

POLITECHNIKA WROCŁAWSKA

PROJEKT ZESPOŁOWY

SYSTEM OBSŁUGI ZAJĘĆ GRUPOWYCH

System do treningu TRIAGE dla ratowników medycznych

Autorzy:

Adam PEĐRACKI 241693

Marta DWORAKOWSKA 235645

Grzegorz HUCHŁA 241265

Paweł PARCZYK 241390

Krzysztof JAROS 226745

Prowadzący:

Dr inż. Jan NIKODEM

29 kwietnia 2020

Spis treści

1	Opis zadania	3
2	Wymagania	4
3	Założenia	5
4	Opis systemu	5
4.1	Poszkodowani	5
4.2	Triażyci	5
4.3	KAM	5
4.4	Instruktor	6
4.5	Zestawienie i charakterystyka sprzętu	6
5	Harmonogram prac na 19.03 - 2.04	6
6	Sieć	7
6.1	Technologia Bluetooth Low Energy	7
6.2	Zasięg BLE	7
6.3	Przebieg transmisji	7
6.4	Topologia sieci	8
6.5	BLE 4.2 vs BLE 5.0	8
7	Wybór sprzętu	9
7.1	Kryteria i wybór	9
7.2	Arduino	9
7.2.1	NANO	9
7.2.2	UNO	10
7.2.3	Inne modele arduino	10
7.3	Wyświetlacz	10
7.3.1	PCF	10
7.3.2	HD	10
7.4	Bluetooth	10
7.5	Zasilanie	10
7.5.1	Przez usb	11
7.5.2	przez regulator napięcia w płytce	11
7.6	Schemat	11
8	Instrukcja instalacji android studio	12
9	Program techniczny - Android	13
9.1	Radio bluetooth	13
9.2	Tryb advertisement	19
10	Instrukcja instalacji Android IDE oraz biblioteki i2C	24

11 Program techniczny - Arduino	24
11.1 Modyfikacja parametrów	25
12 Źródła	26
13 Materiały	26

1 Opis zadania

Zakres prac	Poprawa informacji o procedurze Triage oraz systemie SMART w rozdziałach 1, 3, 4.
Wykonał	Krzysztof Jaros
Sprawdził	PAweł Parczyk
Zatwierdził	Paweł Parczyk

Celem projektu jest stworzenie narzędzia wspierającego ćwiczenia ratowników medycznych w treningu działań podejmowanych w trakcie zdarzeń masowych. Są to zdarzenia, w których zapotrzebowanie na kwalifikowaną pierwszą pomoc i medyczne czynności ratunkowe przekracza możliwości podmiotów ratowniczych obecnych na miejscu zdarzenia. Aby zapewnić przeżycie jak największej liczbie osób ratownicy stosują się do procedur.

W swoich działaniach ratownicy stosują się do zasad segregacji poszkodowanych w wypadku masowym, umożliwiającą kategoryzację rannych na podstawie funkcji życiowych zwanych linią życia. Przydział kategorii odbywa się po przez przypisanie jednego z czterech kolorów. Zielonego - dla osób nie wymagających interwencji lekarskiej, żółtego - osoby wymagające interwencji lekarza, czerwonego - osoby wymagające natychmiastowego przetransportowania do szpitala i czarny - osoby nie do odratowania. Zalecane jest aby na czynności z tym związane ratownik poświęcił maksymalnie 30 sekund.

Trening odbywa się w ograniczonej przestrzeni np. budynku, w hali sportowej o wymiarach 40m na 50m. W szkoleniu uczestniczą triażysty oraz kierujący akcją medyczną (KAM), wszystko nadzoruje instruktor. Do szkolenia przeznaczonych jest kilka symulatorów poszkodowanych, ułożonych w różnych punktach hali. Symulator poszkodowanego przekazuje triażystę wartości podstawowych funkcji życiowych, na podstawie której przydziela kategorię poszkodowanemu, a następnie przekazuje tę informację do KAMA.

Schemat kategoryzacji poszkodowanego:

1. Sprawdzenie czy poszkodowany chodzi, jeśli tak ratownik nadaje mu kategorię koloru zielonego
2. Jeśli poszkodowany nie chodzi, sprawdzamy czy oddycha, jeśli nie oddycha ratownik udrożnia drogi oddechowe, jeśli to nie pomoże przydzielana jest kategoria koloru czarnego.
3. Jeśli poszkodowany oddycha a liczba oddechów na minute jest mniejsza niż 8 lub większa niż 30 zostaje przydzielony do kategorii czerwonej.
4. Jeśli liczba oddechów mieści się w przedziale 8-30/minutę ratownik ocenia krążenie poprzez nawrot papilarny. Polega na uciśnięciu (z siłą wystarczającą do zblednięcia skóry)

przez 5 sekund paznokcia palca ręki położonego na wysokości serca. Po zwolnieniu ucisku mierzy się czas jaki upływa do momentu powrotu koloru otaczającej skóry. Jeśli czas ten jest większy od 2 sekund poszkodowany trafia do kategorii czerwonej

5. Jeśli badanie krążenia zakończy się pozytywnie, ratownik sprawdza czy poszkodowany reaguje na proste polecenia, jeśli tak poszkodowany otrzymuje kategorię koloru żółtego, jeśli nie czerwonego.

W przypadku treningu z naszym systemem, pomijamy sprawdzenie czy poszkodowany chodzi. Wartości takie jak: liczba oddechów na minutę, drożność dróg oddechowych, czas nawrotu papilarnego czy ocena przytomności będą wyświetlane na Baconie.

Instruktor za pomocą panelu sterowania ma wgląd w poczynania uczestników. Widzi czas w jakim przypisana została kategoria, kto ją przypisał i jaki kolor został przypisany a także na bieżąco monitoruje reakcje KAMa na spływające do niego informacje.

Triażysta na podstawie odczytanych wartości funkcji życiowych wypełnia formularz na telefonie z androidem będącym standardowym wyposażeniem ułatwiający mu szybki przydział kategorii, a także przyspiesza przepływ informacji do KAMa.

KAM podejmuje decyzje na podstawie danych zgromadzonych na wyświetlaczu tabletu, będącym standardowym wyposażeniem karetki.

