

RELOJ DIGITAL CON SENSOR DE TEMPERATURA Y HUMEDAD

Proyecto realizado por el grupo 03: Daniel García Marín, Pablo Hortelano Romero, Fernando Peñas Jaramillo, Joan Trujillo Fernández.

Este proyecto pertenece a la práctica final de la asignatura de Diseño de Sistemas Empotrados de Tiempo Real en el curso 2022-2023.

Objetivo:

El objetivo de este proyecto era el diseño y programación de un reloj-termómetro digital. Como indica su nombre las funciones principales son proporcionar la fecha y la hora y con la ayuda de un botón mostrar también la temperatura y humedad actual. Como función secundaria hemos implementado un regulador automático del brillo de la pantalla mediante un LDR y en base a la luminosidad del ambiente que este capte.

La función secundaria de la luminosidad no funciona con consistencia. En futuras versiones arreglaríamos esta función y contemplaríamos añadir una funcionalidad de despertador, añadiendo un zumbador y otro botón para establecer la hora.



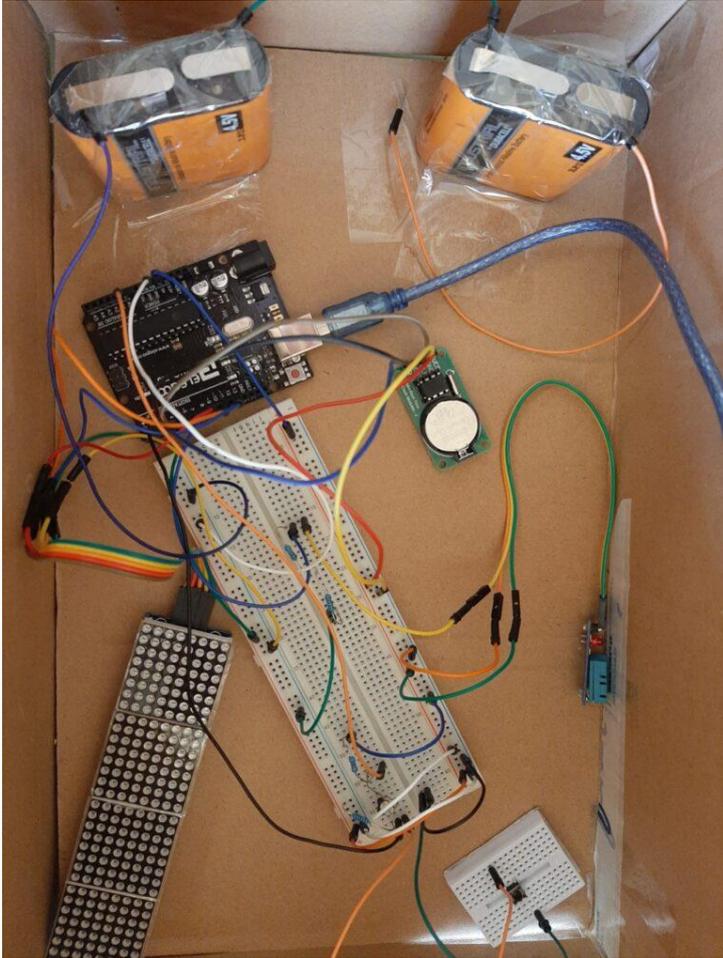
Materiales y presupuesto:

Componente	Precio
Cables	Proporcionado por la URJC

Alargadores	Proporcionado por la URJC
Botón	Proporcionado por la URJC
Protoboard	Proporcionado por la URJC
Placa Arduino	Proporcionado por la URJC
LDR	Proporcionado por la URJC
Pantalla LED 8x32	10,49€
Sensor de Temperatura DHT11	5,99€
Reloj de Tiempo Real DS1302	5,29€
Pila de 4'5 V (x2)	18,50€
Material de montaje (cartón, cutter, celo, cinta adhesiva PVC)	3,73€

Como se puede comprobar el presupuesto total ha sido de 44€. A demás del material, como grupo hemos invertido unas 10 horas a todo el proyecto sin contar la elaboración de la memoria y los videos explicativos.

Esquema de conexiones:



Uso:

El modo de uso es muy sencillo. En primer lugar, debemos conectar el proyecto mediante la placa de Arduino a un ordenador. En ese punto en el código del programa debemos poner la fecha y la hora actual y cargamos el programa en la placa. A partir de aquí el reloj-termómetro actualizará todos sus valores de forma automática hasta que su fuente de alimentación se acabe. En tal caso se deberán conectar unas baterías nuevas y repetir el primer paso. El segundo paso simplemente consiste en pulsar el botón cuando se quiera cambiar la información que muestra la pantalla.

Problemas y Soluciones:

Problemas: Algunos componentes que hemos comprado requerían de librerías específicas para su uso.

La función de adaptación de luz a la pantalla led mediante el LDR no funciona consistentemente.

Soluciones: Hemos investigado y buscado información por internet para encontrar las librerías adecuadas y también las funciones que las componen para usar los componentes.

Tras revisar el código, buscar información y revisar el cableado, no hemos encontrado el problema a la falta de consistencia de la función del brillo. Creemos que el LDR puede estar defectuoso.

