



TALLINNA TEHNIKAÜLIKOOL
MEHAANIKATEADUSKOND

Mehhatroonikainstituut

Mehhatroonikasüsteemide õppetool

MHK40LT

Kristjan Pahtma

TAHHOMEETER SÕIDUAUTOLE VAZ 21053

BSc Lõputöö

Autor taotleb
tehnikateaduste bakalaureuse
akadeemilist kraadi

Tallinn

2016

SISUKORD

EESSÕNA.....	6
SISSEJUHATUS	7
1. RIISTVARA VALIK.....	8
1.1. Kontrollermoodul	8
1.1.1. Arduino Uno	9
1.1.2. Raspberry Pi.....	10
1.1.3. AVR ATmega2561	11
1.1.4. Kontrollermooduli lõplik valik: Arduino Uno.....	12
1.1.5. Kontrollermooduli paigutus	12
1.2. Sensor	13
1.2.1. Magnetsensor MT-190W.....	13
1.2.2. Infrapuna sensor IRS-W	14
1.2.3. Optiline sensor (LED'il põhinev) ROS-W	14
1.2.4. Anduri lõplik valik: Magnetsensor MT-190W	15
1.3. Näidik.....	16
1.4. LED tuled.....	17
1.5. Pingemuundur	18
2. PROGRAMMEERIMINE	20
2.1. Programmi algoritm	21
2.2 Programmi simuleerimine	23
3. MEHAANIKA KONSTRUEERIMINE.....	24
3.1. Lugesirõngas.....	24
3.2. Sensori hoidik	28
KOKKUVÕTE.....	31
SUMMARY	32
KASUTATUD KIRJANDUS.....	33
LISAD.....	36
Programmi kood	36
Kahemõõtmelised joonised.....	39

EESSÕNA

Lõputöö mõte oli omaalgatuslik ja pole ühegi ettevõttega seotud. Töö koostamine ja algandmete kogumine toimus põhiliselt Tallinna Tehnikaülikooli ühiselamus ja selle maa-aluses parklas.

Avaldan tänu mind abistanud ja motiveerinud inimestele: juhendaja ja teadur Robert Hudjakov, kaasüliõpilane Risto Randla ning Aigar Iva ja Robin Eino.

SISSEJUHATUS

Lõputöö ülesandeks on projekteerida tahhomeeter sõiduautole VAZ 21053. Teema lähtub praktilisest vajadusest, kuna ühelgi tehasest tulnud VAZ 2105'1 ei olnud varustuses tahhomeetrit. Tahhomeetri puudumisel on raske ära kasutada mootori täit võimekust või kasutada autot ökonoomselt. Viimaste tõhustamiseks lisatakse ka LED tuled, mis annavad märku parimast käiguvahetuse ajast.

Insener-tehnilisi probleeme on mitmeid. Esimesena riistvara optimaalseim valik, millele on pühendatud esimene peatükk. Raudvara alla kuuluvad näidik, LED'id, mikrokontroller koos arendusplaadiga ja andur.

Teises peatükis tegeldakse töö virtuaalse poolega. Tarkvara peab suutma kontrollida näidikut, töödelda signaali, õigel ajal tulesid lülitada ja olema mikrokontrolleriga kokkusobiv.

Kolmas ja viimane peatükk käsitleb tööga seotud väljakutseid mehaanika vallas. Välja tuleb mõelda, mille järgi hakata hindama mootori pöörete arvu. See probleem käib käsi-käes sensori valikuga esimeses peatükis. Parim variant on selline, millel on võimalikult otsene seos väntvõlli pöörlemise kiirusega. Valiku tegemisel tuleb ka arvestada, mitu lugemit on võimalik ühe ringi jooksul võtta. Mida rohkem, seda täpsem on tulem. Tuleb projekteerida ka anduri kinnitus mootoriruumis. Selle tegemisel tuleb eelkõige lähtuda eelmises lõigus tehtust ja sensori eripäradest.

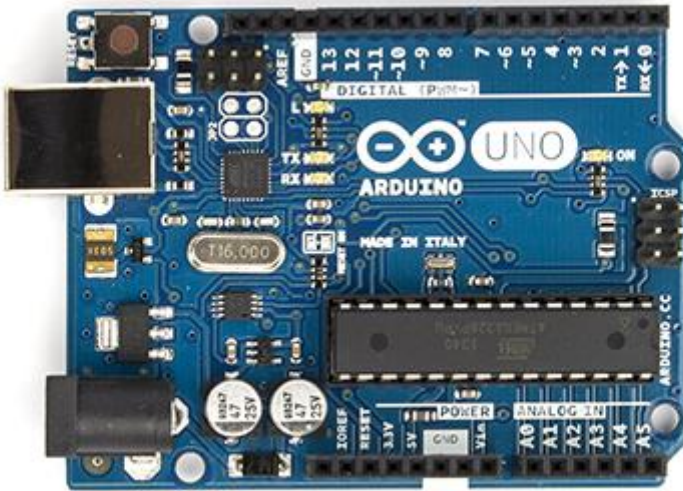
1. RIISTVARA VALIK

Pöördeid mõõdetakse anduriga, mis valitakse järgnevas peatükis. Andur saadab välja kõrge signaali vastavalt sellele, missugune takistus ja mis tüüpi sensoriga on tegemist. Signaali võtab vastu mikrokontrollermoodul, mis omakorda töötleb sisse tulnud info. Tema ülesandeks on ka väljunditele saata ette nähtud signaal. Väljunditeks on näidik, mis kuvab mootoripöördeid ja LED tuled, mis annavad märku optimaalseimast käiguvahetushetkest, võttes arvesse just sellel momendil mootori poolt arendatavaid pöördeid. Ülesandest lähtudes tuleb valida kontrollermoodul, andur, näidik, LED'id ja võimalikud lisakomponendid.

1.1. Kontrollermoodul

Kontrollermooduli ülesandeks on suhelda sisendi (andur) ja väljunditega (näidik ja LED'id). Anduri käest saab ta signaali, mille konverteerib nii väljundile arusaadavaks, et see näitab õiget tulemust. Vastavalt samadele saadud tulemustele süttivad LED'id, mida on kokku neli (sinine, roheline, kollane ja punane). See tähendab, et vaja on vähemalt ühte digitaalset sisendit ja viit digitaalset väljundit. Lisaks tuleb arvestada, et kontroller peab olema programmeeritav ning sellel on piisavalt mälu mulle vajaliku programmi jaoks. Hea oleks ka, kui peab kasutama võimalikult vähe pingemuundureid sisendite, väljundite või volutoite jaoks. Võrreldakse kolme erinevat kontrollerit ning nende seast valitakse sobivaim. Kontrollermoodulile kohti valikul tuleb arvestada, kuhu see paigutada, et välistegurid nagu vesi, päike ja tolm ei hakkaks tema tööd segama.

1.1.1. Arduino Uno



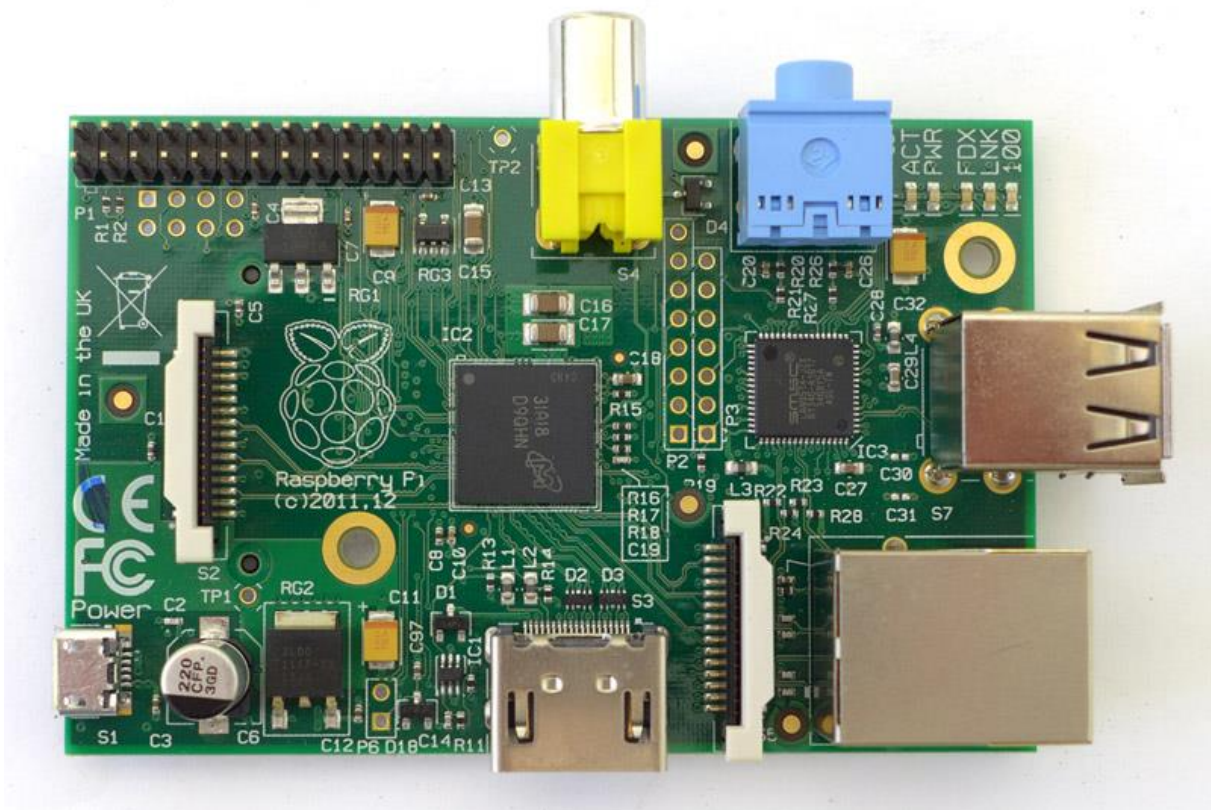
Pilt 1.1. Arduino Uno [1]

Tabel 1.1. Arduino Uno tehnilised andmed [2]

Mikrokontroller	ATmega328P
Protsessori kiirus	16 MHz
Mälu	32 kB
Digitaalsete sisendite/väljundite arv	14
Analoogsisendite arv	6
PWM väljundite arv	6
Sisendpinge maksimum (soovitav)	6 – 20 V (7 – 12 V)

Arduino Uno [Pilt 1.1] on siin mainitute kõige levinum. Kuna riistvara joonised ja tarkvara on vabalt saadaval, siis on olemas palju kolmandate osapoolte andureid ja muud riistvara. Tarkvarasid on võimalik kasutada mitmeid, kuid levinumaks on C/C++ programmeerimiskeel. Tabelist [Tabel 1.1] saab järeldada, et kriitilised aspektid, nagu sisendite ja väljundite arv ning mälu maht, on täidetud.

1.1.2. Raspberry Pi



Pilt 1.2. Raspberry Pi [3]

Tabel 1.2. Raspberry Pi tehnilised andmed [4][5]

Protsessor	ARM1176JZF-S
Protsessori kiirus	700 MHz
Mälu	512 MB
Digitaalsete sisendite/väljundite arv	8
Analoogsisendite arv	0
PWM väljundite arv	Lisamooduliga või tarkvaraliselt
Sisendpinge	4,75 – 5,25 V

Raspberry Pi [Pilt 1.2] on oma olemuselt sisuliselt personaalarvuti. Protsessori kiirus on kümnestes kordades suurem, kui kahel teisel siin toodud kontrollerial [Tabel 1.2]

[Tabel 1.1][Tabel 1.3]. Samuti vajab seade operatsioonisüsteemi, kui ülejäänud valikuvariantidel seda vaja pole, samas on selle tõttu tarkvara valik lai.

