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

**О Т Ч Ё Т**

|  |  |  |
| --- | --- | --- |
| 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) | |

Обучающегося Левченко Матвея Александровича, ИСИб-24-1

(ФИО, группа, подпись)

Руководитель практики от института ИТиАД

Кононенко Р.В., доцент института ИТиАД

(ФИО, должность, подпись)

Ссылка на резюме на сайте superjob.ru:

<https://www.superjob.ru/resume/programmist-razrabotchik-c-55732447.html>

Руководитель образовательной программы

Кононенко Р.В., доцент института ИТиАД

(ФИО, должность, подпись)

Изображение выглядит как шаблон, Графика, пиксель, дизайн

Содержимое, созданное искусственным интеллектом, может быть неверным.Оценка по практике

(ФИО, подпись, дата)

Содержание отчета на \_\_\_ стр. Приложение к отчету на \_\_\_ стр.

Иркутск 2025

Ссылка на резюме на сайте hh.ru:

<https://irkutsk.hh.ru/resume/36d0d197ff0efd13800039ed1f593176734853>

**Индивидуальное задание на прохождение**

**учебной практики: технологической (проектно-технологической) практики**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| для | Левченко Матвея Александровича | | | | |
|  | (ФИО обучающегося полностью) | | | | |
| обучающегося | | 1 | курса | группы | ИСИб-24-1 |

по направлению подготовки Информационные системы и технологии

профиль Интеллектуальные системы обработки информации и управления

Место прохождения практики: ИРНИТУ

Сроки прохождения практики с «16» июня 2025 г. по «\_28\_» июня 2025 г.

Цели и задачи прохождения практики:

Содержание практики, вопросы, подлежащие изучению:

Планируемые результаты практики:

Руководитель практики от

института ИТиАД

\_\_\_\_\_\_\_\_\_\_\_/ Кононенко Р.В. /

(подпись

**Согласовано:**

Руководитель ООП

\_\_\_\_\_\_\_\_\_\_\_/Кононенко Р.В./

(подпись

«\_\_\_»\_\_\_\_\_\_\_\_\_\_2025 г.

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

\_\_\_\_\_\_\_\_\_\_«16» июня 2025 г.

(подпись

**Цель практики:**

Повышение профессиональной квалификации и формирование практических навыков в области инженерии и программирования

**Задачи практики:**

1. Выполнение поставленных заданий по тематике практики
2. Изучение необходимой теории для выполнения задач
3. Участие в экскурсиях на предприятия, потенциально являющиеся базами для прохождения практики

**Содержание практики:**

* Решение базовых задач на программирование, для улучшения навыков
* Изучение и освоение работы с аппаратной платформой **Arduino**
* Применение языка программирования **Python** для моделирования систем машинного зрения
* Посещение экскурсий в IT-компаниях

**Планируемые результаты практики:**

1. Улучшение навыков программирования
2. Освоение навыков работы с платформой Arduino
3. Получение опыта решения разнообразных задач и практического применения методов моделирования с использованием Python
4. Расширение представлений о профессиональной деятельности и предприятиях, предоставляющих возможность прохождения практики

**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.

Изображение выглядит как Шрифт, снимок экрана, число, белый

Содержимое, созданное искусственным интеллектом, может быть неверным.

Приложение 1

Напишите программу, рассчитывающую минимальное время спуска (сумму чисел в пути с вершины до основания).

**Формат входных данных**

В первой строке дано целое число N - высота пирамиды, далее следуют N строк из чисел, разделённых пробелом (в каждой строке на 1 число больше, чем в предыдущей)

**Формат выходных данных**

Сумма чисел в пути с вершины до основания (одно число)

Последовательность участков маршрута (числа, разделённые пробелом)

**Решение:**

#include <iostream>

#include <random>

#include <vector>

#include <algorithm>

using namespace std;

vector<vector<int>> Rand(int n) {

vector<vector<int>> mountain;

random\_device rd;

mt19937 gen(rd());

uniform\_int\_distribution<> dist(1, 99);

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

vector<int> row;

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

row.push\_back(dist(rd));

}

mountain.push\_back(row);

}

for (const auto& row : mountain) {

for (const auto& num : row) {

cout << num << " ";

}

cout << endl;

}

return mountain;

