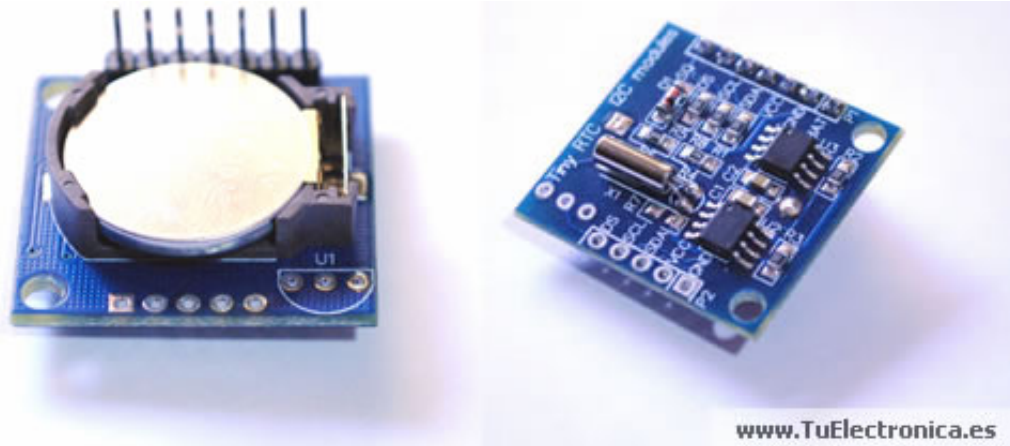


Reloj RTC I2C con Arduino



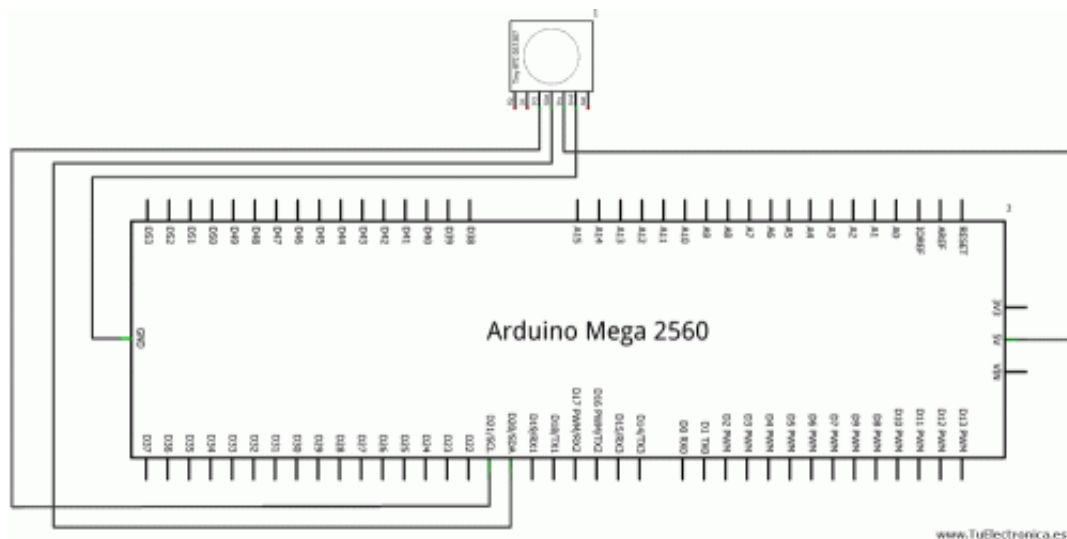
En este segundo tutorial sobre Arduino les enseñaremos a usar un modulo RTC (Real Time Clock) mediante el bus I2C. Usaremos una librería que nos facilitará la comunicación entre este modulo y Arduino. El módulo que emplearemos es el Tiny RTC I2C que incorpora un reloj de tiempo real [DS1307](#), una batería para memorizar los datos y una pequeña memoria EEPROM [24C32](#).

