

ProtoWorks

Руководство

V0.1

2024

Введение

Программа находится в альфа версии, имеет ошибки и недочёты. *От автора: прошу отнестись с пониманием.*

Первая и основная цель программы: создать условия для более удобной работы с оборудованием 3Д печати. Начиная от возможности работы с удалённой печатью, заканчивая мониторингом процесса и ведением таблиц о количестве тех или иных производимых деталей.

Так же предполагается, что программа берёт на себя задачу синхронизации файлов проектов внутри компании. Кроме этого уже реализованы некоторые методы автоматической конвертации файлов из одного внешнего ПО в другое.

Перейдём к **важным понятиям**, используемых в ProtoWorks(далее ProtoWorks подразумевает под собой весь комплекс ProtoWorks и CNCHell).

ProtoWorks – основная программа, ведущая учёт деталей. Так же используется для синхронизации проектов. Имеет дополнительные функции в виде удалённой печати на бумажном принтере и тд.

CNCHell – программа, позволяющая отправлять детали из ProtoWorks на станки. Программа организует очереди файлов на каждый станок.

Деталь ProtoWorks. Также может именоваться как Деталь-PW(PW – сокращение ProtoWorks). Внутри программное обозначение любой детали. Хранит данные о названии детали, идентификационном номере(ID), количестве произведённых экземпляров, требуемом количестве экземпляров, “оригинальном файле” и других файлах детали(производные), состоянии детали.

**Оригинальный файл – файл, по которому деталь была добавлена в программу. Например: гараж.т3d(деталь Компас-3Д перенесена в ProtoWorks). Другие файлы детали – файлы, которые были сконвертированы из оригинального в другие форматы для использования в другом ПО*

Материалы ProtoWorks. Всё то, что не относится к деталям. Тех задания, фотографии от заказчика, чертежи.

Очередь – список работ(задач) для станка. CNCHell в автоматическом или ручном режиме распределяет все, требуемые к производству, детали по станкам в соответствующие очереди.

Работа(job) – задание для станка. Подразумевает под собой наличие двух файлов(минимум одного – расчётного/рассчитанного). Имеет данные о начале работы, состоянии работы, ID станка, на которого работа и возлагается.

*Два файла: Файл, который является оригинальным файлом детали или его производным(читать Деталь-ProtoWorks) называется *файлом расчёта*. По нему станок не способен произвести деталь. Существует так же *рассчитанный файл*. Этот файл уже может использоваться станком. Но такой файл подходит лишь под тот станок, под который и была рассчитана работа(gcode и тд)

Перейдём к устройствам, работающим в системе.

Сервер – все компьютеры, на которых работает ProtoWorks должны синхронизировать файлы проектов. Для этого используется единый сервер.

Клиент – компьютеры, на которых запущен ProtoWorks.

Хаб(НУВ) – основной мини компьютер, задача которого быть “мостом” между сервером и станками. Он собирает данные со станков и отдаёт им задания.

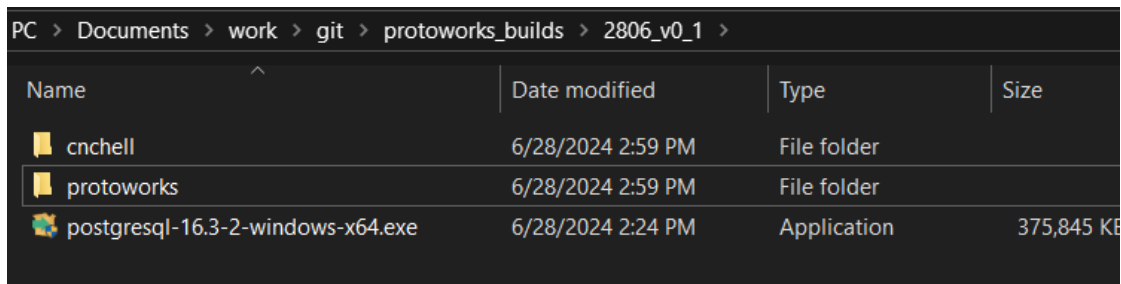
Слейв – устройство, объединяющее несколько станков в небольшую подсеть. Требуется так как сервер и хаб напрямую со станками не могут взаимодействовать.

Таким образом, например, выглядит отправка задания-работы на производство детали:

Пользователь(клиент) выбирает деталь, создаёт по ней Деталь-ProtoWorks.

Отправляет её на распределение в нераспределенную очередь, выбирает станок и отправляет деталь в очередь этого станка. Рассчитывает с помощью ПО “рассчитанный файл” для этого станка и вышлет его на сервер. Далее, когда очередь станка дойдёт до этой детали, сервер отправит деталь на слейв, к которому подключён станок, тот в свою очередь отправит рассчитанный файл станку и тот его выполнит.

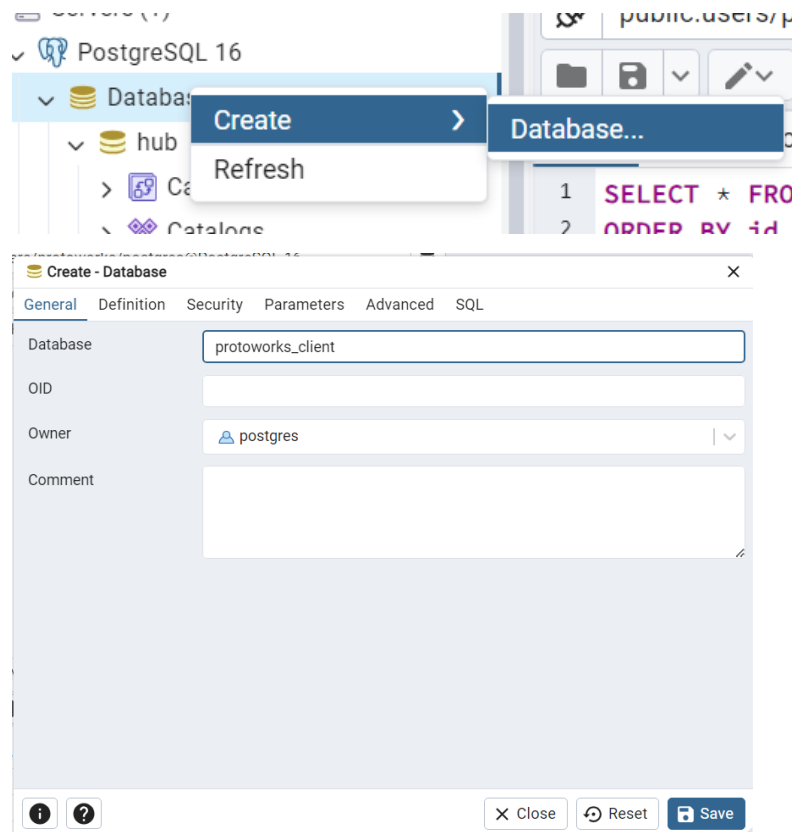
Установка ProtoWorks и CNCHell(клиент)



Name	Date modified	Type	Size
cnchell	6/28/2024 2:59 PM	File folder	
protoworks	6/28/2024 2:59 PM	File folder	
postgresql-16.3-2-windows-x64.exe	6/28/2024 2:24 PM	Application	375,845 KB

В архиве с программой присутствует установщик PostgreSQL16. Нужно его установить, не меняя параметры установки. Пароль установить такой(без кавычек): “Flvbybcnhfnjh”

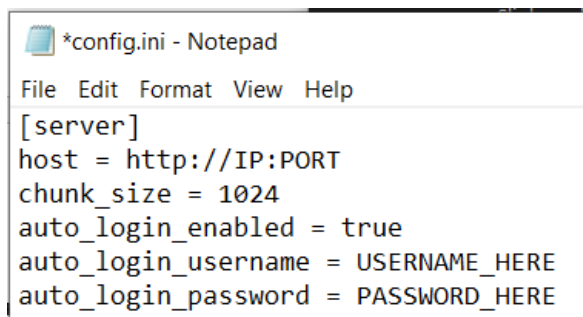
После этого нужно открыть pgAdmin4 и создать базу данных protoworks_client.



Далее идём по инструкции ниже.

