

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Иркутский национальный исследовательский технический
университет»
Институт информационных технологий и анализа данных

ОТЧЁТ

о прохождении _____ учебной практики
(вид практики: учебная/производственная)

технологической (проектно-технологической) практики
(тип практики: технологическая/научно-исследовательская работа/преддипломная и др.)


В _____ ИРНИТУ
(наименование профильной организации)





<https://hh.ru/resume/b02b916cff0efe781f0039ed1f356739437373>



<https://www.superjob.ru/resume/programmist-s-55734466.html>

Обучающегося Цыдендамбаев Д.А., ИСИБ-24-1 
(ФИО, группа, подпись)

Руководитель практики от института ИТиАД
Кононенко Р.В., доцент института ИТиАД 
(ФИО, должность, подпись)

Руководитель образовательной программы
Кононенко Р.В., доцент института ИТиАД 
(ФИО, должность, подпись)

Оценка по практике зачтено
Кононенко Р.В., Тко 30.09.2025
(ФИО, подпись, дата)

Содержание отчета на 31 стр.
Приложение к отчету на ____ стр.

Индивидуальное задание на прохождение
учебной практики: технологической (проектно-технологической)
практики

для Цыдендамбаева Дмитрия Александровича
(ФИО обучающегося полностью)

обучающегося 1 курса группы ИСИБ-24-1
я _____

по направлению подготовки Информационные системы и технологии
профиль Интеллектуальные системы обработки информации и управления

Место прохождения практики: ИРНТУ

Сроки прохождения практики с «16» июня 2025 г. по «29» июня 2025 г.

Цели и задачи прохождения практики:

Закрепление теоретических знаний, полученных в ходе изучения дисциплин «Информатика» и «Программирование», а также развитие практических навыков в области разработки программного обеспечения.

Содержание практики, вопросы, подлежащие изучению:

Совершенствование умений программирования на языке C++; работа с микроконтроллерной платформой Arduino и изучение принципов взаимодействия с различными модулями; освоение основ программирования на языке Python; знакомство с базовыми приёмами машинного зрения

Планируемые результаты практики:

Закрепление и углубление знаний по языку программирования C++; Приобретение навыков работы с Arduino; Формирование практических умений в области разработки программного обеспечения для управления устройствами; Получение первоначального опыта применения методов машинного зрения; Развитие навыков отладки, тестирования и документирования программного кода;

Руководитель практики от
института ИТиАД

Т.В. Кононенко / Кононенко Р.В. /
(подпись)

Согласовано:

Руководитель ООП

Т.В. Кононенко / Кононенко Р.В. /
(подпись)

« 16 » 06 2025 г.

С настоящим индивидуальным заданием и с программой практики
ознакомлен(а), задание принято к исполнению

С « 16 » 06 2025 г.
(подпись)

Содержание

Введение	4
Задание №1	5
Задача №2	8
Задача №3	10
Задача №4	13
Задача №5	15
Задача №6	17
Задача №7	19
Задача №8	20
Задача №9	22
Задача №10	25
Отзыв о посещении филиала АО «СО ЕЭС» Иркутское РДУ	27
Отзыв о посещении компании ISPsystem	27
Задача №11	28
Заключение	30
Список литературы	31

Введение

Учебная практика была проведена на базе Иркутского национального исследовательского технического университета (ИРНИТУ) в период с 16 по 28 июня 2025 года. Главной целью являлось углубление и применение компетенций, сформированных в процессе освоения профильных дисциплин.

Основное внимание в ходе практики уделялось развитию практических навыков программирования, включая проектирование алгоритмов и работу с структурами данных. В рамках одиннадцати учебных заданий осваивались инструменты и методы, актуальные для современной IT-отрасли, что способствовало развитию способности самостоятельно анализировать и решать прикладные задачи.

Результатом прохождения практики стало не только закрепление теоретической базы, но и формирование профессионального подхода к реализации программных решений, что важно для дальнейшего обучения и будущей карьеры в сфере информационных технологий.

Задание №1

Незнайка в своей экспедиции на Луну оказался на вершине лунной горы. Спуск вниз опасен, поэтому он взял с собой карту склона горы, где числами обозначено, сколько минут требуется на этот участок маршрута. Спуск происходит сверху вниз на один из соседних участков. Пример наиболее короткого маршрута выделен красным цветом, сумма чисел = 10. Напишите программу, рассчитывающую минимальное время спуска (сумму чисел в пути с вершины до основания).

```
#include <iostream>
#include <vector>

void buildHeap(int n) {
    std::vector<std::vector<int>> heap;
    std::vector<std::vector<int>> original;

    int value = 1;

    for (int i = 0; i < n; ++i) {
        std::vector<int> row, copyRow;
        for (int j = 0; j < n; ++j) {
            if (j <= i) {
                row.push_back(value * 10 + i + j); // генерируем
                copyRow.push_back(value * 10 + i + j);
                value++;
            }
            else {
                row.push_back(0);
                copyRow.push_back(0);
            }
        }
        heap.push_back(row);
        original.push_back(copyRow);
    }

    // пирамида
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j <= i; ++j) {
            std::cout << original[i][j] << " ";
        }
        std::cout << std::endl;
    }
}
```

Рисунок 1 – Код задачи №1

```

// дп
for (int i = n - 2; i >= 0; --i) {
    for (int j = 0; j <= i; ++j) {
        heap[i][j] += std::min(heap[i + 1][j], heap[i + 1][j + 1]);
    }
}

std::vector<int> path;
int i = 0, j = 0;
path.push_back(original[i][j]);
while (i < n - 1) {
    if (heap[i + 1][j] < heap[i + 1][j + 1]) {
        ++i;
    }
    else {
        ++i;
        ++j;
    }
    path.push_back(original[i][j]);
}

std::cout << "\nPath: ";
for (size_t k = 0; k < path.size(); ++k) {
    std::cout << path[k];
    if (k != path.size() - 1)
        std::cout << " ";
}
std::cout << std::endl;

std::cout << "Min time: " << heap[0][0] << std::endl;
}

int main() {
    int n = 0;
    std::cout << "Enter height: ";
    std::cin >> n;
}

```

Рисунок 1.2 – Код задачи №1.2

Таблица тестов:

Ввод	4	5
Вывод	968 212 427 912 914 221 630 806 272 292 Минимальная сумма: 1888 Путь: 968 427 221 272	928 303 665 64 211 868 49 166 741 533 25 969 242 782 80 Минимальная сумма: 1369 Путь: 928 303 64 49 25

```
Высота: 10
58
468 285
898 342 29
613 136 120 222
618 209 581 179 88
656 566 23 598 321 222
280 192 478 407 845 997 561
699 70 116 811 215 443 138 188
450 296 279 956 71 231 435 801 54
918 323 202 426 50 816 244 695 804 570
Минимальная сумма: 1796
Путь: 58 285 342 136 209 23 407 215 71 50
```

Рисунок 1.3 – Тест №1 для задачи №1

```
4
83
86 77
15 93 35
86 92 49 21
261
83 77 15 86
```

Рисунок 1.4 – Тест №2 для задачи №1

Задача №2

После метеоритной атаки компьютерная сеть для управления лунными заводами разбилась на части, нужно объединить её в единое целое. Каждый фрагмент сети представлен в виде ненаправленного графа.

Вам известно общее число вершин графа (узлы сети, не более 1000) и набор рёбер (сохранившиеся линии связи, не более 1000).

Определите, какое минимальное число линий связи нужно дополнительно построить, чтобы сеть стала единой.

```
#include <iostream>
#include <vector>

void dfs(int v, const std::vector<std::vector<int>>& g, std::vector<bool>& visited) {
    visited[v] = true;
    for (int u : g[v]) {
        if (!visited[u]) {
            dfs(u, g, visited);
        }
    }
}

int main() {
    int n, m;
    std::cin >> n >> m;

    std::vector<std::vector<int>> g(n);
    for (int i = 0; i < m; ++i) {
        int a, b;
        std::cin >> a >> b;
        --a; --b;
        g[a].push_back(b);
        g[b].push_back(a);
    }

    std::vector<bool> visited(n, false);
    int components = 0;

    for (int i = 0; i < n; ++i) {
        if (!visited[i]) {
            dfs(i, g, visited);
            components++;
        }
    }

    std::cout << (components - 1) << std::endl;

    return 0;
}
```

Рисунок 2.1 – Код задачи №2

Таблица тестов:

Ввод	10 6 1 2 2 8 4 10 5 9 6 10 7 9	7 4 1 2 2 3 4 5 6 7
Вывод	3	2

```
5 4
1 2
2 3
3 4
4 5
0
```

Рисунок 2.2 – Тест №1 для задачи №2

```
10 6
1 2
2 8
4 10
5 9
6 10
7 9
3
```

Рисунок 2.3 – Тест №2 для задачи №2

Задача №3

Иркутске раз в году наступает зима. Не смотря на то что событие это довольно регулярное, оно всегда внезапно. Снег буквально заваливает все улицы, не давая проехать на чём-то меньше трактора. В этом году терпение лопнуло и специальным указом был создан кризисный центр по борьбе с сугробами. Центру были переданы спутники, лазеры, метеорологические зонды и несколько десятков лопат.

Вам поручено возглавить отдел разведки снежной ситуации и быть способным чрезвычайно быстро отвечать на запросы центра. Сам город состоит из нескольких, расположенных подряд, улиц, каждая из которых абсолютно похожа на любую другую.

Информация о снеге передается вам в виде тройки чисел – 1 в качестве идентификатора события, уникального индекса улицы и количество миллиметров выпавшего снега.

Запросы в свою очередь так же имеют вид тройки чисел – 2 в качестве идентификатора события, индекс улицы с которой нужно суммировать количество выпавшего снега и индекс улицы по которую нужно суммировать, крайние улицы должны быть включены.

```

1  #include <iostream>
2  #include <vector>
3
4  int main() {
5      int n, k;
6      std::cin >> n >> k;
7
8      std::vector<long long> streets(n + 1, 0); // улицы от 1 до n
9
10     for (int i = 0; i < k; ++i) {
11         int type;
12         std::cin >> type;
13         if (type == 1) {
14             int ix;
15             long long x;
16             std::cin >> ix >> x;
17             streets[ix] += x;
18         }
19         else if (type == 2) {
20             int u, r;
21             std::cin >> u >> r;
22             long long sum = 0;
23             for (int j = u; j <= r; ++j) {
24                 sum += streets[j];
25             }
26             std::cout << sum << std::endl;
27         }
28     }
29
30     return 0;
31 }

```

Рисунок 3.1 – Код задачи № 2

Таблица тестов

Ввод	6 5 2 1 6 1 3 2 2 2 4 1 6 3 2 1 6	5 3 1 3 7 1 1 4 2 1 5
Вывод	0 2 5	11

```

Ввод
6 5
2 1 6
1 3 2
2 2 4
1 6 3
2 1 6
Вывод
0
2
5

```

Рисунок 3.2 – Тест №1 для задачи №3

```
6 5
2 1 6
1 3 2
2 2 4
1 6 3
2 1 6
0
2
5
```

Рисунок 3.3 – Тест №2 для задачи №3

Задача №4

Перестановка P длины n – это упорядоченный набор, содержащий числа от 1 до n , каждое из которых входит в него ровно один раз. Например, перестановкой длины 13 является набор (5 11 13 12 6 1 8 4 10 9 7 2 3). Само название говорит о том, для чего предназначен этот объект. Например, можно при помощи перестановки букв зашифровать слово.

```
1  #include <iostream>
2  #include <vector>
3
4  int main() {
5      int n, k;
6      std::cin >> n >> k; // ВВОД
7
8      std::vector<int> permutation(n);
9      for (int i = 0; i < n; i++) {
10         std::cin >> permutation[i]; // читаем перестановку
11     }
12
13     std::string encrypted;
14     std::cin >> encrypted;
15
16     // Обратная перестановка
17     std::vector<int> inverse(n);
18     for (int i = 0; i < n; i++) {
19         inverse[permutation[i] - 1] = i + 1;
20     }
21
22     std::string result = encrypted;
23     for (int iter = 0; iter < k; iter++) {
24         std::string temp = result;
25         for (int i = 0; i < n; i++) {
26             temp[i] = result[inverse[i] - 1]; // переставляем
27         }
28         result = temp;
29     }
30
31     std::cout << result; // результат
32
33     return 0;
34 }
```

