

Программное обеспечение
ПУИДД-Трафик
Руководство по установке экземпляра ПО
Листов 8

Москва 2026 г.

Содержание

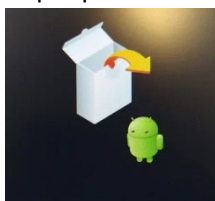
1. Введение	3
2. Установка актуальной версии RED OS	3
3. Установка софта и настройка системы.	4

1. Введение

Настоящий документ содержит описание по установке операционной системы и программного обеспечения «ПУИДД-Трафик» (далее ПО) на специализированное аппаратное обеспечение.

2. Установка актуальной версии RED OS

1. Убеждаемся, что на устройстве установлен Android или Ubuntu и оно запускается.
2. Вставляем настроечную SD-карту в устройство.
3. Перезагружаем его.
4. Устройство должно уйти в режим прошивки. При этом световой индикатор на устройстве начнёт мигать красным цветом. Также мы должны увидеть в центре экрана иконку процесса перепрошивки:



В окне наблюдаем прогресс перепрошивки:

```
Recovery start.....
start sdupdate.update_package = /mnt/sdcard/sdupdate.img
parameter writing...trust writing...uboot writing...boot writing...misc writing...recovery writing...
```

5. Дожидаемся, пока процесс не будет закончен – обычно занимает 2-3 минуты. Затем в тексте должна появиться строка
«Please remove SD CARD!!!, wait for reboot.»

как на фото ниже:

```
Recovery start.....
start sdupdate.update_package = /mnt/sdcard/sdupdate.img
parameter writing...trust writing...uboot writing...boot writing...misc writing...
ing...rootfs writing...parameter checking...trust checking...uboot checking...
isc checking...recovery checking...rootfs checking...update.img Installation s
Please remove SD CARD!!!, wait for reboot.
```

6. Извлекаем SD-карту из устройства. Устройство при этом автоматически уйдёт в перезагрузку.
7. После перезагрузки устройств войдёт в режим восстановления, индикатор при этом начнёт светиться ровным красным цветом, в центре экрана снова будет та же иконка, а слева текст



В данном режиме устройство будет около 5 минут, а затем перезагрузится.

8. В результате успешной установки RED OS после перезагрузки на экране мы увидим консоль

RED OS с уже залогиненным в неё пользователем “root”:

```
firefly login: root (automatic login)

Firefly

Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 4.4.194+ aarch64)

 * Documentation:  http://wiki.t-firefly.com
 * Management:    http://www.t-firefly.com

System information as of Tue Jan 24 16:19:44 UTC 2023

System load:  0.24 0.05 0.02   Up time:    0 min
Memory usage: 2 % of 3841MB   IP:
Usage of /:   1% of 28G

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@firefly:~#
```

3. Установка софта и настройка системы.

Как только устройство загрузится, можно приступить к дальнейшим действиям:

1. Снова вставляем SD-карту в устройство.

2. В консоли выполняем команду:

```
mount /dev/mmcblk0p1 /mnt
```

3. Не должно появиться никаких ошибок, а повторный ввод той же команды должен привести к следующему сообщению:

```
root@firefly:~# mount /dev/mmcblk0p1 /mnt/
mount: /mnt: /dev/mmcblk0p1 already mounted on /mnt.
```

4. Если всё так и есть, как описано выше, в консоли выполняем следующую команду:
`/mnt/install/install.sh`

(* для упрощения ввода можно воспользоваться автозаполнением через клавишу TAB, чтобы не вводить полностью весь текст – вводить следующее ‘+TAB’ – обозначает нажатие на клавишу TAB:
`/mn +TAB i +TAB + i`
)

5. Результатом будет вывод на экран меню настройки:

```
Main menu:
1) Install all soft
2) Network setup
3) Time setup
4) Disable autologin and set root password
5) Setup radar adapters (must be run after all soft installed)
6) Exit
```

Скрипт установки запускается путём выполнения

```
/mnt/install/install.sh
```

Выбор пунктов меню осуществляется путём выбора соответствующих цифр и последующего нажатия клавиши Enter.

Главное меню скрипта предлагает на выбор один из 6 пунктов:

1. Install all soft
2. Network setup

3. Time setup
4. Disable autologin and set root password
5. Setup radar adapters (must be run after all soft installed)
6. Exit

Подробнее о каждом из них.

1. **Install all soft** – данный пункт выполняет копирование необходимых пакетов из каталога install в соответствующие системные каталоги и выполняет их установку. Также он копирует файлы приложений из папки install/git в папку /opt и настраивает соответствующие сервисы.