Описание основных методов работы с ProtoWorks

Авторизация и конфигурация. При входе в программу у пользователя запрашивается логин, пароль и хост(IP и порт сервера запрашиваются у администратора). Для того, чтобы не вводить эти данные при входе каждый раз – их можно единожды ввести в конфигурационный файл(конфиг) config.ini. Он находится в директории ПО.



```
*config.ini - Notepad
File Edit Format View Help
[server]
host = http://IP:PORT
chunk_size = 1024
auto_login_enabled = true
auto_login_username = USERNAME_HERE
auto_login_password = PASSWORD_HERE
```

Достаточно заменить USERNAME_HERE и PASSWORD_HERE, установить auto_login_enabled = true. Login и пароль нужно запросить у администратора(инструкция администратора в конце руководства)

Так же, в config.ini присутствуют данные о путях. Их можно задать вручную как указано ниже(только пути нужно указать на своём компьютере), либо в программе в настройках.

```
[path]
projects_path = C:\Users\DumbOS\Documents\ProtoWorks\ProtoWorksClientData\Проекты\
temp_path = C:\Users\DumbOS\Documents\ProtoWorks\ProtoWorksClientData\Temp\
```

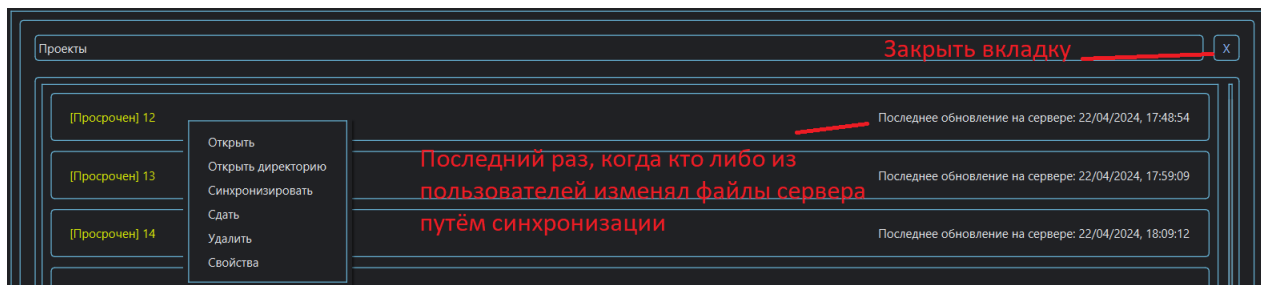
Требуется создать две папки: Для проектов и для временных файлов. Указать их в файле. После перезапустить ПО.

Главное меню.



Все кнопки говорят сами за себя. По кнопке “Нет операции” можно нажать. В окне будут отображаться задачи, которые в данный момент выполняет программа.

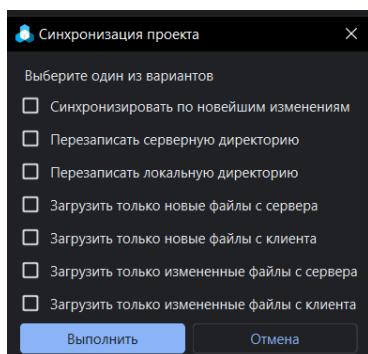
Проекты



Для создания проекта достаточно нажать не по проекту и выбрать “Создать”

Остальные кнопки так же говорят сами за себя.

Синхронизация

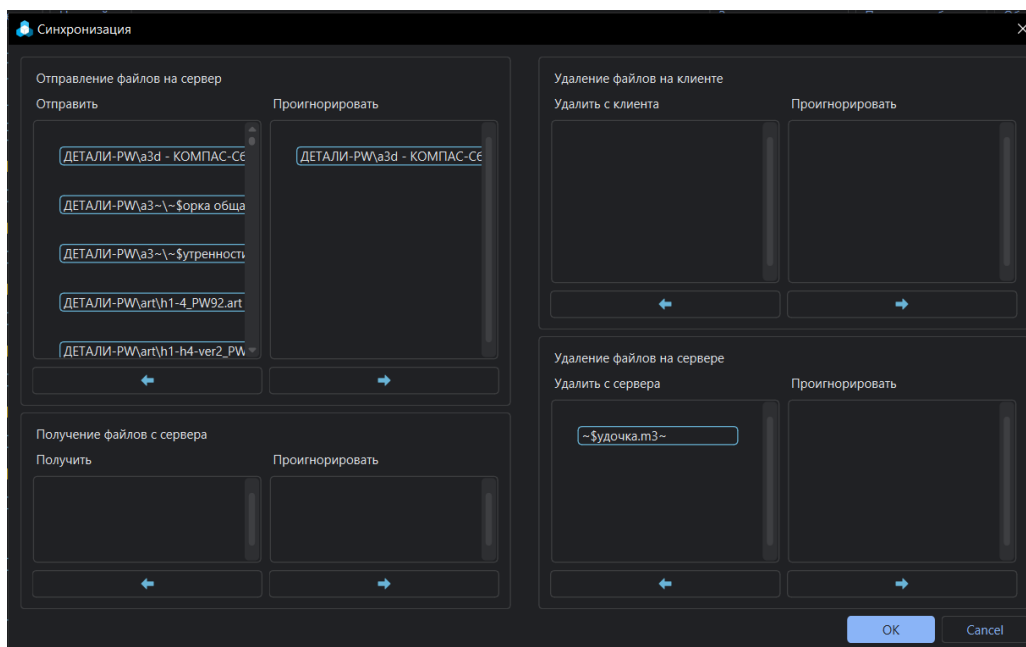


Синхронизировать по новейшим изменениям. Все файлы, которые были созданы после последней синхронизации на клиенте переносятся на сервер и наоборот. Все удалённые на клиенте файлы удаляются и на сервере и наоборот. Все изменённые на клиенте файлы получают такие же изменения на сервере и наоборот.

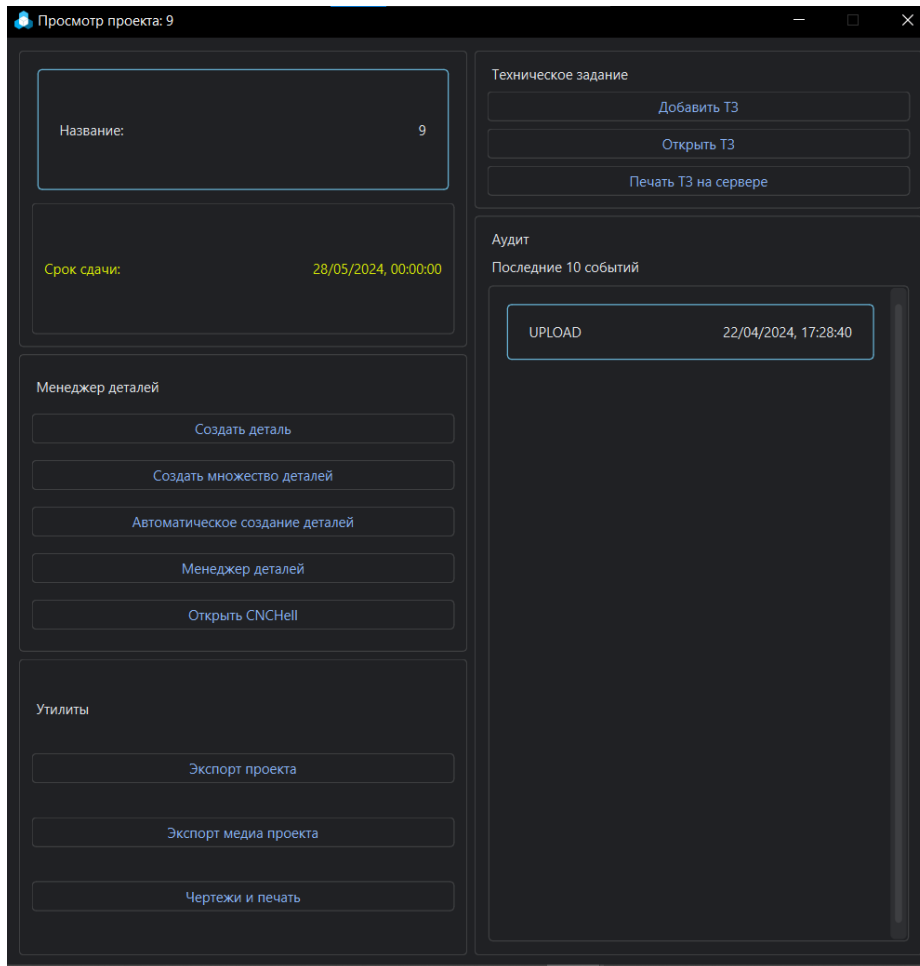
Перезаписать серверную директорию. То есть удалить серверную и записать туда директорию проекта клиента. Используется в тех случаях, когда получаются конфликты файлов. **Перезаписать локальную директорию** делает тоже самое, только на клиенте.

Остальные пункты говорят сами за себя.

Во всех пунктах далее идёт окно, где можно любой файл “проигнорировать”. То есть, если он, например, должен был быть отправлен на сервер, можно его не отправлять (например вы делали две модели. Одну не закончили, поэтому пока решили её не отправлять)



Проект->Открыть



На данный момент не используются функции: *Создать деталь, создать множество деталей*. Их полностью заменяет *Автоматическое создание деталей*.

