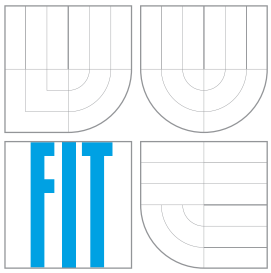
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MONITOROVACÍ SYSTÉM KOMUNIKUJÍCÍ S MOBILNÍM ZAŘÍZENÍM

MONITORING SYSTEM COMMUNICATING WITH MOBILE DEVICE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

FILIP ILAVSKÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PAVEL OČENÁŠEK, Ph.D.

BRNO 2016

Zadání bakalářské práce/18815/2015/xilavs01

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Ilavský Filip**
Obor: Informační technologie
Téma: **Monitorovací systém komunikující s mobilním zařízením**
Monitoring System Communicating with Mobile Device
Kategorie: Počítačové sítě

Pokyny:

1. Seznamte se s technologiemi na bezdrátovou komunikaci, zaměřte se na nízkou spotřebu a kompatibilitu s běžnými zařízeními jako jsou chytré telefony a počítače.
2. Navrhněte protokol pro zabezpečenou komunikaci modulu s klientskými zařízeními a aplikaci pro vybraný komunikační modul.
3. Navrhněte klientskou aplikaci komunikující s modulem a zobrazující naměřená data.
4. Navržené aplikace implementujte.
5. Navrhněte možnosti komunikace s klientem i mimo dosah signálu.
6. Diskutujte získané výsledky a možnosti dalšího rozšíření.

Literatura:

- Tanenbaum, A.S.: Computer Networks. Fourth Edition, Prentice Hall, 2003.
- Kurose, J.F., Ross, K.W.: Computer Networking, A Top-Down Approach Featuring the Internet. Addison-Wesley, 2003.
- Fraden, J.: Handbook of Modern Sensors: Physics, Designs, and Applications, AIP Press, 2003.

Pro udělení zápočtu za první semestr je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Očenášek Pavel, Ing., Ph.D.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2015
Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Táto bakalárska práca sa zaoberá návrhom a implementáciou hardvéru periférie merajúcej teplotu, a implementáciou aplikácie, zobrazujúcej namerané dáta. Bezdrôtová komunikácia medzi perifériou a klientom je implementovaná pod protokolom Bluetooth Smart (verzia 4.0 a vyššie) a zabezpečená bezpečnostným protokolom proti útočníkovi zachytávajúcemu komunikáciu. Výsledkom práce je plne funkčný prototyp periférie merajúcej teplotu a aplikácia zobrazujúca túto teplotu, navrhnutá pre platformu iOS verzie 8.0 a vyššej.

Abstract

This thesis deals with design and implementation of hardware peripheral measuring temperature, and implementation of client application, displaying the measured data. Wireless communication between the peripheral and the client is implemented under the protocol Bluetooth Smart (version 4.0 and higher) and secured by a security protocol against the attacker capturing this communication. The result of this thesis is a fully functional peripheral prototype measuring temperature and application displaying this temperature, designed for iOS version 8.0 and higher.

Kľúčové slová

Teplotný senzor, AES, iOS, Swift, bezdrôtová komunikácia, Bluetooth Low Energy, QR kód

Keywords

Temperature sensor, AES, iOS, Swift, wireless communication, Bluetooth Low Energy, QR code

Citácia

ILAVSKÝ, Filip. *Monitorovací systém komunikující s mobilním zařízením.* Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Očenášek Pavel.

Monitorovací systém komunikující s mobilním zařízením

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Pavla Očenáška Ph.D. Ďalšie informácie mi poskytol Ing. Miloš Beňadik. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Filip Ilavský
10. mája 2016

Poďakovanie

Ďakujem vedúcemu Pavlovi Očenáškovi za usmerňovanie a rady pri rozhodovaní, ako ďalej. Tiež ďakujem Milošovi Beňadikovi za odbornú pomoc a poradenstvo z praxe ohľadom výberu a návrhu hardvéru.

Obsah

Kapitola 1

Úvod

Bezdrôtová komunikácia sa v dnešnom svete rozširuje do viac a viac odvetví, pokrýva čoraz viac problémov kde konektivita pomocou káblového riešenia nie je prakticky použiteľná, alebo vôbec nie je možná. Vývoj posunul hranice bezdrôtovej komunikácie na veľmi žiadanú a stabilnú úroveň najmä v riešení distribúcie internetu. Svoje využitie nachádza aj v rôznych perifériách, ako sú napríklad bezdrôtové klávesnice, myši, audio zariadenia alebo takzvané smart riešenia, meracie prístroje a podobne.

Témou tejto bakalárskej práce je návrh a implementácia zariadenia snímajúceho teplotu, využívajúceho technológiu Bluetooth Low Energy (Bluetooth 4.0 a vyššie) na následné odosielanie tejto teploty na mobilné zariadenie s operačným systémom iOS verzie 8.0 a vyššie. Práca sa skladá z dvoch hlavných častí. Prvá časť je hardvérový a softvérový návrh, a následná implementácia periférie merajúcej teplotu presnú na desatinu stupňa celzia vysielajúcu klientskej aplikácii pomocou čipu od spoločnosti Nordic Semiconductor, konkrétne model nRF51422. Druhá časť je návrh a implementácia aplikácie pre platformu iOS. Uvažujeme nutnosť implementácie bezpečnostného protokolu dostačujúceho potrebám komunikácie a kladieme dôraz na nízku spotrebu periférie, a kompaktnosť. Konkrétny cieľ použitia v praxi budeme ďalej v práci uvažovať napríklad detský teplomer, merajúci teplotu po celý čas periodicky, pokiaľ je fyzicky umiestnený na tele dieťaťa.

V nasledujúcich kapitolách si predstavíme výber súčiastok periférie, hardvérový návrh periférie, implementáciu a komunikáciu s klientskou aplikáciou, bezpečnostné riešenie komunikácie, klientskú aplikáciu a jej možné rozšírenia, a nakoniec potenciálne úpravy tejto práce na základe znalostí z praxe, získaných pri riešení práce.

Kapitola 2

Výber technológií

V tejto kapitole si zdôvodníme výber technológie Bluetooth Low Energy, konkrétneho chipu rady nRF5x[1] od Nordic Semiconductor a ďalších dôležitých súčastí ako sú prevodník, teplotný senzor a akcelerometer. Spoločnosť Nordic Semiconductor patrí medzi najlepších výrobcov riešení bezdrôtovej komunikácie, po boku so spoločnosťami ako napr. Broadcom, Texas Instruments, Microchip a mnoho ďalších.

2.1 Komunikačný protokol

Na bezdrôtovú komunikáciu výrobcovia ponúkajú niekoľko riešení ako napr. klasický Bluetooth, Bluetooth Low Energy, ANT™, ANT™+, Wi-Fi a mnoho ďalších, väčšinou zatiaľ komerčne nepodporovaných protokolov, alebo protokolov na jedno konkrétne použitie. Pri výbere je nutné uvažovať nemožnosť rozšírenia hardvéru bežného zariadenia so systémom iOS, teda nutnosť využitia povolených technológií platformou. Toto nás obmedzuje na klasický Bluetooth a Bluetooth Low Energy. Technológia Wi-Fi je v požadovanom riešení prakticky nepoužiteľná, a to hneď z dvoch dôvodov. Prvým je vysoká spotreba a druhým zbytočná nutnost nadmernej komunikácie, čo si vyžaduje samotná technológia vyplývajúc zo štandardov [5]. Protokoly ANT™a ANT™+ by využiteľné v náš prospech určite boli, a to vďaka spotrebe, v určitých prípadoch dokonca nižšej ako Bluetooth Low Energy, a taktiež pokročilosti protokolu v porovnaní s Bluetooth Low Energy, podľa tvorcu protokolu[2]. Tieto protokoly majú potenciál rastu a budúceho využitia, môžeme nájsť ich podporu už aj v najnovších mobilných zariadeniach so systémom Android. Klasický Bluetooth je jednou z možností, avšak jeho spotreba je príliš vysoká, a nie sme odkázaní na jej využitie, nakoľko nepotrebujeme prenášať veľký objem dát, a nie sme závislí od rýchlosti prenosu, ani reakcií periférie.

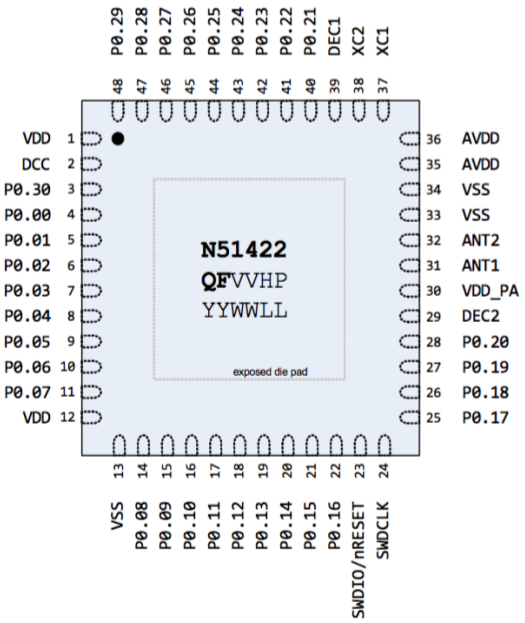
Bluetooth Low Energy úplne spĺňa všetky požiadavky práce. Protokol je podporovaný na väčšine moderných mobilných zariadení a počítačoch, má nízku spotrebu, dostatočný dosah v rámci cieľu použitia, veľkú kompaktnosť a taktiež záujem vývojárov, ktorý udržiava protokol v neustálom vývoji, a rozširuje do rôznych jednoduchých aj smart riešení [6]. Taktiež v uvažovaní nad praktickým využitím ako je detský teplomer, zaváži zdravotná nezávadnosť protokolu Bluetooth pri vysielaní, čo ocenia hlavne rodičia.

[1] uvažujeme konkrétne modely nRF51422 a nRF51822

[2] https://www.thisisant.com

2.2 Chip nRF51422

Riešenia s technológiou Bluetooth Low Energy (ďalej iba BLE) ponúkajú mnohé spoločnosti, jednou z nich je aj Nordic Semiconductor s chipmi rady nRF5x. Taktiež spoločnosť Broadcom ponúka produktovú radu BCM2073x, ktorá však v porovnaní s nRF5x vychádza horšie z pohľadu spotreby, veľkosti a dosahu signálu [3], čo sme dokázali aj na základe porovnávacích pokusov s vývojovým kitom Broadcom WICED Sense™.



Obr. 2.1: Rozloženie pinov chipu nRF51422, prevzaté z [9]

Výber chipu rady nRF5x, pre túto prácu, konkrétne nRF51422, má niekoľko ďalších dôvodov. Chip ponúka intenzitu signálu až do výšky +4dBm, spotrebu pri vysielaní pohybujúcu sa okolo 12mA, a 3μA v stand-by režime. Procesor obsiahnutý v chipe je často používaný a energeticky úsporný 32-bitový Cortex-M0, pamäť použiteľná bez pridávania externej flash pamäte je až do výšky 256kB a RAM až do výšky 32kB. Chip ponúka rôzne ďalšie riešenia, spomenuté sú iba využité v práci: GPIO, I2C (výrobca používa skratku TWI), časovače, zabudovaný analog/digital prevodník, DC-DC prevodník na zníženie celkovej spotreby a kryptovací 128 bitový AES koprocesor. Chip taktiež ponúka už zabudovanú podporu protokolu ANT™. Všetky dáta sú uvedené v referenčnom manuáli [9].

