

Trabajo Final: Arduino.
**“Comedero para mascota
automático”.**

Computación II -2020

Daiana Dominikow.

Legajo: 32529.

Objetivos:

Realizar un programa en Arduino con alguna utilidad práctica, utilizando conocimientos adquiridos en la materia Computación II.

Introducción:

Para este trabajo se ideó un sistema automatizado que lee el nivel de comida en un comedero de animales, y según éste abre una compuerta de un depósito mediante un servomotor para llenarlo.

En este caso particular, se calcula sobre el comedero de Marta, una cobaya de avanzada edad que suele quedarse sola por largas horas durante la jornada laboral y cursada, (fuera del régimen pandémico). Se utilizó un sensor de distancia para determinar si el comedero está o no vacío y asegurar el correcto suministro de alimento.

Esquema Experimental:

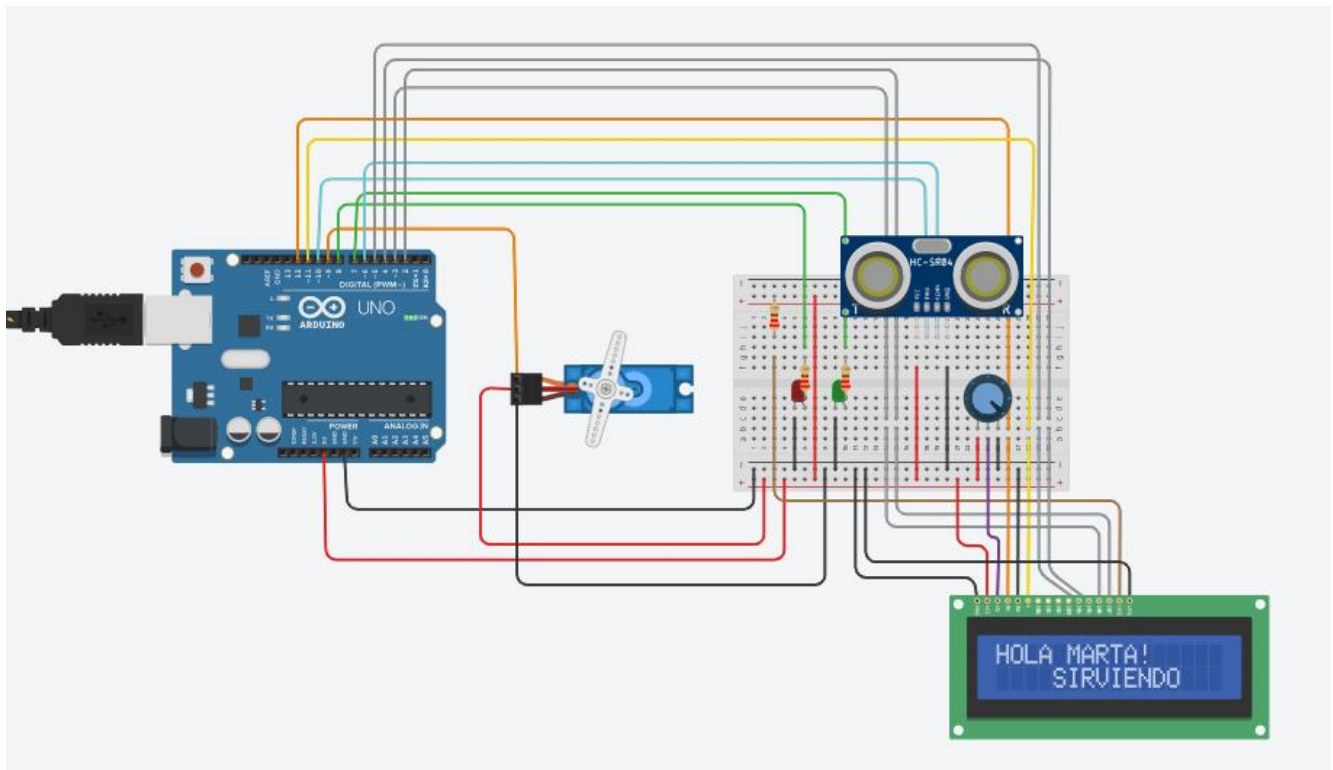


Ilustración 1: Esquema del sistema

Desarrollo:

