

## Tarea para el 25 de Marzo

Tenemos:

- La placa calentadora
- El relé de potencia
- El medidor de volumen (medidor de distancias)
- Sonda Impermeable

### Planteamiento del trabajo:

Desde el ordenador se indicará al programa que encienda la placa calefactora. El programa irá tomando datos de la temperatura y del volumen del agua mediante los sensores de calor y de distancia. Cuando se haya llegado a una temperatura considerable o a un volumen determinado el programa apagará la placa calefactora.

En el proceso se habrán ido tomando los datos del volumen de la mezcla los cuales nos servirán para calcular el volumen del líquido resultado del proceso de destilación con un margen de error.

### Para ello utilizaremos los siguientes componentes:

- MÓDULO KY-019 RELE (1 canal 5V)
- SENSOR DS18B20 (temperatura para líquidos, sumergible)
- SENSOR HC-SR04 (sensor de distancia por ultrasonidos)

### Información del sensor de distancias:

Fuente: <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>

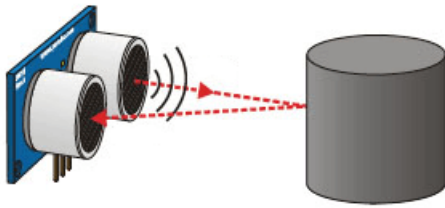
El sensor se basa simplemente en medir el tiempo entre el envío y la recepción de un pulso sonoro. Sabemos que la velocidad del sonido es 343 m/s en condiciones de temperatura 20 °C, 50% de humedad, presión atmosférica a nivel del mar. Transformando unidades resulta

$$343 \frac{m}{s} \cdot 100 \frac{cm}{m} \cdot \frac{1}{1000000} \frac{s}{\mu s} = \frac{1}{29.2} \frac{cm}{\mu s}$$

Es decir, el sonido tarda 29,2 microsegundos en recorrer un centímetro. Por tanto, podemos obtener la distancia a partir del tiempo entre la emisión y recepción del pulso mediante la siguiente ecuación.

$$Distancia(cm) = \frac{Tiempo(\mu s)}{29.2}$$

El motivo de dividir por dos el tiempo (además de la velocidad del sonido en las unidades apropiadas, que hemos calculado antes) es porque hemos medido el tiempo que tarda el pulso en ir y volver, por lo que la distancia recorrida por el pulso es el doble de la que queremos medir.



$$\text{Tiempo} = 2 \cdot (\text{Distancia} / \text{Velocidad})$$
$$\text{Distancia} = \text{Tiempo} \cdot \text{Velocidad} / 2$$

```
const int EchoPin = 5;
const int TriggerPin = 6;

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

    pinMode(TriggerPin, OUTPUT);
    pinMode(EchoPin, INPUT);
}

void loop() {
    int cm = ping(TriggerPin, EchoPin);

    Serial.print("Distancia: ");
    Serial.println(cm);

    delay(1000);
}

int ping(int TriggerPin, int EchoPin) {
    long duration, distanceCm;

    digitalWrite(TriggerPin, LOW); //para generar un pulso limpio
    ponemos a LOW 4us
    delayMicroseconds(4);
    digitalWrite(TriggerPin, HIGH); //generamos Trigger (disparo) de
    10us
```

```
delayMicroseconds(10);

digitalWrite(TriggerPin, LOW);

duration = pulseIn(EchoPin, HIGH); //medimos el tiempo entre pulsos,
en microsegundos

distanceCm = duration * 10 / 292 / 2; //convertimos a distancia, en
cm

return distanceCm;
}
```

## Información del sensor de temperatura:

Fuente: <https://www.luisllamas.es/temperatura-liquidos-arduino-ds18b20/>

El sensor DS18B20 es un sensor barato y, sin embargo, bastante avanzado. Dispone de un rango amplio de medición de  $-55^{\circ}\text{C}$  a  $+125^{\circ}\text{C}$  y una precisión superior a  $\pm 0.5^{\circ}\text{C}$  en el rango  $-10^{\circ}\text{C}$  de  $+85^{\circ}\text{C}$ .

Una de las ventajas del DS18B20 es que se comercializa tanto en un integrado TO-92 como en forma de sonda impermeable, lo que permite realizar mediciones de temperatura en líquidos y gases.

El DS18B20 emplea un bus de comunicación denominado 1-Wire propietario de la empresa Maxim Integrated, aunque podemos usarlo sin tener que pagar por ninguna tasa (es parte del precio del dispositivo).

La principal ventaja del bus 1-Wire es que necesita un único conductor para realizar la comunicación (sin contar el conductor de tierra). Los dispositivos pueden ser alimentados directamente por la línea de datos, o mediante una línea adicional con una tensión de 3.0 a 5.5V.

Dentro del mismo bus 1-Wire podemos instalar tantos sensores como deseemos. Además, el bus 1-Wire permite emplear cables más largos que otros sistemas antes de que se deteriore la comunicación.

El DS18B20 también dispone de un sistema de alarma que permite grabar en la memoria no volátil del DS18B20 los límites inferiores y superiores. El bus 1-Wire permite consultar si algún dispositivo conectado ha activado una alarma.

El DS18B20 es un gran sensor para la medición de temperatura, tanto en ambientes domésticos como industriales. Sus características permiten crear redes con gran

número de sensores para controlar, por ejemplo, la climatización o el sistema HVAC de un edificio comercial.

Para poder leer las temperaturas del DS18B20, necesitamos usar la [librería 1-Wire](#) y la librería Dallas Temperature.

En el primer ejemplo, leeremos un único sensor ubicado en el pin digital 5.

```
#include <OneWire.h>

#include <DallasTemperature.h>

const int oneWirePin = 5;

OneWire oneWireBus(oneWirePin);

DallasTemperature sensor(&oneWireBus);

void setup() {

  Serial.begin(9600);

  sensor.begin();

}

void loop() {

  Serial.println("Leyendo temperaturas: ");

  sensor.requestTemperatures();

  Serial.print("Temperatura en sensor 0: ");

  Serial.print(sensor.getTempCByIndex(0));

  Serial.println(" °C");

  delay(1000);

}
```

## Información del relé:

El módulo de relé KEYES 5V está perfectamente hecho para Arduino. Tiene tres pines, el VCC, GND y Signal. Puede actuar como interruptor si el circuito y el circuito de carga tienen diferente tensión de alimentación. Es de uso común si el circuito de carga es de corriente alterna. Es un interruptor que se utiliza para conectar una conexión aislada del circuito utilizando una señal de circuito. Tiene un LED rojo que se enciende cada vez que la bobina es energizada o el pin de señal tiene una entrada alta.

### Especificaciones:

- Señal de control TTL de 5V – 12 V
- Máxima corriente y voltaje AC: 10A 250VAC
- Máxima corriente y voltaje DC: 10A 30VDC
- La señal de control CC o CA, la carga de 220 V CA se puede controlar
- Hay un contacto normalmente abierto y otro normalmente cerrado
- Para hacer que la bobina del relé se energice, debe tener una entrada de 1 en el pin de la señal.

### Configuración de los pines:

- +: Fuente de alimentación de 5V
- - : Tierra
- S: Señal del Arduino
- NC: normalmente cerrado
- NO : normalmente abierto
- COMÚN: común



### Diagrama:

