ФГБОУ ВО «Глазовский государственный инженерно–педагогический университет имени В.Г. Короленко»

Кафедра математики и информатики

Среднее профессиональное образование

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

**ПМ. 04 СОПРОВОЖДЕНИЕ И ОБСЛУЖИВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ   
КОМПЬЮТЕРНЫХ СИСТЕМ**

**УП. 04 УЧЕБНАЯ ПРАКТИКА**

Специальность: 09.02.07 Информационные системы и программирование

Даты прохождения: 22.11.2025 – 05.12.2025

Тугбаева Полина Михайловна

Группа СИ 132

Руководитель учебной практики

Касаткин К. А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

г. Глазов, 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc215751676)

[1. ВЫПОЛНЕНИЕ ОБЩИХ ЗАДАНИЙ 4](#_Toc215751677)

[1.1 Выполнение тестов для подготовки 4](#_Toc215751678)

[2. МАКЕТ БИОНИЧЕСКОЙ РУКИ 6](#_Toc215751679)

[2.1 Основная задача индивидуального задания 6](#_Toc215751680)

[2.2 Изменение конфигурации приводов 17](#_Toc215751681)

[ЗАКЛЮЧЕНИЕ 21](#_Toc215751682)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 22](#_Toc215751683)

# ВВЕДЕНИЕ

Сопровождение и обслуживание программного обеспечения компьютерных систем становится неотъемлемой частью современных процессов разработки и эксплуатации программного обеспечения, обеспечивая эффективное взаимодействие между командами разработчиков и операционных специалистов. В ходе учебной практики, посвященной изучению ключевых аспектов сопровождения и обслуживания программного обеспечения компьютерных систем, я сосредоточилась на нескольких важнейших разделах, которые способствуют успешной реализации информационных систем.

В процессе практики нами было изучено: правила безопасности и условия работы в учебном кабинете, основы тестирования и документирования программного обеспечения.

Также мы получили индивидуальное задание, которое включало в себя настройку программного обеспечения для работы с оборудованием технопарка на примере комплекта «Макет бионической руки», задания на подготовку к квалификационному экзамену. В ходе выполнения индивидуального задания было необходимо изучить документацию к данному макету, произвести тестирование и настройку программного обеспечения.

# 1. ВЫПОЛНЕНИЕ ОБЩИХ ЗАДАНИЙ

## 1.1 Выполнение тестов для подготовки

В рамках подготовки к квалификационному экзамену по модулю ПМ04, нам было предложено выполнить тесты и практические задания.

Тестовые задания состояли из 3 частей. В 1 части было предложено решить 6 тестов по 15 вопросов, во 2 части – 6 тестов по 15 вопросов и в 3 части – 6 тестов по 10 вопросов. В 1 и 2 частях были предложены несколько вариантов ответа, один из которых был правильным. В 3 части ответ нужно было написать самостоятельно.

Нам было предложено выполнить 21 практическое задание, которые также будут включены в квалификационный экзамен. Все задания были по теме: Основные методы внедрения и анализа функционирования программного обеспечения.

Нами была выполнена регистрация на сайте олимпиады «Траектория будущего», «Нейросетевое искусство» и «Разработка игр».



Рисунок 1 – Результаты курса «Траектория будущего»

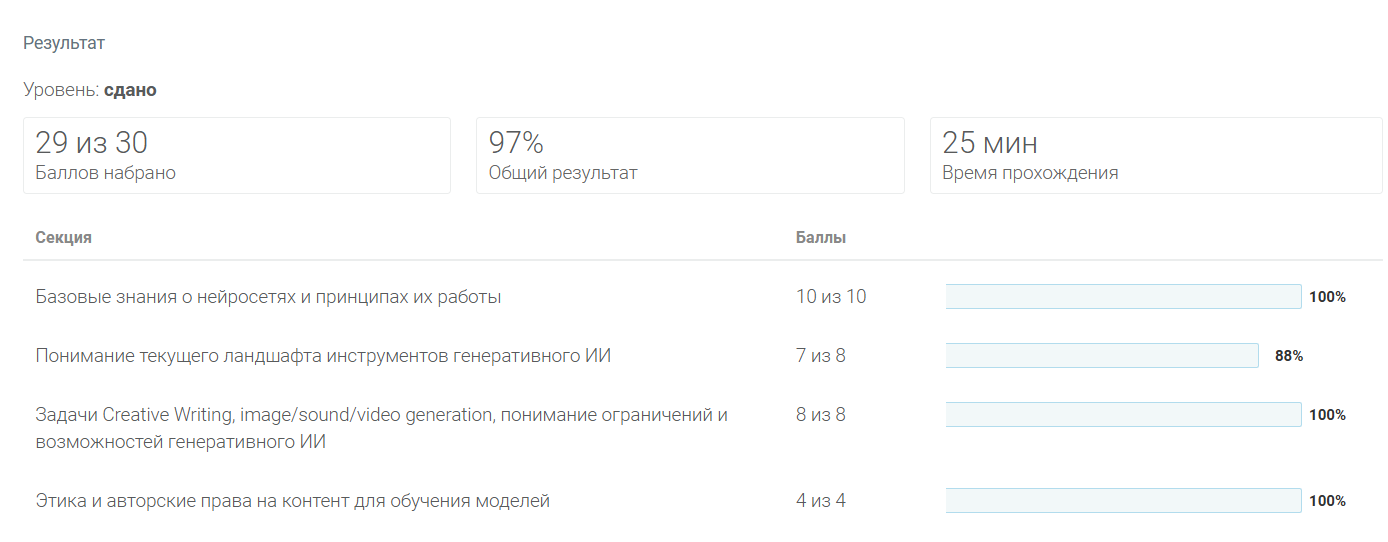


Рисунок 2 – Результаты курса «Нейросетевое искусство»

