



Středoškolská technika 2015

Setkání a prezentace prací středoškolských studentů na ČVUT

USB komunikace mezi počítačem a AVR jednočipem

Jiří Štěpanovský

Gymnázium Třebíč
Masarykovo nám. 9/116, Třebíč

Obsah

Úvod	5
1. USB	5
1. Vývoj jednotlivých verzí	6
2. Komunikační protokol	6
3. Knihovna V-USB	7
4. Přístup k USB z počítačového programu	7
2. Projekty	7
1. Presentation Remote	8
1. Ovladač	8
2. Přijímač	9
2. USB Led Cube	9
1. Konstrukce USB Led Cube	10
2. Firmware	10
3. Program pro komunikaci s kostkou	11
3. PassKeychain	12
1. Klíčenka	12
2. Úprava knihovny V-USB	12
3. Software na ukládání hesel	13
4. Wrist Alarm	14
1. Konstrukce náramku	14
2. Firmware	15
3. Nastavení času a budíků	15
Shrnutí	16
Závěr	16
Seznam použitých zdrojů	17

Úvod

Cílem práce bylo otestovat možnosti komunikace počítače s 8 bitovým AVR jednočipem. Nejprve popíši USB, uvedu rozdíl mezi jednotlivými verzemi a zaměřím se na komunikační protokol této sběrnice. Dále uvedu jednotlivé knihovny, které jsem použil v mých projektech, a lehce nastíním jejich implementaci do kódu. Pak se již budu věnovat mým projektům využívajících USB.

Každý projekt je doprovázen jednoduchou hardwarovou konstrukcí, která se může připojit k USB portu počítače. Zapojení vždy obsahuje AVR jednočip zodpovědný za vlastní USB komunikaci. Projekty by měly představovat alespoň částečně užitečné příslušenství k počítači.

Presentation Remote je dálkové ovládání k prezentaci určené „pouze“ k navigaci mezi snímky. Přestože funkcionalita není rozsáhlá, dokáže toto zařízení odpoutat prezentujícího od počítače a prezentace tak působí daleko přirozeněji.

USB Led Cube je kostka ze 125 červených svítivých diod a vlastní paměti. Z počítače se dají na kostce přehrávat různé animace nebo lze animaci přímo uložit do vnitřní paměti kostky, a pouštět ji i bez počítače.

PassKeychain je speciální klíčenka, která se po připojení k počítači chová jako klávesnice a stiskem tlačítka „napíše“ jedno z hesel uložených v paměti. Hesla se samozřejmě dají v počítačovém programu upravovat, vytvářet, odstraňovat.

Wrist Alarm je minimalistický voděodolný náramek. Každé ráno nebo v jakýkoliv jiný nastavený čas tiše probudí vibracemi. Budí jen a pouze vibracemi.

1. USB

Nejlepším zdrojem informací je specifikace USB 3.1 vydaná organizací USB-IF v roce 2013 [1]. Název sběrnice vznikl jako zkratka z anglického názvu — Universal Serial Bus. V době vzniku bylo jejím hlavním cílem nahradit a sjednotit několik do té doby používaných externích sběrnic v počítačích sloužících k připojení klávesnic, myši a dalších externích zařízení. Netrvalo dlouho a USB se dostalo do téměř každého

Anotace

Cílem práce bylo seznámit se s možnostmi využití USB pro komunikaci mezi počítačem a AVR jednočipem na několika projektech. Mezi ně konkrétně patří Presentation Remote, dálkové ovládání prezentací, USB Led Cube, LED kostka ovládaná z počítače, PassKeychain, USB klíčenka s uloženými hesly, a Wrist Alarm, náramek s funkcí budíku.

Klíčová slova: USB; AVR jednočip; Presentation Remote; USB Led Cube; PassKeychain; Wrist Alarm

Anotation

The goal of this work was to become familiar with all the possibilities of using USB for a communication between a computer and an AVR microcontroller and make several projects. These include Presentation Remote (remote control for presentations), USB Led Cube (LED cube controlled from a computer), PassKeychain (USB keychain with stored passwords) and Wrist Alarm (wristband with alarm feature).

