

# Grove - DS1307 RTC (Real Time Clock) for Arduino

## **Table of contents**

Pre-reading

Specifications

Platforms Supported

Getting started

    Play with Arduino

        Hardware

        Software

    Play With Raspberry Pi

        Hardware

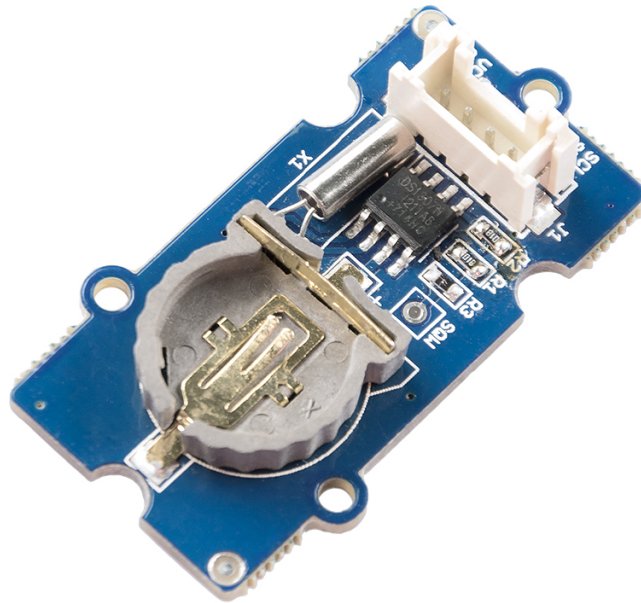
        Software

Schematic Online Viewer

Resources

Projects

Tech Support



The RTC module is based on the clock chip DS1307, which supports the I2C protocol. It utilizes a Lithium cell battery (CR1225). The clock/calendar provides seconds, minutes, hours, day, date, month, and year. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap years. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. And it is valid up to 2100. In order to gain a robust performance, you must put a 3-Volt CR1225 lithium cell in the battery-holder. If you use

the primary power only, the module may not work normally, because the crystal may not oscillate.

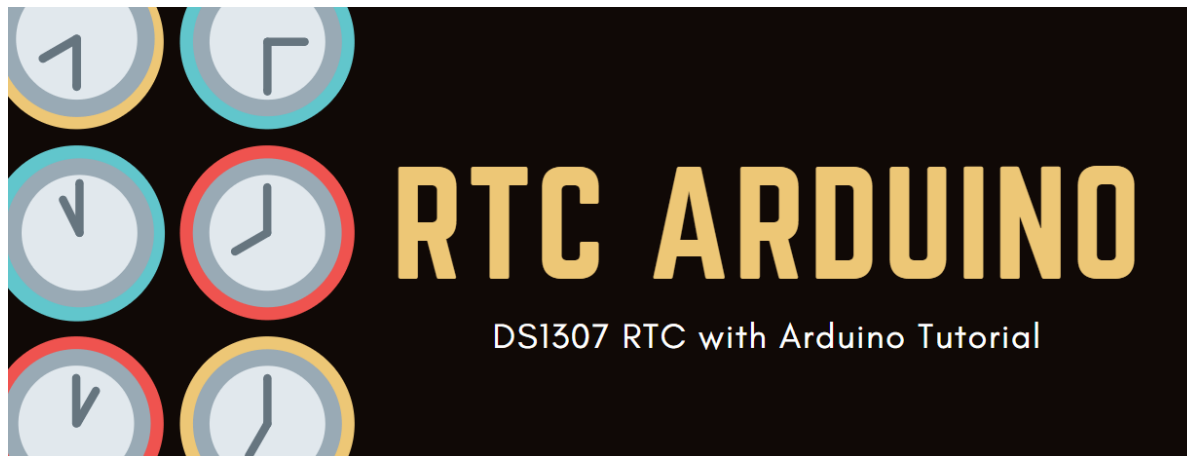


**Note**

The battery is not included.

