

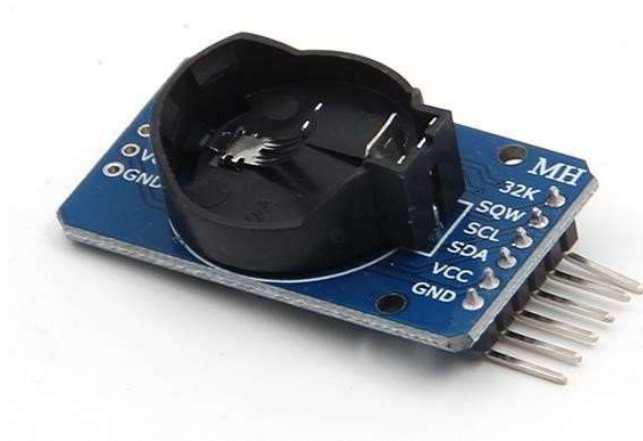


الحمادي للإلكترونيات
ALHAMMADI FOR ELECTRONICS

DS3231 Arduino Clock

The DS3231 is a simple time-keeping chip. The handy thing about it is that there is an integrated battery, so the clock can continue keeping time, even when unplugged.

Set up according to the following image.



Ignore the 32K and SQW pins, you will not need them. Plug the SCL pin into your Arduino Uno A5 port, and the SDA pin into the A4 port. If you are using a newer version of the Arduino than I was, you may have SCL and SDA ports above the AREF port. If you have these ports you can use them instead (with the pins going to their respective ports).. The VCC pin plugs into the 5V port, and the GND plugs into the GND port.

As always, double check your connections before plugging your board into a computer. Improper wiring can damage your sensor.

If you have done everything correctly, when connected to the computer a red light on the clock should light up. If you do not see this light, unplug the clock and check your connections.

Once you are done the clock should be good to go. Chances are it is already running. Open up the serial monitor to check. If it is not working, or you want to set the time, use the following code. You will need to download the DS3231 library first. You can get that here:
<http://m5.img.dxcdn.com/CDDriver/CD/sku.222910.rar>

This will begin the download of a zip file. This contains the original code as well as the needed libraries. Download all the files. Open the DS3231_TEST file and find the files DS3231.cpp, DS3231.H, keywords.txt and Readme.txt. Copy all of these files into a new folder in your Arduino libraries. Name this folder DS3231.

You can just use the code in the file. It is exactly the same as what is listed below, except that I have made a few helpful comments in the code below.

```
/*
```

```
DS3231_test.pde
```

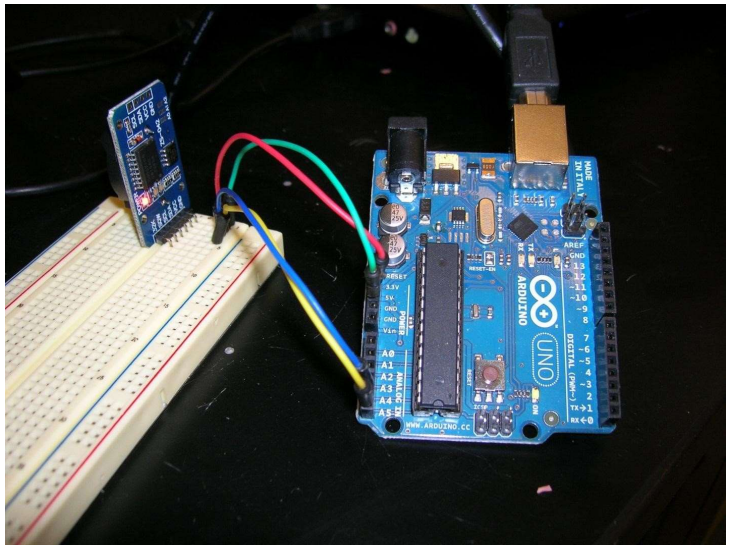
```
Eric Ayars
```

```
4/11
```

```
Test/demo of read routines for a DS3231 RTC.
```

