

POLITECHNIKA WROCŁAWSKA

PROJEKT ZESPOŁOWY

SYSTEM OBSŁUGI ZAJĘĆ GRUPOWYCH

System do treningu TRIAGE dla ratowników medycznych

Autorzy:

Adam PĘDRACKI 241693

Marta DWORAKOWSKA 235645

Grzegorz HUCHLA 241265

Paweł PARCZYK 241390

Krzysztof JAROS 226745

Prowadzący:

Dr inż. Jan NIKODEM

27 maja 2020

Spis treści

1	Opis zadania	3
2	Wymagania	7
3	Założenia	7
4	Opis systemu	8
4.1	Symulatory poszkodowanych	8
4.2	Triażyści	8
4.3	KAM	8
4.4	Instruktor	8
4.5	Zestawienie i charakterystyka sprzętu	9
5	Sieć	9
5.1	Wybór technologii	9
5.1.1	Technologia Bluetooth Low Energy	9
5.1.2	BLE i Android	9
5.1.3	BLE 4.2 vs BLE 5.0	9
5.2	BLE 4.2	10
5.2.1	Zasięg	10
5.2.2	Przebieg transmisji	10
5.2.3	Architektura aplikacji Bluetooth (profile Bluetooth - GAP i GATT)	12
5.3	Sieć w systemie	13
5.3.1	Topologia sieci	13
5.3.2	Protokół komunikacji	14
5.3.3	Ramki protokołu komunikacji	15
6	Wybór sprzętu	16
6.1	Kryteria i wybór	16
6.2	Arduino	16
6.2.1	NANO	16
6.2.2	UNO	17
6.2.3	Inne modele arduino	17
6.3	Wyświetlacz	17
6.3.1	PCF	17
6.3.2	HD	17
6.4	Bluetooth	17
6.5	Zasilanie	17
6.5.1	Przez usb	18
6.5.2	przez regulator napięcia w płytce	18
6.6	Schemat ideowy	19
6.7	Schemat elektryczny	19

7 Instrukcja instalacji android studio	20
8 Android	21
8.1 Oprogramowanie triażysty	21
8.1.1 Algorytm	21
8.2 Oprogramowanie kama	22
8.2.1 Algorytm	22
8.3 Oprogramowanie technika (administrator)	22
8.3.1 Przeznaczenie - rola w systemie	22
8.3.2 Algorytm	23
8.3.3 Opis oprogramowania	23
8.4 Programy techniczne	25
8.4.1 Radio bluetooth	25
8.4.2 Tryb advertisement	30
9 Instrukcja instalacji Arduino IDE oraz biblioteki i2C oraz ESP32	33
10 Realizacja programu Arduino i ESP32	34
10.1 Schematy blokowe programów	38
10.1.1 Arduino	38
10.1.2 ESP32	39
10.2 Modyfikacja parametrów	39
10.3 ESP i komunikacja Bluetooth	40
10.4 Uruchomienie	40
11 Źródła	40
12 Materiały	41

1 Opis zadania

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 liczby osób ratownicy stosują się do procedur.

W swoich działaniach ratownicy stosują się do zasad segregacji poszkodowanych w wypadku masowym, umożliwiających 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
- czarnego - osoby nie do odratowania.

Zalecane jest aby na czynności z tym związane ratownik poświęcił maksymalnie 30 sekund.

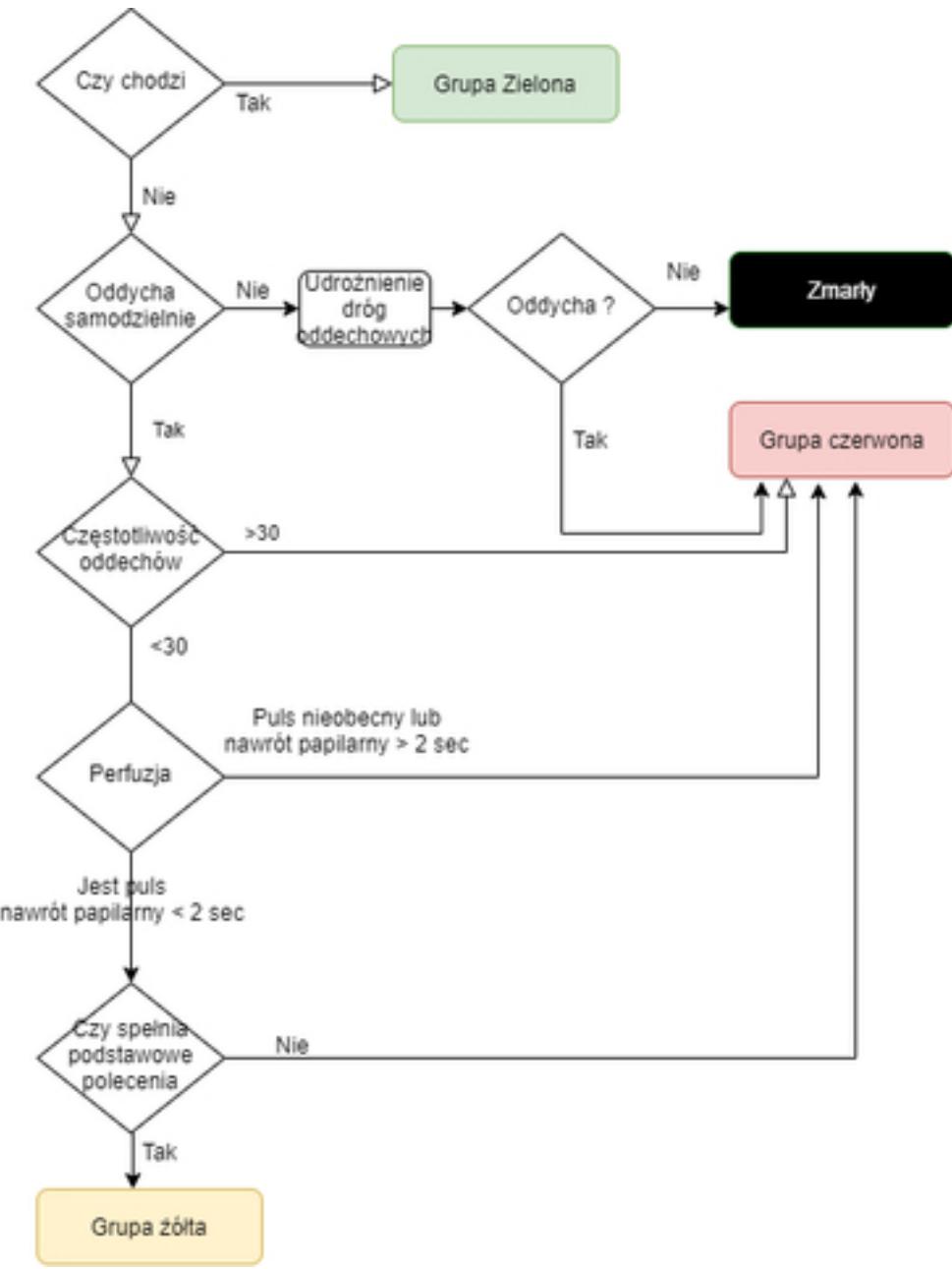
Trening odbywa się w ograniczonej przestrzeni np. budynku, w hali sportowej o wymiarach 40m na 50m. W szkoleniu uczestniczą triażysti oraz kierujący akcją medyczną (KAM), wszystko nadzoruje instruktor. Do szkolenia przeznaczonych jest kilka symulatorów poszkodowanych, ułożonych w różnych punktach hali.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

W przypadku treningu z naszym systemem, do wyboru mamy 3 rodzaje segregacji triage:

1. START Triage

Zakłada nadanie każdemu z poszkodowanych określonego priorytetu określającego kolejność udzielania pomocy/ewakuacji. Poszkodowani zostają oznaczeni poprzez nadanie odpowiedniego koloru (czerwony, żółty, zielony, czarny). Zakłada sprawdzenie takich funkcji życiowych jak:

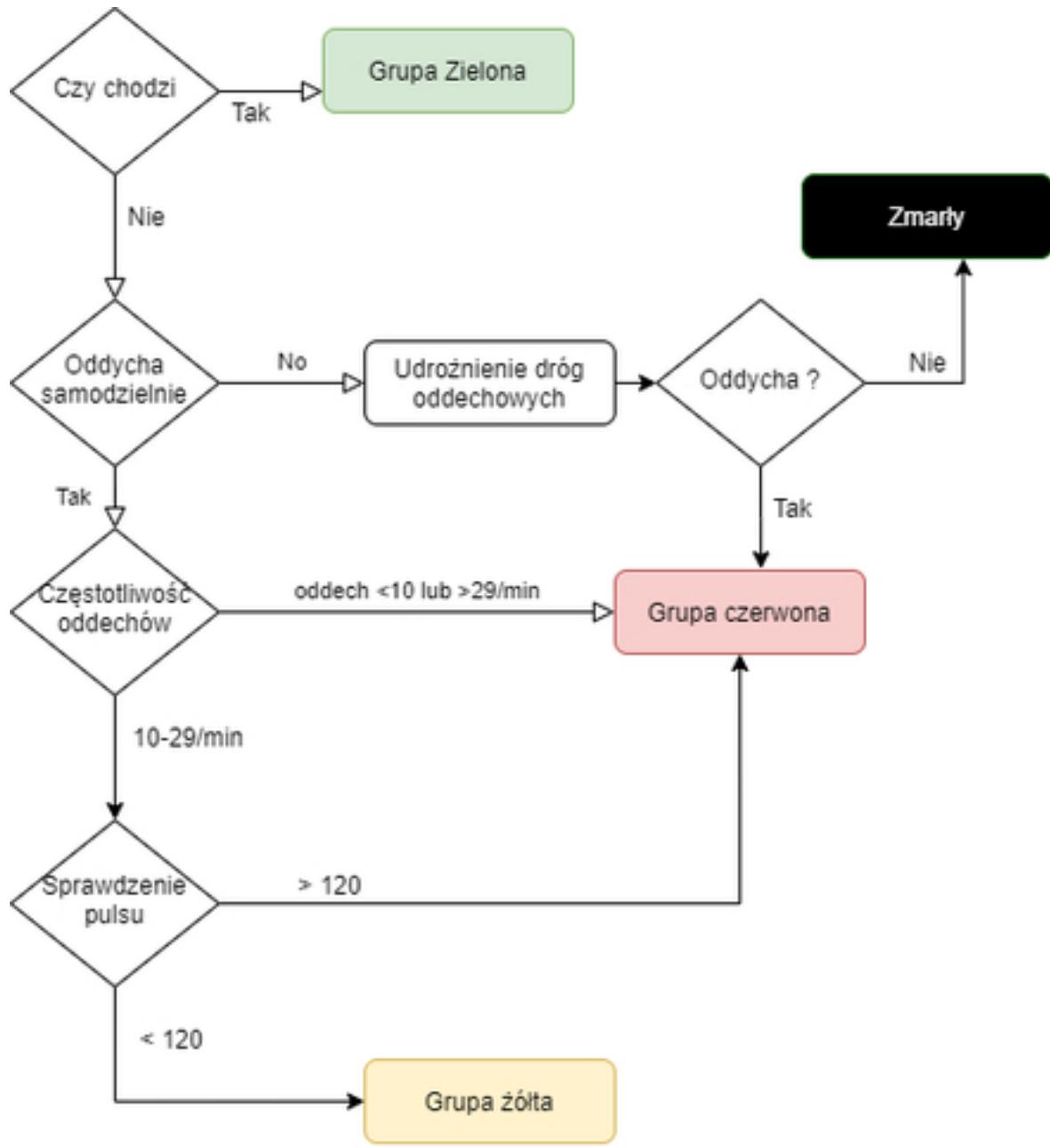
- możliwość chodzenia
- drożności dróg oddechowych
- częstości oddechu
- czas nawrotu kapilarnego lub obecność tężna na tętnicy promieniowej stan świadomości



2. Sieve Triage

stosowany w Wielkiej Brytanii i Australii. Opiera się na podobnych założeniach jak START. Ocena pacjentów opiera się na:

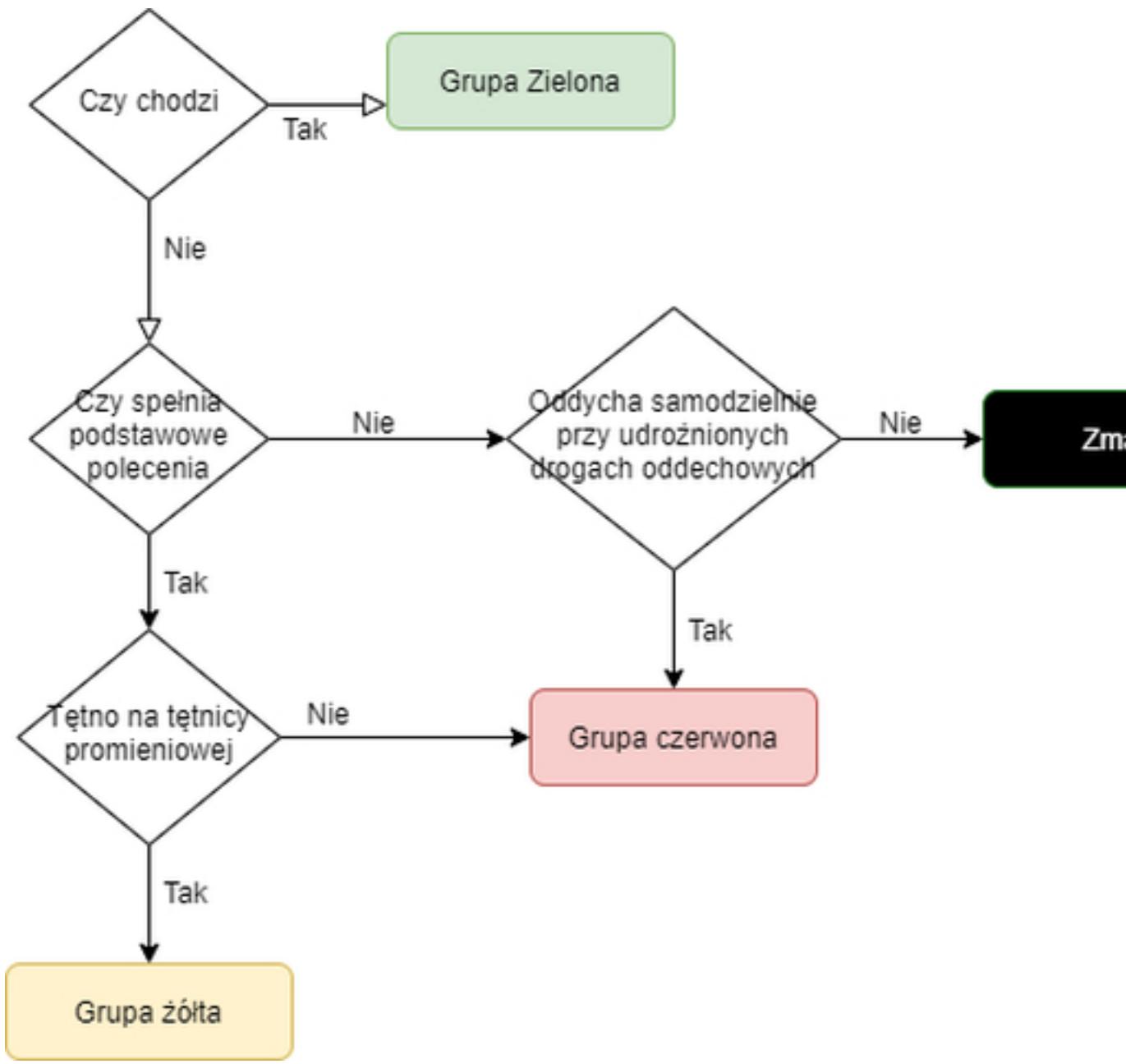
- możliwość chodzenia
- oddechu
- perfuzji obwodowej
- częstości pracy serca



3. CareFlight Triage

Opracowany w Australii. Podobny do systemu START, oparty na ocenie funkcji życiowych:

- możliwości samodzielnego poruszania się
- tępna na tętnicy promieniowej
- oddechu przy udrożnionych drogach oddechowych



Wszystkie parametry życiowe wyświetlane będą na symulatorze poszkodowanego.

Instruktor za pomocą aplikacji magląd w poczynania uczestników. Widzi jaka przypisana została kategoria, kto ją przypisał, 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.

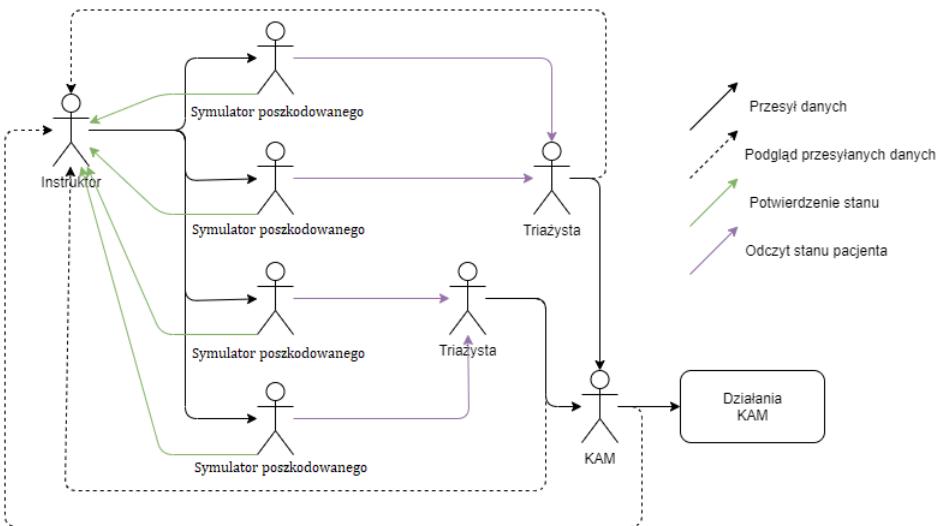
2 Wymagania

1. Linia życia musi być w czytelny sposób przekazana triażyście.
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 liczbie co najmniej jednej załogi karetki.
7. Symulatory poszkodowanego powinny mieć wystarczająco energii do nieprzerwanego działania przez jedną sesję szkoleniową.

3 Założenia

1. Aplikacja symulatora poszkodowanego działa na urządzeniu Arduino UNO wraz z podłączonym wyświetlaczem oraz modułem ESP-32
2. Linia życia jest zbiorem parametrów życiowych określanych znacznikiem czasowym
3. Symulator poszkodowanego prezentuje dane triażyście więc będzie zrealizowany przez mikrokomputer z wyświetlaczem, na którym będą wyświetlane parametry życiowe poszkodowanego.
4. Komunikacja symulatorów poszkodowanych będzie zrealizowana z wykorzystaniem technologii bluetooth w wersji LE.
5. Wszyscy triażyści muszą mieć możliwość przekazywania formularzy KAMowi oraz instruktorowi z dodatkowymi informacjami o czasie ich wypełnienia, więc formularze zrealizowane będą przez dedykowaną aplikację na telefony komórkowe, będące na standardowym wyposażeniu ratownika medycznego.
6. KAM musi mieć możliwość odebrania formularzy, więc posłuży mu do tego aplikacja dedykowana na tablet będący standardowym wyposażeniem karetki.
7. Instruktor nadzorujący szkolenie ma do dyspozycji tablet z dedykowaną aplikacją, służącą jako panel sterowania.

4 Opis systemu



4.1 Symulatory poszkodowanych

Symulatory poszkodowanych są mikrokomputerami komunikującymi się bezprzewodowo. Każdy symulator poszkodowanego 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 Triażysti

Triażysta uczestniczący w szkoleniu będzie wyposażony w telefon, na którym działa aplikacja pełniąca rolę terminala. Gdy ratownik podejdzie do symulatora poszkodowanego, 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, zainstalowaną aplikacją umożliwiającą mu monitorowanie przebiegu szkolenia oraz sprawdzenie stanów na symulatorach poszkodowanych.

4.5 Zestawienie i charakterystyka sprzętu

Jako symulatory poszkodowanych 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 symulatora 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 płytę.

Ratownik będzie posługiwał się dowolnym smartfonem 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 Sieć

5.1 Wybór technologii

5.1.1 Technologia Bluetooth Low Energy

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

5.1.2 BLE i Android

Android od wersji 4.3 (Jelly Bean) wspiera technologię Bluetooth Low Energy, chociaż dostępna wersja BT zależy od urządzenia.

- Android 4.3 (Jelly Bean) - BLE 4.0
- Android 5.0 (Lollipop) - BLE 4.1
- Android 7.0 (Nougat) - BLE 4.2
- Android 8.0 (Oreo) - BLE 5.0

5.1.3 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 korekcyjnych danych, a z kodowaniem 500kb/s lub 125kb/s. Dla technologii LE 5.0 przepływność symbolowa dla niekodowanych danych może zostać zwiększena 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 istotnych ulepszeń w systemie, **rozwiązańem pozostaje technologia BLE 4.2** z uwagi na większą dostępność i niższą cenę urządzeń na rynku.

5.2 BLE 4.2

5.2.1 Zasięg

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).

5.2.2 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. Wysłanie 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 advertisera o nawiązanie połączenia. Połączenie zostaje zaakceptowane, jeżeli advertiser zaakceptuje żądanie, wtedy initiator 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 algorytmiczne 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.

5.2.3 Architektura aplikacji Bluetooth (profile Bluetooth - GAP i GATT)

Transport danych w systemie BLE podzielony jest na warstwy - fizyczną, logiczną i L2CAP. Na tych warstwach zbudowane są różnorodne profile Bluetooth. Podstawowym, implementowanym zawsze, jest profil GAP. W LE, GAP definiuje 4 role, które może pełnić urządzenie:

- Broadcaster (tylko transmituje dane)
- Obserwator (tylko odbiera dane)
- Rola peryferyjna (pojedyncze połączenie - slave)
- Rola centralna (wiele połączeń - master)

Jedno urządzenie może pełnić różne role, a nawet ich kombinacje.

Wokół profilu GAP obudowanych jest wiele innych profili, np.: GATT. Profil ten zbudowany jest dla protokołu ATT. GATT definiuje 2 role:

- Serwer (udostępnia dane i przetwarza żądania)
- Klient (pobiera dane i wysyła żądania)

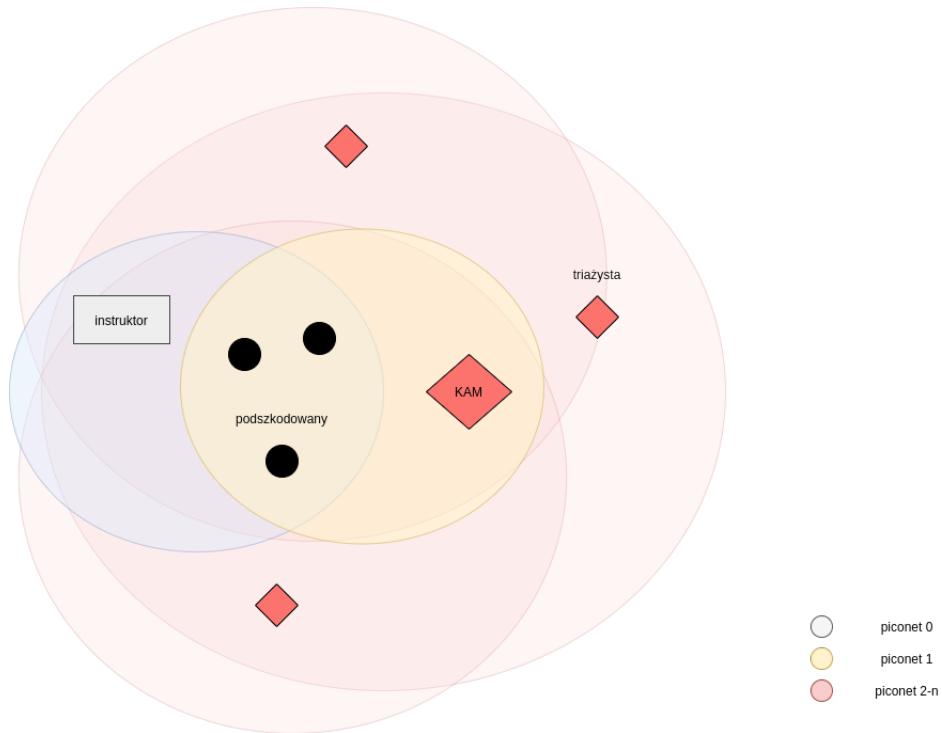
Te role nie są przywiązane do ról GAP, chociaż serwer pełni zazwyczaj rolę peryferyjną, a klient centralną.

Zakres prac	Sieć w systemie
Wykonał	Marta Dworakowska
Sprawdził	Grzegorz Huchla
Zatwierdził	Paweł Parczyk

5.3 Sieć w systemie

5.3.1 Topologia sieci

Rysunek 1 przedstawia sieć scatternet użytą w projekcie.



Rysunek 1: Topologia sieci

- W sieci piconet 0 instruktor pełni funkcję klienta, a poszkodowani serwerów.
- W sieci piconet 1 KAM pełni funkcję klienta, a symulatory poszkodowanych funkcję serwerów.
- W sieciach piconet 2-n triażysta pełni rolę broadcastera, a KAM, symulatory poszkodowanych i instruktor obserwatorów.

5.3.2 Protokół komunikacji

- Instruktor, KAM i triażysta - użytkownicy Android
- Symulator poszkodowanego - urządzenie Arduino

1. Etap identyfikacji urządzeń:

Identyfikacja poszkodowanych:

- Każdy symulator poszkodowanego rozsyła ramkę ping (w 16-bajtowym polu uuid 00000000-0000-1000-8000-00805F9B34FB), informując urządzenia Android o swojej obecności.
- Użytkownicy Android odbierają ramkę ping każdego poszkodowanego, uzupełniając swoją tablicę dostępnych poszkodowanych.

Identyfikacja triażystów:

- Każdy triażysta rozsyła ramkę ping (w 16-bajtowym polu uuid 00000000-0000-1000-8000-00805F9B34FB), informując urządzenia Android i Arduino o swojej obecności.
- Użytkownicy Android odbierają ramkę ping każdego triażysty, uzupełniając swoją tablicę dostępnych triażystów.
- Urządzenia Arduino odbierają ramkę ping każdego triażysty, uzupełniając swoją tablicę dostępnych triażystów.

W konsekwencji:

- użytkownik Android może wysyłać żądania do symulatorów poszkodowanych,
- symulator poszkodowanych może nasłuchiwać triażystów,
- KAM i instruktor mogą nasłuchiwać triażystów.

2. Etap wymiany danych:

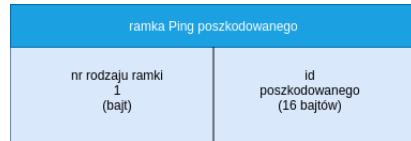
Ustalenie/pobranie charakterystyki symulatora poszkodowanego:

- Poszkodowany nasłuchuje żądań ze swoim identyfikatorem uuid.
- Użytkownik Android wysyła żądanego do symulatora poszkodowanego o podanie charakterystyki.
- Poszkodowany wysyła odpowiedź z danymi: charakterystyką.
- Użytkownik Android odbiera ramkę z odpowiedzią od symulatora poszkodowanego.

Odebranie informacji o decyzji triażysty:

- Użytkownicy Android i symulatory poszkodowanych nasłuchują ramek rozgłoszeniowych, oczekując decyzji triażysty (o klasyfikacji - kolorze).
- Triażysta podejmuje decyzję i nadaje swoją odpowiedź.
- Użytkownicy Android odbierają odpowiedź triażysty.
- Symulator poszkodowanego odbiera odpowiedź triażysty.