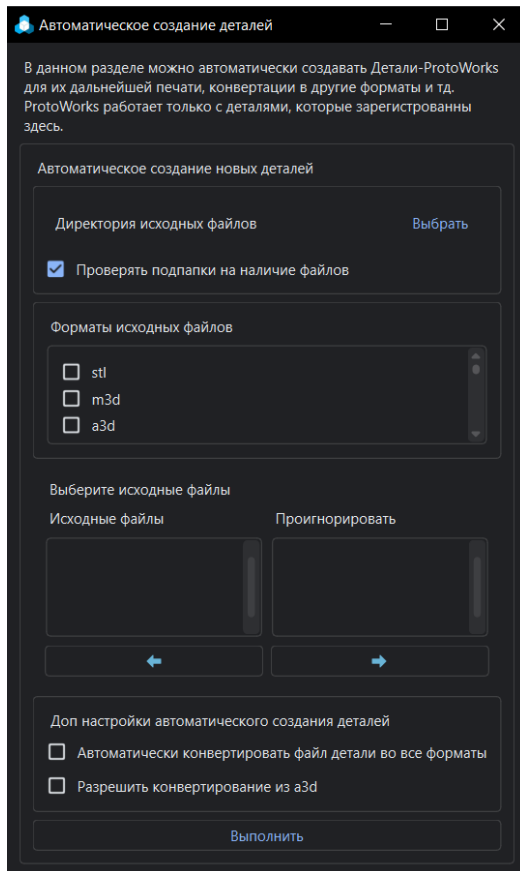
Утилиты на данный момент не реализованы.

Что бы добавить **ТЗ** к проекту, нужно нажать соответствующую кнопку. Тогда далее его можно будет открывать на любом ПК с ProtoWorks. Так же, это ТЗ можно будет напечатать на принтере, подключённом к серверу.

Все ТЗ можно посмотреть в ДиректорияПроекта/МАТЕРИАЛЫ-PW/ для открытия их вручную.

Аудит. Техническая функция, не несущая большого смысла для пользователя. Хранит в себе порядок основных событий в проекте будто загрузка деталей, файлов, удаление файлов и деталей.

Автоматическое создание деталей.



Функция позволяет все существующие файлы проекта конвертировать во все форматы, в которые можно сконвертировать эти файлы. Так же, функция сразу же создаёт и Детали-ProtoWorks для дальнейшей работы с ними.

Функция работает с файлами Компаса и другими CAD'ами только при наличии установленного Компас-3D на компьютере клиента.

Директория исходных файлов. Выбирается либо просто директория проекта, либо более глубокая, если требуется лишь определённая директория. Ниже, можно, как и в окне синхронизации, проигнорировать файлы(например если они ещё совсем не закончены).

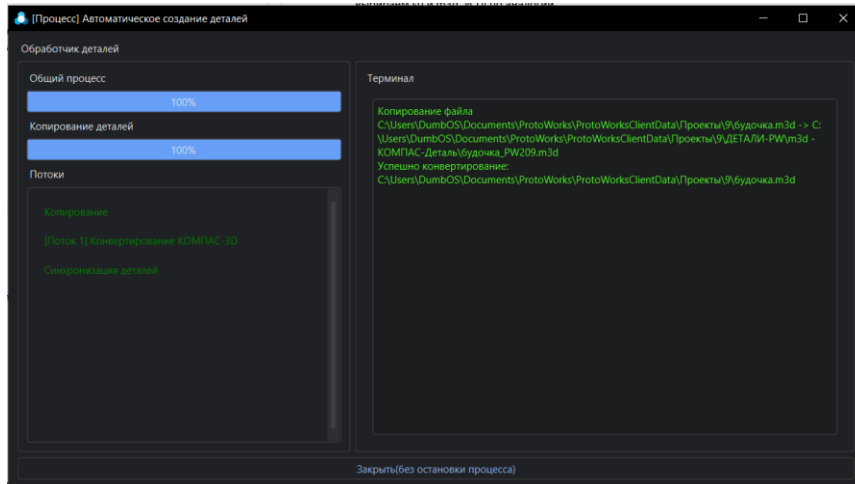
Форматы исходных файлов. Если сейчас требуется создатель детали лишь для печати, то выбираем stl и m3d. И тд по аналогии.

Автоматически конвертировать файл детали во все форматы. Включение одной из главных задач функции, описано выше.

Разрешить конвертирование из a3d(Компас3Д-Сборка). Часто приводит к ошибкам, поэтому к использованию не рекомендуется.

Далее будет открыто окно конвертации. Количество потоков можно указывать в конфигурационном файле

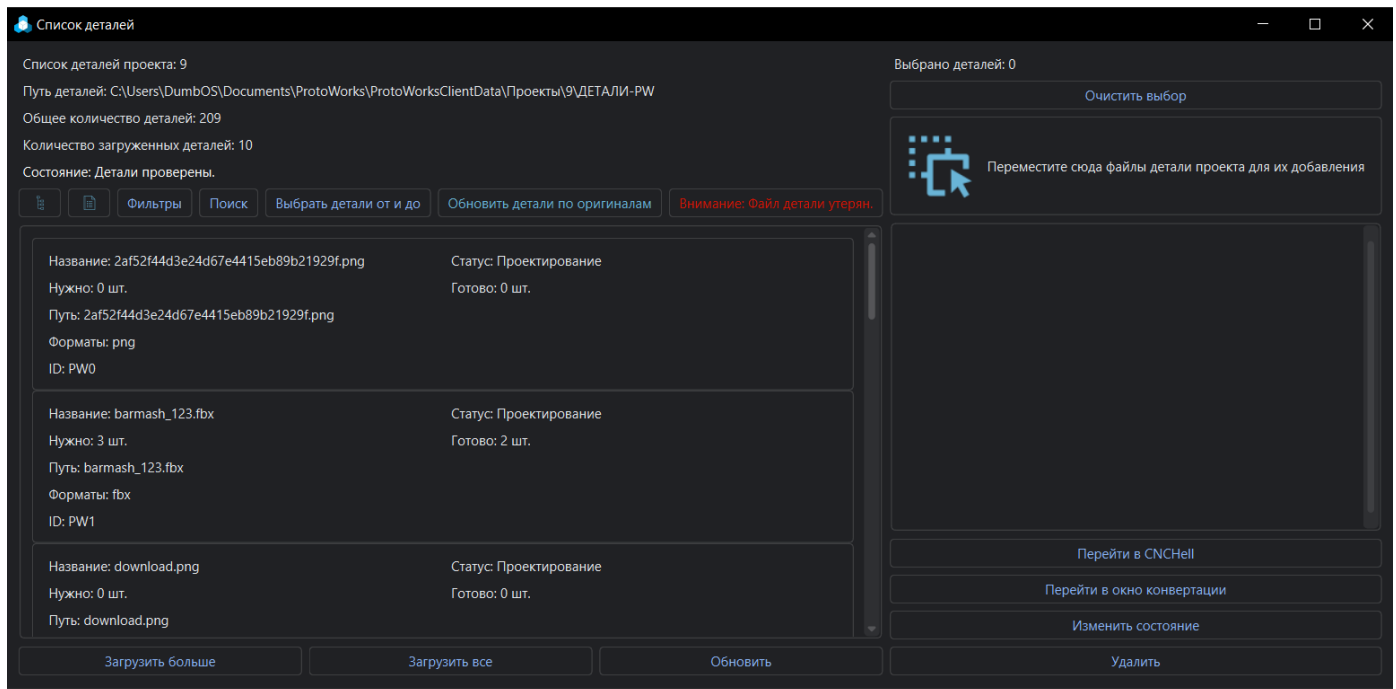
```
[heavy_processing]
task_threads = 5
```



После данной процедуры, в папке ДЕТАЛИ-PW будут созданы файлы детали во всех форматах. Их можно использовать для открытия в других программах.

ВНИМАНИЕ. НЕ РЕКОМЕНДУЕТСЯ ИХ РЕДАКТИРОВАТЬ И УДАЛЯТЬ ВРУЧНУЮ!

Менеджер деталей



Здесь отображены все Детали-ProtoWorks проекта. С помощью фильтров и поиска можно отобразить лишь нужные.

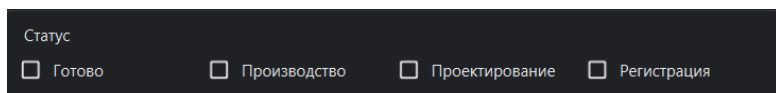
Обновить детали по оригиналам. Объясняя просто. Если изначальный “оригинальный” исходный файл детали был использован для автоматического конвертирования в другие форматы, а потом этот файл был изменён – данная кнопка загорится жёлтым цветом. Это значит, что ранее сконвертированные файлы уже не соответствуют изначальным и для их использования их требуется сначала обновить по оригинальным. Эта кнопка и позволяет это сделать.

Кнопка “Файл детали утерян” возникает лишь в том случае, если был удалён изначальный файл детали. Позволяет либо окончательно удалить деталь, либо восстановить её по локальной копии.

Чтобы взаимодействовать с деталями, нужно перенести их из левого столбца в правый. Далее их можно направить по всем пунктам расписанным в программе.