2.3 Tepelný senzor LMT70

LMT70 od výrobcu Texas Instruments je teplomer vhodný aj na meranie telesnej teploty, a to vďaka uvádzanej presnosti až 0.05°C v tomto intervale, čo môžeme považovať za dostatočné na rozlíšenie zvýšenej teploty (približne 37°C a viac), alebo napríklad aj podchaldenia (približne 36°C a menej). LMT70 má lineárny priebeh prevodu teploty na napätie [14], vhodný na jednoduché spracovanie analog/digital prevodníkom a následný prepočet na teplotu v stupňoch celzia. Jeho veľkosť 0.88 mm x 0.88 mm zaručuje veľmi nízku tepelnú kapacitu, a teda veľmi rýchle meranie. Bežná spotreba sa pohybuje v okolí 9.2μA. V prípade pásovej výroby produktu tejto práce, výrobca teplomera zaručuje pri vyrobených LMT70 z

jedného balenia pri odbere väčšieho množstva maximálny rozdiel nameraných hodnôt teplomerov v hodnote 0.1°C, čo úplne dostačuje pre potreby práce a možnej automatizácie výroby, bez nutnosti pracného premeriavania každého kusu a kalibrácie (postačí kalibrovať iba jeden kus z balenia).

Ako možné alternatívy na trhu sú tepelné senzory série TSYS01 a TSYS02 od spoločnosti Measurement Specialties™. V porovnaní s LMT70 majú výhodu v zabudovanom analog/digital prevodníku a možnosťou prístupu aj cez I2C [12]. Avšak ich veľkosť pohybujúca sa od 2.5 mm x 2.5 mm, spotreba vyššia o najmenej 3μA a presnosť o takmer jeden rád horšia jasne určujú LMT70 v kombinácii s externým prevodníkom za lepšiu voľbu.

2.4 Prevodník ADS1113

V rámci štúdie chipu nRF51422 sme narazili na možnosť využitia integrovaného analog/digitálneho prevodníku. Tento je však na splnenie presnosti merania a udržanie nízkej spotreby v tejto práci nepoužiteľný na presnejšie meranie teploty. Zabudovaný prevodník je maximálne 10-bitový s rýchlosťou merania 68μs a chybovosťou merania pohybujúcou sa až do výšky ±1.5% [9].

Preto sme zvolili externý prevodník, ku ktorému budeme pristupovať cez zbernicu I2C. Prevodník ADS1113 od spoločnosti Texas Instruments disponuje chybovosťou maximálne 0.15% na 16-tich bajtoch. Rýchlosť merania je v maximálne úspornom režime pohybujúca sa tesne nad 1.3μs [13]. Keďže rýchlosť prevodníku je vyššia ako zbernice I2C na nRF5x, nemusíme jeho rýchlosť pri implementácií ani uvažovať.

2.5 Akcelerometer LIS2DH

Chip nRF5x umožňuje režim maximálne zníženej spotreby, avšak jeho zobudenie závisí od hardvérového prerušenia. Akcelerometer je jedna z možností, ako tento problém riešiť. V prípade práce ako teplomeru telesnej teploty, vieme s vysokou pravdepodobnosťou povedať, že ak sa teplomer vôbec nehýbe, nie je pripevnený ku telu meranej osoby, ale napríklad je položený na poličke. V takomto stave, nazvime odloženým, nemá význam merať a odosielať hodnoty, nakoľko ide o telesný/detský teplomer. Avšak pri pohybe senzoru už vieme predpokladať, že stav nie je odložený. Teda akcelerometer tu zohráva dôležitú úlohu v prípade úplneho zníženia spotreby. Vďaka tejto vlastnosti by sa praktická výdrž batérie mohla blížiť až ku hodnote presahujúcej 1 rok.

Akcelerometer je v práci zakomponovaný ako možnosť rozšírenia, je pripravený na doske ale nie je využitý v implementácií hardvérovej časti práce. Model LIS2DH od spoločnosti STMicroelectronics disponuje veľmi nízkou spotrebou, okolo 2μA, dvoma výstupmi na hardvérové prerušenie, vlastným procesorom, zbernicou I2C a niekoľkými prednastavenými pohybmi, konfigurovateľnými množstvom registrov [11]. Z pohľadu spotreby a použiteľnosti sú riešenia od spoločnosti STMicroelectronics naozaj špička na trhu, ich riešenia môžeme nájsť v odvetviach od výrobcov inteligentných hodiniek a telefónov až po výrobcov automobilov, využívajúc riešenia napríklad na kontrolu trakcie. Možnou alternatívou je modul MMA8452Q od spoločnosti Freescale Semiconductor, ktorý má však príliš vysokú spotrebu, pohybujúcu sa nad hranicou 6μA až do 165μA, čo by strácalo zmysel keďže by jeho spotreba prevyšovala spotrebu modulu pri stálom aktívnom móde.

2.6 Batéria CR1220

Z dôvodu požiadavky kompaktnosti a dlhej výdrže sú ideálne gombíkové Lithium batérie. Samovybíjanie Lithium batérií je takmer nulové a kapacita úplne postačuje na predpokladanú spotrebu periférie. Batéria typu CR1220 úplne dostačuje kapacitou okolo 40mAh a vyhovuje aj veľkosťou už samotným porovnaním s chipom nRF5x (10 mm x 10 mm).

Ako referenciu pre odhad výdrže batérie budeme brať vysielanie/meranie každých 10 sekúnd. Použijeme výrobcami špecifikované hodnoty spotreby. Treba uvažovať spotrebu pri vysielaní, spotrebu pri meraní a spotrebu v nečinnom stave (IDLE). Všetky údaje sú uvedené podľa prduktových manuálov výrobcov [9][14][13].

- Jedno vysielanie BLE advertisement paketu v maximálnej intenzite +4dBm podľa výrobcu trvá približne 3ms, a jeho spotreba je priemerne 9mA, čo znamená, že pri jednom vysielaní spotrebujeme z batérie približne 2.7μAh.
- Jedno meranie s trvaním 2ms má nasledovnú spotrebu: spotreba procesora podľa výrobcu približne 2.4mA, spotreba zabudovanej zbernice I2C 400μA, zabudovaný analog/digital prevodník 200μA, AES koprocesor 550μA a externý analog/digital prevodník 260μA. Dokopy o spotrebe 3.81μA, bude pri každom meraní spotrebovávať kapacitu batérie 762nAh.
- Nakoniec spotreba v IDLE režime by sa mala pohybovať okolo 13.2μA, kde prevodník bude spotrebovávať 2μA, teplomer 10μA a samotný nRF5x 1.2μA, dokopy budú brať z batérie kapacitu 13.2μAh.

Celkovo bude výdrž v stálom aktívnom móde približne 2400 hodín, čo je približne 3.2 mesiaca. Údaje zadefinované výrobcami bývajú často veľmi nepresné, preto sa môže skutočnosť od výpočtov značne líšiť.

Kapitola 3

Zabezpečený komunikačný protokol

Jedným z cieľov práce je navrhnúť protokol pre zabezpečenú bezdrôtovú komunikáciu periférie s mobilným zariadením. Takýto protokol bude prevenciou proti útočníkom, ktorí by mohli komunikáciu zachytávať a získavať tak tieto dáta. V práci je použité symetrické šifrovanie, v návrhu protokolu si vysvetlíme prečo.

3.1 Návrh protokolu

Pri návrhu zabezpečenia je nutné prihliadať aj na celkové požiadavky výsledného produktu, a tým nevytvárať zbytočne zložitý komunikačný protokol, ktorý by vo výsledku mal príliš veľký vplyv na úbytok energie. Vieme, že účel periférie je dáta iba vysielať, čo nám značne uľahčuje celkový návrh. Taktiež chip nRF5x ponúka zabudovaný kryptovací AES koprocesor[9], ktorého využitie ušetrí energiu batérie, oproti softvérovo implementovanému protokolu. Teda jedna z hlavných požiadaviek, spotreba energie, nás jasne smeruje na využitie kryptovacieho algoritmu AES.

Keďže už vieme, aký algoritmus budeme používať, máme možnosti kryptovacieho AES kľúča definovaného dynamicky alebo staticky.

3.1.1 Dynamický AES kľúč

Ak máme mať AES kľúč dynamický, musíme ho nastaviť do periférie napríklad pri inicializácii komunikácie s klientom. Periféria si môže kľúč vygenerovať sama a odovzdať kľúč klientskej aplikácii, alebo bude kľúč generovaný na strane klienta a odoslaný periférii. Obe možnosti majú bezpečnostné riziko odchytenia tohto kľúča útočníkom. Napriek možnosti využiť ponúkané kryptovanie samotným protokolom BLE sa jeho využitie neodporúča, nakoľko bolo prelomené[1] a je považované za nie dostatočne bezpečné. V prípade nastavovania kľúča klientskou aplikáciou je potrebné využívať BLE párovanie so zariadním, čo má nasledujúce nevýhody:

- Vyššia spotreba.
- Náročnejšia implementácia, a s tým spojená zvýšená nestabilita.
- Obmedzenie merania na jedno klientské zariadenie.

[1] https://lacklustre.net/projects/crackle/

3.1.2 Statický AES kľúč

V prípade statického kľúča vieme tento odovzdať aplikácii bezpečnejšie. Možnosti sú:

1. Kľúč získaný zo servera na základe výrobného čísla.
2. Text nalepený na periférii slúžiaci na opísanie klientom do aplikácie obsahujúci kľúč.
3. QR kód nalepený na periférii obsahujúci kľúč.

Kľúč uložený na serveri

Klient po kúpe zariadenia získa kľúč periférie cez internet (napr. cez aplikáciu, alebo cez webovú stránku) a následne týmto kľúčom bude vedieť dáta dekryptovať. Periféria o celom procese nevie, iba vysiela dáta. Z pohľadu budúcnosti je tento spôsob výhodný, nakoľko chip nRF5x umožňuje update seba samého cez BLE, pomocou Bootloadera[9]. V tejto práci ale update programu implementovaný nie je, kľúč teda nepotrebuje byť, a ani nebude menený. To znamená strata potreby ukladania kľúča kdekoľvek v elektronickej forme, čím spôsob stráca význam pre využitie v tejto práci.

Text obsahujúci kľúč nalepený na periférii

Klient po zapnutí zariadenia (vložení batérie) opíše kľúč napísaný/nalepený fyzicky na periférii, aby mohol získavať dekryptované dáta z periférie. Problém môže nastať, ak klient nebude vedieť prečítať tento kód, resp. bude mať problém na klávesnici mobilného zariadenia napísať každý zo znakov. Tento problém by bolo možné riešiť obmedzením generovaných znakov AES kľúča na bežne používané znaky, čo však značne oslabuje bezpečnosť. Výhodou oproti serverovému riešeniu je odpadajúca potreba internetu na inicializáciu periférie, a teda nie je nutné prevádzkovať špeciálnu webovú službu pre zákazníkov.

