

TALLINNA TEHNIKAÜLIKOOL
Elektrotehnika instituut

ATR70LT

Vladimir Mišenkin

AUTOPARGI AKUMULAATORITE MONITOOINGUSÜSTEEM

Magistritöö

Instituudi direktor	Tõnu Lehtla
Juhendaja	Taavi Möller
Lõpetaja	Vladimir Mišenkin

Tallinn 2016

AUTORIDEKLARATSIOON

Kinnitan, et käesolev lõputöö on minu iseseisva töö tulemus. Kõik selle koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandus allikatest ja mujalt pärinevad andmed on viidatud. Varem ei ole selle alusel kutse- ega teaduskraadi ega inseneridiplomit taotletud. Töö on koostatud litsentseeritud tarkvara abil.

Tallinn, 18.01.2016

..... Vladimir Mišenkin

ATR70LT

Autopargi akumulaatorite monitooringusüsteem

Vladimir Mišenkin, üliõpilaskood 132436AAAM, Jaanuar 2016. – 69 lk.

TALLINNA TEHNIKAÜLIKOOL

Energeetikateaduskond

Elektrotehnika instituut, Robotitehnika õppetool

Töö juhendaja: Taavi Möller

Töö konsultant:–

Võtmesõnad: Plii–happe akumulaatorid, kontrollid, LAN (*Local Area Network*) võrk, ladustamine, autopark, laadimistase

Referaat:

Lõputöö on 69 lehel, sisaldab 3 tabelit, 59 joonist.

Lõputöö eesmärk on uurida võimalusi autopargis ladustatud sõidukite plii–happe akumulaatorite laadimistaseme pidevaks jälgimiseks mõeldud automatiseeritud süsteemi ehitamiseks.

Antud lõputöö raames uuriti ka müügis olevaid pakutatavaid lahendusi ja mõõtmise teostamise viise lähtudes nende keerulisusest, hinnast, mõõtmise mugavusest ja täpsusest. Autopargi töö spetsiifikast ja akumulaatorite füüsilistest omadusest lähtuvalt valiti süsteemi põhikriteeriumid ja koostati lähteülesanne. Põhjalikumaks uurimiseks ja võrdlemiseks valiti kaks kontrolleri süsteemi – Digital–Measure V10 ja Arduino Leonardo ETH.

Süsteemi ehitamiseks valitud LAN võrgu liidesega kontrolleri Arduino Leonardo ETH jaoks projekteeriti ja katsetati mõõtmiseks vajalik skeem ning kirjutati programmi algkood. Valiti ka skeemi põhikomponendid.

Kokkuvõttes anti lõputöö raames tehtud uurimisele kriitiline hinnang ja pakuti edaspidiste uuringute ja katsetamise suunad.

ATR70LT

Battery monitoring system for vehicle storage facility

Vladimir Michenkin, student code 132436AAAM, January 2016. – 69 pages

TALLINN UNIVERSITY OF TECHNOLOGY *Faculty of Power Engineering

Department of Electrical Engineering

Chair of robotics

Tutor of the work: Taavi Möller

Consultant:–

Keywords: Lead–acid accumulators, controllers, LAN (*Local Area Network*), storage, vehicle storage, state of charge

Summary:

Work is conducted on 69 pages, includes 3 tables and 59 drawings.

The goal of given work is to study the opportunities of surveying the state of charge in stored vehicles lead–acid batteries.

In this work nowadays offered solutions were studied and measurement ways where overlooked according to their price, complexity, specifics and precision. Required criteria where determined for system work and specific requirements where acquired according to vehicle storage facility work specifics. Two controller models where chosen to fulfill the given task – Digital–Measure V10 and Arduino Leonardo ETH after what was made a final choice.

For system building and testing Arduino Leonardo ETH was finally chosen and, basing on provided specifications accumulator measurement and loading circuits where developed. Base components where chosen and tested.

In given work summary critical analysis of processed work and its results are given. Directions of future development and experiments are proposed.

ATR70LT

Система наблюдения за аккумуляторами автопарка

Владимир Мишенькин, код студента 132426AAAM, Январь 2016. – 69 стр.

ТАЛЛИНСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ *Энергетический факультет

Электротехнический институт

Кафедра робототехники

Руководитель работы: Таави Мёллер

Консультант работы: –

Ключевые слова: Свинцовые аккумуляторы, контроллеры, LAN (*Local Area Network*), хранение, степень заряда

Реферат:

Работа выполнена на 69 страницах и включает 3 таблицы и 59 рисунков.

Цель данной работы – изучить возможности слежения за степенью заряда автомобильного свинцово-кислотного аккумулятора в условиях автопарка.

В работе анализируются имеющиеся на данный момент предложения по их цене, сложности, точности и спецификации. Выдвигаются необходимые критерии по условиям работы автопарка и требованиям к работе системы. Две основные модели автоматических контроллеров выбираются для выполнения задачи – Digital–MeasureV10 и Arduino Leonardo ETH – после чего проводится их детальный анализ и окончательный выбор исполнительного элемента.

С выбранным контроллером (Arduino Leonardo ETH), для проверки системы, проводятся эксперименты, и строится необходимая для измерений электросхема с выбором базовых компонентов.

В выводе описаны и оценены результаты работы, приведены варианты дальнейших исследований.

SISUKORD

AUTOPARGI AKUMULAATORITE MONITOORINGUSÜSTEEM	1
MAGISTRITÖÖ ÜLESANNE	7
1. SISSEJUHATUS	8
2. AKUMULAATORITE MONITOORINGUSÜSTEEMID	10
2.1. Nõuded ehitatavale süsteemile	11
2.2. Pliiakude ehitus	12
2.3. Ülevaade pakutavatest plii–happe akumulaatorite monitooringusüsteemidest	12
2.4. Plii–happe akumulaatorite laadimistaseme määramine	17
3. MONITOORINGUSÜSTEEMI PROJEKTEERIMINE	20
3.1. Süsteemi mõõteahela ülesehitamise printsiibid	20
3.2. Süsteemi andmesidevõrgu ülesehitamise printsiibid	23
3.3. Seadmete valik	28
3.3.1. TCP/IPV10 voltmeeter/termomeeter/relee©	28
3.3.2. Arduino Leonardo ETH©	32
3.4. Mõõtmisahela projekteerimine ja katsetamine	35
3.4.1. Potentsiaali jagamise arvutamine ja katsetamine	35
3.4.2. Madalpääs sisendfiltri arvutamine ja katsetamine	39
3.4.3. Mõõtmisahela kaitsmine ülepinge ja lühise eest	45
3.4.4. Temperatuuri mõõtmine	46
3.4.5 Mõõtmisprotseduuride programm	48
3.5. Koormusahel pindlaengu eemaldamiseks	51
3.5.1. Koormusahela arvutamine	51
3.5.2. Koormusahela projekteerimine ja seadmete valik	56
3.6. Mõõtmistulemuste võrdlemine	58
3.7. Andmete edastus võrgus	59
4. KOKKUVÕTE	60
KASUTATUD KIRJANDUS	63
L I S A D	64
Lisa 1: Mõõtmisprotseduuri programm	65
Lisa 2: Näidis programm html liidese jaoks	66
Lisa 3: Mõõtmise ja koormamise ahelate skeem	69

TALLINNA TEHNIKAÜLIKOOL
Elektrotehnika instituut

KOOSKÕLASTATUD

Prof. T. Lehtla

.....

MAGISTRITÖÖ ÜLESANNE

Vladimir Mišenkin, üliõpilaskood 132436 AAAM

Magistritöö teema: Autopargi akumulaatorite monitooringusüsteem

Ülesanne: Uurida võimalusi hajutatud autopargi sõidukite akumulaatorite laadimistaseme mõõtmiseks mikrokontrollerite abil.

Lähteandmed:

1. Info autopargis kasutatavate akude kohta.
2. Olemasolevate lahenduste info internetis.
3. Seadmete ja komponentide andmelehed ning juhendid.
4. Teadusartiklid andmebaasides.

Lahendamisele kuuluvate probleemide loetelu:

1. Olemasolevate lahenduste analüüs.
2. Plii–happe akumulaatorite laadimistaseme määramise protsessi kirjeldamine.
3. Lahenduse skeemi väljatöötamine ja katsetamine, kasutatavate seadmete valik.
4. Tarkvaralise testrakenduse loomine ja testimine.

Magistritöö esitada instituuti eesti keeles 2 eksemplaris hiljemalt 18.01.2016

Juhendaja:

Ülesande vastu võtnud:

T. Möller

Üliõpilane V. Mišenkin

1. SISSEJUHATUS

Sõidukite elektriakumulaatori (või plii–hape akumulaatori, edasi – aku) laadimistaseme määramist on aastate jooksul väga põhjalikult uuritud, mille tulemusena on leiutatud mitmesuguseid seadmeid ja protsesse, mis võimaldavad kindlaks teha, kui palju energiat on antud akul alles jäänud. Protsessid võivad varieeruda oma keerukuse poolest: mõned seadmed võivad väga täpselt näidata nii laadimistaset (edaspidi – SOC – Ingl. – *State Of Charge*) kui ka järgi jäänud energiasalvestamise potentsiaali (edaspidi – SOH – Ingl. – *State Of Health*). Ometi on need seadmed sageli niivõrd keerulised ja nõudlikud varustuse suhtes, et neid pole rakenduslikult mõistlik kasutada ei massilise kontrolli käigus ega süsteemides, kus kõrge mõõtmistulemuste täpsus pole nõutud. Mõned mõõtmisviisid vajavad aku võrgust lahti ühendamist või lahti võtmist – sellega piiratakse seadmete kasutusvaldkonda sellega, et mõõtmisi võivad teostada ainult vastavalt koolitatud spetsialistid spetsiaalsete vahenditega.

Selleks, et täpne mõõtmine oleks teostatav tavakasutajate poolt, hakati tootma seadmed, akutestreid, mis on võimelised kontrollida nii SOC–d kui ka SOH–i. Nende kasutamine on lihtne ja kiire, kuid seevastu nõuab see kasutaja vahetut kohalolekut. On ka automaatseid akutestreid – nad jälgivad aku SOC–d ja vajadusel alustavad laadimisprotsessi iseseisvalt või võrguliidesega – need seadmed pole ainult automaatse laadimisevõimekusega, vaid ka annavad teada aku olekust ja SOC–st võrguprotokolli kaudu.

Laadimistaseme määramise viisi valikut mõjutab olulisel määral ka süsteemi paindlikkus. Alalise akude arvuga süsteeme (akupankasid) on lihtsam varustada automaatsete akutestritega kui süsteeme, kus akude arv on muutuv või teadmata (näiteks, autopargid, kus on vaja kontrollida sõidukite akude SOC–sid).

Eelnevalt mainitud autopargid on levinud riigikaitseasutustes, kus osa sõidukitest peavad konserveerimise tõttu pikalt seisma pidevas valmisolekus. Neid sõidukeid (Joonise 1.1 peal), või mõningaid neist, võib igal hetkel kasutusse võtta ning peale kasutamist tagastada paiknemispaika, kus neid jätkuvalt konserveeritakse ja hoitakse. Vajadus uurimiseks laadimistaseme määramise paindliku süsteemi ehituse suunas tekkis riigikaitse asutustes üleliigse teenindus personali koormamise pärast. Antud süsteemi olemasolu puhul on võimalik hallata suuri ja hajutatud autoparke vähendatud inimressursside kasutamisega.

Võimalik, et just sõidukite pikaajalise seismise pärast ei ole seda tüüpi autopargid levinud tavalistes logistilistes või tööstuslikes autoparkides, kus veokid või sõiduaudod peavad olema pidevas liikumises ja rakenduses. Juhul, kui sõidukit pikaajaliselt ei kasutata, arvestatakse seda autopargist maha. Sellega võibki olla seotud asjaolu, et üldiselt ei pakuta valmis lahendusi konserveeritud sõidukite akude jälgimise süsteemiks.



Joonis 1.1. Maastikuveosõiduk

Selles töös uuritakse teoreetilist materjali selleks, et leida piisavalt lihtsat ja täpset SOC mõõtmisviisi suure autopargi jaoks, kus sõidukite arv ja paigaldus on muutuv, kuid eeldatavalt suur (tõenäoliselt 100 või rohkem ühikut), jälgides allpool toodud süsteemi nõudeid. Seejärel tehakse teoreetiline valik seadmetest ja süsteemidest, mis on võimelised mõõtmeid teostada ning osa süsteemist demonstratsiooniks ja katsetamiseks. Pole välistatud, et sellise põhimõttega süsteem on teostatav turul pakutavate vahenditega: suur osa tööst on suunatud just mugava ja kindla süsteemi projekteerimiseks, kus kõik tööd (nii süsteemi kaudu tehtavad tööd spetsialisti kui ka sõidukite kasutamine autojuhtide poolt) on võimalikult lihtsustatud ja veakindlad.

2. AKUMULAATORITE MONITOOINGUSÜSTEEMID

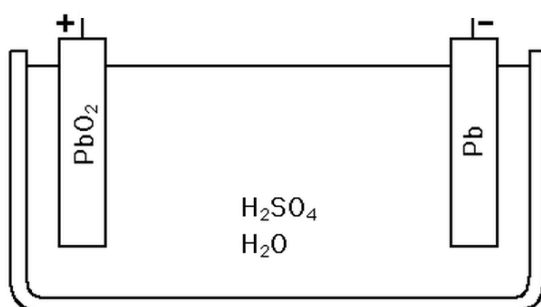
2.1. Nõuded ehitatavale süsteemile

Käesolevas töös projekteeritavale ja testitavale süsteemile kehtivad kindlad nõuded, mis tulenevad riigikaitseasutustes olevate autoparkide ning sõidukite kasutamise ja hoiustamise eripärast. Lisaks tavapärastele baaskriteeriumitele (nt. süsteemi stabiilsus, madal hind või vastavus standartidele) tuleb vajaliku süsteemi projekteerimisel arvestada ka alltoodud tingimustega.

- **Süsteemi ohutus ja lihtsus.** Süsteemis tuleb kasutada ohutut pinget seal, kus kasutaja või autojuht võib kokku puutuda pingestatud osadega. Tuleb vältida ka kasutajate või autojuhtide kokkupuutumist aku jõuahelaga, isegi kui seal on ohutu pinge (enamusele sõidukitest kasutavad 24 V alalist pinget), sest vea puhul võib see kahjustada sõiduki elektrisüsteemi.
- **Süsteemi rakendamine ei tohi muuta sõidukite juhtimist või ladustamisviisi.** Süsteemi kasutajad ja autojuhid ei pruugi olla isikud, kes omavad teadmisi elektrialal. Seetõttu peab mõõtmine olema teostatav võimalikult veakindlalt, et vältida kasutajate või autojuhtide sekkumise vajadust.
- **Süsteemi laiendamisvõimalus.** Lähtudes sellest, et autoparki suurus võib muutuda, tuleb süsteemi ehitamisel kasutada viisi, mis võimaldab vajadusel suurendada, vähendada ja muuta mõõdetavate objektide kogust või asukohta.
- **Mõõdetavate akude laengu säästmine.** On mitmeid viise, kus SOC-d võib väga täpselt määrata aku koormamisega, kuid tuleb pöörata tähelepanu ka asjaolule, et mõõtmine ei tohiks oluliselt SOC-d vähendada.
- **Akumulaatori koormamist tuleb vältida või piirata.** Süsteemi ehitamisel tuleb vältida ka aku laengu kasutamist mõõtmisüsteemi toitmiseks.
- **Süsteemi võrguliidese vajadus.** Kuna autopark võib olla hajutatud, võivad erinevad autoparki sõidukid paikneda erinevates kohtades, mille vahel on suur vahemaa. Seadmed vajavad võrguliidese lahendust, mis võimaldab süsteemi jälgimist kasutajate poolt LAN (*Local Area Network*) või WAN (*Wide Area Network*) võrgu kaudu.

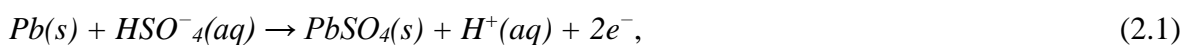
2.2. Pliiakude ehitus

Pliiakku on galvaaniline element, mida kasutatakse elektrienergia säilitamiseks. Pliiakku pingetõstmiseks koosneb aku mitmest eraldi jaotusest – elementidest (Joonis 2.1). Iga pliiaku element sisaldab kahte elektroodi, millest üks on peaaegu puhtast pliiist (Pb), teine elektrod on aga valmistatud pliidioksiidist (PbO₂). Elektrodid asetsevad elektrolüüdis, milleks on väävelhappe (H₂SO₄) 33,6% vesilahus.



Joonis 2.1. Pliiakku element[1]

Negatiivse elektroodi reaktsiooni võib kirjeldada valemiga:



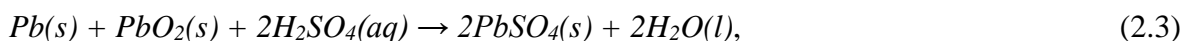
kus Pb – plii,
HSO₄⁻ – bisulfaat,
PbSO₄ – pliisulfaat,
H⁺ – vesinik.

Positiivse elektroodi reaktsiooni valem:



kus PbO₂ – pliidioksiid,
HSO₄⁻ – monosulfaat,
H⁺ – vesinik,
PbSO₄ – pliisulfaat,
H₂O – vesi.

Need reaktsioonid toimuvad välise elektromotoorjõu abil: laadimisel vasakult paremale ja tühjenemisel paremalt vasakule. Üldine valem reaktsioonide jaoks on:



kus Pb – plii,

PbO₂ – pliidioksiid,

H₂SO₄ – väävelhappe,

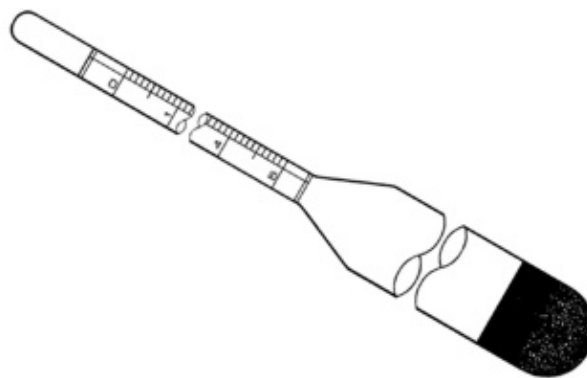
PbSO₄ – pliisulfaat,

H₂O – vesi.

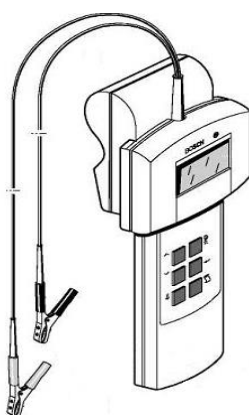
Aku elemendi nimipinge on 2,105 V, mis tähendab, et 6 elemendi puhul (kogu akuga arvestades) keskmine aku pinge on umbes 12,63 V. Pideval laadimisel võib aku klemmide peal pinge olla suurem kui 13,5 V (terve aku jaoks, või kõrgem kui 2,25 V iga elemendi jaoks). Täislaetud koormuseta aku pinge on 12,6 – 12,9 V ja tühjakslaetud – 11,8 – 12,0 V. Tühjakslaetud tarbijaga aku pinge võib olla alla 10,5 V.

2.3. Ülevaade pakutavatest plii–happe akumulaatorite monitooringusüsteemidest

Käesoleval hetkel on turul on mitmeid tooteid akumulaatorite SOC mõõtmiseks, kuid antud töös ei analüüsita neid, kuna antud olukorras võib neid lugeda mittedsobivateks. Nende hulgas on kaasaskantavad akutestred (Joonis 2.3) – need ei sobi, kuna nende kasutamine nõuab kasutaja vahetut kohalolekut mõõtmiste sooritamiseks. Lisaks on elektrolüüdi tiheduse jaoks mõeldud mõõtmisriistad (Joonis 2.2) ja –süsteemid. Need jällegi ei sobi, kuna nende rakendamine nõuab kasutaja vahetu kohalolekut või aku süsteemi ümberehitamist (tegemist on automaatsete süsteemidega, mis voolikutega võtavad elektrolüüdi proove akudest). Mittedsobivateks võib lugeda ka ampertundide lugemiseks mõeldud süsteeme – nende mõõtmisviis pakub küll väga usaldusväärseid tulemusi, kuid on aeganõudev ning viib sõidukeid töökorrast välja testimise ajaks.



Joonis 2.2. Elektrolüüdi tiheduse mõõtmise jaoks kasutatav areomeeter[2]



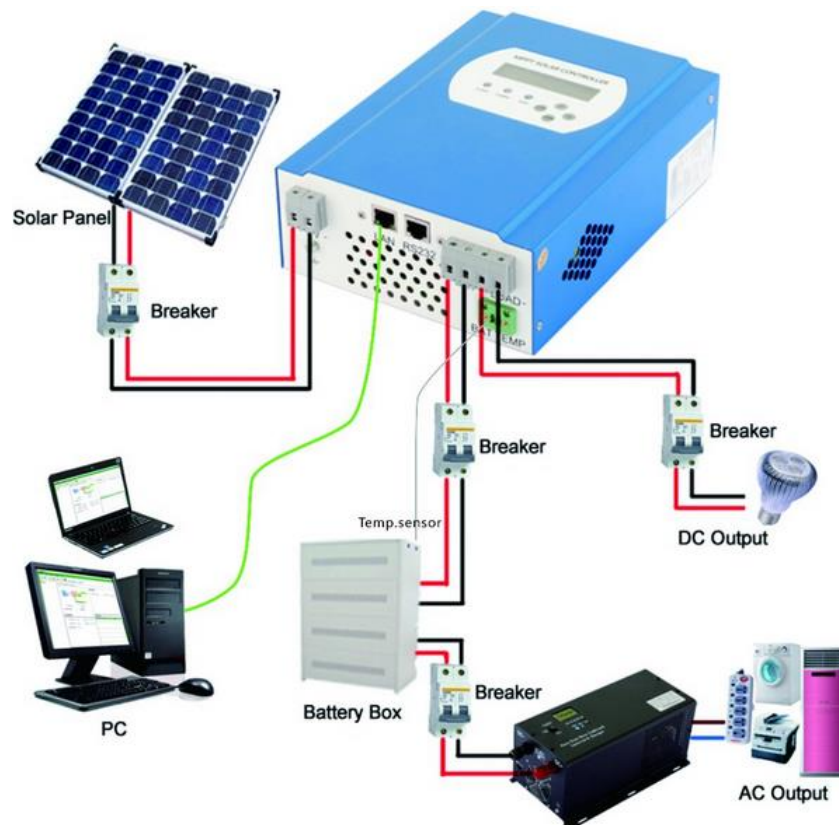
Joonis 2.3. Kaasaskantav akumulaatoritester[3]

Üks viisidest, kuidas aku SOC-d võib kontrollida, on laadimisprotsessi ja SOC jälgimise viimine täielikult automatiseeritud tasemele: selleks võib igale sõidukile ühendada intelligentse akulaadija (Joonis 2.4), mis ise kontrollib aku SOC-d ja vajadusel alustab ise laadimisprotsessi. Selline viis on kõige lihtsam ega nõua keerulist andmeedastusvõrku toimimiseks. Ainsaks vajaduseks on ehitada toitesüsteem, mis oleks võimeline tagada piisava võimsuse suure akumulaatorite hulga laadimiseks. Selline lahendus ei taga siiski pidevat ülevaadet autopargi akumulaatorite SOC-st ega anna võimalust jälgida SOC-d võrguprotokolli kaudu. On olemas keerulisemaid lahendusi, kus intelligentne laadija võimaldab ka võrguühendust, kuid need on tunduvalt kallimad, ega ei ole ettenähtud suure andmebaasi jaoks.