Keywords: USB; AVR microcontroller; Presentation Remote; USB Led Cube; PassKeychain; Wrist Alarm

počítače a postupně nahradilo všechny své předchůdce, mezi které patří mimo jiné i sériový a paralelní port. Zároveň USB vyřešilo dosavadní problém absence napájecích vodičů v tehdejších portech a sběrnicích.

1.1. Vývoj jednotlivých verzí

První verze 1.0 představená v roce 1996 stanovovala rychlosti Low Speed 1,5 Mb/s a Full Speed 12 Mb/s. Byl představen zcela nový protokol a nové USB konektory. Verze 1.1 nepřinesla nic převratného, pouze opravila některé nedostatky první verze.

V roce 2000 byla zveřejněna specifikace USB 2.0. Ta s sebou přinesla rychlost High Speed 480 Mb/s. Ke změně konektorů nedošlo a protokol byl pouze poupraven tak, aby bylo možné rozpoznat zařízení kompatibilní s USB 2.0 od zařízení kompatibilních jen s USB 1.1.

Následující verze USB 3.0 přidává rychlost SuperSpeed 5 Gb/s. Této rychlosti bylo možné docílit pouze radikální změnou komunikace. Již se nejedná o komunikaci half-duplexní, nýbrž o komunikaci full-duplexní. K 2 původním datovým vodičům přibyly další 4 a díky tomu mohou být data přenášena v jedné chvíli jak z master do slave tak naopak. USB 3.1 už přidalo pouze rychlost SuperSpeed+ 10 Gb/s, konektory i protokol zachovává.

Poslední aktualizací je nový konektor typu C. Jedná se pouze o nový konektor, neovlivňuje ani protokol, ani rychlost komunikace.

1.2. Komunikační protokol

Zařízení jsou uspořádána ve stromové struktuře. Základem je master označovaný jako host, na nějž jsou navázána zařízení slave. Hardwarově je většinou počet možných slave zařízení připojitelných k masteru dosti omezen, toto omezení se však dá obejít použitím rozbočovačů (hubs). Ty rozšiřují počet zařízení možných připojit k jednomu USB portu. Počet slave zařízení připojených k jednomu rozbočovači není omezen, ale celkově se může ve stromové struktuře nacházet pouze 5 pater (5 rozbočovačů zapojených sériově) a zároveň nesmí celkový počet slave zařízení překročit 127

(z důvodu adresování). Musí se taktéž dodržet podmínka, že nikdy nedojde k zacyklení struktury.

1.3. Knihovna V-USB

Knihovna V-USB [2] je knihovna napsaná v jazyce C určená pro jednočipy AVR. Zprostředkovává softwarovou implementaci Low Speed USB protokolu. Vyžaduje jen jedno hardwarové přerušení v procesoru (INT0) a jedno z následujících pracovních frekvencí procesoru: 12 Mhz, 15 Mhz, 16 MHz, 20 MHz, nebo frekvence 12,8 MHz či 16,5 MHz. Hardwarově je potřeba jen minimum pasivních součástek. Po kompilaci zabere v paměti jednočipu něco přes 1,2 KB, přičemž k práci vyžaduje přibližně 100 bytů paměti RAM včetně stack pointerů.

1.4. Přístup k USB z počítačového programu

V projektu USB Led Cube jednočip simuluje virtuální sériový port. Není tedy potřeba řešit přístup softwaru přímo k UBS, to řeší operační systém. Stačí nahlížet na zařízení jako na standartní sériový port a ten otevřít. Konkrétně jsem v prostředí Qt použil knihovnu QextSerialPort [3]. Jedná se o multiplatformní knihovnu, která umožňuje přístup k sériovým portům.

