

TARTU ÜLIKOOL

LOODUS- JA TEHNOLOOGIATEADUSKOND

Füüsika Instituut

Madis Sepp

Juhtprogrammi ning riistvara projekteerimine täisautonoomsele
LPC1768 ARM Cortex-M3 protsessoril baseeruvale
andmehõivemoodulile

Bakalaureusetöö Arvutitehnikas (12 EAP)

Juhendaja: Fred Valk, MSc

Kaitsmisele lubatud

Juhendaja

allkiri, kuupäev

Tartu 2012

Sisukord

Tähised, lühendid ja definitsioonid.....	3
1. Sissejuhatus.....	6
2. Ülevaade manussüsteemidest ja ARM arhitektuurist	8
2.1 Ülevaade manussüsteemidest	8
2.2 ARM arhitektuur ja käsustik.....	10
2.3 Cortex-M3 eripärad.....	12
3. NXP LPC1768 protsessori ülevaade.....	13
3.1 NXP LPC1768 protsessori sisendid ja väljundid.....	13
3.2 Erinevad energiasäästurežiimid	13
3.3 Mälud ja registrid.....	14
3.4 RTC katkestuste kasutamine protsessori töörežiimi valikul.....	15
4. Projekteerimine.....	16
4.1 Tehnilised nõuded.....	16
4.2 Funktsionaalsus.....	16
4.3 Ülevaade erinevatest akutüüpidest.....	17
4.4 NiMH aku laadumine ja tühjenemine	18
5. Juhtmooduli sidestamine.....	20
5.1 Analoo-digitaalmuundurite (A/D-muundurite) sidestamine.....	20
5.2 Digitaalsete sisendite sidestamine	23
5.3 Digitaalsete väljundite sidestamine	24
5.4 Digitaalse releeväljundi sidestamine	25
6. Süsteemi toide.....	27
6.1 Andmehõivemooduli toide vahelduvvooluvõrgust.....	27
6.2 Andmehõivemooduli toide tagavaraakult.....	28
7. Andmehõivemooduli tarkvara	29
Kokkuvõte.....	34
Summary	35
Viidatud allikad.....	36
Lisad.....	37

Tähised, lühendid ja definitsioonid

ADC (ingl. Analog-to-Digital converter)	- Analoog-digitaalmuundur e A/D-muundur.
ALU (ingl. Arithmetic Logic Unit)	- Aritmeetika-loogika üksus.
ARM (ingl. Advanced RISC Machine)	- Edasijõudnud lihtsustatud instruksioonidega arvuti.
CAN (ingl. Controller Area Network)	- Multiprotsessorsüsteemis kasutatav andme-sidevõrk, mis on loodud kasutamiseks ühe masina ulatuses.
CISC (ingl. Complex Instruction Set Computer)	- Keeruliste instruksioonidega arvuti.
CPU (ingl. Central Processing Unit)	- Keskprotsessor.
DMA (ingl. Direct Memory Access)	- Otsemällupöördus. DMA kanalit kasutatakse suure kiirusega andmevahetuseks mälu ja välisseadmete vahel ilma arvuti protsessori osavõtuta.
Välkmälu (ingl. Flash Memory)	- Liik püsिमälusid. Võimaldavad kirjutust arvuti töö käigus ja säilitab informatsiooni ka siis, kui seadme toide on välja lülitatud.
GPIO (ingl. General-Purpose I/O)	- Üldotstarbeline sisend/väljund.
I/O (ingl. Input/Output)	- Sisend/Väljund.
I ² C (ingl. Inter-Integrated Circuit)	- Kahejuhtme liides, kasutatakse madala andmevahetuskiirusega seadmete ühendamiseks.
i ² S (ingl. Integrated Interchip Sound)	- Integreeritud kiibisisene heliliides, mida kasutatakse digitaalsete heliseadmete ühendamiseks.
IRC (ingl. Internal RC (Oscillator))	- Sisemine takistist ja kondensaatorist koosnev takti generaator.

ISR (ingl. Interrupt Service Routine)	- Katkestusi teenindav programm.
Kõrge ja madal olek (ingl. high/low state)	- Signaal on kõrge, kui selle olek on 1 (tõene), ja madal, kui see on 0 (väär). Kui mingil signaalil on alumine nivoo (näiteks 0V), nimetatakse seda madalaks, ja kui sellel on ülemine nivoo (näiteks 10V), nimetatakse seda kõrgeks. Kui signaal ei oma kumbagi kirjeldatud olekutest, siis öeldakse, et signaal „ujub“ st tema olek on määramata.
LDO (ingl. Low Dropout Voltage)	- Madal pingelang.
lipp (ingl. flag)	- 1 bitine muutuja, millel on 2 olekut 1 ja 0, kus 1 on tõene ja 0 on väär. Lipp võib näidata staatust, kontrollida mingit protsessi või teha seda mõlemat. Näiteks mingile lipule väärtuse 1 omistamine võib avada ukse (kontrollib). Mingi lipp võib saada väärtuseks 1, kui õues sajab vihma, midagi otseselt korda saatmata (staatus).
MCU (ingl. Microcontroller Unit)	- Mikrokontroller.
NiMH (ingl. Nickel–Metal Hydride)	- Nikkel-metallhüdriid.
OTG (ingl. On-The-Go)	- Laiendus USB 2.0 standardile välisseadmete teineteisega ühendamiseks. USB OTG tooted suudavad teineteisega suhelda ilma PC abita.
PMOP	- p-kanaliga metalloksiid pooljuht.
RISC (ingl. Reduced Instruction Set Computer)	- Lihtsustatud instruksioonidega arvuti.
RTC (ingl. Real-Time Clock)	- Reaalajaline kell.

- SPI (ingl. Serial Peripheral Interface) - Lisaseadmete jadaliides. Sünkroonne ühepolaarne andmevahetus, kus on võimalik samaaegne andmete saatmine ja vastuvõtmine. Mitme seadmega suhtlemisel kasutatakse lisaks valikusignaale.
- SRAM (ingl. Static Random-Access Memory) - Staatileine juhupöördusega mälu.
- SSP (ingl. Synchronous Serial Port) - Sünkroonne jadaliides.
- Thumb/Thumb-2 käsustik - ARM protsessorite jaoks loodud käsustik, mis koosneb nii 16-bitistest kui 32-bitistest käskudest.
- TWI (ingl. Two Wire Interface) - Kahejuhtmeline liides.
- UART (ingl. Universal Asynchronous Receiver/Transmitter) - Universaalne asünkroonne saatja-vastuvõtja. Asünkroonne ühepolaarne andmevahetus, võimalik on samaaegne saatmine ja vastuvõtmine. Standard on täiendava elektroonika abiga võimalik ümber muuta RS-232 standardiks, mis vastab jadaväratile (ingl. Serial port).

1. Sissejuhatus

Arvutusvõimsuse kiire kasv ja erinevate tarkvara- ja riistvaraplatvormide laialdane levik on loonud olukorra, kus klassikalist andmehõivemoodulit saab kergesti luua suvalise manussüsteemides kasutatava kontrolleri baasil. Sellise lahenduse eeliseks on asjaolu, et mõõdetavat andmevoogu saab suunata läbi mõne liidese edasi kohta, kus andmete salvestamine, edasine töötlemine ja esitamine on otstarbekam kui andmehõivemoodulis endas. Klassikaliselt salvestatakse mõõdetavad andmed mõtteseadmises endas või edastatakse läbi USB või RS232 liidese arvutisse, mis lugemite võtmise ajal peab pahatihti kogu aeg töötama. Antud töös uuritakse võimalust mõõdetavate andmete (kohest) edastamist serverisse, mis juba ise tegeleb andmete salvestamise ja visualiseerimisega.

Käesoleva bakalaureusetöö eesmärgiks on projekteerida loodava andmehõivemooduli riistvara sisendite ja väljundite sidestamiseks ning toiteahel, mis koosneb nii vahelduvvooluvõrgu toitest kui ka tagavara aku toitest. Lisaks riistvarale tuleb arendada ka tarkvarapakett, mille ülesandeks on tagada LPC1768 ARM Cortex-M3 protsessoril baseeruva andmehõivemooduli stabiilne ja tõrgeteta töö ning selle külge ühendatud anduritest lugemite võtmine, edasine töötlus ja salvestamine. Oluliseks lisanõudeks on juhtprogrammi kõrge töökindluse tagamine, mis võimaldaks seda kasutada ka tööolukordades, kus andmehõivemoodulile ligipääs on raskendatud. Selliseks kohaks võib näiteks olla ilmajaam mõnel väikesaarel. Tarkvara arendatakse C keeles. Bakalaureusetöö käigus arendatakse välja erinevad programmi moodulid andmevoogude edasiseks töötlemiseks ja salvestamiseks sh liideste automaatne algseadistamine; tarkvara taaskäivitus ja veebikliendi tugi mõõdetavate andmete serverisse edastamiseks. Analoo-digitaalmuunduritest mõõdetavate andmete töötlemisel kasutatakse digitaalseid filtreid mõõdetavate andmete kvaliteedi ja mürakindluse tõstmiseks.

Andmete automaatne ülekanne välisesse serverisse toimub läbi 3G raadiomodemi. Programmi osad, mis sõltuvad palju konkreetsest kasutatavast riistvarast st andurit tüübist,

luuakse viisil, mis võimaldab konkreetse rakenduse hiljem juurde programmeerida, kasutades juba varem loodud liidese definitsioone.

Erinevate andmehõivemoodulite automaatseks eristamiseks ja häälestamiseks toimub tarkvara konfigureerimine serverist. Kuna andmehõivemoodul võib ise paikneda tulemüüri taga, siis kogu süsteemi konfigureerimine käib serverist üksiku andmehõivemooduli algatusel teatud ajaintervalliga.