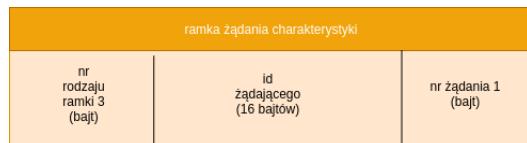
5.3.3 Ramki protokołu komunikacji



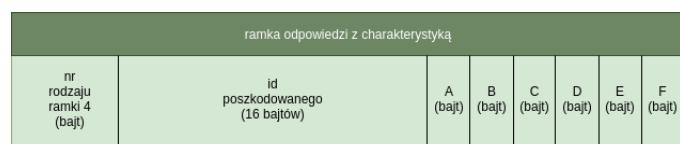
Rysunek 2: Ramka ping symulatora poszkodowanego



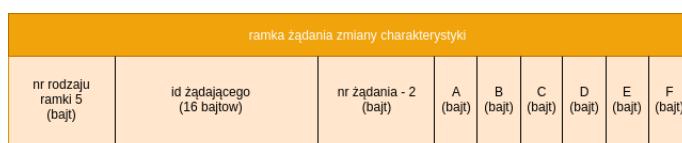
Rysunek 3: Ramka ping triażysty



Rysunek 4: Ramka żądania charakterystyki



Rysunek 5: Ramka odpowiedzi z charakterystyką



Rysunek 6: Ramka żądania zmiany charakterystyki

ramka rozgłoszeniowa triażysty			
nr rodzaju ramki 6 (bajt)	id triażysty (16 bajtów)	id poszkodowanego (16 bajtów)	kolor (bajt)

Rysunek 7: Ramka rozgłoszeniowa decyzji triażysty

6 Wybór sprzętu

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

6.1 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.

6.2 Arduino

6.2.1 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

6.2.2 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

6.2.3 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.

6.3 Wyświetlacz

Są 2 główne standardy podłączenia: PCF8574 i HD 44780.

6.3.1 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.

6.3.2 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.

6.4 Bluetooth

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

6.5 Zasilanie

Znalazłem 2 opcje:

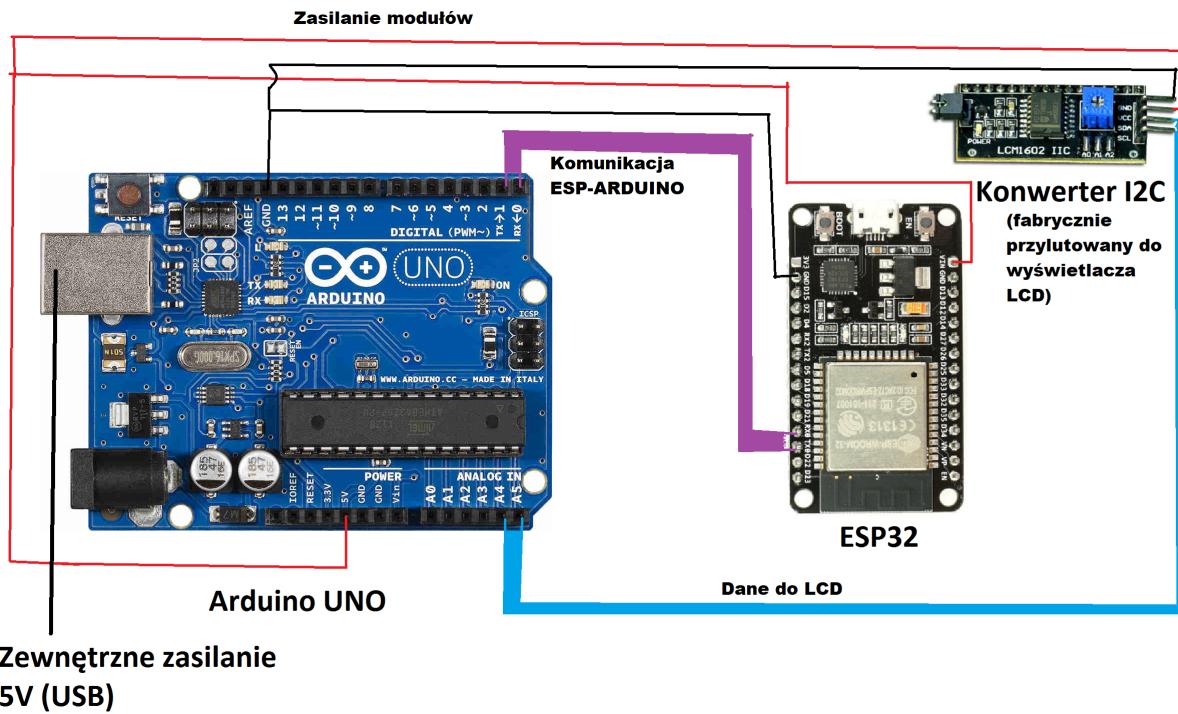
6.5.1 Przez usb

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

6.5.2 przez regulator napięcia w płytce

Płyta 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.

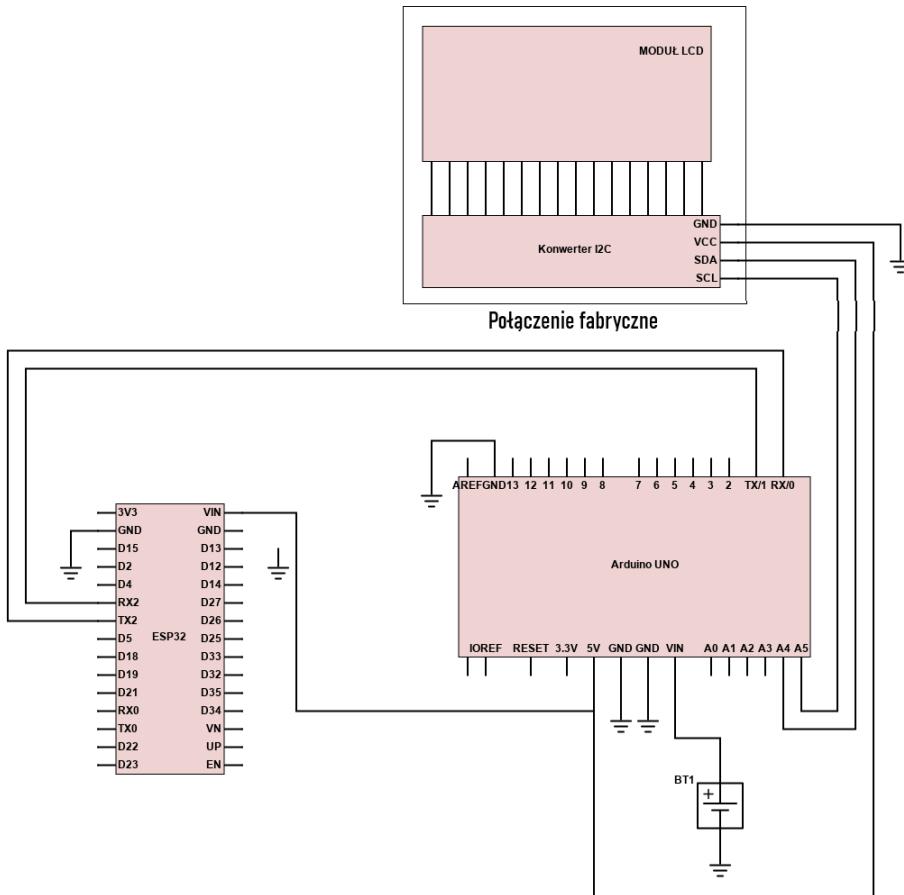
6.6 Schemat ideowy



Rysunek 8: 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. Dokładne połączenia są wyszczególnione na schemacie elektrycznym.

6.7 Schemat elektryczny



Rysunek 9: Schemat elektryczny - symulator poszkodowanego

7 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 klikniesie 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 związanego z "platforms;android-28"
 - (a) znalezcz ./sdkmanager Android/tools/bin i wywołać polecenie ./sdkmanager "platforms;android-28". Lub zainstalować przez AndroidStudio tools -i sdkmanager
 - (b) w przypadku pracy na windowsie należy skorzystać z konsoli zainstalowanej w android studio
5. zbudować
6. wejsc w Run -i Run app. Na belce powinny się pojawić opcje: <https://developer.android.com/training/> app
7. Jesli występują błędy przy znalezieniu urządzenia
 - (a) adb devices -i no permissions Mozna tez spróbować rozłączyć usb
 - (b) sudo adb kill-server
 - (c) sudo adb start-server
 - (d) adb devices -i unauthorized
 - (e) na telefonie nadac uprawnienia do debugowania i przejsc w tryb PTP (wyskoczą dwa dymki)
8. Gotowe!

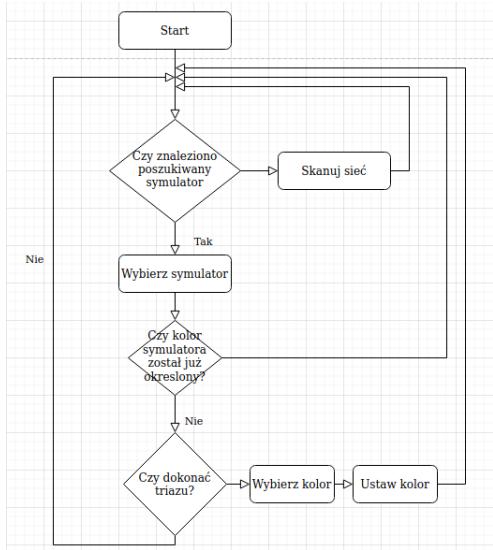
8 Android

8.1 Oprogramowanie triażysty

Zakres prac	Oprogramowanie triażysty
Wykonał	Marta Dworakowska
Sprawdził	Paweł Parczyk
Zatwierdził	Grzegorz Huchla

8.1.1 Algorytm

Zakres prac	Algorytm Triązisty
Wykonał	Paweł Parczyk
Sprawdził	Marta Dworakowska
Zatwierdził	???



