

SKPS laboratorium 2

Sprawozdanie

1. Instalacja OpenWRT

Poleceniem wget pobrano obraz systemu:

```
wget https://downloads.openwrt.org/releases/21.02.1/targets/bcm27xx/bcm2711/openwrt-21.02.1-bcm27xx-bcm2711-rpi-4-ext4-factory.img.gz
```

Następnie poleceniem gzip rozkompresowano pobrany obraz.

Załadowano go jako urządzenie loop:

```
losetup -P -f openwrt-21.02.1-bcm27xx-bcm2711-rpi-4-ext4-factory.img
```

Efekt wyświetlenia stanu wszystkich urządzeń loop komendą:

```
losetup -a
```

```
/dev/loop0: 0 openwrt-21.02.1-bcm27xx-bcm2711-rpi-4-ext4-factory.img
```

Także obraz system został wczytany na urządzeniu o nazwie loop0.

Skopiowano partycję z rootfs z obrazu OpenWRT na kartę SD, na partycję 2:

```
dd if=/dev/loop0p2 of=/dev/mmcblk0p2 bs=4096
```

Stworzono katalogi, w których zamontowano później partycje:

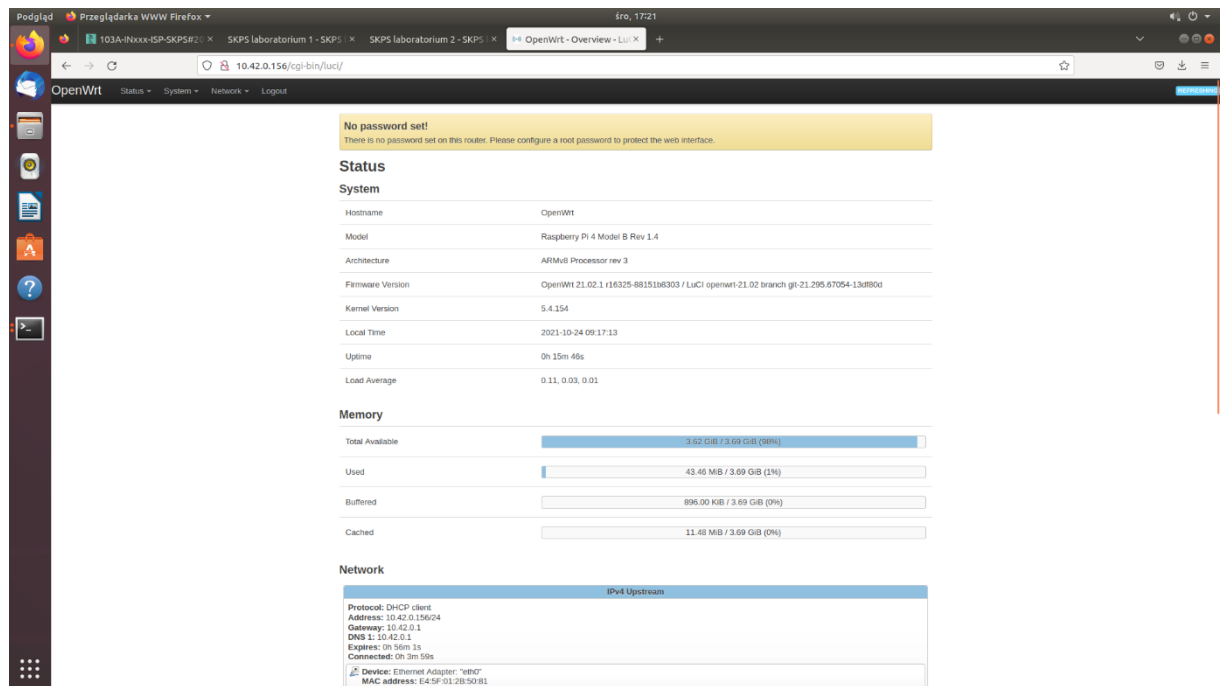
```
mkdir /mnt/boot /mnt/owrt
```

Następnie korzystając z komendy mount zamontowano partycje pierwsze obrazu OpenWRT oraz karty SD.

Skopiowano pliki z obrazu OpenWRT do katalogu user na partycji boot na karcie SD (komenda cp).

Następnie powiększono system plików OpenWRT tak, żeby wypełniał całą partycję.

Po reboocie, trzymając na płycie przycisk SW4, udało się uruchomić OpenWRT na RPi:



2. Eksperymenty w OpenWRT

Przy wykonaniu polecenia `opkg` uzyskaliśmy komunikat 'Failed to download' z powodu braku połączenia z siecią. Zgodnie z zaleceniem odnośnie problemów z DNS, zdefiniowano lokalnie adres serwera nazw w pliku `/etc/resolv.conf` w systemie ratunkowym na RPi oraz w OpenWRT.

Dodano na końcu tego pliku linię `nameserver 8.8.8.8`.

Następnie z powodzeniem wykonano polecenie `opkg update`.

Następnie zainstalowano pythona: `opkg install python3`

Aby zainstalować pakiety pythona za pomocą komendy `opkg install python3-pip` zainstalowano `pip`, a następnie dzięki `pip` zainstalowano `python3-smbus`, `python3-gpiod` oraz `gpio4`.

3. Obsługa akcesoriów przez GPIO

Obsługa GPIO w Pythonie

Zainstalowano pakiet pythona `gpio4`: `pip install gpio4`

Następnie zabrano się za pisanie kodu w pythonie do zadań (korzystając z komendy `vi`).

Zadanie 1. – dioda LED włączała się i wyłączała z odstępem czasowym 0.1 sekundy.

Zadanie 2. – przez 10 sekund jasność diody płynnie się zmienia – wypełnienie rośnie od 0% do 100% a następnie maleje do 0% i w kółko.

Zadanie 3. – najpierw buzzer podpięto do płytki RPi (- do GND, S do GPIO22), z Internetu pobrano częstotliwości gamy C-dur, która następnie została odegrana na buzzerze (po sekundzie na każdej częstotliwości).

Zadanie 4. – przez czas wciskania przycisku SW1 na płytce RPi zapala się dioda LED.

Zadanie 5. – próba z czujnikiem odległości niesfinalizowana z powodu czasu.