La idea del proyecto es colocar el servomotor a un depósito plástico, y a una de sus paletas unirle una puerta que en posición de reposo mantiene cerrada una abertura en el depósito, y al sensor que el comedero está vacío, se mueva formando dejando salir comida por un determinado intervalo de tiempo.

Se enumeran las partes relevantes utilizadas para el experimento, sobre una placa Arduino UNO¹.

Sensor ultrasónico:

El sensor HC SR04² tiene un rango de alcance de 2 a 330 cm, se calibró para ser colocado a 20 cm sobre el nivel del comedero, de manera que, si lee una distancia mayor, (o sea por debajo del nivel del comedero) lo llenará.

Éste funciona enviando una serie de pulsos ultrasónicos luego de recibir una orden del pin “TRIG”, éstos pulsos rebotan sobre una superficie y son detectados por el sensor piezoeléctrico, el pin ECHO cambia al estado HIGH por un tiempo igual al que el pulso demoró en volver, este tiempo es medido por un microcontrolador y con este dato finalmente se calcula la distancia a la que se encuentra el tope del alimento teniendo en cuenta la velocidad del sonido en el aire.

Pines del sensor HC SR04:

- VCC (5V)
- TRIG (*Disparo del ultrasonido*)
- ECHO (*Recepción del ultrasonido*)
- GND (0V)

Se tomó como intervalo de lectura una hora.

Microservomotor:

La idea del trabajo práctico consta de hacer que un microservomotor abra y cierre una abertura en el comedero del animal cuando el sensor detecte que está vacío. El microservomotor puede rotar aproximadamente 180 grados (90° en cada dirección) y se programa tanto la apertura como el tiempo de apertura a conveniencia.

Para invocar al servomotor se utiliza la librería Servo, incluida en el IDE de Arduino.

Los cables en el conector del servomotor están distribuidos de la siguiente forma:

- Marrón = Tierra (GND)
- Rojo = VCC (5V)
- Naranja = Señal de control (PWM)

¹ Consideraciones:

A falta de una placa real de Arduino para diseñar el trabajo práctico, se utilizó el simulador online de marca Autodesk “Tinkercad”, el código fue programado sobre el IDE que otorga la página oficial de Arduino.

² Del emulador ThinkerCad. Hay documentación indicando que su rango real es 2 a 450 cm.

Pantalla LCD 16 x 2:

Finalmente a este circuito se le agregó una pantalla LCD que muestra un mensaje según el estado del circuito (Comedero lleno, Sirviendo, etc.).

Para conectar la pantalla LCD se necesitan 6 pines: 2 de control y 4 de datos, y al igual que el servomotor, invoca una librería incluida en el IDE de Arduino LiquidCrystal.

A este se le agregó un potenciómetro para controlar el nivel del contraste.

Los pines para éste LCD son:

- 1 - GND (Tierra)
- 2 VCC (5V)
- 3 Contraste de pantalla (Conectado al potenciómetro)
- 4 RS – Selector entre comandos y datos (1 / 0)
- 5 RW – Escritura y lectura de comandos y datos (En este caso se conectó a GND, dado que no se utilizó como entrada.)
- 6 Envío de datos a los distintos pines
- 7 al 14 Pines de datos.
- 15 Alimentación luz de fondo (5V)
- 16 GND (Tierra) luz de fondo (0V)

