**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**



Звіт до лабораторної роботи №4

з навчальної дисципліни: «Мікроконтролери ч.1»

**Вивід інформації на семисегментні елементи,**

**що працюють в режимі динамічної індикації**

Варіант №11

**Виконав:**

ст. гр. ІР-22

Матвійчук А.І.

**Прийняв:**

Ст. викладач кафедри КСА

Лагун І.І.

Львів – 2021

**Порядок виконання роботи:**

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. 4. За результатами виконаної роботи оформити звіт та

здати його.

**Завдання:**

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,

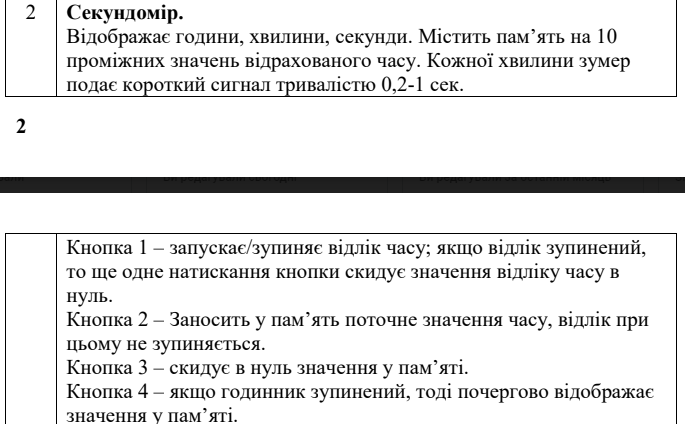
що він тактується від годинникового кварца 32768 Гц.

3. Тривалість звукових сигналів вибирається студентом самостійно,

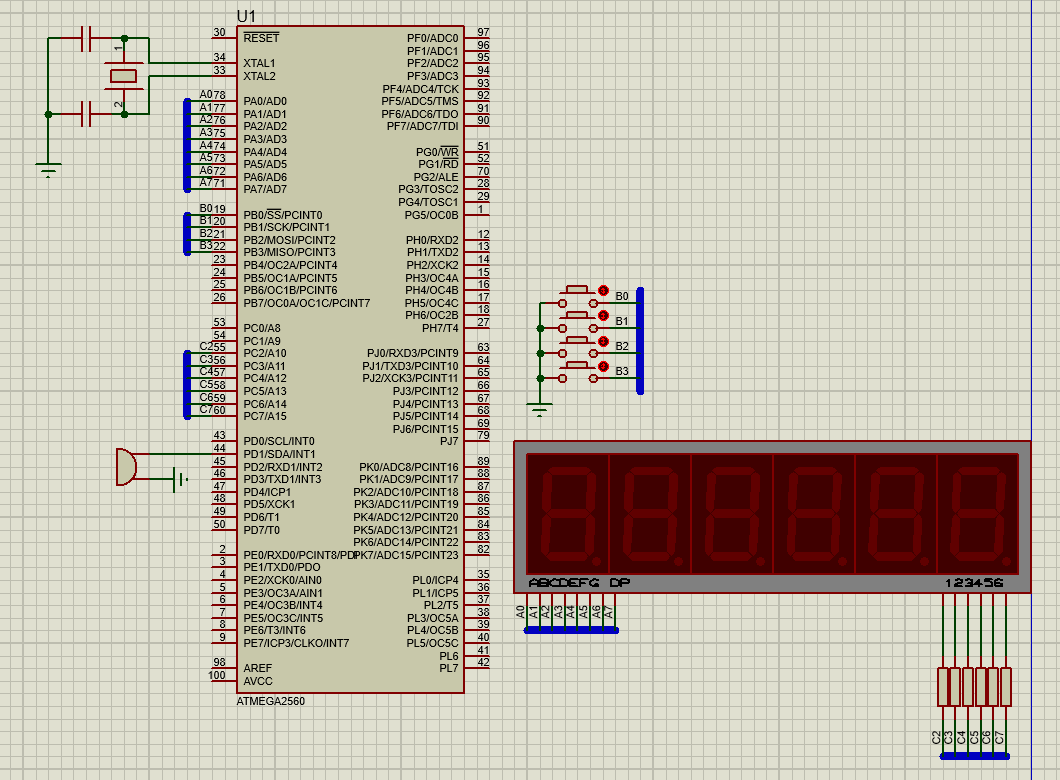
дотримуючись вказаних меж.

4. Парні варіанти вибирають підключення семисегментників зі

спільним анодом, а непарні варіанти – зі спільним катодом. 

