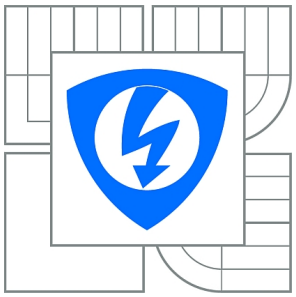
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

INTERFACE PRO DIAGNOSTICKOU SBĚRNICI (SMBUS/I2C) PRO AKUMULÁTORY

INTERFACE FOR DIAGNOSTIC BUSBAR (SMBUS/12C) FOR ACCUMULATORS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

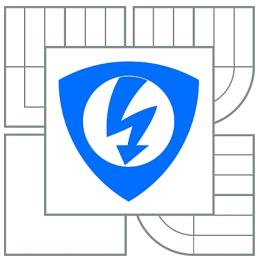
AUTOR PRÁCE — Bc. ZOLTÁN PINTÉR

AUTHOR

VEDOUCÍ PRÁCE — Ing. ONDŘEJ PAVELKA

SUPERVISOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Zoltán Pintér	***ID:***	147761
Ročník:	2	***Akademický rok:***	2012/2013

NÁZEV TÉMATU:

Interface pro diagnostickou sběrnici (SMBus/I2C) pro akumulátory

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte HW část zařízení a příslušný obslužný SW pro komunikaci s řídícími obvody notebookových a jiných "inteligentních" baterií vybavených diagnostickou sběrnicí Smart Battery. Správnou funkci monitoru ověřte čtením údajů na několika notebookových bateriích osazených různými komunikačními čipy. Vytvořte podporu pro existující čipy.

DOPORUČENÁ LITERATURA:

[1] Mann, B. : C pro mikrokontroléry, BEN, 2003. ISBN 80-7300-077-6
[2] Kainka, B.: Měření, řízení a regulace pomocí PC, BEN, 2003, ISBN 80-7300-089-X

Termín zadání:	11.2.2013	***Termín odevzdání:***	29.5.2013

Vedoucí práce: Ing. Ondřej Pavelka
Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Abstrakt

PINTÉR, Z. Interface pro diagnostickou sběrnici (SMBus/I2C) pro akumulátory. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 53 s. Vedoucí diplomové práce Ing. Ondřej Pavelka.

Diplomová práca sa v prvej kapitole venuje analýze problémov pri návrhu zariadenia, ako sú jednotlivé kroky postupu návrhu zariadenia a výber najvhodnejšej možnosti. Zaoberá sa analýzou zadania a požiadaviek na zariadenie, analýzou riadiacich čipov akumulátorov s ktorými má zariadenie komunikovať, analýzou komunikačného protokolu I2C/SMBus, pomocou ktorého zariadenia komunikuje s riadiacimi čipmi akumulátorov a možnosťami komunikácie zariadenia s počítačom. V druhej kapitole sa diplomová práca venuje výberu komunikácie cez USB zbernicu a popisu tejto komunikácie. V ďalšom kroku sa venuje výberu programovateľného mikroprocesora Atmel AT90USB1287, popisuje jeho zapojenie, jeho možnosti a popisuje spôsob zapojenia USB rozhrania, spôsob zapojenia I2C/SMBus rozhrania a jeho ochranu. V poslednej časti tejto kapitoly sa práca venuje programovaniu mikroprocesora Atmel, konkrétne popisuje akým spôsobom sa programuje I2C/SMBus komunikácia, USB komunikácia a akým spôsobom sa program nahráva do mikroprocesora. V tretej kapitole sa práca venuje návrhu samotného zariadenia, teda prvkom ako je schéma zapojenia jednotlivých častí a koncept spôsob fungovania zariadenia, ktorý popisuje spôsob komunikácie so zariadením. Štvrtá kapitola sa zaoberá vývojom samotného zariadenia. Popisuje časti a výsledné zapojenie zariadenia a popisuje spôsob fungovania jednotlivých častí výsledného programu mikroprocesora, ako sú komunikácia s SMBus zariadením a komunikácia cez USB zbernicu. V piatej kapitole sa práca venuje vývoju komunikačnej aplikácie v jazyku Java. V tejto časti sú popísané jednotlivé prvky aplikácie ako komunikácia s USB zariadením, ukladanie údajov a ich otvorenie pre prehliadanie. V poslednej kapitole práca prezentuje výsledné zariadenie a výslednú komunikačnú aplikáciu.

Kľúčové slová: I2C/SMBus, USB, Atmel, mikroprocesor, AT90USB1287.

Abstract

PINTÉR, Z. Interface for diagnostic busbar (SMBus/I2C) for accumulators. Brno: Brno University of Technology. Faculty of Electrical Engineering and Communication. 2013. 53 p. Supervisor of master´s thesis: Ing. Ondřej Pavelka.

The first chapter of the master´s thesis deals with the analysis of problems indesigning of devices, which include individual steps of the designing of the device and the choice of the most suitable option. It deals with the analysis of the proposed assignment and requirements for this device, analysis of battery management chip that the device should be able to communicate with, analysis of the I2C/SMBus communication protocol which is used for communication between the device with the battery management chips and possibilities of communication of the device with computer. The second chapter deals with the communications via USB bus and the description of this communication. The next step is dedicated to the selection of a programmable microprocessor Atmel AT90USB1287, describes his pin configuration, the possibilities, describes the way of USB interface connection, the method of the I2C/SMBus interface connection and its protection. In the last part of the this chapter the thesis describes Atmel microcontroller programming, specifically describes how to program I2C/SMBus communication, USB communication and how to write a program to the microprocessor. The third chapter of the thesis deals with the design of a single device, and components like a scheme of the connection and the concept of functional realization of the communication with device. The fourth chapter deals with the development of the device itself. Describes the parts of final involvement and describes the parts of the final program for microprocessor, such as communication with SMBus device and communication through the USB bus. The fifth chapter is devoted to the development of communication applications in Java. This section describes the different elements of applications such as communication with USB devices, data storage and open for viewing. The last chapter presents the final device and the final communication application.

Keywords: I2C/SMBus, USB, Atmel, microprocessor, AT90USB1287.

Prehlásenie

Prehlasujem, že svoju diplomovú prácu na tému „Interface pro diagnostickou sběrnici (SMBus/I2C) pro akumulátory“ som vypracoval samostatne pod vedením vedúceho diplomovej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb a predovšetkým som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následku porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/200 Zb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka č. 40/2009 Zb.

V Brne dňa 29.5.2013

podpis autora

Obsah

Obsah....6
Zoznam ilustrácií....8
Úvod....9
1 Analýza zariadenia....10
1.1Analýza požiadaviek a funkcií zariadenia....10
1.1.1 Komunikácia zariadenia s riadiacimi čipmi....10
1.1.2 Komunikácia zariadenia s počítačom....10
1.1.3 Výber riadiacej platformy zariadenia....10
1.1.4 Vytvorenie obslužného programu zariadenia....11
1.1.5 Vytvorenie komunikačného programu na počítač....11
1.2 Komunikácia medzi zariadením a PC....11
1.3 Riadiace čipy inteligentných akumulátorov....13
1.3.1 Riadiaci čip BQ2040....13
1.3.2 Zapojenie riadiaceho čipu BQ2040....14
1.3.3 Zbernica I2C a SMBus:....15
1.3.4 Príkazové kódy riadiaceho čipu BQ2040:....20
2 Návrh logických prvkov zariadenia....21
2.1 Výber spôsobu komunikácie zariadenia s počítačom....21
2.1.1 Princíp USB komunikácie....21
2.2 Výber platformy zariadenia....22
2.2.1 Mikroprocesory Atmel AVR....22
2.2.2 Atmel AVR AT90USB1287:....22
2.3 Programovanie mikroprocesorov Atmel....26
2.3.1 Bascom-AVR....26
2.3.2 Programovanie SMBus komunikácie....27
2.3.3 Programovanie USB komunikácie....28
2.3.4 Nahrávanie programu do mikroprocesorov Atmel....29
3 Návrh zariadenia....30
3.1 Princíp fungovania zariadenia....30
3.1.1 Popis princípu fungovania zariadenia....30
3.2 Schéma zapojenia....31

4 Konštrukcia zariadenia....33
4.1 Modul mikroprocesora....33
4.2 Programovanie mikroprocesora Atmel AT90USB1287....34
4.3 Modul SMBus zbernice....38
4.4 Výsledné osadenie a zapojenie zariadenia....41
5 Aplikácia na komunikáciu so zariadením....42
5.1 Komunikácia s USB zariadením....42
5.2 Uloženie údajov do externého súboru....45
5.3 Načítanie a filtrovanie CSV súboru....46
5.4 Spustenie aplikácie....46
6 Výsledný produkt diplomovej práce....47
Záver....50
Bibliografické odkazy....51
Slovník termínov....52
Zoznam príloh....53

Zoznam ilustrácií

Obr. 1.1: Zapojenie BQ2040 [9].....14
Obr. 1.2: Komunikácia na zbernici [5].....15
Obr. 1.3: Čítanie bytu [2].....17
Obr. 1.4: Zápis bytu [2].....17
Obr. 1.5: Čítanie slova [2].....18
Obr. 1.6: Zápis slova [2].....18
Obr. 1.7: Čítanie bloku [2].....19
Obr. 1.8: Zápis bloku [2].....19
Obr. 1.9: Čítanie bloku s PEC [2].....19
Obr. 2.1: Zapojenie AT90USB1287 [3].....23
Obr. 2.2: Zapojenie USB zbernice [4].....24
Obr. 2.3: Zapojenie dvojvodičovej zbernice [1].....25
Obr. 2.4: Bascom-AVR [7].....26
Obr. 3.1: Zapojenie USB zbernice.....31
Obr. 3.2: Zapojenie I2C/SMBus zbernice.....32
Obr. 4.1: AT90USBKey [4].....33
Obr. 4.2: Zapojenie SMBus zbernice.....39
Obr. 4.3: Zapojenie napájania SMBus zbernice.....40
Obr. 4.4: Plošný spoj modulu SMBus zbernice.....40
Obr. 4.5: Osadenie zariadenia.....41
Obr. 6.1: Obrazovka s načítanými údajmi.....47
Obr. 6.2: Obrazovka s filtrovanými záznamami.....48
Obr. 6.3: USB/SMBus Interface.....49

Úvod

V našej diplomovej práci by sme sa chceli venovať návrhu a vytvoreniu zariadenia, ktoré bude vedieť načítavať údaje z riadiacich obvodov inteligentných akumulátorov, napríklad v notebookoch a cez zbernicu SMBus z nich bude čítať údaje o stave akumulátora ako sú zostávajúca kapacita, napätie akumulátora, počet nabíjacích cyklov, výrobca, menovitá kapacita a menovité napätie. Zároveň toto zariadenie bude komunikovať s PC, do ktorého bude odosielať načítané údaje a bude ich možné porovnávať so staršími údajmi zo starších meraní pre sledovanie zmeny stavu akumulátora.

V prvej časti práce by sme sa chceli venovať popisu postupu vývoja zariadenia, ako aj jednotlivým krokom vývoja, ďalej by sme chceli venovať analýze požiadaviek na zariadenie vyplývajúce zo zadania práce, teda aké funkcie má zariadenie obsahovať, analýze I2C a SMBus komunikácie a analýze riadiacich obvodov inteligentných akumulátorov, s ktorými naše zariadenie bude komunikovať. Tu by sme sa chceli podrobne oboznámiť s fungovaním I2C a SMBus komunikácie.

V druhej časti práce by sme sa chceli venovať výberu vhodnej komunikácie nášho zariadenia s počítačom, ako napríklad niektorá z typov sériovej alebo USB komunikácie, výberu mikroprocesora alebo typu riadenia, ktorý bude pre naše zariadenie najvhodnejšie, alebo už hotového riešenia a následne opisu spôsobu programovania tohto riadenia a jeho komunikácie s riadiacimi čipmi a počítačom.

V tretej časti našej práce by sme sa chceli venovať návrhu princípu fungovania nášho zariadenia odvíjajúceho sa od vybraných možností riešenia zariadenia a návrhu jeho zapojenia.

Štvrtá časť práce by mala byť zameraná na výrobu samotného zariadenia a popis výsledných schém zapojenia a výsledných riešení zvolených pre realizáciu.

V piatej časti práce by sme chceli popísať vývoj komunikačnej aplikácie, vďaka ktorej bude možné komunikovať so zariadením a používateľ bude môcť vyčítať údaje z akumulátorov.

1 Analýza zariadenia

1.1 Analýza požiadaviek a funkcií zariadenia

Pri návrhu zariadenia musíme ako prvý krok urobiť analýzu funkcií, ktoré má naše zariadenie vykonávať. Zo zadania vyplýva, že naše zariadenie bude slúžiť na načítavanie údajov z riadiacich čipov inteligentných akumulátorov

1.1.1 Komunikácia zariadenia s riadiacimi čipmi

Inteligentné akumulátory obsahujú riadiace čipy, ktoré monitorujú ich stav a riadia ich nabíjanie. Naše zariadenie bude slúžiť na komunikáciu s tými riadiacimi čipmi. Pre správnu komunikáciu je potrebné vykonať analýzu zapojenia, spôsobu komunikácie, komunikačného protokolu a komunikačných príkazov týchto riadiacich čipov.

1.1.2 Komunikácia zariadenia s počítačom

Zariadenie bude priamo komunikovať s počítačom, preto je potrebné vybrať niektorú z možností komunikácie s počítačom. Pre konkrétny výber komunikácie je nutné špecifikovať požiadavky na komunikáciu, použitie zariadenia a prostredie, kde bude zariadenie používané. Na základe týchto špecifikácií je možné vybrať najvhodnejšiu formu komunikácie

1.1.3 Výber riadiacej platformy zariadenia

Ďalším krokom návrhu je výber vhodnej platformy, na ktorej bude zariadenie postavené. V dnešnej dobe už na trhu existujú mnohé hotové riešenia na rôzne požiadavky, z ktorých si je možné vybrať konkrétne riešenia, alebo je možné navrhnúť vlastné riešenia pomocou programovateľných mikroprocesorov. V oblasti programovateľných mikroprocesorov sú najčastejšie používané mikroprocesory ATMEL AVR od firmy Atmel a mikroprocesory PIC od firmy Microchip Technology.

1.1.4 Vytvorenie obslužného programu zariadenia

Pokiaľ si pre svoje zariadenie vyberieme použitie mikroprocesora, je potrebné pre neho vytvoriť program, ktorý bude riadiť komunikáciu s riadiacimi čipmi a riadiť komunikáciu s počítačom. Na vytvorenie programu pre mikroprocesor existujú rôzne nástroje a rôzne programovacie jazyky, ktoré poskytujú rôzne možnosti a úrovne tvorby programu.

