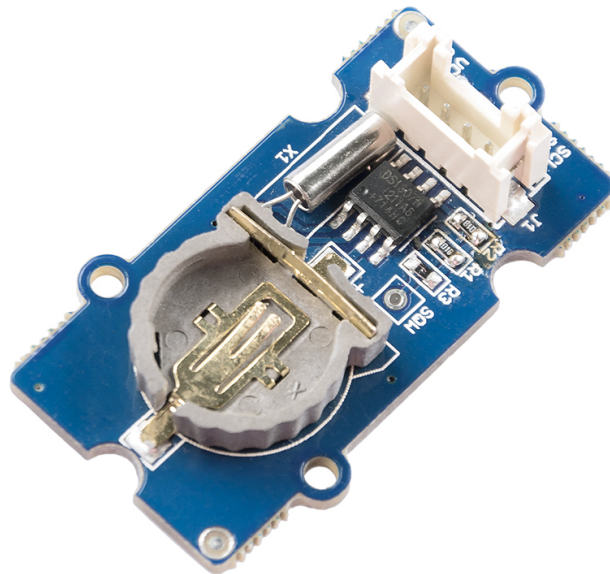


Grove - DS1307 RTC (Real Time Clock) for Arduino



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.

[Get One Now](#) 

[<https://www.seeedstudio.com/Grove-RTC-p-758.html>]

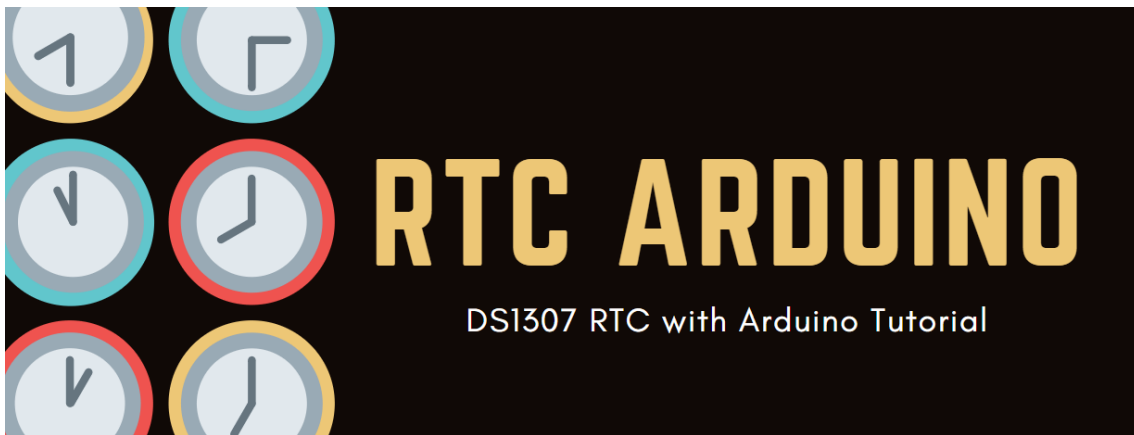


Note

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:



[<https://blog.seeedstudio.com/blog/2019/11/19/arduino-tutorial-using-ds1307-rtc-with-arduino/>]

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

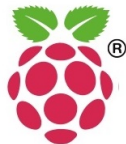
**Tip**

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

[https://wiki.seeedstudio.com/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:

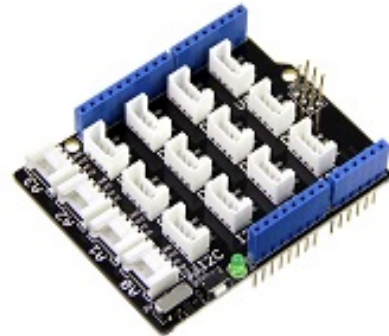
Seeeduino V4.2



[Get One Now](https://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html)

[<https://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html>]

Base Shield



[Get One Now](https://www.seeedstudio.com/Base-Shield-V2-p-1378.html)

