CHAPTER: 9 RTC – REAL TIME CLOCK

PRACTICAL: 9A

AIM: To interface Real Time Clock – DS1307 using Arduino.

ARDUINO CODE :

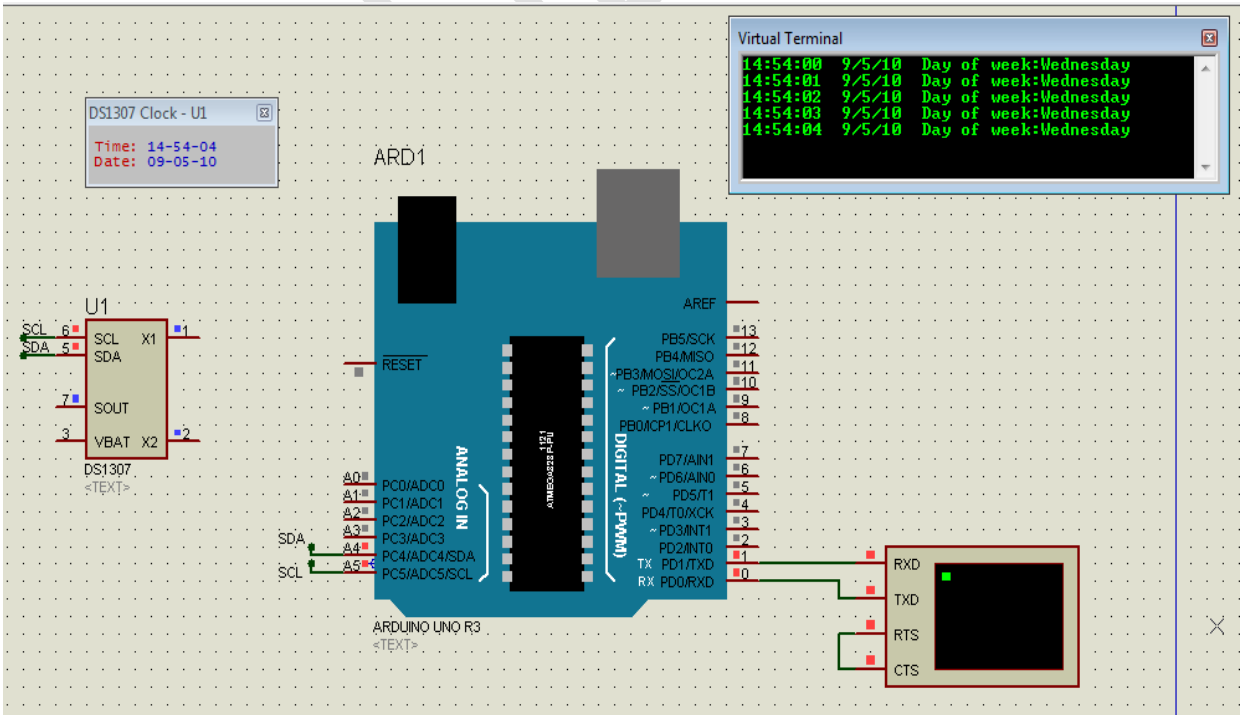
```
/**********************
 * Author: Shreejicharan
 * Title: To interface Real Time Clock – DS1307 using Arduino.
 * Email: shreejicharanelectronics@gmail.com
 **********************/

#include "Wire.h"
#define DS1307_I2C_ADDRESS 0x68

// Convert normal decimal numbers to binary coded decimal
byte decToBcd(byte val)
{
  return ( (val/10*16) + (val%10) );
}

// Convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
  return ( (val/16*10) + (val%16) );
}

// 1) Sets the date and time on the ds1307
// 2) Starts the clock
// 3) Sets hour mode to 24 hour clock
// Assumes you're passing in valid numbers

void setDateDs1307(byte second,        // 0-59
byte minute,        // 0-59
byte hour,          // 1-23
byte dayOfWeek,     // 1-7
byte dayOfMonth,    // 1-28/29/30/31
byte month,         // 1-12
byte year)          // 0-99
{
```

```
  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  Wire.write(0);
  Wire.write(decToBcd(second));    // 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.write(00010000); // sends 0x10 (hex) 00010000 (binary) to control register - turns on square
wave
  Wire.endTransmission();
}

// Gets the date and time from the ds1307
void getDateDs1307(byte *second,
byte *minute,
byte *hour,
byte *dayOfWeek,
byte *dayOfMonth,
byte *month,
byte *year)
{
  // Reset the register pointer
  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  Wire.write(0);
  Wire.endTransmission();
  Wire.requestFrom(DS1307_I2C_ADDRESS, 7);
  // A few of these need masks because certain bits are control bits
  *second     = bcdToDec(Wire.read() & 0x7f);
  *minute     = bcdToDec(Wire.read());
  *hour       = bcdToDec(Wire.read() & 0x3f);  // Need to change this if 12 hour am/pm
  *dayOfWeek  = bcdToDec(Wire.read());
  *dayOfMonth = bcdToDec(Wire.read());
  *month      = bcdToDec(Wire.read());
  *year       = bcdToDec(Wire.read());
}

void setup()
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  Wire.begin();
  Serial.begin(9600);
  // Change these values to what you want to set your clock to.
```

```
  // You probably only want to set your clock once and then remove
  // the setDateDs1307 call.
  second = 0;
  minute = 54;
  hour = 14;
  dayOfWeek = 4;
  dayOfMonth = 9;
  month = 5;
  year = 10;
  setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year);
}


void loop()
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;

  getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);

  Serial.print(hour, DEC);// convert the byte variable to a decimal number when being displayed
  Serial.print(":");

  if (minute<10)
  {
    Serial.print("0");
  }
  Serial.print(minute, DEC);
  Serial.print(":");
            if (second<10)
            {
              Serial.print("0");
            }
  Serial.print(second, DEC);
  Serial.print("  ");
  Serial.print(dayOfMonth, DEC);
  Serial.print("/");
  Serial.print(month, DEC);
  Serial.print("/");
  Serial.print(year, DEC);
  Serial.print("  Day of week:");
  switch(dayOfWeek){
  case 1:
    Serial.println("Sunday");
    break;
```

```
  case 2:
    Serial.println("Monday");
    break;
  case 3:
    Serial.println("Tuesday");
    break;
  case 4:
    Serial.println("Wednesday");
    break;
  case 5:
    Serial.println("Thursday");
    break;
  case 6:
    Serial.println("Friday");
    break;
  case 7:
    Serial.println("Saturday");
    break;
  }
  //  Serial.println(dayOfWeek, DEC);
  delay(1000);
}
```

