```
 #include <Wire.h>                      // for I2C communication
#include <LiquidCrystal_I2C.h>        // for LCD
#include <RTClib.h>                   // for RTC

LiquidCrystal_I2C lcd(0x27, 16, 2); // create LCD with I2C address 0x27, 16
characters per line, 2 lines
RTC_DS3231 rtc;                       // create rtc for the DS3231 RTC module,
address is fixed at 0x6




void updateRTC()
{

  lcd.clear();  // clear LCD display
  lcd.setCursor(0, 0);
  lcd.print("Edit Mode...");

  // ask user to enter new date and time
  const char txt[6][15] = { "year [4-digit]", "month [1~12]", "day [1~31]",
                            "hours [0~23]", "minutes [0~59]", "seconds
[0~59]"};
  String str = "";
  long newDate[6];

  while (Serial.available()) {
    Serial.read();  // clear serial buffer
  }

  for (int i = 0; i < 6; i++) {

    Serial.print("Enter ");
    Serial.print(txt[i]);
    Serial.print(": ");

    while (!Serial.available()) {
      ; // wait for user input
    }

    str = Serial.readString();  // read user input
    newDate[i] = str.toInt();   // convert user input to number and save to
array
```

```
    Serial.println(newDate[i]); // show user input
  }

  // update RTC
  rtc.adjust(DateTime(newDate[0], newDate[1], newDate[2], newDate[3],
newDate[4], newDate[5]));
  Serial.println("RTC Updated!");
}
//The second custom function we will create is the function updateLCD(). This
function will update or refresh the text displayed on the LCD. Inside this
function, we will first get the time and date from the RTC. This is done by
calling rtc.now() function which is included in the RTCLib.h library.

//The function rtc.now() in our code returns a DateTime data type that
contains the current date and time of the rtc. We then assign the data to
different variables for additional formatting on the LCD. After assigning the
variables, we use the functions lcd.setCursor() and lcd.print() from the
LiquidCrystal_I2C.h to position the cursor and to display the text
respectively on the LCD. The code below shows how these functions come
together to get the rtc time, format the text and display it to the LCD.

/*
   function to update LCD text
*/
void updateLCD()
{

  /*
     create array to convert digit days to words:

     0 = Sunday    |   4 = Thursday
     1 = Monday    |   5 = Friday
     2 = Tuesday   |   6 = Saturday
     3 = Wednesday |
  */
  const char dayInWords[7][4] = {"SUN", "MON", "TUE", "WED", "THU", "FRI",
"SAT"};

  /*
     create array to convert digit months to words:

     0 = [no use]  |
     1 = January   |   6 = June
     2 = February  |   7 = July
```

```
    3 = March      |   8 = August
    4 = April      |   9 = September
    5 = May        |   10 = October
    6 = June       |   11 = November
    7 = July       |   12 = December
  */
  const char monthInWords[13][4] = {" ", "JAN", "FEB", "MAR", "APR", "MAY",
"JUN",
                                    "JUL", "AUG", "SEP", "OCT", "NOV",
"DEC"};

  // get time and date from RTC and save in variables
  DateTime rtcTime = rtc.now();

  int ss = rtcTime.second();
  int mm = rtcTime.minute();
  int hh = rtcTime.twelveHour();
  int DD = rtcTime.dayOfTheWeek();
  int dd = rtcTime.day();
  int MM = rtcTime.month();
  int yyyy = rtcTime.year();

  // move LCD cursor to upper-left position
  lcd.setCursor(0, 0);

  // print date in dd-MMM-yyyy format and day of week
  if (dd < 10) lcd.print("0");  // add preceeding '0' if number is less than
10
  lcd.print(dd);
  lcd.print("-");
   lcd.print(monthInWords[MM]);
  lcd.print("-");
  lcd.print(yyyy);

  lcd.print("  ");
   lcd.print(dayInWords[DD]);

  // move LCD cursor to lower-left position
  lcd.setCursor(0, 1);

  // print time in 12H format
  if (hh < 10) lcd.print("0");
  lcd.print(hh);
  lcd.print(':');
```

```
  if (mm < 10) lcd.print("0");
  lcd.print(mm);
  lcd.print(':');

  if (ss < 10) lcd.print("0");
  lcd.print(ss);

  if (rtcTime.isPM()) lcd.print(" PM"); // print AM/PM indication
  else lcd.print(" AM");
}


void setup()
{
  Serial.begin(9600); // initialize serial

  lcd.init();        // initialize lcd
  lcd.backlight();   // switch-on lcd backlight

  rtc.begin();        // initialize rtc
}


void loop()
{
  updateLCD();  // update LCD text

  if (Serial.available()) {
    char input = Serial.read();
    if (input == 'u') updateRTC();  // update RTC time
  }
}
```