1.1.5 Vytvorenie komunikačného programu na počítač

Keďže zariadenie načítané údaje má posielať do počítača, je potrebné vytvoriť aj aplikáciu, ktorá s týmto zariadením bude komunikovať cez zvolený komunikačný protokol a načítané údaje bude zobrazovať a interpretovať používateľovi.

1.2 Komunikácia medzi zariadením a PC

Ďalším krokom návrhu je výber spôsobu komunikácie medzi zariadením a PC. Na základe komunikácie sa totiž bude odvíjať aj výber najvhodnejšieho mikroprocesora a ostatných obvodov zariadenia. Typ komunikácie je potrebné vyberať vzhľadom na typ a určenie vyvíjaného zariadenia. V súčasnej dobe sa najčastejšie využívajú tieto možnosti komunikácie:

Sériová RS-232 komunikácia cez fyzický COM port

Toto je v dnešnej dobe už pomerne starý spôsob komunikácie. Jedná sa o sériovú komunikáciu cez fyzický COM port počítača. Komunikácia prebieha cez sériový protokol cez rozhranie RS-232. Tento typ komunikácie sa používal aj v DOS aplikáciach. Dnes sa používa predovšetkým kvôli jednoduchosti a spoľahlivosti. Väčšina dnešných používateľských počítačov však už sériový port neobsahuje, je však stále rozšírený v priemyselnom prostredí.

Sériová USB-RS-232 komunikácia cez virtuálny COM port

Jedná sa o rovnaký typ komunikácie ako pri sériovej komunikácii cez rozhranie RS-232, avšak fyzický COM port počítača je nahradený prevodníkom z USB portu na sériový port. V súčasnej dobe je to pomerne rozšírený spôsob pripojenia starších zariadení komunikujúcich cez sériový port k novým počítačom, ktoré už sériový port neobsahujú.

Sériová USB-UART komunikácia cez virtuálny COM port

Jedná sa znova o sériovú komunikáciu cez virtuálny COM port, pričom prevodník sa však už nachádza v samotnom zariadení a USB komunikácia sa neprevádza na sériový protokol RS-232 rozhrania, ale už priamo na 5 V TTL logiku, vďaku čomu je možné tento prevodník pripojiť k mikroprocesoru a nie je nutný ďalší obvod, ktorý bude prevádzať úrovne rozhrania RS-232 na TTL.

Sériová USB-UART komunikácia cez USB zariadenie

Jedná sa znova o sériovú komunikáciu, tentokrát sa však z pohľadu počítača jedná o komunikáciu priamo s USB zariadením. Takáto komunikácia je z programátorského hľadiska jednoduchšia ako komunikácia cez COM port. Komunikácia sa však prevádza na štandardnú sériovú komunikáciu s 5 V TTL logikou, ktorú je veľmi jednoduché spracovávať programovateľnými mikroprocesormi. Nevýhodou tohto riešenia je nutnosť ovládačov a knižníc od výrobcov týchto prevodníkov.

USB komunikácia cez Generic HID zariadenie

Jedná sa o pravú USB komunikáciu zo strany počítača aj zo strany zariadenia. Pri Generic Human Interface Device sa používajú štandardné komunikačné príkazy, ktoré už operačné systémy majú priamo v sebe, teda nie sú potrebné žiadne dodatočné ovládače pre komunikáciu so zariadením. Takýto spôsob komunikácie je pomerne jednoduchý na obsluhu zo strany ovládacích programov na strane počítača, sú však pomerne komplikované na obsluhu zo strany mikroprocesora

1.3 Riadiace čipy inteligentných akumulátorov

V inteligentných akumulátoroch sa používajú riadiace čipy, ktoré sa starajú o ich správne nabíjanie, kontrolujú stav akumulátorov a podávajú zariadeniam informácie o stave týchto akumulátorov. Medzi takéto riadiace čipy patria napríklad BQ2040, BQ2060 a BQ2063. Tieto čipy komunikujú so zariadeniami pomocou SMBus zbernice, ktorá je odvodená od zbernice I2C.

1.3.1 Riadiaci čip BQ2040

Riadiaci čip BQ2040 pochádza od firmy Texas Instruments. Bol vyvinutý pre monitorovanie aktuálneho stavu a nabíjania akumulátorov typu NiCD, NiMH a Li-Ion. Riadiaci čip BQ 2040 využíva na komunikáciu so zariadeniami zbernicu typu SMBus (System Management Bus) a podporuje monitorovanie príkazy typu SBData (Smart Battery Data).

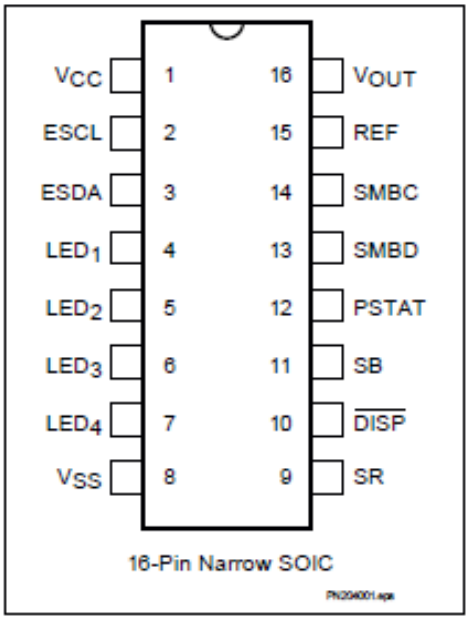
Čip BQ2040 taktiež podporuje SBData príkazy na riadenie nabíjania akumulátora. Stav akumulátora môže byť priamo zobrazovaný pomocou štvorsegmentového LED displeja, zobrazujúceho stav nabitia akumulátora po 25% hodnotách.

Riadiaci čip BQ2040 dokáže monitorovať aj samovoľné vybíjanie akumulátorov na základe vnútorného časovača, externého teplotného senzora a nastavenej hodnote uloženej v externej EEPROM pamäti. Čip sa taktiež dokáže automaticky nastaviť, alebo sa naučiť kapacitu akumulátora na základe plného cyklu vybíjania akumulátora z plnej kapacity do vybitého stavu.

Riadiaci čip potrebuje pre svoju správnu počiatočnú inicializáciu externú EEPROM pamäť, v ktorej sú uložené základné hodnoty o akumulátore.

Naše zariadenie pre zisťovanie stavu akumulátora potrebuje čítať z riadiaceho čipu údaje ako sú sériové číslo akumulátora, aktuálne napätie akumulátora, menovité napätie, počet uskutočnených nabíjacích cyklov, zostávajúca kapacita, maximálna kapacita a stav akumulátora.

1.3.2 Zapojenie riadiaceho čipu BQ2040



Obr. 1.1: Zapojenie BQ2040 [9]

Vcc:	Napájacie napätie 3,0 – 6,5 V
ESCL:	Časovanie pre EEPROM pamäť
ESDA:	Dáta pre EEPROM pamäť
LED1-4:	Segmenty LED displeja
Vss:	Systémová zem
SR:	Odporový vstup na monitorovanie nabíjania a vybíjania
DISP:	Aktivácia LED displeja
SB:	Vstup pre monitorovanie napätie akumulátora
PSTAT:	Vstup ochrany Li-Ion článkov, môže prerušiť nabíjanie akumulátora
SMBD:	SMBus Data, dátový port SMBus zbernice
SMBC:	SMBus Clock, časovací vstup SMBus zbernice
REF:	Referenčný výstup pre regulátor nabíjania
Vout:	Napájací výstup pre externú EEPROM pamäť

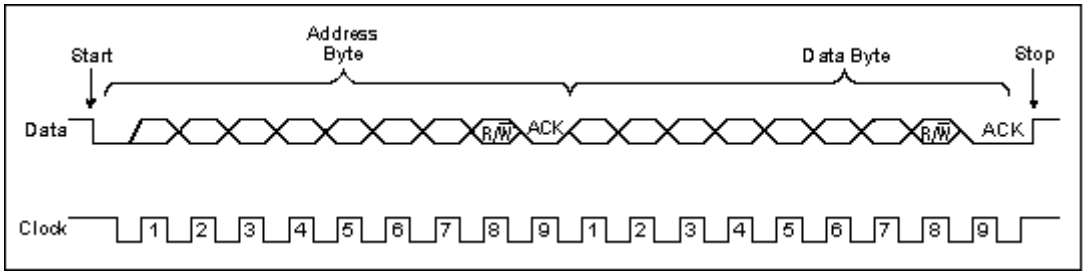
Naše zariadenie sa bude k riadiacemu čipu pripájať pomocou SMBus zbernice, ktorá sa pripája pomocou kontaktov SMBD a SMBC. Tieto kontakty sú pri inteligentných akumulátoroch vyvedené priamo na konektor akumulátora.

1.3.3 Zbernica I2C a SMBus:

Riadiaci čip BQ2040 komunikuje pomocou zbernice SMBus, ktorá je odvodená od zbernice I2C, ktorá je bežne používanou komunikačnou zbernicou medzi zariadeniami a externými EEPROM pamäťami. Zbernica SMBus sa zvyčajne používa na komunikáciu medzi rôznymi obvodmi v zariadeniach, napríklad v notebookoch.

Zbernica I2C a SMBus sú bežne používané dvojvodičové zbernice, ktoré sú v základe kompatibilné medzi sebou. Za normálnych okolností obe strany, master aj slave sú voľne zameniteľné medzi zbernicami. Obe zbernice používajú adresované slave strany, avšak niektoré špecifické adresy sa medzi zbernicami môžu líšiť. Zbernice štandardne komunikujú na rovnakej frekvencii do 100 kHz, avšak I2C zbernica podporuje 400 kHz aj 2 MHz režimy. Zvyčajne sa však kvôli zaisteniu kompatibility používa na všetkých zariadeniach frekvencia 100 kHz.

Základy komunikácie I2C a SMBus:



Obr. 1.2: Komunikácia na zbernici [5]

Start a Stop udalosti: Tieto sú mimoriadne dôležité z dôvodu, aby zariadenie vedelo, kedy más vstúpiť do inicializačného alebo resetovacieho stavu.

Data a Clock musia byť v aktívnom stave pre generovanie Start a Stop: Master zariadenie nemôže generovať Start alebo Stop stav, kým Dáta (SDA pri I2C a SMBD pri SMBus) a Clock (SCL pri I2C a SMBC pri SMBus) nie sú v aktívnej polohe

Start a Stop udalosti sú jediným bodom, kedy prebieha prenos na Data počas toho, ako je Clock v aktívnom stave

Data sa môžu meniť len keď je Clock v nulovom stave: Údaje na Data musia byť nemenné počas aktívnej polohy Clock a môžu byť zmenené len v nízkom stave Clock.

Odčasovanie a časovanie:

Odčasovanie a minimálna rýchlosť časovania sú hlavným rozdielom medzi I2C a SMBus zbernicou. Odčasovanie je prípad, keď slave zariadenie resetuje svoj vstup, keď časovanie na Clock je dlhšiu dobu v nulovej úrovni, zvyčajne je to 35ms. DĺžkA odčasovania taktiež určuje minimálnu rýchlosť časovania, keďže nikdy nemôže zostať v statickej úrovni. Odčasovanie sa však používa iba pri zbernici SMBus. Pri zbernici I2C sa môže držať Clock v statickej úrovni dovtedy, kým je potrebné spravovať údaje pred ďalšou komunikáciou.

V prípade I2C komunikácie, ak slave zariadenie zablokuje komunikáciu a Data aj Clock drží na nízkom stave, nie je možné odstrániť tento chybový stav. Najviac chybových stavov vzniká práve z toho dôvodu, že slave zariadenie ukončilo komunikáciu v stave, keď Data je v nízkej polohe. Master zariadenie v tomto prípade môže iba časovať Clock, kým sa Data nedostane do vysokej úrovne a tým nastane Start stav nasledovaný Stop stavom.

Pri SMBus komunikácii na rozdiel od I2C komunikácie, sa slave zariadenie automaticky resetuje do základného stavu, keď Clock linka je dlhšiu dobu v nízkom stave. Konkrétne podľa SMBus špecifikácií je to 35 ms.

Logické úrovne:

Medzi zbernicou I2C a SMBus je určitý rozdiel v rozsahu povolených napätí prezentujúcich vysoký a nízky stav. Tieto rozsahy sa vo veľkej časti prekrývajú, ale sú aj odlišné, preto nie je vždy zaručená kompatibilita medzi I2C a SMBus zbernicou.

Rozsah napätia pre I2C zbernicu:

Nízky stav: od -0,5 V do 1,5 V

Vysoký stav: od 3 V do 5,5 V

Rozsah napätia pre SMBus zbernicu:

Nízky stav: od 0 V do 0,8 V

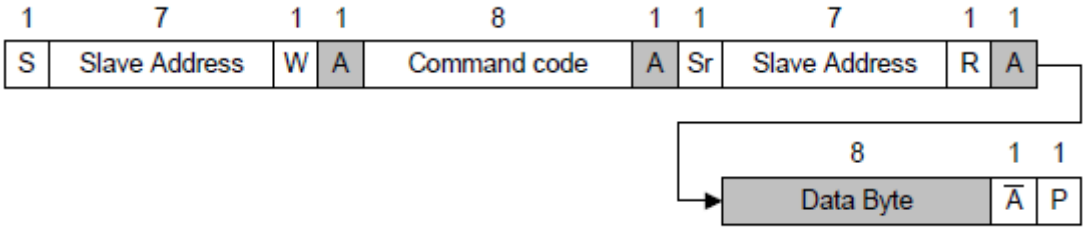
Vysoký stav: od 2,1 V do 5,5V

Z porovnania vidieť, že nie v celom rozsahu sú napätia zbernice kompatibilné, vo väčšej časti však áno, preto sa zväčša aj zariadenia navrhujú na tento spoločný rozsah, aby sa zabezpečila kompatibilita.

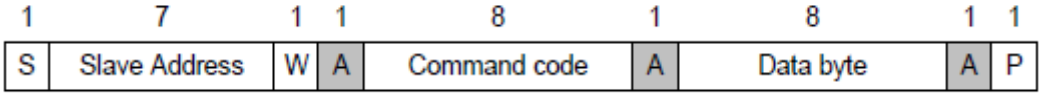
Špecifikácia komunikačného protokolu:

Pri SMBus komunikácii sa typ prenosu delí na zápis do slave zariadenia, čítanie zo slave zariadenia a volanie procesu, ďalej sa zápis a čítanie môže rozdeliť na tri skupiny, zápis/čítanie bytu, zápis/čítanie slova a zápis/čítanie bloku bytov. Všetky tieto typy ešte navyše podporujú aj doplnkový Packet Error Code, čo je kontrolný súčet zasielaných údajov, pre overenie správnosti prijatých údajov.

