**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**Інститут комп’ютерних технологій, автоматики та метрології**

**Кафедра "Комп'ютеризовані системи автоматики"**



**ЗВІТ**

про виконання **лабораторної роботи №3**

з курсу **«Мікроконтролери, частина 1»**

**РОБОТА З ТАЙМЕРАМИ, LCD-ДИСПЛЕЄМ НА БАЗІ HD44780**

**ТА КЛАВІАТУРОЮ 4х4**

**Виконав:**

студент групи ІР-22

Яцків О. О.

Варіант 18

**Прийняла:**

к.т.н., старший викладач КСА

Лагун І. І.

Львів — 2023

**Порядок виконання**

1. Згідно варіанту завдання (таблиця 1) зібрати у пакеті симуляції Proteus схему на основі МК ATmega2560 та написати програму мовою C++ в Arduino IDE для реалізації вказаного завдання.

2. Розробити клієнтську програму для керування МК.

3. Протестувати роботу клієнтської програми зі схемою в Proteus.

4. Залити програму в Arduino-Kit.

5. Отримані результати представити викладачу.

**Завдання (варіант 18)**

1. Непарні номери варіантів використовують 8-ми розрядний інтерфейс LCD для обміну даними, Парні номери варіантів – 4-ри розрядний.

2. У завданні вказується таймер, за допомогою якого потрібно реалізувати відлік інтервалів часу для зазначеного пристрою.

3. Тривалість звукових сигналів вибирається студентом самостійно, дотримуючись вказаних меж.

4. Натиск кожної кнопки супроводжується коротким одинарним сигналом, успішне підтвердження введення даних за допомогою клавіші # супроводжується коротким подвійним сигналом, введення невірних даних за допомогою клавіші # та відміна усієї операції за допомогою клавіші \* супроводжується довгим сигналом.

Таймер відліку часу T3, клавіатура на port-K, порт даних LCD port-C, алгоритм 1:

**Годинник**

Відображає години, хвилини, секунди. Запускається одразу ж при подачі живлення на МК. При співпадінні значень будильника та годинника запускається звуковий сигнал (короткими гудками) тривалістю 0,5-2 хв.

**Кнопка A** – переводить у режим виставлення годин та хвилин годинника.

**Кнопка B** – переводить у режим виставлення годин та хвилин будильника.

\* Значення годин та хвилин вводиться за допомогою клавіш 0-9. Підтвердження введення за допомогою #, при цьому здійснюється перевірка коректності значень годин та хвилин (від 0 до 59).

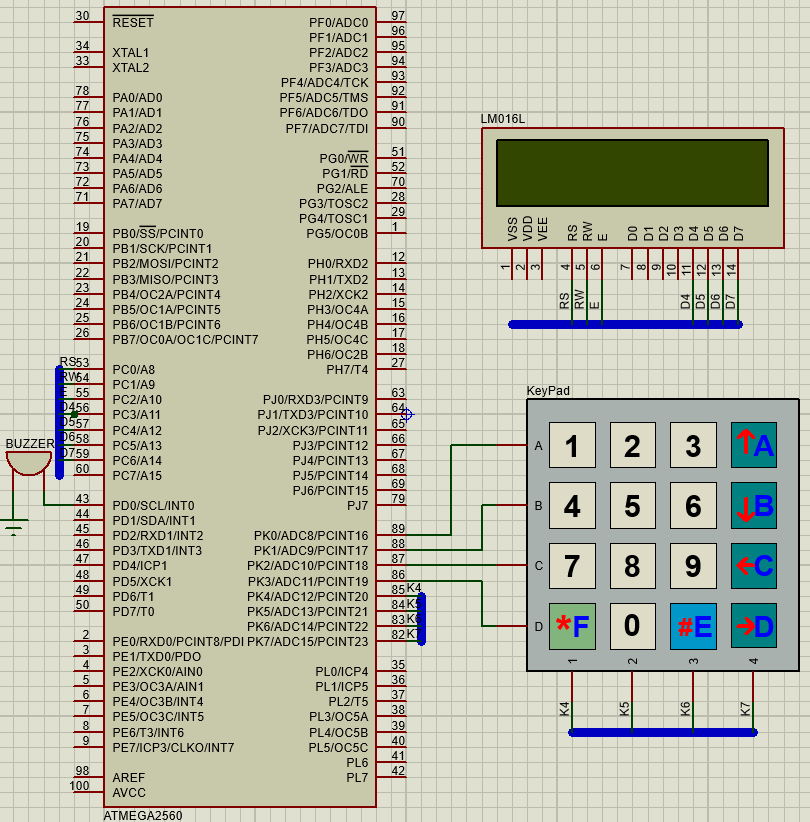
**Кнопка C** – відображає значення поточного будильника.