Töö esimeses osas antakse ülevaade manussüsteemides kasutatavatest riistvara- ning tarkvaralahendustest. Seejärel tutvustatakse ARM tehnoloogiat ja vastavat käsustikku. Edasi antakse ülevaade juba konkreetselt antud töös kasutatava LPC1768 ARM Cortex-M3 protsessori ehitusest ja võimalustest ning esitatakse nõuded projekteeritavale seadmele. Lisaks tutvustatakse erinevaid aku tehnoloogiaid ning valitakse projekti jaoks sobivaim. Edasi jätkatakse riistvara disainimisega loodavale andmehõivemoodulile. Loodava tarkvara arendust alustatakse üksikute liideste jaoks vajalike meetodite kirjutamisest. Lisaks luuakse meetodid, mis tagavad süsteemi stabiilse töö ka siis, kui mõni liides või tööparameeter on vigane. Näiteks salvestatakse mõõdetavad andmed kohalikus mälus, kui serveriühendus puudub. Serveriühenduse tekkimisel sünkroniseeritakse serveriga ka ühenduse katkestuse ajal mõõdetud andmed. Viimases peatükis uuritakse ja pakutakse välja lahendused kuidas välistada andmete ülekandmisel võimalikud tekkivad vead ning esitatakse algoritm tarkvara automaatseks uuendamiseks serverist. Lisaks uuritakse võimalusi vähendamaks riski, et keegi kolmas osapool saaks serverisse valeandmeid logida või muul moel süsteemi tööd häirida.

2. Ülevaade manussüsteemidest ja ARM arhitektuurist

2.1 Ülevaade manussüsteemidest

Manussüsteem on arvuti, mis on mingi seadme lahutamatu integreeritud funktsionaalne osa, mis täidab kindlat ülesannet. Manussüsteem erineb tavapärasest üldotstarbelisest arvutist kasutusviisilt. Kui üldotstarbelise arvuti kasutusfunktsiooni määrab arvuti kasutaja, siis manussüsteem täidab aga ainult kindlat funktsiooni, on ehitatud seadme sisse ja enamasti ei ole see tavakasutajale märgatav. Sellised seadmed võivad olla erinevad juhtsüsteemid, näiteks automootori juhtimine, auto pidurite juhtimine, pesumasina juhtsüsteem jne. [1]

Arhitektuurilt ei erine üldotstarbelised süsteemid manussüsteemidest kuigi palju. Manussüsteemide puhul on enamjaolt tegemist kas mikroprotsessorite või spetsiaalselt valmistatud kiipidega (ASIC - *application specific integrated circuit*), mis on vahetult ühendatud juhitava või jälgitava objektiga. Antud süsteemide üks olulisemaid omadusi on odav massitootmine. Süsteem peab olema oma ülesande jaoks võimalikult lihtne, et seda oleks odavam toota. See on ka põhjus, miks autotööstus püüab siiani läbi saada 8 ja 16 bitiste protsessoritega. [1]

Üks oluline manussüsteemi aspekt on turvalisus. Süsteemi turvalisus tähendab seda, et süsteem ei too endaga kaasa riske. Riskiks loetakse sündmusi või olekuid, mis ei ole soovitud selle süsteemi poolt. Ideaalne süsteem oleks väga turvaline ja ei tooks endaga kaasa ühtegi riski. Reaalsuses on turvaline süsteem selline, mis on põhjalikult üle analüüsitud ja leitud kõik potentsiaalsed riskid, mis süsteemiga tekkida võivad. Kõik leitud riskid on seejärel hinnatud ja teadvustatud ning tehtud kindlaks, et need riskid ei saaks tekitada kahju ei inimestele ega seadmetele. Mittesoovitud ja riskitoovateks sündmusteks süsteemis on näiteks mürgiste ainete vabastamine, mitte teatamine tervistkahjustavates olukordades, energia vabastamine või ohtlikes keskkondades eluks vajalike konditsioonide mittehoidmine. [1]

Teine oluline süsteemi aspekt on töökindlus. Tänapäevastelt manussüsteemidelt on võimalik nõuda juba 99.9999%-st töökindlust, mis on vajalik, kui saata süsteem teele mõne

planeedi poole või kui koguda andmeid maailma sügavaimast ookeanist, siis ei saa vea tekkimisel seda kohe tagasi parandusse tuua. Samuti peavad manussüsteemid tihti olema väga vastupidavad. Suured arvutisüsteemid asuvad tavaliselt kvaliteetse energiatarnega konditsioneeritud ruumides, manussüsteemid peavad tihti aga toimima kuumas, räpases, elektriliselt müra, niiskes ja kehva energiatarnega keskkonnas. See aga on aja edenedes toonud kaasa süsteemide järjest keerukamaks muutumise. Kui vanasti sa võisid olla mehaanik kruvikeeraja, mutrivõtme ja haamriga, siis tänapäeval läheb juba lihtsa hoolduse jaoks vaja erinevaid testsüsteeme. Keerukam disain toob endaga enamjaolt kaasa suurema hulga komponente süsteemis, igaühel neist komponentidest on oma tõrkesagedus. Seega iga komponendi tõrkesagedus panustab süsteemi üldisesse töökindlusesse. [1]

Usaldusväärne süsteem saab alguse projekteerimisfaasis, mitte tootmise ettevalmistusfaasis. Projekteerimisfaasis saab läbi mõelda kõik riskikohad ja nendega tegeleda, et süsteem ei reageeriks neile ohtlikult. Hea projekti eesmärk on see, et kõik vead saaks tuvastatud enne, kui seade jõuab tarbijate kätte. Disaini analüüsimiseks kasutatakse tänapäeval statistilisi meetodeid, et hinnata palju vigu süsteemis veel võib olla. Mõningal juhul võib toote välja lasta juba siis, kui hinnatud vigade arv on väiksem eelseatud lävendist. Mõningate süsteemide puhul on oluline näitaja ka keskmine tõrgete vaheline aeg. See oleneb tootest, näiteks signaalide jälgimisel osakeste kiirendis võib seda mõõta sekundite murdosaga, kuid autotootjad annavad mõningatele osadele 100 000 miilise või pikemagi garantii. Ning on ka seadmeid millel on oluline, et tõrge oleks sujuv. Näiteks kui autoga sõites tekib tõrge raadiosüsteemis, pole see eriti ohtlik, kui aga teha möödasõitu veoautost, kui vastassuunas on lähenemas teine veoauto ning selles olukorras mõni kriitilisem süsteem üles ütleb, on aga olukord väga halb. Seega hea disain nõuab, et kriitilistel süsteemidel oleks olemas ka tagavarasüsteem. [1]

2.2 ARM arhitektuur ja käsustik

Igal protsessoril on oma käsustik ja instruksioonide hulk. Käsustikud jaotatakse jämedalt CISC (keeruliste instruksioonidega arvuti) ja RISC (vähendatud instruksioonidega arvuti) tüüpi.

CISC tüüpi käsustik võimaldab keerulisi instruksioone, mis vähendab kirjutatud koodi mahtu ja täitmiseks kuluvat aega. Instruksioonide täitmiseks kulub mitmeid protsessoritsükleid. Keerulised instruksioonid teevad riistvara keerulisemaks ja kallimaks. RISC tüüpi käsustik võimaldab vähem keerulisi instruksioone, mis tähendab seda, et keerulisemate ülesannete täitmiseks tuleb kasutada mitut instruksiooni. Katkestustel tekkivad viivitused on võrreldes CISC käsustikuga väiksemad, sest lihtsad instruksioonid täidetakse tavaliselt ühe protsessori tsükliga. Lihtsamad instruksioonid muudavad samas riistvara palju lihtsamaks ja odavamaks. [2]

Olgu protsessor 8- või 32-bitine, keerulise või lihtsa käsustikuga, peab sellel olema ligipääs mälule. Igal protsessoril peab olema püsimälu, kus paikneb programm ning konstantsed andmed ning muutmälu, kus hoitakse ajutisi andmeid programmi töö täitmisel. Muutmälust kustuvad andmed pärast toite eemaldamist ning püsimälus need säilivad. Tuntakse kahte mälusüsteemi arhitektuuri: Harvardi arhitektuur ja Von Neumanni arhitektuur. Harvardi arhitektuuris on koodi- ja andmemälu eraldatud. Näiteks Atmeli AVR ja Microchip PIC 8-bitistel mikrokontrolleritel asub programm väikmäälul ja andmed paiknevad SRAM tüüpi määlul. Selline ehitus lubab lugeda andme- ja koodimälu üheaegselt. Von Neumanni arhitektuuri puhul paiknevad andmed ja programm samal määlul. [2]

Peale koodi- ja andmemälu on igal protsessoril veel registrid. Registrid sisaldavad tavaliselt sama palju bitte, kui lai on protsessori siin. Registrid võivad sisaldada informatsiooni ALU viimase tehte tulemuse kohta või juhtida protsessori tööd – näiteks adresseerimise registrid. Peale spetsiifiliste registrite on protsessoril veel tööregistrid. Tööregistreid kasutatakse kõikvõimalike tehete sooritamisel, sest osad protsessorid on võimelised sooritama tehteid

ainult registritega. Kuna neid on tavaliselt vähe, ei võta nende defineerimine käsus palju ruumi, samas kui andmemälu adresseerimine võtab kordades rohkem bitte.

Nimetus ARM tähendas algselt edasijõudnud vähendatud instruksioonidega arvuti (ingl. *Advanced RISC Machine*). ARM käsustik on 32 bitine, aga lisaks sellele on ka kasutusel *Thumb* käsustik, mis on 16 bitiste käskudega, aga väikesema funktsionaalsusega. See annab juurde suurema kooditiheduse ja efektiivsuse. Kahe erineva käsustiku samaaegne kasutamine tähendab aga seda, et protsessoril on kaks olekut, ühes täidab ARM käsustiku käske ning teises *Thumb* käsustiku käske. Ning koodi täitmise ajal peab ta neid olekuid vahetama, mis võtab lisaega. Lisaks on vaja *Thumb* käsustik ja ARM käsustik kompileerida eraldi, see tegi arendamise keerulisemaks. *Thumb* käsustiku edasiarendus *Thumb-2* on juba sama funktsionaalsusega mis ARM käsustik, ning sisaldab lisaks 16-bitistele käskudele ka 32-bitiseid käske. See tähendas, et protsessor ei pidanud enam olekut vahetama käskude täitmise vahel. [3]

ARM protsessorid jagunevad kolme erineva otstarbega protsessorite perekonda [3]:

ARM-A tüüpi (*ARMv7-A*) on rakenduste (ingl. *Application*) protsessor, mis on loodud tegelema keeruliste ülesannetega nagu näiteks manussüsteemide kõrgematasemeliste operatsioonisüsteemide jooksutamise (nt. Symbian, Linux või Windows Embedded). Tänapäeval leidub A-tüüpi ARM protsessoreid tihti mobiiltelefonide sees.

