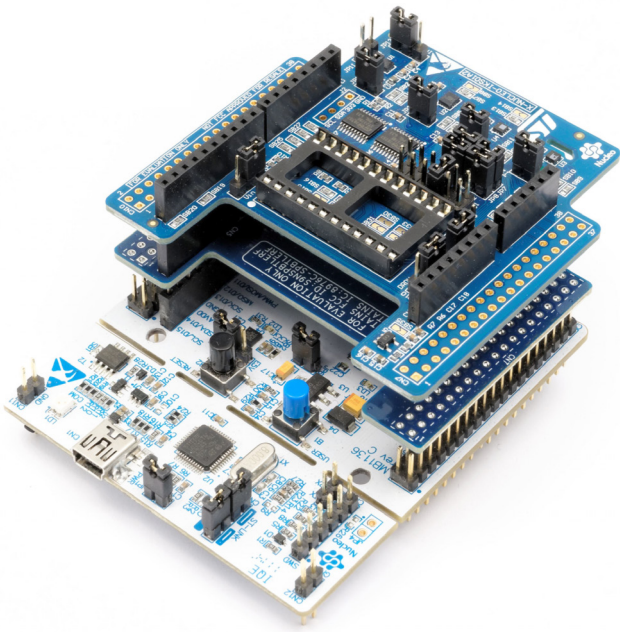
Cube dla STM32: biblioteki dla zestawu X-Nucleo-BLE



W artykule przedstawiamy kolejny „klocek” przygotowany przez firmę STMicroelectronics, także z myślą o użytkownikach środowiska STM32CubeMX. Przybliżymy w nim zestaw do komunikacji bezprzewodowej X-Nucleo-BLE.

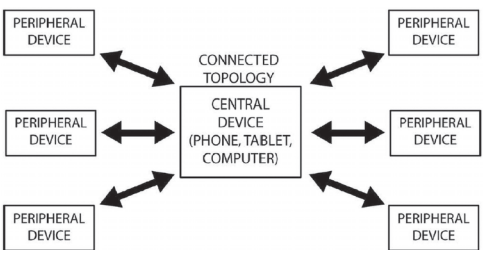
Jednym z fundamentów budowania i użytkowania urządzeń Internetu Rzeczy IoT jest transmisja bezprzewodowa, do której często używa się dwóch sprowadzonych standardów komunikacji: Wi-Fi i Bluetooth. Oba są bardzo dobrze znane i udokumentowane, a wielu producentów oferuje gotowe moduły transceiverów z fimware obsługującym stosy protokołów transmisyjnych.

BLE – podstawowe informacje

Przy transmisji na niewielkie odległości chętnie jest używany standard Bluetooth. Pierwsze wersje Bluetooth były optymalizowane do szybkiego przesyłania dużych ilości danych. Bluetooth 3.0 +HS zapewnia przepływność do 24 Mb/s. Wersja 4.0 została nazwana Bluetooth Low Energy (BLE). W nim priorytetem nie były duże prędkości transmisji danych, ale maksymalne ograniczenie poboru energii przez urządzenia komunikujące się za pomocą standardu BLE.

BLE korzysta z pasma ISM 2,4 GHz podzielonego na 40 kanałów o szerokości 2 MHz każdy. Kanały są logicznie podzielone na 2 grupy: 3 kanały *advertising* i 37 kanałów *data*. Kanały *data* są używane do dwukierunkowej transmisji danych pomiędzy urządzeniami w sieci. Kanały *advertising* biorą udział w procesie wyszukiwania urządzeń BLE znajdujących się w pobliżu, a potem w nawiązywaniu połączenia pomiędzy nimi.

