



Malinowa stacja radiowa

16 > >

Dostałem na święta Raspberry PI. :) Jeszcze wcześniej widziałem jakie ciekawe projekty można wykonać z wykorzystaniem tej płytki. Moją szczególną uwagę zwróciła możliwość nadawania sygnału z modulacją FM na falach używanych przez odbiorniki radiowe bez użycia żadnych dodatkowych układów. Wystarczy podłączyć jakiś kawałek drutu do jednego z pinów GPIO na płycie. Sam program to umożliwiający - pifm - pozwala nadawać jedynie plik wav o odpowiednich parametrach - 1 ścieżka dźwiękowa, 16 bitów na próbkę (sample) i domyślnie 22050 Hz choć to można zmienić podczas wywoływania pifm.

Gdy dostałem Malinkę, od razu chciałem się tym pobawić. W komplecie z pifm znalazłem przykładowy dźwięk - intro Gwiezdných Wojen. Udało mi się to nadać i nawet kolega bez większych problemów odebrał to u siebie w domu - jakieś 30-40 metrów + 2 metrowej grubości ceglane ściany.

Byłem tak zadowolony z wyniku eksperymentu że od razu postanowiłem nadać coś bardziej użytecznego - Spotify i lokalną kolekcję muzyki.



Poza tym że chciałem wykorzystać Mopidy - chyba jedyny obecnie działający odtwarzacz wspierający Spotify, nie miałem zielonego pojęcia jak się za coś takiego zabrać. Na początku planowałem w jakiś sposób wyciągnąć output audio domyślnej i jedynej karty dźwiękowej w formacie wav i wrzucić do go pliku

FIFO, a z stamtąd odczytać przez pifm, okazało się jednak że obsługa audio pod Linuxem jest znacznie bardziej skomplikowana niż cat /dev/snd0. :(



Napisałem więc na Facebookowej grupie Malinowej, oraz później na oficjalnym kanale IRC Mopidy. Dzięki [Łukaszowi C. Jokielowi](#) oraz jednemu z developerów Mopidy udało mi się wreszcie to odpalić. Z tego miejsca, bardzo Wam za to dziękuję. :)

Poniżej przedstawiam instrukcję jak coś takiego zrobić. Zakładam że mamy Raspberry PI z zainstalowanym na karcie Raspbianem i aby uniknąć problemów - czystym i w najnowszej wersji.

Uwaga: Wykonujesz to wszystko na własną odpowiedzialność :)

1. Robimy i podłączamy antenę. Może być to krótki kawałek drutu. Podłączamy go do pinu GPIO4, oznaczanego też jako 7. wg. biblioteki WiringPi oraz na schemacie poniżej. Czerwony pin +5V znajduje się przy krawędzi płytki.



2. Instalujemy pifm - tworzymy tymczasowy katalog [1], wchodzimy do niego [2], pobieramy archiwum zawierające program i przykładowy dźwięk (intro Gwiezdných Wojen) [3], wypakowujemy archiwum [4], kopiujemy właściwy program do folderu `/usr/local/bin` [5], wychodzimy z tymczasowego katalogu [6] i go usuwamy [7].



```
# Text
mkdir /home/pi/pifm_tmp
cd /home/pi/pifm_tmp
wget http://www.icrobotics.co.uk/wiki/images/c/c3/Pifm.tar.gz
tar -xf Pifm.tar.gz
sudo mv pifm /usr/local/bin/pifm
cd /home/pi/
rm -rf /home/pi/pifm_tmp
```

Folder `/usr/local/bin`, wraz z kilkoma innymi, znajduje się domyślnie w zmiennej środowiskowej `$PATH`. Gdy uruchamiamy jakiś program, wpisując jego polecenie, powłoka (najczęściej bash) sprawdza najpierw czy nie jest to alias (przekierowanie do innego programu) lub polecenie wewnętrzne (które realizuje sama powłoka), a następnie próbuje znaleźć plik wykonywalny tego polecenia/programu przeszukując wszystkie foldery podane w zmiennej środowiskowej `$PATH`. Dzięki umieszczeniu pliku wykonywalnego programu `pifm` w katalogu znajdującym się w tej zmiennej mamy możliwość późniejszego uruchomienia go, nie podając pełnej ścieżki, a jedynie nazwę.