}

void MinWay(vector<vector<int>> mountain, int n) {

if (n == 0) {

cout << "0" << endl;

return;

}

int res = mountain[0][0];;

vector<int> way;

way.push\_back(mountain[0][0]);

int current\_num = 0;

for (int i = 1; i < n; i++) {

if (current\_num + 1 >= mountain[i].size()) {

res += mountain[i][current\_num];

way.push\_back(mountain[i][current\_num]);

}

else {

if (mountain[i][current\_num] > mountain[i][current\_num + 1] && mountain[i][current\_num + 1] != 0) {

res += mountain[i][current\_num + 1];

way.push\_back(mountain[i][current\_num + 1]);

current\_num++;

}

else {

res += mountain[i][current\_num];

way.push\_back(mountain[i][current\_num]);

}

}

}

cout << res << endl;

for (const auto& i : way) {

cout << i << " ";

}

cout << endl;

}

int main() {

int n = 0;

cin >> n;

MinWay(Rand(n), n);

}

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Содержимое, созданное искусственным интеллектом, может быть неверным. Изображение выглядит как текст, Шрифт, снимок экрана, черно-белый

Содержимое, созданное искусственным интеллектом, может быть неверным. Изображение выглядит как текст, Шрифт, снимок экрана, типография

Содержимое, созданное искусственным интеллектом, может быть неверным.

Примеры входных/выходных данных для задания 1

**Задание № 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.

Изображение выглядит как круг, белый

Содержимое, созданное искусственным интеллектом, может быть неверным.Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Содержимое, созданное искусственным интеллектом, может быть неверным.

Приложение 2

**Формат входных данных**

В первой строке дано целое число N – количество узлов сети и M – число линий связи. Далее следуют M строк из чисел, разделённых пробелом (узлы, которые связывает данная линия)

**Формат выходных данных**

Число необходимых линий связи (одно число)

**Решение:**

#include <iostream>

#include <vector>

enum Color {

White,

Grey,

Black

};

std::vector<std::vector<int>> vertices;

std::vector<Color> colors;

void DFS(int index) {

colors[index] = Grey;

for (int ner : vertices[index]) {

if (colors[ner] == White) {

DFS(ner);

}

}

colors[index] = Black;

}

int countComponents(int n) {

int count = 0;

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

if (colors[i] == White) {

DFS(i);

count++;

}

}

return count;

}

int main() {

int n = 0;

int m = 0;

std::cin >> n >> m;

vertices.resize(n);

colors.resize(n, White);

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

int from = 0;

int to = 0;

std::cin >> from >> to;

from--;

to--;

vertices[from].push\_back(to);

vertices[to].push\_back(from);

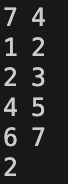
}

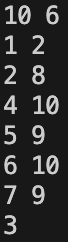
int components = countComponents(n);

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

return 0;

}

 Изображение выглядит как текст, Шрифт, снимок экрана, типография

Содержимое, созданное искусственным интеллектом, может быть неверным. 

Примеры входных/выходных данных для задания 2

**Задание № 3**

В 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 абсолютна 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.

**Формат входных данных**

Первая строка входных данных содержит два целых числа – n (1 или больше) и k (0 или больше) это количество чисел в массиве и количество запросов соответственно.

Следующие k строк содержат:

* либо 1 i x – Учетная информация о количестве, выпавшего на улице i (больше 0) x миллиметров снега.
* либо 2 u r – Запрос на подсчет количества снега на улицах от u до r (u и r больше 0 и могут быть равны друг другу)

**Формат выходных данных**

На каждый запрос второго типа надо вывести единственное число – суммарное выпавшего на них снега с момента начала наблюдения.

**Решение:**

#include <iostream>

#include <vector>

int main() {

int n = 0;

int k = 0;

std::cin >> n >> k;

std::vector<int> vec(n, 0);

std::vector<int> output;

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

int request = 0;

int a = 0;

int b = 0;

std::cin >> request >> a >> b;

if (request == 1) {

vec[a - 1] = b;

} else if (request == 2) {

int res = 0;

for (int j = a; j <= b; j++) {

res += vec[j - 1];

}

output.push\_back(res);

}

}

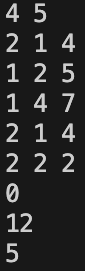
for (int res : output) {

std::cout << res << std::endl;

}

return 0;

}

 Изображение выглядит как текст, Шрифт, типография, число

Содержимое, созданное искусственным интеллектом, может быть неверным. Изображение выглядит как текст, Шрифт, снимок экрана, типография

Содержимое, созданное искусственным интеллектом, может быть неверным.

