

CHAPTER 1

DS1302 REAL-TIME CLOCK AND CALENDAR MODULE

CHAPTER 1

DS1302 Real-Time Clock and Calendar Module

1.1 Introduction

DS1302 is a trickle-charge real-time clock/calendar chip contains 31 bytes of Random Access Memory (RAM). This chip can be access thru simple serial interface. It provides seconds, minutes, hours, day, date, month, and year information and automatically adjust months fewer than 31 days and leap year. It can be operated as 24-hour or 12-hour format with an AM/PM indicator.

1.2 Operation Circuitry

Interfacing the DS1302 with a microcontroller is simplified by using synchronous serial communication. Only three wires are required to communicate with the clock/RAM: CE, I/O (data line), and SCLK (serial clock). Data can be transferred to and from the clock/RAM 1 byte at a time or in a burst of up to 31 bytes. The DS1302 is designed to operate on very low power and retain data and clock information on less than $1\mu\text{W}$.

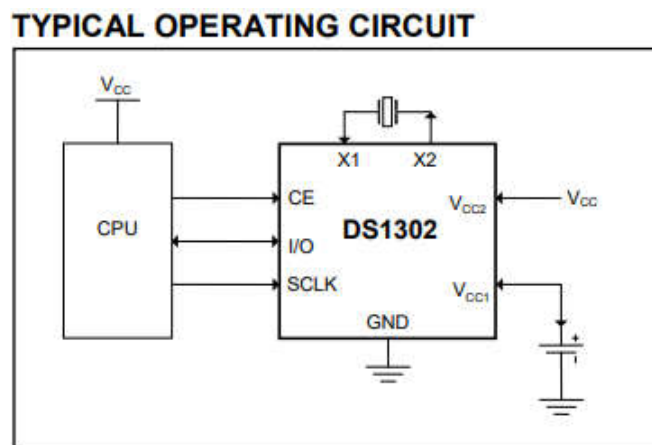


Figure 1: DS1302 Operating Circuitry

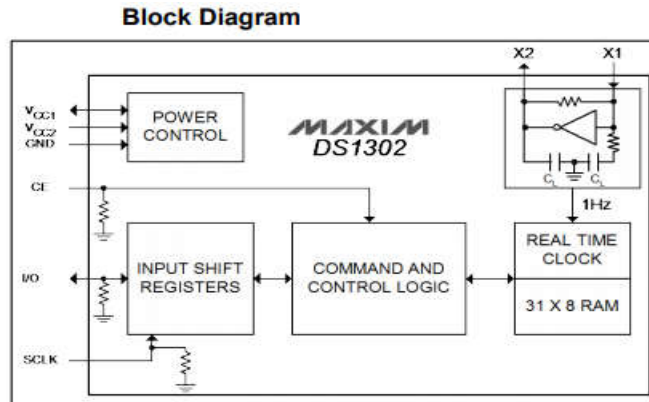


Figure 2: DS1302 Block Diagram

1.3 Specifications

- Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year with Leap-Year Compensation Valid Up to 2100
- 31 x 8 Battery-Backed General-Purpose RAM
- Simple 3-Wire Interface
- TTL-Compatible (VCC = 5V)
- Single-Byte or Multiple-Byte (Burst Mode) Data Transfer for Read or Write of Clock or RAM Data
- 2.0V to 5.5V Full Operation
- Uses Less Than 300nA at 2.0V
- 0°C to +70°C, Optional Industrial Temperature Range: -40°C to +85°C

1.4 Bill of Materials

COMPONENTS	QUANTITY
DS1302 Module	1pc
Arduino/Gizduino Uno	1pc
Alphanumeric LCD 16x2	1pc
Bread Board	1pc
5k Potentiometer	1pc
Jumper Wires	20pcs
USB Cable to program/power Arduino Uno	1pc

Table 1: Bill of Materials for DS1302 Project

1.5 Schematic Diagram

Figure 3 below is the schematic diagram for the project Real Time (RTCC) Clock and Calendar chip DS1302 module. The ground and Vcc is not shown on the figure but it easily to find in the Arduino Uno and RTCC module. In this diagram Arduino or Gizduino Uno was used to simulate the capability of DS1302 to track the not only time but also calendar. Once the correct time and calendar is already configured, this chip will track the right time and calendar.