**Кнопка D** – активізує/відключає будильник (літера А) після натиснення кнопки С.

**Кнопка \*** – відміна операції, перехід у режим ходу годинника.

**Виконання роботи**

У пакеті симуляції Proteus зібрав необхідну схему на основі мікроконтролера ATMega2560:



Програма мовою C++ в Arduino IDE для реалізації вказаного алгоритму:

#include <LiquidCrystal.h>

#define DDR\_KEYPAD DDRK

#define PORT\_KEYPAD PORTK

#define PIN\_KEYPAD PINK

#include "keypad4x4.h"

const int buzzerPin = 21;

bool isAlarmSet = false;

int currentMode = 0;

int inputidx = 0;

int inputidxw = 0;

char currInput[6] = "XXXXXX";

const int rs = 37, rw = 36, en = 35, d4 = 34, d5 = 33, d6 = 32, d7 = 31;

LiquidCrystal lcd(rs, rw, en, d4, d5, d6, d7);

const PROGMEM char sixty[60][3] = {

  {"00"}, {"01"}, {"02"}, {"03"}, {"04"}, {"05"}, {"06"}, {"07"}, {"08"}, {"09"},

  {"10"}, {"11"}, {"12"}, {"13"}, {"14"}, {"15"}, {"16"}, {"17"}, {"18"}, {"19"},

  {"20"}, {"21"}, {"22"}, {"23"}, {"24"}, {"25"}, {"26"}, {"27"}, {"28"}, {"29"},

  {"30"}, {"31"}, {"32"}, {"33"}, {"34"}, {"35"}, {"36"}, {"37"}, {"38"}, {"39"},

  {"40"}, {"41"}, {"42"}, {"43"}, {"44"}, {"45"}, {"46"}, {"47"}, {"48"}, {"49"},

  {"50"}, {"51"}, {"52"}, {"53"}, {"54"}, {"55"}, {"56"}, {"57"}, {"58"}, {"59"}

};

struct Time

{

  unsigned char second, minute, hour;

};

Time T3 = {0, 0, 0};

Time Alarm = {0, 0, 0};

void LCD\_WriteStrPROGMEM(char \*str, int n)

{

  for (int i = 0; i < n; i++)

    lcd.print((char)pgm\_read\_byte(&(str[i])));

}

ISR(TIMER3\_COMPA\_vect)

{

  if (++T3.second == 60)

  {

    T3.second = 0;

    if (++T3.minute == 60)

    {

      T3.minute = 0;

      if (++T3.hour == 24)

        T3.hour = 0;

    }

  }

  if (isAlarmSet) {

    if (T3.hour == Alarm.hour && T3.minute == Alarm.minute && T3.second == Alarm.second) {

      digitalWrite(buzzerPin, HIGH);

    }

  }

  Time end = {0, 0, 0};

  end.second = Alarm.second + 30;

  if (end.second > 59) {

    end.second -= 60;

    end.minute += 1;

  }

  end.minute += Alarm.minute;

  if (end.minute > 59) {

    end.minute -= 60;

    end.hour += 1;

  }

  end.hour += Alarm.hour;

  if (end.hour > 23) {

    end.hour -= 24;

  }

  if (T3.hour == end.hour && T3.minute == end.minute && T3.second == end.second) {

    digitalWrite(buzzerPin, LOW);

  }

  lcd.setCursor(3, 0);

  LCD\_WriteStrPROGMEM(sixty[T3.hour], 2);

  lcd.write(':');

  LCD\_WriteStrPROGMEM(sixty[T3.minute], 2);

  lcd.write(':');

  LCD\_WriteStrPROGMEM(sixty[T3.second], 2);

}

void setup() {

  noInterrupts();

  TCCR3A = 0x00;

  TCCR3B = (1 << WGM32) | (1 << CS32) | (1 << CS30);

  TIMSK3 = (1 << OCIE3A);

  OCR3A = 0x3D08;

  initKeyPad();

  lcd.begin(16, 2);

  interrupts();

  pinMode(buzzerPin, OUTPUT);

  digitalWrite(buzzerPin, LOW);