Примеры входных/выходных данных для задания 3

**Задание на практику № 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. Для примера возьмем приведенную выше перестановку и слово transposition, которое состоит тоже из 13 букв. Далее, следуя перестановке, на первую позицию поставим пятую букву слова, на вторую − одиннадцатую букву и так далее. В итоге получим sinoptsntiora. К этому слову снова применим эту же перестановку и получим poartsnoitsin. Повторив эти стадии шифрования k раз, получим зашифрованное сообщение.



Приложение 3

Вам дано зашифрованное таким образом слово, шифрующая перестановка P и число k. Необходимо восстановить слово.

**Формат входных данных**

Первая строка входных данных содержит 2 числа – n и k (1 или больше, могут быть равны). Следующая строка содержит перестановку длиной n, числа разделяются пробелом. Третья строка содержит зашифрованное слово длиной n.

**Формат выходных данных**

Вывести одну строку − исходное слово.

**Решение:**

#include <iostream>

#include <vector>

#include <string>

int main() {

int n = 0;

int k = 0;

std::cin >> n >> k;

std::vector<int> partition(n);

std::vector<char> word(n);

std::vector<char> temp(n);

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

std::cin >> partition[i];

}

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

std::cin >> word[i];

}

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

for (int j = 0; j < n; j++) {

temp[partition[j] - 1] = word[j];

}

word = temp;

}

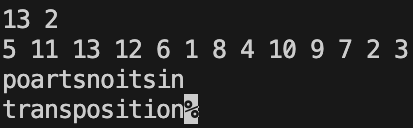
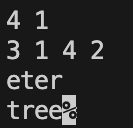
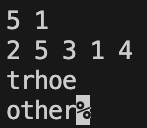
for (int i = 0; i < n; i++) {

std::cout << word[i];

}

return 0;

}



Примеры входных/выходных данных для задания 4

**Задание на практику № 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.

**Формат входных данных**

В первой строке через пробел заданы высота h и ширина w матрицы (длина и ширина 10 или больше, но не больше 50, могут быть равны). В следующих строках заданы значения матрицы по строкам и столбцам. В матрице всегда есть только одна фигура. Фигура отстоит от краев матрицы минимум на один ноль. Начало координат в левом верхнем углу. Координаты растут вниз и вправо.

**Формат выходных данных**

Координаты верхнего левого и правого нижнего угла прямоугольника отделенные пробелами. Координаты задаются номером строки и номером столбца. Нумерация начинается с 0.

**Решение:**

#include <iostream>

#include <vector>

int main() {

int h = 0;

int w = 0;

std::cin >> h >> w;

std::vector<std::vector<int>> matrix(h, std::vector<int>(w));

bool flag = false;

int save\_i = 0;

int save\_j = 0;

int save\_i\_end = 0;

int save\_j\_end = 0;

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

for (int j = 0; j < w; j++) {

std::cin >> matrix[i][j];

if (matrix[i][j] == 1 && !flag) {

save\_i = i;

save\_j = j;

flag = true;

}

if (matrix[i][j] == 1) {

save\_i\_end = i;

save\_j\_end = j;

}

}

}

std::cout << save\_i - 1 << " " << save\_j - 1 << " ";

std::cout << save\_i\_end + 1 << " " << save\_j\_end + 1;

return 0;

}



Примеры входных/выходных данных для задания 5

**Задание на практику № 6**

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

**Формат входных данных**

В первой строке пишем число n (не больше 50). Далее в n строках пишем по 4 натуральных числа (a, b, x, y).

**Формат выходных данных**

Выводим n чисел через пробел, каждое число – максимальное число работающих пар из микроконтроллера и модуля управления мотором можно составить для строки.

