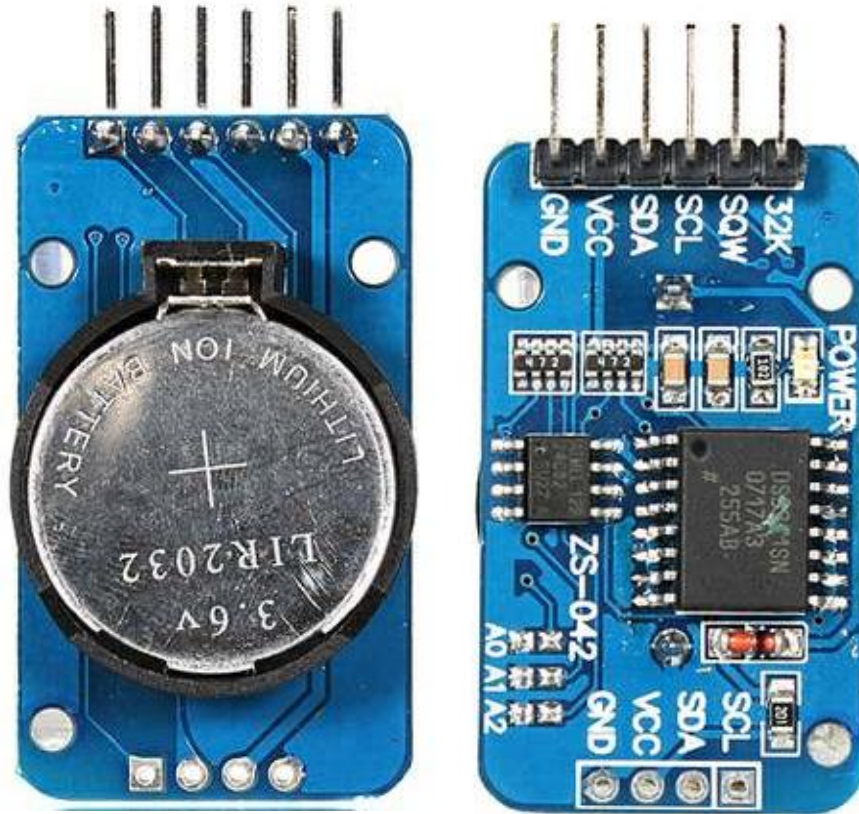


GoldSTEM_Lesson_17_Time_Date_and_Temperature_to_Terminal

tm 3-24-2016

For this experiment we will be using a new component
DS3231 High Precision Real Time Clock



