

Technologie telekomunikacyjne 2020

Raport

BLE HID Controller

Jan Wołowiec

Wojciech Mrzygłód

Paweł Wargacki

Spis treści

1	Opis ogólny	3
1.1	Uzyskane rezultaty	3
2	Wykorzystany sprzęt	3
3	Konfiguracja środowiska programistycznego dla NRF51822	4
3.1	Instalacja oprogramowania	4
3.2	Klonowanie repozytorium	4
3.3	Przygotowanie projektu do kompilacji	4
3.4	Kompilowanie programu i uruchamianie na sprzęcie	4
3.5	Debugowanie programu	4
4	Konfiguracja środowiska programistycznego dla aplikacji na Windows	4
4.1	Wymagane oprogramowanie	4
4.1.1	Uruchomienie aplikacji	4
4.1.2	Kompilacja aplikacji	4
5	Szczegóły implementacji aplikacji na NRF51822	5
5.1	HID over BLE	5
5.2	UART over BLE	6
5.3	Akcelerometr	6
6	Szczegóły implementacji aplikacji na Windows	7
6.1	Cel powstania aplikacji	7
6.2	Implementacja protokołu Nordic Uart Service.	7
6.3	Użycie biblioteki	7
7	Testy opóźnienia komunikacji bluetooth	8

1 Opis ogólny

Projekt przewidywał wykorzystanie modułu Waveshare Kit ewaluacyjny z NRF51822 Bluetooth 4.0 do budowy kontrolera bezprzewodowego.

1.1 Uzyskane rezultaty

- Kontroler zgłasza się w systemie jako Gamepad dzięki zastosowaniu odpowiedniego deskryptora, dane są pobierane przez magistralę I2C z akcelerometru MPU6050.
- Kontroler pozwala na przesyłanie tekstu z komputera poprzez implementację UART'a od NRF i wyświetlanie ich na wyświetlaczu HD44780. Zostały dodane odpowiednie narzędzie uzyskujące wymagane dane z gry i przesyłające je po sformatowaniu na wyświetlacz.
- Kontroler obsługuje 2 przyciski obecne na płytce jako przyciski Gamepad'a
- Kontroler rozgłasza się po BLE i pozwala na parowanie z urządzeniem.