Joonis 2.4. Intelligentne laadija BluePower 24/8©[4]

Selleks, et jälgida akude SOC-d LAN-i või WAN-i kaudu, eriti taastuva energiaga tootmise puhul, oli välja töötatud süsteeme, mis potentsiaalselt võivad olla rakendatud antud probleemi lahendamiseks. Süsteemid, nagu LCD RS232 LAN MPPT©[5] (Joonis 2.5), on võimelised nii laadimistaset jälgima kui ka laadima ning teavet edastada oma olekust LAN protokolliga kaudu. Selliste süsteemide puuduseks võib lugeda seda, et need ei taga suure andmebaasi ülesehitust, paindlikkust ning need on projekteeritud piiratud ülesannete täitmiseks.



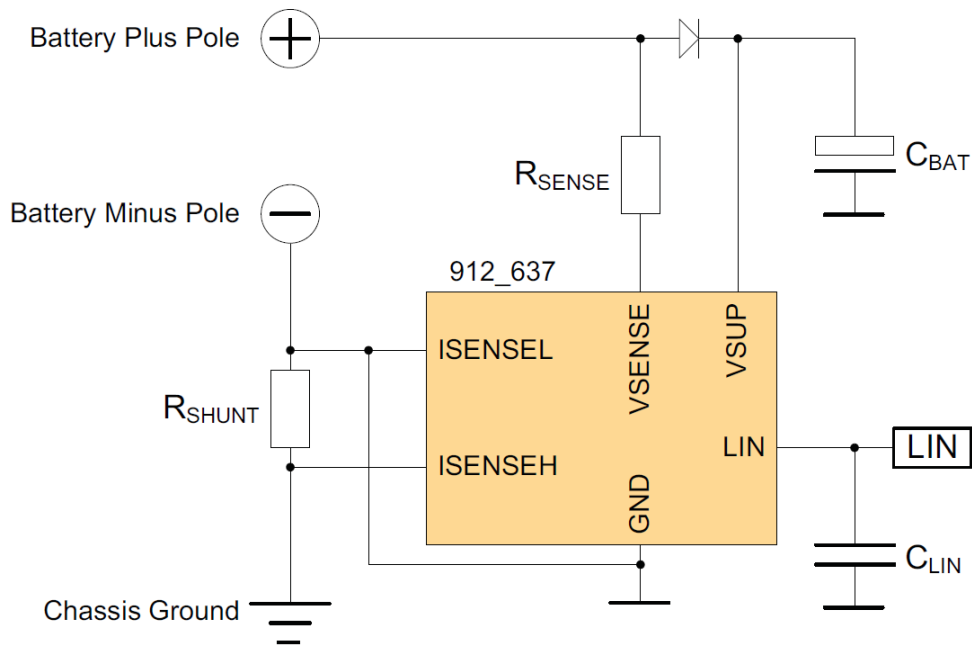
Joonis 2.5. LCD RS232 LAN MPPT© süsteem, toiteallikaks kasutatakse päikesepaneele, süsteemi olekut võib jälgida LAN protokolliga kaudu[5]

Üks kõige paindlikumatest seadmetest, mida on võimalik rakendada, on voltmeetril võrguliidesega. LAN-võrgu liidesega voltmeetril nagu DigitalMeasure TCP/IP© (Joonis 2.6). Voltmeetril, on mugavad mõõtmisseadmed, mida saaks lihtsalt grupeerida andmebaasidesse, kasutades IP (*Internet Protocol*) lahendust. Selliste voltmeetritega tuleb samaaegselt kasutada ka loogikakontrollerit (nagu PIC (*Peripheral Interface Controller*)) selleks, et mõõtmist korrigeerida.



Joonis 2.6. DigitalMeasure TCP/IP© voltmeeter[6]

Lisaks voltmeetritele on ka komplekssemaid protsessoriga seadmed, mis on ehitatud just akude kontrollimiseks ja jälgimiseks. Üks selline lahendus on mikroskeem Freescale MM912 637© (Joonis 2.7, Joonis 2.8), mis kasutab oma algoritme akude mõõtmiseks nii otseselt kui ka mõõtetakistite abil.



Joonis 2.7. Freescale MM912 637© ühendamise skeem[7]

MM912 637© kasutab suhteliselt lihtsat mõõtmiskeemi ja on suuteline edastada informatsiooni LIN-i (*Local Interconnection Network*) kaudu. Sellised kontrollid võivad olla kasulikud antud probleemi lahendamisel eelkõige sellepärast, et nendel on juba läbimõeldud ning töötavad mõõtmise ja andmeedastamise algoritmid. Kuid süsteemi toimimiseks tuleb välja töötada sidumine andmebaasiga ning LIN võrgust LAN võrku üleminek, mis on kallid ja keerulised lahendused autopargi sõidukite puhul.



Joonis 2.8. Freescale MM912 637© 98ASA00343D korpuses[7]

2.4. Plii–happe akumulaatorite laadimistaseme määramine

SOC määramiseks on vaja valida kõige sobivam viis või nende kombinatsioon, millega süsteemi seadmed hakkavad tegelema. *Journal of PowerSources* 96 (2001) 113–120's[8] on toodud kõige levinumate meetodite ülevaatus, nende hulgas – *Discharge test* (ingl. – Tühjakslaadimise test), *Ampere hour counting* (ingl. – Ampertundide lugemine), *Measurement of electrolites physical properties* (ingl. – Elektrolüüdi füüsiliste omaduste mõõtmine) ja *Open circuit voltage* (ingl. – Tühijooksu pinge test).

- **Tühjaks laadimise test.** Tühjakslaadimise test on kõige täpsem ja kindlam kõikide mõõtmisviiside seas. Testi põhimõtteks on akumulaatori tühjakslaadimine koormuse abil kontrollitud tingimustes. See, palju energiat aku suutis tagada oma täislaetud olekust kuni tühjenemiseni, ongi selle aku SOH. Kui SOH on teada, arvutatakse siis SOC samal põhimõttel, kuid akumulaator võib olla mitte täielikult laetud. Isegi kui sellise mõõtmisviisi puhul on kõige paremad mõõtmistulemused (nii täpsuses kui ka usaldusväärsuses), on testi rakendamine keeruline, kuna see nõuab aku taas laadimist, võtab seega palju aega ning see nõuab sõiduki töökorrast väljavõtmist mõneks perioodiks.[8]

- **Ampertundide lugemine.** Ampertundide lugemise mõõtmisviisi põhimõtteks on aku laadimise– ja koormamisvoolu leidmine aja suhtes. SOC määramiseks lähtutakse sellest, et on teada aku esmane laadimistaseme SOC_0 . SOC muutumine sõltub aku sisse– või väljatulevast voolust – I_{batt} leitakse voolu mõõtmise riistadega (ampermeeter või amperklambrid) aku ahelast aja dt suhtes. Selleks, et arvutada akumulaatori SOC–d, kasutakse järgnevat valemit:

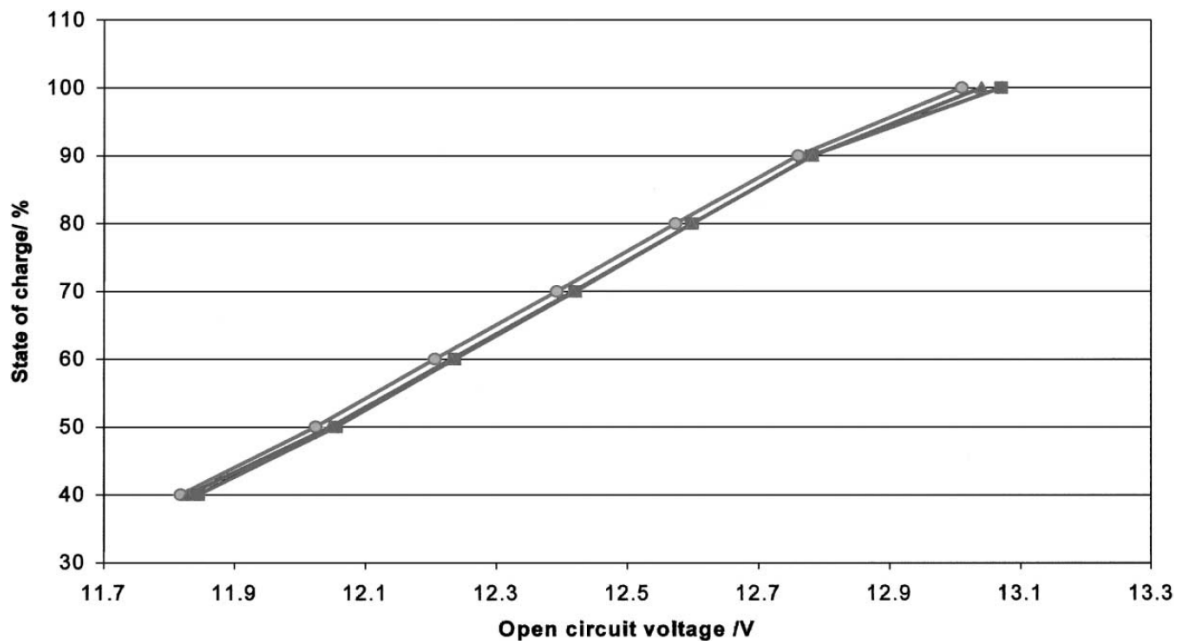
$$SOC = SOC_0 + \frac{1}{C_n} \int_{t_0}^t (I_{batt} - I_{loss}) dt; \quad (2.4)$$

kus SOC – akumulaatori laadimistaseme,
 SOC_0 – akumulaatori laadimistaseme alguses,
 I_{batt} – tarbijate või laadimise vool,
 I_{loss} – kadude vool, mida võetakse konstandina,
 t – mõõtmisperioodi algus,
 t_0 – mõõtmisperioodi lõpp,
 C_n – akumulaatori nominaalmahtuvus[8].

Selline SOC mõõtmine on liigkaudne, ega arvesta aku enda parameetreid. Seda meetodit võib kasutada seal, kus aku on pidevas töös ning selle võrgust lahti ühendamine on keeruline või võimatu. Kuna arvutamine käib aku esmase laadimistase SOC_0 ja aku nominaalmahtuvust C_n arvestades, tuleb neid näitajaid paremate tulemuste saavutamiseks iga kindla aja tagant üle mõõta. Tuleb arvestada ka sellega, et aku nominaalmahtuvus ei pruugi olla võrdne aku tegeliku mahtuvusega ladustamistemperatuuri muutumise või aku vananemise tõttu. Pika ladustamisperioodi järel tekkivate väikeste lekkevoolude mõõtmine võib aga tunduvalt mõjutada laadimistaset, olla keeruline ja nõudlik varustuse suhtes.

- **Elektrolüüdi füüsiliste omaduste mõõtmine.** SOC ja aku elektrolüüdi tiheduse vahel on lineaarsõltuvus[8], mis võimaldab mõõta SOC-d otse akust ning seega saavutada otseseid ja piisavalt täpseid mõõtetulemusi. Mõõtmist võib teostada käsitsi: akult võetakse elektrolüüdi proove ning sellega tehakse kindlaks, mis tihedusega on tegemist ja kui palju laengut on alles. Suurte süsteemide puhul on mugavam kasutada automatiseeritud protsesse, see tähendab mõõtmist ioon-kontsentratsiooni kaudu, ultraheli, tiheduse (tsirkulatsiooni) või juhtivuse abil. Kuigi füüsiliste omaduste otsese mõõtmise puhul on mõõtetulemuste täpsus ja usaldusväärsusega suurem, on iga aku SOC käsitsi kontrollimine seotud suure töömahuga. Automatiseeritud süsteemid on aga liiga tundlikud välitingimuste suhtes, sest sõidukite asukoht võib olla ka õues, või on need keerulised ja kallid suuremahulise kontrolli teostamiseks.

- **Tühijooksu pinge test.** Avatud vooluringi pinge (Ingl. – *Open circuit voltage*) on parameeter, mis võimaldab hinnata aku SOC-d, kui koormus puudub. Avatud vooluringi pinge (või tühijooksu pinge) on lineaarselt seotud aku SOC-ga (Joonis 2.9) ja võib olla väga lihtsa seadmega mõõdetud ja kasutatud SOC hindamiseks.



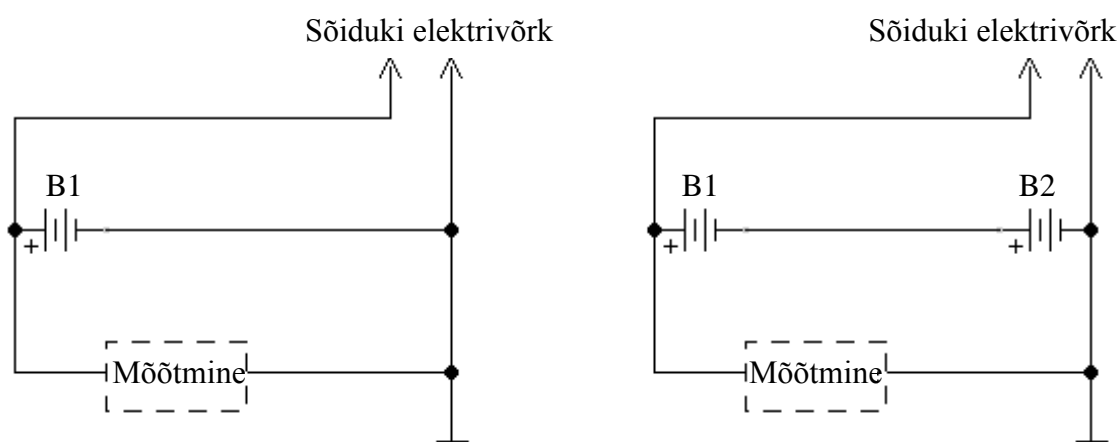
Joonis 2.9. SOC sõltuvus avatud vooluringi pinges kolme akumulaatori võrdlemisel[8]

Tühijooksu pingetest on lihtne ja kiire mõõtmisviis, mis ei nõua keerulisi vahendeid ja on kasutatav näiteks juhul, kus suur mõõtmistäpsus ei ole oluline. Mõõtmisi võib teostada, kui koormus on akust lahti ühendatud ja on eeldatud pikki tööpäuse. Mida pikem on tööpäus, seda stabiilsem on tühijooksu pinget. Mõõtmise täpsust võib küll mõjutada pindlaeng (Ingl. – *Surface charge*) mis jääb akus järele peale laadimist ja madalad voolud, mis on põhjustatud mittetäieliku tarbijate lahti ühendamisega (nt. kellad, tahhograafid ja muud tarbijad, mis jäävad sõidukites peale väljalülitamist tööle). Pindlaengut saab eemaldada kasutades akut lühiaegset koormamates, millega võrdsustatakse plaatide laengud. Seejärel mõõdetud aku pinget kirjeldab juba täpsemalt SOC-d.

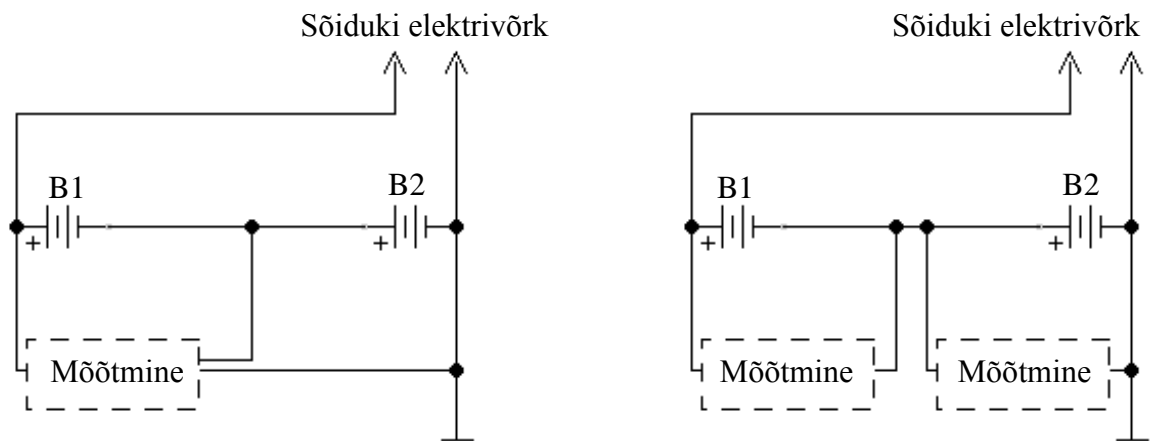
3. MONITOOINGUSÜSTEEMI PROJEKTEERIMINE

3.1. Süsteemi mõõteahela ülesehitamise printsiibid

Selleks, et ehitada sõiduki aku jälgimissüsteemi, tuleb arvestada asjaoluga, et sõidukitel on levinud kaks erinevat toitesüsteemi: 12 V või 24 V alalist pinget. 12 V süsteem on levinud sõiduautodel, 24 V süsteem on kasutusel veoautodel ja maastikusõidukitel. 24 V alalist pinget saavutamiseks pannakse reeglina kaks 12 V akut jadaühendusse. Esiteks on vaja otsustada, kuidas on parem ja otstarbekam jälgida akumulaatorsüsteemi, kas iga aku eraldi või mõlemad korraga (Joonis 3.1). Kahe aku korraga jälgimine lihtsustab süsteemi ülesehitust, kuid mõõtmistulemused ei ole nii täpsed. Pidades silmas, et iga mõõtmisvahendi hind ja kompleksus peavad olema madalad süsteemi paindlikkuse mõttes, on 12 V süsteemi mõõtmisvajadusega parem kasutada eraldatud akumulaatori mõõtmist. Seejärel jääb valida, kuidas mõõta kahte akut korraga: ühe kompleksse vahendiga või kahe eraldi seadmega (Joonis 3.2).

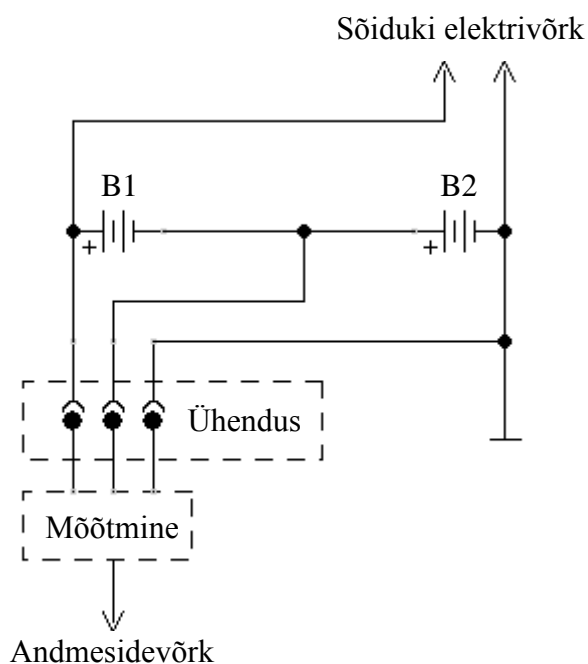


Joonis 3.1. 12 V (parem) ja 24 V (vasak) süsteemide mõõtmine kasutades ühte mõõteriista. B1 ja B2 – akumulaatorid 1 ja 2 vastavalt



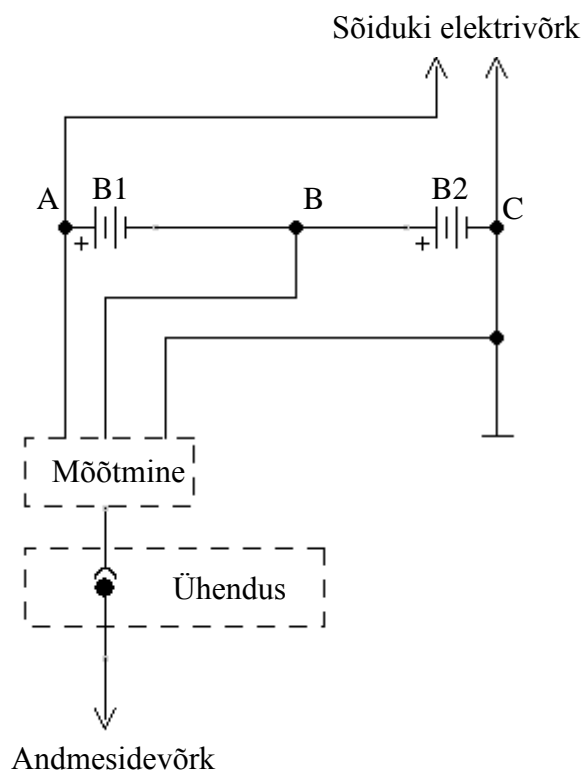
Joonis 3.2. 24 V süsteemi mõõtmine kasutades ühe mõõteriista ja kahte eraldi seadet – Mõõtmine1 ja Mõõtmine2. B1 ja B2 – akumulaatorid 1 ja 2 vastavalt

Kahe aku mõõtmisviis, kas ühe komplektse vahendiga või kahe eraldi seadmega, valitakse lähtudes mõõtmisvahendist ning võrgu üldisest ülesehitusest. Üks tähtsamatest punktidest on see, kus luuakse ühendus sõiduki ja võrgu vahel. Ühelt poolt võib ühendada sõiduki aku läbi pistiku mõõtmisseadmega: see on võimalik kasutades ühe mõõteriista, mis on ühendatud andmesidevõrguga (Joonis 3.3).

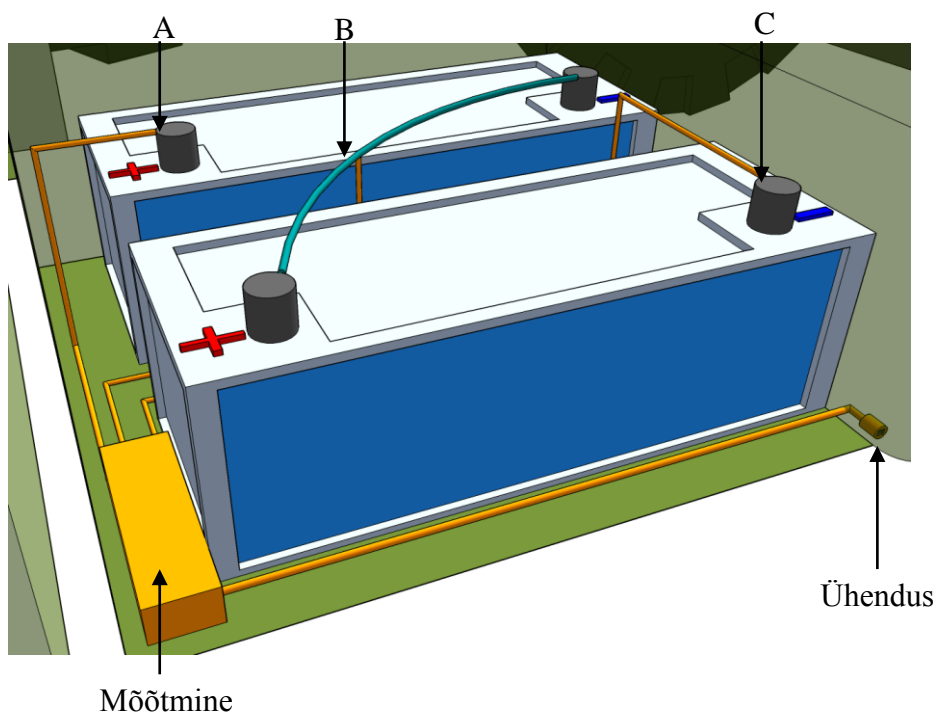


Joonis 3.3. 24 V süsteemi mõõtmine kasutades ühe mõõteriista, ühendus toimub lahtivõetavate kontaktidega akumulaatori ja mõõteriista vahel

Joonise 3.3 peal toodud mõõtmisviisi puuduseks on kasutajate kokkupuude sõiduki jõuahelatega. Arvestada tuleb ka sellega, et ühe mõõteriistaga mitme sõiduki aku kontrollimine võib tuua selle mõõteriista ülekuumenemise, kui ei tehta ajutisi pause mõõtmiste teostamises – see oma korda pikendab mõõtmisprotsessi. Efektivsem oleks kasutada katkestamata ühendust aku ja mõõteriista vahel. Selleks, et sõiduk saaks liikuda, võib ühendust luua lahtivõetava andmesidevõrguna (Joonis 3.4 ja Joonis 3.5) – see võimaldab vältida vajadust kasutajatel voluvõrguga tegeleda. Lisaks jääb mõõtmisahel muutusteta selle ehitamise hetkest, mis teeb mõõtmiste tulemuse usaldamisväärsimateks. Ühendus andmesidevõrguga peab olema teostatud nähtavalt tähistatud kohas ilmastikukindla pistiku või pistikupesaga. Sõiduki pikaajaliseks parkimiseks ühendatakse mõõtmisahel andmesidevõrguga, mille kaudu mõõtmistulemused salvestatakse andmebaasi.



