

УТВЕРЖДАЮ

Генеральный директор ООО
«Алнисофт»

_____ А.С. Азаров

М.П.

«___» _____ 2023 года

Программное обеспечение «ArtInCol»

**Инструкция по установке экземпляра программного обеспечения,
предоставленного для проведения экспертной проверки**

На 28 листах

Аннотация

Настоящий документ представляет собой инструкцию по установке экземпляра программного обеспечения «ArtInCol» (далее – ПО «ArtInCol», Система), предоставленного для проведения экспертной проверки.

Документ содержит описание действий по установке и настройке ПО, инструкцию по разворачиванию экземпляра ПО, а также контакты технических специалистов, которые могут проконсультировать по процессу развертывания и настройки экземпляра ПО и его функционирования.

Содержание

Перечень сокращений	4
1.1 Наименование программного обеспечения	5
1.2 Краткое описание возможностей средства автоматизации	5
2 Назначение и условия применения	6
2.1 Задача	6
2.2 Потенциальные пользователи Решения	6
2.3 Основные процессы, подлежащие автоматизации	6
2.4 Условия применения ПО в соответствии с назначением	7
2.4.1 Внешнее окружение и технологии в части средств сопряжения с оборудованием и обработки видеопотока	7
2.4.2 Внешнее окружение и технологии в части искусственного интеллекта	7
2.4.3 Аппаратная платформа	7
3 Порядок развертывания и проверки работоспособности Системы	8
3.1 Сопряжение с оборудованием и обработка видеопотока	8
3.2 ML-компоненты	25
4 Контактная информация	27

Перечень сокращений

Сокращение	Расшифровка
ADR	(англ. Adenoma Detection Rate) – отношение числа колоноскопий, при которых выявлена одна или несколько гистологически подтвержденных аденом к общему числу выполненных колоноскопий
COM	(англ. serial port, COM-порт, англ. Communications Port) – название интерфейса стандарта RS-232, которым массово оснащались персональные компьютеры
CPU	(англ. Central Processing Unit) – центральное обрабатывающее устройство – электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (код программ), главная часть аппаратного обеспечения компьютера или программируемого логического контроллера
CUDA	(англ. Compute Unified Device Architecture) – программно-аппаратная архитектура параллельных вычислений, которая позволяет существенно увеличить вычислительную производительность благодаря использованию графических процессоров фирмы Nvidia
EULA	(англ. end-user license agreement – «лицензионное соглашение с конечным пользователем»); сокращается до EULA) – договор между владельцем компьютерной программы и пользователем ее копии
FPS	(англ. Frames Per Second) – количество сменяемых кадров за единицу времени
LPT	(англ. Line Print Terminal; также параллельный порт, порт принтера) – международный стандарт параллельного интерфейса для подключения

Сокращение	Расшифровка
	периферийных устройств персонального компьютера
ML	(англ. Machine Learning) – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение за счет применения решений множества сходных задач
PDR	(англ. Polyp Detection Rate) – отношение числа колоноскопий, при которых визуализирован один или несколько полипов к общему числу выполненных колоноскопий
USB	(англ. Universal Serial Bus – «универсальная последовательная шина») – последовательный интерфейс для подключения периферийных устройств к вычислительной технике
ПО	Программное обеспечение

1.1 Наименование программного обеспечения

Программное обеспечение ArtInCol, Система.

1.2 Краткое описание возможностей средства автоматизации

ArtInCol предназначен для автоматизированного детектирования новообразований в видеопотоке с цифровых аппаратов колоноскопии при эндоскопической диагностике внутренней поверхности кишечника. Детектирование осуществляется с помощью технологий искусственного интеллекта (компьютерное зрение).

Использование ArtInCol обеспечивает повышение среднего качества проведения процедуры колоноскопии за счет уменьшения количества как необоснованных удалений гиперпластических полипов, так и оставлений неоплазий со злокачественным потенциалом – как следствие способствует выбору корректной тактики лечения.

2 Назначение и условия применения

2.1 Задача

Колоноскопия – золотой стандарт комплексной диагностики предраковых состояний и новообразований толстой кишки. Пациенты с колоректальным раком составляют около 10% от всех пациентов с онкопрофилем (400 тыс. пациентов). Каждый 20-й человек рискует столкнуться с раком толстой кишки, при этом шанс выздороветь при ранней диагностике – 90%.

Задача детекции эпителиальных образований при колоноскопии является целевой при проведении исследования и отражается в основном показателе качества выполненных эндоскопий: количества выявленных образований (показатели Adenoma Detection Rate (ADR) и Polip Detection Rate (PDR)). При этом на этот показатель влияет не только опыт эндоскописта, так его внимательность и концентрацию.

2.2 Потенциальные пользователи Решения

Целевой аудиторией проекта являются врачи-эндоскописты, выполняющие диагностическую колоноскопию на цифровых колоноскопах различного уровня.

В год в Российской Федерации выполняется около 1 млн. исследований.

Количество колоноскопов – более 7000. Число врачей-эндоскопистов – более 5000.

2.3 Основные процессы, подлежащие автоматизации

Основным вариантом использования Системы врачами-эндоскопистами является работа с исследованием в синхронном режиме (во время проведения исследования):

1) Пользователь с помощью колоноскопа проводит исследование с обработкой видеопотока, основанной на технологиях искусственного интеллекта, оставляет пользовательские метки и снимки, просматривая экран на стойке колоноскопа, в том числе ориентируясь на подсказки интерактивного модуля – рамки, оконтуривающие зоны интереса и находки. Пользователь может добавлять метки врача, включать и останавливать запись видео, включать и выключать отображение рамок;

2) При завершении исследования Система автоматически сохраняет результаты проведения исследования на жестком диске устройства, на котором установлена Система.

2.4 Условия применения ПО в соответствии с назначением

2.4.1 Внешнее окружение и технологии в части средств сопряжения с оборудованием и обработки видеопотока

C++, OpenCV, Visual Studio Code.