Čítanie a zápis jedného bytu:



Obr. 1.3: Čítanie bytu [2]

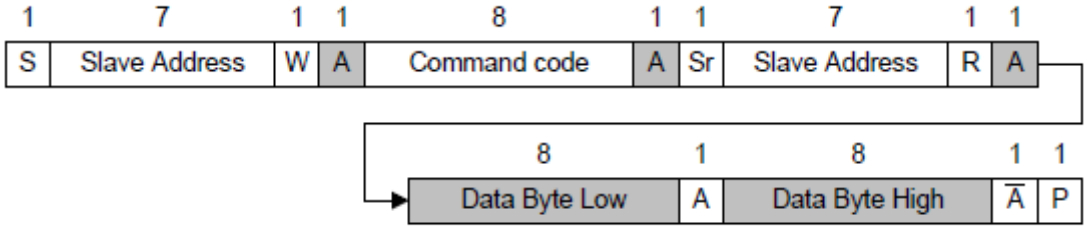


Obr. 1.4: Zápis bytu [2]

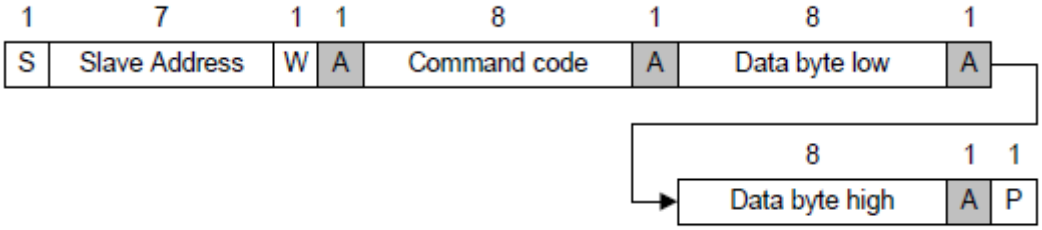
Pri čítaní jedného bytu, master zariadenie odošle Start bit, odošle sedem bitovú adresu slave zariadenia, jeden bit určujúci, že sa jedná o zápis, slave zariadenie vráti jeden ACK bit, master zariadenie pošle príkazový kód, určujúceho oblasť pamäte z ktorej sa má čítať, slave zariadenie vráti ACK bit, ktorý značí prijatie príkazového bytu, master zariadenie znova odošle Start bit, odošle sedem bitovú adresu slave zariadenia, jeden bit určujúci, že sa jedná o čítanie, slave zariadenie vráti jeden ACK bit a pošle dátový byte, master zariadenie vráti NACK bit a pošle Stop bit.

Pri zápise jedného bytu, master zariadenie odošle Start bit, odošle sedem bitovú adresu slave zariadenia, jeden bit určujúci, že sa jedná o zápis, slave zariadenie vráti jeden ACK bit, master zariadenie pošle príkazový kód, určujúceho oblasť pamäte do ktorej sa má zapisovať, slave zariadenie vráti ACK bit, ktorý značí prijatie príkazového bytu, master zariadenie pošle dátový byte, slave zariadenie vráti ACK bit, ktorý značí prijatie dátového bytu a master zariadenie pošle Stop bit.

Čítanie a zápis slova:



Obr. 1.5: Čítanie slova [2]



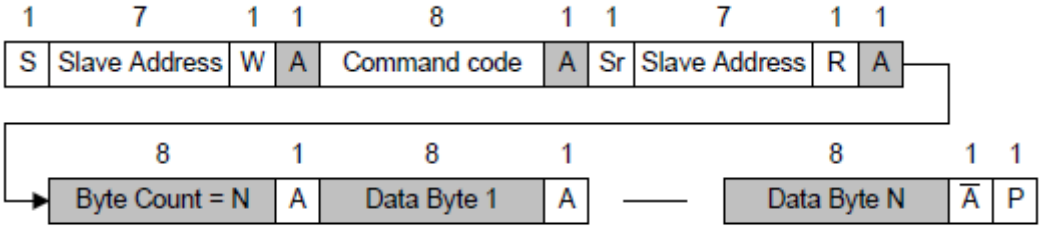
Obr. 1.6: Zápis slova [2]

Čitanie a zápis slova je takmer identický s čítaním a zápisom jedného bytu, posielajú sa však dva dátové byty.

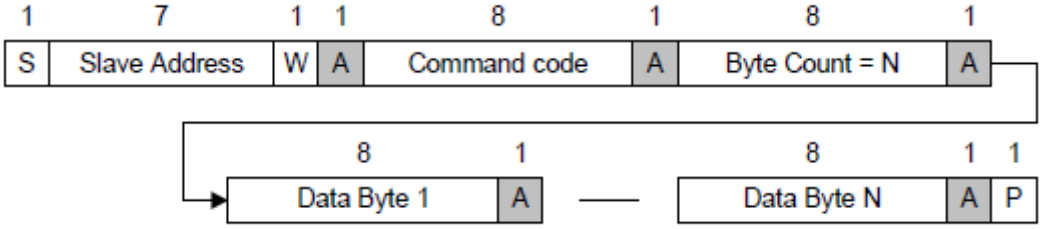
Pri čítaní jedného slova, master zariadenie odošle Start bit, odošle sedem bitovú adresu slave zariadenia, jeden bit určujúci, že sa jedná o zápis, slave zariadenie vráti jeden ACK bit, master zariadenie pošle príkazový kód, určujúceho oblasť pamäte z ktorej sa má čítať, slave zariadenie vráti ACK bit, ktorý značí prijatie príkazového bytu, master zariadenie znova odošle Start bit, odošle sedem bitovú adresu slave zariadenia, jeden bit určujúci, že sa jedná o čítanie, slave zariadenie vráti jeden ACK bit a pošle dolný dátový byte, master zariadenie vráti ACK, slave zariadenie pošle horný dátový byt, master zariadenie vráti NACK bit a pošle Stop bit.

Pri zápise jedného bytu, master zariadenie odošle Start bit, odošle sedem bitovú adresu slave zariadenia, jeden bit určujúci, že sa jedná o zápis, slave zariadenie vráti jeden ACK bit, master zariadenie pošle príkazový kód, určujúceho oblasť pamäte do ktorej sa má zapisovať, slave zariadenie vráti ACK bit, ktorý značí prijatie príkazového bytu, master zariadenie pošle dolný dátový byte, slave zariadenie vráti ACK bit, ktorý značí prijatie dolného dátového bytu, master zariadenie pošle horný dátový byte, slave zariadenie vráti ACK bit a master zariadenie pošle Stop bit.

Čítanie a zápis bloku:



Obr. 1.7: Čítanie bloku [2]

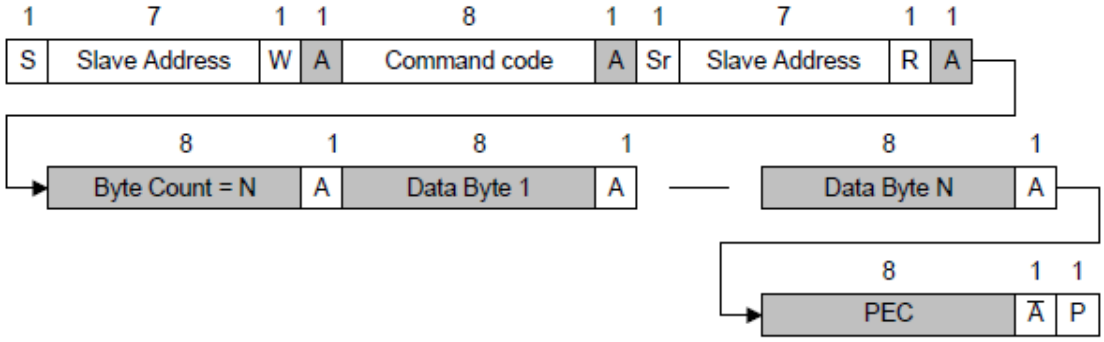


Obr. 1.8: Zápis bloku [2]

Čítanie a zápis bloku je v zásade rovnaký ako zápis slova, nie je však pevne daný počet dátových bytov, ktoré sa budú zapisovať alebo čítať. Tento počet sa posiela pred prvým dátovým bytom.

Packet Error Code (PEC):

Packer Error Code (PEC) je kontrolný súčet dátových bytov posielaný pri komunikácii. Posiela sa za posledným dátovým bytom. Jeho dĺžka je jeden byte.



Obr. 1.9: Čítanie bloku s PEC [2]

1.3.4 Príkazové kódy riadiaceho čipu BQ2040:

Ako vyplýva zo špecifikácie SMBus komunikácie, SMBus slave zariadenie určuje pamäťové miesto do ktorého sa má zapisovať, alebo z ktorého sa má čítať na základe príkazového kódu, ktoré prijalo od master zariadenia. Tento príkazový kód je dlhý jeden byte. Pri riadiacom čipe je priamo určené výrobcom, do ktorého pamäťového miesta sa ktoré hodnoty ukladajú, ktoré sú zapisovateľné, ktoré sú určené iba na čítanie, ktoré sú statické a ktoré ukladá do tohto miesta samotný riadiaci čip na základe aktuálnych údajov akumulátora.

Tabuľka vybraných riadiacich kódov:

Funkcia	Kód	Prístup	Jednotky
Napätie	0x09	čítanie	mV
Teplota	0x08	čítanie	0,1°K
Aktuálny prúd	0x0A	čítanie	mA
Zostávajúca kapacita	0x0F	čítanie	mAh
Maximálna kapacita	0x10	čítanie	mAh
Stav akumulátora	0x16	čítanie	príznak
Počet cyklov	0x17	čítanie	počet cyklov
Menovitá kapacita	0x18	čítanie	mAh
Menovité napätie	0x19	čítanie	mV
Dátum výroby	0x1B	čítanie	-
Sériové číslo	0x1C	čítanie	číslo
Výrobca	0x20	čítanie	reťazec znakov
Režim akumulátora	0x03	čítanie/zápis	príznak

Tabuľka 1.1: Riadiace kódy čipu BQ2040

Typy komunikačných prenosov:

Riadiaci čip pri komunikácii s master zariadením pre prenos stavových práv nevyužíva všetky typy prenosu protokolu SMBus, ale pre svoju komunikáciu využíva iba čítanie a zápis slova a čítanie bloku. Možnosti komunikácie ako zápis jedného bytu alebo zápis bloku sa využíva iba pri konfigurácii riadiaceho čipu pre výrobe akumulátora.

2 Návrh logických prvkov zariadenia

Po zoznámení sa s požiadavkami na zariadenie a jednotlivými možnosťami ich realizácie, sme pristúpili k výberu konkrétnych prvkom systému, ako sú použitá platforma systému, typ komunikácie s počítačom, spôsob vytvárania obslužného programu zariadenia a vytváranie používateľského programu na počítač.

2.1 Výber spôsobu komunikácie zariadenia s počítačom

Pred výberom platformy zariadenia sme potrebovali určiť spôsob komunikácie zariadenia s počítačom, keďže to bude jedným z určujúcich faktorov pre výber najvhodnejšieho riešenia, vzhľadom na rôzne možnosti rôznych riešení.

Pre svoje zariadenie sme sa rozhodli použiť Generic HID USB komunikáciu, z dôvodu podpory USB komunikácie u takmer všetkých dnešných počítačov a z dôvodu, že pri tejto komunikácii nie sú potrebné ďalšie dodatočné ovládače do operačného systému, keďže pre komunikáciu sa využívajú základné štandardné možnosti komunikácie, ktoré vychádzajú už priamo zo špecifikácie Generic HID zariadení.

2.1.1 Princíp USB komunikácie

USB komunikácia funguje na princípe jednej zbernice s jedným master zariadením, ktoré väčšinou tvorí počítač a s niekoľkými slave zariadeniami, ktoré predstavujú pripojené zariadenie ako klávesnice, myši, tlačiarne, pamäťové média a rôzne iné zariadenia, ktoré sa pripájajú pomocou USB zbernice. Dáta sa po zbernici prenášajú v krátkych 8 bytových a dlhých 256 bytových paketoch. Slave zariadenia nikdy nemôžu vysielať dáta do zbernice samé od seba, môže vysielať len na žiadosť master zariadenia. Jedná sa o dvojvodičovú zbernicu, kde Data+ a Data- tvoria jeden krútený dvojpár. Súčasne je cez USB zaistené aj napájanie zariadení s napätím 5 V. Toto napájanie je vedené cez ďalší pár, ktorý však nie je krútený. Najvyšší možný odber USB zariadenia napájaného z master zariadenia je 500 mA.

2.2 Výber platformy zariadenia

Vzhľadom na požiadavky na zariadenie, sme sa rozhodli vybrať použitie programovateľného mikroprocesora od firmy Atmel. Výhodou tohto riešenia, je možnosť vytvorenia riadiaceho programu a prispôsobenia zariadenia presne podľa našich požiadaviek a nie je potrebné upravovať naše požiadavky na základe možností hotových riešení, alebo komplikovať zariadenie zbytočnými funkciami, ktoré by obsahovalo komplexné hotové riešenie, ktoré by presne spĺňalo naše požiadavky.