Рисунок 4.1 – Код задачи №3

Таблица тестов

Ввод	13 2 5 11 13 12 6 1 8 4 10 9 7 2 3 poartsnoitsin	6 3 3 1 4 6 2 5 encode
Вывод	transposition	eodnce

```
13 2
5 11 13 12 6 1 8 4 10 9 7 2 3
poartsnoitsin
transposition
```

Рисунок 4.2 – Тест №1 для задачи №4

```
13 2
5 11 13 12 6 1 8 4 10 9 7 2 3
transposition
snaoptriitosn
```

Рисунок 4.3 – Тест №2 для задачи №4

Задача №5

Дана матрица, состоящая из 1 и 0. Значениями 1 в матрице нарисована некоторая фигура. Необходимо определить координаты верхнего левого и нижнего правого углов параллельного осям ограничивающего прямоугольника, т.е. такого прямоугольника, минимального размера, в который фигура помещается полностью и при этом ни одна точка исходной фигуры не попадает на стороны прямоугольника.

```
1  #include <iostream>
2  #include <vector>
3  #include <climits>
4
5  int main() {
6      int h, w;
7      std::cin >> h >> w; // размерность матрицы
8
9      std::vector<std::vector<int>> matrix(h, std::vector<int>(w));
10
11     for (int i = 0; i < h; i++) {
12         for (int j = 0; j < w; j++) {
13             std::cin >> matrix[i][j];
14         }
15     }
16
17     int min_row = INT_MAX, max_row = -1;
18     int min_col = INT_MAX, max_col = -1;
19
20     for (int i = 0; i < h; i++) {
21         for (int j = 0; j < w; j++) {
22             if (matrix[i][j] == 1) { // если часть фигуры
23                 if (i < min_row) min_row = i;
24                 if (i > max_row) max_row = i;
25                 if (j < min_col) min_col = j;
26                 if (j > max_col) max_col = j;
27             }
28         }
29     }
30
31     // выводим координаты
32     std::cout << min_row << " " << min_col << " ";
33     std::cout << max_row << " " << max_col << std::endl;
34
35     return 0;
36 }
```

Рисунок 4.1 – Код задачи №4

Таблица тестов

Ввод	10 10 0000000000 0000000000 0000000000 0000000000 0001110000 0001110000 0001110000 0001110000 0000000000 0000000000	3 3 000 010 000
Вывод	3 2 8 6	0 0 1 1

```

3 3
0 1 0
1 1 0
0 0 0
1 1 2 2

```

Рисунок 5.2 – Тест для задачи №5

```

3 3
0 0 0
0 1 0
0 0 0
0 0 2 2

```

Рисунок 5.2 – Тест для задачи №5

Задача №6

В школьном кружке робототехники есть два вида микроконтроллеров (условно тип А и тип В) и два вида модулей управления мотором (условно тип 1 и тип 2). Выяснилось, что контроллер типа В и модуль управления типа 2 несовместимы. Использование микроконтроллеров и модулей управления в других комбинациях возможно. Имеется а микроконтроллеров типа А, b микроконтроллеров типа В, x модулей управления типа 1 и y модулей типа 2. Определите, какое максимальное число работающих пар из микроконтроллера модуля управления мотором можно составить. Ваша программа должна ответить на n запросов.

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  int main() {
6      int n = 0;
7      std::cout << "Введите количество запросов: ";
8      std::cin >> n;
9
10     std::vector<std::vector<int>> requests(n, std::vector<int>(4)); // массив для хранения всех запросов
11
12     std::cout << "Введите запросы (a, b, x, y): ";
13     for (int i = 0; i < n; i++) {
14         for (int j = 0; j < 4; j++) {
15             std::cin >> requests[i][j]; // считываем a, b, x, y для каждого запроса
16         }
17     }
18     std::cout << "Результат: ";
19     for (int i = 0; i < n; i++) {
20         std::vector<int> req = requests[i]; // текущий запрос
21         int a = req[0];
22         int b = req[1];
23         int x = req[2];
24         int y = req[3];
25         int used_b = 0;
26
27         if (b >= x) {
28             used_b = x;
29             x = 0;
30         } else {
31             used_b = b;
32             x -= b;
33         }
34
35         // затем подключаем a к оставшимся модулям типа 1 и ко всем модулям типа 2
36         int used_a = std::min(a, x + y);
37
38         std::cout << used_b + used_a << ' ';
39     }
40     return 0;
41 }

```

Рисунок №6.1 – Код задачи №6

Таблица тестов

Ввод	3 5 2 3 7 10 2 12 2 3 8 5 6	2 1 3 4 5 1 0 0 0
Вывод	7 12 8	4 0

```
3
5 2 3 7
10 2 12 2
3 8 5 6
7 12 8
```

Рисунок №6.2 – Тест №1 для задачи №6

```
2
1 3 4 5
1 0 0 0
4 0
```

Рисунок №6.3 – Тест №2 для задачи №6

Задача №7

На компьютере работника автосервиса нашли файл с последовательностью автомобильных номеров, обслуживавшихся в этом автосервисе. Так как файл был поврежден, некоторые данные отображаются неверно. Нужно определить, какие из них остались невредимыми.

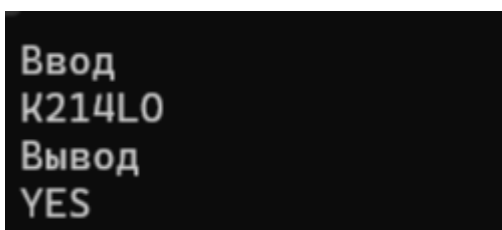
Автомобильным номером является строка из шести символов. Первый символ – заглавная латинская буква, далее следует 3 цифры, и после – две заглавные латинские буквы. Например, строка "P142EQ" является номером. Вам будет дана строка, состоящая из шести символов, необходимо ответить, является ли строка автомобильным номером.

```
1  #include <iostream>
2  #include <cctype>
3
4  int main() {
5      std::string plate;
6      std::cout << "введите строку из 6 символов: ";
7      std::cin >> plate;
8
9      bool format_ok = true; // предполагаем, что всё верно
10
11     // проверяем первую букву
12     if (!std::isalpha(plate[0])) format_ok = false;
13     // проверяем три цифры
14     if (!std::isdigit(plate[1]) || !std::isdigit(plate[2]) || !std::isdigit(plate[3])) format_ok = false;
15     // проверяем две последние буквы
16     if (!std::isalpha(plate[4]) || !std::isalpha(plate[5])) format_ok = false;
17
18     std::cout << "результат: " << (format_ok ? "Yes" : "No");
19     return 0;
20 }
```

Рисунок 7.1 – Код задачи №7

Таблица тестов

Ввод	K045OL	LL01KJ
Вывод	YES	NO



```
Ввод
K214LO
Вывод
YES
```

Рисунок №7.2 – Тест для задачи №7

Задача №8

Составить светодиодную матрицу размером не менее 8 на 8 светодиодов. На матрицу вывести инфографику с различными динамично меняющимися изображениями.