}

// returns "true" if time is valid

bool validateTime (char time[]) {

  for (short i = 0; i <= sizeof(time)/sizeof(time[0]); i++) {

    if (time[i] == 'X') {

      return false;

    }

  }

  int hours = (time[0] - '0') \* 10 + (time[1] - '0');

  int minutes = (time[2] - '0') \* 10 + (time[3] - '0');

  int seconds = (time[4] - '0') \* 10 + (time[5] - '0');

  if (hours > 23) {

    return false;

  }

  else if (minutes > 59) {

    return false;

  }

  else if (seconds > 59) {

    return false;

  }

  else {

    return true;

  }

}

void clearInput () {

  lcd.setCursor(0, 1);

  lcd.write("                ");

  for (short i = 0; i < sizeof(currInput)/sizeof(currInput[0]); i++) {

    currInput[i] == 'X';

  }

  inputidx = 0;

  inputidxw = 0;

}

void showAlarm () {

  lcd.setCursor(0, 1);

  lcd.write("Alarm: ");

  LCD\_WriteStrPROGMEM(sixty[Alarm.hour], 2);

  lcd.write(':');

  LCD\_WriteStrPROGMEM(sixty[Alarm.minute], 2);

  lcd.write(':');

  LCD\_WriteStrPROGMEM(sixty[Alarm.second], 2);

  if (isAlarmSet) {

    lcd.write('+');

  } else {

    lcd.write('-');

  }

}

void loop() {

    if (isButtonPressed()) {

      digitalWrite(buzzerPin, HIGH);

      char pressedKey = readKeyFromPad4x4();

      lcd.setCursor(inputidxw, 1);

      switch (pressedKey) {

        case 'A':

          currentMode = 1;

          break;

        case 'B':

          currentMode = 2;

          break;

        case 'C':

          currentMode = 3;

          showAlarm();

          break;

        case 'D':

          if (currentMode == 3) {

            isAlarmSet = !isAlarmSet;

          }

          showAlarm();

          break;

        case 'F':

          clearInput();

          delay(400);

          currentMode = 0;

          break;

        case 'E':

          if (!validateTime(currInput) || inputidx > 6) {

            delay(400);

            break;

          }

          switch (currentMode) {

            case 1:

              T3.second = (currInput[4] - '0') \* 10 + (currInput[5] - '0');

              T3.minute = (currInput[2] - '0') \* 10 + (currInput[3] - '0');

              T3.hour = (currInput[0] - '0') \* 10 + (currInput[1] - '0');

              lcd.setCursor(3, 0);

              LCD\_WriteStrPROGMEM(sixty[T3.hour], 2);

              lcd.write(':');

              LCD\_WriteStrPROGMEM(sixty[T3.minute], 2);

              lcd.write(':');

              LCD\_WriteStrPROGMEM(sixty[T3.second], 2);

              delay(100);

              digitalWrite(buzzerPin, HIGH);

              break;

            case 2:

              Alarm.second = (currInput[4] - '0') \* 10 + (currInput[5] - '0');

              Alarm.minute = (currInput[2] - '0') \* 10 + (currInput[3] - '0');

              Alarm.hour = (currInput[0] - '0') \* 10 + (currInput[1] - '0');

              isAlarmSet = true;

              delay(100);

              digitalWrite(buzzerPin, HIGH);

              break;

            case 4:

              break;

            default:

              delay(400);

              break;

          }

          clearInput();

          currentMode = 0;

          break;

        default:

          if (currentMode == 0) {

            delay(400);

            break;

          }

          if (inputidxw == 2 || inputidxw == 5) {

            lcd.write(':');

            inputidxw += 1;

          }

          lcd.write(pressedKey);

          currInput[inputidx] = pressedKey;

          inputidx += 1;

          inputidxw += 1;

          break;

