#include <LiquidCrystal\_I2C.h>

\*/

//\*\*\*\*\*\*\*\*\*\*\*\*libraries\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

#include <Wire.h>

#include <RTClib.h>

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

LiquidCrystal\_I2C lcd(0x27,128,32); // Display I2C 128 x 32

RTC\_DS1307 RTC;

//\*\*\*\*\*\*\*\*\*\*\*\*Button\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

int P1=6; // Button SET MENU'

int P2=7; // Button +

int P3=8; // Button -

int P4=9; // SWITCH Alarm

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*Alarm\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

#define LED 13

#define buzzer 10

//\*\*\*\*\*\*\*\*\*\*\*\*Variables\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

int hourupg;

int minupg;

int yearupg;

int monthupg;

int dayupg;

int menu =0;

int setAll =0;

uint8\_t alarmHours = 0, alarmMinutes = 0; // Holds the current alarm time

void setup()

{

lcd.begin();

lcd.backlight();

lcd.clear();

pinMode(P1,INPUT\_PULLUP); // https://www.arduino.cc/en/Tutorial/InputPullupSerial

pinMode(P2,INPUT\_PULLUP);

pinMode(P3,INPUT\_PULLUP);

pinMode(P4,INPUT\_PULLUP);

pinMode(LED,OUTPUT);

pinMode(buzzer, OUTPUT); // Set buzzer as an output

printAllOff();

Serial.begin(9600);

Wire.begin();

RTC.begin();

if (! RTC.isrunning()) {

Serial.println("RTC is NOT running!");

// Set the date and time at compile time

RTC.adjust(DateTime(\_\_DATE\_\_, \_\_TIME\_\_));

}

// RTC.adjust(DateTime(\_\_DATE\_\_, \_\_TIME\_\_)); //removing "//" to adjust the time

// The default display shows the date and time

int menu=0;

}

void loop()

{

// check if you press the SET button and increase the menu index

if(digitalRead(P1)== LOW)

{

menu=menu+1;

}

if((digitalRead(P2)== LOW)&&(digitalRead(P3)== LOW))

{

DisplaySetHourAll();

DisplaySetMinuteAll();

lcd.clear();

lcd.setCursor(5,0);

lcd.print("ALARM");

lcd.setCursor(5,1);

lcd.print(alarmHours, DEC);

lcd.print(":");

lcd.print(alarmMinutes, DEC);

delay(1000);

lcd.clear();

}

// in which subroutine should we go?

if (menu==0)

{

DisplayDateTime(); // void DisplayDateTime

Alarm(); // Alarm control

}

if (menu==1)

{

DisplaySetHour();

}

if (menu==2)

{

DisplaySetMinute();

}

if (menu==3)

{

DisplaySetYear();

}

if (menu==4)

{

DisplaySetMonth();

}

if (menu==5)

{

DisplaySetDay();

}

if (menu==6)

{

StoreAgg();

delay(500);

menu=0;

}

delay(100);

}

void DisplayDateTime ()

{

// We show the current date and time

DateTime now = RTC.now();

lcd.setCursor(0, 2);

lcd.print("Hour : ");

if (now.hour()<=9)

{

lcd.print("0");

}

lcd.print(now.hour(), DEC);

hourupg=now.hour();

lcd.print(":");

if (now.minute()<=9)

{

lcd.print("0");

}

lcd.print(now.minute(), DEC);

minupg=now.minute();

lcd.print(":");

if (now.second()<=9)

{

lcd.print("0");

}

lcd.print(now.second(), DEC);

lcd.setCursor(0, 1);

lcd.print("Date : ");

if (now.day()<=9)

{

lcd.print("0");

}

lcd.print(now.day(), DEC);

dayupg=now.day();

lcd.print("/");

if (now.month()<=9)

{

lcd.print("0");

}

lcd.print(now.month(), DEC);

monthupg=now.month();

lcd.print("/");

lcd.print(now.year(), DEC);

yearupg=now.year();

char DOW[][10]={"Sunday ","Monday ","Tuesday ","Wednesday","Thursday ","Friday ","Saturday "};

lcd.setCursor(0, 0);

lcd.print("Day : ");

lcd.print(DOW[now.dayOfTheWeek()]); // if it appears error in the code, enter the code given below

//lcd.print(DOW[now.dayOfWeek()]);

}

void DisplaySetHour()

{

// time setting

lcd.clear();

DateTime now = RTC.now();

if(digitalRead(P2)==LOW)

{

if(hourupg==23)

{

hourupg=0;

}

else

{

hourupg=hourupg+1;

}

}

if(digitalRead(P3)==LOW)

{

if(hourupg==0)

{

hourupg=23;

}

else

{

hourupg=hourupg-1;

}

}

lcd.setCursor(0,0);

lcd.print("Set time:");

lcd.setCursor(0,1);

lcd.print(hourupg,DEC);

delay(200);

}

void DisplaySetMinute()

{

// Setting the minutes

lcd.clear();

if(digitalRead(P2)==LOW)

{

if (minupg==59)

{

minupg=0;

}

else

{

minupg=minupg+1;

}

}

if(digitalRead(P3)==LOW)