2.2.1 Mikroprocesory Atmel AVR

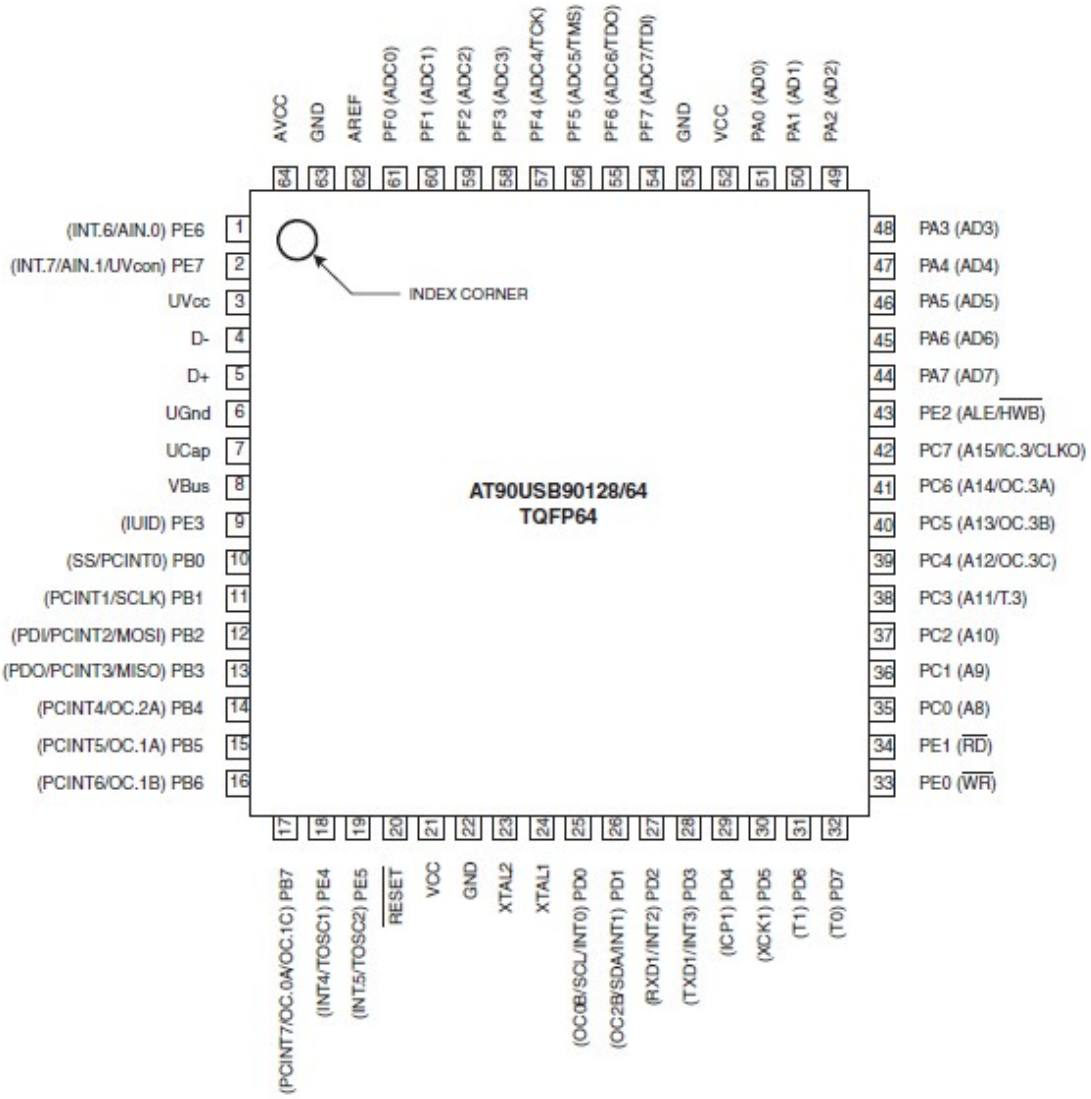
Mikroprocesory Atmel ponúkajú široké možnosti využitia, integrovaný dizajn, overené technológie a inovácie, ktoré sú vhodné na dnešné pokročilé zariadenia. Sú optimalizované pre nízku spotrebu, vysokú rýchlosť komunikácie a širokú podporu komunikačných rozhraní. Široké možnosti naprogramovania umožňujú vyvíjať kompletné systémy pre každý druh použitia. Mikroprocesory Atmel využívajú vysoko efektívnu architektúru zapojenia a strojového kódu, ktoré umožňujú rýchle a efektívne vytváranie zariadení podľa našich predstáv.

2.2.2 Atmel AVR AT90USB1287:

Pre naše zariadenie sme si vybrali mikroprocesor Atmel AVR AT90USB1287, keďže sa jedná o novú sériu základných programovateľných mikroprocesorov, ktorá už má priamo integrované aj komunikačné rozhranie pre USB komunikáciu.

Atmel AVR AT90USB1287 je 8 bitový RISC programovateľný mikroprocesor. Obsahuje 128kB flash pamäte umožňujúcej súčasný zápis aj čítanie, obsahuje 4kB EEPROM pamäte a 8kB SRAM pamäte. Obsahuje 48 vstupno výstupných portov, komunikačné rozhrania pre sériovú USART, PWM, SPI, TWI a USB komunikáciu. Obsahuje 8 kanálový 10 bitový A/D prevodník. Prevádzkový rozsah napájania je 2,7 až 5,5V.

Zapojenie mikroprocesora Atmel AVR AT90USB1287



Obr. 2.1: Zapojenie AT90USB1287 [3]

VCC: Napájacie napätie

GND: Zem

Port A (PA7 - PA0): Obojsmerný 8 bitový vstupno-výstupný port.

Port B (PB7 - PB0): Obojsmerný 8 bitový vstupno-výstupný port.

Port C (PC7 - PC0): Obojsmerný 8 bitový vstupno-výstupný port.

Port D (PD7 - PD0): Obojsmerný 8 bitový vstupno-výstupný port.

Port E (PE7 - PE0): Obojsmerný 8 bitový vstupno-výstupný port.

Port F (PF7 - PF0): Obojsmerný 8 bitový vstupno-výstupný port.

D+ : Kladný dátový port USB rozhrania

D- : Záporný dátový port USB rozhrania

UGND: Zem USB rozhrania

UVCC: Napájanie USB rozhrania

UCAP: Výstup napájania regulátora USB rozhrania

VBUS: Monitor napájania USB zbernice

RESET: Resetovací vstup

XTAL1: Výstup na externý oscilátor

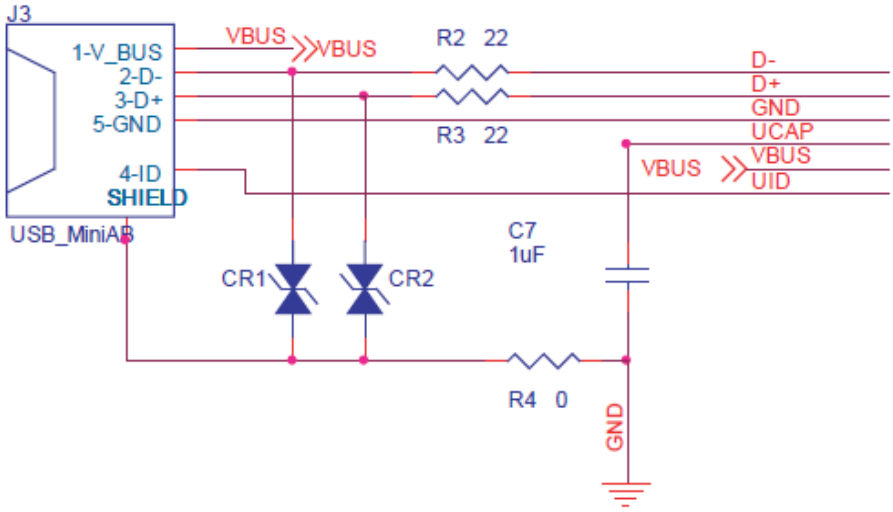
XTAL2: Vstup externého oscilátora

PD1: Dátový port pre dvojvodičové zbernice

PD0: Clock port pre dvojvodičové zbernice

Pre naše zariadenie sú z hľadiska zapojenia dôležité predovšetkým rozhrania pre USB zbernicu a SMBus zbernicu, teda porty UVCC, D+, D- UGND, UCAP pre USB zbernicu a PD1 a PD0 pre SMBus zbernicu.

Zapojenie USB zbernice:



Obr. 2.2: Zapojenie USB zbernice [4]

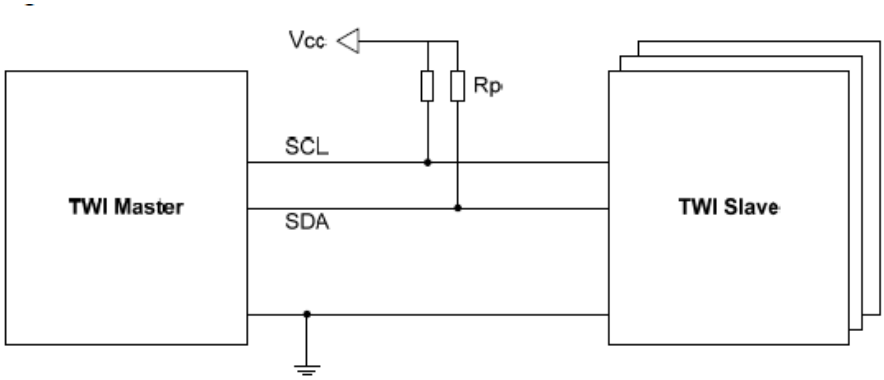
Na schéme zapojenia USB zbernice vidíme pomerne jednoduché a priame pripojenie USB konektora na USB rozhranie mikroprocesora AT90USB1287.

Kontakt VBUS je priamo pripojený VBUS kontakt mikroprocesora, D- je cez rezistor na obmedzenie prúdu pripojený na kontakt D-, D+ je cez rezistor na

obmedzenie prúdu pripojený na kontakt D+, zároveň sú D+ a D- cez ochrannú diódu prepojené na zem pre ochranu pred zvýšeným napätím, kontakt GND je pripojený priamo na kontakt GND mikroprocesora. Výstup regulátora UCAP je pripojený cez kondenzátor na zem podľa odporúčaní. kontakt 4 - ID slúži na určenie režimu USB zariadenia, určuje či sa má správať ako USB zariadenie, alebo ako USB Host.

Zapojenie SMBus zbernice:

Keďže dvojvodičová zbernica je obojsmerná, rozhranie musí byť s otvoreným kolektorom, preto každý z vodičov musí byť pripojený na napájacie napätie cez rezistor. Vodič je vo vysokom stave, keď ho žiadne zo zariadení nedrží otvorený.



Obr. 2.3: Zapojenie dvojvodičovej zbernice [1]

Obrázok č. 10 zobrazuje pripojenie zariadení na dvojvodičovú zbernicu. Hodnota Rp rezistora závisí od napájacieho napätia, kapacity a typu zbernice. Pri SMBus zbernici by to malo byť viac ako 14 kΩ pri napájacom napätí 5 V alebo viac ako 8,5 kΩ pri napájacom napätí 3,3 V.

Ochrana SMBus zbernice:

Keďže zariadenie sa bude pripájať na akumulátor notebooku, kde môže byť napätie až 20V a pri nesprávnom pripojení by sa toto napätie mohlo dostať na SMBus zbernicu, je potrebné ochrániť mikroprocesor pred zničením. Túto ochranu je možné vyriešiť jednoduchým pripojením odporu s hodnotou 100 Ω do série na každú linku a zároveň každú linku pripojiť na zem cez transilovú diódu, ktorá odvedie zvýšené napätie.

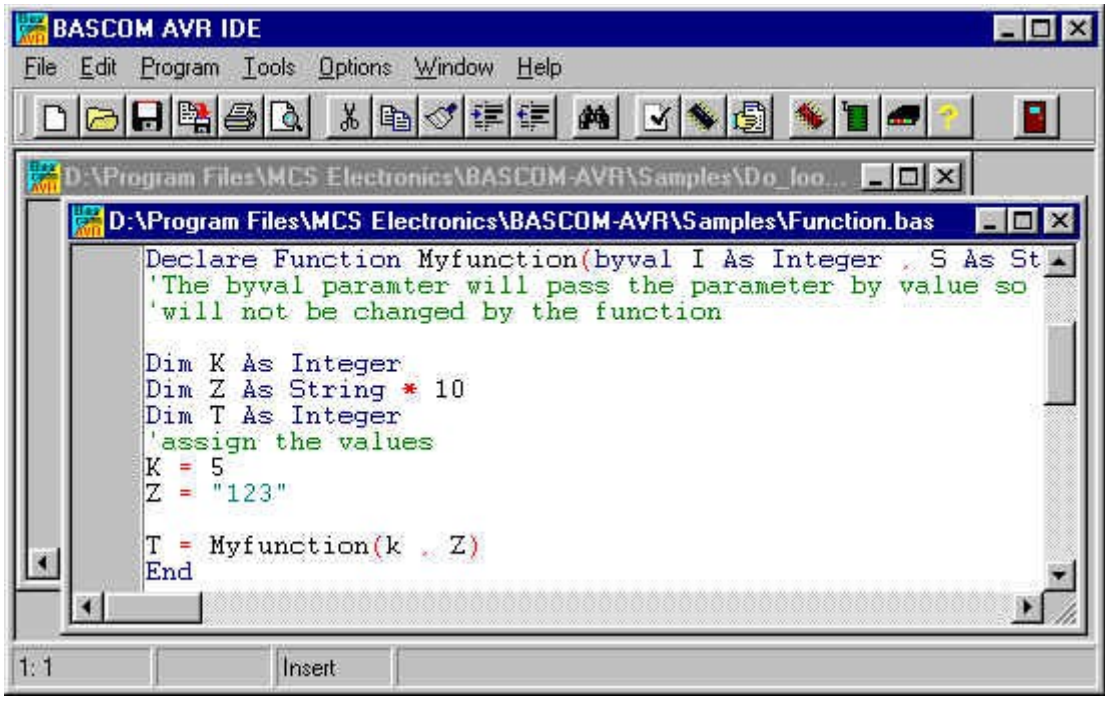
2.3 Programovanie mikroprocesorov Atmel

Existuje množstvo vývojových nástrojov pre programovanie mikroprocesorov Atmel. Sú to napríklad AVR Studio od Atmelu, WinAVR, Bascom-AVR od MCS Electronics.

My sme si pre svoje potreby vybrali nástroj Bascom-AVR od MCS Electronics. kvôli jeho jednoduchému štrukturovanému kódu a integrovaným funkciám pre komunikáciu cez I2C a USB protokol.

2.3.1 Bascom-AVR

Bascom-AVR je vývojové prostredie pre Atmel AVR mikroprocesory. Je založený na programovacom jazyku Basic. Obsahuje široký počet inštrukcií pre podmienky, cykly, procesy, špeciálne inštrukcie pre komunikačné rozhrania, ako sú dvojvodičové I2C zbernice, jednovodičové zbernice, USB zbernicu, sériovú aj paralelnú komunikáciu. Obsahuje simulátor, napísaný program je možné odsimulovať a vyskúšať ešte pred uložením do mikroprocesora.



Obr. 2.4: Bascom-AVR [7]

2.3.2 Programovanie SMBus komunikácie

Vývojové prostredie Bascom-AVR obsahuje v sebe príkazy na komunikáciu cez I2C protokol. Pomocou týchto príkazov je možné priamo posielať START bit, posielať adresu s príznakom čítania alebo zápisu, zapisovať dáta, prijímať dáta, posielať ACK a NACK bity.

```
Dim Mcp23016_adress_w As Byte
Mcp23016_adress_w = &H40 'Write Adress: 0 1 0 0 A2 A1 A0 0

Dim Mcp23016_adress_r As Byte
Mcp23016_adress_r = &H41 'Read Adress: 0 1 0 0 A2 A1 A0 1
```

Výpis kódu 2.1: Zápis adresy [6]

V uvedenom kóde môžme vidieť formát zápisu sedem bitovej adresy slave zariadenia a posledný bit určujúci, či sa jedná o čítanie alebo zápis. Bity A2, A1 a A0 tvoria premennú adresy zariadenia. Posledný bit značí, či sa jedná o volanie zariadenia pre zápis, bit má hodnotu 0, alebo volanie zariadenia pre čítanie, bit má hodnotu 1.

```
Sub I2c_mcp23016_read(byval Cmd As Byte)
' Read data from the I2C MCP23016

 I2cstart 'Generate START condition
 I2cwbyte Mcp23016_adress_w 'Transmit The "ADDRESS and WRITE" Byte
 I2cwbyte Cmd  'Transmit The Command Byte
 I2cstop 'Generate a STOP condition
 I2cstart 'Generate a START condition
 I2cwbyte Mcp23016_adress_r  'Transmit ADDRESS with READ command
 I2crbyte Lsb , Ack 'Receive first DATA byte (LSB) and acknowledge
 I2crbyte Msb , Nack 'Receive second DATA byte (MSB) and don't acknowledge
 I2cstop 'Generate a STOP condition

End Sub
```

Výpis kódu 2.2: Proces čítania slova [6]

Z uvedeného výpisu kódu jasne vidieť štruktúru čítania slova SMBus protokolu, uvedenú na obrázku 1.5 v kapitole 1.2.3.