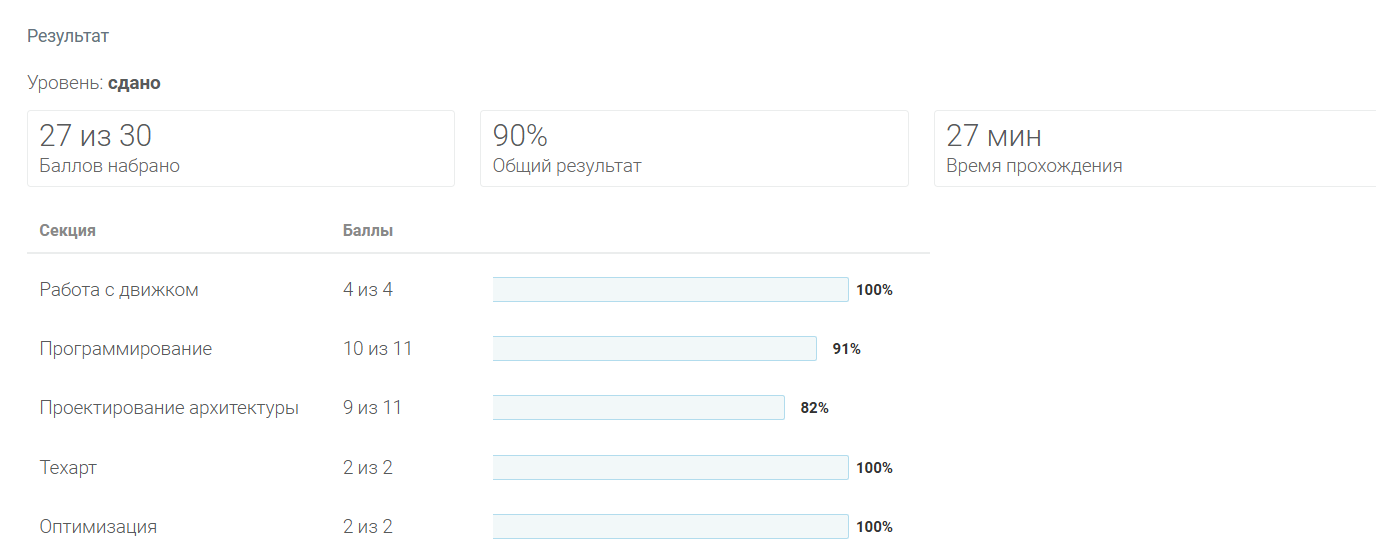


Рисунок 3 – Результаты курса «Разработка игр»

# 2. МАКЕТ БИОНИЧЕСКОЙ РУКИ

## 2.1 Основная задача индивидуального задания

Основным заданием на учебной практике стало индивидуальное задание по теме «Настройка программного обеспечения для работы с оборудованием технопарка на примере комплекта «Макет бионической руки».

Задачи, решаемые в ходе выполнения индивидуального задания:

1.   Запуск и тестирование макета бионического протеза руки человека.

2.   Работа с программной частью в среде Arduino IDE

3. Применение человеко–машинного интерфейса для управления макетом бионического протеза руки человека.

**Программное обеспечение**

В качестве программного обеспечения для выполнения данной работы используется среда разработки Arduino IDE (Рисунок 4), которая позволяет удобно писать код на языке Arduino C и тут же загружать его в плату.

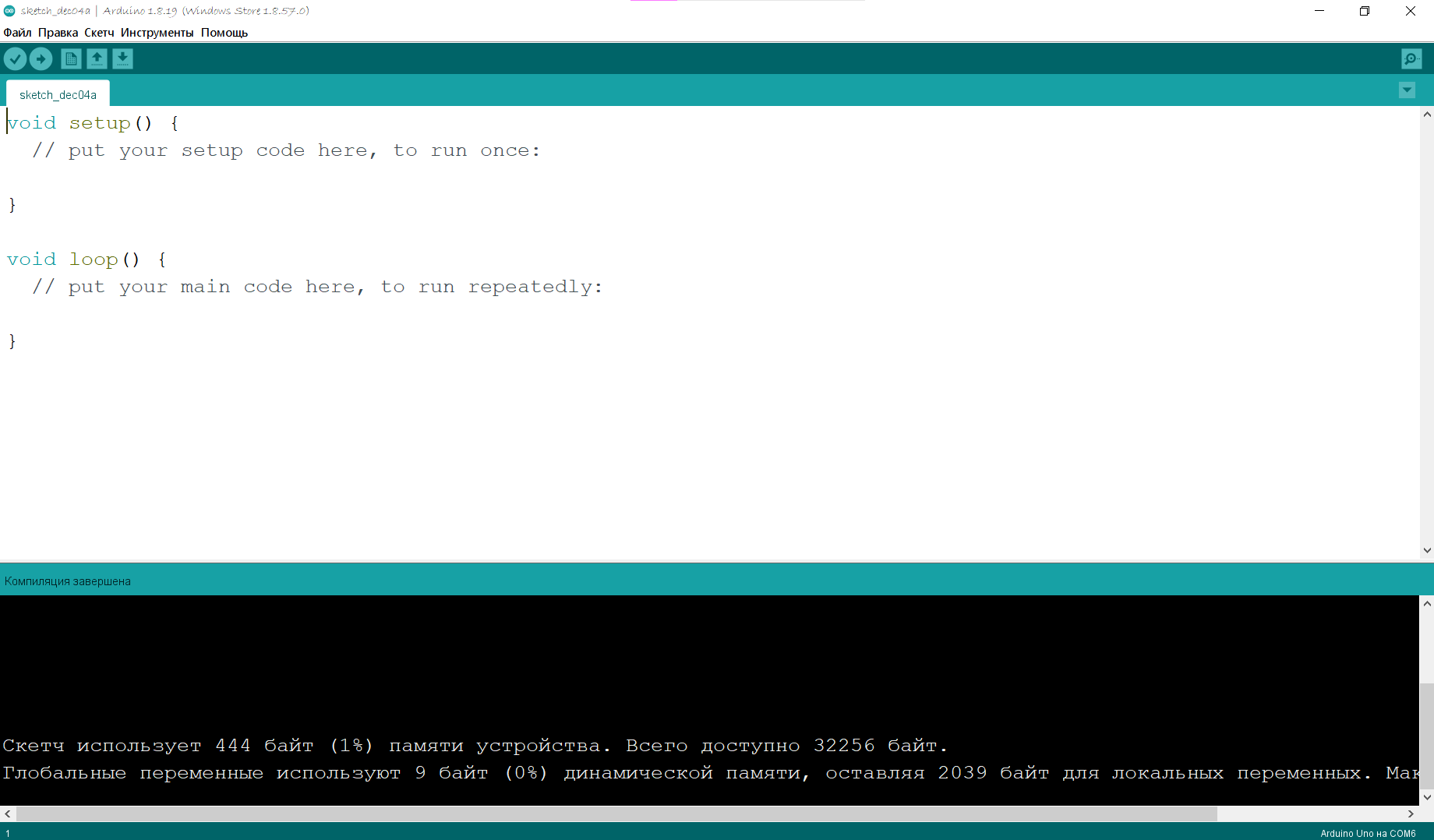


Рисунок 4 – Arduino IDE

**Оборудование**

1. Модуль сбора и отправки данных по радиоканалу («Модуль HUB») – 1 шт.