ARM-R tüüpi (*ARMv7-R*) on reaalaaja (ingl. *Real-time*) protsessor, mis on suunatud suurt jõudlust nõudvatele reaajasüsteemidele. R-tüüpi ARM protsessoreid võib leida kõvaketaste või telefonide kõne/videokõne andmevahetusega tegelevate moodulite seest.

ARM-M tüüpi (*ARMv7-M*) on mikrokontrollerite (ingl. *Microcontroller*) protsessor, mille põhieesmärgid on efektiivsus, odav tükihind, vähene volutarve, kiire katkestustega tegelemise aeg ning kasutamiskihtsus. Käesolevas töös kasutatakse ARM-M tüüpi protsessorit.

2.3 Cortex-M3 eripärad

Cortex-M3 tähtsaim omadus on see, et ta toetab täielikult *Thumb-2* käsustikku ja ei toeta enam ARM käsustikku. See tähendab, et protsessor oskab täita nii 16-bitiseid, kui ka 32-bitiseid käskude vaheldumisi, ilma olekut muutmata, kuid ei ole tagasiühilduv vanema põlvkonna protsessorite jaoks kirjutatud koodiga, kus kasutusel on mõlemad käsustikud. Tänu sellele on kood väga efektiivne ning koodi tihedus on suur. [3]

Cortex-M3 on suure jõudlusega tänu *Thumb-2* käsustikule. Paljud tehted, sealhulgas näiteks korrutamise, võtavad aega ühe tsükli. Nii andmete, kui ka käskude jaoks, on oma siin, mis lubab mõlema poole üheaegselt pöörduda. Lisaks on protsessor madala voolutarbega ning omab erinevaid energiasäästu olekuid. Lisatud on ka erinevaid täiendusi, mis teevad katkestuste töötlemise kiiremaks, näiteks on mõned mitmetsüklilised käsud tehtud katkestatavaks. Varem pidi protsessor kõigepealt lõpetama mitmetsüklilise käsu ning alles siis sai asuda katkestusega tegelema. [3]

3. NXP LPC1768 protsessori ülevaade

3.1 NXP LPC1768 protsessori sisendid ja väljundid

Protsessor NXP LPC1768 omab mitmeid erinevaid sisendeid ja väljundeid [4].

Jadaliidesed:

- 10 Mbps või 100 Mbps Ethernet ühendus;
- USB 2.0 kontrolleri, mis võib töötada seadme, peremehe või OTG režiimis;
- Neli UART ühendust;
- Kahekanaline CAN-võrk;
- Kaks sünkroonset jadapordi kontrolleri;
- SPI kontrolleri samaaegse kahesuunalise andmeedastusega (full-duplex);
- Kolm I²C ühendust;
- I²S ühendus.

Muud välisseadmed:

- 25 üldkasutatavat sisend/väljundviiku;
- 12-bitine A/D muundur, mida saab kasutada kuuel sisendviigul;
- 10-bitine D/A muundur;
- Neli üldkasutatavat taimerit/loendurit;
- Eraldatud voluringiga RTC.

Kasutatud arendusplaadi foto on toodud lisas 1, ning LPC1768 protsessori foto koos väljundviikudega on toodud lisas 2.

3.2 Erinevad energiasäästurežiimid

LPC1768 kontrolleri on energia säästmiseks lisaks mitmetele erinevatele moodulitele, mida eraldi välja lülitada saab, ka neli erinevat energiasäästurežiimi. Nendeks on: unerežiim (ingl. *sleep*), sügav unerežiim (ingl. *deep-sleep*), voolu väljalülitus (ingl. *power-down*) ja sügav voolu väljalülitus (ingl. *deep power-down*). [4]

Unerežiimis ei edastata protsessorile enam takti, aga kõik muud moodulid jäävad tööle. See on kõige suurema energiatarbega energiasäästurežiim, aga samas võtab sellele unerežiimile minek ja sellest väljatulek kõige vähem aega. Unerežiimil olles on koodi täitmine peatatud, kuni katkestuse saabumiseni. [4]

Sügavas unerežiimis lülitatakse välja protsessori taktigeneraator. Mällu jäetakse koodi täitmisesseisu viitaja, registrid ja sisemised SRAM väärtused. Samuti jäävad kehtima ka väljundite väärtused magamamineku hetkest. IRC generaatori väljund keelatakse ära, aga IRC jääb sisselülitatuks, et unerežiimist väljumine oleks kiirem. RTC taktigeneraator jääb tööle ja seega on see ka üks võimalik katkestuse tekitaja protsessori äratamiseks. Kiiremaks ärkamiseks jääb toide sisse ka välmälule. [4]

Voolu väljalülitusrežiimis tehakse sama, mis sügavas unerežiimis ning lisaks lülitatakse välja ka IRC generaator ja välmälu toide. Välmälule saab juurdepääsu 100µs peale unest ärkamist ning kui enne unerežiimi minekut oli IRC kasutusel, siis läheb aega 60µs enne, kui ta edasi takti genereerib. [4]

Sügavas voolu väljalülitusrežiimis lülitatakse vool välja tervest kontrollierist, erandiks on ainult RTC moodul ja taaskäivitamise sisendviik. Sügavast unest saab ärgata taaskäivituse viigu kaudu või RTC võrdluse tekitatud alarmi peale. [4]

3.3 Mälud ja registrid

LPC1768 protsessoril on 512kB välmälu, kus hoitakse koodi ja konstantse väärtusega andmeid. Tegu on säilmäluga ehk seal säilivad andmed ka ilma toiteta. Peale välmälu on antud kontrollieril ka 64kB SRAM mälu, millest 32kB on kasutatav CPU ja DMA poolt ning kaks 16kB mälu, mis kumbki on kasutatav erinevate protsessori väliste seadmete poolt nagu näiteks Ethernet või USB. [4]

Cortex-M3 protsessoril on 16 registrit R0-R15, neist 13 (R0-R12) on üldotstarbelised registrid. R13 on pinuviit, mis viitab kahele pinule, seda kummale ligi pääseda saab, määrab ära kontrollregister. R14 on ühendusregister, kuhu salvestatakse alamprogrammi sisenedes eelmisesse programmi naasmise viit. R15 on käsuloendur, kus salvestatakse

programmi hetkeadress. Seda üle kirjutades saab muuta programmi täitmisjärjekorda. Registrid on 32-bitised ning üldotstarbelised registrid jagunevad kaheks. R0-R7 on madalad registrid, neile saavad ligi nii 16-bitised *Thumb* käsud, kui ka 32-bitised *Thumb-2* käsud. R8-R12 on kõrged registrid, neile saavad ligi kõik *Thumb-2* käsud, aga mitte kõik *Thumb* käsud. Lisaks mainitud registritele on Cortex-M3 protsessoril ka viis eriregistrit. Nendeks on programmi hetkeolukorra register, kus on lipud mis tähistavad negatiivset arvu, nulli, arvutustehete ülekannet/laenamist või ületäitumist. Seller registris on veel kirjas erindi number millega hetkel tegeletakse ning *Thumb* käsustiku tase, mis on ülejääk eelnevast põlvkonnast, kuna antud protsessoris on selle väärtus alati 1 ning selle muutmine põhjustab erindi. [3]

Järgnevad kolm eriregistrit on katkestuste ajutiseks keelamiseks. Viimane eriregister on kontrollregister, kus on kaks lippu. Esimene lipp määrab ära oleku, kas programm töötab kasutaja olekus või eelistatud olekus. Teine lipp määrab ära kumba R13 pinu kasutatakse. [3]

3.4 RTC katkestuste kasutamine protsessori töörežiimi valikul

LPC1768 protsessori RTC on disainitud väga madala voolutarbega, tarbides vähem kui 1 μ A. RTC võimaldab aja mõõtmist ka siis, kui kogu ülejäänud mikrokontroller on välja lülitatud (sügavas voolu väljalülitusrežiimis). RTC tööks vajalik madalaim toitepinge on 2,1V. Taktsagedus 1Hz saadakse madala voolutarbega 32kHz generaatorist. 1Hz takt on kalibreeritud suurima eksimusveaga 1s ühe päeva jooksul konstantse pinge ja temperatuuri juures. Samuti on RTC moodulil väike 20-baidine varuregister andmete hoidmiseks, kui ülejäänud kiip on välja lülitatud. Mooduli katkestuste tähtsaim omadus on see, et ta on võimeline tekitama katkestust, mis äratav protsessori ülesse kõige sügavamast voolusäästmisrežiimist. [4]

4. Projekteerimine

4.1 Tehnilised nõuded

Loodav andmehõivemoodul peab vastama järgmistele tehnilistele nõuetele:

- Andehõivemoodul peab olema universaalne, et teda saaks kasutada paljudes erinevates olukordades. Universaalsus tuleb välja suure sisendpingete vahemikuga (10-24V DC).
- Seadmel peab olema ka oma sisene tagavaruaaku, mis hoiab seadet edasi tööl, ning teavitab serverit toite kadumisest.
- Seadmel peab olema 5 ühenduskanalit A/D-muunduriga, 5 digitaalset väljundit, millest kaks on pooljuhtreleed ning 5 digitaalset sisendit. Samuti peab seadmel olema üks lisa A/D-muunduriga kanal, mis tegeleb aku pinge mõõtmisega.
- Seadmel peab olema võimalus ühenduda interneti, eelistatavalt läbi Ethernet pesa.
- Andmehõivemoodul peab olema võimeline ennast vajadusel üle programmeerima.
- Andmehõivemooduli taktsagedus peab olema piisav mõõtmiste teostamise jaoks.
- Andmehõivemoodulil peab olema piisavalt seadmemälu, et ühenduse puudumisel sinna salvestada mõõtmistulemused.

4.2 Funktsionaalsus

- Süsteem täidab oma põhifunktsioone autonoomselt, ilma inimese sekkumiseta.
- Süsteem on võimeline kontrollima serverist, kas tarkvarale on uuendusi, ning uuendused alla laadima.
- Süsteem saab individuaalselt aktiivsete sisendite/väljundite info, ning samuti muud tööks vajalikud individuaalsed väärtused (nt. andmete uuendamise sagedus) serverist.
- Süsteem on võimeline reageerima mõõtmise tulemustele ettemääratud viisidel.

- Serveriühenduse puudumise korral süsteem salvestab andmed sisemällu, ning ühenduse taastumisel sünkroniseerib andmed serveriga.