Komunikacja w sieci BLE zaczyna się od rozgłaszania przez węzeł sieci (*advertiser*), że oczekuje na połączenia. Węzły sieci (*scanners*) odbierające komunikat żądanie na połączenie wysyłają komunikat z żądaniem połączenia (*connection request*). Ten komunikat musi zawierać dane niezbędne do zestawienia połączenia. Jeżeli połączenie zostanie nawiązane, to węzeł *scanners* staje się węzłem *master*, a węzeł rozgłaszający *advertiser* węzłem *slave*. Tworzy się w ten sposób podsieć pracująca w topologii gwiazdy, w której pojedynczy *master* może komunikować się z jednym lub więcej węzłami *slave*. Urządzenie pracujące jako węzeł *master* jest nazywane **Central Device**, a urządzenia *slave* – **Pheriperial Device** (**rysunek 1**). Central Device to najczęściej komputer, smartfon, tablet itp.

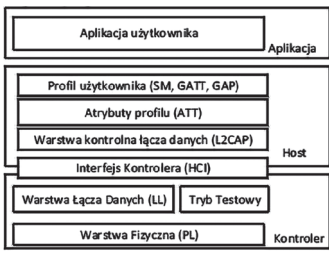


Rysunek 1. Podsieć BLE o topologii gwiazdy

Transmisję zawsze inicjuje węzeł *master*. Węzeł *slave* musi na przesłać w odpowiedzi informację zwrotną. Po nadaniu jednego pakietu danych musi minąć co najmniej 150 µs (*IFS Inter Frame Space*). Każdy z pakietów może zawierać wskaźnik **MD** (*More Data*) sygnalizujący konieczność przesłania kolejnych danych.

Protokół BLE ma budowę warstwową i składa się z warstwy fizycznej, warstwy łącza danych oraz warstwy HCI (*Host Controller Interface*). W warstwach wyższych są umieszczone **GATT** (*Generic Attribute Profile*) i **GAP** (*Generic Access Profile*).

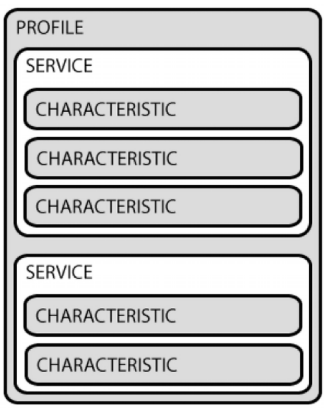
Struktura danych jest podzielona na: charakterystyki, serwisy i profile. Charakterystyki to najmniejsza „porcja” danych, serwis to zbór charakterystyk połączony logiczną zależnością, a profil to predefiniowana kolekcja charakterystyk. Profile są zatwierdzane przez Bluetooth SIG.



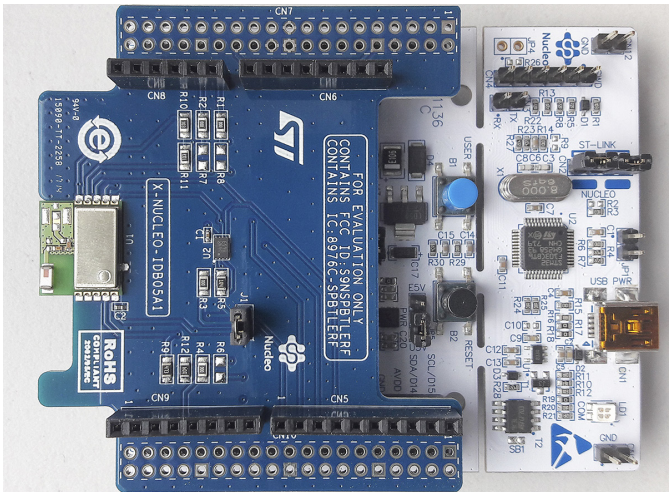
Rysunek 2. Budowa stosu BLE

Warstwa GATT jest zbudowana w oparciu o protokół Atrribute Protocol (ATT), który wykorzystuje dane GATT do określenia sposobu w jaki dwa urządzenia BLE wysyłają i odbierają standardowe wiadomości. GATT jest zbudowany w oparciu o role klienta i serwera.