```
1  #include <FastLED.h> // библиотека
2
3  #define LED_COUNT 64
4  #define LED_PIN 3
5
6  CRGB leds[LED_COUNT];
7
8  void setup() {
9      FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, LED_COUNT);
10     FastLED.setBrightness(100); // яркость
11 }
12
13 void loop() {
14     rainbowCycle(10); // запускаем эффект радуги с задержкой 10
15 }
16
17 void rainbowCycle(uint8_t delayTime) {
18     static uint8_t colorHue = 0; // меняем цвет с помощью переменной
19
20     for (int i = 0; i < LED_COUNT; i++) {
21         leds[i] = CHSV(colorHue + (i * 256 / LED_COUNT), 255, 255);
22     }
23
24     FastLED.show(); // обновляем цвета
25     delay(delayTime);
26     colorHue++; // меняем переменную цвета
27 }
```

Рисунок №8.1 - Код задачи №8

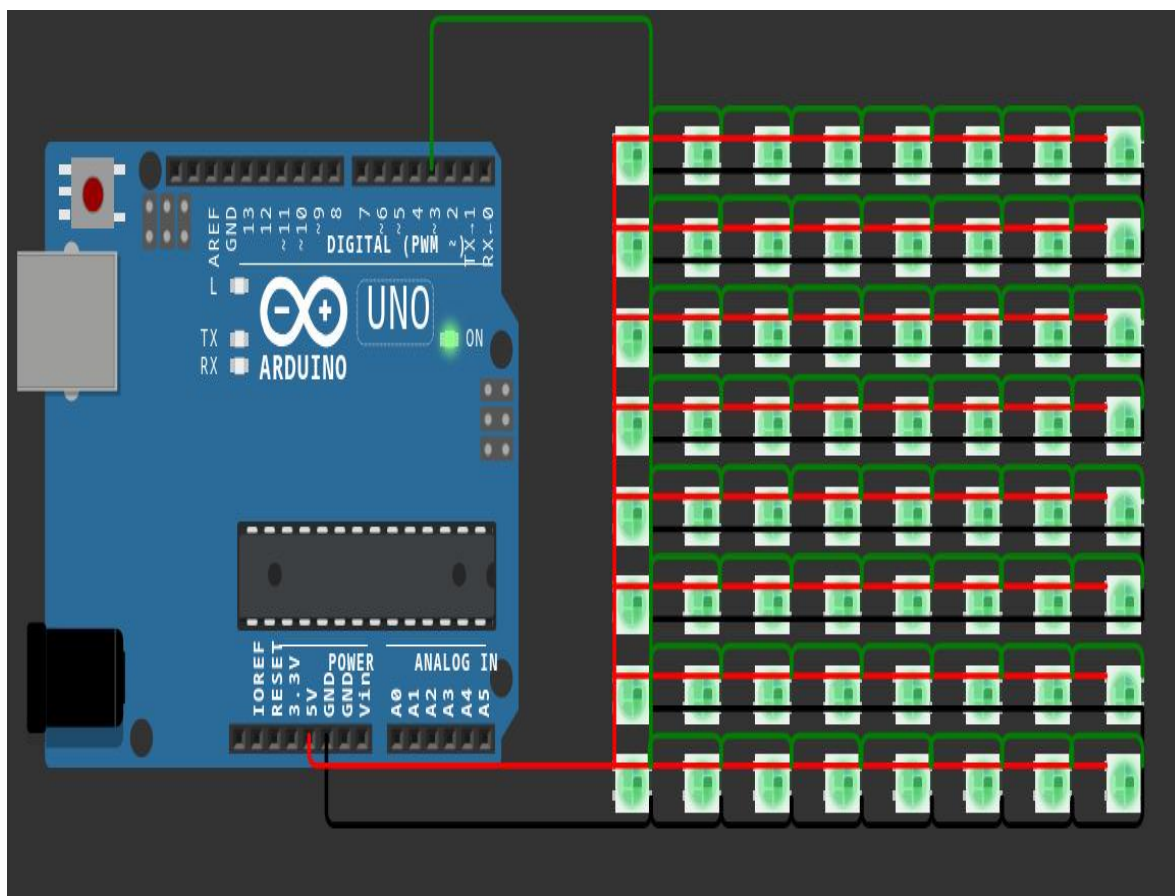


Рисунок №8.1 – Пример работы прошивки №1 для 8 задачи

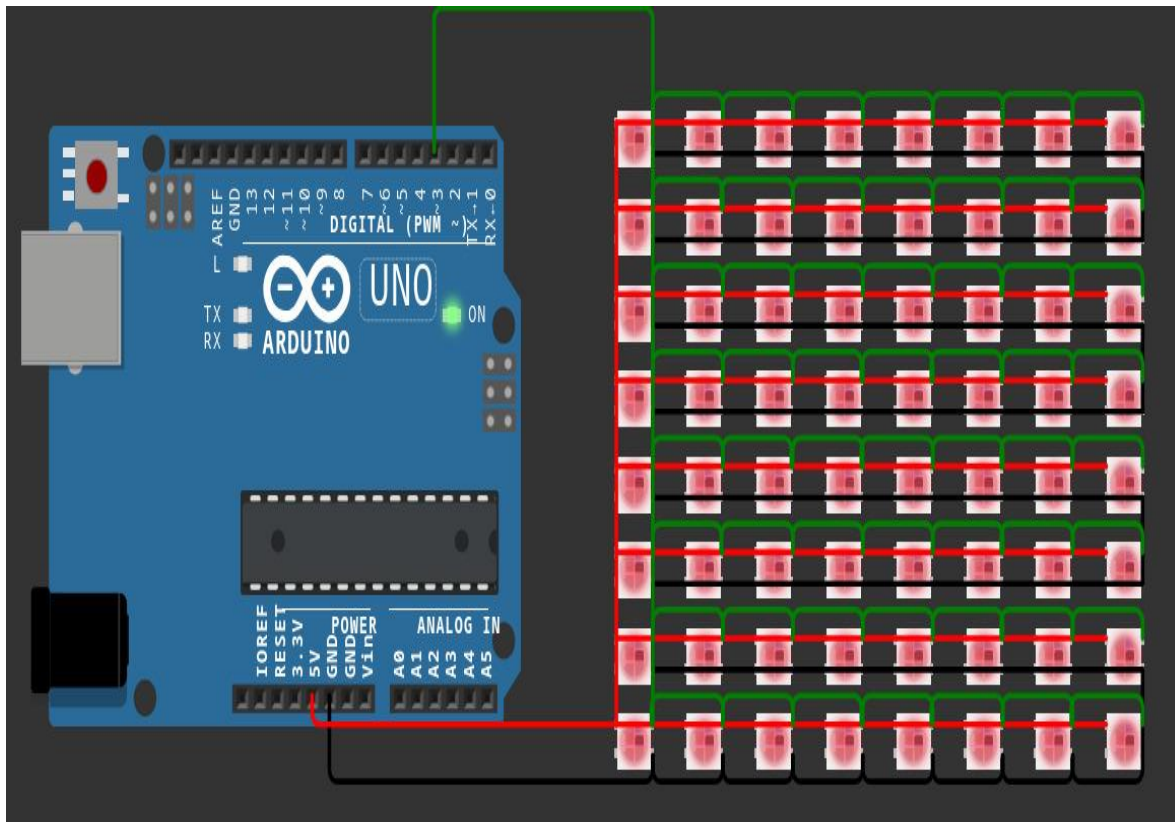


Рисунок №8.2 – Пример работы прошивки №2 для 8 задачи

Задача №9

Собрать схему имитирующую работу автоматических дверей

Подобрать номинал резисторов для светодиодов

Написать программу для управления процессом работы автоматических дверей.

Зеленый светодиод – двери открываются.

Красный светодиод – двери закрываются.

Фоторезистор имитируют процесс приближения-удаления человека от дверей.

Программа управляет светодиодами, имитируя работу автоматических дверей:

При низком уровне освещённости (когда фоторезистор обнаруживает человека рядом), загорается зелёный светодиод (сигнализируя об открытии дверей).

При высоком уровне освещённости (когда человека нет), активируется красный светодиод (указывая на закрытие дверей).


