

RTC DS1307 Tutorial



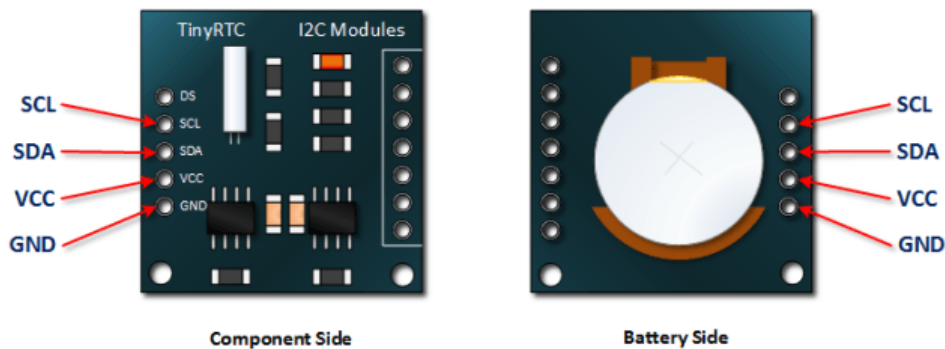
This battery powered clock allows you to keep reasonably accurate time even when your Arduino is powered off. It's very useful in data logging and other time sensitive applications.

The device uses I2C to communicate with your Arduino.

There are several internet libraries that can be used to interface with this device.

Pin Outs

You will only need four connections to make this device work.



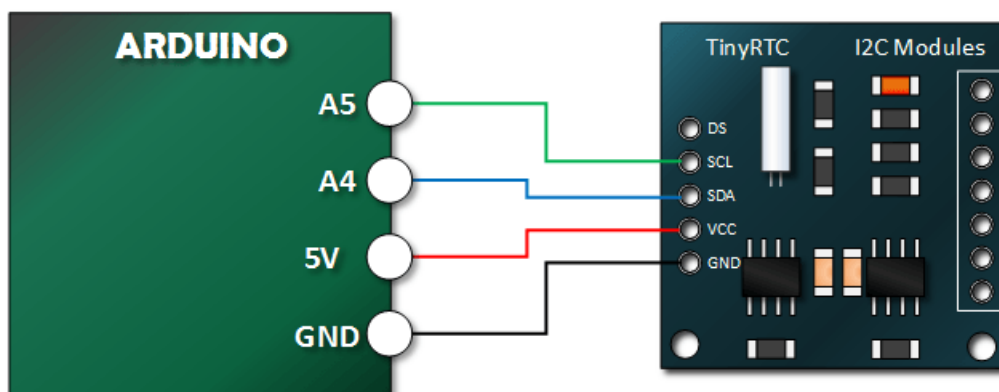
Get the RTC Libraries

There are a variety of libraries available for use with this device. This tutorial will use the Elecrow. If you are new to using libraries with your Arduino, I suggest visiting [this page](#).

This light weight library can be used to set the time and read the time. For many of your applications, this may be all you want to do. The Elecrow RTC library can be found [HERE](#).

Connect the Tiny RTC to your Arduino

This set up will work with all the sketches shown below



Arduino Sketch Using Elecrow Library

```
#include <Wire.h>
#include "RTClib.h"

RTC_DS1307 RTC;

void setup () {
  Serial.begin(9600);
  Wire.begin();
  RTC.begin();
  // Check to see if the RTC is keeping time. If it is, load the time from your computer.
  if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");
    // This will reflect the time that your sketch was compiled
    RTC.adjust(DateTime(__DATE__, __TIME__));
  }
}

void loop () {
  DateTime now = RTC.now();
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
  Serial.print('/');
  Serial.print(now.year(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
  delay(1000);
}
```

Verify Your Tutorial Sketch Output

Open your serial monitor. Your output should look something like screen capture below:

