

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond  
Thomas Johann Seebecki elektroonikainstituut

Margo Raja

## **ARM põhine mõõtemoodul hajutatud andmehõiveks**

Bakalaureusetöö

Juhendaja: Eero Haldre, diplomeeritud insener, vaneminsener

Tallinn 2016

## Autorideklaratsioon

*Deklareerin, et käesolev lõputöö on minu iseseisva töö tulemus, ning kinnitan, et esitatud materjalide põhjal ei ole varem akadeemilist kraadi taotletud. Kinnitan, et antud töö koostamisel on kõikide teiste autorite seisukohtadele, probleemipüstitustele, kogutud arvandmetele jmt viidatud.*

Kuupäev:

Autor: Margo Raja

.....

/allkiri/

# Lõputöö ülesanne

## Lõputöö teema:

ARM põhine mõõtemoodul hajutatud andmehõiveks.

## Teema päritolu:

TTÜ Thomas Johann Seebecki elektroonikainstituut

## Lõputöö eesmärgid:

Luaa andmehõive- ja kommunikatsioonifunktsioonid ARM põhisele platvormile EK-TM4C123GXL. Luaa sünkroniseerimise võimalus, kasutades spetsiaalset lisajuhet. Leida mõõtemooduli poolt tekitatud hilistumised.

## Oodatavad tulemused:

Juhkcontrolleril on võimalus suhelda ühe või mitme mõõtemooduliga korraga. Mõõtemoodul on võimeline genereerima ergutussignaali, teostama pingete mõõtmist ette ning suurema sünkroonsuse saavutamiseks kasutab mõõtetööga alustamiseks sünkroniseerimise juhet.

## Lahendatavad küsimused:

Kas mõõtemoodul on sobilik hajutatud andmehõiveks mitme tootega süsteemis? Kui suurt sünkroonsust on võimalik saavutada sünkroniseerimisjuhtmega? Kas mõõtemoodul on võimeline korraga genereerima ergutus-signaali ning teostama mõõtmisi?

Kuupäev: .....

Üliõpilane: .....

/allkiri/

Juhendaja: .....

/allkiri/

Kinnitaja: .....

/allkiri/

## **Abstract**

This work describes ARM based measurement module EK-TM4C123GXL, which is used for distributed data acquisitions. Goals of this thesis were to create communication functions for communication with controller in a network with up to 32 devices; data acquisition functions: generation of stimulation signal and voltage measurement; and synchronization via additional wire.

During the work all goals were achieved. In addition additional studies were carried out: delays caused by measurement module and its software, synchronization accuracy via special wire and simultaneously generating a stimulation signal and voltage measurements.

The thesis is in Estonian and contains **38** pages of text, **8** chapters, **20** figures and **9** tables.

## Resümee

Käesolev bakalaureusetöö käsitleb ARM põhise mõõtemoodulit EK-TM4C123GXL, mida kasutatakse hajutatud andmehõiveks. Töö eesmärgiks oli luua mõõtemoodulile kommunikatsioonifunktsioonid juhtkontrolleriga suhtlemiseks võrgus, kus on kokku kuni 32 seadet; andmehõive funktsioonid: ergutussignaali genereerimine ja pingete mõõtmine; ning sünkroniseerimise võimalus lisa juhtme abil.

Töö käigus sai täidetud kõik püstitatud eesmärgid ning uurida mõõtemooduli poolt tekitatud hilistumisi, sünkroonsuse täpsust lisa juhtme abil ning samaaegselt genereerida ergutussignaali ning teostada mõõtetööd.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti **38** leheküljel, **8** peatükki, **20** joonist ja **9** tabelit.

## Lühendite ja mõistete sõnastik

ARM – *Advanced RISC Machine*, kärbitud käsustikuga arvutiarhitektuur

CCS – Code Composer Studio

CMSIS - Cortex Microcontroller Software Interface Standard

CRC - *Cyclic Redundancy Check*, tsükkelkoodkontrolli

DMA – *Direct Memory Access*, Otse mälule juurdepääs

GPTM - *General-Purpose Timer Module*, Üldeesmärgiline taimer moodul

ID – identiteet

IDE - Integrated Development Environment - Integreeritud arenduskeskkond

PCB - *Printed Circuit Board*, trükkplaat

PIOSC - *Precision Internal Oscillator*, Sisemine täpsus ostsillaator

PPL - *Phase Locked Loop*, Faasilukklülitus

PWM – *Pulse-Width Modulation*, Impulss-laiusmodulatsioon

SRAM - *Static random-access memory*, staatiline juhupöördlusega mälu

UART - *Universal Asynchronous Receivers/Transmitters*, universaalne a-sünkroonne vastuvõtmine / saatmine.

VREFP - *Reference Positive Voltage*, Näidis positiivne pinge

VREFN - *Reference Negative Voltage*, Näidis negatiivne pinge

## Jooniste ja tabelite nimekiri

JOONIS 1. CRC ARVUTUSE KOOD. ....	16
JOONIS 2. KOMMUNIKATSIOONI LIIDESE TARKAVARA ALGORITM.....	17
JOONIS 3. PWM SIGNAALI GENERAATOR, SIGNAALI PERIOOD 12MS NING TÄITUVUS 33% .....	22
JOONIS 4. PWM GENEREERITUD SIGNAAL, SIGNAALI PERIOOD 4MS NING TÄITUVUS 0.1% .....	23
JOONIS 5. PWM GENEREERITUD SIGNAAL, SIGNAALI PIKKUS 404MS NING TÄITUVUS 50% .....	23
JOONIS 6. PWM GENEREERITUD SIGNAAL, SIGNAALI PIKKUS 40MS NING TÄITUVUS 50%.....	24
JOONIS 7. ADC-DE TEOSTUSE LIHTSUSTATUD JOONIS, [3, LK 800]. .....	25
JOONIS 8. ADC LIHTSUSTATUD MÕÕTE SKEEM, [3, LK 809]. .....	25
JOONIS 9. ADC ÜHELE MÕÕTMISELE KULUNUD AEG. ....	27
JOONIS 10. ADC ÜHELE MÕÕTMISELE KULUNUD AEG KOOS 40 MS HILISTUSEGA. ....	28
JOONIS 11. PROGRAMMI TARKVARALINE ALGORITM. ....	29
JOONIS 12. LOETUD PAKETI ANALÜÜSILE KULUNUD AEG, SISSELUGEMISEST KUNI ANALÜÜSI LÕPUNI. ....	30
JOONIS 13. AJALINE KULU CRC ARVUTUSEL. ....	31
JOONIS 14. AJALINE KULU PAKETI SISSE LUGEMISEST KUNI KÄSKLUSE TEOSTAMISE ALGUSENI.....	31
JOONIS 15. AJALINE KULU PAKETI KÄTTESAAMISEST KUNI ADC-NI KOOS PWM KÄIVITAMISEGA. ....	32
JOONIS 16. AJALINE KULU PAKETI KÄTTESAAMISEST KUNI ADC-NI ILMA PWM KÄIVITAMISETA. ....	32
JOONIS 17. AJALINE KULU: KATKESTUS SÜNKRONISEERIMISJUHTMES KUNI KATKESTUSE KÄSITLEJANI TARKVARAS.....	33
JOONIS 18. AJALINE KULU: KATKESTUS SÜNKRONISEERIMISJUHTMES KUNI PWM KÄIVITAMISE HETKENI .....	34
JOONIS 19. AJALINE KULU: KATKESTUS SÜNKRONISEERIMISJUHTMES KUNI ESIMISE MÕÕTMISENI ILMA PWM KÄIVITAMISETA. ....	34
JOONIS 20. AJALINE KULU: KATKESTUS SÜNKRONISEERIMISJUHTMES KUNI ESIMISE MÕÕTMISENI KOOS PWM KÄIVITAMISEGA. ....	35
TABEL 1. ID BAIDI BITTIDE KIRJELDUS. ....	13
TABEL 2. ID BAIDI VÄÄRTUSTE KIRJELDUS.....	13
TABEL 3. JUHTKONTROLLER - MÕÕTEMOODUL SUUNALISE KOMMUNIKATSIOONI KIRJELDUS. ....	13
TABEL 4. MÕÕTEMOODUL - JUHTKONTROLLER SUUNALISE KOMMUNIKATSIOONI KIRJELDUS. ....	14
TABEL 5. BAIDI ANDMETÜÜPI KIRJELDAVAD BITID. ....	14
TABEL 6. MÕÕTEMOODULI STAATUSE BAIDI BITTIDE KIRJELDUS. ....	14
TABEL 7. MUUDETAVAD KOMMUNIKATSIOONI LIIDESE TARKVARALISED PARAMEETRID.....	15
TABEL 8. MUUDETAVAD PWM SIGNAALI GENERAATORI TARKVARALISED PARAMEETRID. ....	21
TABEL 9. MUUDETAVAD ADC TARKVARALISED PARAMEETRID. ....	26

## Sisukord

Sisukord .....	8
Sissejuhatus .....	10
1. Arenduskeskkond .....	11
1.1. Arenduskeskkonna nõuded .....	11
1.2. Arenduskeskkonna ülesseadmine .....	11
2. Mõõtemooduli riistvara ülevaade .....	11
3. Mõõtemooduli sidekanal .....	11
3.1. Kommunikatsiooni liidese füüsilised ühendused .....	12
3.2. Sideprotokolli koostõlastus .....	12
3.2.1. Sideprotokolli nõuded .....	12
3.2.2. Sideprotokolli disain .....	13
3.3. Sidekanali liidese tarkvaraline lahendus .....	15
3.4. CRC arvutamine .....	16
3.5. Pakettide lugemine .....	17
3.6. Vigaste pakettide käsitlemine .....	18
3.7. Käsklustele käsitlemise protsessid .....	18
3.8. Sidekanali omadused .....	19
4. Sünkroniseerimine .....	19
4.1. Sünkroniseeritult tööga alustamine andmeside kaudu .....	19
4.2. Sünkroniseeritult tööga alustamine spetsiaalse juhtme abil .....	19
5. Ergutus signaali generaatori kasutamine – PWM .....	20
5.1. Signaali generaatori omadused .....	20
5.2. PWM mooduli füüsilise ühendus .....	20
5.3. Tarkvaraline lahendus .....	20
5.3.1. Genereeritava signaali pikkuse ( <i>load</i> ) arvutus .....	21
5.3.2. PWM-ga ilmnenu probleem .....	21
5.4. Genereeritud signaali kuju, täpsus ja näited .....	22
6. Analoo-digitaal-muundamine .....	25
6.1. ADC mõõteparameetrid / omadused .....	25



6.2. ADC füüsilise ühendus valimine.....	26
6.3. ADC mõõtetöö ajalise kulu mõõtmise põhimõte .....	26
6.4. ADC tarkvaraline lahendus .....	26
6.5. ADC mõõtesageduse muutmise võimalused.....	27
6.5.1. Erinevad sisend taktsageduse allika võimalused .....	27
6.5.2. Kahe ADC kasutamine .....	27
6.5.3. Mõõtesageduse seadistamine tarkvara abil.....	28
7. Programmi töö algoritm .....	29
8. Mõõtemooduli poolt põhjustatud viidete mõõtmine .....	30
8.1. Hilistumised sidekanali kaudu.....	30
8.2. Hilistumised spetsiaalse sünkroniseerimisejuhtmega .....	33
8.3. Viidete võrdlus .....	35
Kokkuvõte.....	36
Kasutatud kirjandus.....	37
Lisa 1. Loodud programmi kood.....	38

## Sissejuhatus

Bakalaureusetöö eesmärgiks on luua ARM platvormil baseeruvale mõõtemoodulile kommunikatsiooni- ja andmehõive tarkvaralised funktsionaalsused hajutatud andmehõiveks.

Suure kabariidiliste objektide mõõtmisel on sobiv kasutada sünkroonselt töötavaid mõõtemoduleid, mis paiknevad mõõtepunktidel võimalikult lähedal, ning juhtkontrollerit vähendamaks pikkadest juhtmetest tingitud probleeme. Uuemate kontrolleritega pakutakse mitmeid mõõtetööks vajalikke funktsionaalsusi, mis eemaldab vajaduse kasutada mitmeid erinevaid kontrollereid ühe mõõtetöö teostamiseks, ning suuremaid ja täpsemaid võimalusi erinevate parameetrite näol. Üheks selliseks näiteks on Texas Instruments poolt loodud TM4C123G LaunchPad.

Hajutatud andmehõivesüsteem koosneb ühest juhtkontrollerist ja kuni 31 mõõtemoodulist, mis on ühendatud juhtmetega toite, kommunikatsiooni ja sünkroniseerimise tarvis. Mõõtemoodul peab olema võimeline genereerima ergutussignaali, mõõtma pingeid ette antud sisenditelt, suhtlema juhtkontrolleriga ning sünkroniseerimisjuhtme abil alustama mõõtetööga. Kommunikatsioon toimub RS485 liidese kaudu, mille sideprotokoll võimaldab käskude saatmist ühele või mitmele mõõtemoodulile, mõõtetulemuste saatmist juhkontrollerile ning CRC kasutust.

## 1. Arenduskeskkond

### 1.1. Arenduskeskkonna nõuded

Peamiseks nõudeks on, et loodud tarkava peab olema kasutatav Tallinna Tehnika Ülikooli Thomas Johann Seebecki elektroonikainstituudi laborites kasutatava CCS nimelise IDE-ga. Kusjuures mõõtemoodul, EK-TM4C123GXL, vajab CCS versiooni 6 või uuem, varasematel versioonidel puudub tugi nimetatud mõõtemooduli kontrolleri- le.

