**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 №4**

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

**«Вивід інформації на семисегментні елементи,**

**що працюють в режимі динамічної індикації»**

**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 (порядкового номера в журналі викладача) завдання (таблиця 1) зібрати у пакеті симуляції Proteus схему на основі МК ATmega2560 та написати програму мовою Ci для реалізації вказаного алгоритму (таблиця 2).

2. Отримані на комп’ютері результати симуляції представити викладачу.

3. Виконати перевірку правильності роботи розробленого алгоритму на макеті.

**Завдання (варіант 18)**

1. У завданні вказується таймер за допомогою якого потрібно реалізувати відлік інтервалів часу для зазначеного пристрою. Динамічну індикацію дозволяється реалізувати за допомогою іншого таймера.

2. При використанні таймера 2 в асинхронному режимі передбачається, що він тактується від годинникового кварца 32768 Гц.

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

4. Парні варіанти вибирають підключення семисегментників зі спільним анодом, а непарні варіанти – зі спільним катодом.

Таймер T2(асин.), частота — 10МГц, кнопки PB(0-3), сигнал PC0, порт керування 7-сегментниками PORTA, виводи керування напругами PD(1-6), 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 1** – виставляє значення годин годинника або будильника.

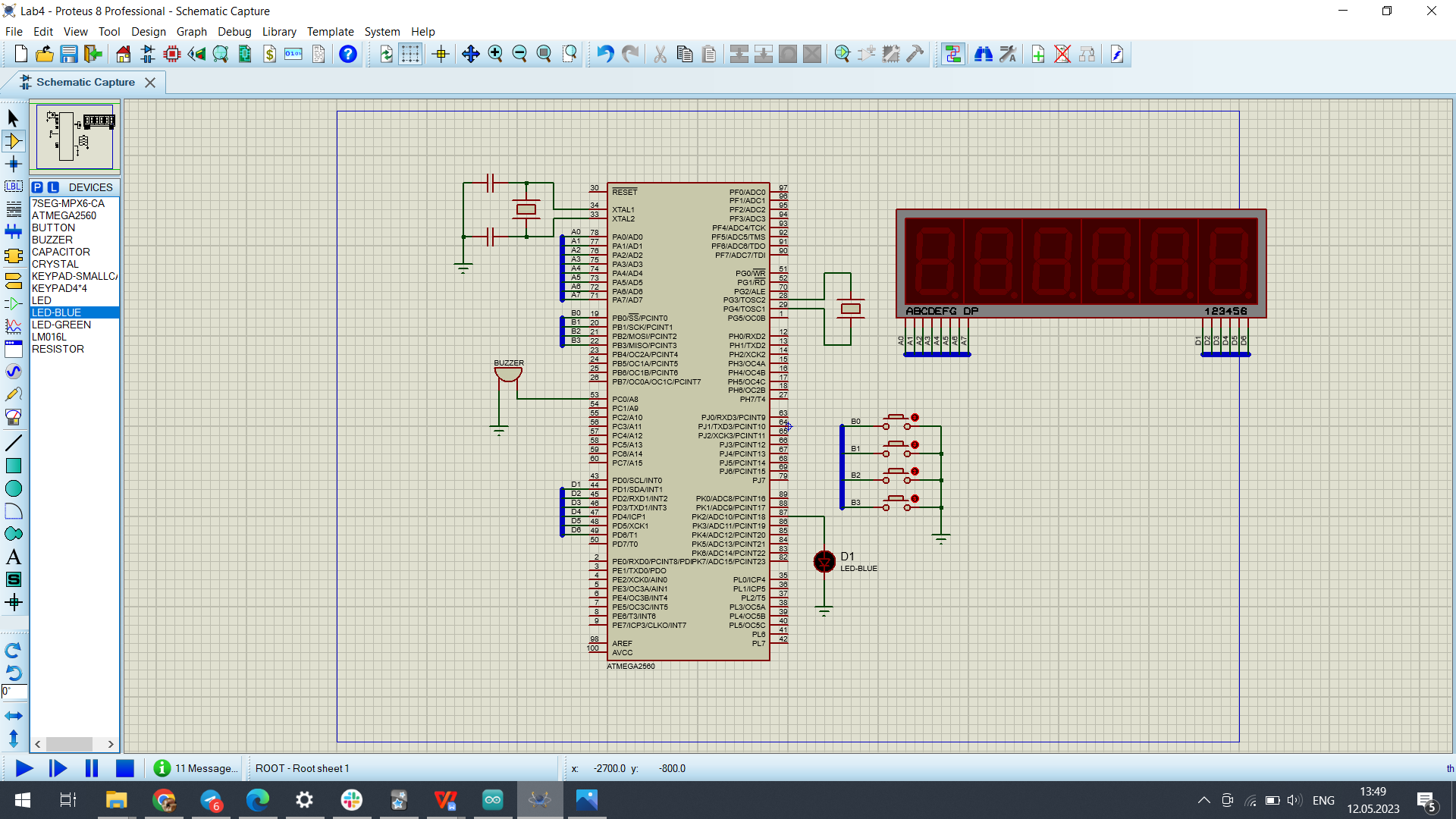
**Кнопка 2** – виставляє значення хвилин годинника або будильника.