3. Instalujemy serweroodtwarzacz Mopidy - pobieramy klucz GPG repozytorium i dodajemy go do APT-a [1], pobieramy i umieszczamy w folderze `/etc/apt/sources.list.d/` wpis o nowym repozytorium [2], aktualizujemy cache wszystkich repozytoriów APT-a [3] oraz instalujemy właściwy program i plugin do obsługi Spotify [4].

Jeśli nie chcemy Spotify, w ostatniej linijce możemy pominąć `"mopidy-spotify"`. Mopidy obsługuje także dużo więcej pluginów do podobnych serwisów i nie tylko. Ich lista znajduje się tutaj: <http://docs.mopidy.com/en/latest/ext/>.



```
# Text
wget -q -O - https://apt.mopidy.com/mopidy.gpg | sudo apt-key add -
sudo wget -q -O /etc/apt/sources.list.d/mopidy.list
https://apt.mopidy.com/mopidy.list
sudo apt-get update
sudo apt-get -y install mopidy mopidy-spotify
```

Repozytorium oprogramowania to serwer lub grupa serwerów z których przy pomocy programu apt-get w Debianie (i pochodnych) możemy pobierać i instalować programy (w pakietach) już przygotowane i skompilowane dla naszej dystrybucji. Każde repozytorium ma parę kluczy - publiczny i prywatny. Przy pomocy jednego z kluczy jesteśmy w stanie odszyfrować dane zakodowane tym drugim. Sumy kontrolne, czyli wartości umożliwiające nam sprawdzenie czy pakiet nie został zmieniony podczas transmisji w sieci, są szyfrowane kluczem prywatnym serwera repozytorium. Następnie przy pomocy pobranego w pierwszej linijce klucza publicznego jesteśmy w stanie odszyfrować pierwotną sumę kontrolną i porównać ją z nowo wygenerowaną dla pobranego pakietu. W ten sposób upewniamy się że nikt (czyt. NSA) po drodze nie podmienił nam pakietu. Oczywiście dzieje się to automatycznie.

4. Uruchamiamy Mopidy. Program wyświetli błąd i się wyłączy, ale wygeneruje przy okazji plik konfiguracyjny, którego teraz nie będziemy musieli pisać od nowa tylko zmienimy odpowiednie wartości.

```
# Text
mopidy
```



5. Edytujemy plik konfiguracyjny Mopidy przy pomocy edytora nano.

Wpisujemy:

```
# Text
nano /home/pi/.config/mopidy/mopidy.conf
```

a) Na początku odnajdujemy sekcję *audio* oraz zmienną *output*, usuwamy znak *#* z początku linijki i zmieniam zawartość zmiennej na podaną poniżej:

```
# Text
[audio]
output = capsfilter caps="audio/x-raw-int,depth=16,channels=1" ! wavenc ! queue
! filesink location=/tmp/mopidy.wav
```



Jest to zapis dla Gstreamera w którym informujemy go co ma zrobić z wyjściem audio programu. Tutaj najpierw następuje połączenie kanałów audio, tak aby uzyskać dźwięk mono oraz zmiana ilości bitów na próbkę dźwięku (sample) z 32 na 16, następnie zamiana/zakodowanie ścieżki dźwiękowej do formatu wav i zapis do pliku */tmp/mopidy.wav*. */tmp/mopidy.wav* to nie do końca prawdziwy plik, ale o tym za chwilę.

b) W kolejnej sekcji decydujemy czy chcemy korzystać z lokalnej biblioteki plików na karcie pamięci (lub jakimkolwiek przymontowanym dysku) i jeśli tak, gdzie ma się ta biblioteka znajdować. Domyślnie wtyczka *local* jest włączona,

więc jeśli chcemy korzystać z lokalnej biblioteki zmienną *enabled* zostawiamy w

dobreprogramy



podajemy w niej ścieżkę do folderu w którym będziemy trzymać pliki mp3, flac itp. Jeśli nie chcemy korzystać z biblioteki lokalnej, odkomentujemy zmienną *enabled* i zmienimy jej zawartość na *false*. *media_dir* zostawiamy w spokoju.

