoTiempo = 0;
unsigned long TiempoTranscurrido;
int contador = 0;
int ValorLM35 = 0;
float TempLM35 = 0;
bool Aireador_1;
int peces = 0;
int posicion;

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 placareal funciona)
```

```

pinMode (Pulsador, INPUT_PULLUP);
pinMode (Calefactor, OUTPUT);
pinMode (Aireador, OUTPUT);
TiempoInicioCuenta=millis();
Serial.begin (9600);

//cambiar la cantidad de peces
peces = EEPROM.read(posicion);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print ("Peces: ");
lcd.setCursor (9, 0);
lcd.print (peces);
pinMode (Pin1Motor,OUTPUT);//las cuatro salidas que comandan el motor
pinMode (Pin2Motor,OUTPUT);
pinMode (Pin3Motor,OUTPUT);
pinMode (Pin4Motor,OUTPUT);
}
void loop()
{
  Serial.println("TiempoInicioCuenta");
  Serial.println(TiempoInicioCuenta);
  Serial.println("CuentaPrevia");
  Serial.println(ChequeoTiempo);
  if (digitalRead(Pulsador)==0){          //Pulsador
    peces = peces + 1;
    EEPROM.write(posicion, peces);
    lcd.setCursor (9, 0);
    lcd.print (peces);
  }
  if (peces > 10){                          //LCD
    lcd.clear();
    peces = 0;
    lcd.setCursor(0, 0);
    lcd.print ("Peces: ");
    lcd.setCursor (9, 0);
    lcd.print (peces);
  }
  ChequeoTiempo=millis();                  //Tiempo del alimento
  TiempoTranscurrido=ChequeoTiempo-TiempoInicioCuenta;
  Serial.println("Tiempo");
  Serial.println(TiempoTranscurrido);

  if ( TiempoTranscurrido >= 15000){
    for (int i=0; peces > i; i++){
      digitalWrite(Pin1Motor, 1);
      digitalWrite(Pin2Motor, 0);
      digitalWrite(Pin3Motor, 0);

```

```

digitalWrite(Pin4Motor, 0);
delay (150);
digitalWrite(Pin1Motor, 1);
digitalWrite(Pin2Motor, 1);
digitalWrite(Pin3Motor, 0);
digitalWrite(Pin4Motor, 0);
delay (150);
digitalWrite(Pin1Motor, 0);
digitalWrite(Pin2Motor, 1);
digitalWrite(Pin3Motor, 0);
digitalWrite(Pin4Motor, 0);
delay (150);
digitalWrite(Pin1Motor, 0);
digitalWrite(Pin2Motor, 1);
digitalWrite(Pin3Motor, 1);
digitalWrite(Pin4Motor, 0);
delay (150);
digitalWrite(Pin1Motor, 0);
digitalWrite(Pin2Motor, 0);
digitalWrite(Pin3Motor, 1);
digitalWrite(Pin4Motor, 0);
delay (150);
digitalWrite(Pin1Motor, 0);
digitalWrite(Pin2Motor, 0);
digitalWrite(Pin3Motor, 1);
digitalWrite(Pin4Motor, 1);
delay (150);
digitalWrite(Pin1Motor, 0);
digitalWrite(Pin2Motor, 0);
digitalWrite(Pin3Motor, 0);
digitalWrite(Pin4Motor, 1);
delay (150);
digitalWrite(Pin1Motor, 1);
digitalWrite(Pin2Motor, 0);
digitalWrite(Pin3Motor, 0);
digitalWrite(Pin4Motor, 1);
delay (500);
}
    TiempolInicioCuenta=millis();
    Aireador_1=Aireador_1!=Aireador_1; //Aireador
    digitalWrite(Aireador, Aireador_1);
}
ValorLM35 = analogRead(LM35);          //Sensor LM35 Y Temperatura
TempLM35=(float) ((ValorLM35*5000.0)/1023.0)/10.0;
lcd.setCursor (0, 1);
lcd.print ("Temp: ");
lcd.setCursor (7, 1);
lcd.print(TempLM35);

```

```
delay (250);

if (TempLM35<=24){           //Calefactor
  digitalWrite(Calefactor, HIGH);
}
if (TempLM35>=26){
  digitalWrite (Calefactor, LOW);
}
}
```

Código para la simulación con componentes;

```
#include <EEPROM.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define I2C_ADDR 0x27
LiquidCrystal_I2C lcd(I2C_ADDR,2,1,0,4,5,6,7);
#include <Stepper.h>
const int stepsPerRevolution = 2048;
Stepper myStepper(stepsPerRevolution, 2, 3, 4, 5);
#define LM35 A0
#define Pulsador 6
#define Calefactor 7
#define Aireador 8
unsigned long TiempoInicioCuenta = 0;
unsigned long ChequeoTiempo = 0;
unsigned long TiempoTranscurrido;
int contador = 0;
int ValorLM35 = 0;
float TempLM35 = 0;
bool Aireador_1;
int peces = 0;
int posicion;
int motorSpeed;

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)
  //lcd.setBacklight(0); // Apaga luz de fondo (Proteus no lo simula, en la placa real funciona)
  pinMode (Pulsador, INPUT_PULLUP);
  pinMode (Calefactor, OUTPUT);
  pinMode (Aireador, OUTPUT);
```

```
// set the speed at 60 rpm:
myStepper.setSpeed(5);
TiempoInicioCuenta=millis();
Serial.begin (9600);

//cambiar la cantidad de peces
peces = EEPROM.read(posicion);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print ("Peces: ");
lcd.setCursor (9, 0);
lcd.print (peces);
}
void loop()
{
  Serial.println("TiempoInicioCuenta");
  Serial.println(TiempoInicioCuenta);
  Serial.println("CuentaPrevia");
  Serial.println(ChequeoTiempo);
  if (digitalRead(Pulsador)==0){          //Pulsador
    peces = peces + 1;
    delay (100);
    EEPROM.write(posicion, peces);
    lcd.setCursor (9, 0);
    lcd.print (peces);
  }
  if (peces > 10){                          //LCD
    lcd.clear();
    peces = 0;
    lcd.setCursor(0, 0);
    lcd.print ("Peces: ");
    lcd.setCursor (9, 0);
    lcd.print (peces);
  }
  ChequeoTiempo=millis();                  //Tiempo del alimento
  TiempoTranscurrido=ChequeoTiempo-TiempoInicioCuenta;
  Serial.println("Tiempo");
  Serial.println(TiempoTranscurrido);

  if ( TiempoTranscurrido >= 15000){
    for (int i=0; peces > i; i++){
      myStepper.step(2048);
    }
    TiempoInicioCuenta=millis();
    Aireador_1=Aireador_1!=Aireador_1; //Aireador
    digitalWrite(Aireador, Aireador_1);
  }
  ValorLM35 = analogRead(LM35);            //Sensor LM35 Y Temperatura
```

```

TempLM35=(float) ((ValorLM35*5000.0)/1023.0)/10.0;
lcd.setCursor (0, 1);
lcd.print ("Temp: ");
lcd.setCursor (7, 1);
lcd.print(TempLM35);
delay (250);

```

```

if (TempLM35<=24){           //Calefactor
    digitalWrite(Calefactor, HIGH);
}
if (TempLM35>=26){
    digitalWrite (Calefactor, LOW);
}
}

```

Imagen de Proteus;