### 1.2. Arenduskeskkonna ülesseadmine

Peamiseks komponendiks on arenduskeskkond, CCS, versiooniga 6 või uuem, mida on võimalik alla laadida Texas Instruments kodulehelt [5]. Lisaks IDE-le on vajalik paigaldada ka CMSIS failid [6]. Õige faili valimisel tuleb lähtuda mõõtemoodulil kasutatavast kontrolleri- st: EK-TM4C123GXL mõõtemoodul kasutab TM4C123GH6PM kontrollereid ning neile vastav TM4F osa number on LM4F230H5QR [7], ehk *header* fail LM4F230H5QR.h, mis võimaldab mõõtemooduli programmeerimist.

## 2. Mõõtemooduli riistvara ülevaade

EK-TM4C123GXL pakub kõiki vajalikke riistvaralisi funktsionaalsusi kommunikatsiooni- ning andmehõive funktsionaalsuste loomiseks. Kontrollereid on olemas kaheks UARTi, mida on väga hea kasutada koos RS-485 standardiga kommunikatsiooni otstarbel, kaks 12 bitist analoog-digitaal muunduri moodulit pingete mõõtmiseks ning kaks 16 bitist impulss-laius modulaatorit ergutussignaali genereerimiseks. Lisaks kuus 64 bitist ja kuus 32 bitist ajastit, mida saab kasutada aegumiste määramiseks ning ajastamiseks. Kontrolleri- ril on 32KB SRAM, mida on võimalik kasutada mõõtetulemuste salvestamiseks. SRAM suurus piirab ära maksimaalse mõõtetulemuste arvu, mida on võimalik korraga talletada mõõtemoodulis.

## 3. Mõõtemooduli sidekanal

Kommunikatsioon on planeeritud toimuma üle RS-485. Omavahel on ühendatud kuni 32 seadet, millest üks on juhtkontroller ning 31 on mõõtemoodulid. Kõik seadmed on lähte olekus kuulaja rollis, *slave*.

Juhtkontroller võib vahetada *slave* olekust *master* olekusse suvalisel sobival hetkel kuid mõõtemoodulid tohivad vahetada *slave* olekust *master* olekusse üksnes juhul kui

juhtkontroller on seda lubanud mõõtetulemuste edastamiseks juhtkontrollerile ning vahetama tagasi *salve* olekusse nii pea kui tulemused on edastatud.

Andmete maht, suunal juhtkontroller - mõõtemoodul, on üldiselt väike, seda kasutatakse üksnes sünkroniseerimiseks, mõõtmiste alustamiseks ning andmete küsimiseks, ehk on kindlalt paika pandud ja muutuvaid andmeid ei esine. Andmete maht vastupidises suunas on suurem, sest sellisel juhul edastatakse mitmeid mõõtetulemusi.

### **3.1. Kommunikatsiooni liidese füüsilised ühendused**

Mõõtemoodulil on kokku kaheksa UART-i, mida on väga hea kasutada koos kommunikatsiooni standardiga RS485[2], millest sobivaima viikude asetusega on UART1 ja UART5 [4, lk 9] viigud, sest need asetsevad vahetus-läheduses toite viikudele. Kasutusele sai võetud UART5 viigud, PE4 ja PE5.

RS485 mooduli vastuvõtu ja väljundi režiimi vahel vahetamiseks on vaja kasutada lisa viiku, algselt sai kasutusele võetud PB4 [4, lk 9], kuna see asetseb valitud UART5 viikude kõrval, PE5-st järgmine. Töö käigus selgus, et kui kasutada PWM mooduliga samas plokis asetsevaid viike, siis PWM võib tekitada soovimatuid katkestusi. Sellest tulenevalt sai kasutusele võtta PD1 [4, lk9].

### **3.2. Sideprotokolli kooskõlastus**

#### **3.2.1. Sideprotokolli nõuded**

- Juhtkontroller peab saama suhelda ühe kindla mõõtemooduliga korraga – unikaalse identifitseerimise numbriga kasutamise kuni 32 tootega võrgus.
- Juhtkontroller peab saama kõikide mõõtemoodulitega korraga suhelda.
- CRC kasutus vigade tuvastamiseks.
- Suhtlus, suunal juhtkontroller-mõõtemoodul, on fikseeritud.
- Suhtlus, suunal mõõtemoodul-juhtkontroller, on muutuva sisuga.
- Paketi pikkus 4 baiti, mis jaguneb vastavalt:
  - 1. bait: Unikaalne ID
  - 2-3 bait: Juhtkontrolleri käsk või mõõtemooduli mõõtetulemused
  - 4. bait: CRC

### 3.2.2. Sideprotokolli disain

Kommunikatsioon mõlemas suunas on ühesuguse paketi suurusega: 4 baiti, kus esimene ja viimane bait on alati kindla sisuga, unikaalne ID ning CRC kood. Esimene bait on kas ühe kindla mõõtemooduli ID, suheldakse ühe kindla mõõtemooduliga, või NULL bait, suheldakse kõikide mõõtemoodulitega korraga, vaata Tabel 1. ID baidi bittide kirjeldus. ja Tabel 2. ID baidi väärtuste kirjeldus. Viimane bait, CRC kood, arvutatakse ja lisatakse viimaseks andme paketi baitiks. Iga sisse loetud paketi puhul kontrollitakse sisu vastavust CRC-ga. Mõõtemooduli poolt väljastatud pakettide ID bait on alati mõõtemooduli enda ID.

2. ja 3. baidi sisu on kirjeldatud Tabel 3. Juhtkontroller - mõõtemoodul suunalise kommunikatsiooni kirjeldus. ja Tabel 4. Mõõtemoodul - juhtkontroller suunalise kommunikatsiooni kirjeldus.

Biti järk	Kirjeldus
0-4	Unikaalne ID.
5-7	Ei kasutata.

Tabel 1. ID baidi bittide kirjeldus.

ID	Kirjeldus
0	NULL bait, paketi sisu mõeldud kõikidele pakettidele.
1 – 31	Unikaalne ID, mis on omane ainult ühele kindlale mõõtemoodulile.

Tabel 2. ID baidi väärtuste kirjeldus.

2. bait	Kirjeldus	3. bait	Kirjeldus
0	Ei kasutata.	0 – 255	Ei kasutata.
1	Alustada mõõtetööga: ergutussignaali ja mõõtmine.	0 – 255	Ei kasutata.
2	Mõõtmistega alustamine.	0 – 255	Ei kasutata.
3	Ergutussignaali genereerimise algus.	0 – 255	Ei kasutata.
4	Ergutussignaali genereerimise lõpp.	0-255	Ei kasutata
5	Saata mõõtetulemused juhkontrollerile.	0 – 255	Ei kasutata.
6	Kustutada mõõtetulemused.	0 – 255	Ei kasutata.
7 – 254	Ei kasutata.	0 – 255	Ei kasutata.
255	Staatuse päring.	0	Ei kasutata.

Tabel 3. Juhtkontroller - mõõtemoodul suunalise kommunikatsiooni kirjeldus.

2. bait	3. bait	Kirjeldus
0 – 255	0 – 255	Mõõtetulemused.

Tabel 4. Mõõtemoodul - juhtkontroller suunalise kommunikatsiooni kirjeldus.

Kuna mõõtetöös kasutatakse 12 bitist analoog-digitaal muundurit ning iga mõõtetulemus on 12 bitine, siis mõõtetulemuste tagastamisel tuleb iga tulemus tagastada kahes osas, mis on jagatud 2. ja 3. baidi vahel võrdselt. Kusjuures kaks suurimat järku bitti igas baidis määrab ära, mis tüüpi andmetega on tegu, mille kirjelduse leiab Tabel 5. Baidi andmetüüpi kirjeldavad bitid.

Bitid	Kirjeldus
00** ****	Andmed on lõplikud, koosneb kuuest väikseimast bitist.
01** ****	Andmete algus, esimene osa saadetavast infost.
10** ****	Andmed jätkuvad, vahepealne osa saadetavast infost.
11** ****	Andmete lõpp, viimane osa saadetavast infost.

Tabel 5. Baidi andmetüüpi kirjeldavad bitid.

Mõõtemooduli hetke staatuse pärimisel korral vastab mõõtemoodulpaketiga, mille sisu on järgmine:

mõõtemooduli ID | staatuse päringu bait (255) | staatuse kood, vt. Tabel 6 | CRC

Staatuse bitid	Kirjeldus
0000 0001	Valmis mõõtetööga alustama: ergutussignaali, mõõtmise või mõlemad.
0000 0010	Mõõtetulemused olemas.
0000 0100	Mõõtetulemused saadetud.
0000 1000	Ei kasutata.
0001 0000	
0010 0000	
0100 0000	
1000 0000	Viimane päring oli vigane.

Tabel 6. Mõõtemooduli staatuse baidi bittide kirjeldus.

### 3.3. Sidekanali liidese tarkvaraline lahendus

Kommunikatsiooni liides on loodud paindlikkust ja võimalust seadistada vastavalt vajadustele silmas pidades. Kõik muudetavad parameetrid, vt. Tabel 7. Muudetavad kommunikatsiooni liidese tarkvaralised parameetrid, on võimalik muuta failis *ARM\_measurement\_unit.h*. Saatumise ja vastuvõtu režiimi vahetamisel on soovitatav kasutada väikest ooteaega veendumaks, et andmeid said edukalt saadetud ja et vajadusel on teine osapool jõudnud ümber lülitada.

Parameeter	Algväärtus	Kirjeldus
ID	1 – 31	Mõõtemooduli unikaalne ID.
GLOBAL_ID	0	Üldine ID, kasutatakse kõikide moodulitega suhtlemiseks.
PACKAGE_SIZE	4	Loetava paketi pikkus, baitide arv pakendis.
CRC_INCLUDED	1	Viimane bait on CRC.
CRC_BIT_CHECK_FOR_FLIP	128	Kontroll bit, mille alusel inverteeritakse CRC arvutamisel teisi bitte.
CRC_BIT_FLIP	7	Bitid, mida CRC arvutamisel inverteeritakse.
BAUD_RATE	1000000	<i>Baud-rate</i> .
INTEGER_BRD	1	<i>Baud-rate</i> täisosa.
FLOAT_BRD	0	<i>Baud-rate</i> murdosad.
RS485_RECEIVE	0	RS485 mooduli vastuvõtu režiim, madal või kõrge.
UART_SWITCHING_DELAY	10	Ajaline hilistus, ms, kui lülitatakse RS485 olekute vahet.
COMMUNICATION_TIMEOUT	16'000'000	Kommunikatsiooni aegumine, kus 16'000'000 on 1 sekund.

Tabel 7. Muudetavad kommunikatsiooni liidese tarkvaralised parameetrid.

### 3.4. CRC arvutamine

CRC-d kasutamise eesmärk on kommunikatsiooni pakettides vigade tuvastamine [1], kuid mitte nende parandamine. CRC kood arvutatakse üle kõigi baitide, kuid välja arvatud viimase baidi, milleks on CRC ise. Iga kord, kui arvutatakse baidilt CRC, siis antakse arvutusele varasemalt arvutatud CRC, mis on esimese baidi arvutamisel null, ning vastav bait, millel arvutatakse. Baidilt CRC arvutamisel nihutatakse baidi sisu vasakule tsüklis kaheksa korda, mis vastab bittide arvule baidis, ning juhul kui kontroll biti väärtus enne nihutamist oli 1, siis pärast nihutamist invertteeritakse invertteeritavad bitid. Vaata Joonis 1. CRC arvutuse kood.. Kontroll bit ning invertteeritavad bitid on seadistatavad vastavalt vajadusele, vt. Tabel 7. Muudetavad kommunikatsiooni liidese tarkvaralised parameetrid.

```
for (i = 0; i < (PACKAGE_SIZE - CRC_BYTE_COUNT); i++){
    for ( i = 0; i < 8; i++ ){
        if (( crc & CRC_BIT_CHECK_FOR_FLIP ) != 0 ){
            crc <<= 1;
            crc ^= CRC_BIT_FLIP;
        } else {
            crc <<= 1;
        }
    }
}
return crc;
```

Joonis 1. CRC arvutuse kood.