Arduino incorpora sus propios temporizadores, uno de ellos es `millis()`. Muchos usuarios se preguntaran para que usar un modulo externo temporizador, la respuesta es sencilla, los temporizadores de Arduino son contadores que cuando se inician empiezan desde 0, por ejemplo `millis()` se inicia una vez conectemos la alimentación de Arduino, es el contador encargado de indicar cuanto tiempo lleva encendido Arduino, y una vez se pierde la alimentación, este valor desaparece. Podemos entender entonces, que para algunos proyectos que necesitemos usar el tiempo como variable, estos contadores no nos servirían y necesitaríamos un RTC externo que posea una batería para almacenar los datos.

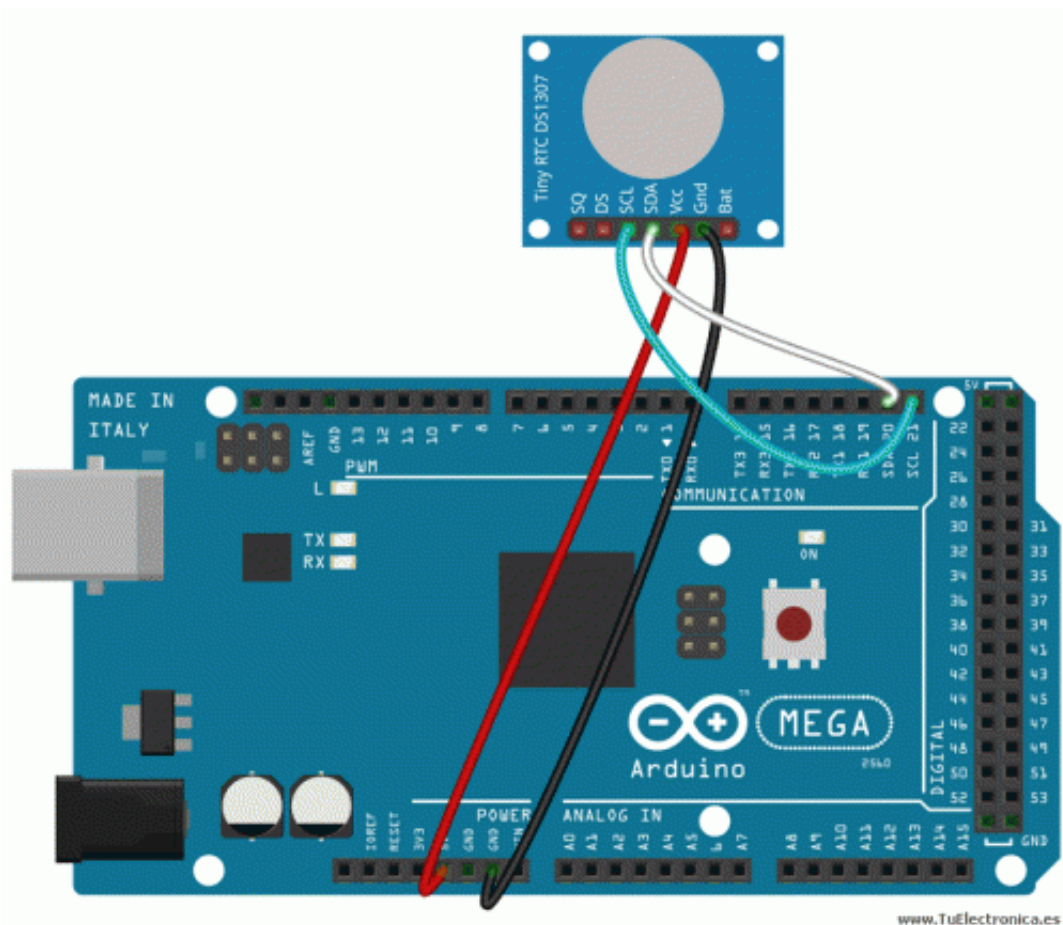


Módulo Tiny RTC I2C.