2 Wymagania

1. Linia życia musi być w czytelny sposób przekazana triażystcie.
2. Wypełnione formularze muszą być na bieżąco przesyłane KAMowi.
3. Instruktor musi mieć możliwość zdalnej zmiany stanu symulatora w dowolnym momencie.
4. System korzysta z komunikacji bezprzewodowej.
5. Instruktor ma możliwość gromadzenia informacji o decyzjach i czasie ich podjęcia przez każdego szkolonego.
6. System umożliwia jednoczesny trening w grupie o liczebności co najmniej jednej załogi karetki.
7. Symulatory poszkodowanych muszą określać swoje współrzędne w hali, każdy symulator ma przydzielony identyfikator.
8. Symulatory poszkodowanego powinny mieć wystarczająco energii do nieprzerwanego działania przez jedną sesję szkoleniową(min 1,5h).

3 Założenia

1. Poszkodowany musi prezentować dane triażyste więc będzie zrealizowany przez mikrokomputer z wyświetlaczem (w dalszej części nazywany beaconem), na którym będą wyświetlane parametry życiowe poszkodowanego.
2. Urządzenia muszą komunikować się między sobą bezprzewodowo więc komunikacja między beaconami będzie zrealizowana z wykorzystaniem technologii bluetooth w wersji LE.
3. Wszyscy triażysty muszą mieć możliwość przekazywania formularzy KAMomwi oraz instruktorowi z dołączonymi dodatkowo informacjami o czasie ich wypełnienia, więc formularze zrealizowane będą przez dedykowaną aplikację na telefony komórkowe, będące na standardowym wyposażeniu ratownika medycznego.
4. KAM musi mieć możliwość odebrania formularzy, więc posłuży mu do tego aplikacja dedykowana na tablet będący standardowym wyposażeniem karetki.
5. Instruktor nadzorujący szkolenie ma do dyspozycji tablet z dedykowaną aplikacją, służącą jako panel sterowania.

4 Opis systemu

4.1 Poszkodowani

Poszkodowani (beacony) będą mikrokomputerami komunikującymi się bezprzewodowo. Będą wymieniały między sobą informacje o stanie całego systemu. Każdy beacon będzie wyposażony w wyświetlacz LCD informujący ratownika o stanie parametrów życiowych symulowanego pacjenta takich jak: liczba oddechów na minutę, drożność dróg oddechowych, czas nawrotu papilarnego, ocena przytomności.

4.2 Triażysty

Triażysta uczestniczący w szkoleniu będzie wyposażony w telefon, na którym działa będzie aplikacja pełniąca rolę terminala. Gdy ratownik podejdzie do beaconsa, wypełnia formularz w aplikacji na podstawie odczytanych danych. Aplikacja po zatwierdzeniu formularza przesyła go do KAMa oraz instruktora poszerzone o dane dotyczące czasu jego wypełniania.

4.3 KAM

KAM posiada tablet z aplikacją informującą go na bieżąco o liczebności poszkodowanych w każdej z kategorii, co ułatwia mu podejmowanie działań koordynacyjnych takich jak: kierowanie kolejnymi przybywającymi zespołami ratownictwa medycznego, przeprowadzenie segregacji, zaopatrzenie i transport poszkodowanych zgodnie z przyjętymi priorytetami.

4.4 Instruktor

Instruktor posiada tablet, z zainstalowaną aplikacją umożliwiającą mu monitorowanie przebiegu szkolenia oraz ustalanie stan beaconów.

4.5 Zestawienie i charakterystyka sprzętu

Jako beacons planujemy użyć mikrokontrolerów Arduino NANO BLE. Zdecydowaliśmy się na to rozwiązanie ze względu na niskie zapotrzebowanie energetyczne tych modułów oraz ich mały rozmiar, dodatkowo moduły te posiadają wbudowany chip nRF52840 pozwalający na komunikację bezprzewodową. Kolejnym elementem każdego beacona będzie wyświetlacz LCD. Mający rozmiar 4x20 pól znakowych, w celu zmieszczenia na nim wszystkich potrzebnych danych dotyczących stanu zdrowia symulowanego pacjenta. Dodatkowym wymaganiem będzie możliwość zasilenia wyświetlacza za pomocą napięcia 3.3V dostępnego na mikrokontrolerze. Zasilanie układu będzie realizowane za pomocą ładowalnych akumulatorów podłączonych do regulatora napięcia wbudowanego w płytke.

Ratownik będzie posługiwał się dowolnym smartphonem z bluetoothem w wersji co najmniej 4.2.

Będzie także potrzebny tablet dla kierownika akcji, który będzie obsługiwać technologię bluetooth w wersji co najmniej 5.0, ze względu na potrzebę dużego zasięgu transmisji.

5 Harmonogram prac na 19.03 - 2.04

1. Poprawienie dokumentacji
 - wykonawca: Paweł
 - sprawdzacz: Marta
2. Wszystko od strony poszkodowanego (sprzęt, kryteria wyboru sprzętu, dokonanie wyboru, zasialnie, schematy)
 - wykonawca: Adam, Grzegorz
 - sprawdzacz: Paweł
3. Wszystko od strony triażysty (transmisja ble, zasady (rodzaje, jak to przebiega, realizacja transmisji przez różne urządzenia, wraz ze zrozumieniem), program techniczny obsługi ble i komunikacja pomiędzy poszkodowanym a triażystą)
 - wykonawca: Marta, Krzysztof
 - sprawdzacz: Adam, Grzegorz
4. Oprogramowanie od strony KAMa (Android) i oprogramowanie instruktora

- wykonawca: Paweł
- sprawdzacz: Marta, Krzysztof

Pomocne linki: Jak zrobić hello world: <https://developer.android.com/training/basics/firstapp>

6 Sieć

Zakres prac	Sieć, Technologia Bluetooth Low Energy
Wykonał	Marta Dworakowska
Sprawdził	Krzysztof Jaros
Zatwierdził	Paweł Parczyk

6.1 Technologia Bluetooth Low Energy

Są dwie formy technologii Bluetooth - Basic Rate (BR) oraz LE (Low Energy). Wybraliśmy technologię Low Energy ze względu na oszczędność energii zużywanej na transmisję.

6.2 Zasięg BLE

Zasięg BLE zależy od klasy urządzenia implementującego technologię. Najbardziej powszechną, używaną w telefonach jest klasa 2. Urządzenia tej klasy mają w większości przypadków zasięg 5-10m (w dobrych warunkach max 20m). Zasięg urządzeń klasy 1 wynosi 20-30m (max 100m).

