

TARTU ÜLIKOOL
LOODUS- JA TEHNOLOOGIATEDUSKOND

TIIT GINTER

Multispektraalse ruumhajumismõõtja riistvara ja
tarkvara funktsionaalne ning arhitektuuriline
disain

Magistritöö arvutitehnikast

Juhendaja: Aivo Reinart

Tartu 2011

Sisukord

1 Sissejuhatus.....	4
2 Mõisted.....	5
3 Kirjanduse ülevaade.....	7
3.1 Tagasihajumiskoeffitsient.....	7
3.2 Hajumise mõõtmine ja arvutamine.....	10
3.2.1 Loodusvete hajumiskoeffitsient.....	10
3.3 Arvutusalgoritmi ja tegevuste partitsioneerimine.....	12
3.4 Kontrolleri töö juhtimise variandid.....	14
3.4.1 Kas kasutada operatsioonisüsteemi?.....	16
3.5 Võimalikud kasutatavad liidesed.....	17
3.5.1 Serial Peripheral Interface Bus	17
3.5.2 Inter Integrated Circuits	17
3.5.3 RS-232.....	18
3.5.4 RS-485.....	19
3.5.5 Ethernet.....	19
4 Ruumhajumismõõtja funktsionaalse ja osalise arhitektuuriline disain.....	21
4.1 Tööpõhimõte.....	21
4.2 Riistvara ja tarkvara partitsioneerimine.....	24
4.3 Ülesannete võimalik jaotamine seadmesises ja välise arvuti vahel.....	26
4.4 MVSM juhtimise partitsioneerimine.....	28
4.5 Kontrolleri ülesanded.....	30
4.5.1 Juhtkontroller.....	30
4.5.2 Mõõtekontroller.....	31
4.6 Juhtarvuti ülesanded.....	33
4.7 Kontrolleri tööskeem (UML).....	34
4.8 Arvutusalgoritmi plokskeem.....	40
4.9 Kontrolleri võimalikud variandid.....	42
4.10 Kontrolleri valik.....	49
4.11 Kontrolleri töö realiseerimine.....	49
4.11.1 Juhtkontroller.....	49
4.11.2 Mõõtekontroller.....	49

4.11.3 Sisendid/väljundid.....	50
4.11.4 Muutujad ja parameetrid.....	52
4.12 Kasutatavad algoritmid.....	58
4.13 Mootorite juhtimine.....	59
4.14 Andmete salvestamine.....	59
5 Kokkuvõte.....	61
6 Summary.....	63
7 Kasutatud kirjandus.....	65

1 Sissejuhatus

Veekogude soojus- ja valgustingimuste modelleerimiseks on vajalik teada vee tagasihajumise funktsiooni. Selle mõõtmiseks kasutatakse enamasti fikseeritud nurgaga mõõteseadmeid, mille mõõtetulemuse järgi arvutatakse hajumisfunktsiooni väärtus. Siiani ei ole tootmises ühtegi universaalset tööstusliku seadet, mis võimaldaks teostada hajumisfunktsiooni mõõtmist kõigi hajumisnurkade ($0 \dots \pi$) korral vahetult veekogus. On mitmeid tööstuslikke fikseeritud nurgaga hajumisfunktsiooni mõõteseadmeid [1][2], samuti mõõteseadmeid, mis võimaldavad mõõta hajumist mingis nurgapiirkonnas erinevate nurkade korral. Üks vähestest teadaolevatest universaalse seadme prototüüpidest on valmistatud Kanada ja Ukraina teadlaste koostöona; selle prototüübi kaasajastamine ja väiketootmiseks ettevalmistamine on plaanis teostada Tartu Observatooriumi, Ukraina TA Mereinstituudi ja AS Interspektrum koostöös. Seadmel on vaja välja töötada ja valmistada uus juhtelektroonika ja tarkvara lähtudes tänapäevasest manussüsteemide väljatöötamise ja koostamise printsiipidest. Samuti on vaja optimeerida seadme optiline ja mehaaniline ehitus tagamaks sobivust väiketootmise võimalustega.

Käesoleva töö eesmärgiks oli multispektraalse ruumhajumismõõtja juhtarvuti riistvara ja tarkvara funktsionaalse disaini koostamine moderniseeritud seadme prototüübi valmistamiseks. Juhtkontrolleri osas oli samuti eesmärgiks analüüsida funktsionaalse disaini sobitamist konkreetsete arhitektuurilise disaini variantidega.

Uurimisülesanneteks oli esmalt tutvuda vee optiliste omaduste määramise algoritmide ja meetodikaga ning tulemuste põhjal analüüsida vee hajumise parameetreid mõõtvate seadme juhtimiseks ja vajalike vee optiliste omaduste arvutamiseks sobivaid algoritme, hinnata nende arvutusmahukust ja tulemustest lähtuvalt koostada seadme funktsionaalne disain. Funktsionaalse disaini tulemuste põhjal analüüsiti arhitektuurilise disaini võimalikke variante.

Antud töö kirjutamisel on arvesse võetud Aare Puussaare (2011) lõputöö [3] raames koostatud multispektraalse ruumhajumismõõtja nõuete kirjeldust ja analüüsi.

2 Mõisted

ADC (*analog digital converter*) – seade, mis muudab analoogsignaali (pinge, vool) digitaalseks koodiks (binaarväärtuseks) [4].

ANSI (*American National Standards Institute*) – Ameerika Rahvuslik Standardite Instituut [5].

Baud rate – boodides mõõdetav andmeedastuskiirus. Väikeste andmeedastuskiiruste ja lihtsate algoritmide korral võrdub edastatavate bittide arvuga sekundis [5].

BGA (*ball grid array*) – kiipide pindmontaaž-pakendamise standard. Ühendusviigud paiknevad kiibi põhja all [6].

DAC (*digital analog converter*) – seade, mis muudab digitaalse koodi (binaarväärtuse) analoogsignaaliks (pinge, vool) [4].

EIA (*Electronic Industries Alliance*) – standardi- ja kaubandusorganisatsioonide liit, kuhu kuulusid USA elektroonika tootjad. Lõpetas tegevuse 2010. a.

Ethernet – üldnimetus arvutivõrgu tehnoloogiatele andmeedastuseks kohtvõrkudes [7].

FAT (*File Allocation Table*) – failipaigutustabel on spetsiaalne failide paiknemist kirjeldav tabel, mida operatsioonisüsteem hoiab andmekandjal (kõvaketas, mälukaart vms) selleks ettenähtud kaitstud piirkonnas ja, kus kirjeldatakse failide füüsilist paigutust andmekandjal [5].

Full-duplex – täisdupleks- ehk lihtsalt dupleksedastus tähendab, et andmeid on võimalik edastada samaaegselt mõlemas suunas [5].

Half-duplex – pooldupleks-andmeedastus tähendab, et andmeid saab signaalikandjal edastada mõlemas suunas, kuid mitte samaaegselt [5].

I²C (*inter-integrated circuit*) – kahesuunaline kahe liiniga sünkroonne jadaliidese standard. Kasutatakse ka lühendit IIC [4].

Kernel – tarkvaraline vahekiht rakenduse ja riistava vahel. Enamasti kasutatakse kerneli mõistet opsüsteemi tuuma keskse osa nimetusena. Kernel korraldab opsüsteemi ressursijaotust ja realiseerib põhifunktsioone [8].

Kiirgusvoog (*radiant flux, radiant power*) – elektronmagnetkiirguse võimsuse mõõt. Mõõtühik W [9].

Kiirgustugevus (*radiant intensity*) – iseloomustab kiirguri kiirgusvoogu antud suunas. Mõõtühik Wsr⁻¹ [9].

Kiiritustihedus (*irradiance*) – elektronmagnetkiirguse võimsus pindalaühiku kohta. Mõõtühik Wm^{-2} [9].

Kiirgusvoo spektraalne tihedus (*spectral radiant power*) – kiirgusvoo jaotus lainepikkuse järgi. Mõõtühik Wm^{-1} [9].

LED (*light-emitting diode*) – pooljuhtdiod, mis kiirgab valgust kui teda läbib vool pärisuunas [10].

MVSM (*multi volume scattering meter*) – multispektraalse ruumhajumisfunktsiooni mõõtja [11].

PID (*proportional–integral–derivative*) – klassikaline tagasidedega juhtimise algoritm [12].

Pimevool – vool kiirgussensori väljundis, kui talle valgust ei lange [13].

PMT (*photomultiplier tube*) – vaakumfotokordisti – tundlik valguse mõõteseade, mis võimendab vahetut kiirguse toimet tekkinud voolu kuni 100 miljonit korda [13].

PWM (*pulse-width modulation*) – moodus digitaalse signaali muundamiseks analoogsignaaliks kasutades impulsi pikkuse modulatsiooni; kasutatakse tihti mootori juhtimiseks [4].

RS-232C – asünkroonne jadaühenduse standard seademetevaheliseks andmevahetuseks [14].

RS-485 – asünkroonse jadaliidese elektrilise signaali standard saate ja vastuvõtu seadmete vaheliseks suhtluseks. Kasutatakse ka lühendit EIA-485 [15].

RTOS (*real-time operating system*) – reaalaajaoperatsioonisüsteem. Operatsioonisüsteem, mis tagab tegumite deterministliku ajalise käitumise [4].

SPI (*serial peripheral interface bus*) – sünkroonne jadaühenduse standard [16].

TIA (*Telecommunications Industry Association*) – globaalne kaubandus ühing mis esindab umbes 600 telekommunikatsiooniettevõtte huve [17].

UART (*universal asynchronous receiver/transmitter*) – riistvara komponent, mis teisendab andmeid jada ja paralleelse esituse vahel. Mõnikord kasutatakse ka RS-232 liidese märkimiseks [18].

USB (*universal serial bus*) – universaalne jadaühendus standard andmevahetuseks elektroonikaseadmete vahel [4].

UML (*unified modeling language*) – standardiseeritud üldotstarbeline modelleerimise keel [4].

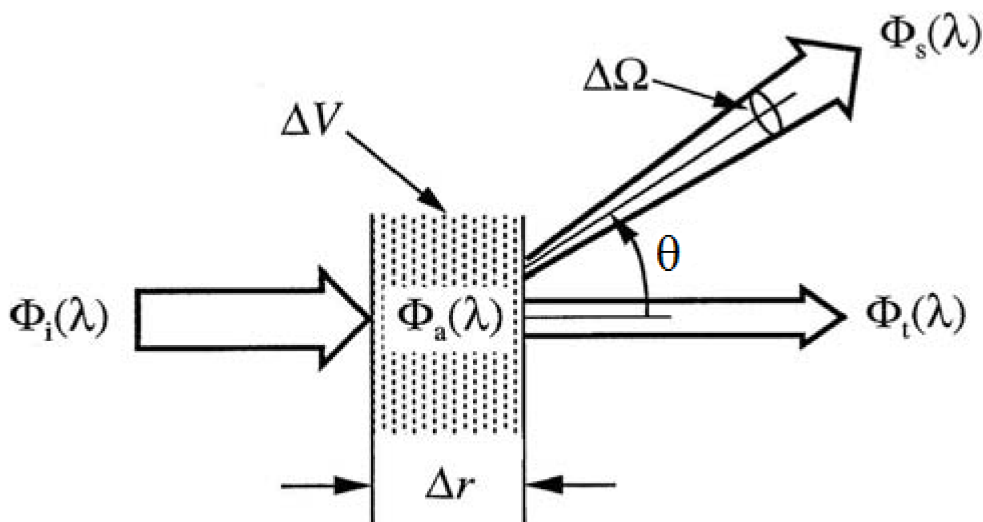
VSF (*volume scattering function*) – ruumhajumisfunktsioon [11].

3 Kirjanduse ülevaade

3.1 Tagasihajumiskoeffitsient

Ruumhajumisfunktsioon (VSF *volume scattering function* ($\text{m}^{-1} \cdot \text{sr}^{-1}$)) on üks vee esmastest optilistest omadustest (*inherent optical property*), mis kirjeldab valguskiire hajumise sõltuvust hajumisnurgast. Sealjuures kirjeldavad VSF ja neeldumiskoeffitsient (a , (m^{-1})) täielikult keskkonna optilised omadused. Kui VSF ja neeldumiskoeffitsient siduda langeva kiirguse nurk- ja spektraalse jaotusega vahetult veepinna all, on võimalik modelleerida kiirguslevi valemi abil kiirgusvoo tasakaalu keskkonnas. Selline arvutus on olulise tähtsusega veekogude pindmise osa soojustasakaalu uurimiseks, samuti vees toimuva fotosünteesi produktiivsuse ja fotoaktiivsete keemiliste ühendite reaktsioonide uurimiseks. Lisaks sellele on spektraalsel ruumhajumisfunktsioonil võtmetähtsus veekogude seisundi hindamisel kaugseire meetoditega (nt satelliitidelt), kui mõõdetakse veekogudest tagasihajunud päikesekiirgust ja selle spektraalset jaotust [19].

Vaatleme väikest kogust vett ruumalaga ΔV ja paksusega Δr . Valgustame seda kitsa paralleelse monokromaatse valguskiirega (joonis 1) mille lainepikkus on λ ning kiirgusvoo spektraalne tihedus (*spectral radiant power*) on $\Phi_i(\lambda)$ (Wnm^{-1}).



Joonis 1: Valguskiire langemine ja hajumine [19].

Algsest kiirgusvoost $\Phi_i(\lambda)$ mingi osa $\Phi_a(\lambda)$ neeldub vees, mingi osa $\Phi_s(\theta, \lambda)$ hajub nurga θ all ja ülejäänud $\Phi_t(\lambda)$ läbib keskkonna suunda muutmata. Kogu kiirgusvoo spektraalse tiheduse saab esitada valemiga 1.

$$\Phi_i(\lambda) = \Phi_a(\lambda) + \Phi_s(\lambda) + \Phi_t(\lambda) \quad (1)$$

Lainepikkusest sõltuv hajumisfunktsioon ehk spektraalne hajuvus (*spectral scatterance*) $B(\lambda)$ on see osa langenud kiirgusest, mis on vees levides algsest suunast kõrvale kaldunud (valem 2):

$$B(\lambda) \equiv \frac{\Phi_s(\lambda)}{\Phi_i(\lambda)}. \quad (2)$$

Spektraalne hajumiskoeffitsient (*spectral scattering coefficient*) $b(\lambda)$ ehk hajuvus pikkusühiku kohta keskkonnas on defineeritud valemiga 3:

$$b(\lambda) \equiv \lim_{\Delta r \rightarrow 0} \frac{B(\lambda)}{\Delta r} (m^{-1}). \quad (3)$$

Vaatame nüüd spektraalse hajuvuse nurkjaotust $B(\lambda, \theta)$, kus langev kiirgus hajub nurga θ all ja hajumise ruuminurk on suurusega $\Delta\Omega$ (joonis 1). Nurka θ nimetatakse hajumisnurgaks (*scattering angle*) ja ta omab väärtusi vahemikus 0 kuni π . Hajuvuse pikkusühiku ja nurgaühiku kohta $\beta(\lambda, \theta)$ saab esitada valemiga 4:

$$\beta(\lambda; \theta) \equiv \lim_{\Delta r \rightarrow 0} \lim_{\Delta\Omega \rightarrow 0} \frac{B(\lambda; \theta)}{\Delta r \Delta\Omega} = \lim_{\Delta r \rightarrow 0} \lim_{\Delta\Omega \rightarrow 0} \frac{\Phi_s(\lambda; \theta)}{\Phi_i(\lambda) \Delta r \Delta\Omega} (m^{-1} sr^{-1}). \quad (4)$$

Spektraalse hajunud kiirguse ruuminurga $\Delta\Omega$ kohta saame arvutada valemiga 5.

$$\Phi_s(\lambda; \theta) = I_s(\lambda; \theta) \Delta\Omega \quad (5)$$

Kui spektraalne kiirgusvoog $\Phi_i(\lambda)$ langeb pinnale ΔA siis sellele vastav spektraalne kiiritustihedus on $E_i(\lambda) = \Phi_i(\lambda) / \Delta A$, ja kuna $\Delta V = \Delta r \Delta A$, saame valemi 4 esitada kujul (valem 6) [19]:

$$\beta(\lambda; \theta) = \lim_{\Delta V \rightarrow 0} \frac{I_s(\lambda; \theta)}{E_i(\lambda) \Delta V}. \quad (6)$$

Edaspidi jätame valemitest välja kiirguse lainepikkuse, kuna mõõtmised tehakse erinevate lainepikkuste jaoks eraldi ja kõik mõõdetud ja arvutatud tulemused vastavad mingile fikseeritud kitsale lainepikkuste vahemikule.

Integreerides funktsiooni $\beta(\theta)$ üle nurkade $\pi/2$ kuni π , saame arvutada tagasihajumiskoeffitsiendi $b_b(m^{-1})$ (valem 7). Hajumiskoeffitsient b arvutatakse kasutades sarnast valemit, erinevuseks on vaid see, et integraal arvutatakse nurga vahemikus 0 kuni π :

$$b_b = 2\pi \int_{\theta=\pi/2}^{\pi} \beta(\theta) \sin(\theta) d\theta. \quad (7)$$

Tagasihajumise tõenäosus on defineeritud kui tagasihajumiskoeffitsiendi ja hajumiskoeffitsiendi suhe b_b/b .

Fotodetektorid mõõdavad kiirgusvoogu (*radiant flux*) (kiirgusenergia ajaühiku kohta (W)). Seega on otstarbekas esitada valem 6 kasutades otsest ja hajunud kiirgusvoogu. Hajunud kiirgusvoog on määratud mõõteriista geomeetriliste ja optiliste parameetritega. Valemis 8 on I kiirgustugevus (*radiant intensity*) (võimsus nurga ühiku kohta $\text{W} \cdot \text{sr}^{-1}$), Ω on fotodetektori vaatenurk (*viewing angle*) (sr), c on kiire nõrgenemise koeffitsient (*beam attenuation coefficient*) (m^{-1}) ja r on vahemaa hajumispunktist fotodetektorini (m):

$$F(\theta) = I(\theta) \cdot \Omega \cdot e^{-c \cdot r}. \quad (8)$$

Kiiritustihedus (*irradiance*) E hajumispunktis on määratud vette siseneva kiirgusvooga F_0 , mis nõrgeneb (osaliselt neeldub ja osaliselt hajub) läbides vees distantssi r_l valgusallikast kuni hajumise ruumi keskkoha, S on langeva valgusvoo ristlõikepindala:

$$E = \frac{F_0}{S} \cdot e^{c \cdot (r + r_l)}. \quad (9)$$

Kombineerides valemeid 6, 8 ja 9 saame valemi 10:

$$\beta(\theta) = \frac{F(\theta) \cdot S \cdot e^{c \cdot (r + r_l)}}{F_0 \cdot \Omega \cdot V(\theta)}. \quad (10)$$

Valemi 10 kasutamiseks peavad kiirgusvoog F_0 ja nõrgenemiskoeffitsient c olema mõõdetud, kui soovetakse saada hajumisfunktsiooni väärtust antud suunas. Kuna nende suuruste mõõtmine ei ole alati võimalik, siis avaldatakse hajunud kiirguse nurkjaotus reeglina suhtelistes ühikutes. Valguse nõrgenemist suunas $\theta = 0$ saab arvutada Lambert-Bouguer'i (valem 11) seadust kasutades [11]. Valemis 11 tähistab l teepikkust, mille valgus läbib, I_0 on langev kiirgustugevus ning I kiirgustugevus kaugusel l [20]. Fotodetektorisse jõudva hajunud kiirguse võime analoogiliselt valemile 11 avaldada valemiga 12:

$$\frac{I}{I_0} = e^{-cl}. \quad (11)$$

$$F(0) = F_0 \cdot e^{-c \cdot (r + r_l)} \quad (12)$$

Kasutades valemeid 10 ja 12 saab tuletada valemi 13. Hajunud ja otsese nõrgenenud kiirgusvoo mõõtmine võimaldab arvutada ruumhajumisfunktsiooni $\beta(\theta)$ absoluutse väärtuse, kuna ülejäänud parameetrid valemis on võimalik määrata seadme geomeetria põhjal [11]:

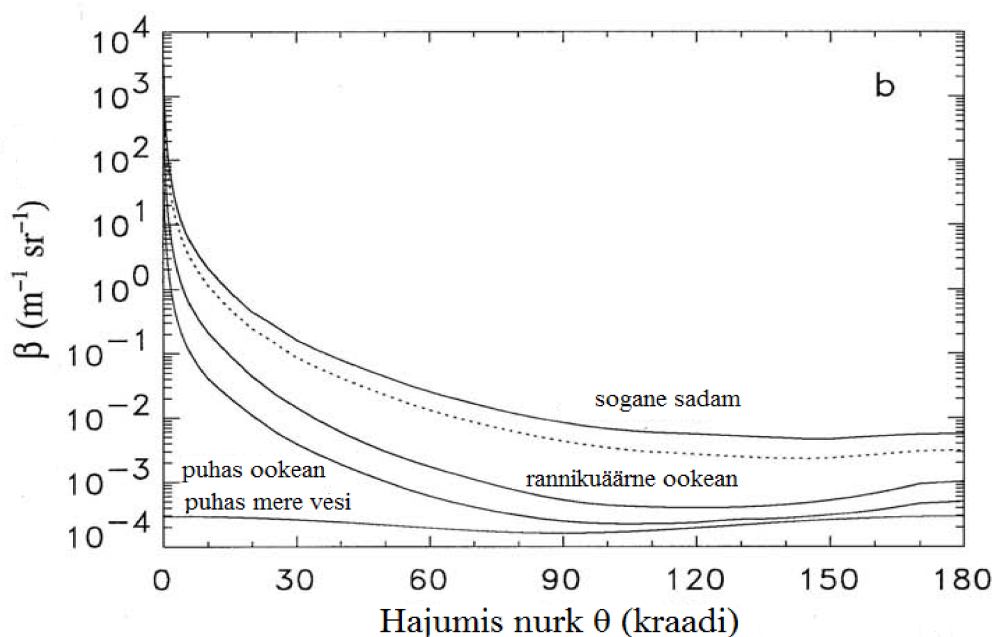
$$\beta(\theta) = \frac{F(\theta) \cdot S}{F(0) \cdot \Omega \cdot V(\theta)} \quad (13)$$

3.2 Hajumise mõõtmine ja arvutamine

Ruumhajumisfunktsiooni mõõdetakse 180° ulatuses ($0 \dots \pi$). See ulatus jaotatakse vastavalt hajumisnurgale kolmeks piirkonnaks: eesmine (*forward*) ($0^\circ - 10^\circ$), üldine (*general*) ($10^\circ - 170^\circ$) ja tagumine (*backward*) ($170^\circ - 180^\circ$). Väikeste hajumisnurkade korral on reeglina ruumhajumisfunktsiooni mõõtmise probleemiks mõõteseadme osadelt peegeldunud ja hajunud otsese kiire genereeritud taustvalgus, kuna otsese kiire intensiivsus on mitu suurusjärku suurem kui mõõdetav hajunud kiirgus. Üldises piirkonnas mõõtmisi tehes on põhiprobleemiks mõõteväärtuste suur dünaamiline ulatus ja väga väike hajunud kiirguse hulk 90° hajumisnurkade läheduses. Suurte nurkade korral (180° lähedal) on samuti probleemiks väga väike hajunud kiirguse hulk. Nende probleemide tõttu on keeruline teha universaalset seadet, mis võimaldaks teostada mõõtmisi kõigis kolmes vajalikus piirkonnas [11].