Ako prvý sa posiela START bit, následne sa posiela adresa slave zariadenia ukončená bitom určujúcim zápis, posiela sa byte riadiaceho kódu a následne sa posiela STOP bit, pre ukončenie zápisovej časti komunikácie. Nasleduje nový START bit pre začiatok novej komunikácie, posiela sa adresa s čítacím bitom, prijíma sa spodný byte slova, vysiela sa ACK bit, prijíma sa horný byte slova, vysiela sa NACK bit a pre ukončenie komunikácie sa posiela STOP bit.

2.3.3 Programovanie USB komunikácie

Komunikáciu USB zbernice vo vývojovom prostredí Bascom-AVR zabezpečujú vstavané knižnice, ktoré zabezpečujú samotnú štruktúru a špecifikáciu komunikácie cez USB zbernicu. Vďaka týmto funkciám sa programátor nemusí zaoberať programovaním samotnej komunikácie, ale iba stačí zadať konfiguračné parametre USB zariadenia a následne volá komunikačné procedúry špecifikované v komunikačnej knižnici.

Prvá časť obslužného programu pre USB komunikáciu sa skladá z nastavenia potrebným konfiguračných parametrov pre komunikáciu. Medzi tieto parametre patrí napríklad premenná Vendor_ID a Product_ID, ktoré zabezpečujú identifikáciu samotného zariadenia na spoločnej USB zbernici. ďalej sú to parametre ako kategória zariadenia, teda či sa jedná o HID zariadenie a akého typu, napríklad klávesnica, myš a iné základné zariadenia. Ďalej parametre ako určenie prerušení, názvu a výrobcu zariadenia, počet prenášaných bytov, ich názvy a konfigurácia rôznych iných konfiguračných parametrov.

Druhým krokom obslužného programu USB komunikácie je inicializácia USB zariadenia, kde sa overí správna funkčnosť komunikácie.

Následne už môže prebiehať samotná komunikácia zariadenia s hostiteľským zariadením, teda v našom prípade s počítačom. Samotnú komunikáciu zabezpečuje USB rozhranie mikroprocesora a komunikačné procesy obsiahnuté vo vývojovom prostredí Bascom-AVR. Pri prijímaní údajov sa sleduje stavový bit „Ueintx.rxouti", či je aktívny, teda, či je zariadenie pripravené na čítanie, následne sa načítajú do premenných prijímané byty a s dátami sa môže pracovať. Pri posielaní údajov sa overí stavový bit „Ueintx.txini", či je zariadenie pripravené posielať dáta, teda či neprebieha iná komunikácia. Ak je pripravené, do premennej „Uedatx" sa postupne vloží 8 bytov, ktoré sa zároveň odosielajú cez USB zbernicu do hostiteľského zariadenia.

Z uvedeného postupu programovania USB komunikácie vo vývojovom prostredí Bascom-AVR vyplýva jednoduchosť používania tohto spôsobu komunikácie, vďaka zabudovaným knižniciam, ktoré obsahujú funkcie, ktoré riadia samotné procesy komunikácie, vďaka tomu programátorovi stačí volať príslušné funkcie a nemusí sám programovať zložité princípy USB komunikácie.

2.3.4 Nahrávanie programu do mikroprocesorov Atmel

Na nahrávanie napísaného programu do Atmel mikroprocesorov existuje niekoľko rôznych možností.

Jednou z možností sú rôzne programátory mikroprocesorov, ktoré sa k počítaču pripájajú cez USB port, priamo spolupracujú s vývojovým prostredím a obsahujú vlastný program, vďaka ktorému dokážu nahrať prijatý program do mikroprocesora. Takýmto programátorom mikroprocesorov je napríklad Atmel Dragon priamo od spoločnosti Atmel.

Druhou možnosťou programovania mikroprocesorov Atmel je programovanie cez špeciálne zabudované sériové vstupy ktoré sa pripájajú priamo na LPT port počítača, ktorý sa v tomto prípade používa pre sériovú komunikáciu. Takto pripojený Atmel mikroprocesor dokáže vývojové prostredie prepnúť do programovacieho režimu a následne cez sériový komunikáciu do neho nahrať napísaný program.

Treťou možnosťou, ktorú obsahujú nové rady Atmel mikroprocesorov s USB rozhraním je špeciálny "DFU" režim, do ktorého dokáže mikroprocesor prepnúť samotný program, alebo je možné do neho vstúpiť sekvenciou aktivácie vstupov, po ktorého aktivácii je možné do neho nahrávať napísaný program cez samotné USB rozhranie. Komunikáciu a nahrávanie programu do mikroprocesora v tomto režime zabezpečuje špeciálny softvérový nástroj "FLIP" priamo od spoločnosti Atmel.

3 Návrh zariadenia

Po analýze komunikačných protokolov, možností riadiacich čipov, možností mikroprocesorov Atmel a ich programovaní sme sa zamerali už na vývoj samotného zariadenia do funkčnej podoby. Táto časť spočíva v návrhu schémy zapojenia, návrhu fungovania obslužného programu, návrhu dátovej komunikácie medzi počítačom a zariadením.

3.1 Princíp fungovania zariadenia

3.1.1 Popis princípu fungovania zariadenia

Naše zariadenie bude slúžiť na čítanie údajov z riadiacich čipov akumulátorov. Na základe tabuľky príkazov riadiacich čipov 1.1 vieme, že pri I2C komunikácii budeme potrebovať čítanie jedného slova a čítanie bloku. Zoznam týchto riadiacich kódov bude uložený v komunikačnej aplikácii na počítači a do mikroprocesora sa bude odosielať riadiaci kód požadovanej hodnoty. Vzhľadom na to, že budeme potrebovať čítanie slova aj čítanie bloku, čo musia byť dva rozdielne procesy, bude musieť zoznam riadiacich kódov obsahovať aj druh vracanej hodnoty.

Samotný sled udalostí čítania bude teda následovný: Pri USB komunikácii sa prenáša osem bytov. V prvom byte sa bude prenášať riadiaci kód požadovanej hodnoty pre riadiaci čip akumulátora, v druhom byte sa bude prenášať typ vracanej hodnoty. Na základe druhého bytu sa program mikroprocesora rozhodne, či má volať procedúru na čítanie slova, alebo na čítanie bloku. Do tejto procedúry vloží do premennej pre riadiaci kód hodnotu prvého prijatého bytu cez USB rozhranie. Následne dostane od riadiaceho čipu požadovanú hodnotu. V prípade, že sa jednalo o slovo, v prvom odosielanom byte odošle informáciu o type posielanej hodnoty, v druhom byte sa bude nachádzať hodnota spodného bytu slova a v tretom byte sa bude nachádzať hodnota horného bytu slova. V prípade, že vracaná hodnota bude blokového typu, teda z hľadiska premenných sa bude jednať o pole, v prvom byte sa znova bude posielať typ vracanej premennej, v druhom byte sa bude posielať počet bytov pola, následne sa v ostatných začnú prenášať jednotlivé prvky poľa, v prípade, že pole bude mať viac ako šesť prvkov, bude sa

posielať ďalších osem bytov, tentokrát však sa však už bude posielať osem prvkov poľa, keďže komunikačný program ich bude očakávať a bude ich vedieť do poľa, nepotrebuje už prvé dva identifikačné údaje o prenose.

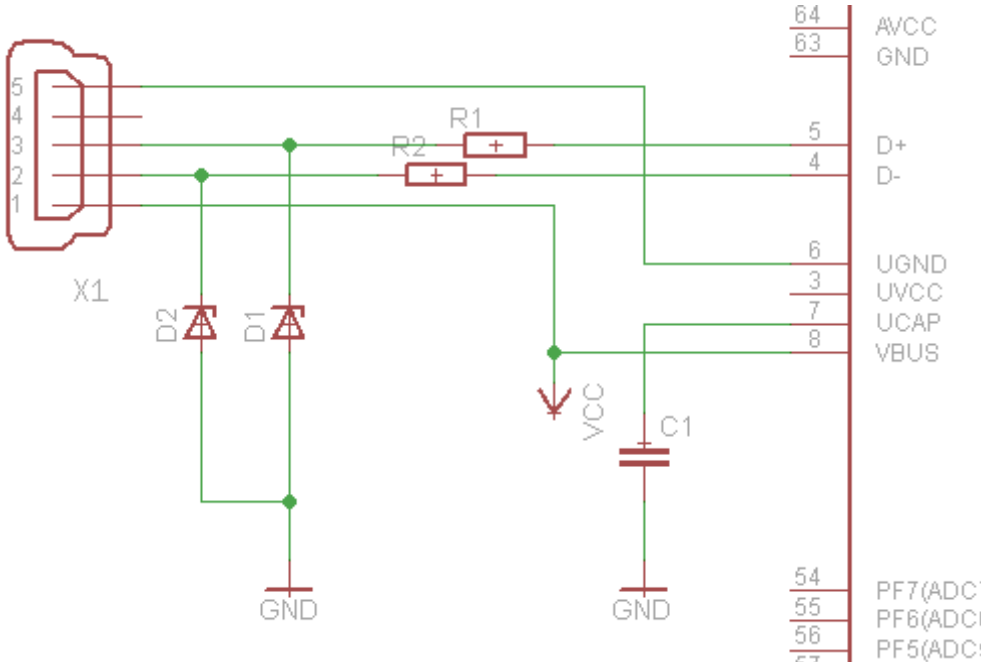
Tým to spôsobom dostaneme jednu z požadovaných hodnôt, pre ďalšie požadované hodnoty sa odošle nové požiadavka s riadiacim kódom požadovanej hodnoty. Následne keď budeme mať všetky požadované údaje prijaté, si bude môcť používateľ v komunikačnom programe tieto údaje uložiť a porovnať ich so staršími uloženými údajmi.

3.2 Schéma zapojenia

Schému zapojenia sme navrhli na základe technickej dokumentácie k mikroprocesoru Atmel AT90USB1287 a vývojovej doske pre USB komunikáciu AT90USBKEY.

Zapojenie pozostáva z dvoch hlavným častí okolo mikroprocesora, prvou časťou je zapojenie USB konektora a jeho pripojenie na USB zbernicu mikroprocesora, druhou časťou je zapojenie zbernice I2C/SMBus.

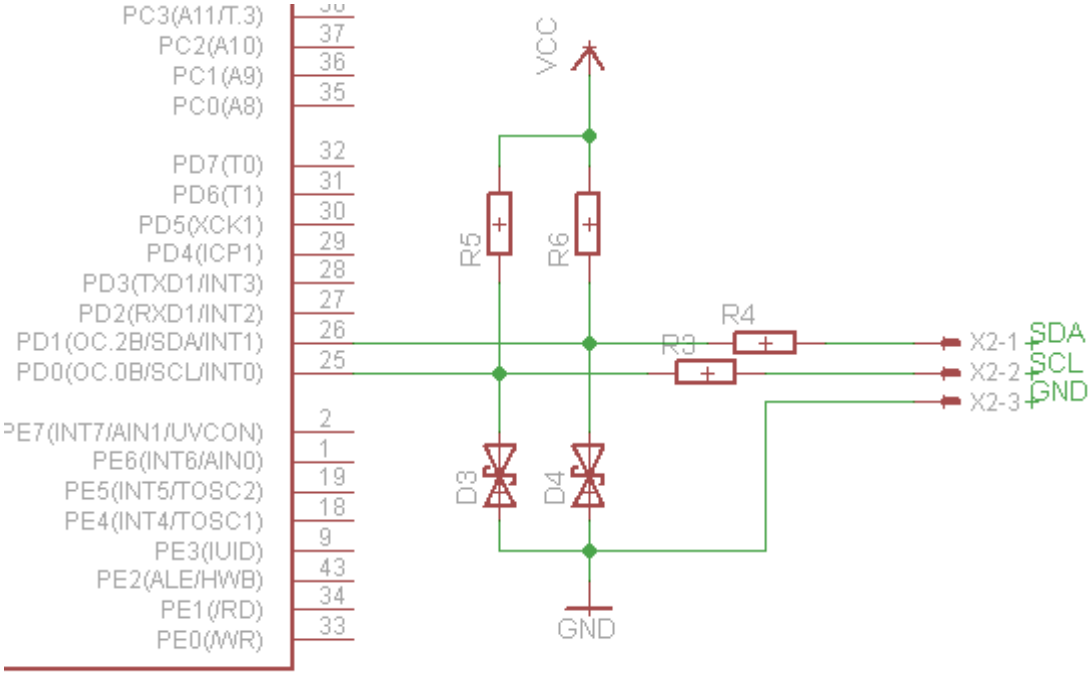
Zapojenie USB zbernice:



Obr. 3.1: Zapojenie USB zbernice

Zapojenie USB zbernice pozostáva z USB konektora X1, ktorého kontakt 1 predstavuje kladnú polaritu napájacieho napätia a je pripojený priamo na kontakt VBUS a VCC mikroprocesora.. Kontakt 5 predstavuje zem a je pripojený priamo na kontakt GND. Kontakt 2 predstavuje zápornú linku USB zbernice, cez ochranný odpor R2 je pripojený na kontakt D-. Kontakt 3 predstavuje kladnú linku USB zbernice, cez ochranný odpor R1 je pripojený na kontakt D+. Komunikácia na USB zbernici používa polaritu 3,3 V, z toho dôvodu sú obe linky zbernice pripojené na zem cez 3,3 V zenerové diódy D1 a D2, ktoré udržia napätie na takejto úrovni aj pri napájaní mikroprocesora napätím 5 V, kedy pri komunikácii používa napätie 5 V.

Zapojenie I2C/SMBus zbernice:



Obr. 3.2: Zapojenie I2C/SMBus zbernice

Zapojenie IC2/SMBus zbernice je veľmi jednoduché, zapojenie obsahuje odpory R3 a R4 ja každej linke zbernice pre obmedzenie maximálneho prúdu. Keďže I2C/SMBus zbernica využíva zapojenie s otvoreným kolektorom, musia byť obe linky pripojené cez odpory k napájaciemu napätiu. V tomto prípade, keďže zariadenie bude napájané pomocou 3,3 V, musia mať tieto odpory R5 a R6 hodnotu 15 kΩ alebo viac. Dióda D3 a D4 sú obojsmerné transilové diódy, ktoré chránia zbernicu pred zvýšeným napätím.

4 Konštrukcia zariadenia

Po analýze požiadavkov na zariadenie a teoretickom návrhu zariadenia sme pristúpili k samotnej realizácii zariadenia.