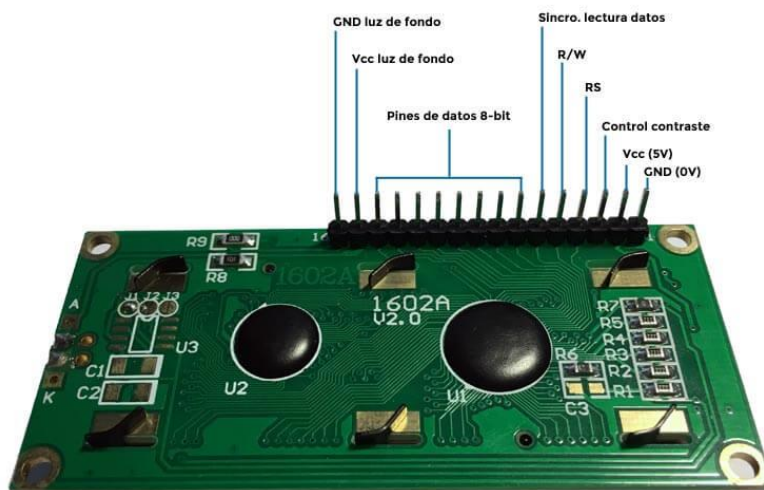


Ilustración 2: Pines al reverso de la pantalla LCD

Anexo:

Script:

```
#include <Servo.h>
```

```
#include <LiquidCrystal.h>
```

```
Servo puerta;
```

```
//sintaxis: LiquidCrystal(rs, enable, d4, d5, d6, d7)
```

```
//si rw no está conectado a tierra: LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)
```

```
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
```

```
int pos = 0;
```

```
int trig=10;
```

```
int echo=6;
```

```
int led_lleno=7;
```

```
int led_vacio=8;
```

```
int distancia=20;
```

```
int duracion;
```

```
int delaysensor=1000;
```

```
//long delaysensor=3600000;//la idea era que cada una hora sense el nivel del comedero
```

```
float v_sonido=0.034;
```

```
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```
void setup() {
```

```
puerta.attach(9);
```

```
pinMode(led_lleno,OUTPUT);
```

```
pinMode(led_vacio,OUTPUT);  
pinMode(trig,OUTPUT);  
pinMode(echo,INPUT);  
Serial.begin(19200); // el 19200 es la velocidad de transmision
```

```
lcd.begin(16,2);  
lcd.print("HOLA MARTA!");
```

```
}
```

```
long calculaDistancia(int duracion){  
long d;  
d=(duracion*v_sonido/2);  
return d;  
}
```

```
void loop() {
```

```
  //Sensor
```

```
  digitalWrite(trig,LOW);  
  delayMicroseconds(4);  
  digitalWrite(trig,HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trig,LOW);
```

```
  duracion=pulseIn(echo,HIGH);  
  distancia = calculaDistancia(duracion);
```

```
Serial.println("DISTANCIA: ");
```

```
Serial.println(calculaDistancia(duracion));
```

```
// se supone que el nivel del comedero está a 20 cm del sensor
```

```
// o sea, si sensa una distancia mayor a 20 cm, el comedero está vacío
```

```
if (distancia>20){
```

```
    digitalWrite(led_vacio,HIGH);
```

```
    digitalWrite(led_lleno,LOW);
```

```
    lcd.setCursor(4,1);
```

```
    lcd.print("SIRVIENDO");
```

```
    delay(1000);
```

```
    for (pos=0;pos<=90;pos++){
```

```
        puerta.write(pos);//para que se ubique a diferentes ángulos.
```

```
        delay(20);
```

```
    }
```

```
    for (pos=90;pos>=0;pos--){
```

```
        puerta.write(pos);//para que se vuelva.
```

```
        delay(10);
```

```
    }
```

```
}
```

```
else {
```

```
    lcd.setCursor(0,1);
```

```
    lcd.print("COMEDERO LLENO");
```

```
    delay(1000);
```

```
    digitalWrite(led_vacio,LOW);
```

```
    digitalWrite(led_lleno,HIGH);
```

```
    puerta.write(0);
```

```
}
```

```
delay(delaysensor);
```

```
lcd.clear();
```

```
}
```