## Pre-reading

An introduction of RTC for Arduino is strongly recommended reading ahead if you are not familiar with it:



## Specifications

- PCB Size: 2.0cm\*4.0cm
- Interface: 2.0mm pitch pin header
- IO Structure: SCL,SDA,VCC,GND
- ROHS: YES
- VCC: 3.3~5.5V
- Logic High Level Input : 2.2~VCC+0.3 V
- Logic Low Level Input : -0.3~+0.8 V
- Battery Voltage: 2.0~3.5 V



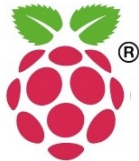
### Tip

More details about Grove modules please refer to [Grove System](#)

## Platforms Supported

Arduino

Raspberry Pi





### Caution


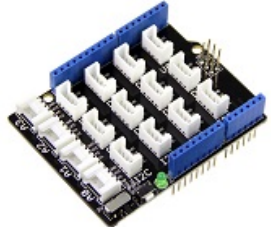

The platforms mentioned above as supported is/are an indication of the module's software or theoretical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

## Getting started

### Play with Arduino

#### Hardware

- **Step 1.** Prepare the below stuffs:

| Seeeduino V4.2   | Base Shield  | Grove-RTC  |
|--|--|--|
|  |  |  |
| <a href="#">Get One Now</a>  | <a href="#">Get One Now</a>  | <a href="#">Get One Now</a>  |

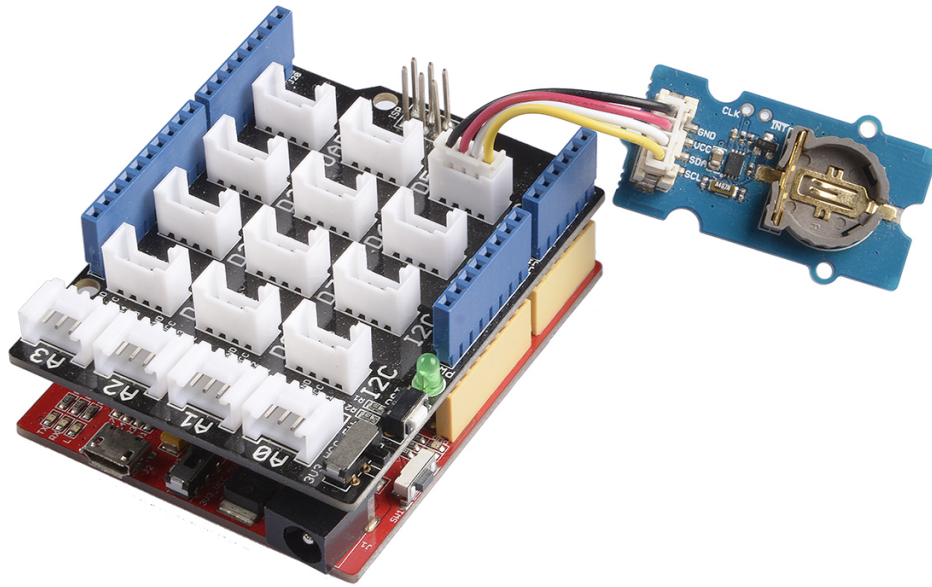
- **Step 2.** Connect Grove-RTC to port **I2C** of Grove-Base Shield.

- **Step 3.** Plug Grove - Base Shield into Seeeduino.
- **Step 4.** Connect Seeeduino to PC via a USB cable.



**Note**

In order to gain a robust performance, you must put a 3-Volt CR1225 lithium cell in the battery-holder. If you use the primary power only, the module may not work normally, because the crystal may not oscillate.