SIMULATION:



PRACTICAL: 10B

AIM: To interface World Real Time Clock – DS1307 using Arduino.

ARDUINO CODE :

```
/**********************
 * Author: Shreejicharan
 * Title: To interface World Real Time Clock – DS1307 using Arduino.
 * Email: shreejicharanelectronics@gmail.com
 **********************/

/*
 World Clock - 10 July 2010
 */

#include <LiquidCrystal.h>
#include <Wire.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
#define DS1307_I2C_ADDRESS 0x68

// Convert normal decimal numbers to binary coded decimal
byte decToBcd(byte val)
{
  return ( (val/10*16) + (val%10) );
}

// Convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
  return ( (val/16*10) + (val%16) );
}

// Gets the date and time from the ds1307
void getDateDs1307(byte *second,
byte *minute,
byte *hour,
byte *dayOfWeek,
byte *dayOfMonth,
byte *month,
byte *year)
```

```
{
  // Reset the register pointer
  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  Wire.write(0);
  Wire.endTransmission();

  Wire.requestFrom(DS1307_I2C_ADDRESS, 7);

  // A few of these need masks because certain bits are control bits
  *second     = bcdToDec(Wire.read() & 0x7f);
  *minute     = bcdToDec(Wire.read());
  *hour       = bcdToDec(Wire.read() & 0x3f);  // Need to change this if 12 hour am/pm
  *dayOfWeek  = bcdToDec(Wire.read());
  *dayOfMonth = bcdToDec(Wire.read());
  *month      = bcdToDec(Wire.read());
  *year       = bcdToDec(Wire.read());
}


void setDateDs1307(byte second,        // 0-59
byte minute,        // 0-59
byte hour,          // 1-23
byte dayOfWeek,     // 1-7
byte dayOfMonth,    // 1-28/29/30/31
byte month,         // 1-12
byte year)          // 0-99
{
  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  Wire.write(0);
  Wire.write(decToBcd(second));    // 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.write(0x10); // sends 0x10 (hex) 00010000 (binary) to control register - turns on square
wave
  Wire.endTransmission();
}


void timeSYD()
{
```

```
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
  lcd.setCursor(0,2);
  hour=hour+10;
  if (hour>23)
  {
    hour=hour-24;
  }
  lcd.print("Sydney GMT+10 ");
  if (hour<10)
  {
    lcd.print("0");
  }
  lcd.print(hour, DEC);
  if (minute<10)
  {
    lcd.print("0");
  }
  lcd.print(minute, DEC);
  lcd.print("h");
}

void timeLHR()
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
  lcd.setCursor(0,0);
  lcd.print("London GMT+00 ");
  if (hour<10)
  {
    lcd.print("0");
  }
  lcd.print(hour, DEC);
  if (minute<10)
  {
    lcd.print("0");
  }
  lcd.print(minute, DEC);
  lcd.print("h");
}

void timeHAV()
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
```

```
  getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
  lcd.setCursor(0,3);
  hour=hour-5;
  if (hour<0)
  {
    hour=hour+24;
  }
  lcd.print("Havana GMT-05 ");
  if (hour<10)
  {
    lcd.print("0");
  }
  lcd.print(hour, DEC);
  if (minute<10)
  {
    lcd.print("0");
  }
  lcd.print(minute, DEC);
  lcd.print("h");
}

void timeSVO()
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
  hour=hour+3; // SVO GMT+03
  if (hour>23)
  {
    hour=hour-24;
  }
  lcd.setCursor(0,1);
  lcd.print("Moscow GMT+03 ");
  if (hour<10)
  {
    lcd.print("0");
  }
  lcd.print(hour, DEC);
  if (minute<10)
  {
    lcd.print("0");
  }
  lcd.print(minute, DEC);
  lcd.print("h");
}
```

```
void setup() {
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  Wire.begin();
  // set up the LCD's number of rows and columns:
  lcd.begin(20, 4);
  // This time data is GMT
  second = 0;
  minute = 37;
  hour = 16;
  dayOfWeek = 6;
  dayOfMonth = 10;
  month = 7;
  year = 10;
  //  setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year);

}

void loop() {
  lcd.clear(); // clear LCD screen
  timeSYD();
  timeLHR();
  timeHAV();
  timeSVO();
  //delay(30000);
  delay(1000);
}
```

SIMULATION: