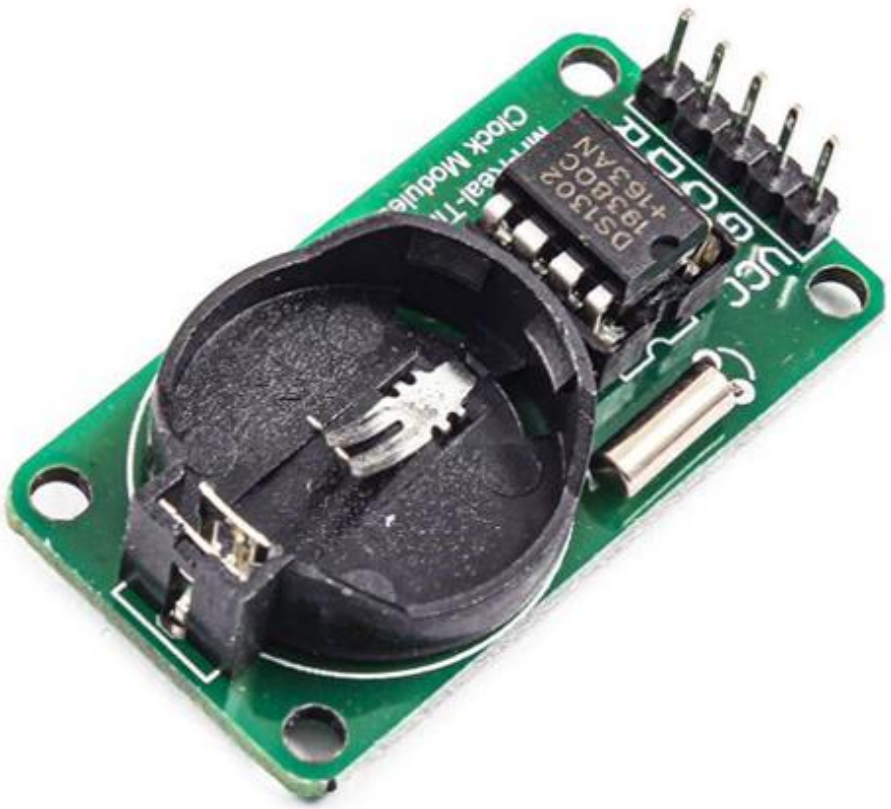
REAL-TIME CLOCK (RTC) MODULE



The DS1302 real time clock module is a cheap module with high accuracy that can be used in different projects. This RTC module provides seconds, minutes, hours, day, date, month, and year information. In this module, date is set automatically based on whether the month is 29, 30 or 31 days and also it is leap year or not. (That's only valid until the year 2100)

Note:

Please be aware that this module does not use I2C communication. Interfacing the DS1302 with a microcontroller is done using a synchronous 3-wire serial communication.

RTC Module Pinout

This module has 5 pins:

- **VCC**: Module power supply – 5V
- **GND**: Ground
- **CLK**: Clock pin
- **DAT**: Data pin
- **RST**: Reset (Must be HIGH for active mode / Active High)

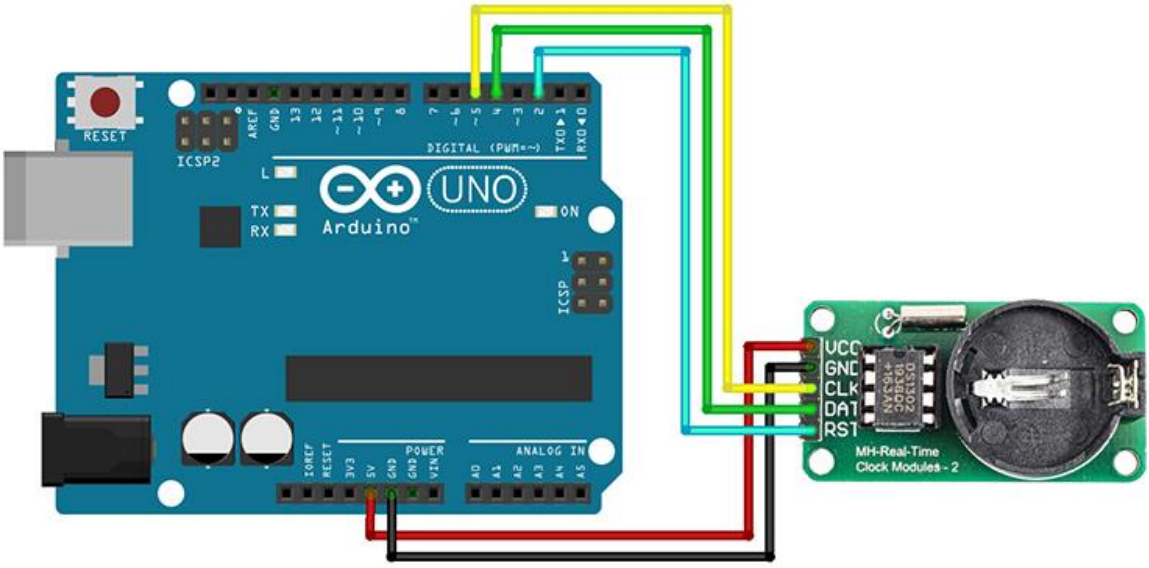
You can see the pinout of this module in the image below.