Joonis 3.4. 24 V süsteemi mõõtmine kasutades ühte mõõteriista, ühendus on lahtivõetavatega kontaktidega mõõteriista ja andmesidevõrgu vahel



Joonis 3.5. Mõõtmisahela vaade sõiduki akumulaatorkastis – tähistatud ühenduskohad A, B ja C; Mõõteriist – Mõõtmine; Ühenduspistik andmesidevõrguga – Ühendus

3.2. Süsteemi andmesidevõrgu ülesehitamise printsiibid

Iga sõiduki juures asuva mõõteahela mõõdetud ja edastatud parameetrite informatsiooni salvestamiseks ja jälgimiseks on vaja ehitada andmesidevõrku, mis tagab töökindla ja efektiivse andmeedastusviisi. Andmesidevõrgu projekteerimisel tuleb arvestada asjaoluga, et esmane objekt selles võrgus on analoogne suurus, mida mõõdetakse akude pealt. Seda tuleb digitaliseerida ja edastada andmebaasidele ja/või TCP/IP ühendusliidesega kliendi terminaalidele. Selleks, et võrk oleks ehitatud kõige ratsionaalsemal viisil, tuleb valida ja paigaldada järgmised süsteemi osad:

Andmebaas – mäluseade TCP/IP liidesega, mis on mõeldud mõõtetulemuste kogumiseks ja hoidmiseks. Võib olla lokaalne, mis salvestab lähedalt paigutatud mõõteobjektide informatsiooni või keskne, mis on mõeldud terve andmesidevõrgu peale. Andmebaasi hallatakse tarkvara rakendusega, mis võimaldab andmeedastust mõõtmisobjektidega ja salvestada mõõtmistulemusi.

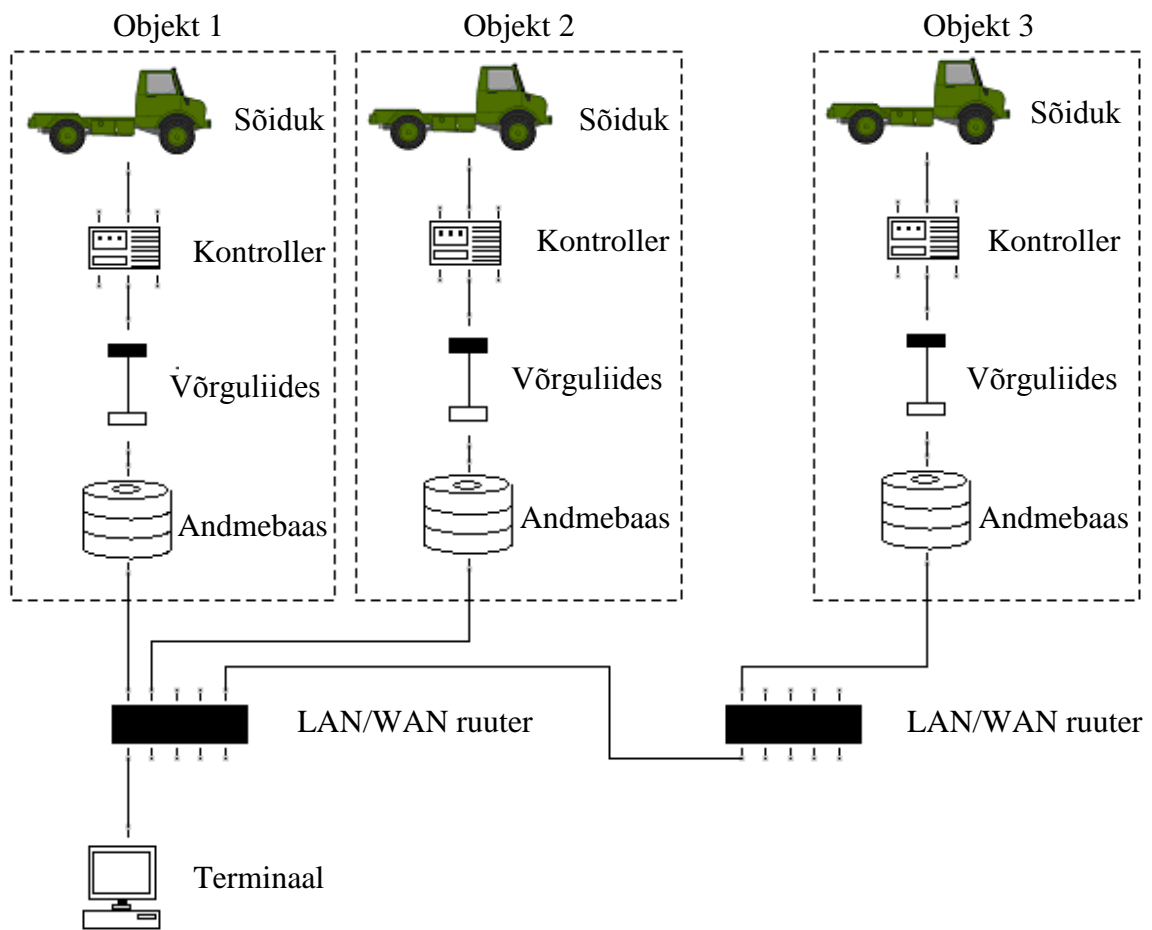
Terminaal – kasutajaliidesega arvuti, mis annab võimaluse kasutajatele või spetsialistidele jälgida andmebaasis kogutud informatsiooni. Mugavamaks ja kiiremaks tulemuseks kasutatakse TCP/IP ühendust ja graafilist info kujutamist.

LAN/WAN ruuter – TCP/IP (*Transmission Control Protocol/Internet Protocol*) võrgu seade, mis võimaldab seadmete vahelist andmeedastust kohaliku võrgu kaudu või internetis. Tagab mitme seadme paralleelselt adresseeritud ühendust.

Kontroller – programmeeritav automaatikaseade, mis võimaldab informatsiooni kogumist, salvestamist ja edastamist seadmetele, millel TCP/IP liidest ei ole. Võib olla varustatud TCP/IP lõpp väljundiga, et kontakteeruda andmebaasidega. Kontroller ühendub mõõtmisahelatega automaatika süsteemi protokollide kaudu, mis võivad olla kas iseprogrammeeritud või juba kasutusel olevad (näiteks sõidukites kasutatavad protokollid LIN või CAN (*Controller Area Network*), või hoonete automaatikas kasutatavad protokollid BACnet (*Building Automation and Control Network*) või muud.

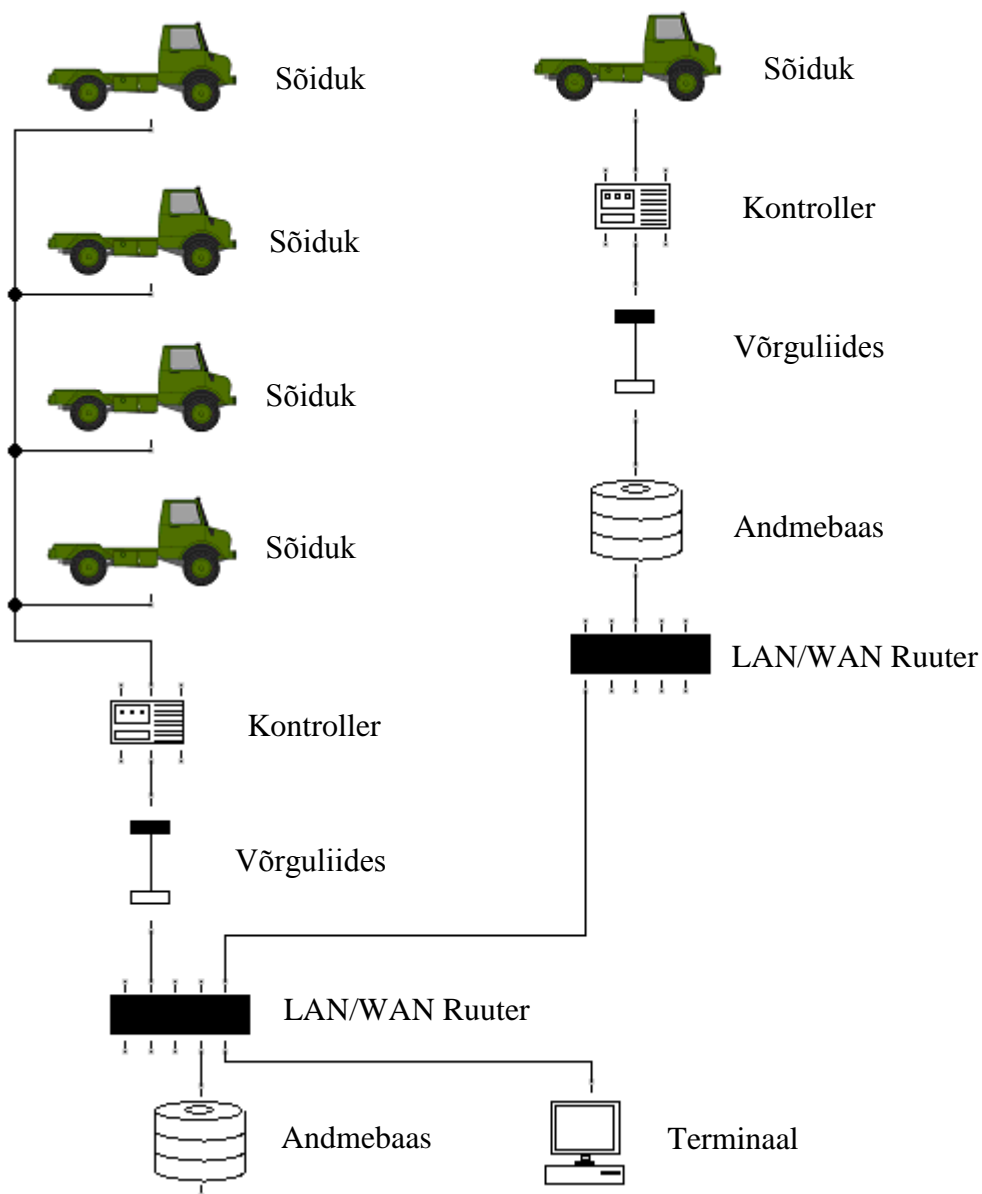
Võrguliides – seade, mis võimaldab ühe võrguliidesega seadet ühendada teise võrguga. Kasutakse siis, kui seadmel on puudu vajalik võrguliides. Võib olla sisse ehitatud (nagu TCP/IP liidesega kontrolleritel) või väline (Nagu Baby LIN[®]). Sõiduk – mõõtmiste objekt. Varustatud mõõtmisahelaga ja lahtivõetava ühendusega (Joonis 4.4).

Kompleksne võrk (Joonis 3.6) – ehitatud komplekssete seadmete baasil. Süsteemid, mis asuvad autode küljes, koosnevad mõõtmisahelatest, kontrolleritest ja andmebaasidest. Sel juhul on igal mõõtmisobjektil oma kohalik andmebaas, kuhu kogutakse mõõtmistulemusi ning kust neid saab vaadata TCP/IP kaudu terminaalist. Kasutades antud lahendust, tagatakse mõõtmistulemuste salvestamine ka siis, kui võrguühendus katketakse, seega suurendades süsteemi töökindlust. Selle süsteemi puhul kasvab ka iga uue objekti loomise hind, kuna iga sõiduki juures vajatakse kontrollerit ja kohalikku andmebaasi. Probleemid võivad tekkida sõiduki kasutamise äärmuslikes tingimustes, mis võivad kahjustada kohalikku võrku moodustavate seadmete keerulist või tundlikku struktuuri. Süsteemi laiendamine on lihtne tänu sellele, et kasutatakse LAN/WAN võrgu ühendusi.



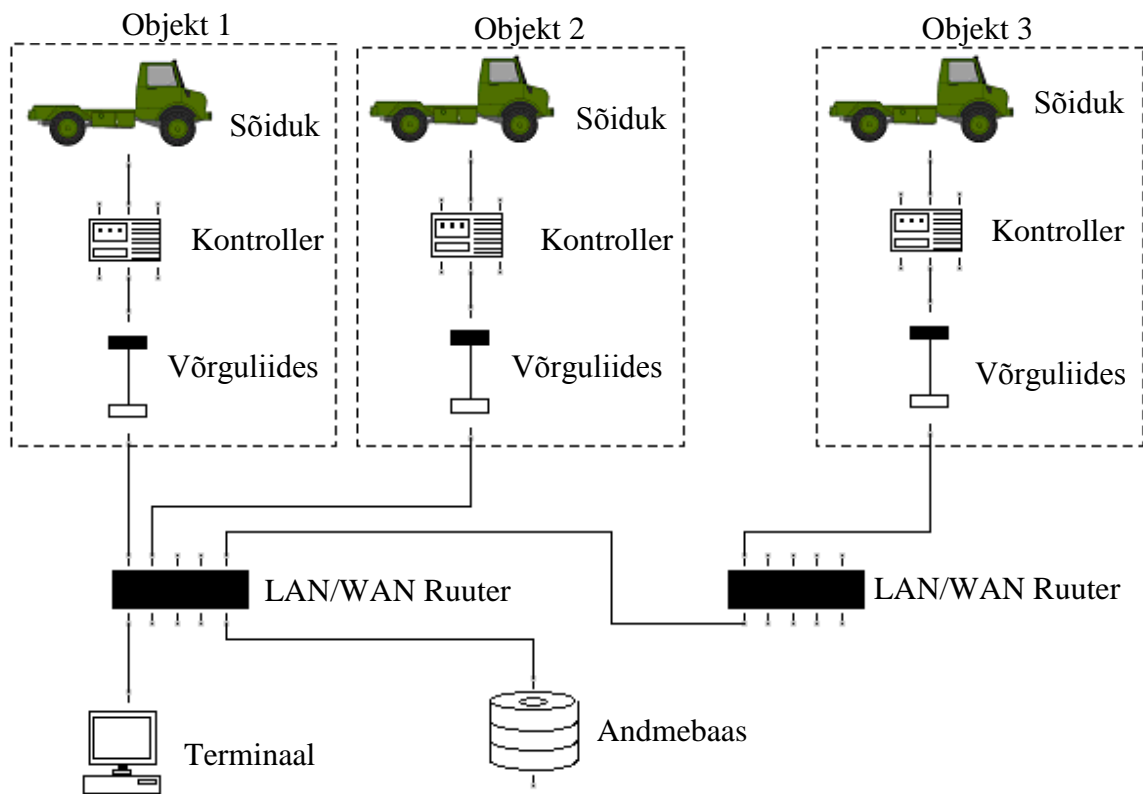
Joonis 3.6. Kompleksne võrk – iga objekti juures on oma kontroller võrguliidese ja andmebaasiga. Kaugühenduse puhul (Objekt3) kasutakse WAN-ühendust

Ahelike võrk (Joonis 3.7) – võrk, kus objekte mõõtmisahelad ühendatakse adresseeritud ahelatesse, mis on ehitatud automaatika sideprotokollide printsiibil. Objektide ja kontrolleri vaheliseks suhtlemiseks kasutatakse LIN, CAN, või hoonete automaatikas kasutatavaid protokolle – BACnet. Igal kontrolleril on TCP/IP liides võrku ühendamiseks. Andmebaasid võivad olla kohalikud või üldised vastavalt vajadusele. Antud lahenduses on kasutusel lihtsamad mõõteseadmed, mis asuvad iga sõiduki juures ja võrk, mis ühendab neid kontrolleritega. Sellise süsteemi puhul on keerulisem programmeerida kohalikku võrku võrreldes LAN-võrguga, lisaks võivad muundurid kontrollerite ja LAN võrgu vahel olla suhteliselt kallid (nagu BABY LIN®, LIN/USB muundur).



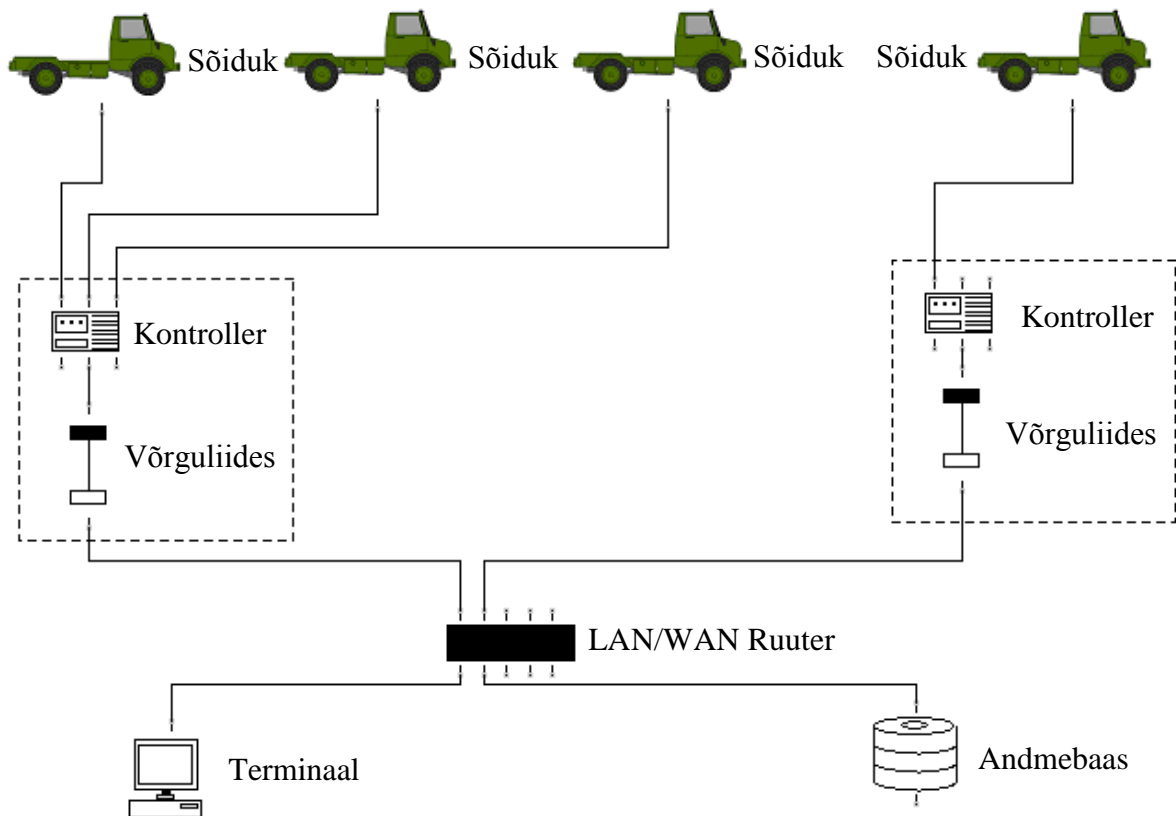
Joonis 3.7. Ahelike võrk – vasakul ahelas üldise ja parema ahelas kohaliku andmebaasiga

Nagu kompleksse võrgu puhul, meenutab see võrk ülesehituse poolest hajutatud võrgu ülesehitust (Joonis 3.8). Erinevuseks on see, et hajutatud võrgu puhul ei ole vaja iga sõiduki juures hoida kohalikku andmebaasi, mis teeb sellise viisi palju lihtsamaks ja odavamaks. Hajutatud võrgu puhul kasutakse TCP/IP-d võrgu mõõtmisobjektide ühendamiseks ja adresseeritud üldise andmebaasiga suhtlemiseks. Üks olulisimaid eeldusi on lihtsa hajutatud võrgu ehitamine, sest seal kasutakse laialt levinud protokollide seadmete vaheliseks suhtlemiseks. Suurematest miinustest võib välja tuua keerulist kohalikku seadet, mis peab asuma iga objekti juures ning koosnema kontrollerist koos võrguliidesega.



Joonis 3.8. Hajutatud võrk

Viimaseks võib tuua kontrolleripõhist võrku (Joonis 3.9), selle põhielemendiks on loogikakontroller TCP/IP liidesega. Võrgu toimimiseks programmeeritakse kontrolleri ja mõõtmisahela vahelist suhtlemisviisi, mis on keeruline kuid võimaldab saavutada lahenduse, mis on kõige sobivam süsteemi toimimiseks. TCP/IP ühendust võimaldavaid kontrollereid on suhteliselt palju, üks nendest on Mitsubishi Electric FX[©] seeria kontroller, kus on LAN moodulitega võimalik kontrollerit ühendada lokaalse võrguga. Antud võrgu puhul on lihtne ehitada kohalikku võrku ning selleks kasutakse vähem varustust. Võrgu laiendamiseks on aga vaja lisada kontrollereid isegi ühest sõidukest koosneva autopargi jaoks.

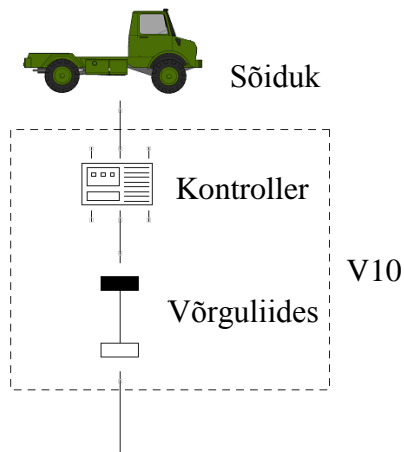


Joonis 3.9. Kontrolleripõhine võrk

3.3. Seadmete valik

3.3.1. TCP/IPV10 voltmeeter/termomeeter/relee©

Üks sobivaimaid variante on kasutada ühte seadmesse sisse ehitatud mõõteriistade ja võrguliidese massiivi, mis on loodud DigitalMeasure firma poolt toodetud TCP/IP V10© Voltmeeter/Termomeeter/Relee (edasi V10). V10 võimaldab saata voltmeetri ja termomeetri mõõtetulemusi läbi TCP/IP protokoll (Joonis 3.10) ning juhtida releed kliendi arvutist.



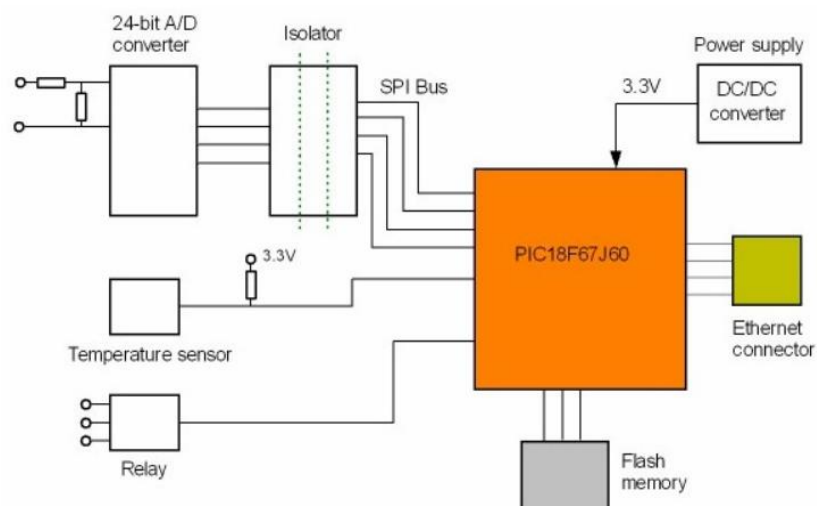
Joonis 3.10. V10'ga mõõtmise printsiip

V10 abil on võimalik mõõta aku pinget ja kohandada seda mõõdetud temperatuuri järgi selleks, et saada võimalikult täpseid andmeid aku laadimistaseme kohta. Lisaks sellele võimaldab relee juhtida koormust, lubades läbi viia pindlaengu eemaldamise protseduuri, kui selleks peaks vajadus tulema (Joonis 3.11).