**Решение:**

#include <iostream>

#include <vector>

int main()

{

int n = 0;

std::cin >> n;

int res = 0;

std::vector<int> results;

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

int a, b, x, y;

std::cin >> a >> b >> x >> y;

int B\_pairs = std::min(b, x);

int remaining\_x = x - B\_pairs;

int A\_pairs = std::min(a, std::max(0, remaining\_x) + y);

int total\_pairs = B\_pairs + A\_pairs;

results.push\_back(total\_pairs);

}

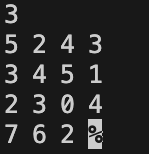
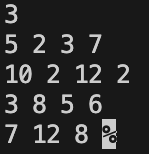
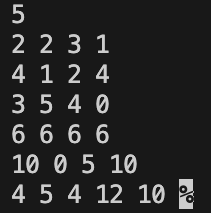
for (int i = 0; i < n; i++) {

std::cout << results[i] << " ";

}

return 0;

}



Примеры входных/выходных данных для задания 3

**Задание на практику № 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.

**Формат входных данных**

В единственной строке находится строка из шести символов, состоящая из цифр и заглавных латинских букв.

**Формат выходных данных**

Если строка является автомобильным номером, то необходимо вывести "Yes", в ином случае – "No" без кавычек.

**Решение:**

#include <iostream>

#include <vector>

int main()

{

std::vector<char> num(6);

for (int i = 0; i < 6; i++) {

std::cin >> num[i];

}

bool flag = true;

for (int i = 0; i < 6; i++) {

if (1 <= i && i <= 3) {

if (!(48 <= static\_cast<int>(num[i]) && static\_cast<int>(num[i]) <= 57)) {

flag = false;

}

} else {

if (!(65 <= static\_cast<int>(num[i]) && static\_cast<int>(num[i]) <= 90)) {

flag = false;

}

}

}

if (flag) {

std::cout << "YES";

}

else {

std::cout << "NO";

}

}



Примеры входных/выходных данных для задания 7

**Задание на практику № 8**

Составить светодиодную матрицу размером не менее 8 на 8 светодиодов (пример на рисунке ниже размером 4 на 4)

!Обратите внимание на ориентацию светодиодов на поле

A circuit board with wires

AI-generated content may be incorrect.

Приложение 4

На матрицу вывести инфографику с различными динамично меняющимися изображениями.

**Решение:**

#include <Adafruit\_NeoPixel.h>

#define LED\_PIN 3

#define LED\_COUNT 64

#define W 8

#define H 8

Adafruit\_NeoPixel leds(LED\_COUNT, LED\_PIN, NEO\_RGB + NEO\_KHZ800);

inline uint16\_t xyToIndex(uint8\_t x, uint8\_t y) {

if (y % 2 == 0) {

return y \* W + (W - 1 - x);

} else {

return y \* W + x;

}

}

inline void setXY(uint8\_t x, uint8\_t y, uint32\_t c) {

if (x < W && y < H) {

leds.setPixelColor(xyToIndex(x, y), c);

}

}

const float AMP = 1.0f;

const float WAVELEN = 1.6f;

const float SPEED = 0.06f;

const float X\_SCALE = 1.4f;

const uint8\_t FPS = 25;

const uint16\_t FRAME\_INTERVAL = 1000 / FPS;

const uint8\_t H\_WHITE = 2;

const uint8\_t H\_BLUE = 3;

const uint8\_t H\_RED = 3;

uint32\_t C\_WHITE, C\_BLUE, C\_RED;

unsigned long lastFrame = 0;

float phase = 0.0f;

void setup() {

leds.begin();

leds.setBrightness(60);

leds.clear();

leds.show();

C\_WHITE = leds.Color(255, 255, 255);

C\_BLUE = leds.Color(0, 0, 255);

C\_RED = leds.Color(0, 255, 0);

}

void loop() {

unsigned long now = millis();

if (now - lastFrame < FRAME\_INTERVAL) {

return;

}

lastFrame = now;

phase += SPEED;

leds.clear();

for (uint8\_t y = 0; y < H; y++) {

for (uint8\_t x = 0; x < W; x++) {

float offset = AMP \* sinf((float)x \* X\_SCALE / WAVELEN + phase);

float yy = (float)y + offset;

uint32\_t color;

if (yy < H\_WHITE) {

color = C\_WHITE;

} else if (yy < H\_WHITE + H\_BLUE) {

color = C\_BLUE;

} else {

color = C\_RED;

