

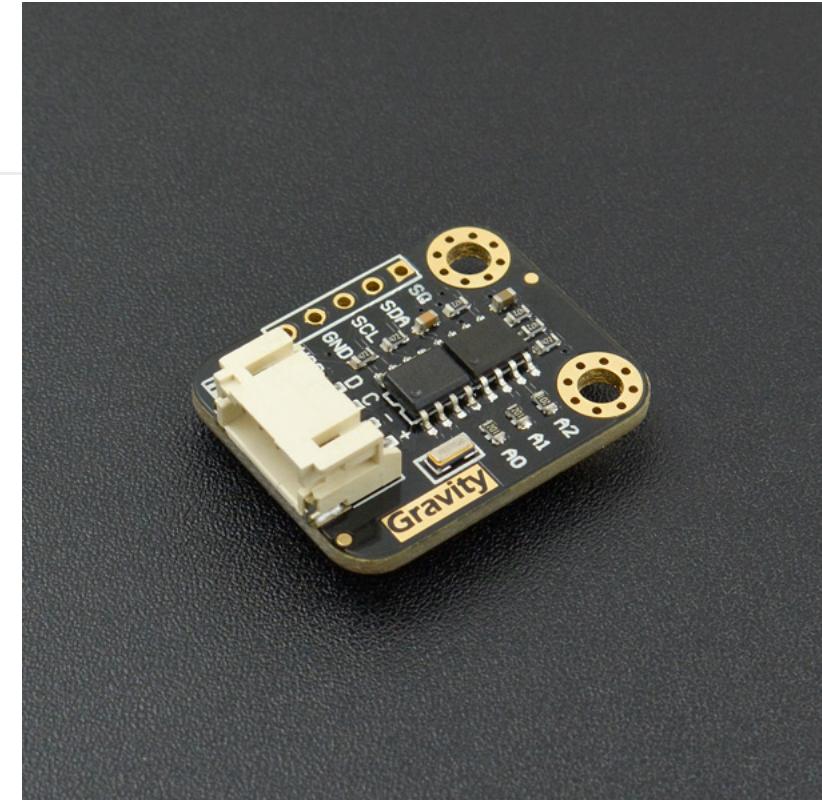
# Gravity I2C DS1307 RTC Module SKU DFR0151

**SKU:DFR0151 (<https://www.dfrobot.com/product-879.html>)**

(<https://www.dfrobot.com/product-879.html>)

## Introduction

After several months "Out of Stock", DS1307 RTC module has a big upgrade. It comes with a Gravity IIC interface, and the size is also reduced to fit miniaturization application. Gravity I2C DS1307 RTC module is a easy-to-use Real Time Clock module. It is cheap and affordable in most applications such as time clock, data logger, etc.. The new version DS1307 RTC module adopts high precision crystal oscillator which reduces time error dramatically. The test conducted shows the time error is only about 1 second in 24 hours (the average time error of Mechanical Watches is about 20s). The module is powered by single cell CR1220 (service life: 3~5 years). It is smaller and keeps EEPROM 4KB ROM to record information related to alarm clocks and events. Besides, you can set the IIC address by resistance adjustment. The Gravity DS1307 RTC module uses Gravity I2C interface and you can plug it on the DFRobot Gravity IO expansion shield directly. Cooperate with Arduino DS1307 library developed by DFRobot, it is easy to realize functions like setting time and time display. It can be widely used in DA (Data Acquisition).

