

```

#include <Wire.h>
#include <I2C_RTC.h>
int sqw_pin_9 = 9 ;
int sqw_pin_10 = 10 ;
int clockPin = 12;
int dataPin = 11;
int j[10] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90};
int time_loop = 0;

static DS3231 RTC;

void setup()
{
    pinMode(clockPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(sqw_pin_9,OUTPUT);
    pinMode(sqw_pin_10,OUTPUT);
    Serial.begin(9600);
    RTC.begin();

    RTC.setHourMode(CLOCK_H12);

    //RTC.setHourMode(CLOCK_H24);

    if (RTC.getHourMode() == CLOCK_H12)
    {
        //RTC.setMeridiem(HOUR_PM);
    }

    //RTC.setDay(22);
    // RTC.setMonth(5);
    // RTC.setYear(2020);
    //RTC.setHours(12);
    ///RTC.setMinutes(24);
    //RTC.setSeconds(34);

    RTC.setWeek(1);
    RTC.enableSqwPin();

    //RTC.setDate(22,07,29);
    //RTC.setTime(22,10,20);
}

void loop()
{
    for(int time_loop = 0; time_loop < 30; time_loop++){

```

```

delay(1000);
while (Serial.available() > 0) {
String teststr_h_0_a = Serial.readString(); //read until timeout

    Serial.println("WELCOME TO CLOCK ADJUST MENU");
    Serial.println("PLEASE ENTER HOURS (e.g. 01 for AM and 13 for PM)");
while (Serial.available() == 0) {} //wait for data available
    String teststr_h_0 = Serial.readString(); //read until timeout
    Serial.println("Hours: ");
    Serial.print(teststr_h_0);
    RTC.setHours(teststr_h_0.toInt());

Serial.println("PLEASE ENTER MINUTES");
while (Serial.available() == 0) {} //wait for data available
    String teststr_m_0 = Serial.readString(); //read until timeout
    Serial.println("Min: ");
    Serial.println(teststr_m_0);
    RTC.setMinutes(teststr_m_0.toInt());

Serial.println("PLEASE ENTER SECONDS");
while (Serial.available() == 0) {} //wait for data available
    String teststr_s_0 = Serial.readString(); //read until timeout
    Serial.println("sec: ");
    Serial.print(teststr_s_0);
    RTC.setSeconds(teststr_s_0.toInt());

    Serial.println("PLEASE ENTER DATE");
while (Serial.available() == 0) {} //wait for data available
    String teststr_D_0 = Serial.readString(); //read until timeout
    Serial.println("Date: ");
    Serial.print(teststr_D_0);
    RTC.setDay(teststr_D_0.toInt());

Serial.println("PLEASE ENTER MONTH");
while (Serial.available() == 0) {} //wait for data available
    String teststr_M_0 = Serial.readString(); //read until timeout
    Serial.println("Month: ");
    Serial.print(teststr_M_0);
    RTC.setMonth(teststr_M_0.toInt());

Serial.println("PLEASE ENTER YEAR");
while (Serial.available() == 0) {} //wait for data available
    String teststr_Y_0 = Serial.readString(); //read until timeout
    Serial.println("YEAR: ");
    Serial.print(teststr_Y_0);
    RTC.setYear(teststr_Y_0.toInt());

time_loop = 0;
main_clock();