4.3 Ülevaade erinevatest akutüüpidest

Elektrienergiat saab salvestada füüsikaliste ning keemiliste vahendite abil. Füüsikalised salvestid on kondensaatorid, kus kahele paralleelsele plaadile, mis on teineteisest dielektrikuga eraldatud, kogunevad laengud. Keemilistes akumulaatorites muudetakse laadimisel elektrienergia keemiliseks energiaks ning tühjenemisel vastupidi. [5]

Akupatarei põhilisteks näitajateks on selle nimipinge ja mahutavus. Aku pinget kirjeldatakse voltides ja mahutavust C_{aku} ampertundides (A·h). Vastavalt sellele saab defineerida suhtelise aku laadimis- ja tühjenemisvoolu $C_{aku}/100$, mis tähendab 100A·h aku juures laadimis- või tühjenemisvoolu 1A.

Erinevaid akumulaatoreid saab võrrelda järgmiste parameetrite kaudu:

- spetsiifiline energia – näitab akusse salvestatud energia ja massi suhet (W·h/kg);
- rakupinge – näitab akupatarei ühe elemendi nominaalpinget voltides;
- elutsüklid – näitab aku laadimiste ja tühjenemiste arvu, mil aku kogumahtuvus on langenud 80% peale;
- enesetühjenemine – näitab, mitu % laetud aku kuus ilma koormiseta ise tühjeneb;
- töötemperatuur – näitab akule sobivat keskkonna-temperatuuri;
- kiire laadimine – näitab aega, mis kulub aku võimalikult kiireks laadimiseks;
- ülelaadimise taluvus – näitab, kui tundlik on aku ülelaadimisele.

Tabel 4.1 Erinevat tüüpi akude võrdlus.

Parameetrid	NiMH	Pliiaku	Li-ion	Li-ion polümeer	Super-kondensaator
spetsiifiline energia (W·h/kg)	45-80	30-50	110-160	100-130	5
Elutsükkel	300-500	200-300	500-1000	300-500	1000000
Kiire laadimine	2-4h	8-16h	2-4h	2-4h	10s
Ülelaadimise taluvus	Madal	Kõrge	Väga madal	Madal	Pole võimalik üle laadida
Rakupinge	1,25V	2V	3,6V	3,6V	2,30-2,75V
Enese-tühjenemine kuus	30%	5%	10%	10%	50%
Töötemperatuur	-20-60°C	-20-60°C	-20-60°C	0-60°C	-40-65°C

4.4 NiMH aku laadumine ja tühjenemine

NiMH akud on akudest ühed raskemini laetavad. Võrreldes nende laadimist pliiakudega, ei ole võimalik NiMH akude laengut lihtsalt pinge ülempiiri järgi hinnata. Olukorda muudab keerulisemaks see, et NiMH akud on üsna tundlikud ülelaadimisele, mis vähendab nende eluiga. Laadimiseks kasutatav pinge ei ole kindlalt paigas, vaid varieerub – oluline on akusid läbiv vool. Akude erinevuse tõttu muutub nende paralleelne laadimine raskeks, sest osad akud võivad suure osa voolust enesele võtta. [5]

On kaks kindlat viisi, kuidas on võimalik laadimise ajal aru saada, et aku on täis. dT/dt – selle meetodiga jälgitakse aku temperatuuri, mis hakkab järsult tõusma, kui aku on täis. Järskus temperatuuri tõusuks loetakse $1-2^{\circ}\text{C}/\text{min}$. dV/dt – selle meetodiga jälgitakse aku pinget, mis teeb aku laadimise lõpus järsu, kuid väikese languse. Järsuks pinge languseks loetakse ühe raku pinge langust ekstreemumist 5-10mV. Antud väärtused kehtivad kiire laadimise puhul, mil laadimisvool jääb vahemikku $0,5-1C_{aku}$. Aku laadimine lõpetatakse

koheselt või laadimist muudetakse olukorras, kus ühe raku pinge ületab 1,8V või aku temperatuur tõuseb üle 50-60°C (vastavalt akule).

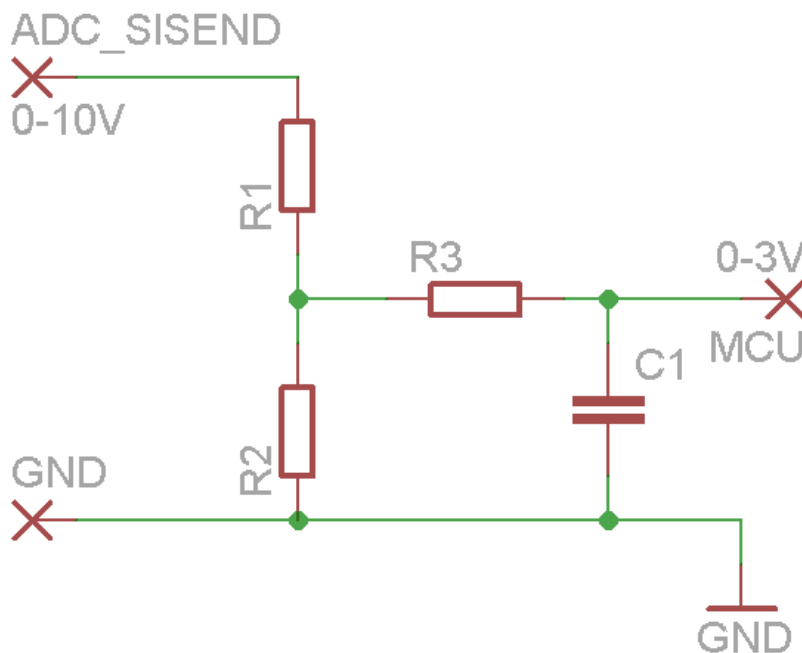
Kui puuduvad võimalused piisavalt täpseks pinge ja temperatuuri mõõtmiseks, saab kasutada aeglast laadimist. On teada, et NiMH akut ei ole võimalik ajalise piiranguga üle laadida, kui laadimiseks kasutatav vool on $C_{aku}/10$. Akut võib laadida sellise vooluga kuni 20 tundi. Aku laengu säilitamiseks pika aja vältel võib kasutada nirstavat laadimist (ingl. *trickle charging*), kus kasutatakse voole $0,03C_{aku}$. [5]

Antud töös kasutatakse NiMH akut, kuna nendega on võimalik kasutada nirstavat laadimist. Nirstav laadimine sobib andehõivemoodulile, kuna aku on kasutusel vaid hädajuhtudel ning nirstavat laadimist on lihtne realiseerida. Kuigi antud töös saaks kasutada ka Li-ion või Li-ion polümeer akut või koguni super-kondensaatorit, siis kõige odavam ja lihtsam on kasutada NiMH akut, kuna spetsiifilise energia suurendamine erilist efekti ei anna – võib alati kasutada rohkem NiMH akusi, rangeid massi ega suuruse piiranguid loodavale süsteemile ei ole. Kokku kasutatakse kuute rakku kogupingega: $1,2V \times 6 = 7,2 V$.

5. Juhtmooduli sidestamine

5.1 Analoo-digitaalmuundurite (A/D-muundurite) sidestamine

A/D-muunduri õige sidestamine on oluline, kui soovime, et A/D-muunduri poolt mõõdetud lugemid kirjeldaksid võimalikult täpselt ka mõõdetava signaali tegelikke väärtusi. Kuigi kasutatava A/D-muunduri lahutus on suhteliselt suur, 12 bitti, siis reaalselt sellise lahutuse saamine nõuab head sidestamist. Kuna käesolevas töös uuritava kontrolleri A/D muundurite võrdluspinge on maksimaalselt 3,3V, siis ei saa ka sisendpinge seda väärtust ületada.



Joonis 5.1 A/D-muundurite sidestamine mõõdetava signaaliga.

Andmehõivemooduli A/D-muunduri poolt mõõdetavaks pingetevahemikuks sai valitud 0-10V. Põhjuseks on tööstuses kasutatavate andurite väljundpinge vahemik, mis on sageli vahemikus 0-10V. Teiseks palju kasutatust leidnud standardiks on variant, kus anduri signaal on lineaarses seoses anduri väljundvooluga, mis on vahemikus 0-20mA [6].

Pingevahemiku 0-10V muundamise pingevahemikule 0-3,3V saab läbi viia tavalise pingejaguriga. Siinkohal on oluline valida õige pingejaguri kogutakistus. Kui pingejaguri kogutakistus on väga suur, siis on ka väga vähe energiat omav signaal võimeline sellel mõõdetava pingete tekitama. See pole soovitatav kuna selliseks signaaliks on just kõiksugu mürad. Kui pingejaguri takistid valida väga väikesed, siis on vaja jälle suurt sisendsignaali võimsust, et A/D-muunduril mõõdetav signaal tekitada. Joonisel 5.1 toodud jaguri puhul on sisendpinge U_s ja väljundpinge U_v seotud järgnevalt:

$$\frac{U_v}{U_s} = \frac{R_2}{R_1 + R_2} \quad (5.1)$$

Valime jagatiseks $3V / 10V = 0,33$. Valides sellise suhte tagame, et kui sisendpinge ületab 10V ei lähe A/D-muundur kohe küllastusse. Lahutus selles palju ei halvene, kuna A/D-muunduri lahutus on suur. Näeme, et pingete vähendamise 0,3 korda saab läbi viia lõpmata arvu erinevate takistite kombinatsiooniga:

$$\frac{R_2}{R_1 + R_2} = 0,3 \quad (5.2)$$

Selleks, et takistid R_1 ja R_2 täpselt määrata, tuleb meil paika panna jaguri kogutakistus. Erinevate kogutakistustega jagurid tarbivad sisendist erinevat võimsust, kuigi pingete vähendavad samamoodi. Võiks arvata, et kõige kasulik on tarbida sisendist võimalikult vähe võimsust, et mõõteahela koormus mõõtesensorile oleks võimalikult väike. Selline lahendus pole aga A/D-muundur sidestamisel kõige otstarbekam. Põhjuseks on asjaolu, et siis on vaja sisendisse väga vähe energiat, et väljundisse mõõdetav pingete tekitada. Sellise pingete võib aga ka tekitada mõõteanduri ühendusjuhtme kaudu süsteemi tulev müra. See tähendab, et suurendades pingejaguri kogutakistust suureneb ka mõõteahela müra-tundlikkus. Kui aga kogutakistus valida liiga väike, siis ei pruugi mõõtesensori väljundvõimsusest piisata, et vajalikku väljundpingete säilitada. Mõistlik oleks valida pingejaguri kogutakistus selline, et mõõtesensori väljundvõimsus oleks suurem, kui pingejaguri poolt tarbitav võimsus ning et pingejaguri kogutakistus oleks palju väiksem

A/D-muunduri sisendtakistusest ($\sim 1\text{M}\Omega$). Kuna pea igasugune mõõtesensor on võimeline välja andma vähemalt 10mA [6], siis saame mõõtesensori väljundvõimsuseks $10\text{mA} \cdot 10\text{V} = 100\text{mW}$. Pingejaguri kogutakistuseks saame Ohmi seadusest:

$$R_1 + R_2 = \frac{10\text{V}}{10\text{mA}} \quad (5.3)$$

Seega on mõistlik valida pingejaguri kogutakistuseks $1\text{k}\Omega$. Takistite R_1 ja R_2 täpsed väärtused saame valemist 5.2. Lisades valemisse kogutakistuse, saame $R_2 / 1000 = 0,3$ kust $R_2 = 300\Omega$, ning $R_1 = 1000\Omega - 300\Omega = 700\Omega$.