Joonis 3.11. V10 trükkplaat[6]

V10 on isoleeritud 24-bitise Analoo/Digitaalse pinget mõõtva PIC18F67J60 mikrokontrolleriga, millele on külge ühendatud digitaalne DS18B20 temperatuuriandur ja relee (Joonis 3.12 ja Tabel 3.1).

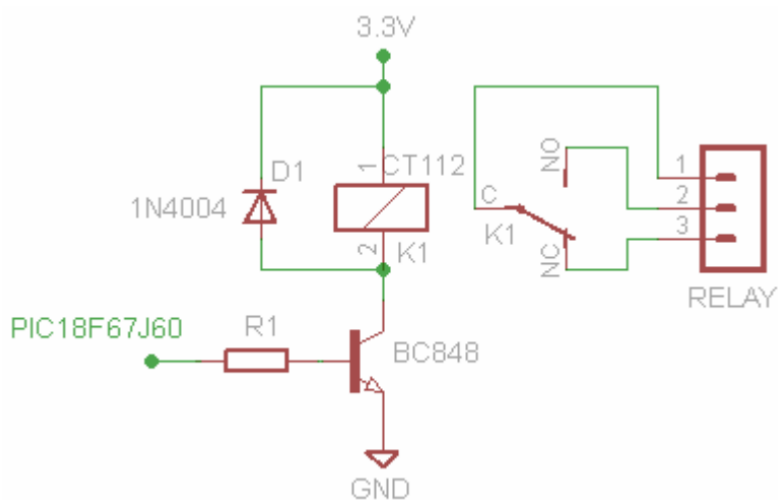


Joonis 3.12. V10 komponentide ühendusdiagramm[6]

Tabel 3.1. V10 põhilised andmed[6]

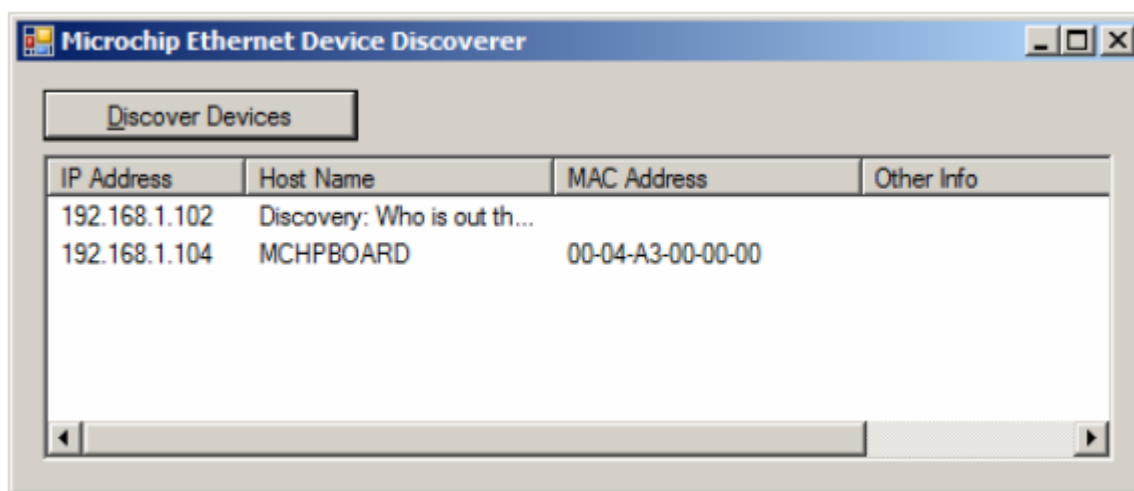
Toide:	5–30 V DC
Voolutarve:	160 mA / 5V
Plaadi suurus:	78 x 67 mm
DC Voltmeeter	
Mõõtepiirkond:	0 – 50 V
Resolutsioon:	0,01 V
Sisendimpedants:	1,0 MΩ
Kiirus mõõtmisel:	2 x sekundis
Termomeeter	
Mõõtepiirkond:	– 25 C° kuni + 120 C°
Resolutsioon:	0,1 C°
Kiirus mõõtmisel:	iga 5 sekundit
Relee	
Maks. Lülitamispinge:	300 V AC (60 V DC)
Maks. Lülitamisvool:	3 A
Maks. Lülitamisvõimsus:	750 VA (90 W)

V10-l on sisse ehitatud FLASH mälu paigaldamisvõimalus, millega saab luua veebiserverit, kui andmebaas nõuab seda. Lisaks on olemas võimalus kasutada digitaalseid sisendeid, mis on kujutatud kontaktidena J3 ja J4 (Joonis 3.11). Relee on ühendatud eraldi seisvate kontaktidega 1, 2, ja 3 (Joonis 3.13) ning on juhitud mikrokontrolleri PIC18F67J60 poolt läbi NPN transistori BC848.



Joonis 3.13. V10 relee ühendusskeem[6]

V10-ga on komplektis tarkvara, mis sisaldab järgmisi tarkvararakendusi: MCHPDetect©, MCHPEXplorer© (Joonis 3.14) ja MPFS2©. MCHPDetect.exe programm näitab, kas veebivoltmeeter on edukalt võrku ühendatud. Samuti näitab see IP-aadressi, mis on eraldatud DHCP serverist. MCHPEXplorer.exe kuvab ühendatud seadmeid ja nende MAC aadresse. MPFS2 on vahend selleks, et laadida lehekülgi üles EEPROM-i.



Joonis 3.14. V10 MCHPEXplorer'i© liides[6]

Mõõdetud väärtusi saab lugeda kasutades veebibrauserit, TCP klienti, SNMP (*Simple Network Management Protocol*) klienti ja XML (*Extensible Markup Language*) failide lugemist. Veebilehitseja on üks kõige lihtsamatest viisidest andmete lugemiseks V10 pealt. Selleks võib kasutada mis tahes AJAX (*Asynchronous Java Script and XML*) tehnoloogiat toetavat veebibrauserit (näiteks IE (*Internet Explorer*©), Firefox©, Chrome©), . Andmed edastatakse brauserile XML – failina, mida loetakse Java Script'i abil. V10 toimib ka TCP serverina, kasutades selleks porti 9760, mis tähendab, et andmed saab lugeda ka TCP kliendi abil. Lähtekood TCP kliendi jaoks on esitatud tarkvara pakettis.

V10 sisaldab SNMP agentit, tänu sellele võib kasutada ükskõik missugust SNMP klienti V10 funktsioonide kontrollimiseks. Seade toetab SNMP 2.0 versiooni. Kui V10 suhtleb brauseriga, siis saadab see oma Status.xml faili, mida saab lugeda ilma brauserita: tarkvarapakettis on selleks tarkvara, mis loeb seda XML-faili otse. See annab võimaluse lugeda andmed skripti abil ning edastada neid andmebaasi vajalikul kujul.

Üleüldiselt V10© on üks parematest ja odavamatest valikutest selleks, et mõõta akumulaatorpatareide SOC-d kaugjuhtimisega – sellel on mugav sisse ehitatud TCP/IP

liides, mis nõuab ainult TCP/IP ruuteri või *switchi* selleks, et ühendada seadmed võrku. RJ45 kaabli pistik on mugav eelkõige sellepärast, et RJ45 kaablid on väga levinud. Minimaalse mõõtmisülesannete täitmiseks V10-l kõik olemas – relee, mis võib juhtida koormust pindlaengu eemaldamiseks, voltmeeter selleks, et pidevalt mõõta aku pinget, ja termomeeter temperatuuri jälgimiseks.

Selle riistvara puudusena võib välja tuua programmeerimisvõimaluse puudumist ja sellest tingitud paindlikkuse puudumise. Puuduvad lisa sisendid ja väljundid, mis tähendab, et pole võimalik juhtida muid protsesse peale pindlaengu eemaldamise. Puuduseid esineb ka V10 hankimisel, sest seda mudelit turustab ainult tootja ise ning selle tellimisega kaasneb pikk ooteaeg, mis omakorda on tingitud vähesest tootmisvõimsusest.

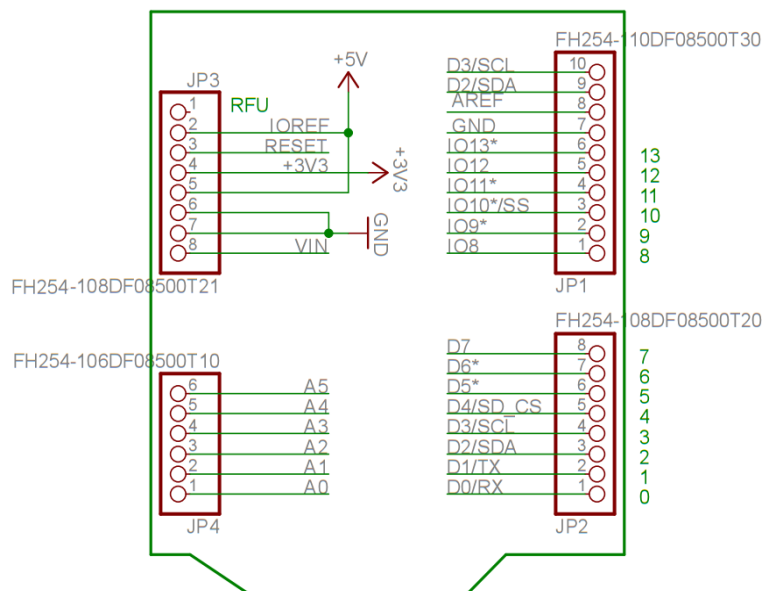
3.3.2. Arduino Leonardo ETH©

Arduino Leonardo ETH© (Joonis 3.15 ja Joonis 3.16, Tabel 3.2) on ATmega32U4© mikrokontrolleri baasil põhinev ja uut W5500 TCP/IP Ethernet kontrolleri sisaldav programmeeritav kontrolleri. Arduino Leonardo ETH – l© on 20 digitaalset sisendit/väljundit (millest seitse on võimalik kasutada PWM (*Pulse Width Modulation*) väljundina ja 12 kui analoogsisendeid). Plaat sisaldab ka 16 MHz kvartsostsillaatorit, RJ45 kaabli ühenduspistikut, mikro USB pistikut ja alalisvooluga toite sisendit.



Joonis 3.15. Arduino Leonardo ETH©[9]

Arduino Leonardo saab toidet mikro USB–ühenduse kaudu või välisest toiteallikast, allikas valitakse automaatselt. Väline toide võib tulla kas vahelduvvoolu–alalisvoolu adapterist või akust, adapter ühendatakse 2.1 mm pistiku abil. Arduino Leonardo ETH–d saab toita ka POE (*Power Over Ethernet*) kaudu, kasutades Arduino POE© adapterit.

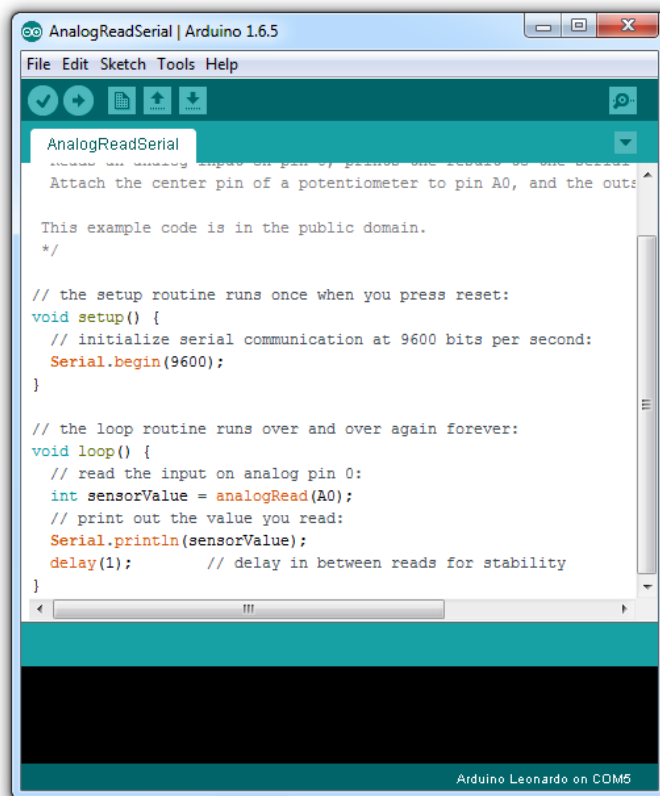


Joonis 3.16. Arduino Leonardo ETH© kontaktid ja nende asukohad[9]

Tabel 3.2 Arduino Leonardo ETH põhilised andmed [9]

Toide:	7–12 V (DC)
Mikrokontroller:	ATmega32u4
Plaadisuurus:	68,6 x 53,3 mm
Tööpinge:	5 V Max
Resolutsioon:	1023 bit
Digitaalsete sisendite arv:	20
PWM kanalitearv:	7
Analoog sisendite arv:	12
Taktsagedus:	16 MHz
Digitaalse I/O vool:	40 mA Max
3.3 V väljundvool:	50 mA Max

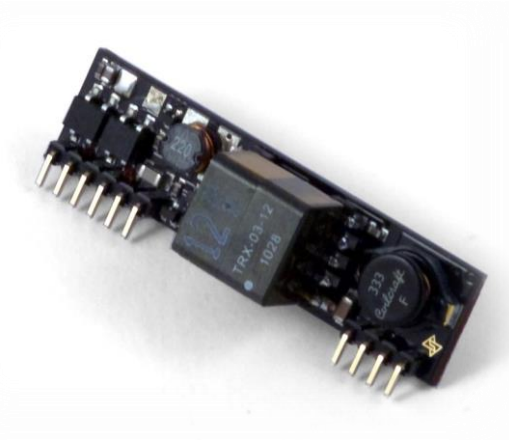
Selleks, et programmeerida Arduino Leonardo't[©] ja teisi Arduino[©] plaate, kasutakse spetsiaalset tarkvara – IDE[©], Joonise 3.17 peal, mis programmeerib kontrollereid läbi virtuaalse COM–porti. Programmeerimine on teostatud C programmeerimiskeeles.



Joonis 3.17. IDE[©] programmeerimiskeskonna liides[9]

Sarnane IDE tarkvaraga on Processing[©] programm, mida võib kasutada selleks, et luua rakendusi, mis töötavad arvutis koos Arduino kontrollritega – need koguvad infot, visualiseerivad mõõtmiste tulemusi ja võimaldavad kontrolleri juhtimist. Lähtudes sellest, et Arduino Leonardo ETH[©] kontrolleri on paindlikum ja omab rohkem võimalusi kui V10 TCP/IP[©] voltmeeter, on see parem valik antud töös. Selle kontrolleri ülemaailmne kasutamine, lähtekoodid ja lihtne kättesaadavus teeb seda veelgi paremaks variandiks masskasutuses, kus süsteem on koostatud mitmekümnest mõõtmisobjektist.

Arduino Leonardo ETH© võimaldab kontrollida ka plii-happe akumulaatorite laadimisprotsessi, sest siis tekib vajadus programmeerida rohkem sisendeid protsessi jälgimiseks ja väljundeid protsessi juhtimiseks. Kasutada saab PoE toitemoodulit (Joonis 3.18), mis võimaldab toita nii kontrolleri kui ka teisi mõõtmiseks vajalikke seadmeid (üheks nõudeks oli, et ei kasutata mõõteobjekti laengut mõõteriista toitmiseks mõõtmisobjekti laengu kiire tühjenemise tõttu).

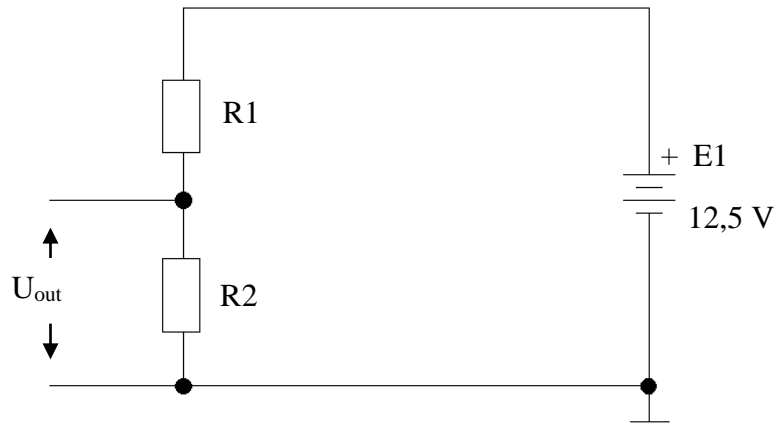


Joonis 3.18. IDE Arduino Leonardo ETH© PoE toitemoodul[9]

3.4. Mõõtmisahela projekteerimine ja katsetamine

3.4.1. Potentsiaali jagamise arvutamine ja katsetamine

Selleks, et Arduino Leonardo ETH© (edaspidi ka – kontrolleri) saaks lugeda plii-happe akumulaatorist pinget, on vaja arvutada pingejagur: plii-happe akumulaatori nimipinge on 12,6 V kuni 13 V (täislaetuna), kontrolleri võib aga lugeda suurused kuni 5 V oma analoog sisendist. Selleks et muuta pinget, rakendatakse potentsiaali jagamise printsiipi – kasutatakse kahte takistit: võimsus on mõõtmisel väike ning sel kujul on see kõige mugavam ja lihtsam.



Joonis 3.19. Pingejagur mõõtmiseks

Joonise 3.19 peal on toodud patarei E1, mille pinget jagatakse kahe takisti vahel – R1 ja R2. U_{out} on jagatud pinge, mida loetakse kontrolleri analoog sisendiga. Pinge jaotakse takistite R1 ja R2 suhteliste väärtuste muutmiseks, valemiga:

$$U_{out} = \frac{R2}{R1+R2} \cdot U_{E1}, \quad (3.1)$$

kus $R1, R2, R3$ – Joonise 3.19 peal vastavalt tähistatud takistid,
 U_{out} – Joonise 3.19 peal tähistatud väljundpinge,
 U_{E1} – Joonise 3.19 peal tähistatud elektromotoorjõu E1 pinge.

Esimeseks sammuks on määrata takistite R1 ja R2 summaarne takistus. See takistus peab olema piisavalt suur selleks, et tekkiv vool ei oleks nii suur, et see nähtavalt vähendaks uuritava objekti SOC–d. Tavaliselt on voltmeetrite sisend takistus ligikaudu 1 M Ω . Sellest lähtudes valime

$$R1 + R2 \approx 1 \text{ M}\Omega, \quad (3.2)$$

kus $R1, R2$ – Joonise 3.19 peal vastavalt tähistatud takistid.

Järgmine tingimus on sisend– ja väljundpingete määramine. Kuna kontrolleri maksimaalne talutav analoogsisendi pinge on 5 V, tuleb arvestada asjaoluga, et võib jääda varu umbes 10 % ulatuses, ning tuleb jälgida, et ohtlikud pinge võnkumised ei kahjustaks kontrolleri. Seega

$$U_{out} = 5 \cdot 0,9 = 4,5 V, \quad (3.3)$$

kus U_{out} – Joonise 3.19 peal tähistatud väljundpinge.

Sisendpinge U_{E1} on laadimisel, maksimaalselt 19 volti, kuid selleks, et kaitsta süsteemi selle eest, et inimliku vea pärast ühendataks külge kahe aku klemmid, võib maksimaalse sisendpingega arvestada 27 V ulatuses. Seega

$$U_{E1} = 27 V, \quad (3.4)$$

$$\frac{U_{E1}}{U_{out}} = \frac{27 V}{4,5 V} = 6, \quad (3.5)$$

kus U_{out} – Joonise 3.19 peal tähistatud väljundpinge,

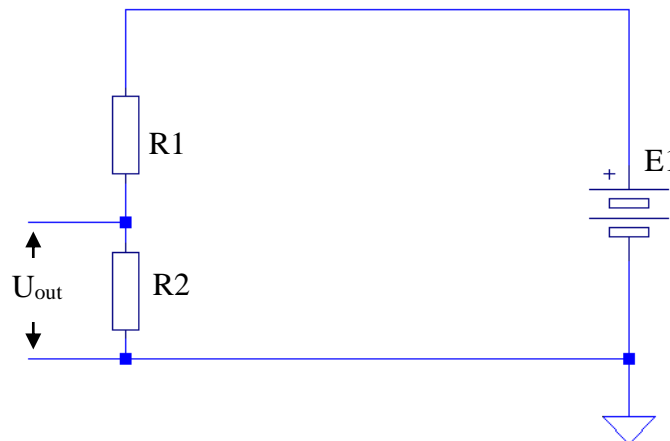
U_{E1} – Joonise 3.19 peal tähistatud elektromotoorjõu E1 pinge.

R1 valiti 820 k Ω takistina ja R2 180 k Ω takistina, seetõttu on arvestatav R1 ja R2 summaarne takistus 1 M Ω . Takistid valiti Velleman© 0.25 W takistite komplektist. Kuna takistus takistitel võib varieeruda 5 % ulatuses, mõõdeti neid üle oommeetriga ning saadi tulemused $R1 = 801 k\Omega$; $R2 = 176 k\Omega$,

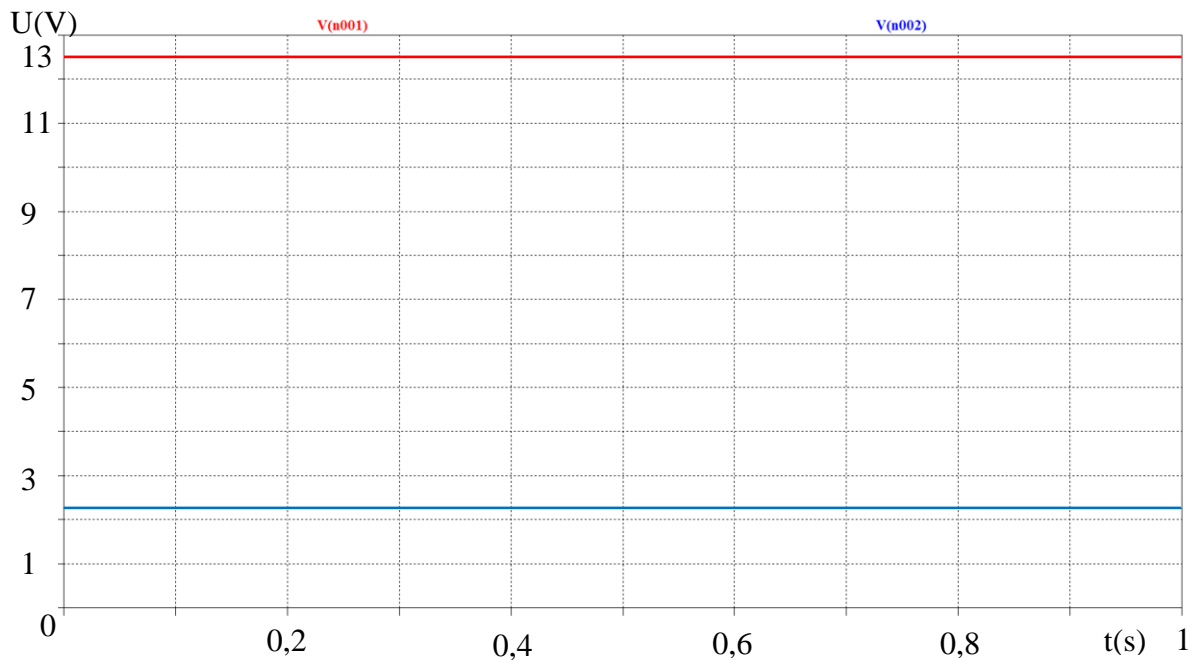
$$(3.6)$$

kus R1, R2, – Joonise 3.19 peal vastavalt tähistatud takistid,

Antud tulemustega tehti skeemi simulatsioon LT Spice keskkonnas (Joonis 3.20) 12,5 V E1 pinge puhul.