Перейти в CNCHell – отправить детали в нераспределенную очередь. Из неё уже можно перенести детали в очередь станков с дальнейшим их производством.

Состояния детали:



Статус Регистрация возникает лишь при ошибках в работе ПО.

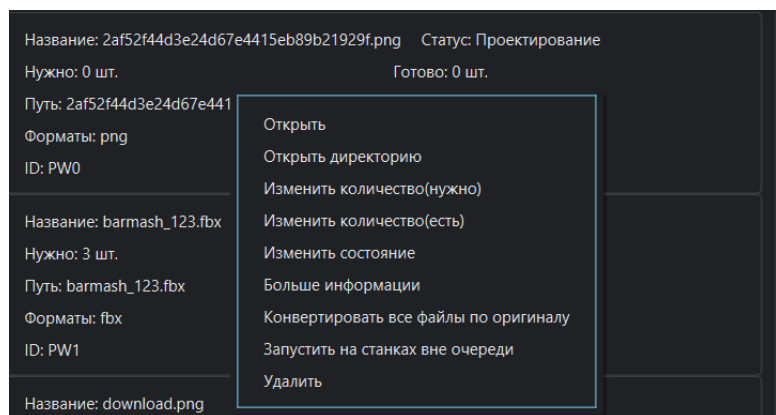
Статус Проектирование отвечает за стадию, когда деталь проектируют... но ещё не производят на станках

Статус Производство возникает тогда, когда деталь начинает производиться на станке впервые

Статус Готово возникает тогда, когда количество произведённых деталей совпадает с количеством требуемых и не равен нулю

Удаление Деталей-ProtoWorks удаляет их лишь из Базы Данных и из программы. Файлы же детали не удаляются.

По детали можно нажать ПКМ, чтобы просмотреть функции, связанные с деталью.



Описание основных методов работы с CNCHell

Авторизация и конфигурация такая же, как и с ProtoWorks. Папки Проектов и временных файлов должны у обоих программ совпадать.

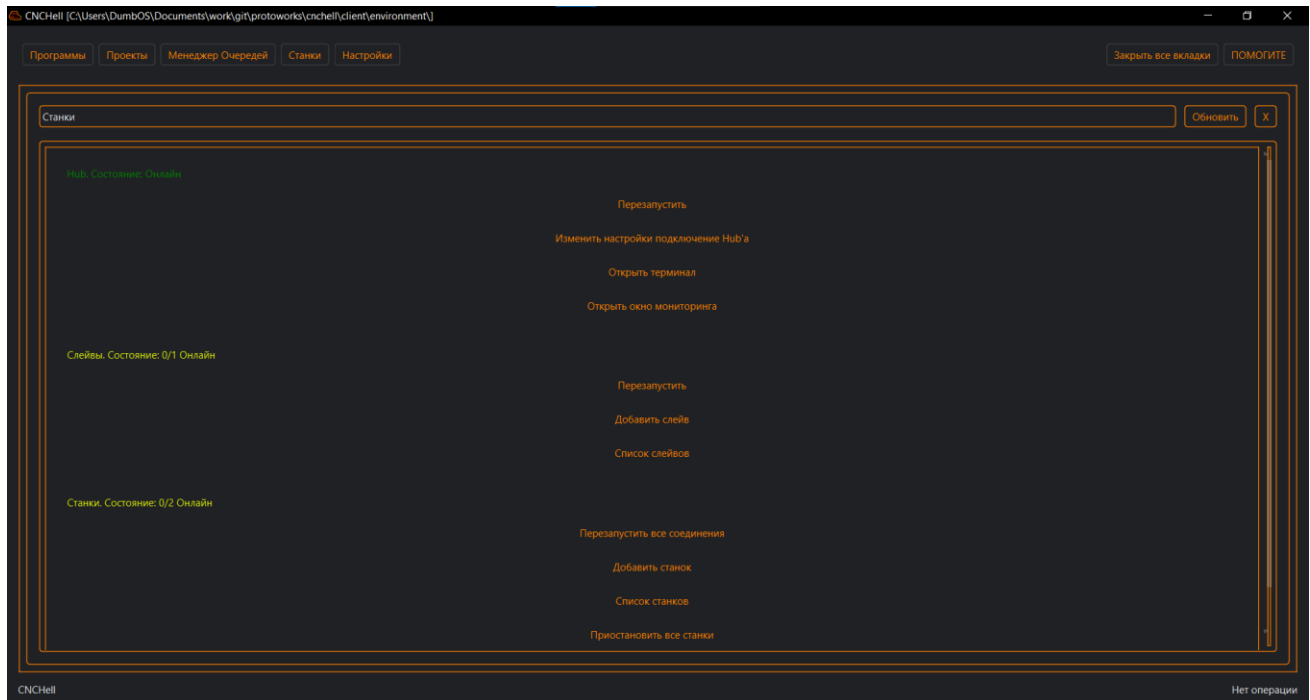
Главное меню.



Присутствуют те же функции, что и в ProtoWorks.

Перейдём к функциям, которые отличаются.

Станки.



Здесь отображён минимальный мониторинг Hub'a, слейвов и станков. Функциям добавления станков, слейвов, изменением настроек Hub'a занимается системный администратор.

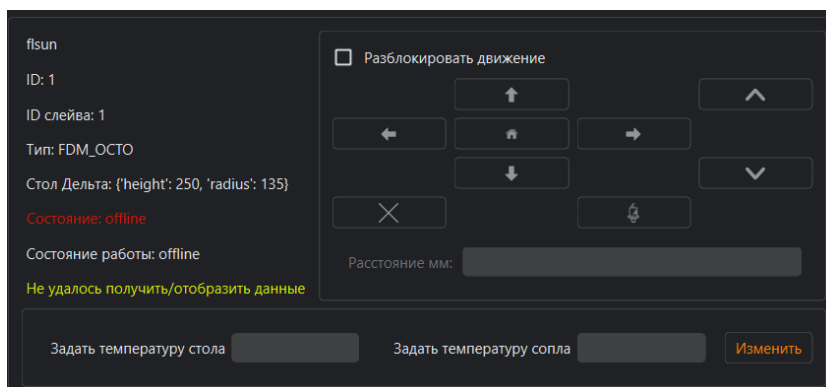
Перейдём к функциям для пользователей.

Станки->Перезапустить все соединения. Применяется в ситуациях после обесточивания помещения и тд, точнее в ситуациях, когда по каким либо причинам станки были отключены, при том все(функция отключает и станки, которые в данный момент работают).

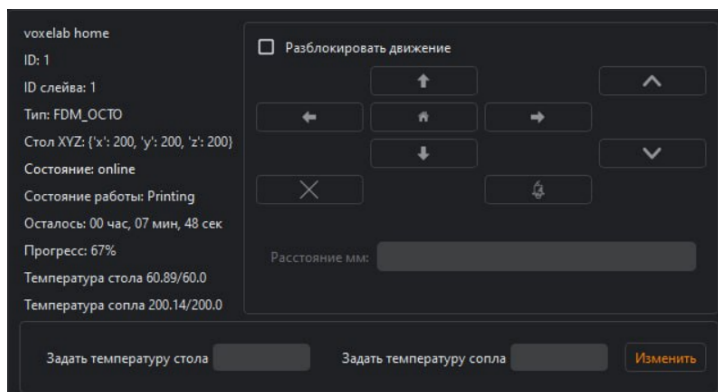
Станки->Приостановить все станки. Относится к аварийному функционалу.

Станки->Отключить все станки. Относится к аварийному функционалу.

Перейдём в Список станков

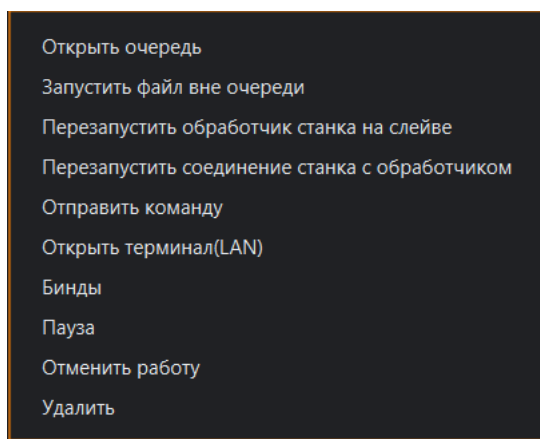


Окно управления одним станком. Здесь есть его название, ID, ID слайва(для системного администратора), тип, данные о рабочей поверхности(стол), состояние(offline/online), состояние работы(offline/Printing/Operational и тд). Во время работы станка отображаются данные о температуре, файле печати, прогрессе и тд.