Rysunek 10: Algorytm pracy oprogramowania triażysty

8.2 Oprogramowanie kama

8.2.1 Algorytm

8.3 Oprogramowanie technika (administrator)

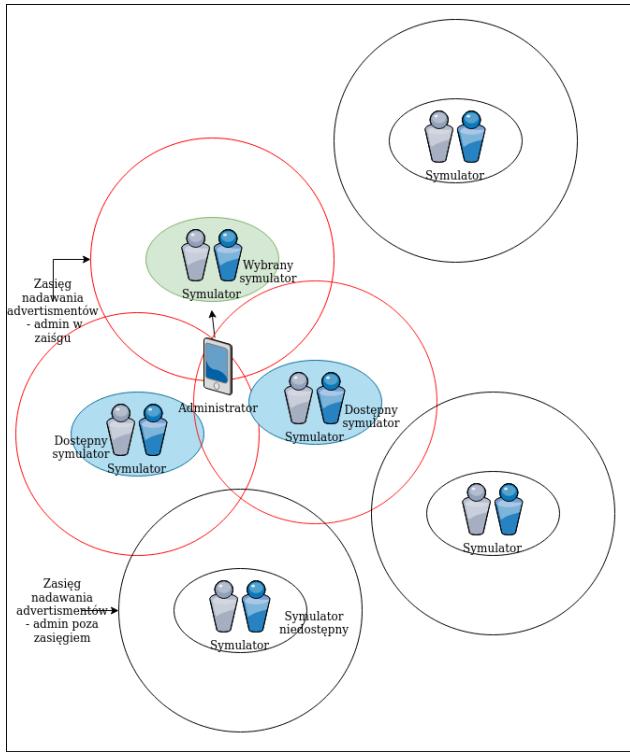
Zakres prac	Oprogramowanie technika
Wykonał	Paweł Parczyk
Sprawdził	Marta Dworakowska
Zatwierdził	Grzegorz Huchla

8.3.1 Przeznaczenie - rola w systemie

Oprogramowanie administratora zostało stworzone w celu realizacji zadań administracji systemem. Dzięki niemu administrator może przygotować system do szkolenia. Umożliwia ono ustawienie parametrów pracy symulatorów poszkodowanych lub sprawdzenie poziomu naładowania symulatorów.

Na rysunku 11 zaprezentowano układ sieci z punktu widzenia oprogramowania administratora systemu. Administrator nasłuchiwa na advertismenty (pingi) od symulatorów. Część z nich dociera ponieważ administrator znajduje się w ich zasięgu (czerwone okręgi). Symulatory od których oprogramowanie odebrało pingi zostają oznaczone jako dostępne (symulatory niebieskie). Spośród urządzeń dostępnych można wybrać jedno, którym się zarządza (symulator zielony). Pozostałe symulatory (białe z czarnymi okregami) nie są dostępne dla administratora ponieważ nie docierają do niego advertismenty.

W celu zarządzania innym symulatorem administrator musi przenieść swoje urządzenie w miejsce, do którego będą docierać advertismenty z pożądanego symulatora. Na rysunku 12 przedstawiono przemieszczenia administratora do takiej lokalizacji.



Rysunek 11: Sieć urządzeń z punktu widzenia administratora

8.3.2 Algorytm

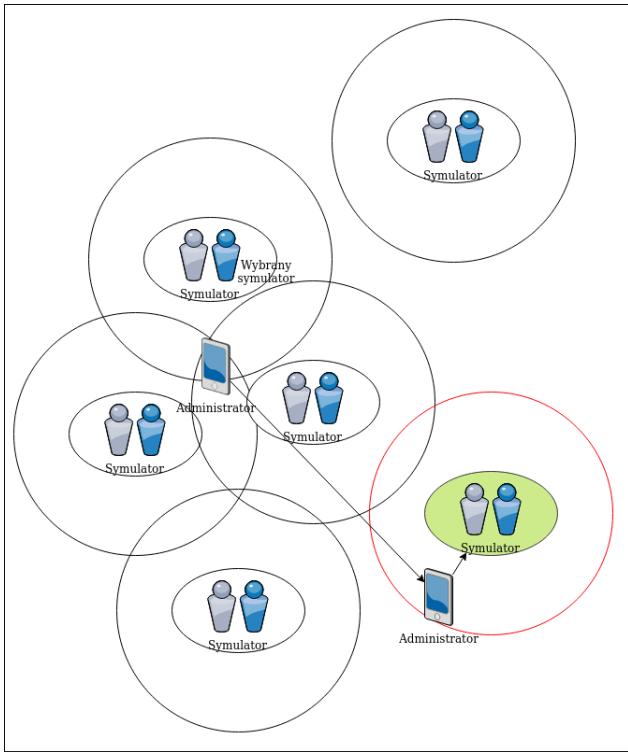
8.3.3 Opis oprogramowania

Urządzenie wyszukuje dostępne urządzenia (takie od których jest w stanie nasłuchać advertismenty), a następnie umożliwia wybranie jednego z nich i operację na nim. Umożliwia zmianę lub sprawdzenie parametrów pracy. Oprogramowanie dzieli się na kilka podstawowych modułów.

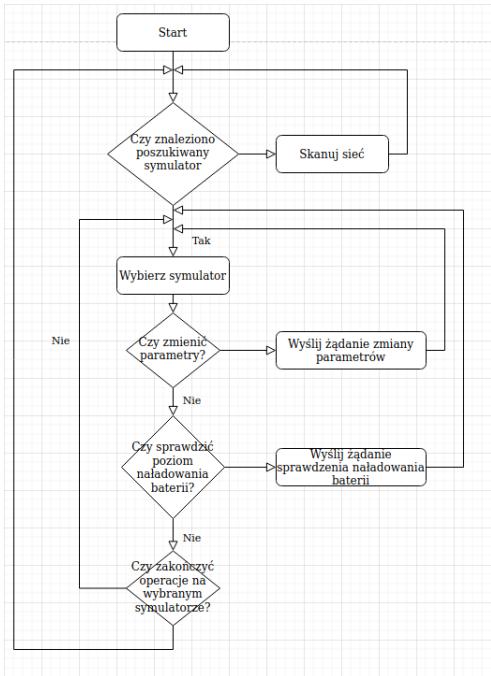
Skanowanie: Oprogramowanie nasłuchuje na ramki ping rozsypane przez symulatory poszkodowanych, gdy otrzyma komunikat określa urządzenie jako dostępne. Proces skanowania trwa nieustannie w tle przez co technik cały czas ma dostęp tylko do osiągalnych (przez sieć bluetooth) urządzeń.

Komunikacja: Oprogramowanie wysyła advertismenty zbudowane na podstawie protokołu opisanego w rozdziale Protokoły komunikacji. Z ich wykorzystaniem oprogramowanie technika może odczytywać i modyfikować parametry innych urządzeń.

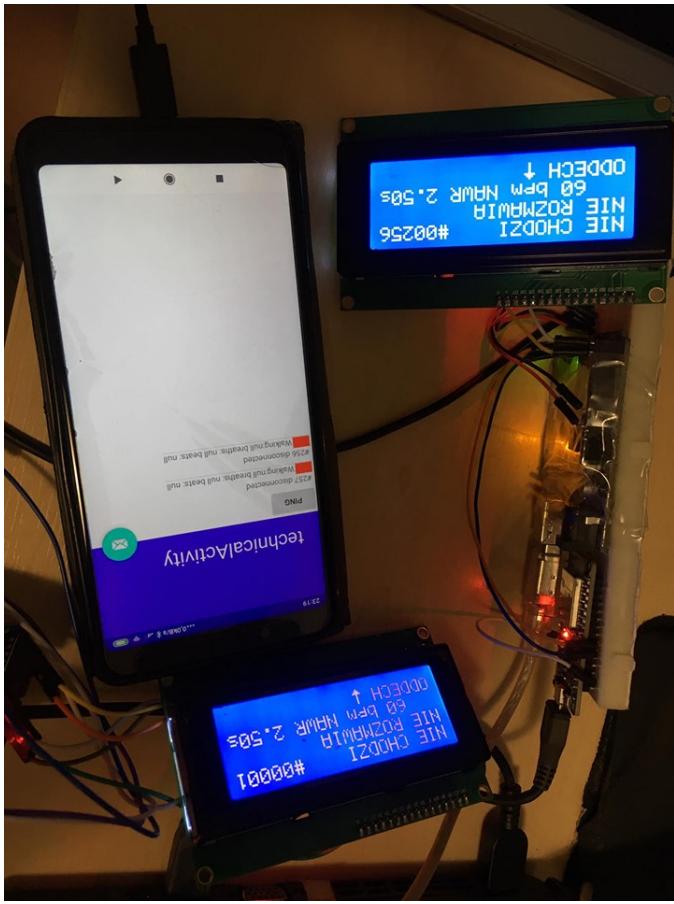
Na rysunku 14 przedstawiono możliwy wynik skanowania sieci w poszukiwaniu symulatorów w obecności dwóch symulatorów. Jeden z identyfikatorem 256 (wyświetlane jako 00256) a drugi z 257 (wyświetlane jako 00001).



Rysunek 12: Przemieszczenie administratora



Rysunek 13: Algorytm pracy oprogramowania administratora



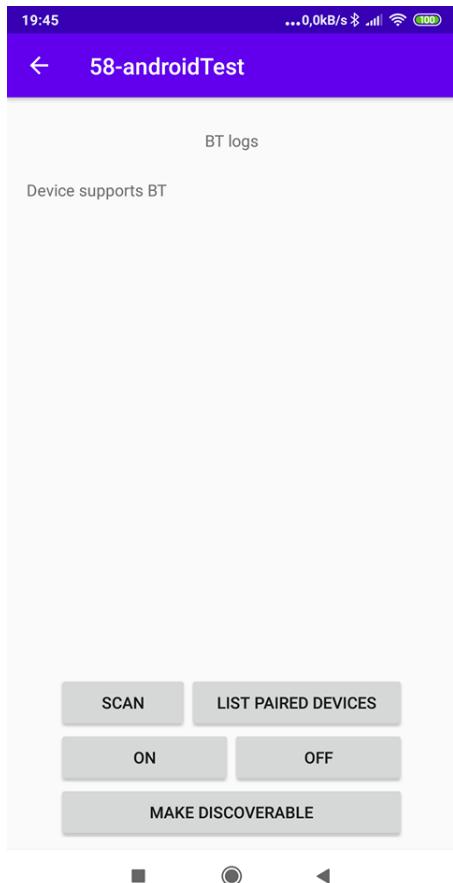
Rysunek 14: Wyniki skanowania sieci w obecności dwóch urządzeń nadających

8.4 Programy techniczne

8.4.1 Radio bluetooth

W związku z koniecznością oprogramowania modułu bt na urządzenie 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.

