

# Dokumentacja projektu **„Reflekto”**

Michał Kwiecień  
Michał Wójcik

9 kwietnia 2017

## **Streszczenie**

Dokumentacja projektu inteligentnego lustra w konwencji IoT komunikującego się ze smartfonem z użyciem interfejsu Bluetooth Low Energy. Projekt powstał na potrzeby konkursu Nordic Semiconductor Student Contest.



## Spis treści

1	Ogólny opis projektu	3
2	Prezentacja działania	3
3	Planowane możliwości personalizacji	4
4	Opis aplikacji systemu nRF52	5
5	Opis aplikacji systemu iOS (stan na 9 kwietnia 2017)	5
5.1	Pobierane dane i sposób ich pobrania . . . . .	5
5.2	Sposób działania . . . . .	6
5.3	Planowany rozwój . . . . .	6

## 1 Ogólny opis projektu

Założeniem projektu jest stworzenie inteligentnego lustra, które podczas wykonywania codziennych czynności, umożliwi podgląd najświeższych i spersonalizowanych informacji. Informacje te zostaną wyświetlone na ekranie umieszczonym za lustrem weneckim, dzięki czemu będą widoczne jednocześnie obok odbicia.

Działanie lustra opiera się na przekazaniu danych poprzez moduł Bluetooth ze smartfona do modułu nRF52 i umieszczeniu ich na podłączonym ekranie. Aktywacja lustra nastąpi w momencie zbliżenia się do niego użytkownika. Z lustra może korzystać wielu użytkowników, gdyż każdorazowo przesyłane są indywidualne dane dla każdego z nich.

W celu wygenerowania danych stworzona została dedykowana aplikacja dla systemu iOS. Po wstępnej konfiguracji umożliwi ona zautomatyzowanie procesu i przesyłanie wiadomości w tle bez późniejszych ingerencji użytkownika.

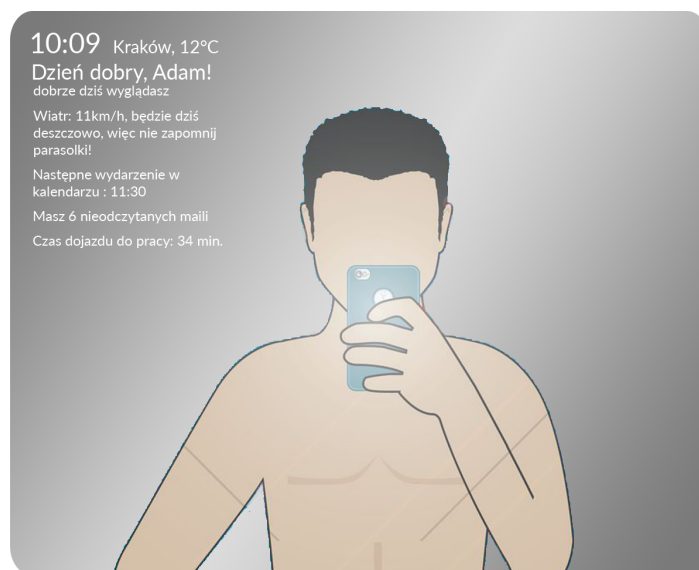
## 2 Prezentacja działania

Kiedy lustro nie jest w bliskim zasięgu jednego ze sparowanych telefonów, wyświetlana jest godzina lub pozostaje wyłączona w zależności od ustawień (Rys. 1)



