Experiment - 12

Interface RTC with ARDUINO DUE and display Date, Time on LCD display

OBJECTIVES :

1. To Interface RTC Module with ARDUINO DUE
2. To display the Date & Time on LCD display

MATERIALS REQUIRED :

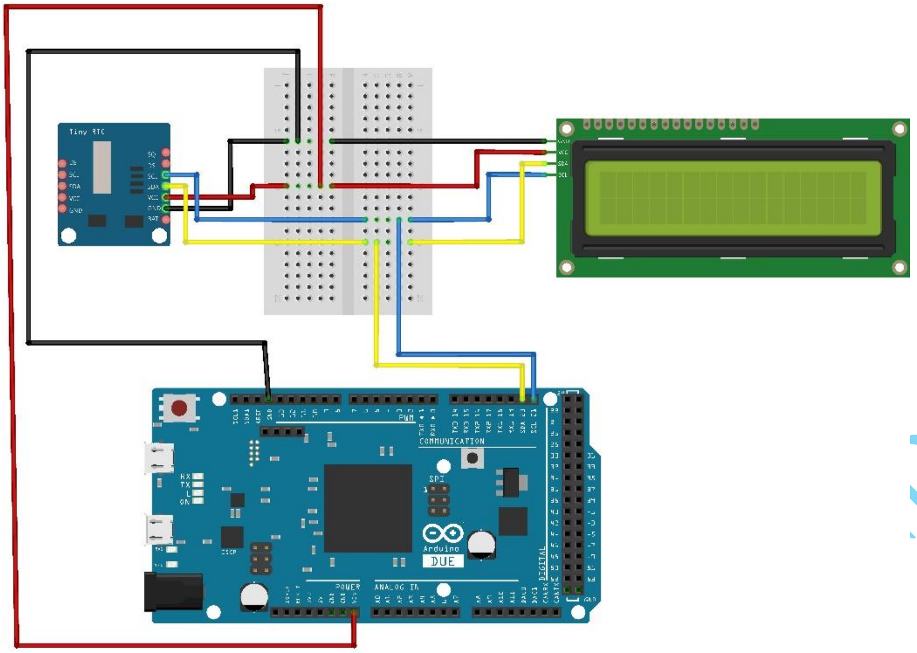
- Arduino DUE board
- RTC module (DS1307)
- I²C LCD module (16x2)
- Jumper wires
- Breadboard
- USB cable for Arduino programming
- Computer with Arduino IDE installed

THEORY :

- Real-Time Clock (RTC): An RTC is an integrated circuit that keeps track of the current date and time even when the main system power is off, using a backup battery. Common modules like DS1307 and DS3231 communicate via the I²C protocol, making them easy to interface with microcontrollers.
- I²C LCD Module: A liquid crystal display (LCD) with an integrated I²C adapter chip (such as PCF8574) can be controlled using only four wires (VCC, GND, SDA, SCL), reducing pin usage and wiring complexity compared to parallel LCDs.
- Arduino Due: The Due is based on the ARM Cortex-M3 microcontroller and supports both 3.3V and 5V logic levels, making it versatile for interfacing with various sensors and displays.

PIN CONFIGURATION :

Arduino Due Pin	RTC Module Pin	I²C LCD Module Pin	Function/Purpose
5V	VCC	VCC	Power Supply
GND	GND	GND	Ground
20 (SDA)	SDA	SDA	I²C Data
21 (SCL)	SCL	SCL	I²C Clock



PROCEDURE :

Hardware Setup :

1. Connect the RTC Module:
   - Connect the VCC pin of the RTC module to 3.3V or 5V on the Arduino Due (check your RTC module's datasheet for voltage requirements).
   - Connect the GND pin to GND.
   - Connect SDA to the SDA pin (pin 20) and SCL to the SCL pin (pin 21) on the Arduino Due.
2. Connect the I²C LCD:
   - Connect VCC to 5V, GND to GND, SDA to the same SDA pin as the RTC, and SCL to the same SCL pin.
   - Note: Both RTC and LCD share the same I²C bus.
3. Power Up: Connect the Arduino Due to your computer via USB.

