

SKPS - Laboratorium 2

Zespół korzysta z karty SD - 105e

1. Przygotowanie stanowiska

1. Płytką została podłączona zgodnie ze schematem z instrukcji
2. Prowadzący zweryfikował poprawność podłączenia

2. Instalacja OpenWRT na RPi4

Ze względu na problemy techniczne pierwszą godzinę laboratorium spędziliśmy na próbie odnalezienia błędu i poprawnego uruchomienia płytki. Koniec końców okazało się, że winny był sprzęt i musieliśmy zmienić stację roboczą - na niej sprzęt był już sprawny. Następnie przystąpiliśmy do instalacji OpenWRT:

1. Przy próbie pobrania prekompilowanego obrazu OpenWRT natrafiliśmy na błąd DNS związany z błędami certyfikatów. Poradziliśmy sobie z nim edytując plik `/etc/resolv.conf` :
nameserver 10.42.0.1 # eth0
nameserver 8.8.8.8

2. Następnie pobraliśmy obraz systemu komendą:

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

3. Kolejno przeszliśmy do rozpakowania pobranego pliku:

```
gzip -d openwrt-21.02.1-bcm27xx-bcm2711-rpi-4-ext4-factory.img.gz
```

4. Zamontowania go w katalogu plików

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

5. Skopiowania obrazu na odpowiednią partycję

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

6. Następnie stworzyliśmy odpowiednie foldery przeznaczone do montowania partycji

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

7. Zamontowaliśmy odpowiednie pliki

```
mount /dev/loop0p1 /mnt/owrt
```

```
mount /dev/mmcblk0p1 /mnt/boot
```

8. Skopiowaliśmy pliki obrazu OpenWRT do katalogu użytkownika korzystając z operacji:

```
cp /mnt/owrt/cmdline.txt /mnt/boot/user/
```

```
cp /mnt/owrt/kernel8.img /mnt/boot/user/
```

```
cp /mnt/owrt/bcm2711-rpi-4-b.dtb /mnt/boot/user/
```

9. Na koniec zmieniliśmy rozmiar systemu plików OpenWRT do wypełnienia partycji

```
resize2fs /dev/mmcblk0p2
```

Po zrestartowaniu płytki udało nam się uruchomić system OpenWRT

[illegible]

Następnie musieliśmy jeszcze skonfigurować ustawienia sieci w celu pobrania niezbędnych pakietów. Konfiguracja pliku `/etc/config/network` wyglądała następująco:


```
root@OpenWrt:/etc# 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 'fd7d:fcff:ac97::/48'

config interface 'lan'
    option device 'eth0'
    option proto 'dhcp'
    option ipaddr '192.168.1.1'
    option netmask '255.255.255.0'
    option ip6assign '60'
```

Po restarcie sieci wszystko działało w porządku i mogliśmy zainstalować niezbędne pakiety:



The screenshot shows a terminal window titled "user@lab-1: ~". The window has a menu bar with options: "Plik", "Edycja", "Widok", "Wyszukiwanie", "Terminal", and "Pomoc". The terminal content shows the command `python3 --version` being executed, resulting in the output `Python 3.9.16`. The prompt `root@OpenWrt:/#` is visible before and after the command.

3. Obsługa akcesoriów przez GPIO

Zadanie 1

Pierwsze zadanie polegające na utworzeniu skryptu w pythonie włączające i wyłączającego 10-krotnie diodę zrealizowaliśmy w pliku "gpio_led_1.py". Program został uruchomiony na płytce i działał bez zarzutów.

Zadanie 2

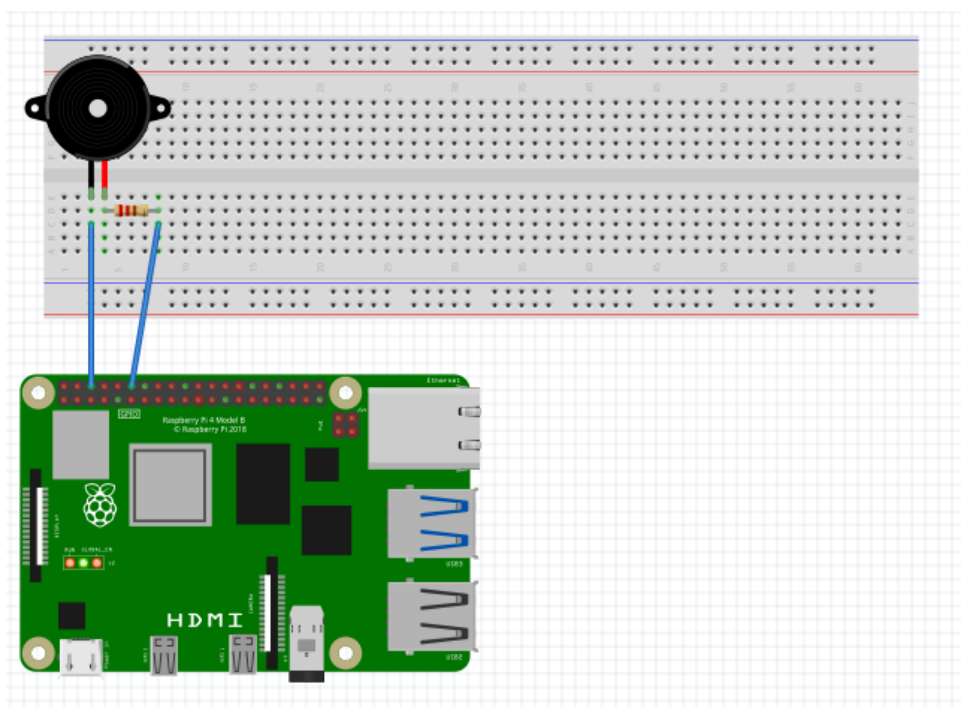
Drugie zadanie polegające na utworzeniu skryptu w pythonie, który przez 10 sekund będzie w płynny sposób zmieniał jasność LED na adapterze do RPi. Zadanie zrealizowaliśmy w pliku "gpio_led_2.py". Program został uruchomiony na płytce i możliwe do zaobserwowania były płynne zmiany jasności diody.

Zadanie 3

Trzecie zadanie polegało na utworzeniu skryptu w pythonie, który będzie reagował na wciśnięcie przycisku na adapterze do RPi. Zadanie zrealizowaliśmy w pliku "gpio_in.py". Program został uruchomiony na płytce i rzeczywiście wciśnięcie przycisku SW4 powodowało zapalenie się diody.

Zadanie 4

W zadaniu czwartym realizowaliśmy układ z buzzerem pasywnym, który przy użyciu skryptu w języku python miał generować kolejne dźwięki gamy C-dur w 2 oktawach. Układ został podłączony według schematu niżej. Rozwiązanie znajduje się w pliku "gpio_snd.py". Program został uruchomiony na płytce - buzzer wydawał kolejne dźwięki z gamy C-dur.



Zadanie 5

W zadaniu piątym jako efektor/czujnik do uruchomienia przez GPIO wybraliśmy serwomechanizm. Podłączenie miało wyglądać jak na schemacie poniżej, a prototypowy kod znajduje się w pliku "gpio_serwo.py". Niestety z powodu problemów technicznych na początku zajęć zabrakło nam czasu na odpalenie skryptu na samej płytce, więc nie jesteśmy w stanie jednoznacznie stwierdzić poprawności jego działania.

