



VŠB-TU Ostrava, Katedra informatiky, Fakulta elektrotechniky a informatiky

pro



ING corporation, spol. s r.o., Mánesova 1259, 739 11 Frýdlant nad Ostravicí

Návrh a realizace HW a SW pro ortoticko-protetické pomůcky s napojením na IoT

Zpráva

OP PIK, Inovační Voucher, Výzva II.

Tato dokumentace popisuje vývoj software – zařízení pro napojení inerciálního senzoru implementovatelného do zvolené ortoticko-protetické pomůcky na IoT (Internet of Things).

Tento výstup byl vytvořen v rámci projektu CZ.01.1.02/0.0/0.0/17_115/0012382, OP PIK, Inovační Voucher, Výzva II. na základě žádosti příjemce ING corporation, spol. s r.o.

Předmět služby/výstupy	
Předmět služby	Výstup služby
Vývoj software	Zpráva

Tuto dokumentaci zpracoval řešitelský tým VŠB-TU Ostrava, Katedra informatiky, Fakulta elektrotechniky a informatiky ve složení:

Ing. David Seidl, Ph.D.

Ing. Pavel Moravec, Ph.D.

Ing. Petr Olivka, Ph.D.

V Ostravě, dne 07. 11. 2018

Ing. David Seidl, Ph.D.

1. Cíl projektu

Cílem je vytvořit software, který bude z embedded procesoru obsluhovat inerciální senzor, naměřená data ukládat do paměti a odesílat pomocí IoT na server.

Je důležité dbát na nízkou spotřebu celého systému. Té bude docíleno uspáváním procesoru do velmi úsporného módu. Procesor se z tohoto módu bude automaticky budit každých 15 minut, aby detekoval, zda je přilba na hlavě dítěte. Informace o nošení přilby se bude ukládat do paměti. Dále se procesor z úsporného módu bude budit po přiložení magnetu. V případě tohoto probuzení se změří vzdálenost hlavičky od čidla na helmě a následně se naměřená data společně s daty o nošení helmy odešlou do sítě sigfox. Podrobnější informace o formátu odeslaných dat lze nalézt v dokumentu Protokol o posílání zpráv na SFX.

2. Zvolené technologie

2.1. IoT

2.1.1. Technologie IoT z pohledu Software

IoT je technologie využívána převážně pro zařízení, u kterých se vyžaduje dlouhý běh s nízkou spotřebou a schopností odesílat malé množství dat na velkou vzdálenost. Pro zdárný běh aplikace v řádů měsíců až let musí být dosaženo celkové spotřeby v jednotkách až setinách mA.

Nízké spotřeby je docíleno výběrem úsporných procesorů a ostatních komponent, které nabízí nízkou spotřebu za běhu a zároveň velmi úsporné stand-by režimy, při kterých je aktivní opravdu jen nutné minimum obvodů. Tyto obvody jsou následně buzeny jen pokud je to pro běh aplikace potřebné, jinak zůstávají ve stand-by režimu.

V dnešní době je drtivá většina procesorů používaných pro aplikace v IoT postavena na jádru procesoru ARM. Výhodou je jednotný instrukční set pro používané procesory a tím pádem velmi dobrá přenositelnost programů a knihoven mezi různými typy procesorů. Tímto se značně zjednodušuje a urychluje návrh a vývoj aplikací. Nevýhodami jednotné architektury jsou poněkud vyšší spotřeba energie a zvýšená komplikovanost daného systému.

2.1.2. Technologie IoT SigFox

Jako IoT síť byl zvolen Sigfox. Sigfox má vybudovanou infrastrukturu jak na straně procesoru, kde má pro komunikaci pro síť již vytvořené různé knihovny tak na straně sítě, kde umožňuje vytváření callbacků pomocí JSONu. Tato síť má neustále se rozšiřující pokrytí ve většině evropských zemí, má velmi dobrou odolnost vůči rušení a jiným vnějším vlivům a vlastní zabezpečení, které je na velmi dobré úrovni.

Na území české republiky je sigfox distribuován firmou Simplecell Networks a.s. Standard.