**Кнопка 3** – відображає значення будильника та дає можливість кнопками 1 та 2 змінювати його значення.

**Кнопка 4** – активізує/відключає будильник (при натисненій кнопці 3), якщо будильник активований, то у секції секунд відображається літера А.

**Виконання роботи**

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



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

const int buzzerPin = 37;

unsigned char number[10]  = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90};

unsigned char bcd\_time[6] = {number[0], number[0], number[0], number[0], number[0], number[0]};

unsigned char alarm\_time[6] = {number[0], number[0], number[0], number[0], number[0], number[0]};

unsigned char portd\_positions[6] = {0b00000010, 0b00000100, 0b00001000, 0b00010000, 0b00100000, 0b01000000};

struct Time

{

  unsigned char second, minute, hour;

};

Time T2 = {0, 0, 0};

Time Alarm = {0, 0, 0};

unsigned char fig\_number = 0;

bool clock\_mode = true;

bool alarm\_state = false;

void bcd(unsigned char number, unsigned char position);

ISR(TIMER2\_COMPA\_vect) {

  if (++T2.second == 60) {

    T2.second = 0;

    if (++T2.minute == 60) {

      T2.minute = 0;

      if (++T2.hour == 24)

        T2.hour = 0;

      bcd(T2.hour, 0);

    }

    bcd(T2.minute, 2);

  }

  bcd(T2.second, 4);

}

ISR(TIMER0\_COMPA\_vect) {

  fig\_number++;

  if (fig\_number == 6) {

    fig\_number = 0;

  }

  PORTD = portd\_positions[fig\_number];

  if (clock\_mode) {

    if (fig\_number == 1 || fig\_number == 3) {

      PORTA = bcd\_time[fig\_number] & ~0x80;

    } else {

      PORTA = bcd\_time[fig\_number];

    }

  } else {

    if (fig\_number == 1 || fig\_number == 3) {

      PORTA = alarm\_time[fig\_number] & ~0x80;

    } else {

      PORTA = alarm\_time[fig\_number];

    }

  }

  if (alarm\_state) {

    if (T2.hour == Alarm.hour && T2.minute == Alarm.minute && T2.second == Alarm.second) {

      digitalWrite(buzzerPin, HIGH);

    }

  }

  if (alarm\_state && !clock\_mode) {

    if (PORTD == 0b01000000) {

      PORTA = 0x88;

    }

  }

  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 (T2.hour == end.hour && T2.minute == end.minute && T2.second == end.second) {

    digitalWrite(buzzerPin, LOW);

  }

}

void setup() {

  noInterrupts();

  DDRD = 0xFF;

  PORTD = 0xFF;

  DDRA = 0xFF;

  PORTA = 0xFF;

  DDRB = 0x00;

  PORTB = 0xFF;

  TCNT0 = 0;

  OCR0A = 25;

  TCCR0A = (1 << WGM01);

  TCCR0B = (1 << CS02) | (1 << CS00);

  TIMSK0 |= (1 << OCIE0A);

  TIMSK2 = (0 << OCIE2A) | (0 << OCIE2B) | (0 << TOIE2);

  ASSR |= 1 << AS2;

  TCNT2 = 0;

  TCCR2A = (1<<COM2A0) | (1<<WGM21);

  TCCR2B = (1<<CS22) | (1<<CS21) | (1<<CS20);

  OCR2A = 32;

  while ((ASSR & ((1 << OCR2AUB) | (1 << OCR2BUB) | (1 << TCR2AUB) | (1 << TCR2BUB) | (1<< TCN2UB))));

  TIFR2 = (1 << TOV2) | (1 << OCF2A) | (1 << OCF2B);

  TIMSK2 = (1 << OCIE2A);

  interrupts();

  pinMode(buzzerPin, OUTPUT);

  digitalWrite(buzzerPin, LOW);

}

void loop() {

  if (clock\_mode) {

    if((PINB&(1<<0))==0) { // increment hours

      if (++T2.hour == 24) T2.hour = 0;

      bcd(T2.hour, 0);

      delay(100000);

    }

    if((PINB&(1<<1))==0) { // increment minutes

      if (++T2.minute == 60) T2.minute = 0;

      bcd(T2.minute, 2);

      delay(100000);

    }

  } else {

    if((PINB&(1<<0))==0) { // increment hours

      if (++Alarm.hour == 24) Alarm.hour = 0;

      alarm(Alarm.hour, 0);

      delay(100000);

    }

    if((PINB&(1<<1))==0) { // increment minutes

      if (++Alarm.minute == 60) Alarm.minute = 0;

      alarm(Alarm.minute, 2);

      delay(100000);

    }

  }

  if((PINB&(1<<2))==0) { // change mode

    clock\_mode = !clock\_mode;

    delay(100000);

