**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 №2**

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

**«Просте керування мікроконтролером за**

**допомогою клієнтської програми на комп'ютері»**

**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.

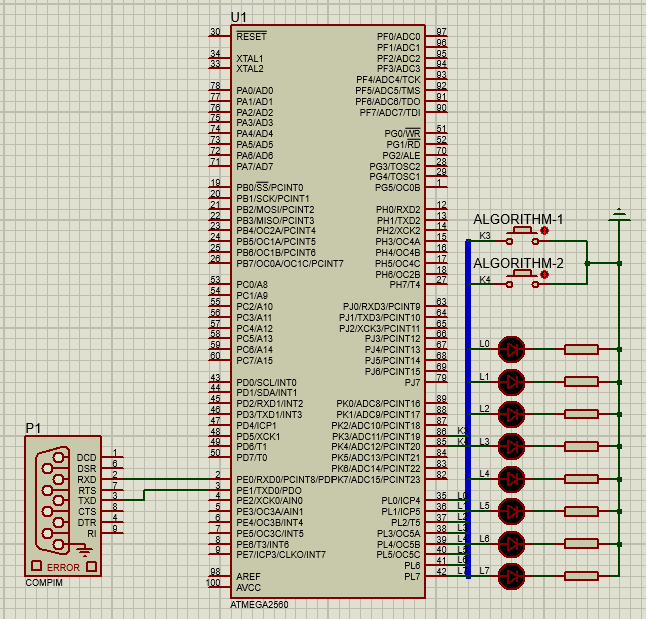
5. Отримані результати представити викладачу.

**Завдання (варіант 18)**

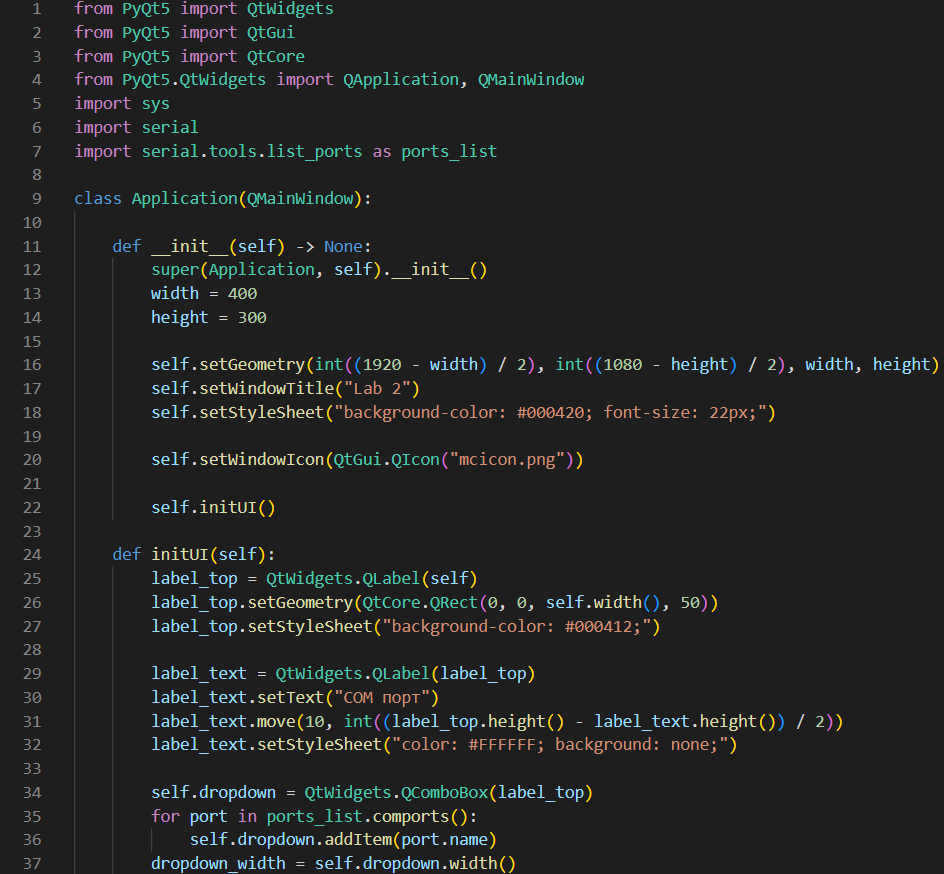
Кнопки PK3, PK4, світлодіоди PORT-L, затримка 0.4 секунди, алгоритми 2 (Лінійка з 8-ми одноколірних світлодіодів. При натисканні кнопки світлодіоди почергово блимають від 7-виводу порту до 0) та 5 (Лінійка з 8-ми одноколірних світлодіодів. При натисканні кнопки світлодіоди починають почергово блимати через один від 0-виводу порту до 6, а потім далі від 1 до 7).