V projektu PassKeychain a Wrist Alarm již komunikace probíhá pomocí HID specifikace. V jazyce JAVA, ve kterém je software naprogramován, je nejlepší knihovnou pro komunikaci s HID zařízeními pravděpodobně knihovna hid4java [4]. Ta zprostředkovává pomocí JNA (Java Native Access) [5] nalinkování na multiplatformní knihovnu hidapi [6], napsanou v jazyce C.

2. Projekty

Cílem bylo vytvořit všemožná zařízení komunikující s počítačem přes USB. Hardwarovou stránku jsem se snažil vždy maximálně zjednodušit, a docílit tak co

nejnižší poruchovosti, přičemž funkcionalitu jsem volil s ohledem na praktičnost.

Základ všech zařízení tvoří AVR jednočip, který je připojený přímo k USB portu. Ve firmwaru se nachází více či méně upravená verze knihovny V-USB. Pro všechny projekty, kromě Presentation Remote, jsem naprogramoval ovládací software, který zajišťuje komunikaci a ovládání.

2.1. Presentation remote

Občas potřebuji odprezentovat své výsledky publiku, a k tomu se perfektně hodí využít prezentaci. Pokud však potřebuji přepnout na další snímek v prezentaci a zrovna se nacházím na opačné straně místnosti než počítač, publikum rychle zrtácí koncentraci a sleduje spíše mě, pobíhajícího po místnosti, nežli prezentaci. Tomu jsem chtěl předejít a postavil jsem si tak dálkové ovládání k prezentaci — Presentation Remote.

Schéma a zdrojové kódy pod licenci MIT najdete na stránkách projektu [7].

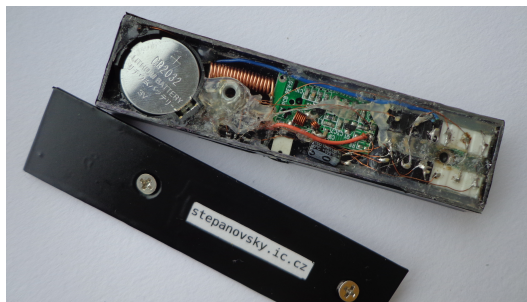
2.1.1. Ovladač

Ovladač obsahuje 4 tlačítka. Stisknutí kteréhokoliv z nich způsobí zapnutí jednočipu Attiny13, který odešle informaci o právě stisknutém tlačítku přes 433 Mhz vysílací modul. Obvod odebírá proud pouze, pokud je stisknuto alespoň jedno tlačítko, čímž je zajištěna dlouhá životnost baterie CR2032 uvnitř. Zapojení je umístěno do plastové krabičky, kde je upevněno tavným lepidlem.

Firmware v jednočipu Attiny13 obsahuje jen několik málo řádků. Sleduje stav tlačítek a pokud je alespoň jedno stisknuté, posílá opakovaně zakódovanou informaci o stavu všech tlačítek do 433 Mhz vysílacího modulu.



Obrázek 1 — Ovladač zvenku



Obrázek 2 — Ovladač uvnitř

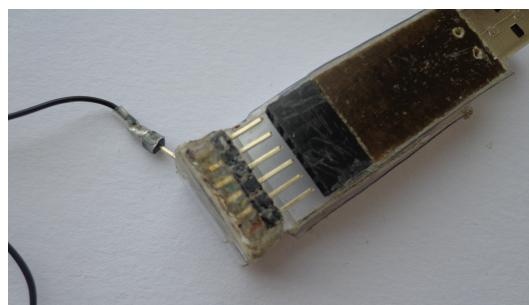
2.1.2. Přijímač

Informaci o stisknutém tlačítku přijme 433 Mhz přijímací modul a zpracuje ji jednočip Attiny85, který je připojen k USB portu.

