

# ECE 106 – Fall 2023 Project

## Real clock timer

Students name:

Jana hariri-S22208115

Ruhana Ruhul Amin-S22208137

Maha Alblwai-S22107905

Instructor:

Dr.Aziza

## Table of Contents:

Introduction.....	3
Procedure.....	4
Results.....	7
Conclusion.....	8

# Introduction

In the world of electronics and microcontrollers, accurate timekeeping is a crucial aspect of many projects. The Real-Time Clock (RTC) module, coupled with an Arduino UNO, provides an elegant solution for maintaining precise time and date information. This project aims to guide you through the process of constructing a robust RTC system using an Arduino UNO, a DS3231 RTC module, and a versatile LCD 20x4 display with I2C interface.

By undertaking this project, you will not only enhance your understanding of Arduino-based projects but also acquire valuable skills in utilizing external modules for specialized functions. The resulting RTC system can be employed in a variety of applications, from simple timekeeping projects to more complex systems requiring precise scheduling and data logging. Let's embark on this journey to create a reliable Real-Time Clock with Arduino UNO, DS3231 and LCD 20x4.

# Procedure

To create a Real Time Clock (RTC) using an Arduino UNO, DS3231 RTC module, and an LCD 20x4 with I2C interface, you can follow the procedure outlined below:

## 1. Gather the necessary components:

- Arduino UNO board
- DS3231 RTC module
- LCD 20x4 with I2C interface
- Jumper wires

## 2. Connect the DS3231 RTC module to the Arduino UNO:

- Connect the VCC pin of the module to the 5V pin on the Arduino.
- Connect the GND pin of the module to the GND pin on the Arduino.
- Connect the SDA pin of the module to the A4 pin or SDA pin on the Arduino.
- Connect the SCL pin of the module to the A5 pin or SCL pin on the Arduino.

## 3. Connect the LCD 20x4 with I2C interface to the Arduino UNO:

- Connect the SDA pin of the LCD to the A4 pin or SDA pin on the Arduino.
- Connect the SCL pin of the LCD to the A5 pin or SCL pin on the Arduino.

## 4. Install the required libraries:

- In the Arduino IDE, go to "Sketch" -> "Include Library" -> "Manage Libraries".
- Search for and install the "RTCLib" and "LiquidCrystal\_I2C" libraries.

## 5. Write the code:

- Include the necessary libraries at the beginning of your code.
- Initialize the RTC and LCD objects.
- Set up the LCD display (number of columns and rows).
- In the setup function, initialize the RTC and LCD.
- In the loop function, read the current time, date and temperature from the RTC and display it on the LCD.