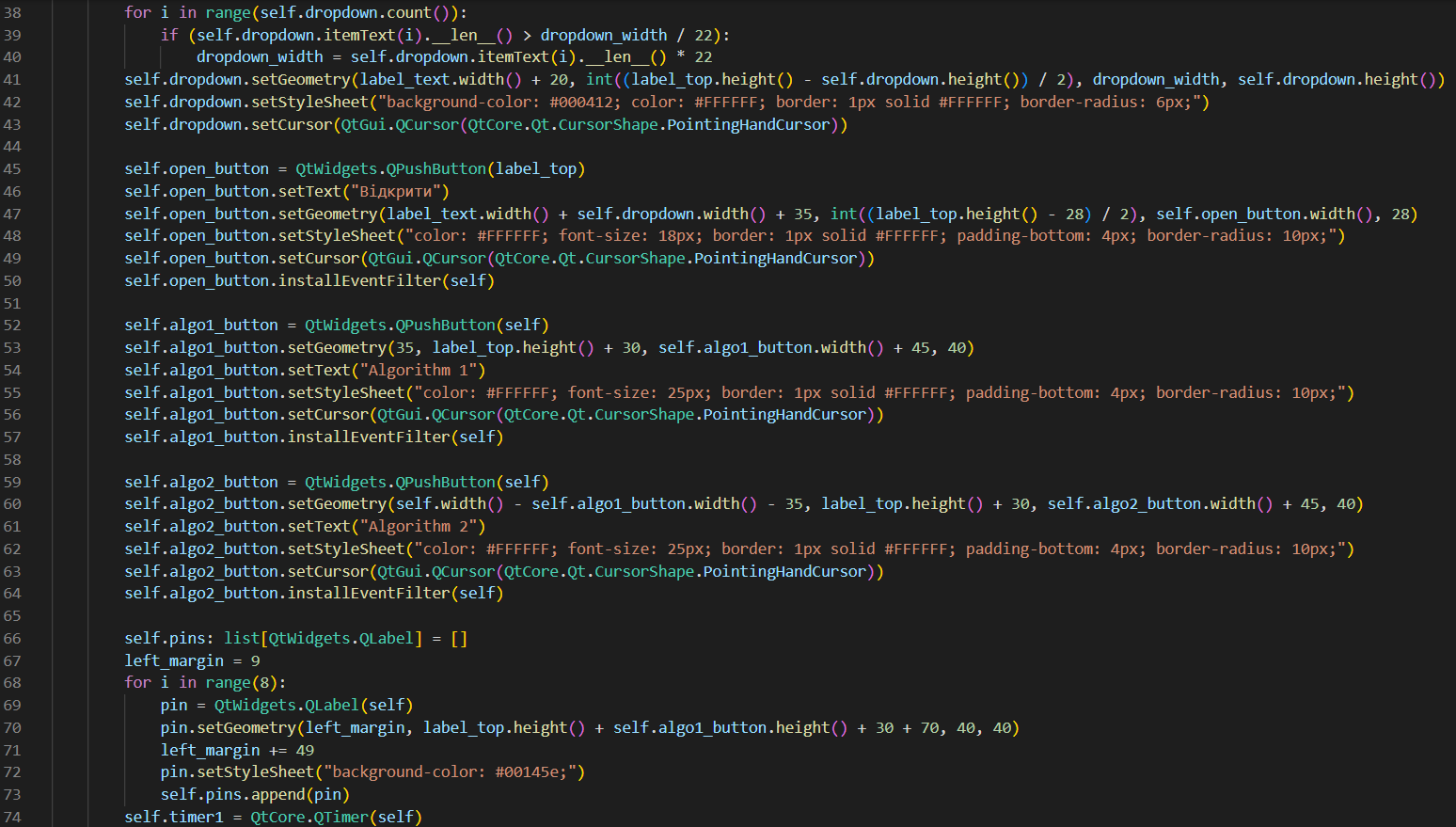
**Виконання роботи**

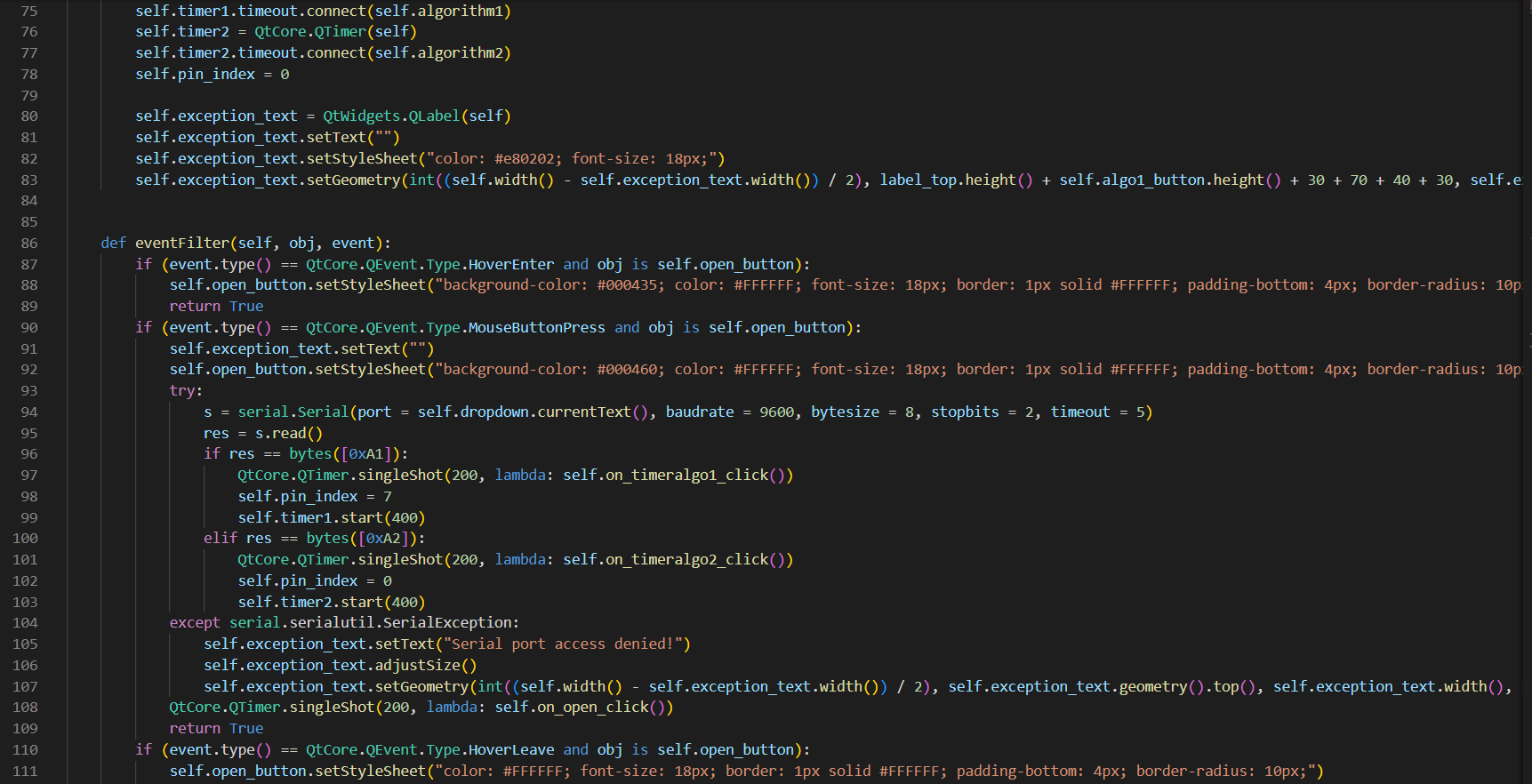
У пакеті симуляції Proteus зібрав необхідну схему на основі мікроконтролера ATMega2560:

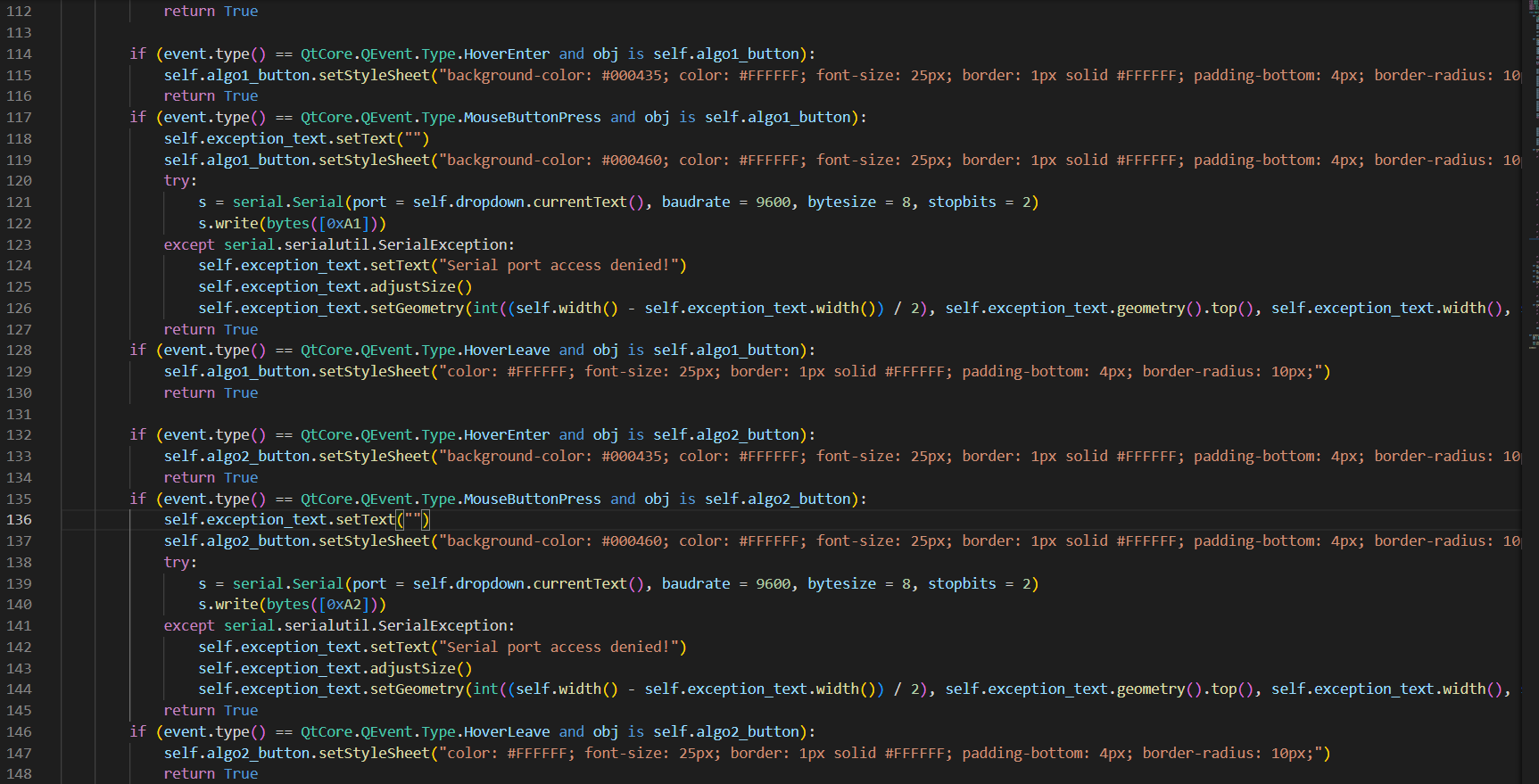


Код клієнтської програми на мові Python з використанням бібліотеки PyQt5 для керування МК:

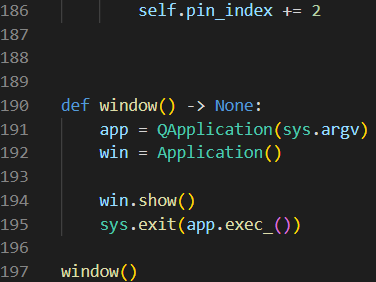












Програма мовою C++ в Arduino IDE для реалізації вказаного алгоритму:

const unsigned int delayInMs = 400;

const unsigned char buttonPinAlgo1 = 65;

const unsigned char buttonPinAlgo2 = 66;

int inByte;

const int algo1\_states[] = {0b00000001, 0b00000010, 0b00000100, 0b00001000, 0b00010000, 0b00100000, 0b01000000, 0b10000000};

const int algo2\_states[] = {0b10000000, 0b00100000, 0b00001000, 0b00000010, 0b01000000, 0b00010000, 0b00000100, 0b00000001};

void setup()

{

  for(uint16\_t i = 42; i < 50; i++)

  {

    pinMode(i, OUTPUT);

    digitalWrite(i, LOW);

  }

  pinMode(buttonPinAlgo1, INPUT\_PULLUP);

  pinMode(buttonPinAlgo2, INPUT\_PULLUP);

  Serial.begin(9600);

}

void loop()

{

  if (Serial.available() > 0)

  {