2. Плата Arduino Uno – 1 шт.

3. Корпусированный модуль для регистрации электромиографического сигнала («Модуль EMG») – 2 шт.

4. Плата расширения для приема данных по радиоканалу от модуля HUB Shield EMG Interface (« Плата расширения HMI») – 1 шт.

5. Макет бионического протеза руки человека – 1 шт.

6. Плата расширения для сервоприводов – 1 шт.

7. Аккумулятор типоразмера 18650 (далее – «Аккумулятор») – 2 шт.

8. Отсек для аккумуляторов – 1 шт.



Рисунок 5 – Модуль HUB с подключенными модулями EMG

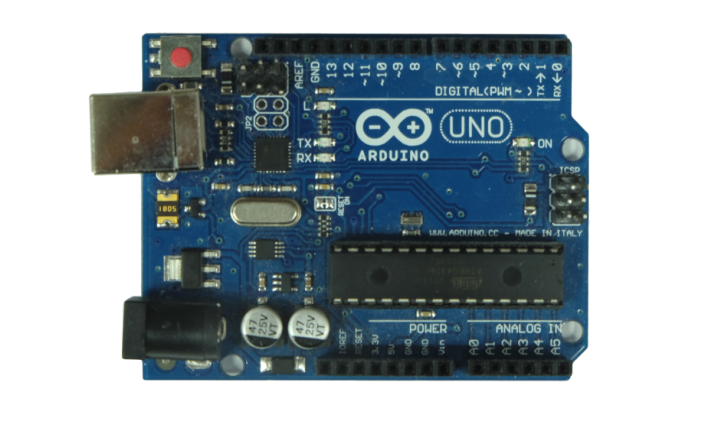


Рисунок 6 – Плата Arduino UNO

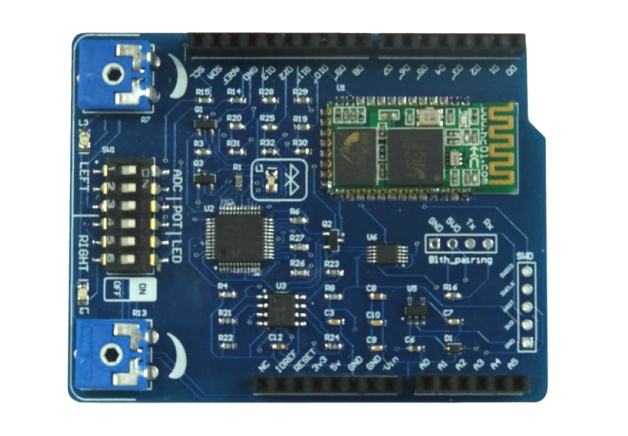


Рисунок 7 – Плата расширения HMI

**Сборка и подключение бионической руки**

Первый шаг – подключение платы расширения HMI к плате Arduino UNO. Для этого необходимо установить плату расширения HMI в пины платы Arduino UNO. Аналоговые и цифровые порты платы Arduino Uno должны совпадать с аналоговыми портами платы расширения HMI. При правильном подключении плата расширения HMI будет расположена над платой Arduino Uno.

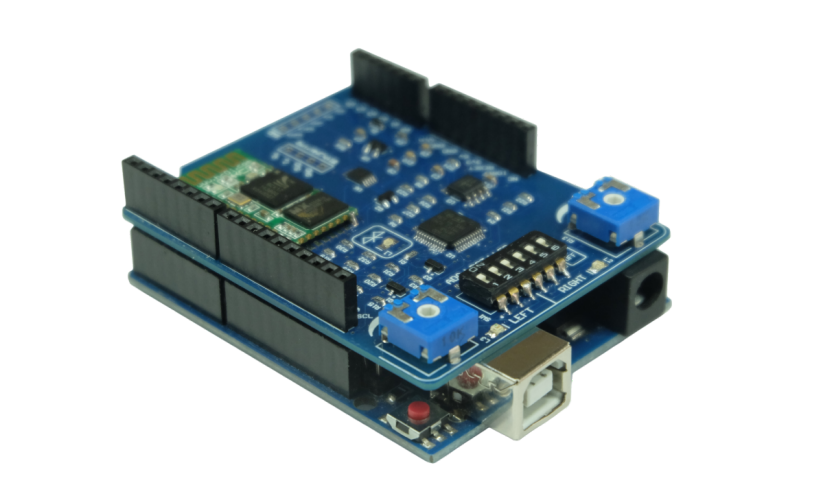


Рисунок 8 – Плата расширения HMI, подключенная к плате Arduino UNO

Затем необходимо подключить плату Arduino Uno к ноутбуку, при этом индикатор на плате расширения HMI начнет часто мигать красным цветом.

Далее установим модули EMG на мышцы предплечья.

