

# Instalace oscamu na RPI

HOWTO: Compile OSCAM on Raspberry Pi, Dockstar or similar NAS running Debian

- [HOWTO: Compile OSCAM on Raspberry Pi, Dockstar or similar NAS running Debian](#)
- [Nová instalace Oscamu na Raspberry PI - kompletní návod](#)
  - [Nová instalace Oscamu na Raspberry PI - kompletní návod 2024](#)

# HOWTO: Compile OSCAM on Raspberry Pi, Dockstar or similar NAS running Debian

1. [linux-satellite-support-forum.blogspot](http://linux-satellite-support-forum.blogspot)
2. Preparations:

Update apt-list and install required packages:

Code:

```
apt-get update
apt-get -y install apt-utils dialog usbutils
apt-get -y install gcc g++ wget
apt-get -y install build-essential subversion libpcsclite1 libpcsclite-dev
apt-get -y install libssl-dev cmake make
apt-get -y install libusb-1.0-0-dev nano
```

If you need PCSC support:

Code:

```
apt-get install pcscd pcsc-tools
```

## 2. Get OSCAM source

Now switch to /usr/src, create a directory for oscam and download sources:

Code:

```
cd /usr/src
svn co http://www.oscam.to/svn/oscam/trunk oscam-svn
```

## 3. Compile OSCAM

Now compile oscam for your machine:

Code:

```
cd oscam-svn
mkdir build
chmod 755 build
cd build
cmake -DWEBIF=1 .. (those 2 dots are required. THIS IS IMPORTANT)
make
```

NOTE: If you get Errors while compiling e.g.

Code:

```
-- no libusb 1.0 found. No smartreader support
```

See the hint at the bottom before proceeding

Now we got our oscam binary  
We're gonna move it to /var/local

Code:

```
cd /usr/src/oscam-svn/build
ls
```

there should be a file called "oscam"  
So lets copy it:

Code:

```
cp oscam /var/local/
cd /var/local
chmod 755 oscam
```

4. Make OSCAM start at system startup:  
We're adding oscam to rc.local to make it start at system boot.

Code:

```
nano /etc/rc.local
```

Now add

Code:

```
/var/local/oscam &
```

right above line

Code:

```
exit 0
```

Oscam will start everytime the system boots up from now on

## 5. configs and logfiles

We still need to get some config files.

By default oscam looks in /usr/local/etc for these file

Code:

```
cd /usr/local/etc  
ls
```

There should be at least 3 files: oscam.user, oscam.server, oscam.conf

If not, copy your configs here

Now we need to create some logfiles and make them writable:

Code:

```
cd /var/local  
mkdir oscam  
chmod 755 oscam  
cd oscam  
mkdir cw  
chmod 755 cw
```

```
nano oscamuser.log (enter a space and save file)
nano oscam.log (enter a space and save file)
chmod 755 oscamuser.log
chmod755 oscam.log
```

## 6. Start oscam

You can now run oscam to check if it works

Code:

```
cd /var/local
./oscam
```

HINT: Oscam compile error: No smartreader support:  
If you get Errors while compiling e.g.

Code:

```
-- no libusb 1.0 found. No smartreader support
```

go to /usr/src, get libusb 1.0.6 from sourceforge and compile for your system:

Code:

```
cd /usr/src/
mkdir libusb
chmod 755 libusb
cd libusb
wget http://downloads.sourceforge.net/project/libusb/libusb-1.0/libusb-1.0.6/libusb-1.0.6.tar.bz2
tar xjf libusb-1.0.6.tar.bz2
make
install
```

After make install of libusb, start over with compiling oscam

# Nová instalace Oscamu na Raspberry PI - kompletní návod

instalace Oscamu s novým úložištěm binárky...

# Nová instalace Oscamu na Raspberry PI - kompletní návod 2024

**Kompletní nový návod: Instalace Oscamu na 64bit Raspbian**  
Tento návod předpokládá, že máte přístup k čisté instalaci Raspbianu a jste přihlášení jako uživatel `root`.

