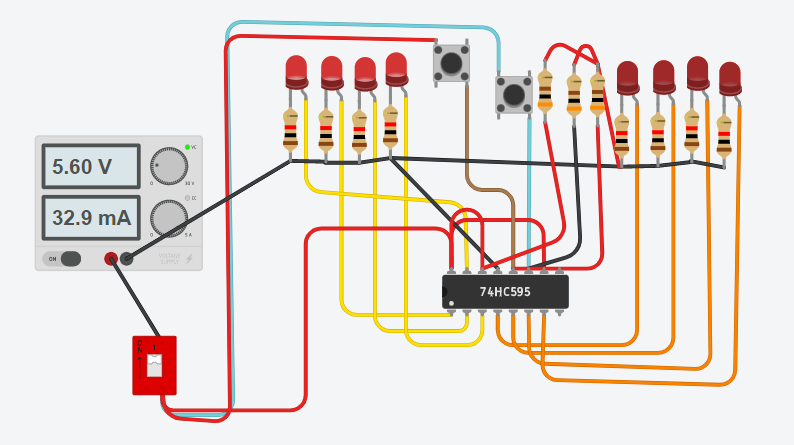
**ЛАБОРАТОРНА РОБОТА №5**

**Мета заняття:** ознайомитися з принципами введення/виведення цифрових сигналів. Вивчити роботу дискретних електронних компонентів (світлодіод, пьезовипромінювач, кнопка тощо) та використання плати Arduino. Ознайомитись з середовищем розробки ArduinoIDE

**Початок роботи**

**Завдання 1: Зібрати схему (рисунок 5.8) та вивчити роботу регістру зсуву 74HC595**

Розпочнемо виконання лабораторної роботи. Для початку, відповідно до отриманого завдання, проведемо ознайомлення з принципом роботи мікросхеми 74HC595.Зазначена вище мікросхема являється простим восьмирозрядним регістром зсуву. Дана схема займається зберігання та виконання зсуву бітів даних, котрів в межах нашої роботи будуть застосовуватися для керування семисигментним індикатором. Для керування схемою використовуються три основні контакти, а саме контакт «ввод», контакт «вивод», контакт «зсув».

****

**Посилання на проект:**

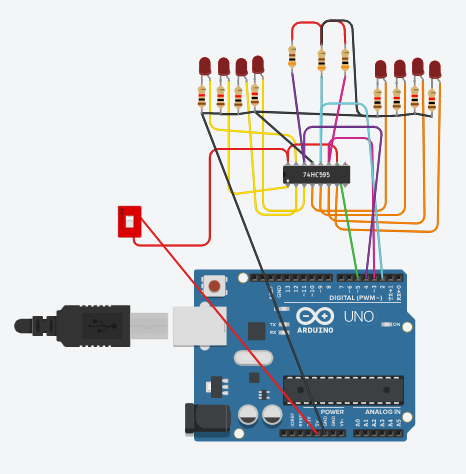
[**https://www.tinkercad.com/things/ldAuBhGpDyl**](https://www.tinkercad.com/things/ldAuBhGpDyl)

**Завдання №2**. Замінити управління перемикачами на управління за допомогою Ардуіно. Реалізувати “Мерехтливі вогні” у відповідності варіанту (див. таблицю 5.1).



****

Виконаємо модифікацію проекту. Для цього додамо платформу Arduino UNO, на основі мікроконтролера якої будемо виконувати керування схеми для побудови мерехтливих вогнів. В результаті проведених модифікацій, схема матиме наступний вигляд.

****

**Код Arduino**

// C++ code

int level = 3;

int output = 2;

int input = 4;

int delTime = 300;

int del = 5;

int step = 1;

/\*відкинув зайві нулі\*/

int a[8] = {1,0,0,0,1};

int a2[8] = {1,0,1,1,1,0,1,1};

int a3[8] = {1,1,0,1,1,1,0,1};

int a4[8] = {0,1,1,0,0,1,1};

void setup()

{

  pinMode(LED\_BUILTIN, OUTPUT);

  pinMode(3,OUTPUT);

  pinMode(4,OUTPUT);

  pinMode(2,OUTPUT);

    pinMode(5,OUTPUT);

}

void loop()

{

  if(step==5){

  step =1;