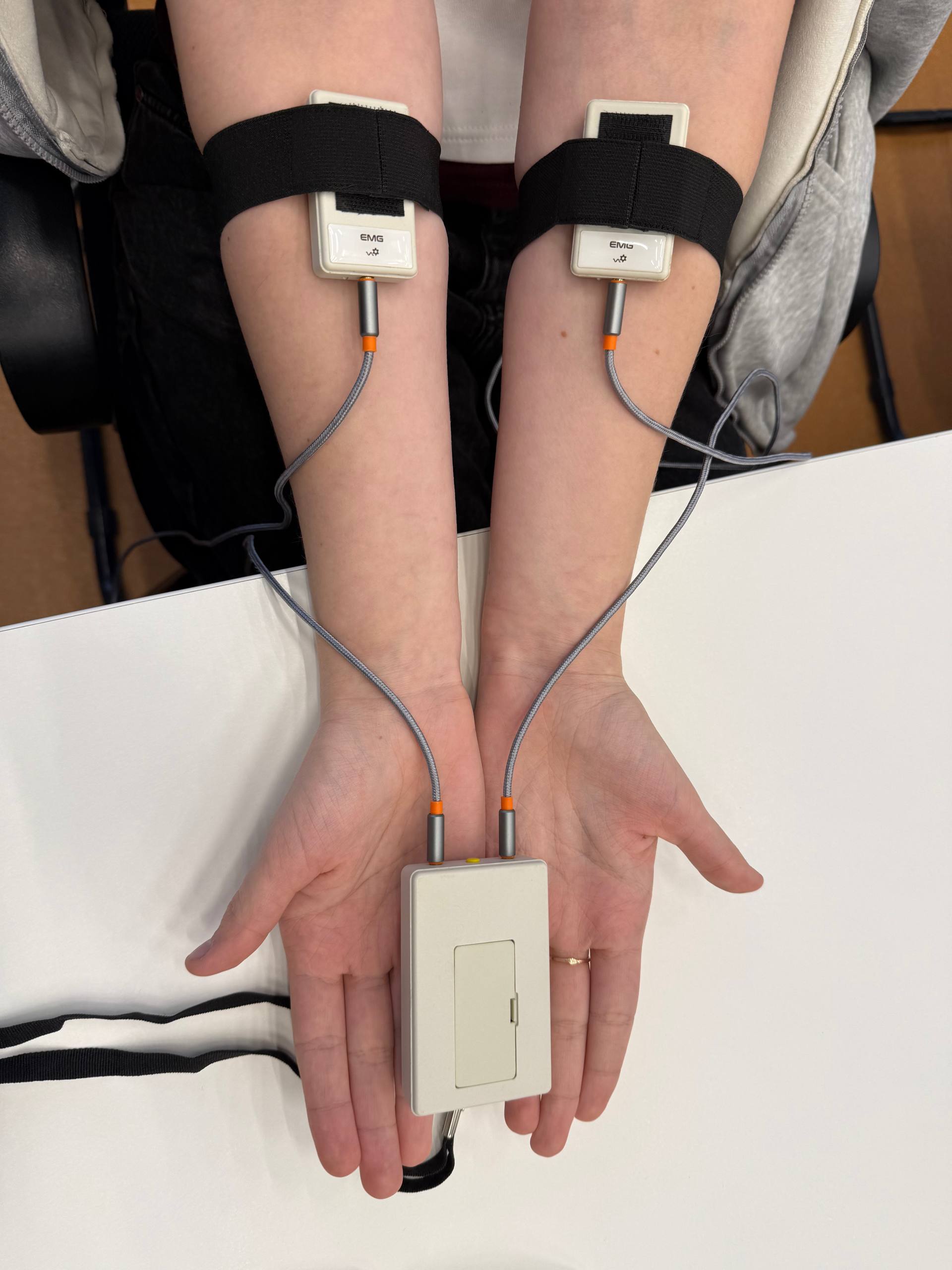


Рисунок 9 – Подключение модулей EMG к мышцам предплечья Поздеевой Александры

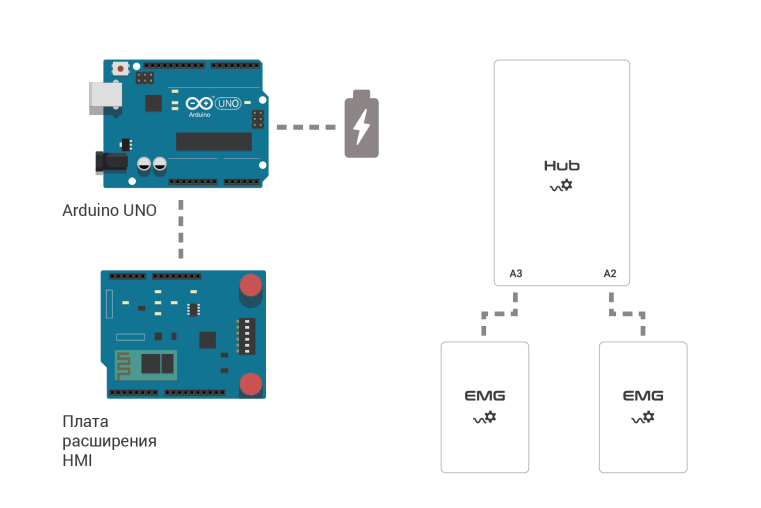


Рисунок 10 – Общая схема сборки и подключения установки

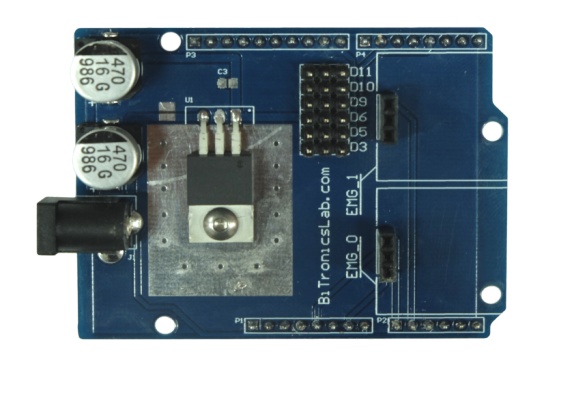


Рисунок 11 – Плата расширения для сервоприводов

Далее необходимо подключить каждый сервопривод руки к плате расширения сервоприводов в соответствии с таблицей подключения, после чего комплект бионической руки будет полностью готов к работе.



Рисунок 12 – Таблица подключения сервоприводов к портам платы расширения для сервоприводов

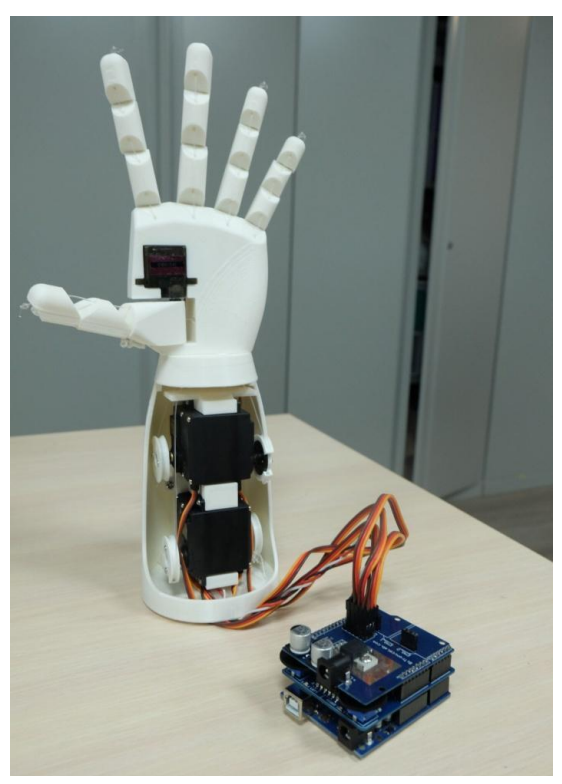


Рисунок 13 – Подключенная бионическая рука

После успешного подключения модулей EMG к плате, нужно загрузить первую программу для проверки работоспособности этих модулей. Перед загрузкой нужно выбрать тип используемой платы, и порт подключения в Arduino IDE.