1.1.3. AVR ATmega2561



Pilt 1.3. AVR ATmega2561 [6]

Tabel 1.3. AVR ATmega2561 tehnilised andmed [7][8]

Mikrokontroller	ATmega2561-16AU
Protsessori kiirus	14,7456 MHz
Mälu	256 kB
Analoog-digitaalmuunduri kanalite arv	8
PWM väljundite arv	6
Sisendpinge maksimum (soovitav)	6 – 15 V (6 – 9 V)

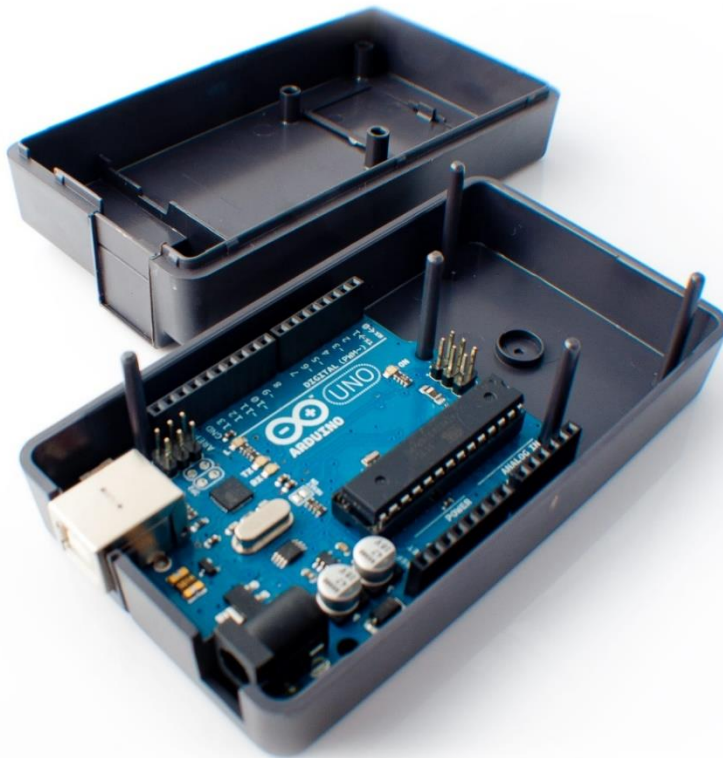
ATmega2561'i [Pilt 1.3] on sisendeid-väljundeid ja mälumahtu piisavalt [Tabel 1.3]. Seade on olemas Mehhatroonika Kodulaboris. Tarkvara osas saab kasutada C programmeerimiskeelt [7][8].

1.1.4. Kontrollermooduli lõplik valik: Arduino Uno

Valituks langes Arduino Uno oma laialdase leviku ja toe tõttu. Rolli mängis ka fakt, et toiteallikaks on autoaku, mille pinge on 12 V, AVR ATmega2561 puhul jääb see soovitatavast piirkonnast välja [7]. See võib esile kutsuda liigse kuumenemise. Raspberry Pi on käesoleva ülesande jaoks ebapraktiline, kuna protsessori kiirus on liiga suur, millega kaasneb kõrgem voolutarve. Lisaks vajab seade eraldi operatsioonisüsteemi [4].

1.1.5. Kontrollermooduli paigutus

Kontrollerit tuleb kaitsta kõrvõimalike väliste elementide, ilmastik, tolm jms, eest. Selle jaoks tuleb valida koht, kus need tegurid arendusplaati ei ohustaks. Otsustasin selleks kohaks valida kindalaeka, kuhu kruvitakse spetsiaalne Arduino jaoks mõeldud karp, „Arduino Project Enclosure“ [44], milles kontrollermoodul veel omakorda sees on [Pilt 1.4]. Nii on ka lihtne arendusplaadile ligi pääseda ja see vajadusel arvutiga ühendada.



Pilt 1.4. Arduino Project Box [45]

1.2. Sensor

Eesmärk on lugeda vastavalt anduri spetsiifikale talle kindlalt ette antud kordusi ja need mikrokontrollerile edastada. Korduste tüüp selgub vastavalt sellele, missugune sensor saab valituks. Kuna töötamistingimused võivad kapoti all olla rasked, peab andur suutma karmides tingimustes, nt vibratsiooni, mustuse ja ekstreemsete temperatuuride käes, vastu pidada. Vaja on tööstuslikul tasemel andurit. Oluline on ka, et andur mootoriruumi ära mahub ja tema saadetavad signaalid on mikrokontrollerile loetavad. Sensori vooluallikana võiks saada kasutada kas autoakut (12 V) või kontrollermoodulit (3,3 ja 5 V). Oluline on, et sensor suudaks lugeda vähemalt nii palju pöördeid, kui suudab arendada mootor. Valikusse jäi kolm eri tüüpi sensorit Monarch Instruments'ilt [9]. Valisin just selle firma, kuna komponendid on piisavalt vastupidavad. Nende kohta on olemas kõik vajalik info ja vajadusel saab reaalselt eraisik neid kasvõi ühekaupa tellida.

1.2.1. Magnetsensor MT-190W



Pilt 1.5. Magnetsensor MT-190W [10]

Tabel 1.4. Magnetsensori MT-190W tehnilised andmed [11][12]

Töökaugus	kuni 6,35 mm
Kiirusvahemik	1 – 99 999 p/min
Töötemperatuur	-73° kuni 107° C
Tööks vajalik pinge	3,3 – 12 V
Väljundsignaal	nelinurksignaal
Juhtme pikkus	2,4 m

Sensor saadab välja signaali, kui töökaugusest on läbi käinud magnetiseeruv metall [12]. Tegemist on digitaalsignaali, kontrollermoodul saab signaale kokku lugedes teatud aja lõikes arvutada signaalide saatmise sageduse ja seega mootori pöörded. Pildilt [Pilt 1.5] on näha, kuidas on võimalik lugeda näiteks hammasratta hambaid.

1.2.2. Infrapuna sensor IRS-W



Pilt 1.6. Infrapuna sensor IRS-W [13]

Tabel 1.5. Infrapuna sensori IRS-W tehnilised andmed [14][15]

Töökaugus	12 – 25 mm
Kiirusvahemik	1 – 999 990 p/min
Töötemperatuur	-40° kuni 85° C
Tööks vajalik pinge	3,3 – 15 V
Väljundsignaal	nelinurksignaali
Juhtme pikkus	2,4 m

Kui töökaugusel on sihtmärk, kus vahelduvad kontrastid või kui infrapunakiires tehakse järsk muutus, saadab andur välja signaali [15]. Pildil [Pilt 1.6] on näha, et kontrastiks on võllile pandud heledamat värvi kleebis.

1.2.3. Optiline sensor (LED'il põhinev) ROS-W



Pilt 1.7. Optiline sensor (LED'il põhinev) ROS-W [16]

Tabel 1.6. Optiline sensori (LED'il põhinev) ROS-W tehnilised andmed [17][18]

Töökaugus	kuni 1 m
Kiirusvahemik	1 – 250 000 p/min
Töötemperatuur	-10° kuni 70° C
Tööks vajalik pinge	3,3 – 15 V
Väljundsignaal	nelinurksignaal
Juhtme pikkus	2,4 m

Andur tuvastab tema enda poolt välja saadetud valguse peegeldumist. Peegeldumise jaoks on spetsiaalne teip, mis tuleb kleepida pöörlevale osale [Pilt 1.7][18].

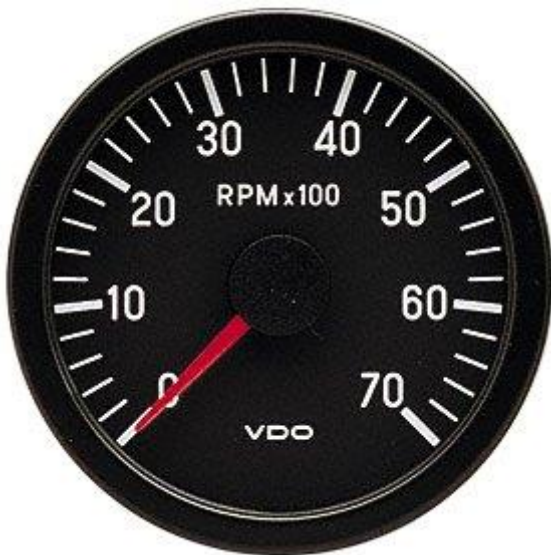
1.2.4. Anduri lõplik valik: Magnetsensor MT-190W

Kiirusvahemik, milles suudavad andurid töötada, on kõigil piisav. Ka tööks vajalik pinge vahemik oli igal anduril sobilik antud projekti teostamiseks [Tabel 1.4][Tabel 1.5][Tabel 1.6]. Optilise sensori otsustasin kõrvale jätta, kuna tema temperatuurivahemik (-10° kuni 70° C) ei ole piisav. Talviti on tihti Eesti kliimas alla kümne miinuskraadi. Lisaks on probleemne fakt, et kuna anduri töökeskkonda võib sattuda kergesti mustust, siis optiline seade ei pruugi anda õiget lugemist [18]. Samal põhjusel otsustasin elimineerida ka infrapuna sensori. Märklaua mustaks saamisel ei annaks see enam täpset infot. Viimase anduri kahjuks räägib ka tõsiasi, et see on tugevalt õhuniiskusest sõltuv ja ei talu kondensaati, mida paratamatult tekkida võib [15].

Magnetsensori puudusteks on vajadus kasutada ferriitset metallist hammastega rõngast ja töökaugus on kõige lühem, mis tähendab, et andur tuleb paigaldada väga täpselt. Nendest probleemidest on aga võimalik lihtsamini üle saada, kui teiste puudustest. Magnetsensori kasuks räägib ka kõige laiem töötamise temperatuurivahemik (-73° kuni 107° C) [12].

1.3. Näidik

Näidiku ülesandeks on kuvada mikrokontrollerist välja saadetud infot. Valisin VDO 333155 Vision Black seeria tahhomeetri näidiku [Pilt 1.8].



Pilt 1.8. VDO 333155 Vision Black seeria tahhomeetri näidik [19]

Tabel 1.7. VDO 333155 tehnilised andmed [20]

Maksimaalsed kuvatavad pöörded	7000 p/min
Tööpinge	12 V
Signaali tüüp	nelinurksignaali
Läbimõõt	8,6 cm

Autol oleva mootori maksimaalsed pöörded on piiratud 6000 p/min juures [43], seega sobib näidik hästi pöördede kuvama [Tabel 1.7]. Arduino'st on kerge välja anda digitaalsignaali. Antud näidik näitab pöördede, kui talle saata iga pöörde kohta kõrge digitaalsignaali. Kuna kontrolleri väljund on kuni 5 V [2], siis 12 V sisendi saamiseks näidikule tuleb kasutada transistorit, mida juhib kontrollermoodul ja juhitav vool tuleb auto akult, mille nimipinge on 12 V. Pea kümne sentimeetrine läbimõõt ning hästi vastanduvad värvid tagavad hea nähtavuse

[Tabel 1.7]. Lisaks on näitur sarnane nendega, millega auto tehasest väljus ning jääb nii üsna diskreetseks.

1.4. LED tuled

LED'id peavad põlema minema, kui mootor on saavutanud teatud pöörded. Ökosõiduks, 2000 p/min juures, sinine tuli. Peatsest parimast käiguvahetushetkest, 5500 p/min juures, annab märku roheline tuli. Jõudluse vaatevinklist optimaalseimaks käiguvahetuseks, 5800 p/min juures, kollane tuli ja punane, selleks kui mootor juba piirajas on, 6000 p/min juures [41][43].

Viimase kolme jaoks kasutatakse sama tüüpi LED'e.