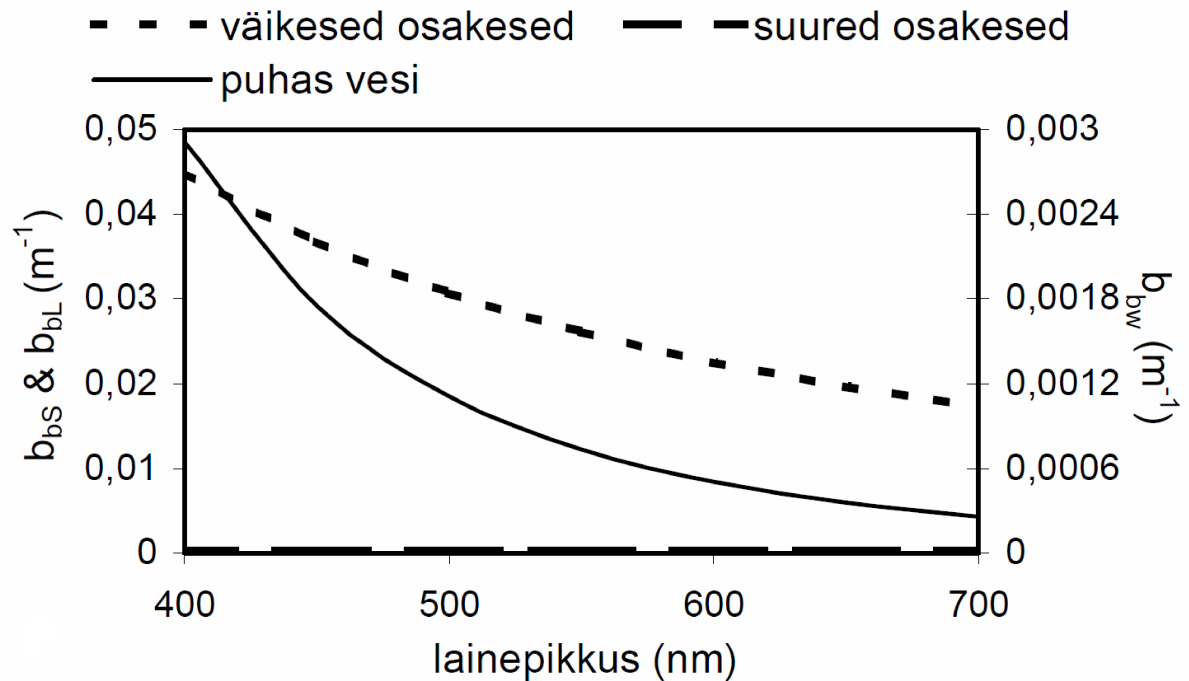
3.2.1 Loodusvete hajumiskoeffitsient

Illustreerimaks suurusi, mida on vaja mõõta, on toodud kaks graafikut. Joonisel 2 on kujutatud tagasihajumisfunktsiooni väärtuse sõltuvus hajumisnurgast. Antud graafikult on näha, et funktsiooni väärtus sõltub väga tugevasti mõõdetava vee omadustest [19]. Samuti on näha hajumiskoeffitsiendi väga suur muutumine hajumisnurga muutudes – koeffitsiendi dünaamiline ulatus on üle kuue suurusjärgu.



Joonis 2: Tagasihajumiskoeffitsient erinevates vetes [19].

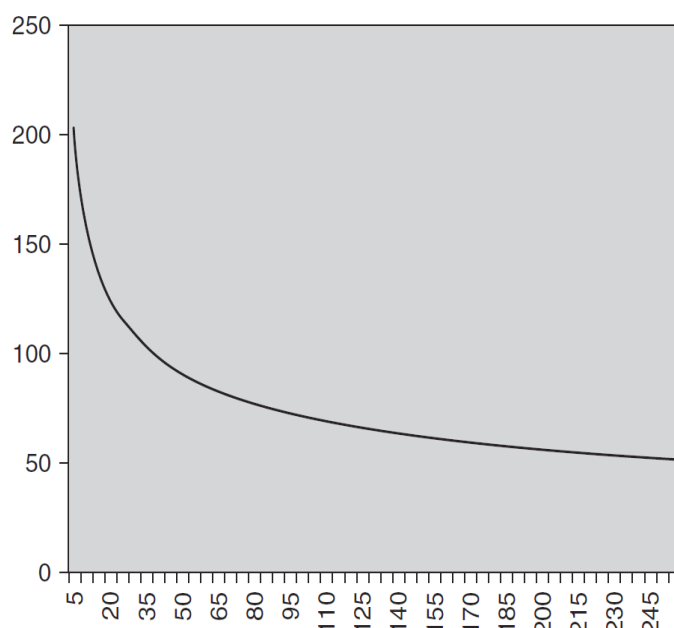
Joonisel 3 on kujutatud tagasihajumiskoeffitsiendi sõltuvusest valguse lainepikkusest ja vees olevate osakeste suurusest: b_{bs} on väikeste osakeste, b_{bL} suurte osakeste ja b_{bw} puhta vee hajumiskoeffitsient. On näha, et sõltuvalt lainepikkusest ja osakeste suurusest koeffitsiendi väärtus erineb rohkem kui 100 korda [21].



Joonis 3: Tagasihajumiskoeffitsiendi sõltuvus valguse lainepikkusest ja vees olevatest osakekest [21].

3.3 Arvutusalgoritmi ja tegevuste partitsioneerimine

Inimese lühimälu on nagu arvuti vahemälu - suudab korraga meeles pidada ainult 5-9 erinevat asja – suurema hulga asjade puhul hakatakse vanu asendama uutega. Keerulisemad funktsioonid ja programmid nõuavad programmeerija mälus rohkemate asjade meelespidamist ja arvestamist kui ta suudab ning seetõttu programmeerijal kulub sellise koodi koostamiseks ja silumiseks palju aega ning vead tekivad kergesti. Kui arvutitehnika arengu algfaasis oli mikrokontrollerite hind võrreldes tarkvaraarendaja tunnitasega suur, siis tänapäeval on olukord vastupidine. Veel enam, lisaks arendamisele kuluvale rahale, mängib suurt rolli ka kvaliteet ja arendamise kiirus. Kõige selle tõttu on keerulisemate ülesannete lahendamisel vajalik jagada realiseeritavad ülesanded sobiva suurusega väiksemateks osadeks. Jaotatud funktsionaalsust on märksa lihtsam realiseerida ja testida. Lisaks sellele võimaldab süsteemi jagamine osadeks sama probleemiga korraga tegeleda rohkematel arendajatel, ilma et nad peaksid kulutama suure osa ajast omavaheliseks suhtluseks ja väikeste muudatuste integreerimiseks suurde süsteemi. Joonisel 4 on kujutatud tüüpilise arendaja produktiivsuse sõltuvust programmi suurusest. On näha, et mida rohkem on programmi ridu, seda vähem produktiivsed on selle projektiga tegelevad programmeerijad. Sellest lähtuvalt on mõttekas hoida lähtekoodi pikkust umbes 20 tuhande rea juures [22].



Joonis 4: Produktiivsuse ja programmi suuruse sõltuvus. Vertikaalteljel on kujutatud produktiivsus tuhat rida lähtekoodi kuus ja horisontaalteljel programmi koodi pikkus tuhandetes ridades [22].

Manussüsteemide korral on tihti mõistlik jagada ülesanne osadeks jaotades tegevused sobivalt

mitme kontrolleri vahel. Lisaks sellele tuleks võimalusel kasutada ülesannete täitmiseks spetsiaalselt nende ülesannete jaoks loodud valmiskomponente. Varem lahendati tihti kõik ülesanded tarkvaraliselt kontrolleri sees, kuid selline lähenemine on väga töömahukas ja vigaderohke. Valmis kontrolleri kasutamise näiteks võib tuua UART liidese realiseerimise, kus praegu kasutatakse enamasti valmis riistvarakomponenti, mis teostab suhtluseks vajalikud ajastus ja signaali teisendamise toiminguid ning lubab põhikoodis keskenduda ainult andmevahetusele. Seega on soovitatav võimaluse korral kasutada spetsiaalseid kontrollereid ja nende vahel informatsiooni vahetamiseks kasutada mõnda lihtsamat standardset suhtlusprotokolli, nagu näiteks I²C [22].

3.4 Kontrolleri töö juhtimise variandid

On mitu põhimõtteliselt erinevat võimalust kontrolleri siseste protsesside juhtimiseks [8]. Olenevalt kontrolleri töötavatest protsessidest tuleb valida viis, mis sobiks kõige paremini vajalike ülesannete lahendamiseks. Lihtsamad juhtimisviisid ei vaja katkestusi, ega ka reaalaajaoperatsioonisüsteemi (RTOS). Programmi töötsükli on võimalik korraldada järgnevate meetoditega:

- **Küsimistsükkel** – kontrolleri käivitamisel minnakse lõpmatusse tsükklisse. Tsükklis kontrollitakse sisend-väljundseadmete seisundit ja vastavalt seisundile teostatakse vajalik tegevus. Selline skeem töötab hästi, kui protsessor on määratud töötleva kiiret sisend-väljundseadet ning ühegi sündmuse vahelejäämine ei ole lubatud. Küsimistsükli kasutatakse ka tagaplaani ülesandena katkestustepõhises süsteemis ning tsüklilise täitmise puhul sündmustepõhiste toimingute täitmiseks. Üldiselt selline töötlus raiskab palju protsessori aega sündmuste kontrollimisele ja ei sobi keerulisemate protsesside juhtimiseks.
- **Sünkroniseeritud küsimistsükkel** – küsimistsükli variant, kus lisatakse küsimisele ja ootamisele mingisugune ajaarvestussüsteem, näiteks taimer. Selline lahendus on hea, kui kontrollitav sündmus toimub lühikese aja jooksul korduvalt, kuid reageerima peab ainult ühele toimumisele. Tüüpiline olukord on näiteks nupuvajutusest tekkiv häireimpulsside jada.
- **Tsükliline täitmine** – kontrolleri töötsükli täidetakse järjest kõik ülesanded. Kui tegemist on kiire kontrolleri ja protsessid on jagatavad piisavalt väikesteks osadeks, on sel viisil võimalik saavutada kiire reageerimisvõime sündmustele. Antud skeem ei ole aga sobiv, kui täidetavad ülesanded on pikad või ei ole neid võimalik ülesannete iseloomu tõttu osadeks jagada, kuid sellest hoolimata on vaja kiiresti reageerida teatud sündmustele.
- **Seisundipõhine täitmine** – vastavalt erinevatele võimalikele süsteemi olekutele jaotatakse kood osadeks. Iga seisundi korral täidetakse selle seisundi jaoks vajalik koodi osa. Sellisel viisil koodi täitmine võimaldab kokku hoida arvutusaega, samuti saab sellist süsteemi matemaatiliste meetoditega optimeerida. Negatiivseks pooleks on see, et kõiki protsesse ei ole võimalik sobivalt seisunditeks jaotada. Lisaks muutub keerukamate süsteemide puhul seisundite tabel väga suureks ning raskesti hallatavaks.
- **Kaasprogrammid** – selle meetodi puhul jaotatakse kõik ülesanded mitmeks osaks. Erinevate ülesannete alamosadid kutsutakse tsükliks välja globaalsete muutujate poolt määratud

järjestuses. Sel viisil saavad kõik protsessid endale piisavalt arvutusaega ja nii on võimalik tekitada illusioon ülesannete samaaegset täitmisest. See skeem on sobiv kasutada koos seisundipõhise ülesannete täitmisega. Nii saavad alamprotsessid sisemiselt vajadusel oodata teatud seisundit, ilma et teised protsessid samuti ootama peaksid. Kaasprogrammide abil on võimalik realiseerida väga keerulisi süsteeme, kuid nende loomine nõuab programmeerijalt ranget distsipliini ja sobivat rakendust.

- Katkestuspõhine juhtimine – puhtalt katkestuspõhise süsteemi põhiprogramm koosneb hüppest iseendale ja erinevate ülesannete täitmist algatatakse riist- või tarkvaralise katkestusega. Tekkivaid katkestusi töödeldakse spetsiaalsetes katkestusprogrammides. Kuna katkestus võib tekkida mõne teise protsessi täitmise ajal, peab iga katkestuse alguses kontrolleri seisu salvestama ja töötlemise lõpus taastama. See toiming võib võtta märkimisväärse osa protsessori tööajast, kui katkestused tekivad tihti.
- Eelisprioriteediga süsteem – sel juhul on erinevatele katkestustele omistatud erinevad prioriteedid. Katkestuse töötlemisel ei vaadata enam ainult katkestuse tekkimise järjekorda, vaid lisaks ka nende prioriteeti. Nii on võimalik kindel olla, et tähtsamad ülesanded saavad varem täidetud kui vähemtähtsad. Samas võib tekkida olukord, kus madalama prioriteediga katkestusi ei töödelda üldse, kui kõrgema prioriteediga katkestusi tekib piisavalt tihti.
- Esiplaani/tagaplaani süsteemid – kombinatsioon katkestuste ja küsimistsükliga juhtimisest. Suurema tähtsusega ülesandeid täidetakse esiplaanil kasutades katkestusi ja väiksema tähtsusega tegevused täidetakse protsessori jõudeajal tagaplaanil. Esiplaani ülesanded on reeglina katkestustega täidetavad ja tagaplaani ülesandeid täidetakse peatsüklis. Lisaks jaotatakse ülesanded ka prioriteetide järgi, mis on eriti oluline katkestuste poolt juhitud esiplaani funktsioonide täitmisel. Sellise lahenduse saab laiendada operatsioonisüsteemiks, lisades täiendavat funktsionaalsust, nagu arvutivõrgu kasutamine, failisüsteemi kasutamine ja seadmete ajurid. Esiplaani/tagaplaani süsteemi kasutatakse kontrolleri põhistes manussüsteemides kõige sagedamini.

- Täiemahuline reaalaajaoperatsioonisüsteem (RTOS) – selle mudeli puhul on kõik funktsioonid realiseeritud eraldiseisvate tegumitena, mida vastavalt süsteemi arhitektuurile täidetakse. Suure tegumite hulga korral võib operatsioonisüsteemi funktsioonide täitmine, nagu näiteks tööolekus tegumite vahetamine, põhjustada märkimisväärse osa protsessori koormusest. Samas on RTOS kasutamise korral võimalik peita programmeerija eest suurem osa tööjaotuse keerukusest, mis võimaldab luua keerulisemat süsteemi väiksema vaevaga [4] [8].

3.4.1 Kas kasutada operatsioonisüsteemi?

Reaalaaja operatsioonisüsteemi vajadus ja kasutamise otstarbekus sõltub väga palju projekti nõuetest ja seadme isaärasustest. Tavaliselt valitakse valmis RTOS, kuna see on töökindel, seda on lihtne kasutada ja RTOS on reeglina portatiivne. Valmis operatsioonisüsteemid varieeruvad suuresti funktsionaalsuse, jõudluse ja toetatavate seadmete poolest. Tihti on nendega kaasas ka arenduskeskkonnad ja silumise vahendid ning nad suudavad töötada paljudes erinevates seadistustes. Valmis RTOS on hea kasutada, kui ta vastab ülesande ajapiirangutele, on mõistliku hinnaga ja kui süsteem peab suutma töötada mitmel erineval riistvaraplatvormil.

Kuigi valmis RTOS garanteerib deterministliku ajalise käitumise ning toetab mitme ülesande täitmist ajajaotusega, on nende kasutamisel ka mitmeid miinuseid. Näiteks, võrreldes enda kirjutatud katkestuste poolt juhitava programmiga, on RTOS aeglasem, kuna tegumitepõhine juhtplokki mudel, mida enamasti kasutatakse, tekitab täiendavat ajakulu. Lisaks sellele sisaldab valmis RTOS reeglina ka mitmeid antud projektile mittevajalikke lisavõimalusi, et olla sobiv laiemale kasutajate gruppidele. Selliste lisade tööaeg ja mäluvajadus võib olla liigne konkreetse ülesande korral. Samuti võivad RTOS näitajad olla eksitavad või kajastada ainult parima juhu näitajaid, mille järgi on tegelikku reageerimisega keeruline määrata. Keerulisema süsteemi, mis tegeleb mitmete katkestustega ning teiste protsessidega, mis omavahel andmeid vahetavad, puhul aga on käsitsi reaalaaja tuuma (*kernel*) kirjutamine väga keeruline. Seetõttu tuleks võimalusel siiski eelistada valmis RTOS-i [8].

3.5 Võimalikud kasutatavad liidesed

MVSM moodulite vaheline suhtlus võiks olla realiseeritud kasutades enamlevinud standardseid jadaliideseid. Võimalik on valida Serial Peripheral Interface Bus (SPI), Inter Integrated Circuits (I²C), RS-232C, RS-485 ja Ethernet hulgast. Neist esimesed kaks on enamasti mõeldud komponentide omavaheliseks suhtluseks ning viimased kolm seadme ja juhtarvuti vaheliseks suhtluseks. Paljudel tänapäeva kontrollritel on sisseehitatud nende liideste kasutamise lihtsustamiseks mõeldud riistvaralised komponendid. Vastavalt sobivale liidesele saab valida seda toetava kontrolleri. Samas, väiksemad mõõteseadmed võivad nõuda neist ühe (kas I²C või SPI) valimist. Valikust jääb välja USB, kuna USB kasutamisel on maksimaalne lubatud kaabli pikkus 5 m [23], mis muudab ta ebasobivaks seadme ja välise arvuti vaheliseks suhtluseks ning seadme moodulite vaheliseks suhtluseks on antud liidese kasutamise protokoll liiga programmeerimis- ja jõudlusmahukas.

3.5.1 Serial Peripheral Interface Bus

Serial Peripheral Interface Bus ehk SPI on Motorola pool välja arendatud sünkroonne jadaühenduse protokoll. SPI toetab samaaegset kahe-suunalist andmevahetust kiirusel kuni 10 Mbit/s. See on lihtne 4 juhtmega jadaliides, mida kasutavad paljud kiibid, mis võimaldavad kontrollril ja välisseadmetel üksteisega suhelda. SPI töötati välja selleks, et kontroll saaks suhelda välisseadmega, kuid sama edukalt on selle abil võimalik panna omavahel suhtlema ka kaks kontrollrit. SPI on mõeldud suhtluseks suurtel andmeedastuskiirustel, aga lühikeste vahemaadega. Seda on võimalik kasutada ka pikemate vahemaade korral (kaugemate seadmega kui kontrolleri plaadil olevad komponendid), kuid siis tuleb vähendada andmeedastuskiirust ning seega kaotab antud protokoll oma peamise eelise. SPI-d kasutatakse peamiselt ühe ülem/alluv rakenduses, kus on vaja edastada kiiret andmevoogu. SPI miinuseks võib pidada, et rohkem kui õhe alluvaga süsteemis on iga alluva jaoks vaja kasutada eraldi valikusignaali (*chip select*), samas ülejäänud ühendused võivad olla paralleelsed [16].