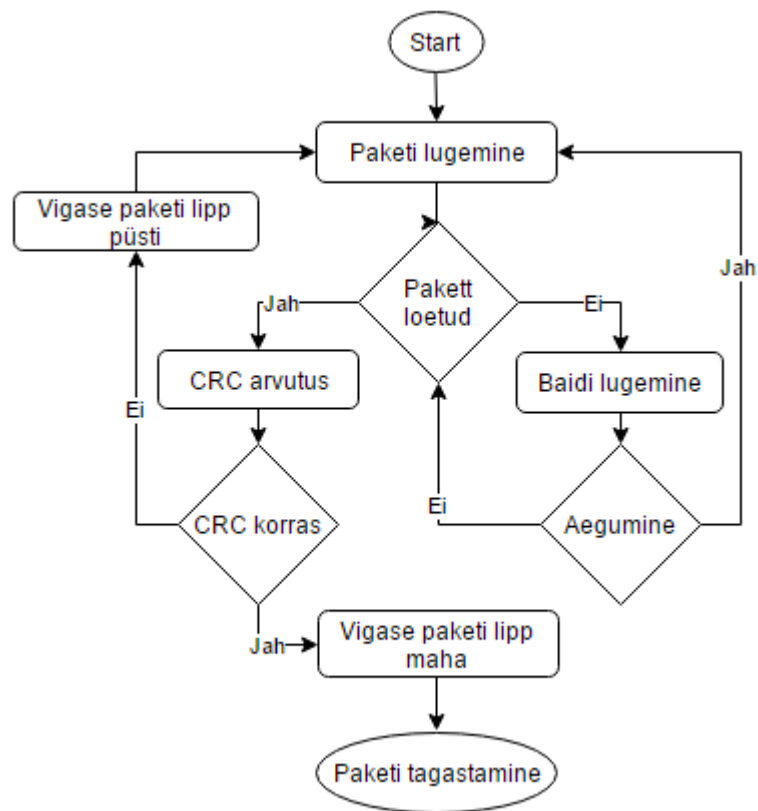
### 3.5. Pakettide lugemine

Pakette loetakse sisse üks bait korraga kuni pakett on täis või kuni uute baitide vastuvõtt on aegunud, vt. Tabel 7. Pakettide aegumise kontrolli kasutatakse üksnes 2. kuni viimase baidi sisse lugemisel, vt. Joonis 2. Kommunikatsiooni liidese tarkavara algoritm.

Edukalt sisse loetud paketi sisult, välja arvatud CRC baidid, arvutatakse CRC ning kontrollitakse arvatud tulemust sisse loetud CRC koodiga. Vigade mitteesinemise korral loetakse pakett korrektseks ning tagastatakse pakett programmi koodi edasiseks tööks.

Paketi lugemise aegumise tuvastamiseks on kasutatud katkestusi ehk *interrupts*. Aegumise katkeskus on väikesemahuline: pannakse aegumise lipp püsti ning puhastatakse katkestuse lipp. Kommunikatsiooni juures see lisakoormust otseselt ei tekita, sest katkestus teostatakse mõne taktiga.

Vigane pakett või paketi lugemise aegumise korral visatakse pakett ära ning hakatakse uut paketti lugema.



Joonis 2. Kommunikatsiooni liidese tarkavara algoritm.

### 3.6. Vigaste pakettide käsitlemine

Vastu võetud vigaseid pakette, paketi sisult arvatud CRC kood ja sisse loetud CRC koodid ei ole võrdsed, ignoreeritakse ning pannakse lipp üles, et viimane saadud päring oli vigane. Uue korrektse päringu saamisel eemaldatakse vigase päringu lipp. Juhul kui uus korrektne päring oli staatuse päring, siis mõõtemoodul saadab hetke staatust kirjeldava vastuse, vt. Tabel 6, mis on kirjeldatud punktis □ Staatuse pärimise protsess mõõtemoodulilt.

Mõõtemoodulil endal puudub teave, kas saadetud paketid jõudsid korrektselt sihtpunkti või mitte. Vea esinemise korral peab juhkontroller saatma uuesti välja vastava sisusuga paketi, mida eelnevalt päriti.

### 3.7. Käsklustele käsitlemise protsessid

- **Staatuse pärimise protsess mõõtemoodulilt**

1. Juhkontroller saadab staatuse päringu mõõtemoodulile.
2. Mõõtemoodul saadab hetke staatust kirjeldava vastuse, vt. Tabel 6.
3. Juhul kui „Viimane päring oli vigane“ lipp oli püsti, siis see eemaldatakse.

- **Mõõtetööga alustamise protsess**

1. Mõõtemoodul initsialiseerib mõõtetööks vajalikud moodulid: ADC ja/või PWM.
2. Juhkontroller saadab käskluse mõõtetööga alustamiseks.
3. Mõõtemoodul alustab koheselt mõõtetööga.

- **Mõõtetulemuste pärimise protsess**

1. Juhkontroller saadab päringu mõõtemoodulile.
2. Mõõte moodul saadab mõõtetulemused juhkontrollerile, üks mõõtetulemus korraga.
3. Juhkontroller veendub, et saadud andmed on korras, vastasel juhul minnakse tagasi punkti 1.

- **Tulemuste kustutamise protsess**

1. Juhkontroller teavitab mõõtemoodulit, et andmed võib kustutada.
2. Mõõtemoodul kustutab mõõtetulemused.
3. Mõõtemoodul seab üles lipu, et mõõtetulemused on kustutatud.
4. Kustutatud mõõtetulemuste lipp eemaldatakse vahetult enne järgmisi mõõtmistoid.

### **3.8. Sidekanali omadused**

Sidekanali modulatsioonikiirustena sai peamiselt kasutada üld-levinuid kiiruseid 9.6kbit/s ning 115.2kbit/s ning lisaks katsetada 1Mbit/s. RS-485 andmeedastuskiirus sõltub kommunikatsiooni juhtmete pikkustest, kus põidla reegel on, et andmeedastuskiiruse (bit/sec.) ja juhtme pikkuse (meetrites) korrutis ei tohiks ületada  $10^8$ . Laboratoorses tingimustes kasutatakse juhtmeid pikkusega kuni 50m ehk andmeedastuskiirus ei tohiks olla suurem kui 2Mbit/s. Mõõtemooduli suurim lubatud kiirus tuleneb süsteemi kellast ning võimalikust väikseimast IBRD väärtusest, milleks võib olla 1 [3, lk 914]. Sellest lähtudes võib olla mõõtemooduli suurimaks lubatud andmeedastuskiiruseks, süsteemi kella kiirusega 16MHz, 1Mbit/s [3, lk 903, BRD arvutus] ning kommunikatsiooni juhtmete pikkuseks sellel kiirusel kuni 100m, mis rahuldab igati laboratoorseid nõudeid.

## **4. Sünkroniseerimine**

### **4.1. Sünkroniseeritult tööga alustamine andmeside kaudu**

Mõõtetööga alustamist on võimalik juhtida kommunikatsiooni kanali kaudu, vt. Tabel 3. Juhtkontroller - mõõtemoodul suunalise kommunikatsiooni kirjeldus. Antud variandi juures ei ole tagatud parim sünkroniseeritus ning ajastatus, kuna see eeldab juhtkontrolleri poolelt vastavasisulise paketi väljasaatmist ning mõõtemooduli poolel sisse loetud pakettide töölust ning vastavate otsuste tegemist. Parem on kasutada spetsiaalset juhhet, vt. 4.2 Sünkroniseeritult tööga alustamine spetsiaalse juhtme abil, millisel juhul aga puudub kontroll üksiku mõõtemooduli üle.

### **4.2. Sünkroniseeritult tööga alustamine spetsiaalse juhtme abil**

Parima võimaliku sünkroniseeritud täpsuse saavutamiseks on vajalik kasutada spetsiaalset juhhet, mille abil anda märku mõõtetöö alustamiseks.

Sünkroniseerimisjuhtme viik peab jääma kommunikatsiooni kanalite vahetuslähedusse, kuna tegu on ühe juhtmega mitmest, mis ühendatakse mõõtemooduliga.

Sobivaima asetusega viigud on need viigud, mis jäävad kommunikatsiooni ja toite viikude vahetuslähedusse, näiteks D0, PB1, PA5 ning teised kõrval olevad viigud. Kasutusele sai esialgu võetud PD0, kuid hilisemal tarkvara loomisel sai PD0 välja vahetatud PE1 vastu.

Sünkroniseerimisjuhtme tarkvaralisel lahendusel sai kasutusele võetud katkestused ehk *interrupts*, sest nende kasutamisel on võimalik suvalisel hetkel katkestada normaalne programmi töö ning käivitada soovitud programmi kood. Katkestuste kasutamiseks tuli lisada

käivitamise koodi (*startup code*) katkestuste käsitleja (*Interrupt Handler*), mille peab hiljem defineerida koodis sobivas kohas.

## **5. Ergutus signaali generaatori kasutamine – PWM**

### **5.1. Signaali generaatori omadused**

Mõõtemoodulil on kaks PWM moodulit, kummalgi neli PWM generaatori plokki ning kella kontroll plokk, kokku 16 väljundiga, kus iga PWM generaatoriga on seotud kaks väljundit [3, lk 1230]. Kõik generaatorid on 16 bitised. Mõlema generaatori mooduli sisendkellaks on võimalik eraldiseisvalt seadistada süsteemi kella või selle jagatist. Süsteemi kella jagamine võimaldab pikemate signaali perioodide genereerimist. Genereeritava signaali keskmine pingeline jääb vahemikku 0V kuni 3.3V, mis sõltub genereeritud signaali täituvusest.

Genereeritava signaali sagedus sõltub sisend taktsagedusest, taktsageduse jagajast ning määratud perioodi pikkusest. Kiireim genereeritav signaal on pool sisend taktsagedusest, kus on kasutatud väikseimat sisend taktsageduse jagajat (võimalikud väärtused on 2 aste, kus astendaja on 0 kuni 5) ning lühimat võimalikku perioodi (võimalikud väärtused on vahemikus 1 kuni 65535). Aeglaseim sagedus on võimalik saavutada kui sisend taktsagedus jagata suurima taktsageduse jagajaga ning pikima võimaliku perioodiga. Genereeritava signaali täituvust on võimalik määrata 16 bit arvuga, 1-st kuni ette antud signaali pikkuseni.

### **5.2. PWM mooduli füüsilise ühendus**

Füüsilise liidese asukoha valimisel oli nõue, et signaali genereerimise viik asuks eemal kommunikatsiooni liidese viikudest, lihtsustamaks lisaraudvara skeemi loomist ning üldist kasutuse mugavust. Kriteeriumi eelduseks on, et kommunikatsiooni liidese ja signaali generaatori liidese viike on võimalik viia erinevatele viikude ridadele.

PWM moodulil on kokku 16 viiku, millest sobivaima asetusega, viikude tulpades J4 ja J2, on järgmised: PB6, PB7, PC4, PC5, PF0, PF2, PF3; millest kasutusele sai võetud PB6. Välistatud said PF0, sest selle kasutus vajab lisaks eraldi lahtilukustamist, ning PF2 ja PF3, sest need on PCB plaadil oleva RGB LED-ga ühendatud.

### **5.3. Tarkvaraline lahendus**

Genereeritava signaali pikkust ning täituvust on võimalik muuta failis *ARM\_measurement\_unit.h*, vt. Tabel 8. Muudetavad PWM signaali generaatori tarkvaralised

parameetrid. Tulevikus on võimalik lisada PWM väljundviike ning kasutusele võtta ka teine PWM moodul enamate ja sõltumatute signaalide genereerimiseks.

Parameeter	Algväärtus	Kirjeldus
PWM_LOAD	1 – 65535	Genereeritava signaali pikkus.
PWM_WIDTH	0 – PWM_LOAD	Genereeritava signaali täituvus.
USE_CLOCK_DIVIDER	0-1	Kasutada kella jagaja või mitte.
CLOCK_DIVIDER	0-7	Süsteemi kella jagaja. Vt. [4, lk 255].

Tabel 8. Muudetavad PWM signaali generaatori tarkvaralised parameetrid.

### 5.3.1. Genereeritava signaali pikkuse (*load*) arvutus

Genereeritava signaali pikkuse arvutus sõltub süsteemi kella jagatisest, kui jagamist ei kasutata, siis süsteemi kellast, kusjuures kella jagaja võimalikud väärtused on 2, 4, 8, 16, 32 või 64 [3, lk 255]. Suurim kasutatav pikkus on 16bit arv ehk 65535.