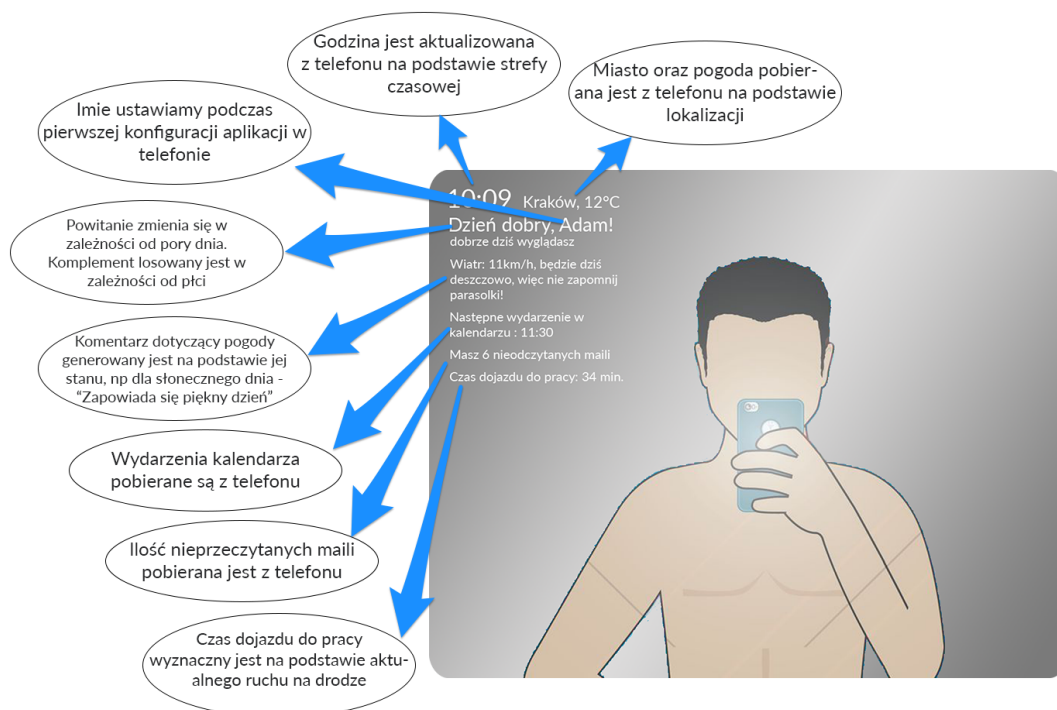
Rysunek 1: Lustro w stanie wyłączonym

W momencie zbliżenia się użytkownika, następuje transmisja danych i wyświetlenie aktualnych wiadomości (Rys. 3).



Rysunek 2: Lustro w stanie aktywnym

### 3 Planowane możliwości personalizacji



Rysunek 3: Proponowane możliwości personalizacji

## 4 Opis aplikacji systemu nRF52

## 5 Opis aplikacji systemu iOS (stan na 9 kwietnia 2017)

Aplikacja mobilna po uruchomieniu automatycznie rozpoznaje lokalizację użytkownika z dokładnością ok. 3km oraz próbuje nawiązać połączenie bluetooth z modulem bluetooth lustra. Aktualny wygląd aplikacji pokazany jest obrazie (Rys. 4)



Rysunek 4: Aktualny wygląd aplikacji

### 5.1 Pobierane dane i sposób ich pobrania

Aplikacja pobiera dane asynchronicznie, komórki z danymi w aplikacji uzupełniają się od razu po pobraniu poszczególnej danej. Nie trzeba czekać na zaaktualizowanie interfejsu aż do momentu pobrania każdego elementu. Poniżej opis pobieranych danych oraz w przypadku bardziej złożonych

danych sposób ich pozyskania.

- Godzina
- Miejsce, pogoda – na podstawie wyznaczonej lokalizacji telefon wysyła zapytanie do Dark Sky API podając w parametrze m. in. długość oraz szerokość geograficzną. Otrzymany w odpowiedzi JSON parsowany jest na obiekt, z którego później wybierana jest temperatura. Miejsce wyznaczone jest również na podstawie szerokości i długości geograficznej używając wbudowanego API Apple Maps.
- Imię – imię użytkownika rozpoznawane jest automatycznie na podstawie nazwy urządzenia. Domyślnie nazwa urządzenia to np. „Michał’s iPhone”. W przypadku innych wersji językowych nazwa urządzenia może być inna, więc metoda wykrywająca imię może nie zadziałać. Imię, które wykryjemy będzie jedynie propozycją podczas metody konfiguracji, jeśli użytkownik stwierdzi, że nie jest to zgodne z prawdą będzie mógł je ręcznie wpisać.
- Płeć – wyznaczana jest na podstawie ostatniej litery imienia, również będzie możliwa zmiana podczas procesu konfiguracji.
- Pogoda, dod. – pobierana z Dark Sky API
- Pogoda, rada – pobierana z Dark Sky API

## 5.2 Sposób działania

Po dotknięciu w poszczególnej komórce zostaje wygenerowany String z dodanymi na początku dwoma bajtami, które pozwalają systemowi wbudowanemu zidentyfikować rodzaj informacji, która jest mu wysyłana. Po stworzeniu odpowiednich danych następuje próba wysłania informacji. Wykonywana jest na osobnym wątku, aby nie blokować interfejsu użytkownika.

W dolnej części interfejsu zostało dodane pole tekstowe ułatwiające proces przygotowania systemu wbudowanego. Można dzięki niemu wpisać dowolną informację, w jego przypadku nie są dodawane dodatkowe dwa bajty na początku.

## 5.3 Planowany rozwój

Aplikacja zostanie rozbudowana o proces konfiguracji, który użytkownik będzie musiał przejść tylko za pierwszym razem. Następne uruchomienia nie będą już wymagane, aplikacja będzie działać w tle i na podstawie skanera bluetooth uruchamiać się w pobliżu lustra. Dane będą dzielone na paczki i wysyłane do modułu bluetooth lustra.

Zostaną dodane kolejne dane pobierane z urządzenia, takie jak komplement generowany na podstawie płci lub informacje o nadchodzących wydarzeniach w kalendarzu.

Dodatkowo aplikacja zostanie przepisana, tak aby używała biblioteki RxSwift, która umożliwia pisanie kodu w sposób reaktywny i funkcjonalny. Do niektórych funkcji dodane zostaną również testy jednostkowe, oraz jeśli czas na to pozwoli integracja z systemem Travis w celu zastosowania continuous integration.