```

1  #define RED_LED 10    // красный светодиод
2  #define GREEN_LED 5   // зеленый светодиод
3  #define LDR_PIN A0    // фоторезистор на аналоговый pin
4
5  #define THRESHOLD 512
6  #define OPEN_TIME 3000 // время открытия
7
8  bool doorIsOpen = false;
9  unsigned long doorTimer = 0;
10
11 void setup() {
12     pinMode(RED_LED, OUTPUT);
13     pinMode(GREEN_LED, OUTPUT);
14     Serial.begin(9600);
15     digitalWrite(RED_LED, HIGH);
16     digitalWrite(GREEN_LED, LOW);
17     Serial.println("Автоматические двери запущены");
18 }
19
20 void loop() {
21     int sensorValue = analogRead(LDR_PIN); // значение с фоторезистора
22
23     // если значение превышает порог и дверь закрыта - открываем
24     if (sensorValue > THRESHOLD && !doorIsOpen) {
25         doorIsOpen = true;
26         doorTimer = millis();
27         digitalWrite(RED_LED, LOW);
28         digitalWrite(GREEN_LED, HIGH);
29         Serial.println("Двери открываются!");
30     }
31
32     // если дверь открыта и прошло время OPEN_TIME
33     if (doorIsOpen && (millis() - doorTimer >= OPEN_TIME)) {