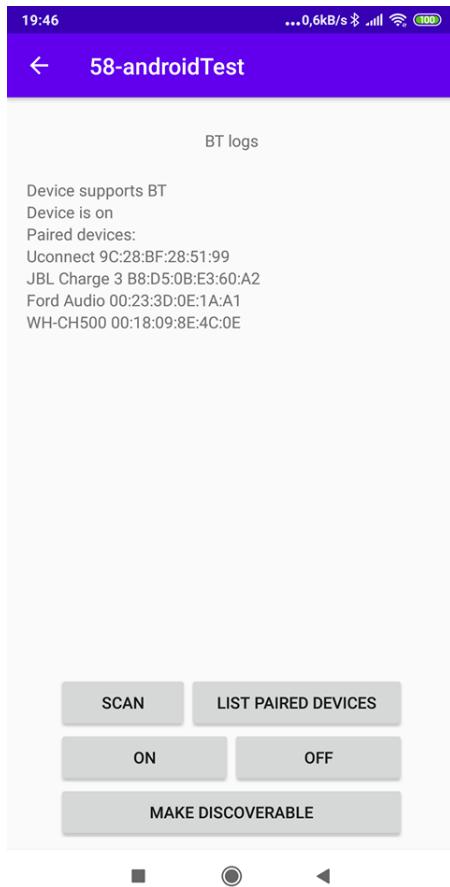
Po uruchomieniu aplikacji należy wybrać właściwy ekran przyciskiem BT. Docelowy ekran został przedstawiony na rysunku 22. Zawiera on pola niezbędne do zarządzania modułem BT w systemie android.



Rysunek 15: Ekran uruchomionej aplikacji testowej

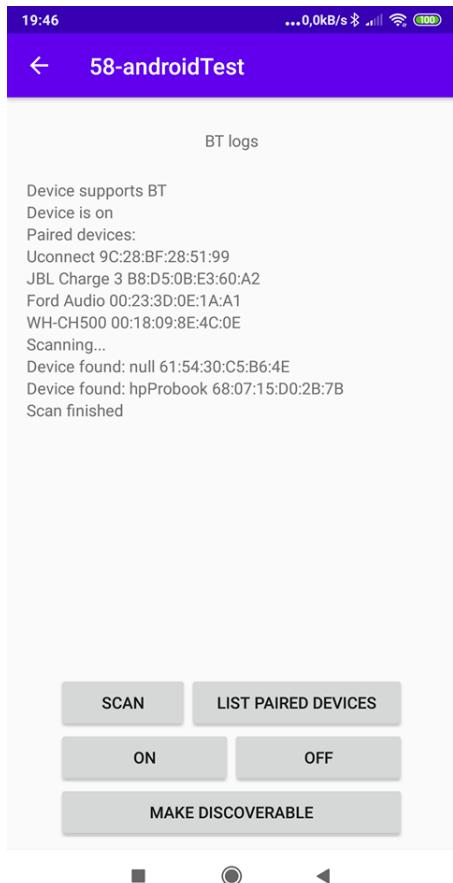
Pierwszy wyświetlony komunikat informuje czy urządzenie na którym został uruchomiony program obsługuje komunikację bluetooth. Jeśli tak (jak w przypadku urządzenia testowego) wyświetli się komunikat *Device supports BT* informujący o wsparciu dla BT.

Przycisk *List Paired Devices* wyświetli liste aktualnie sparowanych urządzeń. Efekt zadziałania został zaprezentowany na rysunku 17. Urządzenie było sparowane z dwoma zestawami głośnomówiącymi, słuchawkami i głośnikiem. Zostały wypisane nazwy urządzeń a także ich adresy MAC.



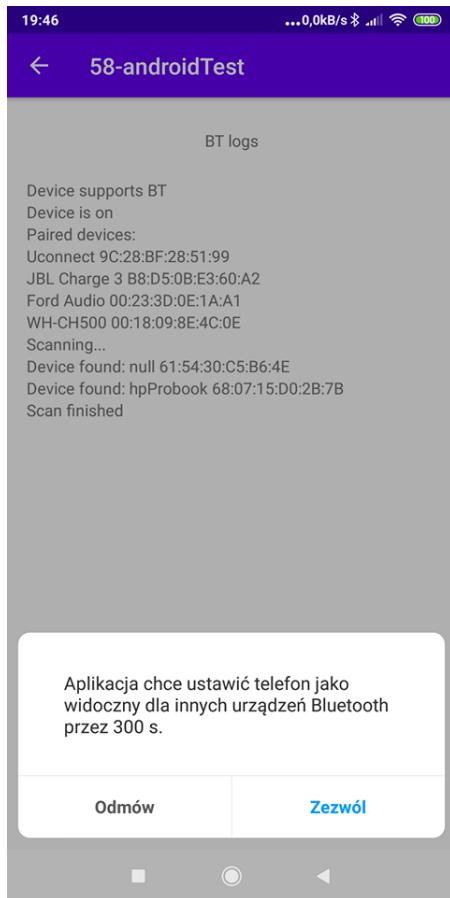
Rysunek 16: Ekran uruchomionej aplikacji testowej

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 wcisnięciu telefon zacznie proces wyszukiwania, który domyślnie trwa 12 sekund. Po jego zakończeniu lista logów zostanie zaktualizowana. Proces skanowania został przedstawiony na rysunku 17. Podobnie jak w przypadku wypisania urządzeń sparowanych zostały wyświetlane nazwy i adresy MAC znalezionych urządzeń. Jeśli nazwa przyjęła wartość *null* znaczy to, że urządzenie się nie przedstawiło.



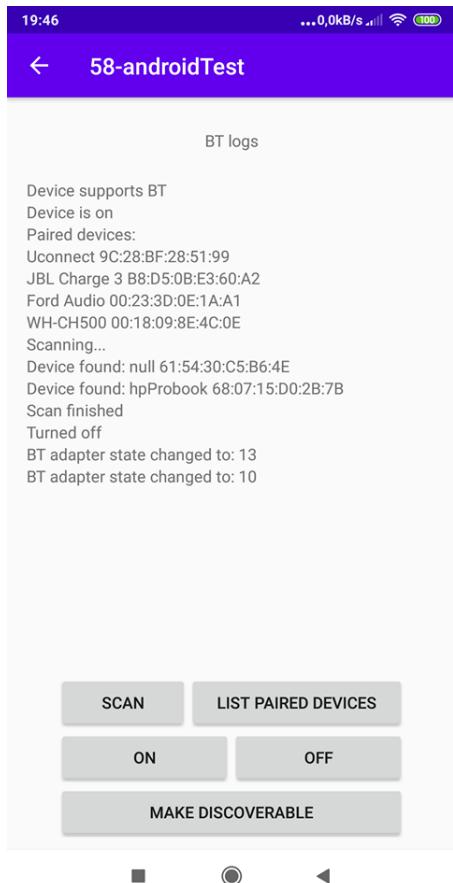
Rysunek 17: Ekran uruchomionej aplikacji testowej

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 wcisnięciu system wyświetla wiadomość z prośbą o potwierdzenie przez użytkownika (zaprezentowane na rysunku 18).



Rysunek 18: 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 19 został zaprezentowany rezultat wyłączenia. Podczas uruchomienia system operacyjny wyświetla wiadomość z prośbą o potwierdzeniełączenia radia (rysunek 20), po akceptacji przez użytkownika zostają wyświetlane kolejne komunikaty. Zaktualizowana lista logów zostały zaprezentowana na rysunku 21