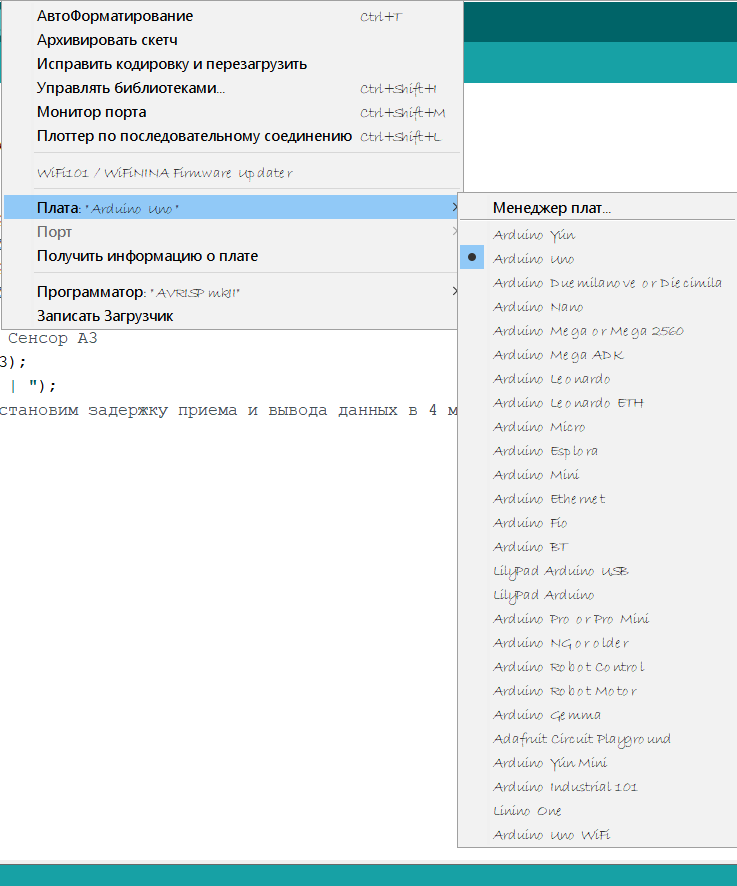


Рисунок 14 – Выбор платы в Arduino IDE

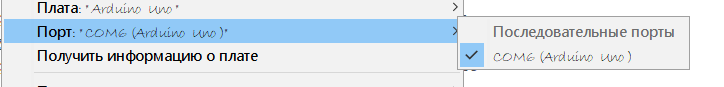


Рисунок 15 – Выбор порта в Arduino IDE

При запуске программы нужно открыть монитор порта, здесь будет видно какие значения выдают оба модуля EMG. При максимально возможном напряжении мышц значения могут доходить до 900, но при обычном сжатии руки в кулак, значения находятся в диапазоне от 170 до 250.

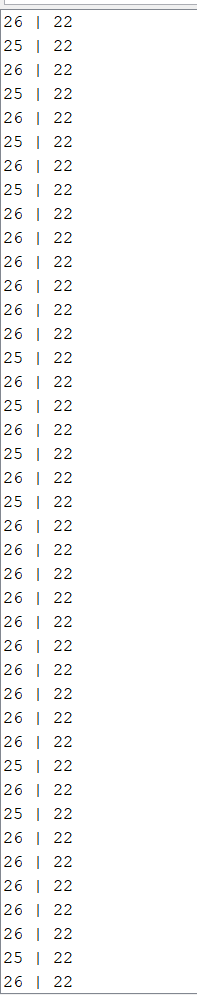


Рисунок 16 –  Значения с модулей EMG при расслабленных мышцах

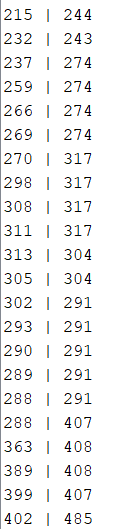


Рисунок 17 – Значения с модулей EMG при сжатии левой руки

Ниже представлен исходный код первой программы

void setup() {

Serial.begin(9600);

}

void loop() {

int a2 = analogRead(A2);

int a3 = analogRead(A3);

Serial.println(a2);

Serial.print(a3);

Serial.print(« | ");

delay(4);

}

После успешной проверки работоспособности модулей EMG, необходимо загрузить вторую программу, которая будет демонстрировать работу бионической руки.