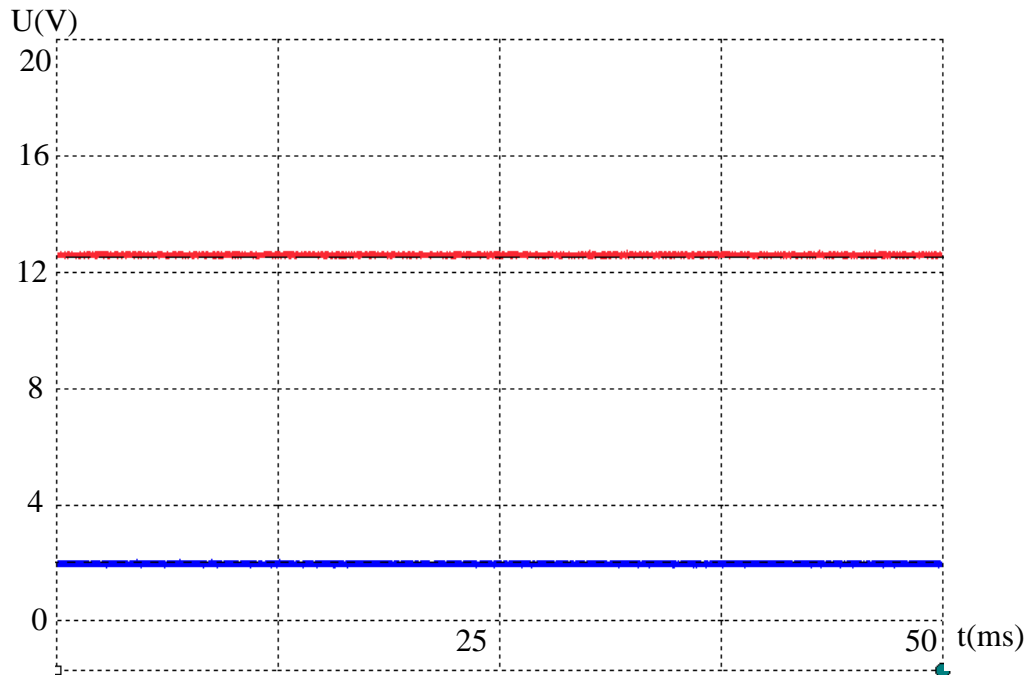


Joonis 3.20. Potentsiaali jagamise simuleerimine LT Spice keskkonnas



Joonis 3.21. LT Spice keskkonnas simuleeritud potentsiaali jagamine – punane(V(n001)) – sisendpinge U_{E1} , sinine(V(n002)) – väljundpinge U_{out}

Simuleerimisel tulid pinged Joonisel 3.21 toodud kujul – pinged olid tulemuste järgi $U_{E1} = 12,5$ V ja $U_{out} = 2,152$ V. Vool, mis tuli läbi takistite oli arvutuslikult vähem kui $14 \mu\text{A}$. Antud tulemused on lubatud piiride sees ja näitavad seda, et arvutus oli tehtud õigesti. Kuna LT Spice keskkonnas simuleerimisest ei saa teha täielikke järeldusi skeemi töö kohta, tuleb korraldada ka reaalse skeemi katsetamise. Reaalse skeemi katsetamiseks kasutakse ülaltoodud Velleman© 0.25 W takisteid, digitaalset ostsiloskoopi PicoScope 5244B, maketeerimislaua MIKROELEKTROONIKA 400© ja laboratoorset toiteplokki. Simuleeritakse skeemi, mis on toodud Joonise 3.22 peal.

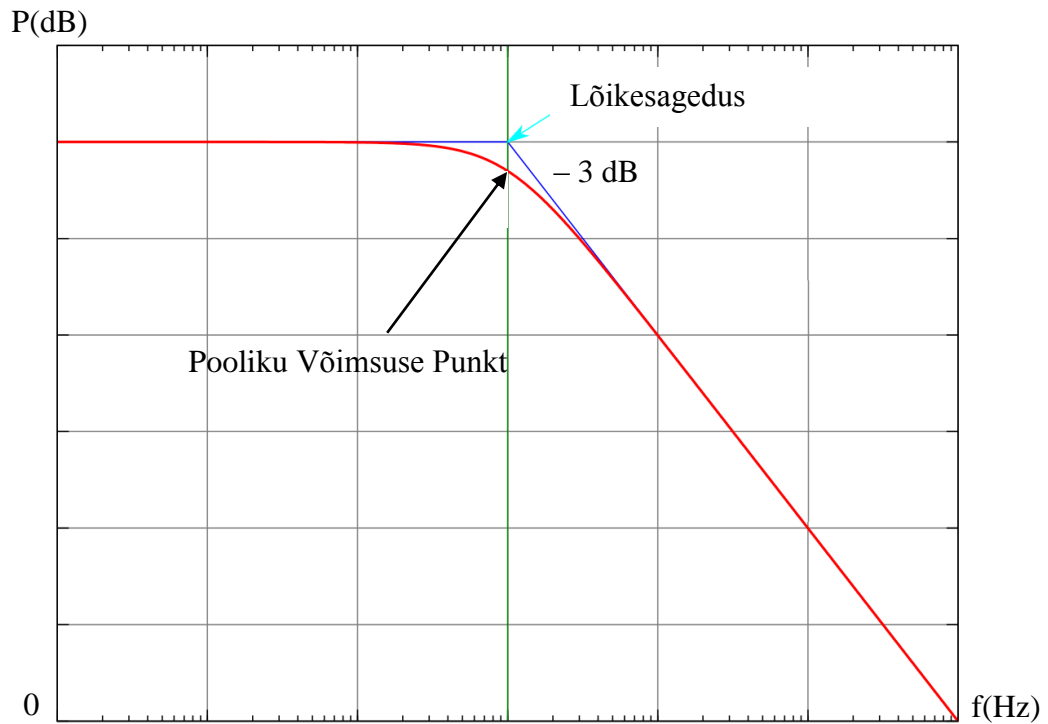


Joonis 3.22. PikoScope 5244B abil saadud pingete pilt – punane– sisendpinge U_{E1} ,
sinine– väljundpinge U_{out}

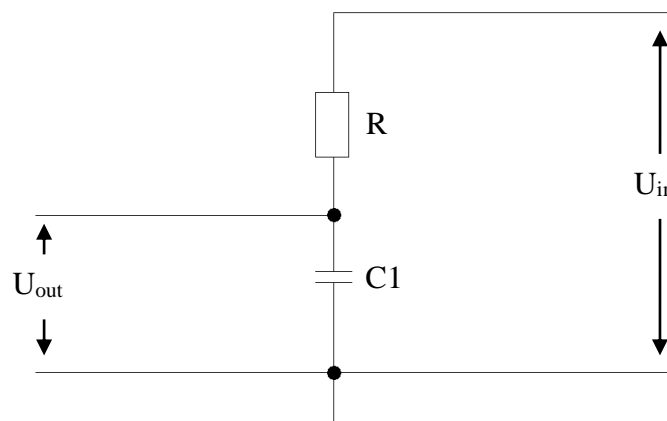
Pinged on mõõtmistulemuste järgi $U_{E1} = 12,51 \text{ V}$ ja $U_{out} = 2,026 \text{ V}$, mis on sarnased simuleeritud ja arvutatud tulemustega. Pinge jaguri arvutamine ja katsetamine on loetud edukaks, paremate mõõtetulemuste saamiseks võib reaalses lahenduses tulla vajadus kalibreerida.

3.4.2. Madalpääs sisendfiltri arvutamine ja katsetamine

Kõrgsageduslike häirete mõju vältimiseks mõõtmisahelas võib sisendile lisada madalpääs filtri. Seda tüüpi filter eemaldab kõik kõrgsageduslike võnkumised, mille sagedused on üle määratud piiri. Antud juhul on tegemist alalisvoolu signaaliga, mistõttu võib piirsageduseks valida suhteliselt madalat sagedust: kõiki häireid, mis ületavad seda, võib lugeda mittekasulikeks ja filtreerida. Madalpääs sagedusfiltri arvutamiseks valitakse kõigepealt madalaim sagedust, mida filter summutab – lõikesagedus. Sellel sagedusel summutab filter võnkumisi võimsusega 3 dB, mis vähendab häirete võimsust kaks korda, seda nimetakse pooliku võimsuse punktiks (Joonis 3.23).



Joonis 3.23. Madalpääs filtri bodeplot



Joonis 3.24. Madalpääs filtri skeem

Tavalist madalpääs filtrit ehitakse Joonisel 3.24 näidatud põhimõttejärgi. Joonisel 3.19 järgi on määratud R_1 väärtus, mis on $801\text{ k}\Omega$, jääb ainult arvutada mahtuvuse C_1 suurus. Seda võib teha valemiga:

$$f_c = \frac{1}{2\pi RC_1}, \quad (3.7)$$

kus f_c – Lõikesagedus,

R – Joonise 3.24 peal tähistatud takisti,

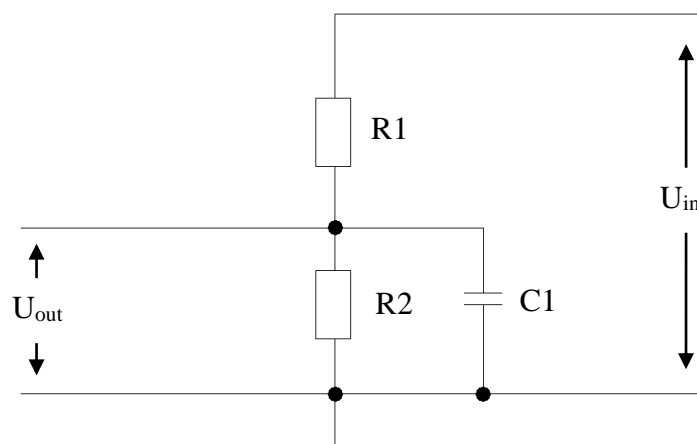
C1 – Joonise 3.24 peal tähistatud kondensaator.

Kui, f_c on lõikesagedus, R – R1 väärtus ja C – C1 väärtus. Lõikesageduseks valitakse 50 Hz, kuna sellel sagedusel võivad tekkida kõige madalamad häired: eelkõige tingituna vahelduvvoolu jaotusvõrgu süsteemi elektromagnet induksioonist ja inverterite töötamisest, mis on sageli on kasutusel autodel. Sellest lähtudes saame:

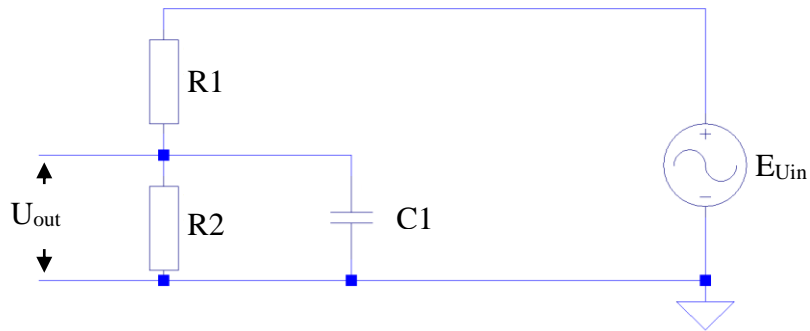
$$50 = \frac{1}{2 \cdot \pi \cdot 801000 \cdot C_1}, \quad (3.8)$$

kus C1 – Joonise 3.24 peal tähistatud kondensaator.

Ülaltoodud valemist võib saada C1 mahtuvuse väärtust, milleks on 3,9 nF. Potentsiaali jagamisel on vaja arvestada sellega, et skeemis peab olema ka R2 takisti, nagu on näidatud Joonise 3.19 peal. Sellega arvestades peab skeem välja nägema nagu on toodud Joonisel 3.25. Kõikide skeemi elementide väärtused on nüüd teada, seda võib simuleerida LT Spice keskkonnas (Joonis 3.26).

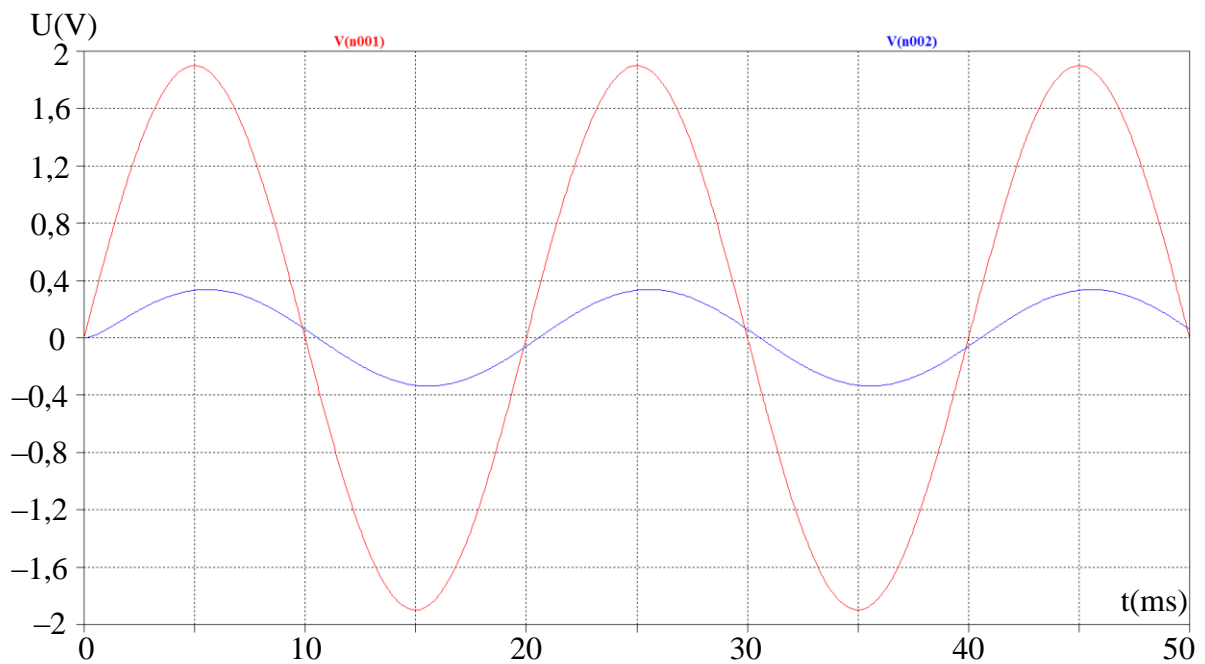


Joonis 3.25. Madalpääs filtri skeem koos potentsiaalijaguriga



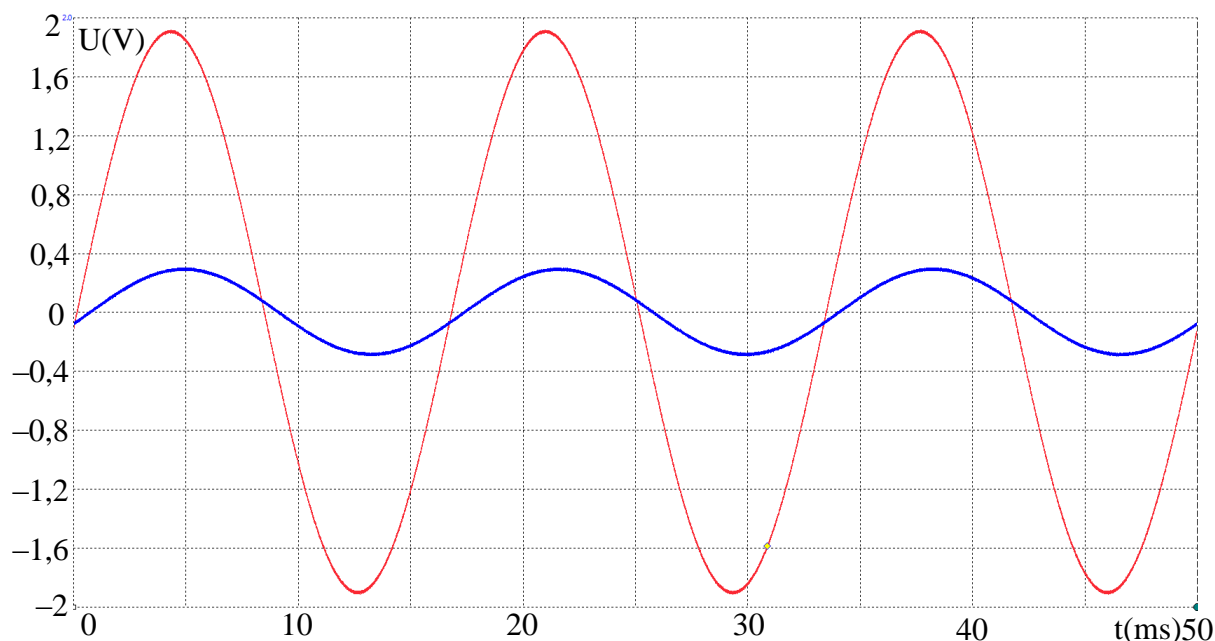
Joonis 3.26. Madalpääs filtri potentsiaalijaguriga skeemi simuleerimine LT Spice keskkonnas

Skeemi katsetamiseks valitakse sisendpingeks U_{in} ilma alalispinge komponendita sinusoidilane vahelduvpinget sagedusega 50 Hz ja amplituudiga 1,9 V. Selle simuleerimise tulemustena tulevad pingete graafikud, mis on toodud Joonise 3.27. peal.



Joonis 3.27. Madalpääs filtri skeemi koos potentsiaalijaguriga simuleerimise tulemused, sisendpinge – U_{in} – punane (V(n001)), väljundpinge – U_{out} – sinine (V(n002))

Joonisel 2.25 toodud skeemi katsetamiseks valiti C1 mahtuvuseks kondensaator 392Z maksimaalse pingega 50 V ja mahtuvusega 3.9 nF. Ostsiloskoobi pealt saadud graafikud on toodud Joonise 3.28 peal.



Joonis 3.28. Madalpääs filtri skeemi koos potentsiaalijaguriga katsetamine, sisendpinge – U_{in} – punane, väljundpinge – U_{out} – sinine

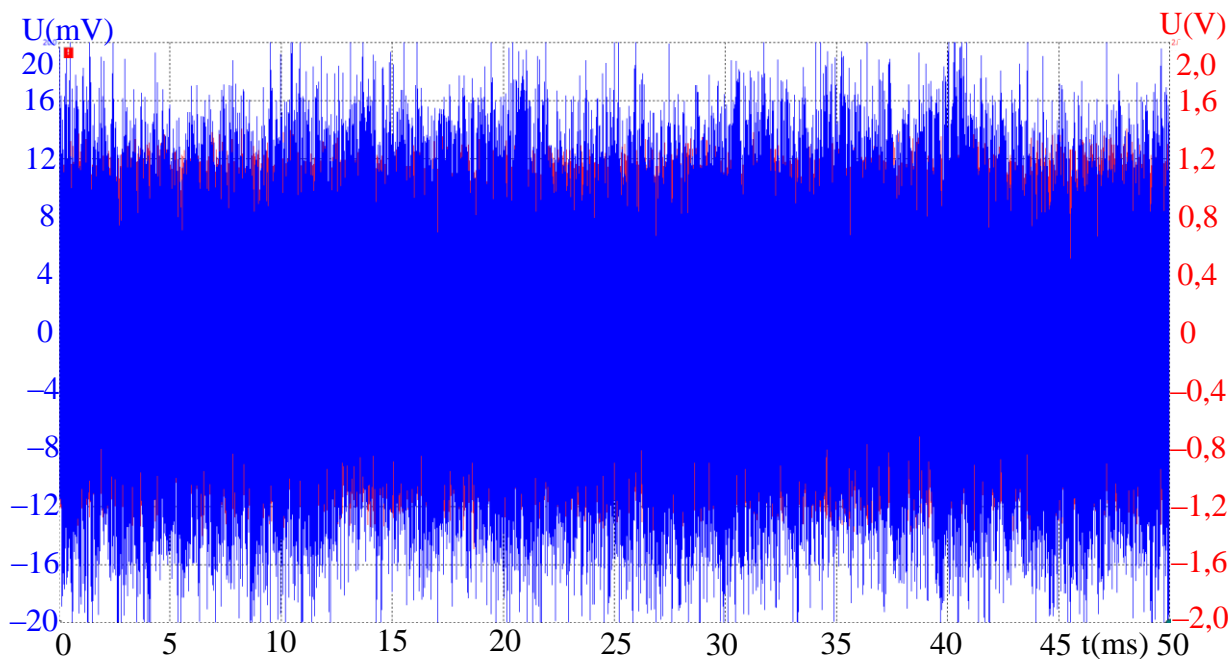
Simuleerimisel on sisendpinge amplituudiks saadud 1,9 V ja väljundpinge U_{out} amplituudiks saadud 336 mV, katsetamisel $U_{out k}$ oli 310,0 mV, mida võib lugeda sarnaseks. Selleks, et kontrollida tulemusi, tuleb arvutada võimendusteguri L_v valemiga

$$L_v = 20 \cdot \log_{10}\left(\frac{U_{out k}}{U_{in}}\right), \quad (3.9)$$

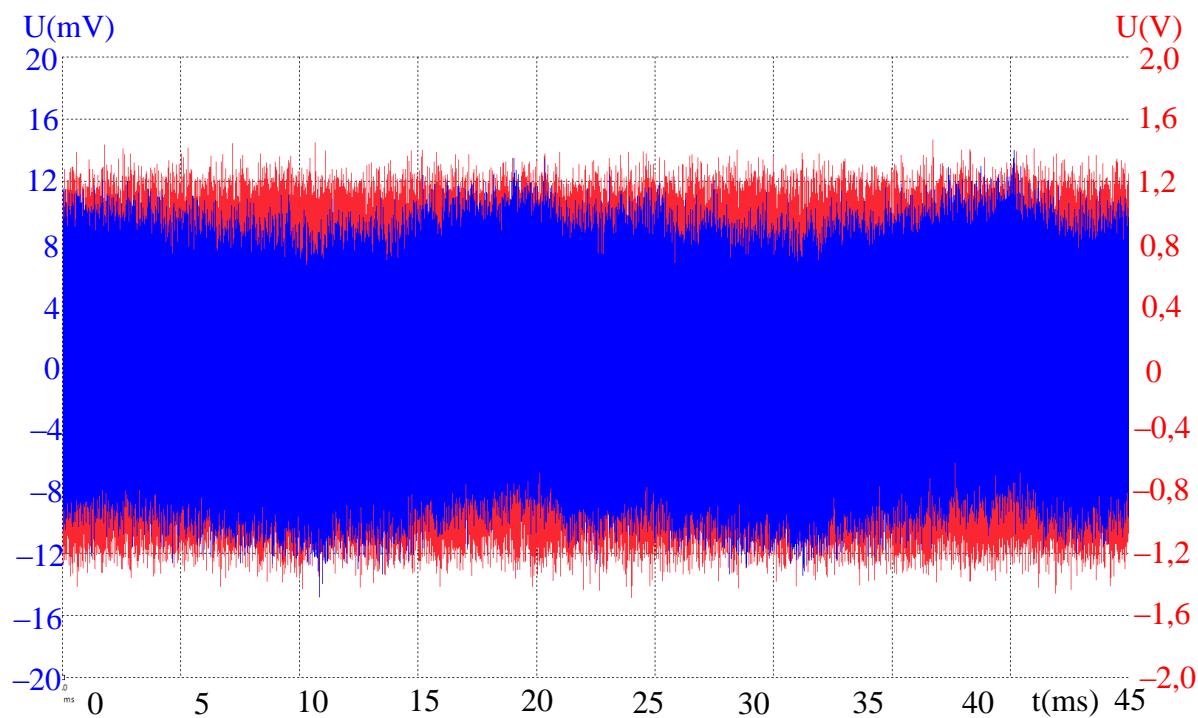
kus $U_{out k}$ – filtri väljundpinge amplituud,

U_{in} – filtri sisendpinge amplituud.

Kus $L_v = -15,7\text{dB}$, kuna antud valem kehtib ilma pingejaguri R2 elemendita skeemile, R2 suurendab sumbuust. Filtri hindamiseks seadistakse sisendpinge U_{in} signaaliks mitted sinusoidaalne laiaribaline (20 Hz kuni 20 000 Hz) *white noise* (Ingl. – *white noise* – valge müra) signaal, Joonise 3.29 peal.