Väljundsignaali sageduse arvutus:

$$\frac{\text{Süsteemi kella kiirus Hz}}{\text{Signaali pikkus, 16 bit arv} * \text{kella jagaja}} = \text{Väljundsagedus}$$

Teisendades antud valmit, on võimalik leida vajalikku signaali pikkust. Kusjuures signaali pikkuse arv mis saab olla täisarv ning olla vahemikus 1 kuni 65535. Valem:

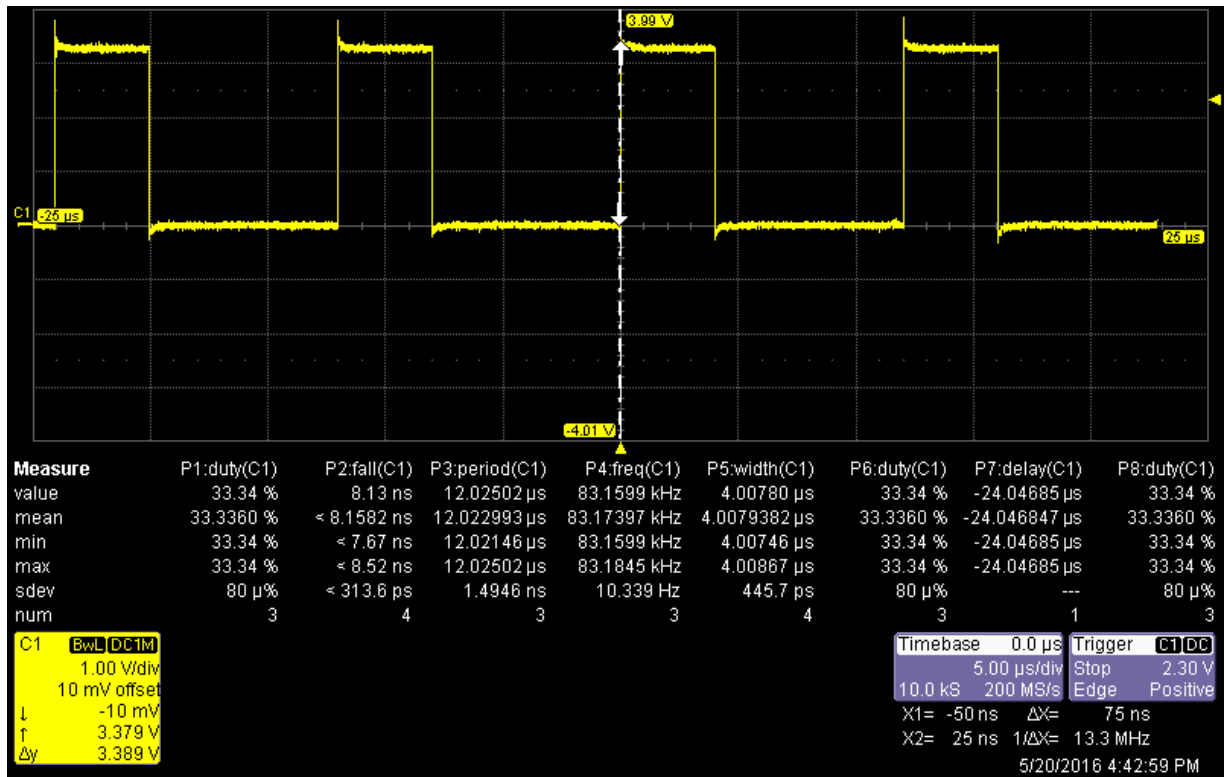
$$\frac{\text{Süsteemi kella kiirus Hz}}{\text{Väljundsagedus} * \text{kella jagaja}} = \text{Signaali pikkus, 16 bit arv}$$

### 5.3.2. PWM-ga ilmnenud probleem

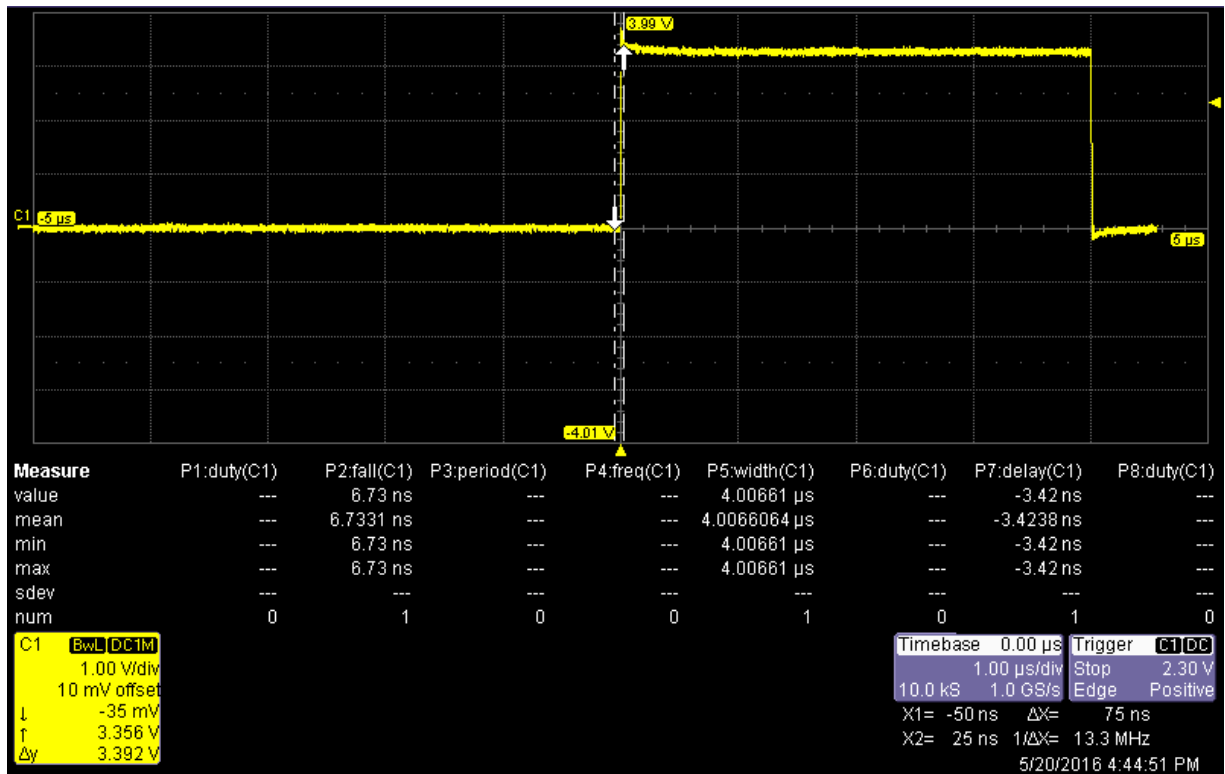
PWM raudvaralisest lahendusest tulenevalt selgus, et kasutatud PWM0 generaator genereerib signaali ka PD0 viigule, mis oli kasutuses sünkroniseerimise eesmärgil. See probleem sai tuvastatud lõpetuseks laboris ostilloskoobi abiga. Tarkvaralisel poolel ei suutnud tuvastada ühtegi viga, mis sarnast nähtust võiks põhjustada. Lahendusena sai muuta sünkroniseerimisviik, milleks sai valitud viik PE1. Põhjus, miks teisi PWM väljundviike kasutusele ei võtnud, seisneb selles, et enamus PWM0 generaatori viigud asuvad just J2 ja J4 viikude tulbas ning see võimaldab tulevikus enamate PWM väljundite kasutust.

## 5.4. Genereeritud signaali kuju, täpsus ja näited

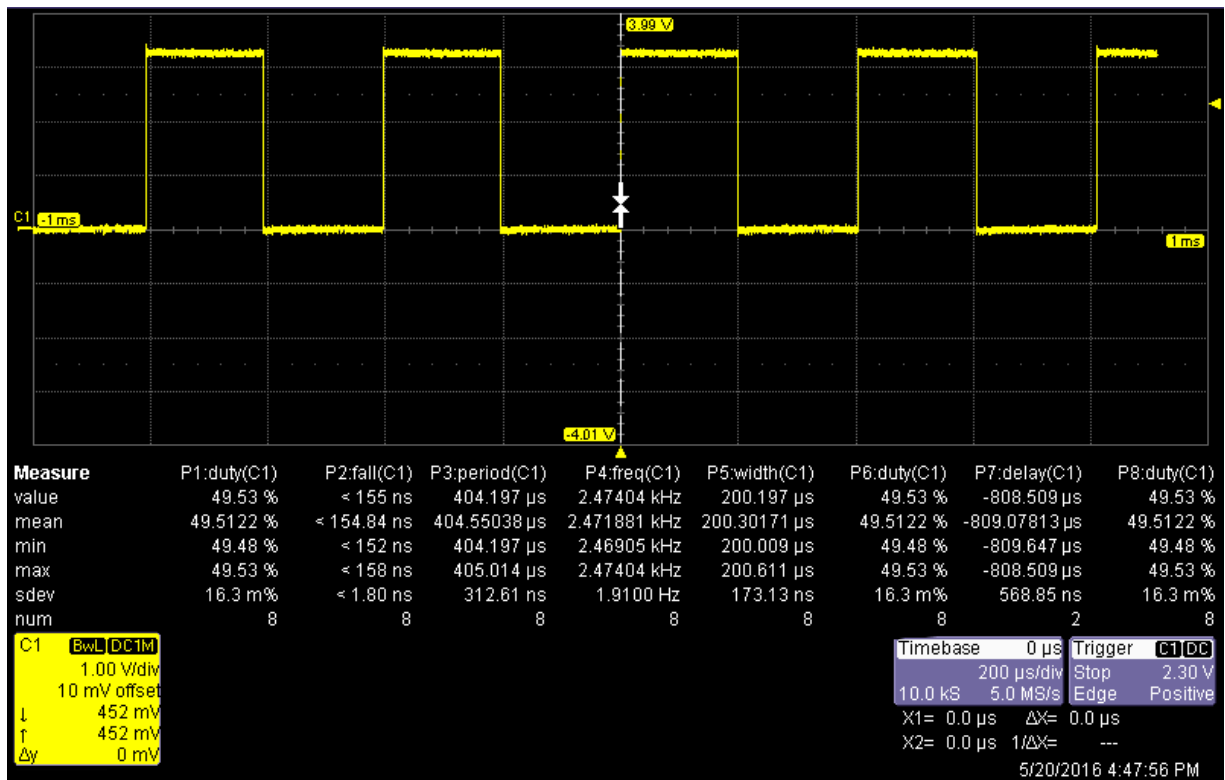
PWM generaatori ühe takti pikkus on umbes 4 mikrosekundit, vt. Joonis 3. PWM signaali generaator, signaali periood  $12\mu\text{s}$  ning täituvus 33% ja Joonis 4. PWM genereeritud signaal, signaali periood  $4\text{ms}$  ning täituvus 0.1%, kui sisend kella ei jagata. Täituvuse täpsuses esineb väikene viga, lühemate genereeritud signaalide puhul vähe suurem kui pikemate signaalide puhul, vt Joonis 5. PWM genereeritud signaal, signaali pikkus  $404\mu\text{s}$  ning täituvus 50% ja Joonis 6. PWM genereeritud signaal, signaali pikkus  $40\text{ms}$  ning täituvus 50%.



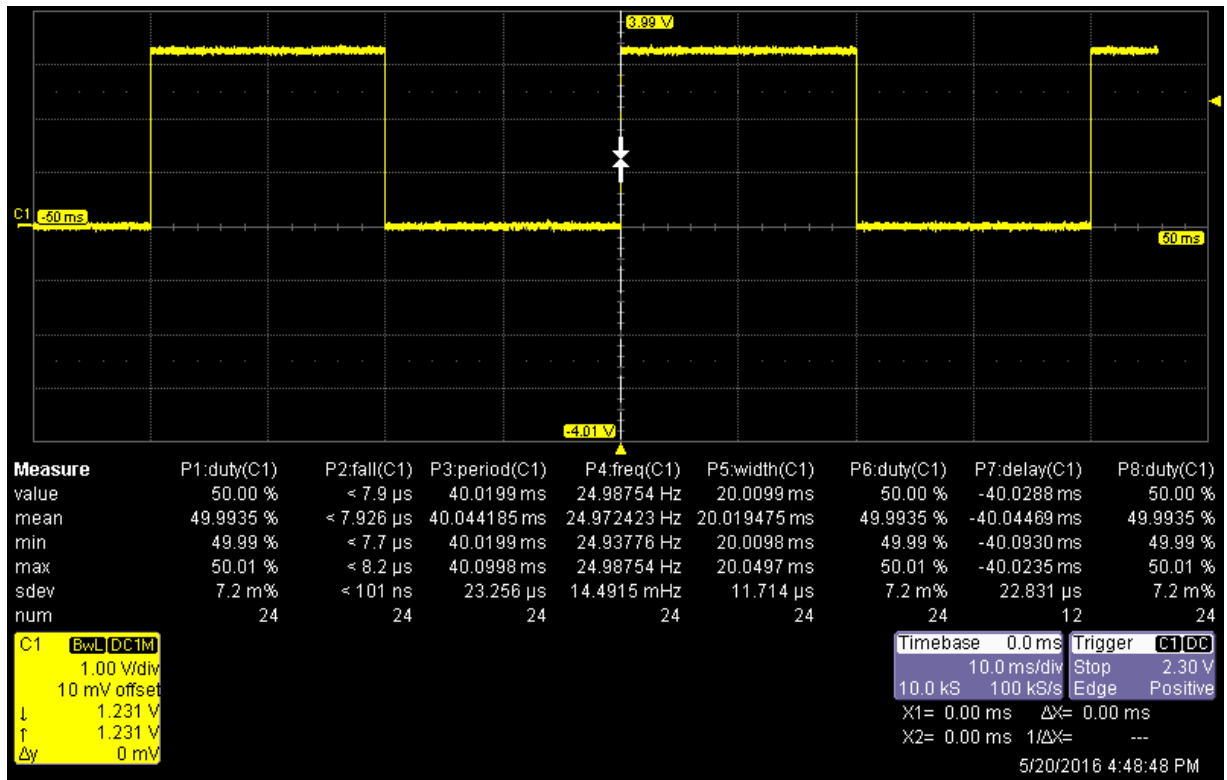
Joonis 3. PWM signaali generaator, signaali periood  $12\mu\text{s}$  ning täituvus 33%



Joonis 4. PWM genereeritud signaal, signaali periood 4ms ning täituvus 0.1%



Joonis 5. PWM genereeritud signaal, signaali pikkus 404μs ning täituvus 50%



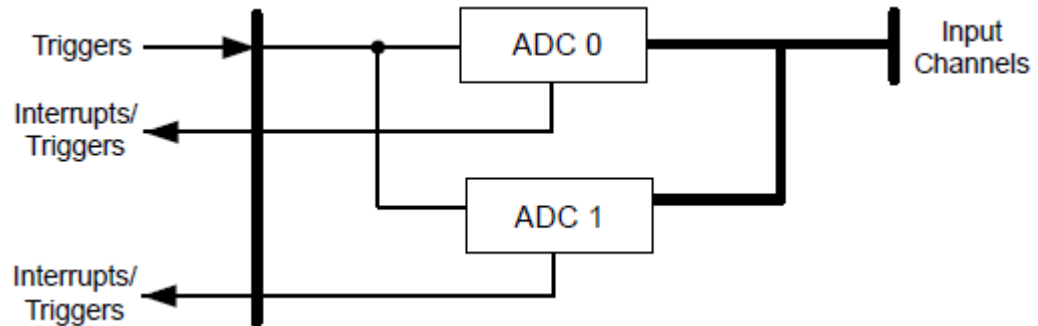
Joonis 6. PWM genereeritud signaal, signaali pikkus 40ms ning täituvus 50%



## 6. Analoo-digitaal-muundamine

### 6.1. ADC mõõteparameetrid / omadused

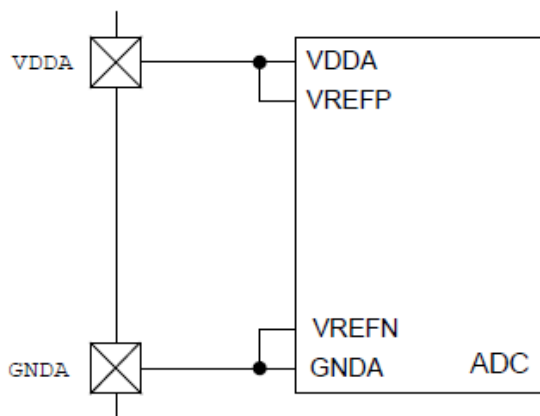
Mõõtemoodulil on kaks analoo-digitaal-muundurit (edaspidi ADC), ADC0 ja ADC1, mis töötavad eraldiseisvalt, kuid kasutavad ühiseid sisendkanaleid [3, lk. 800], vt Joonis 7. ADC-de teostuse lihtsustatud joonis, [3, lk 800]. Kokku on ADC-1 12 sisendkanalit [3, lk 801 tabel 13-1], mida on võimalik kasutada ühiselt või eraldi seisvalt. Suurim võimalik mõõtesagedus on 1M mõõtmist sekundis.