34         sensorValue = analogRead(LDR_PIN);
35
36         if (sensorValue <= THRESHOLD) {
37             doorIsOpen = false;
38             digitalWrite(GREEN_LED, LOW);
39             digitalWrite(RED_LED, HIGH);
40             Serial.println("Двери закрываются");
41         } else {
42             // продлеваем время открытия
43             doorTimer = millis();
44             Serial.println("Двери остаются открытыми");
45         }
46     }
47
48     delay(50); //задержка для стабильности
49 }

```

Код задачи №9

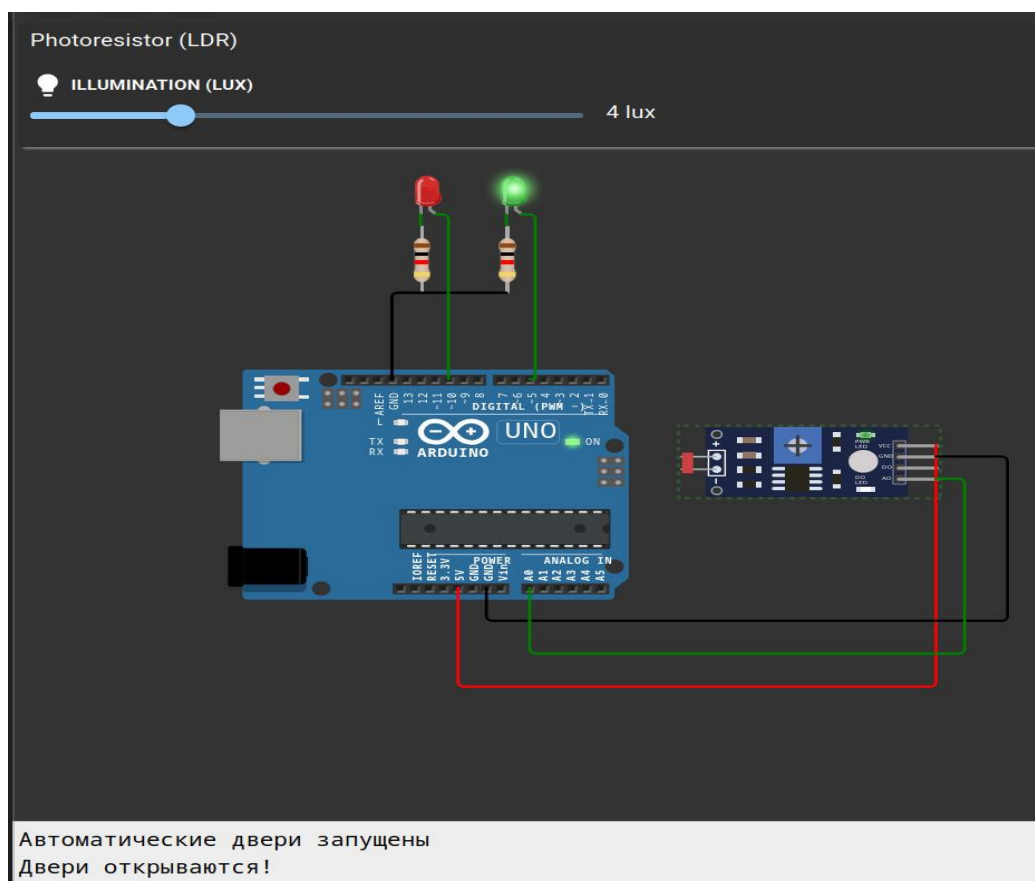


Рисунок №9.1 – Пример работы прошивки №1 для 9 задачи

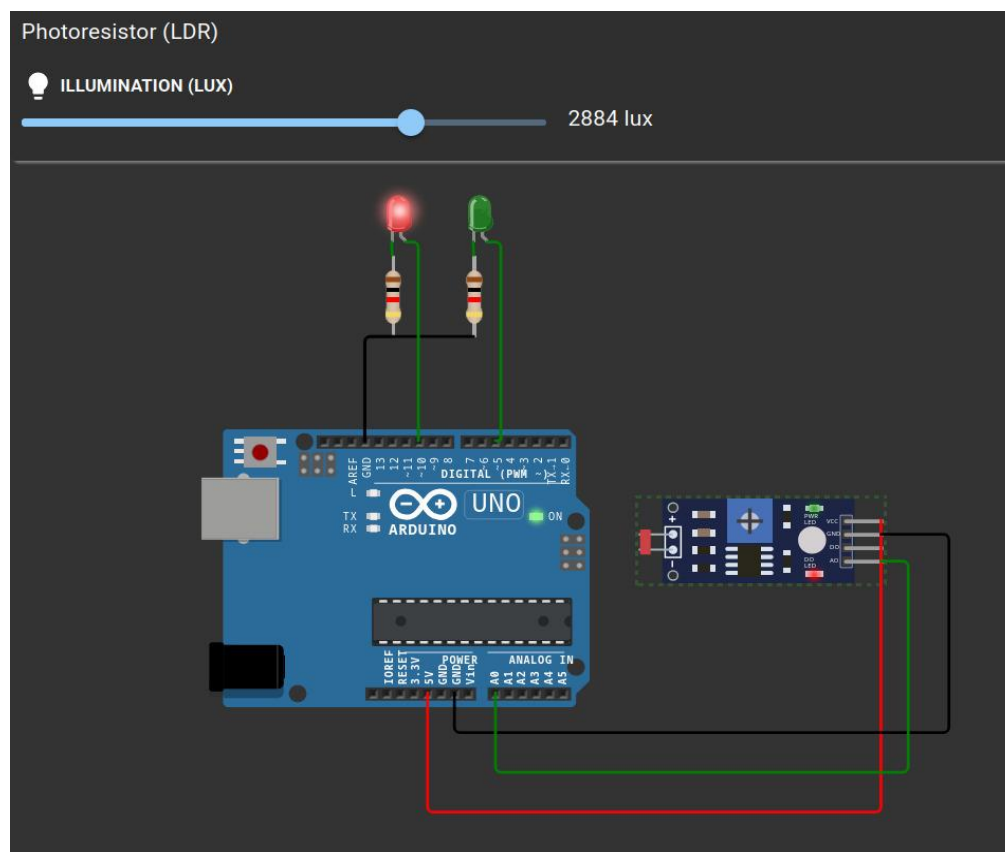


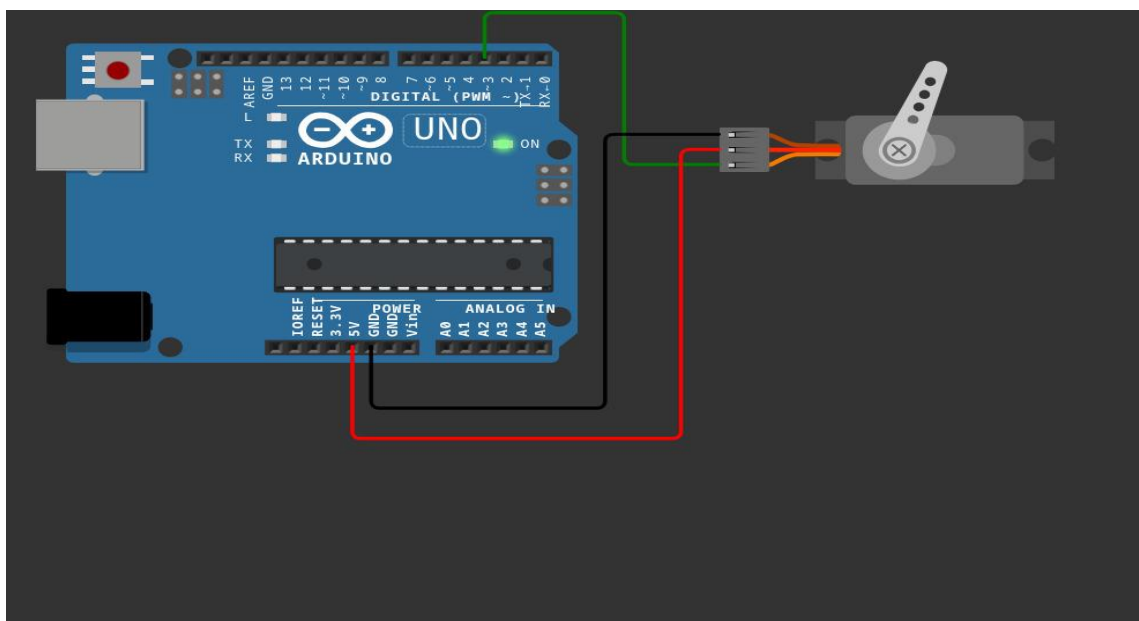
Рисунок №9.2 – Пример работы прошивки №2 для 9 задач

Задача №10

Собрать схему подключения сервопривода
Написать программу для управления сервоприводом через последовательный порт

Логика работы программы
запрашивается угол поворота сервопривода, если он отличен от того, на который повернут привод, то плавно повернуть до указанного. Программа работает в цикле, с возможностью постоянно изменять угол поворота.