Układy Peripherial są serwerami GATT (maja zapisane definicje charakterystyk i serwisów) , a układ Central pełni role klienta, bo wysyła żądania do serwera. W specyfikacji GATT jest opisanych wiele profili w tym na przykład pomiar ciśnienia krwi, monitorowania poziomu glukozy, mierzenia rytmu serca, pomiaru temperatury ciała itp.



Rysunek 3. Struktura danych BLE



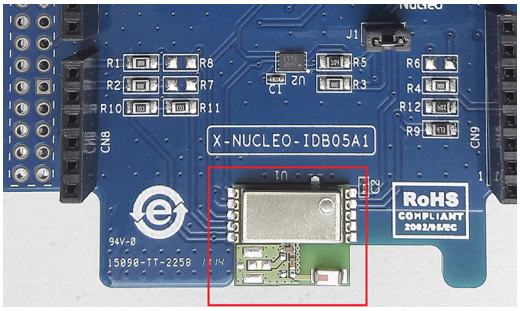
Fotografia 4. Płytka BLE X-Nucleo-IDB05A1 połączona z modułem Nucleo-F401RE

Moduł BLE X-Nucleo-IDB051A1

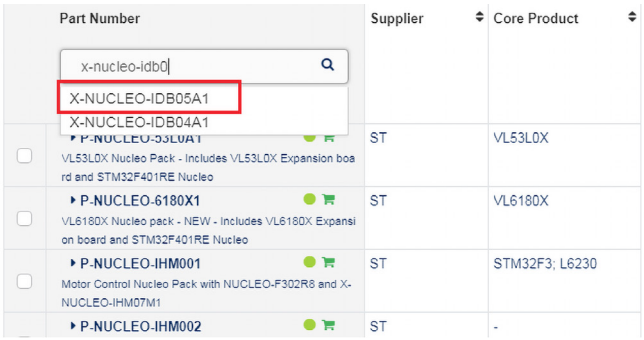
Płytka ewaluacyjna X-Nucleo-IDB051A1 jest wykonana zgodnie ze standardem Arduino R3 i współpracuje z firmowymi modułami STM z serii Nucleo (**fotografia 4**). Podstawowym elementem jest moduł Bluetooth Low Energy SPBLE-RF SPBLE-RF (**fotografia 5**) z wbudowanym procesorem BlueNRG-MS, komunikującym się z hostem przez szybki interfejs SPI.

Połączenia przez BLE z zestawu pokazanego na fot. 4 można szybko przetestować dzięki udostępnieniu przez STM programów demonstracyjnych X-CUBE-BLE1. Zaczynamy od wejścia na stronę ***www.st.com/x-nucleo*** i wybraniu modułu X-Nucleo-IDB05A1 tak jak to zostało pokazane na **rysunku 6**. Po wybraniu modułu trzeba pobrać ze strony spakowany program demonstracyjny en.X-Cube-BLE1. Po rozpakowaniu możemy korzystać z programów demonstracyjnych z gotowymi projektami dla środowisk projektowych: IDE Keil uVision, IAR EWARM, AC6. Oprócz tego, dla każdego z przykładów mamy do dyspozycji gotowe skompilowane pliki wynikowe. Można skorzystać z zamieszczonych plików wynikowych, lub otworzyć projekty w wybranym środowisku IDE i ewentualnie je modyfikować do własnych potrzeb. Niestety nie ma tu projektów dla Atollic True Studio for STM32. Próby szybkiej konwersji projektów za pomocą narzędzia Open Project from File System się nie powiodły. Przekonwertowany projekt dla AC6 opartego o Eclipse, nie dawał się kompilować, a projekt dla EWARM nie działał po skompilowaniu. Być może potrzebne były niewielkie poprawki, ale ja poprzestałem tylko na sprawdzeniu konwersji bez modyfikacji projektów. W czasie testów korzystałem z gotowych plików wynikowych.