6.3 Przebieg transmisji

Częstotliwości używane przez Bluetooth są podzielone na 40 fizycznych kanałów oddzielonych od siebie o 2 MHz. 3 z nich używane są jako kanały rozgłoszeniowe, a pozostałych 37 jako kanały informacyjne. Fizyczny kanał podzielony jest na jednostki czasowe - zdarzenia (ang. events). Informacje przesyłane są podczas trwania zdarzeń. Urządzenie rozgłaszające (ang. advertiser) wysyła jeden pakiet rozgłaszający na jednym kanale, po czym zmienia kanał rozgłoszeniowy. Wysyłanie 3 pakietów na 3 kanałach to jedno zdarzenie. Urządzenia, które chcą nawiązać połączenie (ang. initiators) nasłuchują pakietów rozgłoszeniowych, jeżeli uda im się jeden odebrać wysyłają żądanie do advertiser'a o nawiązanie połączenia. Połączenie zostaje zaakceptowane, jeżeli advertiser zaakceptuje żądanie, wtedy iniciator staje się zarządzającym (ang. master), a advertiser podwładnym (ang. slave). Ich komunikacja odbywa się poprzez 37 kanałów informacyjnych. Informacje przesyłane są między nimi podczas eventów zachodzących w tych kanałach. Master inicjuje zajście każdego eventu. Wysyłanie informacji zachodzi naraz tylko w jednym kanale.

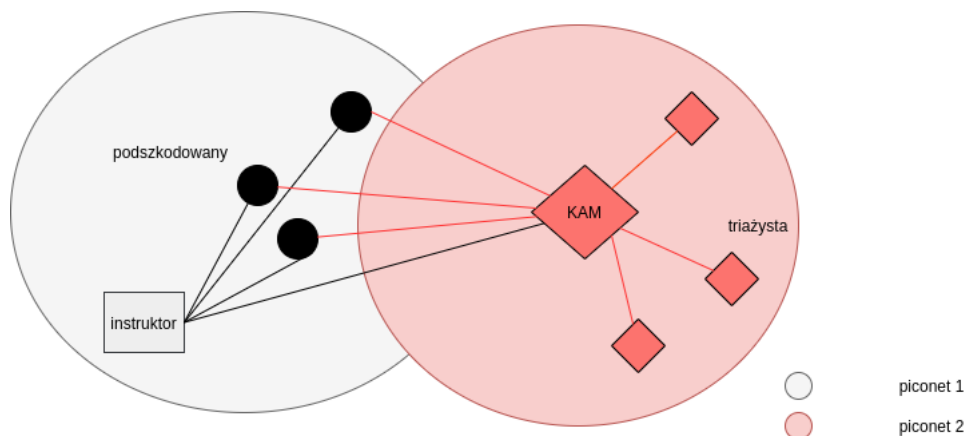
Master i slave tworzą sieć nazwaną piconet. W sieci tej zawsze występuje jeden master i co najmniej jeden slave. Liczba urządzeń slave, mogących być podpięta do jednej sieci piconet jest ograniczona przez zasoby fizyczne mastera. Urządzenia w sieci piconet używają schematu, według którego zmieniają kanały transmisji. Schemat ten jest algorytmicznie zdeterminowany przez mastera, który dostarcza slave'om mechanizm synchronizacji - ustalony czas, co który

urządzenia zmieniają kanał komunikacyjny. Schemat ten może być zmieniany, aby wyłączyć z transmisji kanały, które są mocno obciążone przez pośrednie urządzenia, nieuczestniczące w sieci piconet.

Komunikacja w sieci piconet zachodzi tylko między masterem i jego slave'ami. Master komunikuje się naraz z jednym slave'em, chyba że master pełni funkcję Broadcaster zdefiniowaną w profilu Bluetooth GAP. Jedno urządzenie może pełnić rolę mastera w jednej sieci piconet i slave'a w innej, jak i również mastera albo slave'a w dwóch różnych sieciach piconet. Sieć złożona z sieci piconet nazwana jest siecią scatternet. Zmiana ról w jednej sieci piconet nie jest możliwa.

6.4 Topologia sieci

Rysunek 1 przedstawia sieć scatternet użytą w projekcie.



Rysunek 1: Topologia sieci

- W sieci piconet 1 instruktor pełni funkcję nadajnika, a poszkodowani i KAM odbiornika.
- W sieci piconet 2 KAM pełni rolę nadajnika, a podszkodowani i triażysta odbiorników.

6.5 BLE 4.2 vs BLE 5.0

Dla BLE 5.0 dodano kilka ulepszeń, które mogą wpływać na jakość działania budowanego systemu informatycznego.

1. Dla technologii LE 4.2 przepływność symbolowa wynosi 1Mb/s dla niekodowanych korekcyjne danych, a z kodowaniem 500kb/s lub 125kb/s. Dla technologii LE 5.0 przepływność symbolowa dla niekodowanych danych może zostać zwiększona do 2Mb/s.
2. LE Channel Selection Algorithm 2 (CSA 2) - algorytm przeskakiwania między kanałami, który lepiej unika interferencji i zanikania fal radiowych (multipath fading).

W związku z tym, że technologia BLE 5.0 nie wprowadza wyraźnych ulepszeń w systemie rozwiązaniem pozostaje technologia BLE 4.2 z uwagi na większą dostępność i niższą cenę urządzeń na rynku.

7 Wybór sprzętu

Zakres prac	Wybór urządzeń oraz wykonanie schematu połączeń
Wykonał	Grzegorz Huchla
Sprawdził	Adam Pędracki
Zatwierdził	Paweł Parczyk

7.1 Kryteria i wybór

Naszym wyborem ze względu na koszt oraz potrzeby zostało Arduino UNO wraz z podłączonym wyświetlaczem oraz modułem ESP-32 odpowiadającym za komunikację Bluetooth. Wybór ten jest umotywowany posiadaniem przez nas kilku klonów tej płytki, co pozwoli zminimalizować koszty zakupu potrzebnego sprzętu. Do tej płytki zostanie podłączony także wyświetlacz pracujący za pomocą magistrali I2C, pracujący na zasilaniu 5V dostarczonym z Arduino.

