

Control de temperatura con Arduino

(Tarea 6. Diseño de actividad de aula con Arduino, Mblock y Tinkercad)

Índice

1. Antecedentes
2. Objetivos
3. Materiales
4. Programa Arduino
5. Circuito de Tinkercad
6. Próximas prácticas

1. Antecedentes

Arduino es una plataforma electrónica programable Open Source. Esto último quiere decir que tanto su entorno de programación como el software de multitud de proyectos pueden encontrarse y están accesibles en la red. El software puede ser copiado, reutilizado y modificado libremente. En cuanto al hardware, se pueden encontrar y comprar diferentes versiones de la placa programable, así como conjuntos, más o menos completos y más o menos caros, de sensores, dispositivos y componentes electrónicos para implementar proyectos con diferente grado de complejidad.

<https://www.arduino.cc/>

<https://forum.arduino.cc/c/international/espanol/48>

Para quienes comenzar a trabajar con Arduino, pero sin necesidad de realizar ningún desembolso inicial, **Mblock** es una aplicación que se puede instalar en móvil o PC o manejar en línea que, de forma sencilla, permite iniciarse en la programación. Mediante bloques de instrucciones muy fáciles de comprender y manejar se pueden crear, entre otros, programas que luego pueden ser transferidos a una placa Arduino.

<https://mblock.makeblock.com/en-us/download/>

Siguiendo en la línea argumental del párrafo anterior, **Tinkercad** es un sitio web que permite, entre otros, probar los proyectos ideados para Arduino, ofreciéndonos la posibilidad de realizar una simulación en línea para comprobar tanto la validez del programa desarrollado, cargándolo y corriéndolo en la placa Arduino elegida, como la del montaje de hardware implementado para el proyecto.

<https://www.tinkercad.com/dashboard>

En cuanto al hardware, únicamente señalar que el elemento principal del control a implementar en esta práctica será un sensor de temperatura que sencillamente es un chip que presenta, en uno de sus terminales, un valor de tensión proporcional a la temperatura a la que está sometido. Concretamente, vamos a utilizar un modelo comercial muy extendido y fácil de encontrar por muy poco dinero: **El TMP36**. Lo primero que tendréis que hacer es buscar la hoja de especificaciones del fabricante (*Buscad TMP36 data sheet*) Aunque al principio cuesta familiarizarse con este tipo de documentación (además generalmente vienen en inglés) es la fuente donde acudir cuando se quieran conocer con exactitud las características de funcionamiento de los dispositivos, incluido el modo de conectarlo en el circuito.

2. Objetivos

El objetivo general es construir un control de temperatura que permita la activación ordenada de máquinas de frío y calor para mantener la temperatura ambiente dentro del rango situado entre los 22°C y los 27°C. Además, se señalará mediante led azul la activación de las máquinas de frío y mediante led rojo la activación de las máquinas de calor.

Por su parte, los objetivos de aprendizaje son:

- ⊕ Conocer el funcionamiento de un diodo LED y las resistencias conectadas en su circuito de encendido.
- ⊕ Encontrar la información técnica del sensor de temperatura TMP36.
- ⊕ Conocer el funcionamiento del sensor de temperatura TMP36.
- ⊕ Verificar el funcionamiento del sensor de temperatura TMP36.
- ⊕ Realizar medidas de voltaje con un multímetro digital.
- ⊕ Conocer el funcionamiento de las salidas digitales de la placa Arduino.

- ⊕ Enviar distintos valores a un pin digital de Arduino para encender y apagar un diodo LED.
- ⊕ Conocer el funcionamiento de las entradas analógicas de la placa Arduino.
- ⊕ Mapear los valores de una entrada analógica de la placa Arduino.
- ⊕ Interrelacionar el funcionamiento de un diodo LED y un sensor de temperatura TMP36 conectados a una placa Arduino.
- ⊕ Interrelacionar el funcionamiento de las entradas analógicas y las salidas digitales de una placa Arduino.
- ⊕ Conocer algunos fundamentos básicos de la programación para Arduino.

3. Materiales

Se seleccionaran los siguientes materiales de la lista de componentes disponibles en Tinkercad para llevar a cabo la simulación del montaje y el funcionamiento del control de temperatura.

- ⊕ Arduino UNO conectado al ordenador mediante USB.
- ⊕ Protoboard. Placa de pruebas pequeña.
- ⊕ 2 diodos LED. Uno azul y otro rojo.
- ⊕ 2 resistencias de 220Ω .
- ⊕ Sensor de temperatura TMP36
- ⊕ Multímetro.
- ⊕ Cables de conexión.