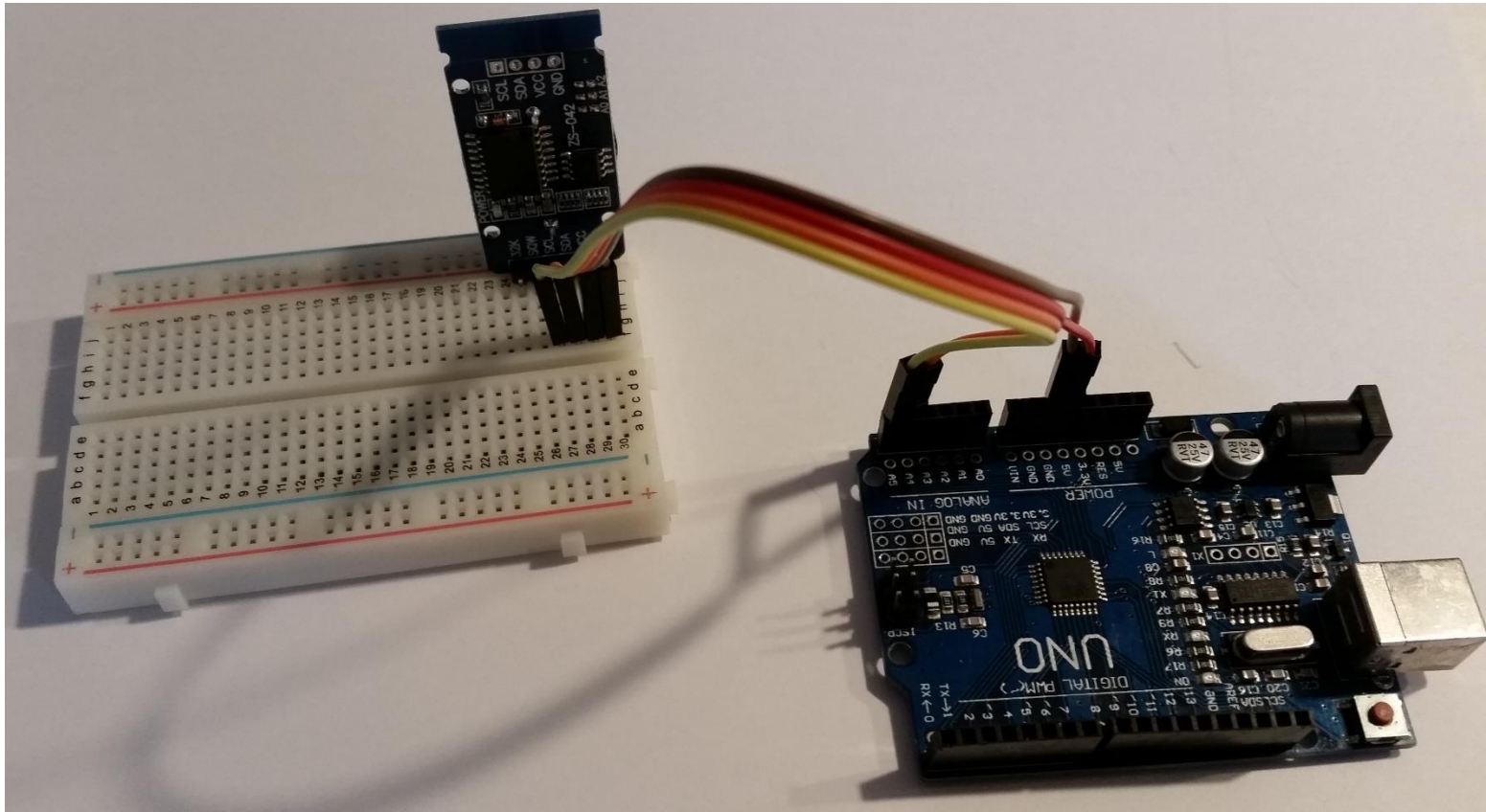
Data Sheet DS3231

<http://datasheets.maximintegrated.com/en/ds/DS3231.pdf>

The first thing we need to do is install the rechargeable Lithium ion battery xxx2032 3.xV.
The lettering on the battery you have may differ from the one in the picture.

While the battery is in the clock module, the module will remember the date and time if the battery is removed the date and time will have to be reprogramed.

The battery will snap into the module.



Wiring the Circuit

Place the Clock module in 25j to 30J GND in 30J

Yellow jumper 27f to UNO ANALOG IN A5

Orange jumper 28J to UNO ANALOG IN A5
Red jumper 29J to UNO POWER 5V
Brown Jumper 30J UNO POWER GND