7.2 Arduino

7.2.1 NANO

- Mniejsze niż Arduino Uno (18 x 45 mm)
- Mniejsze niż w przypadku UNO zużycie energii (ok. 160mW, w zależności od wersji)
- Cena ok 60zł za wersję standardową, 130 za wersję z BT 5.0
- Brak wyjścia 5v, tylko 3,3V
- Mała dostępność obudów gotowych
- Zasilanie 5V microusb albo przez pin i regulator napięcia (6-20V)
- Jest wersja z bt 5.0
- NANO EVERY Najtańsza wersja - ok 60zł, większe zużycie prądu niż w przypadku wersji 33, kosztuje ok 60 złotych i nie posiada wersji z BT.
- NANO zwykłe Droższa wersja (ok 90 zł), brak wersji z BT.
- NANO 33 Wersja z Bluetooth 130 zł, bez 120, są wersje z BT 5.0. Ich użycie niweluje potrzebę dodatkowego modułu, czyli zmniejszenie rozmiaru. Z całej rodziny NANO mają najmniejszy pobór energii

7.2.2 UNO

- Większy rozmiar (68.6 mm × 53.3 mm) niż w przypadku NANO
- Łatwy dostęp obudów i akcesoriów
- Cena ok 95zł
- normalne wyjście 5V
- Brak wersji z BT

7.2.3 Inne modele arduino

Pozostałe wersje Arduino nie posiadają wbudowanej komunikacji Bluetooth. Wybór został więc ze względów ekonomicznych sprowadzony do dwóch wyżej wymienionych wersji - NANO ze względu na wbudowaną komunikację Bluetooth 5.0 oraz UNO ze względu na to, że grupa posiada kilka klonów takiej płytki.

7.3 Wyświetlacz

Są 2 główne standardy podłączenia: PCF8574 i HD 44780.

7.3.1 PCF

Podłączany na dwa wejścia pwm + zasilanie + gnd.

Jest w Botlandzie wersja na 3,3V (pod arduino nano) ODR-15155, co pozwala podłączyć je bezpośrednio do zasilania w NANO. Na zasilanie 5v jest też np. LCD-02640.

7.3.2 HD

6 cyfrowych (wersja na 4 bit) albo 10 (na 8 bit), w Botlandzie nie znalazłem takich pod 3.3V, co wyklucza łatwe użycie z Arduino NANO. Na 5V jest na przykład LCD-04685.

7.4 Bluetooth

Z dwóch opcji: dokupienia osobnego modułu oraz użycia BT wbudowanego w płytkę, co sugeruje użycie NANO 33 BLE. Zdecydowaliśmy się na tą pierwszą opcję, ze względu na brak potrzeby użycia standardu 5.0. Wybrany przez nas modułem Bluetooth został ESP-32 posiadający wbudowany Bluetooth w standardzie 4.2.

7.5 Zasilanie

Znalazłem 2 opcje:

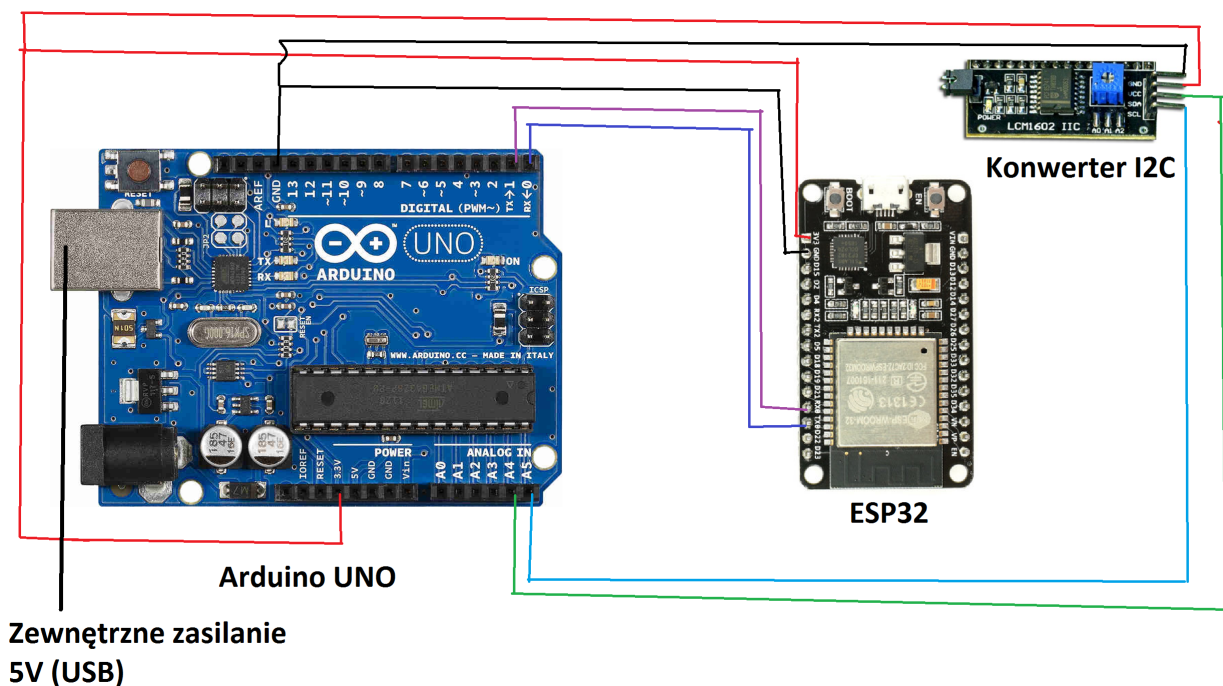
7.5.1 Przez usb

Do każdego pakietu dochodzi powerbank zasilający płytke przez microUSB, powoduje to łatwiejsze ładowanie przez zwykły kabel.

7.5.2 przez regulator napięcia w płytce

Płytka jest do zasilenia czymkolwiek z zakresu 6-21V, czyli można użyć akumulatorów, np 9V. Jest wtedy bardziej skomplikowane ładowanie modułów, potrzeba osobnej ładowarki do baterii.

7.6 Schemat



Rysunek 2: Schemat połączenia modułów do Arduino

Na schemacie zawarte zostały: mikrokontroler Arduino Uno, konwerter wyświetlacza (jest częścią całości modułu wyświetlacza, niezawartego na schemacie ze względu na duże rozmiary oraz brak połączeń realizowanych przez nas - wszystkie potrzebne połączenia pomiędzy sterownikiem a wyświetlaczem będą gotowe w zakupionych modułach) oraz moduł komunikacji Bluetooth ESP-32 połączony z Arduino za pomocą portu szeregowego.

