



Universidad de Costa Rica

Sede Central Ciudad Universitaria Rodrigo Facio

San Pedro de Montes de Oca

Vicerrectoría de Acción Social

Trabajo Comunal Universitario

TC-691 Tropicalización de la Tecnología

Modo de uso del módulo Realtime clock

Brandon Palacio Delgado brandon.palacio@ucr.ac.cr

10 de marzo de 2024

Índice

1. Características generales del módulo RTC	1
2. Partes del módulo	1
3. Modo de conexión del módulo RTC	4
4. Código para hacer funcionar el módulo RTC	5

En este documento se va a mostrar el modo de uso del módulo Real Time Clock (reloj en tiempo real). La finalidad de este módulo es poder dar la fecha en día, mes y año; y la hora en hora, minutos y segundos. La intención es que lo de la manera más precisa posible; los usos pueden llegar a ser muchos, pueden ir desde solo querer tener la fecha y hora, hasta poder controlar distintos módulos de un proyecto definiendo la hora y día que se quiere que entren en funcionamiento. A continuación se va a hablar más a profundidad sobre características generales, partes del módulo, modo de conexión y el código de funcionamiento.

1. Características generales del módulo RTC

De manera resumida, el módulo RTC usa un circuito integrado DS1307 que es el cerebro del módulo para poder hacer el seguimiento de la hora y la fecha; tiene la particularidad de traer una batería de litio integrada con modelo CR1225 que lo que quiere decir que es una batería de 12 mm de diámetro y de 2,5 mm de espesor, y puede tener una duración de hasta 9 años, lo que permite que el módulo pueda seguir funcionando sin importar que no se esté alimentando mediante VCC y por último, se cuenta con el protocolo de comunicación I^2C , que se hace mediante el serial de datos SDA y el serial de reloj SCL. A continuación se presentan otras características que son importantes de exponer:

- Calendario preciso hasta 2100.
- Pin de salida con frecuencias entre 1 Hz y 32 kHz.
- Cuenta con una memoria no volátil de 56 bytes.
- En el caso de querer hacer un entorno con comunicación I^2C , la dirección hexadecimal para el módulo RTC es 0x68.
- Las dimensiones del módulo son 0,75x0,75" (20x20mm).

2. Partes del módulo

Es importante conocer los pines o los componentes que tiene el módulo; para esto, se da una breve explicación sobre que es lo que hacen y se apoya en la Figura 1:

1. El pin SDA es el que se encarga de los datos de entrada o salida, en el protocolo I^2C . Hay que tener en cuenta que es de *open-drain* lo que quiere decir que no tiene una señal lógica en alto, por lo que hay que dar mediante resistores de *pullup* que se deben colocar externamente.
2. El pin SCL va de la mano del SDA, ya que lo que hace el *serialclock* es dar el reloj que sincroniza el movimiento de los datos dentro del protocolo I^2C . Al igual que el anterior, se debe contar con un resistor *pullup*.

3. El pin SQW lo que hace es dar una señal cuadrada de salida que puede ir desde 1 Hz hasta los 32 kHz de frecuencia y al igual que los pines anteriores, necesita de un resistor *pullup*.
4. El pin GND es la referencia a tierra.
5. El pin 5V o VCC es el que da la alimentación al módulo o para ser más concreto, al integrado principal del módulo. Debe ser una alimentación de 5V.
6. El integrado principal del módulo es el DS1307, que es un reloj/calendario, el cual se puede decir que funciona como un módulo *slave* (esclavo) y que tiene como *master* (maestro) al arduino FioV3, esto se logra mediante el protocolo I^2C . El *pinout* del integrado se puede ver en la Figura 2.
7. El oscilador es el encargado de dar la frecuencia de reloj necesaria para llevar la cuenta de la fecha y la hora, se basa en un cristal de cuarzo, el cual se ve que es externo y una de las desventajas de este modelo es que es propenso a las variaciones de temperaturas, lo que hace que se afecte su precisión. Este oscilador tiene una frecuencia de 32.768 kHz.
8. Se cuenta con un jumper de soldadura, así como se ve, está soldado y hace que estén en conexión los resistores de *pullup*, en dado caso de que no se quiere tenerlas, solo se quita la soldadura y se pierde la conexión.
9. El capacitor SMD que se usa es de desacople, para filtrar posibles ondas electromagnéticas o ruido que pueda darse de la fuente de alimentación, este es de 0.1 μF .
10. En la figura proporcionada no se ve donde va puesta la batería de litio, pero tiene un *case* donde va puesta la batería y da la alimentación auxiliar para cuando el módulo no está conectado a arduino y de igual manera sigue funcionando el la hora y la fecha.