Firmware obsahuje knihovnu V-USB nakonfigurovanou do módu HID klávesnice (pomocí USB deskriptoru). Když prostřednictvím přijímacího modulu přijme validní zprávu o stavu tlačítkách a pokud je alespoň jedno z nich stisklé, vytvoří USB report zprávu v níž informuje počítač o stisku klávesy na virtuální HID klávesnici. Klávesy jsou napevno naprogramované v firmwaru, přičemž horním tlačítkům na ovladači odpovídají klávesy šipka vpravo a šipka vlevo a dolním tlačítkům na klávesnici odpovídají klávesy F5 (spuštění prezentace) a B (pozastavení prezentace).



Obrázek 3 — Části přijímače s anténou



Obrázek 4 — Propojení částí přijímače

2.2. USB Led Cube

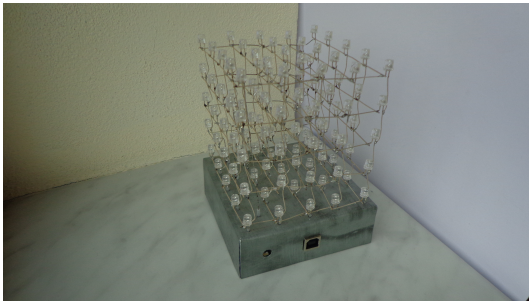
3D svítivé kostky se mně vždy líbily. U většiny mi ale chyběla jednoduchá cesta, jak vytvářet animace. Většinou se natvrdo naprogramují do firmwaru, přičemž jejich tvorbu doprovází problémy s adresováním jednotlivých svítivých bodů kostky a častu dochází k menším chybám v animacích (svítí, co svítit nemá, nebo naopak).

Rozhodl jsem se tedy postavit si kostku vlastní a softwarově daleko lépe vyřešit tvorbu animací. Kostka se bude připojovat k počítači, kde se budou ve speciálním programku tvořit animace a odkud se budou moct uložit do paměti kostky.

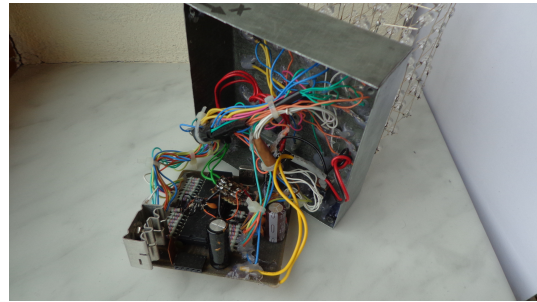
Schéma a zdrojové kódy pod licenci MIT najdete na stránkách projektu [8].

2.2.1. Konstrukce USB Led Cube

Rozhodl jsem se pro kostku velikosti 5x5x5 bodů. Pro jednoduchost jsem zvolil pouze jednobarevnou variantu — červenou. Uvnitř se nachází jednočip Atmega162 a pro ukládání animací jsem zvolil 2 kB I2C EEPROM paměť 24LC16. Elektroniku jsem umístil do plechového boxu pod svítivé diody.



Obrázek 5 — USB Led Cube



Obrázek 6 — Zapojení uvnitř

2.2.2. Firmware

Firmware má tři hlavní části. První se stará o zobrazení obrazových dat na svítivých diodách kostky, druhá obsluhuje USB komunikaci a poslední zajišťuje komunikaci s vnitřní I2C pamětí.

Obrazová data jsou v paměti jednočipu reprezentována 25 byty, přičemž je použito pouze prvních 5 bitů, zbylé 3 jsou nevyužity. Každý byte dat reprezentuje jeden sloupec a každý bit v bytu reprezentuje jednu svítivou diodu v sloupci. Toto rozdělení s sebou přináší nesporné výhody. V jednu chvíli totiž svítí pouze jedna z pěti vrstev kostky a tak se stav diody ve vrstvě jednoduše odvodí od vrstvě odpovídajícího bitu v sloupci odpovídajícím bytu.