3.5.2 Inter Integrated Circuits

Inter Integrated Circuits ehk I²C on Philipsi poolt välja arendatud kahe-suunaline kahe liiniga sünkroonne jadaliidese standard. I²C võimaldab kasutada seadmete eristamiseks 7-bitist aadressiruumi 16 reserveeritud aadressiga, mis teeb kokku võimalike seadmete arvuks ühe liidese korral 112. Seadmete arvu piirab peale aadresside hulga ka liini lubatud maksimaalne mahtuvus 400 pF. Kaks liini, mida antud standardis kasutatakse andmeedastuse realiseerimiseks, on andmete liin (Serial data - SDA) ja kella liin (Serial Clock - SCL). Andmete liinil kantakse üle soovitud infot

ning kella liin on vajalik andmeedastuse protsessi sünkroniseerimiseks. I²C järgib samuti ülem/alluv suhtlusprotokolli, kuid erinevalt SPI-st on tegu mitme võimaliku ülemaga liidesega. See tähendab, et kõik liinil paiknevad seadmed võivad algatada kommunikatsiooni. Seade, mis algatab suhtluse, on ülem ja seade, millega suhtlema hakatakse on alluv. Kuna ülemseade genereerib alati taktsignaali, peavad kõik seadmed, mis tahavad alustada suhtlust, olema võimalised takti genereerimiseks. Olenevalt kasutatavast režiimist varieerub liidese kiirus alates 10 kbit/s kuni 400 kbit/s. Kõige sobivam on I²C juhul, kui on vaja ühendada mitu seadet omavahel vähest hulka juhtmeid kasutades. Puuduseks on tarkvara lisakoormus adresseerimise ja vookontrolli tõttu [24]. Samuti ei saa liidest hästi kasutada, kui soovitakse komponendid üksteisest elektriliselt eraldada, kuna andmete- ja kellaliinil toimub kahe-suunaline liiklus.

3.5.3 RS-232

RS-232C on laialt levinud asünkroonne jadaühenduse standard seadmetevaheliseks andmevahetuseks. RS-232C standard toetab kahte tüüpi ühendusi: 25 klemmiga (DB-25) ja 9 klemmiga (DB-9). Kuigi 25 klemmiga ühendus on defineeritud, kasutatakse seda tänapäeval harva, kuna enamus rakendusi ei vaja kõiki defineeritud signaale. RS-232C puhul edastatakse infot andmesõnu kasutades, kusjuures andmesõna pikkus võib varieeruda 5...8 biti vahel ning bitid edastatakse varem kokkulepitud fikseeritud sagedusel. Lisaks andmetele edastatakse veel sünkroniseerimiseks ja veakontrolliks vajalikke bittide. Saatjal ja vastuvõtjal peavad kasutatavate bittide arv ja saatmissagedus olema defineeritud identselt, muidu andmeid edastada ei õnnestu. Peale esimese (sünkroniseerimise) biti vastuvõtmist arvutab vastuvõtja, mis ajahetkedel on vaja ülejäänud andmebitid vastu võtta. RS-232C liidese maksimaalne andmeedastuskiirus on 1.5 Mbit/s. Signaali edastamiseks kasutatakse positiivset ja negatiivset pingevahemikku; loogilise nulli edastamise korral peab signaal jääma +5...+12 V ja loogilise ühe korral -5...-12 V vahele. Vastuvõetav signaal peab jääma vastavalt +3...+12 V ja -3...-12 V vahele. Standardi järgi on maksimaalne kaabli pikkus 15 meetrit, või nii pikk, et kaabli mahtuvus ei ületaks 2500 pF. Seega, kui kasutada UTP CAT-5 kaablit, on lubatav kaabli pikkus kuni 50 m. Antud maksimumpikkused kehtivad, kui soovitakse edastada andmeid maksimaalsel lubatud kiirusel. Kui andmeedastuskiirust vähendada 2 või 4 korda, suureneb lubatav juhtme maksimaalne pikkus märgatavalt. Kui kasutada näiteks kiirust 2400 boodi võib juba tavakaabli võimalik maksimaalne pikkus ulatuda 900 meetrini. Üldiselt on RS-232C standardne ning häirekindel liides andmete edastamiseks kahe seadme vahel [14].

RS-232C kasutamise korral on võimalik piirduda ka ainult kolme liiniga. Kolmeliinilise versiooni puhul, kui RS-232C kogu funktsionaalsust ei ole vaja kasutada, kasutatakse eraldi liine andmete

saatmiseks, vastuvõtmiseks ja maa ühenduseks. Sellisel juhul riistvaralist vookontrolli teostada ei saa. Kahe liiniga ühenduse puhul saab luua andmeside juhul, kui liiklus käib ainult ühes suunas (saadab ainult üks seade ja mitte kunagi teine) või kui maa liini vaja ei lähe (nt optilise ühenduse puhul) [25].

Mikrokontroller suhtleb RS-232C liidesega üle UART-i, mis teisendab andmed jada- ja paralleelsesituse vahel.

3.5.4 RS-485

RS-485 ehk EIA-485 on ANSI TIA/EIA standard, mis määrab elektrilise signaali omadused seadmete vaheliseks suhtluseks. RS-485 võimaldab mitmel seadmel (kuni 32) suhelda half-duplex viisil ühte keerupaari kasutades kuni 1200 m vahemaaga. Kasutades kaablit kahe keerupaariga, saab luuda full-duplex ühenduse seadmete vahel. RS-485 võimaldab suure häiretekindlusega andmevahetust, kuna kasutades keerupaari, milles edastatakse täiendsignaale (vastasfaasis signaale), on võimalik vastuvõtjal lihtsamalt eristada signaali müra (vastuvõtja jaoks on kasulik signaal keerupaari juhtmetel vastandfaasis ja mürasignaal enamasti samas faasis). Lisaks on liideses kasutusel veel üks lisajuhe saatja ja vastuvõtja maa ühendamiseks, mille suhtes andmesignaale edastatakse. RS-485 standardit saab kasutada nii mitme kui ka ühe ülemaga režiimis. Kasutades ühe ülemaga suhtlust, saab võrgu ressursi kasutada peaaegu 100%-liselt. Mitme ülemusega protokoll kasutamisel on maksimaalne võrgu kasutamise protsent umbes 37%, kuna siis peab tegelema samaaegsete edastajate tuvastamisega ja põrkeolukordade lahendamise. Olenevalt kasutatud kaabli pikkusest võimaldab RS-485 kasutada maksimaalset andmeedastuskiirust kuni 35Mbit/s. Sarnaselt RS-232C-ga suhtleb kontroller RS-485-ga UART-i vahendusel, seega RS-485 kasutamiseks peab kontrollerile lisama sobiva draiveri [15][26]. Liidese galvaaniliseks eraldamiseks peab kasutama kahte keerupaari kaablit, see suurendab võrreldes RS-232-ga kaablite hulka kaks korda, kuid võimaldab samadele liinidele ühendada mitu alluvseadet.

3.5.5 Ethernet

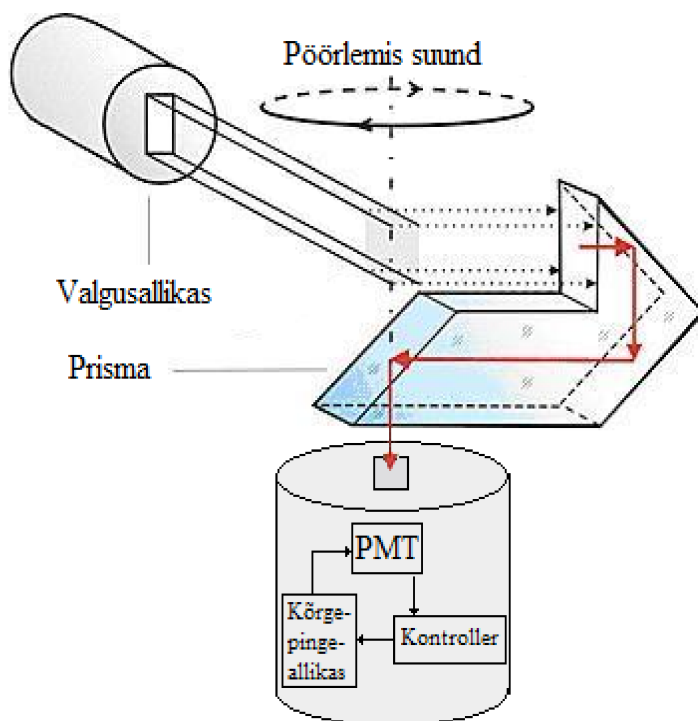
Ethernet on üldnimetus arvutivõrgu tehnoloogiatele andmeedastuseks kohtvõrkudes (LAN – local area network). Ethernet defineerib kaabeldamise ja signaali standardid füüsilise kihi (Physical Layer OSI standardis) jaoks ning üldise adresseerimise formaadi ja erinevad MAC (Medium Access Control) protseduurid lülikihi (Data Link Layer OSI standardis) jaoks. Ethernet on olnud saadaval alates 1980. aastatest, asendades järjest teisi konkureerivaid arvutivõrgu standardeid. Kõige levinum Etherneti kasutamise viis on ühendada seadmeid keerupaari kaabli abil. Selline ühendusviis on osa standardist IEEE 802.3 Ethernet (CSMA/CD). Edaspidi mõtleme mõistet Ethernet kasutades just

seda standardit. Lühend CSMA/CD on lühend inglisekeelsest väljendist „Carrier Sense Multiple Access with Collision Detection“. See tähendab, et ühe ja sama kaabli külge on paralleelselt ühendatud kõik võrgus olevad seadmed. Iga seade võib alata andmeedastust liinil suvalisel ajahetkel, kui ta tuvastab, et liin on vaba. Kui kaks või rohkem seadet üritavad samal ajal andmeid edastada tekib andmete kokkupõrge ning edastamine katkestatakse. Peale kokkupõrget oodatakse juhusliku pikkusega ajavahemik ja seejärel korraldatakse saatmist, kui liin on vaba või oodatakse liini vabanemist. Signaali edastamiseks kasutatakse Manchesteri kodeerimist, et lihtsustada detekteerimist, millal andmeid edastatakse. Tänapäeval on kõige levinum viis kasutada 10Base-T kaablit RJ45 ühendustega. Sellise kaabli maksimaalne lubatud pikkus on 100 m. Ühe füüsilise ühenduse külge võib olla ühendatud kuni 1024 seadet. Standard määrab võimaluse kasutada andmeedastuse kiirust 10, 100 või 1000 Mbit/s. Etherneti kasutamine manusseadmetes sõltub suuresti kontrolleri võimekusest vajalikud toimingud riistvara tasemel realiseerida. Lisaks sellele eeldavad mitmed kontrollid Etherneti kasutamisel, et süsteemis kasutatakse ka reaalaaja operatsioonisüsteemi [7][27].

4 Ruumhajumismõõtja funktsionaalne ja osalise arhitektuuriline disain

4.1 Tööpõhimõte

Üks võimalik ruumhajumise mõõtmise seadme funktsionaalne skeem on toodud joonisel 5. Ruumhajumismõõtja põhielement on sellise seadmes spetsiaalse kujuga hajunud valgust suunav pöörlev periskoopprisma. Prisma pöörlemisteljel paikneb valgust registreeriv fotokordisti. Sellise konstruktsiooni eeliseks on, et valgusallikas ja fotodetektor on mõõtmise ajal fikseeritud asendis. Kasutades sellist prismaga lahendust on võimalik mõõta hajunud valgust kõigi kolme hajumispiirkonna jaoks 0.3° täpsusega (vaata 3.2 Hajumise mõõtmine ja arvutamine). Mõõtmisalgoritmi parameetrid on võimalik määrata teades prisma täpset asendit igal ajahetkel ja teostades lisaks hajunud valguse mõõtmisele otsese valguskiire sumbumise mõõtmist (c, m^{-1}).



Joonis 5: Üks võimalik ruumhajumismõõtja ehituse variant [28].

Kuna sellise konstruktsiooni puhul valgusallikas ja fotodetektor on fikseeritud asendites, siis komponentide vahekauguse suurendamine ei muuda seadet märgatavalt keerukamaks. Samuti lihtsustab selline konstruktsioon valguse detektori ehitust ja võimaldab disainida seadme alamosad väiksemad.

Spektraalse hajumise mõõtmisel on üheks võimaluseks kasutada valgusallikatena erinevates

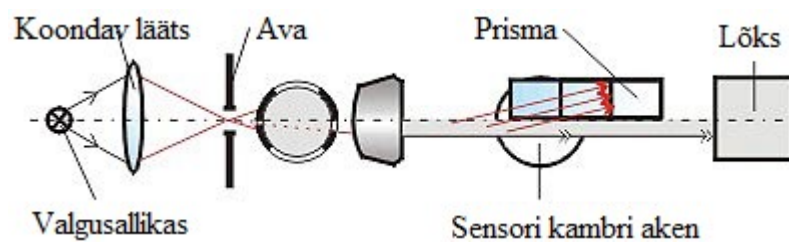
kitsastes lainepikkuse vahemikes valgust emiteerivaid valgusdioode (LED), mille valgusest moodustatakse paralleelne kiirtekimp kasutades sobivat ava ja suure fookuskaugusega (nt 100 mm) läätse. Sel viisil on võimalik moodustada kollimeeritud valguskiir, mille hajumisnurk on väiksem kui 0.1° . See kiir suunatakse läbi akna mõõteruumi, kust hajunud valgus pöördprisma abil fotokordistisse suunatakse.

Kuna tegemist on mõõteseadmega, mis peab töötama ka vee all, siis üks võimalus on paigutada valgusallikas, fotodetektor ja prismaga samm-mootor eraldiasuvatesse hermeetiliselt suletud moodulitesse, mis oleksid kinnitatud ühise raami külge üksteise suhtes 90° nurga all.

Seadme sellise ehituse korral kasutatakse pöörlevat periskoop-prismat hajunud valguse juhtimiseks fotodetektori moodulisse, kus seda läätse abil fokusseeritakse ja fotokordisti abil mõõdetakse.

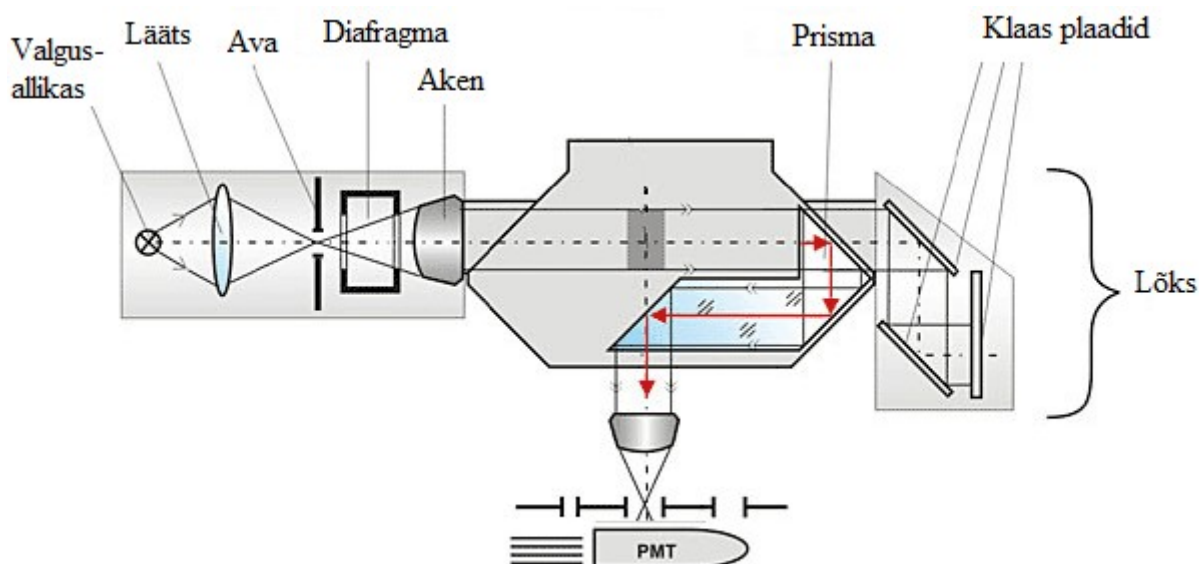
Ruumhajumisfunktsiooni kuju iseärasuse tõttu (vaata 3.2 Hajumise mõõtmine ja arvutamine) peab valgust registreeriva mõõteseadme (fotokordisti) dünaamiline ulatus olema 7 suurusjärku või suurem. Selle saavutamiseks on üks võimalus muuta vastavalt mõõdetava signaali suurusele fotokordisti toitepinget, kasutades eelnevalt kindlaks määratud algoritmi või tagasisidet. Nii on võimalik luua fotokordisti automaatne lineaarne töörežiim, mis tagab suure mõõtmistäpsuse kuni signaali 12 suurusjärgulise muutumise korral [28].

Et laiendada mõõtepiirkonda väikeste mõõtenurkade juures 0.5° -ni, on võimalik kasutada erilist varjumeetodit. Selle realiseerimiseks tuleks kasutada telje suhtes paralleelselt nihutatud periskoopprismat. Nihutatud prisma välisserv paikneb sel juhul täpselt süsteemi optilise teljega paralleelselt vahetult selle kõrval (joonis 6). Sellisel juhul ei kulge otsene valguskiir väikeste hajumisnurkade korral läbi prisma mõõteseadmesse, vaid sellest mööda ja prisma suunab ainult hajunud valguse fotokordistisse vähendades otsese valguse sattumist fotokordistisse mitme suurusjärgu võrra. Ideaalsetes tingimustes oleks võimalik vältida otsese valguse sattumist fotokordistisse täielikult. Lisaks prisma nihutamisele teljel on vajalik vähendada väikeste mõõtenurkade korral valguskiire diameetrit. Kõige lihtsam on seda teostada kasutades erinevate mõõtterežiimide korral valguskiire tee paigutatavaid erineva suurusega avasid, mida vahetatakse samm-mootori abil. Vastavalt valitud mõõtepiirkonnale muudetakse ka ava nii, et 0° kuni 30° ja 160° kuni 180° juures on ava minimaalne, kuid vahepealsete mõõtmisnurkade korral on ava maksimaalne, et suurendada hajunud valguse hulka fotodetektori sisendis.



Joonis 6: Võimalik valguse hajumise mõõtja optiline skeem (vaade ülevalt) [28].

Et võimaldada täpseid mõõtmisi 180° lähedaste mõõtenurkade korral, on optilisse skeemi lisatud spetsiaalne seadme mõõteruumi tagaseinalt otsese valguse tagasipeegeldumist vältiv valguse lõks (joonis 7), mille eesmärk on vältida vees paiknevatest lisanditest ja veeosakestest hajunud mõõdetava valguse segunemist seadme seinalt tagasihajunud valgusega. Sellise konstruktsiooniga on võimalik vähendada tagasihajunud valguse hulka rohkem kui 10^9 korda [28].



Joonis 7: Võimalik seadme optiline skeem (vaade küljelt) [28].

Seadme mehaanilist konstruktsiooni on plaanis täiustada. Seega ei ole MVSM täpne ehitus teada ning sellest tulenevalt võib ka tööpõhimõtte mõningal määral muutuda.

4.2 Riistvara ja tarkvara partitsioneerimine

MVSM siseselt hajumise mõõtmisel teostatavad toimingud ja mõõtmiste teostamiseks vajalikud seadmed on võimalik jagada mitmeks alamosaks. Eraldiseisvateks riistvara komponentideks võiks olla:

- prisma ja selle liigutamise mootor,
- valgusallikas või LED-de kasutamisel valgusallikad koos nende vahetamise mootoriga,
- valgusfiltrid koos nende vahetamise mootoriga, kui LED-ide asemel kasutada halogeenlampi,
- ava muutmise komponendid,
- veepump ja
- valguse mõõtja.

Nende komponentide juhtimist on võimalik sooritada mitmel erineval viisil.