Naše zariadenie sme sa rozhodli rozdeliť na dva samostatné moduly, na modul s mikroprocesorom Atmel AT90USB1286 so zapojením USB zbernice a napájania a na modul SMBus zbernice s napájaním riadiacich čipov akumulátorov BQ2060.

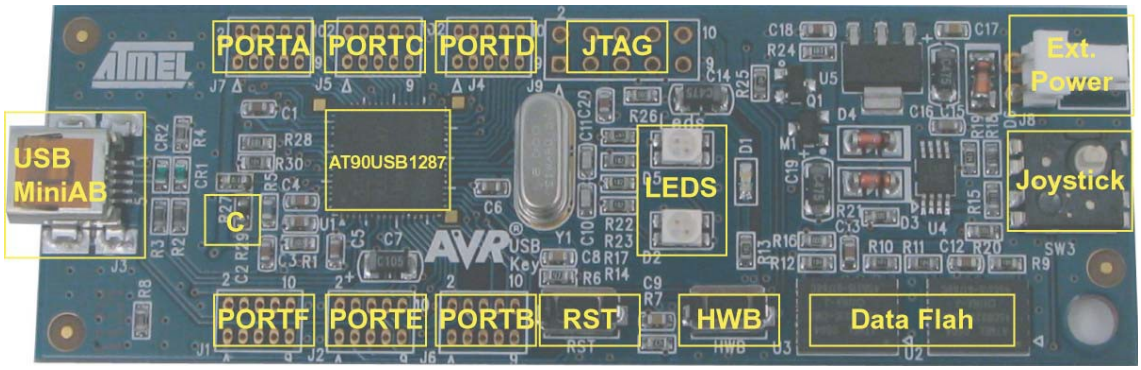
4.1 Modul mikroprocesora

Základ modulu mikroprocesora tvorí mikroprocesor Atmel AT90USB1287 so zapojením USB zbernice. Pre tento modul sme si vybrali produkt od spoločnosti Atmel, Atmel AT90USBKey, ktorý plne zodpovedá našim požiadavkám.

Atmel AT90USBKey:

Atmel AT90USBKey je produktom spoločnosti Atmel, určeným na zjednodušenie vývoja zariadení používajúcich mikroprocesory Atmel série USB s integrovanou USB zbernicou.

Atmel AT90USBKey obsahuje mikroprocesor Atmel AT90USB1287 so zapojením USB zbernice s mini USB konektorom, vyvedenými portmi mikroprocesora a tlačidlami pre prechod do programovacieho režimu.



Obr. 4.1: AT90USBKey [4]

- AT90USB1287: mikroprocesor Atmel AT90USB1287
- USB MiniAB: Mini USB konektor
- PORT A-E: vyvedené porty mikroprocesora
- LEDs : dve dvojfarebné LED diódy
- RST: resetovacie tlačidlo
- HWB: tlačidlo pripojené na jeden so vstupov mikroprocesora
- JTAG: programovací interface JTAG
- Data Flash: flash pamäť
- Joystick: štvorsmerový joystick pripojený na vstupy mikroprocesora
- Ext. Power: externé napájanie

Ďalšou z výhod použitého AT90USBKey je malá veľkosť modulu, vďaka vysokej hustote súčiastok umožnenou vďaka profesionálnemu vyhotoveniu dosky plošného spoja.

Výstupné napájacie napätie pre modul SMBus zbernice je 3.3V. Zbernica SMBus je pripojená na výstup mikroprocesora PD0, ktorý má funkciu časovania dvojvodičovej zbernice a výstup PD, ktorý slúži ako dátový vodič dvojvodičovej zbernice.

4.2 Programovanie mikroprocesora Atmel AT90USB1287

Programovanie mikroprocesora Atmel AT90USB1287 sme vykonávali vo vývojovom prostredí Bascom-AVR. Samotnú USB komunikáciu zabezpečuje komerčné rozšírenie prostredia „BASCOM-AVR - USB Add On“. Toto rozšírenie sa prostredníctvom svoje knižnice „USBINC.lib“ stará o riadenie USB komunikácie, preto v samotnom programe mikroprocesora už túto komunikáciu netreba riešiť, ale stačí priamo volať funkcie pre čítanie a zápis na USB rozhranie.

Samotný program mikroprocesora sa skladá zo štyroch hlavných častí. Prvou časťou je konfigurácia USB komunikácie, kde sa nastavia rôzne parametre, ako VID a PID zariadenie, trieda USB komunikácie, počet bajtov komunikácie atď. Ďalšou časťou je načítanie prijatých údajov z USB zbernice. Tieto údaje sú spracované a sú zavolané požadované funkcie, ktorých výstup sa následne posiela späť na USB zbernicu.

Konfigurácia parametrov USB komunikácie:

```
Const Vendor_id = &H03EB
Const Product_id = &H2022
'specific device constants
Const Ep_control_length = 32
Const User_conf_size = 41
Const Size_of_report = 53
Const Device_class = 0
Const Device_sub_class = 0
Const Device_protocol = 0
Const Release_number = &H1000
Const Length_of_report_in = 8
Const Length_of_report_out = 8
Const Interface_nb = 0
Const Alternate = 0
Const Nb_endpoint = 2
Const Interface_class = 3
Const Interface_sub_class = 0
Const Interface_protocol = 0
Const Interface_index = 0
```

Výpis kódu 4.1: Konfigurácia USB komunikácie

Na výpise kódu 4.1 môžeme vidieť časť konfigurácie USB komunikácie, kde sú parametre ako VID a PID, trieda zariadenia, protokol zariadenia, dĺžka správ atď.

Tieto parametre sú už nastavené do základného režimu vo vzorovom programe priamo dodaného k rozšíreniu Bascom-AVR USB Add-on. Preto tieto nastavenia už nie je potrebné meniť, je však dobré zmeniť hodnoty VID a PID, aby nenastal konflikt pri prípadnom pripojení iného USB zariadenia programovaného rovnakým spôsobom. Taktiež je vhodné zmeniť aj názov výrobku pre jednoduchšiu identifikáciu v systéme.

Načítanie údajov z USB zbernice:

```
Sub Hid_task()
    If Usb_connected = 1 Then


       Usb_select_endpoint Ep_hid_out
       If UEINTX.RXOUTI = 1 Then
          Inbyte1 = UEDATX : Print "Got : " ; Inbyte1
          Inbyte2 = UEDATX : Print "Got : " ; Inbyte2
          Inbyte3 = UEDATX : Print "Got : " ; Inbyte3
          Inbyte4 = UEDATX : Print "Got : " ; Inbyte4
          Inbyte5 = UEDATX : Print "Got : " ; Inbyte5
          Inbyte6 = UEDATX : Print "Got : " ; Inbyte6
          Inbyte7 = UEDATX : Print "Got : " ; Inbyte7
          Inbyte8 = UEDATX : Print "Got : " ; Inbyte8
          Usb_ack_receive_out
       End If
```

Výpis kódu 4.2: Načítanie údajov z USB

Pri načítavaní údajov z USB zbernice sa v cykle „Hid_task“ overí, či je USB zariadenie pripojené, následne sa overí, či buffer obsahuje načítaní údaje a či je zápis údajov už dokončený. Na toto slúži premenná „Ueintx.rxouti“. Ak sú údaje pripravené na čítanie, vyčítajú sa postupne jednotlivé byty do určených premenných. Pri vyčítaní bajtov z buffera je potrebné dbať na to, aby počet vyčítaných bytov bol rovnaký, aký sme do zariadenia poslali z hostiteľského zariadenia. V prípade, ak by sa vyčítalo menej bytov ako sa poslalo, zostali by zvyšné byty v bufferi a nesprávne by sa vyčítali pri ďalšom čítaní, ako prvé byty nového čítania. V prípade, ak by sa vyčítalo viac bytov, problém by nastal v prípade, že v bufferi by sa už nachádzalo viac prijatých sekvencií, teda vyčítal by sa z buffera už aj začiatok novej sekvencie a pri ďalšom vyčítaní z buffera by už tento začiatok chýbal. Po vyčítaní údajov sa vráti správa ACK.

Spracovanie načítaných údajov:

```
'citanie SMBus word
If Inbyte1 = 3 Then

   Call Smb_read(Inbyte2)

   Usb_send
End If

'citanie SMBus block
If Inbyte1 = 4 Then

   Call Smb_block_read(Inbyte2)


End If
```

Výpis kódu 4.3: Volanie príslušných funkcií

Po prijatí údajov, sa najprv skontroluje obsah prvého bytu. Tento byte určuje, akú úlohu žiada od mikroprocesora vykonať. Podľa tejto hodnoty sa volajú požadovanie funkcie na čítanie slova z SMBus zbernice, alebo čítanie bloku z SMBus zbernice. Volanie funkcie zároveň odosiela aj hodnotu druhého bytu pre funkciu, ktorá obsahuje príkazový kód požadovanej hodnoty.

```
'citnaie SMBus word
Sub Smb_read(byval Cmd As Byte)

 I2cstart
 I2cwbyte Smb_adress_w
 I2cwbyte Cmd
 I2cstop
 I2cstart
 I2cwbyte Smb_adress_r
 I2crbyte Lsb , Ack
 I2crbyte Msb , Nack
 I2cstop

 Outbyte1 = 3
 Outbyte2 = Cmd
 Outbyte3 = Lsb
 Outbyte4 = Msb

 Waitms 50

End Sub
```

Výpis kódu 4.4: Čítanie slova z SMBus

Samotnú komunikáciu cez I2C/SMBus zbernicu zabezpečuje priamo mikroprocesor Atmel AT90USB1287 prostredníctvom svojo rozhrania dvojvodičovej zbernice. Taktiež samotnú komunikáciu s tým rozhraním zabezpečuje Bascom-AVR pomocou svojich implementovaných knižníc. V samotnom programe nám už stačí volať správnu postupnosť funkcií, podľa druhu požadovanej komunikácie.

Pri čítaní jedného slova sa ako prvé pošle Start bit, následne sa pošle 7 bitová adresa riadiaceho čipu doplnená o bit určujúceho zápis a zapíše sa riadiaci kód hodnoty, ktorú chceme čítať. Čítanie slova sa ukončí Stop bitom. Následne sa znova pošle Start bit, odošle sa 7 bitová adresa s doplnkom pre čítanie, načíta sa spodný byte, vráti sa ACK, horný byte, vráti sa NACK a pošle sa Stop bit. Následne sa ešte počká 50ms, aby v prípade, že riadiaci čip neprijal všetky odpovede a čaká na ne, nemohlo byť začaté ďalšie čítanie. Slave zariadenie na SMBus zbernici v prípade ak do 50ms nedostane odpoveď, zruší aktuálnu komunikáciu a automaticky sa vráti do základného stavu a čaká na začiatok novej komunikácie. Do výstupných bytov sa vloží typ funkcie a riadiaci kód, aby požívateľská aplikácia vedela identifikovať ku ktorej požiadavke údaje patria a načítané dáta vedela spracovať.

Čítanie bloku cez I2C/SMBus zbernicu je podobné ako čítanie slova, problém však je, že vo väčšine prípadov dopredu nepoznáme počet bytov, ktoré blok obsahuje. Pri čítaní teda treba použiť cyklus, ktorého počet opakovaní sa nastaví až v samotnej funkcii. Ako prvé sa totiž posiela počet bytov v bloku, podľa čoho už vieme nastaviť

počet opakovaní cyklu a vyčítať jednotlivé byty bloku. Pri posielaní údajov do počítača sa postupuje mierne odlišným spôsob ako pri posielaní slova, keďže celý blok sa do jednej 8 bytovej sekvencie nezmestí. Ako prvé sa posiela znova číslo funkcie a druhý byte je príkazový kód hodnoty rovnako ako pri posielaní slova. Rozdiel nastáva v treťom byte, do neho sa zapisuje index prvku poľa, ktoré sa posiela a do štvrtého bytu sa zapisuje hodnota tohto prvku. Takýmto spôsobom sa postupne pošlú všetky prvky načítaného poľa.

Odosielanie údajov na USB zbernicu:

```
Sub Usb_send()

   Usb_select_endpoint Ep_hid_in
      If UEINTX.TXINI = 1 Then
         UEDATX = Outbyte1
         UEDATX = Outbyte2
         UEDATX = Outbyte3
         UEDATX = Outbyte4
         UEDATX = Outbyte5
         UEDATX = Outbyte6
         UEDATX = Outbyte7
         UEDATX = Outbyte8
         Usb_ack_fifocon
      End If

End Sub
```

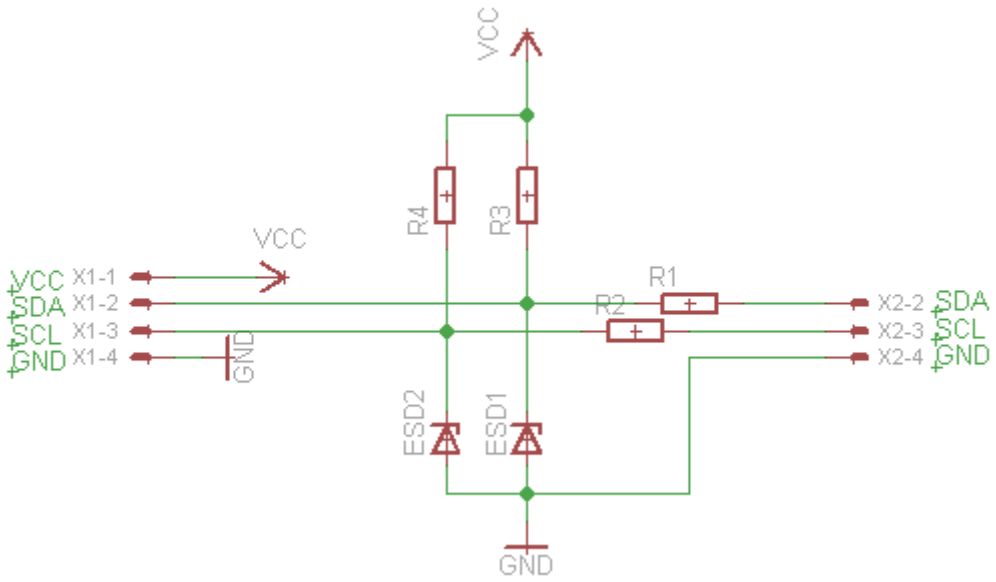
Výpis kódu 4.5: Zápis údajov na USB

Pri odosielaní údajov sa postupuje podobne ako pri ich čítaní. Najprv sa pošle infomácia, že zariadenie je pripravené posielať údaje, následne sa overí, či hostiteľské zariadenie je pripravení prijímať údaje. Potom sa odosielané údaje zapíšu postupne do buffera. Keď je zápis hotový, pošle sa informácia, že údaje sú pripravené.