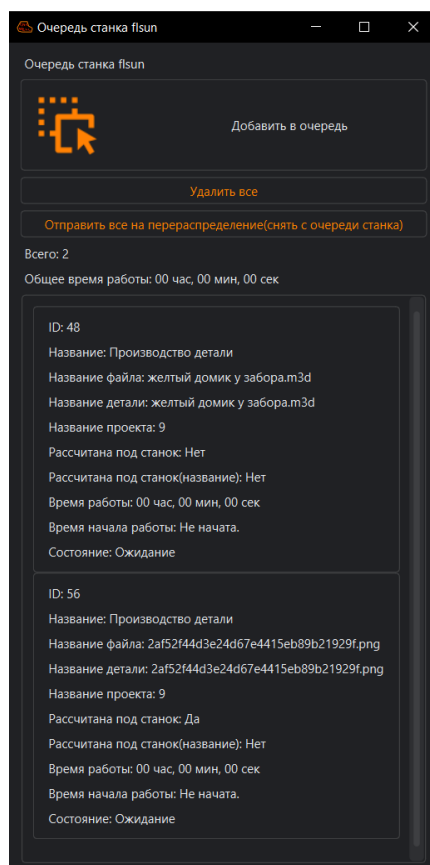
Во время того, как станок не работает, можно удалённо совершать движения, устанавливать температуру и тд, что доступно в правой части окна.

Так же, при нажатии ПКМ открываются остальные функции:



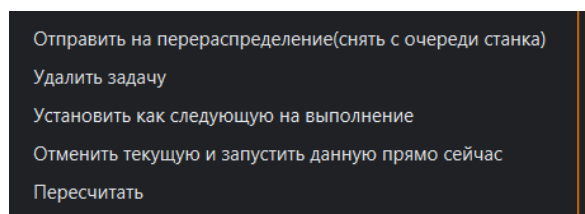
Перезапустить обработчик станка на слейве и Перезапустить соединение станка с обработчиком. Краткое описание. На слейве запущена программа – обработчик станка. Она взаимодействует со станком. Поэтому в случае её отказа следует её перезапускать. В случае, если вы видите сообщения вида “Serial connection Error” или “Disconnected”, следует перезапустить соединение со станком так как в описанном случае, обработчик слейва функционирует, но не может подключиться к станку.

Очередь станка



Сюда можно складывать как файлы Деталей-ProtoWorks(рекомендуется, они будут автоматически учитываться вместе с теми, что были отправлены из ProtoWorks кнопкой “Перейти в ProtoWorks”). Так и можно отправлять файлы, заранее рассчитанные под станок(gcode и тд). Эти файлы будут выполняться без какого либо учёта.

На ПКМ так же есть функции



Пересчитать.

расчёт работы ID: 48

Существуют файлы(gcode, rmta, rmt3, .), которые уже рассчитаны под конкретный станок, они не могут быть перерасчитаны под другой станок. Так же существуют файлы, которые под станок не рассчитаны(stl), их уже можно рассчитать под станок. Ниже нужно выбрать файл, по которому будет проводится расчёт для станка. Если же уже есть рассчитанный в ручную файл - выберите его как рассчитанный

ID: 48
Название: Производство детали
Название файла: желтый домик у забора.m3d
Название детали: желтый домик у забора.m3d
Название проекта: 9
Рассчитана под станок: Нет
Рассчитана под станок(название): Нет
Время работы: 00 час, 00 мин, 00 сек
Время начала работы: Не начата.
Состояние: Ожидание

Скачать файлы с сервера

Файл, по которому проводится расчёт: желтый домик у забора.m3d

Установить файл, по которому должен быть проведен расчёт

Установить для этого оригинальный файл детали

Открыть директорию файла

Рассчитанный под станок файл: Нет

Установить файл, который считать рассчитанным

Открыть директорию рассчитанного файла

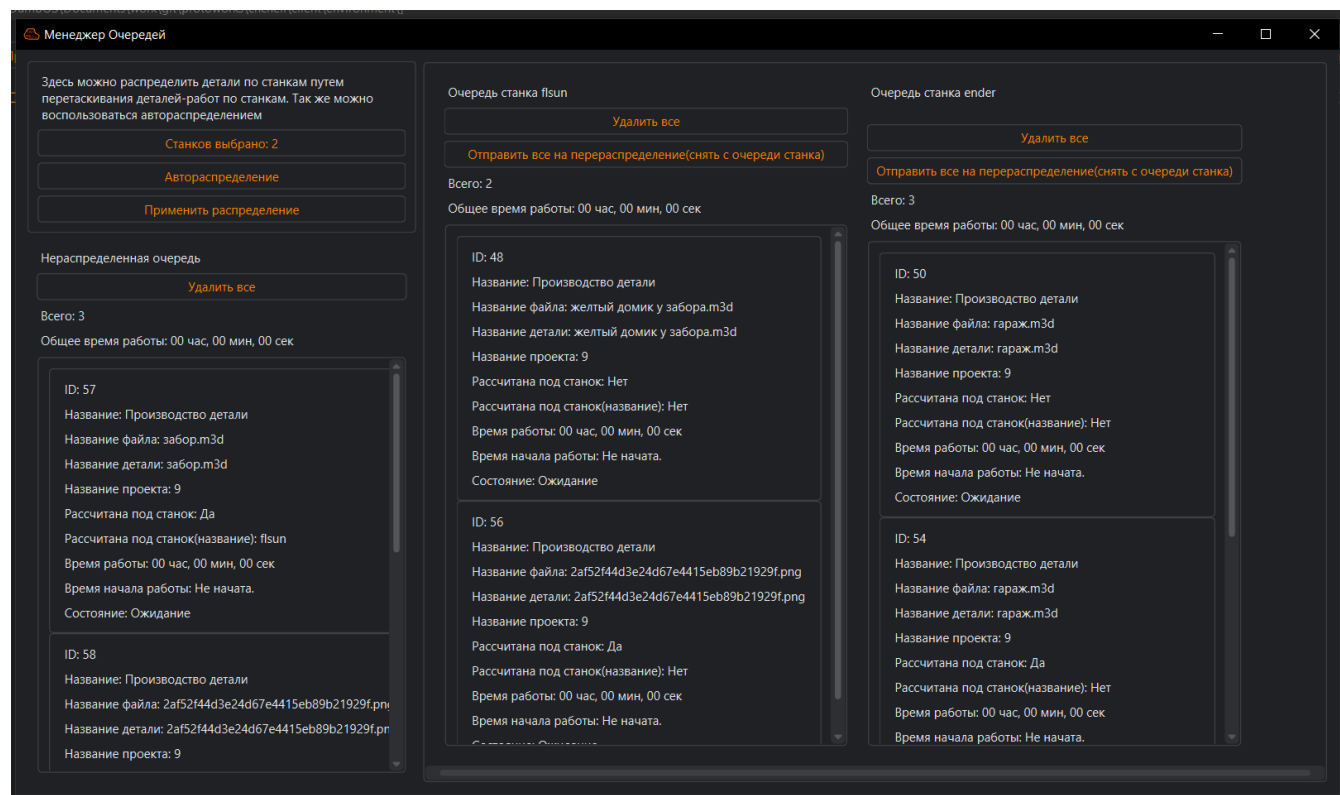
Сохранить расчёт

Ниже представлены варианты автоматического расчёта. Если есть потребность рассчитать внешним ПО просто нажмите кнопку 'Открыть директорию файла', файл который вы рассчитайте выберите в левом окне как рассчитанный

Достаточно сложная в понимании функция.

Существует два типа файлов, с которыми мы работаем. Файл детали(он же будет файлом, по которому проводится расчёт). Есть файл “рассчитанный под станок”, по нему соответственно и будет работать станок. По файлу детали станок не способен работать. Следовательно, для любой поставленной станку задачи требуется указывать “рассчитанный файл” в этой вкладке. Иначе при попытке производства будет выдана ошибка об его отсутствии.

Менеджер Очередей.



Здесь можно распределить нераспределенную очередь по станкам. Для этого требуется выбрать станки, на которые будем распределять детали. Они отобразятся в правой части окна. Из левой части окна перетаскиваем детали на нужный станок. После распределения убеждаемся, что на всех станках лежат лишь детали, рассчитанные под него. Иначе, открываем окно расчёта через ПКМ.

После всего, выбираем “Применить распределение”

Как в целом обращаться с автоматизированным станком

Добавили в очередь станка деталь. Рассчитали под неё файл под станок. После того как деталь была произведена снимаем её с оборудования, после этого нажимаем кнопку около станка.

ЧТОБЫ ОСТАНОВИТЬ СТАНОК ВО ВРЕМЯ ПЕЧАТИ НУЖНО ЗАЖАТЬ КНОПКУ НА НЕСКОЛЬКО СЕКУНД

После того как кнопка была нажата, станок переключится на следующую деталь

Руководство администратора

Перед чтением этого руководства следует прочитать руководство программы!