{

if (minupg==0)

{

minupg=59;

}

else

{

minupg=minupg-1;

}

}

lcd.setCursor(0,0);

lcd.print("Set Minutes:");

lcd.setCursor(0,1);

lcd.print(minupg,DEC);

delay(200);

}

void DisplaySetYear()

{

// setting the year

lcd.clear();

if(digitalRead(P2)==LOW)

{

yearupg=yearupg+1;

}

if(digitalRead(P3)==LOW)

{

yearupg=yearupg-1;

}

lcd.setCursor(0,0);

lcd.print("Set Year:");

lcd.setCursor(0,1);

lcd.print(yearupg,DEC);

delay(200);

}

void DisplaySetMonth()

{

// Setting the month

lcd.clear();

if(digitalRead(P2)==LOW)

{

if (monthupg==12)

{

monthupg=1;

}

else

{

monthupg=monthupg+1;

}

}

if(digitalRead(P3)==LOW)

{

if (monthupg==1)

{

monthupg=12;

}

else

{

monthupg=monthupg-1;

}

}

lcd.setCursor(0,0);

lcd.print("Set Month:");

lcd.setCursor(0,1);

lcd.print(monthupg,DEC);

delay(200);

}

void DisplaySetDay()

{

// Setting the day

lcd.clear();

if(digitalRead(P2)==LOW)

{

if (dayupg==31)

{

dayupg=1;

}

else

{

dayupg=dayupg+1;

}

}

if(digitalRead(P3)==LOW)

{

if (dayupg==1)

{

dayupg=31;

}

else

{

dayupg=dayupg-1;

}

}

lcd.setCursor(0,0);

lcd.print("Set Day:");

lcd.setCursor(0,1);

lcd.print(dayupg,DEC);

delay(200);

}

void StoreAgg()

{

// Variable saving

lcd.clear();

lcd.setCursor(0,0);

lcd.print("SAVING IN");

lcd.setCursor(0,1);

lcd.print("PROGRESS");

RTC.adjust(DateTime(yearupg,monthupg,dayupg,hourupg,minupg,0));

delay(200);

}

void DisplaySetHourAll()// Setting the alarm minutes

{

while(digitalRead(P1)==HIGH){

lcd.clear();

if(digitalRead(P2)==LOW)

{

if(alarmHours==23)

{

alarmHours=0;

}

else

{

alarmHours=alarmHours+1;

}

}

if(digitalRead(P3)==LOW)

{

if(alarmHours==0)

{

alarmHours=23;

}

else

{

alarmHours=alarmHours-1;

}

}

lcd.setCursor(0,0);

lcd.print("Set HOUR Alarm:");

lcd.setCursor(0,1);

lcd.print(alarmHours,DEC);

delay(200);

}

delay(200);

}

void DisplaySetMinuteAll()// Setting the alarm minutes

{

while(digitalRead(P1)==HIGH){

lcd.clear();

if(digitalRead(P2)==LOW)

{

if (alarmMinutes==59)

{

alarmMinutes=0;

}

else

{

alarmMinutes=alarmMinutes+1;

}

}

if(digitalRead(P3)==LOW)

{

if (alarmMinutes==0)

{

alarmMinutes=59;

}

else

{

alarmMinutes=alarmMinutes-1;

}

}

lcd.setCursor(0,0);

lcd.print("Set MIN. Alarm:");

lcd.setCursor(0,1);

lcd.print(alarmMinutes,DEC);

delay(200);

}

delay(200);

}

void printAllOn(){

lcd.setCursor(0,3);

lcd.print("Alarm: ");

if (alarmHours <= 9)

{

lcd.print("0");

}

lcd.print(alarmHours, DEC);

lcd.print(":");

if (alarmMinutes <= 9)

{

lcd.print("0");

}

lcd.print(alarmMinutes, DEC);

}

void printAllOff() {

lcd.setCursor(0, 3);

lcd.print("Alarm: Off ");

}

void Alarm(){

if(digitalRead(P4)== LOW)

{

setAll=setAll+1;

}

if (setAll==0)

{

printAllOff();

noTone (buzzer);

digitalWrite(LED,LOW);

}

if (setAll==1)

{

printAllOn();

DateTime now = RTC.now();

if ( now.hour() == alarmHours && now.minute() == alarmMinutes )

{

lcd.noBacklight();

DateTime now = RTC.now();

digitalWrite(LED,HIGH);

tone(buzzer,880); //play the note "A5" (LA5)

delay (300);

tone(buzzer,698); //play the note "F6" (FA5)

lcd.backlight();

}

else{

noTone (buzzer);

digitalWrite(LED,LOW);

}

}

if (setAll==2)

{

setAll=0;

}

delay(200);

}

Library download：

<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>（LCD library）

<https://github.com/adafruit/RTClib>（RTC library）

Note: we have 4 buttons to finish the function,when we press button 2 and 3, we will see the alarm. And P1 is set(change the hours minutes or seconds). P2 is plus the digital,P3 is minus the digital and P4 is stopped (turn off the alarm).

pin:6(P1) 7(p2) 8(p3) 9(p4) 10 (buzzer) 13(led)