PROJECT 7

Real Time Based system using RTC module for schedule operation

(Realtime Clock)

By: Utkarsh Patel

G11 ES & IOT

Overview

The DS1302 RTC (Real-Time Clock) module is a low-cost, easy-to-use chip designed to provide precise timekeeping. This document outlines the steps and necessary components to build a real-time clock (RTC) using the DS1302 module. This clock will display the current time and date on an LCD.

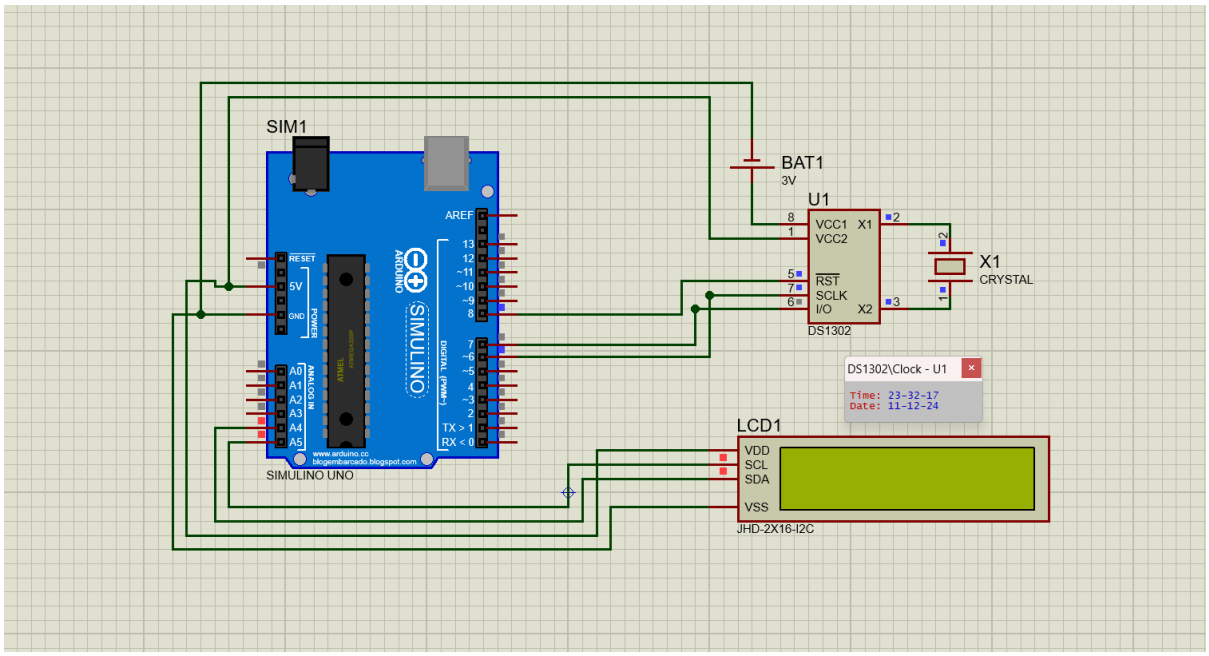
Features

1. Accurate timekeeping with seconds, minutes, hours, day, date, month, and year.
2. Low power consumption with a battery backup.
3. User-friendly interface using a 16x2 LCD.
4. Adjustable date and time via software or optional buttons.

Components Used:

- Arduino UNO R3
- DS1302 RTC Module
- LCD Display I2C
- CMOS battery
- Connecting Wires

Schematic Diagram:



Program:

```
#include <LiquidCrystal_I2C.h>
#include <ThreeWire.h>
#include <RtcDS1302.h>

LiquidCrystal_I2C lcd (0x3F, 16, 2);
ThreeWire mywire(7,6,8);   // DATA, CLK, RST
RtcDS1302<ThreeWire> Rtc(mywire);

void setup()
{
  lcd.init();
  lcd.backlight();
  lcd.clear();
  Rtc.Begin();

  RtcDateTime currentTime = RtcDateTime(__DATE__ , __TIME__);
  Rtc.SetDateTime(currentTime);
}

void loop(){
  RtcDateTime now =Rtc.GetDateTime();

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Date: ");
  lcd.print(now.Day());
  lcd.print("/");
  lcd.print(now.Month());
  lcd.print("/");
  lcd.print(now.Year());
```

```

  lcd.setCursor(0,1);
  lcd.print("Time: ");
  lcd.print(now.Hour());
  lcd.print(":");
  lcd.print(now.Minute());
  lcd.print(":");
  lcd.print(now.Second());

  delay(500);
}

```

How It Works

1. **Initialization**:
    - The DS1302 module is initialized in the setup() function.
    - The LCD is configured to display the time and date.
2. **Time and Date Retrieval**:
    - The rtc.getDateTime() function fetches the current time and date from the DS1302.
3. **Display**:
    - The time and date are displayed on the LCD in the loop() function.
    - Optional: Display the same information on the Serial Monitor for debugging.
4. **Adjusting Time**:
    - Uncomment and modify the rtc.setDateTime() line in setup() to set the desired time and date.
    - Remove or comment it out after the initial setup to avoid resetting the time every upload.

Applications

1. **Digital Clocks**: Simple and reliable real-time clocks.
2. **Timers**: Automate tasks based on specific times.
3. **Data Logging**: Record timestamps for sensor data.
4. **Alarm Systems**: Trigger events at pre-defined times.