Реализация на языке программирования C++. Ядром приложения, обрабатывающим видеопоток является мультимедийный фреймворк gstreamer. За обработку кадров и отрисовку разметки отвечает OpenCV. Взаимодействие с ML (на языке python) реализуется при помощи библиотеки python.h. HTTP клиент и сервер для реализации взаимодействия с бэкендом реализуются на основе библиотеки httplib. Для интеграции зависимостей используется Boost DI. Парсинг json файлов конфигурации реализуется с использованием библиотеки nlohmann. Для взаимодействия с хранилищем MinIO используется AWS SDK.

2.4.2 Внешнее окружение и технологии в части искусственного интеллекта

Python 3.8.10, PyTorch, TensorRT, CUDA, PyCUDA, JetPack, ONNX, ClearML, YOLOv5, OpenCV, SORT, FilterPy, imageio, shapely, scikit-image, scikit-learn, norfair, motpy, pandas, seaborn, matplotlib, scipy, ffmpeg, numpy, psutils, tempfile, jupyter-notebook, poetry, Docker containers for Windows, vscode, WSL.

Реализация на языке программирования Python с использованием фреймворка pytorch для обучения модели на архитектуре YOLOv5. Трекинг экспериментов - ClearML. Модель переводится в формат ONNX и затем в формат engine для работы с библиотекой TensorRT на устройствах класса Jetson. Инициализация модели (Singleton) и предсказание по кадру по запросу из приложения разрабатываемого на c++. Для взаимодействия c++ и python реализуются интерфейсы взаимодействия. Первоначальные эксперименты с последующей оценкой производительности, далее реализация для работы в режиме реального времени. Для фильтрации результатов работы модели YOLOv5 реализуется постобработка, включающая фильтрацию и сглаживание контурной рамки при помощи фильтра Калмана и алгоритма SORT.

2.4.3 Аппаратная платформа

Для развертывания ПО ArtInCol предоставляется масштабируемая и отказоустойчивая аппаратная платформа, с применением технологий виртуализации и контейнеризации, которая базируется на применении серийно выпускаемого оборудования, обеспечивающего исправное функционирование Системы при заданных требованиях по надежности.

Функциональный компонент Интеллектуальный модуль ПО ArtInCol разворачивается на оборудовании со следующими характеристиками:

- производительность ИИ: не менее 200 TOPS;
- количество ядер GPU: не менее 1792;
- максимальная частота GPU: не менее 930 МГц;
- количество ядер процессора: не менее 8;
- максимальная частота процессора: не менее 2,2 ГГц;
- память: не менее 32 ГБ;
- накопитель: не менее 64 ГБ.

3 Порядок развертывания и проверки работоспособности Системы

3.1 Сопряжение с оборудованием и обработка видеопотока

Для настройки среды необходимо выполнить следующие шаги:

- 1) Установить JetPack компоненты;
- 2) Настроить окружение;
- 3) Настроить режим работы Jetson.

Для установки JetPack необходимо выполнить следующие шаги:

- 1) Выполнить установку Ubuntu:
 - Если установка осуществляется с подключенным монитором:
 - Подключить к Jetson DisplayPort-кабель (8) (если монитор имеет только HDMI, использовать адаптер или кабель DisplayPort-HDMI) (рисунок 1):



Рисунок 1 – Подключение DisplayPort-кабеля

- Подключить к Jetson клавиатуру и мышь (12)

(рисунок 2):

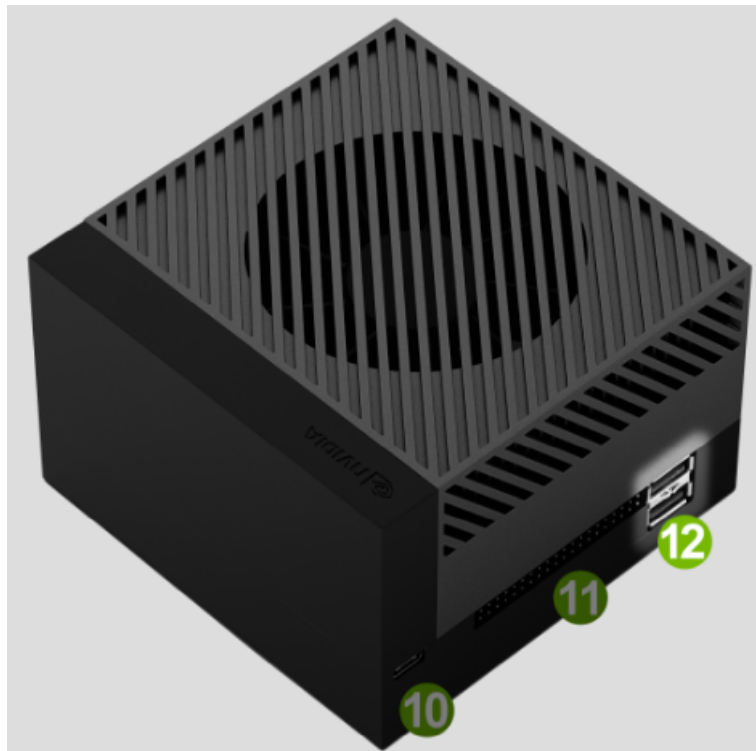


Рисунок 2 – Подключение клавиатуры и мыши

- Подключить к Jetson Ethernet-кабель (6) (рисунок 3);



Рисунок 3 – Подключение Ethernet-кабеля

- Подключить кабель питания из комплекта в USB-порт, находящийся над портом питания (4) (рисунок 4);