Joonis 7. ADC-de teostuse lihtsustatud joonis, [3, lk 800].

Tegu on 12 bitise ADC-ga, mis kasutab sisemisi signaale VREFP ja VREFN näidistena, et arvutada kindel väärtus valitud sisendkanalile [3, lk 808], kus VREFP on ühendatud VDDA-ga ja VREFN on ühendatud GNDA-ga, vt Joonis 8. ADC lihtsustatud mõõte skeem, [3, lk 809]., ehk mõõtevahemik on 0V kuni 3.3V. Täpse mõõtetulemuse saamiseks peab mõõdetav signaal jääma lubatud piiridesse, mis on 0 kuni VDDA, vt. [3, lk 1389].

Mõõtesammu suurus on arvutatav vastavalt valemile [3, lk 809]:  
 $(VREFP - VREFN) / 4096 = (3,3V - 0V) / 4096 = 3,3V / 4096 \approx 0,806 \text{ mV}$  iga ADC koodi kohta.



Joonis 8. ADC lihtsustatud mõõte skeem, [3, lk 809].

## 6.2. ADC füüsilise ühendus valimine

Füüsilise liidese asukoha valimise nõue oli, et mõõtmiste viik asuks eemal kommunikatsiooni liidese viikudest, lihtsustamaks lisa raudvara skeemi loomist ning üldist kasutuse mugavust. Kriteeriumi eelduseks on, et kommunikatsiooni liidese ja mõõtemooduli liidese viike on võimalik viia erinevatele viikude ridadele.

ADC-1 on kokku 12 sisendkanalilt, millest PE0 ainsana asub eraldi viikude reas (J2) [4, lk 10], mis sai ka kasutusele võetud. Lisaks, PE0 viigul ei ole võimalik ühtegi teist lisa funktsionaalsust pordi kontrolli kaudu määrata.

## 6.3. ADC mõõtetöö ajalise kulu mõõtmise põhimõte

ADC mõõtetöö ajalise kulu mõõtmiseks sai kasutada punase LED biti lülitamist. Punane LED lülitati sisse kui algas mõõtetöö. Punane LED lülitati välja kui lõppes mõõtetöö. Kõik mõõtetulemused sisaldavad endas ka punase LED-i biti lülitust, mille ajaline pikkus on määramata.

## 6.4. ADC tarkvaraline lahendus

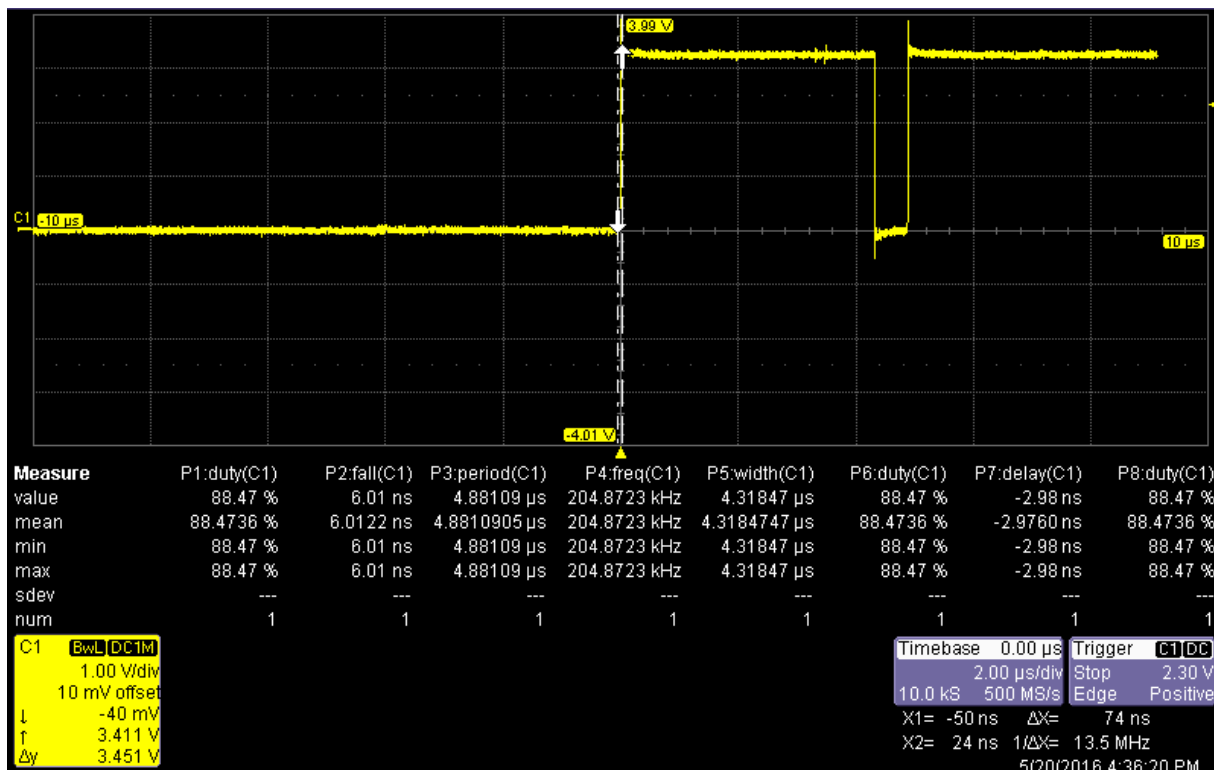
ADC initsialiseeritakse koheselt programmi käimaminemisel ning programmi koodis, täpsemalt *ARM\_measurement\_unit.h* failis, on võimalik ära määratleda, mitu mõõtmist teostatakse, hilistumisi mõõtmiste vahel ning võimalust üle kirjutada saatmata mõõtetulemusi, vt Tabel 9. Muudetavad ADC tarkvaralised parameetrid.

Parameeter	Algväärtus	Kirjeldus
SAMPLE_COUNT	1000	Mõõtemiste arv
SAMPLE_DELAY	0	Hilistus mõõtemiste vahel
OVERWRITE_OLD_RESULTS	0	Mõõtetulemusi mitte üle kirjutada kui varasemad ei ole saadetud.

Tabel 9. Muudetavad ADC tarkvaralised parameetrid.

Mõõtmist ei ole võimalik kommunikatsiooni kanali kaudu ega katkestustega peatada.

Iga mõõtetöö tsükkel, ilma soovitud lisa tarkvaralise viiteta, vt. 6.5.3 Mõõtesageduse vähendamine tarkvara abil, teostatakse 4.8  $\mu$ s jooksul, vt Joonis 9. ADC ühele mõõtmisele kulunud aeg., millest 4.3  $\mu$ s võtab endale reaalne analoog-digitaal muundamine ning mõõtetulemuse salvestamine ning umbes 0.5  $\mu$ s loendurite suurendamine ja kontroll.



Joonis 9. ADC ühele mõõtmisele kulunud aeg.

## 6.5. ADC mõõtesageduse muutmise võimalused

### 6.5.1. Erinevad sisend taktsageduse allika võimalused

ADC suurim töö kiirus on kuni 1M mõõtmist sekundis, mis on mõjutatud ADC mooduli sisend kellast:

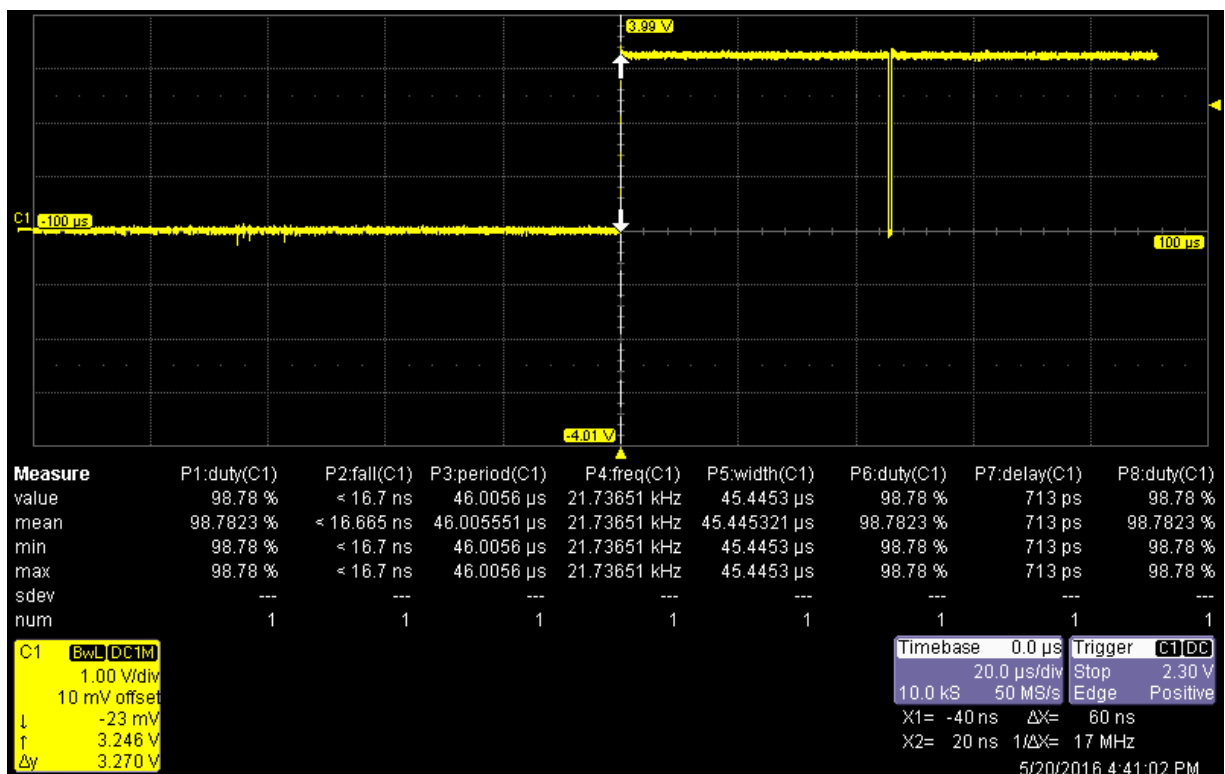
- Süsteemi kella järgi (vähemalt 16MHz) või tuletatuna PPL-st, mis on jagatud 25-ga [3, lk 892].
- PIOSC-i poolt genereeritud 16MHz taktsagedus [3, lk 892].

### 6.5.2. Kahe ADC kasutamine

Kahte eraldiseisvat ADC-d on võimalik korraga kasutada erinevate või samade sisendite mõõtmiseks [3, lk 804] ning lisaks on võimalik määrata mõõtmise faasi, sammuga  $22.5^\circ$  ning vahemikus  $0^\circ$  kuni  $337.5^\circ$  [3, lk 804]. Kui kasutada kahte ADC-d sama sisendi mõõtmiseks ning kahe ADC vaheline mõõtmiste vahe on  $180^\circ$ , siis sedasi on võimalik tõsta mõõtmiste arvu tihendust kahekordseks.

