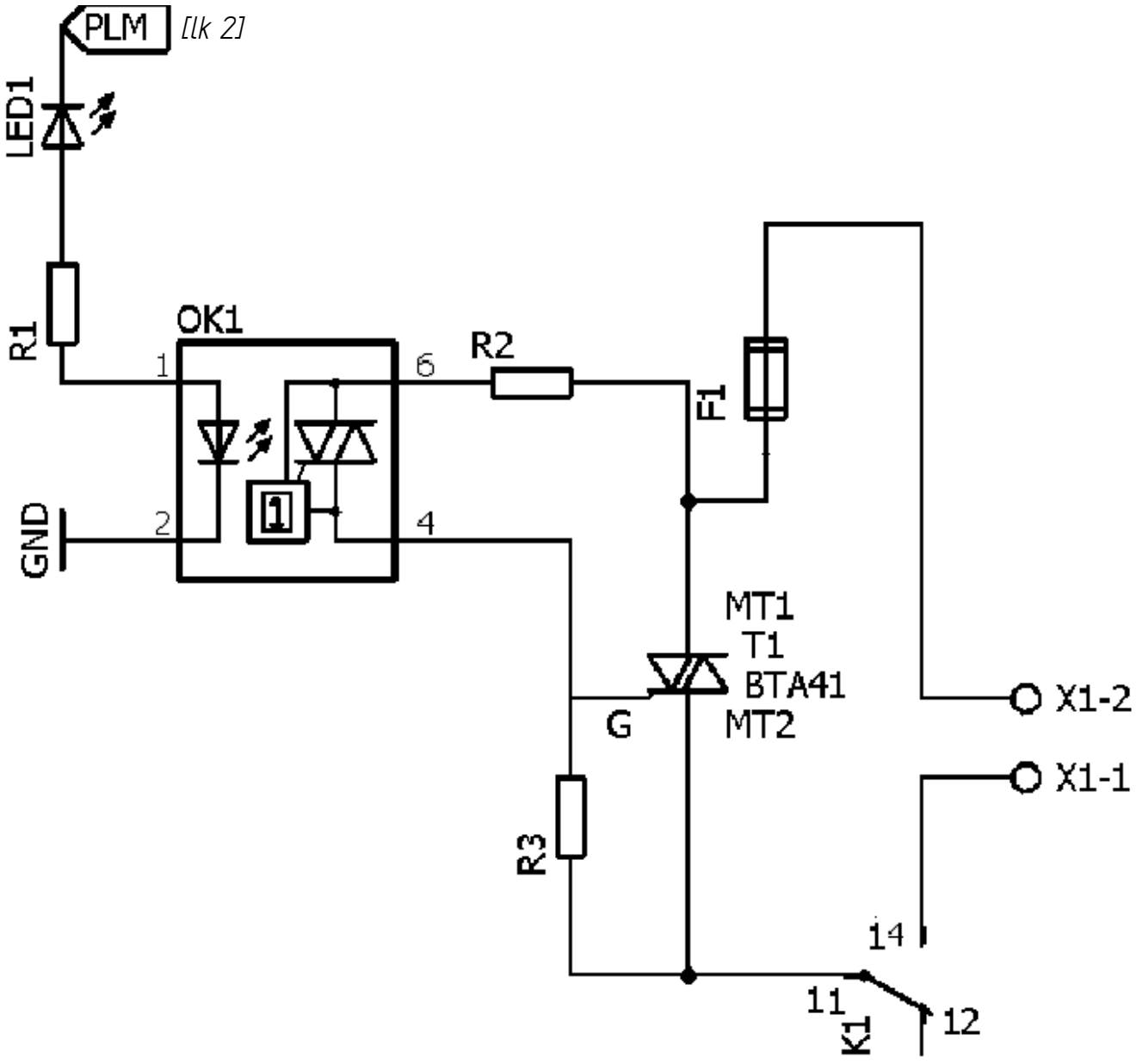
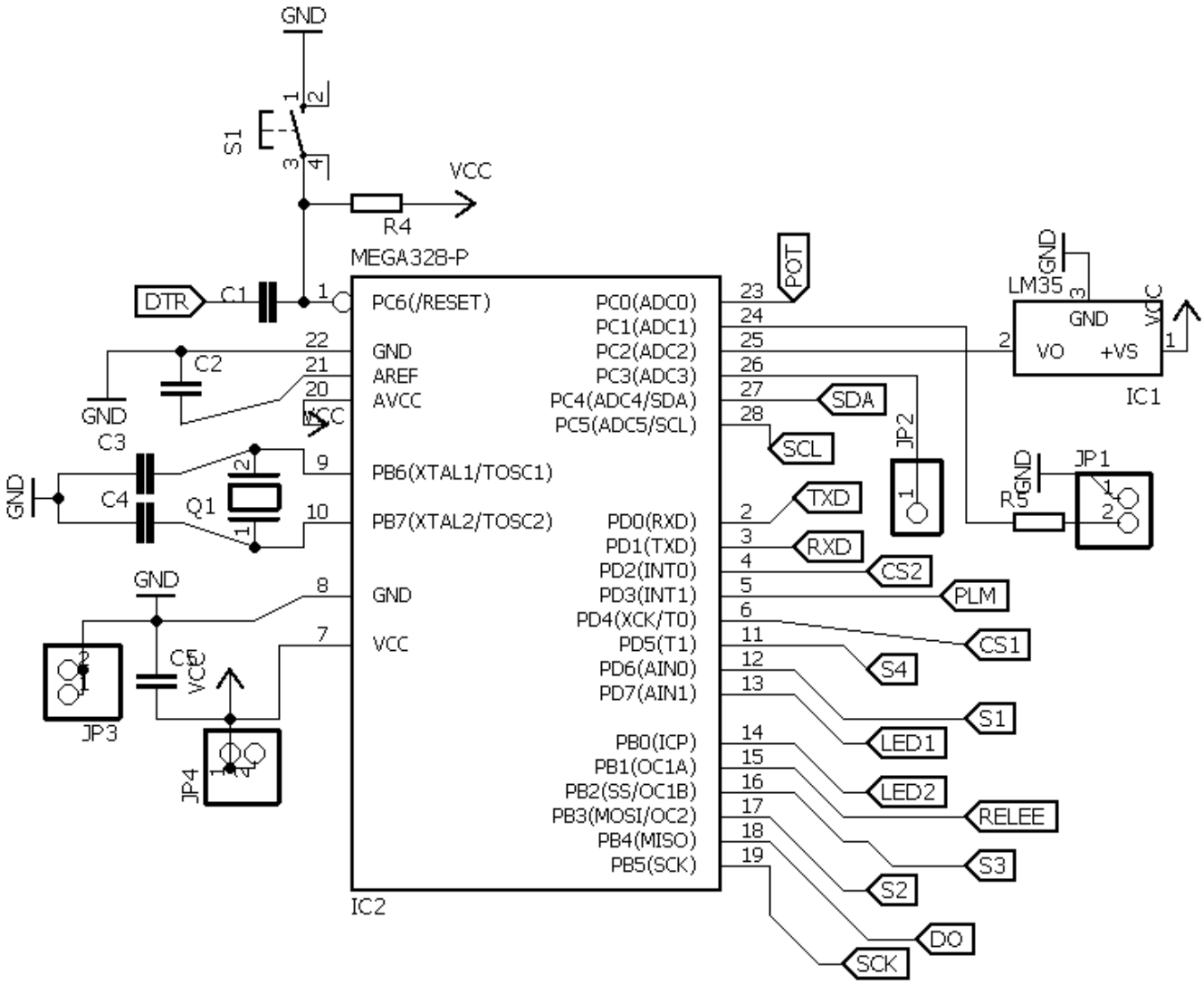
Lisa 1. Regulaatori skeemid



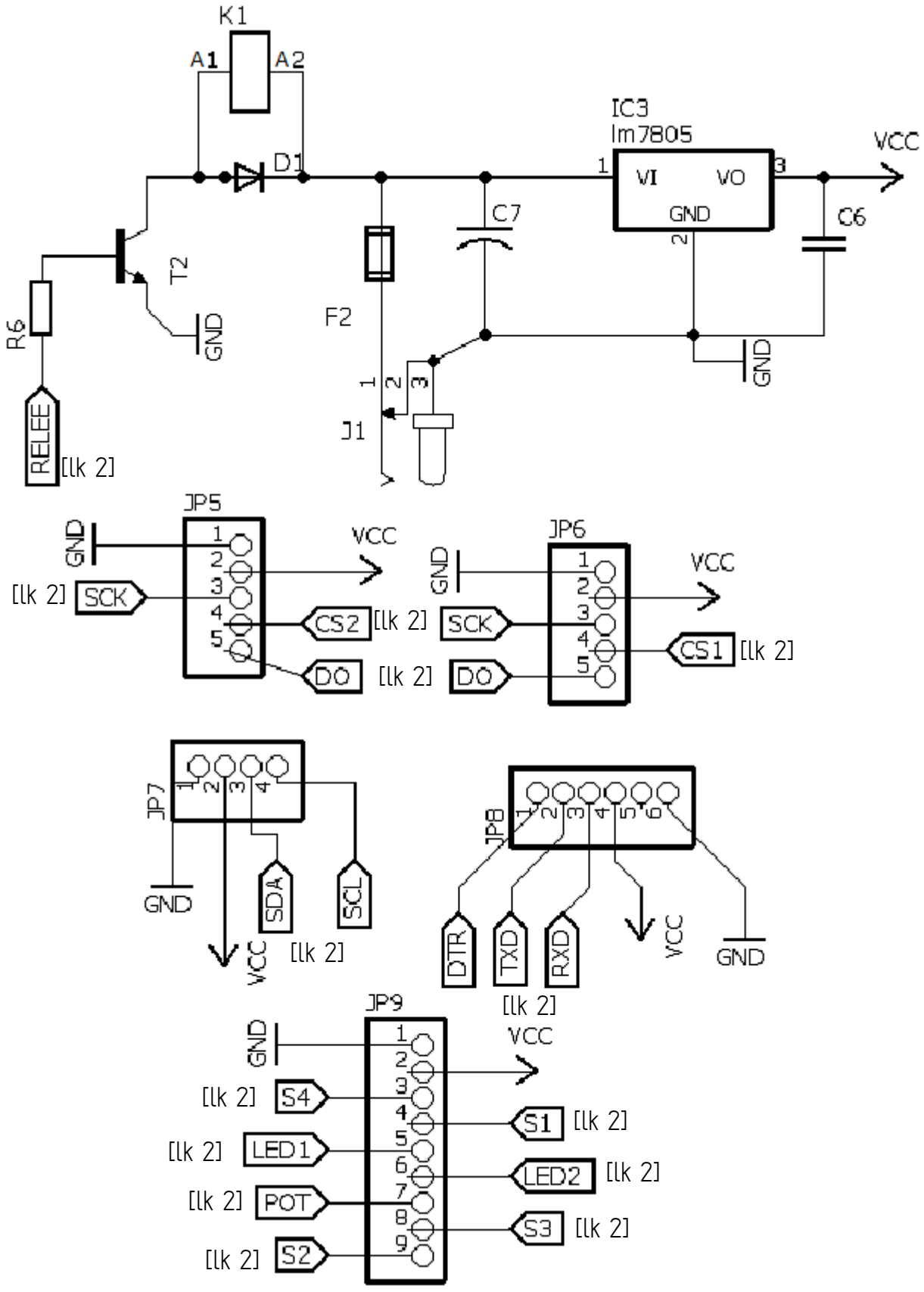
Teostas	Mihkel Keisk	Nimetus: Lisa 1. Optoisolaatori elektriskeem	
Kontrollis	Erkki Jõgi, Vahur Rooni		
Kinnitas	Erkki Jõgi, Vahur Rooni		
EMU		Leht: 1/6	Tähis: TA 18/110585 1 01 01 S

SOLID EDGE ACADEMIC COPY



Teostas	Mihkel Keisk	Nimetus: Lisa 1. Mikrokontrolleri elektriskeem	
Kontrollis	Erkki Jõgi, Vahur Rooni		
Kinnitas	Erkki Jõgi, Vahur Rooni		
EMU		Leht: 2/6	Tähis: TA 18/110585 1 01 02 S