Oshino MD-577**G**2 indicator LED [21]

Oshino MD-577**Y**2 indicator LED [22]

Oshino MD-577**R**2 indicator LED [23]

Pilt 1.9. Oshino MD-577G2 indicator LED [24]

Kõik kolm LED'i on sisult samad, väiksed erinevused nimes tähistavad värve, ülevalt alla, vastavalt roheline [Pilt 1.9], kollane ja punane. Need tuled töötavad kõik 12 V juures. See tähendab, et aktiveerimiseks on vaja pingemuundureid, kuna kontrollermooduli maksimaalne väljundpinge on 5V. Installeerimise mõttes on hea, et neil on juhtmed, mitte ainult klemmid küljes ja LED'i ümber on nailonist pesa [21][22][23].

Sinine LED on teistsugune, mis on kasulik, arvestades tema funktsiooni. Tegemist on Dietz PI45580 indikaatortulega [Pilt 1.10]. Sisendpinge, nagu ülejäänud kolmel, on 12 V [25].

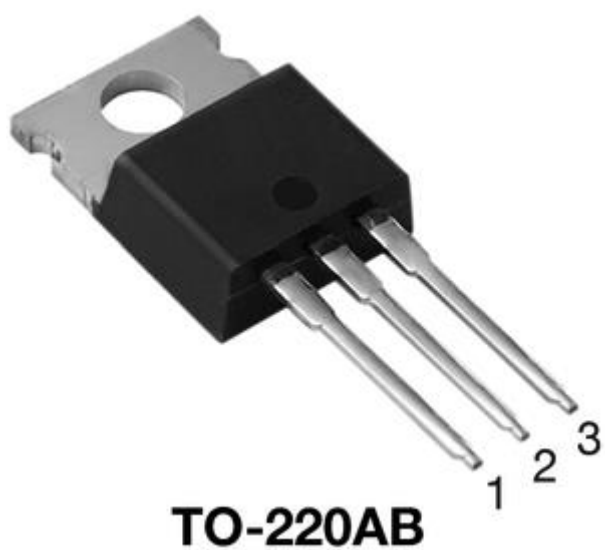


Pilt 1.10. Dietz PI45580 [26]

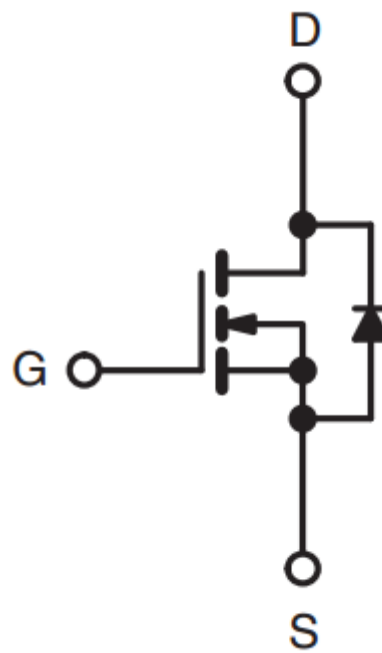
1.5. Pingemuundur

Kuna kõikide väljundite tööpinge on 12 V, aga Arduino plaad suudab välja anda kuni 5 V suurust väljundit [2], siis tuleb need kuidagi omavahel ühilduma panna. Lahendusi on mitmeid, nt relee või võimendi. Võimendi liiga ebapraktiline, kuna tuleb palju lisa riistvara ja autoaku näol on olemas 12 V sisend [43]. Seda tuleks lihtsalt kuidagi juhtida. Releede probleemiks, aga fakt, et toimub reaalne, füüsiline ümberlülitus. See tähendab jälle osade kulumist ja kuna lülitusi võib olla sadu ühe sekundi jooksul, siis ei pruugi relee vastu pidada. Seetõttu otsustasingi transistori kasuks.

Transistoriks valisin IRF520. See kasutab TO-220AB [Pilt 1.11] seadmepakendit, mis hajutab hästi soojust. IRF520 on rikastatud, n-kanaliga metall-oksiid pooljuht-väljatransistor (MOSFET) [27][28][29][Pilt 1.12]. Rikastatud tähendab seda, et kui paisu pinge on null, siis ei lasta voolu ka läbi. MOSFET (metal-oxide-semiconductor field-effect transistor) sobib hästi, kuna on kiireim räni-põhine transistor, juhtida saab võrdlemisi suurt pinget ja on tööstuses väga levinud [29].



Pilt 1.11. TO-220AB [30]



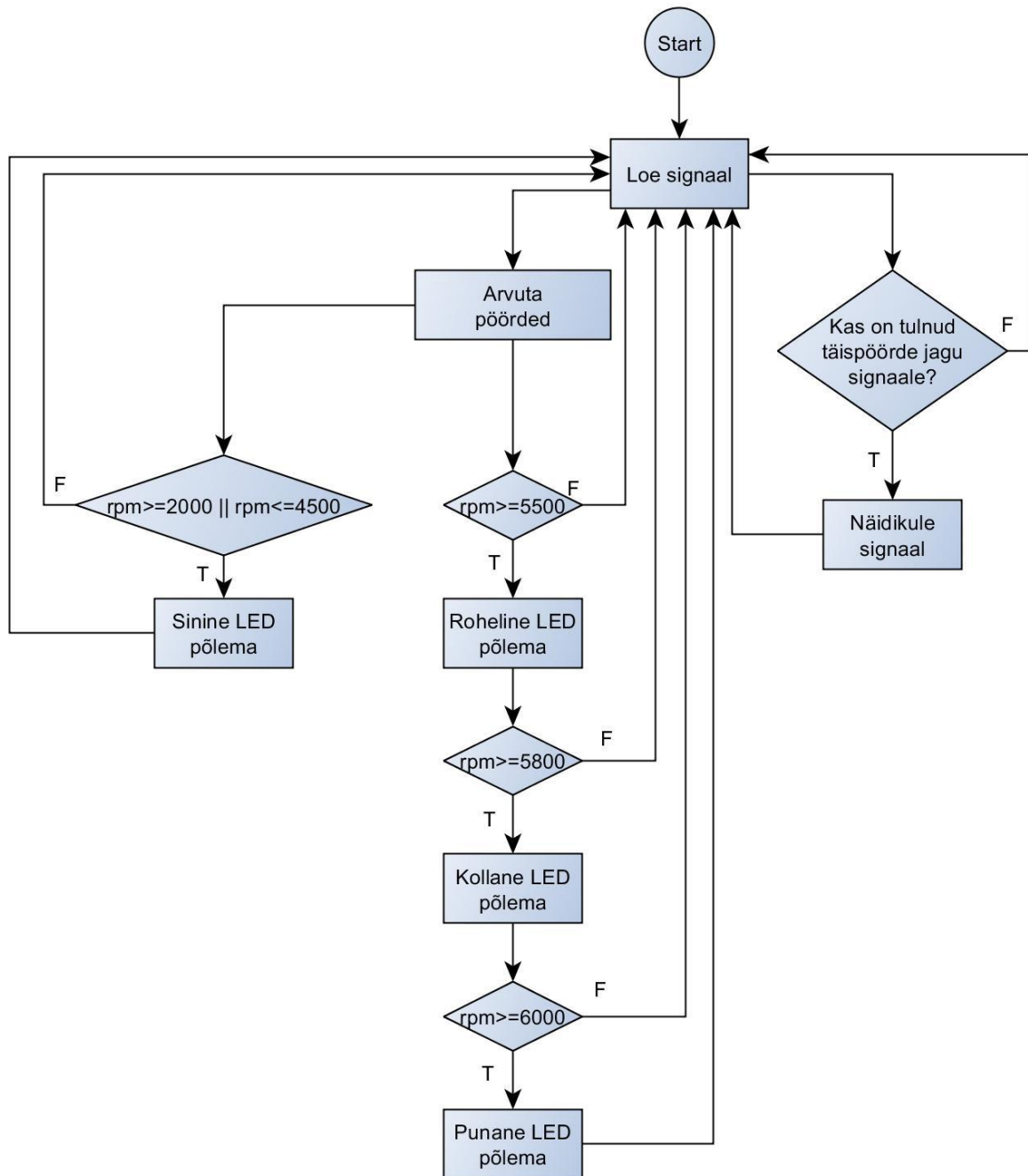
Pilt 1.12. IRF520 skeem [28]

2. PROGRAMMEERIMINE

Programm juhib mikrokontrolleri tegevust. Selle eesmärgiks on vastavalt sisse tulevatele signaalidele arvutada mootori pöörete arv minutis. Vastavalt saadud tulemustele on vaja signaal väljunditele anda. Programm peab olema piisava korratavusega ning suuteline vastu võtma ja mitte vahele jätma ühtegi signaali.

Kuna kõik keeled, mida Arduino arendusplaat ilma pidevalt arvuti külge ühendatuna suudab lahendada, on C/C++ põhised, otsustasin valida Arduino enda keele [35]. Tegemist on sisuliselt C/C++ keeles genereeritud funktsioonidega, mida saab Arduinos välja kutsuda. Kood kompileeritakse kui C/C++. Arduino enda koodi puhul saab kindel olla, et kood täidetakse alati täielikult ja tarkvara poole pealt ei teki takistusi, mis võib juhtuda kolmandate osapoolte programmidega. Lisaks on olemas palju näiteid ja suur kommuuni toetus [36]. Algoritmi genereerimiseks kasutasin tarkvara yEd Graph Editor [37].

2.1. Programmi algoritm



Pilt 2.1. Programmi algoritm

Algoritmist [Pilt 2.1] saab järeldada, et kohe alguses jaguneb programm kaheks. Paremalt pool on signaalide saatmine näidikule. Kuna andur saadab lugemid mitu tükki ringi kohta, aga näidikule tuleb saata üks signaal ühe pöörde kohta, siis tuleb see omavahel sobitada. On teada,

mitu lugemist saadab sensor ringi kohta. Kontrolleri saadab signaali iga selle numbri tagant, mitu lugemist andur saadab pöörde jooksul. See tähendab, et näidiku täpsus on sama, mis see oleks siis, kui lugemist võetaks ühe täispöörde kohta üks. Leian, et see on piisav analoognäidikut ja inimese silma arvestades.

