// PROJECT  :WireLevelRTCLCD

// PURPOSE  :Stage 3 (of 4) for the Wire-level I2C Introduction for ICS3U

// AUTHOR   :C. D'Arcy

// DATE     :2020 02 13

// uC       :328p

// HARDWARE :DS1307 RTC Breakout and LCD Appliance

// STATUS   :Shell to be completed

// REFERENCE:[http://darcy.rsgc.on.ca/ACES/TEI3M/WireLevelRTCExercises.html#time](http://darcy.rsgc.on.ca/ACES/TEI3M/WireLevelRTCExercises.html%23time)

#include <Wire.h>

#define RTCADDRESS 0x00           //DS1307 I2C Address

#include <**LiquidCrystal**.h>        //LCD Library...

#define LCD\_COLUMNS 16            //Number of columns in Character LCD screen

#define LCD\_ROWS    2             //Number of rows on LCD screen

//LCD appliance-compatible pin assignments...

uint8\_t RS = 7, EN = 6, D4 = 5, D5 = 4, D6 = 3, D7 = 2;

**LiquidCrystal** lcd(RS, EN, D4, D5, D6, D7);

**char buff[LCD\_COLUMNS];**      //convenient output buffer for formatted prints

**struct timeDate** {            //tight binding of RTC register data into a struct

  uint8\_t seconds;

  uint8\_t minutes;

  uint8\_t hours;

  uint8\_t date;

  uint8\_t mon;

  uint8\_t YEAR;

} tD;

**void setup()** {

  lcd.begin(LCD\_COLUMNS, LCD\_ROWS);       //initialize LCD screen

  lcd.clear();                            //erase the LCD screen

  lcd.setCursor(0, 0);                    //position display cursor at home

  lcd.print("DATE:");                     //provide a label on the first line

  lcd.setCursor(0, 1);                    //position cursor at start of second row

  lcd.print("TIME:");                     //provide a label on the second line

  pinMode(A2, OUTPUT);                    //provide supply for

  pinMode(A3, OUTPUT);                    // the DS1307 breakout board

  digitalWrite(A2, LOW);                  // inserted as an appliance

  digitalWrite(A3, HIGH);                 //

  Wire.begin();                           //initialize a Wire session

}

uint8\_t BCD2bin(uint8\_t bcd) {

  return 0;                    //compose an expression that converts BCD to binary

}

//the body of this method needs to be developed...

**void getAll(timeDate &td)** { //'**Address Of**' operator, **&**, is the key to no **return**!

??

}

**void loop()** {

  Wire.beginTransmission(RTCADDRESS);

  Wire.write(0);

  Wire.endTransmission();

  Wire.requestFrom(RTCADDRESS, 7);

  //while (!Wire.available()); **//uncomment when going live…**

  getAll(tD);

  sprintf(buff, "%2d/%02d/%02d", tD.date, tD.mon, tD.YEAR);

  lcd.setCursor(5, 0);

  lcd.print(buff);

  sprintf(buff, "%02d:%02d:%02d", tD.hours, tD.minutes, tD.seconds);

  lcd.setCursor(5, 1);

  lcd.print(buff);

  delay(1000); **//to be replaced in the next Stage**

}

# RTC AND RAM ADDRESS MAP (taken from the DS1307 Datasheet, pp.7-8)

**Table 2** shows the address map for the DS1307 RTC and RAM registers. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