Sigfox využívá rádiovou technologii Ultra Narrow Band a operuje v nelicencovaných pásmech. Jednotlivé zprávy zpracovávající síť Sigfox jsou malé (12 Bytů uživatelských dat na zprávu + 1 Byte hlavička.) Sigfox Využívá 200kHz veřejně

dostupných a nelicencovaných pásem k výměně rádiových zpráv. Každá zpráva zabírá 100Hz a přenáší 100 nebo 600 bitů za sekundu dle regionu.

Vysílání mezi zařízeními a sítí není synchronizováno. Zařízení každou zprávu vysílá 3krát na 3 různých frekvencích. Stanice monitorují spektrum a hledají Ultra Narrow Band signály pro demodulaci.

Síť sigfox má omezené povolené množství odeslaných zpráv. Maximální povolený počet zpráv od zařízení na 6 za hodinu, tedy 144 zpráv denně. Tato zpráva dokáže pojmout maximálně 12B uživatelských dat. Při odesílání zpráv ze sítě na zařízení je povoleno odesílat 4 zprávy denně. Tato zpráva může obsahovat maximálně 8B dat.

2.1.3. Backend Sigfox

Sigfox má již vytvořené webové rozhraní pro práci se zprávami přijatými ze zařízení. Toto rozhraní je dostupné na adrese backend.sigfox.com.

Toto rozhraní umožňuje vytvářet a spravovat skupiny uživatelů a těm přidělovat práva k daným zařízením v síti sigfox. Lze zde prohlížet jednotlivá zařízení, jejich odeslané zprávy, vytvářet callbacky a kontrolovat případné chyby při odesílání zprávy ze zařízení na sigfox. Je umožněno také sledovat polohu zařízení, tato služba je však pouze orientačního charakteru, odchylka se totiž pohybuje v řádech desítek km.

Pro registraci nového zařízení je třeba provést následující kroky:

- a) Přihlásit se pod svým účtem na adrese backend.sigfox.com.
- b) V horní liště zvolit menu **Device**.
- c) V pravé horní části obrazovky zvolit možnost **New**.
- d) Vybrat požadovanou skupinu.
- e) Vyplnit informace o zařízení (ID/PAC, Name, Device Type, Product certificate, Základní stav je **Activable**, pokud je tato možnost zakázána všechny zprávy budou Zahozeny. Tuto možnost lze nastavit v editaci zařízení.)
- f) Potvrzení tlačítkem **Ok**.

Pro vytvoření vlastního callbacku je třeba provést následující kroky:

- a) Přihlásit se pod svým účtem na adrese backend.sigfox.com.
- b) V horní liště zvolit menu **Device Type**.
- c) Spravovat Device type a nastavit **Downlink mode** na **callback**.
- d) V **Callbacks** menu vytvořit **custom callback**.
- e) Nastavit následující možnosti:
 - 1.) Type: DATA/BIDIR
 - 2.) Channel: URL
 - 3.) Url pattern: Doména + umístění (<http://host/path>)
 - 4.) Zvolit požnost POST.
 - 5.) Content type: application/json

6.) Body: Data, která budou odeslána na zařízení, nastavte dle potřeby.

2.2. Processor

2.2.1. NCS36510 [1]

Tento procesor je dodáván firmou ON Semiconductor mimo jiné jako součást IoT Development kitu pro Sigfox. Tento procesor patří do rodiny ARM Cortex-M3. Tento procesor je navržen pro aplikace vyžadující minimální spotřebu a umožňuje použití Coma sleep módu, při kterém je spotřeba v jednotkách uA. Procesor nabízí 640kB FLASH paměti pro samotný program a 48kB RAM paměti pro uchovávání dat.

Procesor má následující módy z pohledu spotřeby:

Run

V tomto módu jsou všechny digitální systémy napájeny a spuštěny a procesor vykonává program.

Sleep

V tomto módu jsou napájeny všechny digitální systémy. Spuštěny jsou všechny až na hodiny procesoru. To způsobí zastavení vykonávání programu. Při zjištění přerušení procesor přejde do Run módu a provádí program začínaje posledním známým stavem.

Deep Sleep

Tento mód je podobný Sleep módu. Rozdíl je, že tento mód nenapájí FLASH paměti. Procesor nevykonává program, jelikož jsou jeho hodiny vypnuté. Při detekci přerušení jsou nejprve napájeny FLASH paměti a následně procesor přejde do Run módu.