SOLID EDGE ACADEMIC COPY



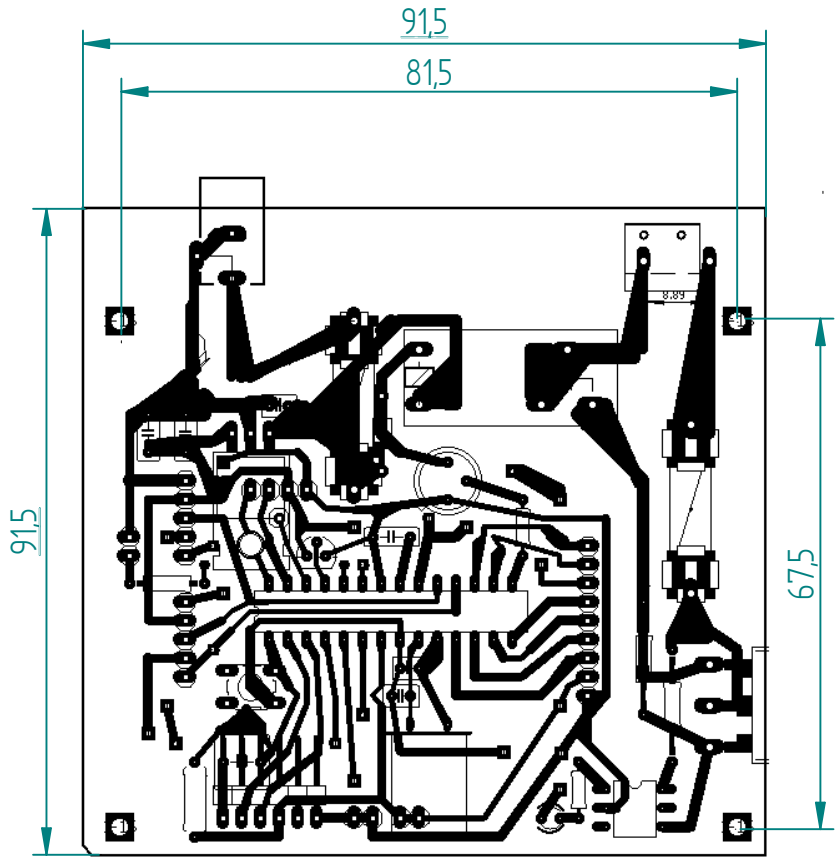
Teostas	*Mihkel Keisk*	*Nimetus:*
Kontrollis	*Erkki Jõgi, Vahur Rooni*	*Lisa 1. Pingeregulaatori ja viikude ühenduspesade elektriskeem*
Kinnitas	*Erkki Jõgi, Vahur Rooni*	

SOLID EDGE ACADEMIC COPY

EMU

Leht: 3/6

Tähis: *TA 18/110585 1 01 03 S*



				Mõõt: 1:1
Teostas	Mihkel Keisk	Nimetus: Lisa 1. Regulaatori trükkplaadi joonis		
Kontrollis	Erkki Jõgi, Vahur Rooni			
Kinnitas	Erkki Jõgi, Vahur Rooni			
EMU		Leht: 4/6	Tähis: TA 18/110585 1 01 04 0	

SOLID EDGE ACADEMIC COPY

C1	keraamiline kondensaator	0,1 uF
C2	keraamiline kondensaator	0,1 uF
C3	keraamiline kondensaator	22 pF
C4	keraamiline kondensaator	22 pF
C5	keraamiline kondensaator	0,1 uF
C6	elektrolüütiline kondensaator	0,33 uF
C7	keraamiline kondensaator	0,1 uf
D1	diood 1N4007	
F1	sulavkaitse 5*20mm	12 A, sulavkaitse + pesa
F2	sulavkaitse 5*20mm	0,5 A, sulavkaitse + pesa
IC1	temperatuuriandur LM35	LP korpus
IC2	mikrokontroller atmega328-P	28-PDIP korpus
IC3	pingeregulaator LM7805	TO-220 korpus
J1	ühenduspesa 21*5,5mm	
JP1	piikriba 1x1	vaba viik
JP2	piikriba 1x2	sumisti ühendus
JP3	piikriba 1x2	maandus ühendus
JP4	piikriba 1x2	5 V ühendus
JP5	piikriba 1x5	reaktoriväline termopaar
JP6	piikriba 1x5	reaktorisisene termopaar
JP7	piikriba 1x4	LCD I2C mooduli ühendus
JP8	piikriba 1x6	programmaatori ühendus
JP9	piikriba 1x9	lülitimooduli ühendus
Tähis	Nimetus	Märkused

Teostas	Mihkel Keisk	Nimetus: Lisa 1. Regulaatori komponentide tükitabel #1
Kontrollis	Erkki Jõgi, Vahur Rooni	
Kinnitas	Erkki Jõgi, Vahur Rooni	
	Leht: 5/6	Tähis: TA 18/110585 1 01 05

SOLID EDGE ACADEMIC COPY

K1	relee SPDT 12A 250V	12VDC, 45 mA
LED1	3mm valgusdiood	
OK1	Optoisolaator MOC3041 või MOC3061	730A-04 korpus
Q1	kvartsostsillaator	16 MHz
R1	takisti 1/4 W	330 oom
R2	takisti 1 W	330 oom
R3	takisti 1 W	380 oom
R4	takisti 1/4 W	10 k oom
R5	takisti 1/4 W	100 oom
R6	takisti 1/4 W	1 k oom
S1	mikronupp 6*6*3,5mm nupp 6mm	
T1	Sümistor BTA 41	TOP3 korpus
T2	transistor 2N1893	TO-39 korpus
X1-1, X1-2	kruviklemm 8,5mm viikude vahega	
Tähis	Nimetus	Märkused

Teostas	Mihkel Keisk	Nimetus: Lisa 1. Regulaatori komponentide tükitabel #2
Kontrollis	Erkki Jõgi, Vahur Rooni	
Kinnitas	Erkki Jõgi, Vahur Rooni	

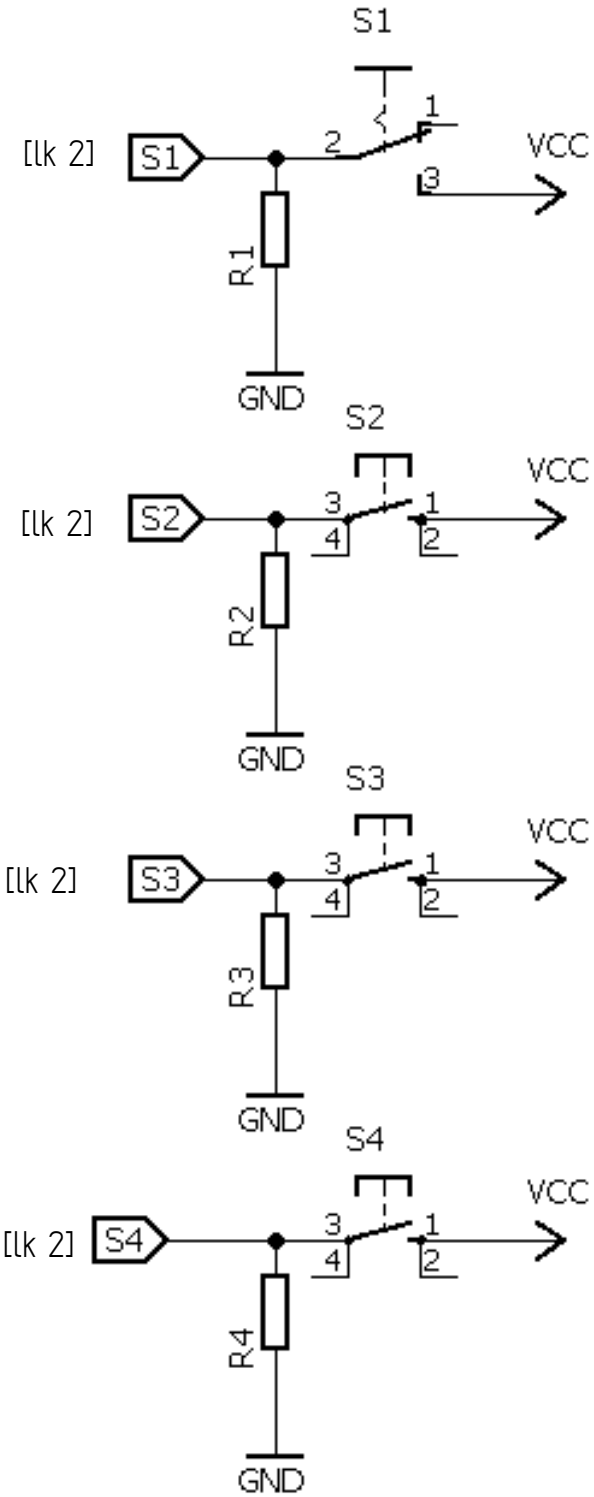
SOLID EDGE ACADEMIC COPY

EMÜ

Leht: 6/6

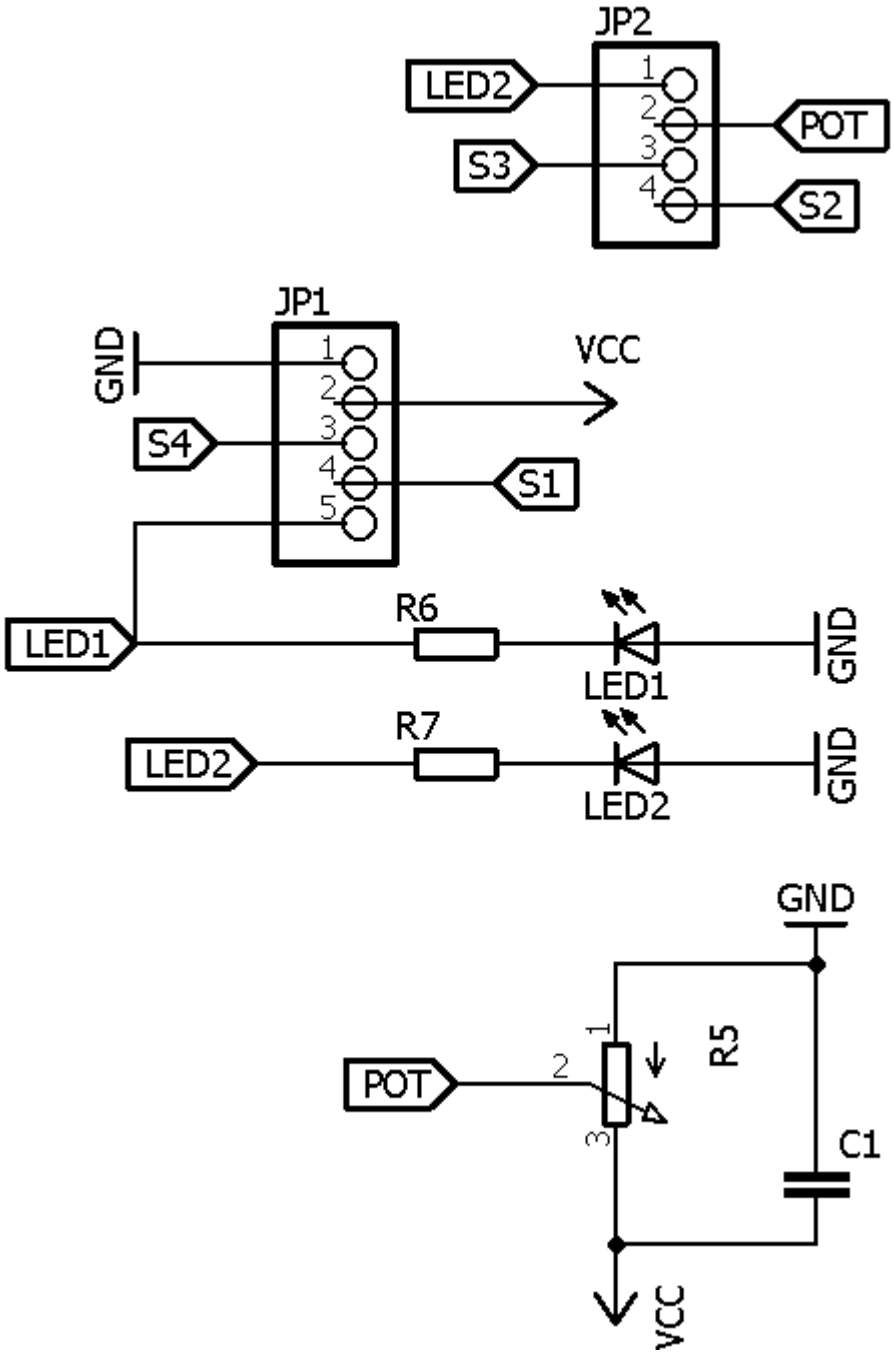
Tähis: TA 18/110585 1 01 06

Lisa 2. Lülitimooduli skeemid



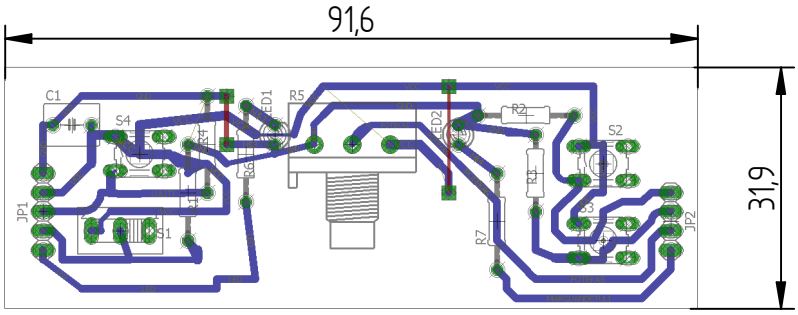
Teostas	Mihkel Keisk	Nimetus: Lisa 2. Lülitite elektriskeem	
Kontrollis	Erkki Jõgi, Vahur Rooni		
Kinnitas	Erkki Jõgi, Vahur Rooni		
EMU		Leht: 1/4	Tähis: TA 18/110585 2 01 01 S