  }

  if(step){

    for(int i=(sizeof(a)/sizeof(a[0]))-1;0<=i;i--){

    int d = a[i];

    digitalWrite(input, a[i]);

    digitalWrite(level,1);

    digitalWrite(level,0);

  }

 }

    if(step==2){

     for(int i=(sizeof(a2)/sizeof(a2[0]))-1;0<=i;i--){

    digitalWrite(input, a2[i]);

    digitalWrite(level,1);

    digitalWrite(level,0);

  }

    }

    if(step==3){

   for(int i=(sizeof(a3)/sizeof(a3[0]))-1;0<=i;i--){

    digitalWrite(input, a3[i]);

    digitalWrite(level,1);

    digitalWrite(level,0);

  }

    }

  if(step==4){

     for(int i=(sizeof(a4)/sizeof(a4[0]))-1;0<=i;i--){

    digitalWrite(input, a4[i]);

    digitalWrite(level,1);

    digitalWrite(level,0);

  }

  }

  digitalWrite(output, 1);

  digitalWrite(output, 0);

  step++;

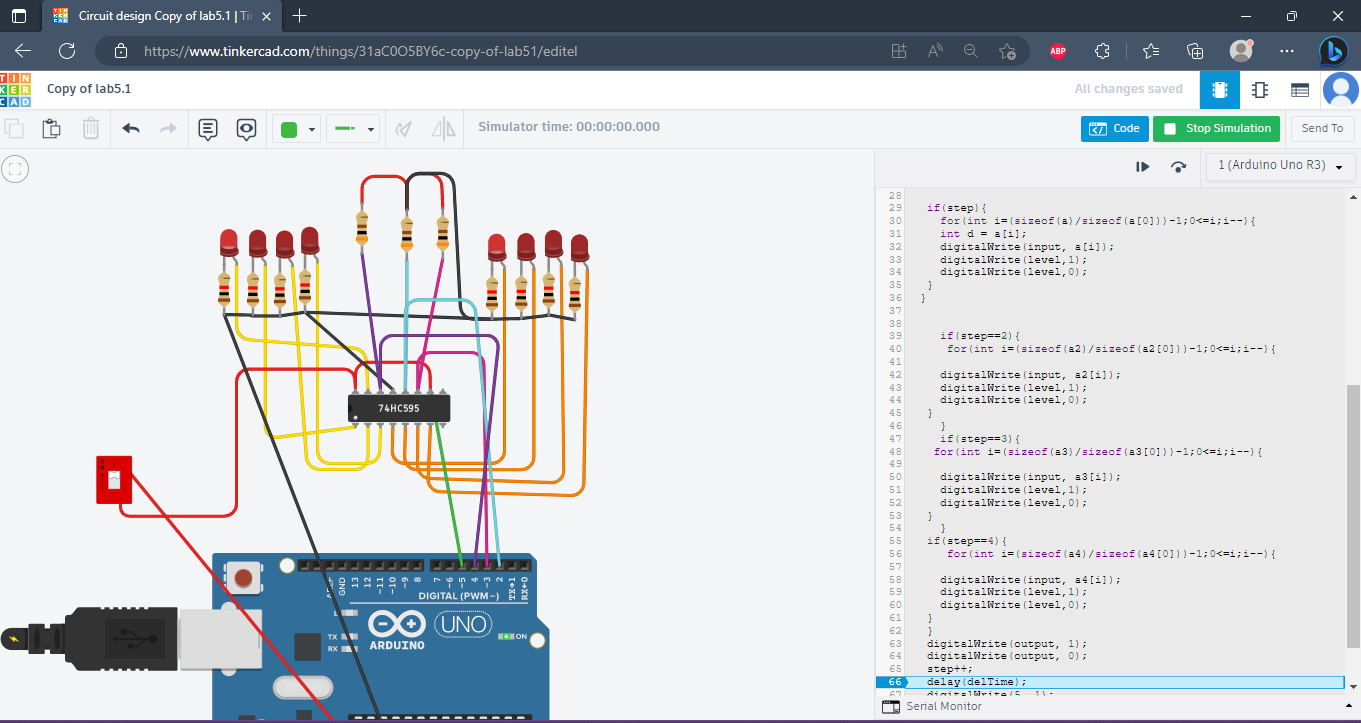
  delay(delTime);

  digitalWrite(5, 1);

  digitalWrite(5, 0);