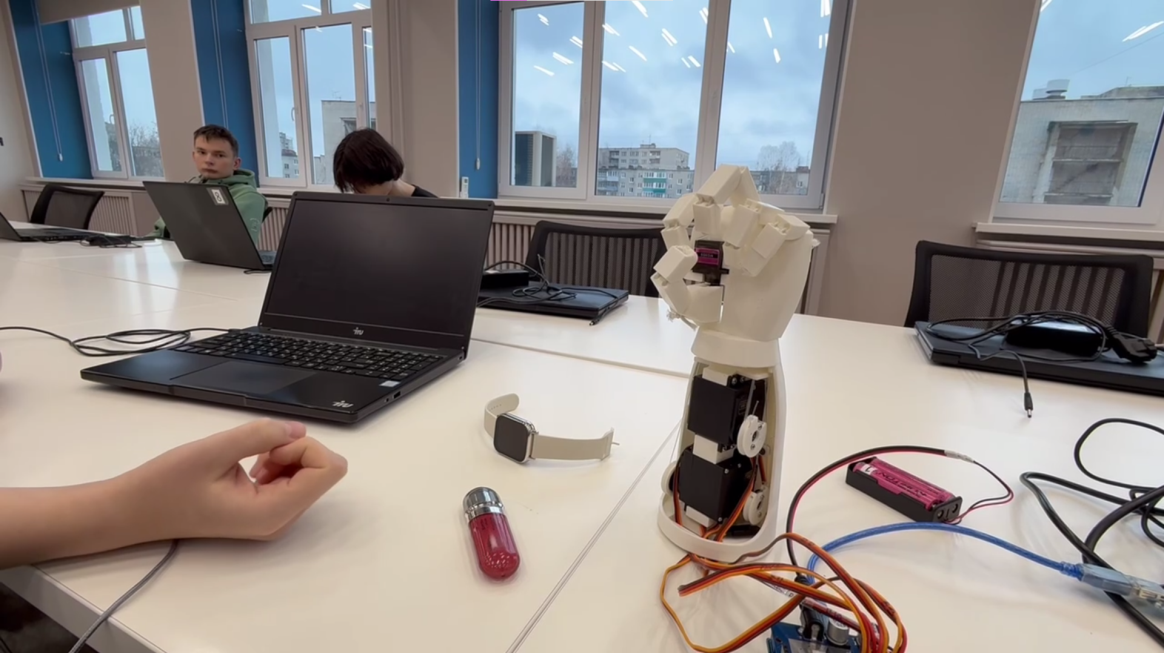


Рисунок 18 – Сжатие руки при помощи напряжения мышц

Ниже представлен исходный код второй программы.

#include <Servo.h>

#define MAX\_THUMB1 160

#define MIN\_THUMB1 80

#define MAX\_THUMB2 180

#define MIN\_THUMB2 30

#define MAX\_INDEX\_FINGER 140

#define MIN\_INDEX\_FINGER 0

#define MAX\_MIDDLE\_FINGER 180

#define MIN\_MIDDLE\_FINGER 0

#define MAX\_LITTLE\_RING\_FINGER 180

#define MIN\_LITTLE\_RING\_FINGER 0

Servo servo\_thumb1;

Servo servo\_thumb2;

Servo servo\_index\_finger;

Servo servo\_middle\_finger;

Servo servo\_little\_ring\_fingers;

int a2;

int a3;

int pos = 0;

int treshold = 150;

bool A = false;

bool B = false;

void ServoClosed() {

    servo\_thumb1.write(MIN\_THUMB1);

    servo\_thumb2.write(MIN\_THUMB2);

    servo\_index\_finger.write(MAX\_INDEX\_FINGER);

    servo\_middle\_finger.write(MAX\_MIDDLE\_FINGER);

    servo\_little\_ring\_fingers.write(MIN\_LITTLE\_RING\_FINGER);

}

void ServoOpen() {

    servo\_thumb1.write(MAX\_THUMB1);

    servo\_thumb2.write(MAX\_THUMB2);

    servo\_index\_finger.write(MIN\_INDEX\_FINGER);

    servo\_middle\_finger.write(MIN\_MIDDLE\_FINGER);

    servo\_little\_ring\_fingers.write(MAX\_LITTLE\_RING\_FINGER);

}

void ServoV() {

    servo\_thumb1.write(MIN\_THUMB1);

    servo\_thumb2.write(MIN\_THUMB2);

    servo\_index\_finger.write(MIN\_INDEX\_FINGER);

    servo\_middle\_finger.write(MIN\_MIDDLE\_FINGER);

    servo\_little\_ring\_fingers.write(MIN\_LITTLE\_RING\_FINGER);

}

void ServoOk() {

    servo\_thumb1.write(MIN\_THUMB1);

    servo\_index\_finger.write(MAX\_INDEX\_FINGER);

    servo\_middle\_finger.write(MIN\_MIDDLE\_FINGER);

    servo\_little\_ring\_fingers.write(MAX\_LITTLE\_RING\_FINGER);

}

int calc(int a, int b) {

    if (a >= treshold) {

        A = true;

    } else {

        A = false;

    }

    if (b >= treshold) {

        B = true;

    } else {

        B = false;

    }

    if (A == false && B == false) pos = 0;

    else if (A == true && B == false) pos = 1;

    else if (A == false && B == true) pos = 2;

    else if (A == true && B == true) pos = 3;

    return pos;

}

void setup() {

    Serial.begin(9600);

    servo\_thumb1.attach(10);

    servo\_thumb2.attach(9);

    servo\_middle\_finger.attach(5);

    servo\_little\_ring\_finfers.attach(3);

}

void loop() {

    a2 = analogRead(A2);

    a3 = analogRead(A3);

    int posLoop = calc(a2, a3);

    switch (posLoop) {

        case 0:

            ServoOpen();

        case 1:

            ServoOk();

            break;

        case 2:

            ServoV();

            break;

        case 3:

            ServoClosed();

            break;

    }

## 2.2 Изменение конфигурации приводов

В рамках выполнения работы с макетом бионической руки была поставлена задача разработки программного обеспечения для демонстрации базовых жестов: «ладонь», «кулак», «указательный палец» и жеста «коза» (поднятые указательный палец и мизинец).

При реализации алгоритма управления был выявлен конструктивный недостаток. Безымянный палец и мизинец были кинематически связаны общей приводной системой и леской, что делало невозможным их независимое движение. Вследствие этого жест «коза» не мог быть корректно выполнен, так как при попытке поднять мизинец синхронно поднимался безымянный палец. Целью работы являлось разъединение кинематической связи между безымянным пальцем и мизинцем для обеспечения возможности их независимого управления и корректного выполнения целевого жеста.

Для устранения выявленного недостатка была выполнена следующая последовательность операций:

1. Проведена частичная разборка макета бионической руки для обеспечения доступа к внутренним тяговым механизмам пальцев.
2. Демонтирована оригинальная леска, связывающая привод мизинца с безымянным пальцем.
3. Установлена новая леска, соединяющая безымянный палец с приводом среднего пальца, тем самым создав новую кинематическую пару «средний палец – безымянный палец».
4. Произведена сборка макета, регулировка натяжения лески и проверка работоспособности новой схемы.