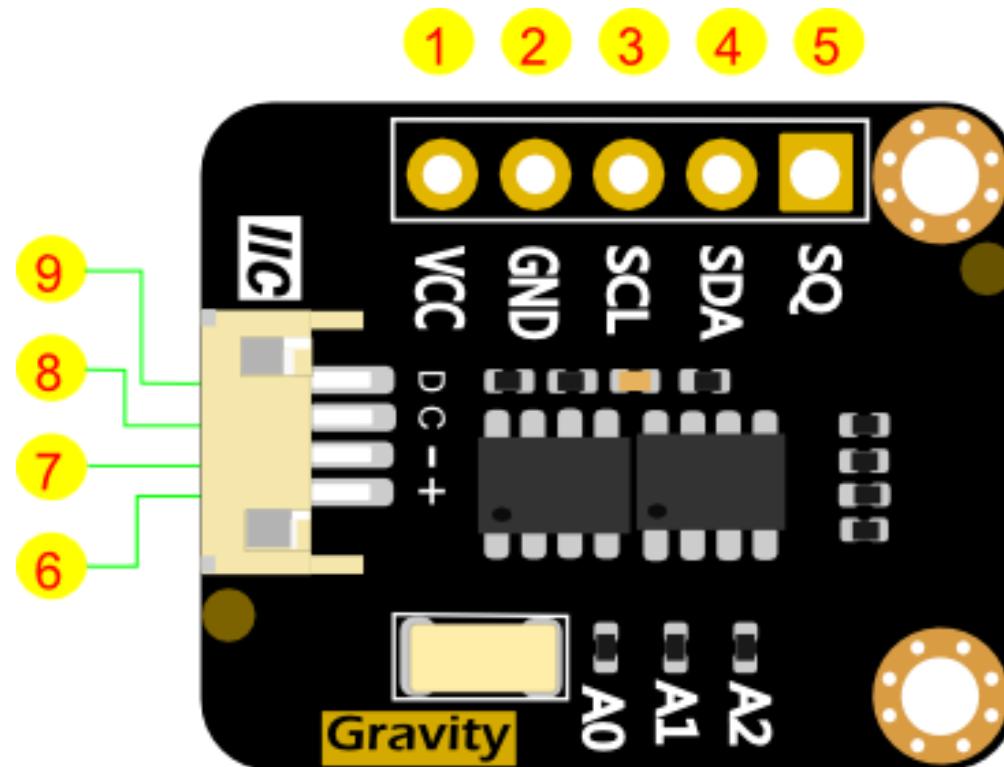


## Specification

- Operating voltage: 5V
- Interface: IIC/I2C (Gravity PH2.0-4P Interface & XH2.54 Pins)

- IC Output Data: Hour, Minutes, Seconds, Week, Day, Month, Year
- Leap-year Compensation
- Accurate calendar up to year 2100
- 1Hz Output Pin
- 56 Bytes of Non-volatile memory available to user
- 4KB ROM of EEPROM
- Size: 0.81inch x 1.06inch (22mm x 27mm)
- Weight: 6g

## Board Overview



Num	Label	Description
1	VCC	5V
2	GND	GND
3	SCL	I2C-Clock
4	SDA	I2C-Data
5	SQ	SQ
6	VCC	5V
7	GND	GND
8	SCL	I2C-Clock
9	SDA	I2C-SDA

&gt;

## Tutorial

In this tutorial, you can set and read time.

### Requirements

- **Hardware**
  - DFRduino UNO (or similar) x 1
  - Gravity: I2C DS1307 RTC Module
  - M-M/F-M/F-F Jumper wires