Подпунктов нет, выбор этого пункта сразу запускает действия, прогресс которых сразу виден на экране. Установка занимает порядка 5 минут. По завершению экран очищается и появляется сообщение «All done», после чего происходит выход из скрипта – для выполнения других пунктов снова запустите скрипт.

```
All done
root@firefly:~# |
```

Проверить статус всех установленных скриптом служб можно, запустив в консоли команду:
`systemctl status config.service auth.service dwh.service radar.service`

Результатом будет картина вроде такой:

```
● config.service - ConfigService
   Loaded: loaded (/etc/systemd/system/config.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-01-25 16:10:57 UTC; 1 years 11 months left
     Main PID: 7385 (run_app)
       Tasks: 3 (limit: 4596)
      Memory: 44.0M
     CGroup: /system.slice/config.service
            └─7385 /bin/bash /opt/ConfigService/run_app
               └─7389 /opt/ConfigService/.venv/bin/python3 /opt/ConfigService/run_app.py

Jan 25 16:11:14 firefly run_app[7389]: 2023-01-25 16:11:14 [INFO] Socket connected: <socket.socket fd=11, family=
Jan 25 16:11:14 firefly run_app[7389]: 2023-01-25 16:11:14 [INFO] Streaming transport linked up: (<pika.adapters.
Jan 25 16:11:14 firefly run_app[7389]: 2023-01-25 16:11:14 [INFO] AMQPConnector - reporting success: <SelectConne
Jan 25 16:11:14 firefly run_app[7389]: 2023-01-25 16:11:14 [INFO] AMQPConnectionWorkflow - reporting success: <Se
Jan 25 16:11:14 firefly run_app[7389]: 2023-01-25 16:11:14 [INFO] Connection workflow succeeded: <SelectConnectio
Jan 25 16:11:14 firefly run_app[7389]: 2023-01-25 16:11:14 [INFO] Created channel=1
Jan 25 16:11:14 firefly run_app[7389]: 2023-01-25 16:11:14 [INFO] 127.0.0.1 - - [25/Jan/2023 16:11:14] "POST /api
Jan 25 16:11:14 firefly run_app[7389]: 2023-01-25 16:11:14 [INFO] make_post_request_with_headers http://localhost
Jan 25 16:11:14 firefly run_app[7389]: 2023-01-25 16:11:14 [ERROR] [Errno 111] Connect call failed ('127.0.0.1',
Jan 25 16:11:14 firefly run_app[7389]: 2023-01-25 16:11:14 [INFO] 127.0.0.1 - - [25/Jan/2023 16:11:14] "PATCH /ap

● auth.service - AuthService
   Loaded: loaded (/etc/systemd/system/auth.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-01-25 16:11:08 UTC; 1 years 11 months left
     Main PID: 7452 (run_app)
       Tasks: 3 (limit: 4596)
      Memory: 42.8M
     CGroup: /system.slice/auth.service
            └─7452 /bin/bash /opt/AuthService/run_app
               └─7456 /opt/AuthService/.venv/bin/python3 /opt/AuthService/run_app.py

Jan 25 16:11:12 firefly run_app[7456]: INFO:werkzeug:Press CTRL+C to quit
Jan 25 16:11:12 firefly run_app[7456]: 2023-01-25 16:11:12 [INFO] Press CTRL+C to quit
Jan 25 16:11:13 firefly run_app[7456]: INFO:werkzeug:127.0.0.1 - - [25/Jan/2023 16:11:13] "POST /api/login HTTP/1
Jan 25 16:11:13 firefly run_app[7456]: 2023-01-25 16:11:13 [INFO] 127.0.0.1 - - [25/Jan/2023 16:11:13] "POST /api
Jan 25 16:11:14 firefly run_app[7456]: INFO:werkzeug:127.0.0.1 - - [25/Jan/2023 16:11:14] "POST /api/validate HTT
Jan 25 16:11:14 firefly run_app[7456]: 2023-01-25 16:11:14 [INFO] 127.0.0.1 - - [25/Jan/2023 16:11:14] "POST /api
Jan 25 16:11:14 firefly run_app[7456]: INFO:werkzeug:127.0.0.1 - - [25/Jan/2023 16:11:14] "POST /api/login HTTP/1
Jan 25 16:11:14 firefly run_app[7456]: 2023-01-25 16:11:14 [INFO] 127.0.0.1 - - [25/Jan/2023 16:11:14] "POST /api
Jan 25 16:11:14 firefly run_app[7456]: INFO:werkzeug:127.0.0.1 - - [25/Jan/2023 16:11:14] "POST /api/validate HTT
Jan 25 16:11:14 firefly run_app[7456]: 2023-01-25 16:11:14 [INFO] 127.0.0.1 - - [25/Jan/2023 16:11:14] "POST /api

● dwh.service - DWHService
   Loaded: loaded (/etc/systemd/system/dwh.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-01-25 16:11:02 UTC; 1 years 11 months left
     Main PID: 7416 (run_app)
       Tasks: 5 (limit: 4596)
      Memory: 44.3M
     CGroup: /system.slice/dwh.service
            └─7416 /bin/bash /opt/DWHService/run_app
               └─7420 /opt/DWHService/.venv/bin/python3 /opt/DWHService/run_app.py

Jan 25 16:11:04 firefly run_app[7420]: INFO:pika.adapters.utils.connection_workflow:AMQPConnector - reporting suc
Jan 25 16:11:04 firefly run_app[7420]: 2023-01-25 16:11:04 [INFO] AMQPConnector - reporting success: <SelectConne
Jan 25 16:11:04 firefly run_app[7420]: INFO:pika.adapters.utils.connection_workflow:AMQPConnectionWorkflow - rep
Jan 25 16:11:04 firefly run_app[7420]: 2023-01-25 16:11:04 [INFO] AMQPConnectionWorkflow - reporting success: <Se
Jan 25 16:11:04 firefly run_app[7420]: INFO:pika.adapters.blocking_connection:Connection workflow succeeded: <Se
Jan 25 16:11:04 firefly run_app[7420]: 2023-01-25 16:11:04 [INFO] Connection workflow succeeded: <SelectConnectio
Jan 25 16:11:04 firefly run_app[7420]: INFO:pika.adapters.blocking_connection:Created channel=1
Jan 25 16:11:04 firefly run_app[7420]: 2023-01-25 16:11:04 [INFO] Created channel=1
Jan 25 16:11:05 firefly run_app[7420]: INFO:root:RabbitMQ listener started.
Jan 25 16:11:05 firefly run_app[7420]: 2023-01-25 16:11:05 [INFO] RabbitMQ listener started.

● radar.service - RadarService
   Loaded: loaded (/etc/systemd/system/radar.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-01-25 16:11:13 UTC; 1 years 11 months left
     Main PID: 7485 (radar-go)
       Tasks: 9 (limit: 4596)
      Memory: 2.6M
     CGroup: /system.slice/radar.service
            └─7485 /opt/radar-go/radar-go
```