Obr. 3.1: Ukážka QR kódu periférie

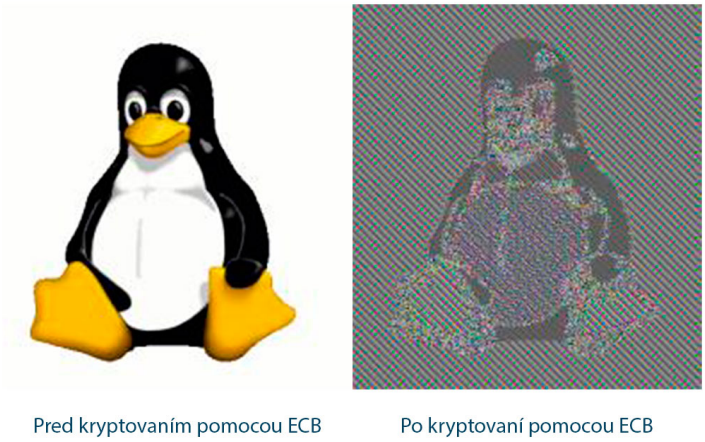
QR kód nalepený na periférii

Po zapnutí zariadenia v rámci aplikácie klient zoskenuje QR kód nalepený fyzicky na periférii, ktorý obsahuje kľúč na dekrytpovanie dát z periférie, vygenerovaný vo výrobe, tak ako pri čistom texte v predchádzajúcej možnosti, viď 3.1. Ku výhodám v porovnaní s čistým textom pribúda skutočnosť, že klient nemusí tento kľúč pracne opisovať, stačí jednoduché naskenovanie QR kódu. Tento QR kód môže obsahovať aj viacero informácií, ako napríklad identifikáciu konkrétnej periférie, čo umožní niekoľkým perifériam byť v jednej oblasti a vysielať simultánne.

3.2 Výsledné riešenie bezpečnosti

Každej periférii je priradený náhodne vygenerovaný AES kľúč pri výrobe. Tento AES kľúč, spolu s Bluetooth MAC adresou budú ako text pretransformované do QR kódu, ktorý napríklad vo forme tvrdej nálepky bude umiestnený na rovnej stene obalu periférie.

Aplikácia pre dekryptovanie dát z periférie bezpečne načíta QR kód z obalu periférie, uloží si načítané dáta, a následne už iba prijíma BLE advertise packety ktoré dekryptuje do výsledných spracovateľných dát (teplota a hodnota batérie). Teda pre získanie kľúča je nutný fyzický prístup ku periférii.



Obr. 3.2: Ukážka výstupu AES ECB algoritmu, prevzaté z [7]

Na šifrovanie použijeme protokol AES, konkrétne formu ECB (Electronic Code Book), ktorá sa neodporúča pri šifrovaní väčšieho množstva dát kvôli svojej periodicite výstupných dát [7], viď obrázok 3.2. My však potrebujeme kryptovať iba jeden blok (128 bitov) pri jednom meraní, takže ECB postačuje.

Pri implementovaní kryptovania a odosielania dát môže nastať situácia opakovaných dát, teda stabilizácie teploty na snímači. To by znamenalo opakujúce sa dáta aj na výstupe kryptovacej metódy, teda spomínaný problém ECB. Tento problém je riešený na strane periférie funkciou `nonce_generator`, ktorá systematicky inkrementuje nevyužité bajty nezakryptovaného bloku. Na týchto nám na strane klienta nezáleží, dosiahli sme hlavne skutočnosti, že útočník pri pokuse zistiť kľúč a následne vedieť dekryptovať dáta, sa nebude môcť na základe opakovaní zamerať na užší interval teploty. Funkcia je implementovaná nasledovne.

```
if((generator_position < TRUE_DATA_ENCRYPTED + BLE_GAP_ADDR_LEN))
    generator_position = TRUE_DATA_ENCRYPTED + BLE_GAP_ADDR_LEN;
if (my_data_plain[generator_position] == UINT8_MAX)
    my_data_plain[generator_position++]++;
if (generator_position >= ENCRYPTION_SIZE)
    generator_position = TRUE_DATA_ENCRYPTED + BLE_GAP_ADDR_LEN;
my_data_plain[generator_position]++;
```

Vo funkcii počítame s veľkosťou MAC adresy a nameraných dát, čo je dokopy 9 bajtov. Na naše generované dáta nám ostáva 7 bajtov, nakoľko jeden blok dát má veľkosť 16 bajtov (128 bitov), viď obrázok 4.5.

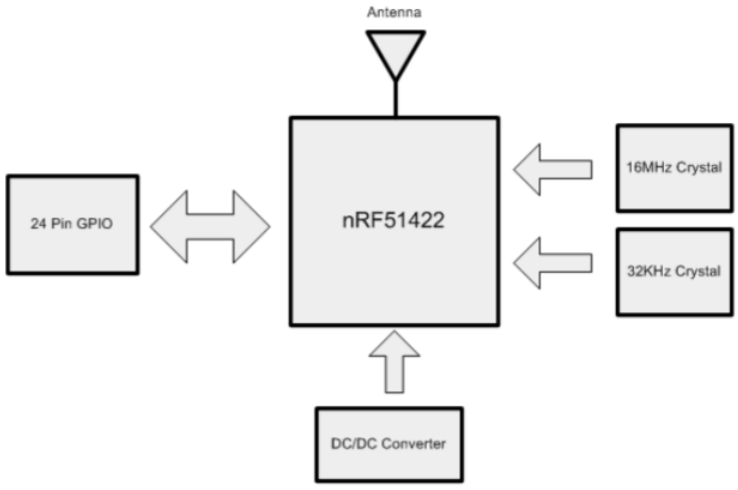
Kapitola 4

Hardvér periférie

V tejto kapitole čitateľovi upresníme fyzický návrh periférie, výber spôsobu BLE vysielania, a samotnú implementáciu na procesore Cortex-M0. Na návrh schémy zapojenia a dosky plošných spojov bol použitý program Eagle.

4.1 Certifikované hardvérové riešenie nRF5x

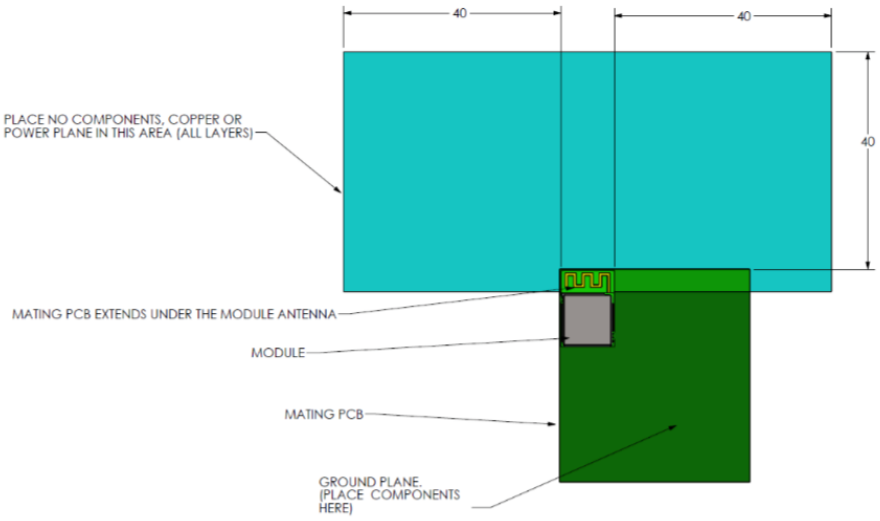
Nakoľko chip nRF51422 samostatne neponúka riešenie antény, a takéto riešenie si vyžaduje špeciálnu certifikáciu spoločnosťou Nordic Semiconductor, za účelom splnenia nám postačuje už certifikované riešenie od spoločností tretích strán. V práci je konkrétne využitý model N550M8CC od spoločnosti Dynastream. Tu je nutné brať ohlad na požiadavky výrobcu Dynastream vzhľadom na umiestnenie antény [4]. Pri zlom návrhu by mohol byť signál z antény úplne vyrušený okolitými súčiastkami a spojmi.



Obr. 4.1: Blokový diagram certifikovaného riešenia chipu nRF51422, od spoločnosti Dynastream, prevzaté z [4]

Na obrázku 4.1 vidíme riešenie modelovej rady N5 od spoločnosti Dynastream, ktoré zahŕňa DC/DC konvertor, 24 pinov GPIO, zabudované frekvenčné kryštály a najdôležitejšie, samotnú anténu. Podľa výrobcu akýkoľvek zásah do špecifikovanej oblasti okolo antény, viď

obrázok 4.2, bude mať negatívny vplyv na intenzitu signálu, v niektorých prípadoch signál úplne zanikne.



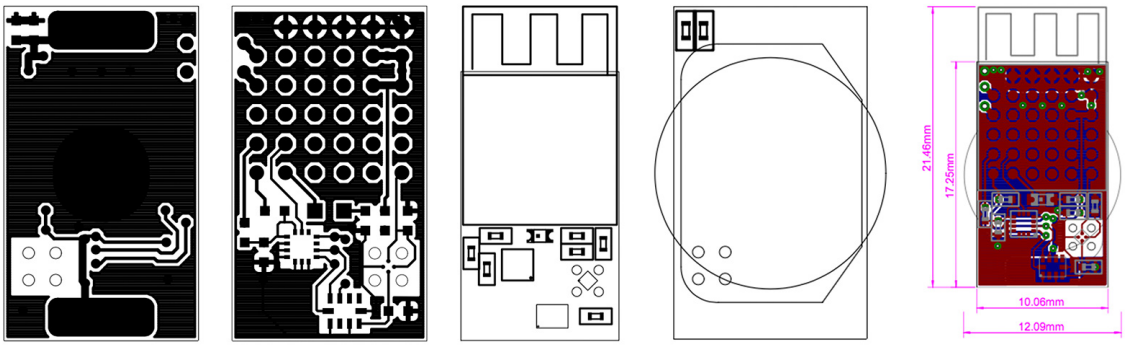
Obr. 4.2: Zóna zakáznej oblasti zásahu do okolia antény modelovej rady N5, prevzaté z [4]

4.2 Schéma zapojenia periférie

Na schéme zapojenia, viď prílohu A, si môžeme všimnúť využitia GPIO pinov a I2C zbernice, ktorá sama taktiež používa GPIO. GPIO pin `T_ON` zapína meranie na tepelnom senzore LMT70 logickou '1'. Nakoľko čas 'zobudenia' tohto čidla je vysoký a spotreba nízka, je tento pin zapnutý od inicializácie programu. Pin `LED` slúži na rozsvecovanie kontrolnej LED diódy, primárne určenej na účely vývoja a rýchly debug aplikácie. Led dióda má v sérii zapojený odpor na zníženie spotreby pri svietení, rozsvecuje sa logickou '0', teda uzemnením. Rezistory R1 a R2 slúžia ako pull-up rezistory pre zbernicu I2C, ktoré však nakoniec neboli nutné, nakoľko tieto už chip nRF5x obsahuje na každom GPIO pine. Schéma obsahuje niekoľko kondenzátorov, ktoré odporúčajú zakomponovať do schémy výrobcovia nRF5x, tepelného senzoru a akcelerometra. Výstup tepelného senzora je priamo zapojený do prevodníku, ktorého jedinou úlohou v tejto práci je konvertovať jeho výstup do digitálneho formátu. Prevodník porovnáva hodnotu so zemou. Akcelerometer má zapojený výstup jedného z dvoch hardvérových prerušení do GPIO nRF5x. Jeho funkcionalita v práci nie je využitá, preto tento vstup nie je nijako konfigurovaný. Akcelerometer a prevodník sú napájané konštantne, ich spotreba pri inaktívnom móde je dostatočne nízka.