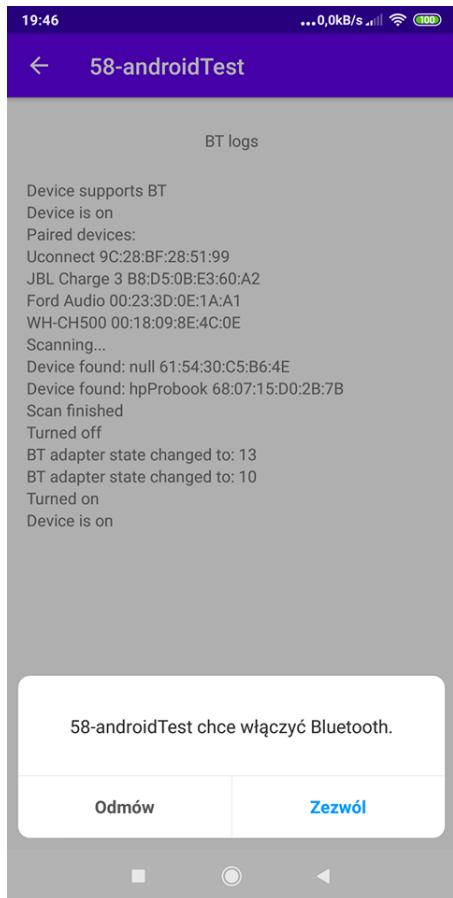


Rysunek 19: Ekran uruchomionej aplikacji testowej

8.4.2 Tryb advertisement

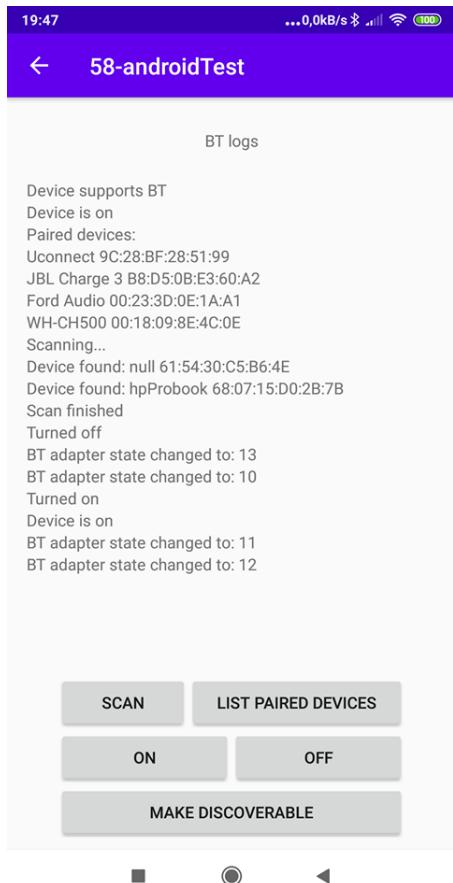
Drugim krokiem pracy nad systemem Android było stworzenie oprogramowania sprawdzającego możliwość komunikacji przez BT z wykorzystaniem trybu advertisement. Komunikacja w tym trybie opiera się na rozsyłaniu ramek bez potwierdzenia odbioru.

Po uruchomieniu aplikacji należy przejść do *BT adv*. Na ekranie pojawi się pole logu i 3 przyciski. Po kliknięciu *advertise*, urządzenie przejdzie w tryb rozgłaszenia i rozpocznie wysyłanie ramek. Przycisk *scan* zmienia tryb pracy na odbiór. Po uruchomieniu odbierania urządzenie przez 10 sekund będzie oczekiwane na dane przychodzące - aplikacja filtry 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 rozgłaszenia a w drugiej do trybu skanowania.



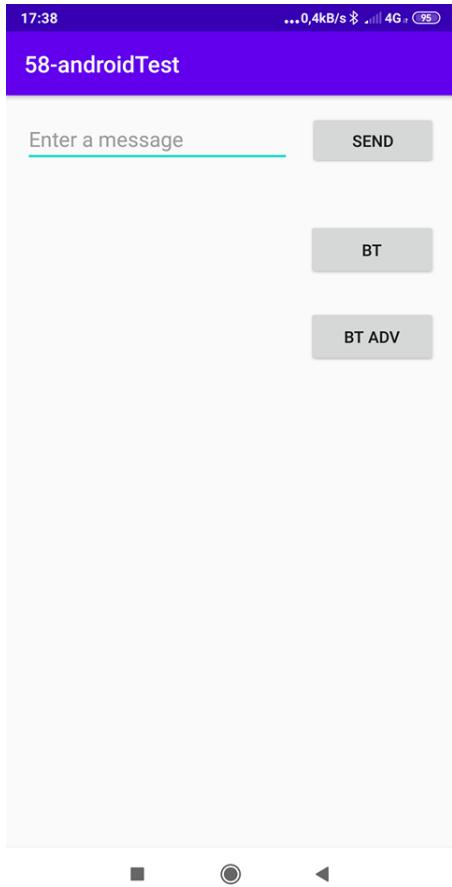
Rysunek 20: Ekran uruchomionej aplikacji testowej

Na rysunku 22 przedstawiono obraz głównego okna aplikacji, na który zaznaczono przycisk przenoszący do opisywanej części aplikacji.



Rysunek 21: Ekran uruchomionej aplikacji testowej

Na rysunku 23 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 24 i 25 przedstawiono logi po zakończeniu procesu.

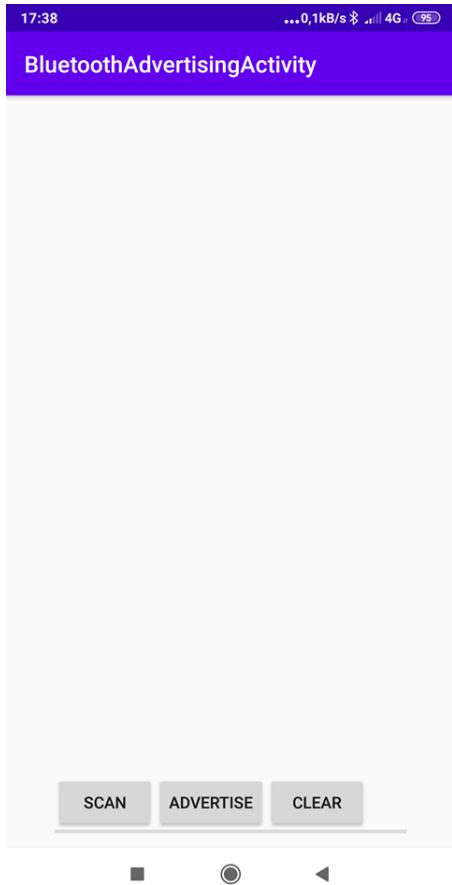


Rysunek 22: Ekran główny aplikacji testowej

9 Instrukcja instalacji Arduino IDE oraz biblioteki i2C oraz ESP32

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

1. Pobrać instalator Arduino IDE 1.8.12 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`



Rysunek 23: Ekran trybu advertisement aplikacji testowej

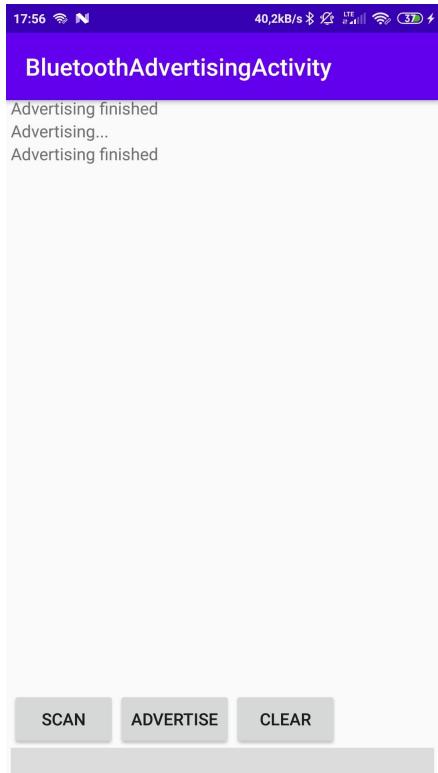
5. Wkleić folder LiquidCrystal_I2C-1.1.2

Następnie potrzebny będzie zestaw narzędzi służący do programowania modułu ESP32. W tym celu:

1. W otwartym programie Arduino IDE wybrać zakładkę Plik, następnie Preferencje,
2. W polu dodatkowe adresy URL dla menedżera płytek wkleić następujący link: <https://dl.espressif.com>
3. Zatwierdzić przyciskiem OK
4. W zakładce narzędzi wejść w menu wyboru płytek, a następnie w pole Menedżera płytek. Po uruchomieniu samodzielnie program zacznie pobierać pakiet dla ESP. Po zakończeniu pobierania można będzie wybrać płytke z menu w zakładce narzędzia.

10 Realizacja programu Arduino i ESP32

W celu funkcjonalnej pracy każdego z symulatorów poszkodowanych konieczne było zaprogramowanie modułów Arduino w taki sposób, aby wyświetlały na module LCD dane potrzebne

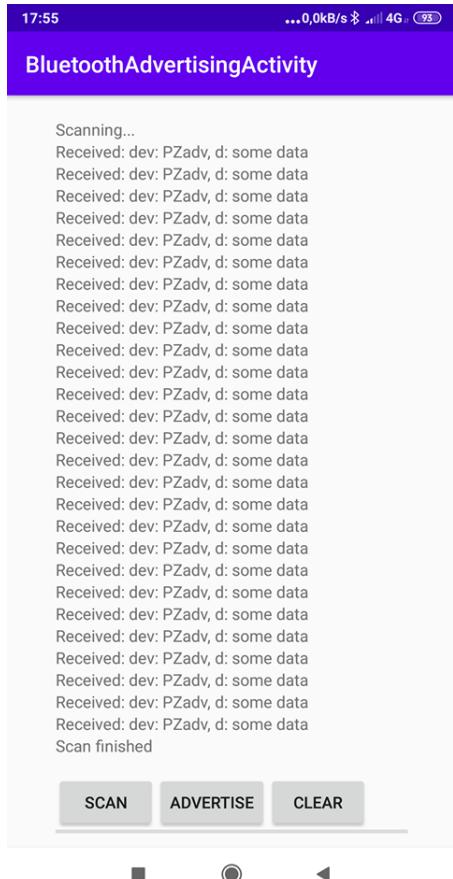


Rysunek 24: Log po przeprowadzeniu procesu wysyłania

ratownikowi do klasyfikacji poszkodowanego. Dodatkowo dane te muszą być zdalnie modyfikowalne, docelowo przez aplikację KAM-a.

Wyświetlacz pokazuje następujące parametry:

- Puls,
- oddech,
- czy poszkodowany chodzi,
- czy rozmawia,
- czas nawrotu kapilarnego.



