Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Санкт-Петербургский государственный университет промышленных технологий и дизайна»

|  |
| --- |
| **Институт информационных технологий и автоматизации** |
| *(Наименование института)* |

|  |  |
| --- | --- |
| Кафедра: | машиноведения |
| Направление подготовки: | 15.03.02 – Технологические машины и оборудование |
| Профиль подготовки: | Информационные технологии в производстве и сервисе технических машин |

|  |  |  |
| --- | --- | --- |
| **ОТЧЕТ** | | |
| о прохождении | учебной | практики |
|  | *(наименование вида практики)* |  |
| тип практики: | практика по получению  умений и навыков | |
|  | *(наименование типа практики)* | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Руководитель  от профильной организации / структурного подразделения СПбГУПТД\*: |  |  | | | | |  |  |
| *(наименование организации)* |  | *(должность, Ф.И.О., печать организации)* | | | | |  | *(подпись, печать)* |
| Руководитель  от СПбГУПТД: |  |  | | | | |  |  |
|  |  | *(должность, ученая степень / звание, Ф.И.О.)* | | | | |  | *(подпись)* |
| Обучающийся: |  | Мадюскин Никита Олегович | | | | |  |  |
|  |  | *(Ф.И.О.)* | | | | |  | *(подпись)* |
| Курс |  | 2 |  | Учебная группа: |  | 2-МДА-14 | |

Санкт-Петербург

2023г

Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный университет промышленных технологий и дизайна»

**Отзыв о практике**

|  |  |
| --- | --- |
| в | ФГБОУ ВО «Санкт-Петербургский государственный университет промышленных технологий и дизайна» |
|  | *(полное наименование профильной организации)* |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Студент (аспирант) | | | | | Мадюскин Никита Олегович | | | | | | |
|  | | | | | *(Ф.И.О.)* | | | | | | |
| Институт | | | Информационных технологий и автоматизации | | | | | | | | |
|  | | | | | *(наименование института)* | | | | | | |
| Курс |  | 2 | | Учебная группа | |  | 2-МДА-14 | | Форма обучения |  | очная |
|  | | | | | | | | | | | |
| Направление подготовки (специальность) | | | | | | | | 15.03.02 – Технологические машины и оборудование | | | |
|  | | | | | | | | *(код и наименование направления (специальности)* | | | |
| Профиль подготовки (специализация) | | | | | | | | ИТ в производстве и сервисе тех. машин | | | |
|  | | | | | | | | *(наименование профиля по учебному плану)* | | | |

|  |  |
| --- | --- |
| проходил (а) | учебно-ознакомительную практику (практика по получению первичных профессиональных умений и навыков) |
|  | *(вид и тип практики)* |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| с « |  | » |  | 2023 года по « |  | » |  | 2023 года |

Инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего распорядка проведен в установленном порядке

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| - компетенции, предусмотренные программой практики | | | |  |
|  | | | | *(указать - сформированы или не сформированы)* |
| - личные и деловые качества | | |  | |
|  | | | | |
|  | | | | |
|  | | | | |
| - качество отчета по практике | | |  | |
|  | | | | |
|  | | | | |
|  | | | | |
| - рекомендации | |  | | |
|  | | | | |
|  | | | | |
|  | | | | |
| - оценка |  | | | |

Руководитель практики от СПбГУПТД \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(должность, Ф.И.О., подпись)*

Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный университет промышленных технологий и дизайна»

**Отзыв о практике**

|  |  |
| --- | --- |
| в |  |
|  | *(полное наименование профильной организации)* |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Студент (аспирант) | | | | | Львов Андрей Сергеевич | | | | | | |
|  | | | | | *(Ф.И.О.)* | | | | | | |
| Институт | | | Информационных технологий и автоматизации | | | | | | | | |
|  | | | | | *(наименование института)* | | | | | | |
| Курс |  | 2 | | Учебная группа | |  | 2-МДА-14 | | Форма обучения |  | очная |
|  | | | | | | | | | | | |
| Направление подготовки (специальность) | | | | | | | | 15.03.02 – Технологические машины и оборудование | | | |
|  | | | | | | | | *(код и наименование направления (специальности)* | | | |
| Профиль подготовки (специализация) | | | | | | | | ИТ в производстве и сервисе тех. машин | | | |
|  | | | | | | | | *(наименование профиля по учебному плану)* | | | |

|  |  |
| --- | --- |
| проходил (а) | учебно-ознакомительную практику (практика по получению первичных профессиональных умений и навыков) |
|  | *(вид и тип практики)* |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| с « |  | » |  | 2023 года по « |  | » |  | 2023 года |

Инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего распорядка проведен в установленном порядке

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| - компетенции, предусмотренные программой практики | | | |  |
|  | | | | *(указать - сформированы или не сформированы)* |
| - личные и деловые качества | | |  | |
|  | | | | |
|  | | | | |
| - качество отчета по практике | | |  | |
|  | | | | |
|  | | | | |
| - рекомендации | |  | | |
|  | | | | |
|  | | | | |
| - оценка |  | | | |

|  |  |
| --- | --- |
| Руководитель практики от Организации |  |
|  | *(реквизиты приказа по организации о назначении руководителя практики)* |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| *(должность,* | *подпись,* | *Ф.И.О)* |

М.П.

Оглавление

[1 Обзор литературных источников 5](#_Toc136438514)

[1.1 Общие сведения об исполнительных механизмах технологических машин 5](#_Toc136438515)

[1.2 Примеры рычажных механизмов, используемых в швейном оборудовании 7](#_Toc136438516)

[1.3 Примеры кулачковых механизмов, используемых в машинах текстильной промышленности 10](#_Toc136438517)

[1.4 Описание лабораторного стенда 12](#_Toc136438518)

[2 Задачи на практику 16](#_Toc136438519)

[2.1 Придумать как считывать ускорение с лабораторного стенда 16](#_Toc136438520)

[2.2 Вывести график ускорения на экран компьютера: 16](#_Toc136438521)

[3 Решение поставленных задач 18](#_Toc136438522)

[3.1 Выбор микроконтроллера 18](#_Toc136438523)

[3.2 Среда разработки 19](#_Toc136438524)

[3.3 Акселерометр MPU6050 20](#_Toc136438525)

[3.4 Как подключить 21](#_Toc136438526)

[3.5 Описание Кода 22](#_Toc136438527)

[3.6 Немного о графическом интерфейсе и как его связать с Arduino 27](#_Toc136438528)

[3.6 Выводы и дальнейшие планы исследования 30](#_Toc136438529)

[4 Список использованных источников 32](#_Toc136438530)

# 1 Обзор литературных источников

## 1.1 Общие сведения об исполнительных механизмах технологических машин

В технологических машинах, для привода рабочих органов, в настоящее время используется достаточно большое число исполнительных механизмов. В большинстве случаев, например в машинах текстильной и легкой промышленности, используются рычажные, кулачковые и кулачково-рычажные механизмы.

Как правило, применяемые в текстильной и легкой промышленности механизмы относятся к классу механизмов циклового действия [1]. На рисунке 1.1 показаны наиболее распространенные разновидности простейших цикловых механизмов: рычажные (рисунок 1.1, а-в), кулачковые (рисунок 1.1, г), механизмы с некруглыми колесами (рисунок 1.1, д), шаговые, среди которых мальтийские (рисунок 1.1, е), храповые (рисунок 1.1, ж) и червячные (рисунок 1.1, з).

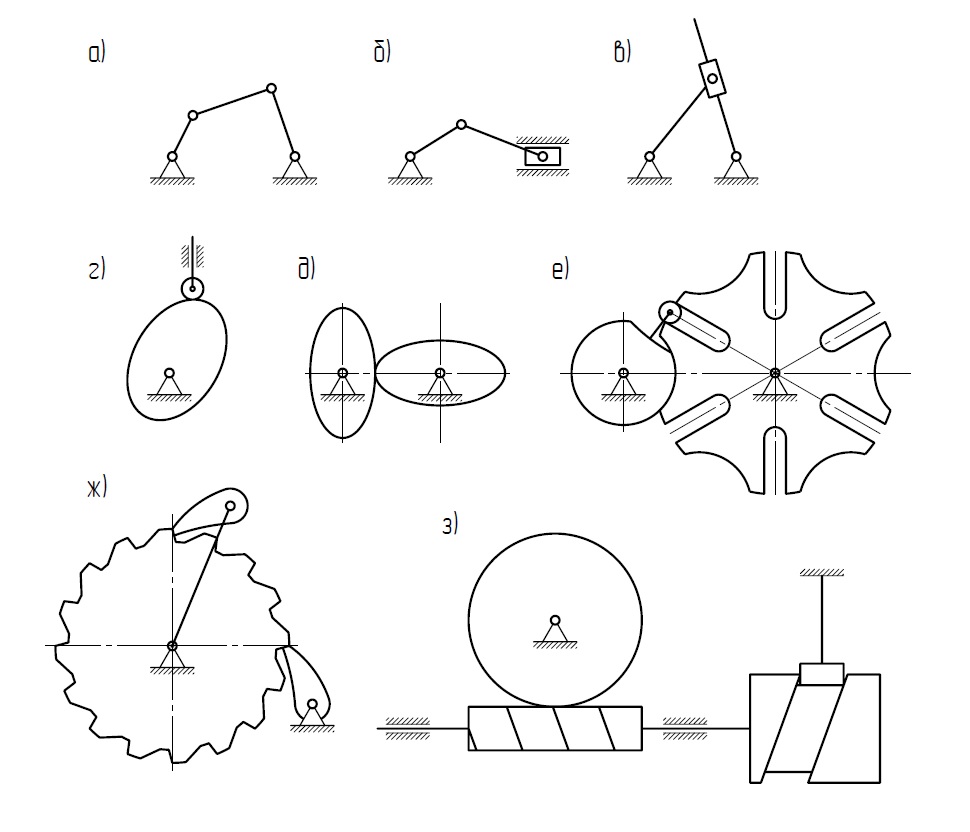


Рисунок 1.1 – Разновидности цикловых механизмов

Возможны различные сочетания указанных простейших механизмов, например рычажно-шаговый, кулачково-шаговый, кулачково-рычажный и прочие. Так же простейшие механизмы могут быть усложнены методом наслоения групп Ассура. Возможно создание циклового механизма шагового типа на базе механизма с двумя степенями подвижности, осуществляющего сложение возвратно-поступательного или колебательного движения с равномерным вращательным движением. Пример подобного механизма, который объединяет свойства червячной передачи и кулачкового механизма показан на рисунке 1.1, з. В данном случае угловые перемещения червячного колеса, вызванные вращением червяка, суммируются с осевым перемещением червяка, управляемым кулачковым механизмом.

Все цикловые механизмы условно делятся на реверсивные и нереверсивные. Если среднее значение первой геометрической передаточной функции ведомого звена равно нулю, то мы имеем возвратно-поступательное или колебательное движение звеньев около неподвижной оси (рисунок 1.1, а-г). При движении ведомого звена с отличной от нуля средней скоростью происходит смещение ведомого звена на один шаг (рисунок 1.1, д-з).

Различают также цикловые механизмы прерывного и непрерывного движения, в которых реализуются функции положения звеньев с выстоями и без выстоев. В многозвенных рычажных механизмах можно получить квазипрерывное движение с приближенным выстоем ведомого звена.

По функциональному назначению цикловые механизмы могут быть исполнительными, передаточными и управляющими, а также осуществлять контроль, питание, транспортировку, сортировку, регулирование и т.д. Каждый из этих механизмов может играть ответственную роль в машине и подвергаться значительным динамическим нагрузкам. Иногда к механизму предъявляются особые требования к уровню допускаемых динамических искажений законов движения, динамических нагрузок и т.п., что должно быть учтено при синтезе механизма. При нелинейной функции положения динамические условия оказываются более напряженные, так как выходные звенья перемещаются с переменной скоростью. Это часто приводит к возникновению значительных инерционных нагрузок.

## 1.2 Примеры рычажных механизмов, используемых в швейном оборудовании

Рассмотрим более подробно конструкции и принцип работы некоторых рычажных механизмов, применяемых в швейных машинах.

Механизмом игловодителя называется механизм, приводящий в движение деталь, к которой крепится игла швейной машины. В зависимости от назначения швейной машины применяются разнообразные механизмы игловодителей, с целью обеспечить нужное перемещение игле. Большинство машин имеет прямолинейно перемещающуюся по вертикали иглу, но бывают машины с криволинейным, горизонтальным или наклонным движением иглы. Прямолинейное перемещение иглы в вертикальной плоскости применяется в швейных машинах, у которых вал игловодителя расположен в рукаве машины. В этом случае требованием к механизму игловодителя является преобразование вращательного движения главного вала в прямолинейное возвратно-поступательное перемещение игловодителя. Этого можно достигнуть применением кривошипно-шатунного (рисунок 1.2), кулисного, коромыслово-шатунного плоского и пространственного механизмов. Кривошипно-шатунный механизма *OAB* может быть центральным и нецентральным. Нецентральным механизм будет в том случае, когда линия движения игловодителя не проходит через центр *O* вращения кривошипа *OA*. Величину *e* смещения центра *O* относительно линии движения игловодителя называют эксцентриситетом.

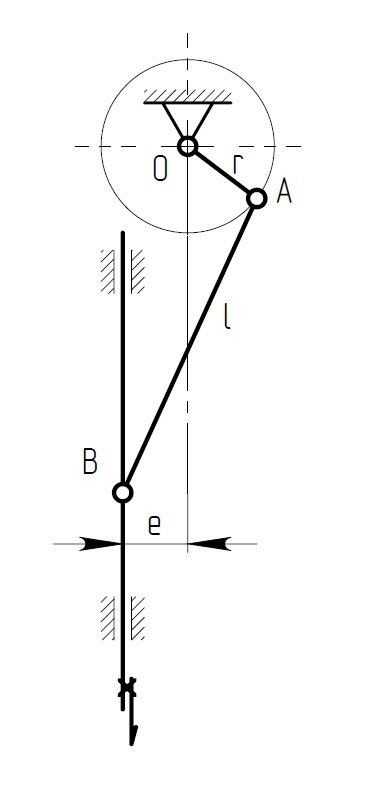


Рисунок 1.2 – Нецентральный кривошипно-шатунный механизм игловодителя

Нитепритягиватель – рабочий орган в челночных швейных машинах, совершающий подачу верхней нити к игле и челноку. Механизм, приводящий его в движение, называют механизмом нитепритягивателя. Данный механизм выполняет подачу верхней нити к игле и челноку, удаляет петлю верхней нити из челночного комплекта, производит утягивание образовавшегося стежка, сматывает очередную порцию нити с катушки. В результате работы нитепритягивателя верхняя и нижняя нити челночного стежка переплетаются в середине сшиваемых материалов. Так как переплетение происходит с силовым утягиванием верхней нити, то после образования стежка материалы плотно прижимаются друг к другу.

Различают несколько основных конструктивных вариантов механизмов нитепритягивателей:

1. Нитепритягиватель выполнен как одно целое с игловодителем, так как глазком *K* для нити служит отверстие, сделанное в самом игловодителе.
2. Нитепритягиватель выполнен в виде качающегося рычага с глазком *K* для нити на конце, получающего движение от цилиндрического кулачка.
3. Нитепритягиватель выполнен в виде качающейся кулисы шарнирно-стержневого механизма, имеющей глазок *K* для нити (рисунок 1.3).

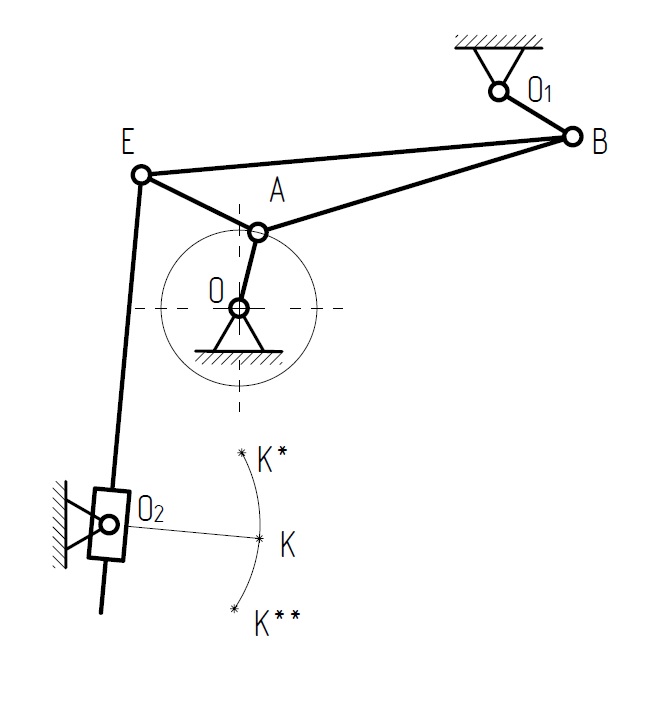


Рисунок 1.3 – Кулисный шарнирно-стержневой механизм нитепритягивателя

Также распространены рычажные механизмы подачи нитки. Механизм подачи нитки освобождает нитку от натяжения в период взаимодействия ее с иглой и челноком. После обведения вокруг челнока механизм снимает нитку с челнока, вытягивает ее из-под материала, производит затяжку стежка и сматывает с катушки нитку, необходимую на новый стежок.

Механизмы подачи нитки должны обеспечивать: подачу и освобождения нитки в период взаимодействия ее с иглой и челноком; своевременное снятие петли игольной нитки с челнока и вытягивание излишка нитки из-под материала; качественную затяжку стежка; сматывание с катушки количества нитки, необходимого для образования стежка; согласование работы механизма подачи нитки с другими механизмами; минимальные нагрузки на нитку и элементы механизма.

Исполнительным инструментом механизма подачи нитки является рычаг-нитепередатчик (нитепритягиватель), имеющий глазок, в который заправляется нитка.

Различают следующие варианты исполнения механизма подачи нити: кулачковый; кривошипный (шатунный, кулисный) и ротационный.

Кривошипно-шатунный механизм (рисунок 1.4) имеет шарнирно-стержневую структуру и может применяться при высоких скоростных режимах, с которыми работают современные машины челночного стежка.

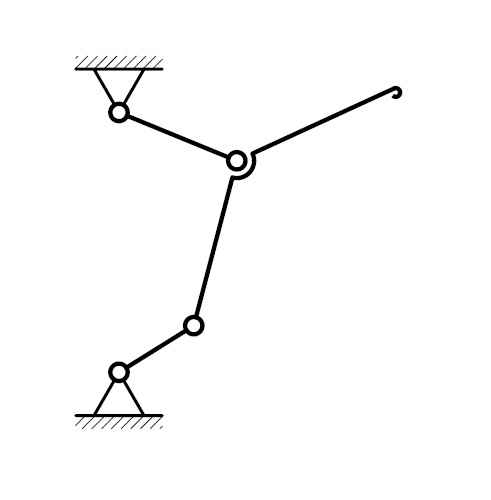


Рисунок 1.4 – Кривошипно-шатунный механизм подачи нитки

## 1.3 Примеры кулачковых механизмов, используемых в машинах текстильной промышленности

Зевообразовательный механизм в ткацком станке открывает зев, чтобы проложить уточную нить и чередует движение ремизок или групп основных нитей так, чтобы получить заданный узор ткани. Различают три типа зевообразовательных механизмов: кулачковые (см. рисунок 1.5), используемые для получения простейших рисунков; ремизоподъемные каретки; жаккардовые машины.

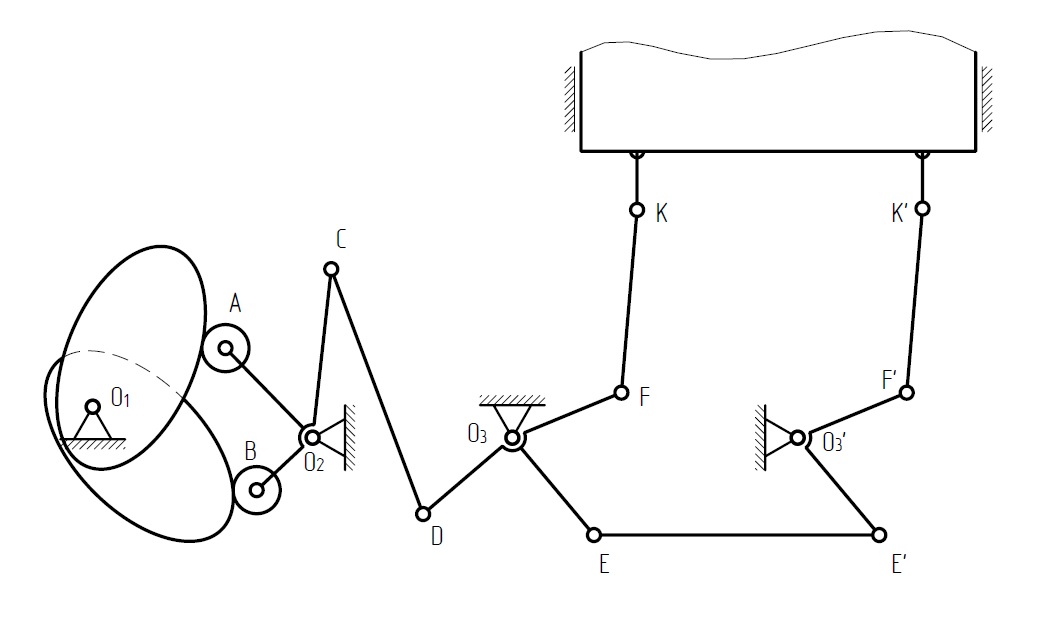


Рисунок 1.5 – Кулачковый зевообразовательный механизм станка АТПР

Перемещение уточной нити вдоль основы, т.е. присоединение ее к предыдущей уточной нити, называется прибоем утка к опушке ткани. В зависимости от способа перемещения и прибоя уточной нити применяются различные механизмы и приспособления. В ткацком производстве наиболее распространен прибой уточной нити бердом, которое укреплено на батане. Батан перемещает уточную нить вдоль нитей основы и прибивает ее к опушке ткани и определяет ширину заправки ткани и плотность ее по основе благодаря берду. Наиболее распространены кривошипно-ползунные и кулачковые (рисунок 1.6) батанные механизмы.

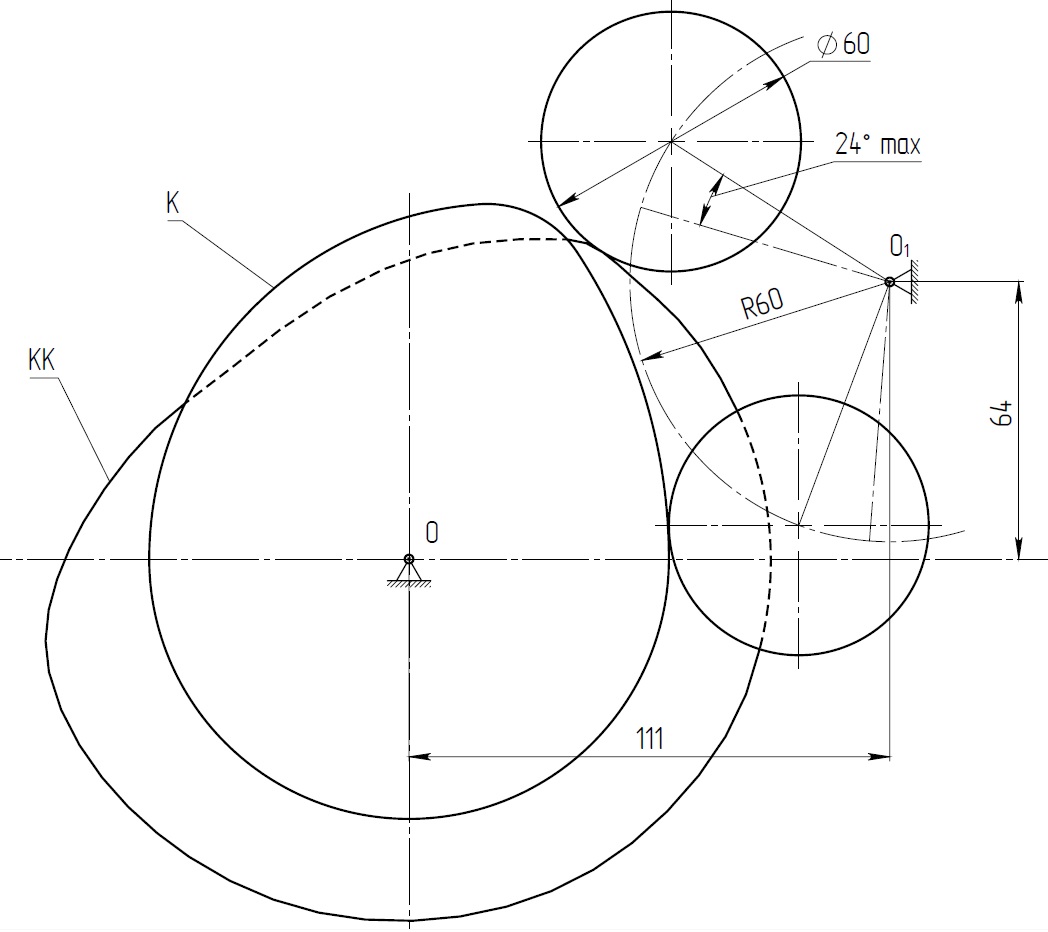


Рисунок 1.6 – Батанный механизм станка СТБ

## 1.4 Описание лабораторного стенда

Как следует из приведенного выше обзора, исполнительные механизмы, как правило, представлены в виде механизмов циклового действия, отличительной особенностью которых является переменность передаточной функции. Из [1] следует, что нелинейность передаточной функции может приводить к возникновению нежелательных колебаний рабочего органа. Наличие упругих и диссипативных элементов в конструкции механизма, а также зазоров в кинематических парах, могут приводить к искажению требуемых движений рабочего органа. Для изучения динамических эффектов, возникающих в цикловых рычажных механизмах, на кафедре машиноведения, при участии профессора Вульфсона И.И., разработан лабораторный стенд, содержащий клиноременную передачу, рычажный механизм и сердечник, размещенный между упругими элементами, и имеющий возможность совершать возвратно-поступательное движение под действием поступательно движущегося ползуна. Внешний вид лабораторного стенда приведен на рисунке 1.7. На рисунке 1.8 представлена структурная схема стенда.



Рисунок 1.7 – Лабораторный стенд (общий вид)

Изображение выглядит как зарисовка, рисунок, диаграмма, Штриховая графика

Автоматически созданное описание

1- Электродвигатель; 2- Нижний шкив клиноременной передачи; 3- Ремень; 4- Верхний шкив клиноременной передачи; 5- Вал; 6- Кривошип; 7- Шатун; 8- Толкатель; 9- Пружина; 10- Сердечник; 11- Датчик; 12- Пружина; 13- Корпус; 14- Датчик

Рисунок 1.8- Структурная схема лабораторного стенда

Предварительное усилие поджатия пружин поз. 9 и 12 может регулироваться с помощью специального узла. С помощью лабораторного стенда имеется возможность исследовать вынужденные колебания сердечника поз. 10, размещенного между пружинами поз. 9 и 12. Датчики поз. 11 и 14. позволяют получать осциллограммы виброускорений. Датчик поз. 14 позволяет получить осциллограмму как бы «идеального» закона движения толкателя, а датчик поз. 11 позволяет оценить искажения, накладываемые на закон движения толкателя, вызванные его упругостью.

Путем регулировки усилия предварительного поджатия пружины поз. 12, можно получить режим нелинейных колебаний сердечника поз. 10 при наличии отрывов пружин поз. 9 и 12 от сердечника поз. 10.

В случае замены кривошипа поз. 6 на кулачок (см. рисунок 1.9) и установке ролика на толкателе поз. 8, лабораторный стенд позволяет исследовать колебания в кулачковом механизме с упругим толкателем, при этом роль упругого толкателя выполняет пружина поз. 9, а пружина поз. 12 представляет собой упругий элемент, соответствующий силовому замыканию.



Рисунок 1.9 – Кулачки (общий вид)

В качестве электродвигателя используется трехфазный асинхронный двигатель АИР71В4. Технические характеристики электродвигателя приведены в таблице 1.1. Скорость вращения вала электродвигателя может плавно изменяться с помощью тиристорного преобразователя частоты (на рисунке 1.8 не показан).

Таблица 1.1 – Технические характеристики электродвигателя АИР71В4

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Электродвигатель | Мощность, кВт | Номинальная частота вращения, мин-1 | Номинальный ток при 220/380В, А | Номинальный крутящий момент, Нм | КПД, % | Iпуск/Iном | Мпуск/Мном | Ммакс/Мном | Ммин/Мном | Масса, кг |
| АИР71В4 | 0,75 | 1350 | 3,7/2,2 | 5,31 | 72,1 | 5,0 | 2,5 | 2,6 | 2,4 | 9,4 |

# 2 Задачи на практику

## 2.1 Придумать как считывать ускорение с лабораторного стенда

1) Выбрать микроконтроллер: Исследуйте различные микроконтроллеры, которые могут быть использованы для считывания данных с акселерометра и передачи их на компьютер. Учтите требования вашего проекта и подберите микроконтроллер, который обеспечит необходимую функциональность.

2) Выбрать среду разработки: Выберите среду разработки, которая будет совместима с выбранным микроконтроллером. Некоторые популярные среды разработки для микроконтроллеров включают Arduino IDE, PlatformIO, MPLAB X и Keil uVision.

3) Выбрать акселерометр: Исследуйте доступные акселерометры и выберите подходящий для вашего проекта. Обратите внимание на диапазон измерений, разрешение и интерфейс связи акселерометра с микроконтроллером (например, I2C или SPI).

4) Написать оптимальный код для считывания ускорения: Напишите программный код для микроконтроллера, который будет считывать данные с акселерометра и передавать их на компьютер через выбранный интерфейс связи. Используйте документацию акселерометра и микроконтроллера, а также примеры кода, чтобы помочь вам реализовать эту функциональность.

## 2.2 Вывести график ускорения на экран компьютера:

После успешного считывания данных ускорения с помощью микроконтроллера и передачи их на компьютер, вы можете визуализировать график ускорения на экране компьютера.

1) Подключите микроконтроллер к компьютеру: Используйте соответствующий интерфейс связи (например, USB) для подключения микроконтроллера к компьютеру.

2) Настройте программное обеспечение на компьютере: Используйте выбранную вами среду разработки или другое программное обеспечение для приема данных ускорения с микроконтроллера через соответствующий интерфейс связи.

3) Разработайте программу для вывода графика или найдите готовое решение в интернете.

2.4) Передача данных с микроконтроллера на компьютер: Внедрите Графический интерфейс в вашу разработку.

# 3 Решение поставленных задач

## 3.1 Выбор микроконтроллера

Arduino Nano - это компактная и универсальная плата на основе микроконтроллера ATmega328P. Вот несколько причин, почему Arduino Nano может быть хорошим выбором для данного проекта считывания ускорения с Лабораторного стенда:

1. Размер и компактность: Arduino Nano имеет небольшой размер и компактный форм-фактор, что делает его удобным для интеграции в различные устройства и стенды. Это позволяет сэкономить место и облегчает монтаж на Лабораторном стенде.
2. Широкая поддержка и сообщество: Arduino Nano является популярной платформой, что означает, что существует большое количество ресурсов, библиотек и примеров кода, доступных в сети. Это может быть полезно при разработке и отладке вашей программы для считывания ускорения.
3. Обширные возможности ввода-вывода: Arduino Nano обладает достаточным количеством пинов ввода-вывода для подключения дополнительных устройств, таких как датчики ускорения или интерфейсы связи. Вы можете использовать эти пины для подключения акселерометра и передачи данных о ускорении на компьютер.
4. Простота использования: Arduino Nano предоставляет простой и дружественный интерфейс для программирования и отладки. Среда разработки Arduino IDE предлагает интуитивно понятный инструментарий для написания кода, загрузки его на плату и мониторинга вывода. Это может быть полезно, особенно если у вас уже есть опыт работы с Arduino-платформой.

## 3.2 Среда разработки

Arduino IDE является официальной интегрированной средой разработки (IDE) для платформы Arduino и хорошим выбором для программирования Arduino Nano. Вот несколько причин, почему Arduino IDE может быть хорошим выбором :

1. Простота использования: Arduino IDE предлагает простой и интуитивно понятный интерфейс, который делает программирование Arduino доступным даже для новичков. Он предоставляет простые инструменты для написания кода, загрузки его на плату Arduino Nano и отладки программы.
2. Широкая поддержка: Arduino IDE имеет огромное сообщество разработчиков и пользователей Arduino, что означает, что вы можете найти множество ресурсов, библиотек и примеров кода для поддержки вашего проекта. Это упрощает разработку и расширение функциональности вашего проекта считывания ускорения.
3. Поддержка различных платформ: Arduino IDE поддерживает не только Arduino Nano, но и другие платы Arduino. Это означает, что если в будущем вы захотите использовать другую плату Arduino для своих проектов, вы сможете легко перенести свой код и настройки на новую платформу.
4. Большое количество примеров и документации: Arduino IDE предоставляет множество примеров и документации, которые помогут вам начать программирование Arduino Nano и реализацию считывания ускорения. Вы можете найти примеры кода для работы с акселерометром и другими датчиками ускорения, а также с графическим выводом данных.

Выбор Arduino IDE позволит быстро приступить к программированию Arduino Nano и разработке проекта.

## 3.3 Акселерометр MPU6050

Акселерометр MPU6050 - это широко используемый модуль, комбинирующий в себе трехосевой акселерометр и гироскоп. Вот несколько причин, почему MPU6050 хороший выбор :

1. Интеграция акселерометра и гироскопа: MPU6050 позволяет одновременно измерять ускорение и угловую скорость. Это полезно, если вам нужно получить более полную информацию о движении объекта на Лабораторном стенде.
2. Высокая точность и разрешение: MPU6050 обеспечивает высокую точность и разрешение при измерении ускорения. Он может измерять ускорения в диапазоне от ±2g до ±16g с разрешением до 16 бит, что позволяет получить точные данные для вашего проекта.
3. Широкая поддержка и доступность: MPU6050 является популярным модулем и имеет обширную поддержку в сообществе Arduino. Вы найдете множество примеров кода, библиотек и руководств, которые помогут вам в подключении и использовании MPU6050 с Arduino Nano.
4. Удобный интерфейс связи: MPU6050 может быть подключен к микроконтроллеру с помощью интерфейса I2C, который требует всего двух проводов для передачи данных. Это упрощает подключение модуля и считывание данных ускорения с Arduino Nano.

Выбор MPU6050 позволит получить надежный и точный модуль акселерометра для проекта.

## 3.4 Как подключить

Подробная инструкция по подключению:

1. Получите Arduino Nano и необходимые компоненты: Вам понадобится Arduino Nano, USB-кабель для подключения к компьютеру, а также провода для подключения акселерометра MPU6050.
2. Подключите USB-кабель: Подключите один конец USB-кабеля к разъему USB на Arduino Nano, а другой конец к свободному USB-порту вашего компьютера. Это позволит питать микроконтроллер и обеспечит связь с Arduino IDE.
3. Подготовьте акселерометр MPU6050: У вас должен быть модуль акселерометра MPU6050. Убедитесь, что он имеет разъемы VCC, GND, SDA и SCL, которые можно подключить с помощью проводов.
4. Подключите акселерометр MPU6050 к Arduino Nano: Соедините пин VCC акселерометра с 5V-пином Arduino Nano, GND-пин с GND-пином, SDA-пин с пином A4 (SDA) и SCL-пин с пином A5 (SCL) на Arduino Nano. Убедитесь, что подключение происходит правильно и надежно.
5. Подключите Arduino Nano к Arduino IDE: Запустите Arduino IDE на вашем компьютере. В меню "Инструменты" выберите правильную плату (Arduino Nano) и порт, к которому подключен Arduino Nano. Это позволит установить связь между компьютером и микроконтроллером.

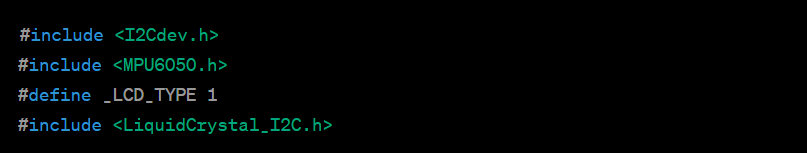
## 3.5 Описание Кода

Прежде чем приступить к написанию кода, я убедился, что акселерометр MPU6050 правильно подключен к Arduino Nano. Я подключил пины VCC и GND акселерометра к соответствующим пинам Arduino Nano для питания, а пины SDA и SCL акселерометра подключил к пинам SDA и SCL Arduino Nano для связи по интерфейсу I2C.

Для работы с акселерометром MPU6050 и считывания данных ускорения, я использовал библиотеки:

1. MPU6050, которая предоставляет удобный интерфейс для взаимодействия с модулем. Эта библиотека позволяет легко получить данные ускорения по осям X, Y и Z.
2. I2Cdev, которая предоставляет методы, упрощающие взаимодействие с устройствами, подключенными к шине I2C.
3. LiquidCrystal\_I2C - библиотека для работы с LCD-дисплеем по протоколу I2C.
4. Math – стандартная библиотека для произведения математический операций

Теперь давайте перейдем к описанию оптимального кода для считывания ускорения с акселерометра MPU6050 с использованием Arduino IDE и Arduino Nano.