8 Instrukcja instalacji android studio

Zakres prac	"Program techniczny - Android"i "Instrukcja instalacji android studio"
Wykonał	Paweł Parczyk
Sprawdził	Krzysztof Jaros
Zatwierdził	Paweł Parczyk

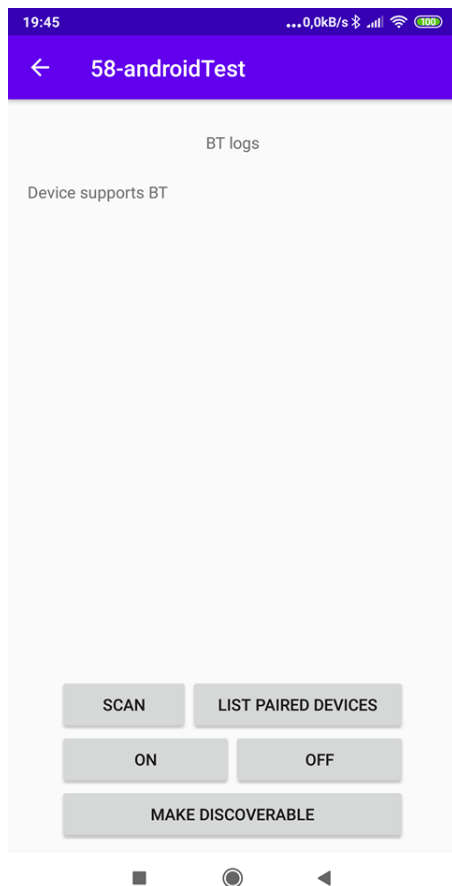
1. zainstalować android studio
 - (a) zainstalować driver do androida `sudo apt-get install adb`
 - (b) uruchomić tryb developerski w urządzeniu (7 kliknieć w numer wersji OS)
 - (c) uruchomić debugowanie przez usb (w ustawieniach developerskich)
 - (d) uruchomić instalowanie przez usb (w ustawieniach developerskich)
2. utworzyć nowy projekt Empty Activity
3. zbudować
4. jeśli wystąpił błąd związany z "platforms;android-28"
 - (a) znaleźć `./sdkmanager Android/tools/bin` i wywołać polecenie `./sdkmanager "platforms;android-28"`. Lub zainstalować przez AndroidStudio `tools -> sdkmanager`
 - (b) w przypadku pracy na windowsie należy skorzystać z konsoli zainstalowanej w android studio
5. zbudować
6. wejść w Run -> Run app. Na belce powinny się pojawić opcje: <https://developer.android.com/training/app>
7. Jeśli występują błędy przy znalezieniu urządzenia
 - (a) `adb devices -l no permissions` Można też spróbować rozłączyć usb
 - (b) `sudo adb kill-server`
 - (c) `sudo adb start-server`
 - (d) `adb devices -l unauthorized`
 - (e) na telefonie nadać uprawnienia do debugowania i przejść w tryb PTP (wyskoczą dwa dymki)
8. Gotowe!

9 Program techniczny - Android

9.1 Radio bluetooth

W związku z koniecznością oprogramowania modułu bt na urządzenia z systemem android została napisana aplikacja pozwalająca zarządzać. Z jej poziomu można włączyć, wyłączyć moduł. Ustanowić urządzenie wykrywalnym przez inne urządzenia, a także wykonać skanowanie. Wyniki pracy są prezentowane na polu tekstowym.

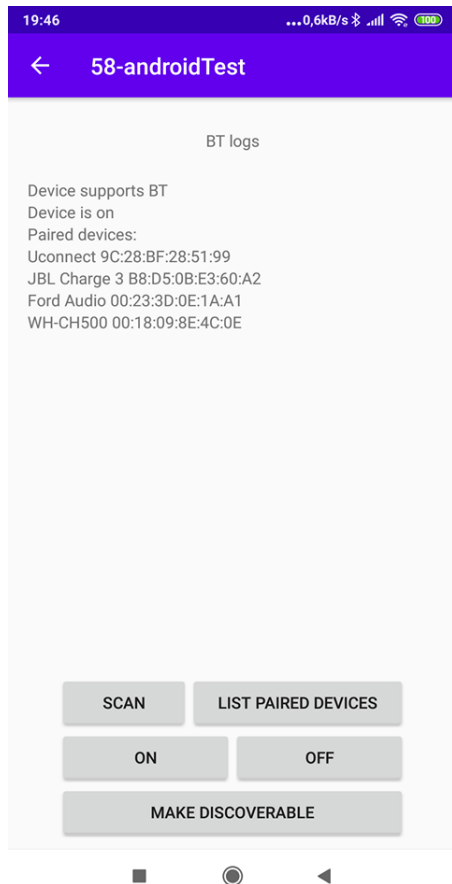
Po uruchomieniu aplikacji należy wybrać właściwy ekran przyciskiem BT. Docelowy ekran został przedstawiony na rysunku 10. Zawiera on pola niezbędne do zarządzania modułem BT w systemie android.



Rysunek 3: Ekran uruchomionej aplikacji testowej

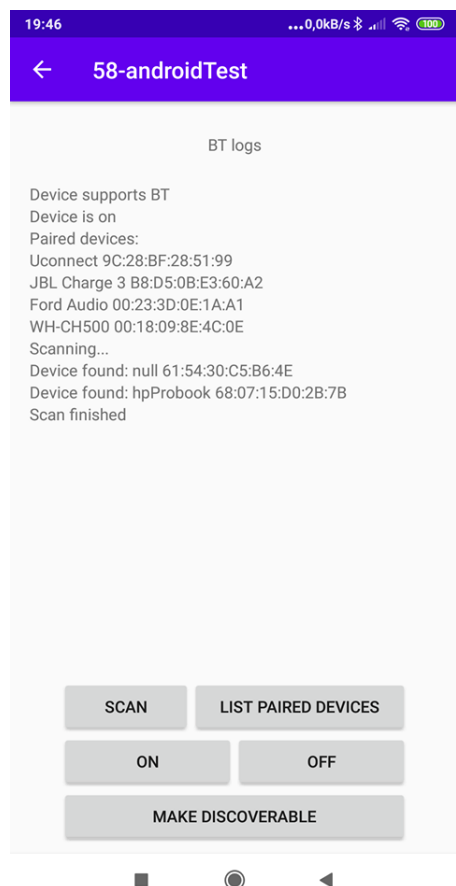
Pierwszy wyświetlony komunikat informuje czy urządzenie na którym został uruchomiony program obsługuje komunikację bluetooth. Jeśli tak (jak w przypadku urządzenia testowego) wyświetli się komunikat *Device supports BT* informujący o wsparciu dla BT.