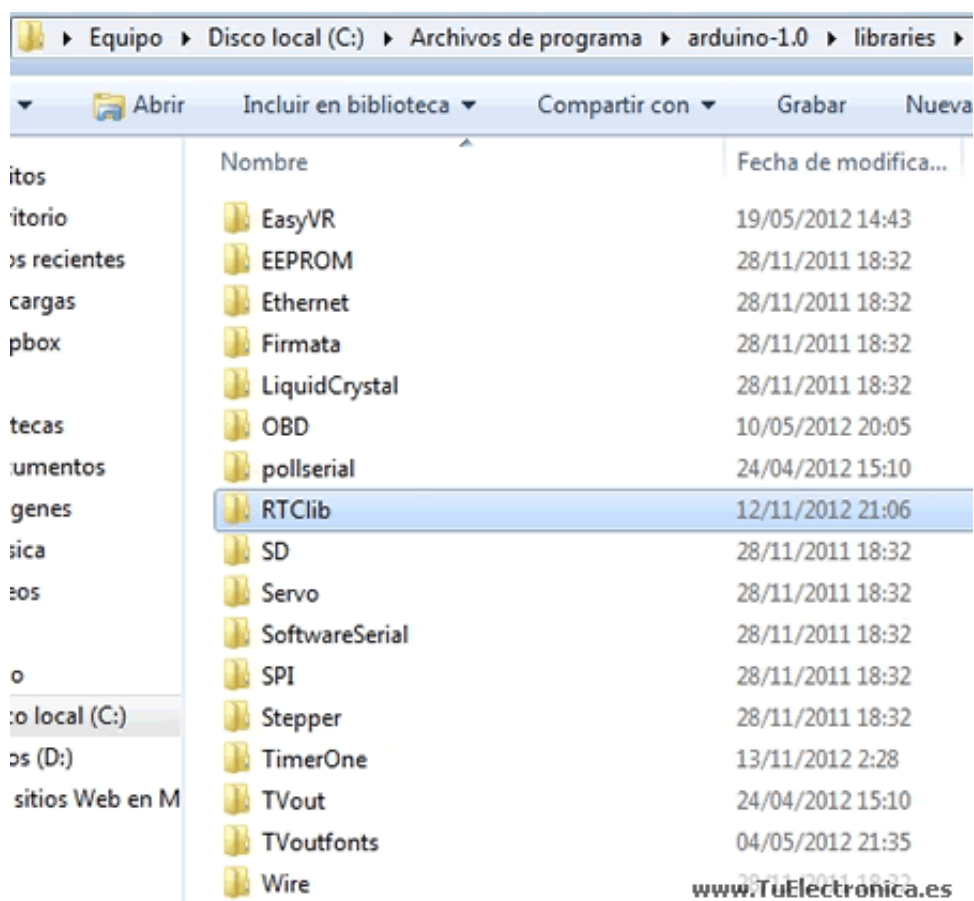
Comenzaremos conectando el modulo a nuestra placa Arduino, es sencillo de conectar ya que usa una comunicación I2C.



Esquema eléctrico Arduino Mega.



Esquema gráfico Arduino Mega.



Una vez conectado, tendremos que añadir la librería [RTCLib](#) a la IDE de Arduino. La descargaremos y descomprimiremos en *C:/directorio arduino/libraries*

Esta librería se apoya en la librería Wire para la comunicación I2C, por lo que tendremos que hacer referencia a ella he incluirla en nuestro código. No os preocupéis, la librería Wire ya viene por defecto añadida a la IDE. RTClib es compatible con cualquier modulo RTC que use el chip DS1307.