Joonis 3.29. Skeemi töö potentsiaalijaguriga ilma kondensaatorita C1, sisendpinge – U_{in} – 1,6 V amplituudiga White Noise – punane, väljundpinge – U_{out} – sinine

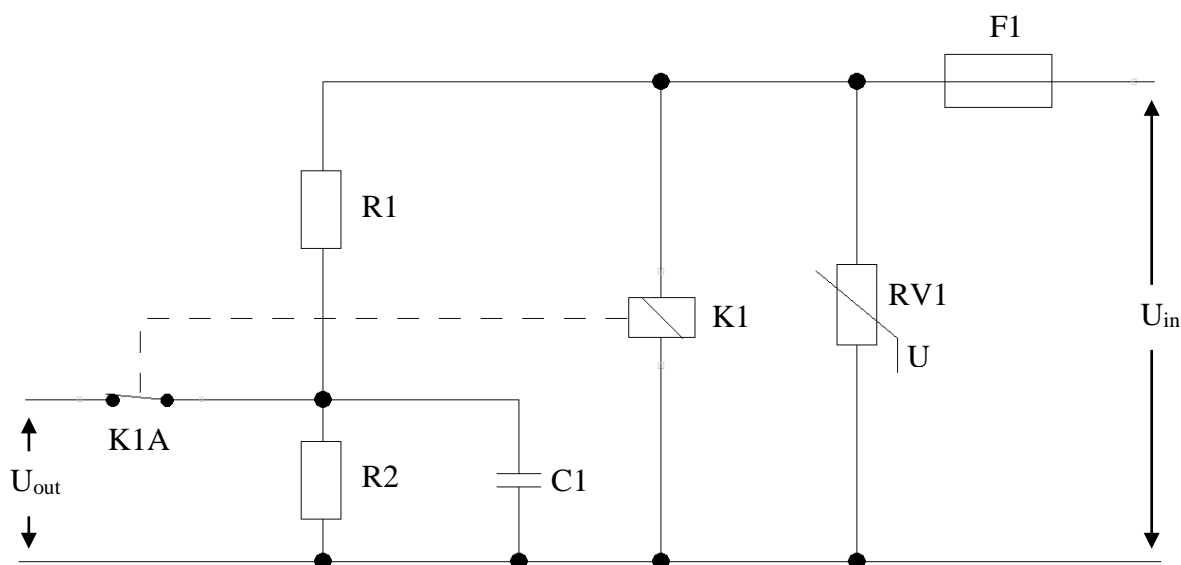


Joonis 3.30. Skeemi töö potentsiaalijaguriga koos kondensaatoriga C1, sisendpinge – U_{in} – 1,6 V amplituudiga White Noise – punane, väljundpinge – U_{out} – sinine

Joonise 3.29 ja 3.30 peal on näha filtri tööd – nähtav vahe väljundsignaalide vahel. Tuleb arvestada sellega, et 1,6 V amplituudiga müra on väga tugev, reaalse skeemi töös on oodata nõrgemaid vahelduvvoolu signaale.

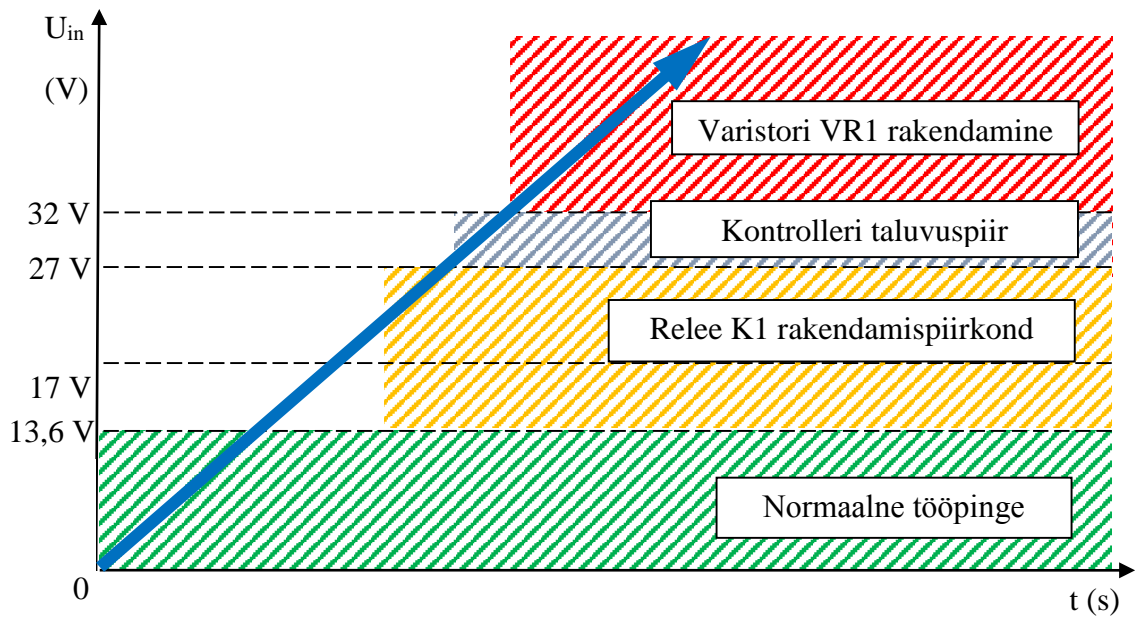
3.4.3. Mõõtmisahela kaitsmine ülepinge ja lühise eest

Kontrolleri ja mõõtmisahela kaitsmiseks ülepinge ja lühise eest võib kasutada mitut viisi korraga, rakendades esimesena taastuvaid kaitsmisviise ja seejärel mittetaastuvaid (Joonis 3.32). Üks taastuvaid ülepinge kaitsmisviise on kaitsereleede kasutamine, mis rakenduvad teatud sisendpinge ületamisel. Mittetaastuvad ülepinge kaitsmed on aga varistorid. Varistorid rakenduvad teatud sisendpinge ületamisel ja ajavad skeemi sisendi lühisesse, mille tagajärjel rakendub sulavkaitse – mittetaastuv lühis kaitsmisvahend.



Joonis 3.31. Mõõtmisahel koos kaitseahelatega – K1 – relee, F1 – sulavkaitse, RV1 – varistor

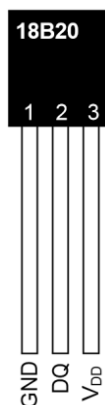
Mõõtmisahel (Joonis 3.31) on arvestatud viisil, kus maksimaalne sisendpinge U_{in} võib olla kuni 27 V. See tähendab, et relee rakendamispinge võib olla 20 V kuni 27 V ja taastumispinge (mis on tavaliselt madalam kui rakenduspinge) 17 V kuni 20 V. Selline süsteem kaitseb kontrolleri valeühendamise eest: näiteks juhul, kui monteerimisel ühendatakse süsteem kahe akuga ja antakse 24 V alalist pinget süsteemi sisendisse. Üle 27 V sisendpinge puhul muutub varistor $RV1$ voolu juhtivaks ning lühistab mõõtmisahela, mille järel põleb läbi sulavkaitse $F1$, katkestades mõõtmisahela. Sulavkaitse $F1$ võib valida väikese nominaaliga – 0,25 A kuni 0,5 A, sest suuremad voole ei ole mõõtmisahelas ette nähtud. Sulavkaitse rakendub iseseisvalt ka juhul, kui mõõtmisahela juhtmed lähevad lühisesse.



Joonis 4.32. Kaitsmisvahendite rakendamispriintiip

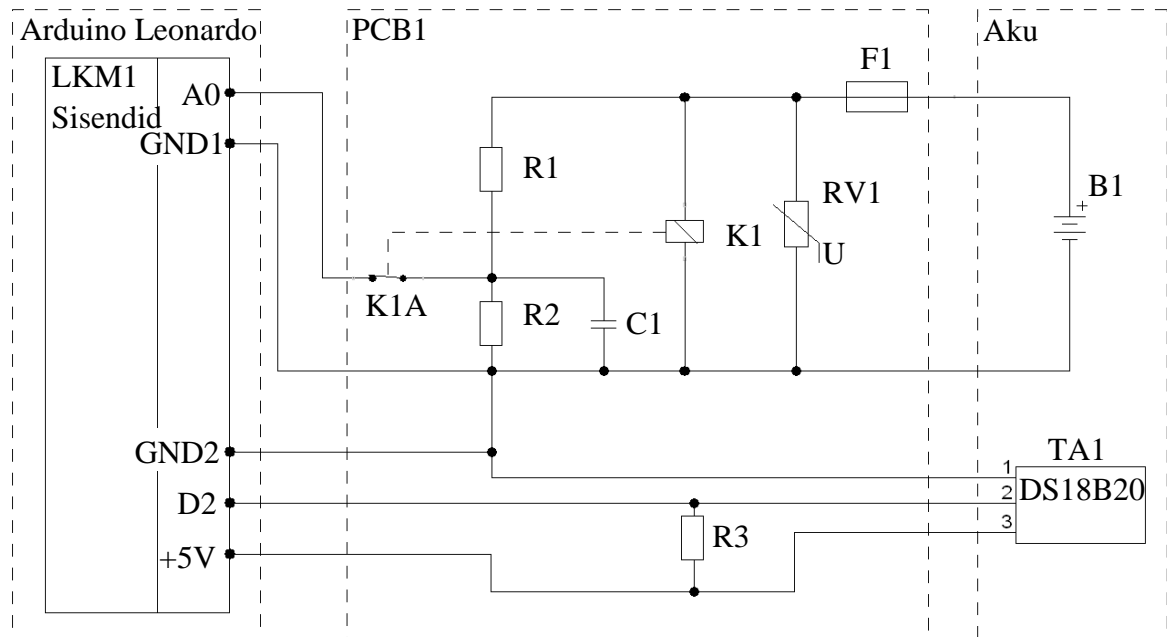
3.4.4. Temperatuuri mõõtmine

Mõõtmistulemuste täpsemaks hindamiseks on tarvis mõõta veel akumulaatori temperatuuri, ehk vaja on temperatuuri andurit. Kõige sobivam on valitud kontrolleri kasutada digitaalset temperatuuri andurit DS18B20 (Joonis 3.33). Selline andur ei vaja kalibreerimist ning on võimeline töötama Arduino Leonardo ETH© toitega, mis on 5 V alalist pinget, ning töötab väga lihtsa skeemi abil.



Joonis 3.33. DS18B20 temperatuuriandur[10]

DS18B20 annab piisava täpsusega mõõtetulemuse ($\pm 0,5 \text{ }^\circ\text{C}$) ja on võimeline töötama temperatuurivahemikus $-55 \text{ }^\circ\text{C}$ kuni $+125 \text{ }^\circ\text{C}$. DS18B20 tööks on vaja paigaldada vaid $4,7 \text{ k}\Omega$ takisti (Joonis 3.34 peal – R3) teise ja esimese terminaali vahel. Toide tuleb kolmandale terminaalile, mass – esimesele, teine terminaal ühendab Arduino Leonardo ETH© digitaalse sisendiga. Arvestades kaitseelementide ja temperatuurianduriga näeb mõõtmisahel välja nagu on toodud Joonisel 4.34.

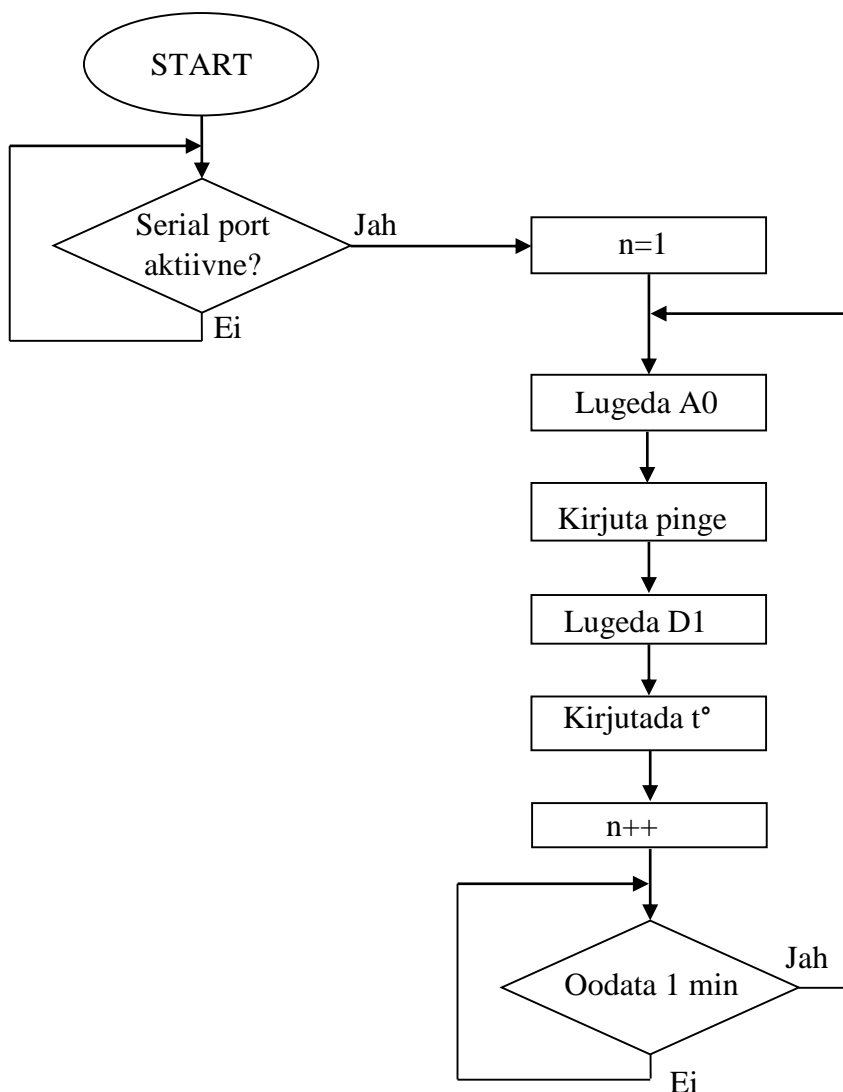


Joonis 3.34. Mõõtmisahel koos temperatuuranduriga – TA1

Joonise 3.34 peal: LKM1 sisend – Arguino Leonardo ETH© sisendid, PCB1 – mõõtmisahela trükkplaat, mis sisaldab kõiki vajalike komponente, B1 – mõõtmisobjekt: akumulaator, milleks on võetud BIGBAT© 7 Ah hooldusevaba plii–happe akumulaator.

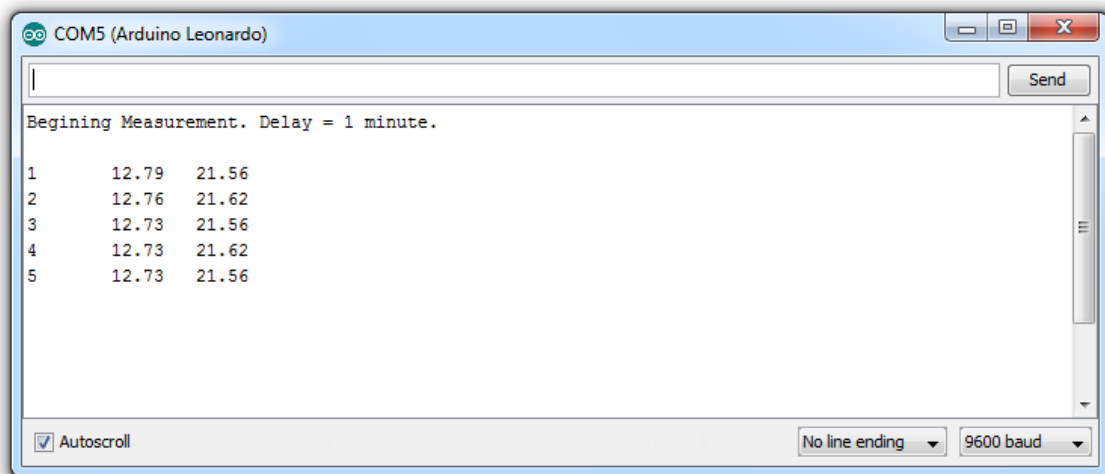
3.4.5 Mõõtmisprotseduuride programm

Mõõtmisahela töö katsetamiseks akumulaatori B1 mõõtmisel on järgmiseks sammuks kontrolleri jaoks programmi koostamine, võimaldades samaaegselt mõõta aku B1 temperatuuri ja pinget. Mõõtmistulemused kantakse mällu ja teostakse iga minuti tagant. Katsetamiseks on mõistlik kasutada jadapordi ühendust arvutiga, millega on suhteliselt lihtne edastada mõõtmistulemusi arvutisse. Programmi kood on toodud Lisa 2 ja programmi tööpõhimõtte plokkiagrammil Joonisel 3.35.



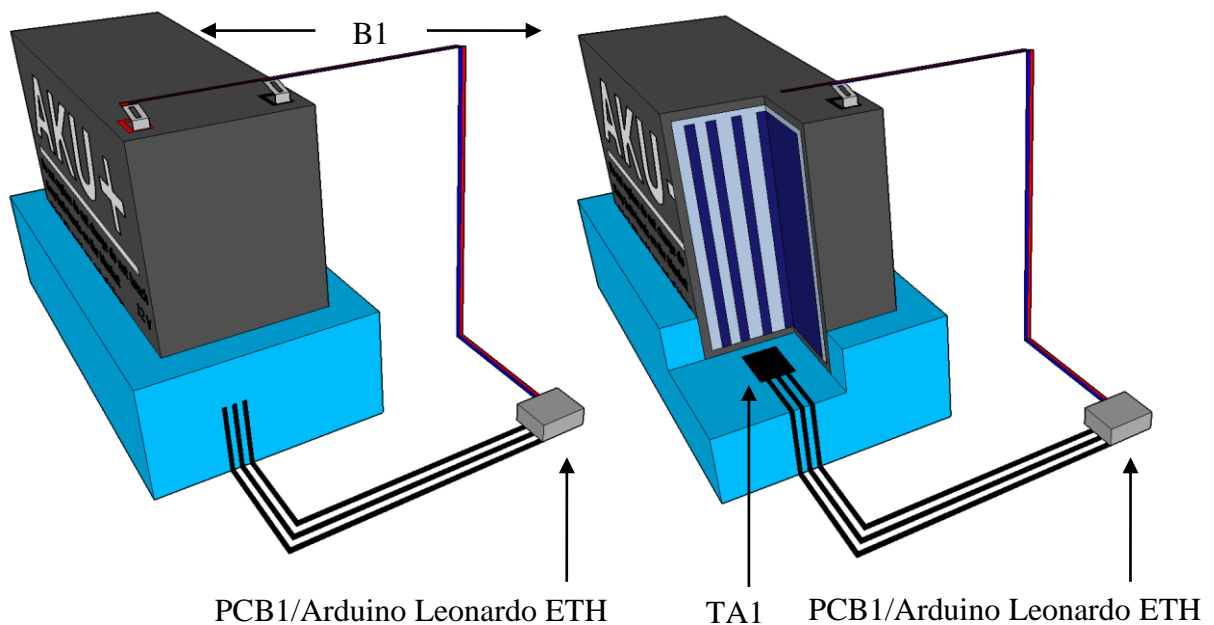
Joonis 3.35. Mõõtmisprogrammi plokkiagramm

Programmi koodis kasutakse OneWire[®] ja DallasTemperature[®] teke temperatuuranduri mõõtetulemuse lugemiseks. Korrektsete pinge näitude saamiseks korrutatakse pingejaguri pealt saadud pinge koefitsiendiga 5,288, milleks on sisend- ja väljundpinge suhe. Jadaporti arvutis lugedes tuleb Arduino Leonardo ETH[®] poolt informatsioon, mis on toodud Joonise 3.36 peal.



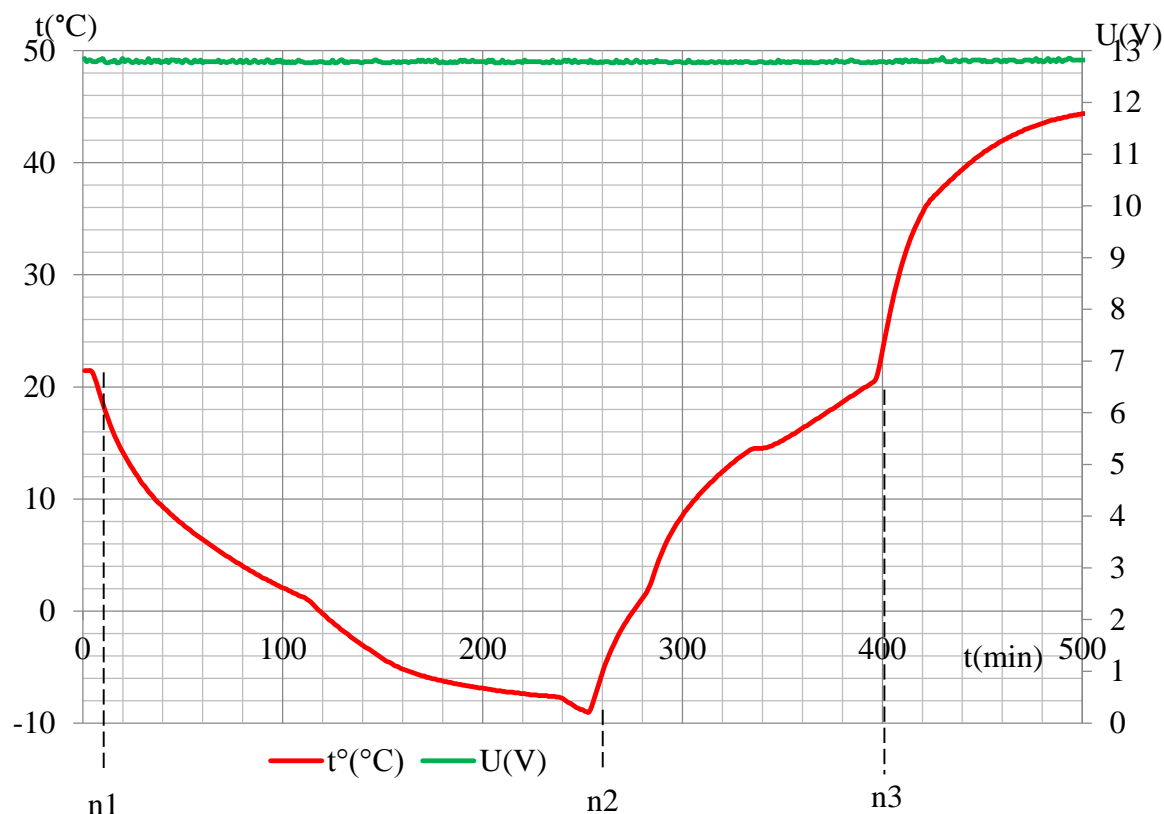
Joonis 3.36. Arduino Leonardo ETH COM5 pordil jälgimine

Joonis 3.36 peal on esimene veerg mõõtmise number, teine veerg on aku B1 pinge (voltides) ja kolmas veerg on aku temperatuur (°C).