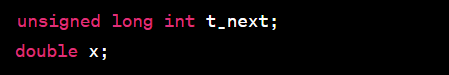
Эти строки представляют собой подключение необходимых библиотек для работы с датчиком MPU6050 и LCD-дисплеем. Библиотеки были описаны ранее



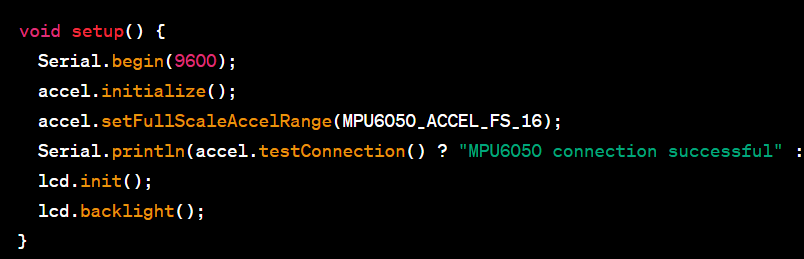
Эта строка кода инициализирует объект lcd для работы с LCD-дисплеем. Параметры LiquidCrystal\_I2C указывают адрес дисплея (в данном случае 0x27), количество символов в строке (16) и количество строк (2).

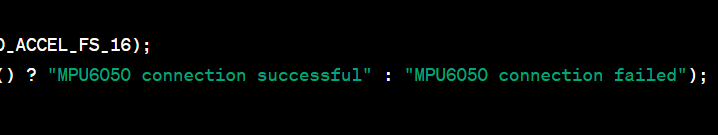


Эта строка кода создает объект accel для работы с датчиком MPU6050.

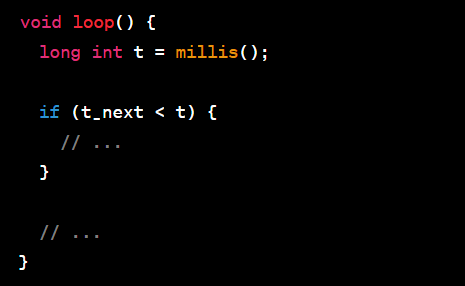


Здесь объявляются переменные t\_next и x. t\_next используется для отслеживания времени следующего измерения, а x будет содержать значение ускорения.



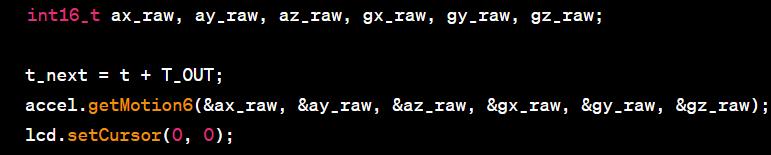


Эта функция setup() вызывается один раз при запуске программы. В ней инициализируются серийный порт для вывода отладочной информации на скорости 9600 бит/с (Serial.begin(9600)). Затем датчик MPU6050 инициализируется (accel.initialize()) и устанавливается полный диапазон измерения ускорения на ±16g (accel.setFullScaleAccelRange(MPU6050\_ACCEL\_FS\_16)). Затем происходит проверка связи с датчиком и вывод соответствующего сообщения в серийный порт. Далее инициализируется LCD-дисплей (lcd.init()) и включается подсветка (lcd.backlight()).



В функции loop() сначала определяется переменная t, которой присваивается текущее время в миллисекундах (millis()).

Затем происходит проверка условия if (t\_next < t). Если текущее время t превышает значение переменной t\_next, то выполняется блок кода внутри условия (блок внутри). Это позволяет контролировать частоту выполнения измерений ускорения.

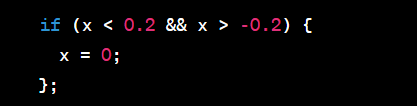


Внутри условия объявляются переменные ax\_raw, ay\_raw, az\_raw, gx\_raw, gy\_raw, gz\_raw, которые будут использоваться для хранения сырых значений ускорения и гироскопа, считанных с датчика MPU6050. Затем переменной t\_next присваивается новое значение, равное текущему времени t плюс задержка T\_OUT. Это гарантирует, что следующее измерение произойдет через определенный интервал времени.

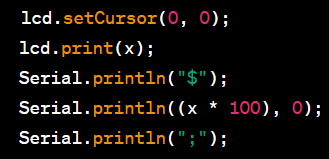


Далее происходит обработка значения ускорения по оси X. Значение сырого ускорения ax\_raw делится на 2048 для получения значения ускорения в g (гравитационных единицах)

(блок внутри)



Этот блок кода отвечает за уничтожение незначительной погрешности значения ускорения по оси X. Здесь происходит проверка: если значение x находится в диапазоне от -0.2 до 0.2, то оно считается незначительным и устанавливается в ноль. Это позволяет избавиться от малых погрешностей и шумов в измерениях, которые могут возникать на датчике.



После обработки значения ускорения x происходит вывод его на LCD-дисплей и в серийный порт.

1. ‘lcd.setCursor(0, 0)’ - устанавливает позицию курсора на LCD-дисплее в начало первой строки.
2. ‘lcd.print(x)’ - выводит значение ускорения x на LCD-дисплей.
3. Serial.println("$") - выводит символ $ в серийный порт для обозначения начала строки. Позже в разделе про графический интерфейс расскажу зачем это нужно.
4. ‘Serial.println((x \* 100), 0)’ - выводит значение ускорения ‘x’, умноженное на 100, в серийный порт. ‘0’ второй аргумент ‘Serial.println()’, и он указывает на вывод без десятичных знаков, чтобы окончательно убрать погрешность, которая скорее всего уже отсутствует.
5. ‘Serial.println(";")’ - выводит символ “ ; ” в серийный порт для обозначения конца строки. Об этом тоже подробнее в разделе графический интерфейс.

Эти операции позволяют отображать значение ускорения на LCD-дисплее и одновременно отправлять его в серийный порт для дальнейшего анализа или обработки внешним устройством.

## 3.6 Немного о графическом интерфейсе и как его связать с Arduino

Я использовал готовое решение SerialPortPlotter.

Перед тем как начать работать с графиком, сначала объясню для чего мы выводили “$” в “Serial.print” и все остальное.

1. Знак “$” является командой SerialPortPlotter для начала считывания значений.
2. Далее выводим нужные нам значения, которые мы хотим видеть на графике

Прошу обратить ваше внимание на то, что значения не могут быть дробными. Именно для этого мы и умножаем значение на 100(а если быть точнее, то мы используем 10n, где “n” – это степень приближения. Все зависит от того насколько точные нам нужны значения).