Peale pingejagurit on RC-ahel, mis eemaldab signaalist müra. Müra spekter on pidev (nn valge müra), aga tegelikul signaalil saab olla ainult lõplik muutumise kiirus, millest madalamaid sagedusi tuleks läbi lasta. Antud olukorras valime muutumise kiiruseks ehk ahela ajakonstandiks $\tau = 0,01\text{s}$. See kiirus sobib antud süsteemi, kuna sellest kiiremini mõõtmisi kindlasti ei teostata ning samuti pole tihti analoogandurid võimelised nii lühikese ajaga muutust tuvastama:

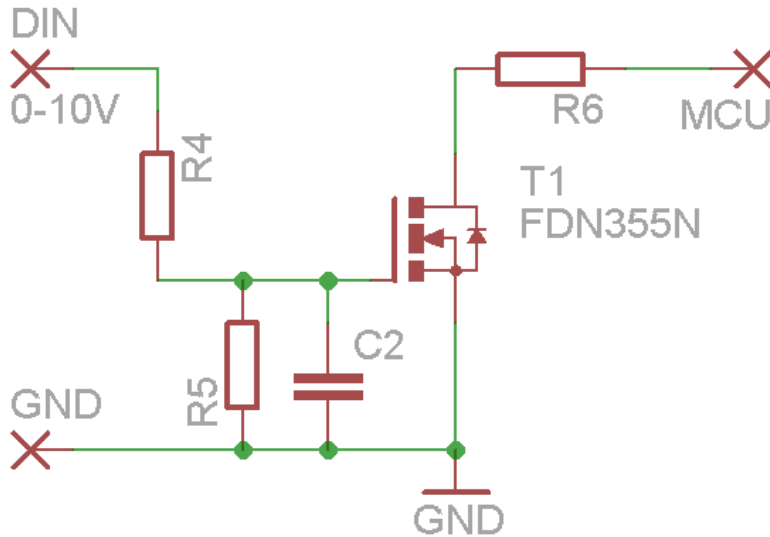
$$\tau = R_3 \cdot C_1 \quad (5.4)$$

Lõikesagedus on võrdeline ajakonstandi pöörväärtusega:

$$f_c \approx \frac{1}{R_3 \cdot C_1} \quad (5.5)$$

Takisti R_3 peab olema palju väiksem, kui sisendi sisetakistus ($R_3 \ll R_{\text{sis}}$). Selle kriteeriumi täitmiseks valime R_3 100 korda väiksema sisetakistusest ($\sim 1\text{M}\Omega$), seega $R_3 = 10\text{k}\Omega$. Asendades takistuse valemis 5.4, saame kondensaatori C_1 väärtuseks $1\mu\text{F}$. Saadud ahela lõikesageduse f_c leiame valemist 5.5 ning selleks on $1 / 0,01\text{s} = 100\text{Hz}$. Kasutatavate komponentide loetelu ning nominaalväärtused on toodud ka tabelis 5.1.

5.2 Digitaalsete sisendite sidestamine



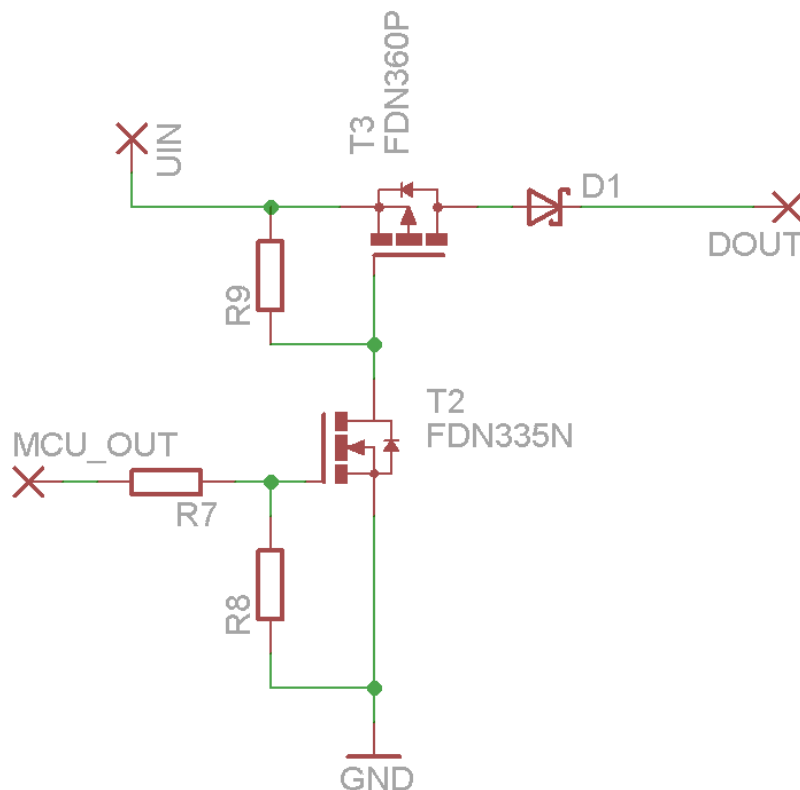
Joonis 5.2 Digitaalse sisendi sidestamine juhtmooduliga.

Digitaalse sisendi sidestamisel on oluline see, et sisend ei oleks otse ühendatud juhtmooduliga, kuna tööstuses kasutatavad pinged vahemikus 10-24V on sisendiks liiga kõrged. Lisaks aitab otseühenduse puudumine ära hoida võimaluse, et liiga kõrge pinge korral kogu juhtmoodul rikki läheb. Hea lahendus selleks on pingejaguriga väljatransistori juhtimine. Transistor on maa ja juhtmooduli sisendi vahel, mille olek on seatud kõrgeks.

Kui sisendisse tuleb signaal, siis transistor avaneb ning seab mooduli sisendi madalasse olekusse, mille omakorda tuvastab tarkvara. Liiga suure pinge või pingepulsi korral läheb suure tõenäosusega läbi transistor, mitte juhtmoodul, mis on oluliselt odavam komponent. See on põhiliseks põhjuseks, miks kasutada lisaks pingejagurile sisendis veel väljatransistorit. Juhtmooduli jala ja maanduse vahel peab olema ka takisti, et moodul liiga palju voolu ei kulutaks, kui transistor avatud on. Selle takisti väärtuse leidmiseks peame paika panema, palju voolu tarbitakse mooduli jalast. Soovime, et tarvitatakse 1mA voolu, siis sobib takisti R_6 takistuseks $3,3V / 1mA = 3,3k\Omega$.

Müra vältimiseks sisendis peab pingejagur sisendi ees tarbima 10mA voolu. Seega sobivad pingejagurisse sama väärtusega takistid nagu A/D-muunduri ees, ehk $R_4 = 700\Omega$ ja $R_5 = 300\Omega$. Takistiga R_5 asub paralleelselt kondensaator C_2 , mis takistab väga lühikesel impulsil katkestuse tekitamist. Kuna transistori enda mahutavus on väga väike, siis peab see kondensaator olema suurusjärgus 100nF.

5.3 Digitaalsete väljundite sidestamine

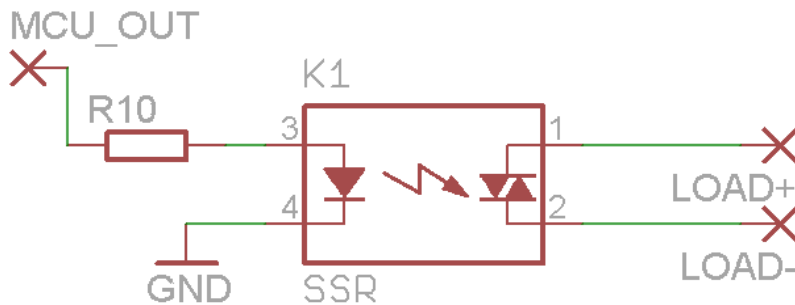


Joonis 5.3 Digitaalse väljundi sidestamine juhtmooduliga.

Digitaalse väljundi sidestamise põhimure on see, et juhtmooduli väljund ise ei suuda juhtida vajalikke seadmeid, kuna tihti vajavad tööstuslikud seadmed samuti sisendpinget 10-24V ja voolu, mis on suurem kui mooduli välja antav 40mA. Lahendus sellele on väljundist n-tüüpi transistori juhtimine, mis avanemisel maandab ära p-tüüpi transistori. P-tüüpi transistor laseb läbi jalast U_{in} tuleva seadme sisendpinge (10-24V) ning voolu, mis suudab juhtida vajalikke seadmeid.

Juhtmooduli väljundi (*MCU_OUT*) järel tulev pingejagur maandab transistori T_2 ära madala oleku korral, et ei tekiks ujuvat olekut. Takistite R_7 ja R_8 väärtusteks sobivad $1\text{M}\Omega$. Takisti R_9 hoiab ära transistori T_3 ujuva oleku ning R_9 väärtuseks sobib $1\text{k}\Omega$. Transistorite valimise kriteeriumid on madal lahtioleku takistus ($0,1\Omega$) ja transistor T_3 peab suutma läbi lasta väljundis tarbitava voolu (valitud transistor laseb läbi kuni 2A). Diodi D_1 valimise kriteerium on see, et diod kannataks sama voolu, mida transistor T_3 .