Turn on the serial monitor after loading this to check if things are working as they should.

```
*/  
  
#include <DS3231.h>  
  
#include <Wire.h>  
  
DS3231 Clock; bool Century=false; bool h12;  
bool PM; byte ADay, AHour, AMinute,  
ASecond, ABits; bool ADy, A12h, Apm;  
  
byte year, month, date, DoW, hour, minute,  
second;  
  
void setup() {  
    // Start the I2C interface  
    Wire.begin();  
  
    // VERY IMPORTANT COMMENTS BELOW!!!  
  
    // If you want to set the time, change these numbers to the date and time you want to set to, and then  
    // upload it to the arduino.  
  
    // once you have finished setting the time, comment out the following clock.set functions and then  
    // reupload it to the board. Otherwise your clock will reset every time you open the serial monitor.  
  
    Clock.setSecond(50); //Set the second  
  
    Clock.setMinute(59); //Set the minute  
  
    Clock.setHour(11); //Set the hour  
  
    Clock.setDoW(5); //Set the day of the week
```



```
Clock.setDate(31); //Set the date of the month

Clock.setMonth(5); //Set the month of the year

Clock.setYear(13); //Set the year (Last two digits of the year)

    // Start the serial interface

    Serial.begin(9600);
}

void ReadDS3231()

{

    int second,minute,hour,date,month,year,temperature;

second=Clock.getSecond(); minute=Clock.getMinute();

hour=Clock.getHour(h12, PM); date=Clock.getDate();

month=Clock.getMonth(Century);

year=Clock.getYear();

temperature=Clock.getTemperature();

Serial.print("20");

Serial.print(year,DEC);

Serial.print('-');

Serial.print(month,DEC);

Serial.print('-');

Serial.print(date,DEC);

Serial.print(' ');

Serial.print(hour,DEC);

Serial.print(':');
```

```

Serial.print(minute,DEC);

Serial.print(':');

Serial.print(second,DEC);

Serial.print('\n');

Serial.print("Temperature=");

Serial.print(temperature);

Serial.print('\n');
}

void loop() {ReadDS3231();delay(1000);

    // send what's going on to the serial monitor.

    // Start with the year

/*    Serial.print("2");    if (Century) {                //
Won't need this for 89 years.

        Serial.print("1");

    } else {

        Serial.print("0");

    }

    Serial.print(Clock.getYear(), DEC);

    Serial.print('-');

    // then the month

    Serial.print(Clock.getMonth(Century), DEC);

    Serial.print('-');

    // then the date

    Serial.print(Clock.getDate(), DEC);

    Serial.print(' ');*/

```

```
// and the day of the week
/*Serial.print(Clock.getDoW(), DEC);
Serial.print(' ');*/
// Finally the hour, minute, and second
/*Serial.print(Clock.getHour(h12, PM), DEC);
Serial.print(':');
Serial.print(Clock.getMinute(), DEC);
Serial.print(':');
Serial.print(Clock.getSecond(), DEC);
// Add AM/PM indicator      if (h12) {
        if (PM) {
Serial.print(" PM "); } else {
        Serial.print(" AM ");
        }
    } else {
        Serial.print(" 24h ");
    }
// Display the temperature
Serial.print("T=");
Serial.print(Clock.getTemperature(), 2); //
Tell whether the time is (likely to be) valid if
(Clock.oscillatorCheck()) {
        Serial.print(" O+");
    } else {
```

```

        Serial.print(" O-");
    }*/
    // Indicate whether an alarm went off
    /*if (Clock.checkIfAlarm(1)) {
        Serial.print(" A1!");
    }
    if (Clock.checkIfAlarm(2)) {
        Serial.print(" A2!");
    }*/
    // New line on display
    //Serial.print("\n");
    // delay(1000);
    // Display Alarm 1 information
    /* Serial.print("Alarm 1: ");
        Clock.getA1Time(ADay, AHour, AMinute, ASecond, ABits, ADy, A12h, Apm);
    Serial.print(ADay, DEC);        if (ADy) {
        Serial.print(" DoW");
    } else {
        Serial.print(" Date");
    }
    Serial.print(' ');
    Serial.print(AHour, DEC);
    Serial.print(' ');
    Serial.print(AMinute, DEC); Serial.print('
    ');

```

```
        Serial.print(ASecond, DEC);
Serial.print(' '); if (A12h) {
    if (Apm) {
        Serial.print('pm ');
    } else {
        Serial.print('am ');
    }
}
if (Clock.checkAlarmEnabled(1)) {
    Serial.print("enabled");
}
Serial.print('\n');
// Display Alarm 2 information
Serial.print("Alarm 2: ");
Clock.getA2Time(ADay, AHour, AMinute, ABits, ADy, A12h, Apm);
Serial.print(ADay, DEC);    if (ADy) {
    Serial.print(" DoW");
} else {
    Serial.print(" Date");
}
Serial.print(' ');
Serial.print(AHour, DEC);
Serial.print(' ');
Serial.print(AMinute, DEC);
Serial.print(' '); if (A12h) {
```

```

        if (Apm) {
            Serial.print('pm');
        } else {
            Serial.print('am');
        }
    }
}

if (Clock.checkAlarmEnabled(2)) {
    Serial.print("enabled");
}*/

/* display alarm bits
Serial.print('\n');
Serial.print('Alarm bits: ');
Serial.print(ABits, DEC);
*/

/*
Serial.print('\n');
Serial.print('\n'); delay(1000);

// Display the time once more as a test of the getTime() function
Clock.getTime(year, month, date, DoW, hour, minute, second);

Serial.print(year, DEC);
Serial.print("/");
    Serial.print(month, DEC);
Serial.print("/");

```

```
        Serial.print(date, DEC);  
Serial.print(" day of the week :");  
        Serial.println(DoW, DEC);  
        Serial.print(hour, DEC);  
Serial.print(":");  
        Serial.print(minute, DEC);  
Serial.print(":");  
        Serial.println(second, DEC);*/  
}
```