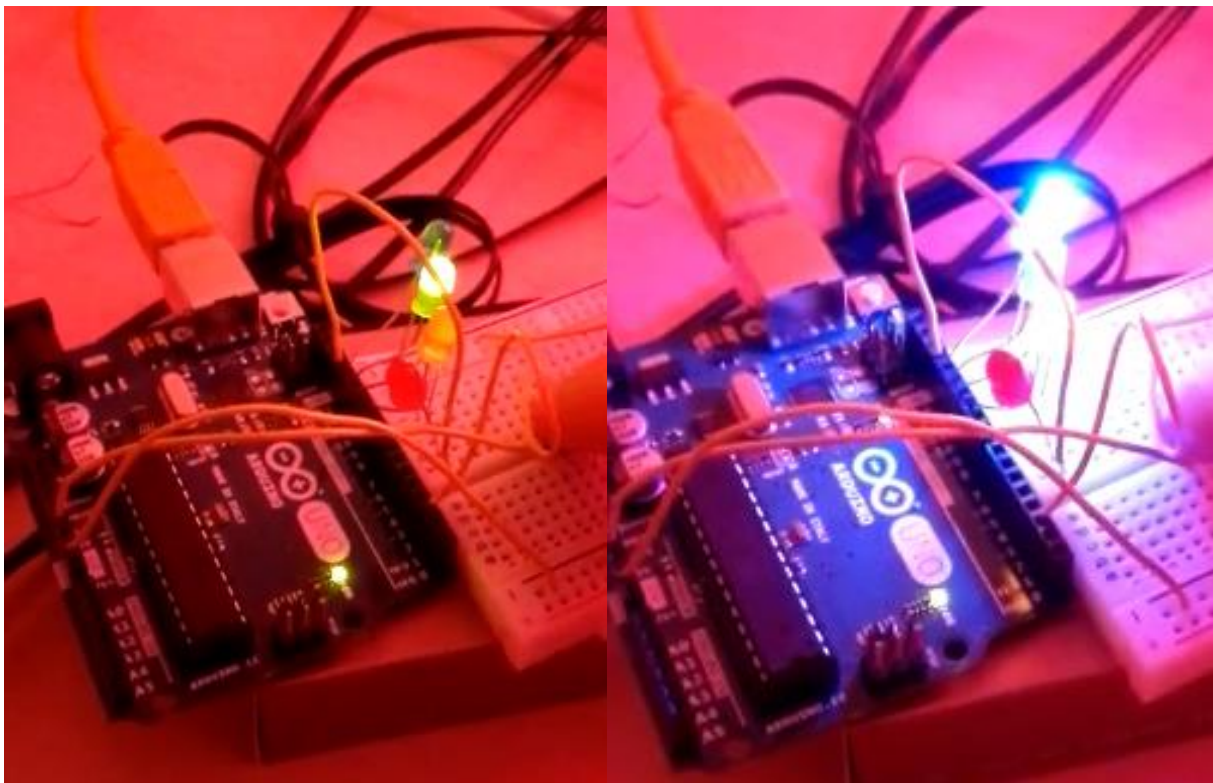
Vasakul pool on näha pöörete arvutamine. Seal kasutatakse ära kõiki lugemist, maksimaalselt täpse tulemuse jaoks. Täpsus on siin olulisem kui näidiku puhul, sest viimast on raske jälgida, aga LED'ide süttimist on lihtne märgata. Et saada õiget pöörete arvu minutis, tuleb lugeda lugemiste arvu ette antud aja jooksul. Seda on üsna raske teostada, kuna programmi täidetakse üks rida korraga. Probleem seisneb selles, et samal ajal kui lugemist loetakse, peaks ka lugema aega. Lihtsam variant on kasutada kindla aja pikkust pausi („delay“), mis on tsüklis ja need tsüklid kokku lugeda. Sellise meetodi puhul kaasneb oht, et pausi ajal tuleb sisse uus signaal, mis jääks sellisel juhul arvutustesse kaasamata ning lisaks ei arvesta selline aja mõõtmise lahendus protsessoril teiste käskude täitmisele kuluva ajaga.

Lahendus on kasutada funktsiooni „millis“ [38]. See näitab, mitu millisekundit on arendusplaat antud programmi jooksutanud. Antud funktsioonist väärtuse lugemine ja nende salvestamine muutujatesse võimaldab sisuliselt jälgida aega. Kord ühe programmiringi jooksul kontrollitakse väärtust. Kui see on võrdne või ületanud ette antud tulemusega, siis arvutatakse pöörded. LED'id pannakse põlema nii, nagu algoritmis näha. Sinine 2000, roheline 5500, kollane 5800 ja punane 6000 p/min juures. „Millis“ funktsiooni kasutamise puuduseks on täisarvu ülevoolu tekkimine. See tähendab, et iga umbes 50 päeva tagant funktsioon nullib iseennast [34]. Kui peaks olema juhtunud, et funktsioon nullib ennast nii, et üks muutuja jääb vana ja teine uue aja järgi, selleks on olemas eraldi koodirida. Teada on maksimaalne signaalide arv, mis on anduril võimalik teatud ajaperioodis saata. Kui see arv on ületatud umbkaudu kolmekordselt, nullitakse kõik „millis“ funktsiooniga seotud muutujad. See tähendab et kord kahe kuu jooksul võib tekkida kuni kolmesekundiline paus tahhomeetri töös. Programm on nii üles ehitatud, et kindlasti kontrollitakse pöörded ainult sinise ja rohelise jaoks. Kui roheline põlema ei lähe, siis kollase ja punase tsüklit minnakse niisama üle. Programm saab nii järgmiste asjadega tegeleda. Sinine tuli kustutatakse, kui pöörded on minutis rohkem kui 4500, kuna see näitab optimaalset käiguvahetust arvestades ökonoomsust. Kui aga pöörded juba nii kõrgel on, siis võib väita, et vähene kütusekulu pole enam prioriteet.

Arduino programm, koos kommentaaridega, on *Lisades*. Kaasa aitavad Arduino näiteprogrammid [38][39].

2.2 Programmi simuleerimine

Koodi kontrollimiseks kasutasin Arduino Uno arendusplaati [2]. Sensori kõrget signaali simuleerib nupuvajutus ning näidikut ja LED tulede süttimist kontrollerplaadiga kaasas olnud LED'id [38][39]. Kontrolli mõttes on ümber arvutamise valemeid muudetud nii, et vähete signaalide korral oleks näha erinevust. Kontrolli tulemusena saab koodi veel optimeerida.



Pilt 2.2. ja 2.3. Pildid programmi simuleerimisest

Pildidel [Pilt 2.2][Pilt 2.3] on näha, mis ulatuses on projekt realselt teostatud.

3. MEHAANIKA KONSTRUEERIMINE

Eesmärgiks on valida andurile koht, kus see kõige paremini toimida saaks ja võimalikult täpseid lugemeid võtaks. Laias laastus jaguneb ülesanne kaheks, kus kõigepealt tuleb täpselt välja mõelda kuidas andur pööreid loeb ja missuguseid elemente on selleks vaja. Teises pooles tuleb leida viis anduri kinnitamiseks nii, et see ka kinni püsiks, tuleb jälgida et oleks piisavalt ruumi ja et lugemid oleksid täpsed.

3.1. Lugemisrõngas

Kuna valisin välja magnetanduri, pean genereerima võimaluse kasutada magnetiseeruvast metallist kordusi, mida lugeda. Pöörete lugemiseks mootoriruumis on mitmeid variante. Anduri saab panna generaatoririhma jälgima. See pole aga väga hea variant, kuna rihm võib libiseda ning rihmale metalsete ribade lisamine võib seda kahjustada.

Parim variant on lugemeid võtta rullikutelt, mille peal antud rihm ringi käib. Kõige mõttekam on kasutada seda, mis tuleb otse väntvõlli küljest [Pilt 3.1], kuna generaatori ja veepumba ajamite puhul peab arvutama ülekandeteguri ja lugem on ebatäpsem, kuna tekivad rihmast ja teistest rullikutest tulenevad määramatused ja lõtkud. Lisaks on valitud rullikul õõnsused, mis soodustab lugemisrõnga paigaldust. Ümbruses on näha ka polte, mille abil saab hiljem kinnitada anduri hoidiku [41][42].



Pilt 3.1. Rullik, mis on väntvõlliga otseühenduses

Magnetsensor on võimeline lugema erinevaid tüüpi sihtmärke, nagu kruvipead või pilud, aga eelistatud on käigu hambad. Ette on antud ka hamba laiusvahemik sensori optimaalseks tööks. (2,692 – 15,875 mm). Sensori maksimaalne kaugus sihtmärgist on 6,35 mm [12].

Kuna sensor suudab ühe hamba korral maksimaalselt 99 999 pööret minutis lugeda [12] ja on teada, et auto mootori pöörded on piiratud 6000 p/min juures [43], siis on võimalik saada lugemeid rohkem kui korra ühe pöörde jooksul. See tõstab täpsust.

$$\frac{99\,999}{6000} = 16,6665$$

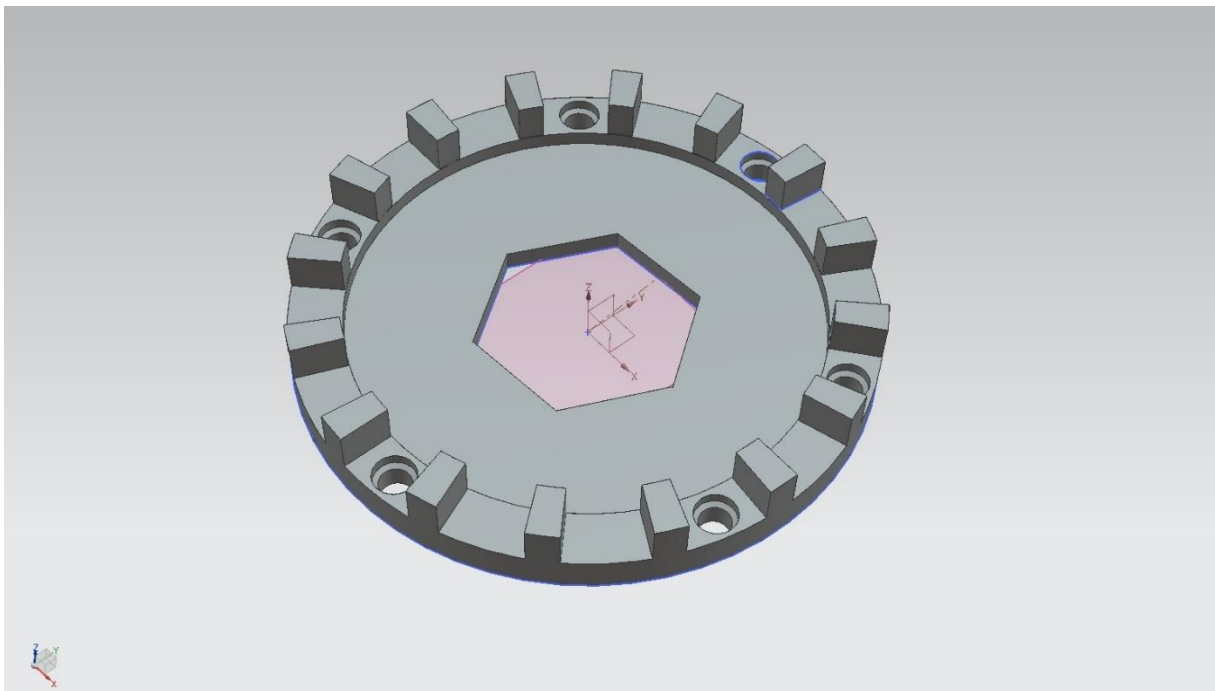
Arvutusest on näha, et ühe pöörde jooksul võiks võtta 16 lugemit.

$$16 * 6000 = 96000$$

16 lugemi juures jääb ka väike varu. Järeldan, et rõngal võiks olla 16 hammast. Seega ühe täispöörde jooksul võetakse 16 lugemit. Rulliku õõnsusega alale tuleb projekteerida rõngas. Antud ala läbimõõt on 116 mm. Arvestades seda ja fakti et ala ääres on väikse raadiusega faas, rõnga läbimõõt võiks olla 110 mm. Rõnga ja anduri teljed hakkavad olema samas sihis, mis rulliku telg. Hamba sügavus peab olema piisav, et isegi kui sensor selle vastu põrkub, suudaks ta infot lugeda, seega üle 6,35 mm, mis on sensori maksimaalne lugemiskaugus. Olgu see väikse

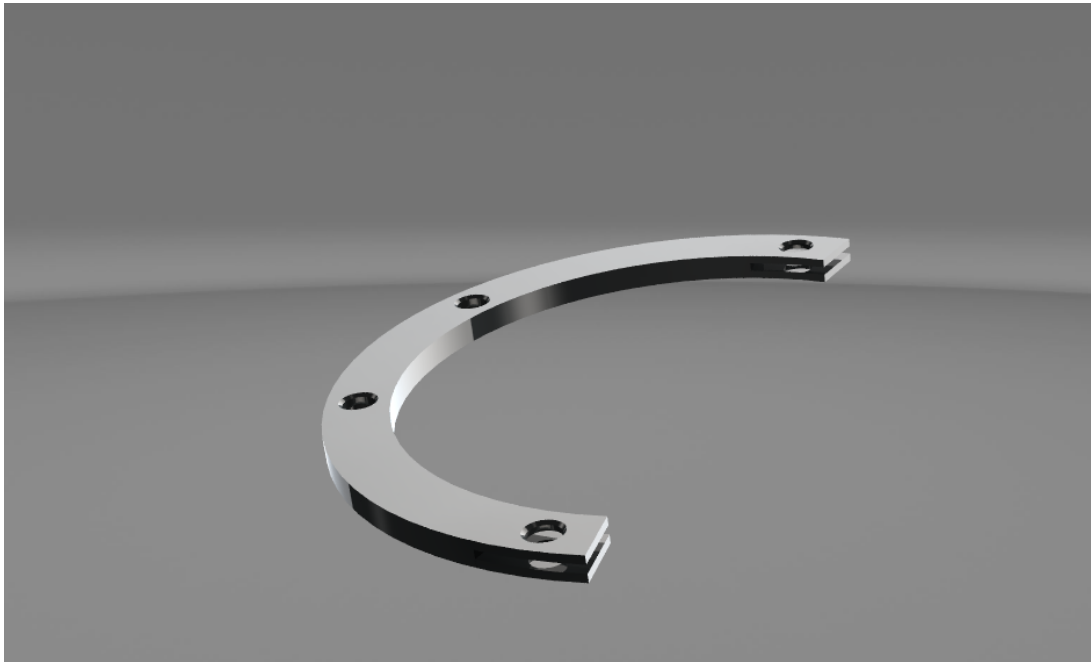
varuga, 8 mm. Hamba laiuse ja kõrguse piirangud on ette antud, need olgu vastavalt 6 ja 10 mm.