Coma

V tomto módu jsou digitálním systémem napájeny pouze paměťové registry. Všechny registry si tedy uchovávají své hodnoty. Dochází k vypnutí rychlých oscilátorů. Pomalé oscilátory slouží jako hodiny pro procesor. Při detekci přerušení se začne napájet FLASH paměť a zbytek digitálních systémů. Procesor následně přejde do Run módu a začne vykonávat program za místem, kde vstoupil do tohoto módu.

Pro tento procesor vydala firma onsemi vlastní vývojové prostředí zvané IDK [2], které je založené na velmi rozšířeném prostředí eclipse. Tento procesor se poté programuje v jazyce C nebo C++, podle volby programátora. Procesor podporuje projekt mbed.

Tento procesor nemohl být použit pro finální aplikaci, neboť nejnížší dodávané množství v době vývoje bylo 2000 kusů, což je pro vývoj nepřijatelné.

2.2.2. LPC824 [3]

Druhým modelem procesoru, na kterém probíhal vývoj byl procesor LPC824 od firmy NXP patřící do rodiny ARM Cortex-M0+. Tento procesor nabízí usporný režim zvaný Power-down mode podobný Coma sleep módu u předchozího procesoru.

Spotřeba v tomto módu by se neměla přehoupnout přes 10uA. Tento procesor má 32kB flash paměti pro program a 8kB flash paměti pro data.

Procesor má následující módy z pohledu spotřeby:

Active

V tomto módu jsou všechny systémy procesoru napájeny a spuštěny. Procesor vykonává program.

Sleep

Při aktivaci Sleep módu jsou hodiny k procesoru zastaveny. Probuzení z tohoto režimu vyžaduje pouze aktivaci těchto hodin. V tomto módu je provádění programu zastaveno dokud nenastane přerušení nebo reset. Periférní funkce v tomto módu stále operují a mohou vygenerovat přerušení pro probuzení procesoru z tohoto módu.

Deep-sleep

V tomto módu je procesor uveden do Sleep módu popsaného výše a všechny hodiny jsou vypnuté s výjimkou IRC a watchdog oscilátoru. Všechny analogové bloky jsou vypnuty a FLASH paměť je ve standby módu. V tomto módu může program nastavit watchdog timer na probuzení dle vlastního načasování. Z tohoto módu lze procesor probudit pomocí resetu, pomocí digitálních pinů zvolených jako vstupy pro přerušení, pomocí časovače watchdog nebo pomocí přerušení přes rozhraní USART(pokud je nakonfigurován v synchronním slave módu), SPI a I2C(Ve slave módu).

Jakékoliv přerušení použité pro buzení z tohoto režimu musí být povoleno v jednom z SYSCON registrů pro umožnění buzení.

Tento mód značně šetří energii a umožňuje rychlé buzení.

Power-Down

V Power-down módu je procesor ve Sleep módu a všechny hodinové zdroje vyjma watchdog oscilátoru. Navíc jsou vypnuty všenebochny analogové bloky a FLASH paměti.

Power-down mód může být ukončen pomocí resetu, pomocí digitálních pinů zvolených jako vstupy pro přerušení, pomocí časovače watchdog nebo pomocí přerušení přes rozhraní USART(pokud je nakonfigurován v synchronním slave módu), SPI a I2C(Ve slave módu).

Jakékoliv přerušení použité pro buzení z tohoto režimu musí být povoleno v jednom z SYSCON registrů pro umožnění buzení.

Tento mód oproti módu Deep-sleep snižuje spotřebu energie za cenu delšího času buzení.

Deep power-down

V Deep power-down módu je energie vypnuta pro celý obvod vyjma WAKEUP pinu a časovače probuzení pokud je umožněn. Pro uchovávání dat v tomto módu jsou k dispozici 4 všeobecné registry.

Z tohoto režimu může být procesor buzen přes externí signál pomocí WAKEUP pinu nebo bez externího signálu pomocí time-out od self-wake-up timeru.

Procesoru může být znemožněno vstoupit do Deep power-down módu nastavením zamykacího bitu v PMU(Power management unit) bloku. Znemožnění tohoto módu zajistí, že watchdog poběží po celou dobu běhu programu.

