

# Clock Module DS3231

## 1. Introduction:

DS3231 is a I2C real-time clock (RTC) of low cost, high precision with integrated temperature compensated crystal oscillator (TCXO) and crystal. The device includes a battery input, so if you disconnect the main power supply, it can still maintain accurate timing. The integrated oscillator improves the long-term accuracy of the device, and reduce the number of components of the production line. DS3231 provides commercial and industrial temperature range, the 300mil SO package with 16 pins. The module itself can adapt to the system of 3.3V and 5V, without the conversion level and this is super convenient!

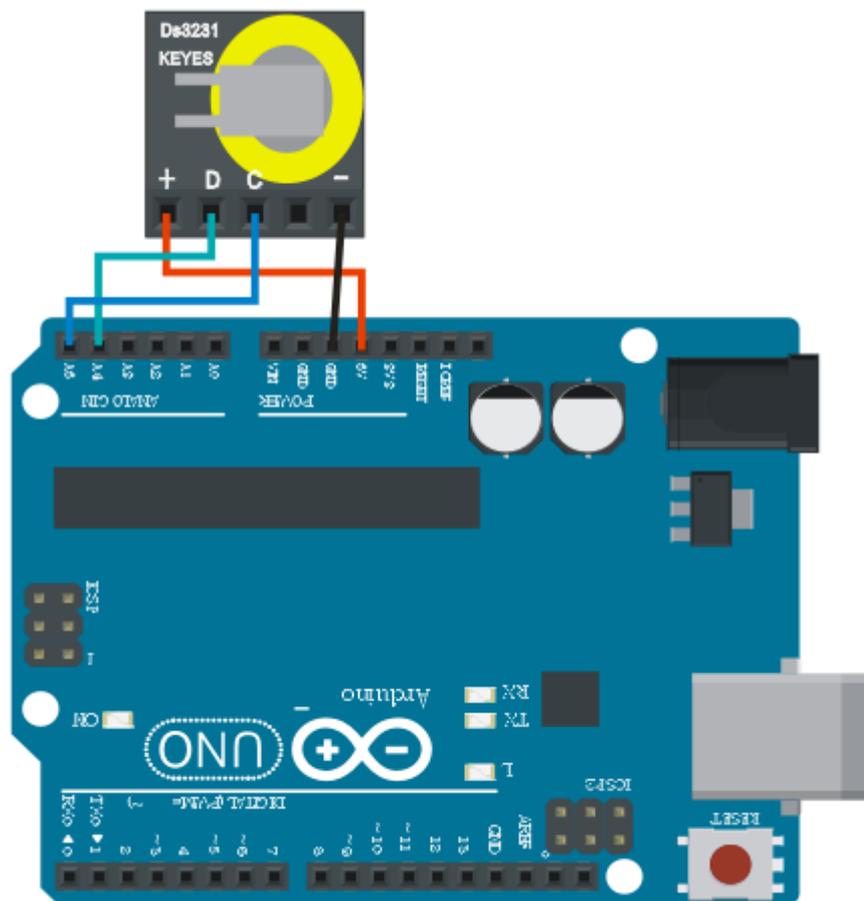
## 2 Feature:

1. -40 to +85 temperature range, the timing accuracy maintained at  $\pm 5\text{ppm}$  ( $\pm 0.432$  seconds / day)
2. provide battery backup for continuous timing
3. low power consumption
4. device package and function compatible with DS3231
5. Complete clock calendar function including seconds and minutes, hour, week, date, month, and year timing, provide leap year compensation until 2100.
- 6 two calendar clock
7. 1Hz and 32.768kHz output
- 8 reset input and button shaking output
- 9 high speed (400kHz), I2C serial bus
10. +2.3V to +5.5V supply voltage
11. digital temperature sensor of which accuracy is  $\pm 3\text{ DEG C}$
12. -40 ~ C to +85 ~ C temperature range
13. 16 pin SO package (300mil)
- 14 by American Association of Underwriters Laboratories (UL) certification

## 3. Module Test

The IIC test methods is used in this module, so we only need to connect D to Arduino A4, C to A5,

+ to VCC, -to GND and that is ok as the follows,



### Test Program Code:

// Date and time functions using a RX8025 RTC connected via I2C and Wire lib

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

```
#include "DS3231.h"
```

```
DS3231 RTC; //Create the DS3231 object
```

```
char weekDay[][4] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };
```

```
//year, month, date, hour, min, sec and week-day(starts from 0 and goes to 6)
```

```
//writing any non-existent time-data may interfere with normal operation of the RTC.
```

```
//Take care of week-day also.
```

```
DateTime dt(2011, 11, 10, 15, 18, 0, 5); //open the series port and you can the time here, and of course you can remand it to be the time you want
```

```
void setup ()
```

```
{
```

```

Serial.begin(57600);//set baud rate to be 57600
Wire.begin();
RTC.begin();
RTC.adjust(dt); //Adjust date-time as defined 'dt' above
}

void loop ()
{
  DateTime now = RTC.now(); //get the current date-time
  Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.date(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
  Serial.print(weekday[now.dayOfWeek()]);
  Serial.println();
  delay(1000);
}

```



Before compiling the code, it is better to put  under the zip file into Arduino libraries, otherwise there might be a lack of library. After finishing the steps above, you can transmit the code to arduino and open the series monitor and you will the result as follow,



When the test is completed, there are other code in the file for you to have a try.