

Trabajo Informática

Reloj Digital

Grupo A109

Miembros del grupo:

- Lucía Pardo Hermosa
- Javier Nicolao Gómez
- Zeqi Lin

Profesor: Joaquín González Gigoso

Curso 2021-2022

1. OBJETIVO Y DESCRIPCIÓN

Realizar un reloj digital que muestre la fecha y hora en cualquier momento (siempre que la placa de Arduino reciba alimentación), lea la fecha y hora del ordenador cuando se desee (esta opción se ha creado debido a que el módulo rtc utilizado para mostrar la hora continuamente, puede tener un error de precisión), poder programar alarmas, crear una alarma, eliminarla, ver las alarmas que hay registradas y disponer de un cronómetro, cuyos datos se pueden recoger en un archivo si se desea guardarlos.

2. ELEMENTOS DEL HARDWARE

○ Indicador fecha y hora (Módulo RTC)

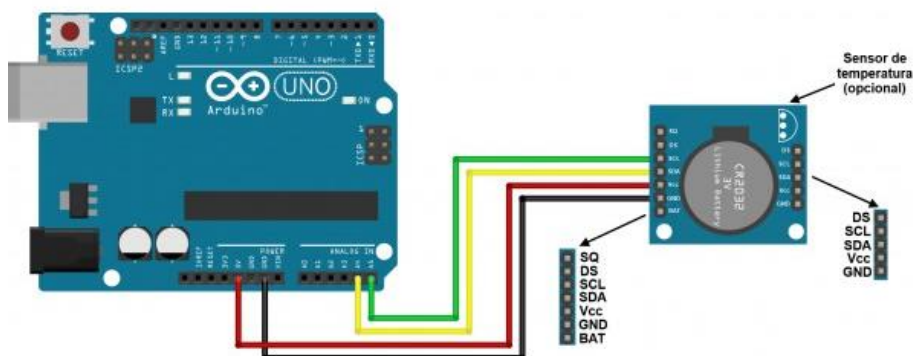
Un reloj de tiempo real (RTC) es un dispositivo electrónico que permite obtener mediciones de tiempo en las unidades de tiempo que usamos de forma habitual.

Los RTC normalmente están formados por un resonador de cristal integrado con la electrónica necesaria para contabilizar de forma correcta el paso del tiempo. Además, la electrónica de los RTC tiene en cuenta el sistema sexagesimal, los meses con diferentes días, o los años bisiestos.

Los RTC aportan la ventaja de mayor precisión y liberar a Arduino de tener que realizar la contabilización del tiempo. Además, frecuentemente los RTC incorporan algún tipo de batería que permite mantener el valor del tiempo en caso de pérdida de alimentación.

En el mundo de la electrónica casera y Arduino existen dos RTC habituales el DS1307 y el DS3231. En este caso, se ha utilizado el DS1307.

Para realizar la lectura de RTC, se recurrirá a la librería Time.h, que proporcionará las funciones necesarias.

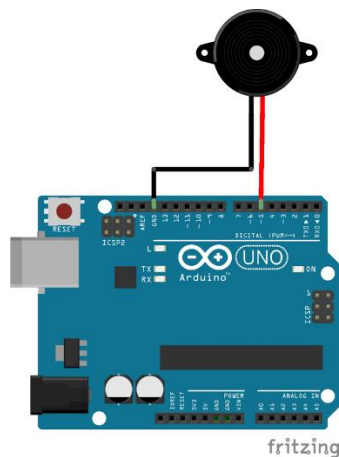


- Salida de audio (Buzzer Activo)

Se trata de un traductor electroacústico que no dispone de electrónica interna, únicamente convierte la señal eléctrica en ondas de sonido. Se puede variar el tono del sonido emitido variando la señal que se le aplica al dispositivo.

El buzzer es un dispositivo piezoeléctrico con bajo consumo eléctrico, por lo que su calidad de sonido es baja.

La función que permite generar la señal de tono en un pin es: `tone(pin, frecuencia)`.



- Pantalla LCD

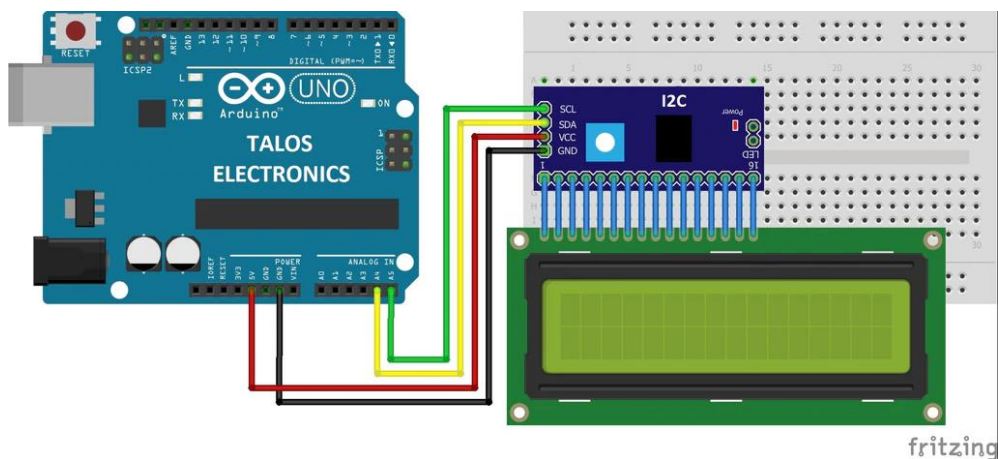
La pantalla LCD (*Liquid Crystal Display*) es un dispositivo que se encarga de convertir las señales eléctricas de la placa en información visual fácilmente entendible por los seres humanos. Está gobernado por un microcontrolador el cual dirige todo su funcionamiento.