Pro programování tohoto procesoru bylo využito IDE MCUXpresso, které sjednocuje práci na všech ARM procesorech vyrobených firmou NXP. Toto IDE běží na Eclipse. Programovací jazyk pro procesor je stejně jako u předchozího modelu C nebo C++ dle volby programátora, procesor podporuje projekt MBED

Tento procesor nemohl být použit pro finální aplikaci, jelikož 32kB flash paměti se prokázalo jako nedostatečné pro požadovaný program.

2.2.3. ATSAM21G18 [4]

Jako konečný typ procesoru byl zvolen procesor Arduino ATSAM21G18, který stejně jako předchozí procesor patří do rodiny Cortex-M0+. Flash paměť na tomto procesoru je pro tento program dostatečnou 256kB. Tento procesor má možnost Stand-by Sleep módu, ve kterém má spotřebu 40uA.

Procesor má následující módy z pohledu spotřeby:

Active

V tomto módu jsou všechny systémy procesoru napájeny a spuštěny. Procesor vykonává program.

Idle

Procesor je zastaven a společně s ním jsou zastaveny různé synchronní zdroje hodiny, které záleží na argumentu.

Standby

Zastaven je procesor a všechny zdroje hodinového signálu s výjimkou těch, u kterých je RUNSTBY bit nastaven na 1. Regulátor je nastaven na nízký výkon, před vstupem do tohoto módu je třeba vypnout většinu periférií aby nedošlo k přetížení regulátoru.

3. Implementace

3.1. Mbed OS [5]

Celý program běží pod projektem mbed. Mbed je open-source operační systém přímo tvořený pro implementaci IoT aplikací na mikroprocesorech. Jakékoliv

knihovny v rámci mbedu lze tedy použít ke komerčním účelům bez potřeby platit za jakékoliv licence.

Za pomoci tohoto projektu lze vyvíjet aplikace na Arm Cortex-M procesorech a součástí projektu jsou knihovny zajišťující práci s různými periferiemi, připojením a zabezpečení aplikací. Tento projekt zajišťuje jednoduchou přenositelnost již existujících programů mezi různými procesory a významně zjednodušuje a zrychluje vývoj programů nových.

Mbed nabízí vlastní online IDE, ale také možnost používat mbed OS v rámci jiných IDE, pokud si tuto možnost zvolí programátor. Programování samotného procesoru probíhá pomocí USB a umožňuje jednoduché a levné prototypování.

3.2. LDC1101 [6]

Jedná se o samotný inerciální senzor od výrobce Texas Instruments. Samotná práce se senzorem probíhá pomocí knihovny LDC1101.h z projektu mbed. Tento senzor komunikuje s procesorem pomocí rozhraní SPI. Ke komunikaci dochází konfigurací vnitřních registrů. Je třeba zajistit, že při konfiguraci těchto registrů je zařízení ve sleep módu. Probuzení obvodu nastane přivedením logické 1 na CSB. Poté probíhá čtení dat z registrů 0x21 a 0x22 a jejich uložení do paměti procesoru.

Pro finální produkt bude použit senzor LDC1612 [7] od stejného výrobce, který funguje obdobně. Rozdíl je v komunikačním rozhraní, kde LDC1612 místo SPI komunikuje pomocí I2C.

3.3. AX8052F143 [8]

Jedná se o obvod pro SFX anténu od firmy on Semiconductor. S tímto obvodem probíhá komunikace pomocí prostředí UART pomocí takzvaných AT příkazů. Samotný obvod nabízí módy Standby, Sleep a Deep Sleep. Standby mód je aktivní kdykoliv skončí samotné vysílání. Do Sleep a Deep Sleep módu lze zařízení dostat pomocí AT příkazů *AT\$P1* pro Sleep mód a *AT\$P2* pro Deep Sleep. Pro odeslání dat na sigfox bez potvrzení přijetí slouží příkaz *AT\$SF* a pro odeslání dat s potvrzením přijetí slouží příkaz *AT\$CB*.

