

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Отчет по лабораторной работе №15

по дисциплине

«Информатика»

«Светодиодные сборки»

Вариант 15

Выполнил студент гр. ИВТб-1301-05-00

_____ /Макаров С.А./

Руководитель преподаватель

_____ /Шмакова Н.А./

Киров 2025

Цель работы

Цель работы: Ознакомление с элементами Arduino IDE, изучение основ работы со средой для программирования, а также сборка схем с семисегментными индикаторами и регистрами.

Задание

1. Секундомер. Необходимо использовать семисегментный индикатор с общим катодом для отображения последовательности тетраэдальных чисел (A000292).
2. Секундомер на 4511. Необходимо использовать семисегментный индикатор с общим анодом и дешифратор 4511 для отображения последовательности чисел Прота (A080075).
3. Сборка на базе счетчика нажатий 595. Необходимо использовать семисегментный индикатор и регистр 595 для отображения последовательности двоичных чисел с весом Хемминга 2 (A018900) с помощью счетчика нажатий.
4. Секундомер с драйвером 4026. Необходимо использовать семисегментный индикатор и драйвер 4026 для отображения последовательности чисел Хиггса (A009003) с помощью счетчика нажатий.
5. Счетчик нажатий на 3ЛС338А. Необходимо использовать семисегментный индикатор 3ЛС338А для отображения последовательности счастливых чисел (A000959) с помощью счетчика нажатий.

Решение

Задание 1

Выполнена живая сборка. Исходный код программы:

```
int numberSegments[10]={
    0b00111111, 0b00001010, 0b01011101, 0b01011110, 0b01101010,
    0b01110110, 0b01110111, 0b00011010, 0b01111111, 0b01111110,
};
int n = 0;
void setup() {
    for (int i = 1; i < 22; i++)
        pinMode(i, OUTPUT);
}
void loop() {
    int number = n * (n + 1) * (n + 2) / 6;
    if (number > 999) {
        n = 0;
        number = 0;
    }
    int maskOne = numberSegments[number % 10];
    int maskTwo = numberSegments[number / 10 % 10];
    int maskThree = numberSegments[number / 100];
    for (int i = 0; i < 7; i++) {
        digitalWrite(i + 1, bitRead(maskOne, i));
        digitalWrite(i + 8, bitRead(maskTwo, i));
        digitalWrite(i + 15, bitRead(maskThree, i));
    }
    n++;
    delay(1000);
}
```

Решение

Задание 2

Выполнена живая сборка. Исходный код программы:

```
int number = 3;
void setup() {
    for (int i = 3; i < 15; i++)
        pinMode(i, OUTPUT);
}
bool isProtNumber(int number) {
    int m = number - 1;
    int n = 0;
    while ((m & 1) == 0) {
        m >>= 1;
        n++;
    }
    return (m < (1 << n));
}
```

```

void loop() {
    if (number > 999) number = 3;
    if (isProtNumber(number)) {
        for (int i = 0; i < 4; i++) {
            digitalWrite(i + 3, bitRead(number % 10, i));
            digitalWrite(i + 7, bitRead(number / 10 % 10, i));
            digitalWrite(i + 11, bitRead(number / 100, i));
        }
        delay(1000);
    }
    number++;
}

```

Решение

Задание 3

Выполнена живая сборка. Исходный код программы:

```

#define DATA_PIN 5
#define LATCH_PIN 6
#define CLOCK_PIN 7
#define BUTTON_PIN 4
int number = 3;
bool buttonWasUp = true;
int segments[10] = {
    0b01111101, 0b00100100, 0b01111010, 0b01110110, 0b00100111,
    0b01010111, 0b01011111, 0b01100100, 0b01111111, 0b01110111
};
void setup() {
    pinMode(DATA_PIN, OUTPUT);
    pinMode(CLOCK_PIN, OUTPUT);
    pinMode(LATCH_PIN, OUTPUT);
    pinMode(BUTTON_PIN, INPUT_PULLUP);
}
bool isHemming(int n) {
    for (int i = 1; i < n; i *= 2) {
        for (int j = 1; j < n; j *= 2) {
            if (i != j && (i + j) == n) {
                return true;
            }
        }
    }
    return false;
}
void loop() {
    if (buttonWasUp && !digitalRead(BUTTON_PIN)) {
        delay(10);
        if (!digitalRead(BUTTON_PIN)) {
            number = (number + 1) % 1000;
            while (!isHemming(number)) {
                number = (number + 1) % 1000;
            }
        }
    }
}

```

```

    }
    buttonWasUp = digitalRead(BUTTON_PIN);
    digitalWrite(LATCH_PIN, LOW);
    shiftOut(DATA_PIN, CLOCK_PIN, LSBFIRST, segments[number / 100]);
    shiftOut(DATA_PIN, CLOCK_PIN, LSBFIRST, segments[number / 10 % 10]);
    shiftOut(DATA_PIN, CLOCK_PIN, LSBFIRST, segments[number % 10]);
    digitalWrite(LATCH_PIN, HIGH);
}