4.3 Doska plošných spojov periférie

Kompaktnosť ako požiadavka nadobúda dôležitosť najmä pri návrhu dosky plošných spojov(ďalej iba DPS) celkového hardvérového riešenia. Keďže nemáme väčšie množstvo súčiastok, ani zložitú schému zapojenia, postačuje dvojvrstvová (obojstranná) DPS.



Obr. 4.3: Výstupy návrhu DPS z programu Eagle

4.4 BLE Advertisement

Protokol BLE ponúka dve základné možnosti komunikácie. Binding a advertisement. Binding, teda párovanie s klientom, obmedzuje periférium ku komunikácii iba s jedným klientským zariadením. Advertisement narozdiel od bindingu vôbec nevníma okolité zariadenia. Jeho výhodu je možnosť vysielania viacerím klientom zároveň.

4.5 SoftDevice a jeho výber

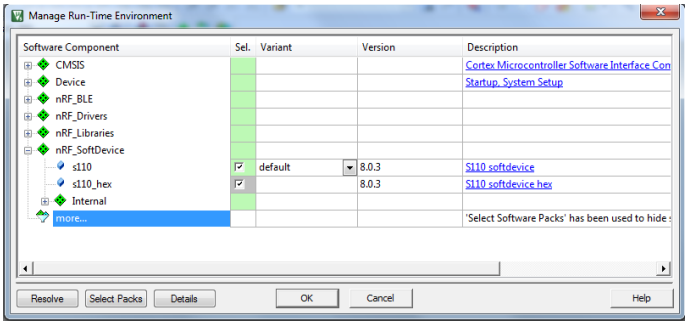
Pri vyberaní chipu na BLE komunikáciu zohrala dôležitú úlohu úroveň podpory od výrobcu pri vývoji aplikácií. Spoločnosť Nordic Semiconductor ponúka na vývoj vlastný druh frameworku, zvaný SoftDevice. Ide o programovú časť aplikácie, ktorá je vyvíjaná výrobcom na zjednodušenie implementácie aplikácií vývojárom. Ide o súbor funkcií a konštánt s určitými obmedzeniami, a taktiež zameraním na konkrétne modelové rady [10].

Pre nRF51422 je ponúkaných 5 SoftDevice druhov, v práci využívame najjednoduchší z nich, SoftDevice S110 verzie 8.0.0, ktorý je cielený na využitie iba ako BLE periféria. SoftDevice za vývojára implementuje funkcionalitu ako napríklad vysielanie dát, používanie AES koprocesora, používanie časovača a tvorbu softvérových prerušení, implementácia I2C zbernice, prístup ku GPIO a následná konfigurácia, práca s vnútorným analog/digital prevodníkom, získavanie informácií o hardvéri, redukcia spotreby energie, a mnoho ďalších v práci nevyužitých funkcionalít.

4.6 Vývojové prostredie Keil µVision

Na vývoj programu pre chip nRF5x sa v práci používa vývojové prostredie Keil µVision verzie 5. Napriek mnohým nedokonalostiam, je jeho využitie najrozumnejšiou voľbou pre väčšinu vývojárov rôznych aplikácií, nielen pre konkrétny chip nRF5x. Jeho hlavnou výhodou je integrovanosť s ponúkanými knižnicami priamo od výrobcu, ktoré je možné bez pracného odladovania kompilácie a dohľadávania vzájomných väzieb použiť pri riešení vývoja. Vývojové prostredie je určené primárne pre procesory ARM, teda na vývoj hardvérových aplikácií.

Výrobca ponúka demoštračné riešenia mnohých aplikácií práve v tomto IDE. Tým podporuje aj samotný vývoj aplikácií na vlastných výrobkoch, nakoľko povoľuje jednoduchší a časovo menej náročný vývoj, ktorý v určitých prípadoch a najmä pri využití SoftDevice,



Obr. 4.4: Package manager vývojového prostredia Keil µVision

vôbec nevyžaduje po vývojárovi expertnú znalosť technológií pri implementácií. Implementácia štandardov je už obsiahnutá v SoftDevice, ktorý je zároveň možné naimportovať do projektu v tomto IDE veľmi jednoducho a natívne využitím softvérových balíkov. Vývojár si jednoducho vyberie z balíkov konkrétne knižnice(možnosť výberu aj konkrétnej verzie) ktoré chce využiť v projekte, v okne znázornenom na obrázku 4.4.

IDE avšak nerieši úplne všetky nutné náležitosti za programátora, ten si musí naštudovať rozloženie RAM a ROM, a následne podľa využitia Bootloadera, SoftDevice a samotného vlastného kódu v projekte nastaviť pri kompilácii vlastného programu offset programu v ROM, a offset v RAM pamäti, ktorá bude rezervovaná pre SoftDevice(prípadne aj pre Bootloader) aby bol výstupný .hex súbor správne vykompilovaný. Taktiež nutné definície preprocesora pri kompilácii a konkrétne nastavenia kompilátora si vývojár nastaví podľa potreby.

Možné alternatívy ponúkané výrobcom sú IAR Embedded Workbench a ARM mbed. V týchto však výrobca neponúka podporu na takej úrovni ako v Keil, ukážkové príklady projektov pre tieto takmer neexistujú a neposledným problémom je samotná inštalácia, v niektorých prípadoch neúspešná kvôli nevyhovujúcemu hardvéru počítača vývojára(problém väčšinou nastáva pri príliš novom hardvéri). Neposledná možnosť je vývoj bez IDE, a priama kompilácia pomocou GCC, ktoré okrem iného používajú už spomenuté IDE na kompiláciu.

4.7 Popis implementácie

Meranie teploty

Snímač LMT70 je napájaný a zároveň zapínaný pomocou GPIO pinu, ktorý sme si nazvali `T_ON`, viď schému v prílohe A. Jeho výstup je priamo zapojený do jedného zo vstupov prevodníku ADS1113. Ku tomu budeme pristupovať zbernicou I2C, pre získanie hodnoty na tepelnom senzore. Do registrov prevodníku môžeme zapisovať, alebo z neho čítať hodnoty.

```
const uint8_t ads1113Write = 0x90;
const uint8_t ads1113Read = 0x91;
```

Nastavíme parametre konverzie do konfiguračného registra prevodníka. Zadefinujeme si mód zápisu, ostatné náležitosti komunikácie pomocou zbernice za nás implementuje knižnica.

```
const uint8_t length_select_config_register = 3;
uint8_t select_config_register[length_select_config_register] = {0x01, 0x81, 0xe0};
status = twi_master_transfer(ads1113Write, select_config_register,
    length_select_config_register, TWI_ISSUE_STOP);
```

Nastavíme ukazateľ ďalšieho čítania z registrov na konverzný register.

```
uint8_t select_conversion_register[1] = {0x00};
status = twi_master_transfer(ads1113Write, select_conversion_register, 1,
    TWI_ISSUE_STOP);
```

Prečítame hodnotu prevnodníku, teda 16 bitov po ukončenej konverzii.

```
uint8_t conversion_register[2] = {0x00};
status = twi_master_transfer(ads1113Read, conversion_register, 2, TWI_ISSUE_STOP);
```

Meranie batérie

Batériu vieme odmerať zabudovaným analog/digital prevnodníkom chipu nRF5x. Nameraná hodnota neurčuje ostávajúcu kapacitu batérie v percentách presne, je to približné informatívne určenie ostávajúcej kapacity získané z aktuálneho napätia batérie.

```
// Konfiguracia a inicializacia ADC
NRF_ADC->CONFIG = (ADC_CONFIG_RES_8bit                          << ADC_CONFIG_RES_Pos)
         |
                  (ADC_CONFIG_INPSEL_SupplyOneThirdPrescaling <<
                      ADC_CONFIG_INPSEL_Pos)  |
                  (ADC_CONFIG_REFSEL_VBG                        <<
                      ADC_CONFIG_REFSEL_Pos)  |
                  (ADC_CONFIG_PSEL_Disabled                     << ADC_CONFIG_PSEL_Pos
                      )    |
                  (ADC_CONFIG_EXTREFSEL_None                    <<
                      ADC_CONFIG_EXTREFSEL_Pos);
NRF_ADC->EVENTS_END = 0;
// Povolenie vnutorneho prevodnika chipu
NRF_ADC->ENABLE     = ADC_ENABLE_ENABLE_Enabled;

// Zastavenie prebiehajucej konverzie prevodnika
NRF_ADC->EVENTS_END  = 0;
// Zacatie merania
NRF_ADC->TASKS_START = 1;

// Cakanie na dokoncenie konverzie
while (!NRF_ADC->EVENTS_END)
{
}

// Konstanty merania
uint16_t vbg_in_mv = 1200;
uint8_t adc_max = 255;
// Vypocet
uint16_t vbat_current_in_mv = (NRF_ADC->RESULT * 3 * vbg_in_mv) / adc_max;

// Zastavenie konverzie
NRF_ADC->EVENTS_END     = 0;
NRF_ADC->TASKS_STOP     = 1;

// Vypnutie vnutorneho prevodnika chipu
NRF_ADC->ENABLE     = ADC_ENABLE_ENABLE_Disabled;

return (uint8_t) ((vbat_current_in_mv * 100) / VBAT_MAX_IN_MV);
```

Vysielanie zakrytpovaných dát

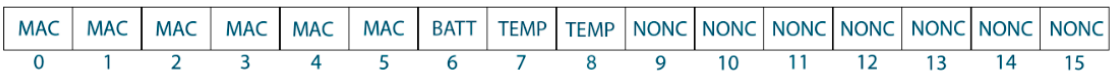
V projekte využívame SoftDevice S110 v8, čo nám rieši väčšinu daných pravidiel a štandardov, ktoré by sme inak museli pracne implementovať. V manažéri balíkov, viď obrázok 4.4, sme si zadefinovali knižnice pre kryptovanie, zbernicu I2C využitú pri meraní teploty, knižnicu pre GPIO, knižnicu pre použitie časovačov a najdôležitejšiu časť, knižnicu na vysielanie

v móde BLE advertising. Táto za nás periodicky vysiela dáta, ktoré si sami špecifikujeme a zadefinujeme cez funkciu tejto knižnice nasledovne.

```
ble_advdata_t advdata;
memset(&advdata, 0, sizeof(advdata));

...

advdata.name_type             = BLE_ADVDATA_FULL_NAME;
advdata.flags                 = flags;
advdata.p_manuf_specific_data = &manuf_specific_data;

err_code = ble_advdata_set(&advdata, NULL);
```

Môžeme si všimnúť napĺňanie štruktúry dát `manuf_specif_data`. Táto štruktúra slúži v BLE advertising móde na presun vlastných dát podľa potreby, my ju v práci používame na prenos zakryptovaného bloku dát. Tieto dáta si pripravíme nasledovne.

```
my_data_plain[0 + BLE_GAP_ADDR_LEN] = battery_level;
my_data_plain[1 + BLE_GAP_ADDR_LEN] = (uint8_t) (temperature_celsius >> 8) & 0xFF;
my_data_plain[2 + BLE_GAP_ADDR_LEN] = (uint8_t) temperature_celsius & 0xFF;

nonce_generator();

nrf_ecb_hal_data_t encryption_data;
memset(&encryption_data, 0, sizeof(nrf_ecb_hal_data_t));

memcpy(encryption_data.key, encryption_key, ENCRYPTION_SIZE);
memcpy(encryption_data.cleartext, my_data_plain, ENCRYPTION_SIZE);

sd_ecb_block_encrypt(&encryption_data);

manuf_specific_data.data.p_data = encryption_data.ciphertext;
manuf_specific_data.data.size   = ENCRYPTION_SIZE;
```



Obr. 4.5: Zloženie dátového bloku protokolu

Definícia `BLE_GAP_ADDR_LEN` slúži ako offset pre konštantne zapísanú MAC adresu zariadenia do dátového bloku, ktorého využitie je vysvetlené v sekcii 5.2. Funkcia `nonce_generator` nám upraví nevyužitú časť dátového bloku tak, aby v prípade dlhšie nezmenenej teploty a hodnoty batérie, výsledný zakryptovaný blok nebol rovnaký. Následne naalokujeme štruktúru `nrf_ecb_hal_data_t`, ktorú potrebuje funkcia `sd_ecb_block_encrypt` ako vstup pre kryptovanie. Táto funkcia výsledok zapíše do tejto štruktúry, z ktorej dáta následne nastavujeme do štruktúry `manuf_specific_data.data.p_data` ako dáta na odoslanie cez BLE advertising.