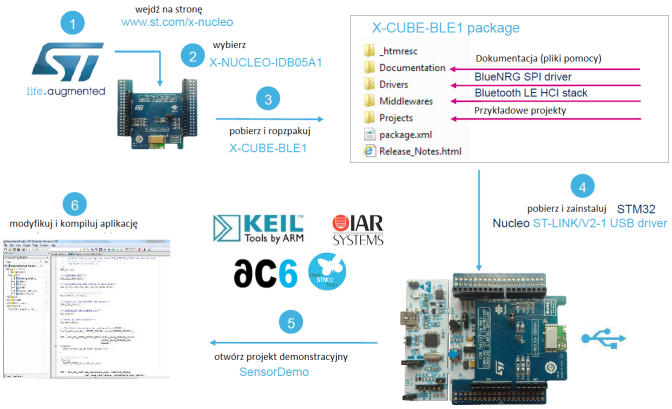
Na **rysunku 7** pokazano schematycznie proces pobierania i uruchamiania programów demonstracyjnych. Oprogramowanie demonstracyjne bazuje na warstwie abstrakcji sprzętu STM32CubeHAL dla mikrokontrolerów STM32 (**rysunek 8**). Aplikacja wykorzystuje również pakiet wsparcia BSP (Board Support Package) dla płytki Nucleo, oraz dla płytki rozszerzenia BlueNRG/BlueNRG-MS. Zastosowany tu procesor BLE BlueNRG-MS charakteryzuje się bardzo małym



Fotografia 5. Moduł Bluetooth Low Energy SPBLE-RF SPBLE-RF



Rysunek 6. Wybór typu modułu w wyszukiwarce *www.st.com*

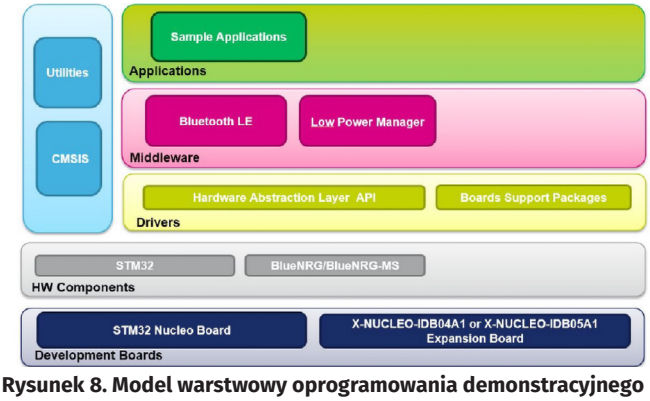


Rysunek 7. Pobieranie i uruchamianie programów demonstracyjnych

poborem mocy, a jego firmware jest zgodne ze specyfikacją Bluetooth 4.0/4.1. Warstwa sterowników (Drivers) zapewnia komponentom warstw wyższych (Middleware) dostęp do urządzenia BlueNRG-MS niezależnie od szczegółów sprzętowych. Inaczej mówiąc warstwy wyższe nie musza znać budowy i szczegółów sterowania procesora BLE. Middleware Low Power Manager optymalizuje pobór mocy.

Pakiet oprogramowania zawiera szereg przykładowych aplikacji. Kiedy testowany układ pełni rolę układu peryferyjnego (peripherial device - slave), to wspierana jest obsługa profili specyfikacji GATT: Alert Notification Client, Blood Pressure Sensor, Find Me Locaqtor, Find Me Target, Glucose Sensor, Health Thermomenter, Heart Rate, Human Interface Device, Phone Alert Client, Proximity Monitor, Proximity Reporter, Timer Server. W przypadku pełnienia roli układu centralnego (Central Device – Master) wpierana jest obsługa profili: Alert Notification Client, Blood Pressure Collector, Find Me Locator, Glucose Collector, Health Thermometer Collector, Heart Rate Collector, Time Client.