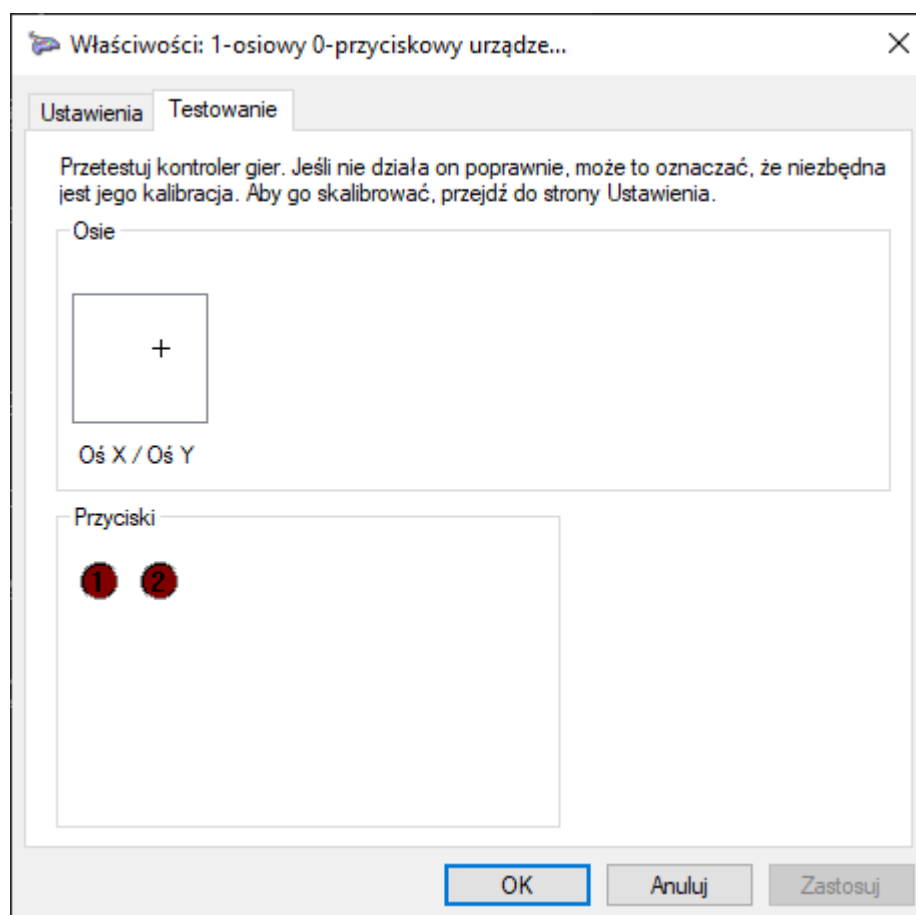


Figure 1 Urządzenie widoczne w systemie

2 Wykorzystany sprzęt

- Waveshare Bluetooth 4.0 NRF51822 Eval Kit
- J-Link programmer
- MPU6050 accelerometer module
- HD44780 LCD
- BLE compatible Windows PC

3 Konfiguracja środowiska programistycznego dla NRF51822

3.1 Instalacja oprogramowania

Do pisania i kompilowania programów na procesor NRF51822 potrzebne są następujące komponenty programowe:

- Git
- mingw32 – z tego pakietu używane jest narzędzie **make** automatyzujące przebieg kompilacji, po zainstalowaniu należy dodać folder /bin/ do zmiennej środowiskowej PATH
- arm-gcc – kompilator GCC pozwalający kompilować programy na architekturę ARM,
- nRF Command Line Tools – narzędzie do programowania układu NRF51822,
- oprogramowanie do programatora J-Link – element opcjonalny - wymagany do debugowania programu, dostępny na stronie producenta, po zainstalowaniu należy dodać ścieżkę instalacji do zmiennej środowiskowej PATH,
- Edytor Visual Studio Code z rozszerzeniem cortex-debug – używany do pisania i debugowania programu, po instalacji należy dodać ścieżkę instalacji pakietu arm-gcc do zmiennej **cortex-debug.armToolchainPath** w ustawieniach programu (settings.json).

3.2 Klonowanie repozytorium

- W katalogu do którego będzie klonowane repozytorium w terminalu wpisujemy:
 - `git clone https://github.com/NVi5/NRF_BLE_HID_Controller`

3.3 Przygotowanie projektu do kompilacji

- W pliku /ble_hid_controller/Paths do zmiennej GNU_INSTALL_ROOT wpisujemy ścieżkę instalacji pakietu arm-gcc.

3.4 Kompilowanie programu i uruchamianie na sprzęcie

- Otwieramy folder repozytorium w edytorze Visual Studio Code,
- Kompilujemy program i programujemy procesor poprzez **Terminal > Run Build Task...** lub skrót **Ctrl+Shift+B**.

3.5 Debugowanie programu

- W edytorze wybieramy opcję **Debug > Start Debugging** lub klawisz **F5**.

4 Konfiguracja środowiska programistycznego dla aplikacji na Windows

Wszystkie poniższe komponenty nie wymagają dodatkowej konfiguracji po instalacji.

4.1 Wymagane oprogramowanie

4.1.1 Uruchomienie aplikacji

- Python 3 64-bit.

4.1.2 Kompilacja aplikacji

- Visual Studio 2019,
- Visual Studio C++ development tools,
- Windows Driver Kit.

5 Szczegóły implementacji aplikacji na NRF51822

5.1 HID over BLE

W przypadku wykorzystywanego układu NRF51822, producent dostarcza biblioteki służące do obsługi tego typu komunikacji co w dużym stopniu ułatwia konfigurację tej funkcjonalności. Z racji wykorzystania BLE jest to implementacja wykorzystująca GATT, należy zdefiniować serwis HID o identyfikatorze UUID 0x1812, który będzie rozgłaszany przez nasze urządzenie.

Aby komputer wyświetlał odpowiednią ikonę przy naszym urządzeniu można również ustawić BLE Appearance na Gamepad.