Software Setup

1. Install Required Libraries in the Arduino IDE Library Manager :
   - RTClib (for RTC communication)
   - LiquidCrystal_I2C (for I²C LCD control)

Programming Code :

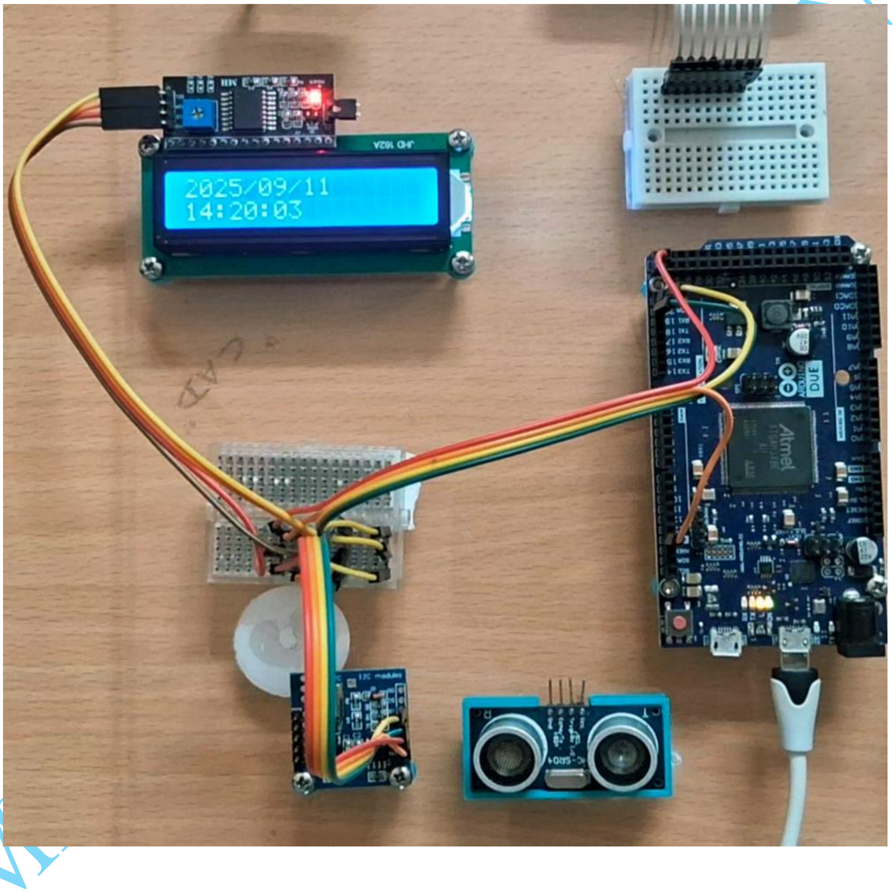
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "RTClib.h"

RTC_DS1307 rtc;  // Use DS1307 instead of DS3231
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Try 0x3F if not working
void setup() {
  Serial.begin(9600);
  delay(3000);

  if (!rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }

  if (!rtc.isrunning()) {
    Serial.println("RTC is not running, setting time...");
   rtc.adjust(DateTime(year, month, day, hour, minute, second));
    rtc.adjust(DateTime(2025, 9, 11, 14, 20, 0));
  }

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("DS1307 Ready");
  delay(2000);
  lcd.clear();  }
```

```
void loop() {
  DateTime now = rtc.now();
  Serial.print(now.year()); Serial.print('/');
  Serial.print(now.month()); Serial.print('/');
  Serial.print(now.day());   Serial.print("  ");
  Serial.print(now.hour());  Serial.print(':');
  Serial.print(now.minute());Serial.print(':');
  Serial.print(now.second());Serial.println();

 lcd.setCursor(0, 0);
  lcd.print(now.year());
  lcd.print('/');
  if (now.month() < 10) lcd.print('0');
  lcd.print(now.month());
  lcd.print('/');
  if (now.day() < 10) lcd.print('0');
  lcd.print(now.day());

  lcd.setCursor(0, 1);
  if (now.hour() < 10) lcd.print('0');
  lcd.print(now.hour());
  lcd.print(':');
  if (now.minute() < 10) lcd.print('0');
  lcd.print(now.minute());
  lcd.print(':');
  if (now.second() < 10) lcd.print('0');
  lcd.print(now.second());
  delay(1000);
}
```

2. Upload the Code:
   Open the Arduino IDE, type the provided code, select the correct board (Arduino DUE), and upload the sketch.

3. Set the RTC Time (First Run):
   Uncomment the line rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); in the setup() function, upload once, then comment it out and upload again. This sets the RTC to your computer's current time.

4. Observe the Output:
   The LCD should display the current date and time updated every second from the RTC.



RESULT :

CONCLUSION :