Ďalšou dôležitou časťou implementácie je použitie časovača, na prerušením spúšťané meranie a nastavovanie BLE advertising dát. V našom prípade implementujeme pomocou SoftDevice, ktorý si časovače rezervoval na výhradné použitie. Preto výrobca ponúka možnosť softvérového riešenia časovača pomocou SoftDevice plánovača (scheduler), ktorý inicializujeme cez preddefinované makro a následne spustíme.

```
APP_TIMER_INIT(APP_TIMER_PRESCALER, APP_TIMER_OP_QUEUE_SIZE, false);
app_timer_create(&m_measure_timer_id, APP_TIMER_MODE_REPEATED, main_timer_handler);

app_timer_start(m_measure_timer_id, MAIN_TIMER_INTERVAL, NULL);
```

Po spustení bude časovač periodicky volať handler prerušenia `main_timer_handler` v intervale definovanom v `MAIN_TIMER_INTERVAL`. Tento handler implementuje meranie a následné nastavenie BLE advertising dát.

```
temperature_celsius = temp_measure();
battery_level = battery_level_get();
advertising_data_set();
```

Nakoľko je pre nás dôležitý aj dosah signálu, nastavíme intenzitu signálu na maximálnych +4dBm pri inicializácii periférie a pre zníženie spotreby pri vysielaní zapneme DC/DC konvertor, znižujúci celkové napätie chipu nRF5x.

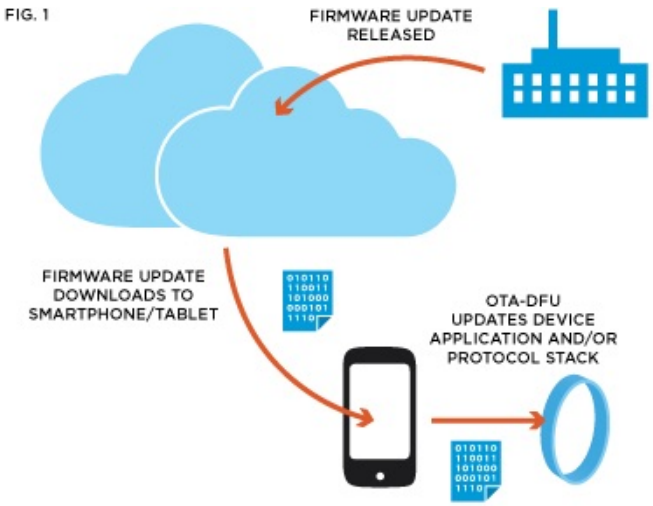
```
sd_ble_gap_tx_power_set(DEVICE_TX_POWER_LEVEL);
sd_power_dcdc_mode_set(NRF_POWER_DCDC_ENABLE);
```

4.8 Programovanie nRF5x

Nordic Semiconductor ponúka mimo štandardného káblového spôsobu prepisu programu aj bezdrôtový Over The Air (ďalej iba OTA) spôsob.

4.8.1 Over-the-Air Device Firmware Upgrade

Programovanie chipu nRF5x je možné aj bezkáblovo, pomocou Bootloadera. Možnosť tohto programovania je voliteľná, programátor musí chip základne naprogramovať, aby OTA podporoval. Princíp je veľmi jednoduchý. Periféria sa nachádza v stave čakajúca na pripojenie s cieľom aktualizácie firmvéru. Klientská strana(zariadenie disponujúce s novým firmvérom) začne s komunikáciou, a následne odošle periférii nový firmvér, tá si ho uloží do rezervovanej časti v pamäti RAM. Následne si overí správnosť podpisu tohoto firmvéru, v prípade nesprávnosti operáciu zruší a oboznámi klienta. V prípade správnosti môže stále nastať chyba, napríklad nekonzistentnosť samotného firmvéru. Teda bootloader zámerne ponecháva pôvodný firmvér v zálohe pre prípad, že nový zlyhá. Podľa výsledku testu firmvéru je operácia úspešná alebo neúspešná.



Obr. 4.6: Príklad použitia OTA-DFU, prevzaté z [8]

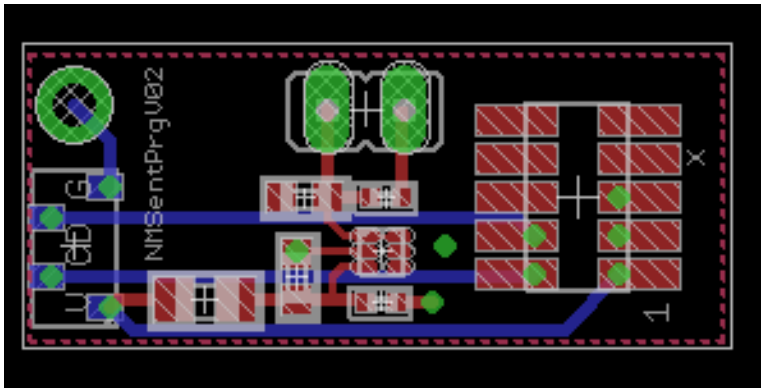
Tento spôsob nie je odporúčaný pri vývoji, nakoľko program býva často nestabilný, čo môže vo výsledku úplne zablokovať prepisovanie programu. Upgrade OTA je odporúčaný zo stabilnej verzie na novú stabilnú verziu, teda napríklad pri aktualizácií už predanej periférie zákazníkovi, pomocou aplikácie od samotného výrobcu a frekvencia takýchto aktualizácií je veľmi malá, viď obrázok 4.6.

4.8.2 Káblové programovanie

Klasické káblové programovanie si vyžaduje programátor a k tomu prislúchajúci driver/softvér pre počítač. V práci je využitý programátor od spoločnosti SEGGER Microcontroller. Model J-Link LITE CortexM verzie 8.1, určený pre procesory Cortex M0, M0+, M1, M3, M4 a M7 s podporou na väčšine operačných systémov je vhodný na účely tejto práce. Keďže výsledná periféria neobsahuje piny na zpojenie 9-pinového konektoru, a úprava 9-pinového káblu sa nejaví ako správne riešenie, je ku periférii navrhnutá ešte jedna DPS, určená výhradne na vývoj/programovanie periférie, pozri prílohu B.

Pri vývoji a odladovaní programu, je prepis programu častý, a ten spotrebuje značné množstvo energie. Preto je pomocný programátor navrhnutý tak, aby bolo možné pripojiť externú veľkokapacitnú batériu. Napätie tejto batérie znižujeme pomocou LDO na hodnotu 3V. Na schéme môžeme vidieť niekoľko rezistorov a kondenzátorov, kde všetky slúžia na filtráciu a stabilizáciu prúdu citlivých mikrosúčiastok.

Okrem toho sa na pomocnom programátore nachádza 9-pinový konektor, do ktorého budeme zapájať kábel z programátora, a nakoniec výstup smerom ku periférii. Tento výstup je priamo neodpojiteľne letovaný na periférii, po dokončení programovania je nutné kábliky odstrániť, inak by mohli spôsobovať rušenie a tým zoslabovať signál. Návrh DPS si môžeme pozrieť na obrázku 4.7



Obr. 4.7: DPS pomocného programátora

Kapitola 5

Klientská aplikácia

BLE protokol je podporovaný rôznymi platformami, nie len mobilných zariadení. Za účelom prezentácie periférie je v práci implementovaná aplikácia pre operačný systém iOS8.0 a vyššie.

5.1 Návrh aplikácie

Aplikácia na mobilné zariadenie je pomerne jednoduchá s prioritným účelom prezentácie zabezpečenej komunikácie. Pre vývoj natívnym spôsobom je nutné vývoj vykonávať na platforme OS X vo vývojovom prostredí XCode. Primárny programovací jazyk aplikácie je Swift 2 [2], spätne kompatibilný so svojím predchodcom Objective-C a taktiež s čistým C.

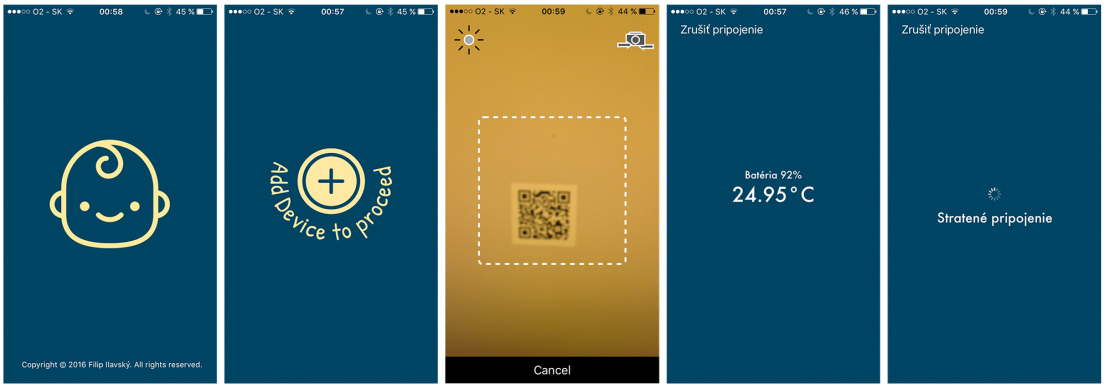
Cieľ aplikácie je zobraziť aktuálne dáta z periférie. To teda znamená nasledovnú postupnosť, ktorá musí byť splnená.

1. Získanie prístupu ku kamere.
2. Získanie prístupu k Bluetooth mobilného zariadenia.
3. Naskenovanie QR kódu.
4. Overenie obsahu QR kódu.
5. Uloženie informácií o perifériii z QR kódu (identifikácia periférie, AES kľúč).
6. BLE skenovanie.
7. Dekryptovanie dát z periférie.
8. Zobrazenie dát na obrazovke.

5.2 Popis implementácie

Podľa prvých dvoch požiadaviek potrebujeme použiť dve základné natívne knižnice. Driver CoreBluetooth na prístup a manipuláciu s Bluetooth mobilného zariadenia. Knižnicu AVFoundation na prístup ku kamere a niektorým jej definíciám.