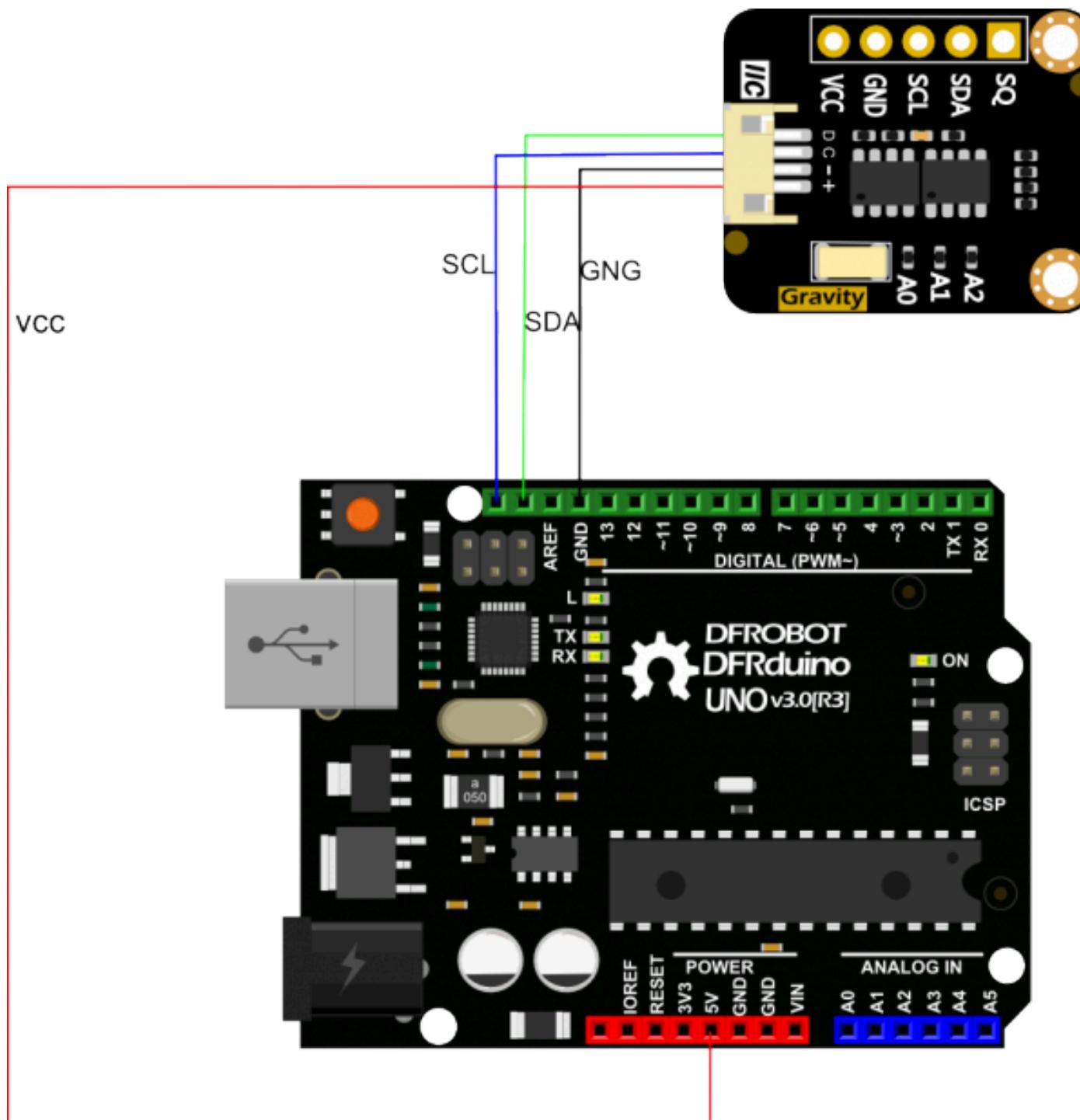
- **Software**

- o Arduino IDE, Click to Download Arduino IDE from Arduino® (<https://www.arduino.cc/en/software>)

>

## Connection Diagram

>



## Sample Code

Please download DFRobot DS1307 RTC library ([https://github.com/DFRobot/DFRobot\\_DS1307](https://github.com/DFRobot/DFRobot_DS1307)). How to install Libraries in Arduino IDE (<https://docs.arduino.cc/software/ide-v1/tutorials/installing-libraries#.UxU8mdzF9H0>)

**Note: #define updata\_flag 0 //1 updata time,0 only show time**

- When flag is 0, the sample code just shows time
- When flag is 1, the sample code only to set time
- Please set time first and modify the value from 1 to 0 to only show time

&gt;

```
/*
 * @file setTimeAndSqw.ino
 * @brief Set time and start timing, set pin sqw output
 * @details Read the time and pin sqw mode
 * @copyright Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @license The MIT License (MIT)
 * @author [qsjhyy](yihuan.huang@dfrobot.com)
 * @version V1.0
 * @date 2021-11-18
 * @url https://github.com/DFRobot/DFRobot_DS1307
 */
#include <DFRobot_DS1307.h>

/* Constructor */
DFRobot_DS1307 DS1307;

void setup()
{
    Serial.begin(115200);

    // Initialize sensor
    while( !(DS1307.begin()) ){
        Serial.println("Communication with device failed, please check connection");
        delay(3000);
    }
    Serial.println("Begin ok!");

    /**

