

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

«ПАУК Видеоаналитика»

ООО «НПО «ИТС СОФТ»

«__»_____2026 г.

МОСКВА

2026

БАЗОВОЕ СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Системные программные средства, на которых обеспечивается эффективная работа ПО:

- ❑ процессорный модуль Jetson Orin Nano;
- ❑ процессор – шестиядерный 64-разрядный процессор NVIDIA Carmel с архитектурой ARM®v8.2 6 МБ L2 и 4 МБ L3
- ❑ графический процессор – архитектура NVIDIA Volta™ с 384 ядрами NVIDIA CUDA® и 48 тензорными ядрами
- ❑ оперативная память – 8 ГБ LPDDR4x 128-бит, 59,7 Гбит/с
- ❑ операционная система – RED OS 8.0

Инструкция по установке RedOS на Jetson

1. Записать образ redos8-custom.iso на USB носитель в режиме dd

Linux:

```
sudo dd if=./redos8-custom.iso of=/dev/sdb bs=1M status=progress && sync && sudo eject /dev/sdb
```

где ./redos8-custom.iso — путь к образу, /dev/sdb — путь к usb-носителю.

Windows:

Использовать программу Ventoy, создав загрузочный носитель и записав туда образ.

2. Распаковать архив Jetson_Linux_R36.4.4_aarch64.tbz2 на ПК с Linux (путь к директории распаковки архива не должен содержать кириллических символов и пробелов).
3. Подключить Jetson к ПК.
4. Запустить Jetson в recovery mode, замкнув перемычкой два контакта.
5. Прошить QSPI с помощью команды:

```
sudo bash ./flash.sh p3768-0000-p3767-0000-a0-qspi external
```

6. После прошивки подключить USB накопитель с RedOS к Jetson
7. Включить Jetson и нажимать F11 до появления меню выбора загрузчика
8. Выбрать USB накопитель **!!! Все данные будут удалены с Jetson !!!!**
9. Установка производится в автоматическом режиме. Форматируется носитель nvme, создаются разделы, пользователи, устанавливаются необходимые пакеты.
10. После установки будет доступен пользователь user с паролем 1111. Аналогичный пароль у суперпользователя root.
11. Если установка производилась без подключения к сети, то при первой загрузке необходимо её настроить. Для этого нужно получить список подключений и включить желаемое:

```
nmcli connection show
```

```
nmcli connection modify <имя_подключения> ifname <имя_подключения>
```

```
nmcli connection up <имя_подключения>
```

Инструкция по установке

1. Подключиться к Jetson по SSH.
2. Добавьте пользователя в группу Docker и перезагрузите устройство.

```
$ sudo usermod -aG docker user
```

```
$ sudo reboot
```

3. Загрузить на Jetson Docker-образы и веса нейросетей:
4. Разархивировать загруженный архив на Jetson.

```
$ unzip Local-VA.zip
```

5. Загрузить все распакованные докер-образы в системный докер:

```
$ docker load -i <путь до файла Docker образа>
```

Пример:

```
$ docker load -i ./Local-VA /images/cartrack4.tar.gz
```

```
$ docker load -i ./Local-VA /images/cartrack25-fapi.tar
```

```
$ docker load -i ./Local-VA /images/nginx_car.tar
```

6. Удалить все скаченные докер-образы (опционально):

```
$ rm <docker image file>
```

Пример удаления всех образов:

```
$ rm -rf ./Local-VA /images
```

7. Скопировать ПО на Jetson или клонировать с Git-репозитория:

```
$ git clone <ссылка на Git-репозиторий> <путь для расположения файлов>
```

8. Перейти в директорию проекта (путь, указанный в прошлом пункте):

```
$ cd <путь для расположения файлов>
```

9. Перенести веса в директорию ./cartrack/weights/

Пример:

```
$ cp ./Local-VA/weights ./cartrack/weights
```

10. Перед первым запуском внести желаемый логин и пароль от веб-интерфейса в файл ./env. Его дальнейшая замена возможна только из базы данных. Пример файла:

```
$ cat .env
```

```
ADMIN_USER = admin
```

```
ADMIN_PASSWORD = password
```

11. Рекомендуется настроить подключение к NTP-серверу (подробнее в разделе «Настройка подключения к NTP-серверу»).