## 1. Aktualizace systému

Před instalací bychom měli se ujistit, že je systém aktuální

```
apt-get update && apt-get upgrade -y
```

## 2. Instalace potřebných nástrojů a závislostí

Ověření dostupnosti balíčku

Zkontrolujte, zda je balíček `libssl-dev` dostupný:

```
apt-cache search libssl-dev
```

Pokud se zobrazí výsledky, znamená to, že je balíček k dispozici.

## 3. Instalace správného balíčku

Pokud máte 64bitový systém, je důležité zajistit, že máte správné repozitáře. Ujistěte se, že používáte následující příkazy:

```
sudo apt-get install -y libusb-1.0-0-dev
```

zkuste nainstalovat konkrétní verzi OpenSSL pomocí následujícího příkazu:

```
sudo apt-get install -y openssl
```

Nainstalujte všechny balíčky nezbytné pro kompilaci Oscamu:

```
apt-get install -y git build-essential cmake libusb-1.0-0-dev libpcsclite-dev \
libssl-dev pcscd pcsc-tools gcc-aarch64-linux-gnu g++-aarch64-linux-gnu
```

- **Vysvětlení:**

- **git** : Pro stažení zdrojového kódu.
- **build-essential** , **cmake** : Pro kompilaci.
- **libusb-1.0-0-dev** , **libpcsclite-dev** , **libssl-dev** : Závislosti pro podporu USB, PCSC a SSL.
- **pcscd** , **pcsc-tools** : Služba a nástroje pro ?te?ky karet.
- **gcc-aarch64-linux-gnu** , **g++-aarch64-linux-gnu** : Kompilátory pro 64bit ARM architekturu.

### 3. Příprava systému pro PCSC ?te?ky

1. Ujist?te se, že služba **PCSC** b?ží:

```
bash
```

```
systemctl start pcscd systemctl enable pcscd
```

2. Ov??te, zda ?te?ka funguje:

```
bash
```

```
pcsc_scan
```

Výstup by m?l detekovat p?ipojenou ?te?ku karet.

### 4. Stažení zdrojového kódu Oscamu

Přesuňte se do adresáře **/usr/src** a stáhněte kód:



```
cd /usr/src
git clone https://git.streamboard.tv/common/oscam.git oscam-svn
```

1. Přesuňte se do složky se zdrojovým kódem:

```
cd oscam-svn
```

## 5. Instalace CMake

Spusťte příkaz pro instalaci CMake:

```
apt-get update
apt-get install -y cmake
```

## 5. Kompilace Oscamu

1. Vytvořte adresář pro kompilaci:

```
mkdir build && cd build
```

Spusťte konfiguraci:

```
cmake -DCMAKE_C_COMPILER=aarch64-linux-gnu-gcc -DHAVE_LIBUSB=1 -DHAVE_PCSC=1 \ -DWEBIF=1 -
DHAVE_LIBCRYPTO=1 -DWITH_SSL=1 ..
```

1. Spusťte kompilaci:

```
make
```

Zkopírujte binární soubor Oscamu do **/var/local/** :

```
cp oscam /var/local/oscam chmod 755 /var/local/oscam
```

## 6. Nastavení služby Oscam

1. Vytvořte nový soubor služby:

```
bash
```

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

2. Vložte následující obsah:

```
ini
```

```
[Unit] Description=Oscam Service After=network.target [Service] ExecStart=/var/local/oscam Restart  
=always User=root [Install] WantedBy=multi-user.target
```

3. Uložte a aktivujte službu:

```
bash
```

```
systemctl daemon-reload systemctl enable oscam systemctl start oscam
```

4. Ověřte stav služby:

```
bash
```

```
systemctl status oscam
```

---

## 7. Nastavení webového rozhraní Oscamu

1. Vytvořte konfigurační adresář:

```
bash
```

```
mkdir -p /usr/local/etc
```

2. Vytvořte soubor **oscam.conf** :

```
bash
```

```
nano /usr/local/etc/oscam.conf
```

3. Přidejte minimální konfiguraci:

[Zkopírovat kód](#)

```
ini
```

```
[webif] httpport = 8888 httpallowed = 0.0.0.0-255.255.255.255
```