Alamülesanded, mida on vaja teostada, on järgmised:

- valgusvoo juhtimine – kasutatakse sisemist valgusallikat (LED-id) valgusvoo tekitamiseks. Valgusvoo juhtimisel peab kontrolleri suutma muuta valguse lainepikkust (vahetades LED-i), muuta kiire diameetrit (vahetades kiire ees paiknevat ava) ja reguleerida mõõtesensori ees olevat ava fotokordistisse jõudva valguse hulga muutmiseks. Valitavaid lainepikkusi on planeeritud kuni 12 ja iga lainepikkusega valguse emiteerimiseks on kasutusel erinev LED. Lainepikkuse seadmiseks tuleb positsioneerida sobiv LED õigesse kohta ja juhtida LED-i läbivat voolu sobiva valgusvoo tekitamiseks. Valguskiire ees paikneval aval peaks olema vähemalt kaks muudetavat suurust kiire diameetri muutmiseks erinevate mõõtepiirkondade jaoks. Fotokordisti ees oleval aval peaks olema vähemalt kolm võimalikku olekut – suletud; väike ava valgushulga vähendamiseks ja suur ava vähese hajunud valguse korral teostatavateks mõõtmisteks.
- prisma liigutamine – prisma liigutamiseks kasutatakse samm-mootorit. Juhul, kui mõistliku suurusega samm-mootor ei jõua prisma vees liigutada, on vaja lisada spetsiaalne väikese loksuga reduktor. Kontrolleri peab olema suuteline määrama prisma asendit ja pöörama seda etteantud positsioonile. Asendi määramise andur peab paiknema samm-mootoriga samas seadme hermeetilises osas, et vältida anduri kokkupuudet veega. Hajumiskoeffitsiendi piisava ruumilise lahutuse saamiseks peab minimaalne prisma nurga muutmise võimalus

olema umbes kolmandik kraadi. Võimaluse korral tuleks kasutada valmis komponenti samm-moorori juhtimiseks. Sobiva reduktoriga ja piisava võimsusega mootori kasutamisel on võimalik prisma asendi määramisel piirduda ainult ühe fikseeritud asendi määramisega; kõige sobivam oleks kasutada prisma nullpositsioonile või prisma 180° positsioonile vastavat asendiandurit.

- valguse mõõtja – kuna mõõdetava signaali dünaamiline ulatus on väga suur (vaata 3.2 Hajumise mõõtmine ja arvutamine), siis sobiv seade hajunud valguse registreerimiseks on fotokordisti (PMT - *photomultiplier tube*). Kuna mõõdetava valguse intensiivsus varieerub väga laiades piirides, siis on vajalik muuta PMT toitepinget töö ajal fotokordisti tundlikkuse muutmiseks ja lisaks on vaja muuta fotokordistisse sisenevat valguse hulka mehaaniliselt, kasutades erineva suurusega avasid. Lisaks peab olema võimalus blokeerida mehaaniliselt täielikult valguse sisenemine fotokordistisse pimevoolu mõõtmiste teostamiseks. Pimevoolu väärtust tuleks kasutada mõõdetava signaali korrigeerimiseks. Saavutamaks võimalikult täpset tulemust võib olla vajalik mõõteseadme jahutamine. Jahutamisprotsessi juhtimiseks peab olema võimalus mõõta PMT temperatuuri.
- andmete salvestamine – kui seadet kasutatakse välimõõtmistel ilma välise juhtarvutita või piiratud võimalustega välise juhtseadmega, siis on vajalik mõõtetulemused salvestada. Selleks on sobiv kasutada mälukaarti, millele salvestamine ja kirjutamine oleks lihtsalt teostatav ning mis oleks töökindel ja piisavalt suure mahutavusega. Standardne SD tüüpi mälukaart kasutab oma failisüsteemi, millele on võimalik juurdepääs SPI liidese abil.
- suhtlemine kasutajaga – MVSM peab olema võimeline suhtlema kasutajaga: võtma vastu kasutaja käske ja tagastama küsitud andmeid. Kasutajaga suhtlemine võiks olla mitmetasandiline – alates võimalusest teostada kogu hajumise mõõtmise töötsüklil ühe nupuvajutuse peale (või lihtsa käsu peale) kuni keeruliste võimalusteni juhtida välise kontrollseadme abil paindlikult kõiki mõõtmisalgoritmi osi. Lisaks peab kasutajaliides võimaldama kontrollida seadme alamosade seisundit, et avastada võimalikke tõrkeid.
- suhtlemine komponentidega – seadme juhtkontroller peab olema võimeline suhtlema kõigi oma alamkomponentidega.
- arvutamine – mõõtmiste tulemuste põhjal tuleb arvutada välja hajumiskoeffitsiendid.

4.3 Ülesannete võimalik jaotamine seadmesisese ja välise arvuti vahel

Seadme juhtimist on võimalik realiseerida mitmel erineval viisil. MVSM võib olla nii autonoomne mõõteriist kui ka välise arvuti abil juhitud. Lisaks võiks olla lihtne väline juhtseade halbades ilmastikutingimustes välimõõtmiste teostamiseks, kus standardse arvuti kasutamine on raskendatud – soolase vee pritsmed, mõõtmised ereda valguse käes, mõõtmine talvisel ajal jms.

Autonoomse seadmena teostab MVSM ise andmete kogumist ja töötlemise. Seade peab olema piisavalt suure jõudlusega, et suudaks juhtida mõõtmisprotsessi ja samaaegselt teostada kogu vajaminev andmetöötlus. Lisaks peab MVSM suutma salvestada kõik vajalikud andmed ja tulema toime kõigi võimalike veaolukordadega välise sekkumiseta. Autonoomse süsteemi realiseerimine on kõige keerulisem. Lahenduse positiivseks küljeks on kasutaja seisukohalt seadme lihtsam ja universaalsem kasutamine, sest mõõtmise teostamiseks ei ole vaja pidevat ühendust arvutiga ning ei ole nõutav kasutajapoolne sekkumine.

Välise arvuti abil juhitava seadme korral toimub andmetöötlus ja mõõteprotsessi juhtimine personaalarvutis, kusjuures MVSM ise töötab lihtsa arvutijuhitava mõõteseadmena. Sellisel juhul on vaja kiiret ja stabiilset andmesidekanalit. MVSM ülesandeks oleks täita arvuti poolt saadud korraldusi, nagu näiteks prisma liigutamine soovitud asendisse, valguse seadmine etteantud režiimi, PMT väljundi mõõtmine jne. Protsessi juhtimine ning andmete töötlus ja salvestamine toimuks aga välises arvutis. Nii on seadme sisene tööalgoritm märkimisväärselt lihtsam, kuna on vaja teostada ainult lihtsamaid toiminguid. Samuti ei tekitaks probleemi arvutuste keerukus ja sellele kuluv aeg, kuna võrreldes mikrokontrolleriga on personaalarvuti jõudlus väga suur. Positiivseks omaduseks võib lugeda ka kiiremat ja täpsemat tagasisidet mõõtmisprotsessi käigust - kui tekib ettenägematu olukord või on vaja muud kasutajapoolset sekkumist, saab kasutaja kiiresti reageerida ja probleemide olemasolu ei selgu alles hiljem andmeid töödeldes. Negatiivseks omaduseks võib kujuneda mõõteprotsessi aeglus, kuna mõõtmistsükli jooksul on välise arvuti ja MVSM-i vahel vaja vahetada palju andmeid.

Hübriid-süsteemi korral oleks võimalik kasutada ära nii autonoomse seadme mõõtekiirus kui universaalarvuti arvutuskiirus, jättes mõõtmise juhtimine MVSM enda hooleks, aga vajalike hajumiskoeffitsiendi arvutuste teostamise universaalarvutile. Andmete töötlemine ja talletamine jääks sel juhul märksa võimsama arvuti hooleks. Ka selline lahendus eeldab välise arvuti olemasolu ning selle pidevat ühenduses olemist MVSM-iga. Positiivseks küljeks on võimalus saada operatiivset tagasisidet, kui mõõtmisprotsessis tekib häireid või kui tuleks mõõtmisi mingil põhjusel koheselt korrata.

Autonoomne seade kaugjuhtimisega – mõõteseade ise on autonoomne ja saab kõige vajalikuga iseseisvalt hakkama. Kaugjuhtimisseade on mõõtmisprotsessi alustamiseks ning põhiliste mõõtmistulemuste ja teadete kuvamiseks. Selline variant on kõige sobilikum välitöödeks. Kasutajal ei ole mõõtmiste läbiviimiseks vaja personaalarvutit, vaid talle piisab lihtsast juhtseadmest, mis suudab edastada MVSM-le vajalikud käsud ning kuvada esmast tagasisidet. Nii omab kasutaja ülevaadet toimuvast mõõteprotsessist ja tal on selle juhtimise võimalus.

Antud seadme kasutust silmas pidades on kõige sobivam variant autonoomne seade kaugjuhtimisega, kuna selline seade sobib kõige paremini tööks välitingimustes. Tuleviku perspektiivis kaalutakse spetsiaalse juhtpuldi loomist välitingimustes töötamiseks. Seega edaspidi võtame aluseks, et seade peab iseseisvalt toime tulema mõõteprotsessiga ning tema juhtimine juhtseadme või välise arvuti abil peab olema võimalikult lihtne.

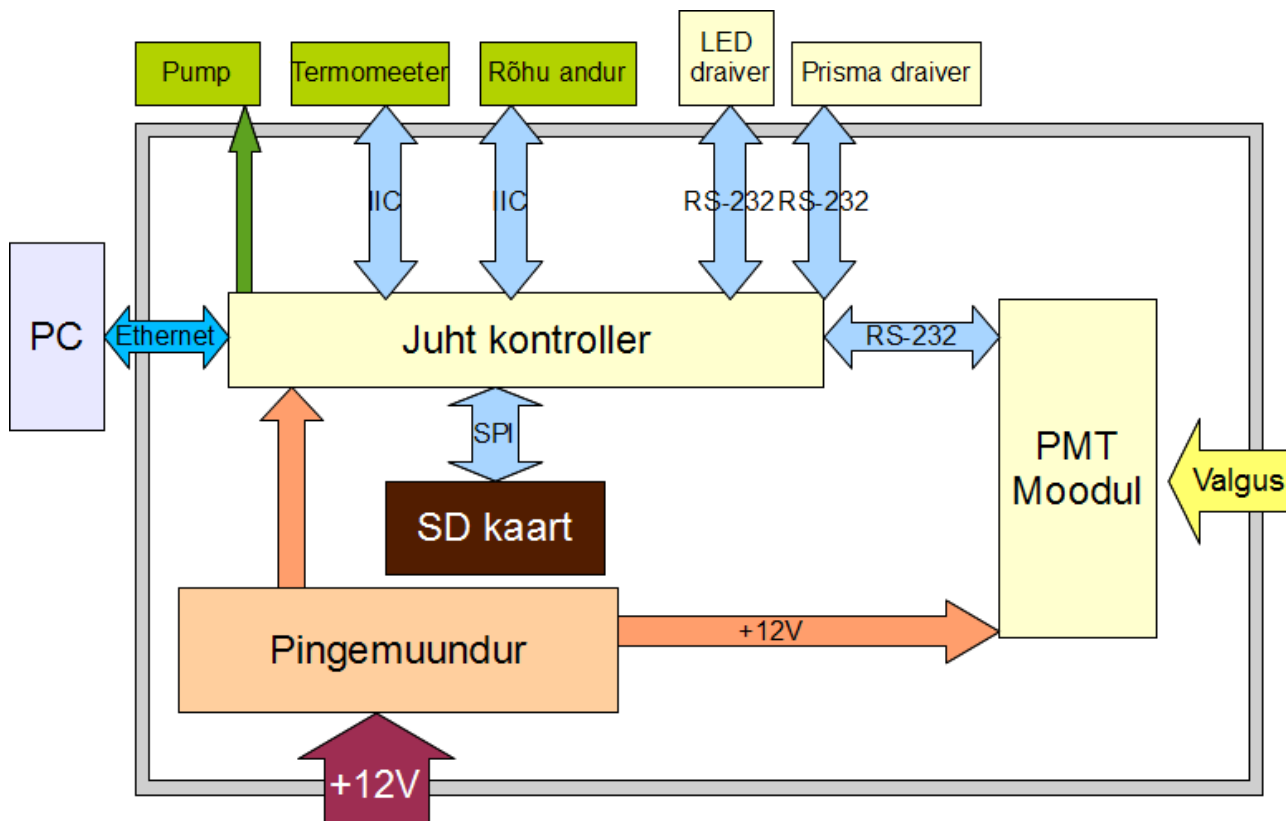
4.4 MVSM juhtimise partitsioneerimine

MVSM töö juhtimiseks võiks kasutada kahte kontrolleri. Põhjuseks see, et hinnanguliselt on ühe kontrolleri jaoks ülesandeid liiga palju ning programmi koostamine kujuneks pigem keeruliseks ja rohkemate kontrolleri puhul hakkaks komponentide omavaheline suhtlus nõudma liiga suurt osa tööle kuluvast ajast. Lisaks kahele põhikontrollerile oleks mõistlik kasutada mootorite juhtimiseks täiendavaid kontrollereid koos mootori draiveritega, et vähendada juhtmete hulka moodulite vahel, kuna komponendid paiknevad eraldi konteinerites.

Nende kahe kontrolleri vahel võib töö jaotada kahel viisil:

1. Põhikontroller komponentide juhtimiseks, andmete salvestamiseks ning kasutajaga suhtlemiseks ja mõõtekontroller PMT kasutamiseks. Sellise tööjaotuse plokk skeem on kujutatud joonisel 8.
2. Mõõtekontroller on mõeldud mõõtmisprotsessi läbi viimiseks ja põhikontroller kasutajaga suhtlemiseks, andmete töötlemiseks ning salvestamiseks. Sellise tööjaotuse plokk skeem on kujutatud joonisel 9.

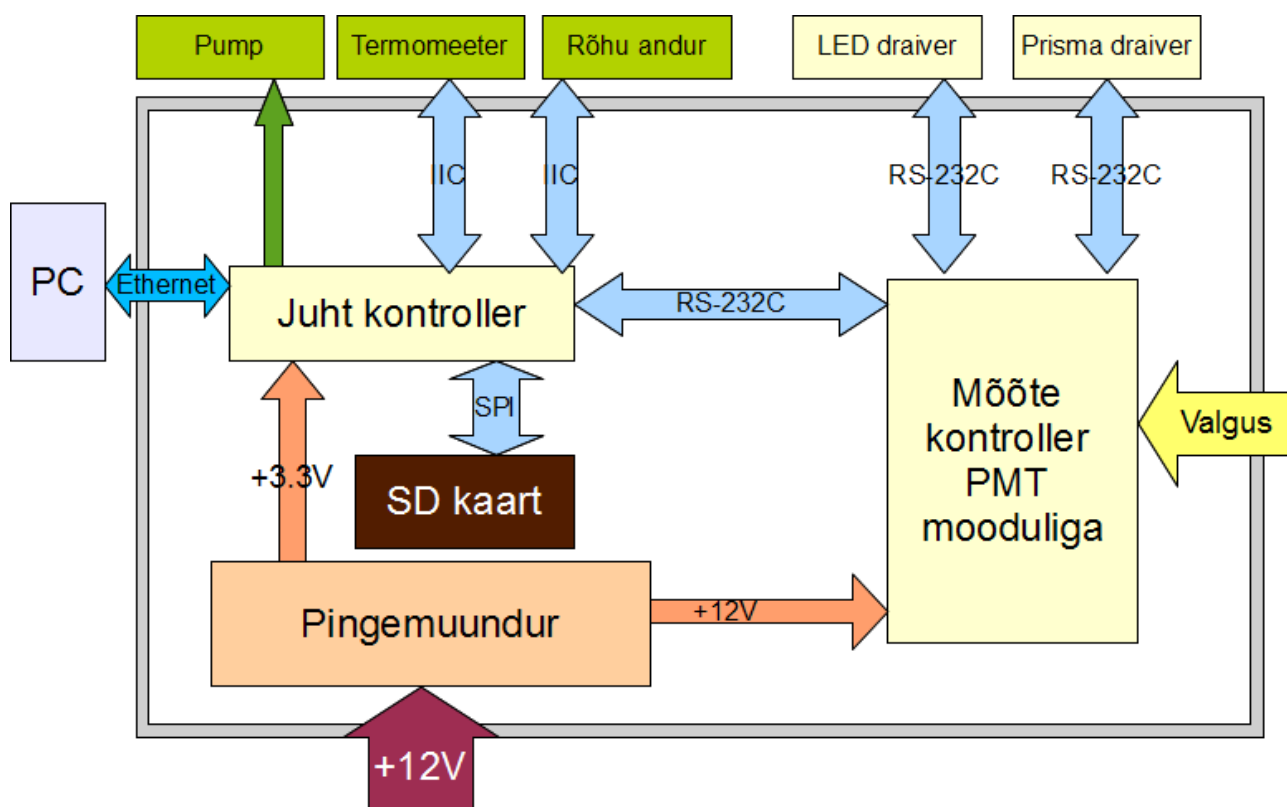
Esimese partitsioneerimise versiooni kasutamine on sobiv juhul, kui ühe kontrolleri jaoks on kogu mõõteprotsessi läbiviimine liialt keerukas. Sellisel juhul jääb mõõtmisest kõige keerukam osa, ehk



Joonis 8: Juhtimisloogika plokk skeem, kui põhikontroller juhib kogu süsteemi tööd ja teostab hajumiskoeffitsiendi arvutused ning mõõtekontroller juhib ainult PMT kasutamist

PMT kasutamine, eraldi kontrolleri tööks. Eraldiseisvate komponentidena töötavad LED draiver ja prisma draiver on juhitavad juhtkontrolleri poolt. Miinuseks võib kujuneda asjaolu, et juhtkontrollerile jääb liiga palju ülesandeid ning erinevate seadmete juhtimiseks ja arvutuste teostamiseks kuluv aeg venitab mõõtmisprotsessi. Selle variandi korral võib positiivse omadusena välja tuua PMT mooduli eraldatuse. Nii on võimalik eraldiseisvalt PMT moodulit, kui kõige keerukamat ja olulisemat komponenti, hiljem vahetada või täiendada ülejäänud seadme ehitust muutmata.

Teise variandi eeliseks on see, et kõik mõõtmisega seotud tegevused teostatakse ühes kontrollerris, mõõtekontrolleris. Juhtkontrolleri ülesandeks on suhtlemine kasutajaga, andmete töötlus ning salvestamine. Juhtkontrolleril jääb rohkem aega arvutuste tegemiseks ja kasutajaga andmete vahetamiseks. Negatiivseks küljeks on mõõtekontrolleri võimalik ülekoormamine. PMT kasutamine on küllaltki keerukas toiming ning teiste seadmetega suhtlemine tekitab lisakeerukust.



Joonis 9: Juhtimisloogika, kui põhikontroller suhtleb kasutajaga ja teostab hajumiskoeffitsiendi arvutused ning mõõtekontroller juhib lisaks PMT-le veel valgust ja prisma mootorit.

4.5 Kontrolleri ülesanded

Juhtimise partitsioneerimisest on tõenäoliselt otstarbekam kasutada esimest varianti (vaata 4.4 MVSM juhtimise partitsioneerimine), kuna sel juhul on PMT ja tema juhtimine eraldatud muust süsteemist ning mõõtekontrollerile jääb suurem jõudlusvaru vajalikeks toiminguteks, samuti on võimalik PMT juhtimise moderniseerimine süsteemi muid osi muutmata.

4.5.1 Juhtkontroller

Juhtkontroller peab täitma mitut olulist ülesannet: kasutajaga suhtlemine, hajumise mõõtmisprotsessi juhtimine, andmete töötlemine, kasutaja käskude interpreteerimine ja täitmine ning andmete salvestamine (joonis 8):

Kasutajapoolsed käsud mida kontroller peaks teenindama võiksid olla;

- hajumiskoeffitsiendi mõõtmine. Kasutajalt saadud korralduse peale viiakse läbi kogu hajumiskoeffitsiendi mõõtmisprotsess.
- Pumba sisse- ja välja lülitamine. Kasutaja käsu peale lülitatakse mõõtekambri pump sisse või välja.
- Hajumiskoeffitsientide arvutamine. Viimaste mõõtmistulemuste järgi arvutatakse hajumiskoeffitsiendid ja edastatakse nende väärtused kasutajale.
- Prisma liigutamine. Liigutab prisma etteantud asendisse. Peamiselt mõeldud prisma viimiseks algasendisse.
- Viimaste mõõteandmete küsimine. Kasutaja käsu peale edastatakse kasutajale viimase mõõtettsükli ajal mõõdetu andmed.
- Temperatuuri ja rõhu küsimine. Edastatakse kasutajale rõhu- ja temperatuurianduri näidud.

Kõige tähtsama ülesande, mõõtmisprotsessi läbiviimise kohta, on toodud koostööskeem joonisel 14. Koostöö organiseerimine on juhtkontrolleri ülesanne, nagu ka üldise mõõtmisprotsessi juhtimine, mille kohta on järgnevusskeem joonisel 12.

Juhtkontrolleriga on SPI liidese abil ühendatud mälukaart, kus hoitakse mõõtmistulemusi. Juhtkontroller kasutab mälukaardile sobivat failisüsteemi andmete salvestamiseks ja küsimiseks.