Seknovanie QR kódu a dekryptovanie si vyžaduje implementáciu, no nakoľko cieľom projektu nie je implementácia skeneru QR kódu, ani implementácia kryptovacieho algoritmu,



Obr. 5.1: Screenshoty z iOS aplikácie

ale zabezpečeného protokolu, v projekte sme využili externé knižnice obsahujúce implementáciu týchto funkcionalít. Pre zjednodušenie importu týchto knižníc, ktoré nie sú natívne ponúkané v prostredí XCode, sme v projekte použili dependency manager CocoaPods [1]. Cez tento sme naimportovali dve externé knižnice: `QRCodeReader` a `CryptoSwift`. Každá z knižníc si rekurzívne vyžiada všetky potrebné knižnice(dependencies) sama, pri procese importovania cez CocoaPods. Pre použitie postačuje už iba zadefinovať v zdrojovom kóde import do súboru, v ktorom bude použitý. Nakoľko je aplikácia jednoduchá, obsahuje implementáciu v jednom súbore, ViewController.swift. Tento view controller teda obsahuje zdrojový kód vykonávaný v našej jedno-okennej aplikácii. Nastavenia projektu sa nachádzajú v projektovom súbore, grafické užívateľské rozhranie aplikácie sa nachádza primárne v Main.storyboard. Launch Screen.storyboard je štartovacia obrazovka, ktorá nedisponuje priradeným view controllerom.

Aplikácia je obmedzená na portrait mód zobrazenia, akákoľvek rotácia je zablokovaná. Obsahovo je aplikácia veľmi jednoduchá, nie je obmedzená na konkrétne zariadenie, čo vo výsledku znamená podporu už napríklad od iPhone 4S až po iPad Pro.

Najdôležitejšia časť implementácie sa týka konkrétne používania Bluetooth modulu v aplikácii. Ako bolo spomenuté, musíme si zadefinovať jeho použitie:

```
import CoreBluetooth
```

Následne pridáme delegát knižnice(porovnateľné s interface v C#) a zadefinujeme povinné funkcie.

```
//! Prijatie advertisment packetov
func centralManager(central: CBCentralManager,
    didDiscoverPeripheral peripheral: CBPeripheral,
    advertisementData: [String : AnyObject],
    RSSI: NSNumber)
{
    ProcessAdvertisementData(advertisementData)
}

//! Zmena stavu Bluetooth modulu
func centralManagerDidUpdateState(central: CBCentralManager) {
    if (central.state == CBCentralManagerState.PoweredOn){
        central.scanForPeripheralsWithServices(nil,
            options: nil)
    }
}
```

Funkcia `ProcessAdvertisementData` overuje, či prijaté dáta pochádzajú z očakavánej periférie a následne volá funkcie na dekrytovanie, a nakoniec spracovanie, čiže zobrazenie získaných dát(teplota, hodnota betérie). V prípade, že aplikácia čakala po naskenovaní QR kódu na prvý advertisement packet, aby si overila dostupnosť periférie, uloží si nastavenia(MAC adresu a AES kľúč) do perzistentnej aplikačnej pamäte (UserDefaults). Z tejto následne po štarte aplikácie tieto nastavenia načíta, aby nebolo nutné skenovať QR kód vždy po úplnom vypnutí procesu aplikácie.

```
//! Spracovanie beacon dat
func ProcessAdvertisementData(advertisementData : [String : AnyObject]) {
    // Neziadane data
    if ScanningType == ScanType.OFF {
        return
    }
    if UpdateGlobalsFromLoadedData(GetPeripheralData(advertisementData)) {
        // Pri parovani je potrebne ulozit nastavenia periferie a zmenit stav na
            meranie
        if ScanningType == ScanType.PAIRING {
            ScanningType = ScanType.DATA
            SavePairedPeripheral(mac, key: key)
        }
        ScanningSuccess()
    }
}
```

QR kód a dekryptovanie dát

QR kód je dvojrozmerný čiarový kód, ktorého obsahom sú dáta, väčšinou text. V tejto práci je použitý na distribúciu identifikácie periférie a zároveň kľúča na dekrytpovanie dát z periférie. Na overenie, či naskenovaný QR kód naozaj slúži na identifikáciu periférie, je na jeho začiatok vložený prefix `"sentinel_key="`. Ak sa tento na začiatku nenachádza, ide o QR kód slúžiaci na iný účel. Ak prefix obsahuje, môžeme pri ďalšiom rozkladaní načítaného QR kódu tento ignorovať.

Majme príklad QR kódu zobrazeného na obrázku 3.1 s obsahom `"sentinel_key=E4:D9:2D:86:7C:A8'a[tFD.S?i1<F*t#"`. Ten je rozdelený na MAC adresu `"E4:D9:2D:86:7C:A8"` a AES kľúč `"'a[tFD.S?i1<F*t#"`. Ich rozdelenie robíme na základe konštantnej dĺžky MAC adresy, ale aj 128 bitového AES kľúča (v textovej forme 16 bajtov, teda znakov). Teda po načítaní QR kódu aplikácia disponuje všetkými potrebnými informáciami na dekryptovanie dát vysielaných perifériou.

Nakoľko iOS od verzie 7.0 nepovoľuje aplikáciam prístup ku MAC adresám okolitých periférií, je nutné túto identifikáciu vykonať explicitne. Vieme, že jeden zakryptovaný vysielaný 16 bajtový blok dát z periférie obsahuje iba tri dátové bajty, obsahujúce teplotu a hodnotu batérie. Využijeme teda 8 bajtov na odosielanie vlastnej MAC adresy v tomto bloku pre ľahšiu identifikáciu mobilným zariadením. Ten musí najskôr dátový blok dekryptovať, a následne si overiť, či naozaj ide o požadovanú perifériu porovnaním očakávaných konštantných dát s hodnotou načítanej MAC adresy z QR kódu. Ak sa zhoduje, dáta z periférie budú uložené a zobrazené.

Dekryptovanie dát algoritmom AES, konkrétne verziou ECB, nám zaobstaráva knižnica CryptoSwift. Jej použitie je ukázané na funkcii DecryptData. V prípade akéhokoľvek zlyhania vraciame prázdne pole bajtov.

```
func DecryptData(encryptedDataString : [UInt8]) -> [UInt8] {
    if key == "" {
        return []
    }
    do {
        // Inicializacia a dekryptovanie vstupu funkcie
        return try AES(key: Array(key.utf8), blockMode: .ECB).decrypt(
            encryptedDataString)
    } catch {
    }
    return []
}
```

Testovanie iOS aplikácie

Prostredie XCode ponúka možnosť testovať aplikácie na vlastnom telefóne nahratím cez kábel, alebo pomocou testovacieho prostredia TestFlight. Toto prostredie je vhodné hlavne v prípade externých testerov aplikácie. Vývojár, ak chce takúto možnosť použiť, musí svoj projekt zaregistrovať na stránke https://itunesconnect.apple.com. Takáto registrácia, avšak nie je zadarmo. Jedným z dôvodov je možnosť následného predaja v AppStore.

V prípade prizívania osoby na testovanie, musí táto dostať pozvánku od vývojara, resp. jeho spoločnosti. Po prijatí tejto pozvánky na mailovú adresu, a následnom súhlase kliknutím na odkaz môže danú verziu aplikácie testovať, avšak maximálne po dobu 60 dní od prvého dňa aktivácie verzie na testovanie.

V prípade nahrávania aplikácie na AppStore, musí každú verziu, ktorá má byť na AppStore prístupná na stiahnutie, schváliť spoločnosť Apple v overovaciom procese. Spoločnosť týmto zabraňuje potenciálnym útočníkom akokoľvek uškodiť, a taktiež vo výsledku znižuje spotrebu batérie mobilných zariadení zákazníkov neprijímaním aplikácií s falošným využitím služieb v pozadí, a podobne.

Kapitola 6

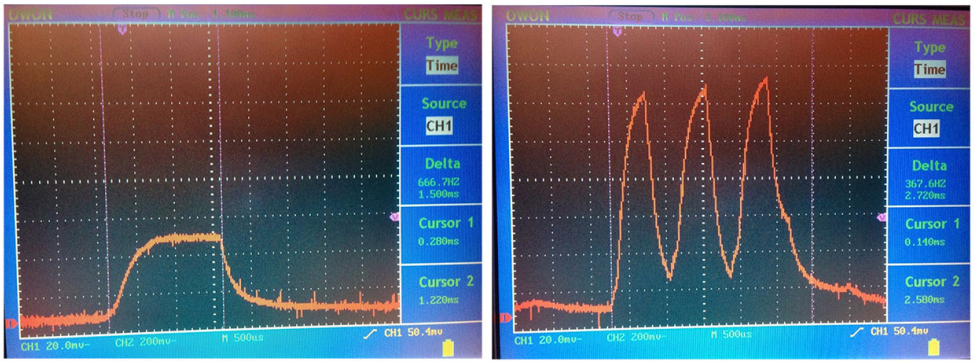
Merania, výpočty a testy

Pre overenie použiteľnosti a predpokladaných výpočtov sme spotrebu periférie odmerali pomocou osciloskopu. Taktiež nás zaujíma približná používateľná vzdialenosť a to najmä v byte, priechodnosť signálu stenami.

6.1 Meranie spotreby

Ako prvú sme odmerali spotrebu mimo merania a vysielania, od ktorej budeme pri meraní osciloskopom definovať teoretickú nulu. Tu nám na klasickom ampérmetri nameralo hodnotu 14μA.

Pri meraní osciloskopom sme postupne našli ideálny odpor pre meranie spotreby, čo je 10Ω. Nižšia hodnota(1Ω) nevyhovovala pre citlivosť osciloskopu, priebeh merania bol nečitateľný. Vyššia hodnota (100Ω) spôsobovala zlyhávanie chipu nRF5x a jeho následné resetovanie, takže znovu nevhodné na meranie. Následne nás zaujímali dve nasledujúce časti priebehu spotreby.



Obr. 6.1: Priebeh spotreby pri meraní(vľavo) a pri vysielaní(vpravo)

Vysielanie

Na obrázku je vidieť tri špičky spotreby. Toto je spôsobené vysielaním na všetky frekvencie BLE Advertising[1] vykonávaným za sebou, pre vyššiu pravdepodobnosť zachytenia klien-

[1]BLE používa pásmo 2402MHz - 2480MHz, pričom advertise iba konkrétne tri 2402MHz, 2426MHz, 2480 MHz

tom, ktorý počúva na jednom z nich. Z grafu si počítanie uľahčíme vizuálnou úpravou na jednoduchší priebeh, teda nie na tri špičky, ale na konštantný odber prúdu v rovnakom intervale času, čo vychádza približne 7.2mA počas 2.6ms. To znamená, pri jednom vysielaní spotrebujeme z batérie 1.872μAh.

Meranie

Spotreba pri meraní je ľahko odčítateľná na obrázku 6.1. Z priebehu je vidieť zobudenie procesora v nRF5x a všetkých ďalších komponentov uvedených v návrhu, a následné odmeranie a prechod do IDLE módu (ukončenie merania). Z grafu vieme vyčítať spotrebu 4.2mA po dobu 1.5ms, čo nám z batérie spotrebuje 630nAh.

Vo výsledku nám celková výdrž pri vysielaní každých 10 sekúnd vychádza na približne 3.2 mesiaca. Ak si meranie porovnáme s výpočtom iba na základe údajov od výrobcov zistíme, že spotreba zodpovedá predpokladanej vypočítanej hodnote.

V skutočnom prípade používania ako telesného teplomera, po implementácii akcelerometra na úplne uspatie periférie, by tento meral vo väčšine prípadov iba v stave choroby človeka. To zvyšuje výdrž batérie, nakoľko by periféria nevysielala a nemerala neustále.

Predstavme si situáciu až štvorčlennej rodiny, v ktorej by bolo nutné použit náš teplomer každé dva mesiace, napríklad kvôli chorobe. Ak by táto choroba trvala 5 dní pokaždé, za jeden cyklus choroby by periféria spotrebovala 6mAh z batérie (za predpokladu spotreby akcelerometra podľa výrobcu, teda 3μA). To by vo výsledku znamenalo približnú výdrž 400 dní, čo je takmer štvornásobok aktuálne meranej hodnoty.