Ve firmwaru je opět implementována knihovna V-USB. Tentokrát je lehce upravena tak, aby počítač identifikoval kostku jako virtuální sériový port. Komunikace mezi kostkou a počítačem tedy probíhá po virtuálním sériovém portu přes USB. Firmware identifikuje dva typy zpráv. Jeden typ nastavuje obrazová data, druhý typ ukládá obrazová data přímo do vnitřní paměti. Pokud nejsou delší dobu přijmuta žádná data, kostka automaticky začne přehrávat animaci uloženou ve vnitřní paměti.

Jednočip Atmega162 nedisponuje I2C sběrnici a tak se komunikace s I2C pamětí

musí řešit přes softwarově emulovanou I2C sběrnici. Emulace I2C sběrnice není nikterak náročná, avšak jelikož jsem našel již hotovou knihovnu právě pro tento účel, přišlo mi zbytečné ji znova programovat a použil jsem tak kód z www.eXtremeElectronics.co.in [9].

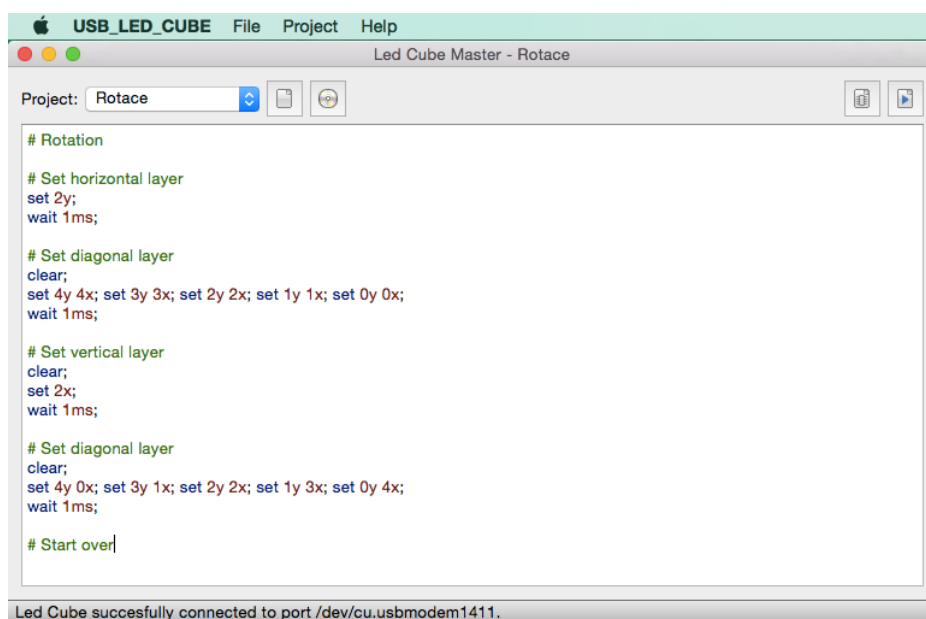
2.2.3. Program pro komunikaci s kostkou

Program jsem chtěl vyvíjet pro všechny tři nejrozšířenější operační systémy, a tak jsem se rozhodl pro použití prostředí Qt a jazyka C++. Pro komunikaci s USB Led Cube, která se k počítači připojí jako virtuální seriový port, jsem použil knihovnu QextSerialPort [3].

V aplikaci člověk může vytvářet a ukládat animace, a pokud je kostka připojená k počítači, může animaci v aktuálním čase přehrát na kostce nebo uložit do vnitřní paměti kostky a přehrát kdykoliv, k tomu jsou využívány dva typy zpráv popsané výše.

Vytváření animací je vyřešeno naprosto originálně. Animace se totiž vytváří pomocí textových příkazů, což je podle mě nejlepší způsob zadávání obrazových dat. Pracuje se celkově se 3 příkazy — set, clear a wait. Pomocí parametrů se pak určuje, které body na kostce se mají rozsvítit, zhasnout nebo jak dlouho má program počkat.

