

Linux w systemach wbudowanych – Laboratorium 5

Tymon Felski

2 czerwca 2017

1 Treść zadania

Podczas piątego laboratorium należało przenieść projekt stworzony w ramach czwartego laboratorium do środowiska innego niż Buildroot. Cel laboratorium był taki sam jak poprzednio, czyli przygotowanie przy pomocy płytki Raspberry Pi urządzenia wyposażonego w złożony interfejs użytkownika, w którym:

- przyciski i diody LED powinny być użyte do podstawowej obsługi urządzenia,
- interfejs WWW lub inny interfejs sieciowy powinien być użyty do bardziej zaawansowanych funkcji,
- przetwarzany powinien być dźwięk lub obraz.

2 Odtwarzanie projektu z załączonego archiwum

Należy wykonać kroki opisane poniżej.

3 Opis rozwiązania

3.1 Przygotowanie

Wybór środowiska padł na LEDE Project. Pobrano obraz systemu w wersji 17.01. Znajduje się on w dołączonym archiwum pod nazwą lede.img.gz.

Pierwszym krokiem było wgranie systemu na kartę pamięci płytki. W tym celu uruchomiono płytkę w trybie ratunkowym (przerywając bootowanie i wpisując `run rescue`), a następnie wykonano polecenie

```
# ssh felskit@192.148.143.XXX "cat /malina/felskit/lede.img.gz" \  
> | gunzip -c | dd of=/dev/mmcblk0 bs=4M
```

Po rozpakowaniu i wgraniu obrazu oraz ponownym uruchomieniu płytki można było przystąpić do konfiguracji systemu. Niezbędne było skonfigurowanie połączenia sieciowego.

```
# udhpcp -i br-lan  
# uci set network.lan.proto=dhcp  
# uci commit  
# /etc/init.d/network restart
```

To zapewniło działające połączenie z siecią. Kolejnym krokiem była aktualizacja listy paczek dostępnych przez `opkg`.

```
# opkg update
```

Aplikacja przygotowana na poprzednim laboratorium korzysta z `mpd` oraz `mpc` przy odtwarzaniu muzyki, więc należało je zainstalować poleceniami

```
# opkg install mpd  
# opkg install mpc
```

Domyślnie instalującym się `mpd` jest `mpd-mini`, jednak w zupełności wystarczy do zaspokojenia potrzeb aplikacji. Niezbędny był też Python w wersji 3 i odpowiadający mu package manager.

```
# opkg install python3  
# opkg install python3-pip
```

Aby uniknąć komplikacji przy instalacji pozostałych pakietów, zaktualizowano `setuptools`.

```
# pip3 install --upgrade setuptools
```

Do działania przygotowanej aplikacji niezbędny jest moduł `flask`, jednak jeden z pakietów, od których on zależy, wymaga modułu `six`. Wobec tego wykonano polecenia

```
# pip3 install six
# pip3 install flask
```

Aplikacja korzysta także z interfejsu GPIO do zarządzania diodami i przyciskami na płytce Raspberry Pi, więc wymagany jest moduł, który spełni to wymaganie. Niestety instalacja wykorzystywanego podczas poprzedniego laboratorium modułu `RPi.GPIO` nie jest możliwa pod używanym obrazem LEDE, chociażby ze względu na brak `ccache`. Wybrano jego uproszczoną alternatywę - `gpio`.

```
# pip3 install gpio
```

Tak przygotowany system był w stanie uruchomić aplikację, jednak należało w niej najpierw wprowadzić kilka modyfikacji, które opisane są w następnym punkcie.

3.2 Aplikacja

Stworzona w ramach laboratorium aplikacja to internetowe radio napisane w Pythonie, którym można sterować przy pomocy interfejsu GPIO oraz interfejsu webowego. Ze względu na zmianę modułu obsługującego interfejs GPIO, aplikacja musiała zostać nieznacznie zmodyfikowana.

Pierwszym krokiem była zmiana importowanego modułu.

```
- import RPi.GPIO as GPIO
+ import gpio as GPIO
```

Ponadto, należało usunąć linię konfigurującą tryb działania interfejsu.

```
- GPIO.setmode(GPIO.BCM)
```

Konieczna była zmiana sposobu inicjacji przycisków.

```
- GPIO.setup(button, GPIO.IN, pull_up_down = GPIO.PUD_UP)
+ GPIO.setup(button, GPIO.IN, initial=True)
```

Niezbędna okazała się także zmiana odniesień do stanu przycisków i diód poprzez `GPIO.HIGH` oraz `GPIO.LOW` na odpowiednio 1 oraz 0. Ostatnim krokiem było odpowiednie wywołanie czyszczenia stanu diód, które polega już na jednorazowym wywołaniu `GPIO.cleanup()`.

```
for led in [17, 18, 23, 24]:
    GPIO.cleanup(led)
```

Tak zmodyfikowana aplikacja mogła zostać bez problemu uruchomiona na wcześniej skonfigurowanym obrazie systemu dostarczonym przez LEDE Project. Przy pomocy `ssh` folder z radiem znajdujący się w dostarczonym archiwum został skopiowany z komputera laboratoryjnego na kartę pamięci płytki.

```
$ scp -r radio/ root@192.168.143.XXX:/radio
```