При этом, если ваше устройство не подключено к локальной сети и сетевой интерфейс неактивен, сервис radar.service скорее всего не будет в состоянии “Active”, так как он требует наличие работающей сети.

2. **Network setup** – этот пункт предлагает различные варианты настройки сети. Результатом работы каждого из них будет создание файла конфигурации /etc/netplan/00-eth0-config.yaml и перезапуск службы Netplan с обновлёнными параметрами.

```
Choose:
1) Use standard IP Address 192.168.93.2/255.255.255.0 (gateway 192.168.93.1)
2) Manual Input
3) Use dhcp
4) Exit
Input number and press ENTER
|
```

- 1) Use standard IP... - данный пункт устанавливает настройки с указанными в самом пункте параметрами.
- 2) Manual Input – позволяет указать IP-адрес, маску и дефолтный шлюз вручную – при этом адреса и маска сети проверяются на валидность.
- 3) Use dhcp – данный пункт включает получение IP-адреса от службы DHCP в локальной сети.

3. **Time setup** – позволяет изменить настройки времени на сервере

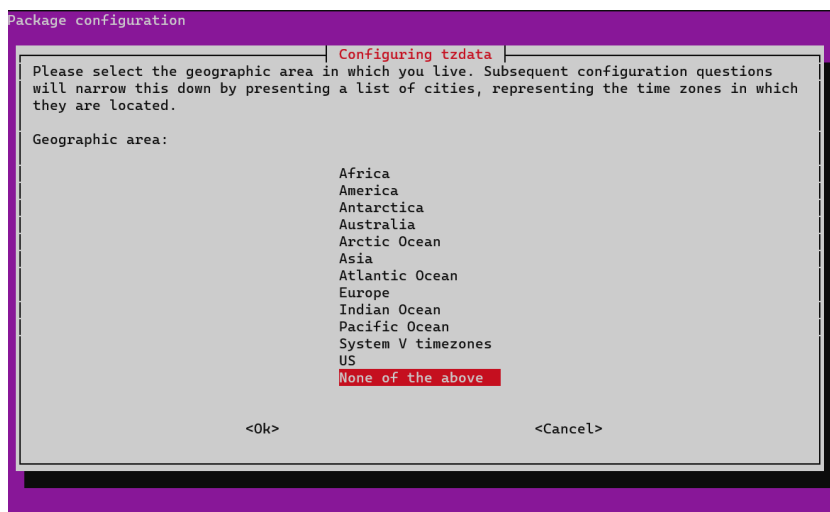
```
Set time settings:
1) Set ntp-server
2) Set date and time manually
3) Change timezone
4) Exit
|
```