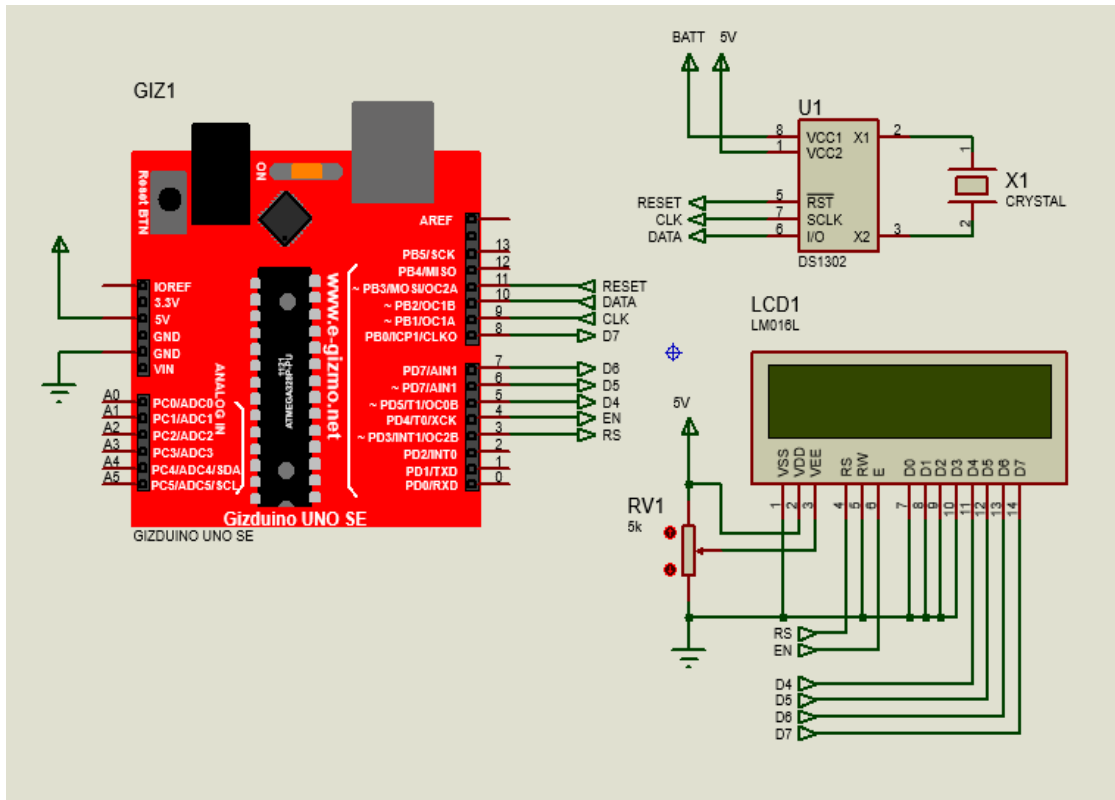


Figure 3: Schematic Diagram of DS1302 Project

1.6 Pictorial Diagram

The Figure 4 below is the point-to-point connections of the RTCC project using jumper wires. It is very easy to construct and the potentiometer was used as the contrast of the alphanumeric Liquid Crystal Display (LCD) 16X2. It also shown below how to connect LCD to Arduino Uno.