Начнём с сервера.

```
[path]
server_path = C:\Users\DumbOS\Documents\ProtoWorks\ProtoWorksData\
projects_path = C:\Users\DumbOS\Documents\ProtoWorks\ProtoWorksData\Projects\
temp_path = C:\Users\DumbOS\Documents\ProtoWorks\ProtoWorksData\Temp\
machines_path = C:\Users\DumbOS\Documents\ProtoWorks\ProtoWorksData\Machines\

[networking]
chunk_size = 5120
```

Нужно создать все папки как в config.ini. Желательно подключить принтер по бумаге к серверному ПК.

Далее открываем pgAdmin4 и создаём БД protoworks. Теперь открываем main.exe и сервер должен работать.

Добавление новых пользователей

Для этого открываем userlist.txt и пишем в формате

USERNAME PASSWORD PRIVILEGES

Перезапускаем сервер.

Получение хоста

Если работаете только в локальной сети:

```
C:\Users\DumbOS>ipconfig

Windows IP Configuration

Wireless LAN adapter Local Area Connection* 9:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::3e91:4d7c:d5e5:748e%13
    IPv4 Address. . . . . : 192.168.0.183
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

C:\Users\DumbOS>
```

Берём локальный IP, порт 5000 и составляем <http://IP:PORT>

Например: <http://192.168.0.183:5000>

Если работаете не только в локальной сети:

Заходим в настройки роутера, где стоит сервер, берём локальный ip адрес ПК сервера и открываем порт на этот ip. Внешний порт ставим любой свободный выше 1000, внутренний ставим 5000. Протокол ставим TCP/UDP

Пример:

The screenshot shows a router's configuration page with two main sections: 'Общие настройки' (General settings) and 'Настройки внутренней сети' (Internal network settings). In the 'Общие настройки' section, the 'Включить' (Enable) toggle is turned on. Under 'Шаблон' (Template), 'Custom' is selected. Under 'Интерфейс' (Interface), '<Все>' (All) is selected. Under 'Протокол' (Protocol), 'TCP/UDP' is selected. The 'NAT Loopback' option is disabled. In the 'Настройки внутренней сети' section, the 'Внутренний IP*' (Internal IP) is set to '192.168.0.183' and the 'Внутренний порт*' (Internal port) is set to '5000'. Below this, there is a section for 'Настройки публичной сети' (Public network settings) with a sub-section 'Удаленный IP' (Remote IP) which is currently empty. A button 'ДОБАВИТЬ УДАЛЕННЫЙ IP' (Add remote IP) is visible. The 'Внешний порт*' (External port) is set to '1379'. At the bottom left of the configuration area is a blue button labeled 'ПРИМЕНИТЬ' (Apply).

Далее заходим в интернет и узнаём свой публичный IP. Например через <https://2ip.ru>

И получаем хост в стиле <http://83.169.216.0:1379>

Установка ПО на HUB (Orange Pi Zero)

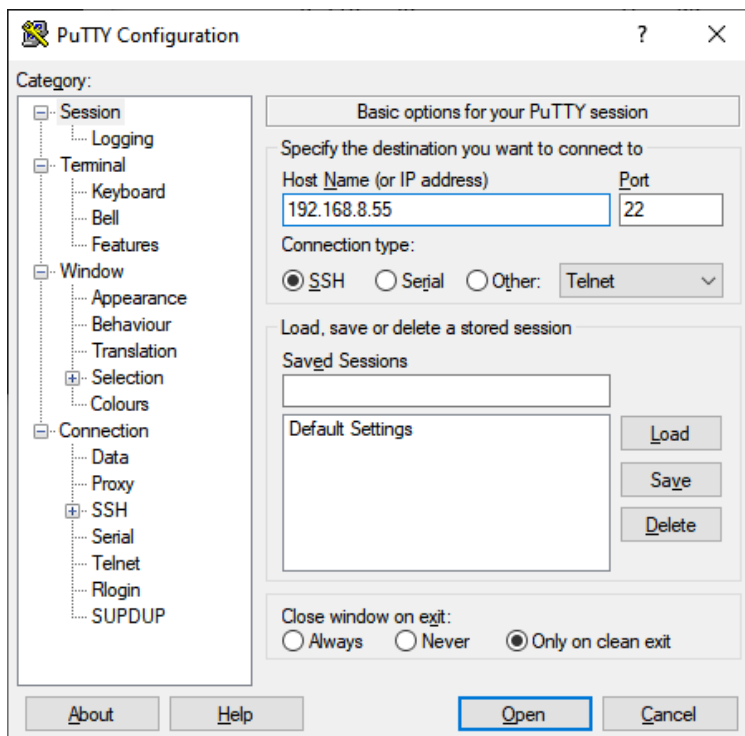
Нужно установить операционную систему. Для этого SD карту вставляем в кардридер и через программу Rufus(<https://rufus.ie/ru/>) устанавливаем образ ОС, указанный ниже.

Образ **Armbian**(<https://www.armbian.com/orange-pi-zero/>)

После вставляем флешку в плату и подключаем её в сеть.

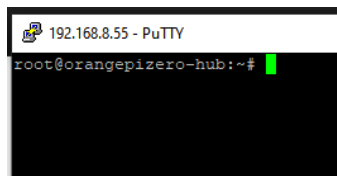
ВНИМАНИЕ*: Плата стартует несколько минут. Работает только по проводному интернету. Если в дальнейшем нужен Wi-Fi: после установки ОС пропишите **nmtui и подключитесь к Wi-Fi*

Нужно войти в систему через программу PuTTY(<https://www.putty.org/>) по IP адресу из роутера. Порт 22, type: SSH



Если это сделали успешно. Входим в аккаунт **root**. Обычный пароль **1234**. Следуем инструкциям, запоминаем пароль.

После увидели консоль в стиле:



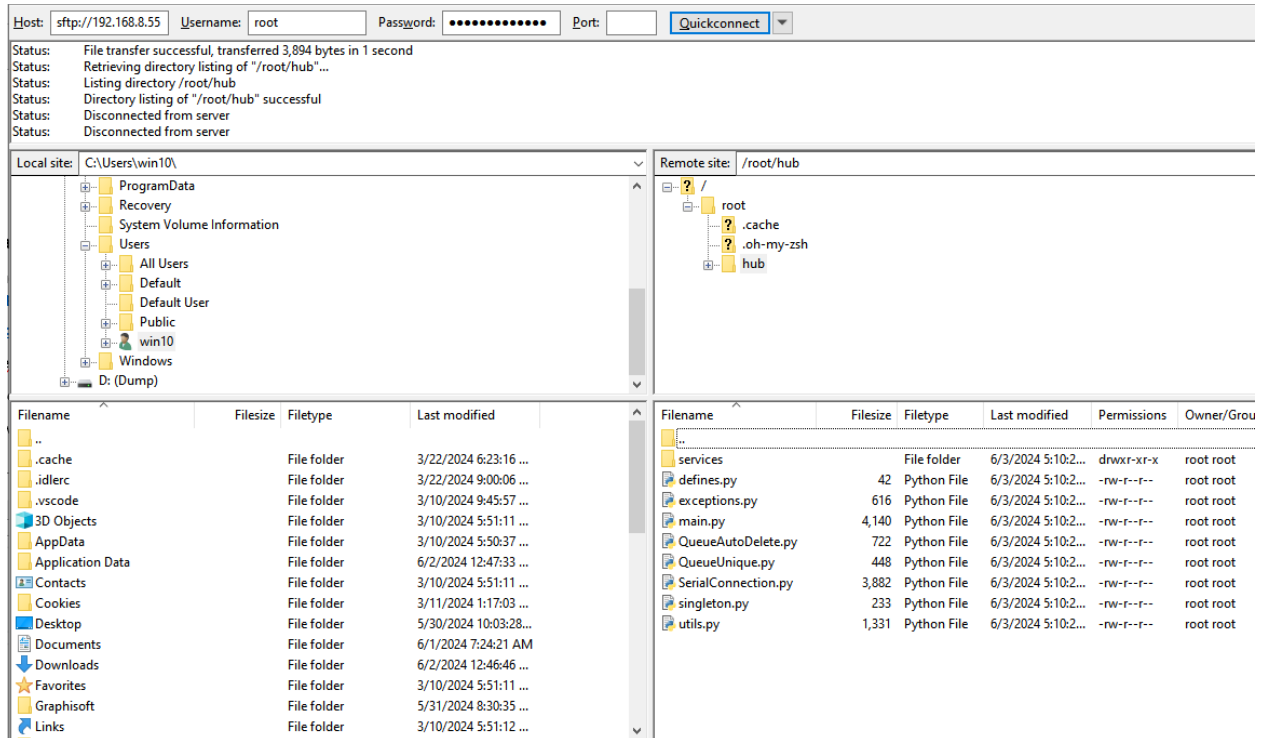
Вводим следующие команды:

```
apt update && apt upgrade  
apt install python3.11-full
```

```
apt-get -y install python3-pip
cd
mkdir hub
```

Открываем на компьютере **FileZilla**(<https://filezilla.ru/>)

Подключаем к плате и заходим в папку hub



В папку hub сбрасываем файлы из репозитория([protoworks/cnchell/hub](https://github.com/protoworks/cnchell/hub)/opi)

Далее возвращаемся в консоль и пишем:

```
cd
cd hub
python3.11 -m pip install -r requirements.txt
```

Проверяем установку с помощью: `python3.11 main.py`

Если видим что-то подобное(Нет ошибок и приложение работает):

```
root@orange-pi-zero-hub:~/hub$ python3.11 main.py
hub server v0.1
* Serving Flask app 'main' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://192.168.8.55:5001/ (Press CTRL+C to quit)
```

То нажимаем Ctrl+C и переходим к установке авто запуска.

```
nano /etc/systemd/system/hub.service
```

И копируем текст ниже

```
[Unit]

Description=Hub
After=network-online.target
Wants=network-online.target

[Service]
Environment="LC_ALL=C.UTF-8"
Environment="LANG=C.UTF-8"
Type=exec
User=root
ExecStart=python3.11 /root/hub/main.py

[Install]
WantedBy=multi-user.target
```

Нажимаем Ctrl+X и выходим.