Przycisk *List Paired Devices* wyświetli listę aktualnie sparowanych urządzeń. Efekt zadziałania został zaprezentowany na rysunku 5. Urządzenie było sparowane z dwoma zestawami głośnomówiącymi, słuchawkami i głośnikiem. Zostały wypisane nazwy urządzeń a także ich adresy MAC.



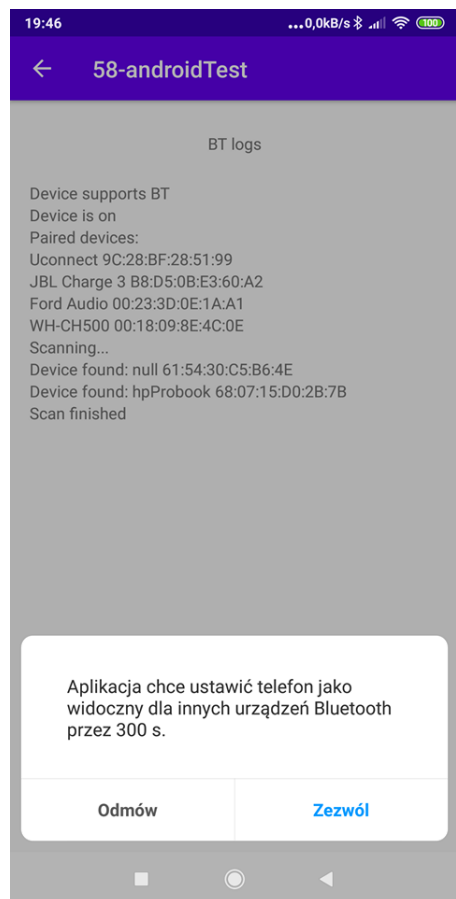
Rysunek 4: Ekran uruchomionej aplikacji testowej

Jedną z najważniejszych funkcji urządzenia jest wykrywanie pobliskich urządzeń BT, aby móc się z nimi skomunikować. Do sprawdzenia tej funkcji służy przycisk *Scan*. Po jego wciśnięciu telefon zacznie proces wyszukiwania, który domyślnie trwa 12 sekund. Po jego zakończeniu lista logów zostanie zaktualizowana. Proces skanowania został przedstawiony na rysunku 5. Podobnie jak w przypadku wypisania urządzeń sparowanych zostały wyświetlone nazwy i adresy MAC znalezionych urządzeń. Jeśli nazwa przyjęła wartość *null* znaczy to, że urządzenie się nie przedstawiło.



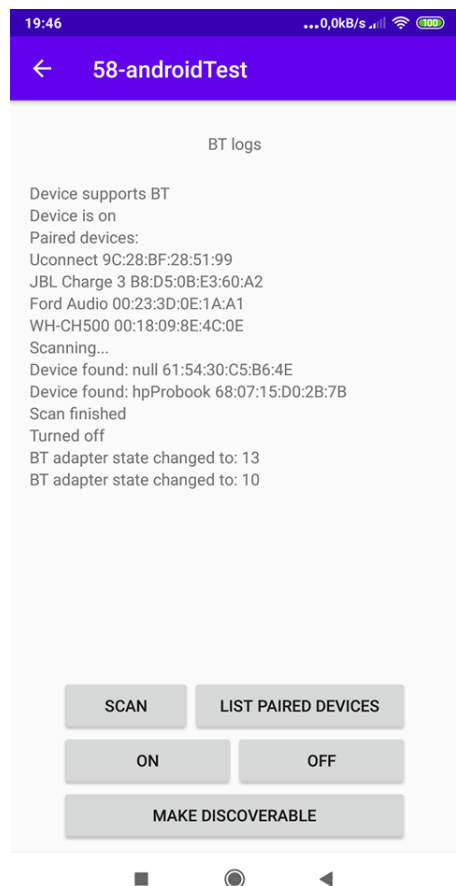
Rysunek 5: Ekran uruchomionej aplikacji testowej

W technologii BT urządzenie może nasłuchiwać na rozgłoszenia, ale nie być widoczne podczas skanowania przez inne urządzenia. Aby umożliwić innym urządzeniom wykrywanie należy włączyć tryb *discoverable*. Służy do tego przycisk *Make Discoverable*. Po jego wciśnięciu system wyświetla wiadomość z prośbą o potwierdzenie przez użytkownika (zaprezentowane na rysunku 6).