**Note**

If we don't have the base shield, we also can directly connect the Grove-RTC to Arduino board. Please follow below connection.

| Seeeduino_v4 | Grove-RTC |
|--------------|-----------|
| 5V           | VCC       |
| GND          | GND       |
| SDA          | SDA       |
| SCL          | SCL       |

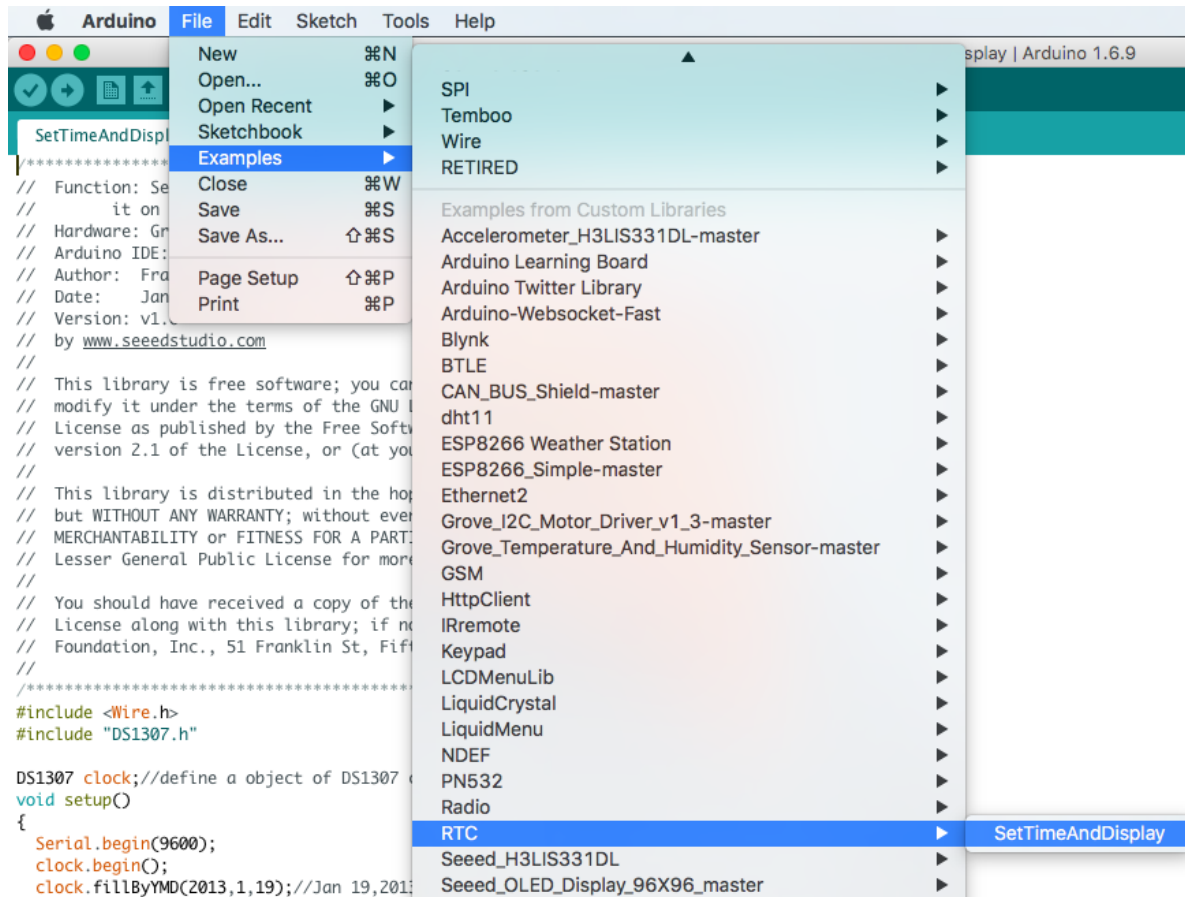
## Software

**Step 1.** Download the [RTC Library](#).

**Step 2.** Refer [How to install library](#) to install library for Arduino.

**Step 3.** Create a new Arduino sketch and paste the codes below to it or open the code directly by the path: **File -> Example ->RTC->SetTimeAndDisplay**.





```
1 #include <Wire.h>
2 #include "DS1307.h"
3
4 DS1307 clock; //define a object of DS1307 class
5 void setup()
6 {
7     Serial.begin(9600);
```