Пишем

```
systemctl enable hub
systemctl start hub
systemctl status hub
```

Если видим что то такое:

```
192.168.8.55 - PuTTY
systemd/system/hub.service.
root@orangezero-hub:~/hub# systemctl start hub
root@orangezero-hub:~/hub# systemctl status hub
● hub.service - Hub
   Loaded: loaded (/etc/systemd/system/hub.service; enabled; vendor preset: e
   Active: active (running) since Mon 2024-06-03 12:54:27 UTC; 2s ago
     Main PID: 7100 (python3.11)
        Tasks: 3 (limit: 882)
       Memory: 16.3M
          CPU: 2.356s
      CGroup: /system.slice/hub.service
              └─7100 python3.11 /root/hub/main.py

Jun 03 12:54:27 orangezero-hub systemd[1]: Started Hub.
Jun 03 12:54:30 orangezero-hub python3.11[7100]: hub server v0.1
Jun 03 12:54:30 orangezero-hub python3.11[7100]: * Serving Flask app 'main'
Jun 03 12:54:30 orangezero-hub python3.11[7100]: * Environment: production
Jun 03 12:54:30 orangezero-hub python3.11[7100]: WARNING: This is a develo
Jun 03 12:54:30 orangezero-hub python3.11[7100]: Use a production WSGI ser
Jun 03 12:54:30 orangezero-hub python3.11[7100]: * Debug mode: off
Jun 03 12:54:30 orangezero-hub python3.11[7100]: * Running on all addresses.
Jun 03 12:54:30 orangezero-hub python3.11[7100]: WARNING: This is a develo
Jun 03 12:54:30 orangezero-hub python3.11[7100]: * Running on http://192.168
lines 1-20/20 (END)
```

То нажимаем Ctrl+C и выходим из системы(exit)

ПО установлено.

Установка ПО на слейв(slave)

Повторяем все шаги с HUB, но ставим скрипт в соответствии с нужным.

Установка ПО на Arduino

Скачиваем Arduino IDE и открываем protoworks/cnchell/hub/arduino_btn_hub/arduino_btn_hub.ino и заливаем скетч на Arduino

Драйвера CH340G в папке с protoworks

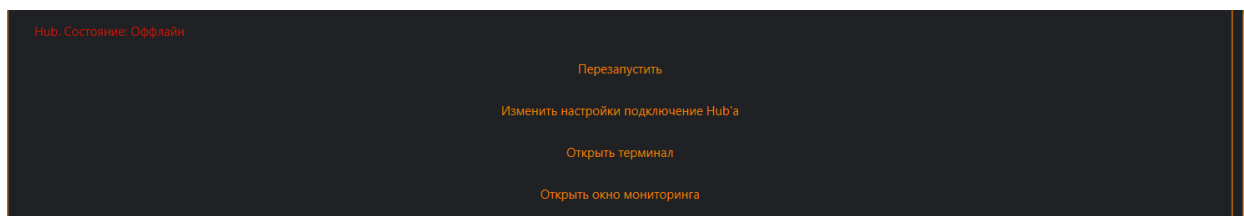
Как вообще должна работать система

Существует сервер, на нём запущено приложение сервера. Есть хаб, он тоже запущен и находится в одной локальной сети с сервером. К хабу по USB подключена arduino. В одной локальной сети с хабом и сервером есть слейвы, к ним уже подключены станки по USB.

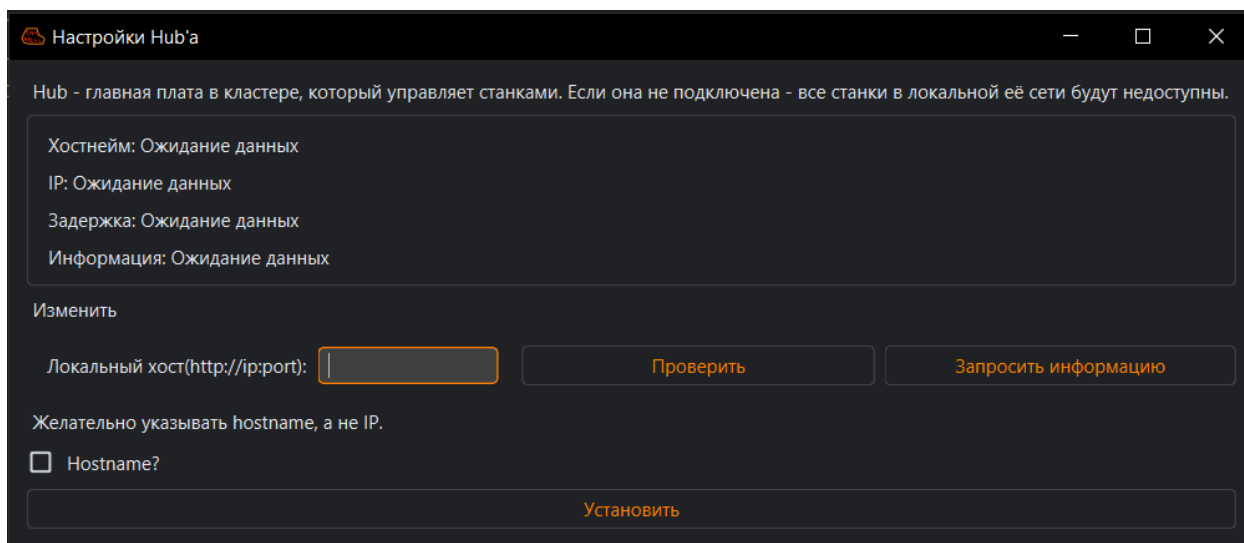
От Arduino уходит множество jack3.5. Они для кнопок завершения печати. Что бы их сделать нужно просто припаять два контакта кнопки к двум контактам jack3.5(прозвонить мультиметром на оригинальной схеме что бы понять, что и куда)

Подключение хаба, слейвов, станков

Начинаем с хаба. После его запуска, открываем CNCHell->Станки.



Выбираем **Изменить настройки подключения Hub'a**



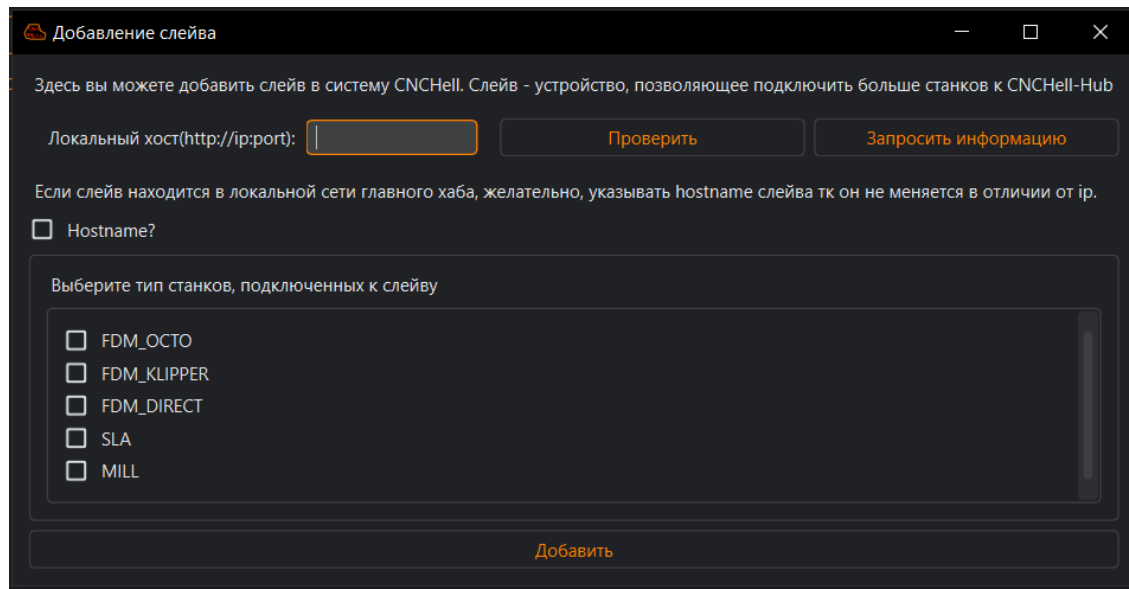
Здесь указываем локальный хост HUB'a. Его можно узнать через SSH к хабу командой ifconfig. Порт 5001.