SOLID EDGE ACADEMIC COPY



Teostas	Mihkel Keisk	Nimetus:	Lisa 2. Lülitimooduli elektriskeem
Kontrollis	Erkki Jõgi, Vahur Rooni		
Kinnitas	Erkki Jõgi, Vahur Rooni		
EMU		Leht: 2/4	Tähis: TA 18/110585 2 01 02 S

SOLID EDGE ACADEMIC COPY



				Mõõt: 1:1
Teostas	Mihkel Keisk	Nimetus: Lisa 2. Lülitimooduli trükkplaadi joonis		
Kontrollis	Erkki Jõgi, Vahur Rooni			
Kinnitas	Erkki Jõgi, Vahur Rooni			
EMU		Leht: 3/4	Tähis: TA 18/110585 2 01 03 0	

SOLID EDGE ACADEMIC COPY

C1	keraamiline kondensator	0,1uF
LED1	3mm valgusdiood	
LED2	3mm valgusdiood	
JP1	kaabliühendus 1x5	pistik
JP2	kaabliühendus 1x4	pistik
R1	1/4 W takisti	10 k oom
R2	1/4 W takisti	10 k oom
R3	1/4 W takisti	10 k oom
R4	1/4 W takisti	10 k oom
R5	Potentsiomeeter	1 k oom
R6	1/4 W takisti	330 oom
R7	1/4 W takisti	330 oom
S1	kahepositsiooniline DIP liuglüliti	3 viiguga SPDT
S2	mikronupp 6*6*9.5mm nupp 6mm	
S3	mikronupp 6*6*9.5mm nupp 6mm	
S4	mikronupp 6*6*9.5mm nupp 6mm	
Tähis	Nimetus	Märkus

Teostas	Mihkel Keisk	Nimetus: Lisa 2. Lülitimooduli komponentide tükitabel
Kontrollis	Erkki Jõgi, Vahur Rooni	
Kinnitas	Erkki Jõgi, Vahur Rooni	

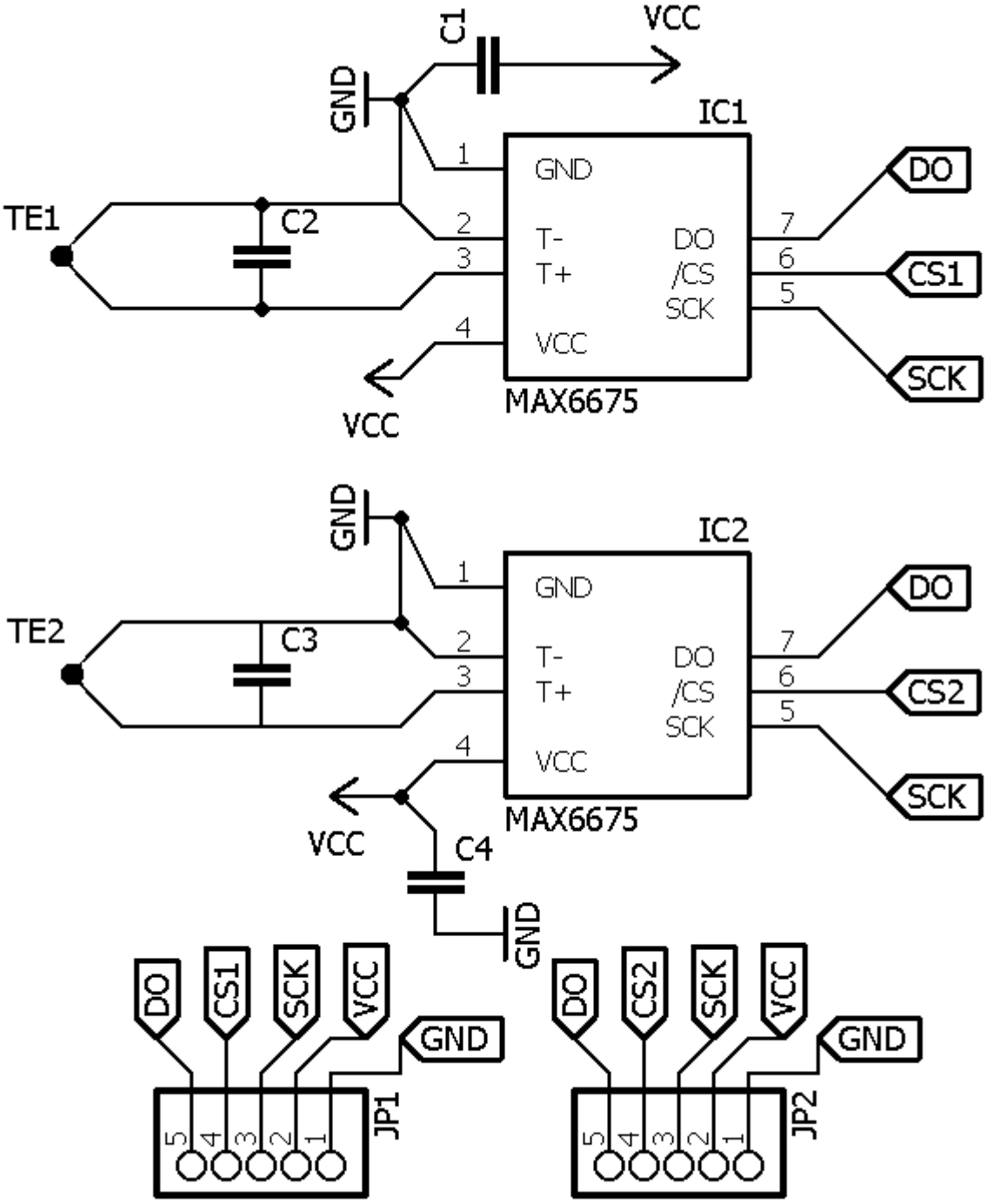
SOLID EDGE ACADEMIC COPY

EMU

Leht: 4/4

Tähis: TA 18/110585 2 01 04

Lisa 3. Analoog-digitaalmuunduri elektriskeem



Teostas	*Mihkel Keisk*	*Nimetus:* *Lisa 3. Termopaari analoog-digitaalmuunduri MAX6675 elektriskeem*
Kontrollis	*Erkki Jõgi, Vahur Rooni*	
Kinnitas	*Erkki Jõgi, Vahur Rooni*	
		Leht: 1/2
		Tähis: *TA 18/110585 3 01 02 S*

SOLID EDGE ACADEMIC COPY

C1	SMD kondensaator	0,1 uF
C2	Keraamiline kondensaator	0,1 uF
C3	SMD kondensaator	0,1 uF
C4	Keraamiline kondensaator	0,1 uF
IC1	MAX6675 mikroskeem	8-SO pakett
IC2	MAX6675 mikroskeem	8-SO pakett
JP1	piikriba 1x5	pistik
JP2	piikriba 1x5	pistik
TE1	K-tüüpi termopaar	Küttekeha termopaar
TE2	K-tüüpi termopaar	Reaktorisisene termopaar
Tähis	Nimetus	Märkused

Teostas	*Mihkel Keisk*	*Nimetus:* *Lisa 3. Termopaari analoog-digitaalmuunduri MAX6675 tükitabel*
Kontrollis	*Erkki Jõgi, Vahur Rooni*	
Kinnitas	*Erkki Jõgi, Vahur Rooni*	