```
1  #include <Servo.h>
2  Servo servo;
3  int currentPos = 0;
4
5  void setup() {
6      Serial.begin(9600);
7      servo.attach(3);
8      servo.write(currentPos);
9      Serial.println("Введите угол от 0 до 180:");
10 }
11
12 void loop() {
13     if (Serial.available() > 0) {
14         String input = Serial.readStringUntil('\n');
15         input.trim();
16
17         bool validInput = true;
18         for (unsigned int i = 0; i < input.length(); i++) {
19             if (!isDigit(input[i])) {
20                 validInput = false;
21                 break;
22             }
23         }
24         if (validInput) {
25             int targetAngle = input.toInt();
26             if (targetAngle >= 0 && targetAngle <= 180) {
27                 if (targetAngle != currentPos) {
28                     Serial.print("Поворот на: ");
29                     Serial.println(targetAngle);
30                     if (targetAngle > currentPos) {
31                         for (int pos = currentPos; pos <= targetAngle; pos++) {
32                             servo.write(pos);
33                             delay(10);
34                         }
35                     } else {
36                         for (int pos = currentPos; pos >= targetAngle; pos--) {
37                             servo.write(pos);
38                             delay(10);
39                         }
40                     }
41                     currentPos = targetAngle;
42                 }
43             } else {
44                 Serial.println("Ошибка: допустимый диапазон 0-180");
45             }
46         } else {
47             Serial.println("Ошибка: вводите только цифры");
48         }
49         Serial.println("Введите угол от 0 до 180:");
50     }
51 }
```

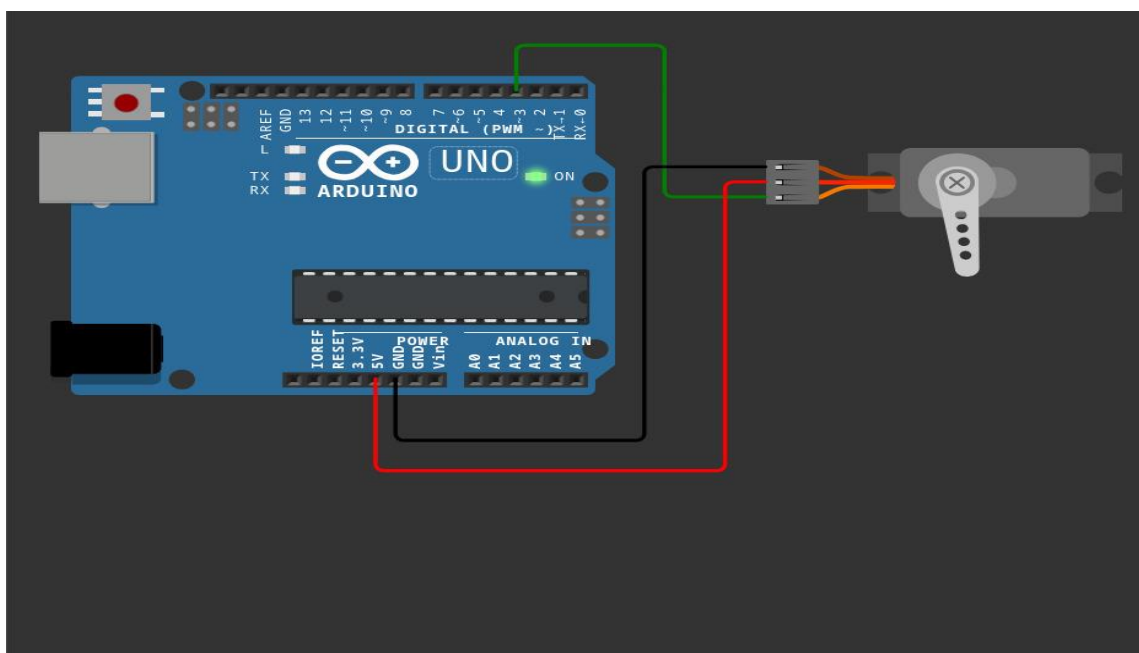


Введите угол от 0 до 180:

Поворот на: 20

Введите угол от 0 до 180:

Рисунок №10.1 – Пример работы прошивки №1



Введите угол от 0 до 180:

Поворот на: 175

Введите угол от 0 до 180:

Рисунок №10.2 – Пример работы прошивки №2 для 10 задачи

Отзыв о посещении филиала АО «СО ЕЭС» Иркутское РДУ

В рамках учебной практики наша группа посетила Иркутское региональное диспетчерское управление — филиал АО «Системный оператор ЕЭС», обеспечивающий стабильную работу энергосистемы области. Основная задача РДУ — оперативное управление: координация электростанций и сетевых объектов, регулирование выработки и распределения электроэнергии. Во время визита нам рассказали о значении цифровых технологий в энергетике и о работе IT-отдела, который разрабатывает системы для поддержки диспетчеров. Мы узнали о применении аналитических комплексов и симуляционных платформ для моделирования работы энергообъектов. Эта экскурсия помогла мне понять, насколько важны процессы диспетчеризации и роль информационных технологий в надежности энергосистемы региона.

Отзыв о посещении компании ISPsystem

В рамках учебной практики наша группа посетила компанию ISPsystem, занимающуюся разработкой программного обеспечения для управления IT-инфраструктурой. Важно отметить, что компания создаёт собственные продукты, а не работает по заказу или на сторонних платформах, что позволяет ей полностью контролировать качество и развитие своих решений.

Особое впечатление произвёл коллектив: сотрудники открытые, заинтересованные в своей работе, охотно делились опытом и рассказывали о своей деятельности. Это создавало ощущение дружелюбной и профессиональной атмосферы.

Также нам рассказали о том, какие специалисты востребованы в компании. Помимо программистов, там работают тестировщики, дизайнеры, менеджеры проектов, маркетологи, специалисты по документации и локализации. Таким образом, попасть в компанию можно, имея разные профессиональные компетенции.

Посещение ISPsystem позволило получить более чёткое представление о том, как организована работа современной IT-компании, и дало дополнительную мотивацию развиваться в этой сфере.

Задача №11

Найдите все жёлтые объекты на изображении. Найдите центры жёлтых объектов. Отметьте центр красной точкой.