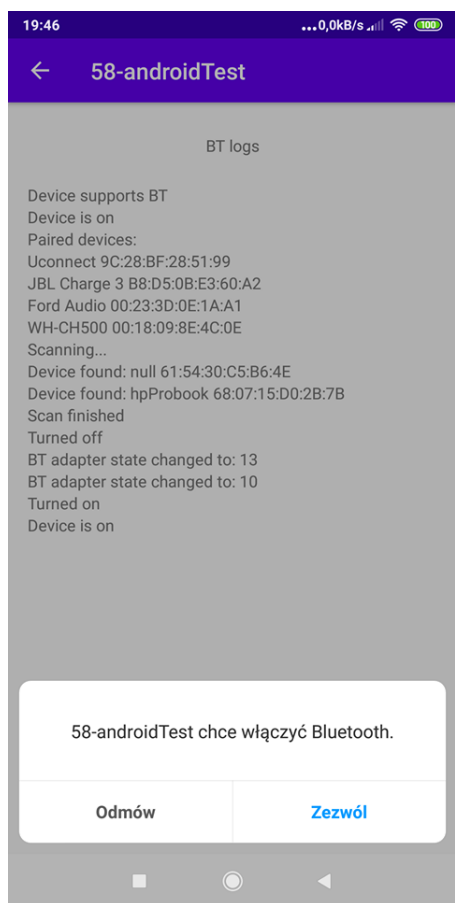


Rysunek 6: Ekran uruchomionej aplikacji testowej

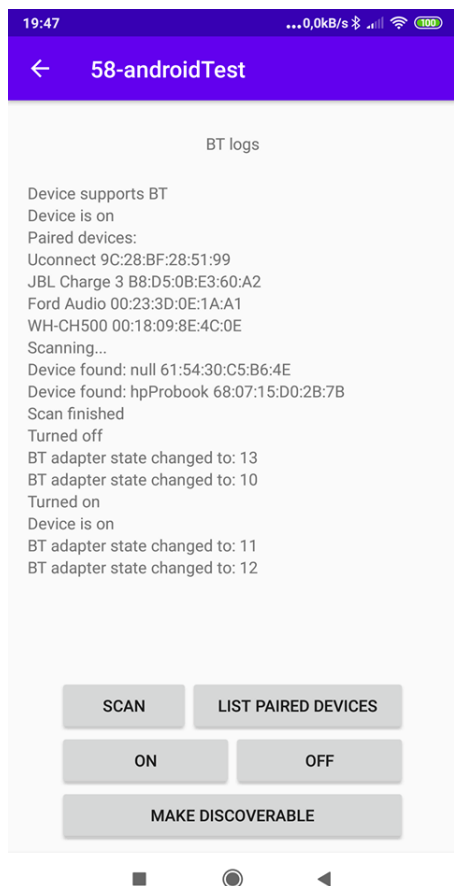
Z poziomu aplikacji możliwe jest zarówno uruchomienie jak i wyłączenie radia. Służą do tego kolejno przyciski *On* i *Off*. Na rysunku 7 został zaprezentowany rezultat wyłączenia. Podczas uruchomienia system operacyjny wyświetla wiadomość z prośbą o potwierdzenie włączenia radia (rysunek 8), po akceptacji przez użytkownika zostają wyświetlone kolejne komunikaty. Zaktualizowana lista logów została zaprezentowana na rysunku 9



Rysunek 7: Ekran uruchomionej aplikacji testowej



Rysunek 8: Ekran uruchomionej aplikacji testowej



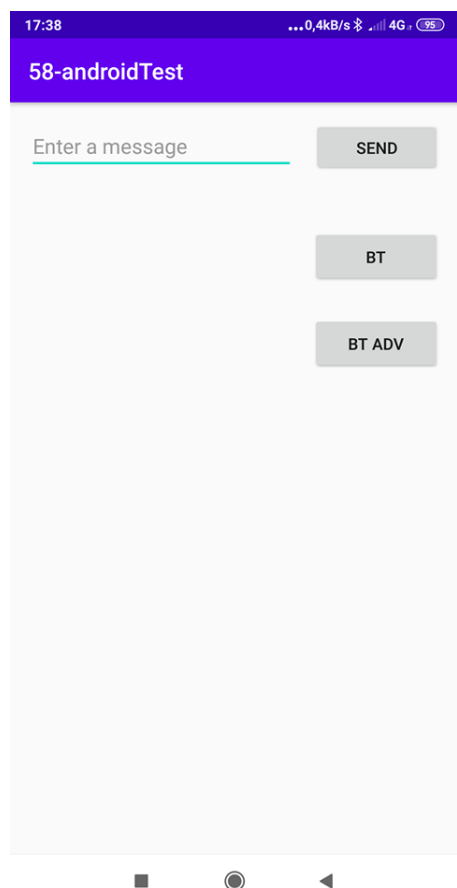
Rysunek 9: Ekran uruchomionej aplikacji testowej

9.2 Tryb advertisement

Drugim krokiem pracy nad systemem Android było stworzenie oprogramowania sprawdzającego możliwość komunikacji przez BT z wykorzystaniem trybu advertisement. Komunikacja w tym trybie opiera się na rozsyłaniu ramek bez potwierdzenia odbioru.

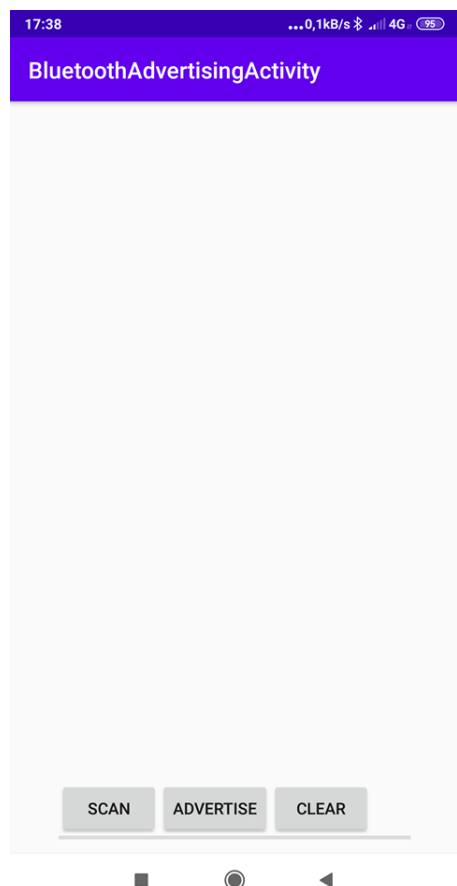
Po uruchomieniu aplikacji należy przejść do *BT adv*. Na ekranie pojawi się pole logu i 3 przyciski. Po kliknięciu *advertise*, urządzenie przejdzie w tryb rozgłaszania i rozpocznie wysyłanie ramek. Przycisk *scan* zmienia tryb pracy na odbiór. Po uruchomieniu odbierania urządzenie przez 10 sekund będzie oczekiwać na dane przychodzące - aplikacja filtruje cały ruch który nie jest wysyłany przez nią samą. Więc poprawność działania można sprawdzić jedynie uruchamiając aplikację na dwóch różnych urządzeniach i przechodząc w jednej do trybu rozgłaszania a w drugiej do trybu skanowania.

Na rysunku 10 przedstawiono obraz głównego okna aplikacji, na który zaznaczono przycisk przenoszący do opisywanej części aplikacji.