6.2 Testovanie maximálnej vzdialenosti

Nakoľko pre získavanie meraní používame bezdrôtovú komunikáciu, je dôležité zistiť použiteľný dosah signálu periférie. V prípade otvoreného priestoru je dosah signálu blížiaci sa ku 30 metrom. V prípade detského teplomeru je veľmi nepravdepodobné, aby bol teplomer a klientské zariadenie v takejto vzdialenosti bez akejkoľvek prekážky, preto tento údaj nie je veľmi relevantný. Preto sme merali dosah signálu na klientskom zariadení[2] cez najbežnejšiu prekážku v domácnosti, stenu. Takýto údaj už dokážeme považovať za interiérový dosah. Merania priechodnosti cez steny sú veľmi približné, uvádzame priemer meraní v niekoľkých priestoroch, s rušením spotrebičov aj bez nich, najmä cez betónové/tehlové steny. Namerané výsledky prekvapili nasledovnými hodnotami

- Cez 1 stenu 11 metrov.
- Cez 2 steny 7 metrov.
- Cez 3 a viac stien signál neprešiel.

Napriek veľmi dobrým výsledkom dosahu v porovnaní s vynaloženou energiou nás všeobecný údaj o vzdialenosti utvrdzuje o nutnosti riešenia získavania údajov aj na zariadení mimo dosahu periférie. Toto meranie je čisto informatívne, nakoľko rôzny materiál obsiahnutý v stenách značne ovplyvňuje maximálny dosah signálu, údaj nie je možné použiť ako referenčný.

[2] Apple iPhone 5, iOS verzia 9.3.1

6.3 Cena

Najdrahšia časť výroby prototypov, ak nezapočítavame čas človeka na štúdium, testovanie a hodnotu stratenú pri chybnom návrhu a výrobe, je výroba DPS. V tabuľke 6.1 sú uvedené ceny najdrahších a najdôležitejších súčiastok, fyzicky použitých na prototype periférie.

Výroba DPS	120 €
N550M8CC	9.1 €
Držiak batérie	0.5 €
LMT70	2.5 €
LIS2DH	2 €
ADS1113	4 €
Celkom	138.1 €

Tabuľka 6.1: Cena prototypu periférie

Výroba prototypu je nákladná najmä kvôli výrobe DPS. Výrobca DPS musí stroje naprogramovať na výrobu, následne vyrobiť a ceny akýchkoľvek výrobkov na mieru v malých množstvách sú rádovo vyššie. Uvedená cena výroby jednej DPS nie je presná na objednávku 1ks, nakoľko sme dali vyrobiť niekoľko kusov naraz. Keďže išlo o objednávku na mieru, cena bola taktiež na mieru. Taktiež túto cenu nie je možné jednoducho podeliť počtom vyrobených kusov, výrobca udáva cenu celkovej objednávky, akoby bola objednávka na 1ks. Pri zníženej kvantite je veľmi pravdepodobná rovnaká cena od aktálnej.

V prípade pásovej výroby bez obalu, náklad pri 100 kusoch bude celkovo okolo 18.2 € . V tabuľke 6.2 je rozpis špecifických súčiastok a ich približných nákupných cien pri množste 100ks. Môžeme si všimnúť pribúdajúcej položky osadenia súčiastok, ktorá pri prototype nebola uvedená z dôvodu vlastného ručného osádzania.

Výroba DPS	1.5 €
Osadenie súčiastok	1.5 €
N550M8CC	8.5 €
Držiak batérie	0.2 €
LMT70	2 €
LIS2DH	1.5 €
ADS1113	3 €
Celkom	18.2 €

Tabuľka 6.2: Cena periférie pri kvantite 100ks, za 1ks

Ak by sme chceli uvažovať o výrobe prototypu predajného výrobku, je prototyp navrhnutý tak, aby sa ušetrila celková cena. Ako si môžeme všimnúť, periféria neobsahuje žiadne káble, pohyblivé časti ani vypínače. To znamená možnosť absolútnej automatizácie výroby a osádzania DPS, výroby obalu, a poskladanie obalu s perifériou. Jediná nutná požiadavka je možnosť výmeny batérie. Táto požiadavka je riešiteľná rozdeliteľnosťou obalu na dve časti tak, aby bolo možné batériu vymeniť. Je teda možné periférium pevne pripevniť na jednu stranu obalu teplomerom otočeným ku tejto časti, kde sa bude nastálo dotýkať pevne umiestneného, tepelne vodivého materiálu vedúceho až na povrch obalu. Pre zabránenie poškodenia vlhkosťou vložíme medzi časti obalu silikónové tesnenie. Tepelne vodivý materiál tepelného senzora bude technologicky riešený tak, aby neprepúšťal vlhkosť.

Kapitola 7

Ďalší vývoj

Práca sa skladá z mnohých častí a nadobúda vysokú komplexnosť. Preto nie všetky časti, ktoré by v reálnom využití periférie a aplikácie boli priam nutné, boli v práci aj implementované. V tejto kapitole sú najdôležitejšie z nich.

Background service iOS

Aplikácia bez notifikácií, najmä ak ide o detský teplomer stráca význam skutočného využitia. Avšak platforma iOS má silné bezpečnostné obmedzenia, jedným z nich je aj nutnosť presne špecifikovať pri BLE skenovaní konkrétnu službu na vyhľadávanie. Keďže tej neostalo miesto v adverisement packete, do budúcna bude nevyhnutné implementovať možnosť takzvaného Scan Response packetu, ktorý posiela dodatočných 31 bajtov dát na vyžiadanie. To má avšak za následok zvýšenie spotreby, nakoľko jedno vysielanie bude trvať približne dvojnásobok pôvodného času bez Scan Response. Následne je potrebné v definícii projektu zadefinovať požiadavku o povolenie aplikácie bežiacej v pozadí na základe používania BLE periférie a zadefinovať BLE identifikáciu služby. V tom prípade bude aplikácii v pozadí povolené získavanie BLE dát a následné spracovávanie a vykonávanie podľa potreby.

Exsitujú aj iné spôsoby, ktoré sú však všeobecne považované za hack a striktne odmietnuté v prípade požiadavky o uverejnenie aplikácie do AppStore(aplikácia iba pre vlastné použitie), čo úplne stráca zmysel pri komerčnej výrobe.

Obal periférie

Periféria je v aktuálnom stave v podstate holá, nie je ničím chránená. Pre reálne využitie ako komerčného produktu je nutné túto zabaliť do pevného púzdra. Púzdro by malo byť ideálne vodotesné, otvárateľné špecificky kvôli výmene batérie a musí mať tepelne dobre vodivú časť, prenášajúcu teplotu medzi telom a tepleným senzorom(dotýkať sa oboch zároveň). Prototyp takéhoto obalu je možné vďaka kompaktnosti vyrobiť aj v modernej 3D tlačiarni, čo však nebolo cieľom projektu a zároveň tlačiareň a/alebo tlačenie je finančne náročné.

Chyby v návrhu DPS

Počas testovania sa ukázala chyba návrhu, spočívajúca v umiestnení tepleného senzoru. Nakoľko je DPS veľmi kompaktná, jej minimálny rozmer je aktuálne limitovaný najmä veľkosťou batérie, resp. jej držiaku. Čo však znamená, ako je uvedené na návhu DPS 4.3, že tepelný senzor sa nachádza pod batériou. To spôsobuje nežiaducé ovplyvňovanie merania

teploty, spôsobené tepelnou kapacitou batérie. Pre rýchlejšie a presnejšie meranie je potrebné batériu umiestniť úplne mimo DPS, čo umožnuje ešte mierne zmenšenie DPS, alebo zväčšenie DPS a umiestnenie tepelného senzoru mimo oblasť pokrytú batériou. Táto zmena značne ovplyvní aj návrh obalu celej periférie.

Snímanie viacerých veličín

Ak uvažujeme rozšírenie snímania bez nutnosti zmeny DPS alebo pridávania ďalších senzorov, je možné pomocou akcelerometra overovať aj napríklad či dieťa neprestalo dýchať. Najväčší význam pre rozšírenie aj za cenu pridávania senzoru by bol senzor EKG. Periféria by bola použiteľná aj ako lekárská pomôcka pri diagnostikovaní pacientov so srdcovými problémami(prieskum trhu ukázal, že priamo porovnateľná periféria na trhu nie je).

Meranie mimo dosahu BLE

Jeden zo zásadných problémov je dosah signálu periférie. Najjednoduchšie riešenie pre iOS sa ponúka využitie push notifikácií a nadbytočného iOS zariadenia, nachádzajúceho sa v dosahu signálu periférie (napr. v rovnakej miestnosti). Takýto spôsob umožňuje značné rozšírenie aplikácie bez nutnosti akéhokoľvek zásahu do periférie alebo nutnosti výroby inej špecifickej periférie. Taktiež tento spôsob nevyžaduje implementáciu a udržiavanie serverovej časti riešenia. Avšak riešenie je limitujúce na zariadenia s operačným systémom iOS a OS X.

Rozšírenie na iné platformy

Dôležitým faktorom pri komerčnom výrobku je podpora klientskej aplikácie na viacerých platformách, nie len mobilných zariadení. Potenciálny zákazník, spokojný so svojím mobilným zariadením, nekúpi kvôli produktu iné mobilné zariadenie, a vôbec nepravdepodobne s iným operačným systémom. Preto by bolo potrebné implementovať klientskú aplikáciu minimálne na operačné systémy Android od verzie 4, rozširujúci sa Windows Phone 8 a vyššie, a nakoniec pre úzky trh aj BlackBerry. Mimo mobilné zariadenia by bola pozitívna aj implementácia na prenosných a stacionárnych počítačoch.

Ukladanie dát a analýza

Pri neustálom meraní je možné dáta ukladať a následne analyzovať. Teoretická využiteľnosť takéhoto ukladania dát by mohlo byť pri diagnostikovaní choroby lekárom. Pre bežného užívateľa by mohlo byť dôležité aj meranie teploty seba samého, napríklad počas spánku. Preto by mohla aplikácia ponúkať výstup vo forme grafu obsahujúceho čas a teplotu.

Kapitola 8

Záver

Cieľom tejto bakalárskej práce bolo navrhnúť a zostaviť hardvérovú periférium, komunikujúcu s mobilným zariadením, zabezpečenou bezdrôtovou komunikáciou. Na implementáciu tejto komunikácie sme použili Bluetooth Smart, čo vo výsledku zaručuje maximálnu možnú kompatibilitu so všetkými najpoužívanejšími klientskými operačnými systémami, a to nie len v rámci mobilných zariadení ako sú Android, iOS a Windows Phone, ale aj Microsoft Windows, MAC OS X, Linux a mnoho ďalších. Pre implementáciu klientskej aplikácie postačuje pre bežného programátora vedieť navyše iba samotný protokol, teda časť kryptovania a časť získavania konkrétnych dát z dekryptovaného bloku.

Testovaním výdrže vyrobeného prototypu periférie sme sa presvedčili o dostačujúcej výdrži batérie, pri stálom vysielaní 3.2 mesiaca, ako je uvedené v sekcii 6.1. Odchýlka v porovnaní s predbežnými výpočtami podľa údajov od výrobcov bola minimálna. Dokázali sme stabilitu softvéru a hardvéru periférie neustálym meraním teploty po dobu jedného mesiaca bez vypnutia periférie.