Kolejnym krokiem jest konfiguracja samego serwisu HID. Należy podać sam deskryptor który zostanie przesłany do sparowanego urządzenia. W naszym przypadku identyfikujemy się jako gamepad posiadający 2 osie, X i Y oraz 2 przyciski. Dodatkowo biblioteka wymaga aby podać report id który wykorzystujemy oraz maksymalny rozmiar danych które otrzymamy, co prawdopodobnie pozwala uniknąć parsowania deskryptora. Po skonfigurowaniu reszty wymaganych parametrów możemy uruchomić serwis i wysyłać dane korzystając z właściwych funkcji.

Dane poprzez HID są wysyłane okresowo co 33ms, następuje wtedy odczyt danych z akcelorometru jak i stanu przycisków.

```
0x05, 0x01, // USAGE_PAGE (Generic Desktop)
0x09, 0x05, // USAGE (Game Pad)
0xa1, 0x01, // COLLECTION (Application)
0x85, 0x01, // REPORT_ID (1)
0x09, 0x01, // USAGE (Pointer)
0xa1, 0x00, // COLLECTION (Physical)
0x05, 0x01, // USAGE_PAGE (Generic Desktop)
0x09, 0x30, // USAGE (X) - Left Analog Left(-ve), Right(+ve)
0x09, 0x31, // USAGE (Y) - Left Analog Up(-ve), Down(+ve)
0x15, 0x81, // LOGICAL_MINIMUM (-127)
0x25, 0x7f, // LOGICAL_MAXIMUM (127)
0x75, 0x08, // REPORT_SIZE (8)
0x95, 0x02, // REPORT_COUNT (2)
0x81, 0x02, // INPUT (Data,Var,Abs)
0xc0, // END_COLLECTION
0x05, 0x09, // USAGE_PAGE (Button)
0x19, 0x01, // USAGE_MINIMUM (Button 1)
0x29, 0x02, // USAGE_MAXIMUM (Button 2)
0x15, 0x00, // LOGICAL_MINIMUM (0)
0x25, 0x01, // LOGICAL_MAXIMUM (1)
0x95, 0x02, // REPORT_COUNT (2)
0x75, 0x01, // REPORT_SIZE (1)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x95, 0x01, // REPORT_COUNT (1)
0x75, 0x06, // REPORT_SIZE (6)
0x81, 0x43, // INPUT (Cnst,Var,Abs,NULL)
0xc0 // END_COLLECTION
```

Figure 2 Deskryptor HID

5.2 UART over BLE

Z racji że BLE w przeciwieństwie do klasycznego Bluetooth'a nie obsługuje natywnie protokołu RFCOMM, dostępne są jedynie rozwiązania dostarczane przez producentów. W przypadku układu NRF51 dostępna jest usługa NUS(Nordic UART Service), która zawiera 2 charakterystyki RX i TX działające w podobny sposób do natywnego połączenia Bluetooth. Główną różnicą jednak jest to że w przeciwieństwie do automatycznie rozpoznawanego przez system protokołu RFCOMM, w tym wypadku należy stworzyć własne sterowniki pozwalające na zapis i odczyt do charakterystyk. Z racji braku gotowych bibliotek obsługujących BLE pod systemem Windows byliśmy zmuszeni wykorzystać WinAPI.

W przypadku dodawania serwisu NUS, wygląda to podobnie jak w przypadku serwisu HID. Najpierw należy zdefiniować UUID serwisu odsyłanego w odpowiedzi na scan response. W tym wypadku podobnie dysponujemy bibliotekami dostarczonymi przez producenta więc konfiguracja sprowadza się w głównej mierze do dodania handlera który będzie wywoływany w przypadku otrzymania danych przez układ.

Dodatkowo w tym kroku został skonfigurowany klasyczny UART, aby ułatwić nam debugowanie programu.

5.3 Akcelerometr

Moduł akcelerometru z którego skorzystaliśmy to MPU6050 wykorzystujący magistralę I2C do komunikacji. W przypadku NRF51 magistrala ta nosi nazwę TWI, jej konfiguracja korzystając z bibliotek sprowadza się w dużej mierze do podania pinów SDA i SCL oraz wykorzystywanej częstotliwości.