}

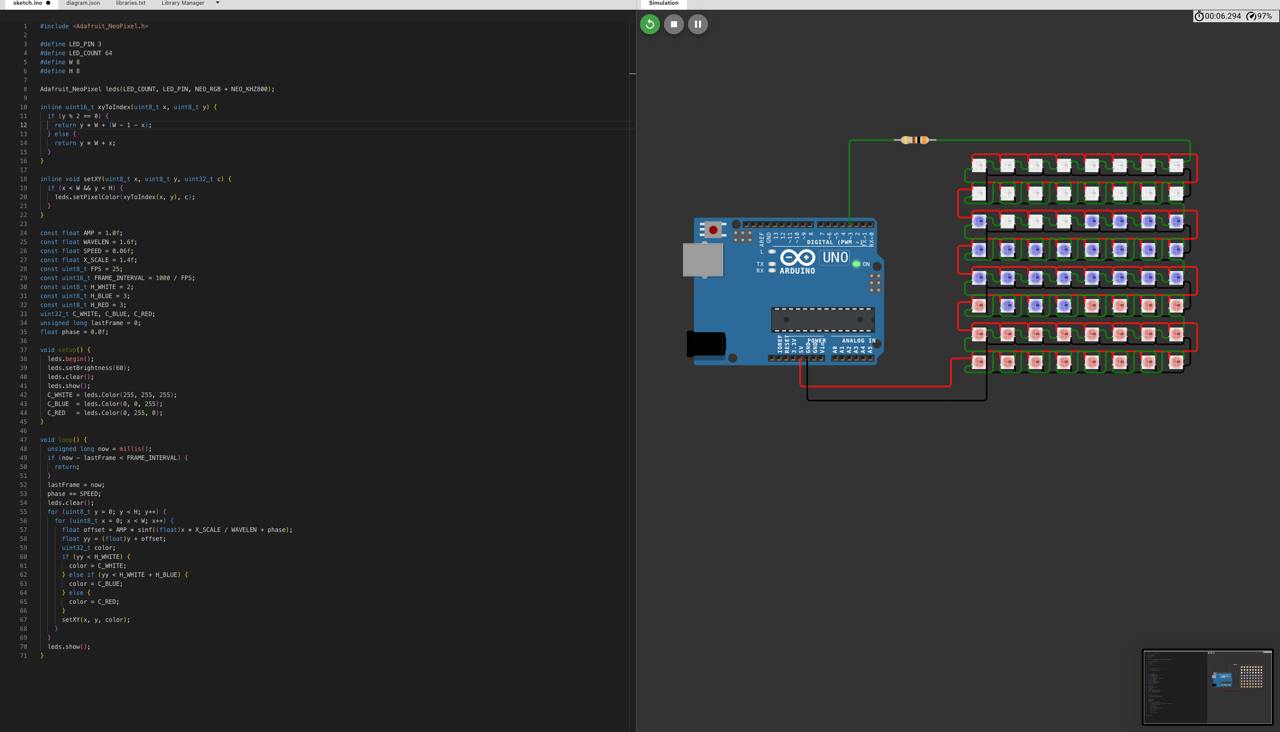
setXY(x, y, color);

}

}

leds.show();

}



Результат задания 8 “анимация флага России” (1)

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Результат задания 8 “анимация флага России” (2)

**Задание на практику № 9**

**Задачи:**

1. Собрать схему имитирующую работу автоматических дверей
2. Подобрать номинал резисторов для светодиодов
3. Написать программу для управления процессом работы автоматических дверей.

Схема приведена на рисунке 1.

Зеленый светодиод – двери отрываются.

Красный светодиод ­– двери закрываются.

Фоторезистор имитируют процесс приближения-удаления человека от дверей.

A circuit board with wires

AI-generated content may be incorrect.

Приложение 5 – Схема управления работой автоматических дверей

Изменение значений фоторезистора осуществляется при помощи ползунка (рисунок 2), изменение значения фоторезистора доступно только, когда запущен процесс моделирования.

A screenshot of a computer

AI-generated content may be incorrect.