Mõõtmed annavad hea ülevaate, kuid olukorra paremaks piltlikkustamiseks otsustasin luua kahemõõtmelised joonised, mis on *Lisades* ja kolmemõõtmelised (3D) mudelid [Pilt 3.2]. Modelleerimistarkvarana kasutan Siemensi poolt pakutavat NX 10't [40]. Valik langes selline, kuna programm on piisav sellise ülesande täitmiseks, olen ise just seda tarkvara viimati kasutanud ning see on olemas ka kooli arvutiklassis.



Pilt 3.2. Hammasrõnga 3D mudel

Stabiliseerimise ja tsentreerimise eesmärgil saab ära kasutada rulliku keskel olevat kuusnurkset poldi pead. [Pilt 3.1] See aitab ka kinnitusprotsessil. Rõngas kinnitatakse teise rõngaga, mis hakkab olema rulliku taga. See koosneb aga kahest osast, et rõnga kinnitamiseks ei peaks rullikut ära võtma [Pilt 3.3][Pilt 3.4]. Kahte rõngast ühendab 6 kruvi. Poldide pead lähevad hammastega rõnga sisse, et mitte valesid lugemeid toota. Tagumise rõnga augud on keermestatud, mis tähendab mutrite vajalikkuse kadumist.



Pilt 3.3. 3D mudel kinnitusrõnga esimesest poolest

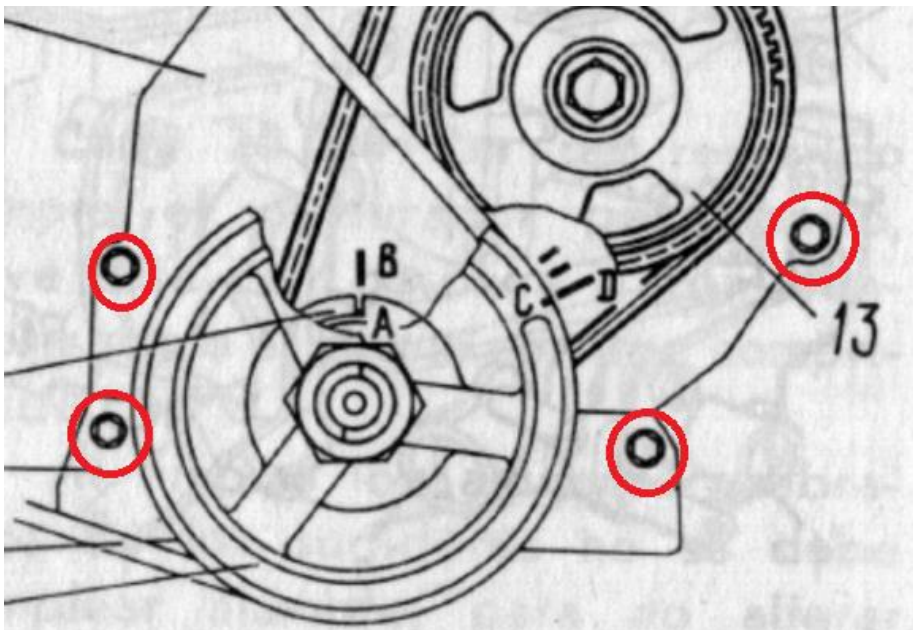


Pilt 3.4. 3D mudel kinnitusrõnga teisest poolest

3.2. Sensori hoidik

Andur peab olema piisavalt tugevasti kinnitatud, et see välisjõudude mõjul ära ei kaoks või katki ei läheks. Samas peab ta olema kinnitatult väga täpselt ja stabiilselt, et sensor saadaks häiringuteta lugemeid.

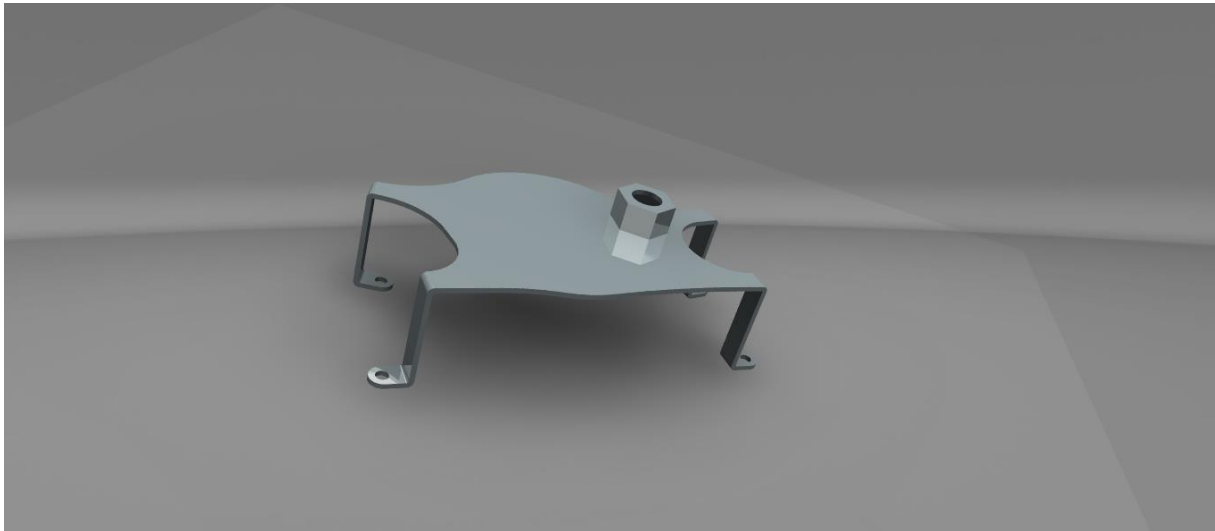
Eelmainitud nõudeid arvestades, ei saa andurit kinnitada kere külge. Arvestades lõtkusid mootoripatjades tuleks leida hoidikule koht mootori küljes. Otsustasin valida neli polti, millega on kinnitatud korpus [Pilt 3.5], mille eesmärk on kaitsta hammasrihma ümbritsevate elementide eest. Pildil punasega märgitud poldid on need, mida kasutatakse hoidiku kinnitamisel. See, et poldid jäävad 2,5 mm väljapoole, kui nad muudu on, ei tekita probleemi, kuna need on tehtud varuga. Arvestada tuleb ka sellega, et hoidik tuleb projekteerida vastavalt oma ümbrusele. Ruumi on vähe ja ümbruskonnas on palju liikuvaid osi [41][42][43].



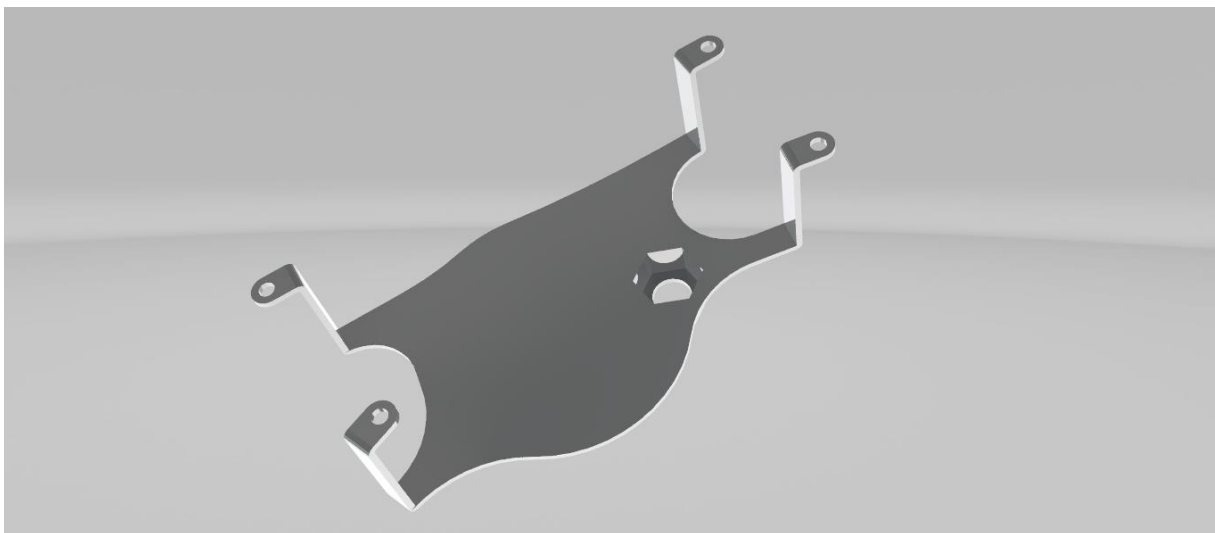
Pilt 3.5. Skeem poltide paigutusest ja rihma ning rullikute asukohast [41]

On teada, et rullik ulatub augu kohtadest 31 mm kaugemale. Hammasrõngas asub 3 mm sügavuses õõnsuses, rulliku sees. Hammasrõnga kogu kõrgus on 15 mm. Seega hambad ulatuvad 43 mm ($31 - 3 + 15 = 43$) kaugusele aukude tasandist.

Hoidik toimib mingil määral ka hammasrõnga väliste elementide, nagu pinnase ja vee eest kaitsjana. Hoidiku materjali paksus on 2,5 mm.



Pilt 3.6. Hoidik pealtvaates



Pilt 3.7. Hoidik altvaates

Piltidelt [Pilt 3.6][Pilt 3.7] on näha, et hoidik kinnitatakse nelja poldiga. Põhiplaadi kõrgus on sama, mis maksimaalne anduri kaugus poltide kinnituskohast ehk 49,35 mm. See tuleneb lugemisrõnga hammaste kõrgusest aukude tasandist (43 mm) ja anduri maksimaalsest töökaugusest (6,35 mm). Nii saab anduri vajadusel enne hoidiku mootori külge kinnitamist õigesse kaugusesse paika panna.

Sensor kinnitatakse augu kohast, mõlemalt poolt mutriga. Sensor pealt terves ulatuses keermestatud, põhimõtteliselt on tegemist 16x50 poldiga [12]. Anduri jaoks on hoidjasse tehtud

süvis, et see oleks võimalik alumise tasapinnaga tasa panna. See on kuusnurkne, seega saab sinna asetada mutri ja anduri saab fikseerida välispoolset mutrit kasutades.