  }

  if((PINB&(1<<3))==0) { // toggle alarm state

    alarm\_state = !alarm\_state;

    delay(100000);

  }

}

void bcd(unsigned char fig\_in, unsigned char position) {

  unsigned char bcdL = fig\_in;

  unsigned char bcdH = 0;

  while (bcdL >= 10) {

    bcdL -= 10;

    bcdH++;

  }

  bcd\_time[position] = number[bcdH];

  bcd\_time[position + 1] = number[bcdL];

}

void alarm(unsigned char fig\_in, unsigned char position) {

  unsigned char bcdL = fig\_in;

  unsigned char bcdH = 0;

  while (bcdL >= 10) {

    bcdL -= 10;

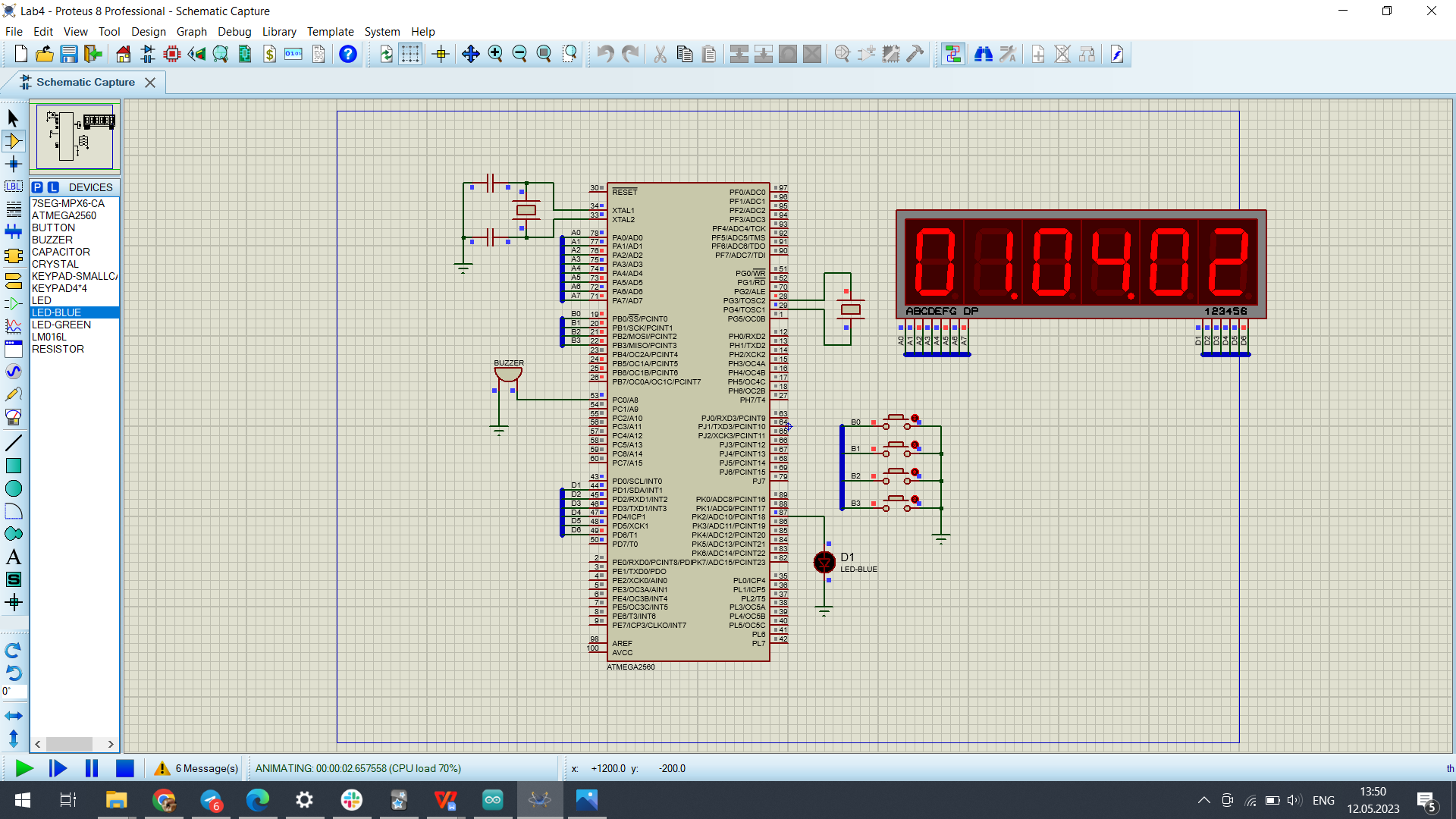
    bcdH++;

  }

  alarm\_time[position] = number[bcdH];

  alarm\_time[position + 1] = number[bcdL];

}



Демонстрація виконання програми на МК

**Висновок:**

На цій лабораторній роботі я навчився працювати з асинхронним таймером, а також познайомився і навчився працювати із семисегментними елементами.