Do testowanie działania łącza BLE będzie potrzebna płytka BLE X-Nucleo-IDB05A1 połączona z modułem Nucleo F-401RE, oraz smartfon z pobraną i zainstalowaną aplikacją STM32 BLE Profiles. Wszystkie programy demonstracyjne obsługujące profile peripherial device symulują dane z czujników zewnętrznych. Z oczywistych względów

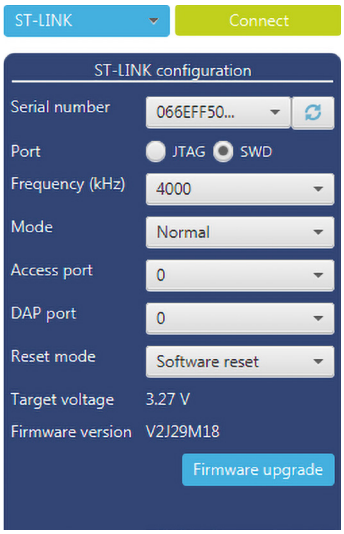


Rysunek 8. Model warstwowy oprogramowania demonstracyjnego

trudno byłoby testować rzeczywiste pomiary takich parametrów jak poziom glukozy, czy ciśnie krwi.

Tak jest też w wypadku pierwszego wykonywanego testu symulującego pomiar stężenia glukozy we krwi. Test zaczynamy od zaprogramowania mikrokontrolera w module Nucleo F-401RE plikiem wynikowym ProfPerip.GlucoseSensor_F401RE.bin umieszczonym w katalogu Projects/Multi_Applications-Profiles_LowPower/Binary/STM32F401RE_Nucleo. Do programowania pamięci można użyć programu STM32CubeProgrammer. Po podłączeniu modułu do komputera przez złącze USB trzeba kliknąć na przycisk Connect w oknie ST_LINK. W oknie ST-LINK configuration zaznaczamy opcję SWD, jak na **rysunku 9**.

Po kliknięciu na Connect następuje połączenie z układem programatora/debuggera na płytce Nucleo i można przejść do programowania. Najpierw wybieramy plik binarny (lub w formacie .hex) i po jego otwarciu klikamy na Start Programming (**rysunek 10**). Po zaprogramowaniu mikrokontrolera trzeba uruchomić aplikację STM32 BLE Profiles na smartfonie z aktywnym łączem Bluetooth. Aplikacja jest dostępna bezpłatnie w sklepie Google Play (dla systemu Android). Jeżeli Bluetooth nie jest aktywny, to aplikacja poprosi o jego włączenie. Po nawiązaniu łączności pomiędzy urządzeniem centralnym (smartfon), a urządzeniem peryferyjnym (płytka) aplikacja STM32 BLE Profiles zgłasza wykrycie profilu. Jak wspominałem w naszym przypadku będzie to pomiar poziomu glukozy (**fotografia 11**).

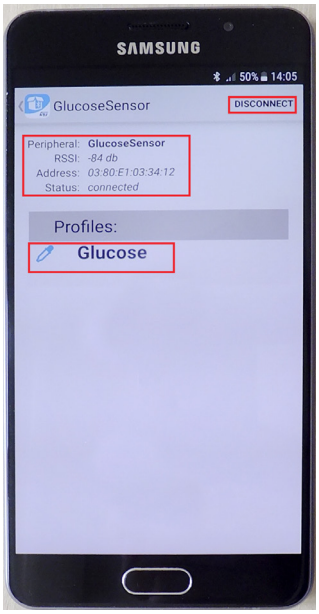


Rysunek 9. Okno obsługi ST-LINK programu STM32CubeProgrammer

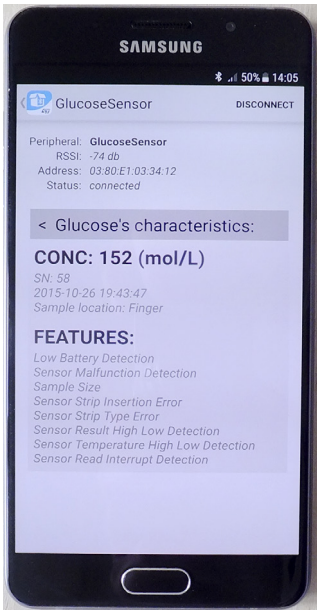
Na ekranie jest wyświetlana:

- Nazwa układu peryferyjnego: GlucoseSensor.
- Poziom sygnału radiowego – 74 dB.
- Adres.
- Status połączenia – connected.
- Nazwa profilu = Glucose.