Приложение 6 – Изменение значения фоторезистора

Логика работы программы:

1. По умолчанию горит светодиод, имитирующий закрытую дверь
2. Микроконтроллер считывает значение фоторезистора с аналогово пина
3. Если значение на пине превышает 512, на определённое время загорается светодиод, имитирующий открытую дверь, в последовательный порт выводится сообщение о событии.
4. После истечения заданного временного промежутка проверяется значение фоторезистора, если оно всё ещё превышает 512, дверь должна остаться открытой, в противном случае нужно включить индикацию закрытой двери, в последовательный порт выводится сообщение о событии.

**Решение:**

const int photoPin = A0;

const int redLedPin = 9;

const int greenLedPin = 5;

const int threshold = 512;

const int openTime = 5000;

bool doorOpen = false;

unsigned long doorTimer = 0;

void setup() {

pinMode(redLedPin, OUTPUT);

pinMode(greenLedPin, OUTPUT);

digitalWrite(redLedPin, HIGH);

digitalWrite(greenLedPin, LOW);

Serial.begin(9600);

Serial.println("Door close");

}

void loop() {

int photoValue = analogRead(photoPin);

if (photoValue > threshold) {

if (!doorOpen) {

openDoor();

}

doorTimer = millis();

}

if (doorOpen && (millis() - doorTimer > openTime)) {

if (analogRead(photoPin) <= threshold) {

closeDoor();

} else {

doorTimer = millis();

}

}

delay(100);

}

void openDoor() {

digitalWrite(redLedPin, LOW);

digitalWrite(greenLedPin, HIGH);

doorOpen = true;

doorTimer = millis();

Serial.println("Door open");

Serial.println(analogRead(photoPin));

}

void closeDoor() {

digitalWrite(greenLedPin, LOW);

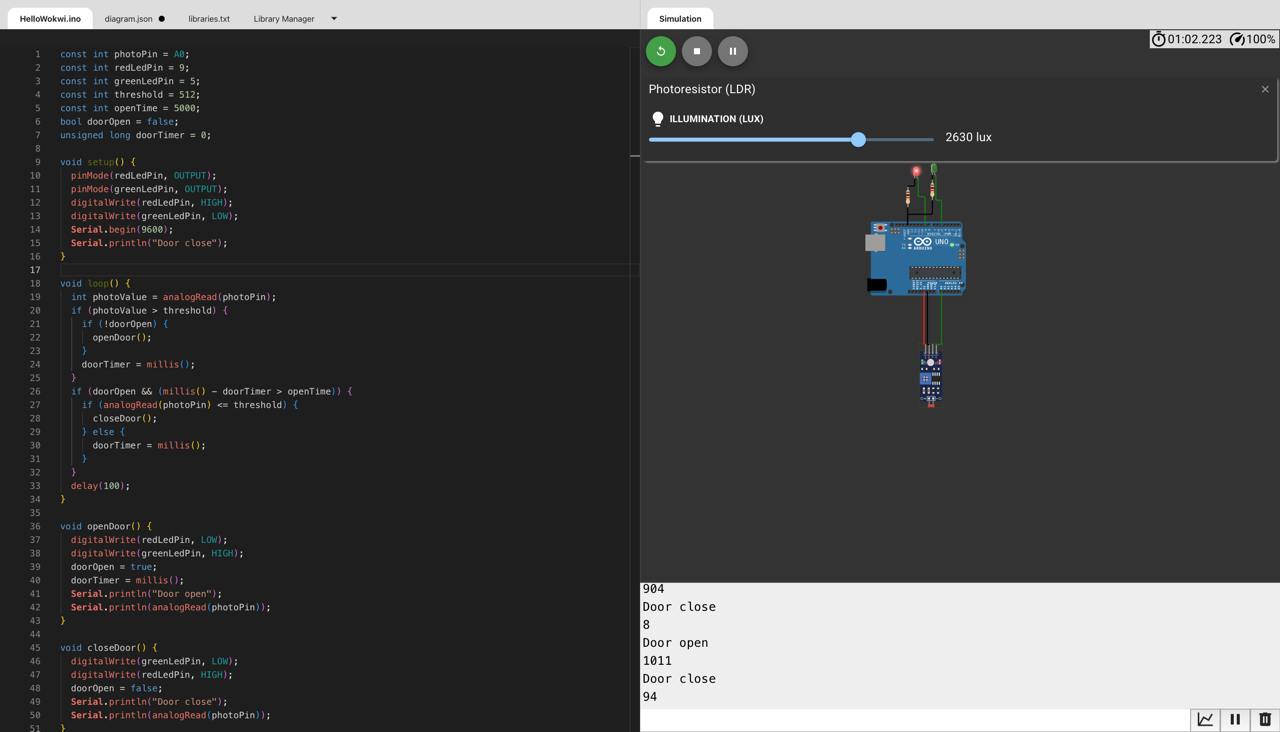
digitalWrite(redLedPin, HIGH);

doorOpen = false;