```

```
* According to the set time type, input the value of the corresponding range, set time
* type Time type to be obtained, eTimeType_t:
*   eSEC, eMIN, eHR, eDOW, eDATE, eMTH, eYR
* typeTime According to the selected time type, write the value of the corresponding range
*   When setting eSEC type, range: 00-59
*   When setting eMIN type, range: 00-59
*   When setting eHR type, range: 00-23
*   When setting eDOW type, range: 01-07
*   When setting eDATE type, range: 01-31
*   When setting eMTH type, range: 01-12
*   When setting eYR type, range: 2000-2099
* Note: Values out of range will result in a setting error
*/
DS1307.setTypeTime(DS1307.eYR, 2000);
```

>

```
/**
 * Get the time of one type from rtc module and convert it to uint16_t
* type Time type to be obtained, eTimeType_t:
*   eSEC, eMIN, eHR, eDOW, eDATE, eMTH, eYR
* According to the selected time type, return the value uint16_t of the corresponding range
*   When reading eSEC type, range: 00-59
*   When reading eMIN type, range: 00-59
*   When reading eHR type, range: 00-23
*   When reading eDOW type, range: 01-07
*   When reading eDATE type, range: 01-31
*   When reading eMTH type, range: 01-12
*   When reading eYR type, range: 2000-2099
*/
Serial.print("get type time: ");
```