12. Запустить ПО командой: `docker-compose up -d`

Настройка подключения к серверу точного времени (NTP)

1. Включить синхронизацию времени в `timedatectl` на случай, если она отключена:

```
sudo timedatectl set-ntp true
```

2. Прописать NTP сервер (для примера используется адрес сервера 91.207.136.55. Стоит обратить внимание, что между адресом и \1 нет и не должно быть пробела):

```
sudo sed -i -E 's/^\#*(NTP=).*\1<Адрес сервера>/g' /etc/systemd/timesyncd.conf
```

Пример:

```
sudo sed -i -E 's/^\#*(NTP=).*\191.207.136.55/g' /etc/systemd/timesyncd.conf
```

3. Запустить сервис синхронизации на случай, если он отключен:

```
sudo systemctl enable --now systemd-timesyncd.service
```

4. Перезапустить сервис синхронизации:

```
sudo systemctl restart systemd-timesyncd.service
```

5. Проверить синхронизацию:

```
timedatectl timesync-status
```

Результат:

```
Server: 91.207.136.55 (91.207.136.55)
Poll interval: 34min 8s (min: 32s; max 34min 8s)

    Leap: normal
    Version: 4
    Stratum: 2
    Reference: 82AD5B3A
    Precision: 1us (-24)
Root distance: 44.959ms (max: 5s)

    Offset: +1.929ms
    Delay: 26.386ms
    Jitter: 3.297ms

Packet count: 127

Frequency: +39.452ppm
```

После «Server:» должен быть адрес, указанный на 2 шаге.

Инструкция по обновлению программного обеспечения

Для текущей версии ПО существует 3 различных сценария обновления. Выбор способа обновления зависит от того, какие файлы были обновлены. Каждый последующий способ полностью заменяем предыдущим, но они требуют меньшее количество команд и перезапусков обработки.

Обновлен файл `docker-compose.yml` (данный способ можно использовать при любом обновлении):

1. Перейти в директорию проекта (по умолчанию название директории `vd-main-service`):
`cd <Путь>`
2. Остановить все контейнеры:
`docker-compose down`
3. Обновить код (через Git или замену файлов):
`git pull <ссылка на репозиторий>`
4. Запустить контейнеры:
`docker-compose up -d`

Обновлен файл из директории `./flaskapi` (данный способ можно использовать при любом обновлении, кроме случая, описанного выше):

1. Перейти в директорию проекта (по умолчанию название директории `vd-main-service`):
`cd <Путь>`
2. Обновить код (через git или замену файлов):
`git pull <ссылка на репозиторий>`
3. Перезапустить контейнеры:
`docker-compose restart`

Остальные изменения (данный способ можно использовать при любом обновлении, кроме случаев, описанных выше):

1. Перейти в директорию проекта (по умолчанию название директории `vd-main-service`):
`cd <Путь>`
2. Обновить код (через Git или замену файлов):
`git pull <ссылка на репозиторий>`
3. Перезапустить обработку через веб-интерфейс.

Инструкция по обновлению весов нейронных сетей и Docker-образов

Обновление весов нейросетей:

1. Загрузить новые веса на Jetson.
2. Перейти в директорию ПО, где хранятся веса:

```
cd <Путь_до_ПО>/cartrack/
```

3. (Опционально) Удалить старые веса для уменьшения занимаемого дискового пространства:

```
sudo rm -rf ./weights
```

4. Переместить новые веса (папку weights) в <путь_до_ПО>/cartrack/:

```
cp -r <путь_до_папки_weights_с_новыми_весами> ./weights/
```

5. Перезапустить обработку через веб-интерфейс.

Обновление Docker-образов:

1. Загрузить новые образы на Jetson.
2. Удалить все Docker-контейнеры:

```
docker rm -v -f $(docker ps -qa)
```

3. Удалить все Docker-образы:

```
docker rmi -f $(docker images -aq)
```

4. Поочередно загрузить Docker-образы в систему:

```
docker load -i <путь_до_файла_с_образом>
```

5. (Опционально) Удалить файлы с образами для уменьшения занимаемого дискового пространства:

```
rm <путь_до_файла_с_образом>
```

6. Перезапустить обработку через веб-интерфейс.