### 6.5.3. Mõõtesageduse seadistamine tarkvara abil

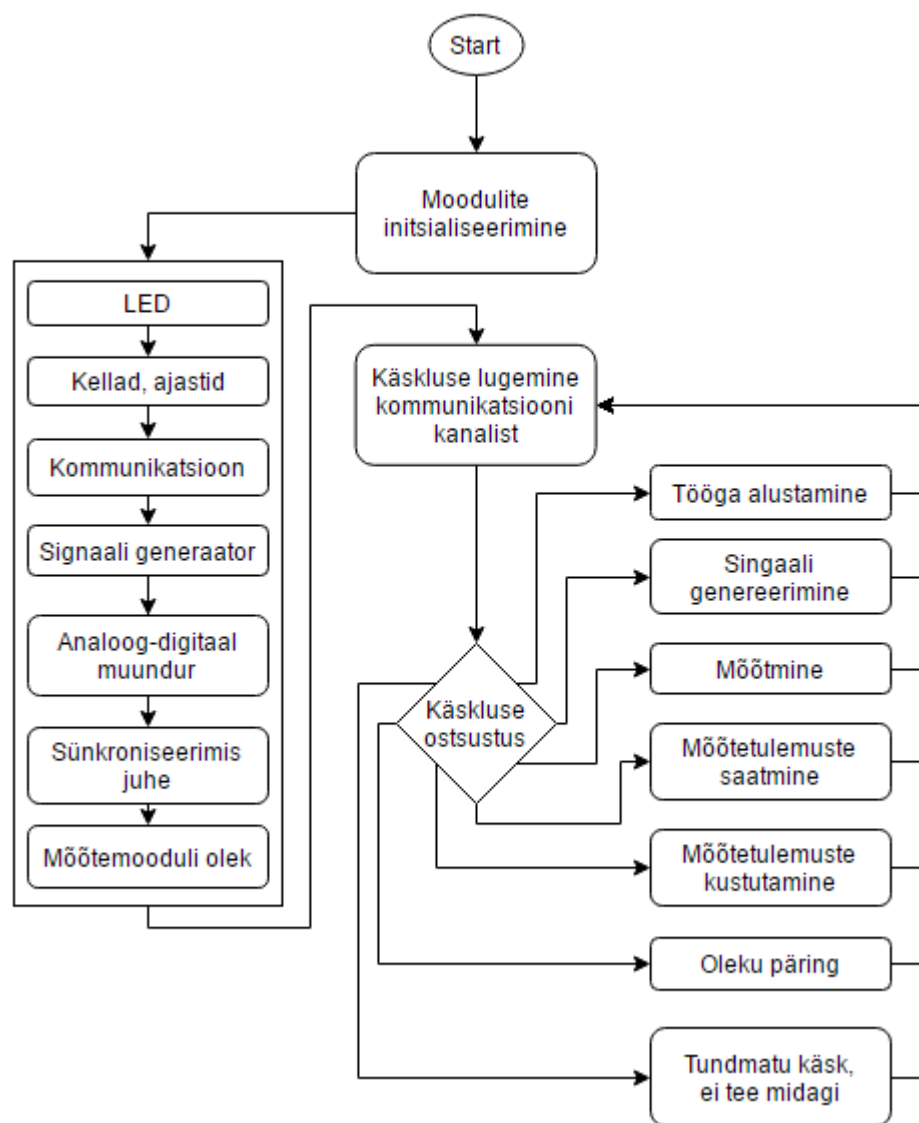
ADC töötab 16MHz sagedusel ning sellest tulenevalt on võimalik mõõte sagedust muuta üksnes tarkvaraliselt. Hilistumise teostamiseks on kasutatud mõõtemooduli GPTM-i [3, lk 704], mis töötab perioodiliselt, ning hilistumist teostatakse iga mõõtesammu vahel, mille pikkus on määratav failis *ARM\_measurement\_unit.h*, vt. Tabel 9. Muudetavad ADC tarkvaralised parameetrid. Hilistumise suurusel piirangud puuduvad, vähim võimalik hilistus on 0, vt Joonis 9. ADC ühele mõõtmisele kulunud aeg. ja Joonis 10. ADC ühele mõõtmisele kulunud aeg koos 40  $\mu$ s hilistusega.



Joonis 10. ADC ühele mõõtmisele kulunud aeg koos 40  $\mu$ s hilistusega.

## 7. Programmi töö algoritm

Programm töötab lõputus tsüklis. Programmi käivitamisel initsialiseeritakse mooduli tööks vajalikud osad: kommunikatsioon, LED, kellad ja ajastid, signaali generaator, ADC, PWM ning sünkroniseerimise juhtme sisend. Edasi jääb programm jälgima kommunikatsiooni kanalit edasiste käskluste lugemiseks. Käskluse saades teostatakse vastav tegevus ning pöörduakse tagasi kommunikatsiooni kanalit jälgima. Vaata Joonis 11. Programmi tarkvaraline algoritm.



Joonis 11. Programmi tarkvaraline algoritm.

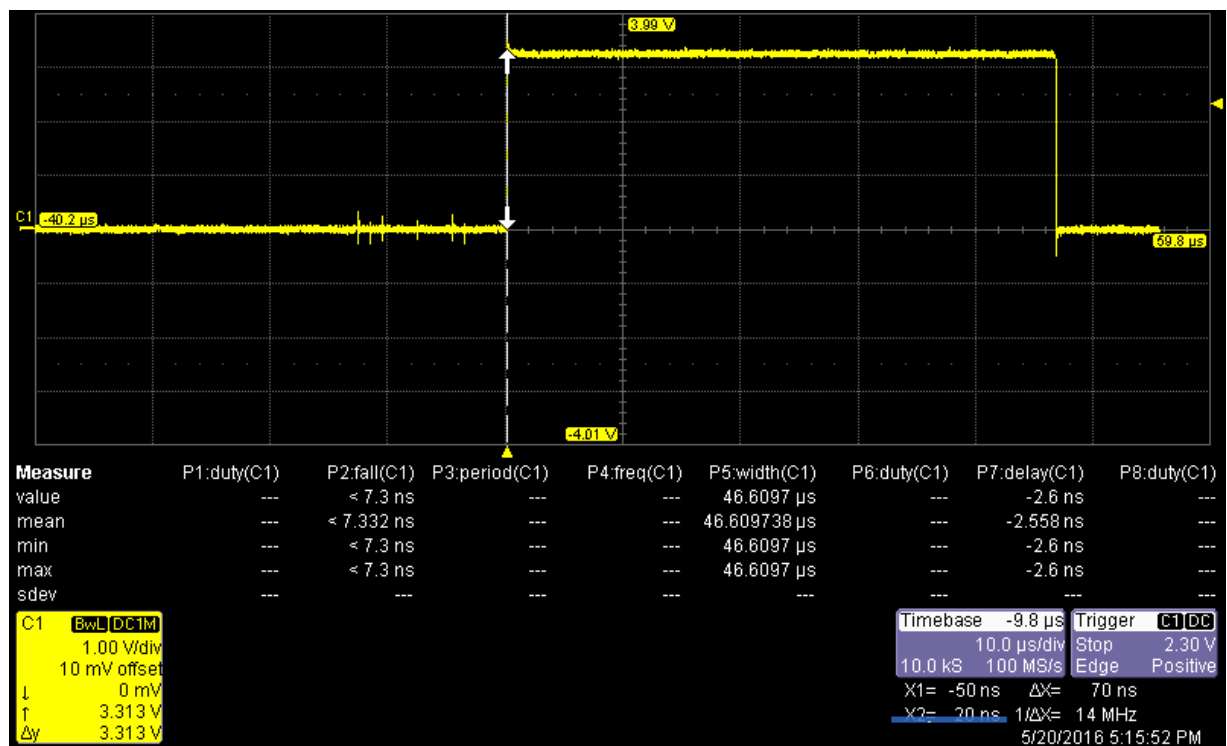
## 8. Mõõtemooduli poolt põhjustatud viidete mõõtmine

Viidete mõõtmise määratlemiseks sai kasutada punase LED biti lülitamist. Tõusev nivoo määrab mõõtmise alguse ja langev nivoo mõõtmise lõpu. Tulemused sisaldavad punase LED biti lülitamist, mille ajaline kulu on määratlemata.

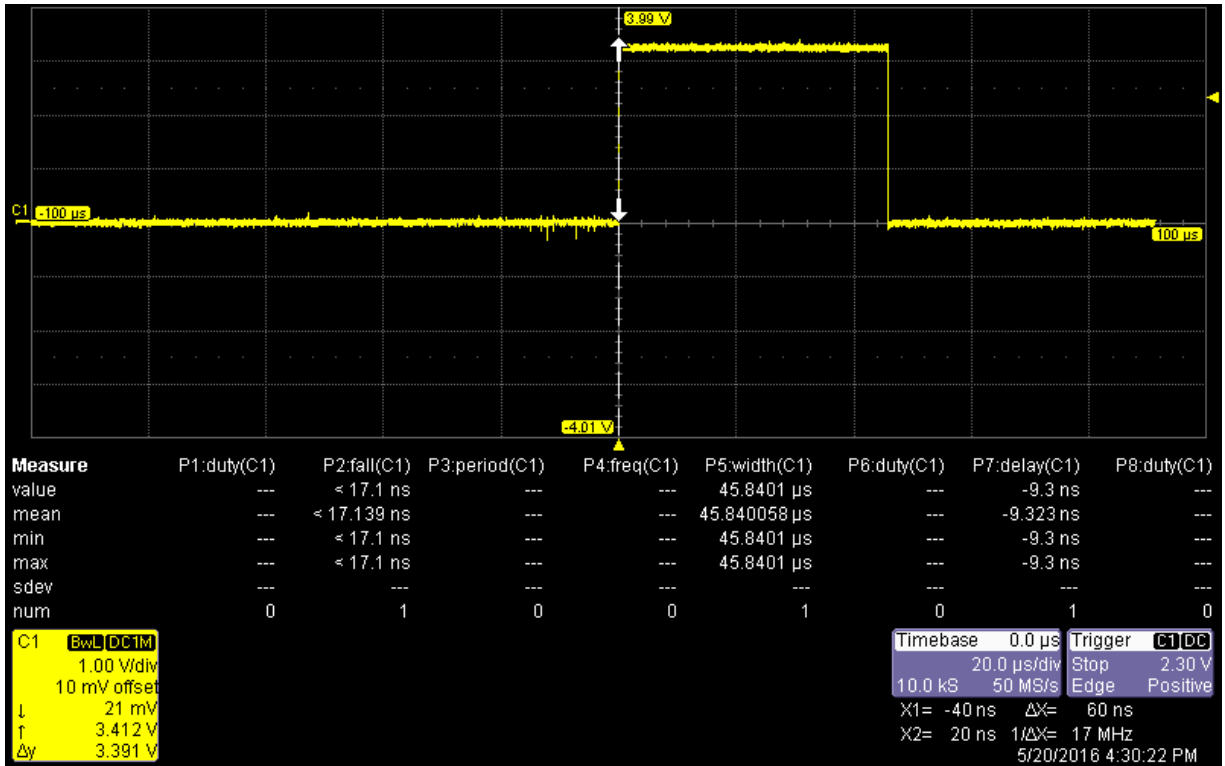
### 8.1. Hilistumised sidekanali kaudu

Iga sisse loetud pakett analüüsitakse, kas saadud paketti peab kuulama konkreetsele mõõtemoodulile ning CRC korrektsus, 47us jooksul, vt Joonis 12. Loetud paketi analüüsile kulunud aeg, sisselugemisest kuni analüüsi lõpuni. CRC koodi kontroll moodustab peamise ajakulu, umbes 45,8  $\mu$ s vt. Joonis 13. Ajaline kulu CRC arvutusel. Korrektse paketi korral teostatakse vastav käsklus, milleni jõudmine võtab aega umbes 1.3 $\mu$ s, vt Joonis 14. Ajaline kulu paketi sisse lugemisest kuni käskluse teostamise alguseni.

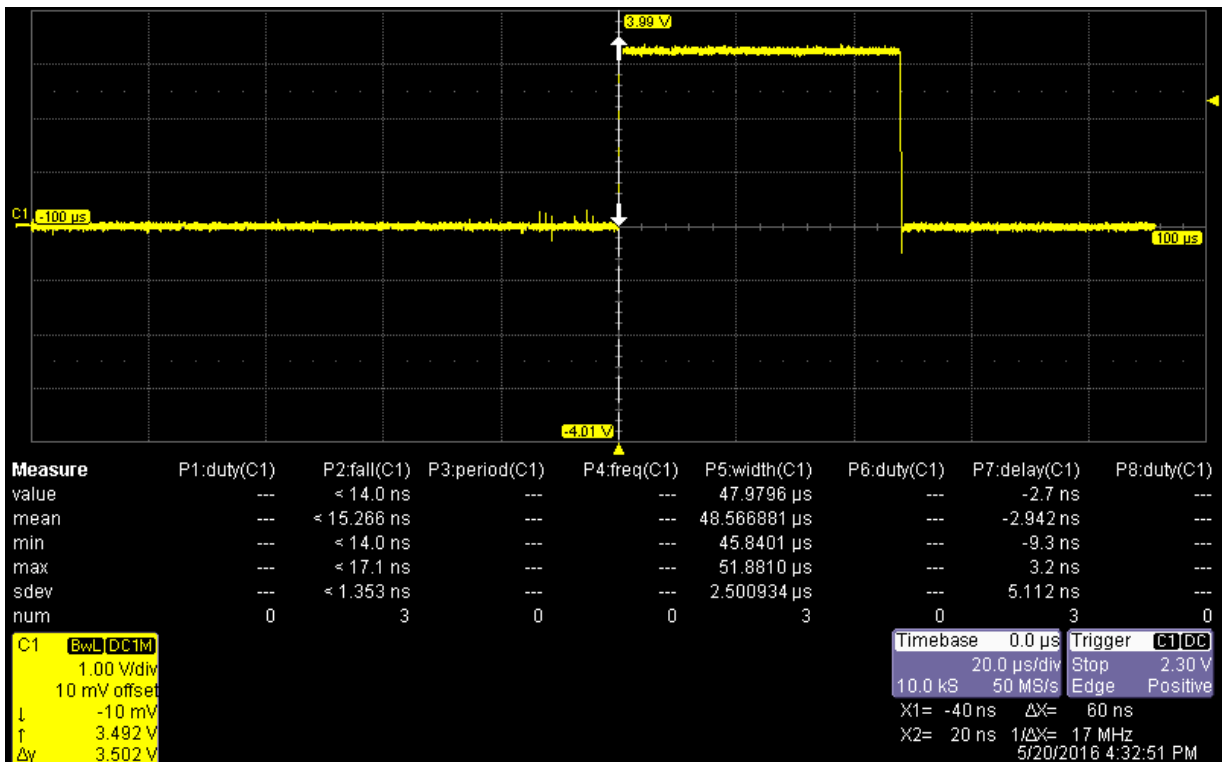
Juhul kui kasutada PWM-i, mis käivitatakse vahetult enne mõõtetöö algust, ja ADC-d, siis esimene mõõtmine teostatakse 51,9  $\mu$ s pärast paketi kättesaamist, vt. Joonis 15 Joonis 16, lisa ajaline kulu tuleneb, 2.8 $\mu$ s, on tingitud muutujate lähtestamisest mõõtefunktsioonis. Ajaline kulu paketi sisse lugemisest kuni esimese mõõtmiseni ilma PWM käivitamiseta on 50,7  $\mu$ s, vt. Joonis 16 Joonis 15.