W ramach samej komunikacji z układem tworzy się struktury danych, które są potem kolejgowane do niego. Aby uruchomić sam akcelerometr należy wyłączyć tryb uśpienia w rejestrze konfiguracyjnym. Dane które możemy z niego odczytać to między innymi odczyty 3 osi akcelerometru i 3 osi żyroskopu. W naszym przypadku zastosowaliśmy najprostsze podejście i kąty nachylenia są obliczane jedynie na podstawie danych z akcelerometru, z racji trudności w implementacji filtru Kalmana lub konieczności całkowania odczytów z żyroskopu.

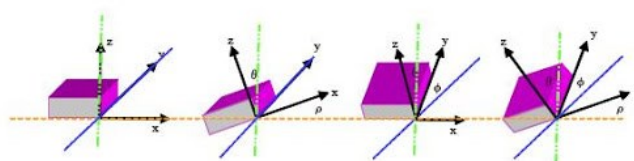


Figure 8. Three Axis for Measuring Tilt

$$\rho = \arctan\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right)$$

$$\phi = \arctan\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right)$$

$$\theta = \arctan\left(\frac{\sqrt{A_x^2 + A_y^2}}{A_z}\right)$$

Figure 3 Wykorzystane równania do obliczenia kąta nachylenia akcelerometru

6 Szczegóły implementacji aplikacji na Windows

6.1 Cel powstania aplikacji.

W projekcie do komunikacji PC -> Gamepad wykorzystany jest protokół Nordic Uart Service emulujący port szeregowy poprzez interfejs Bluetooth Low Energy.

Nordic Uart Service Jest to niestandardowa usługa BLE stworzona przez Nordic Semiconductor, więc nie jest domyślnie obsługiwana przez żaden system operacyjny. Co prawda producent udostępnia aplikacje na systemy Android, iOS i Windows, jednak ich funkcjonalność ogranicza się do ręcznego wpisania przesyłanej wiadomości, co więcej aplikacja na system Windows współpracuje tylko z adapterem USB Bluetooth od Nordic.

Celem powstania aplikacji na Windows była możliwość wyświetlania informacji na wyświetlaczu gamepada wykorzystując dowolny adapter Bluetooth oraz dowolne narzędzie/środowisko np. C, C++, Python czy LabVIEW. W naszym przypadku został wybrany Python.

Biblioteka umożliwiająca komunikację została zaimplementowana jako plik *.dll z użyciem języka C++ i winAPI, co umożliwia wykorzystanie jej we wspomnianych narzędziach/środowiskach.

6.2 Implementacja protokołu Nordic Uart Service.

Serwis Bluetooth Low Energy Nordic Uart Service (identyfikator UUID: 6E400001-B5A3-F393-E0A9-E50E24DCCA9E) składa się z dwóch charakterystyk:

- RX Characteristic (UUID: 6E400002-B5A3-F393-E0A9-E50E24DCCA9E) – do odbierania danych z układu NRF51822,
- TX Characteristic (UUID: 6E400003-B5A3-F393-E0A9-E50E24DCCA9E) – do przesyłania danych do układu NRF51822.

W bibliotece zaimplementowana jest wyłącznie TX Characteristic, komunikacja w kierunku RX poprzez HID obsługiwana jest przez system operacyjny.

Działanie biblioteki wygląda następująco:

- Wyszukanie urządzenia po nazwie i sprawdzenie czy jest w danym momencie połączone z komputerem,
- Wylistowanie wszystkich serwisów Bluetooth Low Energy obsługiwanych przez urządzenie,
- Znalezienie serwisu o UUID odpowiadającemu Nordic Uart Service UUID.
- Wylistowanie wszystkich charakterystyk serwisu.
- Znalezienie charakterystyki o UUID odpowiadającym TX Characteristic UUID.
- Wysłanie wiadomości do urządzenia.

6.3 Użycie biblioteki

Biblioteka NordicNUS_win32.dll udostępnia dla użytkownika następujące funkcje:

- `int OpenBleNusHandle()` – nawiązuje połączenie z urządzeniem. W przypadku poprawnego połączenia zwraca 0, w przeciwnym przypadku -1,

- `int SendNusMessage(char* message, unsigned int len)` – wysyła do urządzenia wiadomość o treści `message` i długości `len`. W przypadku powodzenia wysłania wiadomości zwraca 0, w przeciwnym przypadku -1.