Interfacing DS1302 RTC Module with Arduino

To establish a connection between Arduino and the DS1302 module, follow these steps:

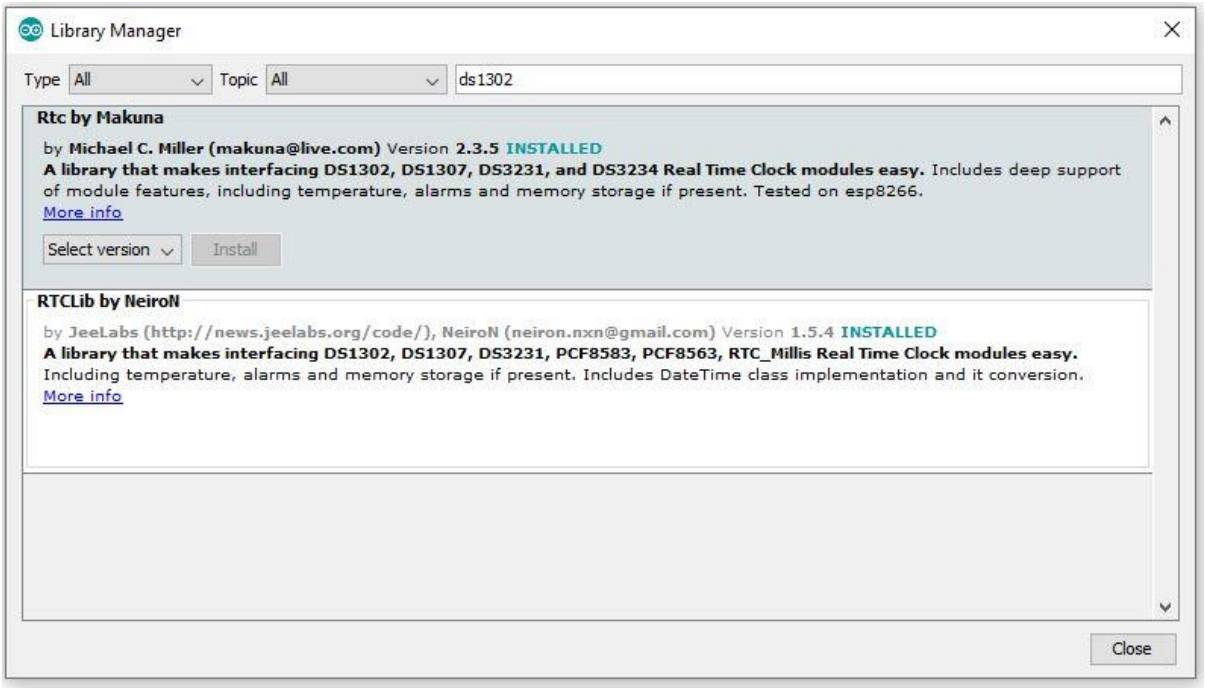
Step 1: Circuit Setup

The following circuit shows how you should connect Arduino to DS1302 module. Connect wires accordingly.



Step 2: Library Installation

Go to **Library manager** and install the **Rtc by Makuna** library.



Step 3: Code Implementation

Upload the following code to Arduino. After that open Serial Monitor.

```
*/



// CONNECTIONS:
// DS1302 CLK/SCLK --> 5
// DS1302 DAT/IO --> 4
// DS1302 RST/CE --> 2
// DS1302 VCC --> 3.3v - 5v
// DS1302 GND --> GND

#include <ThreeWire.h>
#include <RtcDS1302.h>

ThreeWire myWire(4,5,2); // IO, SCLK, CE
RtcDS1302<ThreeWire> Rtc(myWire);

void setup ()
{
    Serial.begin(9600);

    Serial.print("compiled: ");
    Serial.print(__DATE__);
    Serial.println(__TIME__);

    Rtc.Begin();

    RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__);
    printDateTime(compiled);
    Serial.println();
```

