

Przygotowanie do zajęć

Sklonuj repozytorium <https://github.com/pwr-zak/iot-lab> do swojego katalogu domowego. W repozytorium znajdziesz katalog „Zajecia_nr_4”, w którym znajdują się pliki potrzebne do wykonania zadań.

Ustaw zmienną `SDK_ROOT` w pliku `Makefile` tak, aby wskazywała na SDK.

Zadanie

1. Zmodyfikuj plik `main.c` w taki sposób, aby odczytać adres MAC lub EUI64 interfejsu Bluetooth. Wykorzystaj API oferowane przez nRF SDK. Odczytany adres lub EUI64 wyświetl przy pomocy makra `APPL_LOG`.
2. Znając adres MAC lub EUI64 nawiąż trwałe połączenie IPv6/Bluetooth pomiędzy routerem brzegowym (Raspberry Pi) oraz swoją płytką nRF52.

1. Nawiąż sesję SSH z routerem brzegowym (prowadzący poda Ci potrzebne informacje) poleceniem:

```
ssh login@adres_ip_routera
```

2. W sesji SSH zaloguj się na konto administratora poleceniem:

```
sudo su
```

3. Upewnij się, czy twoja płytka nRF52 poprawnie rozgłasza swoją obecność (wysyła pakiety Advertisement po Bluetooth). Przeanalizuj w tym celu wyjście wyniku polecenia:

```
hcidtool lscan --passive
```

4. Nawiąż połączenie ze swoją płytką nRF52 poleceniem poniżej. Zwróć uwagę na zachowanie LEDów na płycie oraz na komunikaty wyświetlane na UARTcie

```
echo "connect adres_MAC_płytki 1" > /sys/kernel/debug/bluetooth/6lowpan_control
```

5. Przeanalizuj konfigurację interfejsów Bluetooth oraz IP na routerze brzegowym wykonując następujące polecenia:

```
ip addr show dev bt0
```

```
hciconfig
```

6. Sprawdź, czy połączenie pomiędzy twoją płytką nRF52 i routerem brzegowym zostało poprawnie zestawione. Wykorzystaj narzędzie `ping`

```
ping adres_IPv6_twojej_płytki -c 1 -I bt0
```

3. W procedurze obsługi wciśnięcia przycisku „Button 1” napisz kod, który wyśle pakiet UDP do serwera działającego na routerze brzegowym:

1. Adres routera: `2001:db8::1`

2. Numer portu serwera: `12345`

3. Zawartość pakietu UDP: twój nr indeksu

Skorzystaj z API oferowanego przez nRF SDK.

Zawartość pliku `main.c`

Krótki opis wybranych fragmentów kodu.

- Początkowe ~380 linii kodu to funkcje odpowiedzialne za poprawną inicjalizację płytki.
- Funkcja `rx_udp_port_app_handler()`: Wywoływana po odebraniu pakietu UDP adresowanego na port zadany przez użytkownika.
- Funkcja `icmp6_handler()`: Wywoływana po odebraniu pakietu ICMPv6.
- Funkcja `button_event_handler()`: Wywoływana po wciśnięciu dowolnego z przycisków na płycie.

Dokumentacja

Dostępna lokalnie w pobranym nRF5 SDK lub online pod adresem <http://infocenter.nordicsemi.com/>. Sugerowane sekcje dokumentacji przydatne w pracy nad zadaniem:

- Omówienie bibliotek związanych ze stosem IPv6
Software Development Kit → nRF5 SDK → nRF5 SDK v15.0.0 → Libraries → IoT libraries → Experimental IoT libraries → Nordic's IPv6 stack
- Omówienie biblioteki odpowiedzialnej za alokację buforów np. na pakiety UDP
Software Development Kit → nRF5 SDK → nRF5 SDK v15.0.0 → Libraries → IoT libraries → Experimental IoT libraries → Packet Buffer
- Dokumentacja API
Software Development Kit → nRF5 SDK → nRF5 SDK v15.0.0 → API Reference → IoT components