KOKKUVÕTE

Töö esimene peatükk käsitleb riistvara ja selle valikut. Tehti kindlaks vajalikud komponendid. Selekteeriti kontrolleri, anduri, näidiku, LED tuled ja transistorid. Kontrollermooduliks võeti laialt levinud Arduino Uno. Monarch Instruments'ilt on magnetsensor MT-190W, mis tööstustasemel seadmena peab vastu ka karmides oludes. Tahhometri näidik on VDO poolt, 333155 Vision Black. LED'id tootjatelt Oshino ja Dietz. Transistori kasutamise vajadus ilmnis väljundite spetsiifika tõttu. Selleks valiti tööstuses tõestatud ja TO-220AB seadmepakendit kasutav MOSFET, IRF520.

Teine peatükk keskendub projekti tarkvaralisele poolele. Programmeerimiskeskondadest on valikuid palju, valituks sai Arduino keel, mis sobib kõige paremini kokku valitud kontrollermooduliga. Enne programmi kirjutamist, parema ülevaate saamiseks, genereeriti selle algoritm. Viimane teostati programmiga yEd Graph Editor. Arduino programmi testiti Arduino Uno peal, võimalike vigade leidmiseks.

Kolmas ja viimane peatükk tegeleb lõputöö mehaanilise osaga. Otsustati, et kasutatakse metallist, hammastega rõngast, mille kordusi loeb andur. Sensori võimekust arvestades arvutati välja, et rõngal on 16 hammast. Rõngale leiti koht rullikul, mis on otseühenduses väntvõlliga. Rõnga kinnitamine on korraldatud nii, et kuhugi ei ole vaja puurida auke ja midagi ei pea auto küljest ära võtma. Anduri kinnituse koha valikul ja projekteerimisel arvestati nõutava täpsuse ja keerulise töökeskkonnaga. Ka on arvestatud sellega, et kinnitamine ja sensori kauguse paika ajamine oleks võimalikult lihtne. Kõik vajalikud osad projekteeriti programmis NX 10.

SUMMARY

First chapter of the thesis deals with the selection of hardware. Necessary components were identified. Microcontroller board, sensor, gauge, LED lights and transistors were selected. Common Arduino Uno board was chosen. Monarch Instruments provided the magnetic sensor MT-190W, which as an industrial grade device can handle the harsh environment. Tachometer gauge is by VDO, 333155 Vision Black. LEDs are from the manufacturers of Oshino and Dietz. The necessity for a transistor occurred. The industry proven and TO-220AB package using MOSFET, IRF520 was chosen.

The second chapter concentrates on the software of the project. There were many choices among the programming environments, ultimately Arduino language was selected, which is also the most compatible with the microcontroller board. Before writing the program, an algorithm was generated for a better overview. It was done with yEd Graph Editor. Arduino program was tested on Arduino Uno, for debugging purposes.

Third and final chapter looks at the mechanical part of the thesis. It was decided, sensor will read ferrous pitch gears. Given the capabilities of the sensor, the gear should have 16 teeth. The toothed ring was found a place on the pulley, which is connected to crankshaft. Fastening of that ring is arranged, so there is no need to drill additional holes or disassemble anything. When developing the bracket for holding the sensor and choosing the place for it, accuracy and difficult working conditions were considered. The ease of fastening and setting up the right distance for the sensor was also given thought to. All the details were designed in NX 10.

KASUTATUD KIRJANDUS

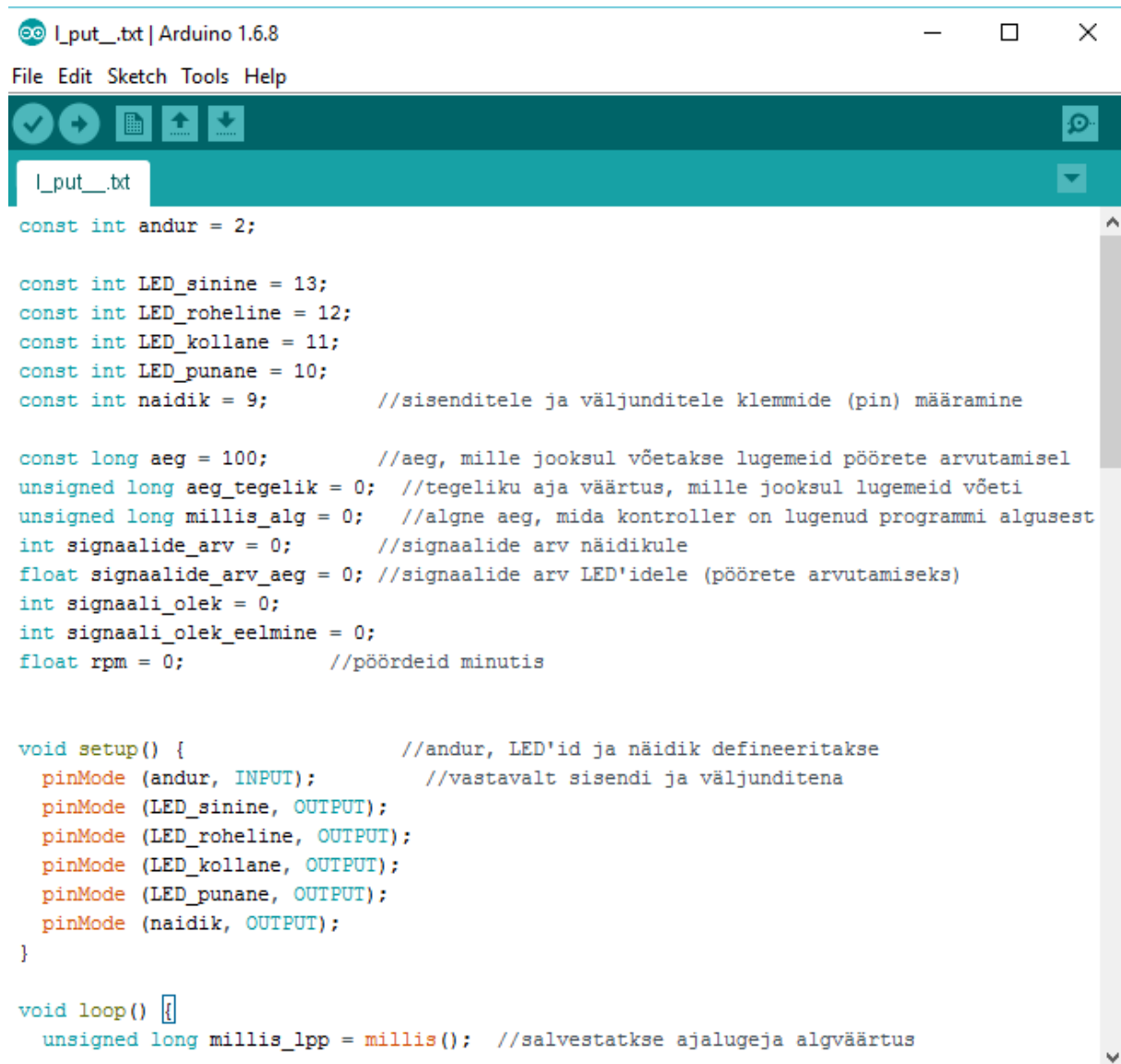
- [1] [WWW] https://www.arduino.cc/en/uploads/Main/ArduinoUno_R3_Front_450px.jpg
(29.04.2016)
- [2] [WWW] <https://www.arduino.cc/en/Main/ArduinoBoardUno> (29.04.2016)
- [3] [WWW] <http://www.mbtechworks.com/wp-content/uploads/2015/05/RP-1-B-800x.jpg>
(29.04.2016)
- [4] [WWW] <https://www.raspberrypi.org/help/faqs/> (29.04.2016)
- [5] [WWW] <http://raspberrypi.stackexchange.com/questions/298/can-i-use-the-gpio-for-pulse-width-modulation-pwm> (29.04.2016)
- [6] [WWW] <http://www.roboticlab.eu/oo/homelab/inc/atmega2561.png> (29.04.2016)
- [7] [WWW] <http://home.roboticlab.eu/et/hardware/homelab/controller/atmega2561>
(29.04.2016)
- [8] [WWW]
http://www.ttu.ee/public/m/Mehaanikateaduskond/Instituudid/mehhatroonika/Tudengile/Loputoo_2b.pdf (29.04.2016)
- [9] [WWW] <http://www.monarchinstrument.com/index.php> (29.04.2016)
- [10] [WWW] <http://monarchinstrument.com/images/Mt190wbe.GIF> (29.04.2016)
- [11] [WWW] <http://www.monarchinstrument.com/product.php?ID=40> (29.04.2016)
- [12] [WWW] <http://www.monarchinstrument.com/pdfs/manuals/1071-4853-112R.pdf>
(29.04.2016)
- [13] [WWW] <http://monarchinstrument.com/images/NEWIRSBEAM.jpg> (29.04.2016)
- [14] [WWW] <http://www.monarchinstrument.com/product.php?ID=42> (29.04.2016)
- [15] [WWW] http://www.monarchinstrument.com/pdfs/1071-4856-125R_IRS.pdf
(29.04.2016)
- [16] [WWW] <http://monarchinstrument.com/images/rosill.GIF> (29.04.2016)

- [17] [WWW] <http://www.monarchinstrument.com/product.php?ID=38> (29.04.2016)
- [18] [WWW] <http://www.monarchinstrument.com/pdfs/manuals/1071-4854-118R.pdf>
(29.04.2016)
- [19] [WWW] <http://ecx.images-amazon.com/images/I/41VWMFEDW0L.jpg> (29.04.2016)
- [20] [WWW]
http://www.usa.vdo.com/generator/www/us/en/vdo/main/hidden/downloads/instruments_accessories/EnthusiastCatalog/flc_EnthusiastCatalog_en.pdf?redirect=false (29.04.2016)
- [21] [WWW] <http://www.oshino-lamps.de/de/indi2g.html> (29.04.2016)
- [22] [WWW] <http://www.oshino-lamps.de/de/indi2y.html> (29.04.2016)
- [23] [WWW] <http://www.oshino-lamps.de/de/indi2g.html> (29.04.2016)
- [24] [WWW] <http://yeint.ee/image/data/imported/md500.jpg> (29.04.2016)
- [25] [WWW] <http://www.auto-radio-land.de/shop/themes/kategorie/detail.php?artikelid=7490&kategorieid=84&source=1&refertype=10> (29.04.2016)
- [26] [WWW] <https://www.automaailm.ee/admin/Uploads/Images/1f54f141-93df-44b6-beb8-f907306828f7.jpeg> (29.04.2016)
- [27] [WWW] <http://www.irf.com/product-info/datasheets/data/irf520.pdf> (29.04.2016)
- [28] [WWW] <http://www.vishay.com/docs/91017/91017.pdf> (29.04.2016)
- [29] [WWW]
<http://www.ene.ttu.ee/elektriamid/oppeinfo/materjal/IN660/ELEKTROONIKA%20ja%20j6upooljuhttehnika.pdf> (29.04.2016)
- [30] [WWW] <http://www.vishay.com/images/product-images/pt-large/94380-pt-large.jpg>
(29.04.2016)
- [34] [WWW] <https://www.arduino.cc/en/Reference/Millis> (15.05.2016)
- [35] [WWW] <https://www.arduino.cc/en/Main/Software> (29.04.2016)
- [36] [WWW] <https://www.arduino.cc/en/Main/FAQ> (29.04.2016)
- [37] [WWW] <https://www.yworks.com/products/yed> (29.04.2016)