}

**Результат роботи:**

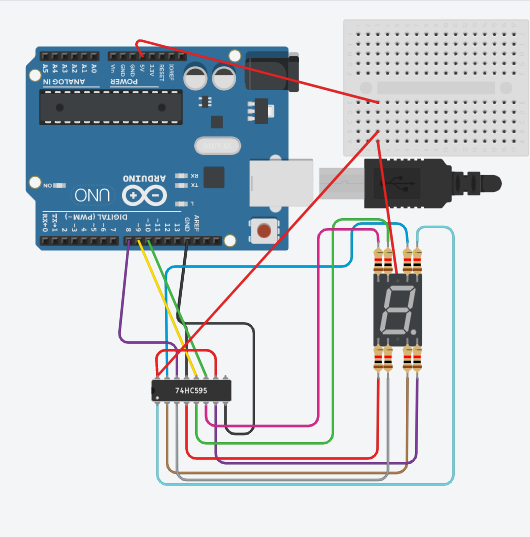
****

**Посилання на проект:**

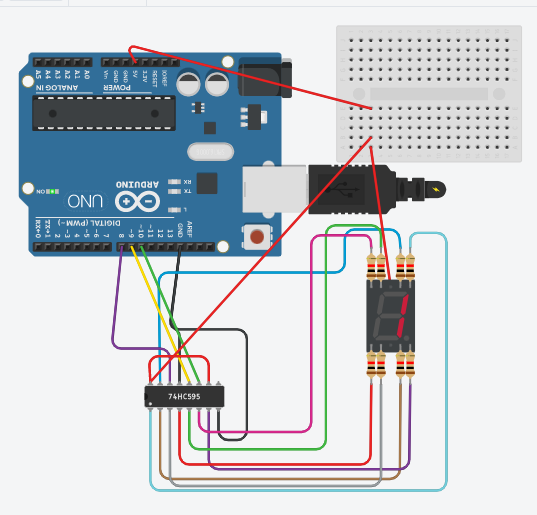
[**https://www.tinkercad.com/things/31aC0O5BY6c**](https://www.tinkercad.com/things/31aC0O5BY6c)

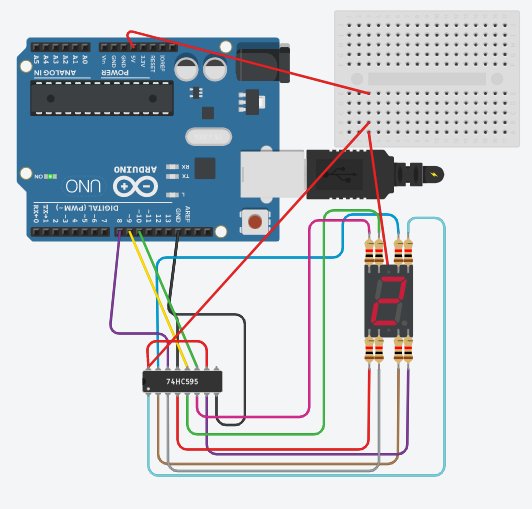
**Завдання №3.** Замінити світлодіоди на семисегментний індикатор та реалізувати функцію відображення цифр.

Виконаємо модифікацію схеми, додамо до неї семисигментний індикатор та напишемо функцію керування ним, для відораження числових символів. Схема матиме наступний вигляд.

****

**Результат роботи програми**

****

****

**Код:**

// C++ code

int level = 10;

int output = 9;

int input = 8;

int delTime = 300;

int del = 5;

int zero[8] = {0,0,1,1,1,1,1,1};

int one[8] = {0,0,0,0,0,1,1,0};

int two[8] = {0,1,0,1,1,0,1,1};

int three[8] = {0,1,0,0,1,1,1,1};

int four[8] = {0,1,1,0,0,1,1,0};

int five[8] = {0,1,1,0,1,1,0,1};

int six[8] = {0,1,1,1,1,1,0,1};

int seven[8] = {0,0,0,0,0,1,1,1};

int eight[8] = {0,1,1,1,1,1,1,1};

int nine[8] = {0,1,1,0,1,1,1,1};

int step = 1;

/\*відкинув зайві нулі\*/

void setup()