- 1) Set ntp-server – предлагает ввести адрес или имя хоста ntp-сервера в локальной или глобальной сети (так как работа ntp-клиента предполагает установку дополнительных пакетов, использовать данный пункт стоит только после того, как был выполнен пункт «1) Install all soft» главного меню). Валидация вводимого адреса или имени не производится, значение записывается «как есть», после чего служба NTP запускается.
- 2) Set date and time manually – ручная установка даты и времени. Обратите внимание, что по умолчанию в системе установлен часовой пояс «Asia/Shanghai» - если данный вариант не подходит (с вероятностью 99% так и есть), сначала проделайте настройки под пунктом 3 – Change timezone. После этого уже введите значения текущей даты и времени в таком формате, как это указано в самих сообщениях. Например:

```
Set time settings:
1) Set ntp-server
2) Set date and time manually
3) Change timezone
4) Exit
2
Enter current date in YYYYMMDD format: 20221103
Enter current time in HH:MM:SS format: 13:45:43|
```

Данные параметры установят дату в 3 ноября 2022 года, а время в 13 часов 45 минут 43 секунды. Изменения применяются сразу

- 3) Change timezone – данный пункт запускает интерактивное меню выбора часового пояса:



Выберите из списка регион, а затем нужный часовой пояс. Если требуется установить локальное время в UTC, просто выберите пункт как на скриншоте выше – «None of the above» и в следующем окне выберите «UTC».

4. Disable autologin and set root password – выбор данного пункта сразу отключит автологин в локальную консоль и предложит ввести новый пароль для пользователя root, который нужно будет ввести дважды. Вводимые символы при этом отображаться не будут.

```
Set new root password:  
New password:  
Retype new password: |
```

После установки нового пароля автоматически откроется главное меню скрипта.

5. Setup radar adapters... - данный пункт следует выполнять ТОЛЬКО ПОСЛЕ(!) успешного выполнения пункта 1 главного меню, так как в противном случае скрипту будет нечего настраивать.

Скрипт предложит ввести следующие параметры адаптеров:

IP хоста

порт хоста (можно выбрать стандартное значение 55555, либо ввести порт вручную)

IP датчика

порт датчика (можно выбрать стандартное значение 55555, либо ввести порт вручную)

IP группы

порт группы (можно выбрать стандартное значение 60000, либо ввести порт вручную)

тип адаптера (можно выбрать тип UMRR-11, либо ввести значение вручную)

параметр «active» - true или false

протокол, по которому работает адаптер – TCP либо UDP

Вариант заполнения представлен на скриншоте:

```
Adapter host IP: 192.168.1.101
Adapter host port:
1) 55555
2) Enter manually
1
Adapter sensor IP: 192.168.1.102
Adapter sensor port:
1) 55555
2) Enter manually
1
Adapter group IP: 239.144.0.0
Adapter group port:
1) 60000
2) Enter manually
1
Adapter type:
1) UMRR-11
2) Enter manually
1
Active:
1) true
2) false
1
Connection protocol:
1) TCP
2) UDP
1
What's next?
1) Complete radar setup
2) Add one more adapter
```

IP-адреса и вводимые номера портов проверяются на валидность и в случае неверного ввода, скрипт требует ввести их заново.

После того, как внесены данные по адаптеру, можно выбрать один из двух вариантов.

```
What's next?
1) Complete radar setup
2) Add one more adapter
```

Первый вариант завершит настройку – запишет введенные параметры в файл конфига /opt/radar-go/adapter.json

Второй вариант перезапустит меню ввода (при этом сохранив параметры первого введенного адаптера в память) и можно будет ввести параметры следующего адаптера.

По окончании ввода снова появится меню «What's next?» и таким образом можно настроить нужное количество адаптеров.

ВАЖНО! Настройка адаптеров из данного пункта меню не добавляет адаптеры в конфиг к уже имеющимся, а полностью перезаписывает файл /opt/radar-go/adapter.json – для того чтобы добавить ещё один адаптер после завершения настройки, придётся заново вбить параметры и ранее добавленных.

После завершению настройки адаптеров можно применить изменения, перезапустив службу:
`systemctl restart radar.service`