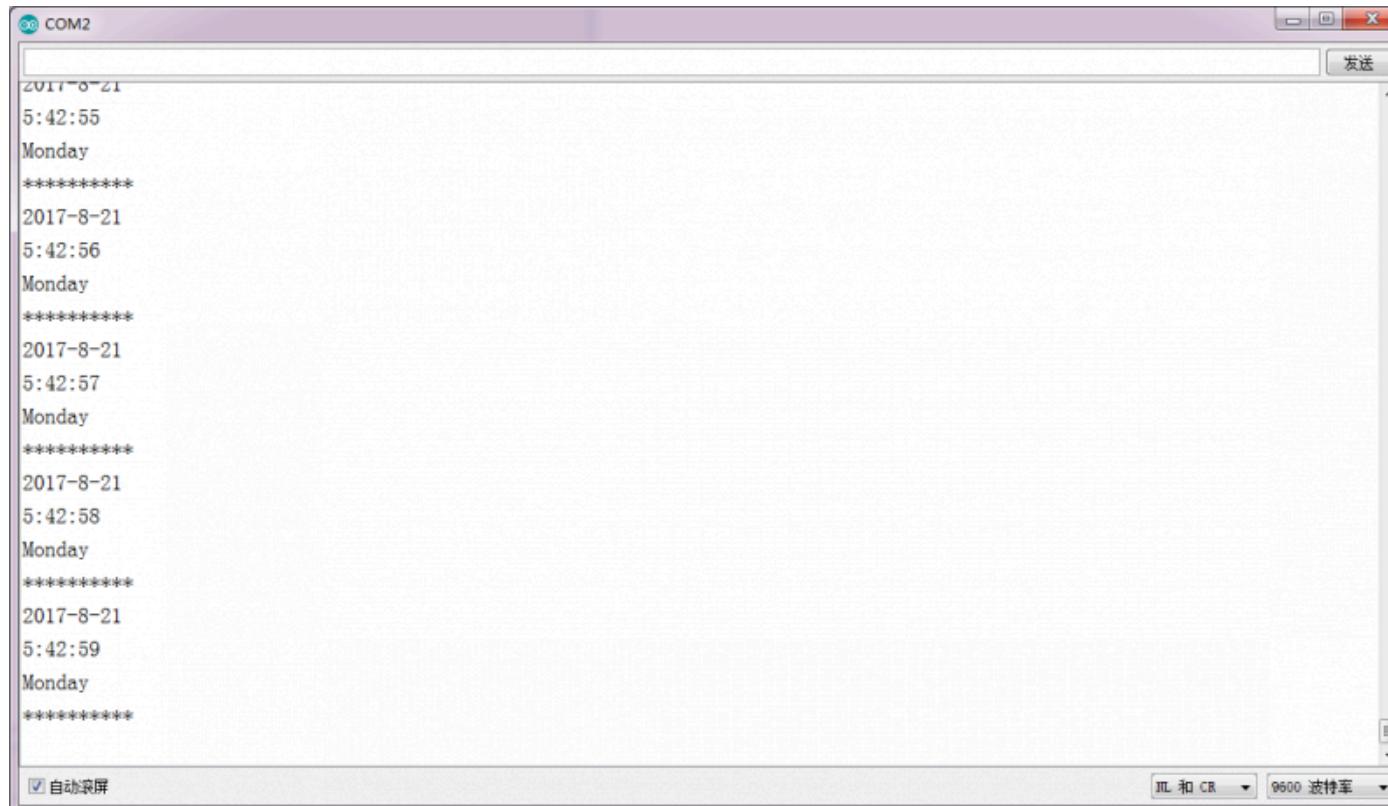
```
Serial.println(DS1307.getTypeTime(DS1307.eYR));  
  
/**  
 * Stop RTC timer function  
 * this bit is part of the seconds byte  
 */  
DS1307.stop();  
/**  
 * According to the available array, set all the time  
 * setTimeBuff Array in the following format (type is uint16_t)  
 * setTimeBuff[0] for eSEC type, range: 00-59  
 * setTimeBuff[1] for eMIN type, range: 00-59  
 * setTimeBuff[2] for eHR type, range: 00-23  
 * setTimeBuff[3] for eDOW type, range: 01-07  
 * setTimeBuff[4] for eDATE type, range: 01-31  
 * setTimeBuff[5] for eMTH type, range: 01-12  
 * setTimeBuff[6] for eYR type, range: 2000-2099  
 * Note: Values out of range will result in a setting error  
 */  
uint16_t setTimeBuff[7] = {5, 1, 7, 6, 9, 9, 2021};  
DS1307.setTime(setTimeBuff);  
/**  
 * Start RTC timing function  
 * this bit is part of the seconds byte  
 */  
DS1307.start();  
  
/**  
 * control the operation of the SQW/OUT pin
```

```
* mode SQW Pin output mode:  
* eSquareWave_LOW, eSquareWave_HIGH, eSquareWave_1Hz,  
* eSquareWave_4kHz, eSquareWave_8kHz, eSquareWave_32kHz  
*/  
DS1307.setSqwPinMode(DS1307.eSquareWave_1Hz);  
  
}  
  
void loop()  
{  
/**  
 * Get the current output mode of SQW/OUT pin  
 * eSqwPinMode_t:  
 * eSquareWave_LOW, eSquareWave_HIGH, eSquareWave_1Hz,  
 * eSquareWave_4kHz, eSquareWave_8kHz, eSquareWave_32kHz  
*/  
if(DS1307.eSquareWave_1Hz == DS1307.getSqwPinMode()){  
    Serial.print("SQW/OUT pin: 1Hz | ");  
}  
  
/**  
 * Get the time from rtc module and convert it to uint16_t  
 * getTimeBuff Array for buffering the obtained time, uint16_t *  
 * getTimeBuff[0] for eSEC type, range: 00-59  
 * getTimeBuff[1] for eMIN type, range: 00-59  
 * getTimeBuff[2] for eHR type, range: 00-23  
 * getTimeBuff[3] for eDOW type, range: 01-07  
 * getTimeBuff[4] for eDATE type, range: 01-31  
 * getTimeBuff[5] for eMTH type, range: 01-12
```