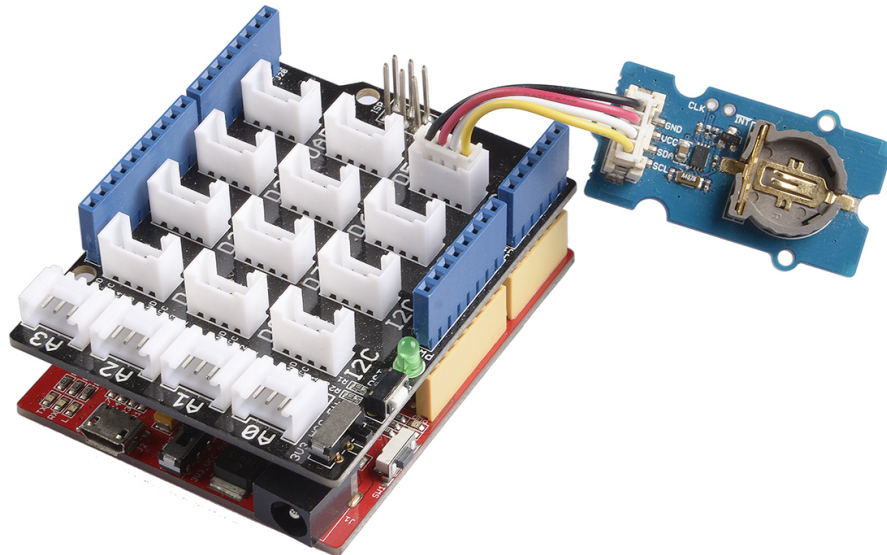
[<https://www.seeedstudio.com/Base-Shield-V2-p-1378.html>]

- **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.

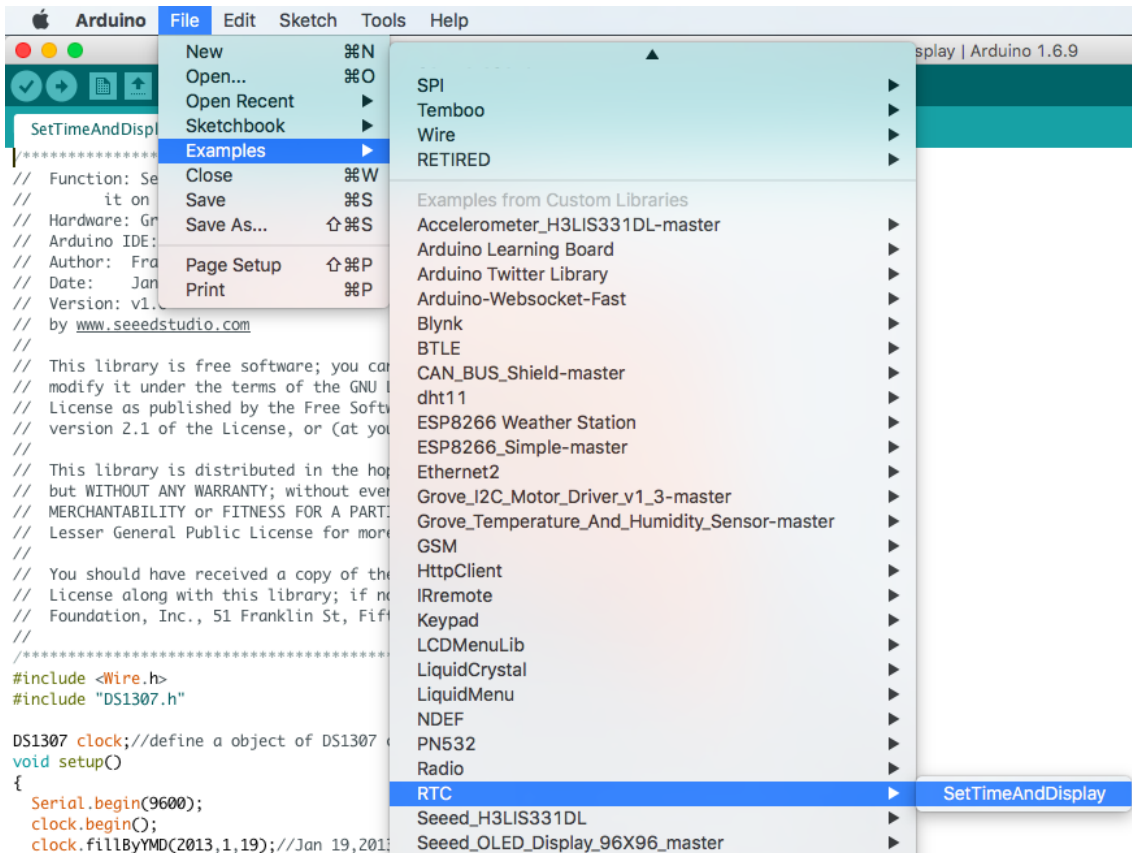
Seeeduino_v4	Grove-RTC
5V	VCC
GND	GND
SDA	SDA
SCL	SCL