4. Programa de Arduino

A los alumnos se les pedirá incluir en este espacio una captura de los bloques de programación creado en Mblock para la práctica y, además, el texto del programa que trasladado a Tinkercad.

cuando Arduino Uno se inicia

fija pinA a 7

fija pinR a 8

fija pinTMP a 0

pon el pin digital pinA a bajo

pon el pin digital pinR a bajo

para siempre

fija leeTMP a lee pin analógico pinTMP

fija Voltios a $5.0 / 1024 * \text{leeTMP}$

fija gradsVolt a $\text{Voltios} * 100$

fija temperatura a $\text{gradsVolt} - 50$

si temperatura > 27 entonces

pon el pin digital pinR a bajo

pon el pin digital pinA a alto

si no

si temperatura < 22 entonces

pon el pin digital pinA a bajo

pon el pin digital pinR a alto

si no

pon el pin digital pinR a bajo

pon el pin digital pinA a bajo

!!!IMPORTANTE!!!
Al llevar el código a Tinkercad se deber poner 5.0 en lugar de solo 5 para que C++ maneje decimales y no trunque a enteros, en cuyo caso esta división sería siempre 0, la temperatura 0 °C y siempre estaría encendido el led rojo.

El TMP36 varia 10 mV por cada grado °C. Entonces, cada voltio representa 100 grados °C.

El rango del TMP36 va desde -50 °C a 150 °C que corresponde a un excursión desde 0 voltios a 2 voltios. La temperatura del TMP36 modelado en Tinkercad se puede variar desde -40 °C a 125 °C o lo que es lo mismo, desde 100 mV a 1.75 voltios.

Dejar de inyectar calor e inyectar frío.

Dejar de inyectar frío e Inyectar calor.

Dejar de inyectar frío o calor.

```
// generated by mBlock5 for <your product>
```

```
// codes make you happy
```

```
//#include <Arduino.h>
```

```

#include <Wire.h>

#include <SoftwareSerial.h>


float pinR = 0;

float pinA = 0;

float leeTMP = 0;

float pinTMP = 0;

float temperatura = 0;

float Voltios = 0;

float gradsVolt = 0;


void _delay(float seconds) {

    long endTime = millis() + seconds * 1000;

    while(millis() < endTime) _loop();

}


void setup() {

    pinMode(pinA,OUTPUT);

    pinMode(pinR,OUTPUT);

    pinMode(A0+pinTMP,INPUT);

    pinA = 7;

    pinR = 8;

    pinTMP = 0;

```

```

digitalWrite(pinA,0);

digitalWrite(pinR,0);

while(1) {

    leeTMP = analogRead(A0+pinTMP);

    // ¡¡¡¡¡IMPORTANTE!!!

    // Al llevar el código a Tinkercad se deber poner 5.0 en

    // lugar de solo 5 para que C++ maneje decimales y no trunque

    // a enteros, en cuyo caso esta división sería siempre 0, la

    // temperatura 0 °C y siempre estaría encendido el led rojo.

    Voltios = (5.0 / 1024) * leeTMP;

    // El TMP36 varia 10 mV por cada grado °C. Entonces, cada voltio representa
    100 grados °C.

    gradsVolt = Voltios * 100;

    // El rango del TMP36 va desde -50 °C a 150 °C que corresponde a

    // un excursión desde 0 voltios a 2 voltios. La temperatura del

    // TMP36 modelado en Tinkercad se puede variar desde -40 °C a

    // 125 °C o lo que es lo mismo, desde 100 mV a 1.75 voltios.

    temperatura = ((gradsVolt - 50));

    if(temperatura > 27){

        // Dejar de inyectar calor e inyectar frío.

        digitalWrite(pinR,0);

        digitalWrite(pinA,1);

    }else{