5.4 Digitaalse releväljundi sidestamine



Joonis 5.4 Digitaalse releväljundi sidestamine.

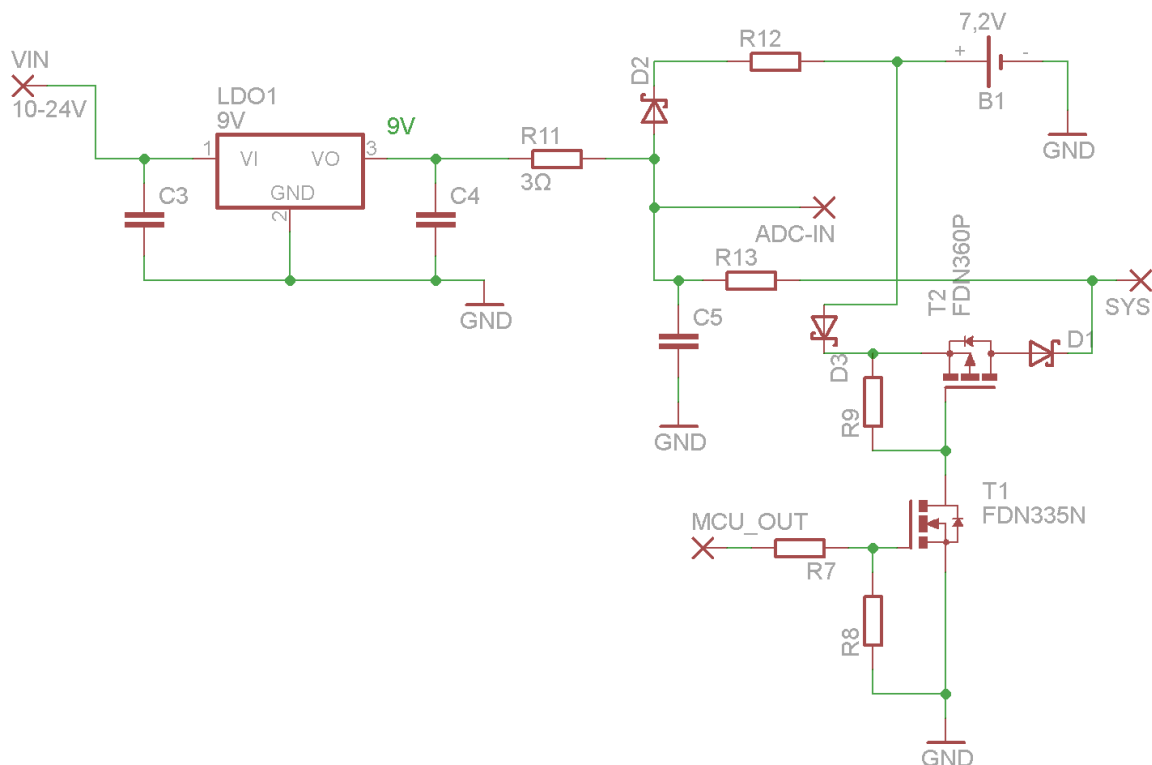
Pooled digitaalsetest väljunditest on ühendatud releega. Tänu sellele on nad võimelised lülitama ka vahelduvvooluga töötavaid seadmeid. Takisti R_{10} piirab voolu, mida tarbitakse relee lülitamiseks. Kuna tavaliselt pooljuhtreleed avanevad juba $5\text{-}10\text{mA}$ vooluga, aga juhtmoodul annab välja kuni 40mA , siis on mõttekas seda voolu piirata, et mitte pooljuhtreleest rohkem voolu läbi ajada, kui tegelikult vaja on. Voolu piiramiseks 10mA peale peab takisti R_{10} väärtus olema $3,3\text{V} / 0,01\text{A} = 330\Omega$, arvestamata pooljuhtrele enda sisendtakistust, st tegelik takistus on sellest väiksem olenevalt pooljuhtrele enda takistusest pingel $3,3\text{V}$.

Tabel 5.1 Komponentide nominaalväärtused.

Komponent	Nominaalväärtus	Komponent	Nominaalväärtus
R_1	700 Ω	C_1	1 μ F
R_2	300 Ω	C_2	100nF
R_3	10k Ω	C_3	10 μ F
R_4	700 Ω	C_4	10 μ F
R_5	300 Ω	C_5	1 μ F
R_6	3,3k Ω	D_1	$I_{\max}=2A$
R_7	1M Ω	D_2	$I_{\max}=2A$
R_8	1M Ω	D_3	$I_{\max}=2A$
R_9	1k Ω	B_1	NiMH 7,2V, 600mAh
R_{10}	330 Ω	T_1	$R_{\text{avatud}}=0.079\Omega$, $I_{\max}=1,7A$
R_{11}	1 Ω	T_2	$R_{\text{avatud}}=0.079\Omega$, $I_{\max}=1,7A$
R_{12}	500 Ω	T_3	$R_{\text{avatud}}=0.1\Omega$, $I_{\max}=2A$
R_{13}	1 Ω	K_1	$I_{\text{lülitus}}=5-20mA$

6. Süsteemi toide

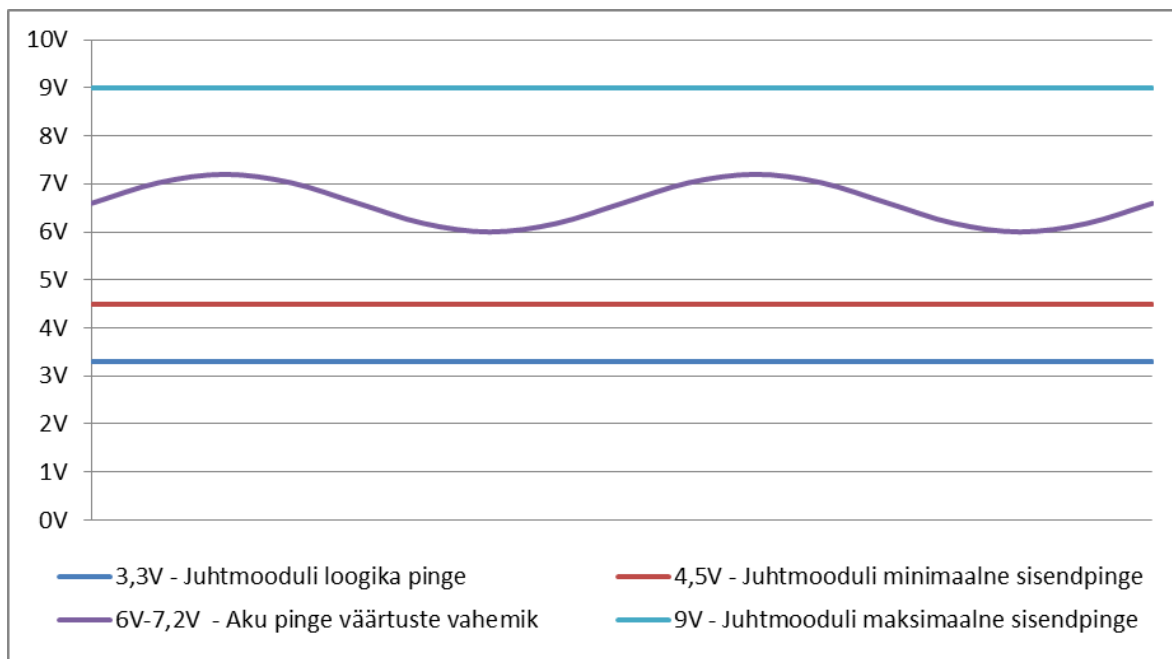
6.1 Andmehõivemooduli toide vahelduvvooluvõrgust



Joonis 6.1 Juhtmooduli toiteahel

Andmehõivemooduli toiteks sai valitud 10-24V, aga juhtmoodul vajab tööks 4,5-9V. Kuna aga süsteemil on ka varutoide, milleks on 7,2V aku, siis on optimaalne valida toitepingeks 9V, sest selle pealt saab laadida akut ja samuti töötab juhtmoodul selle pealt. 9V saamiseks kasutame madala voolutarbega pingeregulaatorit. Pingeregulaatori järel on 1Ω takisti (R_{11}), mis stabiliseerib A/D-muunduri näitu ning juhtmooduli ees on 1Ω takisti (R_{13}) ja 1μF kondensaator (C_5), mis hoiavad ära võnkumise pingeregulaatori ja juhtmooduli enda pingeregulaatori vahel. Saadud pinge läheb veel ka aku laadimiseks läbi schottky diodi ning takisti akuni. Schottky diod on kasutusel tavalise diodi asemel tema madalama pingelangu pärast. Takisti R_{12} väärtus sõltub aku näitajatest, kuna aku laadimisvool peaks

olema $0,03C_{aku}$, mis tagab aku pikema eluea. $600\text{mA}\cdot\text{h}$ aku puhul on see 18mA või vähem, ning $R_{I2} = 9\text{V} / 18\text{mA} = 500\Omega$.



Joonis 6.2 Andmehõivemoodulis kasutatavad pinged.