Software

Step 1. Download the [RTC Library](https://github.com/Seeed-Studio/RTC_DS1307/archive/master.zip) [https://github.com/Seeed-Studio/RTC_DS1307/archive/master.zip].

Step 2. Refer [How to install library](https://wiki.seeedstudio.com/How_to_install_Arduino_Library) [https://wiki.seeedstudio.com/How_to_install_Arduino_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 {

```

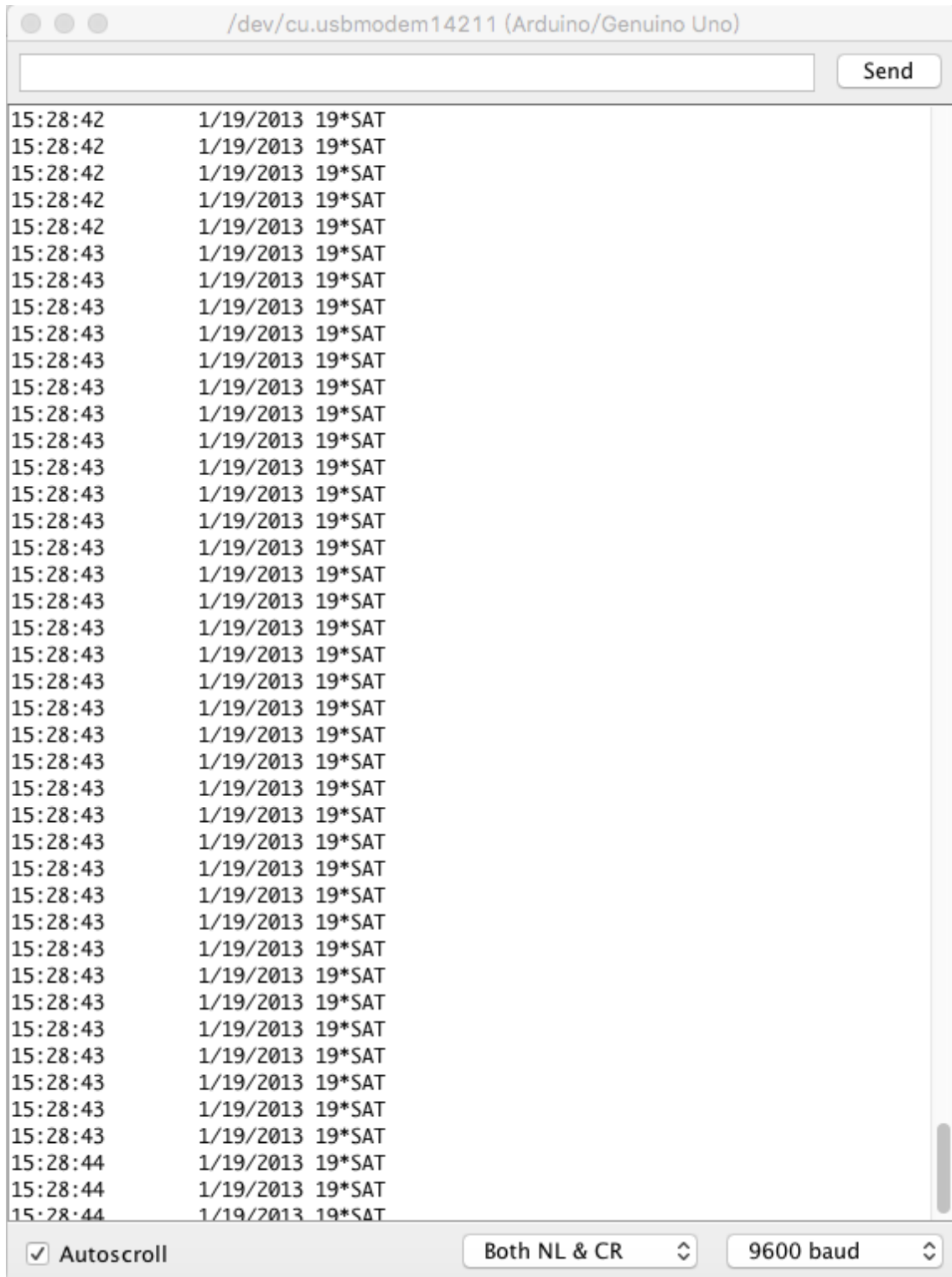
```
21     clock.getTime();
22     Serial.print(clock.hour, DEC);
23     Serial.print(":");
24     Serial.print(clock.minute, DEC);
25     Serial.print(":");
26     Serial.print(clock.second, DEC);
27     Serial.print(" ");
28     Serial.print(clock.month, DEC);
29     Serial.print("/");
30     Serial.print(clock.dayOfMonth, 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 wee.
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