SOLID EDGE ACADEMIC COPY

EMU

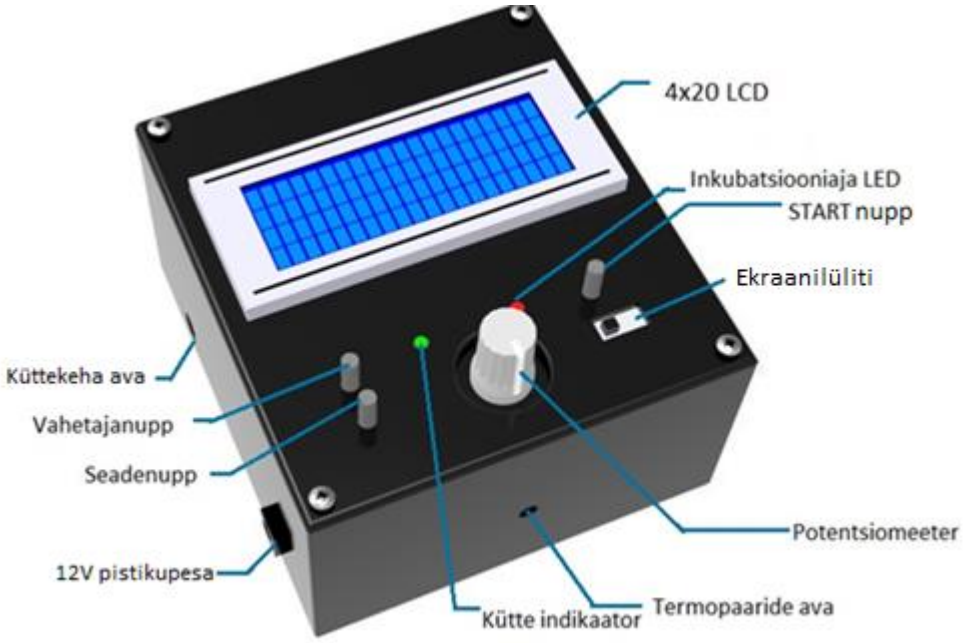
Leht: 2/2

Tähis: *TA 18/110585 3 01 02*

Lisa 4. Regulaatori kasutusjuhend

Kaskaadjuhtimisega temperatuuriregulaatori kasutusjuhend

SEADME ÜLESEHITUS

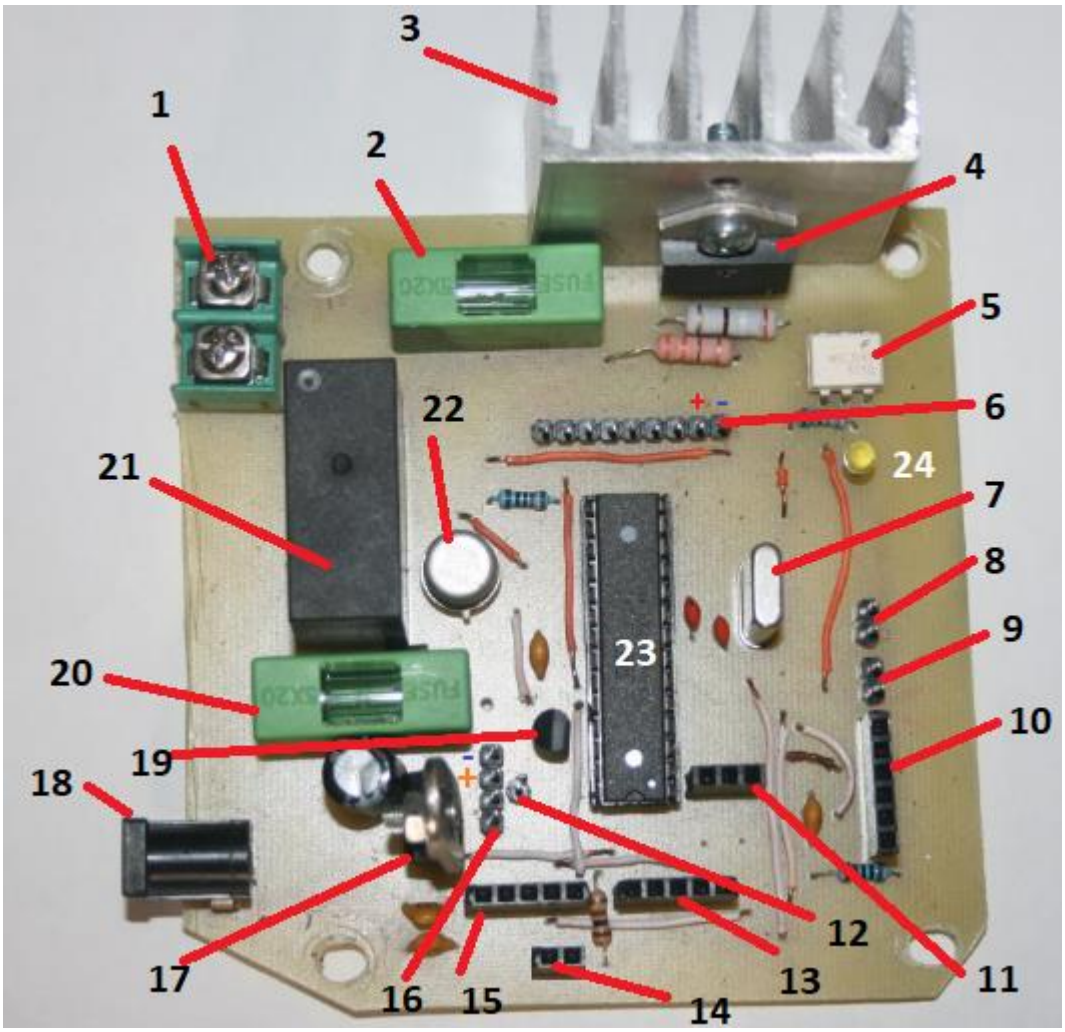


SPETSIFIKATSIOON

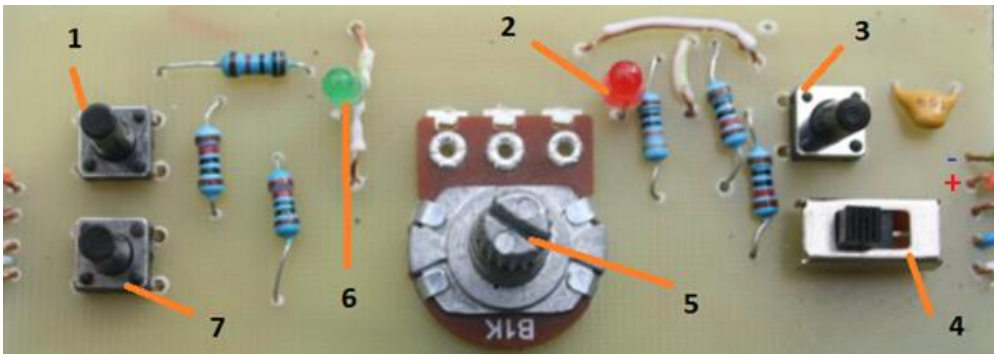
Tabel 1. Regulaatori andmed

Tähis	parameeter	väärtus	Ühik
U_I	SISENDPINGE	9…12	V
U_O	TÖÖPINGE	5	V
I_F	TÖÖVOOL	0,4	A
U_T	KÜTTEKEHA SISENDPINGE	230	V
I_T	KÜTTEKEHA MAKSIMAALNE VOOL	12	A
	TERMOPAARI TÜÜP	K	
T_{1max}	MAKSIMAALNE KÜTTEKEHA TEMPERATUUR	230	°C
T_{2max}	MAKSIMAALNE REAKTORISISENE TEMPERATUUR	210	°C
L	KORPUSE MÕÕTMED	120x110x63	mm

TRÜKKPLAATIDE ÜLESEHITUS



Joonis 1. Regulaatori trükkplaat. 1 – kruviklemm; 2 – sulavkaitse, 12 A; 3 – jahutusradiaator 4 – sümistor BTA41; 5 – optoisolaator MOC3041; 6 – (1 x 9) piikriba, lülitite mooduli ühendus; 7 – kvartsostsillaator 16 MHz; 8 – 1 * 2 piikriba, 5 volti; 9 – (1 x 2) piikriba 0V ühendus; 10 – (1 x 6) piikriba, programmaatoriga ühendamiseks; 11 – (1 * 3) piikriba, lähtestus regulaatorile; 12 – (1 x 1) piikriba, vaba viik; 13 – (1 x 5) piikriba, reaktorivälise analoog-digitaalmuunduriga MAX6675 ühendamiseks; 14 – (1 x 2) piikriba, sumisti ühendus; 15 – (1 x 5) piikriba, reaktorisisese analoog-digitaalmuunduriga MAX6675 ühendamiseks; 16 – (1 x 4) piikriba, LCD I2C mooduliga ühendamiseks(+ ja + klemm on joonisel näidatud); 17 – pingeregulaator LM7805; 18 – 9 – 12 V ühenduspesa; 19 – temperatuuriandur LM35; 20 – sulavkaitse 0,5 A; 21 – Relee; 22 – transistor; 23 – mikrokontroller Atmega328-p; 24 – 3 mm LED.

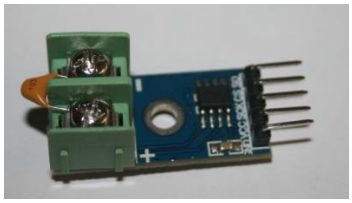


Joonis 2. Lülitite moodul. 1 – Vahetajalüliti; 2 – inkubatsiooniaja *LED*; 3 – START lüliti; 4 – ekraanivahetuslüliti; 5 – potentsiomeeter; 6 – kütte indikaator; 7 – seadelüliti.

TERMOPAARIDE ÜHENDAMINE

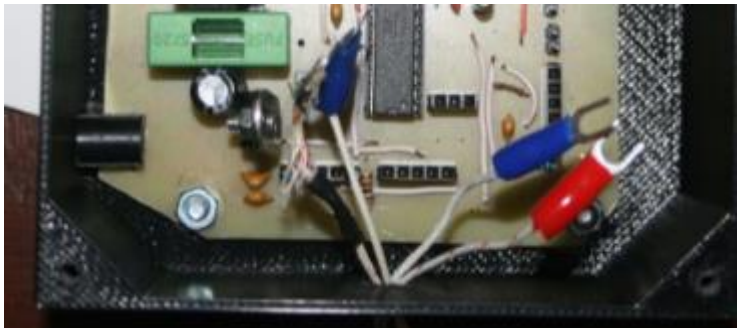
1. Termopaaride ühendamiseks tuleb avada korpus ning eemaldada LCD ja lülitimooduli ühendusjuhtmed, et pääseda ligi regulaatori trükkplaadile.

2. Seadmele lisatavad termopaarid peavad olema K-tüüpi. Termopaari analoog-digitaalmuunduriks on MAX6675 mikroskeem. Termopaaride ühendamisel MAX6675 mooduliga tuleb lisada + ja – kruviklemmide vahele keraamiline kondensaator mahtuvusega 0,01…0,1 µF .



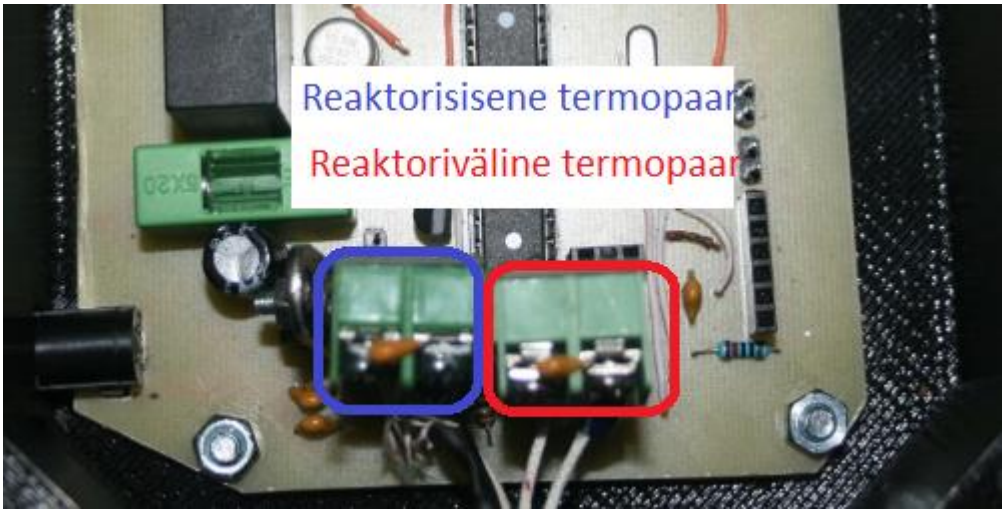
Joonis 3. MAX 6675 analoog-digitaalmuunduri moodul.

3. Termopaari ühendusjuhtmed tuleb esmalt sisestada korpuse avasse ilma moodulita. Ühendusjuhtmed kinnitada mooduliga MAX6675.



Joonis 4. Termopaari juhtmete sisestamine korpusesse.

4. MAX6675 moodul tuleb ühendada trükkplaadile õiget pidi, vastasel juhul edastatakse signaale valeledele viikudele.



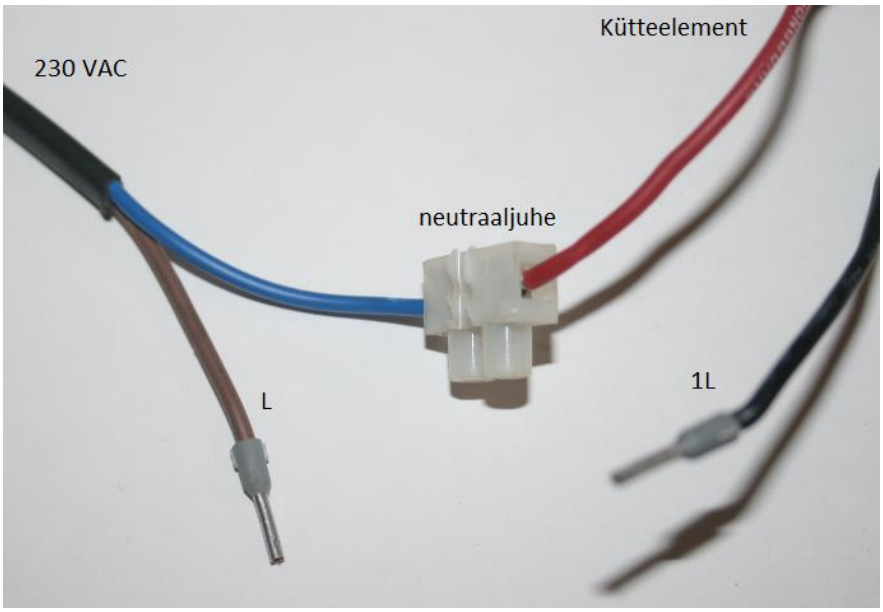
Joonis 5. MAX6675 ühendamine trükkplaadile.

5. Korrata eelnevaid samme teise termopaariga, Termopaari moodilite ühendamisel tuleb veenduda, et reaktorisisene ja väline termopaar on õigetes kohtades joonisel näidatult.

KÜTTEELEMENDI ÜHENDAMINE

Kütteelemendi ühendamisel või lahtiühendamisel on kõige olulisem kontrollida, et kütteelemendi voolukaabel on vooluvõrgust lahti ühendatud.

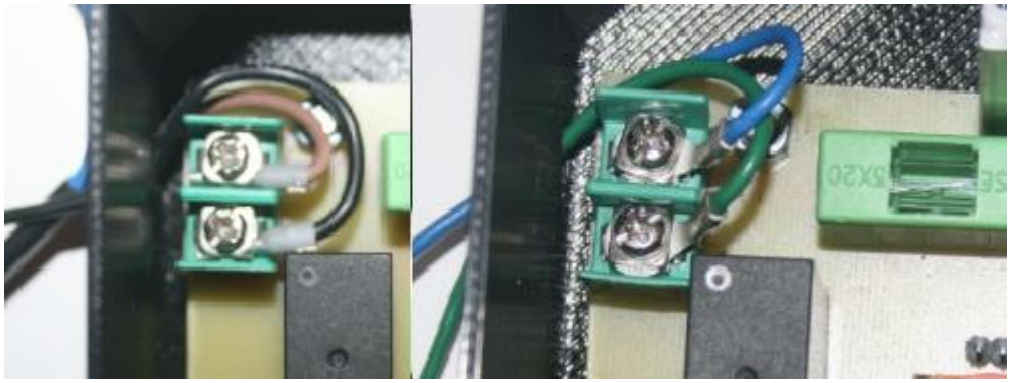
1. Eemaldada kaablilt varjestus joonisel näidatult. Jätta ühe faasi kummalegi poole varu 50mm. Varjestus traadilt eemaldada mõlemalt faasi poolelt 10mm varuga. Nii toite, kui ka kütteelemendi kiudkaablid sisestada hülssi või ühendada rõngasklemmiga. Neutraaljuhe ühendada klemmliistu abil.



Jooni 6. Kütteelemendi kaabel.