Prakticky se toto zařízení programuje pomocí knihovny AX8052F143_SFX poskytnuté přímo společností on Semiconductor. Tato knihovna obsahuje metodu použitou v programu SendIntFrame, která slouží k odeslání pole 8-bitových celých čísel. Přímě v této metodě se nastavuje i čekání na odpověď ze sigfox serveru a doba, po kterou má zařízení na tuto odpověď čekat. Při úspěšném odeslání zprávy a obdržení ACK od serveru tato funkce vrací 0. V opačném případě vrací celé číslo s kódem chyby. Při komunikaci mohou nastat následující chyby:

<i>SFX_FAILURE</i>	(-1) Zprávu se nepodařilo odeslat.
<i>SFX_TIMEOUT</i>	(-2) Odpověď ze serveru přesáhla očekávanou dobu.
<i>SFX_MAX_SIZE</i>	(-3) Přesažení maximální velikosti přijímacího bufferu.

`SFX_UNSUPPORTED(-4)` Nepodporovaná délka posílaných informací.

3.4. Použité SW knihovny

3.4.1. Mbed.h

Tato knihovna je základem celé aplikace a jsou na ni postaveny všechny funkce a ostatní použité knihovny. Knihovna obsahuje nástroje pro práci se samotnými piny a také nástroje pro různá komunikační rozhraní jako například SPI nebo I2C.

3.4.2. Shields.h

Tato knihovna je knihovna dodávaná firmou on Semiconductor. Slouží pro ovládání modulu AX8052F143 a obsahuje jak patřičnou knihovnu AX8052F143.h tak i jiné pomocné knihovny pro práci s tímto obvodem. Knihovna funguje na základě mbedu. Pomocí této knihovny se opět vytváří instance třídy AX8052F143 a probíhá její inicializace. K odesílání dat je použita metoda `SendIntFrame`. Dále je zde funkce `powerMode`, která nastavuje příslušný úsporný režim a funkce `getVoltage`, pomocí které lze měřit napětí na obvodu.

3.4.3. LDC1101.h [9]

Tato knihovna je podřízená projektu mbed a slouží pro práci s čidlem LDC1101. Knihovna obsahuje implementaci všech funkcí čidla. Program využívá funkcí `get_RP_Data`, která vrací výsledek měření a kontrolní funkci `is_Oscillation_Error`. Senzor je zde reprezentován jako třída a pro fungování je potřeba vytvořit jeho instanci, do konstruktoru je potřeba zadat jednotlivé piny, kapacitu a odpor použitého senzoru. Třída pak automaticky provádí inicializaci, buzení a uspávání samotného čidla.

3.4.4. Sensor_Data.h

Tato knihovna byla vytvořena námi přímo pro tuto aplikaci. Obsahuje strukturu `Data` pro ukládání dat v domluveném formátu. Dále obsahuje třídu `Sensor_Data`, která kromě samotných dat obsahuje pomocné informace pro práci se sigfoxem a metody pro měření vzdálenosti a kontrolu nasazení přílby.

3.5. Struktura programu

Celý program se odehrává v nekonečné smyčce a po většinu času je procesor uspán do nejúspornějšího módu. Buzení procesoru probíhá dvěma různými způsoby.

První způsob je buzení pomocí časovače. Při tomto buzení se vždy pouze zkontroluje zda má dítě přílbu na sobě nebo nikoliv. Tato kontrola probíhá změřením vzdálenosti helmy od hlavy a následnou kontrolou proti referenční hodnotě. V tomto případě se naměřena hodnota neukládá. Po kontrole se inkrementuje čítač těchto měření a v případě, že je detekováno nenošení i čítač nenošených intervalů. Buzení časovačem bylo v rámci demonstračního programu nastaveno na každých 10 sekund. Na praktické nasazení je toto buzení nastaveno na každých 15 minut.