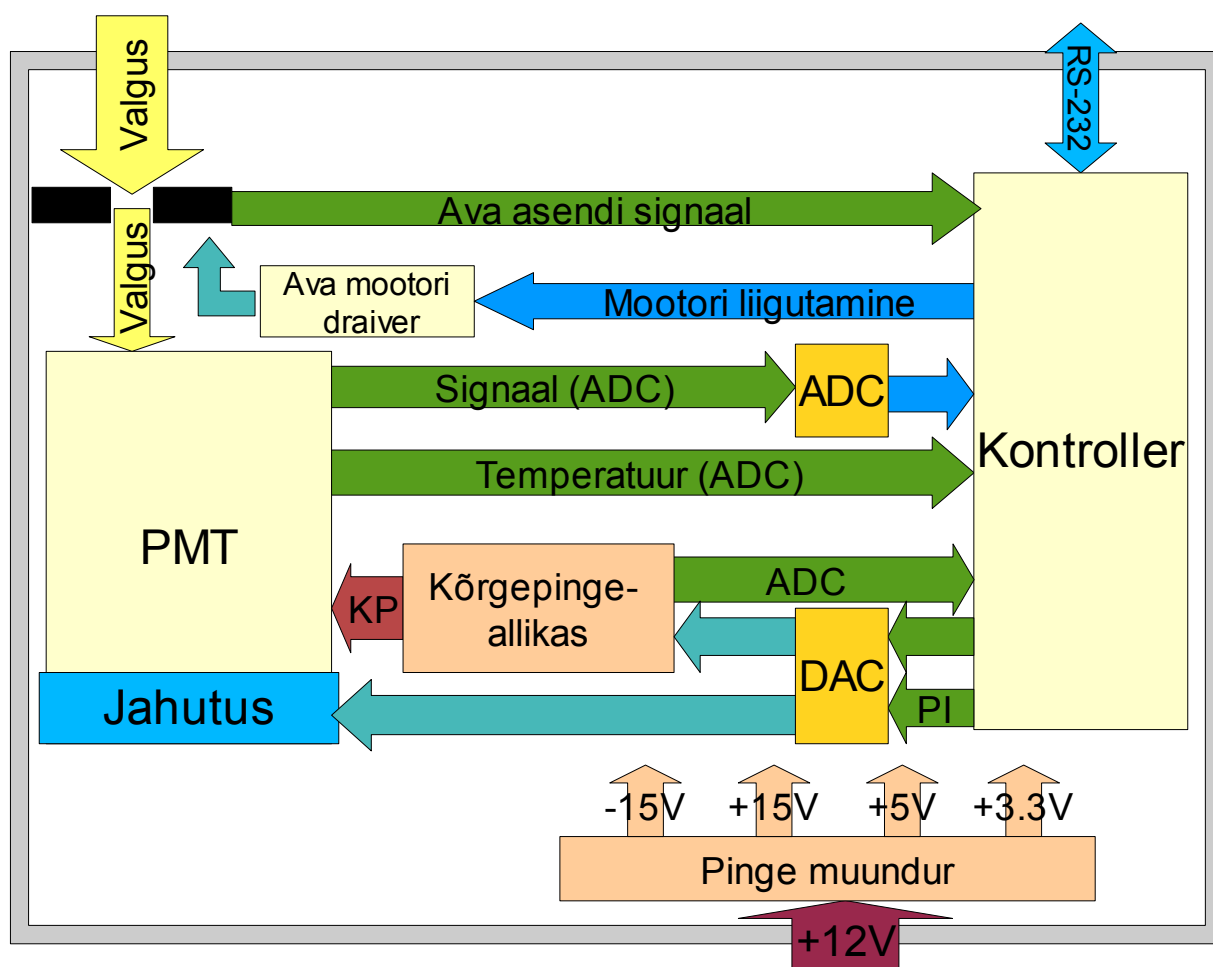
Otseselt juhtkontrolleri juhtida on pump, rõhuandur ja termomeeter.

Kontrollerist elektriliselt eraldatud PMT moodul, LED draiver ja prisma draiver on juhitavad

suhtades RS-232C liidest. Nendega suhtlemine ja nende juhtimine on samuti otseselt mikrokontrolleri ülesanne.

4.5.2 Mõõtekontroller

Mõõtekontrolleri peamine ülesanne on PMT juhtimine ja valguse mõõtmine. Joonisel 10 on toodud PMT mooduli plokk skeem. Kontroller peab juhtima PMT sisendava suurust, PMT jahutust ja kõrgepinget ning mõõtma PMT väljundvoolu ja temperatuuri. Lisaks tuleb mõõtekontrolleril suhelda juhtkontrolleriga: alustada juhtkontrolleri edastatud käsu peale mõõtmist ja tagastada pärast mõõtmise teostamist tulemus. PMT ees paikneva valguse hulka reguleeriva ava muutmine toimub samm-mootoriga. Kontroller muudab vastavalt vajadusele ava suuruse üheks kolmest võimalikust suurusest: suur ava, väike ava, suletud ava. Ava asendi kontrollimiseks (nt algseadistamisel) on vajalik asukohaandur, mis annab signaali, kui ava juhtimismootor on algasendis.



Joonis 10: PMT mooduli plokskeem

PMT kõrgepinget ja jahutust juhitakse digitaal-analoog muundit (DAC) kasutades. Kontrolleri arvutab sobiva kõrgepinge tekitamiseks vajamineva pinge väärtuse, mis edastatakse DAC-ile, mis

omakorda seab sobiva väljundväärtuse. PMT toitepinge seadmisel kontrollitakse ADC abil tegelikku kõrgepinge väärtust, et seda oleks võimalik hiljem arvutustes kasutada.

PMT temperatuuri ja väljundvoolu mõõtmiseks kasutatakse eraldiseisvat analoog-digitaal muundit (ADC), et tagada suurem lahutusvõime (hinnanguliselt on vajalik minimaalselt 16-bitti). ADC juhtimine ja andmete töötlemine on samuti mõõtekontrolleri ülesanne.

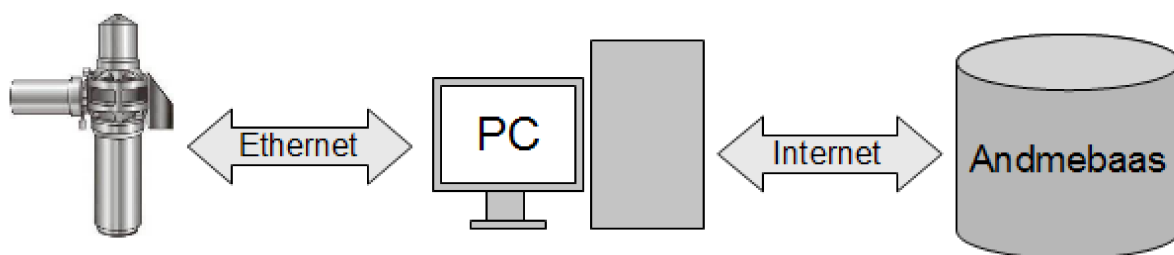
Mõõtekontrolleri põhifunktsioon mõõteprotsessis oleks seega mõõteruumis hajunud valguse mõõtmine. Selle ülesande täitmise järgnevusskeem on kujutatud joonisel 15.

4.6 Juhtarvuti ülesanded

Juhtarvuti peamine ülesanne on võimaldada kasutajal suhelda mõõteseadmega ning edastada kasutajale infot mõõteprotsessi kulgemisest ja tulemustest. Samuti võimaldab juhtarvuti laadida mõõtmisandmeid mälukaardilt, kui mõõtmisi on teostatud kasutades välimõõtmiste lihtsat juhtkontrollerit.

Mõõteseadmega suhtlemine peab olema kasutaja jaoks võimalikult lihtne. Mõõtmisprotseduuri läbiviimine peaks olema selline, et ei oleks vaja kasutada suure jõudlusega arvutit, ega eriväljaõppega kasutajat. Samuti peaks vähest jõudlust nõudma seadme juhtimine, kuna veekogule kaasavõetavate seadmete valik on reeglina piiratud. Kasutajalt ei tohi nõuda teadmisi elektroonika ja tarkvara valdkonnast, vaid kogu sisemine juhtimisloogika peab olema peidetud kasutajaliidese taha.

Mõõtetulemuste operatiivseks analüüsimiseks peab juhtarvutis kasutajaliides suutma kuvada arvutuse tulemusena saadud hajumiskoeffitsiendid, kasutatud lainepikkuse, temperatuuri ja sügavuse väärtused ning vajadusel kuvama need graafikul.

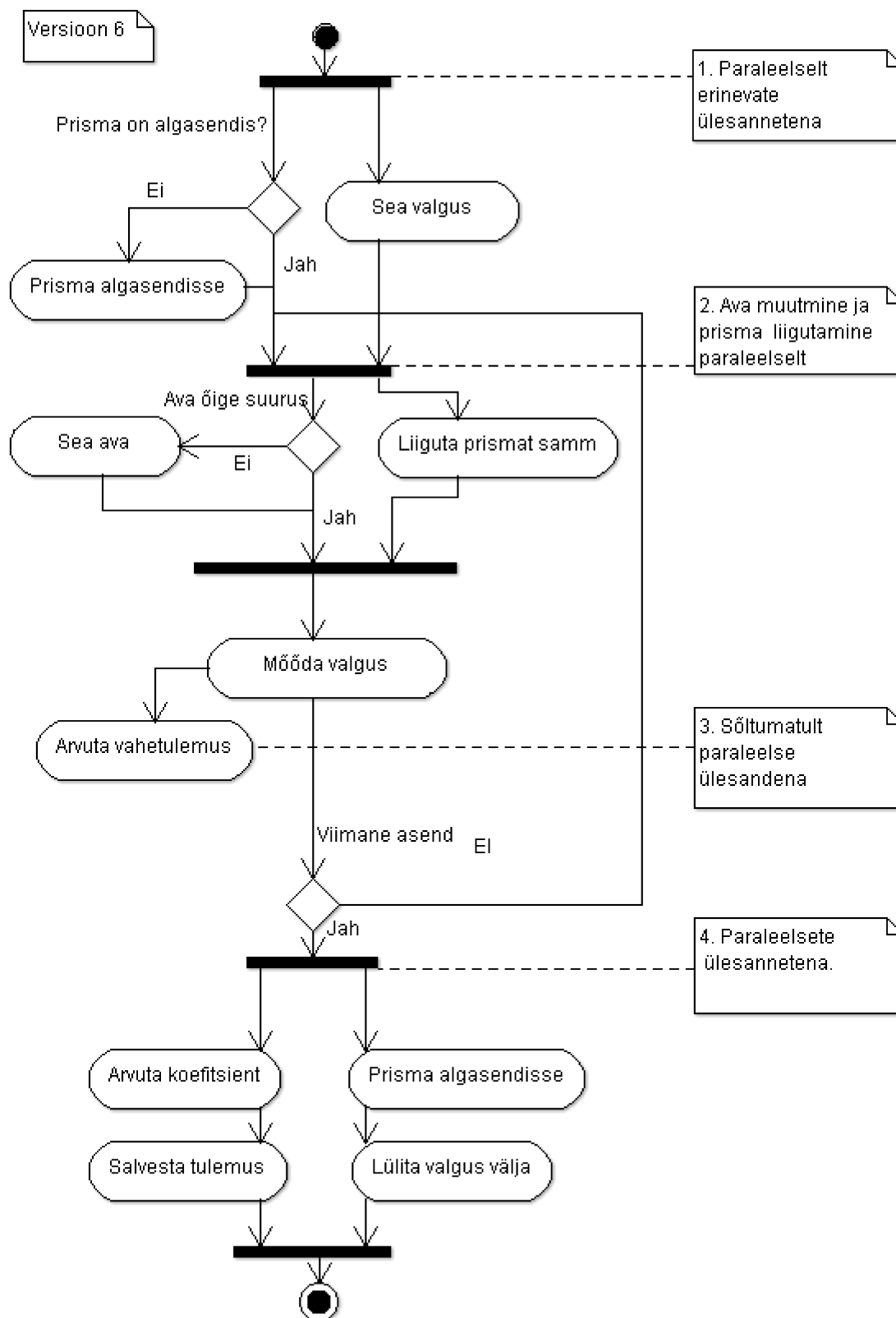


Joonis 11: Mõõteseadme kasutamine välise juhtarvutiga ja mõõtmisandmete hoidmine andmebaasis.

Andmete pikaajaliseks hoidmiseks ja kasutamiseks luuakse veebipõhine andmebaas, kuhu saab salvestada mõõtmistulemused koos lisaandmetega selle kohta, millises olukorras ja kus mõõtmine teostati. Seadmest andmete andmebaasi salvestamiseks võiks juhtarvuti kasutajaliideses olla sobiv funktsionaalsus. Joonisel 11 on kujutatud skeem, kuidas andmed seadmest andmebaasi salvestada.

4.7 Kontrolleri tööskeem (UML)

Mõõtmisprotsessi töö järgnevusskeem on toodud joonisel 12. Mõõtmisprotsess algab mõõtmiseks sobiva valguse seadmisega ja prisma algasendisse liigutamisega. Mõlemat toimingut teostavad



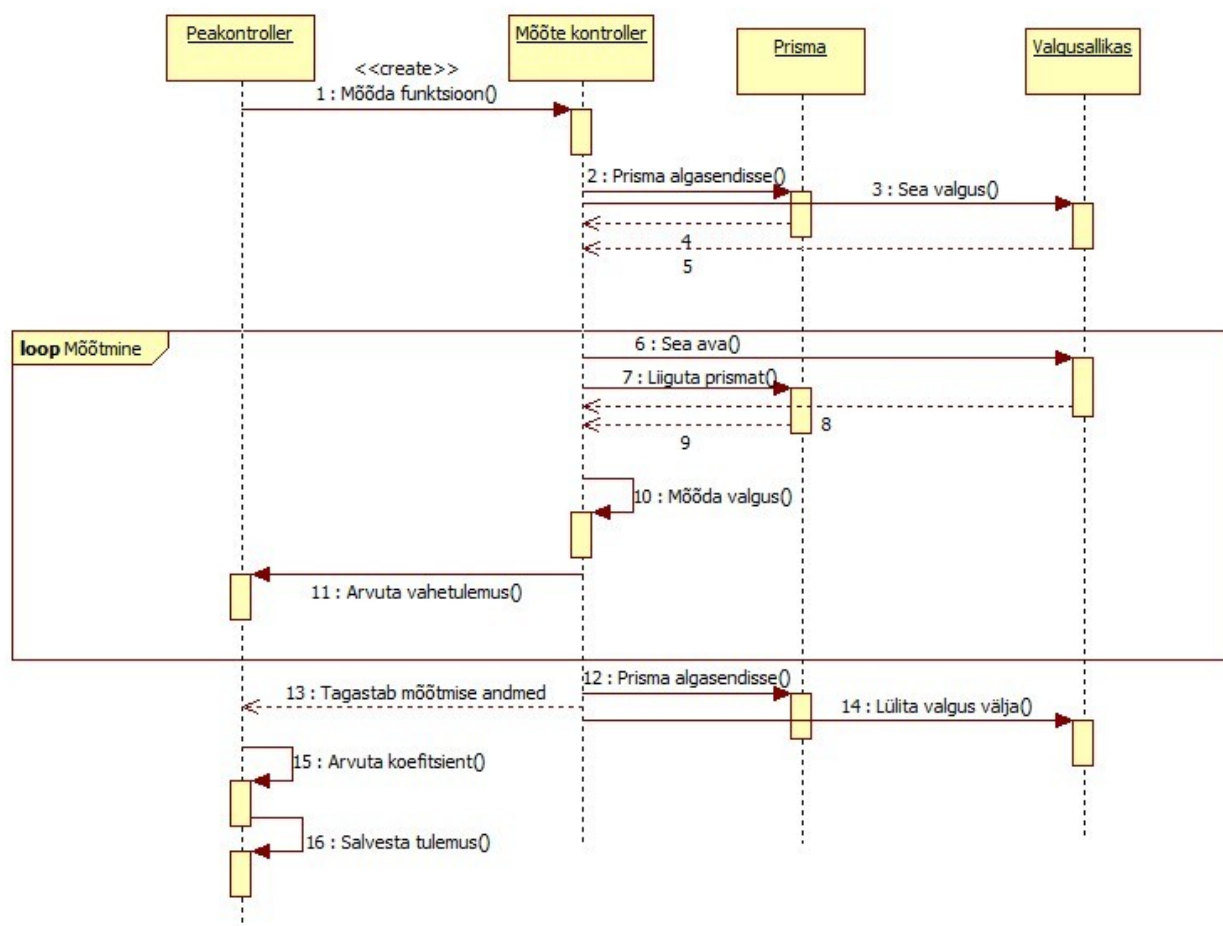
Joonis 12: Mõõtmiste järgnevusskeem.

erinevad füüsilised komponendid, seega nende teostamine võib toimuda paralleelselt.

Edasi hakatakse tsüklliliselt teostama valguse mõõtmist. Tsükkel koosneb mitmest vajalikust toimingust. Esmalt liigutatakse prisma sammu võrra edasi ja vajadusel korrigeeritakse valgusallika ava. Nende kahe toimingu teostamine toimub samuti paralleelselt. Järgneb PMT abil valguse mõõtmine (valguse mõõtmist on täpsemalt kirjeldatud peatüki lõpus). Pärast hajunud valguse mõõtmist arvutatakse vajalikud vahetulemused. Tsükli korratakse kuni mõõtmised on teostatud kogu 180 kraadi ulatuses.

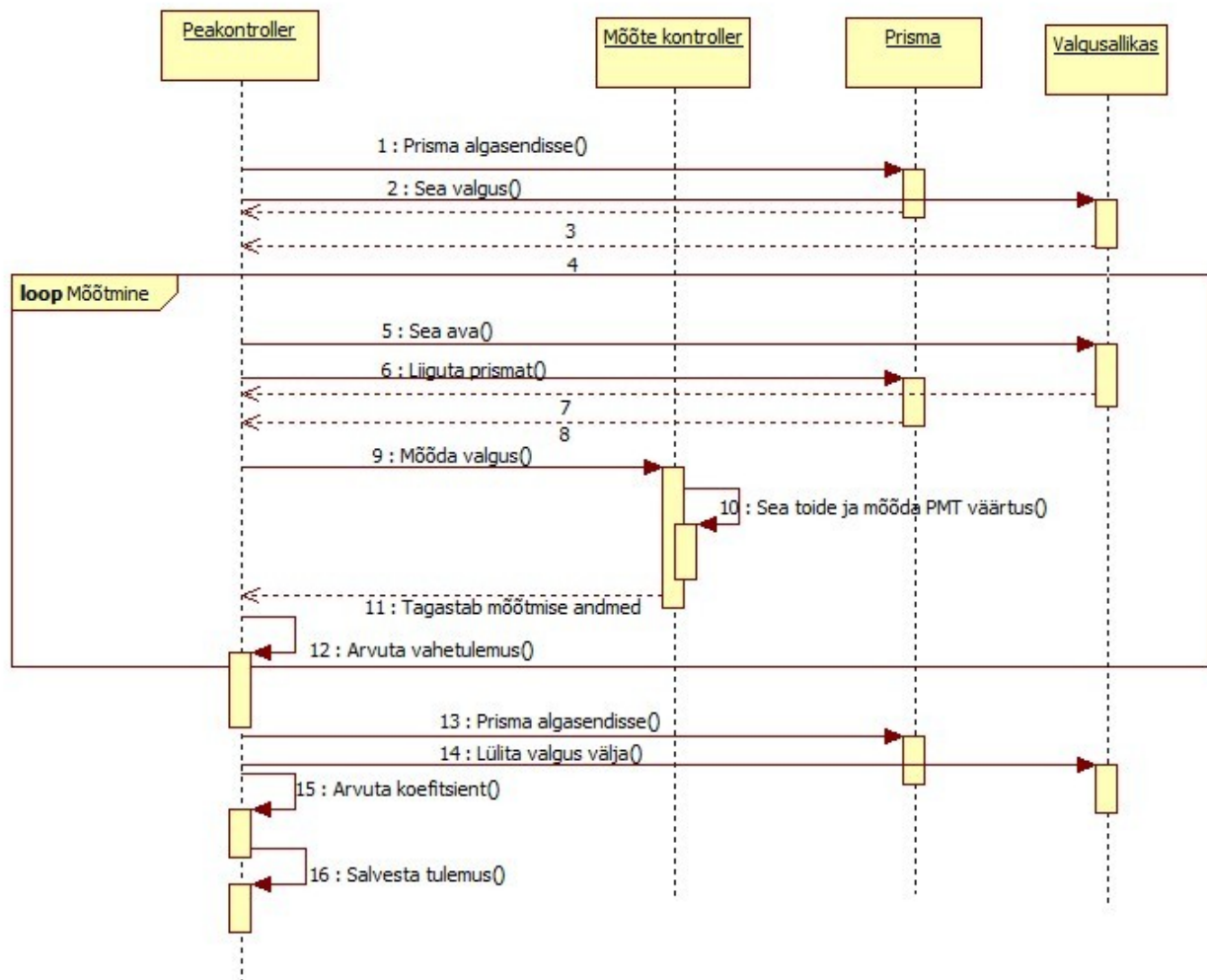
Mõõtmistsükli lõpus arvutatakse vahetulemusi kasutades hajumiskoeffitsiendi ja tagasihajumiskoeffitsiendi väärtus ning salvestatakse need koos mõõtmise vahetulemustega. Peale mõõtmistoimingute lõppu seatakse prisma algasendisse ja lülitatakse valgus välja.

Antud seadme puhul on kõige otstarbekam kasutada ühte kahest järgnevast koostööskeemist. Esimene neist (joonis 13) kujutab tööjaotust juhul, kui PMT abil valguse mõõtmine on lihtne ja kontrolleri suudab teostada ka muid toiminguid. Selline koostööskeem vastab ehituse plokkiskeemile, mis on kujutatud joonisel 9.



