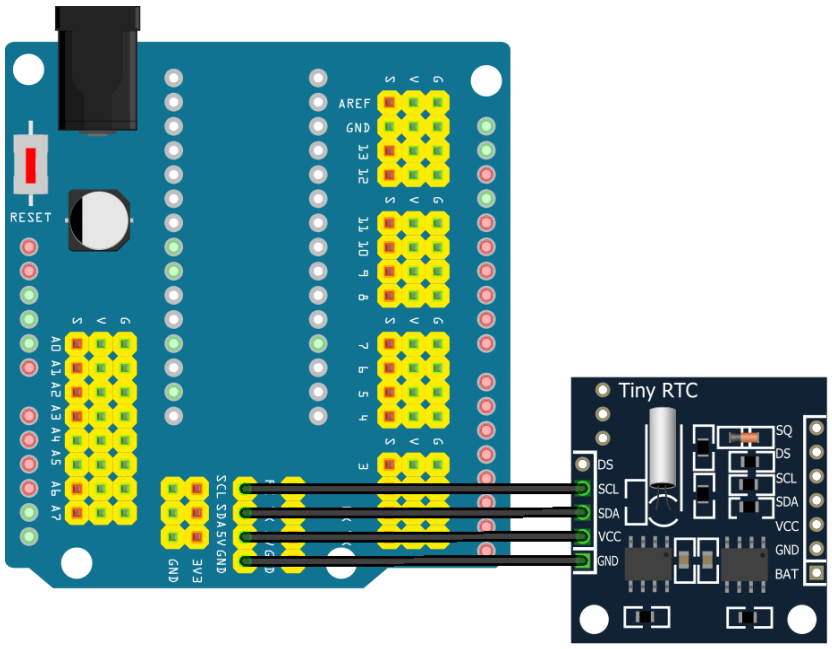
Task.1. Connect the circuit as shown in the picture.



The tiny RTC module is based on the clock chip DS1307 which supports the I2C protocol. It uses a lithium cell battery (CR2032). The DS1307 serial **real - time clock** (RTC) is a lowpower, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. The clock/calendar provides seconds, minutes, hours, day, date,month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator.

Code example:

```
#include <Wire.h>
#define BAUDRATE 115200
#define DS1307_I2C_ADRESS 0x68
byte sec, minute, hour;
byte dayOfWeek, dayOfMonth, month, year;
```

```
byte decToBcd(byte dec) {
return ((dec/10*16)+(dec%10)); }

byte bcdToDec(byte bcd) {
return ((bcd/16*10)+(bcd%16)); }

void setDateDs1307() {
 sec = 45; minute = 29; hour = 11;
 dayOfWeek = 2;
 dayOfMonth = 31; month = 1; year= 23;

 Wire.beginTransmission(DS1307_I2C_ADRES);
 Wire.write(decToBcd(0));
 Wire.write(decToBcd(sec)); // 0 to bit 7 starts the clock
 Wire.write(decToBcd(minute));
 Wire.write(decToBcd(hour));
 Wire.write(decToBcd(dayOfWeek));
 Wire.write(decToBcd(dayOfMonth));
 Wire.write(decToBcd(month));
 Wire.write(decToBcd(year));
 Wire.endTransmission();
}

void getDateDs1307() {

 Wire.beginTransmission(DS1307_I2C_ADRESS);
 Wire.write(decToBcd(0));
 Wire.endTransmission();
 Wire.requestFrom(DS1307_I2C_ADRES, 7);
 sec = bcdToDec(Wire.read() & 0x7f);
 minute = bcdToDec(Wire.read());
 hour = bcdToDec(Wire.read() & 0x3f);
 dayOfWeek = bcdToDec(Wire.read());
 dayOfMonth = bcdToDec(Wire.read());
 month = bcdToDec(Wire.read());
 year = bcdToDec(Wire.read());
 Serial.print(hour, DEC);
 Serial.print(":");
 Serial.print(minute, DEC);
 Serial.print(":");
 Serial.print(sec, DEC);
 Serial.print(" ");
```

```
 Serial.print(month, DEC);
 Serial.print("/");
 Serial.print(dayOfMonth, DEC);
 Serial.print("/");
 Serial.print(year,DEC);
 Serial.println(" ");
}

void setup() {
 Wire.begin();
 Serial.begin(BAUDRATE);
 setDateDs1307();
}

void loop() {
 delay(2000);
 getDateDs1307();
}
```

Task.2. Use the `RTClib`.

Add the `RTClib` to Your Arduino IDE using the *Library Manager*.

```
#include<Wire.h>
#include<RTClib.h>
#define BAUDRATE 115200

RTC_DS1307 RTC;

void setup () {
 Serial.begin(BAUDRATE);
 Wire.begin();
 RTC.begin();
 RTC.adjust(DateTime(2014, 1, 21, 3, 0, 0));
 if (! RTC.isrunning()) {
  Serial.println("RTC is NOT running!");
  while(1);
 }
}

long p_millis = 0;
#define TIM_DELAY 2000
```

```
void loop () {
  if(millis() - p_millis > TIM_DELAY) {
   DateTime now = RTC.now();
   Serial.print(now.year(), DEC);
   Serial.print('.');
   Serial.print(now.month(), DEC);
   Serial.print('.');
   Serial.print(now.day(), DEC);
   Serial.print(' time:');
   Serial.print(now.hour(), DEC);
   Serial.print(':');
   Serial.print(now.minute(), DEC);
   Serial.print(':');
   Serial.print(now.second(), DEC);
   Serial.println();
   p_millis() = millis();
  }
}
```

Reference:

adafruit.github.io/RTClib/html/index.html

Task.3. Display the date and time on the LCD.

Task.4. Display current date and day of week on the LCD.

For those interested:

1. Guide for Real Time Clock (RTC) Module with Arduino:

   randomnerdtutorials.com/guide-for-real-time-clock-rtc-module-with-arduino-ds1307-and-ds3231/