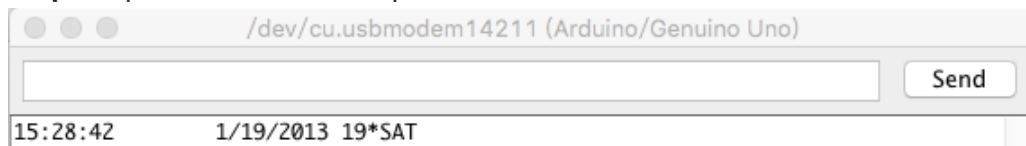
```
8     clock.begin();
9     clock.fillByYMD(2013,1,19); //Jan 19,2013
10    clock.fillByHMS(15,28,30); //15:28 30"
11    clock.fillDayOfWeek(SAT); //Saturday
12    clock.setTime(); //write time to the RTC chip
13 }
14 void loop()
15 {
16     printTime();
17 }
18     /*Function: Display time on the serial monitor*/
19 void printTime()
20 {
21     clock.getTime();
22     Serial.print(clock.hour, DEC);
23     Serial.print(":");
24     Serial.print(clock.minute, DEC);
25     Serial.print(":");
26     Serial.print(clock.second, DEC);
27     Serial.print(" ");
28     Serial.print(clock.month, DEC);
29     Serial.print("/");
30     Serial.print(clock.dayOfMonth, DEC);
31     Serial.print("/");
32     Serial.print(clock.year+2000, DEC);
33     Serial.print(" ");
34     Serial.print(clock.dayOfMonth);
35     Serial.print("*");
36     switch (clock.dayOfWeek) // Friendly printout the weekday
37     {
38         case MON:
39             Serial.print("MON");
40             break;
41         case TUE:
42             Serial.print("TUE");
```

```
43     break;
44     case WED:
45     Serial.print("WED");
46     break;
47     case THU:
48     Serial.print("THU");
49     break;
50     case FRI:
51     Serial.print("FRI");
52     break;
53     case SAT:
54     Serial.print("SAT");
55     break;
56     case SUN:
57     Serial.print("SUN");
58     break;
59 }
60 Serial.println(" ");
61 }
```

**Step 4.** Set the time. Change function arguments to current date/time. Please pay attention to arguments' format.

```
1  clock.fillByYMD(2013,1,19); //Jan 19, 2013
2  clock.fillByHMS(15,28,30); //15:28 30"
3  clock.fillDayOfWeek(SAT); //Saturday
```