- [38] [WWW] <http://www.arduino.cc/en/Tutorial/BlinkWithoutDelay> (29.04.2016)
- [39] [WWW] <http://www.arduino.cc/en/Tutorial/StateChangeDetection?from=Tutorial.ButtonStateChange> (29.04.2016)
- [40] [WWW] https://www.plm.automation.siemens.com/en_us/products/nx/10/index.shtml (29.04.2016)
- [41] [WWW] <https://es.scribd.com/doc/31901703/Manual-Vaz-2105#scribd> (29.04.2016)
- [42] Мирзоев Г.К., Вершигора В.А., Золотарев Е.М., Игнатов А.П., Новокшенов К.В., Пятков К.Б. Автомобиль ВАЗ-2105. Многокрасочный альбом. Москва : Машиностроение, 1994.
- [43] Soodla U., Vainola K. Sõiduauto „Žiguli“ : ekspluatatsioon ja hooldus. Tallinn : Valgus, 1979.
- [44] [WWW] <https://www.sparkfun.com/products/10088> (14.05.2016)
- [45] [WWW] <https://virtuabotix-virtuabotixllc.netdna-ssl.com/core/wp-content/uploads/2013/08/ardbox2.jpg> (14.05.2016)

LISAD

Programmi kood



```
Arduino 1.6.8
File Edit Sketch Tools Help
_put_.txt
const int andur = 2;

const int LED_sinine = 13;
const int LED_roheline = 12;
const int LED_kollane = 11;
const int LED_punane = 10;
const int naidik = 9; //sisenditele ja väljunditele klemmide (pin) määramine

const long aeg = 100; //aeg, mille jooksul võetakse lugemeid pöörete arvutamisel
unsigned long aeg_tegelik = 0; //tegeliku aja väärtus, mille jooksul lugemeid võeti
unsigned long millis_alg = 0; //algne aeg, mida kontroller on lugenud programmi algusest
int signaalide_arv = 0; //signaalide arv näidikule
float signaalide_arv_aeg = 0; //signaalide arv LED'idele (pöörete arvutamiseks)
int signaali_olek = 0;
int signaali_olek_eelmine = 0;
float rpm = 0; //pöördeid minutis

void setup() { //andur, LED'id ja näidik defineeritakse
  pinMode (andur, INPUT); //vastavalt sisendi ja väljunditena
  pinMode (LED_sinine, OUTPUT);
  pinMode (LED_roheline, OUTPUT);
  pinMode (LED_kollane, OUTPUT);
  pinMode (LED_punane, OUTPUT);
  pinMode (naidik, OUTPUT);
}

void loop() {
  unsigned long millis_lpp = millis(); //salvestatakse ajalugeja algväärtus
```

```

if(millis_lpp - millis_alg >= aeg) //kontrollitakse, kas lugemite võtmise aeg on täis
{
  aeg_tegelik = millis_lpp - millis_alg; //arvutatakse tegelik aeg
  rpm = (signaalide_arv_aeg/16)*60000/aeg_tegelik; //arvutatakse rpm
  millis_alg = millis_lpp; // eelmine ajalugeja lõppväärtus salvestatakse
  //kui algväärtus
  if (rpm >= 2000 || rpm <= 4500)//kontrollitakse sinise LED'i tingimusi
  {
    digitalWrite (LED_sinine, HIGH); //sinine LED põlema
  }
  else
  {
    digitalWrite (LED_sinine, LOW); //sinine LED kustub
  }

  if(rpm >= 5500)//kontrollitakse roheline LED'i tingimusi, kui tingimus pole
  {
    //täidetud, ei kontrollita kollast ja punast
    digitalWrite (LED_roheline, HIGH); //roheline LED põlema
    if (rpm >= 5800) //kontrollitakse kollase LED'i tingimusi
    {
      digitalWrite (LED_kollane, HIGH); //kollane LED põlema
      if (rpm >= 6000) //kontrollitakse punase tingimusi
      {
        digitalWrite (LED_punane, HIGH); //punane põlema
      }
      else
      {
        digitalWrite (LED_punane, LOW); //punane kustutada
      }
    }
  }
}

```

```

    }
  }
  else
  {
    digitalWrite (LED_punane, LOW); //punane ja kollane kustutada, siin on
    digitalWrite (LED_kollane, LOW); //ka punane kuna nende tsüklid on üksteise
  } //sees ja on olemas teoreetiline variant, et pöörded langevad kiiremini,
} //kui kontroller jõuab teha programmile tiiru peale
else
{
  digitalWrite (LED_punane, LOW); //sama asi, mis eelmises tsüklis
  digitalWrite (LED_kollane, LOW);
  digitalWrite (LED_roheline, LOW);
}
signaalide_arv_aeg=0; //pööretearvutamise signaalide arvu nullimine

}
signaali_olek = digitalRead (andur); //anduri lugemine

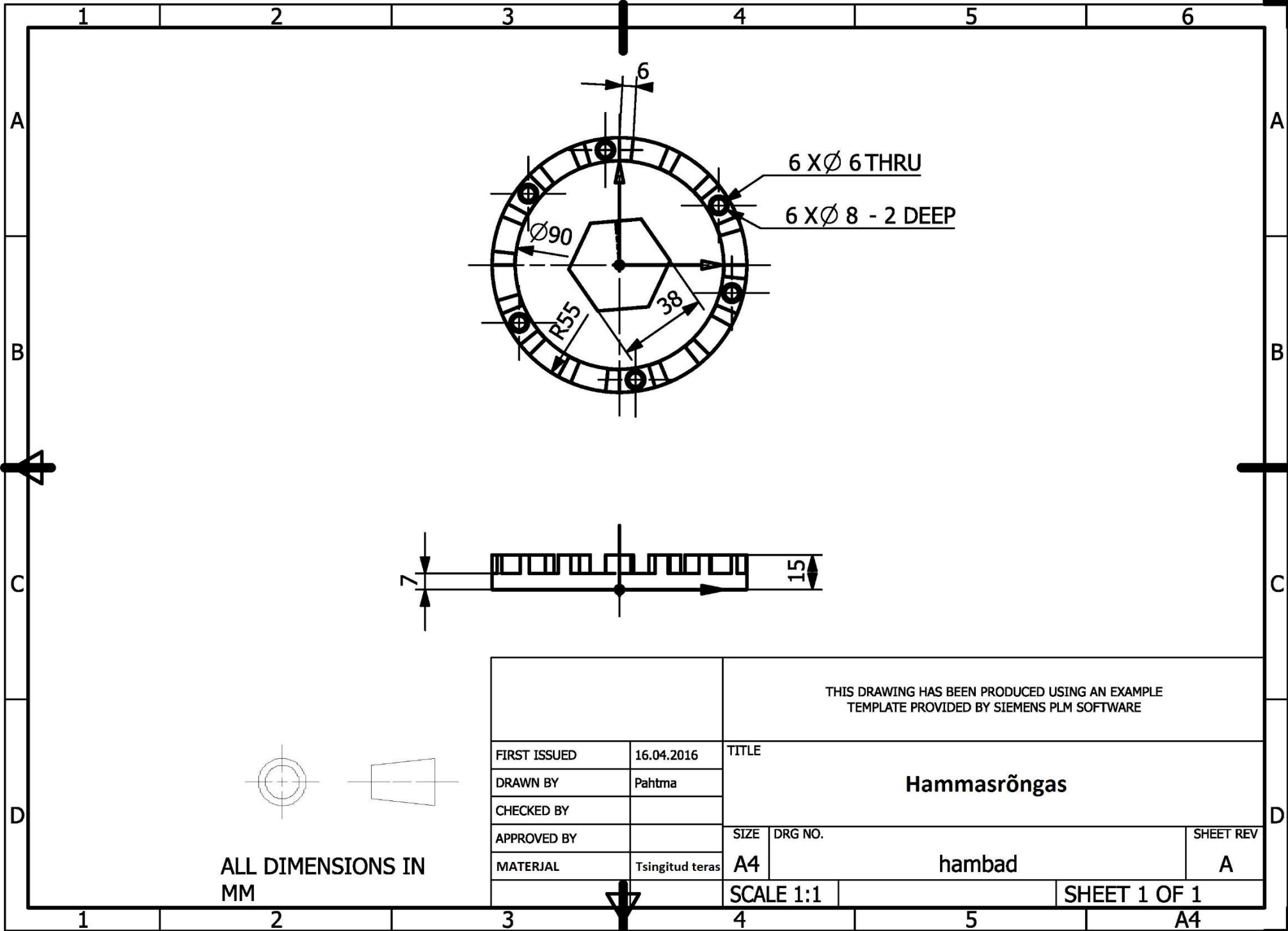
if(signaali_olek != signaali_olek_eelmine) //kontrollitakse kas anduri olek on muutunud
{
  //võrreldes oma eelmise olekuga, et mitte lugeda ühte signaali mitu korda
  if(signaali_olek == HIGH)//kontrollitakse, kas anduri olek on kõrge
  {
    signaalide_arv ++; //lisatakse signaalide arvudele 1
    signaalide_arv_aeg ++;
  }
}

signaali_olek_eelmine = signaali_olek; //signaali olek salvestatakse kui signaali
//eelmine olek
if(signaalide_arv == 16)//signaalide arv ühe ringi kohta on 16, kui see saab 16,
{
  //saadetakse näidikule signaal
  digitalWrite (naidik, HIGH);
}
else
{
  digitalWrite (naidik, LOW);
}

if(signaalide_arv >= 17)//kui signaalide arv on 17, siis asendatakse see ühega, et
{
  //toimiks ringlus ja signaal saadetakse välja iga 16nda sissetulnud signaaliga
  signaalide_arv = 1;
}
if(signaalide_arv_aeg>=500)//kontrollitakse "millis" funktsiooni reseti
{
  //signaalide arvu järgi
  millis_lpp=0; //nullitakse mõlemad "millis" funktsiooniga
  millis_alg=0; //seotud muutujad
}
}
}