```

```
}  
}
```

```
time_loop = 0;  
main_clock();
```

```
}
```

```
void main_clock()
```

```
{
```

```
    switch (RTC.getWeek())  
    {
```

```
        case 1:  
            Serial.print("SUN");  
            break;
```

```
        case 2:  
            Serial.print("MON");  
            break;
```

```
        case 3:  
            Serial.print("TUE");  
            break;
```

```
        case 4:  
            Serial.print("WED");  
            break;
```

```
        case 5:  
            Serial.print("THU");  
            break;
```

```
        case 6:  
            Serial.print("FRI");  
            break;
```

```
        case 7:  
            Serial.print("SAT");  
            break;
```

```
    }
```

```
    Serial.print(" ");
```

```
    Serial.print(RTC.getDay());
```

```
    Serial.print("-");
```

```
    Serial.print(RTC.getMonth());
```

```
    Serial.print("-");
```

```
    Serial.print(RTC.getYear());
```

```

Serial.print(" ");

Serial.print(RTC.getHours());
Serial.print(":");
Serial.print(RTC.getMinutes());
Serial.print(":");
Serial.print(RTC.getSeconds());
Serial.print(" ");

if (RTC.getHourMode() == CLOCK_H12)
{
    if(RTC.getMeridiem() == HOUR_AM)
        Serial.println(" AM");
    if (RTC.getMeridiem() == HOUR_PM)
        Serial.println(" PM");
}

for (int time_loop = 0; time_loop < 11; time_loop++) {
int count_1 = RTC.getHours() / 10 ; // 1 o/p
int count_1_r = RTC.getHours() % 10 ;// 2 o/p
int count_2 = RTC.getMinutes() / 10 ; // 1 o/p
int count_3 = RTC.getMinutes() % 10 ;// 2 o/p

digitalWrite(clockPin, LOW);
shiftOut(dataPin, clockPin, MSBFIRST, j[count_3]);
//delay(10);
digitalWrite(clockPin, LOW);
shiftOut(dataPin, clockPin, MSBFIRST, j[count_2]);
//delay(10);
digitalWrite(clockPin, LOW);
shiftOut(dataPin, clockPin, MSBFIRST, j[count_1_r]);
//delay(10);
digitalWrite(clockPin, LOW);
shiftOut(dataPin, clockPin, MSBFIRST, j[count_1]);
//delay(10);
delay(1000);
digitalWrite(sq_w_pin_9, !digitalRead(sq_w_pin_9));
digitalWrite(sq_w_pin_10, !digitalRead(sq_w_pin_10));
// delay(1);

}

// -----
// -- CLOCK ENDS HERE -----
//-----

```

```

// -----
// -- DISPLAY TEMPERATURE HERE -----
//-----
time_loop = 0;
for (int time_loop = 0; time_loop < 11; time_loop++) {
Serial.print(RTC.getTemp());
float temp_now = RTC.getTemp();
int count = temp_now*100;
int count_3 = count/1000 ; // display it 9

int count_3_r = count % 1000 ; //812
//int count_2 = count_3_r;
int count_2 = count_3_r / 100 ; // display it 8
int count_2_r = count_3_r % 100; // 12
//int count_1 = count_2_r ;
int count_1 = count_2_r / 10 ; // 1 o/p
int count_1_r = count_2_r % 10 ;// 2 o/p

digitalWrite(clockPin, LOW);
shiftOut(dataPin, clockPin, MSBFIRST, j[count_1_r]);
//delay(10);
digitalWrite(clockPin, LOW);
shiftOut(dataPin, clockPin, MSBFIRST, j[count_1]);
//delay(10);
digitalWrite(clockPin, LOW);
shiftOut(dataPin, clockPin, MSBFIRST, j[count_2]);
//delay(10);
digitalWrite(clockPin, LOW);
shiftOut(dataPin, clockPin, MSBFIRST, j[count_3]);
//delay(10);
digitalWrite(sqw_pin_9, HIGH);
digitalWrite(sqw_pin_10, LOW);
delay(1000);
}
// -----
// -- TEMPERATURE ENDS HERE -----
//-----

time_loop = 0;
for (int time_loop = 0; time_loop < 11; time_loop++) {

int count_1 = RTC.getDay() / 10 ; // 1 o/p
int count_1_r = RTC.getDay() % 10 ;// 2 o/p
int count_2 = RTC.getMonth() / 10 ; // 1 o/p
int count_3 = RTC.getMonth() % 10 ;// 2 o/p

digitalWrite(clockPin, LOW);
shiftOut(dataPin, clockPin, MSBFIRST, j[count_3]);

```

```

//delay(10);
digitalWrite(clockPin, LOW);
shiftOut(dataPin, clockPin, MSBFIRST, j[count_2]);
//delay(10);
digitalWrite(clockPin, LOW);
shiftOut(dataPin, clockPin, MSBFIRST, j[count_1_r]);
//delay(10);
digitalWrite(clockPin, LOW);
shiftOut(dataPin, clockPin, MSBFIRST, j[count_1]);
//delay(10);
delay(1000);
digitalWrite(sqw_pin_9, LOW);
digitalWrite(sqw_pin_10, LOW);
// delay(1);

}

time_loop = 0;

// this part checks any data on serial port


}

```