Serial.println("Door close");

Serial.println(analogRead(photoPin));

}



Результат работы программы для задания 9

**Задание на практику № 10**

**Задачи:**

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

A circuit board with wires and a switch

AI-generated content may be incorrect.

Приложение 7 – Подключение сервопривода

Логика работы программы:

1. Программа находится в ожидании ввода данных в последовательный порт
2. Когда в последовательный порт вводятся данные, программа проверяет их на корректность (Должны приниматься только числовые значения в диапазоне от 0 до 180)
3. Если введён угол поворота, отличный от текущего угла, сервопривод плавно поворачивается в заданное положение.

**Решение:**

#include <Servo.h>

Servo myservo;

int currentAngle = 0;

int targetAngle = 0;

void setup() {

myservo.attach(9);

Serial.begin(9600);

Serial.println("Enter angle (0-180):");

myservo.write(0);

}

void loop() {

if (Serial.available() > 0) {

String input = Serial.readStringUntil('\n');

input.trim();

if (isDigits(input)) {

int newAngle = input.toInt();

if (newAngle >= 0 && newAngle <= 180) {

targetAngle = newAngle;

Serial.print("Moving to: ");

Serial.print(targetAngle);

Serial.println("°");

} else {

Serial.println("Error: Angle must be 0-180°");

}

} else {

Serial.println("Error: Numbers only (0-180)");

}

}

if (currentAngle != targetAngle) {

if (currentAngle < targetAngle) currentAngle++;

else currentAngle--;

myservo.write(currentAngle);

delay(20);

}

}

bool isDigits(const String &str) {

for (unsigned int i = 0; i < str.length(); i++) {

if (!isdigit(str[i])) return false;

}

return true;

}

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Результат работы программы для задания 10

**Задание на практику № 11**

Find all green and yellow objects in the image. Find the centers of all green and yellow objects. Highlight the boundaries of green objects using a blue frame. Select the boundaries of yellow objects using a green frame.

(*Найдите все зеленые и желтые объекты на изображении. Найдите центры всех зеленых и желтых объектов. Выделите границы зеленых объектов, используя синюю раму. Выберите границы желтых объектов, используя зеленую раму*.)

**Решение:**

import cv2

import numpy as np

image = cv2.imread("/Users/matvey/PyCharmMiscProject/kuk.jpg")

cv2.imshow("original", image)

cv2.waitKey(0)

blurred\_image = cv2.GaussianBlur(image, (11, 11), 0)

cv2.imshow("blurred", blurred\_image)

cv2.waitKey(0)

hsv\_image = cv2.cvtColor(blurred\_image, cv2.COLOR\_BGR2HSV)

cv2.imshow("hsv", hsv\_image)

cv2.waitKey(0)

hsv\_min = np.array((36, 25, 25), np.uint8)

hsv\_max = np.array((70, 255, 255), np.uint8)

green\_mask = cv2.inRange(hsv\_image, hsv\_min, hsv\_max)

cv2.imshow("mask\_green", green\_mask)

cv2.waitKey(0)

contours, hierarchy = cv2.findContours(green\_mask.copy(), cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

for i in contours:

if cv2.contourArea(i) < 100:

continue

x, y, w, h = cv2.boundingRect(i)

cx, cy = int(x + w / 2), int(y + h / 2)

cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 2)

cv2.circle(image, (cx, cy), 9, (0, 0, 255), 3)

hsv\_min = np.array((20, 100, 100), np.uint8)

hsv\_max = np.array((36, 255, 255), np.uint8)

yellow\_mask = cv2.inRange(hsv\_image, hsv\_min, hsv\_max)

cv2.imshow("mask\_yellow", yellow\_mask)

cv2.waitKey(0)

contours, hierarchy = cv2.findContours(yellow\_mask.copy(), cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

for i in contours:

if cv2.contourArea(i) < 100:

continue

x, y, w, h = cv2.boundingRect(i)

cx, cy = int(x + w / 2), int(y + h / 2)

cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv2.circle(image, (cx, cy), 9, (0, 0, 255), 3)

cv2.imshow('result', image)

cv2.waitKey(0)

cv2.destroyAllWindows()

Изображение выглядит как текст, Мультимедийное программное обеспечение, программное обеспечение, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Результат работы программы для задания 11 (тест 1)

Изображение выглядит как Красочность

Содержимое, созданное искусственным интеллектом, может быть неверным.