{

  pinMode(LED\_BUILTIN, OUTPUT);

  pinMode(level,OUTPUT);

  pinMode(output,OUTPUT);

  pinMode(input,OUTPUT);

  pinMode(del,OUTPUT);

}

void loop()

{

 setNumber(one);

 delay(1000);

  setNumber(two);

  delay(1000);

  setNumber(three);

  delay(1000);

  setNumber(four);

  delay(1000);

  setNumber(five);

  delay(1000);

  setNumber(six);

  delay(1000);

  setNumber(seven);

   delay(1000);

  setNumber(eight);

     delay(1000);

  setNumber(nine);

}

void setNumber(int a[]){

  for(int i=0;i<8;i++){

  digitalWrite(input,!a[i]);

  digitalWrite(level, 1);

   digitalWrite(level, 0);

}

 digitalWrite(output, 1);

 digitalWrite(output, 0);

}

**Посилання на схему:**

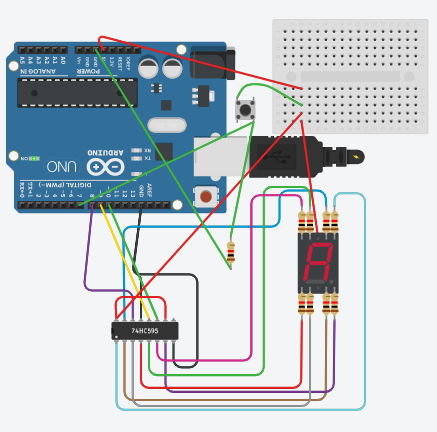
**<https://www.tinkercad.com/things/eSDSOOZy8hn>**

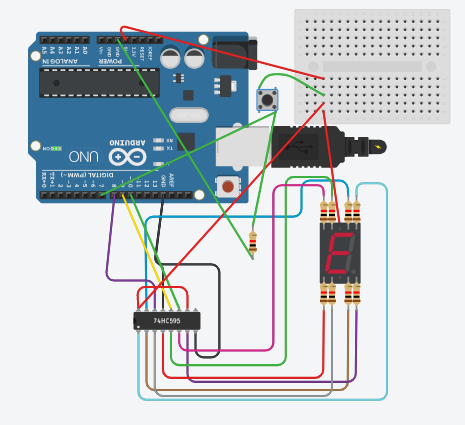
**Завдання 4.** Розробити програму дешифрування літерів A, B, C, D, E, F азбуки Морзе та відображення їх на семисегментному індикаторі. Телеграфний ключ реалізувати за допомогою кнопки. Тривалість: однієї точки – 100 мс; тире – 300 мс; паузи між елементами азбуки Морзе – 100 мс; паузи між літерами – більше 300 мс. Допустима похибка затримок – 10 % (відображати символ “П” на індикаторі, якщо похибка затримки більша). Відображати символ “0”, якщо символ не розпізнано.

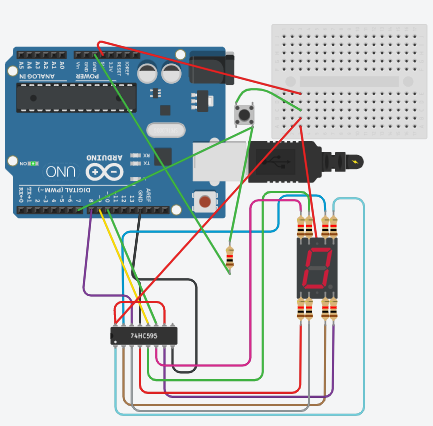
Реалізуємо роботи програми дешифрування літер згідно азбуки Морезе. В нашому випадку, реалізуємо програмний код для дешифрування зазначених символів. Для цього виконаємо модифікацію проекту з семисегментним індикатором додавши до нього кнопку. Результат зображений на рисунку нижче.