2. Sisestada ühe faasi mõlemad varjestatud kaablid avasse.

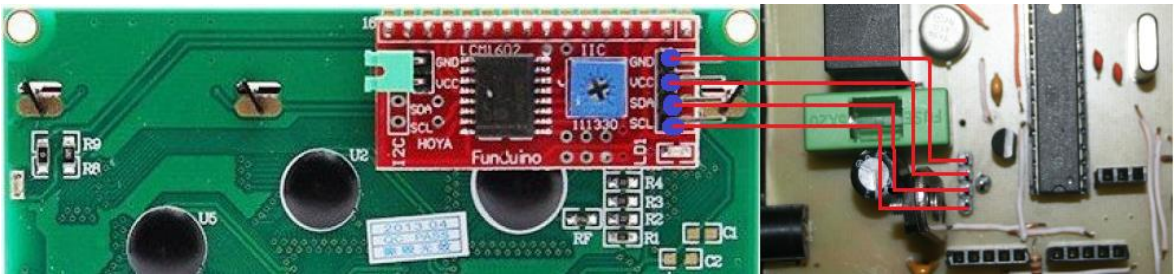
3. Keerata lahti kruviklemmid ning sisestada hülsid või spetsiaalsed rõngasklemmid kruviklemmidele. Veenduda, et kumbki faasiühendus pole üksteisega kontaktis, vastasel juhul läbib vool kütteelementi otse, maksimaalse koormusega. Kindlustuseks võib kontrollida testeriga juhtivust kummalgi klemmil. Juhtivust ei tohi olla. Kuna tegemist on vahelduvvooluga, siis ei ole oluline kumma kruviklemmiga on kütteelement ning toide ühendatud.



Joonis 7. Kütteelemendi ja faasijuhtme ühendamine trükkplaadile.

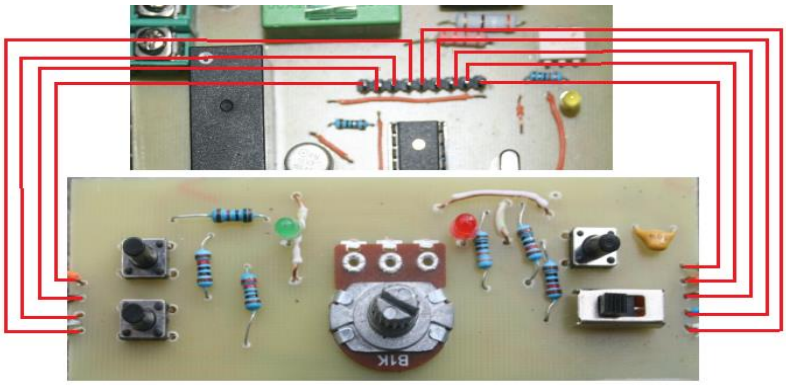
LCD ühendamine

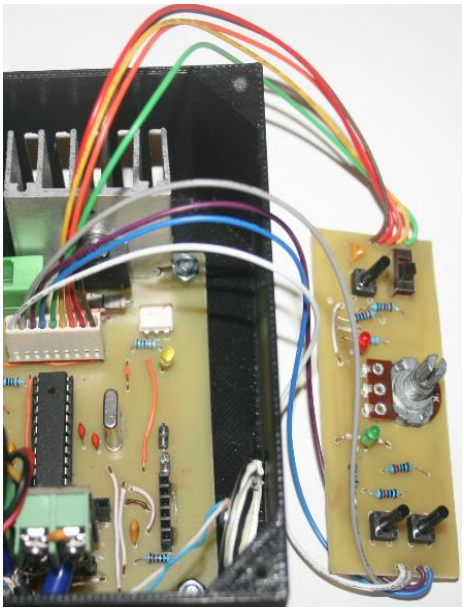
Vedelkristallekraani I^2C mooduli ühendus regulaatori trükkplaadiga.



Joonis 8. LCD ühendamine trükkplaadile.

Lülitimooduli ühendamine

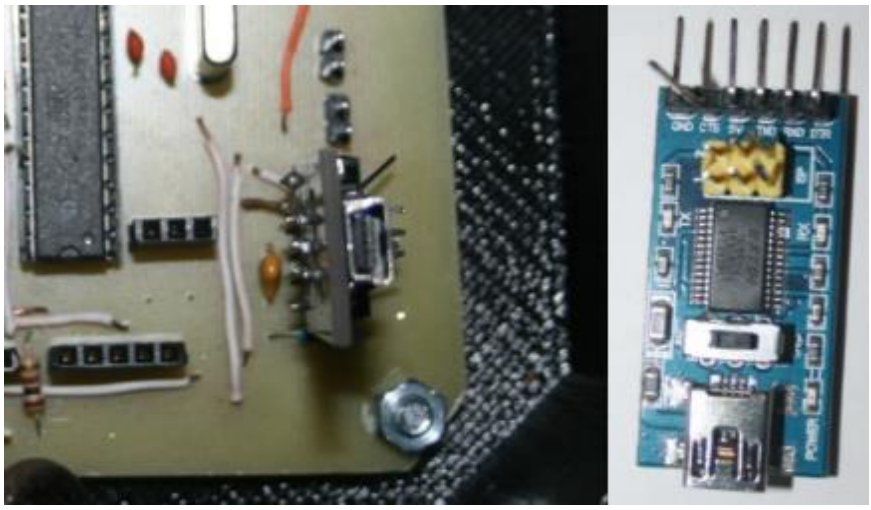




Joonis 9. Lülitimooduli ühendamine trükkplaadile.

PROGRAMMAATORI ÜHENDAMINE

Seadme programmeerimiseks saab kasutada RS232 programmaatorit. Programmaator ühendada joonisel näidatult.

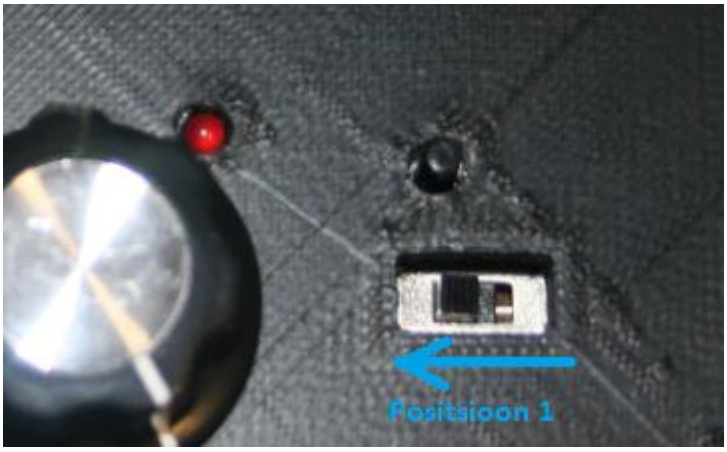


Joonis 10. Programmaatori ühendamine trükkplaadile.

REGULAATORI SEADISTAMINE

Regulaatori kasutamiseks tuleb veenduda, et termopaarid ja kütteelement on ühendatud.

Kütteväärtuste ehk inkubatsiooniaja ja sättetemperatuuri muutmiseks peab ekraanivahetuslüliti olemas positsioonis 1.



Joonis 11. Ekraanilüliti positsioon 1.

Ekraanile kuvatav kiri käseb valida inkubatsiooniaja ja sättetemperatuuri. Seadmel on püsimälus meeles viimasena kasutataud väärtused, mis on joonisel allajoonitud sinise värviga. Paremal all nurgas kuvatakse hetkel mõõdetavad temperatuurid termopaaridel (punaste joontega allajoonitud).

Joonis 12. Sättetemperatuuri ja inkubatsiooniaja valimise kuva.

Vahetaja nupu vajutamisel ilmub ekraanile + märk real 3 või 4, mis annab ühtlasi märku, kumba parameetrit hetkel muudetakse. Plussmärgi ilmumisel kummalegi reale, aktiveeritakse potentsiomeeter, mille reguleerimisel on võimalik seadetmperatuuri ja inkubatsiooniaja väärtuseid muuta. Sisse on programmeeritud sättetemperatuuri valiku piirang, milleks on …200 °C, ning inkubatsiooniaja piirang, milleks on 60 minutit.

Joonis 13. Lülitite nimetused.

Joonis 14. Vahetajalüliti vajutamisel liigutatakse + märki ridade 3,4 vahel.

Pärast väärtuse valimisel potentsiomeetri abil, tuleb väärtuse salvestamiseks püsimälusse vajutada seadenuppu.

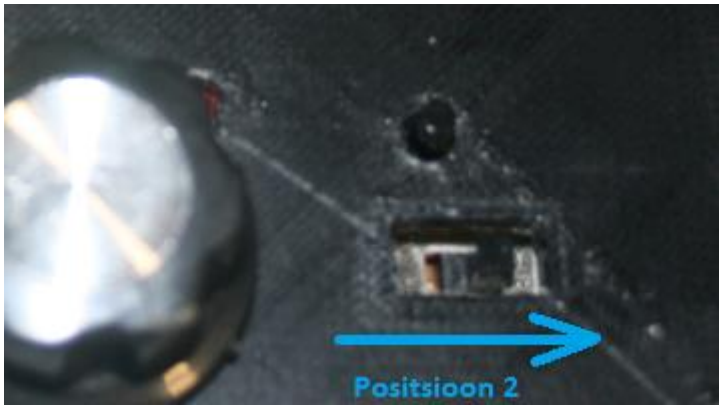
Joonis 15. Seadelüliti vajutamisel salvestatakse väärtus püsimälusse.

Kütmise alustamiseks on nupp START, mille vajutamisel käivitatakse kütmine ja ühtlasi ka reguleerimise algoritm. Mittesoovitud nupu vajutamisel saab seadme kütmist lõpetada pistiku eemaldamisega.

VÕIMENDUSTEGURITE MUUTMINE

Enne antud osa kirjeldust tuleb märkida, et antud väärtuste muutmisel ei tööta regulaator konkreetsel süsteemil häälestatud ja ettenähtud viisil.

Juhul, kui on tarvis reguleerimisparameetreid muuta tuleb seada ekraanilüliti positsioonile 2. Antud ekraan (joonis 17.) kuvatakse vaid siis, kui kütmist ei toimu.



Joonis 16. Ekraanilüliti positsioon 2.

Ekraanile ilmub esimesel real kütmise alguses kiiremalt reageeriva kontuuri reguleerimise võimendustegurid Kp1 ja Ki1. Teisel real on aeglaselt reageeriva kontuuri reguleerimise võimendustegurid Kp2 ja Kd2.

Joonis 17. Võimendustegurite kuvamine.

Kp1, Kp2, Ki1, Kd2 muudetud väärtused salvestatakse seadme püsimälusse. Võimendustegurite väärtuseid on võimalik muuta vahemikus 0…10. Ning resolutsioon on igal väärtusel 0,1.

Kolmandal real kuvatakse temperatuur, kui palju soovitakse kiirema reageerimisega kontuuri soojemaks kütta aeglasemast, st küttekeha, kui palju soojemaks, kui reaktorisisene temperatuur. Vaikimisi on antud väärtus 30, kuid on võimalik muuta antud väärtust vahemikus 1…40. Sarnaselt eelnevale salvestatakse antud väärtus püsimälusse.

Neljandal real on toodud juhttoime piirväärtus. Juhttoime maksimaalne väärtus, mis on vaikimisi seatud ~50 % ehk 125. Pulsilaiuse väärtust saab muuta vahemikus 1…255. Sarnaselt eelnevale salvestatakse antud väärtus püsimälusse.

Väärtuste vahetamiseks tuleb esmalt vajutada vahetaja nuppu, mis kuvab + märgi muudetava parameetri juurde. Vahetaja nuppu uuel vajutamisel liigutatakse + märk järgmisele muudetavale parameetrile. Kui + märk ilmub soovitud väärtuse juurde, tuleb see aktiveerida, START nupuga. Start nupu vajutus teeb parameetri aktiivseks ning seda on võimalik potentsiomeetri abil muuta. Uuendatud väärtus salvestatakse automaatselt püsimälusse. Kui soovitakse muuta mitut väärtust, siis pärast ühte muudetud väärtust tuleb vajutada uuesti START nuppu, et deaktiveerida potentsiomeeter.

Joonis 18. Võimendustegurite muutmine.

REGULAATORI TÖÖTAMINE

Enne kütmise sisse lülitamist nupuga START tuleb veenduda, et termopaarid on küttekehale ja köetavale süsteemile fikseeritud.

Kütmise käimisest annab märku roheline, indikaatortuli. Inkubatsiooniajast antakse märku punase indikaatortulega.

Joonis 19. Indikaatortuled.