Joonis 13: Koostööskeem seadme erinevate osade jaoks mõõteprotsessi korral, kui mõõtekontroller juhib lisaks PMT-le ka prisma ja valgusallika kontrolleri.

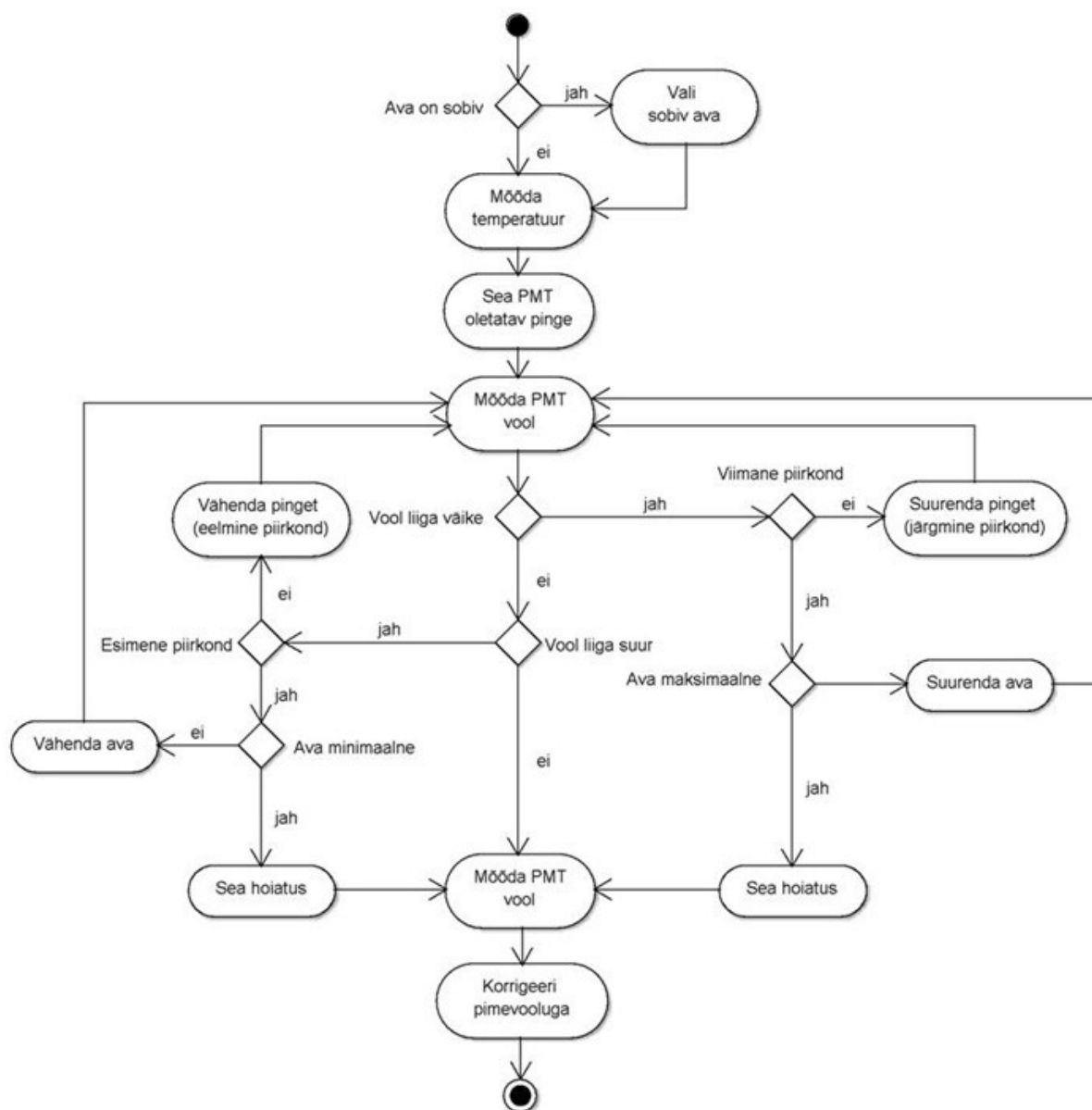
Teine koostööskeem (joonis 14) kujutab tööjaotust juhul, kui PMT kasutamine on ressursimahukas ning teiste komponentidega suhtlemine jääb peakontrolleri ülesandeks. Selline koostööskeem vastab seadme ehituse plokkiskeemile, mis on kujutatud joonisel 8. Erinevalt eelmisest koostööskeemist (joonis 13) tegeleb antud juhul erinevate seadme osadega suhtlemisega juhtkontroller.



Joonis 14: Koostööskeem seadme erinevate osade jaoks mõõteprotsessi korral, kui mõõtekontrolleri ülesandeks on ainult PMT juhtimine ning prisma ja valgusallika kontrollit juhib peakontroller.

Joonisel 15 on kujutatud mõõtekontrolleri töö järgnevusskeemi, mis kirjeldab täpsemalt valguse mõõtmist PMT abil. Antud algoritmi täidetakse iga kord, kui teostatakse valguse mõõtmine. Mõõtmisprotsess algab avaseadmisega. Vastavalt prisma nurgale ja eelmisele mõõtmisele, valitakse sobiv ava. Järgmise sammuna mõõdetakse PMT temperatuur ja arvutatakse ning seatakse eeldatav PMT kõrgepinge. Edasi hakatakse mõõtma hajunud valgust. Selleks mõõdetakse tsükliliselt PMT signaali väärtus, hinnatakse tulemust ja korrigeeritakse mõõtmise tingimusi (ava ja kõrgepinge), kuni jõutakse sobivates tingimustes mõõdetud tulemusteni või mõõteparameetrite ääritingimusteni, mille korral enam korrigeerimist teha ei ole võimalik. Hindamisel ja korrigeerimisel suurendatakse

PMT kõrgepinget, kui mõõdetav väljundpinge jääb liiga väikseks. Kui enam ei ole võimalik kõrgepinget suurendada, suurendatakse fotokordisti ees olevat ava. Kui aga mõõdetud pinge oli liiga suur vähendatakse PMT kõrgepinget ja kui seda enam vähendada ei saa, vähendatakse ava suurust. PMT kõrgepinge muutmine käib astmete kaupa ja aval on vähemalt kaks fikseeritud suurust. Peale mõõtmise sooritamist korrigeeritakse mõõtetulemust vastavalt pimevoolule ning edastatakse see peakontrollerile. Juhul, kui mõõtmine sooritatakse kõrgepinge piirväärtust kasutades, edastatakse mõõtmistulemus, kuid lisatakse märge (hoiatus), et mõõtmine on sooritatud mõõtepiirkonna servas ning võimalusel tuleks mõõtmist korrata muutes eelnevalt valgusallika seadeid.

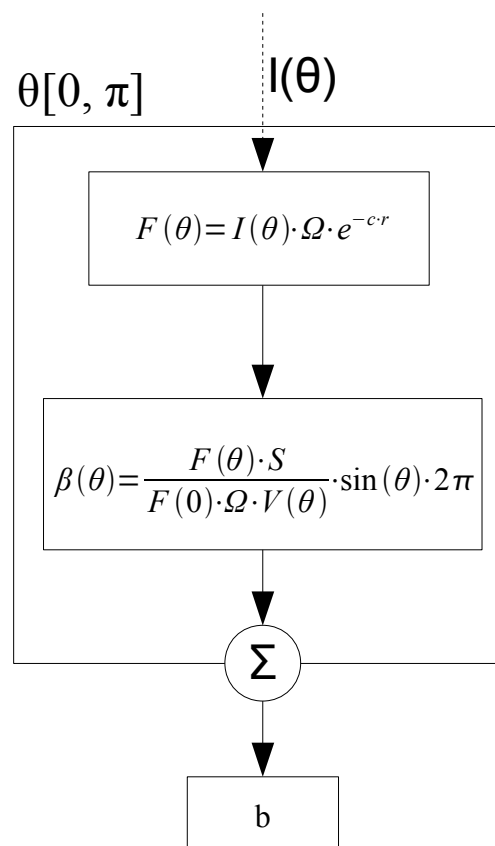


Joonis 15: PMT-le langeva valguse mõõtmise järgnevusskeem.

Lisaks nimetatud toimingutele võib vajadusel lisanduda PMT jahutuse juhtimine. Selleks tuleks teostada pidevat temperatuuri mõõtmist ning vastavalt temperatuurile muuta jahutuse intensiivsust kasutades PID algoritmi.

4.8 Arvutusalgoritmi plokkskeem

Joonisel 16 on kujutatud kontrolleris teostatava hajuvuskoefitsiendi (vaata 3.1 Tagasihajumiskoefitsient) arvutuse plokkskeemi. Sisendiks on valgusvoog $I(\theta)$, mis mõõdetakse vahemikus 0 kuni π PMT abil. Samuti mõõdetakse seadet läbinud otsese valgusvoo väärtus, mis on vajalik nõrgenemiskoefitsiendi c arvutamiseks. Vahearvutuste tulemused summeritakse ja nii saadakse otsitav hajumiskoefitsient. Tagasihajumiskoefitsient saadakse, kui arvutamisel kasutatakse mõõtmistulemusi vahemiku $\pi/2$ kuni π .



Joonis 16: Hajumiskoefitsiendi arvutamise skeem.

Joonisel on:

$I(\theta)$ – valgusvoog ($\text{W} \cdot \text{sr}^{-1}$), mõõdetakse PMT-ga,

θ – nurk mille juures mõõtmise tehti,

Ω – fotokordisti vaatenurk (sr), mille määrab PMT ees oleva ava suurus,

c – kiire nõrgenemiskoefitsient (m),

r – kaugus hajumise ruumi keskkohast kuni fotodetektorini (m),

$F(0)$ – vette sisenev kiirgusvoog (W),

S – siseneva kiirgusvoo ristlõikepindala (m^2),

$V(\theta)$ – hajumise ruumala (m^3),

b – hajumiskoeffitsient (m^{-1}).

Suurused Ω ja r on antud seadme jaoks konstandid (Ω – väärtus on erineva PMT ava korral erinev). $F(0)$ ja S on konstantsete väärtustega - nende täpne väärtus sõltub kasutatavast valgusallikast ja avast. Need väärtused määratakse eksperimentaalselt pärast seadme valmimist ning neid kasutatakse antud valemis kui konstante.

Ruumhajumisfunktsiooni mõõtmise käigus varieerub hajumisruumala $V(\theta)$ seadme mehaanilise ehitusega määratud viisil. Hajumisfunktsiooni määramisel, eriti väikeste ja suurte nurkade korral, on väga oluline teada hajumisruumala suurust võimalikult täpselt. Arvutuste jaoks saab piisavalt täpselt defineerida ruumala muutust üldises piirkonnas ($10^\circ \dots 170^\circ$) valemiga $V(\theta) = V(90^\circ) / \sin(\theta)$ kus $V(90^\circ)$ on hajumisruumala 90° juures, mille omakorda määravad seadme ehitusparameetrid. Seda valemit saab aga kasutada ainult üldises piirkonnas, eraldi on vaja määrata hajumisruumala väikeste ja suurte hajumisnurkade korral ($0^\circ \dots 10^\circ$ ja $170^\circ \dots 180^\circ$).

Antud arvutuses saab kontrolleri jõudlust efektiivsemalt kasutada tabuleerides osade parameetrite väärtused. Näiteks saab kasutada tabuleeritud arvulisi väärtusi $\sin(\theta)$ ja e^{-cr} jaoks. Nii saab oluliselt vähendada kontrolleri poolt teostatavate ujukomaarvutuste mahtu, mis võimaldab märkimisväärselt vähendada arvutusaega.

4.9 Kontrolleri võimalikud variandid

Kõike eelnevat kokku võttes peaks kontrolleri valikul arvestama järgnevate piirangutega:

- kontrolleri, mis suhtleb teiste komponentidega, peab toetama piisaval hulgal valitud välisliideseid (RS-232C).
- Kasutajaga suhtlev kontrolleri peab olema võimeline suhtlema kasutades Ethernet' ühendust.
- PMT kontrolleri peab suutma toime tulla AD-DA muunditega (hinna-kvaliteedi-kiiruse suhtelt sobivad võiksid olla SPI liidesega ühendatavad AD-DA muundid).
- Andmete salvestamiseks peab juhtkontrolleri suutma suhelda mälukaartiga SPI liidest kasutades.
- Kõigil kontrolleritel peab olema piisavalt mälu arvutuseks vajalike muutujate hoidmiseks.
- Kontrolleri peavad omama piisavalt arvutusressurssi tulemaks toime neile vajalike arvutusülesannetega.

Lisaks sellele tuleks kontrolleri valikul veel tähelepanu pöörata tema kasutamise keerukusele. Ehk siis võimalusel tuleks eelistada levinumat kontrolleri, et probleemide tekkimisel ja hilisemate arenduste puhul oleks võimalik leida kompetentseid arendajaid ja samuti peaks kontrolleri olema piisavalt laialt levinud, vältimaks probleemide tekkimist, kui tootjafirma mingil põhjusel otsustab kontrolleri tootmise lõpetada. Samuti tuleb tähelepanu pöörata sellele, mis arendusvahendeid on võimalik valitud kontrolleri tarkvara koostamiseks kasutada ning kas on olemas vajalikud arenduse ja silumise vahendid. Eelistatavalt peaksid juhtimis- ja mõõtekontrolleri olema sama seeria omad, et lihtsustada nende programmeerimist.

Kontrolleri valiku tegemiseks koostame tabeli järgmiste kontrolleri omaduste kohta (sulgudes on iga omaduse oletatav kaalufunktsiooni väärtus skaalas 0...10 – mida suurem number, seda olulisem on antud kriteerium):

1. Jõudlus (10) – kontrolleri kiirus, arhitektuur, mälumaht, bittide arv.
2. Vajalikud liidesed (10) – kas ja kuidas on toetatud MVSM jaoks vajalikud liidesed (Ethernet, UART, SPI jne).
3. Laiendatavus (8) – kas on lisaks liideseid või muud hiljem vaja minna võivat?
4. Energiatarve (3) – energiatarve puhkeolekus ja maksimumenergiatarve.
5. Keskkonnatingimused (10) – kas kontrolleri suudab töötada eeldatavates

keskkonnatingimustes $-5^{\circ} \dots 50^{\circ} \text{C}$?

6. Arendustarkvara tugi (10) – nii kommerts- kui ka vaba arendustarkvara valik. Tarkvara kasutatavus ja tugi.
7. Kooskonna (Community) olemasolu ja tugi (5) – erinevad toe saamise võimalused nagu teised kasutajad foorumis või tootja pakutav tugi.
8. Süsteemi koostajate eelnevad teadmised ja oskused (10) – kui palju on süsteemi koostajatel kokkupuuteid antud kontrolleri, arhitektuuri ja arendusvahenditega?
9. Süsteemi koostajate valmisolek uusi kontrollereid proovida (5) – kas süsteemi koostajad on nõus õppima uusi tehnoloogiaid ja valmis riskima võimaliku pikema ajakuluga tarkvara koostamisel?
10. Sobivus väiketootmiseks ja prototüüpimiseks (10) – kas kontrolleri on võimalik käsitsi joota (elimineerib BGA korpuses kontrolleri kasutamise) ning kas tootja või kolmanda osapoole poolt on saadaval arendusplaate prototüübi valmistamise lihtsustamiseks?
11. RTOS (5) – kas ja milliseid reaalaaja operatsioonisüsteeme kontrolleri toetab. Kas on kättesaadavad vabakasutuse reaalaajaoperatsioonisüsteemid või ainult tasulised? Kas valitud RTOS-i on võimalik kasutada ka teise kontrolleri. Kas RTOS toetab failisüsteemi ja Etherneti?
12. Ülesannete jaotamine kontrolleri vahel (10) – kuidas sobib antud kontrolleri talle antud ülesannete sooritamiseks?
13. Hind ja kättesaadavus (2) – kontrolleri hind (väheoluline, kuna tõenäoliselt ei ole kavandatud seade masstootmise objekt) ja kättesaadavus erinevate vahendajate kaudu. Eeldatav toote ja toe kättesaadavus lähema 10 aasta jooksul.

Juhtkontrolleri valimiseks on võrreldud erinevate kiibitootjate kontrollereid, mis toetaksid minimaalselt Etherneti, kolme UART-i, ühte I²C-d ja ühte SPI liidest, et kontrolleri oleks võimalik juhtida kõiki seadme komponente. Lisaks piiras valikut nõue, et kontrolleri võiks olla 32-bitine, kuna enamused RTOS-e töötavad ainult 32-bitiste kontrolleri. Võrdlemiseks on iga firma korral valitud võimalikult suure taksageduse ja mäluga kontrolleri [29][30][31][32][33].

Tabel 1: Valik võimalikke kontrollereid 1.

	Texas Instruments ARM Cortex-M3 Stellaris 8000 seeria LM3S8938	Atmel 32-bit AVR UC3 AT32UC3A0512	NXP Cortex-M3 LPC1700 seeria LPC1768FBD100
Jõudlus	50 MHz, 62.5 DMIPS, ARM Cortex-M3, RAM 64 KB, flash 256 KB [34]	66 MHz, 91 DMIPS, AVR RISC, RAM 64 KB, flash 512 KB [35]	100 MHz, ARM Cortex-M3, RAM 64 KB, flash 512 KB [36]
Liidesed	1 Ethernet, 3 UART, 2 I ² C, 2 SPI, 1 USB, 2 CAN	1 Ethernet, 4 UART, 1 I2C, 6 SPI, 1 USB	1 Ethernet, 4 UART, 3 I2C, 1 SPI, 1 USB, 2 CAN
Laiendatavus	Firma pakub mitmeid sarnase kasutusloogikaga, aga suurema jõudluse ja võimalustega kontrollereid.	Laiendatavus piiratud konkreetse kontrolleri perekonnaga.	Firma pakub mitmeid sarnase kasutusloogikaga, aga suurema jõudluse ja võimalustega kontrollereid.
Energia-tarve	Täielik jõudeolek 0.25mA, magamine 16mA, töörežiim 108mA (Ethernet ühendatud)	Täielik jõudeolek 0.047mA, magamine 16.8mA, töörežiim 36.3mA.	Täielik jõudeolek 0.3mA.
Keskkond	-40...85° C	-40...85° C	-40..85° C
Tarkvara	Pakkuda on nii komerts- kui ka vabavaralisi arendusvahendeid tootjafirma enda ja kolmandatelt osapoolte firmade poolt. Lisaks saadaval mitmeid tasuta Eclipse põhiseid arendusvahendeid (Yagatro). GCC tugi.	Tootja poolt pakutakse tasuta arendustarkvara vahendeid. Lisaks on saada kolmanda osapoole tööriistu ja tasuta liideseid teiste arendusvahendite jaoks (Eclipse). GCC tugi.	Tootja pakub koostöös firmaga Code Red Technologies Eclipse-il põhinevat arendusplatvormi LPCXpresso koos JTAG siluriga. Code Red pakub lisaks tasuta rohkemate võimalustega versiooni. Olemas on IAR ja KEIL arendusvahendid. GCC tugi.

Tabel 1: Valik võimalikke kontrollereid 1.

	Texas Instruments ARM Cortex-M3 Stellaris 8000 seeria LM3S8938	Atmel 32-bit AVR UC3 AT32UC3A0512	NXP Cortex-M3 LPC1700 seeria LPC1768FBD100
Tugi	Tootja poolt on olemas kasutajaskonna foorum, samuti mitmed treeningharjutused ning näidised. Lisaks on pakkuda liitumisega kasutajatoe teenus.	Väga suure kasutajaskonnaga foorum (AVRFreaks). Pakutakse ka toe teenust.	Olemas tootjapoolne foorum ja kasutajatugi. Lisaks on NXP poolt pakkuda mitmeid õppematerjale ja juhendeid.
Kogemused	Arendusmeeskonnal on olnud varasemaid kokkupuuteid antud seeria kontrollritega.	Arendusmeeskond omab teoreetiliselt ülevaadet kontrollrist, probleemiks arvatavasti kontrolleri tegelik vähene levik võrreldes ARM kontrollritega.	NXP kontrollrite kasutamine on perspektiivikas nende laia leviku tõttu. Kuna tuumaks on Cortex-M3, siis probleeme kasutamisega ei tohiks tekkida.
Valmisolek		Küsitav, arvatavasti ei soovita omandada.	Perspektiivne uus kontroll omandamiseks.
Väike-tootmine	On saada ka pindjoodetava pakendiga. Tootja pakub omalt poolt mitmeid suuruselt ja võimalustelt sobivaid arendusplaate.	On saadaval pindjoodetava pakendiga. Samuti on saadaval nii tootja enda kui kolmanda osapoole arendus plaate.	On saadaval pindjoodetava pakendiga. Samuti on saadaval kolmanda osapoole pakutavaid arendusplaate.
RTOS	Saadaval on mitmeid tasulisi ja tasuta reaalaaja operatsioonisüsteeme. Näiteks toetavad antud protsessorit FreeRTOS, Pumpkin, CMX ja ThreadX. Neist esimene	Saadaval tasuta ja tasulisi RTOS-e. Näiteks RTOS CMX, FreeRTOS, ThreadX, MicroJvm.	Saadaval tasuta ja tasulisi RTOS-e. Näiteks CoOS, FreeRTOS, Micrium.