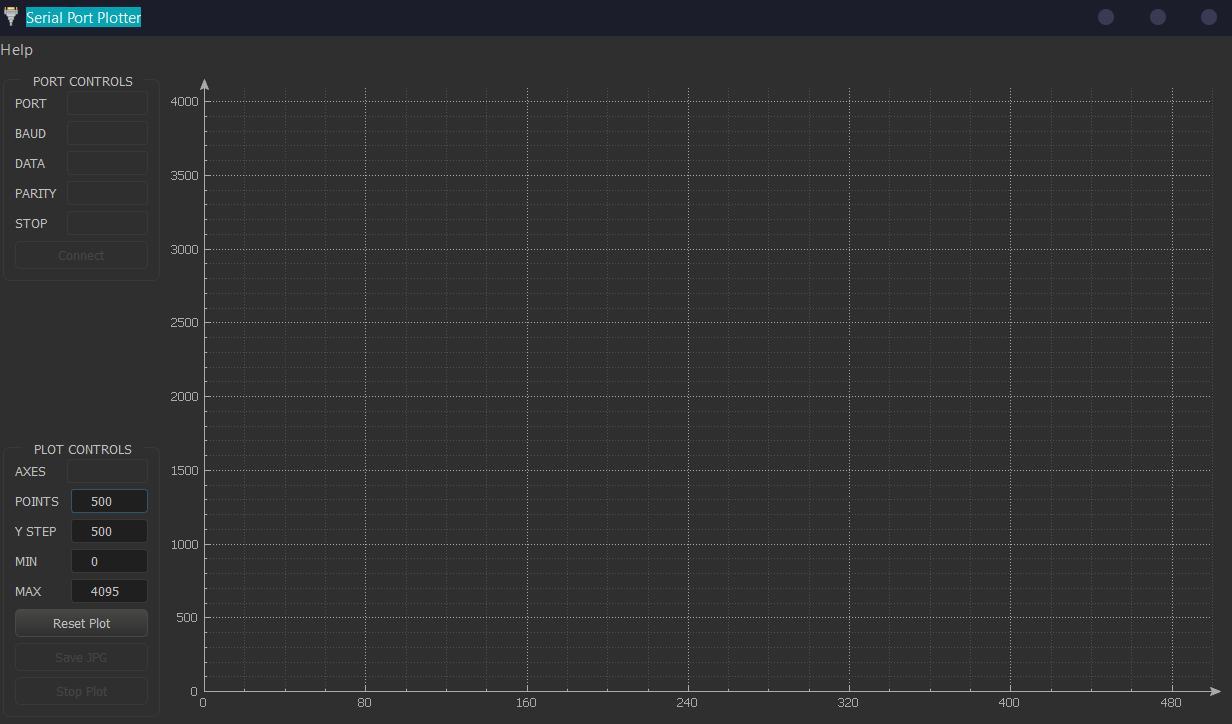
1. Далее мы отделяем значение символом

Еще обратите внимание на то что мы используем метод ”println”

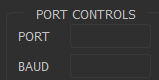
Он выводит значение в отладчик и переводит курсор на следующую строку.

Перейдем же наконец к данной утилите:

Вот интерфейс данной программы



В интерфейсе сверху слева мы видим Port и Baud



В первом мы выбираем порт, куда подключена Arduino Nano, а во второй вводим 9600 бит / c(или другое значение, которое мы указали в коде).

Далее просто тыкаем на кнопку



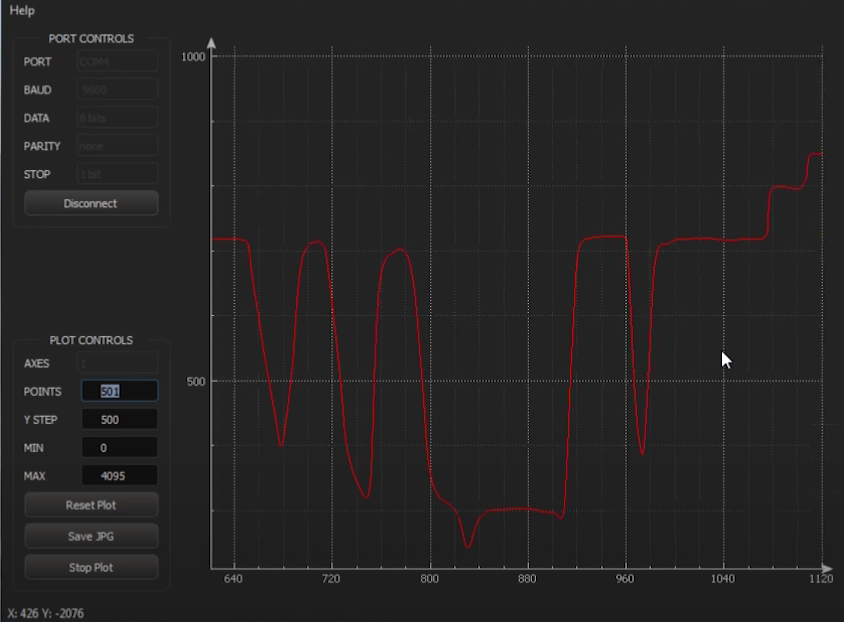
Важно!!!

Если по какой-то причине после всех вышеперечисленных действий у вас не Появился график, то нужно проверить значение

Оно должно совпадать со значением Baud.

Далее после появления графика мы можем добавить аннотацию с подписью, что это значения G умноженные на 100.

В целом для учебного проекта это необязательно.



Что-то подобное мы увидим на экране.

## 3.6 Выводы и дальнейшие планы исследования

В рамках практики я научился многим фичам и изучил как минимум 4 библиотеки Arduino. Но хотелось бы довести это до конца возможно уже в рамках следующей практики.)))

Кратко о том, что бы я хотел добавить:

1. Графический интерфейс SerialPortPlotter хоть решил мою проблему, но хотелось бы самому написать интерфейс и добавить туда функционал, которого нет тут.
2. Поменять датчик MPU6050 на какой-нибудь аналоговый, который хоть и стоит дороже, но будет более точен в своих измерениях.
3. Подсоединить к Arduino Nano второй датчик, который будет мерить угол оборота кулачка.
4. И последнее. Убрать зависимость от ПК.

<https://github.com/NickMad17/Practic.git> - репозиторий с проектом на github.

# Список использованных источников

1. <https://www.arduino.cc/en/software/> - Официальный сайт Arduino.
2. <https://3d-diy.ru/wiki/arduino-platy/arduino-nano/> - Подробное описание Arduino Nano.
3. <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-EV-Board1.pdf> - документация MPU-6000.
4. <https://docs.arduino.cc/learn/electronics/lcd-displays> - документация Liquid Crystal Display.