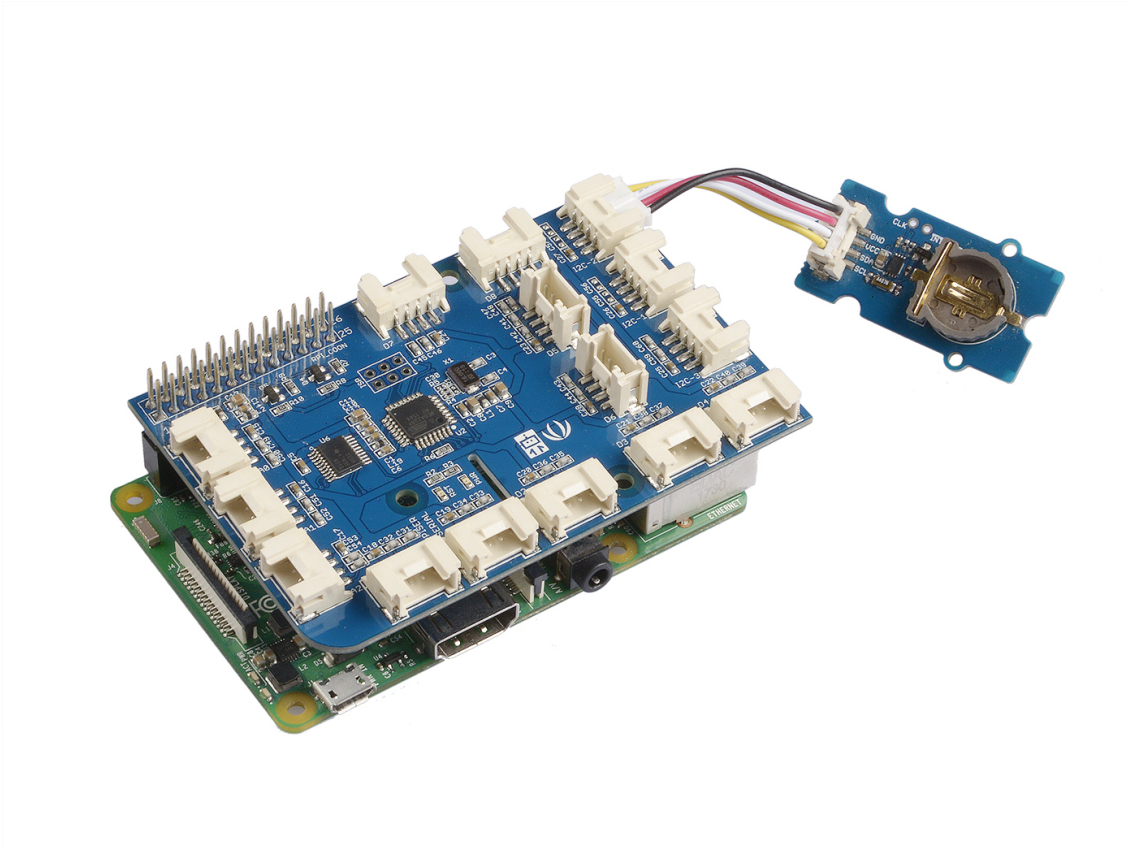
Play With Raspberry Pi

Hardware

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

Raspberry pi	GrovePi_Plus
	
<p>Get One Now [https://www.seeedstudio.com/Raspberry-Pi-3-Model-B-p-2625.html]</p>	<p>Get One Now [https://www.seeedstudio.com/GrovePi-Plus-p-2241.html]</p>

- **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](https://www.dexterindustries.com/GrovePi/get-started-with-the-grovepi/setting-software/) [https://www.dexterindustries.com/GrovePi/get-started-with-the-grovepi/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://wi
4  #
5  # The GrovePi connects the Raspberry Pi and Grove sensor.
6  #
7  # Have a question about this example? Ask on the forums
8  #
9  '''
10 ## License
11 The MIT License (MIT)
12 GrovePi for the Raspberry Pi: an open source platform fo
13 Copyright (C) 2017 Dexter Industries
14 Permission is hereby granted, free of charge, to any per
15 of this software and associated documentation files (the
16 in the Software without restriction, including without li
17 to use, copy, modify, merge, publish, distribute, public
18 copies of the Software, and to permit persons to whom th
19 furnished to do so, subject to the following conditions:
20 The above copyright notice and this permission notice sh
21 all copies or substantial portions of the Software.
22 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF AN
23 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF I
24 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN
25 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAI
26 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTH
27 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
28 THE SOFTWARE.
29 '''
30
31 import time
32 import grovepi
33
34 # Connect the Grove Real Time Clock to any I2C port eg.
35 # Can be found at I2C address 0x68
36 # SCL,SDA,VCC,GND
37
38 while True:
```



```
10      *
11      * The MIT License (MIT)
12      *
13      * Permission is hereby granted, free of charge, to
14      * of this software and associated documentation fi