```

```

    if(temperatura < 22){

        // Dejar de inyectar frio e Inyectar calor.

        digitalWrite(pinA,0);

        digitalWrite(pinR,1);


    }else{

        // Dejar de inyectar frío o calor.

        digitalWrite(pinR,0);

        digitalWrite(pinA,0);


    }

}

_loop();

}

void _loop() {

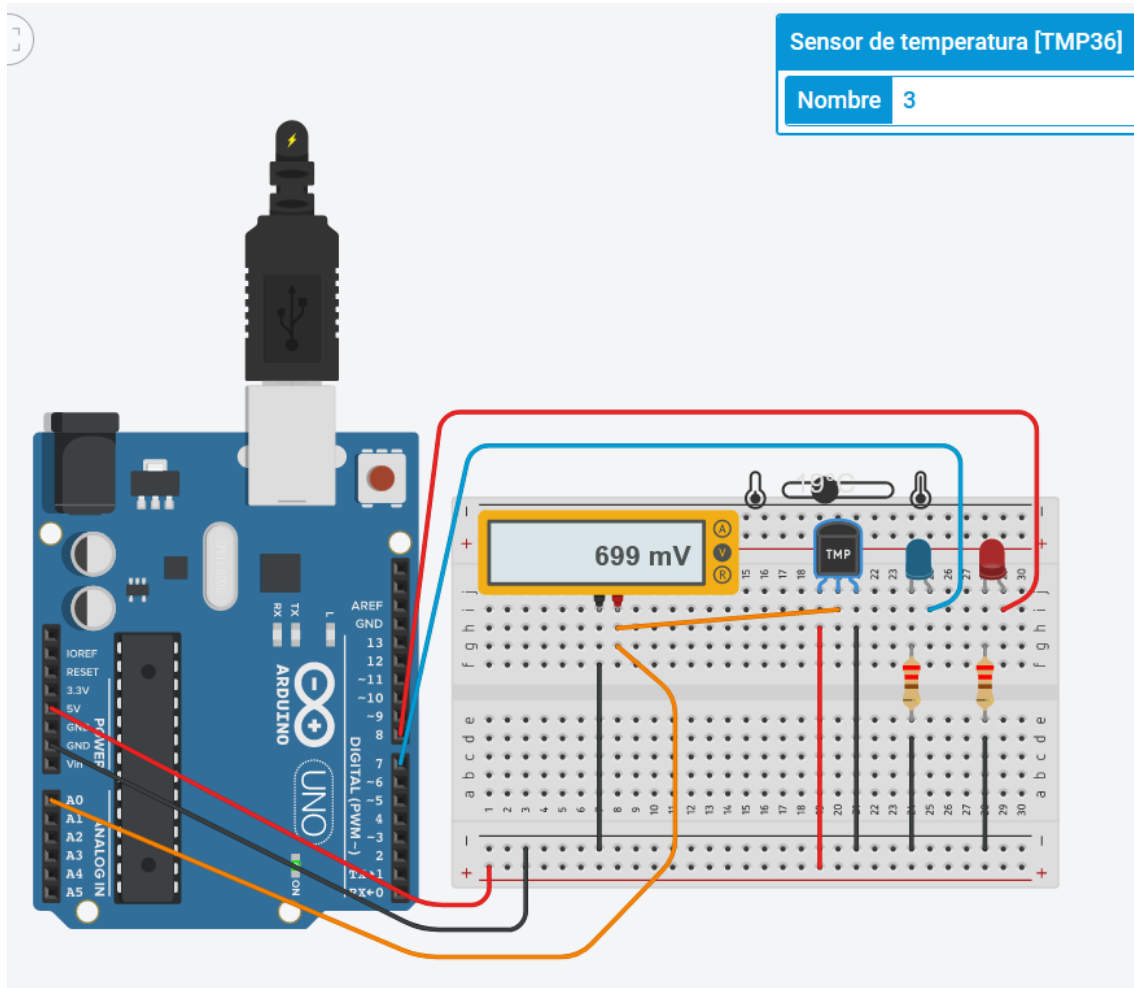
}

void loop() {

```

```
_loop();  
}
```

5. Circuito de Tinkercad



A los alumnos se les pedirá incluir en este espacio una captura del montaje realizado en Tinkercad. La captura se deberá hacer mientras se esté realizando la simulación del control de temperatura.

6. Próximas prácticas

La actividad está pensada para el módulo de Sistemas Eléctricos y Electrónicos del primer curso del Grado Superior de Mecatrónica Industrial. Siendo la primera de una serie que engloban la construcción de un control de temperatura con Arduino. En

esta se aprende a utilizar herramientas en línea como Mblock y Tinkercad para realizar tareas sencillas, y ya en las posteriores se trabajaría con el IDE de Arduino para la totalidad de un proyecto de cierto calado. Se informará a los alumnos que las siguientes prácticas se ocuparán de las restantes fases hasta la finalización del control de temperatura. En ellas habrá que comenzar a trabajar con el entorno de programación de Arduino y dispositivos reales una vez superada esta primera fase de simulación del núcleo básico del control, o sea, está práctica. Luego, y en primera instancia, se debería incorporar un teclado de tipo matricial que permita reprogramar los umbrales de temperatura para la inyección de frío y calor. Con posterioridad, una pantalla LCD que posibilite visualizar en tiempo real la medida de temperatura, así como los umbrales de disparo configurados y la reprogramación de los mismos. Por último, se incluirán los elementos que permitan encender y apagar las máquinas de frío y calor más allá de su mera señalización.