Joonis 3.37. Akumulaatori testimise stand

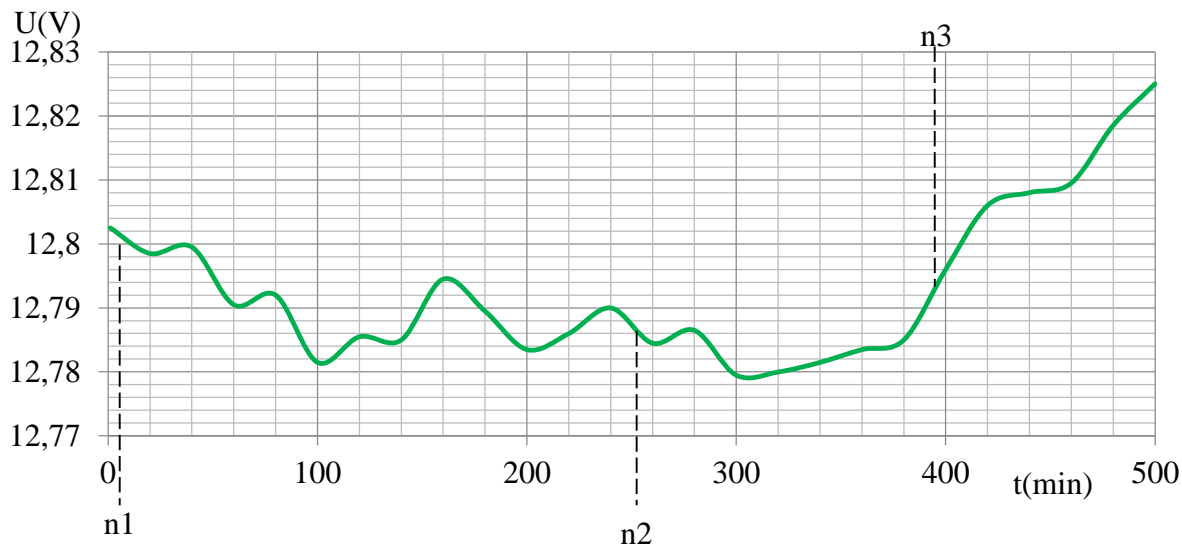
Aku temperatuuri ja pinget jälgimiseks oli ehitatud Joonise 3.37 peal näidatud stand – aku paigaldati termoisoleeritava aluse peale (Joonis 3.37 – sinine) ja aku alla paigaldati temperatuuriandur TA1. Aku klemmid ja TA1 ühendati omakorda Arduino Leonardo ETH©-ga piisavalt pika juhtmetega, et võimaldada ümberpaigaldamist ühe temperatuuriga keskkonnast teise.



Joonis 3.38. Akumulaatori testimise tulemused

Aku testimisel tehti üle 500 mõõtmise, mille jooksul logiti aku pinget ja temperatuuri väärtused (Joonis 3.38). Katsel n1 (kui $n = 5$) jahutati aku keskkonnas temperatuuriga umbes -10 °C , katsel n2 (kui $n = 251$) soojendati aku toa temperatuurini (mis oli ligikaudu $+21\text{ °C}$) ja katsel n3 (kui $n = 398$) soojendati aku keskkonnas temperatuuriga $+50\text{ °C}$.

Nagu on näha Joonise 3.38 pealt, pinget ei muutu oluliselt temperatuuri muutumisega. Kui võtta keskmised pingete väärtused, et vältida mittevajalikke võnkumisi ja suurendada pilti, siis on võimalik saada selgem pilt pinget muutmisest, mis on toodud Joonise 3.39 peal.



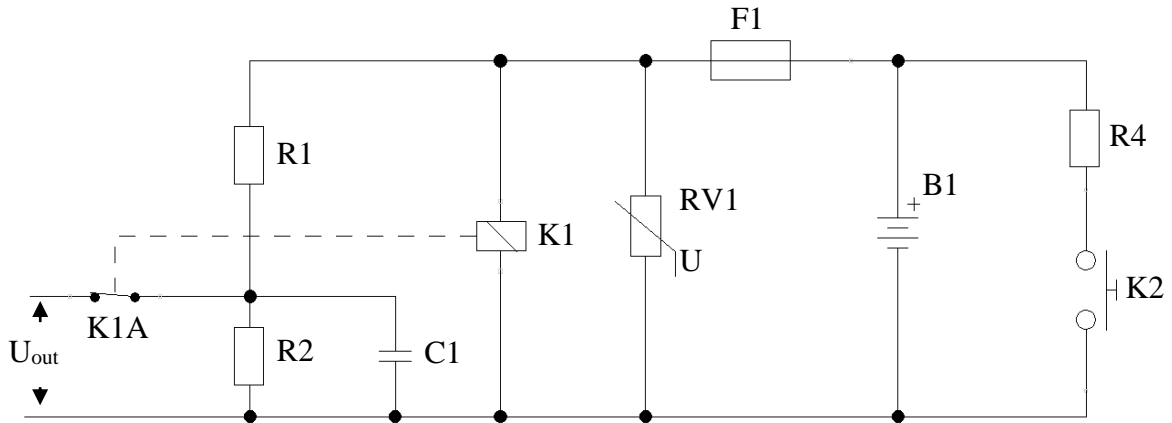
Joonis 3.39. Akumulaatori pinge muutumine

On näha, et temperatuuri mõõtmine oli tehtud viitega pinge suhtes, sest et aku polnud võrdselt soojendatud või jahutatud. Joonise 3.39 peal toodud informatsiooni abil on aga võimalik rakendada temperatuuri koefitsienti, mis parandab pinge muutmist temperatuuri suhtes paremate SOC andmete saamiseks. Pikaajalisel ladustamisel on sellisest koefitsiendist suurem kasu, kuna akumulaatorite temperatuur muutub siis sujuvamalt ja temperatuurijaotus on tänu sellele akumulaatorites ühtlasem.

3.5. Koormusahel pindlaengu eemaldamiseks

3.5.1. Koormusahela arvutamine

Pindlaengu eemaldamiseks ja aku defektide diagnostikaks on tarvis akut koormata, kasutades aktiivtakistust. Pindlaenguefekti (mis tekib lühiaegse laadimise puhul ebavõrdse mahulise plaatide laadimise pärast) eemaldamiseks akumulaatori plaatides soovitakse koormata akut ligikaudu 50 W koormusega (näiteks sõiduki täistuledega), et võrdsustada plaatide laengut. Selle jaoks skeemile tehti vajalikud täiendused.



Joonis 3.40. Akumulaatori koormamise skeem

Joonise 3.40 peal on katsetamiseks lisatud pingemõõtmisahelasse $3,9 \Omega$ takisti R4, mis akut $3,23 \text{ A}$ vooluga siis, kui lüliti K2 on suletud. Sel juhul on aku koormatud 41 W koormusega.

U_{out} on loetud Arduino Leonardo ETH© kontrolleri abil ning selleks, et pingenäitudel poleks liigseid võnkumisi, arvutatakse tulemust iga kümme sekundi kohta mitme mõõtmise aritmeetilise keskmisena.

$$U_{out\ avg} = \frac{\sum_{i=1}^{10} U_{out}}{10}, \quad (3.10)$$

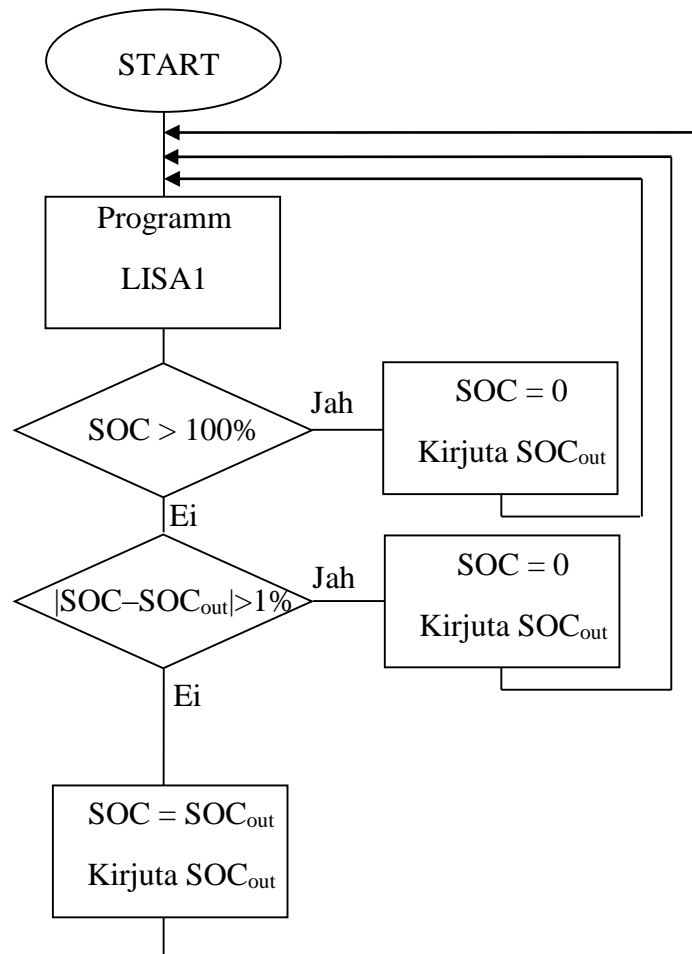
kus $U_{out\ avg}$ – aritmeetiliselt keskmine pinge 10 sekundi jooksul,
 U_{out} – Joonise 3.40 peal tähistatud pinge.

SOC arvutamiseks, kasutades $U_{out\ avg}$, tuleb teisendada seda protsentidesse, mida tehakse arvestades sellega, et aku pinge on täis laetuna umbes $13,1 \text{ V}$ (Joonis 2.9). Sellest saame valemi:

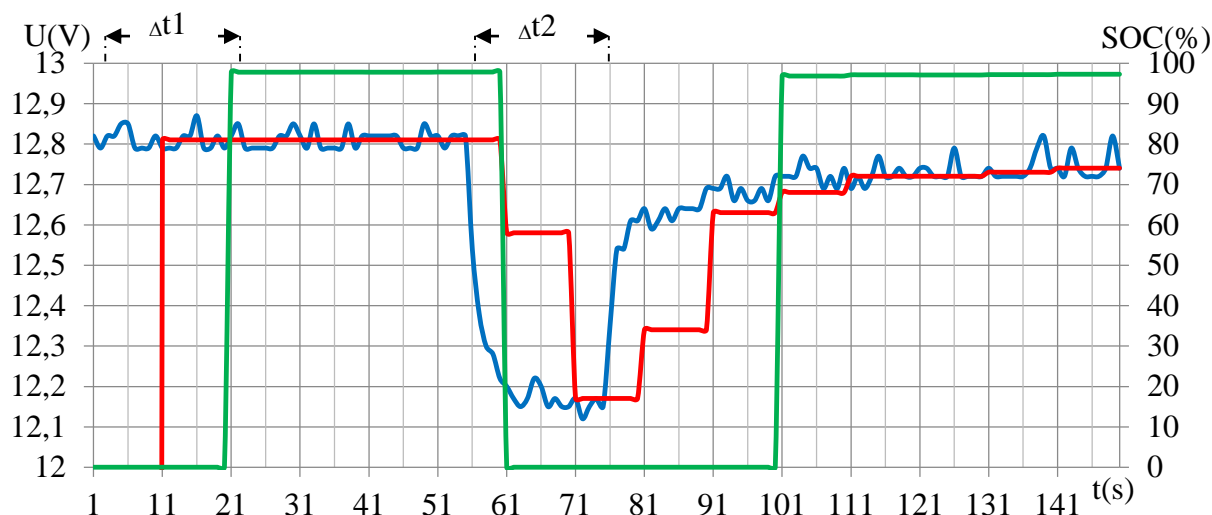
$$SOC = \frac{U_{out\ avg} - 11,8}{13,1 - 11,8} \cdot 60 + 40 (\%), \quad (3.11)$$

kus $U_{out\ avg}$ – aritmeetiliselt keskmine pinge 10 sekundi jooksul,
 SOC – aku laadimisetase.

Valede SOC väärtuste vältimiseks (st. kui aku pinge on kõrgem kui 13,1 V või ei ole veel stabiliseeritud peale laadimist või koormamist), lisatakse programmile kontrollalgoritm (Joonis 3.41), kus SOC_{out} on viimase 10 sekundi jooksul mõõdetud laadimistase, mida kasutakse hetkelise laadimistase võrdlemiseks. See võrdlus võimaldab veenduda selles, et laadimistase näit on stabiliseeritud. Antud algoritmi mõõtmisgraafikud on toodud Joonise 3.42 peal.

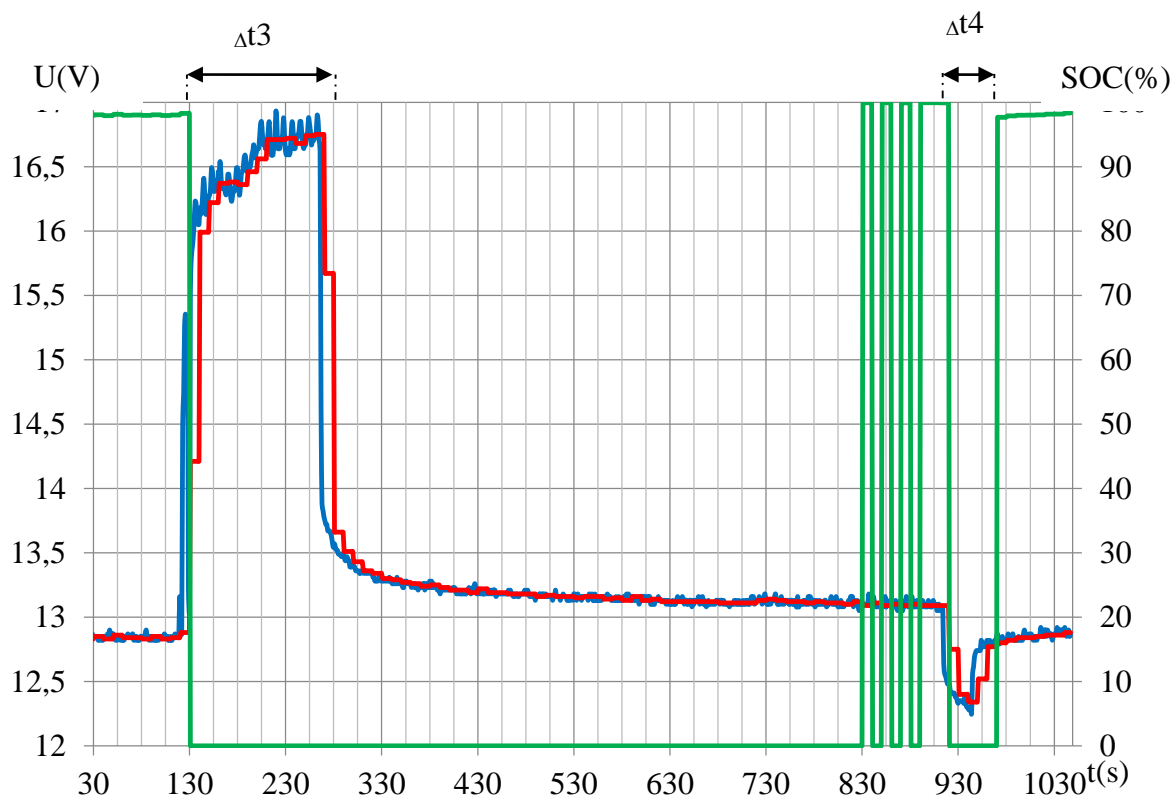


Joonis 3.41. SOC mõõtmise algoritm



Joonis 3.42. SOC mõõtmise katsetamine – SOC – roheline, U_{out} – sinine ja U_{outavg} – punane

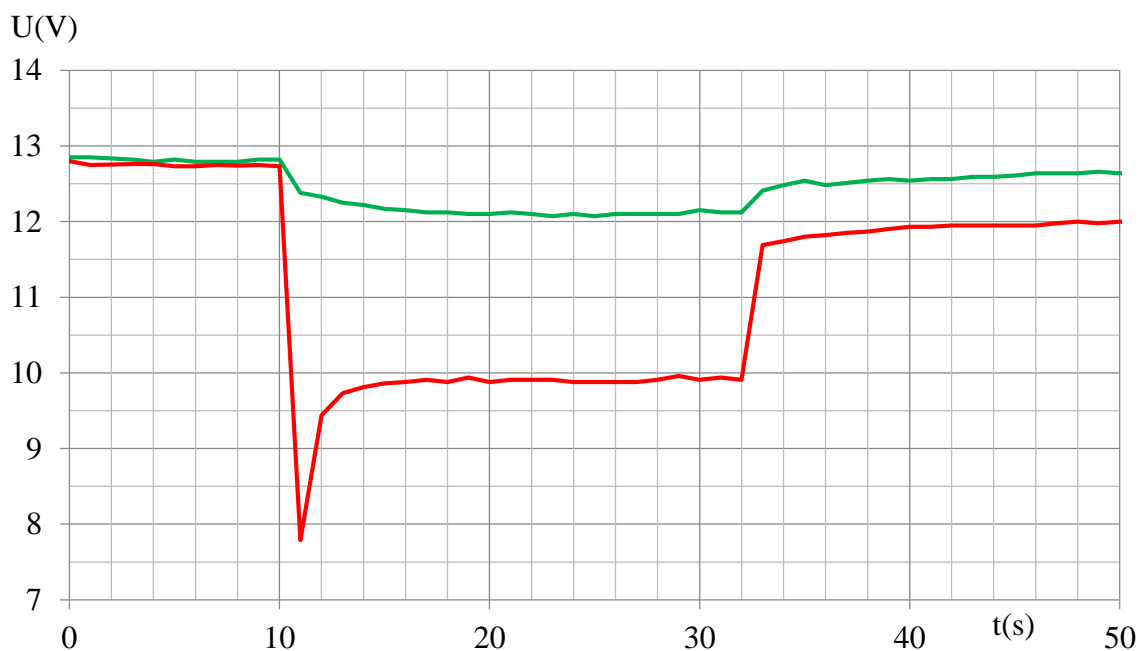
Andmed võeti ühe sekundi vahedega parema resolutsiooni jaoks. Aeg Δt_1 kulub süsteemi töö stabiliseerimiseks programmi alguses. Koormus rakendati lüliti K2 sulgemisega kahekümneks sekundiks – seda tähistab aeg Δt_2 . Aku SOC mõõtmist katsetati koos lühiaegse laadimise tsükliga ja pindlaengu eemaldamisega.



Joonis 3.43. SOC mõõtmise katsetamine koos laadimisprotsessiga – SOC – roheline, U_{out} – sinine ja U_{outavg} – punane

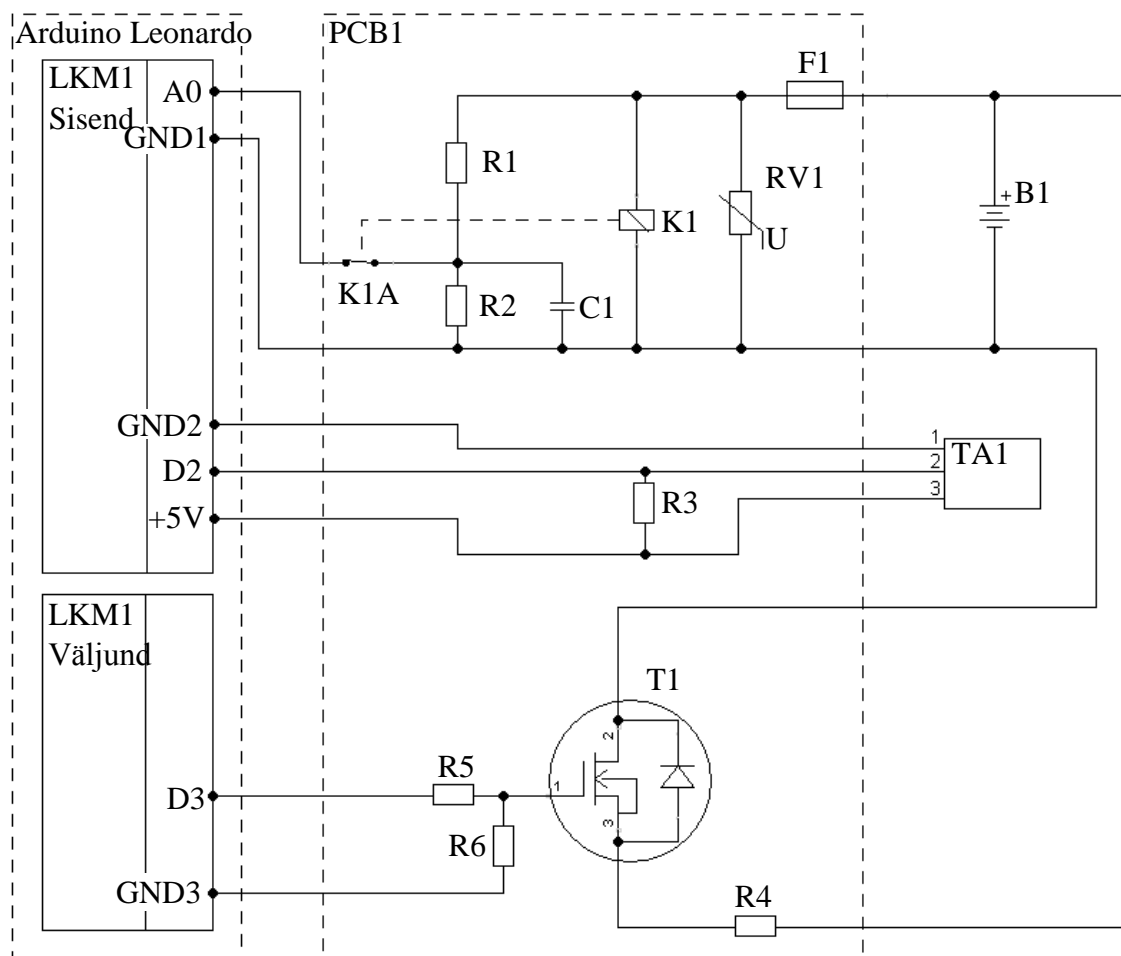
Joonise 3.43 peal toodud katses on akumulaator laetud aja Δt_3 jooksul (umbes 100 sekundit) ja koormatud Δt_3 jooksul (umbes 30 sekundit). On näha, kuidas peale laadimist ja enne koormamist (aeg: 830 s kuni 900 s) arvutatud SOC väärtus hakkab õige tulemuse poole muutuma, kuid siiski jääb natuke kõrgemaks. Koormamisel taastub SOC õige väärtus, nagu oli alguses (98.4 %).

Antud koormamise protsess võib anda ülevaadet plii-happe aku tervise seisundist (SOH), näidates aku defekte koormamise protsessi käigus. Joonisel 3.44 on võrdlemiseks toodud kaks kõverat: roheline, mis kirjeldab 20 sekundilist terve aku koormamist ja punane, mis kirjeldab lühis sektsiooniga akukoormamist. Mõlemad akud on sama mahtuvusega. Ajal $t = 10$ koormati mõlemat akut ja ajal $t = 32$ võeti koormus maha. On näha, kuidas pinge langeb rohkem lühissektsiooniga aku puhul, pinge taastamiseks eelmisele tasandile kulub sel ka rohkem aega (üle 100 sekundi).