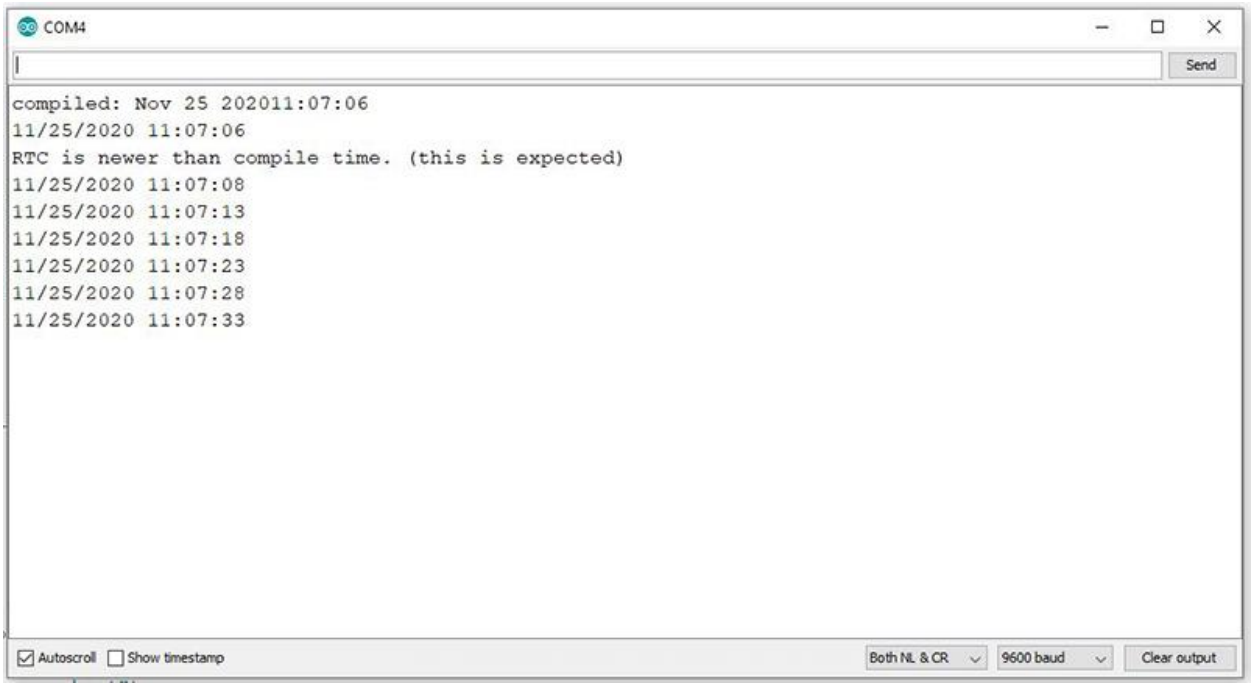
```
    if (!Rtc.IsDateTimeValid())
    {
        // Common Causes:
        //    1) first time you ran and the device wasn't running yet
        //    2) the battery on the device is low or even missing

        Serial.println("RTC lost confidence in the DateTime!");
        Rtc.SetDateTime(compiled);
    }

    if (Rtc.GetIsWriteProtected())
    {
        Serial.println("RTC was write protected, enabling writing
now");
        Rtc.SetIsWriteProtected(false);
    }

    if (!Rtc.GetIsRunning())
    {
        Serial.println("RTC was not actively running, starting now");
        Rtc.SetIsRunning(true);
    }

    RtcDateTime now = Rtc.GetDateTime();
    if (now < compiled)
    {
        Serial.println("RTC is older than compile time!  (Updating
DateTime)");
        Rtc.SetDateTime(compiled);
    }
    else if (now > compiled)
    {
```

```
        Serial.println("RTC is newer than compile time. (this is
expected)");
    }
    else if (now == compiled)
    {
        Serial.println("RTC is the same as compile time! (not expected
but all is fine)");
    }
}

void loop ()
{
    RtcDateTime now = Rtc.GetDateTime();

    printDateTime(now);
    Serial.println();

    if (!now.IsValid())
    {
        // Common Causes:
        //    1) the battery on the device is low or even missing and
the power line was disconnected
        Serial.println("RTC lost confidence in the DateTime!");
    }

    delay(5000); // five seconds
}

#define countof(a) (sizeof(a) / sizeof(a[0]))

void printDateTime(const RtcDateTime& dt)
{
    char datestring[20];
```

```
    snprintf_P(datestring,
            countof(datestring),
            PSTR("%02u/%02u/%04u %02u:%02u:%02u"),
            dt.Month(),
            dt.Day(),
            dt.Year(),
            dt.Hour(),
            dt.Minute(),
            dt.Second() );
    Serial.print(datestring);
}
```

In this code, at first, the time information is given to module as the starting point. Then module starts working and the updated time appears on Serial Monitor every 5 seconds.

Here's the output you can expect to see:



REFERENCE:

https://electropeak.com/learn/interfacing-ds1302-real-time-clock-rtc-module-with-arduino/