

Laboratorium nr 2

SKPS

1. Przygotowanie środowiska.

Po otrzymaniu zestawu początkowego przeszliśmy do składania zestawu. Zestaw został złożony i zatwierdzony przez prowadzącego. Wszystkie urządzenia i kable zostały ułożone w sposób ułatwiający pracę na komputerze.

2. Pierwsze uruchomienie RPi.

Uruchomienie zasilania spowodowało włączenie się urządzenia. Wpisanie komendy ***tio /dev/ttyUSB0*** skutkowało poprawnym włączeniem się ***tio***. Otrzymaliśmy informację *connected*. Pojawiły się logi z bootloadera i uzyskaliśmy możliwość zalogowania się. Załadował się system ratunkowy Raspberry Pi OS. Wpisanie podanych danych logowania zakończyło się poprawnym wejściem. Sprawdziliśmy stan połączenia sieciowego na RPi poleceniem ***ifconfig*** oraz ***ping 10.42.0.1***.

3. Kopiowanie plików na RPi

Na komputerze host postawiliśmy serwer HTTP przy użyciu polecenia: ***python3 -m http.server***.

4. Instalacja OpenWRT z wykorzystaniem systemu ratunkowego

Pobraliśmy obraz systemu poleceniem **wget**: **wget https://downloads.openwrt.org/releases/21.02.1/targets/bcm27xx/bcm2711/openwrt-21.02.1-bcm27xx-bcm2711-rpi-4-ext4-factory.img.gz**.

Potem dokonaliśmy rozkompresowania poleceniem: **gzip -d openwrt-21.02.1-bcm27xx-bcm2711-rpi-4-ext4-factory.img.gz**. Następną czynnością było montowanie: **losetup -P -f openwrt-21.02.1-bcm27xx-bcm2711-rpi-4-ext4-factory.img**. Wykonaliśmy kopiowanie partycji z rootfs z obrazu OpenWRT na kartę SD, na partycję 2: **dd if=/dev/loop0p2 of=/dev/mmcblk0p2 bs=4096**. Stworzyliśmy katalogi, w których zamontujemy partycję: **mkdir /mnt/boot /mnt/owrt**. Zamontowaliśmy partycję pierwszą obrazu OpenWRT w katalogu /mnt/owrt: **mount /dev/loop0p1 /mnt/owrt**, a następnie zamontowaliśmy partycję pierwszą karty SD (partycja boot) w katalogu /mnt/boot: **mount /dev/mmcblk0p1 /mnt/boot**, otrzymaliśmy tu ostrzeżenie, że źle dokonano poprzedniego odmontowania, jednak prowadzący powiedział, że można to pominąć. Zwiększyliśmy system plików OpenWRT tak, żeby wypełnił całą partycję: **resize2fs /dev/mmcblk0p2**. Zrestartowaliśmy system poleceniem **reboot** i w trakcie ponownego uruchomienia przytrzymaliśmy przycisk SW4 na RPi. Zgłosił nam się system OpenWRT. Przystąpiliśmy do konfiguracji sieci przy użyciu edytora vi: **vi /etc/config/network**. Zakomentowaliśmy i zmodyfikowaliśmy podane w skrypcie do laboratorium linie. Zapisaliśmy zmiany i dokonaliśmy restartu sieci: **/etc/init.d/network reload**.

```

user@lab-3: ~/skps21z_kamilsulkowski_karolkasperek/cw2
Plik Edycja Widok Wyszukiwanie Terminal Karty Pomoc
user@lab-3: ~/skps21z_k... x user@lab-3: ~/skps21z_k... x user@lab-3: ~ x
BusyBox v1.33.1 (2021-10-24 09:01:35 UTC) built-in shell (ash)

 _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _
| |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |
|_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|
W I R E L E S S F R E E D O M

-----
OpenWrt 21.02.1, r16325-88151b8303
===== WARNING! =====
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
=====
root@OpenWrt:/# vi /etc/config/network
root@OpenWrt:/# cat /etc/config/network

config interface 'loopback'
    option device 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'

config globals 'globals'
    option ula_prefix 'fd1d:de45:d44b::/48'

#config device
#    option name 'br-lan'
#    option type 'bridge'
#    list ports 'eth0'

config interface 'lan'
    option device 'eth0'
    option proto 'dhcp'
#    option ipaddr '192.168.1.1'
#    option netmask '255.255.255.0'
#    option ipassign '60'

root@OpenWrt:/# █

```

5. Interfejs http dla OpenWRT.

W przeglądarce na urządzeniu PC wpisaliśmy adres: <http://10.42.0.227>, podaliśmy login: root i bez hasła udało nam się zalogować. Wszystko było poprawnie skonfigurowane.

6. Eksperymenty w OpenWRT

Dokonaliśmy uaktualnienia pakietów poleceniem ***opkg update***. Zainstalowaliśmy wszystkie wymienione w skrypcie

pakiety: ***gpiod-tools***, ***i2c-tools***, ***spi-tools***. Następnie dokonaliśmy instalacji potrzebnych pakietów takich jak: python, nano i pip. Na koniec zainstalowaliśmy pakiety potrzebne do obsługi urządzeń I/O z poziomu Pythona: ***python3-smbus***, ***python3-gpiod*** oraz ***gpio4*** (ten zainstalowany za pomocą pip).

7. Obsługa akcesoriów przez GPIO

Następną częścią było przystąpienie do wykonania zadań. Rozpoczęliśmy od zadania numer jeden. Kod źródłowy zadania umieściliśmy na gitlabie pod nazwą gpio_led_1.py. Włącza on 10-krotnie nasz LED, a przerwa między włączeniem a wyłączeniem wynosi 1 sekundę.

Potem zrobiliśmy zadanie numer 4 i z reakcją na naciskanie przycisku na RPi 4 i działało prawidłowo. Zadanie znajduje się pod nazwą gpio_in.py.