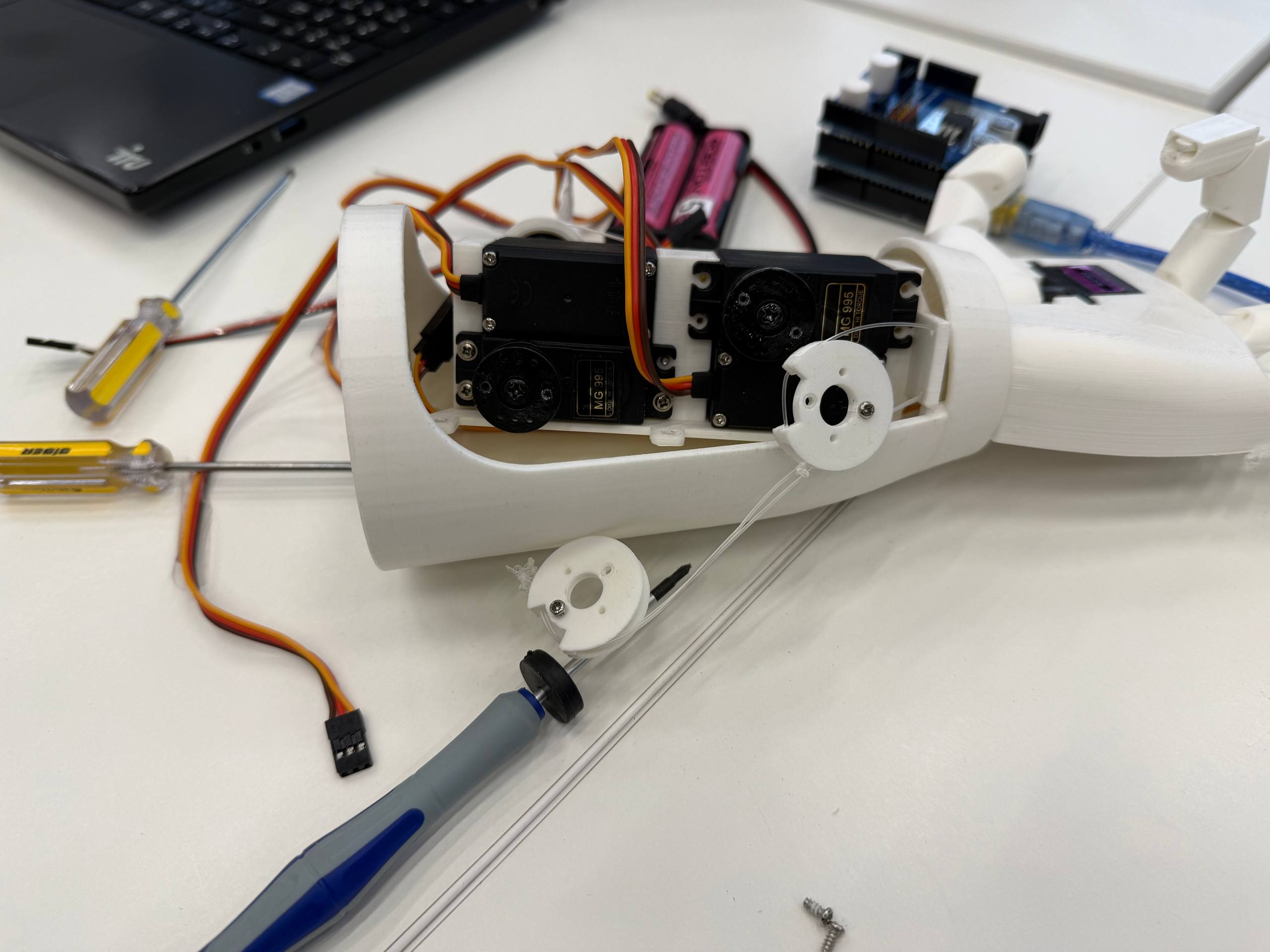


Рисунок 19 – Разбор макета руки

В результате проведенной модификации кинематическая схема привода пальцев была изменена. Безымянный палец был переведен на общую систему управления со средним пальцем, а мизинец получил независимый привод. Это позволило программно реализовать все целевые жесты, включая жест «коза», который стал выполняться за счет одновременного поднятия независимо управляемых указательного пальца и мизинца при сжатых остальных пальцах.