Código

```
1  #include <Wire.h>
2  #include "RTClib.h"
3  RTC_DS1307 RTC;
4  void setup () {
5  Wire.begin(); // Inicia el puerto I2C
6  RTC.begin(); // Inicia la comunicación con el RTC
7  RTC.adjust(DateTime(__DATE__, __TIME__)); // Establece la fecha y hora
8  Serial.begin(9600); // Establece la velocidad de datos del puerto
9  }
10 void loop () {
11 }
```

Descripción del código

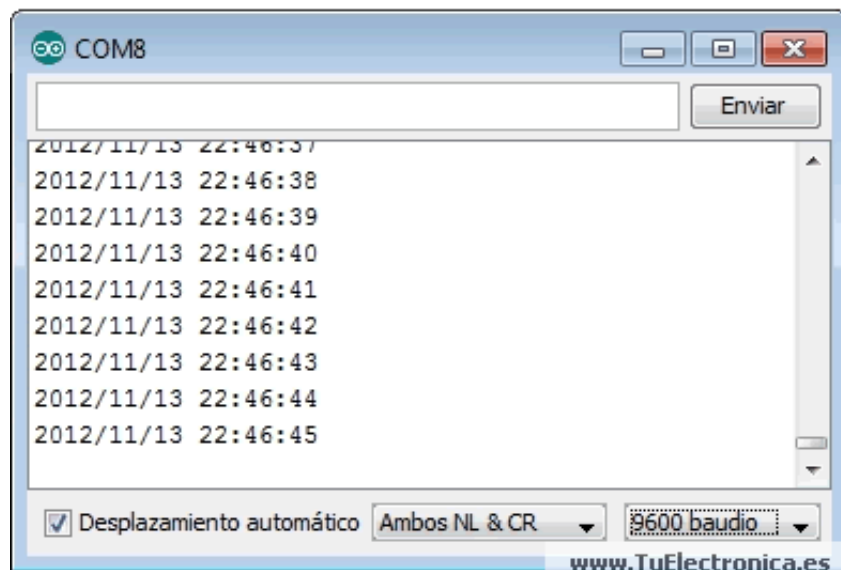
RTC.adjust(DateTime(__DATE__, __TIME__));

Esta función establecerá la fecha y hora en nuestro reloj DS1307. La fecha y hora usada para cargarla en nuestro modulo, será la fecha de creación del archivo .hex generado al compilar el sketch.

Código

Una vez cargado el código anterior en nuestro Arduino, ya debería estar configurada la fecha y hora en el módulo RTC. Para mostrar la fecha y hora actual usaremos el código siguiente.

```
1  #include <Wire.h>
2  #include "RTClib.h"
3  RTC_DS1307 RTC;
4  void setup () {
5  Wire.begin(); // Inicia el puerto I2C
6  RTC.begin(); // Inicia la comunicación con el RTC
7  //RTC.adjust(DateTime(__DATE__, __TIME__)); // Establece la fecha y hora
8  Serial.begin(9600); // Establece la velocidad de datos del puerto
9  }
10 void loop(){
11   DateTime now = RTC.now(); // Obtiene la fecha y hora del RTC
12
13   Serial.print(now.year(), DEC); // Año
14   Serial.print('/');
15   Serial.print(now.month(), DEC); // Mes
16   Serial.print('/');
17   Serial.print(now.day(), DEC); // Dia
18   Serial.print(' ');
19   Serial.print(now.hour(), DEC); // Horas
20   Serial.print(':');
21   Serial.print(now.minute(), DEC); // Minutos
22   Serial.print(':');
23   Serial.print(now.second(), DEC); // Segundos
24   Serial.println();
25   delay(1000); // La información se actualiza cada 1 seg.
26 }
```