Animaci pak program před odesláním do kostky nejprve „zkompiluje“ do obrazových dat, případně nahlásí chybu, pokud příkazy nejsou správně zadány.



```
# Rotation
# Set horizontal layer
set 2y;
wait 1ms;

# Set diagonal layer
clear;
set 4y 4x; set 3y 3x; set 2y 2x; set 1y 1x; set 0y 0x;
wait 1ms;

# Set vertical layer
clear;
set 2x;
wait 1ms;

# Set diagonal layer
clear;
set 4y 0x; set 3y 1x; set 2y 2x; set 1y 3x; set 0y 4x;
wait 1ms;

# Start over|
```

Led Cube successfully connected to port /dev/cu.usbmodem1411.

Obrázek 7 — USB Led Cube software

2.3. PassKeychain

Do počítače zadávám hesla takřka neustále a pamatovat si každé je pro mě nemožné. Tak jsem si vytvořil program pro ukládání hesel a díky znalosti USB jsem jej rozšířil o klíčenku. S klíčenkou mi stačí stisknout tlačítko a heslo se mi automaticky napíše.

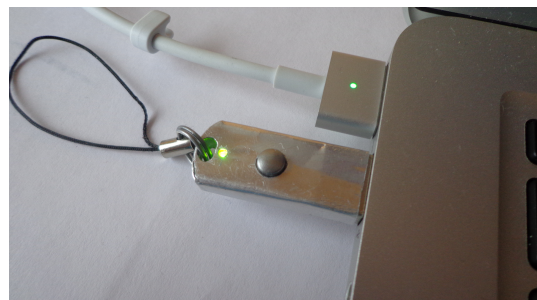
Schéma a zdrojové kódy pod licencí MIT najdete na stránkách projektu [10].

2.3.1. Klíčenka

Klíčenku tvoří jen jednočip Attiny85, svítivá dioda a tlačítko. Klíčenka se připojuje k počítači přes USB, přičemž se připojí buďto jako HID klávesnice nebo jako HID specifické zařízení, mezi režimy lze přepínat dlouhým stiskem tlačítka. Pokud je klíčenka v režimu HID specifického zařízení, může se přes počítačový program do vnitřní paměti klíčenky nahrát různá hesla. Po přepnutí do režimu HID klávesnice po krátkém stisku tlačítka pak klíčenka bude simulovat stisk kláves na virtuální klávesnici tak, že napíše první heslo uložené v paměti. Pokud se tlačítko zmáčkne rychle dvakrát po sobě, napíše se druhé heslo v pořadí. A tak až do teoreticky možných 127 hesel, neboli 127 stisků tlačítka.



Obrázek 8 — PassKeychain



Obrázek 9 — PassKeychain v počítači

2.3.2. Úprava knihovny V-USB

Aby bylo možné připojit zařízení v obou režimech (klávesnice a specifické zařízení), bylo nutné poměrně dost upravit knihovnu V-USB. Ta je totiž schopna