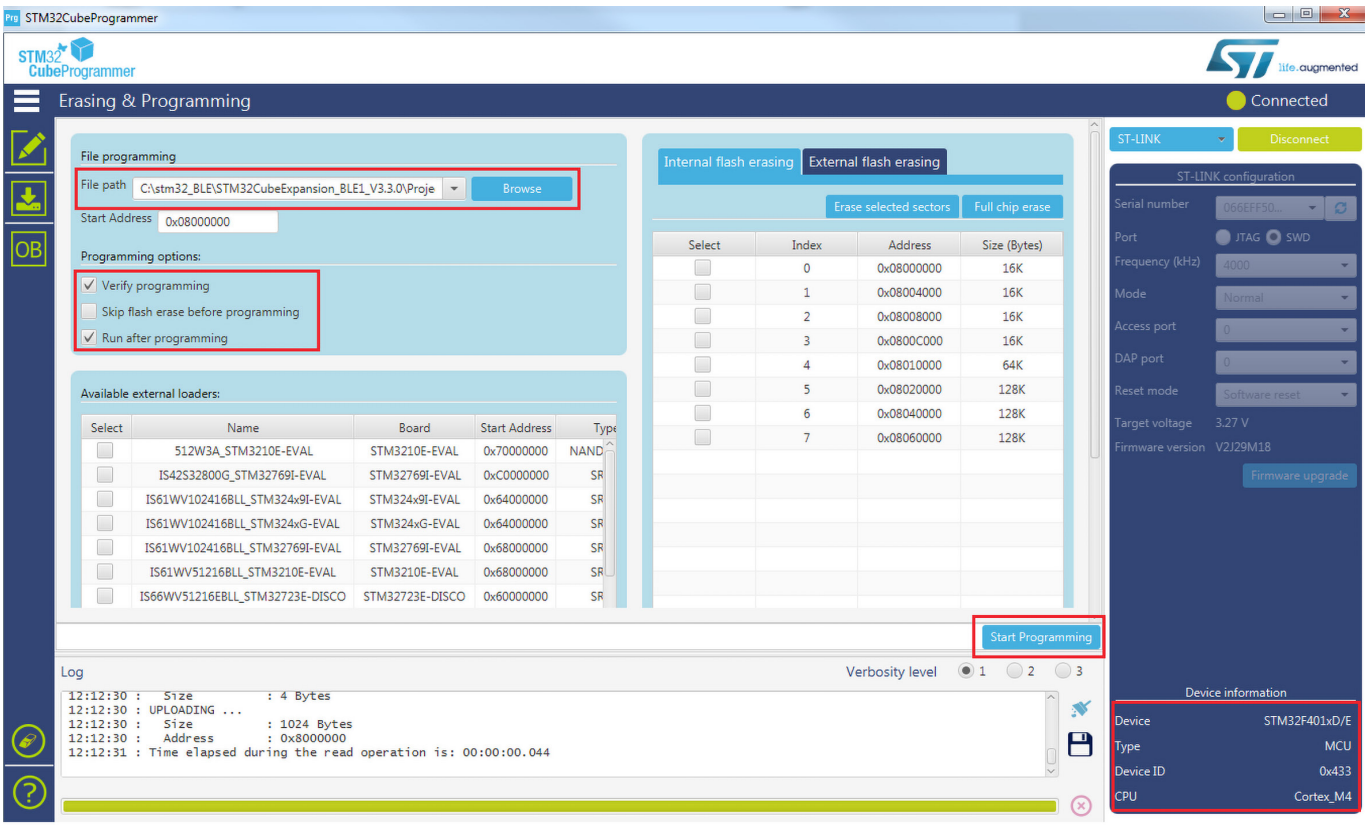
Można tez tu rozłączyć połączenie dotykając ikony Disconnect i ponownie je nawiązać naciskając Connect. Po naciśnięciu na nazwę profilu (Glucose) aplikacja przechodzi do wyświetlania symulowanego pomiaru jak na **fotografii 12**. Jak już wiemy dane o poziomie glukozy są zapisane w programie demonstracyjnym. Mamy tu wartość pomiaru (152 mol/L), datę i godzinę pomiaru, oraz miejsce pobrania próbki (z palca). W dolnej części ekranu są wyświetlane właściwości profilu.