Kütmisprotsessi kestel on võimalik jälgida antud ekraanilt hetketemperatuure termopaaridel. Samuti on ekraanil kuvatud seadetemperatuur ja seadeinkubatsiooniaeg. Inkubatsiooniaja kestel käivitatakse loendur, mis hakkab ekraanil kuvama, mitu minutit on inkubatsiooniaja lõppemiseni jäänud.

Selleks, et jälgida kütmisega seotud väärtuseid tuleb ekraanilüliti vahetada positsioonile 2 (joonis 16.).

Kütmisprotsessi kestel on võimalik jälgida temperatuuri regulaatori korpuses - Tseade. Teisel real kuvatakse hetkel käimasolev algoritmijärk, milleks võib olla: P-kütmine, PI-kütmine, sättetemperatuur saavutatud, kütmine lõpetatud ja plahvatuseks valmis.

Joonis 20. Kütmise käigus kuvatavad parameetrid.

Kolmandal real kuvatakse hetkel kasutusel olevad PID regulaatori võimendustegurite väärtused.

Neljandal real kuvatakse reaalajas hetkel kütteelemendile rakenduv juhttoime väärtus. Juhttoime väärtus on mikrokontrollerilt tulev pulsilaiuse väärtus, mis edastatakse optoisolaatorile, mis omakorda reguleerib sümistori abil küttekeha võimsust.
Kütmise ajal on ülejäänud nuppude funktsioonid välja lülitatud. Seadme lähtestamiseks või töö peatamiseks tuleb eemaldada 12 V sisendpinge pistik, mille tulemusel kommuteeritakse lahti relee ühendused.

SEADME OHUTUS

Kasutaja, ja süsteemi ohutuse tagamiseks on elektriahelas ülekoormuskaitsmed. Kütteelemendiga jadamisi ühendatud ülekoormuskaitse puruneb 12 A voolu ületamisel.

Kütmise ajal on ohutuse tagamiseks lisatud programmisisesed ohutuslaused. Temperatuuridele on lisatud maksimaalsed temperatuuride piirangud, kus piiratakse maksimaalset temperatuuri küttekehal 230 °C ning reaktori sees 210 °C. Temperatuuri ületamisel antud väärtuse seadet ei lähtestata vaid kuvatakse ekraanile teade “temp ületas max piiri” ning piiratakse optoisolaatorile väljastatavat pulsi laiust. Temperatuuri langemisel alla maksimaalsete väärtuste jätkub kütmine eelneval režiimil.

Mikrokontrolleri koostööl analoogdigitallmuunduriga antakse märku vigasest termopaarist või katkenud termopaari ühendusest. Analoogdigitaalmuunduril MAX6675- le on programmeeritud edastada 0-bitine väärtus, kui termopaar ei anna muundurile korrektset EMJ väärtust. Selle tulemusel edastatakse ekraanile veateade ja keelatakse programmisiseselt optoisolaatorile signaali edastada. Reaktorisisese temperatuuri veateade on “T2 error #2”, küttekeha temperatuuri veateade on “T1 error #2”. Antud viga võib paraneda, millest tulenevalt seadet täiesti välja ei lülitata, juhul kui esineb üksik T = 0 väärtus, siis temperatuuri normaliseerimisel jätkatakse kütmist eelnevalt pooleli jäänud programmist

Juhul, kui unustatakse vms. põhjusel jäävad termopaarid kuumutussüsteemile ühendamatta, siis lülitatakse seade välja pärast teatud aja möödumist, et vältida kuumutussüsteemi ülekuumenemist. Kui temperatuur reaktoril ei tõuse esimese ühe minuti jooksul 10 $°C$, siis lülitatakse seade välja ja kui reaktorisisene temperatuur ei kasva esimese viie minuti jooksul 5 °C, siis lülitatakse seade välja. Veateated, mis kuvatakse on reaktorisisesel anduril “T1 error #2” ja küttekeha anduril “T1 error #1”.

Kõikide vigade tekkimisel antakse märku punase LED lambi ühe sekundilise intervallise vilkumisega.

VEAOTSING

Järgnevas jaotises on toodud tüüpilisemad vead, ning nende põhjustajad ja võimalikud parandused.

Ekraanil sobimatud märgid ja sümbolid. – on põhjustatud LCD I^2C mooduli juhtmete halvast ühendusest. Aitab juhtmete ühenduse parendamine. Samuti võib tuleneda mikrokontrolleri ülekoormamisest, mille parandamiseks piisab seadme taaskäivitamine.

Nupud ei tööta, LED lambid ei tööta või seade käitub ettearvamatult, alustades näiteks ise kütmist vms. – Põhjuseks on ilmselt lülitimooduli juhtmete halb ühendus. Kui mikrokontrolleri ja nuppude vahel puudub elektriline ühendus võib juhtuda nähtus nagu lüliti põrkumine, kus kontroller ei saa nupu olekust aru ning põhjustab väärlülitusi. Aitab juhtme kontakti parendamine. Kui eelnev ei aidanud, siis võib viga olla lülitimooduli ühendustes. Tuleb teha kindlaks, kas elektrivool jõuab lülititeni ning lampideni.

Kütmine kestab väga kaua –

1. Kütmise kestust võib põhjustada näiteks termopaari isolatsiooni kulumisest, kus termopaar satub kontakti teise metalliga, ning EMJ väärtust ei saa analoog-digitaalmuundur õigesti fikseerida. .Lahendusena – parandada termopaari isolatsioon.
2. Teise võimaliku põhjusena võib olla elektriline müra seadmes, mis põhjustab temperatuuri kõikumist .
3. Kolmanda võimaliku põhjusena on konkreetse kütteelemendi halb kontakt termopaariga, kus termopaar ei saa vahetut maksimaalset küttekeha või reaktorisisest temperatuuri vaid mõõdab madalamalt. Parandus –pingutada kütteelemendi kinnituspolte.

Seade ei käivitu, sellest annab märku LCD mittetöötamine. – Selle põhjuseks võib olla pingeregulaatorist LM7805 põhjustatult. Seda saab kontrollida, mõõtes väljalülitatud olekus LM7805 viikude takistusi (Input,GND— 0 Ω, OUTPUT, GND – 5 kΩ) või sisselülitatud olekus sisend, ja väljundpinget. Samuti kontrollida ülekoormuskaitse töötamist.

Viga võib olla ka trükkplaadi alalisvoolu radades, kus mõnel viigul puudub ühendus. Lahendus – testriga kontrollida komponendi väljaulatuva jootetud tipu suhtes sama raja piirkondi. Võib aidata üle jootmine. Vahetada välja pingeregulaator.

Seade töötab, lülitatakse kütma, kuid küttekeha ei tööta. Probleeme võib olla mitmeid, järgnevas nimekirjas võib olla põhjustaja. Veaotsingut tuleks teostada järgnevas järjekorras.

1. Kuulata, kas lülitamisel on kuulda relee kommuteerimist. Iga kütmise alguses ja lõpus kommuteeritakse elektromehaaniline relee, mille lülitamist on võimalik kuulda. Juhul, kui relee ei lülitu, on probleem seadmes olevas *NPN* transistoris;
2. Ülekoormuskaitse purunenud- tekkis lühis või ületati voolutugevus 12 A.
3. Trükkplaadi vahelduvvoolu rada ei juhi voolu. Testriga kontrollida väljalülitatud olekus radasid.
4. 1 W takistid, mõõta seadme väljalülitatud olekus. , takistite väärtused peavad olema trükkplaadilt mõõtes: ~330 Ω ja ~60 Ω;
5. Sümistor on riknenud. – vahetada välja sümistor bta41;

Termopaaridel esineb vigu kuus, kummagi termopaari kohta kolm: **T1/T2 error #1/#2/#3.** Vea tekkimisel kuvatakse ekraanile veateade.

Error#1 juhtub, kui jäetakse termopaarid kuumutussüsteemile ühendamatta. Kütmise alustamisel käivitatakse aja loendurid. Reaktorisisene temperatuur peab kasvama esimese 5 minuti jooksul vähemalt 5 kraadi. Kui temperatuur ei kasva pärast kütmise alustamist, on sisse programmeeritud relee väljalülitus, sest termopaar on ühendamatta. Reaktorivälisel peab temperatuur kasvama esimese minuti jooksul vähemalt 10 kraadi.

Error#2 tekib, kui MAX6675 ei saa ühendust mikrokontrolleriga. Aitab ühenduste ülejootmine.

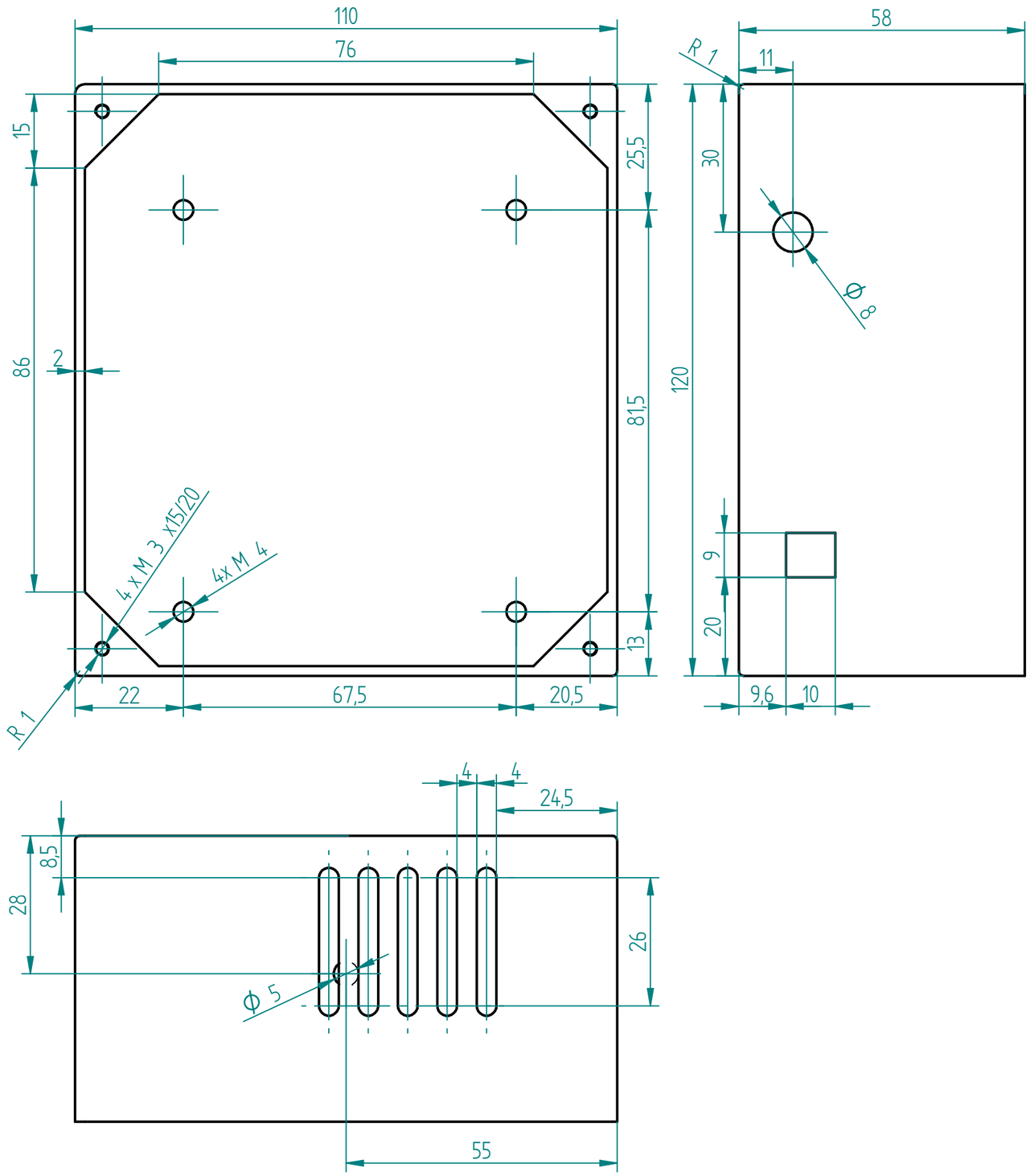
Error#3 on Analoog-digitaalmuundurist tulev mittearvuline väärtus. Juhul, kui termopaar on katki või ühendus *ADC* ja termopaari vahel on katkenud. Väljastatakse temperatuuri asemel NaN – Not a Number väärtus.