```

Решение

Задание 4

Выполнена живая сборка. Исходный код программы:

```

#define CLOCK_PIN 8
#define RESET_PIN 9
#define BUTTON_PIN 4
#define LIMIT 141
bool buttonWasUp = true;
int numbers[100] = { 0 };
int k = 0;
int sequenceLength = 0;
void generateHiggsSequence() {
    bool is_prime[LIMIT + 1];
    for (int i = 2; i <= LIMIT; i++) {
        is_prime[i] = true;
    }
    for (int p = 2; p * p <= LIMIT; p++) {
        if (is_prime[p]) {
            for (int i = p * p; i <= LIMIT; i += p) {
                is_prime[i] = false;
            }
        }
    }
    int index = 0;
    for (int n = 5; n <= LIMIT; n++) {
        for (int p = 2; p <= n; p++) {
            if (is_prime[p] && n % p == 0 && p % 4 == 1) {
                numbers[index++] = n;
                break;
            }
        }
    }
    sequenceLength = index;
}
void setup() {
    pinMode(CLOCK_PIN, OUTPUT);
    pinMode(RESET_PIN, OUTPUT);
    pinMode(BUTTON_PIN, INPUT_PULLUP);
    digitalWrite(RESET_PIN, HIGH);
    digitalWrite(RESET_PIN, LOW);
    generateHiggsSequence();
}

```

```

}
void loop() {
    if (buttonWasUp && !digitalRead(BUTTON_PIN)) {
        delay(50);
        if (!digitalRead(BUTTON_PIN)) {
            if (k < sequenceLength) {
                updateDisplay(numbers[k]);
                k++;
            }
            if (k >= sequenceLength) {
                k = 0;
                digitalWrite(RESET_PIN, HIGH);
                digitalWrite(RESET_PIN, LOW);
            }
        }
    }
    buttonWasUp = digitalRead(BUTTON_PIN);
}
void updateDisplay(int num) {
    digitalWrite(RESET_PIN, HIGH);
    digitalWrite(RESET_PIN, LOW);
    for (int i = 0; i < num; i++) {
        digitalWrite(CLOCK_PIN, HIGH);
        digitalWrite(CLOCK_PIN, LOW);
    }
}
}

```

Решение

Задание 5

Выполнена живая сборка. Исходный код программы:

```

#define DATA_PIN 5
#define LATCH_PIN 6
#define CLOCK_PIN 7
#define BUTTON_PIN 4
int number = 1;
bool buttonWasUp = true;
int segments[10] = {
    0b01111101, 0b00100100, 0b01111010, 0b01110110, 0b00100111,
    0b01010111, 0b01011111, 0b01100100, 0b01111111, 0b01110111
};
void setup() {
    pinMode(DATA_PIN, OUTPUT);
    pinMode(CLOCK_PIN, OUTPUT);
    pinMode(LATCH_PIN, OUTPUT);
    pinMode(BUTTON_PIN, INPUT_PULLUP);
}
bool is_lucky(int number) {
    int numbers[number * 2 + 2];
    int count = 0;
    int index = 1;

```

```

for (int i = 1; i <= number * 2 + 2; i += 2) {
    numbers[count++] = i;
}
while (index < count) {
    int step = numbers[index];
    int new_count = 0;
    for (int i = 0; i < count; i++) {
        if ((i + 1) % step != 0) {
            numbers[new_count++] = numbers[i];
        }
    }
    count = new_count;
    index++;
}
for (int i = 0; i < count; i++) {
    if (numbers[i] == number) {
        return true;
    }
}
return false;
}
void loop() {
    if (buttonWasUp && !digitalRead(BUTTON_PIN)) {
        delay(10);
        if (!digitalRead(BUTTON_PIN)) {
            number = (number + 1) % 1000;
            while (!is_lucky(number)) {
                number = (number + 1) % 1000;
            }
        }
    }
    buttonWasUp = digitalRead(BUTTON_PIN);
    digitalWrite(LATCH_PIN, LOW);
    shiftOut(DATA_PIN, CLOCK_PIN, LSBFIRST, ~segments[number % 10]);
    shiftOut(DATA_PIN, CLOCK_PIN, LSBFIRST, ~segments[number / 10 % 10]);
    shiftOut(DATA_PIN, CLOCK_PIN, LSBFIRST, ~segments[number / 100]);
    digitalWrite(LATCH_PIN, HIGH);
}

```

Вывод

В ходе лабораторной работы были изучены основы работы в среде Arduino IDE, а также собраны схемы с семисегментными индикаторами (с общим анодом и катодом), регистром 595, дешифратором 4511, драйвером 4026 в соответствии с вариантом задания.