Нажимаем **Проверить** что бы получать время задержки – пинг. Нажимаем **Запросить информацию**. Должен быть ответ **CNCHell HUB VER0.1**

Делать через hostname не желательно. Это просто плохо работает на практике.

Теперь при нажатии кнопки **Обновить**, хаб должен светиться зелёным **Онлайн**.

Теперь настраиваем слейвы.



В целом, всё тоже самое. Надпись должна быть **CNCHell OCTO SLAVE VER0.1**

Только вместо OCTO то, что вам было нужно установить.

FDM_OCTO – FDM принтеры с обработчиком OctoPrint

FDM_KLIPPER – FDM принтеры с обработчиком Klipper(на момент v0.1 не поддерживаются)

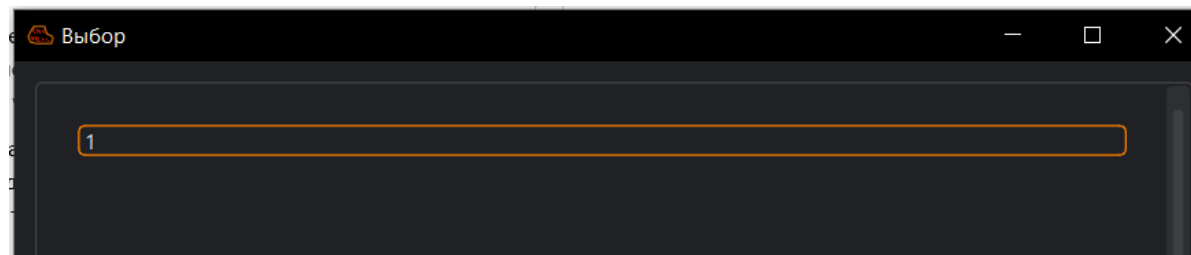
FDM_DIRECT – FDM принтеры через собственный обработчик(на момент v0.1 не поддерживаются)

SLA – Фотополимерные станки(на момент v0.1 не поддерживаются)

MILL – Фрезерные станки(на момент v0.1 не поддерживаются)

Добавляем станки

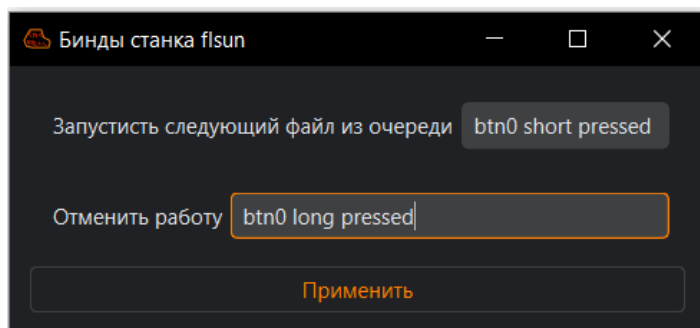
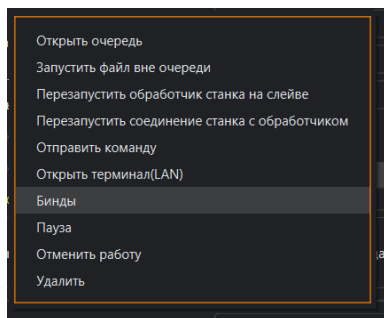
Первым выбором просят выбрать слейв



Выбираем, очевидно, тот, к которому мы подключили новый станок

Далее всё интуитивно понятно.

После добавления станка нужно прописать ему бинды



Бинды – кнопка, которую нужно нажать чтобы получить соответствующий результат.
Пример на скриншоте. Число – номер джека на ардуино.

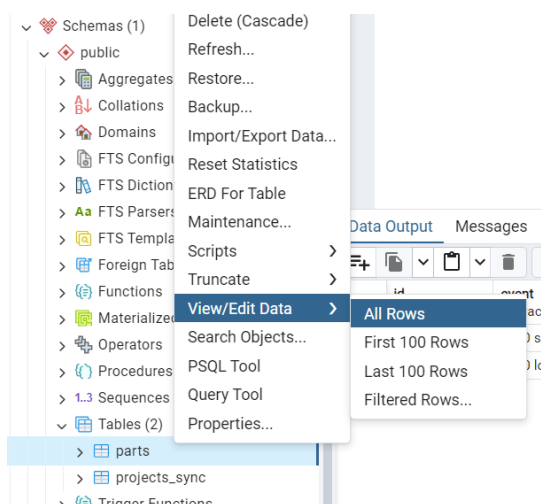
Short pressed – короткое нажатие

Long pressed – длинное нажатие(более 500мс)

Справочник по базе данных

Используется на случай различных ошибок, которые не удаётся исправить программно

Открываем pgAdmin4. Что бы открыть таблицу нужно открыть вкладку Схемы и там уже таблицы, нажать ПКМ и выбрать чтение



Существует база данных пользователя protoworks_client и сервера protoworks

protoworks_client

parts не используется

projects_sync. База данных хранящая данные о том, какой проект когда последний раз был синхронизирован с сервером. В случае конфликтов может помочь изменение этой БД

protoworks

audit. Аналог projects_sync для сервера. Хранит: действие, момент исполнения действия, информацию. В информации указано какие конкретно файлы и детали были изменены

bindings. База данных биндов для станков. Можно использовать для того что бы на одну кнопку быстро назначить взаимодействие с множеством станков. Хранит: станки, к которым обращается binding, сам binding, действие которое будет совершено

1	1	btn0 short pressed	{"devices": ["MACHINE1"], "action": "NEXT_JOB"}
2	2	btn0 long pressed	{"devices": ["MACHINE1"], "action": "CANCEL_JOB"}

MACHINEN – MACHINE и её ID

files. Файлы проектов. Хранит путь, момент изменения файла, размер, ID проекта

files_logging. Тоже самое, но эта база данных при удалении файлов не удаляет файл

hub. Здесь хранятся данные о хабе. А именно лишь его ip(с портом) и хостнейм

machines. Здесь хранятся данные о станках. ID, имя, размеры стола, ID слейва, к которому подключён станок, gcode_manager(MARLIN/REPETIER.... На момент v0.1 не имеет никакого значения) baudrate - Баудрейт станка. На данный момент не имеет никакого значения

```
144 GCODE_TYPES_STR = ["Marlin"]
```

materials. Хранение материалов. То есть ТЗ например. Хранит путь, ID проекта, тип(

```
MATERIAL_TECH_TASK = 0  
MATERIAL_MEDIA = 1
```

monitoring. Здесь в реальном времени данные мониторинга устройств. Хранит ID устройства и его тип в том же стиле, что и в machines. time_scanned – unix время момента, в которой были запрошены данные о состоянии. status, info – данные мониторинга

parts. Содержит данные о Деталях-ProtoWorks. Хранит ID детали, ID проекта, название, путь к оригинальному файлу, количество штук произведённых и требуемых, состояние

```
PART_DONE = 0 # done  
PART_PRODUCTION = 1 # printing/milling  
PART_IN_WORK = 2 # cad  
PART_IN_COORDINATION = 3 # waiting
```

programs_configs. В v0.1 не используется.

projects. Данные о проектах. Хранит ID проекта, описание, заказчика, время появления, дедлайн, состояние, путь на сервере.

```
4 PROJECT_IN_WORK = 0  
5 PROJECT_DONE = 1  
6
```

slaves. Данные о слайвах. ID, hostname, ip(и порт), тип.

```
GENE_MACHINE_ID = 0
FDM_OCTO = 0
FDM_KLIPPER = 1
FDM_DIRECT = 2
SLA = 3
MILL = 4
```

storage. Не используется

tokens. База данных токенов входа в систему. Если очень надо войти по старому токenu – можно expiration поменять на число побольше

users. Вместо userlist.txt можно использовать это. Можно применять для того, чтобы добавлять пользователей без перезапуска сервера

work_queue. Хранение очередей. ID станка(-1 для нераспределённой очереди), время начала работы, время конца работы, состояние(можно поменять если очень надо), индекс – порядковый номер в своей очереди, уникальная информация(данные о тех самых двух файлах расчёта, название работы). Зная отсюда job_send_pre_calculated_filename можно получить файл, который должен быть отправлен на станок в папке Machines/WorkingDirectory

На этом Руководство на момент 28.06.2024 версии программы v0.1 заканчивается.

Спасибо.