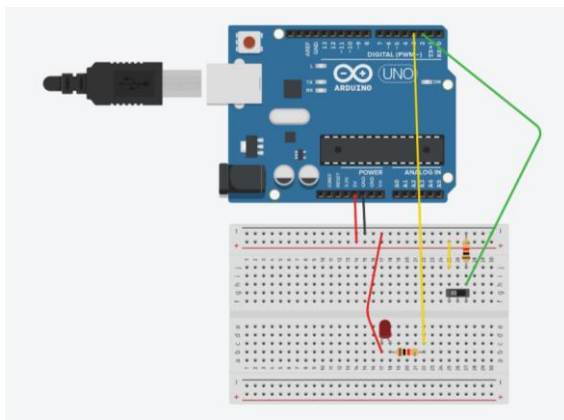


Componentes:

Nuestra nevera se compondrá de una placa base Arduino uno, sensores de temperatura y humedad(DHT22), sistema de refrigeración, fuente de alimentación(por determinar), luces led , servo motores(SG90),ultrasonidos(HC-SR04) y un módulo display Bundle(HD44780) además de una aplicación móvil adicional con la que podremos monitorizar y controlar nuestro dispositivo

En cuanto a la compatibilidad de los componentes, al tratarse de piezas utilizadas en los circuitos expuestos abajo y dando por hecho que trabajaremos en Arduino, suponemos que los componentes serán compatibles entre sí.

Esquema del montaje en Arduino:



Programa básico para luces led:

(Para este programa se utiliza un interruptor en vez de un sensor magnético)

Programa_simple_led_1

```
void setup() {  
  pinMode(2, INPUT);  
  pinMode(3, OUTPUT);  
}  
  
void loop() {  
  if(digitalRead(2)==HIGH) {  
    digitalWrite(3,HIGH);  
  }  
  else{  
    digitalWrite(3,LOW);  
  }  
}
```

En nuestro código, la puerta jugará un papel fundamental ya que será esta quien mantenga el contenido de su interior seguro, su funcionamiento será el siguiente: Tras poner la contraseña correcta, el programa que se encarga de abrir las puertas (será algo parecido a lo de abajo), se pondrá en marcha y finalizará cuando este se haya abierto por completo(o haya llegado al número de grados asignados), para cerrarse, dos componentes entrarán en marcha, el primero, será un sensor de ultrasonidos ya que será este quien detecte si hay algún tipo de presencia para así no cerrarse, si pasado un tiempo(unos 5 seg) no detecta ninguna presencia, la puerta comenzará a cerrarse hasta quedar completamente cerrada(0 grados) poniéndose de nuevo en marcha la contraseña.

Como detalle adicional, contará con dos luces led en la puerta que indicaran cuando este esté abierto(verde) y cuando esté cerrado(rojo)

El siguiente programa (que al tratarse de una versión temprana no cuenta con los componentes de la contraseña ni del ultrasonido) es de una puerta que se abre de manera automática de 10 en 10 grados y que pasado un tiempo se vuelve a cerrar.

programa_cerrar_puertas_1

```
#include<Servo.h>
Servo motor;
void setup() {
  motor.attach(3);
}
void loop() {
  AbrirPuerta();
  CerrarPuerta();
}
void AbrirPuerta(){
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH);
  motor.write(9);
  delay(500);
  motor.write(20);
  delay(500);
  motor.write(30);
  delay(500);
  motor.write(40);
  delay(500);
  motor.write(50);
  delay(500);
  motor.write(60);
  delay(500);
  motor.write(70);
  delay(500);
  motor.write(80);
  delay(500);
  motor.write(90);
  delay(500);
  motor.write(100);
  delay(500);
  motor.write(110);
  delay(500);
  motor.write(120);
```

Compilado

El Sketch usa 2496 bytes (7%)
Las variables Globales usan 50

```

programa_cerrar_puertas_1
motor.write(110);
    delay(500);
motor.write(120);
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
delay(5000);
}
void CerrarPuerta(){
    motor.write(120);
    delay(500);
    motor.write(110);
    delay(500);
    motor.write(100);
    delay(500);
    motor.write(90);
    delay(500);
    motor.write(80);
    delay(500);
    motor.write(70);
    delay(500);
    motor.write(60);
    delay(500);
    motor.write(50);
    delay(500);
    motor.write(40);
    delay(500);
    motor.write(30);
    delay(500);
    motor.write(20);
    delay(500);
    motor.write(90);
    digitalWrite(4,LOW);
    digitalWrite(5,HIGH);
}

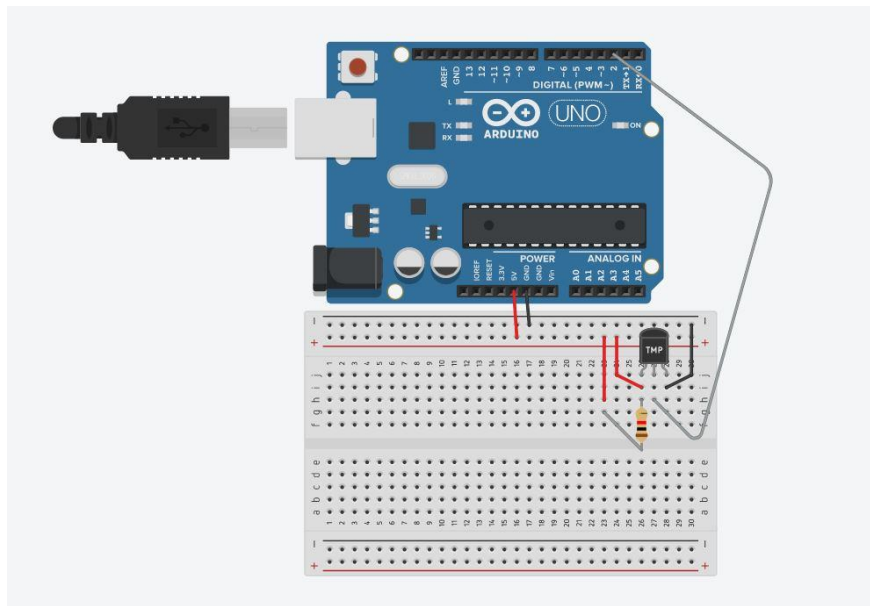
```

Compilado

El Sketch usa 2498 bytes (7%) de memoria.
 Las variables Globales usan 50 bytes de memoria.

Finalmente, haremos un breve ejemplo de un programa que recoge la temperatura leída y la muestra en pantalla.

Este sensor será capaz de detectar la temperatura y humedad para que los productos que se encuentren en su interior no se deterioren.



Así quedaría diseñado finalmente el programa para Arduino del sensor encargado de medir la temperatura y la humedad dentro del aparato, en este caso hemos decidido programarlo para el sensor DHT22, aunque también es perfectamente compatible para el sensor DHT11, solo habría que sustituir el nombre de uno por el otro dentro de nuestro programa de Arduino, en correspondencia con el que se haya escogido para realizar el trabajo.

```
Sensor_de_Temperatura_y_Humedad__DHT22_o_DHT11_§
```

```
#include <DHT.h>
#include <DHT_U.h>

int SENSOR = 2;
int TEMPERATURA;
int HUMEDAD;

DHT dht (SENSOR, DHT22);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  TEMPERATURA = dht.readTemperature();
  HUMEDAD = dht.readHumidity();
  Serial.print("Temperatura: ");
  Serial.print(TEMPERATURA);
  Serial.print("Humedad: ");
  Serial.println(HUMEDAD);
  delay(500);
}
```