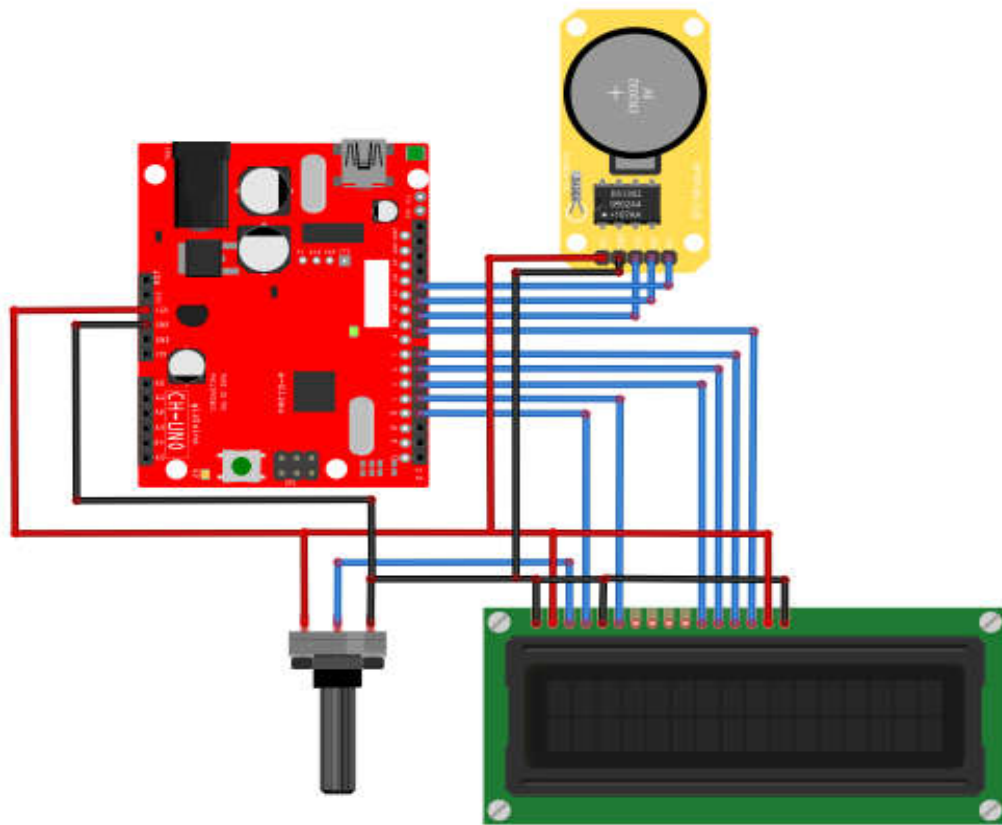


Figure 4: Pictorial Diagram of DS1302

1.7 Circuit Construction

To construct the circuit for the DS1302 project, we can use a Bread Board to mount the LCD and potentiometer as shown in the Figure 5 below.

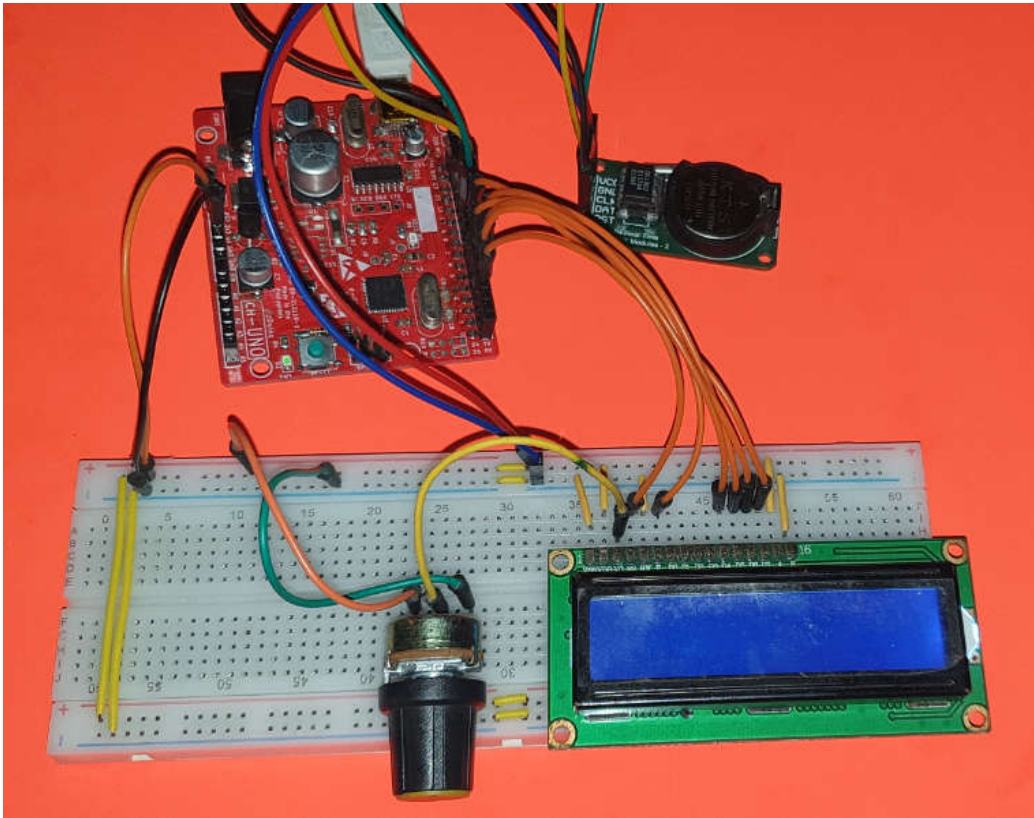


Figure 5: The sample circuit construction of DS1302 Project

1.8 Source Code

This is the final steps on how to make this project complete using the Arduino IDE (Integrated Development Environment) to program the Arduino Uno using the source code listing below.