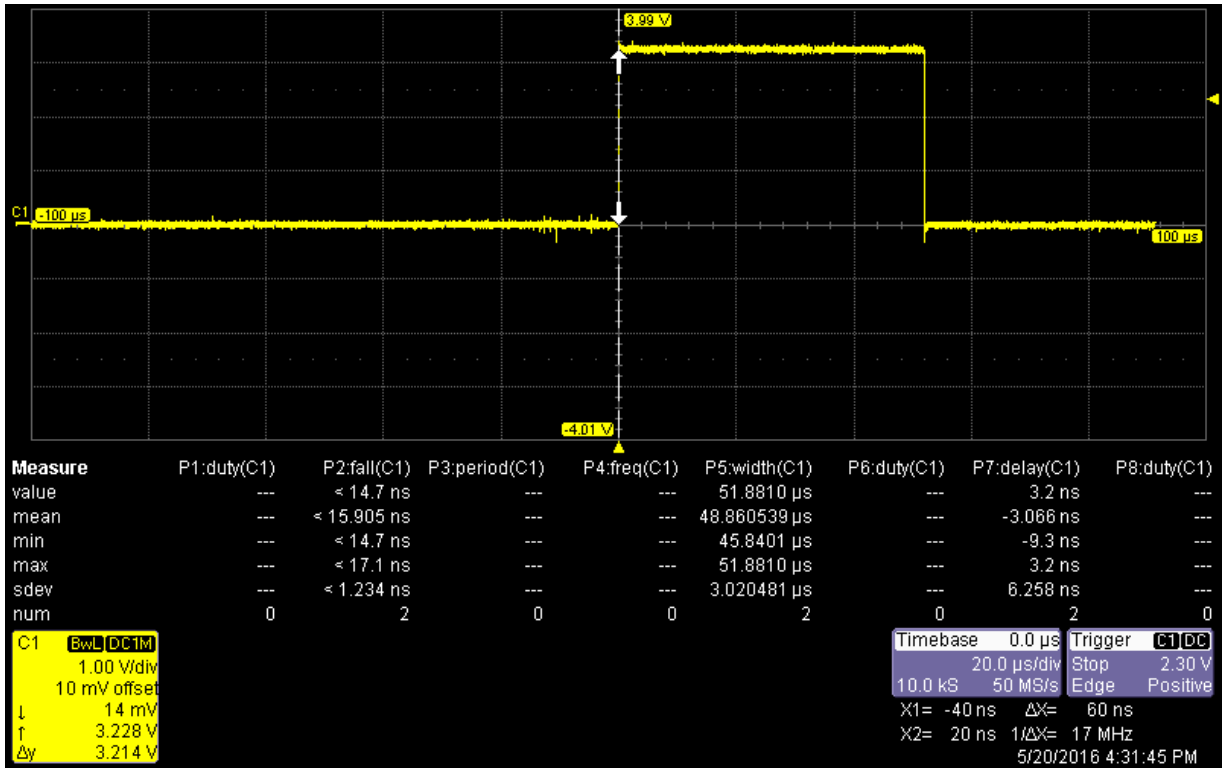
Joonis 12. Loetud paketi analüüsile kulunud aeg, sisselugemisest kuni analüüsi lõpuni.



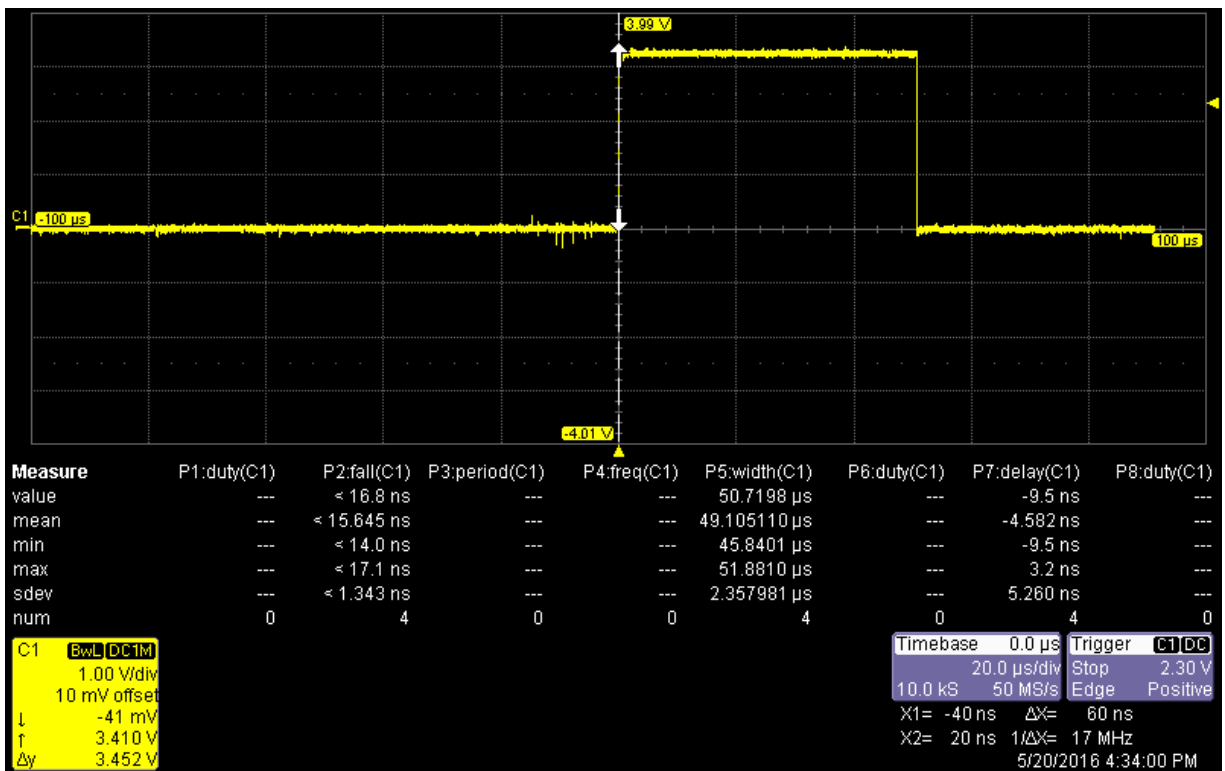
Joonis 13. Ajaline kulu CRC arvutusel.



Joonis 14. Ajaline kulu paketi sisse lugemisest kuni käskluse teostamise alguseni.



Joonis 15. Ajaline kulu paketi kättesaamisest kuni ADC-ni koos PWM käivitamisega.



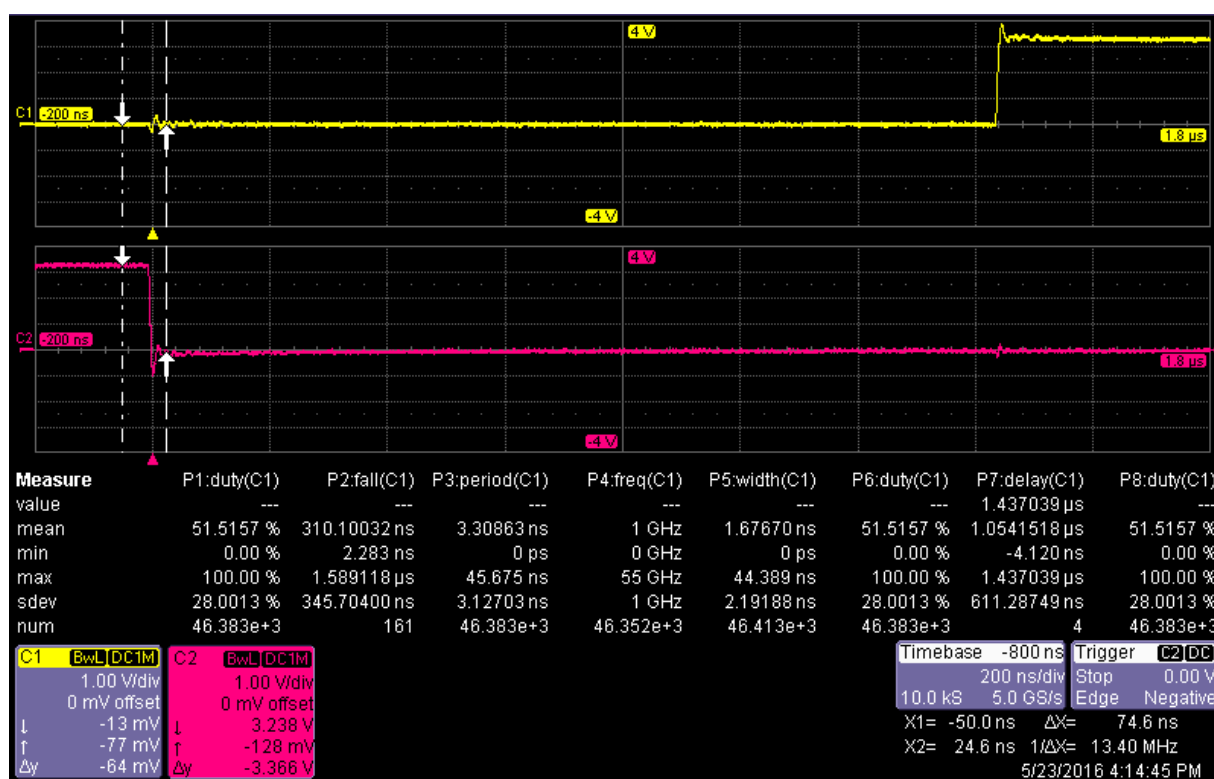
Joonis 16. Ajaline kulu paketi kättesaamisest kuni ADC-ni ilma PWM käivitamiseta.



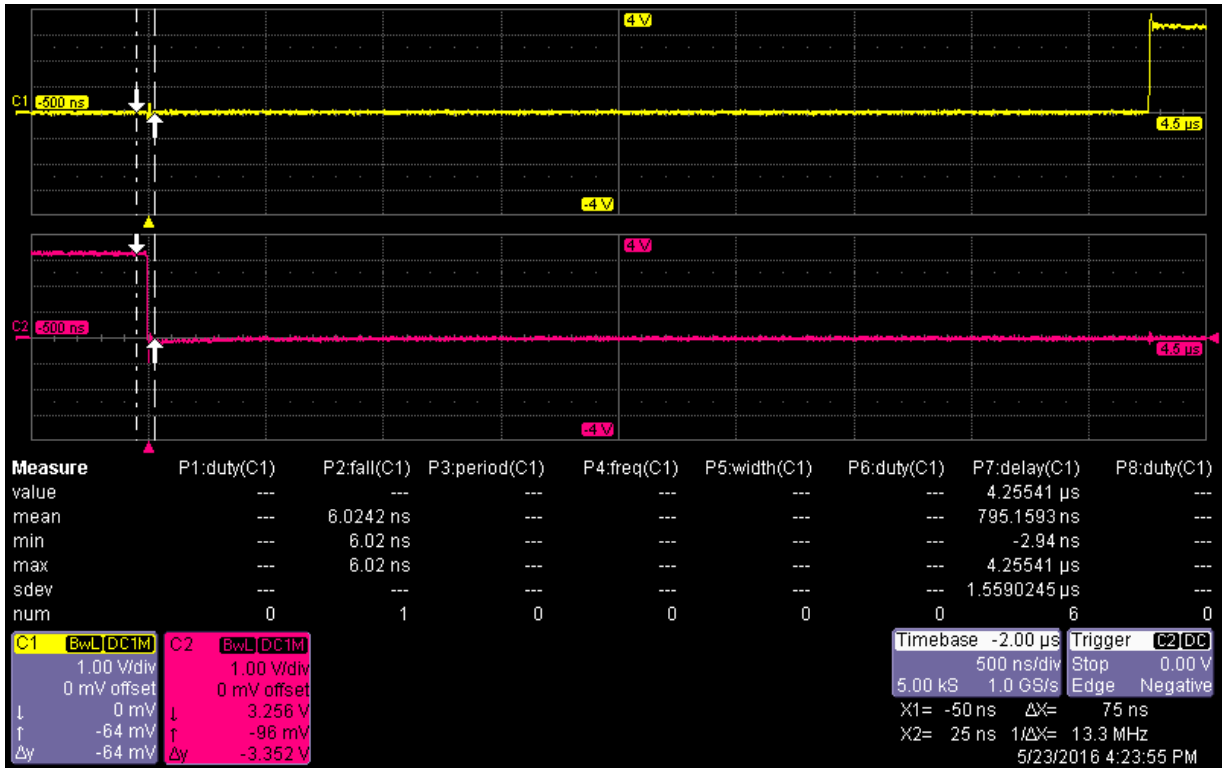
## 8.2. Hilistumised spetsiaalse sünkroniseerimisejuhtmega

Sünkroniseerimisjuhtme kasutamisel viidete mõõtmiseks sai kasutada punase LED biti lülitamist, mis lülitati kõrgeks kui jõuti soovitud sihtkohta, ning lisaks teist ostsilloskoobi kanalit sünkroniseerimisjuhtme jälgimiseks, millel sai kasutada langeva fronti päästikut. Kõik mõõtetulemused sisaldavad endas punase LED biti lülitust, mille ajaline kulu on määratlemata.