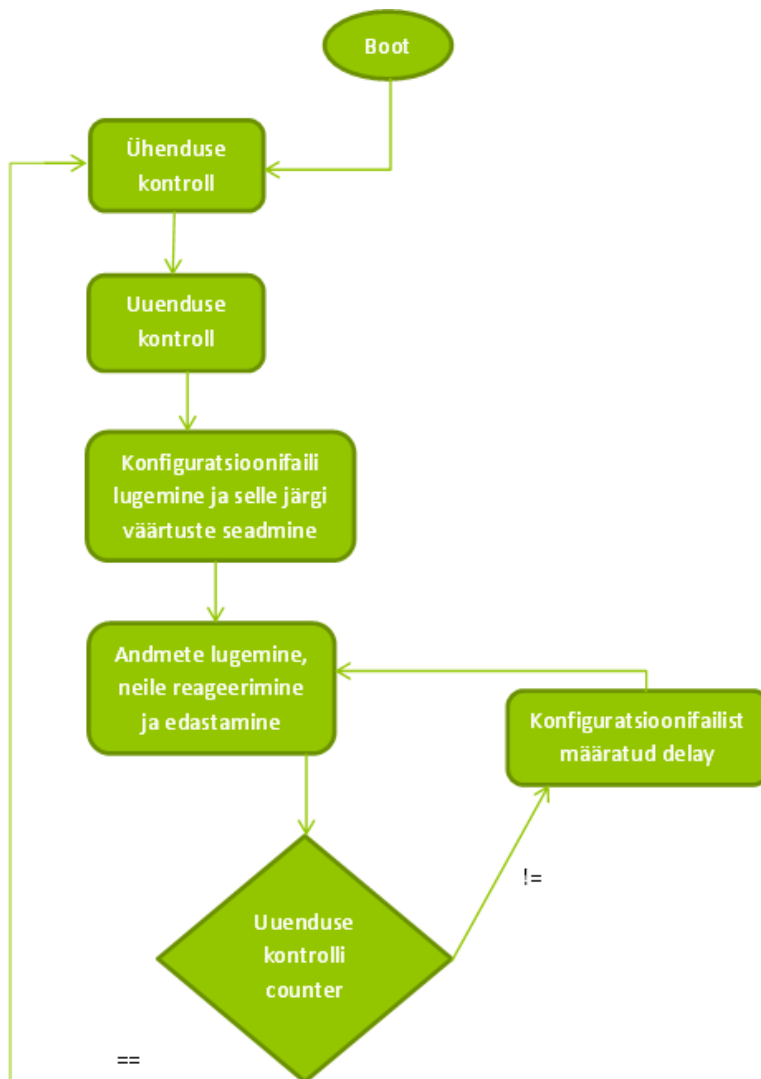
6.2 Andmehõivemooduli toide tagavaraakult

Sisendpinge mõõtmiseks on kasutusel ka üks A/D-muunduri kanal, mis tuvastab selle, kui väline toide on kadunud ning seade töötab aku pealt. A/D-muunduri kaudu on vaja aku pinget mõõta selleks, et vältida kahju aku elementidele. $7,2\text{V}$ aku koosneb kuuest $1,2\text{V}$ elemendist ning kahju vältimiseks ei tohiks ühegi elemendi pinge kukkuda alla 1V . Seega protsessor mõõdab pinget, 6V juures lülitab aku ahelast välja ning sellega ka endal toite välja. Vahelduvooluvõrgu toite ühendamisel saab protsessor sisendisse jälle 9V , mille pealt hakatakse akut uuesti laadima.

Joonisel 6.2 on näidatud ka aku pinge vahemikus $6,0\text{V}$ - $7,2\text{V}$. $6,0\text{V}$ vastab tühjale akule ja $7,2\text{V}$ vastab laetud akule. See pingevahemik on täiesti piisav protsessori töötamiseks, kuna protsessor vajab sisendis minimaalselt $4,5\text{V}$. Selle voolu pealt alandatakse pinge $3,3\text{V}$ peale, mis on protsessori loogika tööpinge.

7. Andmehõivemooduli tarkvara

Protsessor NXP LPC1768 programmeerimine toimus mbed arendusplatvormil [7]. See sisaldab endas nii sisseehitatud teeke, kui ka kasutajate poolt pidevalt lisatavaid ja täiustatavaid teeke. Programmeerimine toimus C++ keeles ja võrgus kasutatavas programmeerimiskeskkonnas.



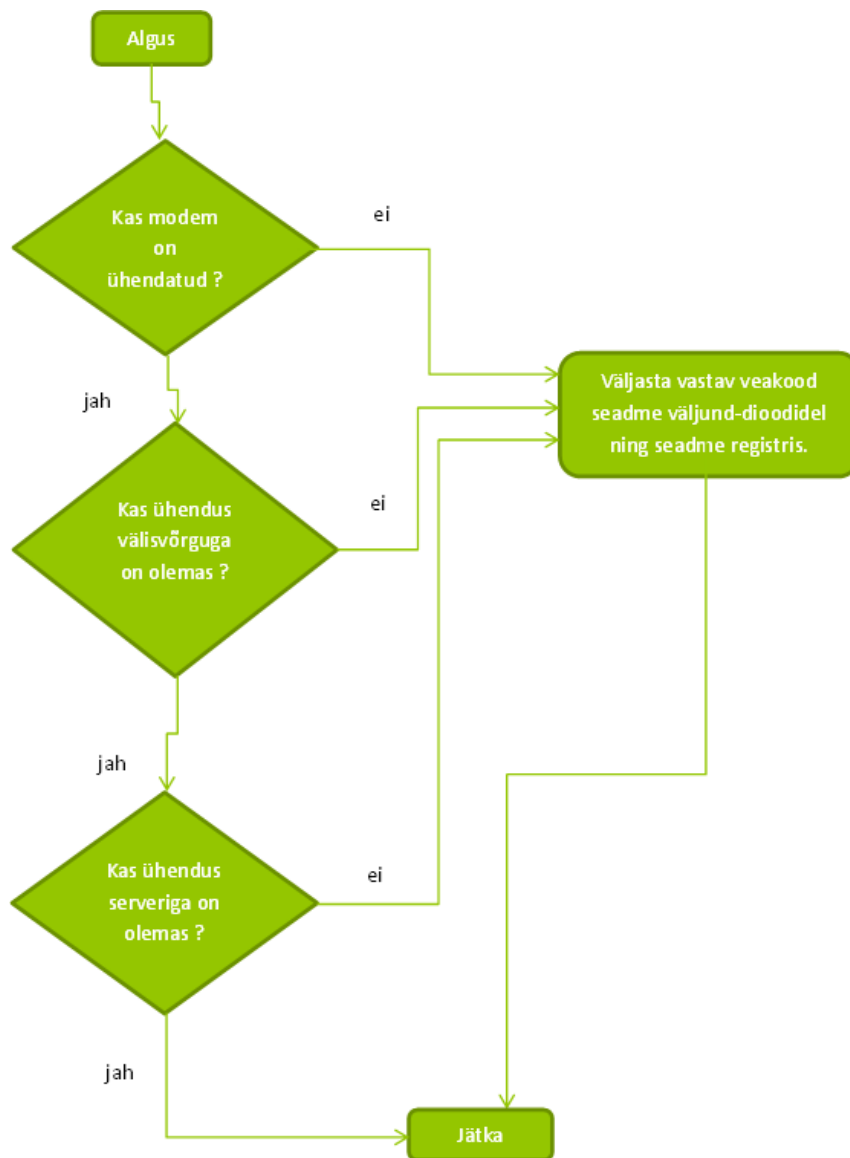
Joonis 7.1 Süsteemi töö blokkdiagramm.

Süsteemi põhifunktsioonide ülevaade on toodud joonisel 7.1. Töö saab alguse internetiühenduse kontrollist, et saaks teada, kas on võimalik luua ühendust serveriga. Kui internetiühendus on olemas, siis uuritakse, kas serveris on uuem konfiguratsioonifail või tarkvara kui hetkel süsteemis kasutusel. Kui leitakse uuem, siis see laetakse alla ja võetakse kasutusele. Edasi hakkab süsteem andmeid lugema ja neid edastama ning vajadusel ka andmetele reageerima.

Tarkvara loomise esimene etapp oli luua teek, mis luges seadme sisemälust konfigureerimiskäskude faili ning võtaks saadud väärtused kasutusele peasisüsteemis. Selliselt failist loetavateks väärtusteks on näiteks aktiveeritud sisend/väljundpordid või andmete uuendamise kiirus. Failist andmete lugemist ja kasutusele võtmist teostava koodi põhiidee on välja toodud lisa nr 3.

Peale aktiivsete sisendite lugemist tuli luua teek, mis loeks sisendinfot. Esimeseks teegiks oli digitaalsisendi info lugemine ning salvestamine, kuna seda sai kõige lihtsamalt testida. Esialgses tarkvaras töötas loogika 3,3V pealt ehk ilma eelnevalt kirjeldatud sidemooduliteta.

Võrgu ühenduvuse kontrolliks tuli luua teek, mis testis interneti ühenduvust ning suutis väljastada kolm erinevat veateadet, kas ühendus modemi on olemas, kas ühendus internetti on olemas ja kas ühendus serveriga on olemas. Selle kontrolliga on võimalik tuvastada, millises ahelas on viga, kui ühendust ei saada. Interneti ühenduse testimiseks üritatakse luua ühendust ette antud serveritega (nt. google.com). Veateade väljastatakse seadme peal olevate valgusdiodidega. Ühenduvuse kontrolli ja vigade väljastamist illustreerib joonis 7.2.