Steps are follows below:

1. Open the Arduino IDE then select Arduino Uno/Genuino in Tools->Board->Arduino/Genuino Uno then select the COM Port using Tools->Port-> and select name of the COM Port. After the IDE setting was done, download the library [virtuabotixRTC](#) in this link then add to the Arduino IDE library by using menu item Sketch->Include Library->Add .ZIP Library and browse the file where you saved it then add the file ArduinoRTClibrary-master.
2. After adding the required library in Arduino IDE, copy the code listing below then click the Verify button then after compiling the sketch it should be Done compiling appears on the status bar.
3. After compiling the sketch, press the Upload button so that the sketch will be uploaded in Arduino/Gizduino Uno. Test the circuit if it is working.

```
//This code is all about DS1302 RTC module using LCD 1602
```

```
//Download this code to the link  
https://github.com/chrisfryer78/ArduinoRTClibrary then  
add the .zip file to the Arduino Library
```



```

#include <virtuabotixRTC.h> //DS1302 Library
//Include LCD1602 library
#include <LiquidCrystal.h>

//Choose pinouts of LCD1602
const int RS = 3, EN = 4, D4 = 5, D5 = 6, D6 = 7, D7 =
8;
LiquidCrystal lcd(RS, EN, D4, D5, D6, D7);

//Wiring for DS1302 Module SCLK, I/O, CE Or CLK, DAT,
Reset
virtuabotixRTC myRTC(9, 10, 11);

void setup()
{
    serial.begin(115200);
    lcd.begin (16,2); //Initialize the LCD
    lcd.setCursor(0,0);
    lcd.print("MARIC DS1302 RTC");
    lcd.setCursor(0,1);
    lcd.print("USING GIZDUINO");
    delay(1000);

    //Set the updated time
    myRTC.setDS1302Time(0, 3, 3, 5, 31, 7, 2020);
    //Comment this code after setting the right time and
    date
}

```

```

void loop()
{
    lcd.clear();
    myRTC.updateTime();
    lcd.setCursor(4,0);
    lcd.print(myRTC.hours);
    lcd.print(":");
    lcd.print(myRTC.minutes);
    lcd.print(":");
    lcd.print(myRTC.seconds);
    lcd.setCursor(4,1);
    lcd.print(myRTC.dayofmonth);
    lcd.print("-");
    lcd.print(myRTC.month);
    lcd.print("-");
    lcd.print(myRTC.year);

    delay(1000);
}

```

Code Listing 1.8: DS1302_LCD1602.ino

1.9 Testing the DS1302 Project

This is the output of the LCD shown below in Figure 6 and 7. As we can see, the LCD displays the first screen as all about the project for 1 second timer delay then the next screen will be the display of real time clock and calendar set by the user using the source code of the project above.

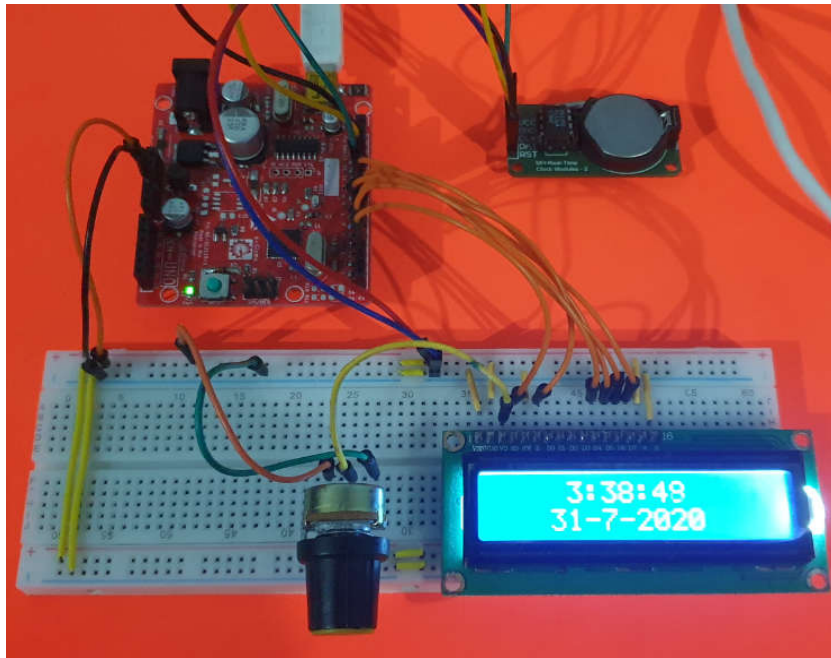


Figure 6: Testing of DS1302 Project