

# Drift Corrected DS1307 Real Time Clock

---

 [limchonghan.wordpress.com/driftcorrecteds1307/](http://limchonghan.wordpress.com/driftcorrecteds1307/)

I've recently purchased the DS1307 real time clock for my data logging project only to find that it drifts about 3 seconds per day, which is not acceptable for me. I expect my data logger to be maintenance free for at least 10 years. In which time it would have drifted 7.6 days!

At first I was considering switching to the DS3231 RTC which should be accurate enough for me but then its more expensive, not available locally and I'd have to mess around with SOIC breakout boards.

So after looking around the net and not found any solution that I'm happy with I did some quick modifications to the Time and DS1307RTC Libraries to do drift correction and store the drift info in the RTC battery backed memory.

I've also written an Arduino sketch to do the RTC time setting and drift calibration. Take a look at the Arduino sketch below if you're interested. Most of the codes are supporting codes. You only need two lines of code to get the drift corrected time. One line to load the DriftInfo and one line to read the drift corrected time. You'll need the modified Time and DS1307RTC Libraries(included in the "Drift Corrected DS1307 codes" link).

Howard

[Drift Corrected DS1307 codes](#)

[Arduino Time Library](#)

[Discussion about drift correction for DS1307 on Arduino.cc Forum](#)

[DS1307 RTC Tutorial at Ladyada](#)

[DS1307 RTC at bildr Blog](#)

[An I2C Bus Example Using the DS1307 Real-Time Clock](#)

[DS1307 Datasheet](#)

## Implementation Note:

- I assume you are somewhat familiar with using the Arduino Time library and have a working DS1307 hookup to your Arduino. If not follow the links above first.
- The time in your RTC is not changed in anyway as the drift correction is computed every time now2() or now3() functions are called.
- DriftInfo contains the drift start datetime and the drift rate which is used by now2() and now3() to compute the drift corrected time. DriftInfo can be optionally stored on the DS1307 battery backed memory using 8 Bytes of memory starting at location 0x08.
- This is my preferred solution as I don't have to worry about accumulated rounding error, also switching the Arduino board off for extended period won't affect the drift correction.
- The sketch and libraries are written using Arduino 1.0.1 IDE. Just comment out the LCD and Temperature Sensor codes if you don't have an LCD and LM35 attached.
- The sketch below provides a user friendly interface to set, calibrate and monitor the DS1307. I hope others find it useful. Feedbacks, corrections, improvements are welcomed. Oh, I forgot one very important thing NO WARRANTY. Have fun! :)