**Step 5.** Upload the code and open the serial monitor to receive the sensor's data








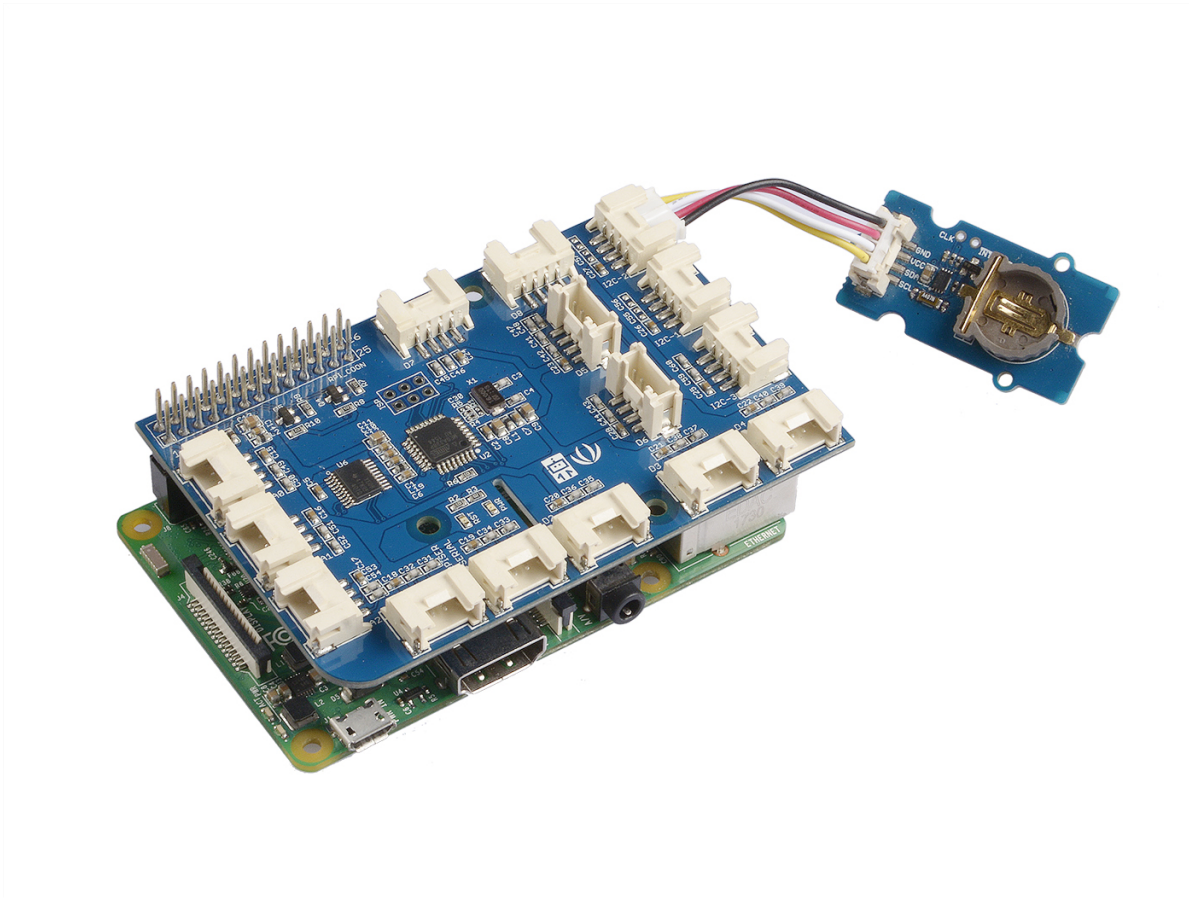
## Play With Raspberry Pi

### Hardware

- **Step 1.** Prepare the below stuffs:

| Raspberry pi  | GrovePi_Plus  | Grove-RTC   |
|---|---|---|
|  |  |  |
| <a href="#">Get One Now</a>   | <a href="#">Get One Now</a>   | <a href="#">Get One Now</a>   |

- **Step 2.** Plug the GrovePi\_Plus into Raspberry.
- **Step 3.** Connect Grove-RTC to **I2C** port of GrovePi\_Plus.
- **Step 4.** Connect the Raspberry to PC through USB cable.



## Software

- **Step 1.** Follow [Setting Software](#) to configure the development environment.
- **Step 2.** Git clone the Github repository.



```
1 cd ~
2 git clone https://github.com/DexterInd/GrovePi.git
```

- **Step 3.** Excute below commands to use this sensor

```
1 cd ~/GrovePi/Software/Python
2 python grove_i2c_rtc.py
```

Here is the code

```
1 #!/usr/bin/env python
2 #
3 # GrovePi Example for using the Grove I2C RTC (https://www.seeedstudio.com
4 #
5 # The GrovePi connects the Raspberry Pi and Grove sensors. You can learn
6 #
7 # Have a question about this example? Ask on the forums here: http://foi
8 #
9 '''
10 ## License
11 The MIT License (MIT)
12 GrovePi for the Raspberry Pi: an open source platform for connecting Gro
13 Copyright (C) 2017 Dexter Industries
14 Permission is hereby granted, free of charge, to any person obtaining a
15 of this software and associated documentation files (the "Software"), to
16 in the Software without restriction, including without limitation the ri
17 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
18 copies of the Software, and to permit persons to whom the Software is
19 furnished to do so, subject to the following conditions:
20 The above copyright notice and this permission notice shall be included
21 all copies or substantial portions of the Software.
```

```
22 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
23 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
24 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
25 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
26 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
27 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
28 THE SOFTWARE.
29 '''
30
31 import time
32 import grovepi
33
34 # Connect the Grove Real Time Clock to any I2C port eg. I2C-1
35 # Can be found at I2C address 0x68
36 # SCL,SDA,VCC,GND
37
38 while True:
39     try:
40         print(grovepi.rtc_getTime())
41         time.sleep(.5)
42
43     except IOError:
44         print ("Error")
45
```