4. Uložte soubor a restartujte službu:

```
bash
```

```
systemctl restart oscam
```

---

## 8. Testování instalace

1. Ověřte verzi Oscamu:

```
bash
```

```
/var/local/oscam -V
```

2. Otevřete webové rozhraní Oscamu:

- Do prohlížeče zadejte:

```
arduino
```

```
http://<IP vašeho Raspberry Pi>:8888
```

## 1. Automatický aktualizací skript

Vytvoříme jednoduchý skript pro aktualizaci Oscamu, který stáhne nejnovější verzi, zkompileje ji a nahradí stávající binární soubor.

### 1.1 Vytvoření aktualizacího skriptu

1. Vytvořte nový soubor:

```
bash
```

```
nano /usr/local/bin/oscam_update
```

Přidejte následující obsah:

---

```
#!/bin/bash
# Oscam Update Script
set -e

# Cesta k aktuálnímu Oscamu
OSCAM_BIN="/var/local/oscam"

# Adresář se zdrojovým kódem
SRC_DIR="/usr/src/oscam-svn"

echo "Aktualizuji Oscam..."

# Stáhněte nejnovější verzi kódu
if [ -d "$SRC_DIR" ]; then
    cd "$SRC_DIR"
    git pull
else
    git clone https://git.streamboard.tv/common/oscam.git "$SRC_DIR"
    cd "$SRC_DIR"
fi

# Přesuňte se do složky pro kompilaci
mkdir -p build
cd build

# Vyčistěte předchozí build a spusťte konfiguraci
make clean
rm -rf *
cmake -DCMAKE_C_COMPILER=aarch64-linux-gnu-gcc -DHAVE_LIBUSB=1 -DHAVE_PCSC=1 \
    -DWEBIF=1 -DHAVE_LIBCRYPTO=1 -DWITH_SSL=1 ..

# Zkompilujte novou verzi Oscamu
make

# Zastavte službu Oscam, aktualizujte binární soubor a restartujte službu
```

```
systemctl stop oscam
cp oscam "$OSCAM_BIN"
chmod 755 "$OSCAM_BIN"
systemctl start oscam

echo "Aktualizace dokončena. Oscam byl úspěšně aktualizován."
```

#### 1. Uložte a nastavte spustitelný p?íznak:

```
bash
```

```
chmod +x /usr/local/bin/oscam_update
```

## 1.2 Spuštění aktualizace

Pro aktualizaci Oscamu jednoduše spusťte:

```
bash
```

```
sudo /usr/local/bin/oscam_update
```

## 2. Automatizace aktualizací

Pokud chcete, aby se Oscam aktualizoval automaticky (nap?. jednou týdn?), p?idejte tento skript do crontabu:

#### 1. Otevřete crontab:

```
bash
```

```
crontab -e
```

#### 2. P?idejte řádku pro automatickou aktualizaci každou ned?li v 3:00 ráno:

```
bash
```

```
0 3 * * 0 /usr/local/bin/oscam_update
```

3. Uložte a zavězte editor.

### 3. Ruční kontrola verze

Po aktualizaci můžete ověřit, že se nainstalovala nejnovější verze Oscamu:

```
bash
```

```
/var/local/oscam -V
```

Tento příkaz zobrazí verzi Oscamu a další informace, jako je datum a číslo revize.

### 4. Záloha před aktualizací

Pro jistotu je dobré zálohovat aktuální binární soubor a konfiguraci před každou aktualizací:

1. Zálohujte binární soubor:

```
bash
```

```
cp /var/local/oscam /var/local/oscam.bak
```

2. Zálohujte konfiguraci:

```
bash
```

```
tar -czvf oscam_config_backup.tar.gz /usr/local/etc
```