Посилання на схему:

<https://www.tinkercad.com/things/1BVeCztGtTE>







Лістинг коду:

// Значення символів Морзе для літер A, B, C, D, E, F

const char\* morseCode[] = {".-", "-...", "-.-.", "-..", ".", "..-."};

// Змінні для дешифрування символу Морзе

String currentSymbol = "";

int decodedCharacter = 0;

int a[8] = {0,1,1,1,0,1,1,1};

int b[8] = {0,1,1,1,1,1,0,0};

int c[8] = {0,0,1,1,1,0,0,1};

int d[8] = {0,1,0,1,1,1,1,0};

int e[8] = {0,1,1,1,1,0,0,1};

int f[8] = {0,1,1,1,0,0,0,1};

int zero[8] = {0,0,1,1,1,1,1,1};

// C++ code

int level = 10;

int output = 9;

int input = 8;

int inputButton = 7;

const int dotDuration = 100;  // 100 мс

const int dashDuration = 300;  // 300 мс

const int elementPause = 100;  // 100 мс

const int letterPause = 300;  // 300 мс

const int delayTolerance = 10;  // 10% похибка затримки

int buttonState = LOW;

int prevButtonState = LOW;

void setup()

{

  pinMode(level,OUTPUT);

  pinMode(output,OUTPUT);

  pinMode(input,OUTPUT);

  pinMode(inputButton,INPUT);

  Serial.begin(9600);

}

void loop()

{

  buttonState = digitalRead(inputButton);

 if (buttonState == HIGH && prevButtonState == LOW) {

    decodeMorse();

  }

 int inputSignal  = digitalRead(inputButton);

}

void decodeMorse() {

  while (buttonState == HIGH) {

    // Отримання тривалості сигналу

    int signalDuration = getSignalDuration();

    // Якщо тривалість сигналу менша за дотирання, вважаємо його крапкою

    if (signalDuration <= dotDuration + dotDuration \* delayTolerance / 100) {

      currentSymbol += ".";

    }

    // Якщо тривалість сигналу більша за дотирання, вважаємо його тире

    else if (signalDuration >= dashDuration + dashDuration \* delayTolerance / 100) {

      currentSymbol += "-";

    }

    // Якщо тривалість сигналу в межах паузи між елементами Морзе,

    // дешифруємо поточний символ і очищуємо його значення

    else if (signalDuration >= elementPause - elementPause \* delayTolerance / 100) {

      decodeSymbol();

    }

    // Затримка перед наступним вимірюванням тривалості сигналу

    delay(10);

    // Оновлення стану кнопки

    buttonState = digitalRead(inputButton);

  }

   Serial.println(currentSymbol);

  decodeSymbol();

  // Відображення розшифрованого символу на семисегментному індикаторі

  printLetter();

  delay(letterPause);

  // Очищення значень для наступного дешифрування

  currentSymbol = "";

  decodedCharacter = 0;

  }

void printLetter(){

  if(decodedCharacter==-1){

  set(zero);

  }

  if(decodedCharacter==0){

   set(a);

  }

   if(decodedCharacter==1){

   set(b);

  }

   if(decodedCharacter==2){

   set(c);

  }

   if(decodedCharacter==3){

   set(d);

  }

  if(decodedCharacter==4){

   set(e);

  }

   if(decodedCharacter==5){

   set(f);

  }

}

void decodeSymbol() {

  if (currentSymbol == "") {

    return;

  }

  // Дешифрування символу Морзе

  for (int i = 0; i < sizeof(morseCode) / sizeof(morseCode[0]); i++) {

    if (currentSymbol==morseCode[i]) {

      decodedCharacter = i;

    break;

    }

    else{

    decodedCharacter = -1;

    }

  }

}

int getSignalDuration() {

  int duration = 0;

  // Вимірювання тривалості сигналу, поки кнопка утримується

  while (buttonState == HIGH) {

    delay(1);

    duration++;

    buttonState = digitalRead(inputButton);

  }

  return duration;

}

void set(int a[]){

  for(int i=0;i<8;i++){

  digitalWrite(input,!a[i]);

  digitalWrite(level, 1);

   digitalWrite(level, 0);

}

 digitalWrite(output, 1);

 digitalWrite(output, 0);

}

**Висновки:** В ході виконання лабораторної роботи було проведено роботу з восьмирозрядним регістром зсуву 74HC595 та семисегментним індикатором, на базі яких було проведено ряд досліджень та тестувань. В результаті виконання роботи було проведено закріпленно знання та навички основних принципів керування семисигментним індикатором, а також покращено знання про роботу з регістрами зсуву. Також, в ході виконання роботи було проведено процес побудови різних схем керування, на основі яких було продемонстровано основні принципи керування, а також реалізовано декілька програм на базі мікроконтролера Arduino для керування керування роботою семисигментного індикатора на базі регістра зсуву.