Tabel 1: Valik võimalikke kontrollereid 1.

	Texas Instruments ARM Cortex-M3 Stellaris 8000 seeria LM3S8938	Atmel 32-bit AVR UC3 AT32UC3A0512	NXP Cortex-M3 LPC1700 seeria LPC1768FBD100
	on tasuta ning võimaldab Etherneti ja SD kaarti kasutamist.		
Ülesanne jagamine		Ujukoma arvutuste jaoks eraldi floating-point unit.	
Hind	Hind tootja lehel 6.40\$. Saadaval nii firma enda veebilehelt kui ka paljude edasimüüjate käest 12.44€ [37]. Uuemate perekondade kontrolleri kättesaadavus võib osutuda probleemiks lähitulevikus.	13.68\$ - 15.23€ [37]. Saada mitme internetipoe vahendusel nii tükikaupa kui ka suuremate kogustena.	9.14€ erinevate interneti poodide hinnakirjas [37].
Kokku			

Tabel 2: Valik võimalikke kontrollereid 2.

	Freescal ColdFire V2 MCF52259CAG80	Microchip 32-bit MIPS M4K PIC PIC32MX795F512L	ST Microelectronics 32-bit ARM Cortex MCUs STM32F seeria STM32F107VC
Jõudlus	80 MHz, 76 DMISP, ColdFire, RAM 64 KB, flash 512 KB [38]	80 MHz, 124 DMISP, PIC, RAM 128 KB, flash 512 KB[39]	72 MHz, 90 DMISP, ARM Cortex-M3, RAM 64, flash 256 KB [40]
Liidesed	1 Ethernet, 3 UART, 2 I2C, 1 SPI, 1 USB, 1 CAN	1 Ethernet, 5 UART, 5 I2C, 4 SPI, 1 USB, 2 CAN	1 Ethernet, 5 UART, 2 I2C, 3 SPI, 1 USB, 2 CAN
Laiendatavus			
Energia-tarve	Täielik jõudeolek 0.15mA, jõudeolek 17mA, töörežiim 81mA	Täielik jõudeolek 0.12mA, jõudeolek 42mA, töörežiim 98mA	Täielik jõudeolek tüüp. 0.03mA max 0.6mA, jõudeolek 49mA, töörežiim 68.4mA
Keskkond	-40...70° C	-40...85° C	
Tarkvara	Tootja poolt Eclipsei põhine CodeWarrior, millest on piirangutega tasuta kui ka tasuline versioon. IAR arenduskeskkond.	Tootja poolt tasuta MPLAB ja C kompilaator.	Mitmeid kolmandate osapoolte arendusvahendeid. Enamus tasulised, nende hulgas ka IAR ja KEIL. GCC tugi.
Tugi	Olemas nii foorum kui ka online klienditugi. Samuti on tootja poolt mitmeid treening näidiseid.	Suure kasutajaskonnaga foorum.	Tootja hallatav foorum. Lisaks online tugi ja ka e-õppe vahendite olemasolu.
Koge-	Puuduvad.	Puuduvad	Kuna protsessoril on Cortex M3 tuum, siis

Tabel 2: Valik võimalikke kontrollereid 2.

	Freescall ColdFire V2 MCF52259CAG80	Microchip 32-bit MIPS M4K PIC PIC32MX795F512L	ST Microelectronics 32-bit ARM Cortex MCUs STM32F seeria STM32F107VC
mused			kasutamine ei tohiks valmistada probleeme.
Valmis- olek	Arvatavasti mitte.	Arvatavasti mitte, kuigi on perspektiivikas.	Perspektiivikas kontrollerr.
Väike- tootmine	Pindjoodetavas pakendis. Tootja poolt pakkuda demonstratsioonkomplekt.	Pindjoodetavas pakendis. Tootja poolt saada väga erinevaid arendusplaate. Lisaks saadaval ka kolmanda osapoole arendusplaate.	Pindjoodetavas pakendis. Saadaval nii tootja kui ka mitmete kolmanda osapoolte arendusplaate
RTOS	Tootja arendatav Freescale MQX, ei toeta SD kaarti. Lisaks toetavad veel näiteks FreeRTOS ja CMX	Spetsiaalselt PIC'i jaoks mõeldud AVIX. Lisaks CMX, ThreadX , FreeRTOS, Micrium, SEGGER- embOS	Saadaval tasuta ja tasulisi RTOS-e. Näiteks FreeRTOS, ChibiOS/RT, ThreadX, CMX, PowerPac, Micrium , SEGGER- embOS
Ülesan- nete jagamine			
Hind	9,83 € - 32.91€ [37]	10.27 € [37]	11.91€ [37]
Kokku			

4.10 Kontrolleri valik

Tabelites 1 ja 2 välja toodud andmete põhjal on soovitatav valida Texas Instruments LM3S8938 kontrolleri. Seda põhjusel, et antud tootele pakutav arendusvahendite hulk on kõige suurem: tootjalt on laias valikus sobivaid arendusplaate ning ka lai arendustarkvara valik ja saadaval on mitmeid sobivaid RTOS-e. Antud kontrolleri kasuks räägib ka see, et tootjal on hiljuti valminud uue, võimsama seeria kontrolleri, millele üleminek hiljem, kui vajadus tekib, oleks küllaltki lihtne. Nagu ka kahel konkureerival kontrolleri on LM3S8938 tuumaks Coretx-M3 protsessor, mis tähendab, et kasutamise poolest ei tohiks antud kontrolleri olla keerulisem kui teised ja üleminek teistele sama tuumaga kontrolleritele peaks olema raskusteta teostatav.

Samas peab märkima, et kontrolleri on peamiste näitajate poolest küllaltki võrdsed. Selle tõttu võib arendaja soovi korral valida mõne teise kontrolleri perekonna, millega on olnud varasemaid kokkupuuteid või mille kasutamist tulevikus peetakse perspektiivikaks.

4.11 Kontrolleri töö realiseerimine

4.11.1 Juhtkontroller

Juhtkontrolleri töö on otstarbekas realiseerida kasutades RTOS-i, kuna kontroller peab paralleelselt suhtlema mitme erineva jadaliidesega ning teostama arvutusi ja tegelema andmete haldamisega mälukaardil.

4.11.2 Mõõtekontroller

Mõõtekontrolleri töö realiseerimiseks võiks samuti kasutada RTOS-i, kuna siis on mõlema põhilise kontrolleri töö realiseeritud sarnaselt. Samuti võimaldab RTOS hõlpsamalt vahetada kasutatavat kontrollerit, kui selleks peaks tekkima vajadus.

4.11.3 Sisendid/väljundid

Juht- ja mõõtekontrollerite sisendid ja väljundid on toodud tabelites 3 ja 4.

Tabel 3: Juhtkontrolleri sisendid ja väljundid.

Suund	Nimi	Andme-/ühenduse liik
Sisendid		
	LAN	Ethernet
	Temperatuuri andur	I ² C
	Rõhu andur	I ² C
	Mõõtekontroller	RS-232C
	LED-ide ajur	RS-232C
	Prisma mootori juhtimine	RS-232C
	Pump sees/väljas	boolean
	Toide	12V
Väljundid		
	LAN	Ethernet
	Temperatuuri andur	I ² C
	Rõhu andur	I ² C

Tabel 3: Juhtkontrolleri sisendid ja väljundid.

Suund	Nimi	Andme-/ühenduse liik
	Mõõte kontroller	RS-232C
	LED-ide ajur	RS-232C
	Prisma mootori juhtimine	RS-232C
	Pump sisse/välja	boolean
	Mälukaart	SPI

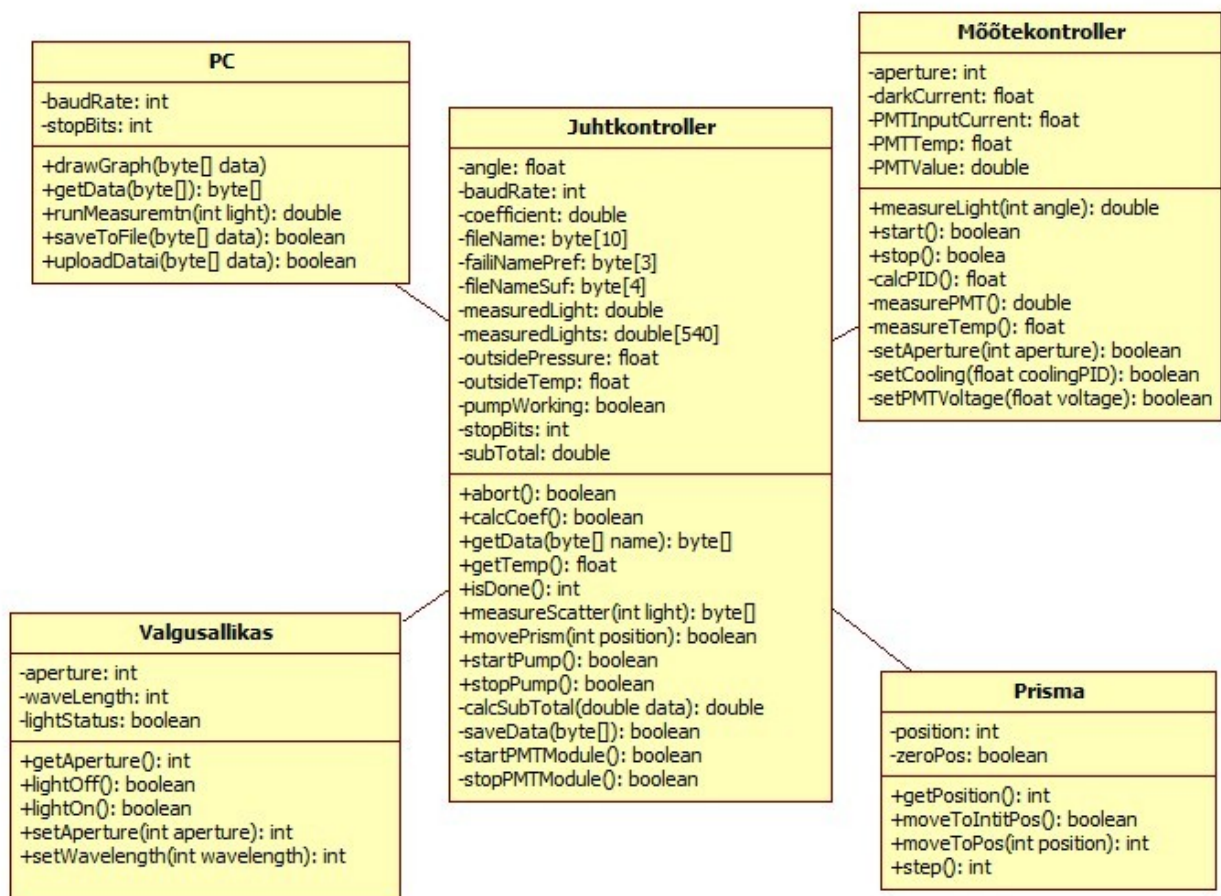
Tabel 4: Mõõtekontrolleri sisendid ja väljundid.

Suund	Nimi	Andme-/ühenduse liik
Sisendid		
	Juhtkontroller	RS-232C
	Ava nullpositsioon	boolean
	PMT voolu väärtus	16-bit ADC
	PMT temperatuur	10-bit ADC
	Mõõdetav kõrgepinge	10-bit ADC
	Toide	+12V
Väljundid		
	Juhtkontroller	RS-232C
	Ava mootori ajur	Samm, suund
	Seatav kõrgepinge	10-bit DAC

4.11.4 Muutujad ja parameetrid

Süsteemis kasutavatest muutujatest ja meetoditest annab ülevaate klassidiagramm, mis on kujutatud joonisel 17.

Tabelis 5 on toodud üks variant vajaminevatest muutujatest ning välja- ja sissepoole näha olevatest meetoditest. Ära on toodud kõige olulisemad elemendid koos lühiselgitusega.



Joonis 17: Süsteemi klassi diagramm.

Tabel 5: Kontrollerite muutujate ja meetodite kirjeldus.

Nimi	Tüüp	Kirjeldus
Juhtkontroller		
<i>Muutujad</i>		
fileName	private: byte[10]	Faili nimi, kuhu viimase mõõtmise tulemused salvestatakse.
fileNamePref	private: byte[3]	Faili nime algus, mis on kõigil failidel ühine.
fileNameSuf	private: byte[4]	Faili nime lõpp (nt “.csv”).
coefficient	private: double	Arvutatud koefitsiendi väärtus. Muutuja arvutuses kasutamiseks ja hilisemaks salvestamiseks.
angle	private: float	Prisma nurk mille juures mõõtmist hetkel teostatakse.
pumpWorking	private: boolean	Pumba sisendit määrav muutuja.
outsidePressure	private: float	Väline rõhk. Mõõdetud väline vee rõhk. Salvestatakse koos mõõtetulemustega, et vee sügavust arvutada.
outsideTemp	private: float	Mõõdetud välistemperatuuri väärtus.
subTotal	private: double	Koefitsiendi arvutamise vahetulemus
measuredLight	private: double	Viimane valgusmõõtmise väärtus, tagastatud mõõteseadmelt.
measuredLights	private: double[540]	Kõik mõõtmiste väärtused. Massiivi suurus määratud prisma sammude arvuga (esialgne hinnang suurusele 180*3).
<i>Meetodid</i>		
abort()	public: boolean	Katkestatakse mõõtmine.
getData(byte[] nimi)	public: byte[]	Tagastab vastavalt faili nimele mälukaardile salvestatud mõõtmise andmed. Edastatakse terve fail baitidena.

Tabel 5: Kontrollerite muutujate ja meetodite kirjeldus.

Nimi	Tüüp	Kirjeldus
getTemp	public: float	Mõõdab välise temperatuuri ja tagastab selle väärtuse.
calcCoef	public: double	Arvutab ja tagastab viimaste mõõtmistulemuste põhjal hajumisekoefitsiendi.
isDone	public: int	Tagastab vastuseks 0 kui mõõtmine on sooritatud, 1 kui mõõtmine on pooleli ja 2 kui mõõtmine ebaõnnestus.
measureScatter(int valgus)	public: byte[]	Alustab hajuvuse mõõtmist soovitud valguse lainepikkusega valgust kasutades. Edastab failinime, mida kasutades saab küsida mõõtmise andmeid.
movePrism(int asend)	public: boolean	Liigutab prisma etteantud asendisse. Tagastab 0 kui toiming õnnestus ja 1 kui ei õnnestunud.
startPump()	public: boolean	Lülitab mõõtekambri pumba sisse. Tagastab 0 kui toiming õnnestus ja 1 kui ei õnnestunud.
stopPump()	public: boolean	Lülitab mõõtekambri pumba välja. Tagastab 0 kui toiming õnnestus ja 1 kui ei õnnestunud.
calcSubTotal(double valgus)	private: double	Kasutades viimast mõõtetulemust arvutab vahetulemuse ja tagastab selle väärtuse.
saveData(byte[] andmed)	private: boolean	Salvestab kõik mõõtmisega seotud andmed mälukaardile.
stopPMTModule	private: boolean	Äratab PMT mooduli. Tagastab 0 kui toiming õnnestus ja 1 kui ei õnnestunud.

Tabel 5: Kontrollerite muutujate ja meetodite kirjeldus.

Nimi	Tüüp	Kirjeldus
startPMTModule	private: boolean	Annab PMT moodulile käsu minna puhkeolekusse. Tagastab 0 kui toiming õnnestus ja 1 kui ei õnnestunud.
Mõõtekontroller		
<i>Muutujad</i>		
aperture	private: int	PMT ees oleva ava positsioon, 0 tähendab suletud ava.
darkCurrent	private: float	Viimati mõõdetud pimevoolu väärtus.
PMTinputCurrent	private: float	Mõõdetud PMT toitepinge.
PMTTemp	private: float	Mõõdetud PMT temperatuur. Vajalik jahutuse reguleerimiseks ja pimevoolu arvutamiseks.
PMTValue	private: double	PMT väljund väärtus
<i>Meetodid</i>		
measureLight(int nurk)	public: double	Mõõdab kordistile langeva valguse. Etteantud nurga väärtust kasutatakse eeldatava ava ja kõrgepinge seadmiseks. Teostatakse mõõtmiskorrektuurid ja pimevooluga kompenseerimine. Tagastatakse mõõteväärtus või negatiivne mõõteväärtus, kui mõõtmine toimus piirialal (hoiatus).
stop	public: boolean	Lülitab välja PMT ja tema jahutuse. Sulgeb PMT ees oleva ava. Tagastab 0 kui toiming õnnestus ja 1 kui ei õnnestunud.
start	public boolean	Komponendid seatakse töövalmis. Kontrollitakse PMT temperatuuri ja mõõdetakse pimevool. Tagastab 0 kui toiming õnnestus ja 1 kui ei õnnestunud.

Tabel 5: Kontrollerite muutujate ja meetodite kirjeldus.

Nimi	Tüüp	Kirjeldus
calcPID	private: float	Arvutab temperatuuri järgi uue jahutuse juhtväärtuse.
measurePMT	private: double	Mõõdab PMT voolu väärtuse ADC abil.
measureTemp	private: float	Mõõdab PMT temperatuuri.
setAperture(int aperture)	private: boolean	Seab PMT ees oleva ava suurust. Sisend 0 tähendab kinnist ava. Tagastab 0 kui õnnestus, 1 kui tekib viga ava seadmisel.
setCooling(float coolingPid)	private: boolean	Seab etteantud väärtuse (PID arvutuseks) jahutuse väärtuseks. Tagastab 0 kui toiming õnnestus ja 1 kui ei õnnestunud.
setPMTVoltage(float voltage)	private: boolean	Seab PMT kõrgepinge DAC abil. Tagastab 0 kui toiming õnnestus ja 1 kui ei õnnestunud.

Tabelis 6 on toodud välja kontrolleri vahel kasutatavate käskude üks võimalik variant.

Käsu koostamisel kasutatakse kuju <MOODUL><TEGEVUS><NIMI>[<PARAMEETER>] vastust oodatakse aga kujul <MOODUL><NIMI> [OK/FAIL][<PARAMEETER>].

- MOODUL – tähistab käsu saaja nime. Antud juhul on võimalikud variandid mõõtekontroller M, juhtkontroller C, valguse moodul L ja prisma moodul P.
- TEGEVUS – käsu tüüp. Andmete küsimise käsk R, väärtuse seadmise käsk W ja toimingu alustamise käsk S.
- NIMI – käsu nimi. See osa käsust, mis määrab, millist ülesannet käsk täidab.
- PARAMEETER – käsuga kaasaantav parameeter. Parameetri ja ülejäänud käsu eraldajaks on tühik.
- OK/FAIL – Lühike kirjeldus kas käsk õnnestus või mitte. Võimalikud väärtused O OK tähistamiseks ja F FAIL tähistamiseks, ehk vastavalt, kas toiming õnnestus või ebaõnnestus.

Märkide [ja] vahel olev osa ei ole käsu koostamisel tingimata vajalik.

Protsessi seisundi küsimiseks kasutatakse võimalikult lühikesi käske, kuna neid tuleb tõenäoliselt edastada mitu korda. Seisundi küsimise käskude vastused peavad samuti olema lühikesed kujul <MOODUL><NIMI> <PARAMEETER>. Parameetri väärtus võib olla 0, kui tegevus on lõpetatud või 1, kui tegevust sooritatakse ning 1-st suurem number veakoodina, kui tegevuse käigus tekkis viga. Käskude ja vastuste koostamisel peab jälgima, et aeglasel protsessil vastaksid kohe, et tegevuse teostamisega on alustatud; tegevuse tulemust peab küsima eraldi käsuga, kui seisundi küsimine annab vastuseks, et tegevus on lõpetatud.

Tabelis kasutakse andmetüübi sümboleid järgnevates tähendustes:

- <I+> – positiivne täisarv (integer),
- <D> – topelttäpsusega reaalarv (double),
- <F> – reaalarv (float),
- <S> – parameeter sõne kujul (byte[]),
- <BA> – baitide jada (byte[]).

Sellisel käsu süntaksi kasutamisel on mitu põhjust. Esiteks on sel viisil koostatud käske võimalik kasutada, kui ühe liidese külge on ühendatud mitu seadet, kuna käsu- ja vastusestruktuur tagab, et seadmed ei sega üksteist. Teiseks on lihtne jälgida, kellele käske saadetakse ja millisele käsule parajasti vastatakse.

Tabel 6: Kontrollerite suhtlemisel kasutatavad käsud.