15      * in the Software without restriction, including w
16      * to use, copy, modify, merge, publish, distribute
17      * copies of the Software, and to permit persons to
18      * furnished to do so, subject to the following con
19      *
20      * The above copyright notice and this permission n
21      * all copies or substantial portions of the Softwa
22      *
23      * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANT
24      * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANT
25      * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT
26      * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY C
27      * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT
28      * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE
29      * THE SOFTWARE.
30      '''
31
32      #!/usr/bin/python
33      import time
34      import smbus
35
36
37      bus = smbus.SMBus(1)      # 0 = /dev/i2c-0 (port I2C0
38
39      class DS1307():
40          def __init__(self):
41              self.MON = 1
42              self.TUE = 2
43              self.WED = 3
44              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)
64         bus.write_byte_data(self.DS1307_I2C_ADDRESS, 0x00, self.second)
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)
71         bus.write_byte_data(self.DS1307_I2C_ADDRESS, 0x00, self.second)
72
73         print 'stopClock..'
74
75     def setTime(self):
76         data = [self.decToBcd(self.second), self.decToBcd(self.hour), self.decToBcd(self.minute),
77                 self.decToBcd(self.dayOfMonth), self.decToBcd(self.month), self.decToBcd(self.year)]
78
79         bus.write_byte(self.DS1307_I2C_ADDRESS, 0x00)
80         bus.write_i2c_block_data(self.DS1307_I2C_ADDRESS, 0x00, data)
81
82         print 'setTime..'
83
84     def getTime(self):
85
86         bus.write_byte(self.DS1307_I2C_ADDRESS, 0x00)
87         data = bus.read_i2c_block_data(self.DS1307_I2C_ADDRESS, 1, 2)
88         #A few of these need masks because certain
89         self.second = self.bcdToDec(data[0] & 0x7f)
90         self.minute = self.bcdToDec(data[1])
```