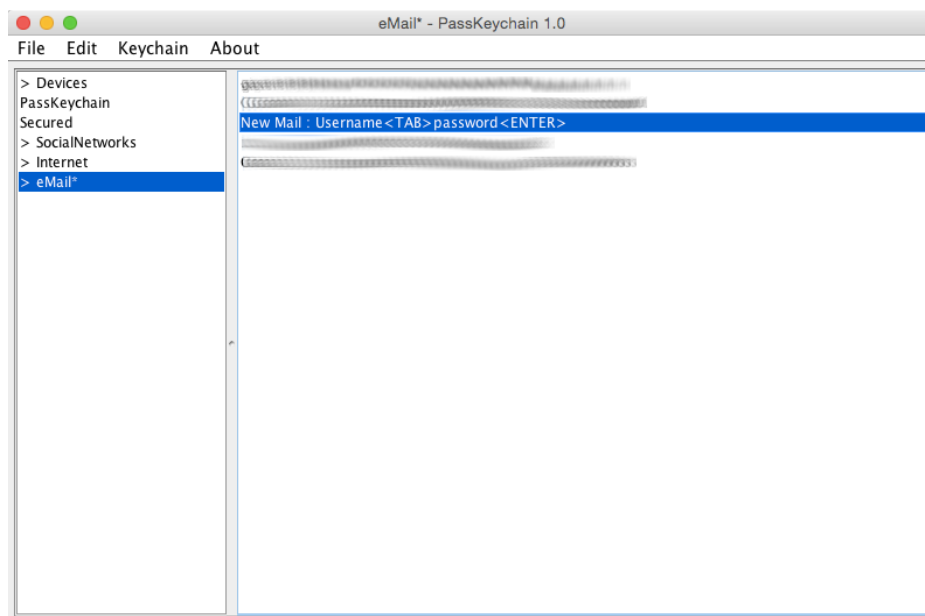
pracovat pouze v jednom režimu. Ve firmwaru jsem to nakonec vyřešil tak, že jsem spojil do sebe dvě na sobě nezávislé knihovny V-USB. Manuálně pak ve firmwaru zapínám zrovna aktuální režim a vypínám ten neaktivní.

2.3.3. Software na ukládání hesel

Program jsem napsal v jazyce JAVA, což přináší s sebou výhodu podpory všech rozšířených operačních systému bez nutnosti opětovné kompilace, jako tomu je v případě Qt. Pro komunikaci s klíčenkou v režimu HID specific device jsem využil knihovnu hid4java [4].

V aplikaci může člověk vytvářet nové hesla, může upravovat již vytvořené nebo může hesla uspořádat do seznamů. Tyto seznamy lze ukládat v počítači jak v nešifrované podobě tak v šifrované podobě s využitím šifrování AES.

Nebo lze seznam nahrát přímo do klíčenky, kde k nim má člověk přístup jen díky stisku tlačítka.



Obrázek 10 — PassKeychain software

2.4. Wrist Alarm

Nositelná elektronika v poslední době vzrůstá na popularitě. Zvláště pojem inteligantní hodinky slýchávám čím dál tím častěji. Jejich účel jsem však zatím příliš nepochopil. Oproti chytrým telefonům nenacházím žádné pozitivum — malý display, ztížené ovládání... Přece se mi jedna funkce nositelné elektroniky velmi zamlouvá.

Na zvuk budíku jsem již alergický, a tak jsem se nechal inspirovat moderní nositelnou elektronikou a vyrobil jsem si budící náramek. Budí pomocí vibrační tiše a klidně.

Schéma a zdrojové kódy pod licenci MIT najdete na stránkách projektu [11].

2.4.1. Konstrukce náramku

Náramek obsahuje relativně obrovský akumulátor li-pol s kapacitou 160 mAh na který dokáže běžet až 1 měsíc! Attiny85 používá k měření času obvod RTC. Náramek má jedno jediné tlačítko, jímž se zjišťuje stav akumulátoru a vypíná aktuálně vibrující budík.

Konstrukce je volena s ohledem na kruhovitý tvar náramku. Deska plošných spojů je rozdělena na několik částí a je pospojována několika vodiči, čímž je zajištěna ohebnost, pružnost a v neposlední řadě i odolnost zařízení.