Loading the Code

GoldSTEM_Lesson_17_Time_Date_and_Temperature_to_Terminal

```
// GoldSTEM_Lesson_17_Time_Date_and_Temperature_to_Terminal      GoldSTEM tm 3-23-2016
```

```
#include <Wire.h>
```

```
#include "ds3231.h"
```

```
#define BUFF_MAX 128
```

```
uint8_t time[8];
```

```
char recv[BUFF_MAX];
```

```
unsigned int recv_size = 0;
```

```
unsigned long prev, interval = 1000;
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  Wire.begin();
```

```
  DS3231_init(DS3231_INTCN);
```

```
  memset(recv, 0, BUFF_MAX);
```

```
// Serial.println("Setting time");
```

```
// parse_cmd("T023413228032016",16);
```

```
}
```

```
void loop()
```

```
{
```

```
  char in;
```

```
char tempF[6];
float temperature;
char buff[BUFF_MAX];
unsigned long now = millis();
struct ts t;

// show time on terminal
if ((now - prev > interval) && (Serial.available() <= 0)) {
    DS3231_get(&t); //Get time
    parse_cmd("C",1);
    temperature = DS3231_get_treg(); //Get temperature
    dtostrf(temperature, 5, 1, tempF);

    Serial.print("Time ");
    Serial.print(t.hour);
    Serial.print(":");
    if(t.min<10)
    {
        Serial.print("0");
    }
    Serial.print(t.min);
    Serial.print(":");
    if(t.sec<10)
    {
        Serial.print("0");
    }
    Serial.print(t.sec);
    Serial.print(' ');

    Serial.print("Date ");
    Serial.print(t.mon);
    Serial.print("-");
```

```
Serial.print(t.mday);
Serial.print("-");
Serial.print(t.year);

Serial.print(" Temp ");
Serial.print(tempF);
Serial.print((char)176);
Serial.println("C ");
prev = now;
}

if (Serial.available() > 0) {
  in = Serial.read();

  if ((in == 10 || in == 13) && (recv_size > 0)) {
    parse_cmd(recv, recv_size);
    recv_size = 0;
    recv[0] = 0;
  } else if (in < 48 || in > 122) { // ignore ~[0-9A-Za-z]
  } else if (recv_size > BUFF_MAX - 2) { // drop lines that are too long
    // drop
    recv_size = 0;
    recv[0] = 0;
  } else if (recv_size < BUFF_MAX - 2) {
    recv[recv_size] = in;
    recv[recv_size + 1] = 0;
    recv_size += 1;
  }
}
}
```

```

void parse_cmd(char *cmd, int cmdsize)
{
    uint8_t i;
    uint8_t reg_val;
    char buff[BUFF_MAX];
    struct ts t;

    //snprintf(buff, BUFF_MAX, "cmd was '%s' %d\n", cmd, cmdsize);
    //Serial.print(buff);
    // TssmmhhWDDMMYYYY aka set time

    if (cmd[0] == 84 && cmdsize == 16) {
        //T355720619112011
        t.sec = inp2toi(cmd, 1);
        t.min = inp2toi(cmd, 3);
        t.hour = inp2toi(cmd, 5);
        t.wday = inp2toi(cmd, 7);
        t.mday = inp2toi(cmd, 8);
        t.mon = inp2toi(cmd, 10);
        t.year = inp2toi(cmd, 12) * 100 + inp2toi(cmd, 14);
        DS3231_set(t);

    } else if (cmd[0] == 49 && cmdsize == 1) { // "1" get alarm 1
        DS3231_get_a1(&buff[0], 59);
        Serial.println(buff);
    } else if (cmd[0] == 50 && cmdsize == 1) { // "2" get alarm 1
        DS3231_get_a2(&buff[0], 59);
        Serial.println(buff);
    } else if (cmd[0] == 51 && cmdsize == 1) { // "3" get aging register
        Serial.print("aging reg is ");
        Serial.println(DS3231_get_aging(), DEC);
    } else if (cmd[0] == 65 && cmdsize == 9) { // "A" set alarm 1

```

```

DS3231_set_creg(DS3231_INTCN | DS3231_A1IE);
//ASSMMHHDD
for (i = 0; i < 4; i++) {
    time[i] = (cmd[2 * i + 1] - 48) * 10 + cmd[2 * i + 2] - 48; // ss, mm, hh, dd
}
byte flags[5] = { 0, 0, 0, 0, 0 };
DS3231_set_a1(time[0], time[1], time[2], time[3], flags);
DS3231_get_a1(&buff[0], 59);
Serial.println(buff);
} else if (cmd[0] == 66 && cmdsize == 7) { // "B" Set Alarm 2
    DS3231_set_creg(DS3231_INTCN | DS3231_A2IE);
    //BMMHHDD
    for (i = 0; i < 4; i++) {
        time[i] = (cmd[2 * i + 1] - 48) * 10 + cmd[2 * i + 2] - 48; // mm, hh, dd
    }
    byte flags[5] = { 0, 0, 0, 0, 0 };
    DS3231_set_a2(time[0], time[1], time[2], flags);
    DS3231_get_a2(&buff[0], 59);
    Serial.println(buff);
} else if (cmd[0] == 67 && cmdsize == 1) { // "C" - get temperature register

} else if (cmd[0] == 68 && cmdsize == 1) { // "D" - reset status register alarm flags
    reg_val = DS3231_get_sreg();
    reg_val &= B11111100;
    DS3231_set_sreg(reg_val);
} else if (cmd[0] == 70 && cmdsize == 1) { // "F" - custom fct
    reg_val = DS3231_get_addr(0x5);
    Serial.print("orig ");
    Serial.print(reg_val, DEC);
    Serial.print("month is ");
    Serial.println(bcdtodec(reg_val & 0x1F), DEC);
} else if (cmd[0] == 71 && cmdsize == 1) { // "G" - set aging status register

```

```

    DS3231_set_aging(0);
} else if (cmd[0] == 83 && cmdsize == 1) { // "S" - get status register
    Serial.print("status reg is ");
    Serial.println(DS3231_get_sreg(), DEC);
} else {
    Serial.print("unknown command prefix ");
    Serial.println(cmd[0]);
    Serial.println(cmd[0], DEC);
}
}

```

If this is the first time the battery was placed in the clock module the date and time must be set.