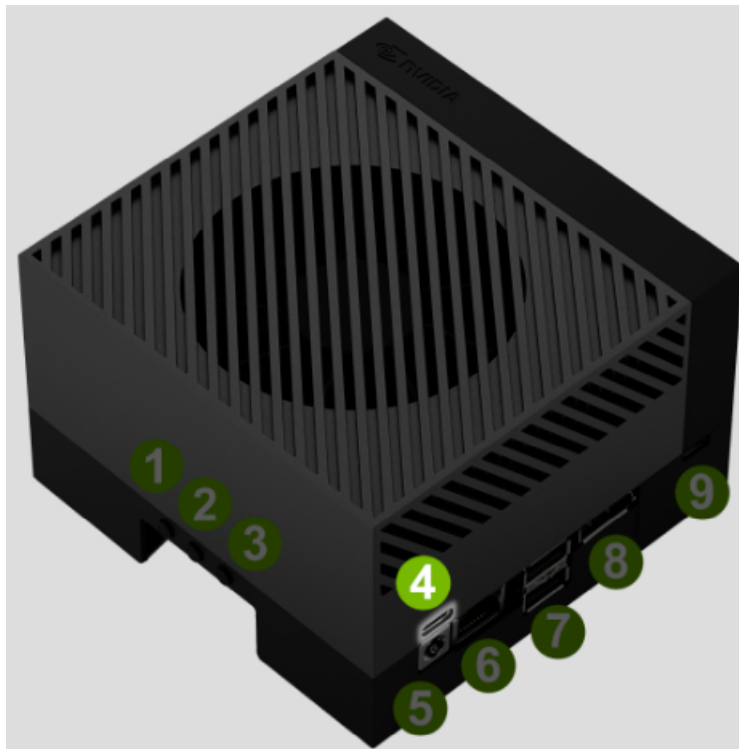


Рисунок 4 – Подключение питания

- Дождаться автоматического включения устройства и загорания белого светодиода (0) (рисунок 5);

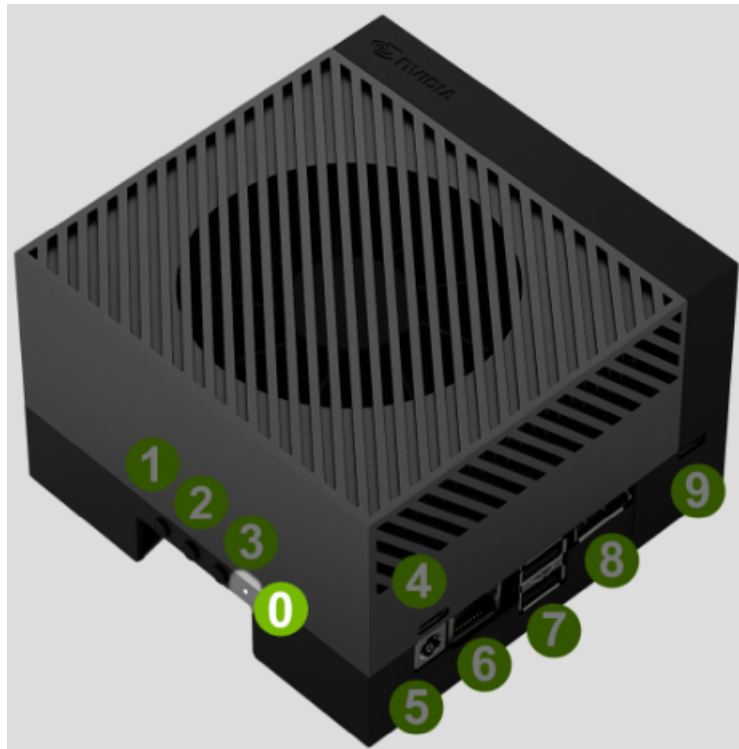


Рисунок 5 – Включение устройства

- Если светодиод (0) не загорелся, нажать на кнопку питания (1) (рисунок 6);

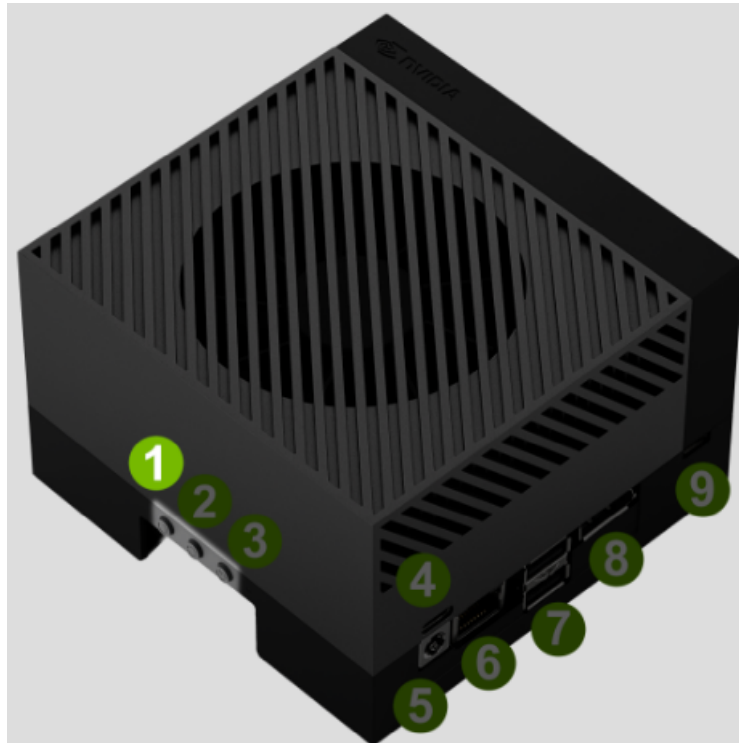


Рисунок 6 – Кнопка питания

- Дождаться загрузки Ubuntu (время ожидания около 1 минуты) после чего ознакомиться и согласиться с EULA, выбрать системный

язык и часовой пояс, задать имя компьютера, имя и пароль пользователя, а также сконфигурировать беспроводную сеть;

- После настроек перезагрузить устройство, дождаться загрузки стартовой страницы (рисунок 7) и перейти к установке Jetpack:

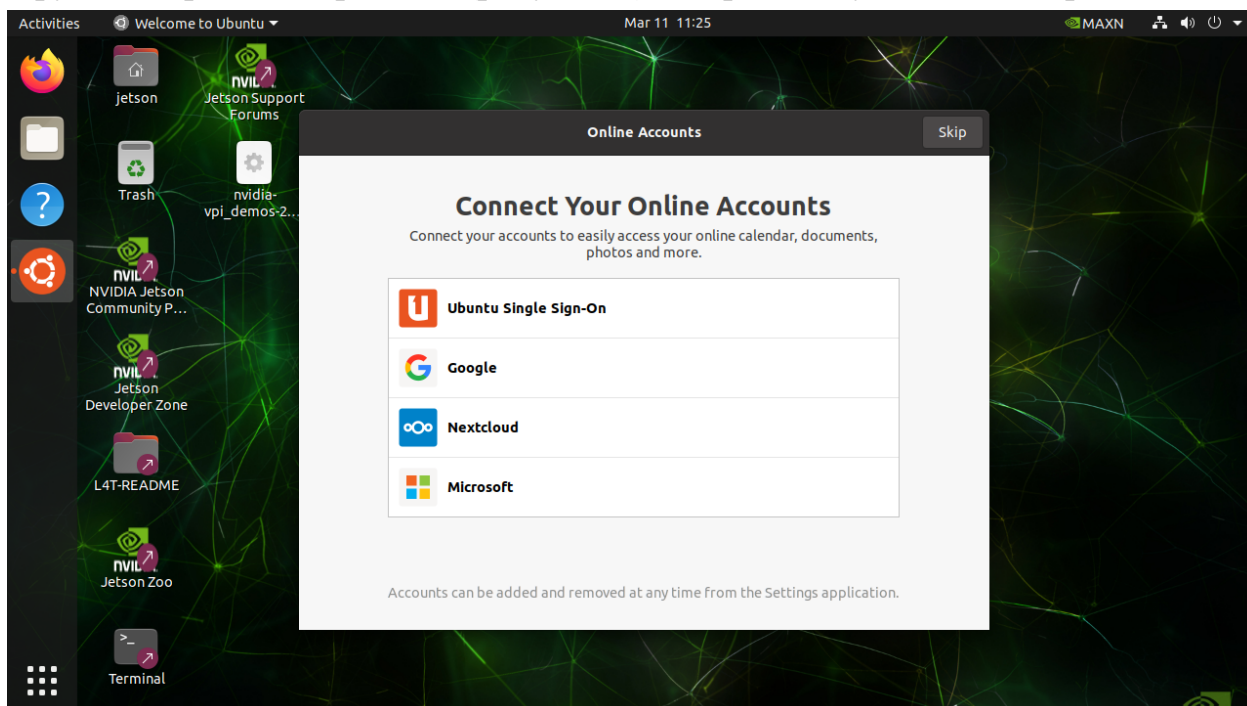


Рисунок 7 – Стартовая страница Ubuntu

- Если установка осуществляется с внешнего компьютера:
- Соединить USB кабелем внешний компьютер и устройство через USB порт, расположенный рядом с 40-контактным портом (10) (рисунок 8):

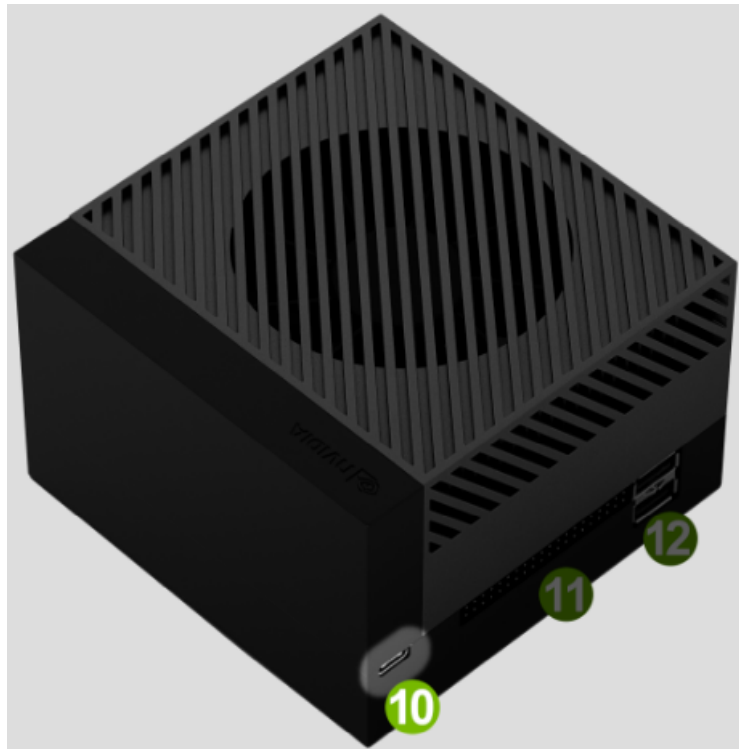


Рисунок 8 – Подключение к внешнему компьютеру

- При необходимости подключить к Jetson Ethernet-кабель (6) (рисунок 9);



Рисунок 9 – Подключение Ethernet-кабеля

- Убедиться, что ничего не подключено к DisplayPort;

- Подключить кабель питания из комплекта в USB-порт, находящийся над портом питания (4) (рисунок 10);