```
1  import cv2
2  import numpy as np
3
4  image = cv2.imread("test2.jpg")
5  result_image = image.copy()
6
7  # Предобработка изображения
8  blurred = cv2.GaussianBlur(image, (11, 11), 0)
9  hsv = cv2.cvtColor(blurred, cv2.COLOR_BGR2HSV)
10 |
11 # Диапазон желтого цвета в HSV
12 lower_yellow = np.array([20, 100, 100])
13 upper_yellow = np.array([36, 255, 255])
14
15 # Создание маски для желтого цвета
16 yellow_mask = cv2.inRange(hsv, lower_yellow, upper_yellow)
17
18 # Поиск контуров желтых объектов
19 contours, _ = cv2.findContours(yellow_mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
20
21 print(f"Найдено желтых объектов: {len(contours)}")
22
23 # Обработка каждого контура и отметка центров
24 for contour in contours:
25     if cv2.contourArea(contour) < 100:
26         continue
27
28     # Вычисление центра через ограничивающий прямоугольник
29     x, y, w, h = cv2.boundingRect(contour)
30     center_x = int(x + w/2)
31     center_y = int(y + h/2)
32
33     # Красная точка в центре объекта
34     cv2.circle(result_image, (center_x, center_y), 10, (0, 0, 255), -1)
35
36 cv2.imshow("Yellow Objects", result_image)
37 cv2.waitKey(0)
38 cv2.destroyAllWindows()
```

Рисунок №11.1 – Код задачи №11

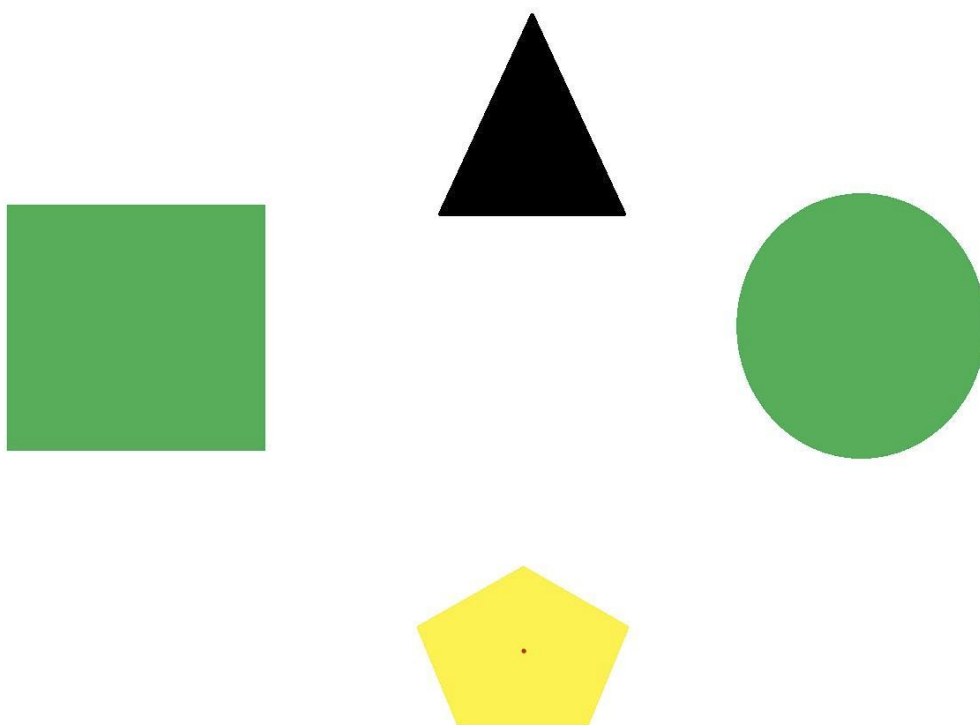


Рисунок №11.2 – Пример работы

Заключение

В период практики мне удалось применить и значительно углубить теоретические знания, полученные за время изучения курсов “С++” и “Алгоритмы и структуры данных”. Основной практической задачей стало развитие навыков программирования на языках программирования (я пользовался С++ и Python) и освоение работы с аппаратной частью на платформе Arduino. Работа с различными модулями и датчиками позволила наглядно изучить взаимодействие между программным кодом и физическими устройствами. Отдельным и крайне интересным направлением работы стало знакомство с основами машинного зрения. В рамках этого блока я разрабатывал программы на Python, используя библиотеки OpenCV и NumPy. Это дало возможность освоить на практике базовые алгоритмы обработки изображений и методы распознавания объектов по цветовым признакам.

Важным дополнением к технической части стали экскурсии в компании ISPsystem и Иркутское РДУ. Они предоставили ценный взгляд изнутри на работу современных IT- и энергетических предприятий, их технологический стек и подходы к управлению проектами.

Весь приобретенный опыт не только систематизировал пройденный материал, но и сформировал четкое понимание, как полученные в университете знания применяются в реальных производственных задачах. Уверен, что это будет фундаментом для дальнейшего обучения и поможет в будущей профессиональной деятельности.

Список литературы

1. Wokwi Arduino Simulator: онлайн-симулятор Arduino-проектов [Электронный ресурс]. – URL: <https://wokwi.com/arduino>
2. AlexGyver: подробное руководство по работе со светодиодными лентами WS2812B [Электронный ресурс]. – URL: https://alexgyver.ru/ws2812_guide
3. AlexGyver: руководство по работе со светодиодными матрицами [Электронный ресурс]. – URL: https://alexgyver.ru/matrix_guide
4. AlexGyver: работа с аналоговыми входами Arduino [Электронный ресурс]. – URL: <https://alexgyver.ru/lessons/analog-pins>
5. AlexGyver: работа с последовательным портом Serial [Электронный ресурс]. – URL: <https://alexgyver.ru/lessons/serial>
6. AlexGyver: подключение и управление сервоприводами [Электронный ресурс]. – URL: <https://alexgyver.ru/lessons/servo>
7. Стуков И. Компьютерное зрение OpenCV: где применяется и как работает в Python [Электронный ресурс] / И. Стуков. — Skillbox, 2023. — URL: <https://skillbox.ru/media/code/kompyuternoe-zrenie-opencv-gde-primenyaetsya-i-kak-rabotaet-v-python/>