Es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. Los datos son enviados desde el PC por medio del puerto de video a los circuitos de la pantalla LCD, donde el microcontrolador se encarga de determinar la posición de cada píxel. Una vez analizados los datos recibidos y determinado el píxel a colorear, la celda cuenta con 3 sustancias propensas a recibir corriente y colorearse de algún color básico (verde, rojo y azul) por medio de polarización. La corriente que le llega a cada píxel determina la saturación para cada color y así se generan los textos a mostrar.

Para este proyecto, se utilizará una pantalla LCD 16x2 (dos filas de dieciséis caracteres), para ello se dispondrá de un Adaptador Serial con interfaz I2C compatible con pantallas 16x2 gracias a librerías de código abierto. Este módulo proporciona una transmisión de alta velocidad y puede convertir la

transmisión de datos en paralelo de un Display LCD en una señal I2C en serie, lo que implica que solo se necesitan cuatro puertos en lugar de dieciséis. Este módulo permite alimentar la pantalla con dos cables y transferir los datos con solo dos conexiones. El bus serie I2C es muy cómodo de utilizar y permite una transferencia de datos eficaz y segura.

Para trabajar con el módulo Adaptador LCD a I2C se usará la librería LiquidCrystal_I2C.h. Las funciones que utiliza esta librería son similares a la librería LiquidCrystal de Arduino.



3. GUÍA DE USO DE LOS PROGRAMAS

El programa está dividido en dos menús principales, uno que se ejecuta cuando el Arduino está conectado y otro que se ejecuta cuando no lo está. En cualquiera de los dos casos se pueden elegir distintas funciones a realizar sin necesidad de tener que ejecutar el programa de nuevo cada vez que se quiera cambiar de función, no pudiéndose ejecutar ambos menús simultáneamente (por razones obvias de incompatibilidad). Es decir, mientras que no se indique que se quiere salir del programa, se le dará la opción de realizar una nueva función continuamente.

- Si se quiere obtener la fecha y hora del sistema:

Estando el Arduino conectado, se mostraría el menú principal 1. Dentro de las opciones del menú, la opción 1 es la encargada de obtener la fecha y hora del sistema. Esta opción permite visualizar la hora actual eliminando el error de precisión que se da con el rtc.

- Si se quiere programar una alarma:

Esta parte se divide en otras dos, en función de si el Arduino está conectado o no lo está:

- Si el Arduino no está conectado: se ejecutará el menú principal 2, compuesto por las siguientes funciones que se pueden realizar, en cuanto a las alarmas:
 1. Consultar alarmas programadas: en esta función se podrá ver el registro de alarmas que se tiene o, en caso de que no las hubiera, se mostrará un mensaje indicando que no existen alarmas programadas.
 2. Crear nueva alarma: para añadir una alarma se le pedirán los datos de esta: nombre de la alarma (este apartado es importante para poder localizar la alarma en el fichero de datos), año, mes, día, hora, minutos y segundos, en ese mismo orden.
 3. Eliminar alarma programada: si se desea eliminar una alarma anteriormente programada, en esta función se le pedirá el nombre de la alarma, si no existe se le indicará que no se ha encontrado ninguna alarma con ese nombre; en caso de que sí se haya encontrado una, se le preguntará si desea eliminar la alarma encontrada, a lo que se deberá responder con una s (SI) o una n (NO), pueden ser mayúsculas o minúsculas, ambos casos se tomarán como válidos.
- Si el Arduino está conectado: para enviar los datos de la alarma por el puerto serie (con el objetivo de poder ejecutar la función de alarma), se deberá escoger la opción 2 del menú principal 1. Esta función mostrará por pantalla los datos recibidos para que se pueda comprobar que se han enviado correctamente.

○ Si se quiere utilizar la función de cronómetro:

En este caso, también se dan dos opciones distintas en función de si existe una conexión lógica con el puerto serie:

- Si hay conexión: la opción 3 del menú principal 1, muestra por pantalla un nuevo menú, que le permite controlar las opciones del cronómetro que se están llevando a cabo en Arduino. Las opciones que se le presentan son:
 1. Inicio de la cuenta del cronómetro.
 2. Pausa.
 3. Reset/guardar marca: para reiniciar la cuenta del cronómetro y almacenar los datos obtenidos en un fichero.
 4. Salir de la aplicación.

En este caso, no se deberá pulsar intro para ejecutar las distintas funciones del menú, ya que se leerá como un carácter más, lo que dará

lugar a que el programa interprete la opción recibida como un error. Por lo tanto, para que funcione correctamente, simplemente se deberá pulsar el número de la opción que se desee ejecutar.

- Si no hay conexión: la opción 4 del menú principal 2, muestra los datos del cronómetro recogidos anteriormente, a partir de la lectura del fichero en el que han sido almacenados.

Mientras que no se estén ejecutando ninguna de las opciones anteriores, el programa de Arduino siempre mostrará la fecha y hora por la pantalla LCD, a través del módulo de reloj en tiempo real, siempre que esté recibiendo la alimentación necesaria.

Todos los casos en los que existe conexión con el puerto serie, cuentan con una función en Arduino que ejecuta el programa correspondiente en función de los datos obtenidos por el puerto serie.

4. MONTAJE