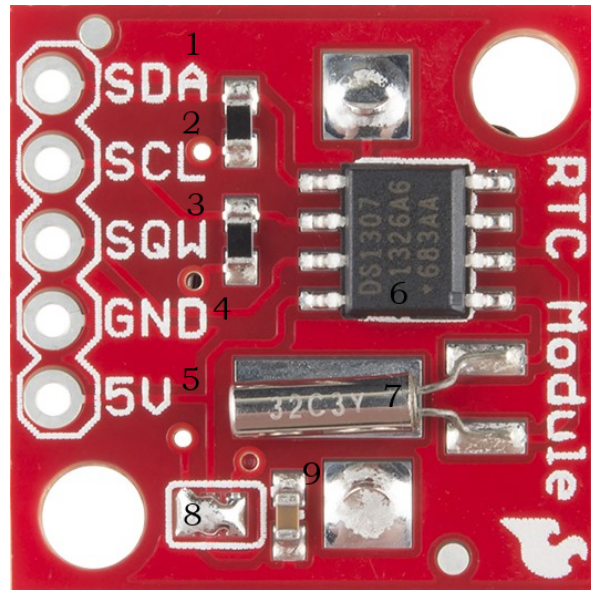


Figura 1: Componentes y pines del módulo RTC.

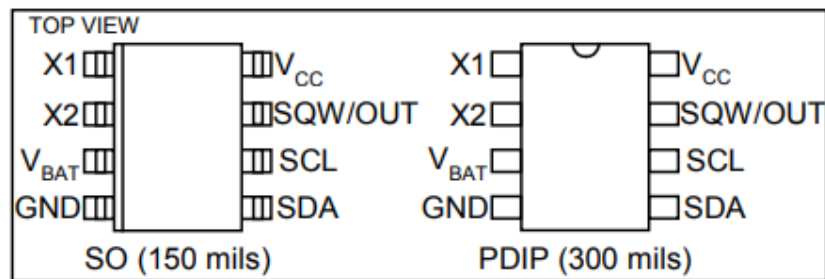


Figura 2: Pinout del integrado DS1307.

En la Figura 3 se puede ver el esquemático del módulo, donde se puede apreciar de mejor forma como se dan las conexiones y a entender de mejor manera cómo funcionan los pines y los componentes antes mencionados.

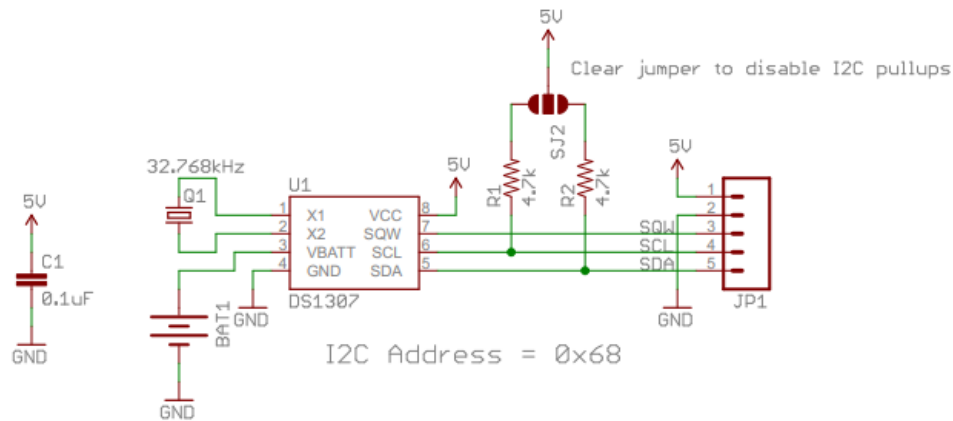


Figura 3: Esquemático del módulo RTC.

3. Modo de conexión del módulo RTC

Para hacer funcionar al módulo hay que tener en cuenta dos pasos muy sencillos, el **primero** es generar la conexión del protocolo I^2C y la **segunda** es darle alimentación de 5 V y la conexión a tierra. Ya se mostraron los pines del módulo; ahora, hay que mostrar los pines del arduino FioV3 de Sparkfun, en la Figura 4 se puede ver el *pinout* del arduino.