****

**Схема у Proteus:**



**Відео презентація лабораторної роботи:**

<https://drive.google.com/file/d/1TeeTYjjRcITR7RMmsrUG_O2qajiYi3ir/view>

**Код прошивки:**

**Lab4.cpp**

#define NUM\_DIGIT 6

#define PORT\_CONTROL\_SEGMENT PORTA

#define DDR\_CONTROL\_SEGMENT DDRA

#define PORT\_CONTROL\_VOLTAGE PORTC

#define DDR\_CONTROL\_VOLTAGE DDRC

#define PORT\_BUTTONS PORTB

#define DDR\_BUTTONS DDRB

#define PIN\_BUTTONS PINB

#define BUZZER 20

struct Time

{

unsigned char second, minute, hour;

};

Time T4 = {0, 0, 0};

String timeString = "";

boolean startCounting = 0;

Time arrTime[10];

byte countTime = 0;

boolean journalMode = 0;

void timeToString(Time pattern){

timeString = pattern.hour>9 ? String(pattern.hour) : '0' + String(pattern.hour) + '.';

timeString += pattern.minute>9 ? String(pattern.minute) : '0' + String(pattern.minute) + '.';

timeString += pattern.second>9 ? String(pattern.second) : '0' + String(pattern.second);

}

ISR(TIMER1\_COMPA\_vect)

{

if(!journalMode){

if (startCounting){

if (++T4.second == 60)

{

T4.second = 0;

if (++T4.minute == 60)

{

T4.minute = 0;

if (++T4.hour == 24)

T4.hour = 0;

}

}

}

timeToString(T4);

}

}

const byte segmentNumber[10] = {

0b00111111,

0b00000110,

0b01011011,

0b01001111,

0b01100110,

0b01101101,

0b01111101,

0b00000111,

0b01111111,

0b01101111,

};

void setup() {

noInterrupts();

TCCR1A = 0x00;

TCCR1B = (1 << WGM12) | (1 << CS12) | (1 << CS10);

TIMSK1 = (1 << OCIE1A);

OCR1A = 0x3D08;

interrupts(); // Enable global interrupts

DDR\_CONTROL\_VOLTAGE = 0b11111111;

PORT\_CONTROL\_VOLTAGE = 0b00000000;

DDR\_CONTROL\_SEGMENT = 0b11111111;

PORT\_CONTROL\_SEGMENT = 0b00000000;

DDR\_BUTTONS = 0x0F;

PORT\_BUTTONS = 0xF0;

Serial.begin(9600);

pinMode(53,INPUT\_PULLUP);

pinMode(52,INPUT\_PULLUP);

pinMode(51,INPUT\_PULLUP);

pinMode(50,INPUT\_PULLUP);

pinMode(BUZZER,OUTPUT);

}

void loop() {

showSegment(timeString);

if(T4.second == 59) digitalWrite(BUZZER,HIGH);

else if (T4.second == 0)digitalWrite(BUZZER,LOW);

static byte postDown = 2;

static byte journalScroll = 0;

static byte pinButtonsOld = PIN\_BUTTONS;

byte pinButtonsNow = PIN\_BUTTONS;

if(pinButtonsNow == 255){

if (pinButtonsOld == 0b11111110){

if (journalMode)journalMode = 0;

else {

if (postDown<2)postDown++;

else postDown = 0;

if (postDown==0)startCounting = 1;

else if (postDown==1)startCounting = 0;

else T4 = {0, 0, 0};

}

}

else if (pinButtonsOld == 0b11111101 && !journalMode){

arrTime[countTime] = T4;

if(countTime<10)countTime++;

}

else if (pinButtonsOld == 0b11111011 && !journalMode){

countTime=0;

}

else if (pinButtonsOld == 0b11110111){

if (startCounting == 0 && countTime>0)journalMode = 1;

if (journalMode){

timeToString(arrTime[journalScroll]);

if(journalScroll<countTime-1)journalScroll++;

else journalScroll = 0;

}

}

}

pinButtonsOld = pinButtonsNow;

}

void showSegment(String value){

String clearValue = value;

byte pointRow[6]={0,0,0,0,0,0};

while(true){

int index = clearValue.indexOf(".");

if (index == -1) break;

clearValue.remove(index,1);

pointRow[index-1] = 0b10000000;

}

int clearValueLength = clearValue.length();

if (clearValueLength > NUM\_DIGIT+1) return;

PORT\_CONTROL\_VOLTAGE = 0b01111111;

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

PORT\_CONTROL\_SEGMENT = segmentNumber[clearValue[clearValueLength-i-1] - '0'] | pointRow[clearValueLength-i-1];

delay(1);

PORT\_CONTROL\_VOLTAGE = (PORT\_CONTROL\_VOLTAGE>>1) ^ 0b10000000;

}

}

**Висновок:**

В даній лабораторній роботі я навчився виводити інформацію на семисегментні елементи, що працюють в режимі динамічної індикації.