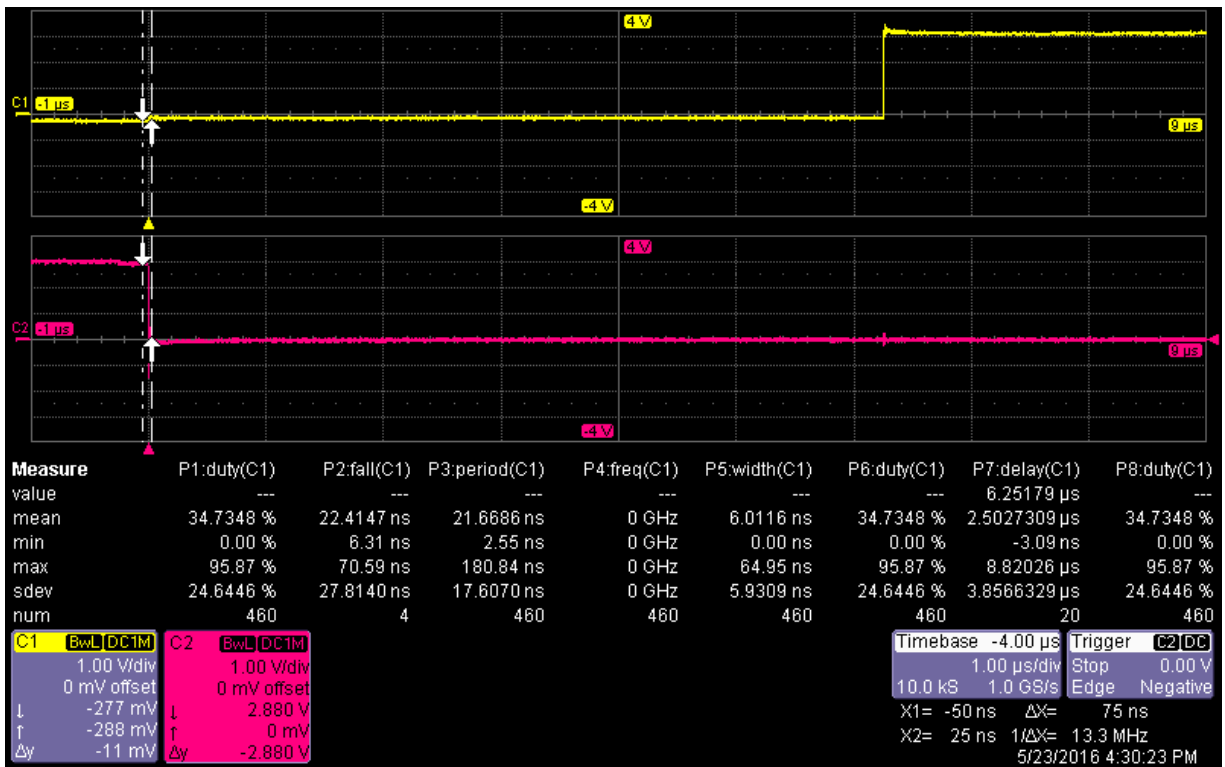
Katkestuse hetkest sünkroniseerimisejuhtmelt kuni katkestuse käsitlejani (*interrupt handler*) tarkvaras võttis 1.4  $\mu\text{s}$ , vt. Joonis 17; PWM mooduli käivitamiseni kulus 4.3  $\mu\text{s}$ , vt. Joonis 18; esimese mõõtmiseni ilma PWM mooduli käivitamiseta kulus 6.3  $\mu\text{s}$ , vt. Joonis 19; ning esimese mõõtmiseni koos PWM mooduli käivitamisega kulus 8.7  $\mu\text{s}$ , vt. Joonis 20.



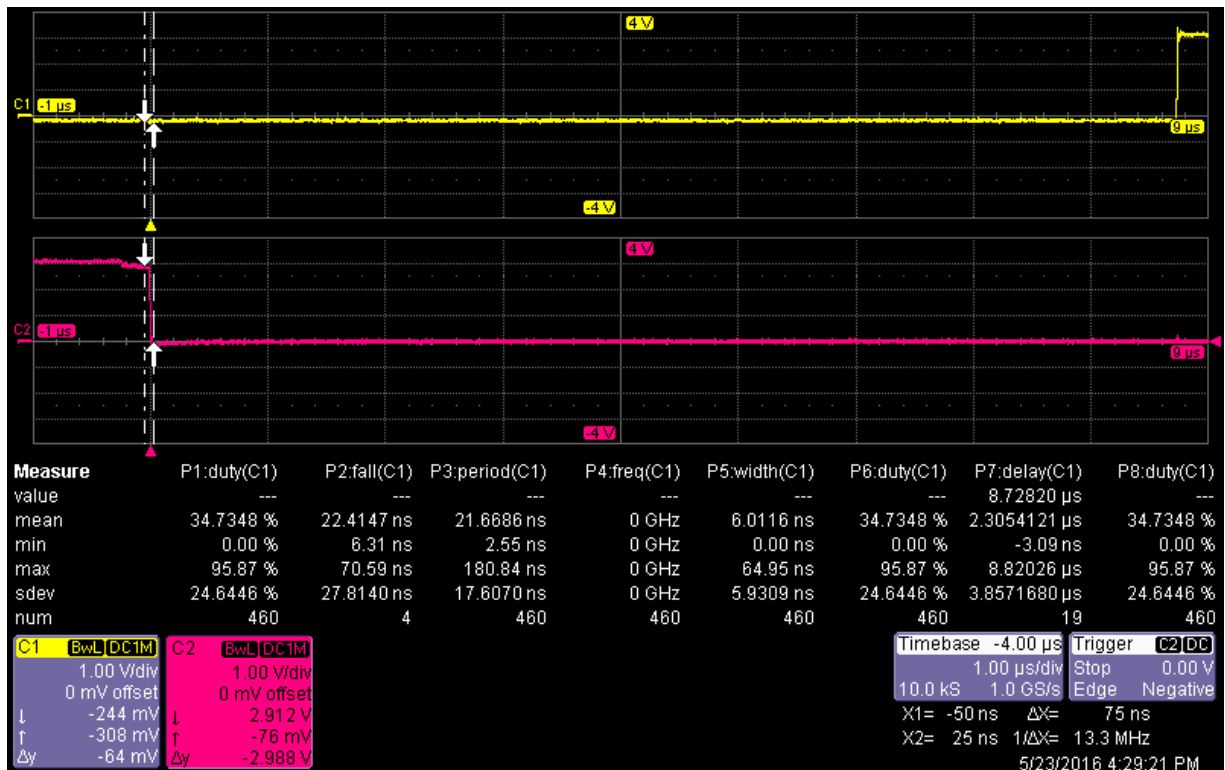
Joonis 17. Ajaline kulu: katkestus sünkroniseerimisjuhtmes kuni katkestuse käsitlejani tarkvaras



Joonis 18. Ajaline kulu: katkestus sünkroniseerimisjuhtmes kuni PWM käivitamise hetkeni



Joonis 19. Ajaline kulu: katkestus sünkroniseerimisjuhtmes kuni esimese mõõtmiseni ilma PWM käivitamiseta.



Joonis 20. Ajaline kulu: katkestus sünkroniseerimisjuhtmes kuni esimese mõõtmiseni koos PWM käivitamisega.

### 8.3. Viidete võrdlus

Hilistumised sünkroniseerimisjuhtme abil on märkimisväärselt väiksemad kui sidekanali kaudu, katkestusest kuni esimese mõõtmiseni ilma PWM käivitamiseta vastavalt 6.3 μs ja 50.7 μs. Sidekanali peamiseks viite põhjuseks on CRC arvutamine ning lisaks erinevad kontrollid ja otsuste tegemine, mis aga sünkroniseerimisjuhtme juures enamjaolt puuduvad. Sellest tulenevalt on sünkroniseerimisjuhtme kasutus soovitatav, kui on vaja mitme tootega teostada mõõtmisi võimalikult sünkroonselt.

## Kokkuvõte

Töö eesmärgiks oli luua andmehõive- ja kommunikatsioonifunktsioonid ning juhtmega sünkroniseerimise võimalus ARM põhisele platvormile EK-TM4C123GXL, mida oleks võimalik kasutada laboratoorsetes tingimustes katsetuste läbiviimiseks. Lisaks veel uurida sünkroniseerimise täpsust ning mõõtemooduli poolt tekitatud hilistumisi. Kõik püstitatud eesmärgid said täidetud. Loodud programm on kasutatav ning võimalik edasi arendada, kuid esineb mitmeid kohti lahenduses, mida on võimalik täiustada.

Mõõtemooduli arenduse juures piirduti 16MHz kiirusel töötava süsteemikellaga, mille tõstmisega kuni 80MHz-ni on võimalik tõsta nii kommunikatsiooni kiirust kui ka lühendada genereeritava impulss-laius modulatsiooni signaali.

Hetkel eksisteeriv analoog-digitaal muunduri mõõtetöö ei ole eriti paindlik ning sellest tulenevalt on mõistlik mõõtetöö viia üle katkestuste põhiseks, kus iga katkestuse peale, ajasti-, või sünkroniseerimisjuhtme katkestkus, tehakse üks mõõtmine. Lisaks on võimalik määrata ka impulss-laius modulatsiooni generaatorit päästikuks, kus igal sobival tingimusel teostatakse üks mõõtmine, mis võimaldaks suuremat täpsust ning paindlikust määrata mõõtetöö hetkel.

PWM-ga signaalide genereerimisel tasub võimaldada ka eelnevalt kirjeldatud signaalide listi kasutamist generaatori sisendina, mis võimaldaks keerukamate signaalide genereerimist, mille erinevad signaalid jooksutatakse läbi automaatselt või on vahetatavad katkestuste abil.

Üleüldise programmi lihtsuse ja loetavuse eesmärgil tasub kasutada katkestusi ka kommunikatsiooni mooduli alusena.

## Kasutatud kirjandus

1. Cyclic redundancy check [WWW]  
[https://en.wikipedia.org/wiki/Cyclic\\_redundancy\\_check](https://en.wikipedia.org/wiki/Cyclic_redundancy_check) (20.04.2015)
2. Universal asynchronous receiver/transmitter  
[https://en.wikipedia.org/wiki/Universal\\_asynchronous\\_receiver/transmitter](https://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter)  
(09.03.2015)
3. TM4C123G mikrokontrolleri andmeleht [WWW]  
<http://www.ti.com/lit/ds/spms376e/spms376e.pdf> (20.06.2014)
4. TM4C123G kasutusjuhend [WWW] <http://www.ti.com/lit/ug/spmu296/spmu296.pdf>  
(01.03.2016)
5. CCS IDE [WWW] <http://www.ti.com/ltds/ti/tools-software/ccs.page> (17.05.2016)
6. CMSIS failid [WWW]  
[http://www.ti.com/tool/CMSIS\\_DEVICE\\_HEADERS?keyMatch=cmsis&tisearch=Search-EN-Everything](http://www.ti.com/tool/CMSIS_DEVICE_HEADERS?keyMatch=cmsis&tisearch=Search-EN-Everything) (17.05.2016)
7. Tiva TM4C ja LM4F osa numbrite vastavus [WWW]  
[https://e2e.ti.com/support/microcontrollers/tiva\\_arm/f/908/t/271820](https://e2e.ti.com/support/microcontrollers/tiva_arm/f/908/t/271820) (17.05.2016)
8. RS-422 and RS-485 Standards Overview and System Configurations [WWW]  
<http://www.ti.com/lit/an/slla070d/slla070d.pdf> (04.2010)

## **Lisa 1. Loodud programmi kood**

Loodud programm on üles laetud pilve keskkonda github ning on avalikult kättesaadav:

[https://github.com/margoraja/ARM\\_based\\_measurement\\_module\\_for\\_distributed\\_DAO](https://github.com/margoraja/ARM_based_measurement_module_for_distributed_DAO)