Результат работы программы для задания 11 (тест 2)

**Отчёт по посещению экскурсии 24.06.2025 в филиале АО «Системный оператор Единой энергетической системы»**

**Профиль деятельности**: Иркутское РДУ отвечает за управление электроэнергетикой Иркутской области. Основная его работа — следить за тем, чтобы электростанции вырабатывали ровно столько энергии, сколько нужно потребителям, и чтобы линии электропередачи и подстанции работали без перебоев. Под их контролем находятся крупные гидроэлектростанции — Братская, Усть-Илимская, Иркутская, а также тепловые станции в Иркутске, Ангарске, Братске и других городах. Всего это десятки электростанций и сотни линий и подстанций по всему региону.

**Задачи и подразделения:** Управление строится по нескольким направлениям. Диспетчеры следят за режимами работы станций и сетей, регулируют нагрузку и контролируют подачу энергии. Технические службы занимаются состоянием линий и оборудования. Есть отделы, которые планируют развитие и модернизацию системы, а также группы, отвечающие за устранение аварий. Помимо этого, в организации есть IT-подразделения: серверные, системы мониторинга, администраторы, которые обеспечивают работу программного обеспечения и инфраструктуры.

Во время экскурсии нам рассказали об истории создания диспетчерских служб и показали современный диспетчерский зал с большим экраном, где в реальном времени отображается работа всех объектов энергетики области. Позже мы посетили серверные комнаты, где находятся стойки с оборудованием, и пообщались с администратором, который отвечает за IT-часть. Это позволило взглянуть на работу организации не только со стороны энергетики, но и со стороны информационных технологий.

Общее впечатление осталось положительным: стало понятно, какое значение имеет эта структура для всего региона. Однако я понял, что направление деятельности Иркутского РДУ всё же ближе тем, кто хочет развиваться в энергетике и смежных сферах, тогда как мои интересы больше связаны с другими областями IT.

**Отчёт по посещению экскурсии 26.06.2025 в АО «ISPsystem»**

**Профиль деятельности:** ISPsystem — это российская компания-разработчик программного обеспечения для управления IT-инфраструктурой. Проще говоря, они делают программы, которые помогают работать с серверами, виртуальными машинами, дата-центрами и системами хостинга. Эти продукты используют не только в России, но и во многих странах мира. Среди основных решений — VMmanager (для управления виртуальными серверами), DCImanager (для работы с «железом» и учётом ресурсов), BILLmanager (для автоматизации продаж и биллинга), а также DNSmanager и IPmanager для управления сетевыми ресурсами.

**Задачи и подразделения**: Нам объяснили, что компания разделена на разные отделы. Есть команды разработчиков (frontend и backend), тестировщики (QA), дизайнеры интерфейсов (UI/UX), менеджеры и аналитики. Отдельно работают специалисты технической поддержки, которые помогают пользователям, а также администраторы, следящие за серверами и безопасностью систем. По сути, каждый отдел занимается своей частью работы: одни создают продукт, другие проверяют его качество, третьи общаются с клиентами и поддерживают его работу.

Во время экскурсии мы посмотрели не только рабочие места сотрудников, но и серверные стойки, узнали о внутренней инфраструктуре и организации процессов. Удивило, что в ISPsystem нужны не только программисты, но и специалисты других профилей, так что здесь можно развиваться в IT даже без глубоких знаний кода. Кроме того, компания заботится о сотрудниках — есть кухня, зоны отдыха, комфортные офисные пространства.

В целом экскурсия произвела хорошее впечатление. Было видно, что это современная компания, где создают серьёзные продукты, востребованные в IT-сфере. Для меня посещение стало полезным опытом: я понял, чем живёт IT-команда изнутри и какие направления могут быть интересны для дальнейшего развития.