Bibliotekę proponuję umieścić w */home/pi/mopidy*.

```
# Text
[local]
enabled = false
--lub--
media_dir = /home/pi/mopidy
```

c) Tutaj decydujemy czy chcemy korzystać ze wtyczki Spotify czy nie. Jeśli nie instalowaliśmy wtyczki *mopidy_spotify*, tej sekcji nie będzie w configu, więc nie musimy nic tutaj wyłączać. Jeśli chcemy korzystać ze Spotify, usuwamy znak *#* z przed linii *username* i *password* i podajemy tam nasz login i hasło do Spotify. Jeśli rozmyśliliśmy się ze Spotify już po instalacji wtyczki, podobnie jak poprzednio, ustawiamy zmienną *enabled* na *false*, oczywiście wcześniej ją odkomentowując.

```
# Text
[spotify]
enabled = false
--lub--
username = aaa
password = bbb
```

Aby wyjść z nano wciskamy Ctrl+X oraz T lub Y (w zależności od wersji językowej systemu) i zatwierdzamy enterem że chcemy zapisać zmiany.

6. Tworzymy plik specjalny FIFO. Wszystko co do takiego pliku zostanie zapisane jest przekazywane od razu do odczytujących go programów, ale nie jest nigdzie zapisywane. U nas zapisywać będzie Mopidy, a odczytywać pifm.

```
# Text
mkfifo /tmp/mopidy.wav
```

7. Jeśli korzystamy z lokalnej biblioteki muzycznej, tworzymy katalog w którym będziemy trzymać nasze empetrójki i je tam wrzucamy [1], a następnie

indeksujemy bibliotekę [3].

```
# Text
mkdir /home/pi/mopidy
--wrzucamy pliki--
mopidy local scan
```

Bibliotekę będziemy musieli zawsze indeksować [3] po dodaniu do niej plików. Po zaindeksowaniu konieczne będzie ponowne uruchomienie Mopidy.

8. Przy pomocy tych poleceń instalujemy screena [1] i uruchamiamy w tło serwer odtwarzacz mopidy [2] oraz transmitter FM pifm [3]. W trzeciej linii przedostatni parametr - u mnie 87.6 to częstotliwość na jakiej nadajemy. Możemy podać tutaj dowolną od 1 do 250 MHz.

```
# Text
sudo apt-get -y install screen
screen -dmS mopidy mopidy
screen -dmS pifm sudo pifm /tmp/mopidy.wav 87.6 44100
```

screen -dmS uruchomi nasze polecenia w tło. Wartość po *S* to nazwa sesji - dzięki niej będziemy mogli łatwo dostać się do konsoli z uruchomionymi poleceniami, choćby po to żeby sprawdzić co się dzieje gdy coś nie będzie działało prawidłowo, wpisując *screen -dr nazwa*. W dalszej części znajdują się właściwe polecenia uruchamiające programy.

9. Instalujemy klienta MPC do sterowania naszym odtwarzaczem:

```
# Text
sudo apt-get -y install ncmtcpp
```

10. Uruchamiamy klienta MPC i wybieramy utwór z naszej kolekcji lub wyszukujemy jakiś z bazy Spotify.

```
# Text
ncmtcpp
```

Aby przejść do opcji z górnej belki pilota wybieramy ich numer na klawiaturze. Dalej poruszamy się strzałkami.



--

Zdjęcie anteny oraz schemat pinów GPIO pochodzi ze strony <http://malinowepi.pl/post/45982694540/malinowe-radio-pierwsze-doswiadc...> i jest udostępniony za zgodą autora.

Zdjęcie Raspberry B rev 2.0 pochodzi z oficjalnej strony projektu.

--

Edit: Pojawiają się głosy że do działania radia może być potrzebny jeszcze pakiet gstreamer0.10-plugins-bad. Instalujemy go tak:

```
# Text
sudo apt-get install gstreamer0.10-plugins-bad
```

ę



ó ż ą ę

ś