```
92         self.hour = self.bcdToDec(data[2] & 0x3f)
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, " ",
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



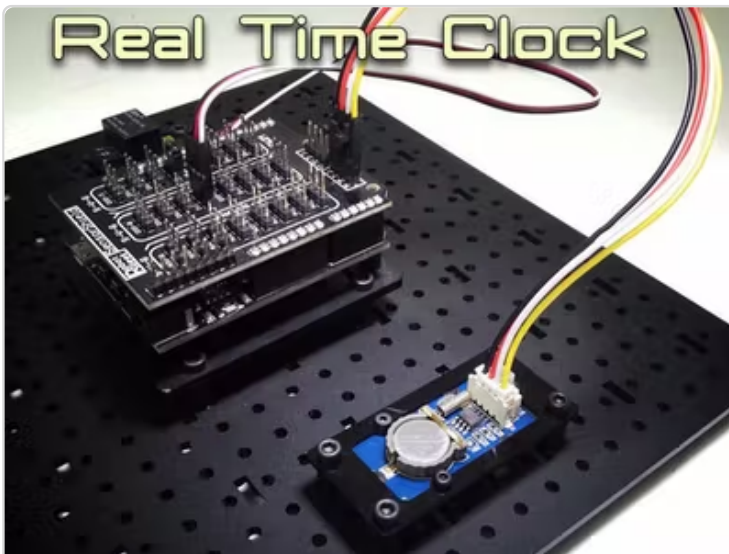
Resources

- **[Eagle]** Grove-RTC in Eagle format
[https://files.seeedstudio.com/wiki/Grove-RTC/res/Real_Time_Clock.zip]
- **[PDF]** Grove-RTC Schematic in PDF format
[<https://files.seeedstudio.com/wiki/Grove-RTC/res/Grove%20-%20RTC%20v1.1%20Sch.pdf>]

- **[PDF]** [Grove-RTC PCB in PDF format](https://files.seeedstudio.com/wiki/Grove-RTC/res/Grove%20-%20RTC%20v1.1%20PCB.pdf)
[<https://files.seeedstudio.com/wiki/Grove-RTC/res/Grove%20-%20RTC%20v1.1%20PCB.pdf>]
- **[Library]** [Github repository for RTC](https://github.com/Seeed-Studio/RTC_DS1307/archive/master.zip) [https://github.com/Seeed-Studio/RTC_DS1307/archive/master.zip]
- **[Datasheet]** [DS1307 Datasheet](https://files.seeedstudio.com/wiki/Grove-RTC/res/DS1307.pdf)
[<https://files.seeedstudio.com/wiki/Grove-RTC/res/DS1307.pdf>]

Projects

Using a Real Time Clock With Arduino: Ever wanted to make a project that activated on a schedule? Using a Real Time Clock (RTC) is a great way to do it!



(<https://www.hackster.io/robotgeek-projects-team/using-a-real-time-clock-with-arduino-6b3896>)

Colorful Clock: A clock that shows your energy level and greets you when you approach



(<https://www.hackster.io/zou-wei/colorful-clock-4cea6b>)

Tech Support

Please submit any technical issue into our [forum](https://forum.seeedstudio.com/) [<https://forum.seeedstudio.com/>].



[<https://www.seeedstudio.com/act-4.html?>

utm_source=wiki&utm_medium=wikibanner&utm_campaign=newpr
oducts]