4.3 Modul SMBus zbernice

Druhým modulom zariadenia je modul zbernice SMBus. Na tomto module sa nachádzajú pull-up odpory, ochranné odpory a diódy zapojenia SMBus zbernice ako aj napájanie SMBus zbernice s obmedzovačom prúdu pre riadiace čipy bez vlastného zdroja napájania.

Zapojenie SMBus zbernice:

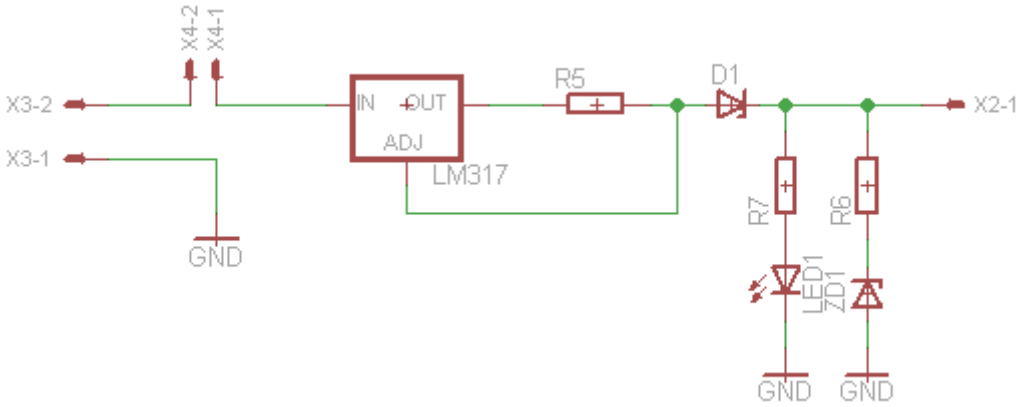


Obr. 4.2: Zapojenie SMBus zbernice

Zapojenie SMBus zbernice obsahuje dva pull-up odpory R3 a R4 s hodnotou 15 kΩ, keďže SMBus zbernica je zapojenie s otvoreným kolektorom. Obsahuje ochranné odpory R1 a R2 s hodnotou 100 Ω a ochranné diódy ESD1 a ESD2. Tieto diódy zvedú prebytočné napätie na zem. Tieto diódy môžu byť 5 V transilové alebo zenerove diódy. Nevýhodou zenerovej diódy je vlastnosť, že pri prepálení sa dióda preruší a zvýšené napätie sa dostane na vstupy zbernice, zatiaľ čo pri prepálení transilovej diódy dôjde ku skratu na zem, teda zvýšené napätie sa nedostane na vstupy zbernice.

Zapojenie napájania SMBus zbernice:

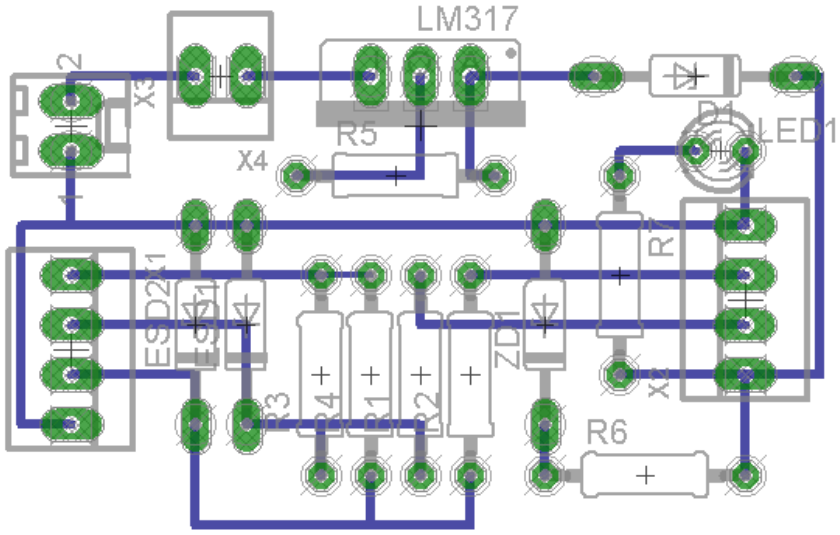
Napájanie SMBus zbernice slúži na napájanie riadiacich čipov, ktoré nemajú vlastné napájanie. Obmedzovač prúdu je súčasťou napájania riadiacich čipov. Riadiace čipy za normálnych podmienok dostávajú napájanie z akumulátora. V prípade, že akumulátor je už zničený a nemá žiadne napätie, alebo je už príliš vybitý a nemá dostatočné napätie pre napájanie riadiaceho čipu, je potrebné riadiaci čip napájať externe. Pri externom napájaní sa však začne nabíjať aj akumulátor. Z toho dôvodu externé napájanie musí obsahovať obmedzovač prúdu, ktorý zabezpečí napájanie riadiacemu čipu beztoho, aby sa akumulátor začal nabíjať.



Obr. 4.3: Zapojenie napájania SMBus zbernice

Napájacia časť sa skladá z regulátora prúdu, ktorý je tvorený regulovateľným stabilizátorom LM317 a odporom R5, ktorým sa nastavuje hodnota maximálneho prúdu. Ďalej napájanie obsahuje diódu D1, ktorá bráni tomu, aby regulátor LM317 nedostával napätie zo vstupnej strany, v prípade, ak riadiaci čip má vlastné napájanie z akumulátora. LED dióda slúži na indikáciu napájania zbernice. Zenerova dióda ZD1 znižuje napätie dodávané akumulátorom, aby sa nezničila LED dióda. Napájanie zbernice je zabezpečené bežnou 9 V batériou typu 6LR61.

Plošný spoj modulu SMBus zbernice:



Obr. 4.4: Plošný spoj modulu SMBus zbernice

4.4 Výsledné osadenie a zapojenie zariadenia

Obr. 4.5: Osadenie zariadenia

Následne sme oba moduly prepojili, pripojili sme konektory a spínače a celé zariadenie sme osadili do krabičky vhodnej veľkosti, na ktorej sme ešte spravili otvory pre konektory USB a SMBus zbernice, LED diódy a spínač napájania SMBus zbernice.

5 Aplikácia na komunikáciu so zariadením

Ďalšou najdôležitejšou časťou práce bolo vytvorenie aplikácie na komunikáciu so zariadením. V tejto aplikácii používateľ vidí načítané údaje z riadiaceho čipu, následne si tieto údaje vie uložiť a v prípade potreby znova prezerať a filtrovať údaje podľa sériového čísla akumulátora.

Pre vytvorenie aplikácie sme si zvoli jazyk Java, pre jeho multiplatformitu, keďže aj zariadenie pracuje v HID režime, teda je aj to multiplatformové.

Samotnú komunikáciu so zariadením v režime USB HID zabezpečuje komunikačná knižnica AtUsbHidJni od Atmelu, preto samotné základy USB komunikácie už pri písaní aplikácie nebolo potrebné riešiť, ale stačilo volať požadované funkcie z knižnice, ako sú pripojenie k USB HID zariadeniu, čítanie údajov zo zariadenia a odosielanie údajov do zariadenia.

5.1 Komunikácia s USB zariadením

Pripojenie USB zariadenia:

```
int state = usbDevice.findHidDevice(vid, pid);

switch (state)
{
case 1: // Find Device
  System.out.println("AtUsbHidSimple.main():Connected to the device");
  btn_Nacitat.setEnabled(true);
  btn_Nacitat_Hlavne.setEnabled(true);
  break;
case -1: // ERROR_USB_DEVICE_NOT_FOUND
  System.err.println("AtUsbHidSimple.main():Usb Device Not Found");
  JOptionPane.showMessageDialog(Frame1,"USB zariadenie nenajdene.","USB
  break;
case -2: // ERROR_USB_DEVICE_NO_CAPABILITIES:
  System.err.println("AtUsbHidSimple.main():Usb no capabilities");
  JOptionPane.showMessageDialog(Frame1,"USB zariadenie nie je pripraven
  break;
```

Výpis kódu 5.1: Pripojenie USB zariadenia

Pripojenie USB HID zariadenia je realizované volaním funkcie „findHidDevice()“, v ktorom sa odovzdávajú parametre VID a PID zariadenia. Táto funkcia po pripojení zariadenia vracia hodnotu stavu, teda či zariadenie bolo pripojené, alebo chybový kód. Podľa tejto hodnoty je možné následne vykonať ďalšie funkcie, ako výpis chybovej hlášky, alebo aktiváciu prvkov pre komunikáciu so zariadením.

Konkrétne na výpise kódu 5.1 môžeme vidieť, ako sa po vrátení stavu 1, čo znamená úspešné pripojenie, aktivujú tlačidlá na čítanie údajov zo zariadenia. V prípade vrátenia stavu -1, čo znamená, že USB zariadenie s požadovaným VID a PID nebolo nájdené, sa objaví chybové okno s hlásením „USB zariadenie nenájdené“.

Odosielanie údajov USB zariadeniu:

```
writeByteArray[0] = 3;
writeByteArray[1] = 0x1c;
usbDevice.writeData(writeByteArray);
```

Výpis kódu 5.2: Odosielanie údajov

Pre odosielanie údajov na USB zariadenie je vytvorené pole typu Byte s potrebným počtom prvkov. Do tohto poľa sú vložené požadované údaje a následne je toto pole odoslané USB zariadeniu volaním funkcie „writeData()“ a parametrom na toto pole.

Na výpise kódu 5.2 vidíme vloženie hodnoty 3 do prvého prvku poľa, čo znamená pre naše USB zariadenie, že od neho budeme požadovať čítanie slova. Do druhého prvku poľa je vložená hodnota príkazového kódu 0x1C požadovanej hodnoty.

Prijímanie údajov z USB zariadenia a ich spracovanie:

```
readByteArray = usbDevice.readData();
```

Výpis kódu 5.3: Prijímanie údajov

Pri prijímaní údajov z USB zariadenia sa do poľa typu Byte uložia údaje z USB zariadenia zavolaním funkcie „readData()“.

```
if(readByteArray.length != 0)
{

    //citanie slova
    if(readByteArray[0] == 3)
    {

        if(readByteArray[1] == 0x09)
            {
                p_napatie = (((float) readByteArray[2] ) + (((float) readByteArray[3] )*256))/
                System.out.println("Napatie je: " + p_napatie + " V");
                s_napatie = String.format("%.2f" , p_napatie);
                t_napatie.setText(s_napatie + " V");
            }


        if(readByteArray[1] == 0x08)
            {
                p_teplota = (float) ((((float) readByteArray[2] ) + (((float) readByteArray[3]
                System.out.println("Teplota je: " + p_teplota + " °C");
                s_teplota = String.format("%.2f" , p_teplota);
                t_Teplota.setText(s_teplota + " °C");
            }
```

Výpis kódu 5.4: Spracovanie načítaných údajov

Po načítaní údajov z USB zariadenia sa ako prvé overí, či boli údaje načítané. Ak boli, skontroluje sa hodnota prvého prvku, ak je hodnota 3, znamená to, že načítané údaje sú odpoveď na žiadosť o čítanie slova. V druhom prvku na nachádza príkazový kód vracanej hodnoty. V treťom prvku sa nachádza spodný byte slova a vo štvrtom prvku sa nachádza horný byte slova.

Tieto údaje je potrebné následne vhodne upraviť. Hodnotu horného byte je potrebné vynásobiť hodnotou 256 a následne k nej pričítať hodnotu spodného bytu. Zároveň tieto hodnoty je potrebné pretypovať podľa typu výslednej premennej, do ktorej sa ukladajú a hodnotu upraviť do požadovanej fyzikálnej jednotky, napríklad napätie sa udáva v mV a teplota sa udáva v °K. Tieto jednotky je vhodné upraviť na V a °C.

Po spracovaní údajov je potrebné ich zobraziť pre používateľa vo vhodnej forme. V našej aplikácii je toto zobrazovanie riešené výpisom údajov do textový polí. Tieto textové polia vedia zobrazovať premenné typu String, preto je potrebné uložené údaje pretypovať do typu String. Pretypovanie sa môže urobiť priamo pri zápise do textového poľa, alebo pretypované údaje uložiť do premennej typu String a tú následne vložiť do textového poľa. My sme si zvolili uloženie do premennej, aby sa údaje dali následne použiť pri ukladaní do výstupného súboru bez potreby opakovaného pretypovania.

5.2 Uloženie údajov do externého súboru

Načítané údaje z USB zariadenia je možné uložiť do externého súboru pre ďalšie spracovanie a ich opätovné prezeranie.

Súbory sa ukladajú do CSV súboru. CSV súbor je textový súbor obsahujúci hodnoty oddelené čiarkou. V prvom riadku súboru sa môže nachádzať hlavička stĺpcov, ale nie je to potrebné, záleží od využitia súboru.

```
boolean alreadyExists = new File(file).exists();

FileWriter writer = new FileWriter(file, true);

if (!alreadyExists)
{
    writer.write("serial");
    writer.write(';');
    writer.write("cas");
    writer.write(';');

    writer.write("dkapacita");
    writer.write(';');
    writer.write("zoskapacita");
    writer.write(';');

    writer.write("mennapatie");
    writer.write(';');
    writer.write("napatie");
     writer.write(';');
```

Výpis kódu 5.5: Zápis do súboru

Zápis do CSV súboru prebieha pomocou štandardnej funkcie jazyku Javy „FileWriter“. Ako prvé sa overí, či súbor s daným menom už existuje, následne sa súbor otvorí. Ak takýto súbor ešte neexistuje, pri požiadavke o otvorenie sa automaticky vytvorí. Ak súbor neexistoval, ako prvé zapíše sa hlavička stĺpcov. Následne sa zapíšu uložené údaje. Všetky hodnoty sú oddelené znakom „ ; “

Výber umiestnenia a názvu súboru je realizovaný pomocou funkcie „JfileChooser“, čo je štandardná funkcia jazyka Java pre vyvolanie výberového okna súboru. Súbor je automaticky vytvorený koncovkou „.csv“ čo zabezpečuje filter výberového okna súboru, ktorý za zadané meno súboru automaticky doplní „.csv“.

5.3 Načítanie a filtrovanie CSV súboru

Načítavanie údajov z CSV súboru sme v našej aplikácii riešili pomocou voľne dostupnej knižnice OpenCSV, keďže pri načítavaní súboru môžu nastať pri bežnom textovom čítaní rôzne problémy pri načítaní nesprávneho súboru.

```
CSVReader reader;
try {
    reader = new CSVReader(new FileReader(file), ';', '\t', 1);

String [] nextLine;
int i = 0;


while ((nextLine = reader.readNext()) != null) {
    System.out.println("Data: " + nextLine[0] + " " + nextLine[1
    data[i][0] = nextLine[0];
    data[i][1] = nextLine[1];
    data[i][2] = nextLine[2];
    data[i][3] = nextLine[3];
    data[i][4] = nextLine[4];
    data[i][5] = nextLine[5];
    data[i][6] = nextLine[6];
    data[i][7] = nextLine[7];
    data[i][8] = nextLine[8];

    i++;

}
```

Výpis kódu 5.6: Načítanie CSV súboru

Pri čítaní je pomocou funkcie „FileReader“ otvorený požadovaný súbor. Následne sú do pripraveného dvojrozmerného poľa postupne načítané jednotlivé prvky aktuálneho riadku. Riadky sa postupne posúvajú, až kým cyklus nepríde na koniec súboru.