7 Testy opóźnienia komunikacji bluetooth

Testy opóźnienia komunikacji bluetooth zostały wykonane przy pomocy programu Microsoft Message Analyzer. Oprogramowanie pozwala na analizy różnych interfejsów komunikacji PC takich jak Ethernet, USB, Bluetooth.

Pierwszy test miał na celu zasymulowanie idealnych warunków pracy urządzenia. Kontroler został położony w odległości kilkudziesięciu cm od urządzenia PC, w pobliżu nie działało żadne inne urządzenia Bluetooth. W tabelce poniżej widnieją opóźnienia kolejnych paczek danych kontrolera bluetooth

61358	2020-11-17T16...	0,0005359	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61360	2020-11-17T16...	0,0008478	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61362	2020-11-17T16...	0,0136493	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61364	2020-11-17T16...	0,0005554	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61366	2020-11-17T16...	0,0008161	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61368	2020-11-17T16...	0,0136402	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61370	2020-11-17T16...	0,0005290	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61372	2020-11-17T16...	0,0008277	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61374	2020-11-17T16...	0,0136518	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61376	2020-11-17T16...	0,0005222	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61378	2020-11-17T16...	0,0008278	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61380	2020-11-17T16...	0,0137142	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61382	2020-11-17T16...	0,0004797	4120	8980	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61384	2020-11-17T16...	0,0008073	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61386	2020-11-17T16...	0,0136396	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61388	2020-11-17T16...	0,0005066	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61390	2020-11-17T16...	0,0009087	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61392	2020-11-17T16...	0,0135540	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61394	2020-11-17T16...	0,0005116	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61396	2020-11-17T16...	0,0008357	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61398	2020-11-17T16...	0,0136349	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61400	2020-11-17T16...	0,0005374	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61402	2020-11-17T16...	0,0007944	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61404	2020-11-17T16...	0,0137090	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61406	2020-11-17T16...	0,0005271	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61408	2020-11-17T16...	0,0007805	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61410	2020-11-17T16...	0,0136530	14020	2492	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61412	2020-11-17T16...	0,0005350	14020	13740	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61414	2020-11-17T16...	0,0007936	14020	16772	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61416	2020-11-17T16...	0,0136755	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61418	2020-11-17T16...	0,0005146	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61420	2020-11-17T16...	0,0008367	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61422	2020-11-17T16...	0,0136382	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61424	2020-11-17T16...	0,0005429	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61426	2020-11-17T16...	0,0008336	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61428	2020-11-17T16...	0,0136432	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61430	2020-11-17T16...	0,0005073	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61432	2020-11-17T16...	0,0008392	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
61434	2020-11-17T16...	0,0136644	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI

Można zauważyć, że opóźnienie nie jest deterministyczne, jednak zawiera pewien schemat. Obserwujemy powtarzające się opóźnienia 0.5, 0.8, 13ms. Cykliczna natura różnych opóźnień danych związana jest z tzw. „frequency hopping”. Duże opóźnienie cykliczne prawdopodobnie związane jest z zajętością pasma w konkretnym zakresie częstotliwości.

Kolejny test miał na celu sprawdzenie opóźnienia danych przy zasymulowanej dużej zajętości pasma. Kontroler został położony w odległości kilkudziesięciu cm od urządzenia, dodatkowo w pobliżu został włączony transfer plików z komputera do telefonu, dodatkowo została podłączona myszka bluetooth.