Obrázek 11 — Wrist Alarm



Obrázek 12 — Konstrukce náramku

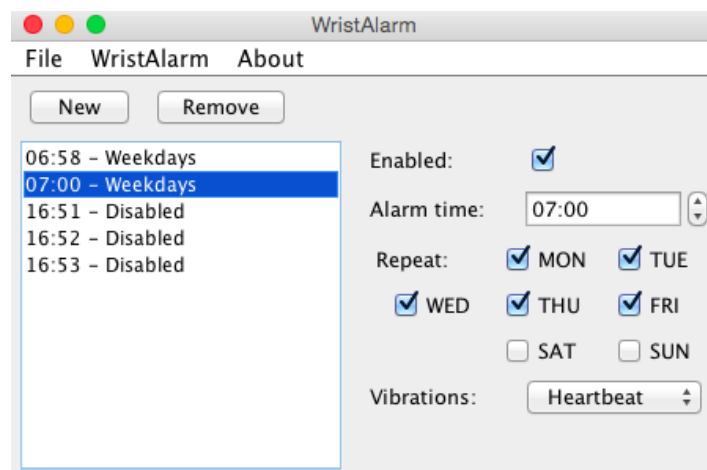
2.4.2. Firmware

Firmware obsahuje opět knihovnu V-USB, tentokrát napevno v módu USB HID specific device. Pokud náramek je připojen k počítači, očekává nastavování času případně časů budíků. Jestliže počítač připojen není, jednočip dokola porovnává aktuální čas RTC a budíků uložených v jeho EEPROM paměti. Po většinu času však jednočip stráví v power-down módu, převážně kvůli šetření akumulátoru.

2.4.3. Nastavení času a budíků

Aplikace je napsána v multiplatformním jazyce JAVA. Pro komunikaci s náramkem jsem použil již ověřenou knihovnu z předchozího projektu — hid4java [4].

Program umožňuje seřízení času v náramku podle času v počítači a dovoluje nastavit časy budíků, jejich opakování, způsob vibrace. Budíky jsou uloženy pouze a jenom v paměti náramku, pokud tedy člověk připojí svůj náramek k úplně cizímu počítači, vždy uvidí svoje časy budíků. Ty se mu nikdy neztratí.



Obrázek 13 — Wrist Alarm software

Shrnutí

Vytvořil jsem několik zařízení, která implementují USB v 8 bitových AVR jednočipech a komunikují s počítačem. Zažil jsem si programování USB komunikace z obou stran, jak ze strany master (počítače), tak ze strany slave (AVR jednočipu).

Závěr

S využitím USB lze vytvořit množství užitečného příslušenství k počítači. Zařízení jako PassKeychain nebo Wrist Alarm používám každodenně a na trhu mi podobné alternativy chybí.

Projekty nebyly po hardwarové stránce nikterak náročné, avšak složitost firmwaru a softwaru to vynahradila.

Seznam použitých zdrojů

- [1] USB-IF. USB 3.1 Specification. 2013. Dostupné z:
<http://www.usb.org/developers/docs/>

- [2] Objective Development. V-USB. 2012. Dostupné z:
<http://www.obdev.at/products/vusb/index.html>

- [3] Wayne Roth. Qt knihovna QextSerialPort. 2011. Dostupné z:
<https://code.google.com/p/qextserialport/>

- [4] Gary Rowe. Knihovna hid4java. 2014. Dostupné z:
<https://github.com/gary-rowe/hid4java>

- [5] Timothy Wall. Java Native Access. 2015. Dostupné z:
<https://github.com/twall/jna>

- [6] Alan Ott. Knihovna hidapi. 2014. Dostupné z:
<https://github.com/signal11/hidapi>

- [7] Jiří Štěpanovský. Presentation Remote. 2013. Dostupné z:
<http://jiristepanovsky.cz/projects/presenter/>

- [8] Jiří Štěpanovský. USB Led Cube. 2014. Dostupné z:
<http://jiristepanovsky.cz/projects/ledcube/>

- [9] Avinash Gupta. Software I2C library. 2011. Dostupné z:
<http://extremeelectronics.co.in/avr-tutorials/software-i2c-library-for-avr-mcus/>

- [10] Jiří Štěpanovský. PassKeychain. 2014. Dostupné z:
<http://jiristepanovsky.cz/projects/passkeychain/>
- [11] Jiří Štěpanovský. Wrist Alarm. 2015. Dostupné z:
<http://jiristepanovsky.cz/projects/wristalarm/>