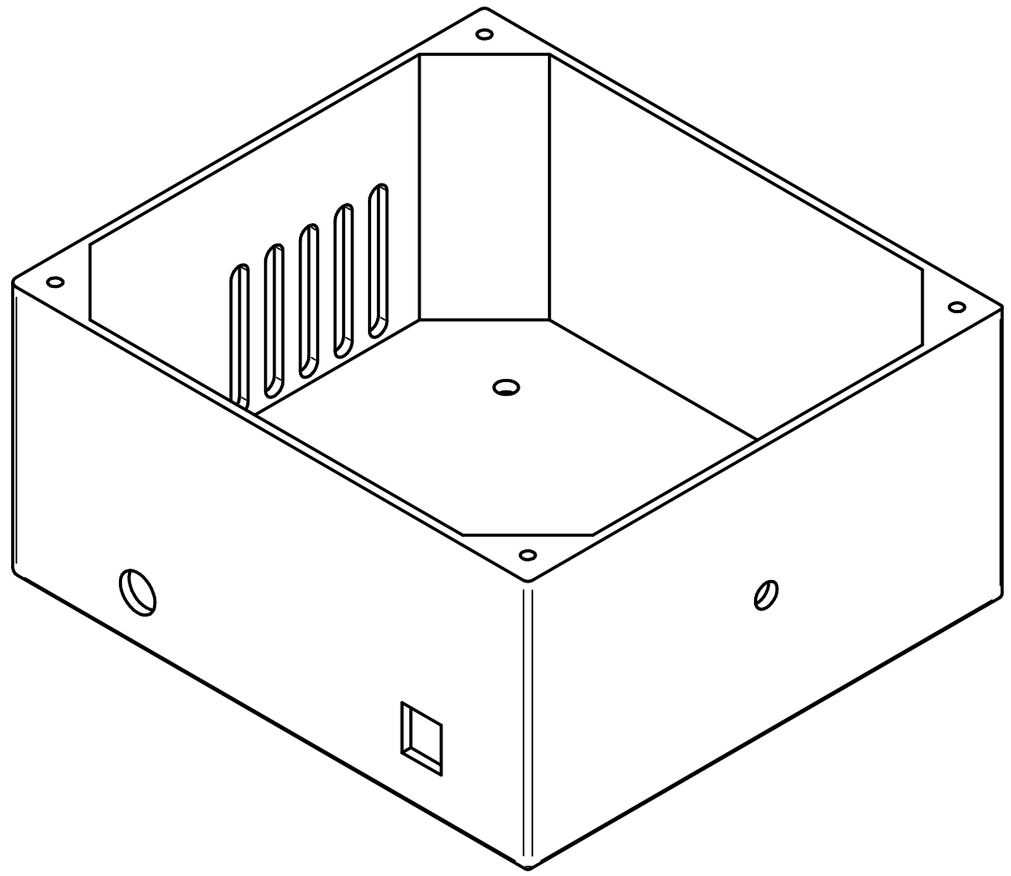
Errorite #2 ja #3 puhul releed välja ei lülitata. Regulaatori väljund kirjutatakse juhttoime väärtuse asemel 0-ks. Selle tulemusel valenäitudel ei toimu ülereguleermist.

Ekraan tühi – mitu termopaari viga esines korraga ja kuhjas mikrokontrolleri üle. Aitab vigade eemaldamine ja seadme taaskäivitamine.

Lisa 5. Korpuse detailijoonised ja koostejoonis

Ra 6,3

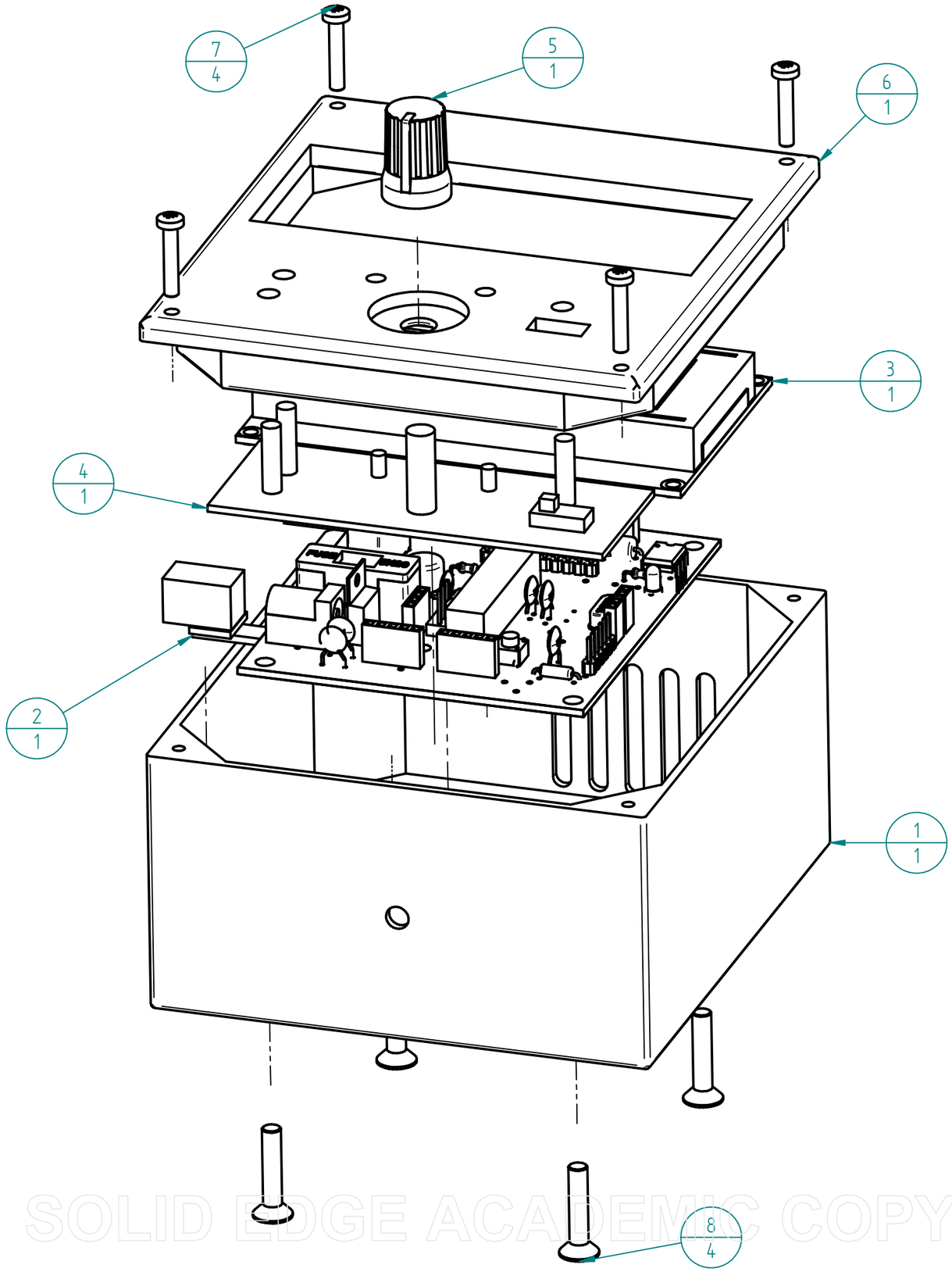


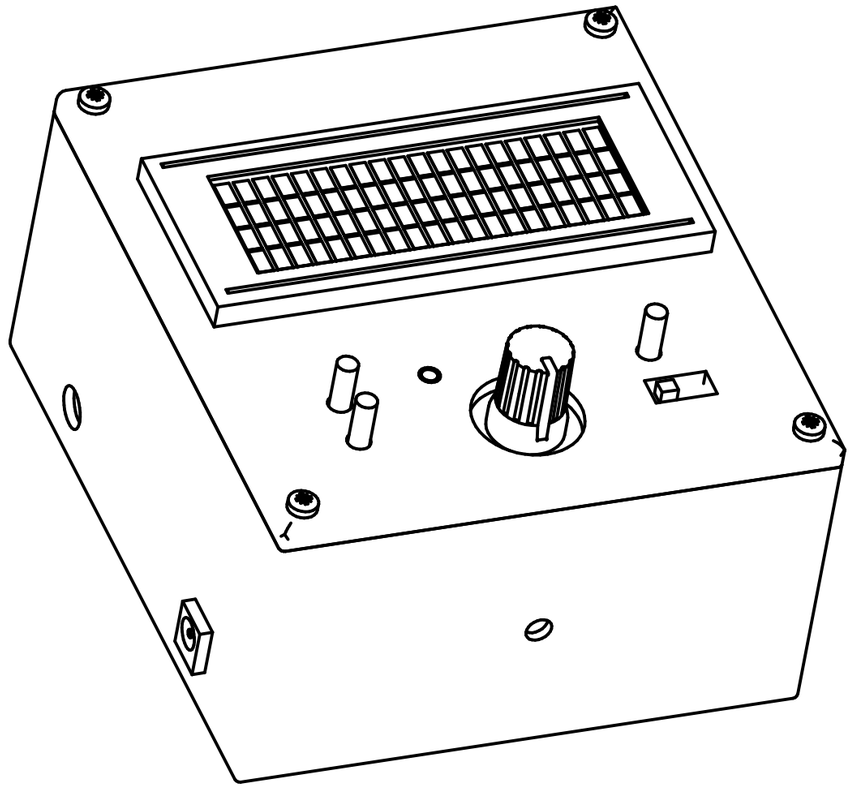


Põhja paksus – 5 mm
Detail 3D printida

	Materjal ABS	Märkimata piirhälbed DIN ISO 2768-mk	Mass 0,14kg	Mõõt 1:1
Teostas	Mihkel Keisk	Nimetus		
Kontrollis	Vahur Rooni, Erkki Jõgi	Lisa 5. Korpuse põhi		
Kinnitas	Vahur Rooni, Erkki Jõgi			
		Leht 1/3	Tähis TA 18/110585 5 01 01 D	

SOLID EDGE ACADEMIC COPY



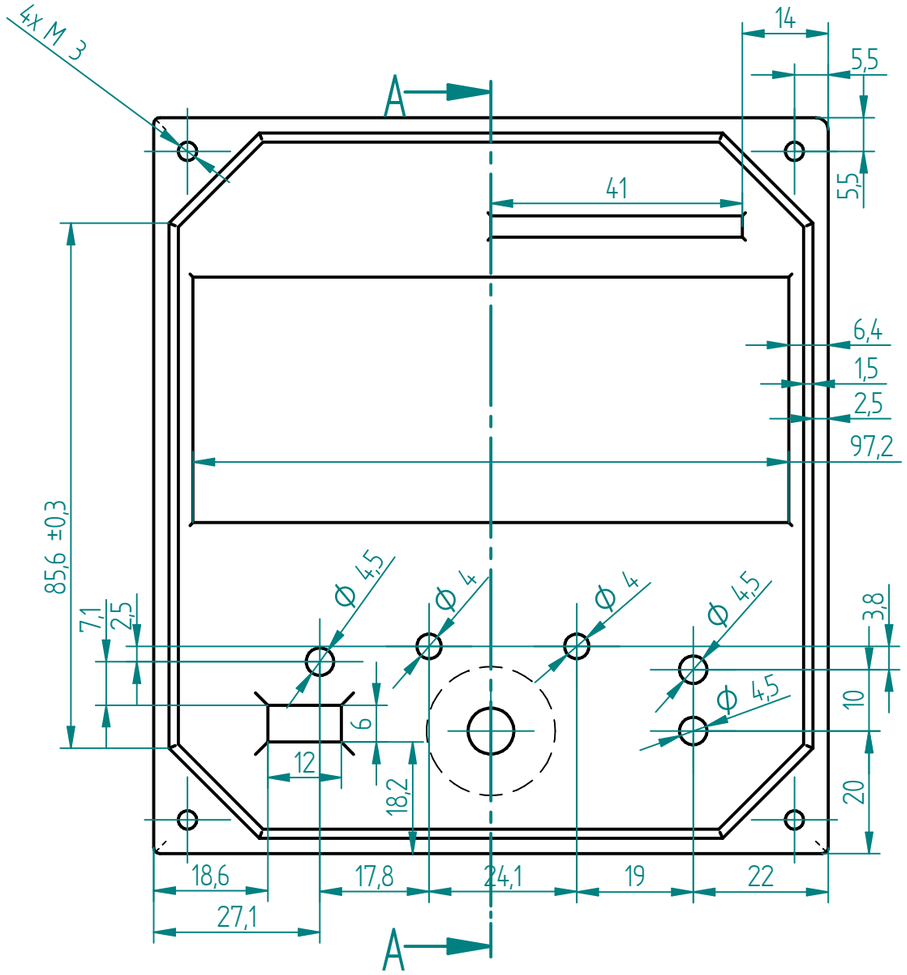


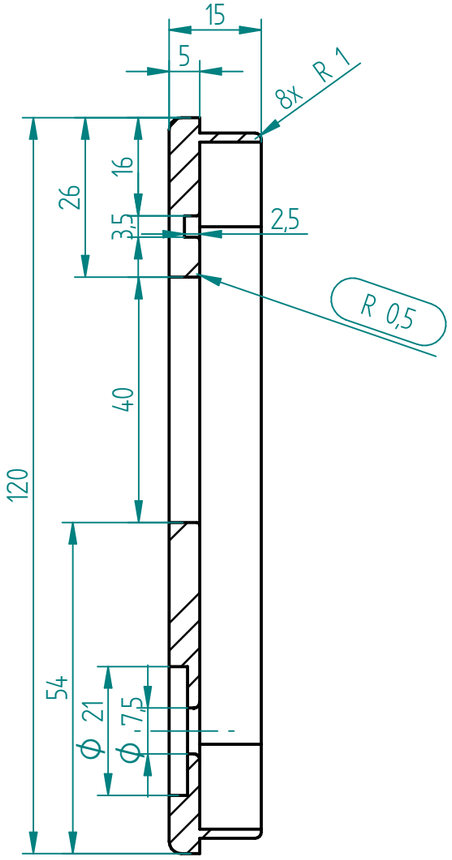
1		Korpuse põhi, ABS	TA 18/110585 C 01 01 D	1	
2		Regulaatori trükkplaat	TA 18/110585 A 01 04 D	1	
3		4x20 LCD		1	
4		Lülititemoodul	TA 18/110585 B 01 03 D	1	
5		Potentsiomeetri pöördnupp		1	
6		Korpuse kaan, ABS	TA 18/110585 C 01 02 D	1	
7		Polt M3		4	
8		Polt M4		4	
Osa	Väli	Nimetus, Materjal	Tähis	Hulk	Märkus