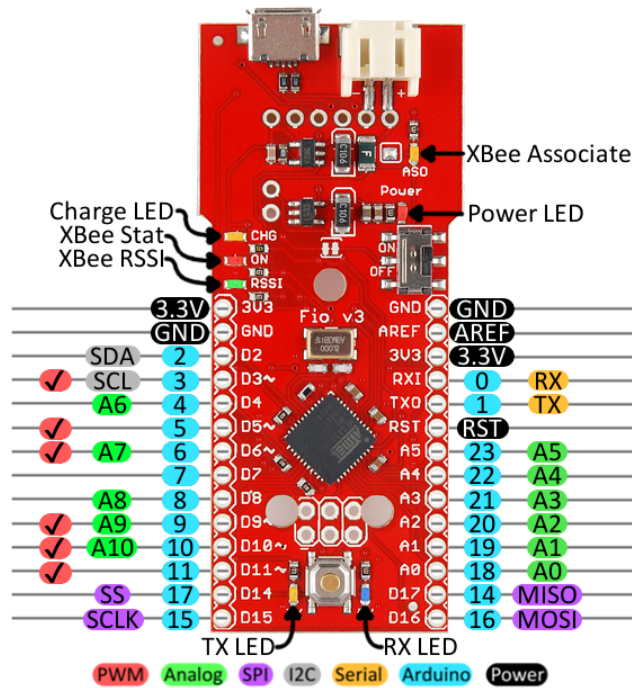


Figura 4: Pinout del arduino FioV3.

Apoyado en esto, entonces lo primero que hay que hacer es conectar ya sea el SDA o el SCL, se puede empezar por el SDA por lo que hay irse al arduino en el **pin 2**, de color celeste, se debe conectar por medio de un jumper (cable) al pin 1 del módulo RTC (de la Figura 1); ahora, el SCL que se ubica en el **pin 3** del arduino en color celeste, se conecta al pin 2 del módulo RTC; de esta manera ya se genera la conexión por protocolo I^2C , donde se dice que el maestro es el arduino y el esclavo es el módulo RTC, siempre teniendo en cuenta que las resistencias de *pullup* estén, estas pueden venir en el módulo (como este caso) o se deben agregar de manera externa, típicamente pueden ser de 4,7 k Ω .

Como segunda parte, hay que alimentar al módulo RTC, por lo que del arduino el pin de 5 V está un poco oculto, en la parte superior hay 6 huecos, donde podrían ir soldados pines macho, el de 5 V es el que está más a la derecha y este mediante un jumper se conecta al pin 5 del módulo RTC. Para la tierra, se puede ver que el arduino cuenta con 2 pines para este propósito, por lo que se puede escoger cualquiera y conectarla con la tierra del módulo que está en el pin 4; con estas indicaciones se realiza la conexión de manera correcta y tomando en cuenta alimentar al arduino, que puede ser mediante *laptop* con el cable *USB*.

4. Código para hacer funcionar el módulo RTC

En esta parte se explica el código para poner en marcha al módulo RTC, no es tan largo y es fácil de entender, se hizo basado en un video de youtube, hay que tener en cuenta que el modelo RTC que se está usando y que para este se uso el IDE (entorno de desarrollo integrado) de arduino, el que se descarga de la página oficial y que el lenguaje que se usa es *C++*. Lo primero es incluir las galerías que se necesitan, la primera es *Wire.h* que se usa para la comunicación I^2C y la librería *RTClib.h* que es para manejar el módulo RTC y hacer más fácil su uso, esto se puede ver en la Figura 5. Lo siguiente más importante es crear el objeto de tipo *RTC_DS1307*, en el *void setup* se define la comunicación serial, para poder ver información en el monitor serial, luego se define una condición para comprobar que la comunicación se ha establecido, de lo contrario se vería en el monitor serial un mensaje; si todo está bien, mediante *rtc.adjust* y *DateTime*, se establece por primer vez la fecha y la hora; luego de la primer vez, no es necesario tener esta línea de código activa, se puede comentar y en el módulo debería poder seguir con el conteo de la hora y la fecha. Con *__DATE__* y *__TIME__*, se establece la hora y fecha actual, también se puede definir manualmente, poniendo en formato largo y no necesariamente la hora y fecha actual, pero lo conveniente es inicializar el módulo con el tiempo presente.

```

10 #include <Wire.h> // incluye libreria para interfaz I2C
11 #include <RTClib.h> // incluye libreria para el manejo del modulo RTC
12
13 RTC_DS1307 rtc; // crea objeto del tipo RTC_DS1307
14
15 void setup () {
16   Serial.begin(9600); // inicializa comunicacion serie a 9600 bps
17
18   if (!rtc.begin()) { // si falla la inicializacion del modulo
19     Serial.println("Modulo RTC no encontrado !"); // muestra mensaje de error
20     while (1); // bucle infinito que detiene ejecucion del programa
21   }
22   rtc.adjust(DateTime(__DATE__, __TIME__)); // funcion que permite establecer fecha y hora
23   // al momento de la compilacion. Comentar esta linea
24   // y volver a subir para normal operacion

```

Figura 5: Primera parte del código, establecer por primera vez la fecha y hora.