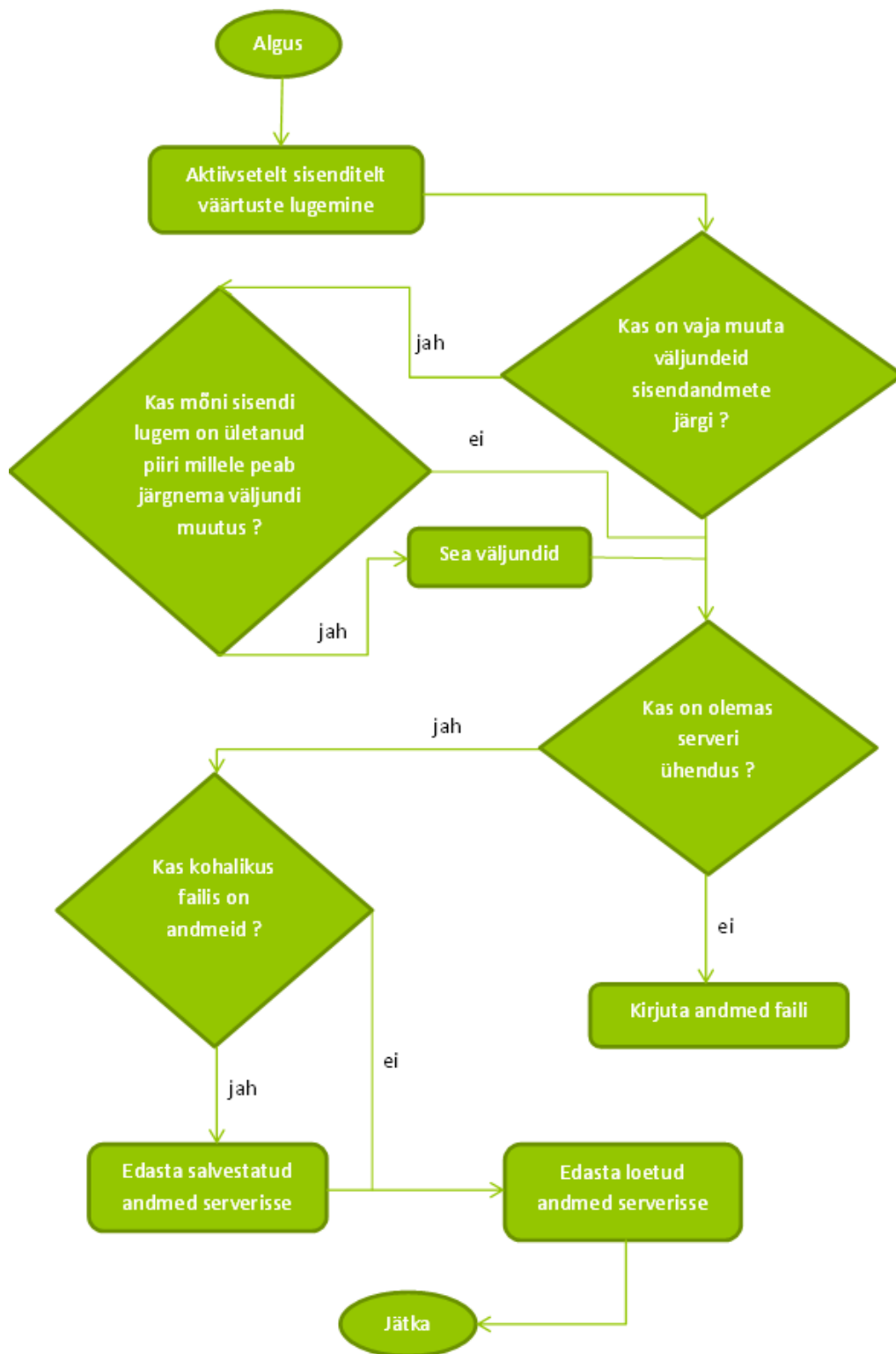


Joonis 7.2 Ühenduse kontrolli blokkdiagramm.

Järgmiseks tuli luua teek, mis koostaks loetud andmetest andmeprotokolli. Andmeprotokolli esmase versiooni ülesehitus algab seadme unikaalse identifitseerimise väljaga (nt. number) ning peale seda tuleb iga aktiivse sisendi lugem, ning viimased kaks baiti on andmepaketi kontrollsumma. Edasiarendusplaan on luua SQL ühendus serveriga, mis eristab seadmeid ning seadmete üheaegset andmeteedastust.

Realiseerida tuli ka algeline veaparandus, mis kontrollis, et A/D-muunduri andmed ei muutuks üle 50% 0,01s jooksul. Kui selline muutus toimus, siis võis arvestada sellega, et muutuse põhjustas müra ning seda lugemit ei edastata.

Andmehõivemooduli üks funktsioon on andmetele reageerimine digitaalväljundeid kasutades. Selle toimimiseks on vaja määrata sisendite piirid ja nende ületamisega kaasnevad tegevused. On defineeritud ka üks lipp, mis määrab selle kas andmetele peab reageerima ehk kas on vaja sisendandmeid üldse kontrollida. Teine oluline funktsioon on andmete salvestamine kohalikku faili, kui serveriühendus puudub ning ühenduse taastumisel faili salvestatud andmed serverisse edastada. Nende funktsioonide käitumist tutvustab joonis 7.3. Selline andmetele reageerimise omadus on väga tähtis paljudes kriitilistes süsteemides ja selle tõttu peavad olema seadmes alati olemas ka väljundite turvalised väärtused, kui serveriga ühendus peaks puuduma.



Joonis 7.3 Andmetele reageerimine ja edastamine.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli disainida andmehõivemooduli riistvara ehk toitemoodul ja sisendite/väljundite sidestus. Esmalt tuli uurida erinevaid akutüüpe ning valida seadmele sobivaim. Siis tuli koostada toiteskeem, mis sisaldaks nii vahelduvvoolu toidet, kui ka tagavara akutoidet. Toiteskeemis oli vaja realiseerida aku laadimine ja tühjenemise jälgimine A/D-muunduri abil. Lõpptulemuseks oli toiteskeem, mille sisendpinge on nõutud piirides ning skeem sisaldas endas ka tagavaratoidet aku näol.

Riistvara disaini teine osa käsitles A/D-muunduri sisendi, digitaalse sisendi, digitaalse väljundi ning digitaalse releeväljundi sidestamist. Neist igaühe peamine takistus oli pingete ja voolude vahe sisendis ja väljundis. Väljundid pidid suutma väljastada piisavat pinget ja voolu seadmete juhtimiseks ning sisendid pidid taluma kõrgemat pinget, kui mikrokontrollerile sisendiks ette nähtud. Vastavalt tuli disainida riistvara, mis manipuleeris signaale nii, et neid saaks kasutada ettenähtud funktsioonidega. Tulemuseks on neli erinevat skeemi, millest igaüks võimaldab vastaval skeemil oma nõudmisi täita.

Lisaks tuli disainida andmehõivemoodulile tarkvara. Sissejuhatuses kirjeldatud tarkvara funktsionaalsuseni päris jõuda ei õnnestunud, kuna töö osutus mahukamaks, kui algselt tundus. Sellele vaatamata võib väita, et valitud juhtmoodul sobib kirjeldatud andmehõiveseadmele väga hästi, kuna kõik vajalikud funktsioonid ja omadused on olemas.

Andmehõivemooduli edasi arendamiseks tuleb valmis teha serveripool ja projekteeritud riistvara jaoks luua ühtne trükiplaat ning sellele seade kokku panna. Serveri pool on antud moodulil väga suur - ilma selleta puudub väga suur osa mooduli funktsionaalsusest. Peale seda saab reaalselt katsetada kõikide moodulite tööd ning vigade ilmunisel nendega tegeleda.

**Software and hardware for fully autonomous data acquisition module based on
LPC1769 ARM Cortex-M3 processor**

Madis Sepp

Summary

The aim of the present project is to design software and hardware for a practical data acquisition system.

The system should be autonomous, suitable for many different situations and also should have low power consumption. The project starts with theoretical study on embedded systems, ARM architecture and a closer look at the Cortex-M3 microprocessor. Also we review different types of rechargeable batteries and choose the most suited for this project.

NXP LPC1768 microprocessor is chosen for this project and the features of the microprocessor are highlighted. Next system's design elements are discussed and a specific design is put forth. First, technical requirements for the intended objectives, including the software and hardware, are defined. Thereafter coupling of the digital inputs and outputs is designed. Also a versatile power scheme is created for use in different industrial realms and environmental conditions.

Next, the software for the project is designed and implemented. Further some are described steps in detail and also illustrated with diagrams. All the key functions of the data acquisition module are explained.

Finally, after thorough study the research and design of the circuit is complete and further development and testing plans are formed.

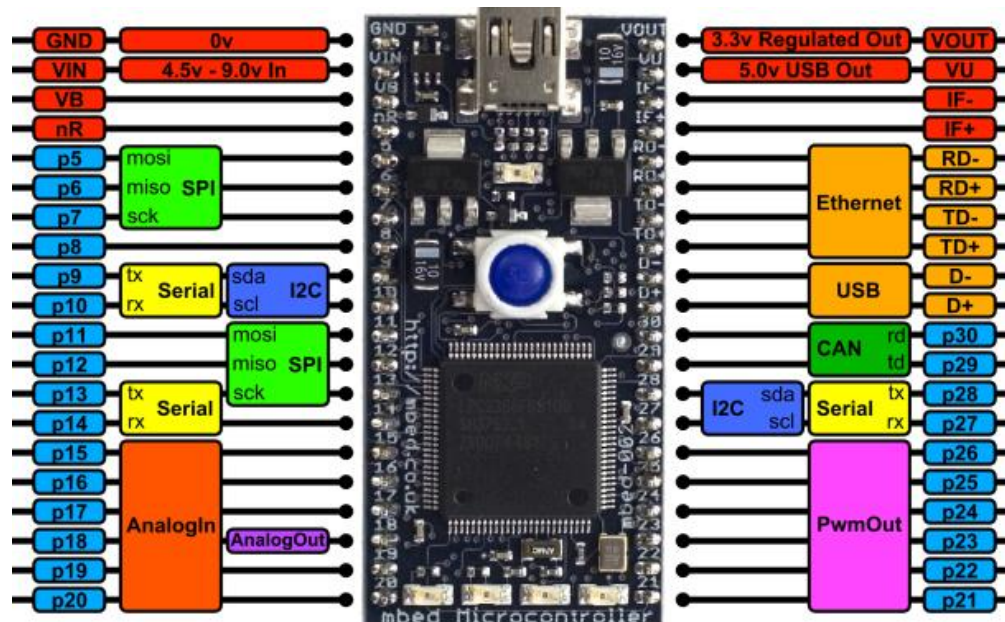
Viidatud allikad

1. James K. Peckol, Embedded Systems: A Contemporary Design Tool (Wiley, Hoboken (N.J.), 2008)
2. Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, <http://LeeSeshia.org>, ISBN 978-0-557-70857-4, 2011.
3. Joseph Yiu, The Definitive Guide to the ARM Cortex-M3 (Elsevier, Oxford, 2010)
4. UM10360 LPC17xx User manual Rev. 2 — 19 August 2010
(http://www.nxp.com/documents/user_manual/UM10360.pdf)
5. Thomas Roy Crompton, Battery Reference Book Third edition (Newnes, Great Britain, 2000)
6. Terry L. M. Bartelt, Industrial Control Electronics: Devices, Systems, and Applications 3-rd Edition (Delmar, New York, 2006)
7. www.mbed.org

Lisad



Lisa 1. Juhtmoodul koos arendusmooduliga.



Lisa 2. Juhtmooduli viikude skeem.

Lisa 3. Muutujate algväärtustamine failist.

```
#include "mbed.h"
#include <string>

using namespace std;

LocalFileSystem local("local");
char buffer[25];
string nimi;
int v22rtus;

void funktsioon(string muutuja_nimi, int v22rtus) {
    if (muutuja_nimi=="muutuja1")
        muutuja1=v22rtus;
    if (muutuja_nimi=="muutuja2")
        muutuja2=v22rtus;
    ...
}

int main() {
    FILE *fp = fopen("/local/config.txt", "r");
    while(fscanf(fp,"%s %d",buffer,&v22rtus)==2)
    {
        nimi=buffer;
        funktsioon(nimi,v22rtus);
    }
    fclose(fp);
}
```