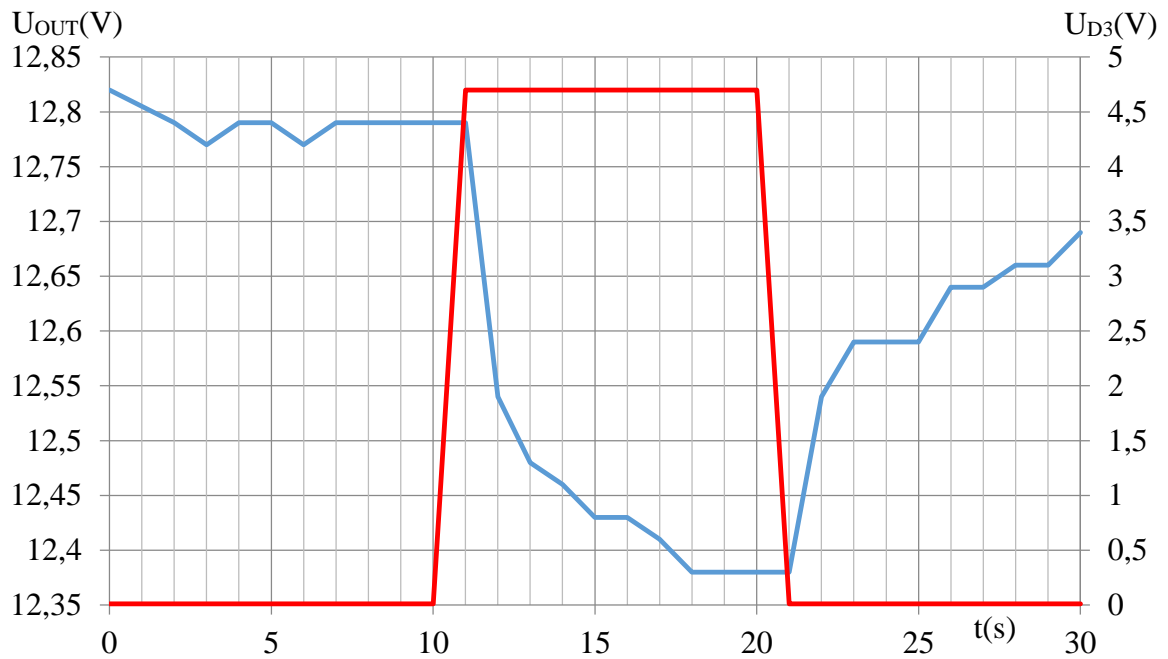
Joonis 3.44. Terve ja lühissektsiooniga akumulaatori koormamine – roheline U_{out} terve akumulaatori puhul, punane – U_{out} lühissektsiooniga akumulaatori puhul

3.5.2. Koormusahela projekteerimine ja seadmete valik



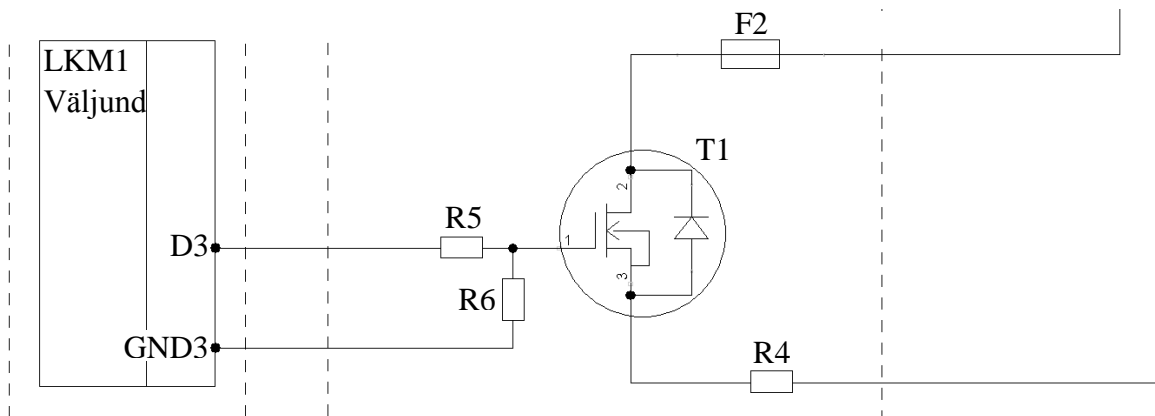
Joonis 3.45. Skeem koos koormamisahelaga

Selleks et kontrolleri LKM1 saaks koormata R4 takistiga akut B1, oli T1 transistoriks valitud N-MOSFET (*N-channel Metal-Oxide-Semiconductor Field-Effect Transistor*) transistor, mida saab avada + 5 V pingega rakendamiseks 1 ja 2 viikude vahel (paisu-neelu pingega) (Joonis 3.45). Väljundvoolu piiramiseks (maksimaalselt 40 mA) oli digitaalse väljundi ahelale lisatud takisti R5, mis on 150 Ω nominaaliga. Takisti R6, mis on 1 M Ω väärtusega, on staatilise laengu eemaldamiseks neelu ja paisu vahel siis, kui on vaja transistori sulgeda. T1 transistoriks oli valitud IRF530. IRF530 lahtise seisuga takistus neelu ja lättevahel on 0,6 Ω , mis vähendab koormamisvoolu 0,5 A võrra, mis on võimalik kompenseerida koormamisega pikendamise või koormustakisti R4 vähendamise teel.



Joonis 3.46. U_{in} , koormamisahela skeemiga akumulaatori koormamine – U_{in} – sinine, juhtimispinge U_{D3} – punane

Joonise 3.46 peal toodud katsetamisel lülitati sisse digitaalväljund D3 hetkel $t = 10$ s ja väljahetkel $t = 20$ s. Koormamise ajal $t = 10$ s kuni $t = 20$ s oli transistorile T1 antud juhtimispinge $+ 4,7$ V, koormustakisti R4 oli akuga ühendatud. Koormusahela ülekoormusest kaitsmiseks tuleb lisada ahelasse sulavkaitse F2 Joonisel 3.47 toodud viisil.



Joonis 3.47. Koormamisahel koos lisatud sulavkaitsmega – F2, osa skeemist

Jooniselt 4.45

3.6. Mõõtmistulemuste võrdlemine

Projekteeritud süsteemi mõõtmistulemuste hindamiseks valiti kolm erinevad sõiduki plii-happe akut erinevate mahtuvuste ja laadimistasemetega. SOC mõõtmise kontrollimiseks kasutakse suure täpsusega Telwin DTP 800© akutestrit. Iga aku koormatakse esialgu pindlaengu eemaldamiseks ja seejärel mõõdetakse SOC Arduino Leonardo ETH© baasil ehitatud süsteemiga (Tabel 3.3 – Tulemus). Selle järgi teostakse mõõtmist kasutades Telwin DTP 800© testrit (Tabel 3.3 – Kontroll). Kõik mõõtmised tehakse sama keskkonna temperatuuriga ja väikeste ajavahemikega, tulemused on toodud Tabelis 3.3.

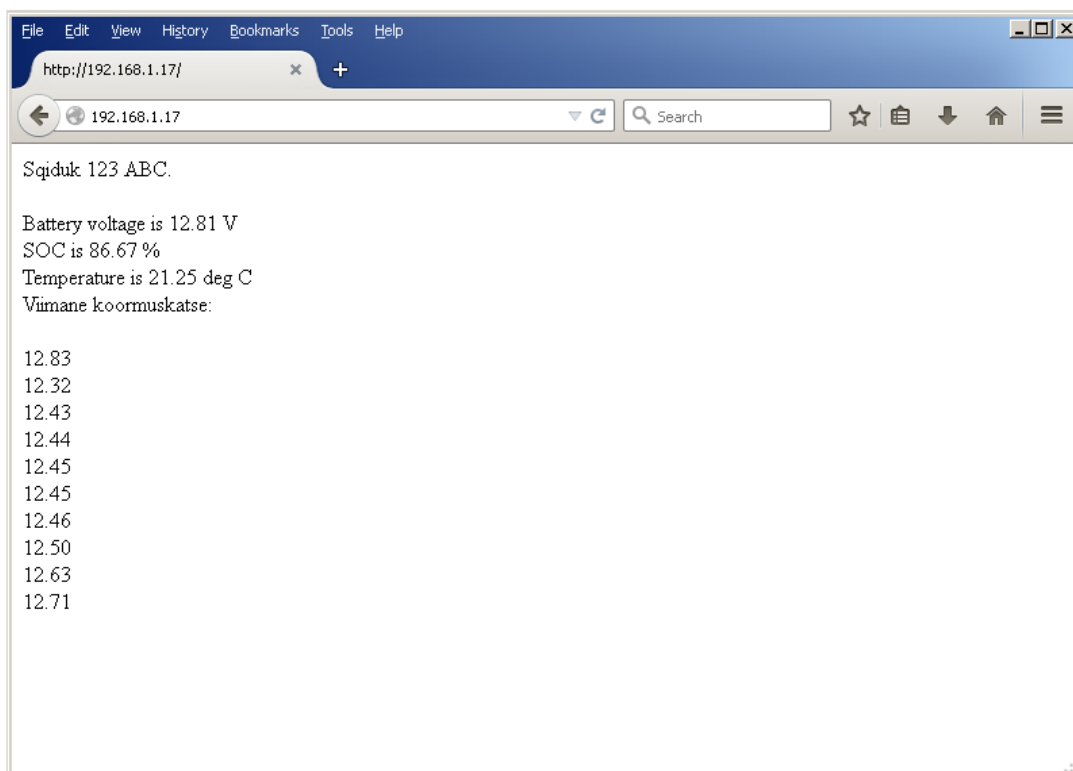
Tabel 3.3. Mõõtmistulemuste võrdlemine

Aku number	Tootja/Mudel	Mahtuvus	Rating	Pinge	SOC	
					Tulemus	Kontroll
1	Exide/72AH	72 Ah	DIN – 475	12,27 V	62 %	57 %
2	Bosch/85AH	85 Ah	EN – 800	12,56 V	75 %	80 %
3	Varta/F18	85 Ah	EN – 800	12,13 V	55 %	45 %

Tabelis 3.3 toodud mõõtmiste järgi on näha, et projekteeritud süsteemis on SOC mõõtmiste tulemused sarnased (umbes $\pm 5\%$ veaga) testriga Telwin DTP 800 mõõdetud tulemustega, kuni laadimistase SOC on piisavalt kõrge (50 % ja kõrgem). Madalama laadimistase puhul on oodata mittelineaarse pinge ja SOC vahelist suhet[8]. Akumulaatorit, mille laadimistase SOC on alla 40 %, soovitatakse enne mõõtmist laadida parema mõõtmistulemuse saamiseks. Kui teha SOC mõõtmist vahemikega (näiteks sammuga 5 %), on hindamine lihtsam. Kui SOC langeb alla 50 % võib alustada laadimist, kuna on oht, et tegelik akumulaatori SOC langeb alla 40 % mis pole hea akumulaatori ladustamise tingimusi silmas pidades.

3.7. Andmete edastus võrgus

Võimaldamaks mõõtmisüsteemil edastada mõõtmistulemuste andmeid LAN võrgu kaudu andmebaasile, kus neid kogutakse, või kliendi arvutisse, kus neid töödeldakse, kasutakse Arduino Leonardo ETH© Etherneti liidest. Tagamaks võimalust suhelda kontrolleri läbi võrgu, tehakse temast server *Ethernet2* teeke abil. *David A. Mellis* ja *Tom Igoe* veebiserveri lähtekoodi baasil koostati näidisprogramm. Kommunikatsioon käib võrgus läbi TCP 80 pordi HTML (*Hyper Text Markup Language*) liidese kaudu. Liidese HTML rakendus sisaldab sõiduki registreerimisnumbrit ning aku pinget, temperatuuri ja SOC-d. Lisaks sellele kujutab antud liides viimase koormamiskatse tulemusi (akumulaatori pinge iga sekundi tagant koormamise jooksul). Tulemuseks saab standardses veebilehitsejas, näiteks Fire Fox©, vaadata kõiki vajalikke mõõtmistulemusi (Joonis 3.48).



Joonis 3.48. HTML-koodis leht, mis kujutab Arduino Leonardo ETH© mõõtmiste tulemusi

4. KOKKUVÕTE

Antud töö käigus tehtud pakutavate lahenduste uuring näitas, et vabas müügis ei ole suure autopargi aku monitooringu süsteemi jaoks sobivat paindlikku toodet. Kõige sobivam variant oli LIN – või muu võrguprotokolli peal ehitatud kontrolleriite massiiv, mis siiski ei sobinud antud juhul, sest väiksemate võrkude puhul või üksikutel autodel võrgu ülesehitamine oleks liialt kallis.

Teoreetiliste aluste uurimine plii–happe akumulaatorite laadimistaseme määramise kohta tõestas töö alguses pakutatud valiku: kõige lihtsam ja kindlam viis plii–happe aku laadimistaseme määramiseks on antud juhul tühijooksupinge mõõtmine (Joonis 2.9). Sellega saab anda hinnangut isegi akule, mille seisund (näiteks vanus, ladustamise tsüklid, ladustamise tingimused) on teadmata. Tühijooksu pinge mõõtmine ei ole kõige täpsem või universaalsem viis, kuid käesoleval juhul pole täpne laadimistaseme määramine niivõrd oluline – piisab sellest, et süsteem annab teada, kui aku laadimistaseme läheneb 50 (alla selle on mõõtmisviga tühijooksu pinge mõõtmisel suur). Siis tuleb akut laadida, kuna edaspidine ladustamine madala laadimistasemega võib plii–happe akut kahjustada.

Võrgu ülesehituse printsiibi valiku poolt oli valitud kõige ratsionaalsem ja lihtsam variant: programmeeritavate kontrolleriitega (näiteks Arduino Leonardo ETH©) pole keeruline luua kohalikku andmebaasi iga sõiduki juures (Joonis 3.6), kuna hetkel toetavad paljud kontrolleriid SD-kaardi (*Secure Digital Card*) kasutamist informatsiooni hoidmiseks. Kohalik andmebaas võimaldab koguda mõõtmistulemusi isegi juhul, kui võrguühendus üldise andmebaasi või kliendiga on katkestatud. Edaspidiseks uurimiseks on üks tähtsaimatest punktidest andmete töökindel ja liigse sidekoormuseta edastamine ning üldandmebaasi kogumine.

Kontrollerite valikul on Arduino Leonardo ETH© sobivaim platvorm – mudel võib olla isegi erinev ja vähemate võimalustega. Süsteemi ehitus ja programmeerimine on Arduino kontrolleriitel suhteliselt lihtne ja sarnane kõikide mudelitega (Joonis 3.17). Lihtsama kontrolleri, näiteks Arduino Nano©, valimisel langeks süsteemi hind ühe sõiduki kohta mitu korda. Küll aga võib kasutada ka tellimusel valmistatud PCB–d, mis on näiteks ehitatud PIC mikrokontrolleri peal ja sisaldab ainult kõige vajalikumaid funktsioone nagu Digital Measure V10©. Uurimiseks valitud Arduino Leonardo ETH© oli mugavam kontrolleri

võimaluste kasutamise ja programmeerimise poolest, kuid antud skeem ja süsteem võib olla ehitatud mistahes teise kontrolleri baasil, kui täidetakse kõik süsteemile esitatud nõuded.

Mõõtmisahela ehitus ja kaitse sai tehtud suhteliselt universaalseks (Joonis 3.31), sobides ka teiste controllerite jaoks väheolulise programmi muutustega. Koormamisahela jaoks valitud takisti ja transistori asjus võib tulevikus teha sügavamaid katsetamisi ja uurimisi ning vajadusel kaalutleda kaasaegsemaid meetodeid aku seksioonide defektide tuvastamiseks nagu andurid igas seksioonis lühilainelise sidega[12]. Üldise ahela ehitamisel peab arvestama ka sellega, et soojusenergia, mis eraldub transistorilt T1 ja takistilt R4 (Joonis 3.45), on suhteliselt suur. Sellega tuleb arvestada koormustsüklite arvu – ning ajalise piiranguga. T1 ja R4 (Joonis 4.1 peal) elementidel peab olema võimalus ka jahutusvõimaluse projekteerimiseks. Süsteemi töö käigus tuleb uurida ka aku suuruse mõju koormamisprotsessi pikkusele – erinevatel akudel on erinev plii plaatide suurus, mis mõjutab pindlaengu suurust ning selle eemaldamiseks vajalikku koormamise pikkust.



Joonis 4.1. Koormamiseks kasutatud takisti R4 – ZDST21X60 H

Programmid, mida tehti kontrolleri töö katsetamiseks, vajavad veel ka optimeerimist. Kuigi kõik vajalikud funktsioonid said katsetatud ja demonstreeritud, ei saa veel Lisas 1 ja Lisas 2 toodud programme lugeda lõplikeks. Antud programmidel vajab täiendamist ka HTML-osa, et võimaldada ka kontrolleri juhtimist selle kaudu. Selleks on vaja konsulteerida IT spetsialistidega.

Mõõtmiste tulemuste täpsemaks hindamiseks tuleb süsteemi laadimistaseme mõõtmist kontrollida etalonakuga, kus tegelik mahtuvus on teada ning laadida seda akut tühjaks teatud koormusvooluga. See annab täpsemat informatsiooni aku laadimistaseme kohta ampertundide lugemisega (valem 2.4), mida võib võrrelda süsteemi poolt antud andmetega. Katsetamist tuleb läbi viia erinevate mahtuvustega akudega.

Üheks oluliseks edaspidise uuringu võimalustest on aku temperatuuri ja pinge suhte analüüs. Erinevad akud reageerivad temperatuuri muutmisele erinevalt, sellest tulenevalt võib määrata kasutatavate akutüüpide seast, kuidas võib rakendada parandusi laadimistaseme lugemisel temperatuuri suhtes[11]. Uuring eeldab pikaajalisi katsetamisi, kus on teada aku temperatuuri täpne väärtus ja see, et akus on ühtlane temperatuuri jaotus.

Mugava kasutajaliidese loomine on samuti oluline edaspidise arenduse suund: luua lihtne ja arusaadav liides, võimaldades kontrollida süsteemi tööd ilma suurema väljaõppeta. Antud süsteem võib töötada ka sõiduki varastamisvastase vahendina, sest sõiduki mõõtmisüsteemi lahti ühendamisel võrgust võib edastada teate võrku kontrollivale isikule. Sellise süsteemi olemasolu puhul on autopargis palju lihtsam määrata sõidukite tegelikku asukohta ka eemal olles: kontrolleri IP-või MAC-aadressid võib kinnitada kindlate sõidukite numbritele andmebaasis või kontrolleri programmis.

Antud süsteemile tuleb kasuks ka laadimisprotsesside juhtimise võimaldamine. Autopargi hooldamise mõttes oleks kaugjuhitav laadimine väga suureks eelduseks süsteemi ehitamisel. Selleks tuleb arvestada mõned sisendid/väljundid valitud kontrolleri reserviks, siis võib tulevikus ehitada sama kontrolleri baasil automaatse/kaugjuhitava laadimissüsteemi juurde.

Kui täiendavad uuringud ja arendustööd on tehtud ning süsteem vastab nõuetele, ehitatakse komplektne prototüüp süsteem väikse autopargi osa jaoks ja katsetatakse seda veendumaks süsteemi ja võrgu katkestusteta töös. Eduka testperioodi järel võib laiendada süsteemi tervele autopargile.

KASUTATUD KIRJANDUS

- 1) Georgia State University Department of Physics and Astronomy kodulehekül <http://hyperphysics.phy-astr.gsu.edu>. Vaadatud 29.11.2015.a.
- 2) Tech In Laboratuar Cihazları Sanayive Ticaret Ltd. Şti.© kodulehekül <http://www.techin.com.tr>. Vaadatud 29.11.2015.a.
- 3) ООО "Автомобильные инновации"© kodulehekül <http://www.autoinno.by>. Vaadatud 29.11.2015.a.
- 4) Victron Energy B.V. © kodulehekül <http://www.victronenergy.com>. Vaadatud 29.11.2015.a.
- 5) Ipanda Co. Ltd. © kodulehekül www.wipanda.com. Vaadatud 29.11.2015.a.
- 6) Digital Measure Co.© kodulehekül <http://www.digital-measure.com>. Vaadatud 29.11.2015.a.
- 7) Freescale© Inc. kodulehekül <http://www.freescale.com>. Vaadatud 29.11.2015.a.
- 8) Piller, S.; Perrin, M.; Jossen, A. Methodes for state-of-charge determination and their applications, – Ulm: Journal of power sources 96, 2001.– Lk. 113...120.
- 9) Arduino Co.© kodulehekül <http://www.arduino.cc>. Vaadatud 29.11.2015.a.
- 10) Maxim Integrated © kodulehekül <http://www.maximintegrated.com>. Vaadatud 29.11.2015.a.
- 11) Casacca, M.; Salameh Z. Determination of Lead-Acid Battery Capacity Via Mathematical Modeling Tehniques,– Lowell: Transaction on Energy Conversion, Vol. 7, No. 3, 1992.–Lk. 442...443.
- 12) Schneider M., Ilgin S., Jegenhorst N., Kube R., Püttjer S., Riemschneider K., Vollmer J. Automotive Battery Monitoring by Wireless Cell Sensors,– Hamburg: Hochschule für Angewandte Wissenschaften, 2012.– Lk. 1...5.

LISAD

Mõõtmisprotseduuri programm

Lisa 1

```
1  #include<OneWire.h>
2  #include<DallasTemperature.h>
3  #define ONE_WIRE_BUS 2
4
5  OneWire oneWire(ONE_WIRE_BUS);
6  DallasTemperature sensors(&oneWire);
7
8  void setup()
9  {
10  Serial.begin(9600);
11  sensors.begin();
12  }
13
14  void loop()
15  {
16
17  while (Serial.available()==0);           ///oota serial port
18
19  int x;                                   ///mootmise algus
20  x=1;
21  Serial.print("Begining Measurement. Delay = ");
22  Serial.print(x);
23  Serial.print(" minute.");
24  Serial.println("");
25  Serial.println("");
26
27  int n;
28  n=1;                                     ///esimene katse number
29
30  Loop:
31  int sensorValue = analogRead(A0);       ///pinge lugemine
32  float voltage = sensorValue * (5.0 / 1023.0)*5.288;
33  Serial.print(n);                         ///katse number
34  Serial.print("\t");
35  Serial.print(voltage);
36  Serial.print("\t");
37  sensors.requestTemperatures();          ///temperatuuri lugemine
38  Serial.print(sensors.getTempCByIndex(0));
39  Serial.println("");
40  n++;
41  delay(20*1000);                          ///paus 1 min
42  delay(20*1000);
43  delay(20*1000);
44  goto Loop;                               ///tagasi
45  }
```

```
1  #include<SPI.h>
2  #include<Ethernet2.h>
3  #include<OneWire.h>
4  #include<DallasTemperature.h>
5  #define ONE_WIRE_BUS 2
6
7  OneWire oneWire(ONE_WIRE_BUS);
8  DallasTemperature sensors(&oneWire);
9  byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
10 IPAddress ip(192, 168, 1, 17);
11 EthernetServer server(80);
12
13 void setup() {
14   sensors.begin();
15   Serial.begin(9600);
16   while (!Serial) {}
17
18   Ethernet.begin(mac, ip);
19   server.begin();
20   Serial.print("server is at ");
21   Serial.println(Ethernet.localIP());
22 }
23
24 void loop()
25 {
26   EthernetClient client = server.available();
27   if (client) {
28     Serial.println("new client");
29
30     boolean currentLineIsBlank = true;
31     while (client.connected()) {
32       if (client.available()) {
33         char c = client.read();
34
35         float p1;
36         float p2;
37         float p3;
38         float p4;
39         float p5;
40         float p6;
41         float p7;
42         float p8;
43         float p9;
44         float p10;
45
46         Serial.write(c);
47         if (c == '\n' && currentLineIsBlank) {
48           sensors.requestTemperatures();
```

```

59  client.println("HTTP/1.1 200 OK");
60  client.println("Content-Type: text/html");
61  client.println("Connection: close");
62  client.println("Refresh: 500");
63  client.println();
64  client.println("<!DOCTYPE HTML>");
65  client.println("<html>");
66  {
67  int sensorValue = analogRead(A0);
68  float voltage = sensorValue * (5.0 / 1023.0)*5.2634;
69  float SOC = (((sensorValue * (5.0 / 1023.0)*5.2634)-11.8)/1.3)*60)+40;
70
71  client.print("Sqiduk 123 ABC.");
72  client.println("<br />");
73  client.println(" ");
74  client.println("<br />");
75
76  client.print("Battery voltage is ");
77  client.print(voltage);
78  client.print(" V ");
79  client.println("<br />");
80
81  client.print(" SOC is ");
82  client.print(SOC);
83  client.print(" % ");
84  client.println("<br />");
85
86
87  client.print(" Temperature is ");
88  client.println(sensors.getTempCByIndex(0));
89  client.print(" deg C ");
90  client.println("<br />");
91
92  client.print(" Viimane koormuskatse: ");
93  client.println("<br />");
94  client.println(" ");
95  client.println("<br />");
96
97  client.print(p1);
98  client.println("<br />");
99  client.print(p2);
100 client.println("<br />");
101 client.print(p3);
102 client.println("<br />");
103 client.print(p4);
104 client.println("<br />");
105 client.print(p5);
106 client.println("<br />");
107 client.print(p6);

```

```
108 client.println("<br />");
109 client.print(p7);
110 client.println("<br />");
111 client.print(p8);
112 client.println("<br />");
113 client.print(p9);
114 client.println("<br />");
115 client.print(p10);
116 client.println("<br />");
117     }
118 client.println("</html>");
119 break;
120     }
121 if (c == '\n') {
122     currentLineIsBlank = true;
123     }
124 else if (c != '\r') {
125     currentLineIsBlank = false;
126     }
127     }
128     }
129 delay(1);
130 client.stop();
131 Serial.println("client disconnected");
132     }
133
134 }
```