Fotografia 11. Ekran smartfona z wykrytym profilem BLE



Fotografia 12. Ekran z wyświetlanymi symulowanymi danymi pomiarowymi



Rysunek 10. Programowanie za pomocą STM32CubeProgrammer

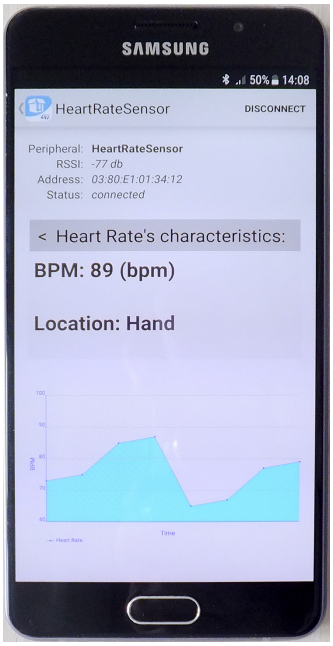
Po wgraniu plików wynikowych ze skompilowanych projektów można przetestować działanie połączenia i symulacji przesyłania danych z kolejnych czujników. Na **fotografii 13** pokazano ekran z pomiarem tętna z czujnikiem umieszczonym na ręce. Zmiany tętna są pokazywane w formie graficznej na wykresie. Kolejny przykład, to pomiar ciśnienia pokazany na **fotografii 14**.

Podobnie sprawdziłem wszystkie programy demonstracyjne z katalogu Projects/Multi_Applications-Profiles_LowPower/Binary/STM32F401RE_Nucleo.

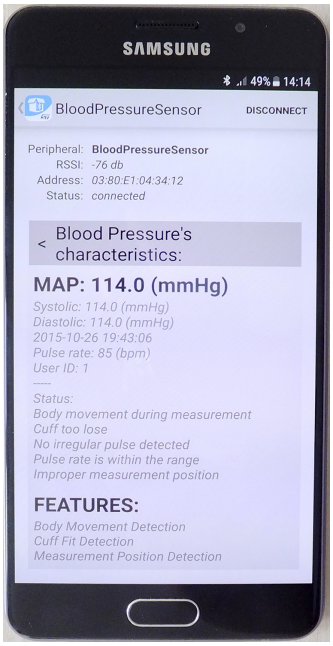
Programów demonstracyjnych jest o wiele więcej. Można testować miedzy innymi aplikacje z katalogu SampleAppThT. Potrzebne są dwa zestawy płytek BLE X-Nucleo-IDB05A1 połączonych z modułem Nucleo F-401RE. Jeden jest skonfigurowany jako central, a drugi jako Peripherial. Naciśnięcie przycisku User na jednym z modułów zapala i gasi diodę LED na drugim z modułów (i vice versa). Doświadczenia z przeprowadzonych testów pozwalają sadzić, że pozostałe tez będą działać poprawnie.

Przykładowe, działające programy z projektami zawierającymi kompletne pliki źródłowe mogą być doskonałą bazą do opracowywania własnych projektów. Dołączenie czujników mierzących rzeczywiste parametry pozwala na błyskawiczne tworzenie użytecznych aplikacji użytkownika przy minimalnym nakładzie pracy własnej.

Tomasz Jabłoński, EP



Rysunek 13. Symulacja pomiaru tętna



Rysunek 14. Symulowany pomiar ciśnienia tętniczego