Código:

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Max72xxPanel.h>
#include <DHT.h>
#include <EEPROM.h>
#include <DS1302.h>

DS1302 rtc(3, 4, 5);

#define DHTPIN 8 // Definimos el pin digital donde se conecta el sensor DHT11
#define DHTTYPE DHT11 // Dependiendo del tipo de sensor

DHT dht(DHTPIN, DHTTYPE); // Inicializamos el sensor DHT11

const int CS = 10; // pin cs del modulo reloj a la placa del arduino
const int numMatricesVertical = 1; // Cantidad de matrices led vertical
const int numMatricesHorizontal = 4; // Cantidad de matrices led
const int SensorLDR= 0; //pin LDR a a0 de placa arduino
int valorLDR=0; //inicializamos la variable a 0 para posteriormente asignar otro valor

Max72xxPanel matriz = Max72xxPanel(CS, numMatricesHorizontal, numMatricesVertical);
// Configuración de la matriz
```

```

String mensajePanel = ".. ";
String humedad = " H=";
String stringHumedad = ".. ";
String porcentajeYTemperatura = " % T=";
String stringTemperatura = ".. ";
String temperatura = ".C";
String dia = " ..";
String mes = "..";
String anio = ".. ";
String barra = "/";
String espacio = " ";
int botonPin = 2;
int valorBoton = 0;
bool botonPulsado;

const int rapidez = 43; //Velocidad con la que el mensaje recorre la matriz
const int espacioLetras = 1; // Constante para el espacio entre las letras
const int ancho = 5 + espacioLetras; // cantidad de pixeles por cada letra escrita

byte value; // Variable de tipo byte para lectura y escritura de la memoria rom eeprom del
real time clock

Time t; // Variable para el tiempo

void setup() {
  Serial.begin(9600);
  pinMode(botonPin, INPUT);
  matriz.setIntensity(0); //configuramos el brillo de las matrices led

  /*configuramos la posición de las letras y número
  que se mostrarán en cada una de las cuatro matrices*/
  matriz.setPosition(0, 0, 0);
  matriz.setPosition(1, 1, 0);
  matriz.setPosition(2, 2, 0);

```

```

matriz.setPosition(3, 3, 0);

/*configuramos el tipo de rotación de las letras y número
//que se mostrarán en cada una de las cuatro matrices*/
matriz.setRotation(0, 1);
matriz.setRotation(1, 1);
matriz.setRotation(2, 1);
matriz.setRotation(3, 1);

rtc.halt(false); // Habilita el reloj
rtc.writeProtect(false); // Deshabilita protección contra escritura

value = EEPROM.read(21); //leemos de la memoria rom eproom
if (value != 0) {
    rtc.setDOW(WEDNESDAY); // configuramos del día
    rtc.setTime(11, 27, 00); // configuramos de la hora
    rtc.setDate(27, 04, 2023); // configuramos de la fecha
    EEPROM.write(21, 1); // escribimos en la memoria rom eproom
}
dht.begin(); //Iniciamos el sensor del real time clock

}

void loop() {
    valorBoton = digitalRead(botonPin);
    // Verificar si el botón está pulsado
    if (valorBoton == HIGH) {
        botonPulsado = true;
        Serial.println("Boton no pulsado");
    } else {
        botonPulsado = false;
        Serial.println("Boton pulsado");
    }
}

```

```
t = rtc.getTime(); //metemos el valor de la hora que conseguimos con getTime() en la variable t
```

```
valorLDR = analogRead(SensorLDR); //metemos el valor de la luminosidad en la variable valorLDR
```

```
//En caso de oscuridad la intensidad descenderá a 0 y en caso de luminosidad, la intensidad subirá a valor 15
```

```
if(valorLDR<50){  
    matriz.setIntensity(0);  
}else{  
    matriz.setIntensity(15);  
}
```

```
// Leemos la temperatura en grados centígrados captada por el sensor de temp-hum
```

```
int tem = dht.readTemperature();
```

```
// Leemos la humedad captada por el sensor de temp-hum
```

```
int h = dht.readHumidity();
```

```
//leemos el dia
```

```
int d = t.date;
```

```
//leemos el mes
```

```
int m =t.mon;
```

```
//leemos el año
```

```
int a = t.year;
```

```
//guardamos los valores como Strings para mostrarlos en la matriz
```

```
stringHumedad = h;
```

```
stringTemperatura = tem;
```

```
dia = d;
```

```
mes = m;
```

```
anio = a;
```

```
String tiempo=rtc.getTimeStr();
```

```
//mostramos los mensajes en forma de scroll uniendo todos los strings que queremos mostrar y dependiendo de la hora te sale un mensaje u otro
```

```
if(botonPulsado == false){
```

```

    mensajePanel = humedad + stringHumedad + porcentajeYTemperatura +
stringTemperatura + temperatura;
}else{
    mensajePanel = tiempo+espacio+ dia +barra+ mes + barra+anio;
}
//con este for repetimos el proceso de escritura de la cadena de caracteres de manera
correcta en la matriz led
for (int i = 0; i < ancho * mensajePanel.length() + matriz.width() - 1 - espacioLetras; i++) {
    matriz.fillScreen(LOW);

    int letra = i / ancho;
    int x = (matriz.width() - 1) - i % ancho;
    int y = (matriz.height() - 8) / 2; //centrar el texto de manera vertical

    while (x + ancho - espacioLetras >= 0 && letra >= 0) {
        if (letra < mensajePanel.length()) {
            matriz.drawChar(x, y, mensajePanel[letra], HIGH, LOW, 1);
        }
        letra--;
        x = x - ancho;
    }
    matriz.write(); //Enviamos los datos para mostrarlos en la matriz.
    delay(rapidez);
}
}

```

Video de explicación:

- Video hardware: <https://youtu.be/vgZftovVK4E>
- Video software: <https://youtu.be/FUrPzjHwNs4>