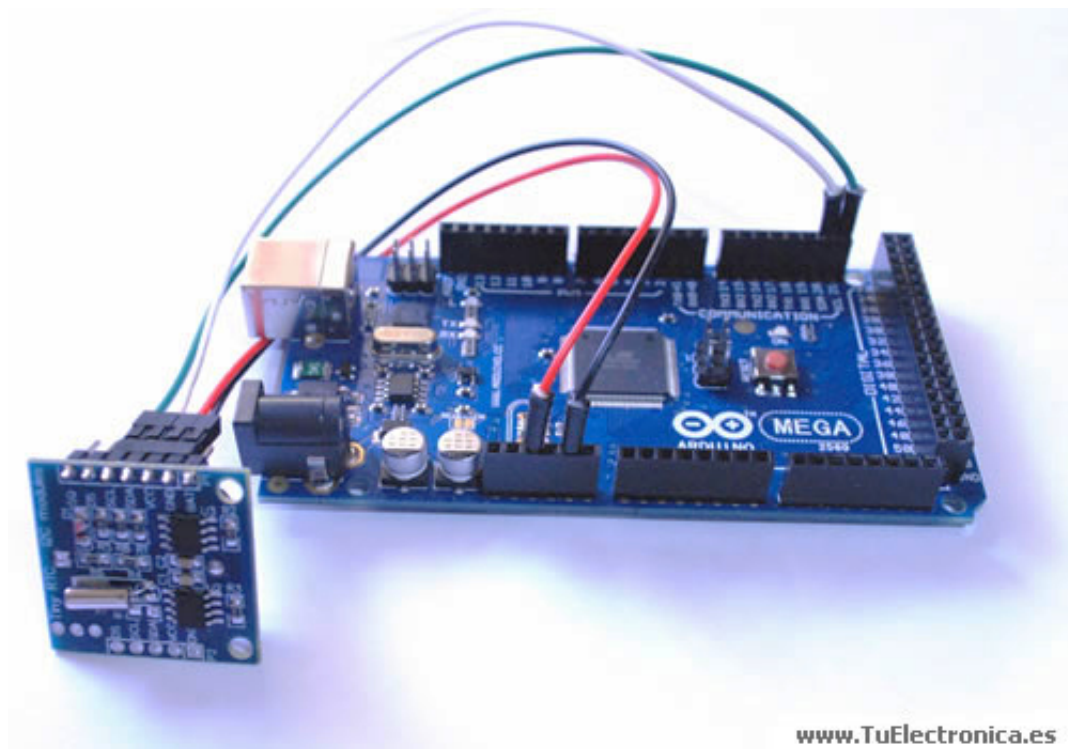
Si todo es correcto veremos la fecha actual como se muestra en la imagen.

Podremos comprobar que desconectando la alimentación, y no la batería, la fecha y hora seguirá siendo la correcta.

La librería RTClib tiene una función que merece la pena mencionar, esta función es `dayOfWeek()`. Es una función que devuelve el valor del día de la semana, es decir, si es lunes retornará un 1, si es martes un 2, si es miércoles un 3 y así sucesivamente. En el siguiente código vemos como se usa:

Código

```
1  #include <Wire.h>
2  #include "RTClib.h"
3  RTC_DS1307 RTC;
4  void setup () {
5    Wire.begin(); // Inicia el puerto I2C
6    RTC.begin(); // Inicia la comunicación con el RTC
7    //RTC.adjust(DateTime(__DATE__, __TIME__)); // Establece la fecha
8    Serial.begin(9600); // Establece la velocidad de datos del puerto
9  }
10 void loop () {
11   DateTime now = RTC.now();
12   Serial.print("Dia de la semana: ");
13   Serial.println(now.dayOfWeek(), DEC); // L(1), M(2), X(3), J(4),
14   delay(1000); // La información se actualiza cada 1 seg.
15 }
```



Resultado final.

Solución de errores

Numerosos usuarios nos han comentado alguno de los problemas que han tenido a la hora de usar este módulo. Estas son las recomendaciones a seguir para resolverlos.

Se muestra la fecha 2000/1/1 0:0:0

Este resultado se debe a que la batería del reloj está agotada o esta se ha quitado con anterioridad. Sustituya la batería por una nueva o coloque la batería y configure la fecha de nuevo.

Se muestra la fecha 2165/165/165 165:165:85

Se debe a un fallo de conexión o algún otro fallo interno producido por la manipulación del módulo mientras que esta alimentado. Comprueba las conexiones, desconectar el módulo de Vcc, quitar la pila y volver a conectar Vcc. Desconectar nuevamente Vcc poner la pila y volver a conectar Vcc.

Intenta no manipular el modulo mientras está conectado a Vcc, ya sea de Arduino u otra fuente de alimentación externa.