To accomplish this un comment the following line

```

// parse_cmd("T023413228032016",16); about the 15 line down from the top, backspace the // so the line looks like this
parse_cmd("T023413228032016",16);
// TssmmhhWDDMMYYYY aka set time

```

Change the sequence of numbers as follows

02 Seconds set this to 02 it will take about two seconds to compile and load the code.

34 Minutes set this for one minute past what your system clock says

13 Hours in military time 1-24

2 Day of week 1= Sunday

28 Day of month

03 Month of year

2016 Year

When the system clock matches the appropriate minute Click the Upload button on the program page.

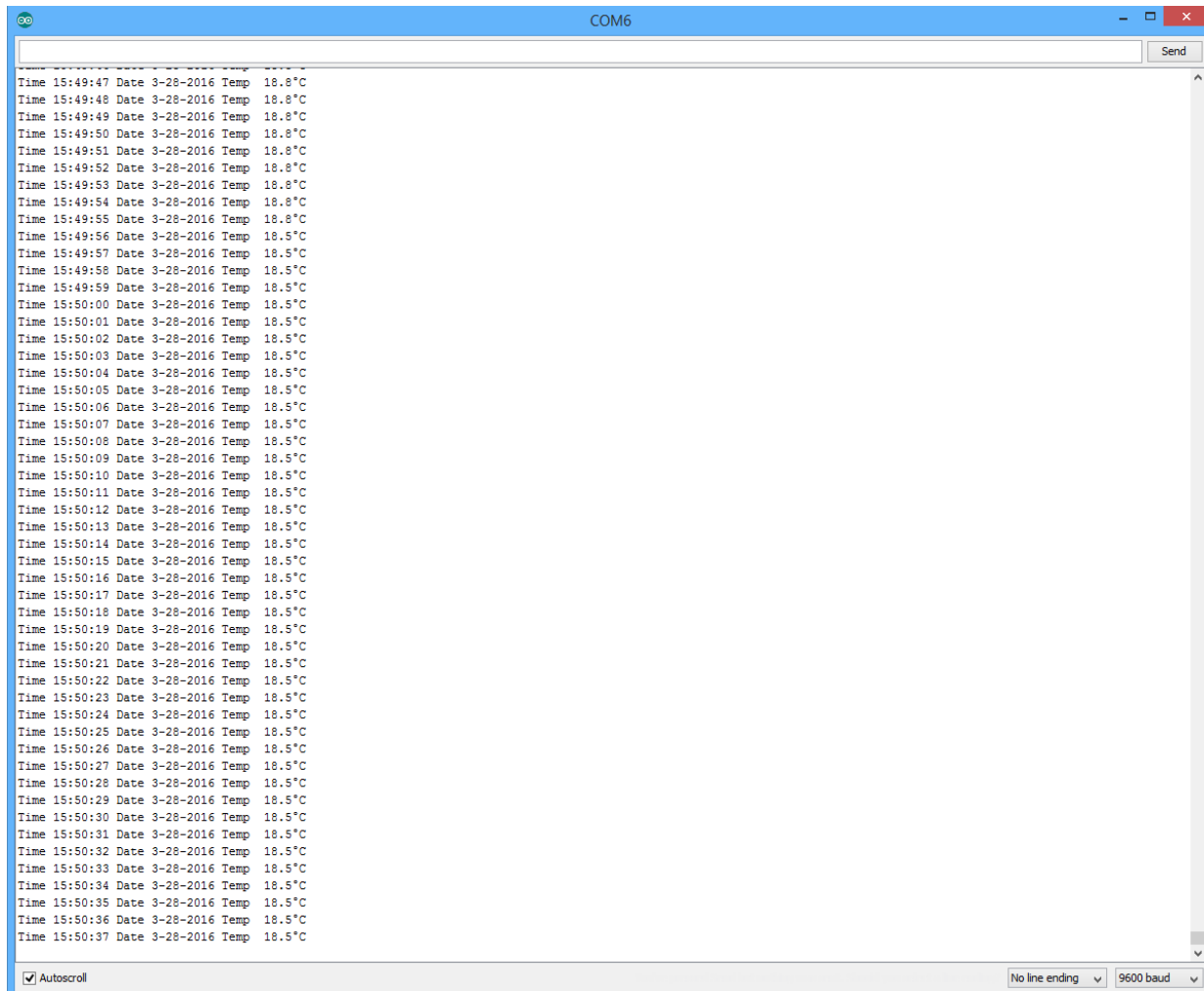
When the code is loaded click the Serial Monitor button.

Check the time if it matches the system time **comment out** the parse_cmd, otherwise try to set the time again.

Output to monitor

To see the Monitor output click on the Serial Monitor Icon, on the Sketch page it looks like a magnifying glass on the right side of the page.

This is what you should see.



Ok is it working?

End of Lesson