      }

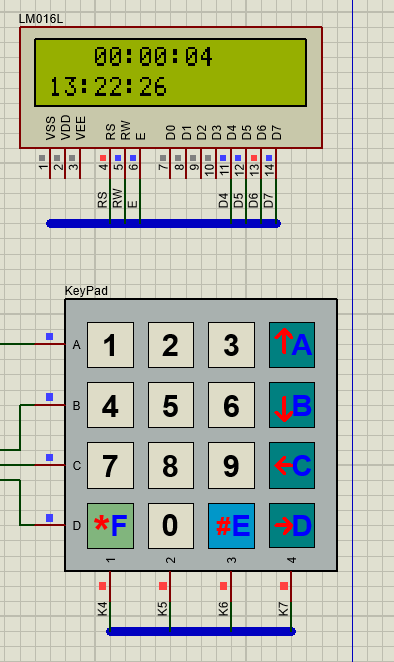
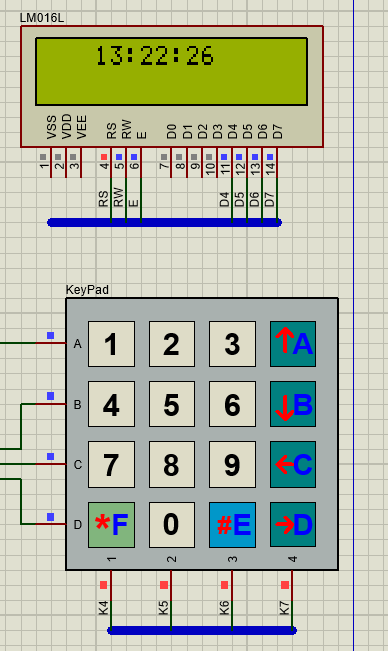
      delay(100);

      digitalWrite(buzzerPin, LOW);

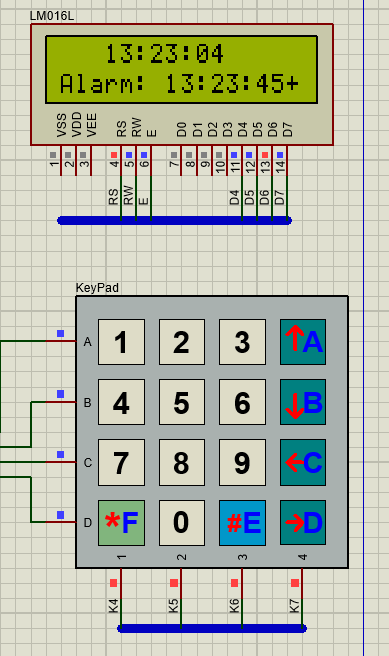
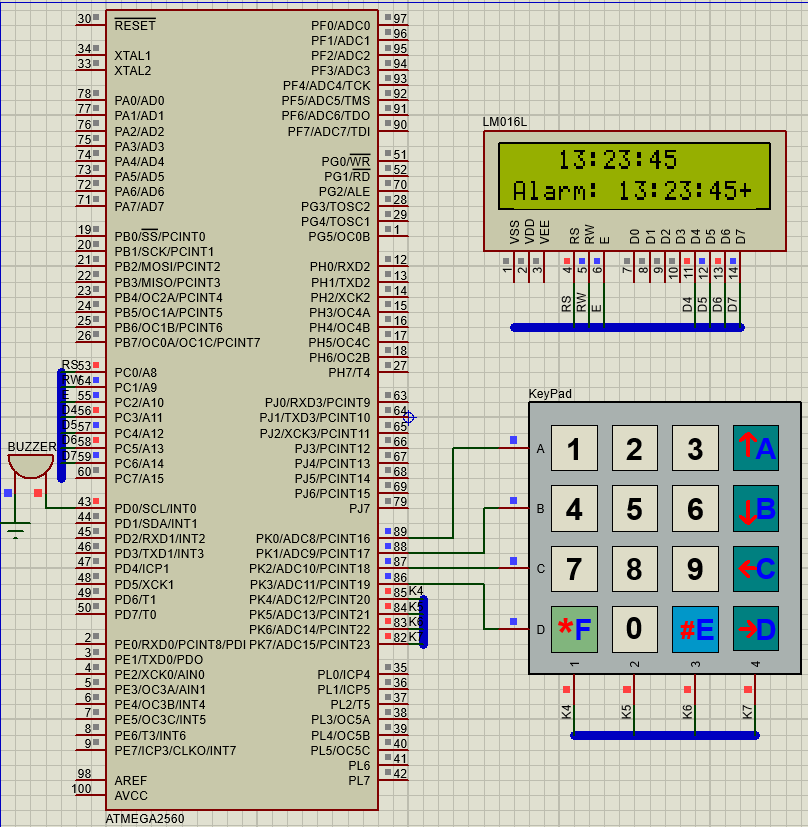
    }

}

Приклад роботи програми:

Встановлення часу на годиннику

Приклад встановленого будильника і його відпрацювання (Buzzer пищить протягом 30-ти секунд)

**Висновок:**

На цій лабораторній роботі я навчився працювати з перериваннями та таймерами на мікроконтролері, а також із LCD дисплеєм.