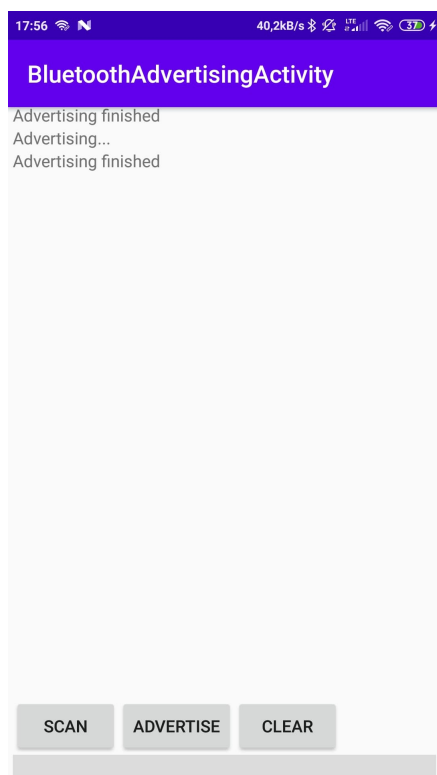


Rysunek 10: Ekran główny aplikacji testowej

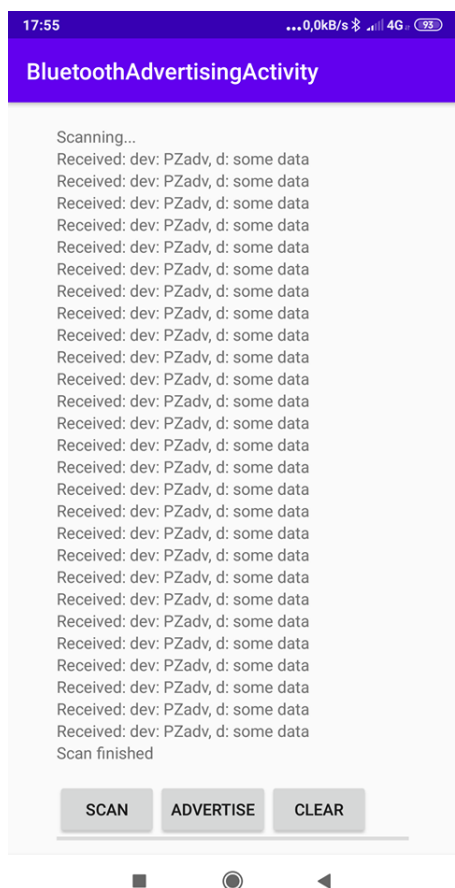
Na rysunku 11 przedstawiono ekran trybu advertisement aplikacji. Test połączenia przebiegał w następujący sposób. Na jednym z urządzeń uruchomiono skanowanie. Po otrzymaniu pozytywnego komunikatu uruchomiono wysyłanie na drugim urządzeniu, które przez okres 3 sekund nadawało. Na rysunkach 12 i 13 przedstawiono logi po zakończeniu procesu.



Rysunek 11: Ekran trybu advertisement aplikacji testowej



Rysunek 12: Log po przeprowadzeniu procesu wysyłania



Rysunek 13: Log po przeprowadzeniu procesu skanowania

10 Instrukcja instalacji Android IDE oraz biblioteki i2C

Zakres prac	Instrukcja instalacji Android IDE oraz biblioteki i2C, Program techniczny - Arduino
Wykonał	Grzegorz Huchla
Sprawdził	Adam Pędracki
Zatwierdził	Paweł Parczyk

1. Pobrać instalator Arduino IDE ze strony <https://www.arduino.cc/en/Main/Software> oraz go zainstalować,
2. Pobrać bibliotekę I2C ze strony:
https://downloads.arduino.cc/libraries/github.com/marcoschwartz/LiquidCrystal_I2C-1.1.2.zip
3. Rozpakować archiwum biblioteki
4. Odszukać folder Users/*[nazwa_użytkownika]*/Documents/Arduino/libraries
5. Wkleić folder LiquidCrystal_I2C-1.1.2

11 Program techniczny - Arduino

W celu funkcjonalnej pracy każdego z beaconów konieczne było zaprogramowanie modułów Arduino w taki sposób, aby wyświetlały na module LCD dane potrzebne ratownikowi do klasyfikacji poszkodowanego. Dodatkowo dane te muszą być zdalnie modyfikowalne, docelowo przez aplikację KAM-a.

Wyświetlacz pokazuje następujące parametry:

- Puls,
- oddech,
- ciśnienie skurczowe i rozkurczowe,
- temperaturę ciała,
- saturację krwi.

Docelowo zostanie dodana również graficzna reprezentacja pracy zastawek serca.

Wszystkie parametry, oprócz oddechu, są pokazane w formie liczbowej. Oddech natomiast jest reprezentowany przez zmieniającą się strzałkę. Skierowana w górę symuluje wdech, a w dół oznacza wydech. Dodatkowo na wyświetlaczu przed tętnem znajduje się serduszko migające z prędkością odpowiednią dla obecnie wyświetlanego tętna.



Rysunek 14: Parametry życiowe na wyświetlaczu LCD

11.1 Modyfikacja parametrów

Modyfikacja parametrów wyświetlanych przez moduł odbywa się obecnie, ze względu na brak wszystkich koniecznych urządzeń i aplikacji, za pomocą portu szeregowego działającego poprzez kabel USB. Modyfikacja parametrów odbywa się za pomocą komend przesyłanych do Arduino, według następującego wzoru:

Parametr	Kod	Jednostka	Wartość
Puls	A	1bpm	000-999
Wdech	B	10ms	000-999
Wydech	C	10ms	000-999
RR skurcz	D	1mmHg	000-999
RR rozkurcz	E	1mmHg	000-999
Temperatura	F	0,1 st.C	000-999
Saturacja	G	1%	000-100

Komenda składa się z kodu i bezpośrednio po nim trzycyfrowej wartości z podanego zakresu. W przypadku saturacji wartości wyższych niż 100 wykonywana jest na parametrze operacja modulo 101. Przykładowo, chcąc ustawić ciśnienie rozkurczowe na 80mmHg należy użyć komendy E080. Możliwym jest łączenie komend, oddzielając je pojedynczym znakiem !. Koniec ciągu komend, niezależnie od ilości modyfikowanych parametrów należy oznaczyć znakiem #. Przykładowa komenda modyfikująca kilka parametrów na raz może mieć więc postać: A111!D021!E037!G069#

Aby otworzyć okno komunikacji szeregowej w komputerze należy w otwartym środowisku Android IDE użyć kombinacji klawiszy Ctrl-Alt-M

12 Źródła

1. <https://www.gov.pl/web/zdrowie/zdarzenia-mnogie/masowe>
2. https://en.wikipedia.org/wiki/Simple_triage_and_rapid_treatment
3. <https://www.bluetooth.com/specifications/archived-specifications/>
4. <https://en.wikipedia.org/wiki/Bluetooth>
5. <https://www.bluetooth.com/blog/periodic-advertising-sync-transfer/>

13 Materiały

<https://github.com/NordicSemiconductor/Android-nRF-Mesh-Library>

Zakres prac	asdasd
Wykonał	ZZZZ
Sprawdził	XXXX
Zatwierdził	YYYY