- **Step 4.** Here is the result.



```
pi@raspberrypi: ~/software/GrovePi/Software/Python
pi@raspberrypi ~/software/GrovePi/Software/Python $ sudo python grove_i2c_rtc.py
[0, 12, 43, 22, 3, 5, 15, 5, 4, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]
[0, 12, 43, 23, 3, 5, 15, 5, 4, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]
[0, 12, 43, 24, 3, 5, 15, 5, 4, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]
[0, 12, 43, 24, 3, 5, 15, 5, 4, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]
[0, 12, 43, 25, 3, 5, 15, 5, 4, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]
[0, 12, 43, 25, 3, 5, 15, 5, 4, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]
[0, 12, 43, 26, 3, 5, 15, 5, 4, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]
[0, 12, 43, 27, 3, 5, 15, 5, 4, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]
[0, 12, 43, 27, 3, 5, 15, 5, 4, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]
[0, 12, 43, 28, 3, 5, 15, 5, 4, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]
[0, 12, 43, 28, 3, 5, 15, 5, 4, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]
```

## Demo 2: Grove\_rtc

You can also use this demo to show the time in common. Please create grove\_rtc.py and copy below codes.

```
1      '''
2      * Grove-RTC.py
3      * Demo for Raspberry Pi
4      *
5      * Copyright (c) 2014 seeed technology inc.
6      * Website   : community.seeedstudio.com/
7      * Author    : Lambor
8      * Create Time: Nov 2014
9      * Change Log :
```

```
10      *
11      * The MIT License (MIT)
12      *
13      * Permission is hereby granted, free of charge, to any person obtain
14      * of this software and associated documentation files (the "Software")
15      * in the Software without restriction, including without limitation
16      * to use, copy, modify, merge, publish, distribute, sublicense, and
17      * copies of the Software, and to permit persons to whom the Software
18      * is furnished to do so, subject to the following conditions:
19      *
20      * The above copyright notice and this permission notice shall be
21      * included in all copies or substantial portions of the Software.
22      *
23      * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
24      * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
25      * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
26      * NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
27      * BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
28      * ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
29      * CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
30      * THE SOFTWARE.
31      *
32      * '''
33
34      #!/usr/bin/python
35
36      import time
37      import smbus
38
39      bus = smbus.SMBus(1) # 0 = /dev/i2c-0 (port I2C0), 1 = /dev/i2c-1
40
41      class DS1307():
42          def __init__(self):
43              self.MON = 1
44              self.TUE = 2
45              self.WED = 3
46              self.THU = 4
```

```

45         self.FRI = 5
46         self.SAT = 6
47         self.SUN = 7
48         self.DS1307_I2C_ADDRESS = 0x68
49
50         print 'begin'
51
52     def decToBcd(self, val):
53         return ( (val/10*16) + (val%10) )
54
55     def bcdToDec(self, val):
56         return ( (val/16*10) + (val%16) )
57
58     def begin(self, news):
59         print news
60
61     def startClock(self):
62         bus.write_byte(self.DS1307_I2C_ADDRESS, 0x00)
63         self.second = bus.read_byte(self.DS1307_I2C_ADDRESS) & 0x7f
64         bus.write_byte_data(self.DS1307_I2C_ADDRESS, 0x00, self.sec
65
66         print 'startClock..'
67
68     def stopClock(self):
69         bus.write_byte(self.DS1307_I2C_ADDRESS, 0x00)
70         self.second = bus.read_byte(self.DS1307_I2C_ADDRESS) | 0x80
71         bus.write_byte_data(self.DS1307_I2C_ADDRESS, 0x00, self.sec
72
73         print 'stopClock..'
74
75     def setTime(self):
76         data = [self.decToBcd(self.second), self.decToBcd(self.minu
77                 self.decToBcd(self.hour), self.decToBcd(self.dayOfW
78                 self.decToBcd(self.dayOfMonth), self.decToBcd(self.
79                 self.decToBcd(self.year) ]