## Here's the code for the following project

```
// Autor: Zineb KANDOUSSI (https://www.facebook.com/SchoolSurSeineFR/)
// Creation : Le 21/04/2023
// Description : Display the current time, date and temperature on the 20x4 LCD screen
// Inspired from this code : File => Examples => RTCLib => ds3231

#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include "RTCLib.h"

LiquidCrystal_I2C lcd(0x27, 20, 4);

RTC_DS3231 rtc;

char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

byte char_temp[8]={B00100,B01010,B01010,B01110,B01110,B11111,B11111,B01110}; // for thermometer icon

void setup()
{
    rtc.begin();

    lcd.init();

    lcd.backlight();

    lcd.createChar(0,char_temp);

    lcd.setCursor(0, 0);

    lcd.print(" School Sur Seine!");

    lcd.setCursor(7, 1);

    lcd.print(":");

    delay(5000);

    lcd.clear();

    if (rtc.lostPower()) {

        // When time needs to be set on a new device, or after a power loss, the
        // following line sets the RTC to the date & time this sketch was compiled
        rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

    }

}

void loop()
{
    DateTime now = rtc.now();
```

```
//lcd.setCursor(column,row);

//-----Date display -----//

lcd.setCursor(6,0);

//lcd.print(daysOfTheWeek[now.dayOfTheWeek()]);

lcd.setCursor(0,1);

//lcd.print("Date:");

lcd.setCursor(6,1);

if (now.day() <=9 )

{

    lcd.print("0");

    lcd.setCursor(7,1);

    lcd.print(now.day(),DEC);

}

else {lcd.print(now.day(),DEC);}

lcd.setCursor(8,1);

lcd.print(":");

lcd.setCursor(9,1);

if (now.month() <=9 )

{

    lcd.print("0");

    lcd.setCursor(10,1);

    lcd.print(now.month(),DEC);

}

else {lcd.print(now.month(),DEC);}

lcd.setCursor(11,1);

lcd.print(":");

lcd.setCursor(12,1);

lcd.print(now.year(),DEC);

//-----Time display -----//

lcd.setCursor(0, 2);

lcd.print("Time:");

lcd.setCursor(4,0);

if (now.hour() <=9 )

{

    lcd.print("0");
```

```
lcd.setCursor(5,0);

lcd.print(now.hour(),DEC);

}

else {lcd.print(now.hour(),DEC);}

lcd.setCursor(6,0);

lcd.print(":");

lcd.setCursor(7,0);

if (now.minute() <=9 )

{

    lcd.print("0");

    lcd.setCursor(8,0);

    lcd.print(now.minute(),DEC);

}

else {lcd.print(now.minute(),DEC);}

lcd.setCursor(9,0);

lcd.print(":");

lcd.setCursor(10,0);

if (now.second() <=9 )

{

    lcd.print("0");

    lcd.setCursor(11,0);

    lcd.print(now.second(),DEC);

}

else {lcd.print(now.second(),DEC);}

//-----Temperature display -----//

lcd.setCursor(0,3);

//lcd.print("Temp:");

lcd.setCursor(0,1);

lcd.print(rtc.getTemperature());

lcd.setCursor(4,1);

//lcd.write((char)223);

lcd.setCursor(4,1);

lcd.print("C");

lcd.setCursor(9,1);

//lcd.print(char(0));}
```

6. Upload the code to the Arduino UNO board using the Arduino IDE.

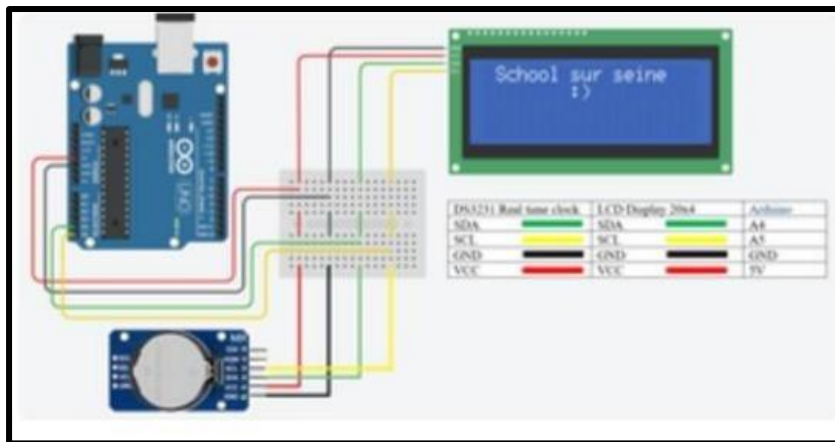
7. Connect a power supply to the Arduino UNO and observe the LCD display. It should show the current date and time as obtained from the DS3231 RTC module.

This is a basic example to display the date and time on the LCD. You can further enhance the functionality by adding alarm settings, temperature readings, or other features based on your requirements.



## Results

For the results we got exactly what we were looking for which is a real clock timer that you can set it according to your needs. Furthermore, it's easy to use since all you need to do is to adjust the code through the Arduino.



This is the circuit diagram that we used and helped us through the project so we can get the expected result.

## Conclusion

In conclusion, creating a real-time clock using an Arduino UNO, DS3231 RTC module, and an LCD 20x4 with I2C interface is a straightforward process that involves connecting the components, installing the required libraries, and writing the necessary code. By following the outlined procedure, you will be able to display the current time on the LCD screen. This project provides a practical and functional solution for keeping track of time accurately using Arduino and RTC technology. With a solid understanding of the procedure and some creativity, this project can be further enhanced by adding additional features such as alarm functionality or date display.