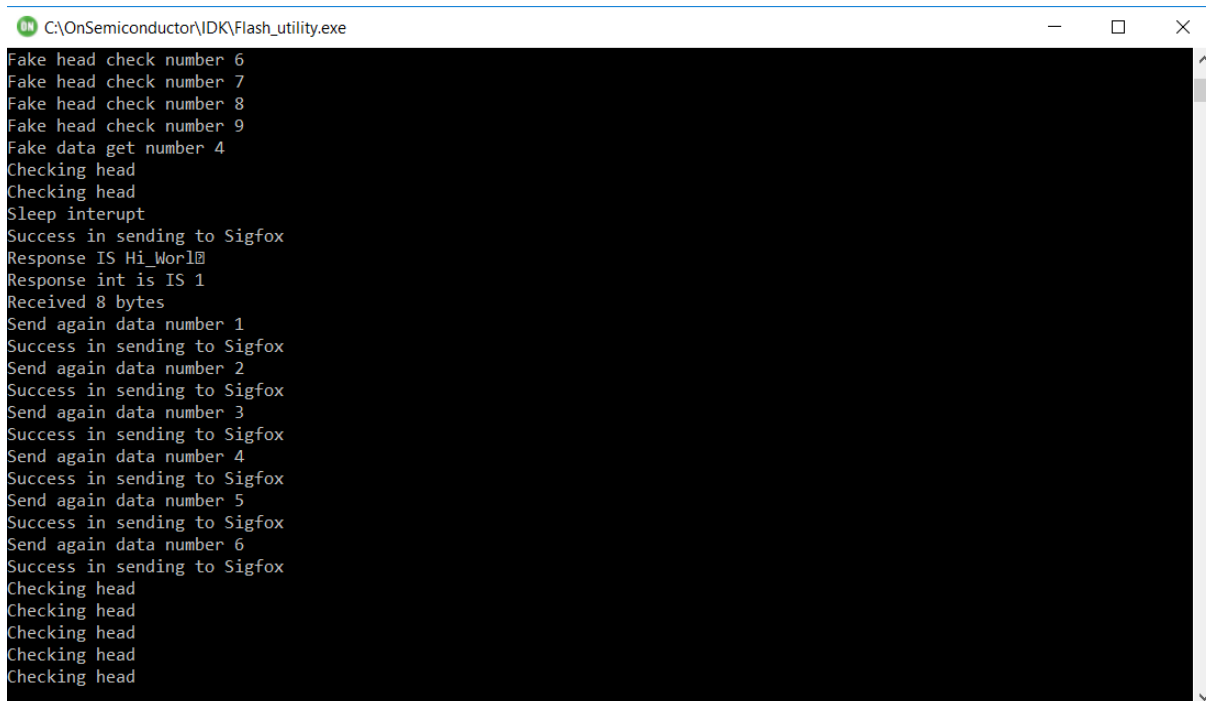
Druhý způsob buzení je pomocí vnějšího přerušení. V případě demonstračního programu k tomuto buzení došlo po stisknutí tlačítka SW5. V případě praktického nasazení k němu dojde po přiložení magnetu ke spínacímu čidlu. V případě probuzení tímto způsobem program provede kompletní měření a

uloží naměřenou vzdálenost. Jakmile je měření hotovo program odešle zaznamenaná data na sigfox ve formátu uvedeném v protokolu o odesílání zpráv na sigfox. Program následně čeká 60 sekund na ack signál, který potvrzuje úspěšné přijetí dat na sfx. Pokud je tento signál obdržen program pokračuje přechodem do úsporného režimu. Pokud program neobdrží potvrzení o doručení zprávy, je u dané zprávy nastaven příznak a při příštím odesílání dat dojde k jejímu opětovnému odeslání. Uživatel ví, že zpráva je odesílána díky LED signalizaci na přílbě. LED svítí pokud probíhá komunikace.

Data jsou uchovávána ve struktuře `Sensor_Data`. Tato struktura kromě dat samotných obsahuje i metody pro práci s daty. V této struktuře se ukládají informace o počtu kontrol nošení, počtu nenošení a samotná naměřená vzdálenost. Převod z obdržených dat na skutečnou vzdálenost bude probíhat až na straně serveru především z důvodu šetření baterie.

Demonstrační program navíc obsahuje funkci `fillFakeData`, která umožňuje simulovat neúspěšně odeslanou zprávu. Dále tento program vypisuje aktuální stav po sériové sběrnici. Příklad této komunikace je k vidění na Obr. 1.