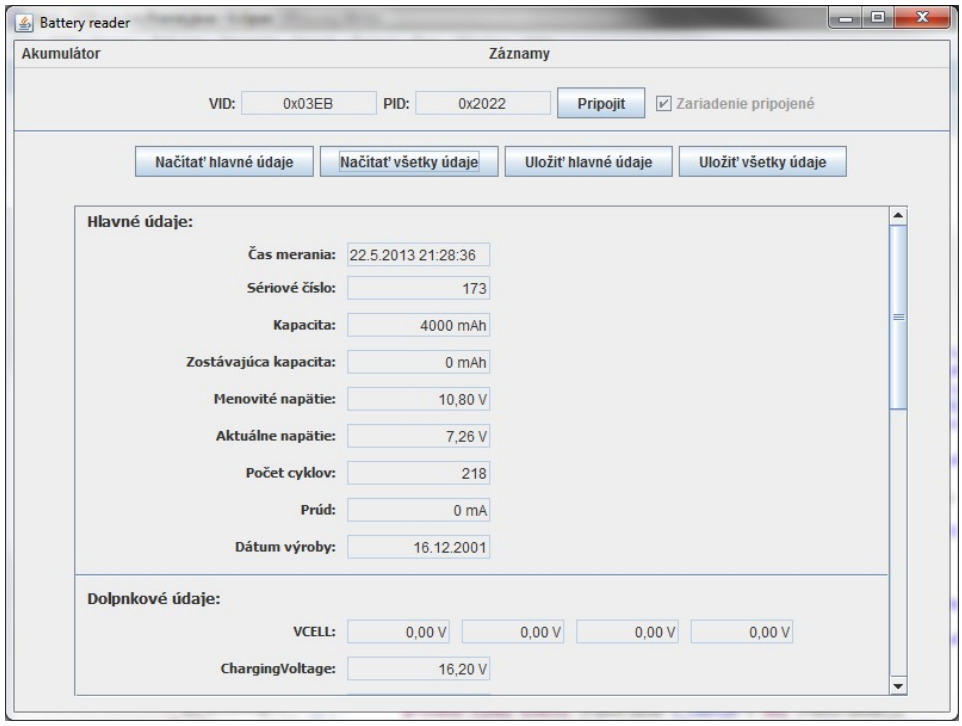
5.4 Spustenie aplikácie

Pre spustenie aplikácie v jazyku Java je potrebné mať v operačnom systéme mať nainštalovanú podporu Java. Túto podporu je možné stiahnuť na stránke http://www.java.com/en/download/ a je taktiež súčasťou Prílohy A, pre operačný systém Microsoft Windows, Linux a Mac OS X.

6 Výsledný produkt diplomovej práce

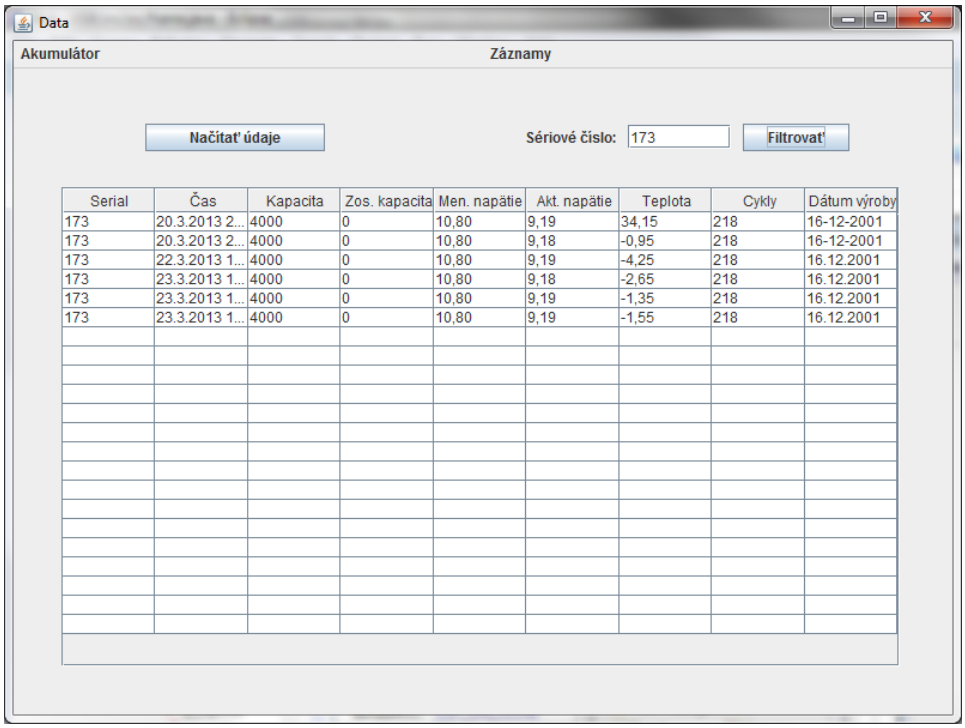
Ako výsledný produkt našej diplomovej práce sme dostali požadovaný SMBus Interface pre komunikáciu s inteligentnými akumulátormi. Zariadenie sa pripája pomocou USB zbernice a pracuje v režime USB HID, teda nepotrebuje žiadne dodatočné ovládače, ale používa štandardné ovládače operačných systémov na rozdiel od hotových . Zariadenie je vďaka svojmu HID režimu schopné pracovať pod dnešnými bežnými operačnými systémami.

Súčasťou produktu je aj vlastná komunikačná aplikácia, vďaka ktorej vie používateľ pracovať so zariadením, vidí načítané údaje z akumulátora, ktoré si vie následne uložiť do CSV súboru, ktorý je možné ďalej spracovávať v rôznych aplikáciach, napríklad Microsoft Excel alebo Open Office Calc. Tieto údaje je možné znovu otvoriť aj v aplikácii pre ich opätovné prezeranie. Aplikácia je písaná v jazyku Java, vďaka ktorej aplikácia dokáže taktiež fungovať pod rôznymi operačnými systémami.



Obr. 6.1: Obrazovka s načítanými údajmi

Obrázok 6.1 zobrazuje hlavné okno aplikácie s načítanými údajmi z akumulátora a riadiacimi prvkami ako sú pripojenie zariadenia, načítanie hlavných a všetkých údajov a ich uloženie.



Serial	Čas	Kapacita	Zos. kapacita	Men. napätie	Akt. napätie	Teplota	Cykly	Dátum výroby
173	20.3.2013 2...	4000	0	10,80	9,19	34,15	218	16-12-2001
173	20.3.2013 2...	4000	0	10,80	9,18	-0,95	218	16-12-2001
173	22.3.2013 1...	4000	0	10,80	9,19	-4,25	218	16.12.2001
173	23.3.2013 1...	4000	0	10,80	9,18	-2,65	218	16.12.2001
173	23.3.2013 1...	4000	0	10,80	9,19	-1,35	218	16.12.2001
173	23.3.2013 1...	4000	0	10,80	9,19	-1,55	218	16.12.2001

Obr. 6.2: Obrazovka s filtrovanými záznamami

Na obrázku 6.2 je zobrazené okno s údajmi ktoré boli opätovne načítané z externého súboru a filtrované podľa sériového čísla akumulátora.

Na obrázku 6.3 sa nachádza samotné zariadenie USB/SMBus Interface. Na prednom paneli sa nachádzajú LED diódy na indikáciu pripojenia USB zbernice a prítomnosti napájacieho napätia na SMBus zbernici. Ďalej sa na paneli nachádza tlačidlo na zapnutie napájania pre SMBus zbernicu z externého zdroja, ktorý v tomto prípade predstavuje 9 V batéria typu 6LR61. Konektory pre pripojenie USB a SMBus zbernice sa nachádzajú na vrchnej strane zariadenia.

Obr. 6.3: USB/SMBus Interface

Výhodou nášho zariadenia oproti bežne predávaným hotovým riešeniam pre komunikáciu USB/I2C je, že zariadenie pracuje v režime USB HID, teda nepotrebuje žiadne dodatočné ovládače, ako bežné riešenia. Taktiež vďaka mikroprocesoru Atmel AT90USB1287 je možné ďalej rozširovať možnosti nášho zariadenia, teda je ľahko modifikovateľné aj pre prípadné nasadenie v oblasti automatizácie.

Záver

V prvej kapitole našej diplomovej práce sme sa zamerali na popis oblastí a možností, ktoré pri návrhu zariadenia treba zvážiť a vybrať si z nich pre naše zariadenie tú najvhodnejšiu podľa požiadaviek na zariadenie. Ďalej sme sa podrobne venovali komunikácii cez protokol I2C a SMBus, kde sme popísali ich fungovanie, špecifikácie, spôsoby komunikácie. V prvej kapitole sme sa taktiež venovali podrobne riadiacim čipom typu BQ2040.

V druhej kapitole sme sa venovali výberu komunikácie zariadenia s počítačom. Vybrali sme si komunikáciu cez USB HID zariadenie, keďže tento spôsob komunikácie nepotrebuje dodatočné ovládače do operačného systému. Ďalej sme sa v kapitole venovali výberu vhodného spôsobu riadenia zariadenia. Ako najvhodnejšiu možnosť sme zvolili výber programovateľného mikroprocesora. Vzhľadom na USB komunikáciu s počítačom sme si zvolili mikroprocesor Atmel AT90USB1827. V poslednej časti druhej kapitoly popisujeme spôsob programovania I2C/SMBus komunikácie a USB komunikácie cez vývojové prostredie Bascom AVR od spoločnosti MCS Electronics.

V tretej kapitole sme sa venovali návrhu samotného zariadenia podľa získaných poznatkov. Kapitola obsahuje návrh schémy zapojenia výsledného zariadenia a popis jeho fungovania.

V štvrtej kapitole sme venovali postupu tvorby samotného zariadenia. Popisu výsledného zapojenia a popisu jednotlivých častí programu mikroprocesora ako sú komunikácia cez SMBus a USB zbernicu.

V piatej kapitole popisujeme komunikačnú aplikáciu a jej jednotlivé časti ako komunikáciu s USB zariadením, ukladanie načítaných hodnôt a ich opätovné otvorenie pre prehliadanie a filtráciu.

Ako výsledný produkt našej diplomovej práce sme dostali požadovaný SMBus Interface pre komunikáciu s inteligentnými akumulátormi. Zariadenie sa pripája pomocou USB zbernice a pracuje v režime USB HID, teda nepotrebuje žiadne dodatočné ovládače, ale používa štandardné ovládače operačných systémov.

Súčasťou produktu je aj vlastná komunikačná aplikácia, vďaka ktorej vie používateľ pracovať so zariadením, vidí načítané údaje z akumulátora, ktoré si vie následne uložiť do CSV súboru.

Bibliografické odkazy

[1] Atmel Corporation. 1.2010. *AVR315: Using the TWI module as I2C master* Dostupné z: <http://www.atmel.com/Images/doc2564.pdf> [cit. 2012-10-20].

[2] Atmel Corporation. 10.2005. *AVR316: SMbus Slave Using the TWI Module.* Dostupné z: <http://www.atmel.com/Images/doc2583.pdf> [cit. 2012-10-20].

[3] Atmel Corporation. 4.2012. *AT90USB64/128 Datasheet* Dostupné z: <http://www.atmel.com/Images/doc7593.pdf> [cit. 2012-11-8].

[4] Atmel Corporation. 4.2006. *AT90USBKey Hardware User Guide* Dostupné z: <http://www.atmel.com/Images/doc7627.pdf> [cit. 2012-11-10].

[5] Maxim integrated. 2012. *Comparing the I²C Bus to the SMBus* Dostupné z: <http://www.maximintegrated.com/app-notes/index.mvp/id/476> [cit. 2012-11-15].

[6] MCS Electronics. 2000. *AN #139 - Using MCP23016* Dostupné z: <http://www.mcselec.com/index.php?option=com_content&task=view&id=107&Itemid=57> [cit. 2012-11-16].

[7] MCS Electronics. 2012. *BASCOM-AVR* Dostupné z: <http://www.mcselec.com/index.php?option=com_content&task=view&id=14&Itemid=41> [cit. 2012-10-15].

[8] MCS Electronics. 2012. *BASCOM-AVR Help: CONFIG USB* Dostupné z: <http://avrhelp.mcselec.com/config_usb.htm> [cit. 2012-11-17].

[9] Texas Instruments. 6.1999. BQ*2040 Datasheet* Dostupné z: <http://www.ti.com/lit/ds/symlink/bq2040.pdf> [cit. 2012-10-15]

Slovník termínov

Zbernica - je elektrický vodič, alebo sústava vodičov, ktoré prepájajú viaceré elektrické či elektronické zariadenia a slúžia na vzájomný prenos energie, alebo údajov vo forme elektrických impulzov medzi nimi.

UART - Universal Asynchronous Receiver/Transmitter – univerzálny asynchrónny prijímač/vysielač, rozhranie pre sériovú komunikáciu.

TTL logika – je štandardom, používaným v digitálnych obvodoch. Obvody TTL používajú napájacie napätie 4,5 až 5,5 V.

RS-232 – komunikačné rozhranie, umožňuje vzájomnú sériovú komunikáciu dvoch zariadení. Pracuje s napätím od -15 do +15 V, kde na dátovej komunikácii logická 0 predstavuje úroveň +3 až +15 V a logická 1 úroveň -3 až -15 V.

TWI - two-wire interface - rozhranie komunikujúce po dvoch vodičoch, teda dvojvodičová zbernica, ako sú zbernica I2C a SMBus.

I2C – Dvojvodičová master-slave zbernica.

SMBus – Dvojvodičová zbernica odvodená od zbernice I2C.

USB – Universal serial bus – univerzálna sériová zbernica, súčasné štandardné pripojenie periférnych zariadení k PC.

Generic HID – Generic Human Interface Device – sú to základné USB zariadenia, ktorých komunikácia je priamo špecifikovaná v štandarde USB, preto operačné systémy s nimi dokážu komunikovať pomocou svojich zabudovaných ovládačov a nie sú potrebné doplnkové ovládače od výrobcov týchto zariadení.

Master zariadenie - hlavné zariadenie na zbernici, ktoré začína a riadi komunikáciu.

Slave zariadenie - zariadenie na zbernici, ktoré iba čaká na požiadavku od master zariadenia a následne na túto požiadavku odpovedá.

Zoznam príloh

Príloha A: CD s elektronickou verziou diplomovej práce, komunikačnou aplikáciou, zdrojovými kódmi komunikačnej aplikácie a programu mikroprocesora a podporou jazyka Java