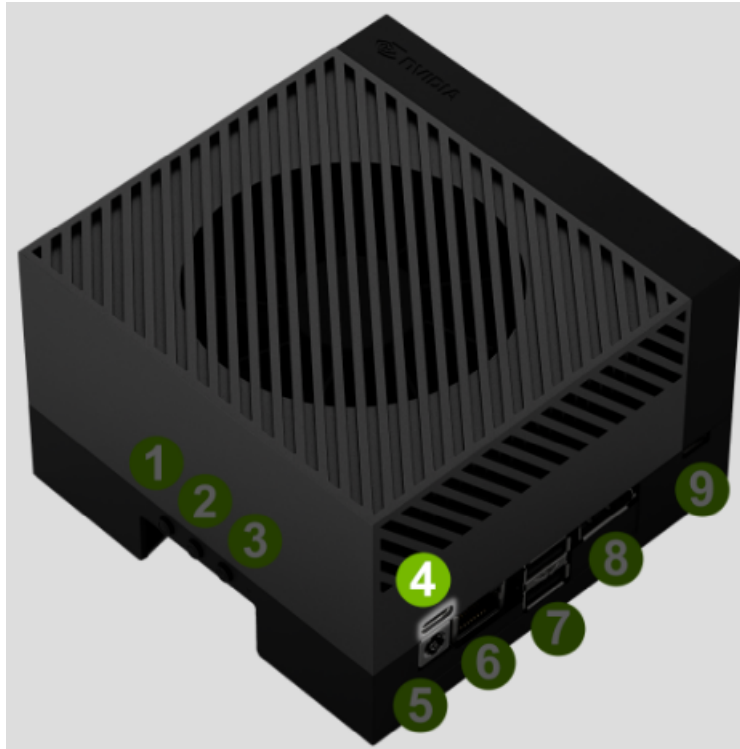


Рисунок 10 – Подключение питания

- Дождаться автоматического включения устройства и загорания белого светодиода (0) (рисунок 11);

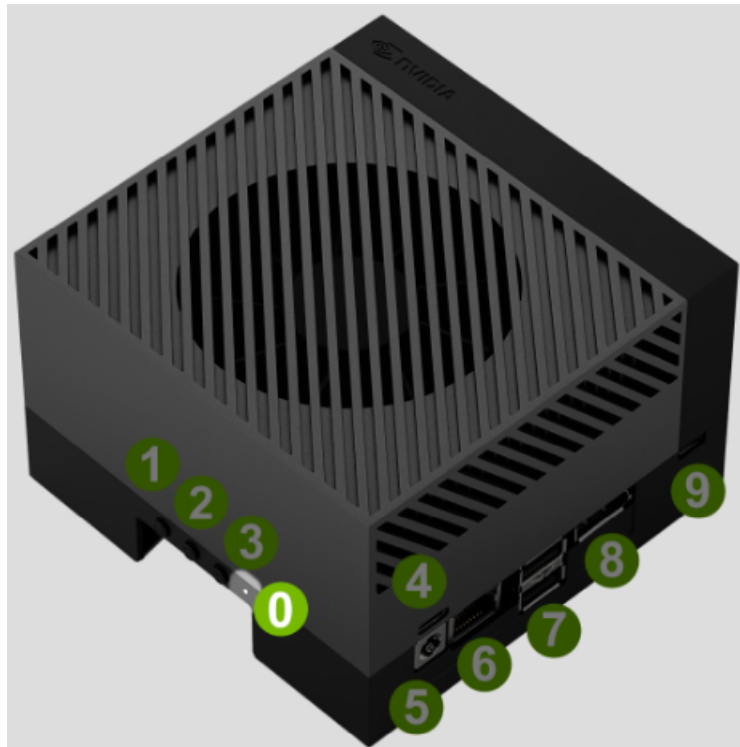


Рисунок 11 – Включение устройства

- Если светодиод (0) не загорелся, нажать на кнопку питания (1) (рисунок 12);



Рисунок 12 – Кнопка питания

- Подождать 1 минуту и выполнить подключение через внешний компьютер:

1) Для Windows:

1) Открыть оснастку Диспетчер устройств, перейти к параметрам USB Serial Device в группе Порты (COM & LPT) и убедиться, что VID и PID равны 0955 и 7020 (рисунок 13):

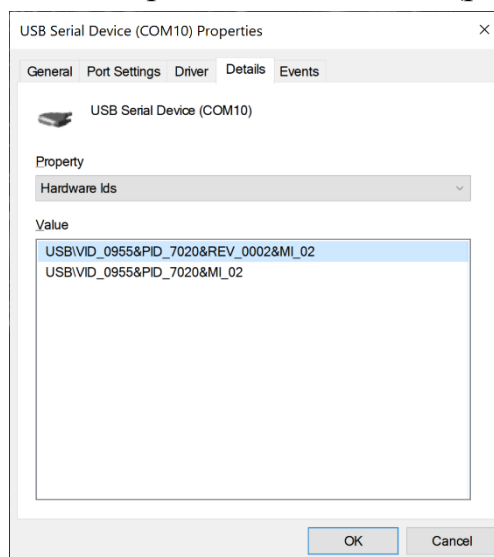


Рисунок 13 – Параметры USB Serial Device

2) Открыть PuTTY или аналогичное ПО и создать новое подключение с типом Serial и скоростью 115200 (рисунок 14):

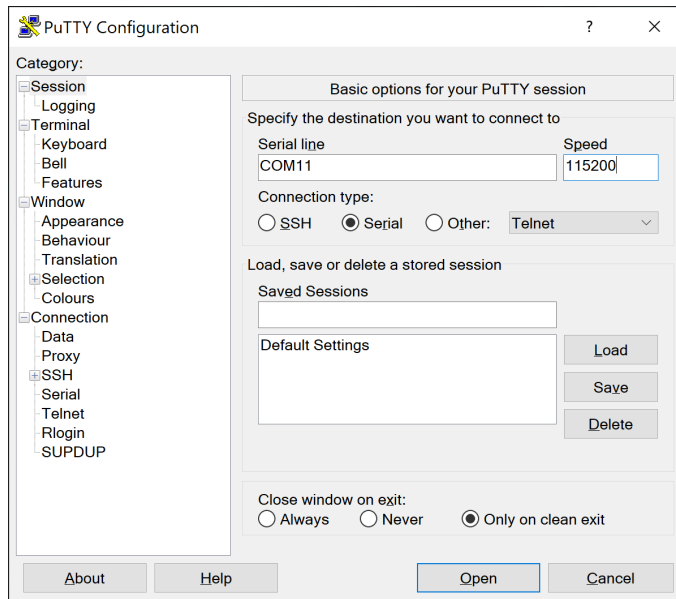


Рисунок 14 – Создание нового подключения к устройству

3) Выполнить подключение и последующую конфигурацию устройства (рисунок 15):