Käsk	Kirjeldus	Vastus
Mõõtekontroller		
MSM <I+>	Alustab valguse mõõtmist. Ette antakse nurk mille juures valgust mõõdetakse.	MMO / MMF <I+>
MRS	Küsib mõõteprotsessi staatust.	MS <I+>
MRM	Tagastab mõõteprotsessi tulemuste.	MMO <D> / MMF <I+>
MSW	Äratab mõõte mooduli.	MWO / MWF <I+>
MSS	Seiskab mõõte mooduli.	MSO / MSF <I+>
Juhtkontroller		
CSC	Alustab koefitsiendi arvutamist.	CCO / CCF <I+>
CRC	Tagastab koefitsiendi arvutus staatuse.	CC <I+>
CRCV	Tagastab arvutatud koefitsiendi väärtuse.	CCVO <D> / CCVF <I+>
CRF <S>	Tagastab mõõte andmed failist.	CFO <BA> / CFF <I+>
CRT	Tagastab seadme välistemperatuuri.	CTO <F> / CTF <I+>
CRS	Tagastab mõõtmisprotsessi staatuse.	CS <I+>
CSM <I+>	Alustab hajumise mõõtmist. Vastuseks tagastab faili nime kuhu tulemused salvestatakse.	CMO <S> / CMF <I+>
CWPR <I+>	Liigutab prisma antud asendisse.	CPRO <I+> / CPRF <I+>
CWPU <I+>	Lülitab pumpa. 1 käivitab pumpa ja 0 lülitab pumpa välja.	CPUO / CPUF <I+>
Prisma		
PRP	Tagastab prisma positsiooni.	PPO <I+> / PPF <I+>
PSZ	Liigutab prisma alg asendisse.	PZO / PZF <I+>
PWN <I+>	Liigutab prisma antud positsioonile.	PNO / PNF <I+>
PSS	Pöörab prisma sammu võrra ja tagastab uue positsiooni.	PSO <I+> / PSF <I+>

Tabel 6: Kontrollerite suhtlemisel kasutatavad käsud.

Käsk	Kirjeldus	Vastus
Valgusallikas		
LEA	Tagastab ava asendi.	LAO <I+> / LAF <I+>
LWL	Lülitab valgust. Parameeter 1 lülitab valguse sisse ja 0 lülitab valguse välja.	LLO / LLF <I+>
LWP <I+>	Seab ava etteantud asendisse. Tagastab uue ava asendi.	LPO <I+> / LPF <I+>
LWW <I+>	Seab valgus allika sobivale lainepikkusele. Tagastab uue lainepikkuse väärtuse.	LWO <I+> / LWF <I+>

4.12 Kasutatavad algoritmid

PMT kasutamisel tuleb väikese müra ja korrektsete mõõtmistulemuste saamiseks hoida fotokordisti temperatuur võimalikult stabiilsena ja soovitavalt ka võimalikult madalana, kuna fotokordistite pimevool suureneb temperatuuri tõustes. Temperatuuri stabiliseerimiseks on sobiv kasutada klassikalist PID juhtimisalgoritmi, mis põhineb tagasiside kasutamisel juhtsignaali väärtuse leidmiseks. PID algoritmi korral kasutatakse soovitava juhtimisobjekti seisundi ja mõõdetud seisundi vahe väärtust ehk viga (P komponent), selle vea integraali (I komponent) ja tuletist (D komponent) selleks, et määrata juhtimisobjektile toimiva juhtsignaali väärtus. See tähendab, et PID algoritmi kasutamiseks peab pidevalt mõõtma juhitava parameetri väärtust. Pärast iga mõõtmist arvutatakse vastavalt mõõtetulemuse ja soovitud tulemuse erinevusele parandus, mida rakendatakse juhtsignaalile. Antud seadme puhul mõõdetakse PMT temperatuuri ning temperatuuri kõrvalekaldumisel soovitud tulemusest korrigeeritakse seadme jahutust jahutusseadet läbiva voolu muutmisega [12].

Proportsionaalne komponent PID algoritmis mõjutab väljundväärtust võrdeliselt vea suurusega, selle komponendi väärtus arvutatakse valemiga 14:

$$P = K_p e(t). \quad (14)$$

Integraalne komponent võimaldab elimineerida süsteemi püsitalitusvea, ehk hoida juhtimisobjekti täpselt soovitud seisundis. Seda komponenti arvutatakse valemi 15 abil:

$$I = K_i \int_0^t e(\tau) d\tau. \quad (15)$$

Diferentsiaalne komponent näitab vea muutumise kiirust ja võimaldab stabiliseerumisprotsessi kiirendada. Seda komponenti arvutatakse valemi 16 abil:

$$D = K_d \frac{d}{dt} e(t). \quad (16)$$

Juhttoime väärtus (valemis tähistatud kui $u(t)$) saadakse, kui liidetakse kokku kõigi kolme komponendi poolt tekitatud toimed (valem 17):

$$u(t) = P + I + D. \quad (17)$$

Valemities kasutatud $e(t)$ tähistab vea väärtust ehk näitab juhtimisobjekti soovitud väärtuse erinevust tegelikust väärtusest.

Valemis kasutatud konstandid K_p , K_i ja K_d määravad iga komponendi kaalu juhttoime moodustamisel. Nende konstantide väärtused määratakse konkreetse süsteemi jaoks empiiriliselt kasutades klassikalise juhtimisteooria reegleid. Olenevalt rakenduse omapärast võib PID algoritmist ära jätta I või D komponendi [12].

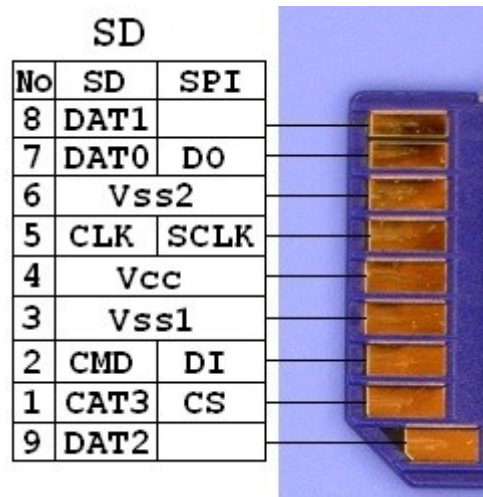
4.13 Mootorite juhtimine

Kõik kasutatavad mootorid on samm-mootorid, mille juhtimiseks kasutatakse spetsiaalset mootori juhtimise draiverit. Samm-mootorite juhtimiseks on saadaval mitmesuguseid draivereid. Keerulisemad ja kallimad võimaldavad paljude mikrosammude kasutamist ja on juhitud mingi standardse jadaliidese abil (nt I²C või RS-232). Lihtsamate komponentide kasutamise korral tuleb vajaliku täpsuse tagamiseks ise rohkem vaeva näha sobiva algoritmi programmilise realiseerimisega.

4.14 Andmete salvestamine

Andmete salvestamisel on otstarbekas kasutada SD (*Secure Digital Memory Card*) mälukaarti. Väiksemamõõdulisema seadme puhul võib kasutada ka sama kaardi vähendatud versioone miniSD või microSD. Andmeid hoitakse kaardil FAT12/16/32 failisüsteemi kasutades failidena.

Kaardiga suhtlemiseks on peamiselt kolm võimalust: ühe- ja neljabitine SD standard ning SPI standard. Kaardil on kokku 9 kontakti, millest 3 on toite ja 6 andmevahetuse jaoks (joonis 18). Väiksemates mikrokontrollersüsteemides kasutatakse peamiselt SPI suhtlust, kuna see on märgatavalt lihtsam programmeerida ning kontrollritel on reeglina sisseehitatud tugi selle standardi kasutamiseks. Lisaks sellele on kõigil SD kaartidel sisseehitatud kontrolleri andmete haldamiseks, mis suudab tavaliselt ka SPI standardile vastavalt suhelda. SPI ühenduse puhul kasutatakse ära ainult 4 andmevahetuseks mõeldud SD kaardi kontakti. Kõik SD kaardid toetavad standardseid käske. Kõik käsud, mida kontrolleri SD kaardile saadab, on fikseeritud kuuebaidise pikkusega. Kaart saadab alati vastuseks ühe kolmest võimalikust vastuse tüübist. SD kaardiga andmeid vahetades on kasutatava andmebloki suuruseks reeglina 512 baiti. SD kaart vajab töötamiseks 2.7-3.6 V toidet ja voolu kuni 100 mA [41].



Joonis 18: SD kaardi kontaktid SPI ja SD ühenduse jaoks.

Kaardi programmilise kasutamise lihtsustamiseks on võimalik kasutada vabavarana saadaolevat FAT failisüsteemi moodulit FatFs. See võimaldab kasutajat olemasoleva API abil hallata faile ning

teostada andmete lugemist ja kirjutamist. Antud failisüsteemi moodul on laialt levinud ja seetõttu on tema jaoks paljude kontrolleriperekondade tarbeks kirjutatud riistvaralise SPI suhtluse jaoks vajalik I/O kiht, mis vähendab märgatavalt andmete salvestamise ja lugemise programmeerimiseks vajalikku tööd. Tegemist on tasuta levitatava komponendiga, mille kasutamist ei piirata ka ühegi litsentsiga.

5 Kokkuvõte

Käesolevas töös vaatluse all oleva ruumhajumismõõtja (MVSM) esimene prototüüp vee ruumhajumise mõõtmiseks töötati välja Ukraina ja Kanada teadlaste koostöös. See prototüüp osutus aga sobimatus, kuna ei vastanud paljudele ootustele. Vajaka jäi nii optilises ja mehhaanilises kui ka elektroonilises ja tarkvaralises lahenduses.

Käesoleva magistritöö raames on koostatud moderniseeritud multispektraalse ruumhajumismõõtja juhtkontrolleri funktsionaalne disain, mille põhjal on analüüsitud arhitektuurilise disaini võimalikke variante. Töö tulemusi on kavas rakendada moderniseeritud vee ruumhajumismõõtja prototüübi valmistamisel.

Töös on toodud välja vajaliku riistvara ja tarkvara võimalik partitsioneerimine ning kontrolleritele jäävad ülesanded ning nende lühikirjeldused. Täpsemalt on lahti seletatud hajumiskoeffitsiendi arvutamise võimalused ning selle osana teostatav hajunud valguse mõõtmisprotsess. Lisaks on toodud hajumiskoeffitsiendi arvutamiseks vajalikud valemid ning kirjeldatud PMT jahutamiseks vajalik PID algoritmi põhiolemus. Töös on esitatud ka võimalik kontrollerite sisendite ja väljundite nimekiri ning seadme komponentide tarkvara meetodite ja põhimuutujate struktuur. Samuti on pakutud üks võimalik variant komponentidevahelises suhtluses kasutatavatest käskudest.

- Tulevikuperspektiivina nähakse ette minimalistliku juhtpuldi loomist, et tagada seadme sobivust välitöödeks. Selle realiseerimiseks on kõige sobivam selline MVSM-i juhtimisstiil, kus seade on piisavalt autonoomne, et kõik vajalikud mõõtmis- ja juhtimistoimingud ise sooritada ning seadmele vajalikud käsud saab edastada välise arvuti või lihtsama juhtpuldi abil kasutades Ethernet-i. Sellise juhtimisstiili korral peavad põhilised juhtimiskäsud olema lihtsad, lisaks peab olema võimalik välist arvutit kasutades saada ligi kõigile mõõtetulemustele, et neid vajadusel täpsemalt analüüsida ja kasutada.
- Mõõteseadme juhtimistegevuste jaotamisel on soovitatav selline variant, kus kasutatakse kahte eraldi kontrollerit: üks kontroller on vajalik fotokordisti (PMT) kasutamiseks ning teine mõõtmisprotsessi juhtimiseks, andmete salvestamiseks ja suhtlemiseks. Sellisel viisil juhtimist partitsioneerides on PMT kasutamine eraldatud teistest komponentidest, mis võimaldab edaspidi selle eraldiseisvat täiendamist. Lisaks sellele väheneb nii teiste komponentide poolt potentsiaalselt tekitatav mõõtemüra (nt mootorid), mis segaks PMT kasutamist. Juhtkontroller tegeleks sel juhul peamiselt komponentide töö koordineerimise ja andmevahetuse organiseerimisega.

- Andmevahetuseks seadme sees sobib kõige paremini RS-232C liides, kuna selle kasutamine on lihtne ja vajaduse korral on võimalik lihtsalt seadme erinevaid osi galvaaniliselt eraldada. Ühenduseks juhtarvuti või juhtpuldiga on perspektiivis kõige sobivam Ethernet ühendus, kuna see on laialt kasutatav, sobib suurte andmehulkade kiireks edastamiseks ja võimaldab kasutada pikki kaableid. Kontrolleri programmeerimise lihtsustamiseks võib kasutada valmis RS-232C Ethernet konverterit. Seadmesiseste parameetrite mõõtmiseks (nt termomeeter) sobib kasutada I²C ehk IIC-d, kuna paljud sobivad andurid seda kasutavad.
- Andmete salvestamiseks peaks kasutama microSD kaarti FAT32 failisüsteemiga, sobivaks tarkvarapaketi võiks olla FatFs. Sellisel juhul on lihtne koostada näiteks csv vormingus faili, milles hoitavat infot on hiljem lihtne juhtarvutis töödelda ja kasutada.
- Kontrolleri valikul oleks üks sobiv variant Texas Instruments-i Cortex M3 tuumaga ARM kontrolleri, kuna selle kasutamiseks pakub tootja mitmeid sobivaid näidisarendusi nii arendusplaatide kui ka näidisprojektidena. Samuti toodab TI mitmeid teisi sobivaid elektroonikakomponente, mida seadme valmistamisel vaja läheb (nt RS-232C draiver, ADC, DAC). Kuna aga võrreldud erinevate firmade kontrollereid on üldjoontes väga sarnased, võib valiku teostamisel olulise punktina lähtuda ka arendusmeeskonna eelnevatest kogemustest.

6 Summary

Author: Tiit Ginter

Title: „Multispectral Volume Scattering Meter hardware and software functional and architectural design“

The aim of the Masters' Thesis is to prepare multispectral volume scattering meter software and hardware functional and architectural design. First prototype of MVSM was created in collaboration of Canada and Ukraine scientists, but it did not meet all requirements. Along with optical and mechanical component changes it is needed to develop new electronics and software components according modern embedded systems design principles.

In given work necessary calculations for determining volume scattering function are presented and described. The algorithm for doing those calculations during the measurement process is presented. Furthermore, description of VSF measurement and light measurement process is given. Also the flow- and collaboration charts with explanations are presented for illustrating necessary tasks and work distribution between components.

The thesis also brings out one possible version of needed variables and methods which should be implemented. In addition one possible set of commands for communication between different components is proposed.

As a result of present study it was acknowledged that:

- For best usage MVSM should be an autonomous measuring device with remote control capability. Therefore system is designed to be as independent as possible and only data catering and starting measurement is done over Ethernet connection with computer or dedicated remote control device.
- By created design it is preferred to use two separate controllers for MVSM: one for using PMT for measuring and the other for controlling the measurement and communication process. This design version is preferred to others because it allows the developer to dedicate one controller mainly for PMT and so it can be upgraded separately if needed. The advantage of this schema is also that communications to all other components are then made from one single controller. That simplifies the implementation of the other controllers.
- Possible connection standards for communication between MVSM components and between MVSM and its controlling device are presented. For communicating between components it

is recommended to use RS-232C standard, because it is simple and common way for doing it. Also it enables galvanic isolation between separate parts. Between MVSM and external controller it is suggested to use Ethernet connection because it offers high speed and bandwidth over relatively long distance and is very common connection type.

- Comparison table for choosing suitable main controller and controller for PMT module with different possible controllers are presented. For ease of programming it is advised to use the same model or at least the same family of controllers for both those tasks. It is proposed to use Texas Instruments ARM Cortex-M3 Stellaris 8000 series controller because for these TI provides many different development boards and many other necessary development tools can be found for these. For remark it has to be mention that all controllers where very similar by their specifications, so any of these may be chosen if developer has experience with it or finds it to be more promising in the future.

Because MVSM will become an autonomous measurement device it needs to store results. In this work it is suggested that FatFs software module should be used to store data on SD memory card in FAT32 file system.

Given work will be used for creating new MVSM prototype.

7 Kasutatud kirjandus

1. "HydroScat Backscattering Sensor Family", <http://www.hobilabs.com/cms/index.cfm/37/152/1253/1254>, 24.05.2011.
2. "ECO Scattering Meters", <http://www.wetlabs.com/products/ebb/bbvsfindex.htm>, 24.05.2011.
3. Aare Puusaar, „Requirements and analysis of Multispectral Volume Scattering Meter“, 2011.
4. James K. Peckol, *Embedded Systems: A Contemporary Design Tool*, Wiley, 2007.
5. Heikki Vallaste, "e-Teatmik: IT ja sidetehnika seletav sõnaraamat", <http://vallaste.ee/>, 23.05.2011.
6. "Ball grid array", http://en.wikipedia.org/wiki/Ball_grid_array, 23.05.2011.
7. "Ethernet", <http://en.wikipedia.org/wiki/Ethernet>, 15.05.2011.
8. Phillip A. Laplante, *Real-Time Systems Design And Analysis*, 2004.
9. Hans Korge "RADIOMEETRIA. FOTOMEETRIA.", <http://www.physic.ut.ee/instituudid/efti/loengumaterjalid/opt/optika/foto.pdf>, 21.05.2011.
10. "Light-emitting diode", http://en.wikipedia.org/wiki/Light-emitting_diode, 23.05.2011.
11. Michael E. Lee, Marlon R. Lewis, "A New Method for the Measurement of the Optical VolumeScattering Function in the Upper Ocean", 2003.
12. "PID controller", http://en.wikipedia.org/wiki/PID_controller, 23.05.2011.
13. "Photomultiplier", <http://en.wikipedia.org/wiki/Photomultiplier>, 23.05.2011.
14. Lammert Bies, "RS232 Specifications and standard", http://www.lammertbies.nl/comm/info/RS-232_specs.html, 17.05.2011.
15. "Basics of the RS-485 Standard", http://www.bb-elec.com/tech_articles/rs485_basics.asp, 19.05.2011.
16. "Online course on Embedded Systems - SPI Bus interface", <http://www.eeherald.com/section/design-guide/esmod12.html>, 19.04.2011.
17. "Telecommunications Industry Association", http://en.wikipedia.org/wiki/Telecommunications_Industry_Association, 23.05.2011.
18. "Serial UART information", <http://www.lammertbies.nl/comm/info/serial-uart.html>, 23.05.2011.
19. Curtis D. Mobely, *Light and Water: Radiative Transfer in Natural Waters*, 1994.
20. "Beer–Lambert law", http://en.wikipedia.org/wiki/Beer–Lambert_law, 22.05.2011.
21. Birgot Paavel, "Peipsi järve vee optilised omadused", 2004.
22. Jack Ganssle, *The Art Of Designing Embedded Systems*, 2008.
23. "Universal Serial Bus", http://en.wikipedia.org/wiki/Universal_Serial_Bus, 19.05.2011.
24. "Online course on Embedded Systems - I2C Bus interface", <http://www.eeherald.com/section/design-guide/esmod11.html>, 19.04.2011.
25. "Embedded Systems Course - RS232 serial communications", <http://www.eeherald.com/section/design-guide/esmod8.html>, 15.05.2011.
26. Lammert Bies, "RS485 serial information", <http://www.lammertbies.nl/comm/info/RS-485.html>, 22.05.2011.
27. "IEEE 802.3", http://www.eeherald.com/section/design-guide/ieee802_3.html,
28. Michael E. Lee, "MVSM Manual", 2002.
29. "Stellaris® ARM® Cortex™-M3-based MCUs", <http://focus.ti.com/mcu/docs/mculuminaryprodsearch.tsp?sectionId=95&tabId=2485&familyId=1755>, 12.05.2011.
30. "MCU Product Selector", <http://www.microchip.com/productselector/MCUProductSelector.html>, 12.05.2011.

31. "NXP Microcoltrllers Cortex-M3 Selection guide",
[http://www.nxp.com/#/ps/ps=\[i=71568,f=c7487:7804e\]|pp=\[t=pfp,i=71568\]](http://www.nxp.com/#/ps/ps=[i=71568,f=c7487:7804e]|pp=[t=pfp,i=71568]), 12.05.2011.
32. "ColdFire", <http://www.freescale.com/webapp/sps/site/taxonomy.jsp?nodeId=018rH3YTLC00M9>, 12.05.2011.
33. "ATMEL 32-bit ", http://www.atmel.com/dyn/products/param_table.asp?category_id=163&family_id=607&subfamily_id=2138, 12.02.2011.
34. "LM3S8938 Microcontroller", <http://www.luminarymicro.com/products/lm3s8938.html>, 12.05.2011.
35. "AT32UC3A0512 Atmel AVR UC3",
http://www.atmel.com/dyn/products/product_card.asp?category_id=163&family_id=607&subfamily_id=2138&part_id=4117, 12.05.2011.
36. "NXP Semiconductor", <http://www.nxp.com>, 12.05.2011.
37. "Microcontrollers (MCU)", <http://ee.farnell.com/jsp/bspoke/bspoke4.jsp?bspokepage=farnell/>, 12.05.2011.
38. "MCF527X: V2 ColdFire® Integrated Microprocessor",
http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MCF527X&nodeId=018rH3YTLC00M91752, 12.05.2011.
39. "PIC32MX795F512L", <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en545660#3>, 12.05.2011.
40. "STM32F107VC STMicroelectronics", <http://www.st.com/internet/mcu/product/221020.jsp>, 12.05.2011.
41. "How to use MMC/SDC", http://elm-chan.org/docs/mmc/mmc_e.html, 13.05.2010.