Zobrazení samotných zpráv uložených u sigfoxu řeší backend sigfoxu. Příklad sekvence zpráv z demonstračního programu je k vidění na Obr.2.



```
C:\OnSemiconductor\IDK\Fish_utility.exe
Fake head check number 6
Fake head check number 7
Fake head check number 8
Fake head check number 9
Fake data get number 4
Checking head
Checking head
Sleep interrupt
Success in sending to Sigfox
Response IS Hi_World
Response int is IS 1
Received 8 bytes
Send again data number 1
Success in sending to Sigfox
Send again data number 2
Success in sending to Sigfox
Send again data number 3
Success in sending to Sigfox
Send again data number 4
Success in sending to Sigfox
Send again data number 5
Success in sending to Sigfox
Send again data number 6
Success in sending to Sigfox
Checking head
Checking head
Checking head
Checking head
Checking head
```

Obr.1. Výpis stavu programu pomocí sériového rozhraní do konzole.

Device 22AB7F - Messages

page 1

Time	Data / Decoding	Link quality	Callbacks	Location
2018-11-14 15:54:19	3400ca82c9db00 ASCR: 4...			
2018-11-14 15:53:56	1e00ee8aca5b00 ASCR: ...[
2018-11-14 15:53:48	140059d0ca5b00 ASCR: ...[
2018-11-14 15:53:34	0a00a9e4cadb00 ASCR:			
2018-11-14 15:52:57	3400ca82c9db00 ASCR: 4...			
2018-11-14 15:50:09	36003e23c15b00 ASCR: 6+P[
2018-11-14 15:49:27	34005a87c0db00 ASCR: 4-2...			

Obr.2. Reprezentace zpráv na serveru sigfox.

3.6. Příklad užití

- 1.) Helma je nasazena do provozu, automaticky probíhá pravidelná kontrola nošení.
- 2.) Zákazník přiloží magnet k senzoru, senzor aktivuje přerušení na procesoru.
- 3.) Procesor vykoná měření vzdálenosti, rozsvítí LED diodu a odešle data na sigfox.
- 4.) Procesor očekává potvrzení o přijetí dat ze sigfoxu.
- 5.) Procesor přijme potvrzení o doručení dat. Pokud není indikována žádná ztracená zpráva LED indikátor pro vysílání zhasne a procesor přejde zpátky do úsporného režimu. Pokud procesor indikuje chybějící zprávu dojde k jejímu znovu odeslání a očekávání na potvrzení. Pokud potvrzení nedorazí LED indikátor zhasne a zpráva bude odeslána při příštím měření. Proces pokračuje prvním krokem.

3.7. Budoucí modifikace

Nejočekávanější příklad by byl změna intervalu pro kontrolu nošení helmy. V tomto případě je ve funkci main potřeba změnit konstantu `TIME_BETWEEN_CHECKS_MS` na požadovanou hodnotu. Další možnou modifikací nezávislou na HW by byla změna formátu odchozích dat. V tomto případě je třeba nejdřív modifikovat strukturu Data z knihovny `Sensor_Data`, následně je třeba upravit v argumentu funkce `sendIntFrame` konstantu `SEND_DATA_SIZE`, jejíž definice se nachází na začátku programu.

Seznam použité literatury a knihoven:

- [1] – Low Power System-on-Chip For 2.4 GHz IEEE 802.15.4 Applications
<https://www.onsemi.com/pub/Collateral/NCS36510-D.PDF>
- [2] – IoT IDK User Guide <https://www.onsemi.com/pub/Collateral/AND9666-D.PDF>
- [3] – LPC82x <https://www.nxp.com/docs/en/data-sheet/LPC82X.pdf>
- [4] – SAM D21 Family <http://ww1.microchip.com/downloads/en/DeviceDoc/SAMD21-Family-DataSheet-DS40001882D.pdf>
- [5] – Mbed OS Documentation <https://os.mbed.com/docs/v5.10/>
- [6] – LDC1101 1.8-V High-Resolution, High-Speed Inductance-to-Digital Converter
<http://www.ti.com/lit/ds/symlink/ldc1101.pdf>
- [7] – LDC1612, LDC1614 Multi-Channel 28-Bit Inductance to Digital Converter (LDC) for Inductive Sensing <http://www.ti.com/lit/ds/symlink/ldc1614.pdf>
- [8] – Ultra-Low Power, AT Command / API Controlled, Sigfox Compliant Transceiver IC for Up-Link and Down-Link <https://www.onsemi.com/pub/Collateral/AX-SFEU-D.PDF>
- [9] – LDC1101.h <https://os.mbed.com/users/bobgiesberts/code/LDC1101/>