Рисунок 15 – Конфигурация устройства

2) Для Linux:

1) Перед установкой проверить какие Serial устройства определены на внешнем компьютере, выполнив команду:

```
dmesg | grep --color 'tty'
```

2) Соединить USB кабелем внешний компьютер и устройство через USB порт, расположенный рядом с 40-контактным портом и повторно выполнить команду для определения порта, используемого устройством:


```
~$ dmesg | grep --color 'tty'
...
...
[xxxxxxx.xxxxxx] cdc_acm 1-5:1.2: ttyACM0: USB ACM device
```

3) Установить пакет screen, выполнив команду:

```
sudo apt-get install -y screen
```

4) Выполнить команду с определенным ранее портом и последующую конфигурацию устройства:

```
sudo screen /dev/ttyACM0 115200
```

2) Выполнить установку JetPack:

- Проверить версию L4T, выполнив команду:

```
cat /etc/nv_tegra_release
```

- Если версия старше 34, выполнить команды:

```
sudo bash -c 'echo "deb
https://repo.download.nvidia.com/jetson/common r34.1 main" >>
/etc/apt/sources.list.d/nvidia-l4t-apt-source.list'
sudo bash -c 'echo "deb
https://repo.download.nvidia.com/jetson/t234 r34.1 main" >>
/etc/apt/sources.list.d/nvidia-l4t-apt-source.list'
```

- Выполнить установку, выполнив команды:

```
sudo apt update
sudo apt dist-upgrade
sudo reboot
sudo apt install nvidia-jetpack
```

Установка может занять около часа (в зависимости от скорости интернет-соединения).

Для последующей настройки окружения необходимо выполнить следующие шаги:

1) Установить пакеты git и pkg-config, выполнив команды:

```
sudo apt install git
sudo apt install pkg-config
```

3) Установить и настроить Python, выполнив команды:

```
sudo apt install python3.9-dev
sudo apt install curl
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python3.9 -m get-pip
python3.9 -m pip install torch
python3.9 -m pip install opencv-python
python3.9 -m pip install pandas
python3.9 -m pip install IPython
python3.9 -m pip install psutil
python3.9 -m pip install torchvision
python3.9 -m pip install tqdm
python3.9 -m pip install matplotlib
python3.9 -m pip install seaborn
python3.9 -m pip install pyyaml
python3.9 -m pip install --upgrade scipy
python3.9 -m pip install matplotlib --force-reinstall
python3.9 -m pip install norfair
python3.9 -m pip install scikit-image
python3.9 -m pip install wheel
python3.9 -m pip install shapely
python3.9 -m pip install -U Pillow
python3.9 -m pip install
tensorrt-8.2.1.9-cp39-none-linux_aarch64.whl
python3.9 -m pip install
tensorrt-8.2.1.9-cp39-none-linux_aarch64.whl
python3.9 -c 'import tensorrt; print("TensorRT version:
{}").format(tensorrt.version)'
```

4) Установить и настроить Kitware, выполнив команды

```

sudo apt update && sudo apt install -y software-properties-common
lsb-release && sudo apt clean all
wget -O - https://apt.kitware.com/keys/kitware-archive-latest.asc
2>/dev/null | gpg --dearmor - | sudo tee
/etc/apt/trusted.gpg.d/kitware.gpg >/dev/null
sudo apt-add-repository "deb https://apt.kitware.com/ubuntu/
$(lsb_release -cs) main"
sudo apt update
sudo apt install kitware-archive-keyring
sudo rm /etc/apt/trusted.gpg.d/kitware.gpg
sudo apt update

```

После выполнения команды в консоли должен появиться ключ, который необходимо использовать вместо KEY:TAG перед выполнением следующей команды:

```

sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
KEY:TAG

```

5) Установить и настроить CMake, выполнив команды:

```

sudo apt update
sudo apt install cmake

```

6) Установить и настроить библиотеки C++, выполнив команды:

```

sudo apt install libyaml-cpp-dev
sudo apt install libcrypto++-dev
sudo apt install libssl-dev
sudo apt-get install -y udev
sudo apt-get install libgtk-3-dev
sudo apt-get install libegl-mesa0

```

7) Установить и настроить CUDA, выполнив команды:

```

echo "export PATH=\${PATH}:/usr/local/cuda-11.4/bin/" >>
~/.bashrc
source ~/.bashrc

```

```

export
CPATH=$CPATH:/usr/local/cuda-11.4/targets/aarch64-linux/include
export
LIBRARY_PATH=$LIBRARY_PATH:/usr/local/cuda-11.4/targets/aarch64-l
inux/lib
python3.9 -m pip install pycuda
echo "export
LD_PRELOAD=${LD_PRELOAD}:/usr/lib/aarch64-linux-gnu/libgomp.so.1
" >> ~/.bashrc
source ~/.bashrc

```

8) Установить и настроить AWS, выполнив команды:

```

git clone https://github.com/aws/aws-sdk-cpp.git
cd aws-sdk-cpp
git checkout main
git pull origin main
git submodule update --init --recursive
mkdir sdk_build
cd sdk_build
sudo apt update && sudo apt upgrade && sudo apt install curl &&
sudo apt-get install libcurl4-openssl-dev
cmake ../ -DCMAKE_BUILD_TYPE=Debug
-DCMAKE_PREFIX_PATH=/usr/local/
-DCMAKE_INSTALL_PREFIX=/usr/local/ -DBUILD_ONLY="s3"
-DBUILD_SHARED_LIBS="OFF"
make
sudo make install

```

Для настройки режима работы Jetson необходимо выполнить следующие шаги:

1) Перед первым запуском необходимо сгенерировать файл с доступами к хранилищу `credentials.json`:

- Открыть вкладку «access keys» (рисунок 16):

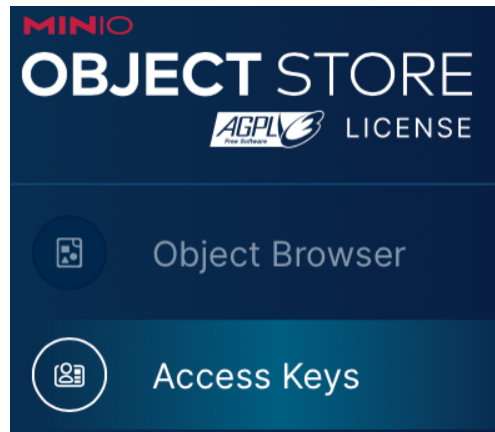


Рисунок 16 – Переход к вкладке «access keys»

- На открывшейся вкладке нажать на кнопку «Create access key»;
- Для формирования файла с доступами к хранилищу нажать на кнопку «Create» (рисунок 17);

 A screenshot of the 'Create Access Key' form. It features a key icon and the title 'Create Access Key'. There are two input fields: 'Access Key' containing the text 'K00vEt2qtu4LJ4US' and 'Secret Key' which is masked with dots. Below these fields is a toggle switch for 'Restrict beyond user policy' currently set to 'OFF'. At the bottom, there are 'Clear' and 'Create' buttons.

Рисунок 17 – Формирование файла с доступами к хранилищу

- Для выгрузки файла с доступами к хранилищу нажать на кнопку «Download for import» (рисунок 18);

 A screenshot of a notification dialog titled 'New Access Key Created'. It contains the text: 'A new Access Key has been created with the following details:'. Below this, there are two text boxes: 'Access Key:' with the value 'K00vEt2qtu4LJ4US' and 'Secret Key:' with the value 'H86ZuEHYcd9CGGVpeQFL6x9XYhUMptas'. A red warning icon and text state: 'Write these down, as this is the only time the secret will be displayed.' At the bottom right, there is a 'Download for import' button with a download icon.

Рисунок 18 – Выгрузка файла с доступами к хранилищу

- Сформированный файл credentials.json разместить в папке hardware_bin

2) Настроить следующие параметры, отредактировав файл config.json:

- Масштабирование видео для передачи кадра в ML;

```
"cropParams": {
  "left": 550,
  "right": 15,
  "top": 0,
  "bottom": 0
},
```

- Адрес локального сервера для взаимодействия с Backend;

```
"localServerParams": {
  "ip": "127.0.0.1",
  "port": 1234
},
```

- Параметры сервиса для автозапуска ПО;

```
"localServiceParams": {
  "ip": "127.0.0.1",
  "port": 1236
},
```

- Название корзины хранилища для выгрузки видеозаписей исследований;

```
"videosBucketName" : "studies",
```

- Название корзины хранилища для выгрузки “скриншотов” исследований;

```
"imagesBucketName" : "protocol-images",
```

- Разрешение входного видеопотока;

```
"cameraParams": {
  "inputWidth": 1920,
  "inputHeight": 1080
},
```

- Текущий режим работы: «sync» для работы в режиме реального времени с подключенным колоноскопом или «async» для асинхронной обработки видеофайла;

```
"defaultMode": "sync",
```

- Офлайн режим: «false» - выключено, «true» - включено. При включенном офлайн режиме видео и снимки врача сохраняются на диск, координаты рамок записываются в текстовый файл в формате, аналогичном передаваемому на Backend. По умолчанию запись происходит в папку hardware_bin/recordings/. Для каждого исследования создается отдельная папка (название является датой создания исследования) со всеми данными;

```
"isOnline": false,
```

- Путь до видео в офлайн режиме. Для изменения пути до записей в офлайн режиме, необходимо задать путь до папки. По данному пути создастся recordings/. Для каждого исследования создается отдельная папка (название является датой создания исследования) со всеми данными;

```
"videoDestPath" : "/media/user-orin/Transcend/123/",
```

- Отображение рамок в записанном видеофайле: «false» - выключено, «true» - включено;

```
"isBoxesInRecording": false,
```

3) При необходимости использования инструментов отладки настроить следующие параметры, отредактировав файл config.json:

- Показ FPS при работе: «false» - выключено, «true» - включено (отладочный функционал);

```
"showFPS": true,
```

- Использование предсказаний от ML: «false» - выключено, «true» - включено (тестирование производительности);

```
"isMLEnabled": true,
```

- Логирование таймаутов на ресурсоемких участках кода: «false» - выключено, «true» - включено (тестирование производительности);

```
"isDebugLogsEnabled": true,
```

- Частота считывания и выдачи кадров;

```
"FPS" : 30,
```

- Режим interlance: «false» - выключено, «true» - включено;

```
"interlanceMode": false,
```

- Настройка изображения. Указанные параметры стандартные - картинка без изменений;

```
"videobalance": {
  "saturation": "1.0",
  "contrast": "1.0",
  "brightness": "0.0",
  "hue": "0.0"
},
```

- Входной формат:

- AYUV (packed 4:4:4 YUV with alpha channel);
- ARGB (RGB with alpha channel first);
- BGRA (reverse RGB with alpha channel last);
- ABGR (reverse RGB with alpha channel first);
- RGBA (RGB with alpha channel last);
- Y444 (planar 4:4:4 YUV);
- xRGB (sparse RGB packed into 32 bit, space first);
- RGBx (sparse RGB packed into 32 bit, space last);
- xBGR (sparse reverse RGB packed into 32 bit, space first);
- BGRx (sparse reverse RGB packed into 32 bit, space last);
- RGB (RGB);
- BGR (reverse RGB);
- Y42B (planar 4:2:2 YUV);
- YUY2 (packed 4:2:2 YUV);
- UYVY (packed 4:2:2 YUV);
- YVYU (packed 4:2:2 YUV Same as "YUY2" but with U and V planes swapped);
- I420 (planar 4:2:0 YUV);