Rysunek 25: Log po przeprowadzeniu procesu skanowania

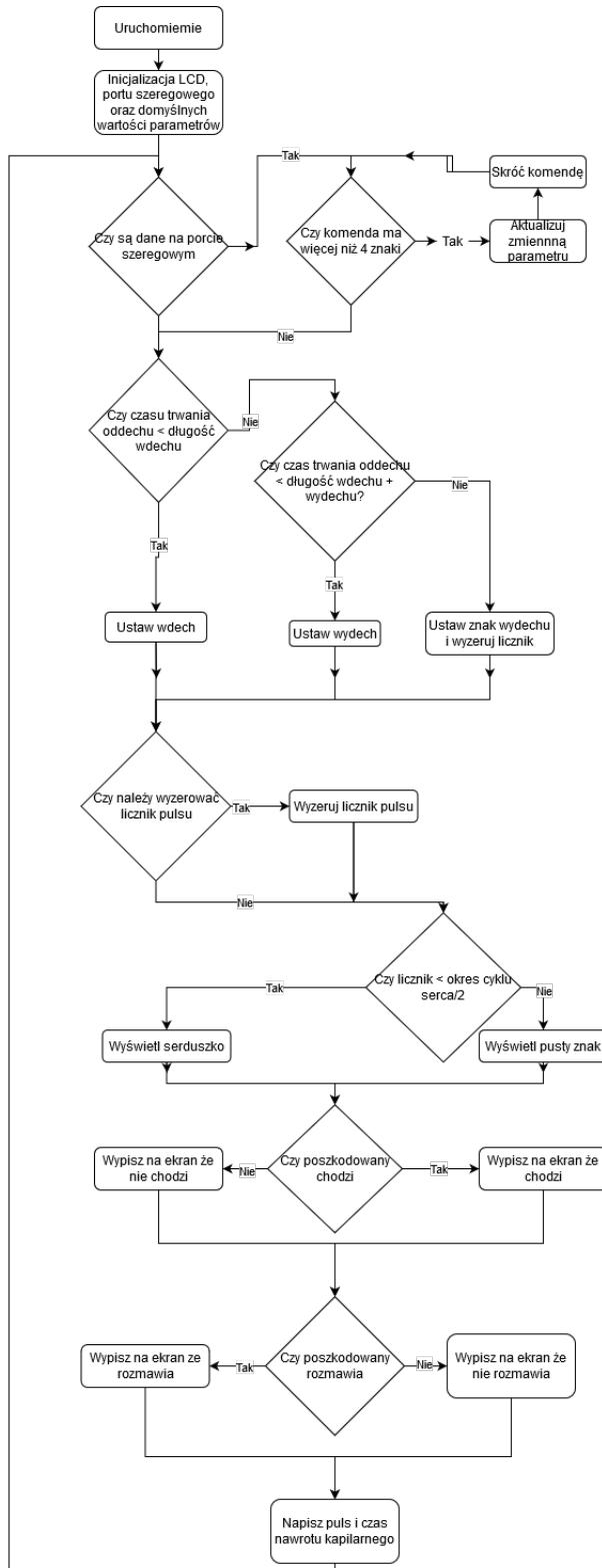
Puls oraz czas nawrotu prezentowany jest w formie liczbowej, informacje czy pacjent chodzi i rozmawia jest prezentowane w formie tekstowej, a oddech za pomocą strzałki zmieniającej się w zależności od trwającego procesu - strzałka w góre reprezentuje wdech, a strzałka w dół wydech.



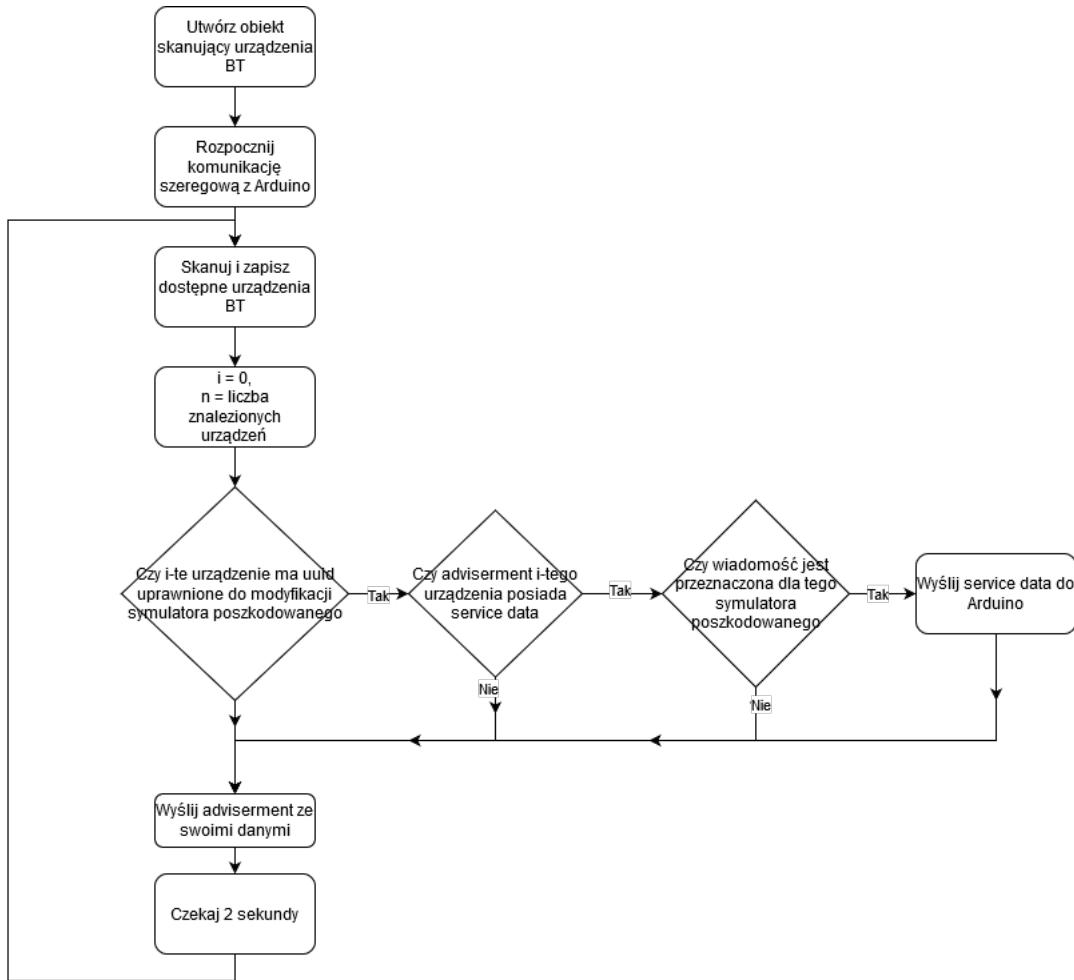
Rysunek 26: Parametry życiowe na wyświetlaczu LCD

10.1 Schematy blokowe programów

10.1.1 Arduino



10.1.2 ESP32



10.2 Modyfikacja parametrów

Modyfikacja parametrów wyświetlanych przez moduł odbywa się za pomocą portu szeregowego, podłączonego do ESP32. Komenda przesłana z ESP zawiera informacje dotyczące stanu parametrów pacjenta, według następującego kodu:

Parametr	Kod	Jednostka	Wartość
Puls	A	1bpm	000-999
Oddech	B	10ms	000-999
Chodzenie	C	Boolean	000-999
Rozmowa	D	Boolean	000-999
Nawrót kap.	E	0,1s	000-999
Numer simulatora	F	1	000-999
Prośba o dane	G	brak	dowolna

Komenda składa się z kodu i bezpośrednio po nim trzycyfrowej (np 001, 100, 050) wartości z podanego zakresu. Dla parametrów chodzenia i rozmawiania jeżeli wartość parametru jest parzysta oznacza to że nie, w przeciwnym przypadku że tak. Przykładowo, chcąc usta-

wić nawrót kapilarny na 0,5s użyć należy komendy 005. 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!B021!C037#. Dodatkowo komendy pozwalają na ustawienie numeru symulatora poszkodowanego. Ostatnia komenda prośby o dane jest komendą wewnętrzną symulatora poszkodowanego. Powoduje ona wysłanie przez moduł Arduino ciągu danych obecnie ustawionych w symulatorze aby mozna było te dane wysłać przez sieć Bluetooth.

Komendy są przesyłane w ramkach opisanych w protokole komunikacyjnym. Na obecnym etapie są one izolowane do zmiennych tekstowych, nie powodują jeszcze modyfikacji parametrów na wyświetlaczu.

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

10.3 ESP i komunikacja Bluetooth

Aby ESP32 modyfikowało parametry wyświetlane przez Arduino potrzebne jest dostarczenie tych danych do wspomnianego modułu. W tym celu aplikacja osoby modyfikującej te parametry wysyła Adviserment przez Bluetooth, który jest odbierany przez ESP32 i dekodowany. Jako Service dostarczany przez Adviserment jest przekazywany ciąg znaków z określeniem parametrów takich komenda modyfikująca, docelowy moduł poszkodowanego oraz inne.

10.4 Uruchomienie

Aby uruchomić cały symulator poszkodowanego należy poczynić następujące kroki:

1. Podłączyć Arduino odpowiednim kablem USB do komputera,
2. uruchomić kod źródłowy przeznaczony dla Arduino w programie Arduino IDE
3. upewić się że wybrana jest odpowiednia płytka (Arduino UNO),
4. skompilować a następnie wgrać program na płytke za pomocą okrągłych przycisków w lewym górnym rogu programu,
5. powtórzyć kroki 2-4 dla układu ESP32,
6. odłączyć płytke od komputera,
7. podłączyć według schematu wszystkie elementy układu a następnie podłączyć zasilanie.

11 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/>
6. https://pl.wikipedia.org/wiki/Wersje_systemu_Android

12 Materiały

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

Zakres prac	asdasd
Wykonał	ZZZZ
Sprawdził	XXXX
Zatwierdził	YYYY