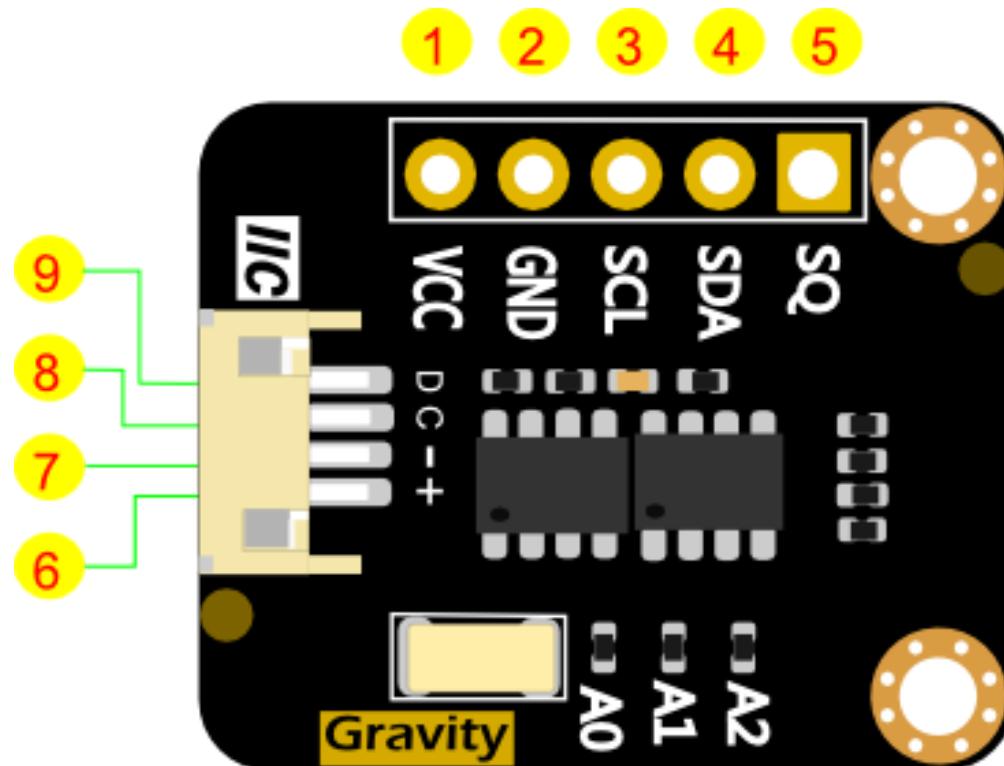
```
*      getTimeBuff[6] for eYR type, range: 2000-2099
*/
uint16_t getTimeBuff[7] = {0};
DS1307.getTime(getTimeBuff);
char outputarr[128];
sprintf(outputarr, "time: %d/%d/%d-%d %d:%d:%d\r\n",
        getTimeBuff[6],
        getTimeBuff[5],
        getTimeBuff[4],
        getTimeBuff[3],
        getTimeBuff[2],
        getTimeBuff[1],
        getTimeBuff[0]
    );
Serial.print(outputarr);

>     delay(1000);
}
```

## Expected Results



## Appendix: EEPROM IIC Address Setting



A0	A1	A2	Add
ON	ON	ON	0x50
OFF	ON	ON	0x51
ON	OFF	ON	0x52
OFF	OFF	ON	0x53
ON	ON	OFF	0x54
OFF	ON	OFF	0x55
ON	OFF	OFF	0x56
OFF	OFF	OFF	0x57

## > FAQ

For any questions, advice or cool ideas to share, please visit the DFRobot Forum (<https://www.dfrobot.com/forum/>).

## More Documents

- DFRobot DS1307 RTC Library ([https://github.com/DFRobot/DFRobot\\_DS1307](https://github.com/DFRobot/DFRobot_DS1307))
- [Schematic & Layout] ([https://github.com/DFRobot/DS1307\\_RTC/raw/master/Schematic & Layout.pdf](https://github.com/DFRobot/DS1307_RTC/raw/master/Schematic%20&%20Layout.pdf))
- DFRobot DS1307 RTC Repostory ([https://github.com/DFRobot/DS1307\\_RTC](https://github.com/DFRobot/DS1307_RTC))
- Time Synchronization with Computer (<http://www.lucadentella.it/en/2013/11/27/rtcsetup/>)

**Back to Top** 

>