Una vez ya se tiene la hora y fecha, en el *void loop* se define una función para devolver el tiempo actual y se guarda en la variable fecha como se puede ver en la Figura 6; las siguientes líneas son para definir, mediante *print*, el formato que se quiere ver tanto la fecha como la hora y esto se ve en monitor serial.

```

26 void loop () {
27   DateTime fecha = rtc.now(); // funcion que devuelve fecha y horario en formato
28   // DateTime y asigna a variable fecha
29   Serial.print(fecha.day()); // funcion que obtiene el día de la fecha completa
30   Serial.print("/"); // caracter barra como separador
31   Serial.print(fecha.month()); // funcion que obtiene el mes de la fecha completa
32   Serial.print("/"); // caracter barra como separador
33   Serial.print(fecha.year()); // funcion que obtiene el año de la fecha completa
34   Serial.print(" "); // caracter espacio en blanco como separador
35   Serial.print(fecha.hour()); // funcion que obtiene la hora de la fecha completa
36   Serial.print(":"); // caracter dos puntos como separador
37   Serial.print(fecha.minute()); // funcion que obtiene los minutos de la fecha completa
38   Serial.print(":"); // caracter dos puntos como separador
39   Serial.println(fecha.second()); // funcion que obtiene los segundos de la fecha completa
40
41   delay(1000); // demora de 1 segundo
42 }

```

Figura 6: Segunda parte del código, se le da el formato de impresión.

En las Figuras 7 y 8, se pueden ver las pruebas que se hicieron y se ve correctamente el formato que se definió, va contando el tiempo cada segundo como parte del *delay* que se le dio. Para la Figura 8, se hizo la prueba de desconectar por 55 segundos el arduino de la alimentación, la hora antes de hacer esto marcaba 16:57:36 y al volver a alimentar el arduino, marcaba 16:58:31, con esto se demuestra que aunque se desconecte de la alimentación, sigue el conteo normal y esto gracias a la batería o alimentación auxiliar que tiene el módulo RTC.

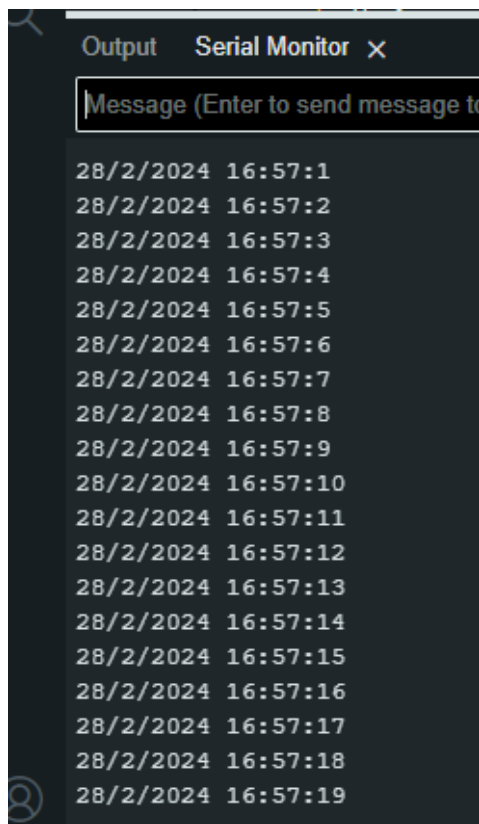


Figura 7: Prueba para mostrar el formato en el monitor serial

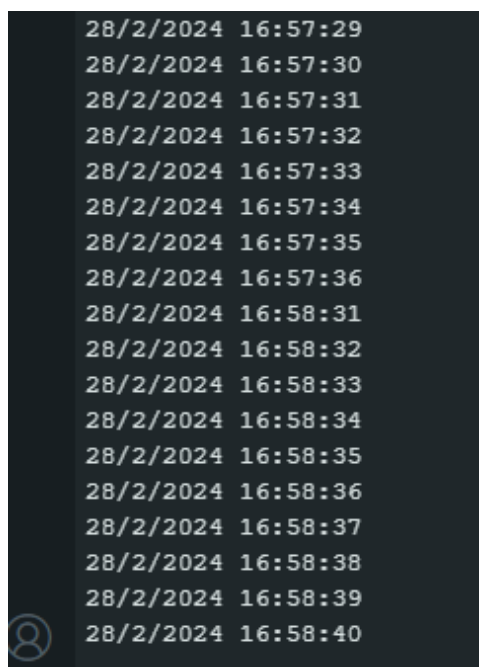


Figura 8: Prueba para comprobar que el módulo funcionar con la alimentación auxiliar.