Informácie na riešenie tejto práce sme zbierali najmä z internetu, včetne elektronických kníh, z dôvodu nutnosti riešenia väčšiny problémov podľa produktových manuálov konkrétnych súčiastok od výrobcu, na ktoré neexistuje knižný výtlačok.

Ciele tejto bakalárskej práce boli splnené. Po štúdii technológií na bezdrôtovú komunikáciu, prihliadajúc na kompatibilitu a nízku spotrebu, sme vybrali Bluetooth Low Energy. Navrhli sme protokol pre zabezpečenú komunikáciu periférie s mobilným zariadením, dodržujúc požiadavky nízkej spotreby, sme použili kryptovací algoritmus AES podporovaný chipom nRF51422 (AES koprocesor), a dekryptovací kľúč sme pretransformovali do QR kódu, ktorý aplikácia mobilného zariadenia musí naskenovať pre získavanie dát z periférie. Taktiež sme navrhli a implementovali klientskú aplikáciu pre platformu iOS8 a vyššie, zobrazujúcu namerané dáta. V kapitole 7 sme si uviedli možné rozšírenia práce, jedným z nich je aj získavanie dát mimo dosahu signálu periférie. Monitorovací systém (periféria) je vo forme plne funkčného prototypu tepelného senzora, ktorý spĺňa požiadavky tejto práce.

Komplikácie pri návrhu a vývoji aplikácie boli úspešne vyriešené. Bola otestovaná a stanovená približná používateľná vzdialenosť periférie od prijímajúceho mobilného zariadenia. Taktiež bol zjednodušený spôsob inicializácie periférie v aplikácii skenovaním QR kódu a testovanie aplikácie vďaka použitiu aplikácie TestFlight.

Projekt má potenciál budúceho využitia aj napríklad ako diplomová práca. Taktiež po dokončení všetkých náležitostí pre komerčné použitie, je možné výsledok použiť ako produkt na predaj. Všetky minimálne náležitosti sú spomenuté postupne v práci, ale hlavne v kapitole 7.

Literatúra

[1] Apple Inc.: *The Swift Programming Language (Swift 2.2)*. 2014-06-02 [cit. 2016-05-07], [Online].
URL
https://itunes.apple.com/sk/book/swift-programming-language/id881256329

[2] Apple Inc.: *Using Swift with Cocoa and Objective-C (Swift 2.2)*. 2014-06-02 [cit. 2016-05-07], [Online].
URL https:
//itunes.apple.com/sk/book/using-swift-cocoa-objective/id888894773

[3] Broadcom Ltd.: *BCM2073X Product Matrix*. [cit. 2016-05-07], [Online].
URL https://www.broadcom.com/collateral/hs/2073X-HS100-R.pdf

[4] Dynastream Inovations Inc.: *TN5 ANT SoC Module Series Datasheet*. 2016-02-08 [cit. 2016-05-07], [Online].
URL https://www.thisisant.com/assets/resources/Datasheets/D00001598_N5_ANT_SoC_Module_Series_Datasheet_Rev1.9.pdf

[5] Erina Ferro, Francesco Potori: *Bluetooth and Wi-Fi wireless protocols: a survey and a comparison*. IEEE, 2005, ISBN 1536-1284.

[6] Kevin Townsend, Carles Cufí, Akiba, Robert Davidson: *Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking*. O'Reilly Media, 2014, ISBN 978-1-491-94951-1.

[7] Larry Ewing: *ECB: you're doing it wrong*. 2012-11-24 [cit. 2016-05-07], [Online].
URL http://www.rogdham.net/2012/11/24/ecb-youre-doing-it-wrong.en

[8] Nordic Semiconductor ASA: *S110 SoftDevice v7.0*. [Online].
URL https://www.nordicsemi.com/eng/Products/S110-SoftDevice-v7.0

[9] Nordic Semiconductor ASA: *nRF51422 Product Specification v3.1*. 2014-10 [cit. 2016-05-07], [Online].
URL https:
//www.nordicsemi.com/eng/nordic/download_resource/20360/9/78502290

[10] Nordic Semiconductor ASA: *S110 nRF51 SoftDevice Specification v2.0*. 2015-02 [cit. 2016-05-07], [Online].
URL https:
//www.nordicsemi.com/eng/nordic/download_resource/30088/15/19406577

[11] ST Electronics Ltd.: *MEMS digital output motion sensor: ultra low-power high performance 3-axes femto accelerometer*. 2011-11 [cit. 2016-05-07], [Online].
URL http://www2.st.com/resource/en/datasheet/lis2dh.pdf

[12] TE Connectivity Ltd.: *TSYS01 Digital Temperature Sensor*. 2013-08-04 [cit. 2016-05-07], [Online].
URL http://meas-spec.com/product/temperature/TSYS01.aspx#

[13] Texas Instruments Inc.: *Ultra-Small, Low-Power, 16-Bit ADC with Internal Reference (Rev. B)*. 2009-10-16 [cit. 2016-05-07], [Online].
URL http://www.ti.com/lit/ds/symlink/ads1113.pdf

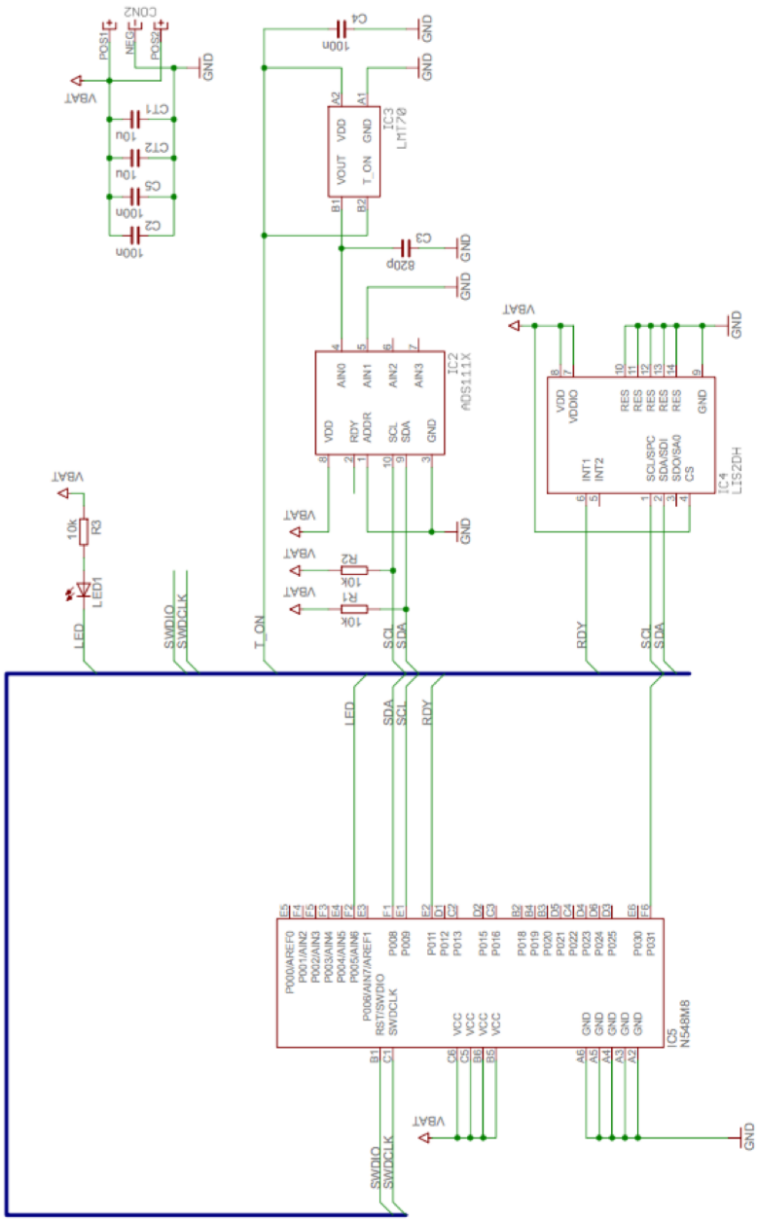
[14] Texas Instruments Inc.: *LMT70, LMT70A ±0.05°C Precision Analog Temperature Sensor, RTD and Precision NTC Thermistor IC (Rev. A)*. 2015-05-11 [cit. 2016-05-07], [Online].
URL http://www.ti.com/lit/ds/symlink/lmt70.pdf

Prílohy

Zoznam príloh

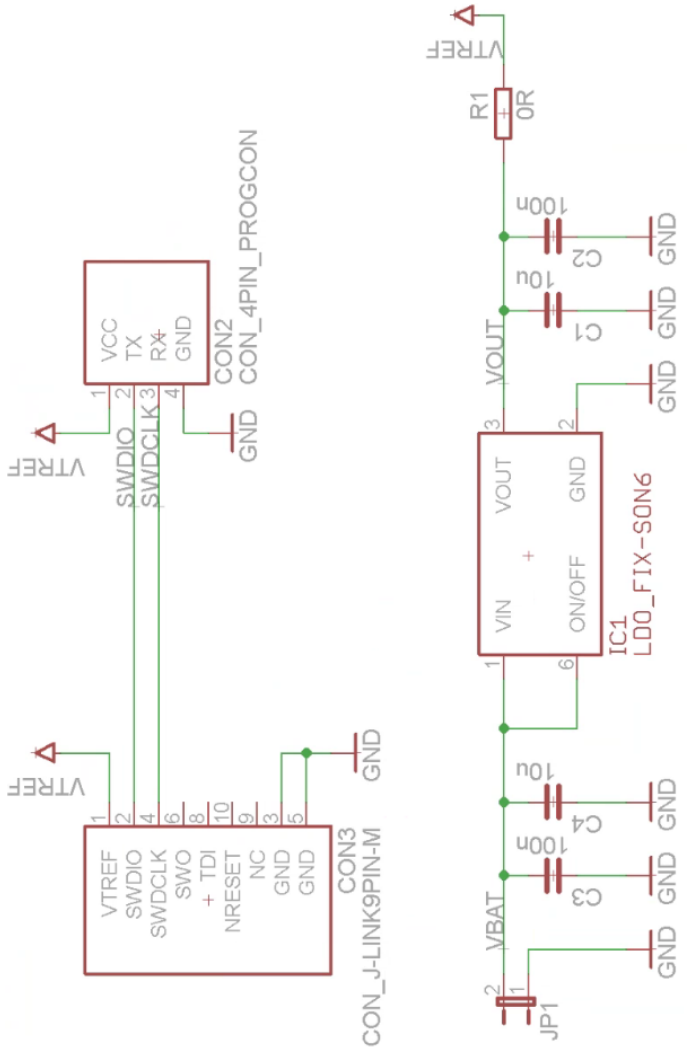
Príloha A

Schéma zapojenia periférie



Príloha B

Schéma zapojenia pomocného programátora



Príloha C

Obsah CD

Na priloženom CD sa nachádza nasledovný obsah:
`iOS8/` - projekt v prostredí XCode pre operačný systém iOS8 a vyššie
`Periferia/`
- `HW/` - schéma zapojenia periféria, DPS a zoznam súčiastok v návrhu
- `SW/` - implementácia softvéru periférie v programe Keil µVision 5

`Data/` - dáta používané pri vývoji
`DOC/` - zdrojové kódy k tomuto dokumentu
`Instalacie/` - inštalátor programu Keil µVision 5
`xilavs01.pdf` - tento dokument