```

Kahemõõtmelised joonised



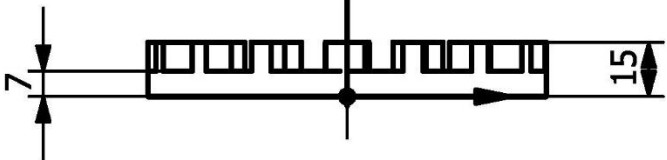
6 X Ø 6 THRU

6 X Ø 8 - 2 DEEP

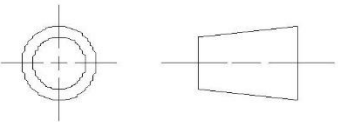
Ø90

R55

38



ALL DIMENSIONS IN
MM



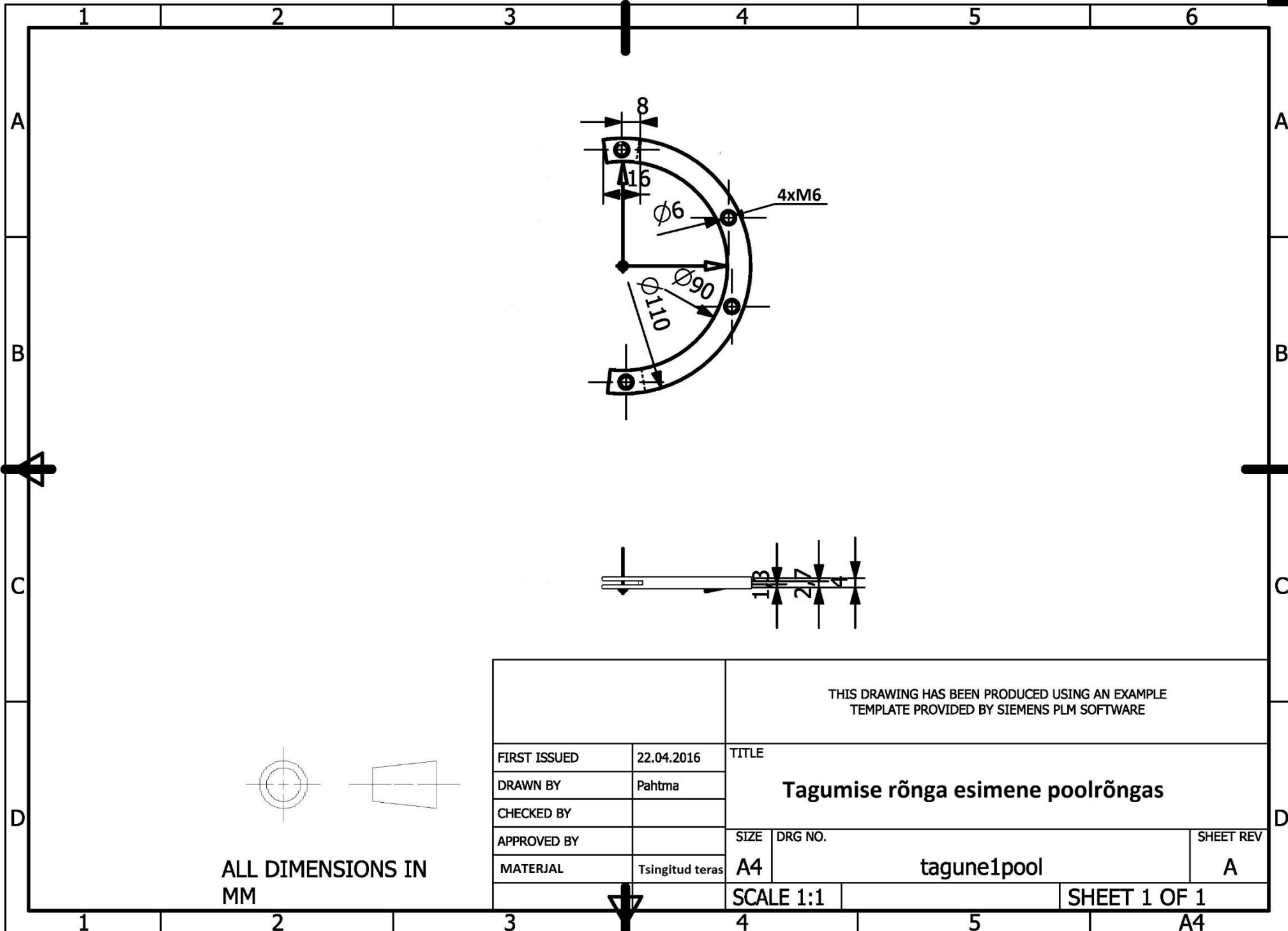
THIS DRAWING HAS BEEN PRODUCED USING AN EXAMPLE
TEMPLATE PROVIDED BY SIEMENS PLM SOFTWARE

FIRST ISSUED	16.04.2016
DRAWN BY	Pahtma
CHECKED BY	
APPROVED BY	
MATERJAL	Tsingitud teras

TITLE		Hammasrõngas	
SIZE	DRG NO.		
A4	hambad	SHEET REV	A

SCALE 1:1 SHEET 1 OF 1

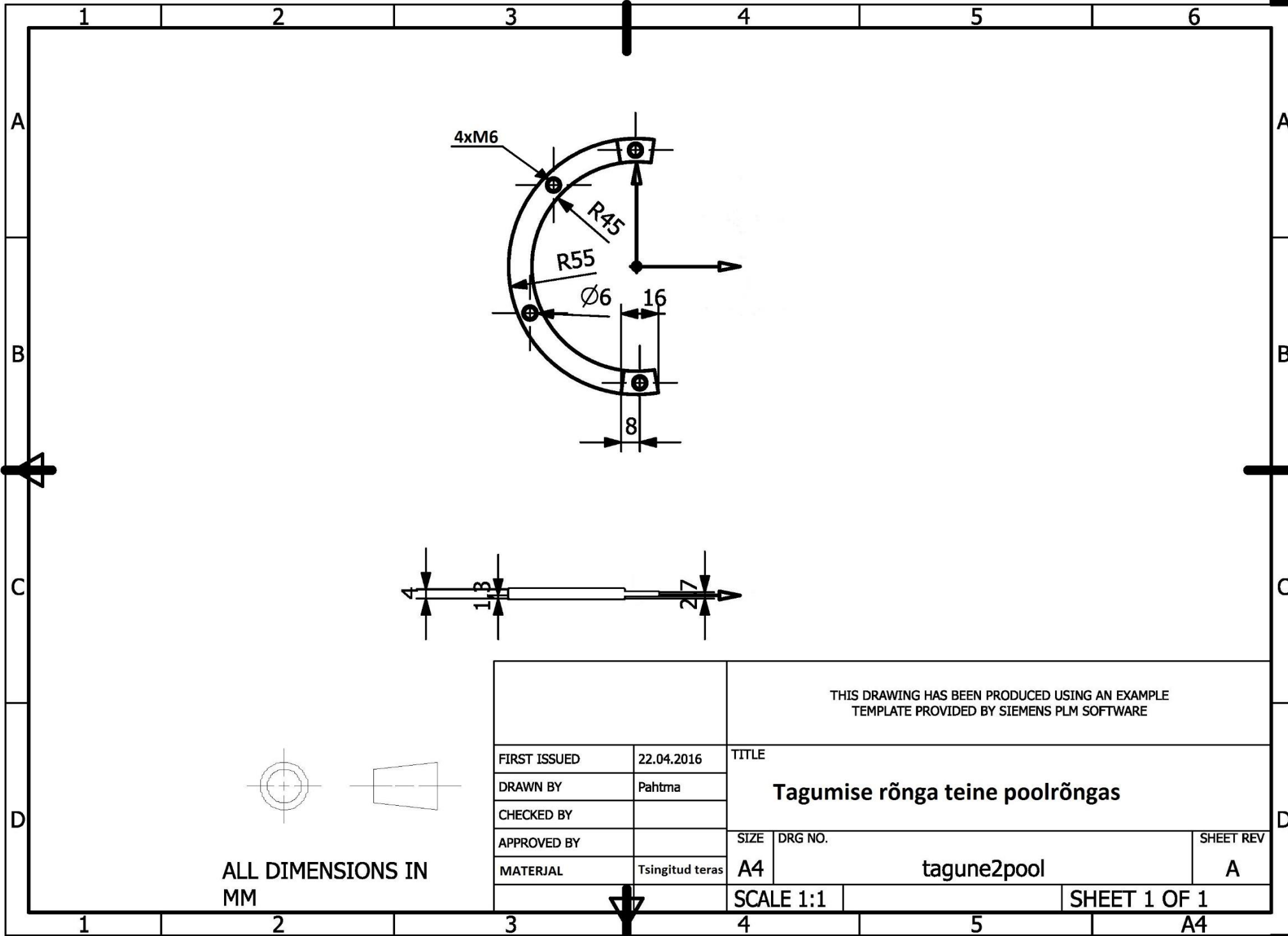
A4



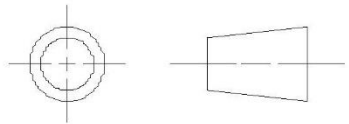
ALL DIMENSIONS IN
MM

		THIS DRAWING HAS BEEN PRODUCED USING AN EXAMPLE TEMPLATE PROVIDED BY SIEMENS PLM SOFTWARE		
FIRST ISSUED	22.04.2016	TITLE		
DRAWN BY	Pahtma	Tagumise rõnga esimene poolrõngas		
CHECKED BY				
APPROVED BY		SIZE	DRG NO.	SHEET REV
MATERJAL	Tsingitud teras	A4	tagune1pool	A
		SCALE 1:1	SHEET 1 OF 1	

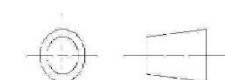
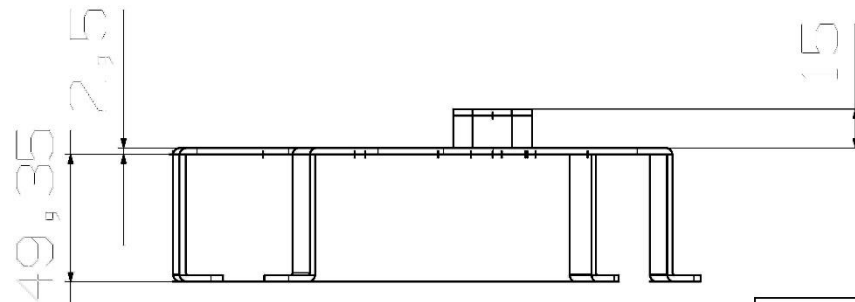
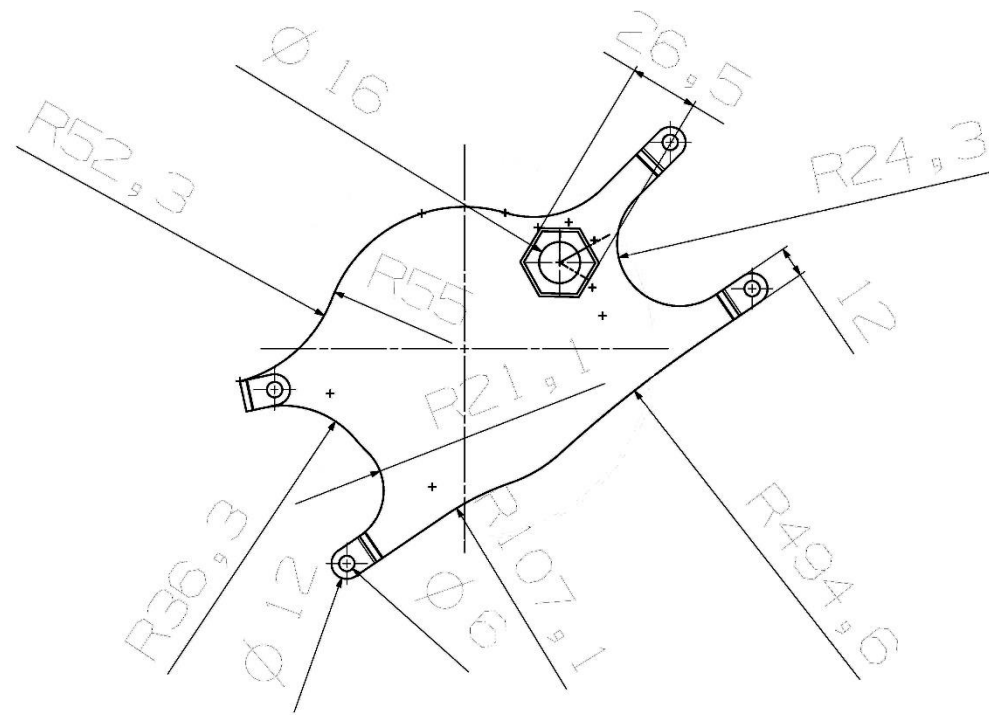
A4



ALL DIMENSIONS IN
MM



		THIS DRAWING HAS BEEN PRODUCED USING AN EXAMPLE TEMPLATE PROVIDED BY SIEMENS PLM SOFTWARE		
FIRST ISSUED	22.04.2016	TITLE		
DRAWN BY	Pahtma	Tagumise rõnga teine poolrõngas		
CHECKED BY				
APPROVED BY		SIZE	DRG NO.	SHEET REV
MATERJAL	Tsingitud teras	A4	tagune2pool	A
		SCALE 1:1	SHEET 1 OF 1	



ALL DIMENSIONS IN
MM

THIS DRAWING HAS BEEN PRODUCED USING AN EXAMPLE
TEMPLATE PROVIDED BY SIEMENS PLM SOFTWARE

FIRST ISSUED	26.04.2016	TITLE		
DRAWN BY	Pahtma	Anduri hoidik		
CHECKED BY				
APPROVED BY		SIZE	DRG NO.	SHEET REV
MATERJAL	Tsingitud teras	A1	bracket1	A
		SCALE 1:1		SHEET 1 OF 1