163057	2020-11-17T16...	0,0150602	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163072	2020-11-17T16...	0,0150377	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163084	2020-11-17T16...	0,0149192	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163096	2020-11-17T16...	0,0150137	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163107	2020-11-17T16...	0,0149769	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163119	2020-11-17T16...	0,0150277	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163122	2020-11-17T16...	0,0149816	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163126	2020-11-17T16...	0,0150298	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163128	2020-11-17T16...	0,0149797	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163130	2020-11-17T16...	0,0149865	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163132	2020-11-17T16...	0,0150017	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163135	2020-11-17T16...	0,0150306	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163155	2020-11-17T16...	0,0150106	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163162	2020-11-17T16...	0,0150249	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163173	2020-11-17T16...	0,0149831	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163183	2020-11-17T16...	0,0150425	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163196	2020-11-17T16...	0,0149446	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163210	2020-11-17T16...	0,0149918	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163221	2020-11-17T16...	0,0150090	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163244	2020-11-17T16...	0,0150425	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163260	2020-11-17T16...	0,0149733	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163275	2020-11-17T16...	0,0149408	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163286	2020-11-17T16...	0,0149952	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163304	2020-11-17T16...	0,0150368	4428	6792	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163323	2020-11-17T16...	0,0150660	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163339	2020-11-17T16...	0,0149204	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163366	2020-11-17T16...	0,0150650	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163380	2020-11-17T16...	0,0149938	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163396	2020-11-17T16...	0,0150411	14020	16772	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163408	2020-11-17T16...	0,0149988	14020	6768	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163419	2020-11-17T16...	0,0149619	14020	13712	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163434	2020-11-17T16...	0,0150463	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163445	2020-11-17T16...	0,0149728	14020	15448	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163464	2020-11-17T16...	0,0150232	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163470	2020-11-17T16...	0,0150600	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163474	2020-11-17T16...	0,0148939	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163480	2020-11-17T16...	0,0150501	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163487	2020-11-17T16...	0,0150116	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163503	2020-11-17T16...	0,0150327	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
163507	2020-11-17T16...	0,0149965	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI

Możemy zauważyć że maksymalne opóźnienie wzrosło do 15ms. Opóźnienie średnie wzrosło 3-krotnie.

W ostatnim teście dodatkowo została włączona komunikacja bluetooth w odświeżaczu powietrza znajdującym się w odległości kilku m od urządzenia PC.

168964	2020-11-17T16...	0,0149966	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
168967	2020-11-17T16...	0,0149948	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
168972	2020-11-17T16...	0,0150331	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
168978	2020-11-17T16...	0,0149876	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
168981	2020-11-17T16...	0,0150166	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
168985	2020-11-17T16...	0,0150190	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169000	2020-11-17T16...	0,0149741	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169004	2020-11-17T16...	0,0299989	14020	12384	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169007	2020-11-17T16...	0,0149905	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169010	2020-11-17T16...	0,0149906	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169015	2020-11-17T16...	0,0300867	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169017	2020-11-17T16...	0,0149151	2436	2416	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169019	2020-11-17T16...	0,0150056	4	708	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169026	2020-11-17T16...	0,0300121	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169035	2020-11-17T16...	0,0149816	2436	2416	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169059	2020-11-17T16...	0,0150039	4428	6792	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169068	2020-11-17T16...	0,0150123	1280	1588	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169079	2020-11-17T16...	0,0149873	4	708	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169090	2020-11-17T16...	0,0150660	2436	10660	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169099	2020-11-17T16...	0,0149341	2436	14256	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169112	2020-11-17T16...	0,0150006	4	708	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169126	2020-11-17T16...	0,0150023	4428	6792	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169137	2020-11-17T16...	0,0149940	4428	6792	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169147	2020-11-17T16...	0,0149977	12692	12696	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169166	2020-11-17T16...	0,0150010	2436	2416	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169171	2020-11-17T16...	0,0150054	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169182	2020-11-17T16...	0,0150068	11712	13792	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169191	2020-11-17T16...	0,0149985	4	708	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169202	2020-11-17T16...	0,0149964	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169215	2020-11-17T16...	0,0149965	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169225	2020-11-17T16...	0,0149924	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169236	2020-11-17T16...	0,0150082	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169246	2020-11-17T16...	0,0149971	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169256	2020-11-17T16...	0,0150393	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169266	2020-11-17T16...	0,0149525	14020	9644	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169289	2020-11-17T16...	0,0150288	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169300	2020-11-17T16...	0,0149801	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169309	2020-11-17T16...	0,0150022	4428	6792	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI
169320	2020-11-17T16...	0,0149975	0	0	Microsoft_Windows_BTH_BTHPORT	Odczytaj pakiet HCI

Możemy zauważyć że czasami opóźnienie wzrasta z 15ms do 30ms z powodu nowego urządzenia wysyłającego dane z małą częstotliwością