- YV12 (planar 4:2:0 YUV Same as I420 but with U and V planes swapped);

- IYUV (packed 4:4:4 YUV, U-Y-V order);
- Y41B (planar 4:1:1 YUV);
- NV12 (planar 4:2:0 YUV with interleaved UV plane);
- NV21 (planar 4:2:0 YUV with interleaved VU plane);

```
"inputFormat": "YUY2",
```

- Выходной формат (возможные значения аналогичны предыдущему параметру. Лучше оставлять значение «BGR»;

```
"outputFormat": "BGR",
```

- Насыщенность видео: «jpeg», «mpeg2», «dv», «none». Если строка пустая данное поле не добавится в пайплайн;

```
"chroma-site": "",
```

- Колориметрия: «bt601», «bt709», «smpte240m». Если строка пустая данное поле не добавится в пайплайн;

```
"colorimetry": ""
```

Для запуска ПО необходимо перейти в папку hardware_bin и запустить файл artincol_fw.

Для запуска видео при эмуляции видеопотока на Jetson Nano необходимо из консоли запустить следующие команды:

```
sudo systemctl stop gdm
sudo loginctl terminate-seat seat0
gst-launch-1.0 filesrc location=~/.example.mp4 ! decodebin !
nvdrmvideosink -e
```

Для выхода в стандартный режим:

```
sudo systemctl start gdm
```

3.2 ML-компоненты

Для развертывания ML-компонентов необходимо выполнить следующую последовательность действий:

1) Клонировать проект:

```
git clone --depth 1 --branch 0.1.3
https://bitbucket.org/waveaccessllc/artincol-ml cd artincol-ml
```

4) Установить Python:

```
sudo apt-get install python3.9-dev -y
sudo apt-get install python3-pip -y
```

5) Создать и активировать виртуальное окружение (опционально):

```
sudo apt-get install
python3.9-venv python3.9 -m venv venv
source venv/bin/activate
```

6) Установить зависимости:

```
export PATH=/usr/local/cuda-11.4/bin:$PATH
export
LD_LIBRARY_PATH=/usr/local/cuda-11.4/lib64:$LD_LIBRARY_PATH
sudo apt-get install ffmpeg -y
python3.9 -m pip install -r requirements.txt
```

7) Загрузить wheel tensorrt на Jetson в ~/ и установить:

```
python3.9 -m pip install
../tensorrt-8.2.1.9-cp39-none-linux_aarch64.whl
```

8) Загрузить веса модели:

```
sudo apt-get install git-lfs -y
git-lfs pull
```

9) Конвертировать модель в engine:

```
python3.9
import os
from artincol_ml.export import export_engine

PROJECT_ROOT = "/home/user-orin/artincol-ml"
```

```

onnx_fpath = os.path.join(PROJECT_ROOT, "checkpoints",
"best_detection.onnx")
output_fpath = os.path.join(PROJECT_ROOT, "checkpoints",
"best_detection.engine")

export_engine(onnx_fpath, output_fpath)

```

10) Проверить работоспособность:

- установить pytest:

```
python3.9 -m pip install pytest
```

- ВЫПОЛНИТЬ ТЕСТЫ

```
python3.9 -m pytest tests
```

11) Для проверки в рабочем режиме необходимо загрузить на устройство тестовое видео и выполнить запуск:

```
python3.9 interface/ml_interface.py -f /path/to/video
```

Пример логов успешной отработки:

```

user-orin@userorin-desktop:~/artincol-ml$ python3.9
interface/ml_interface.py -f /home/user-orin/artincol-ml/10.1.mp4
2023-05-18 13:48:41,639 - ML_logger - WARNING - detection_model
is not initialized.
2023-05-18 13:48:41,639 - DetectionRtInference - INFO - Batch
size is configured: 1
2023-05-18 13:48:41,777 - DetectionRtInference - INFO -
CUDA_MODULE_LOADING=LAZY is enabled.
[TensorRT] INFO: [MemUsageChange] Init CUDA: CPU +213, GPU +0,
now: CPU 336, GPU 6312 (MiB)
[TensorRT] INFO: Loaded engine size: 29 MiB
[TensorRT] INFO: [MemUsageChange] Init cuDNN: CPU +620, GPU +587,
now: CPU 1017, GPU 6956 (MiB)
[TensorRT] INFO: [MemUsageChange] TensorRT-managed allocation in
engine deserialization: CPU +0, GPU +27, now: CPU 0, GPU 27 (MiB)

```

```
2023-05-18 13:48:43,554 - DetectionRtInference - INFO - Model
/home/user-orin/artincol-ml/checkpoints/best_detection.engine is
loaded successfully.
[TensorRT] INFO: [MemUsageChange] Init cuDNN: CPU +1, GPU +0,
now: CPU 988, GPU 6930 (MiB)
[TensorRT] INFO: [MemUsageChange] TensorRT-managed allocation in
IExecutionContext creation: CPU +0, GPU +34, now: CPU 0, GPU 61
(MiB)
2023-05-18 13:48:43,600 - ML_logger - INFO - Postprocessing is
enabled: True
703it [00:24, 28.44it/s]
2023-05-18 13:49:09,304 - ML_logger - INFO - FPS (prediction
only): 38.194161857264355
```

4 Контактная информация

Фактический адрес размещения инфраструктуры разработки:

СПб, Каменноостровский пр., д. 11, корп. 2, литер А
индекс 197046

Фактический адрес размещения разработчиков:

СПб, Каменноостровский пр., д. 11, корп. 2, литер А
индекс 197046

Фактический адрес размещения службы поддержки:

СПб, Каменноостровский пр., д. 11, корп. 2, литер А
индекс 197046

Email: Support@alnisoft.ru

Адрес веб-сайта: _____