

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 (stan na 9 kwietnia 2017)	5
4.1	Włączenie urządzenia	5
4.2	Zasilanie	5
4.3	Sposób rozgłaszania	5
4.4	Realizacja komunikacji	5
4.5	Umieszczanie informacji na wyświetlaczu	5
5	Opis aplikacji systemu iOS (stan na 9 kwietnia 2017)	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.

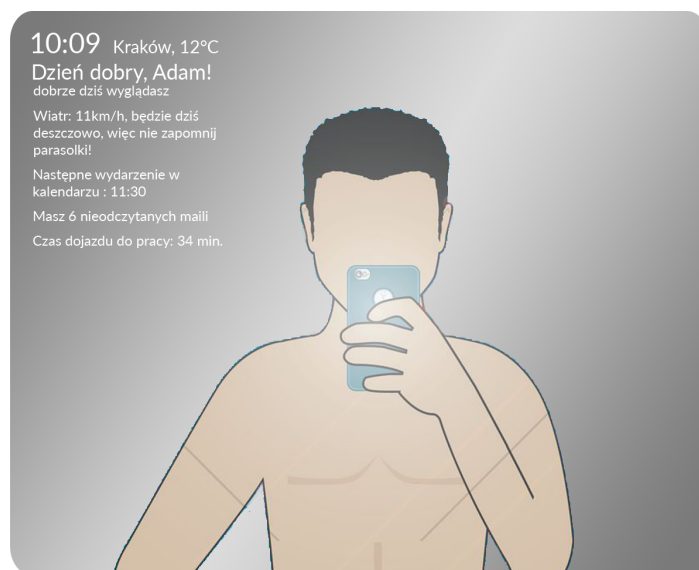
2 Prezentacja działania

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



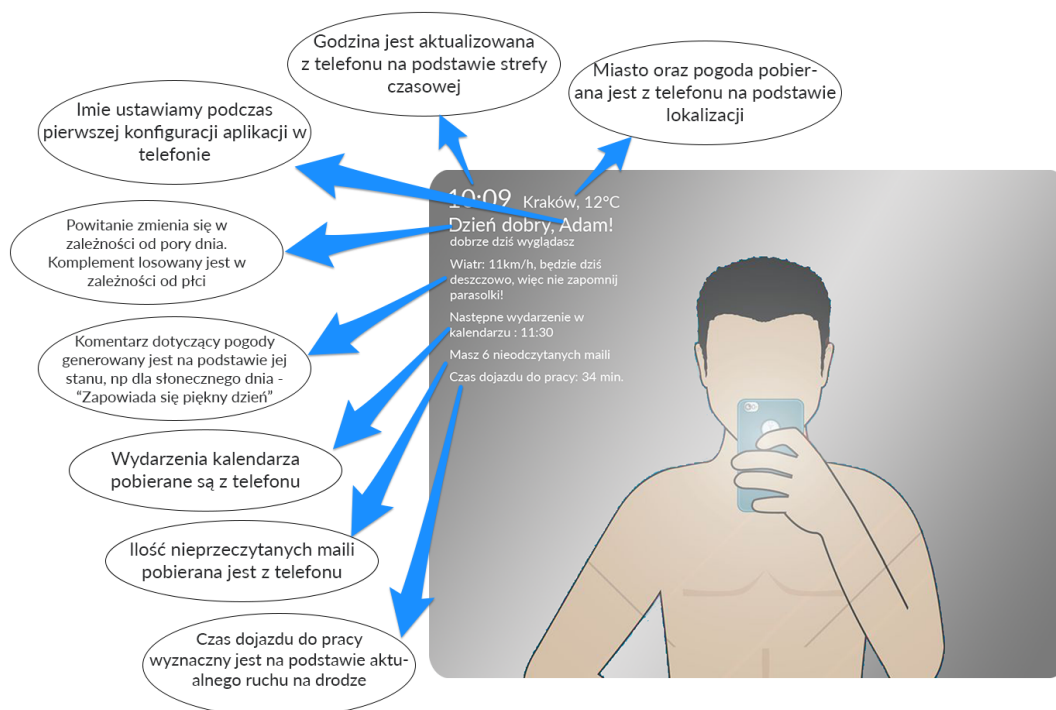
Rysunek 1: Lustro w stanie wyłączonym

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 (stan na 9 kwietnia 2017)

4.1 Włączenie urządzenia

Urządzenie po uruchomieniu inicjalizuje wszystkie niezbędne usługi: timer, uart, stos BLE, serwisy oraz obsługę wyświetlacza. Dalej rozpoczyna się rozgłaszanie (pod nazwą „Reflekto_UART”) i urządzenie pozostaje w stanie oczekiwania aż do wyłączenia zasilania.

4.2 Zasilanie

Ze względu na użycie wyświetlacza TFT, wymagane jest zasilanie przez zasilacz zewnętrzny bądź port USB (bateria pastylkowa nie pokrywa zapotrzebowania na prąd). Obecnie szacowany pobór energii w stanie aktywnym wynosi mniej niż 1W. W stanie nieaktywnym pobór energii przez wyświetlacz może zostać zredukowany do zera (sterowanie PWM), jednakże wymaga to wlutowania odpowiedniej zworki w płytce wyświetlacza.

4.3 Sposób rozgłaszania

Do urządzenia w tym samym czasie może być podłączony jeden smartfon. W zależności od ustawień, może być on połączony do momentu zachowania bliskiego zasięgu (potencjalnie blokując innych użytkowników), lub rozłączać się od razu po przesłaniu informacji.

Układ rozgłasza się pod wcześniej zdefiniowaną nazwą i UUID, udostępniając jeden serwis, z charakterystyką „write” oraz „notify” (obecnie niewykorzystana).

4.4 Realizacja komunikacji

Układ w momencie otrzymania danych wysyła je do komputera przez port szeregowy (w celu debugowania), oraz rzutuje je na wektor typu *char*, który przekazywany jest do kolejnej funkcji realizującej umieszczenie danych na wyświetlaczu.

Ze względu na ograniczenie ilości danych w pakiecie do 20 bajtów zastosowano specjalne kodowanie, gdzie pierwsze dwa bajty analizowane są w następujący sposób:

- bajt „0”

Wskazuje na typ otrzymanych danych (np. data, godzina, pogoda, kalendarz, e-mali)

- bajt „1”

4 bity górne – numer aktualnie otrzymanego pakietu

4 bity dolne – ilość wszystkich pakietów danego typu

Wynika z tego maksymalna długość przesyłanych danych jednego typu do 288 znaków. Podział na paczki realizowany jest przez aplikację na smartfonie. Założono także obsługę błędnych pakietów przez SoftDevice’a, dlatego nie uwzględnione zostały bajty na sumę CRC.

4.5 Umieszczanie informacji na wyświetlaczu

Po otrzymaniu wszystkich paczek danego typu, dane „drukowane” są na wyświetlaczu (ze względu na metodę komunikacji niemożliwe jest przesłanie jednorazowo całej ramki obrazu). Wyświetlacz podzielony został na sekcje (od góry):

Możliwe korzystanie przez kilku użytkowników jednocześnie

4 bity można użyć do „szyfrowania” transmisji

- 20% – aktualna data i godzina
- 20% – aktualna pogoda
- 60% – kalendarz, czas dojazdu do pracy, emaile itd. wyświetlane cyklicznie bądź jednocześnie

4.6 Konfiguracja

W planach jest obsługa danych konfiguracyjnych, które ustalałyby np. sposób wyświetlania informacji, lub działania układu w trybie nieaktywnym.

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