    inByte = Serial.read();

    if (inByte == 0xA1)

    {

      algorithm1();

    }

    else if (inByte == 0xA2)

    {

      algorithm2();

    }

  }

  if(digitalRead(buttonPinAlgo1) == LOW) {

    Serial.write(0xA1);

  }

  if(digitalRead(buttonPinAlgo2) == LOW) {

    Serial.write(0xA2);

  }

}

void algorithm1()

{

  for (uint16\_t i = 41; i <= 50; i++)

  {

    if (i != 50)

    {

      digitalWrite(i, HIGH);

    }

    if (i > 42)

    {

      digitalWrite(i-1, LOW);

    }

    delay(delayInMs);

  }

}

void algorithm2()

{

  for (uint16\_t i = 49; i > 41; i -= 2)

  {

    digitalWrite(i, HIGH);

    if (i < 48)

    {

      digitalWrite(i + 2, LOW);

    }

    if (i == 48)

    {

      digitalWrite(43, LOW);

    }

    if (i == 43)

    {

      i = 50;

    }

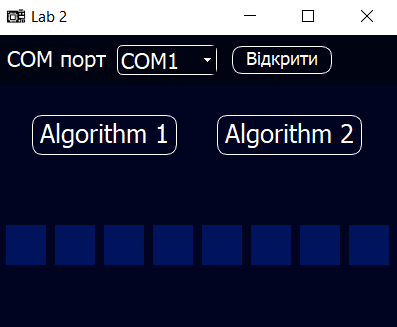
    delay(delayInMs);

  }

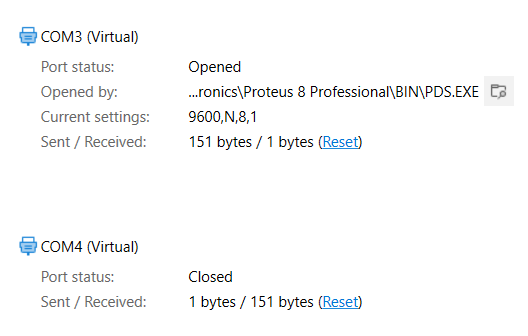
  digitalWrite(42, LOW);