```

```
80
81     bus.write_byte(self.DS1307_I2C_ADDRESS, 0x00)
82     bus.write_i2c_block_data(self.DS1307_I2C_ADDRESS, 0x00, data)
83
84     print 'setTime..'
85
86     def getTime(self):
87         bus.write_byte(self.DS1307_I2C_ADDRESS, 0x00)
88         data = bus.read_i2c_block_data(self.DS1307_I2C_ADDRESS, 0x00)
89         #A few of these need masks because certain bits are control
90         self.second = self.bcdToDec(data[0] & 0x7f)
91         self.minute = self.bcdToDec(data[1])
92         self.hour = self.bcdToDec(data[2] & 0x3f) #Need to change
93         self.dayOfWeek = self.bcdToDec(data[3])
94         self.dayOfMonth = self.bcdToDec(data[4])
95         self.month = self.bcdToDec(data[5])
96         self.year = self.bcdToDec(data[6])
97
98         print 'getTime..'
99
100     def fillByHMS(self, _hour, _minute, _second):
101         self.hour = _hour
102         self.minute = _minute
103         self.second = _second
104
105         print 'fillByHMS..'
106
107     def fillByYMD(self, _year, _month, _day):
108         self.year = _year - 2000
109         self.month = _month;
110         self.dayOfMonth = _day
111
112         print 'fillByYMD..'
113
114     def fillDayOfWeek(self, _dow):
```

```
115         self.dayOfWeek = _dow
116
117         print 'fillDayOfWeek..'
118
119     if __name__ == "__main__":
120         clock = DS1307()
121         clock.fillByYMD(2015,3,5)
122         clock.fillByHMS(12,42,30)
123         clock.fillDayOfWeek(clock.THU)
124         clock.setTime()
125         while True:
126             clock.getTime()
127             print clock.hour, ":", clock.minute, ":", \
128                   clock.second, " ", clock.dayOfMonth, "/", \
129                   clock.month, "/", clock.year, " ", "weekday", \
130                   ":", clock.dayOfWeek
131             time.sleep(1)
132
```

Excute below commands to use this sensor

```
sudo python grove_rtc.py
```



Here is the result.

```
pi@raspberrypi: ~/software/Raspi_Grove/Grove_Adapter/Grove_RTC
pi@raspberrypi ~/software/Raspi_Grove/Grove_Adapter/Grove_RTC $ sudo python Grove_RTC.py
begin
fillByYMD..
fillByHMS..
fillDayOfWeek..
setTime..
getTime..
12 : 42 : 30   5 / 3 / 15   weekday : 4
getTime..
12 : 42 : 31   5 / 3 / 15   weekday : 4
getTime..
12 : 42 : 32   5 / 3 / 15   weekday : 4
getTime..
12 : 42 : 33   5 / 3 / 15   weekday : 4
getTime..
12 : 42 : 34   5 / 3 / 15   weekday : 4
getTime..
12 : 42 : 35   5 / 3 / 15   weekday : 4
getTime..
12 : 42 : 36   5 / 3 / 15   weekday : 4
getTime..
12 : 42 : 37   5 / 3 / 15   weekday : 4
getTime..
```

Schematic Online Viewer



## Resources

- **[Eagle]** [Grove-RTC in Eagle format](#)
- **[PDF]** [Grove-RTC Schematic in PDF format](#)

- **[PDF]** [Grove-RTC PCB in PDF format](#)
- **[Library]** [Github repository for RTC](#)
- **[Datasheet]** [DS1307 Datasheet](#)



## Tech Support

Please submit any technical issue into our [forum](#).