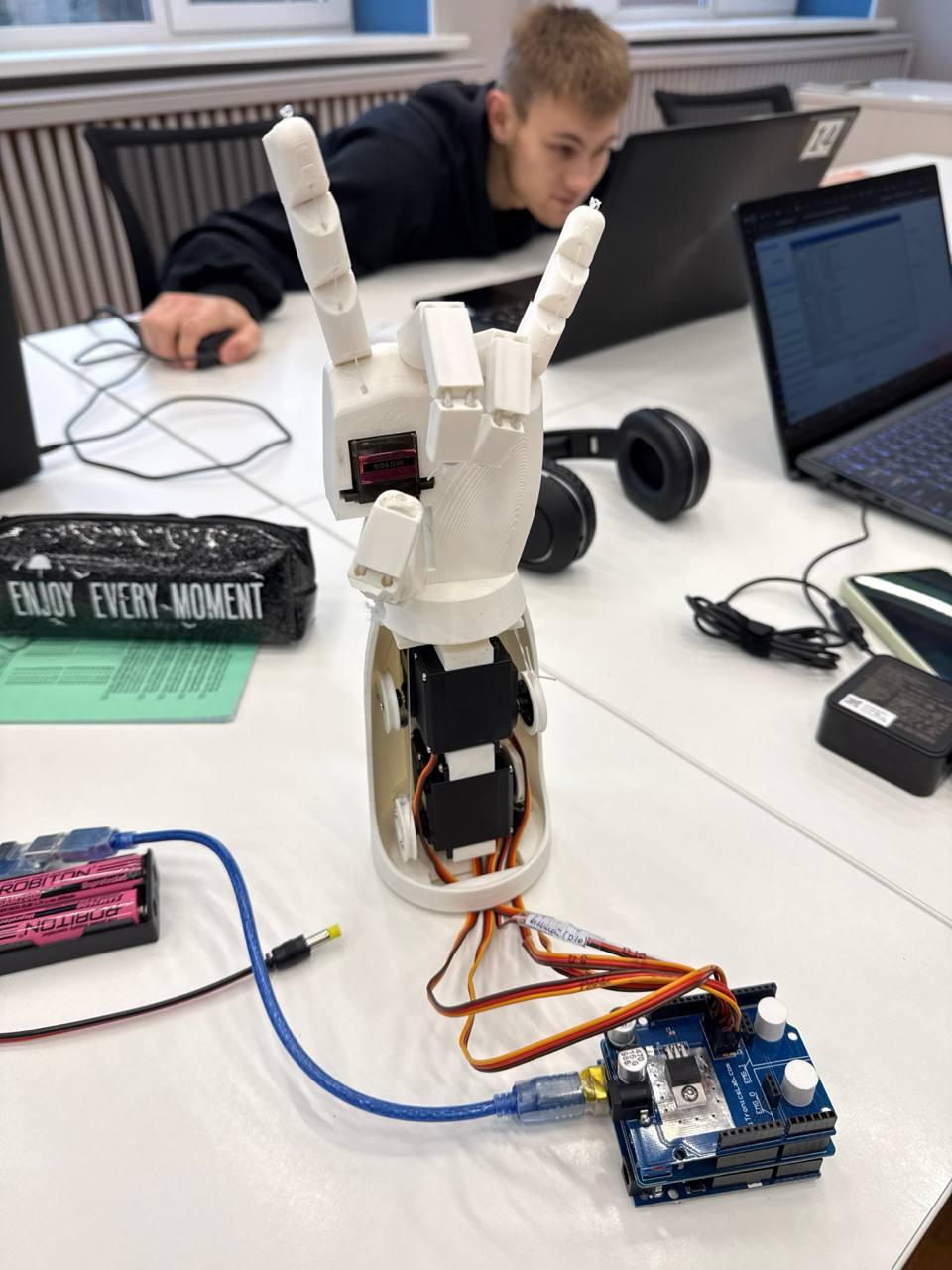


Рисунок 20 – Макет руки после модификации

Код программы для демонстрации базовых жестов.

#include <Servo.h>

#define MAX\_THUMB1 160

#define MIN\_THUMB1 80

#define MAX\_THUMB2 180

#define MIN\_THUMB2 30

#define MAX\_INDEX\_FINGER 180

#define MIN\_INDEX\_FINGER 0

#define MAX\_MIDDLE\_FINGER 180

#define MIN\_MIDDLE\_FINGER 0

#define MIN\_LITTLE\_FINGER 180

#define MAX\_LITTLE\_FINGER 0

// Примечание: названия исправлены для логичности (MAX = закрыт, MIN = открыт)

Servo servo\_thumb1Servo servo\_thumb2;

Servo servo\_index\_finger;

Servo servo\_middle\_finger;

Servo servo\_little\_finger;

void MakeFist()

{

servo\_thumb1.write(MIN\_THUMB1);

servo\_thumb2.write(MIN\_THUMB2);

servo\_index\_finger.write(MAX\_INDEX\_FINGER);

servo\_middle\_finger.write(MAX\_MIDDLE\_FINGER);

servo\_little\_finger.write(MAX\_LITTLE\_FINGER);

}

void OpenPalm()

{

servo\_thumb1.write(MAX\_THUMB1);

servo\_thumb2.write(MAX\_THUMB2);

servo\_index\_finger.write(MIN\_INDEX\_FINGER);

servo\_middle\_finger.write(MIN\_MIDDLE\_FINGER);

servo\_little\_finger.write(MIN\_LITTLE\_FINGER);

}

void ShowIndexAndLittleFinger()

{

servo\_thumb1.write(MIN\_THUMB1);

servo\_thumb2.write(MIN\_THUMB2);

servo\_index\_finger.write(MIN\_INDEX\_FINGER);

servo\_middle\_finger.write(MAX\_MIDDLE\_FINGER);

servo\_little\_finger.write(MIN\_LITTLE\_FINGER);

}

void ShowOnlyIndexFinger()

{

servo\_thumb1.write(MIN\_THUMB1);

servo\_thumb2.write(MIN\_THUMB2);

servo\_index\_finger.write(MIN\_INDEX\_FINGER);

servo\_middle\_finger.write(MAX\_MIDDLE\_FINGER);

servo\_little\_finger.write(MAX\_LITTLE\_FINGER);

}

void ShowFiveFingers()

{

OpenPalm();

}

void setup()

{

Serial.begin(9600);

servo\_thumb1.attach(10);

servo\_thumb2.attach(9);

servo\_index\_finger.attach(6);

servo\_middle\_finger.attach(5);

servo\_little\_finger.attach(3);

Serial.println("Бионическая рука запущена!");

}

void loop()

{

Serial.println("Жест: КУЛАК");

MakeFist();

delay(2000);

Serial.println("Жест: ЛАДОНЬ (5 пальцев)");

OpenPalm();

delay(2000);

Serial.println("Жест: ТОЛЬКО УКАЗАТЕЛЬНЫЙ ПАЛЕЦ");

ShowOnlyIndexFinger();

delay(2000);

Serial.println("Жест: УКАЗАТЕЛЬНЫЙ ПАЛЕЦ И МИЗИНЕЦ (жест 'рога')");

ShowIndexAndLittleFinger();

delay(2000);

Serial.println("===============================");

}

Таблица подключения сервоприводов стала выглядеть следующим образом.

Таблица 1 – Новая таблица подключения сервоприводов к портам платы расширения для сервоприводов

| **Палец** | **Сервопривод** | **Разъем на плате расширения** |
| --- | --- | --- |
| Большой палец | Tower Pro MG90S | D10 |
| Большой палец | Tower Pro MG995 | D9 |
| Указательный палец | Tower Pro MG995 | D6 |
| Средний и безымянный палец | Tower Pro MG995 | D5 |
| Мизинец | Tower Pro MG995 | D3 |

# ЗАКЛЮЧЕНИЕ

В ходе работы были достигнуты цель и задачи практики: проведена работа над изучение программной части «Макет бионической руки», изучена программа Arduino IDE, получены знания о работе с электронными схемами.

Создавала и решала тесты, что позволило мне дать понимание предметной области и развить аналитические способности. Всего был разработано и решено 18 тестов, каждый из которых способствовал закреплению теоретических знаний и практических навыков. Кроме того, в рамках практики я ежедневно выполняла по два практических задания, что способствовало развитию навыков работы с различными инструментами и методами.

Полученные во время работы знания и навыки могут быть применены в инженерной и медицинской области, что может быть использовано в дальнейших исследованиях и разработках для решения актуальных проблем.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Методическое пособие «Человеко–машинное взаимодействие». – BitronicsLab.
2. Руководство пользователя BiTronics Studio.
3. Инструкция пользователя BiTronics Studio версия 4.3.5.8.