}

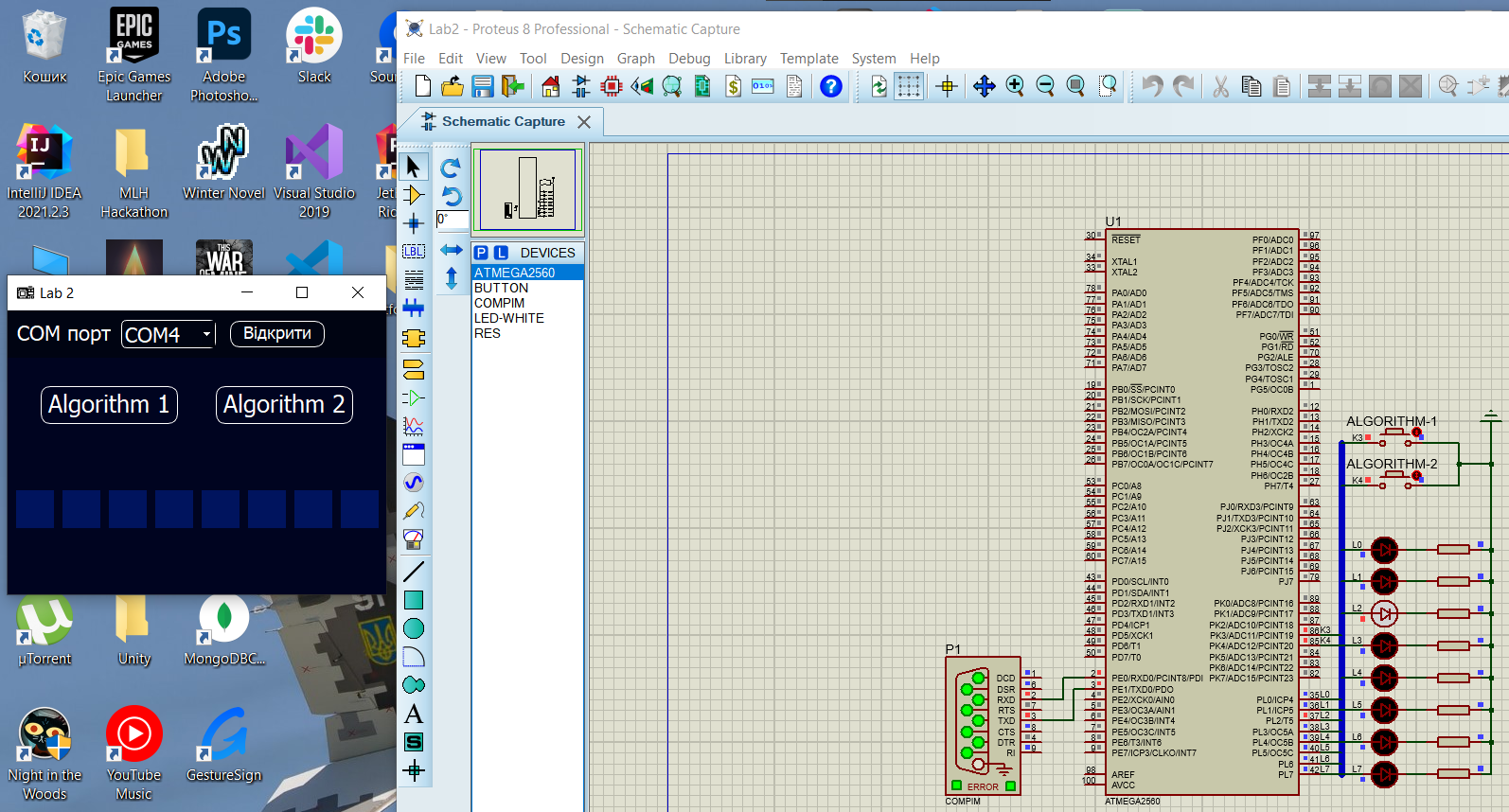
Інтерфейс клієнтської програми:



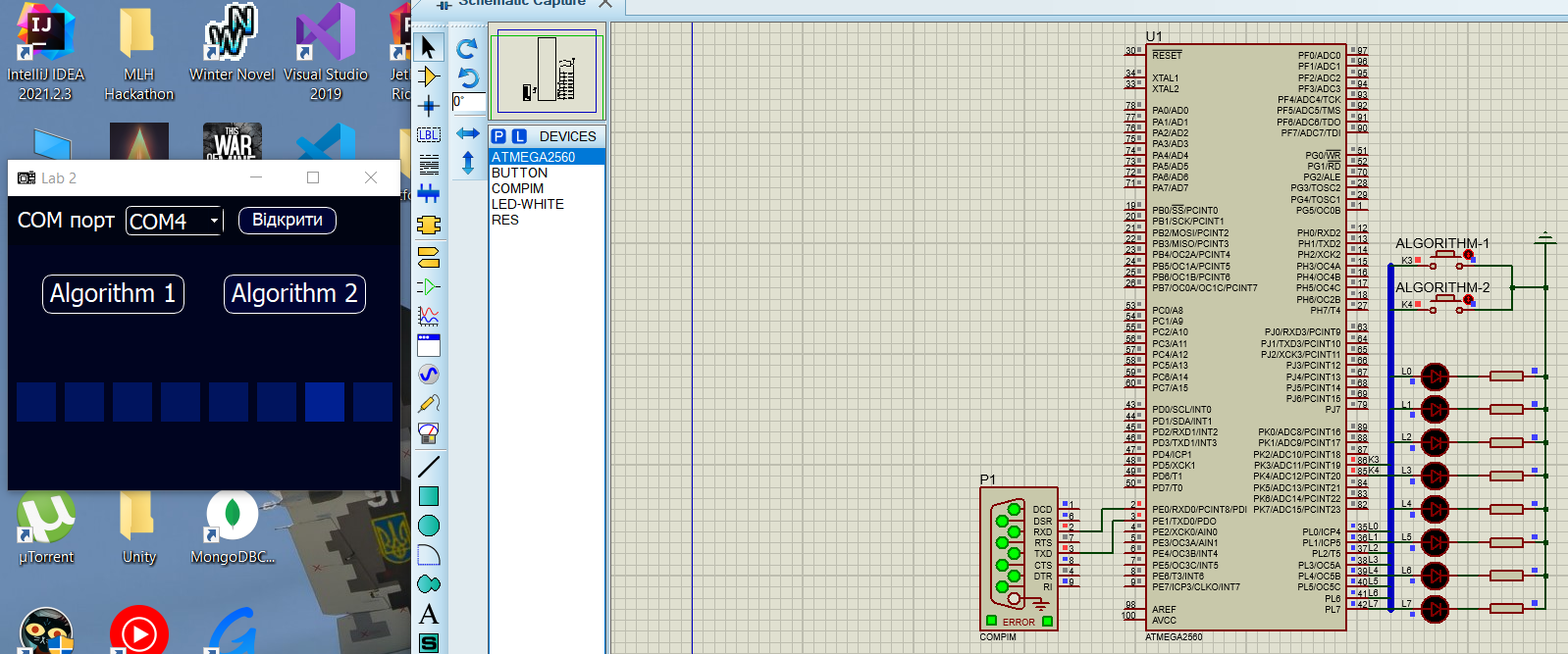
Створені віртуальні COM-порти для комунікації з МК:



Приклад виконання алгоритму:



Виконання алгоритму на Arduino Mega 2560 в програмі симуляції



Виконання алгоритму в клієнтській програмі

**Висновок:**

На цій лабораторній роботі я навчився писати клієнтські програми з допомогою PyQt, створювати віртуальні зв’язки COM-портів у програмі Virtual Serial Port Driver, передавати інформацію по COM-порту з клієнтської програми на МК та навпаки, і аналізувати отриману інформацію.