		Mass 0,4kg	Mõõt 1:1
Teostas	Mihkel Keisk	Nimetus: Lisa 5. Korpuse koostejoonis	
Kontrollis	Vahur Rooni, Erkki Jõgi		
Kinnitas	Vahur Rooni, Erkki Jõgi		
EMÜ		Leht: 3/3	Tähis: TA 18/110585 5 01 01 K

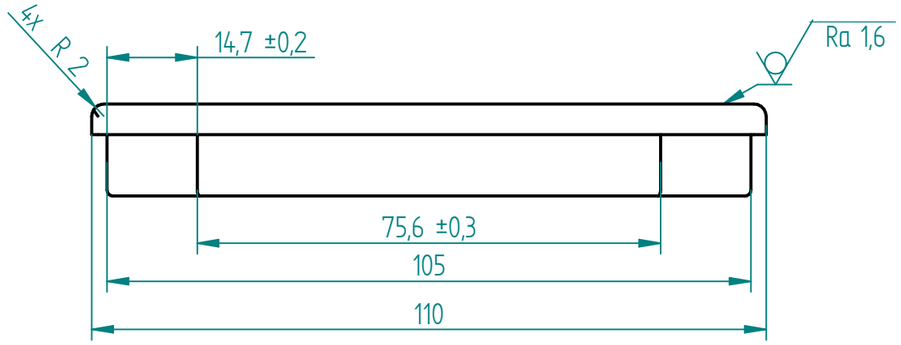
SOLID EDGE ACADEMIC COPY

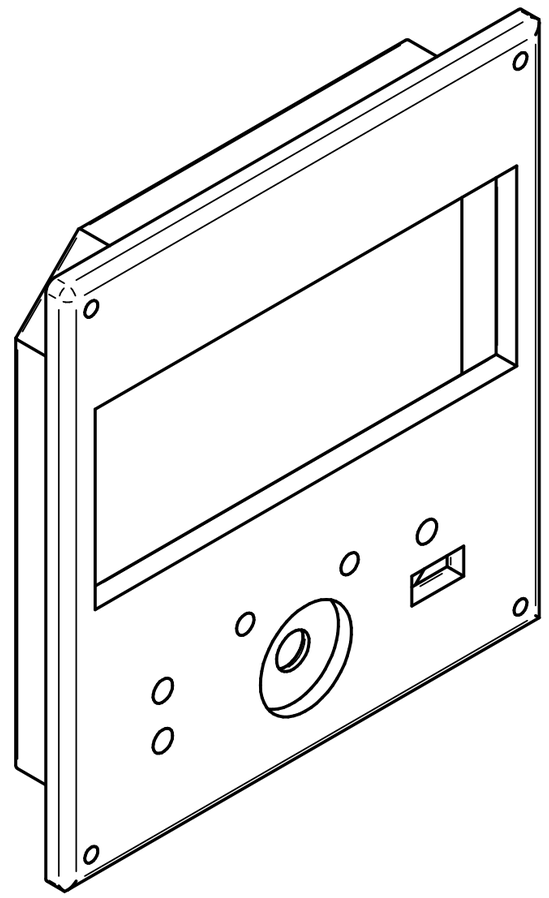
Ra 6,3





Lõige
A–A





Detail 3D printida
Kõik avad seespool nurgaraadiusega R = 0,5...1

Materjal: ABS		Märkimata piirhälbed: DIN ISO 2768-mk	Mass: 0,05kg	Mõõt: 1:1
Teostas: Mihkel Keisk	Nimetus: Lisa 5. Korpuse kaas			
Kontrollis: Vahur Rooni, Erkki Jõgi				
Kinnitas: Vahur Rooni, Erkki Jõgi				
EMU		Leht: 2/3	Tähis: TA 18/110585 5 01 02 D	

SOLID EDGE ACADEMIC COPY

Lisa 6. Programmkood

```
#include <SPI.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "max6675.h"
#include <PID_v1.h>
#include <elapsedMillis.h>
#include <PreciseLM35.h>
#include "EEPROM.h"
int aadress0 = 0;                                   //setpoint temperatuur
int aadress1 = 1;                                   //inkubatsiooniaeg
int aadress2 = 2;                                   //Kpalgus
int aadress3 = 3;                                   //Kpenne
int aadress4 = 4;                                   //Kienne
int aadress5 = 5;                                   //Kpparast
int aadress6 = 6;                                   //Kdparast
int aadress7 = 7;                                   //ülekütmise temp
int aadress8 = 8;                                   //PWM
elapsedMillis timeElapsed;                          //inkubatsiooniaja loendur
elapsedMillis termopaar1Elapsed;                    //välise termopaari ajaloendur
elapsedMillis termopaar2Elapsed;                    //sisese termopaari ajaloendur
boolean releelaks = false;
boolean kutmistalustati = false;                    //booleanid:
boolean setpoindinijoudmas = false;
boolean setpoindinijoutud = false;
boolean kutminelabi = false;
boolean plahvatus = false;
boolean labi = false;
boolean puhastamine = false;
boolean lcdpuhastamine = false;
boolean lcdpuhastamine2 = false;
boolean lcdpuhastamine3 = false;
boolean lcdpuhastamine4 = false;
boolean lcdpuhastamine5 = false;
boolean lcdpuhastamine6 = false;
boolean naitakorra = false;
boolean makstempyletatud = false;
boolean termopaar1juhtus = false;
boolean termopaar1juhtus2 = false;
boolean termopaar2juhtus = false;
boolean termopaar2juhtus2 = false;
boolean termopaar1juhtus3 = false;
boolean termopaar2juhtus3 = false;
boolean relee1juhtus = false;
boolean relee2juhtus = false;
 const long interval = 1000;                        //millid
const long interval2 = 300;
const long interval3 = 3000;
const long interval5 = 5000;
unsigned long previousMillis = 0;
unsigned long previousMillis2 = 0;
unsigned long previousMillis3 = 0;
unsigned long previousMillis4 = 0;
unsigned long previousMillis5 = 0;
unsigned long previousMillis6 = 0;
unsigned long previousMillis7 = 0;
unsigned long previousMillis8 = 0;
int moodatemp1;
int moodatemp2;
int Inkubatsiooniaeg;
long int sample = 5000;//pid kalkulatsiooni ajaintervall
double Setpoint, Input, Output;
double algusKp = (EEPROM.read(aadress2)) / 10.0, algusKi = 0, algusKd = 0;                                  //alustus parameetrid
double enneKp = (EEPROM.read(aadress3)) / 10.0, enneKi = (EEPROM.read(aadress4)) / 10.0, enneKd = 0;   //integraalajaga
double parastKp = (EEPROM.read(aadress5)) / 10.0, parastKi = 0, parastKd = (EEPROM.read(aadress6)) / 10.0;//pärast setpointi parameetrid
PID myPID(&Input, &Output, &Setpoint, algusKp, algusKi, algusKd, DIRECT);
unsigned long SetSampleTime (sample);               //pid arvutust korratakse iga 5 sek tagant
double PTterm, DTterm, ITterm;
float oldInput;
 int min1 = 0;                                      //min1 ja max1 abil piiratakse OUTPUT_PINi pulsilaiust
int max1 = EEPROM.read(aadress8);
LiquidCrystal_I2C lcd(0x3f, 20, 4);                 //20x4 lcd, A4 viik ühendatud lcd SDA viiguga, A5 ühendatud lcd SCL viiguga.
const int thermoDO = 12;
```

```
const int thermoCS = 4;
const int thermoCS2 = 2;
const int thermoCLK = 13;
#define csTC1  4
#define csTC2 2
MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);
MAX6675 thermocouple2(thermoCLK, thermoCS2, thermoDO);
const int PIN_OUTPUT = 3;                                   // led ja optosümistori pwm pin
const int marguandetuli = 7;                                //indikaatortuli: kas temperatuur on saavutatud ja inkubatsiooniaeg on alustanud
const int led = 8;                                          //led, mis annab märku, kas seade töötab.
const int relee = 9;                                        //relee, mis lülitab kütteelemendile voolu
int releestate = LOW;                 //relee olek, lülitub sisse D2 luliti vajutamisel.. lülitub välja inkubatsiooniaja täitumisel või vea tekkimisel
const int luliti = 5;                                       //lülitab küttekeha kütma. START lüliti
int buttonState;                                            // luliti olek
int lastButtonState = LOW;                                  // luliti eelmine olek
const int valikululiti = 6;                                 //ekraanivahetajalüliti
const int vahetaja = 11;                                    //lüliti, mis vahetab inkubatsiooniaja ja setpoint temperatuuri vahel.
int buttonState2;                                           // vahetaja nupu olek//
int lastButtonState2 = LOW;                                 // va eelmine olek
int buttonState3;
int lastButtonState3 = LOW;                                 // va eelmine olek
const int fiksaator = 10;                                   //lüliti, mis salvestab ink aja ja setpoint temp.
int buttonState4;
int lastButtonState4 = LOW;                                 // va eelmine olek
long int aeg ;                                              //inkubatsiooniaja muutuja
unsigned long lastDebounceTime = 0;  //
unsigned long debounceDelay = 50;    //
const int sensorPin = A0;                                   // potekas, millega saab muuta soovitud temperatuuri Setpoint või Inkubatsiooniaega
const int sumisti = A1 ;                                    //piezo buzzer
const int pinLM35 = A2;                                     //temp andur korpuses
PreciseLM35 lm35(pinLM35, DEFAULT);
int sound = 500;                                            // helisagedus 500Hz
int val = analogRead(A0);
int val2;                                                   //lcdle prinditav jääv setpoint
int val22;                                                  //lcdle prinditav jääv inkubatsiooniaeg
int val1;
int counter;
int counter2;
int voimendustegr;
int voim;
int tehtud;
float protsent;
int temp = EEPROM.read(aadress7);                           //reaktori välise termopaari ülekütmise temperatuur
int SP2 = EEPROM.read(aadress0);                            //sättetemperatuur

void setup() {
  Inkubatsiooniaeg = EEPROM.read(aadress1);
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.clear();
  myPID.SetMode(AUTOMATIC);
  pinMode(PIN_OUTPUT, OUTPUT);
  pinMode(marguandetuli, OUTPUT);
  pinMode(relee, OUTPUT);
  pinMode(led, OUTPUT);
  pinMode(sumisti, OUTPUT);
  pinMode(luliti, INPUT);
  pinMode(vahetaja, INPUT);
  pinMode(fiksaator, INPUT);
  pinMode(valikululiti, INPUT);
  digitalWrite(led, LOW);
  digitalWrite(relee, releestate);}

void PIDkalk() {                                                                    //pid kalkulatsioon
  if(!makstempyletatud && !termopaar1juhtus2 && !termopaar2juhtus2 && !termopaar1juhtus3 && !termopaar2juhtus3){
  myPID.Compute();
  analogWrite(PIN_OUTPUT, Output);}}

void termopaari1kaitse() {                                                          //juhul kui välimine termopaar pole ühendatud
  if (!relee1juhtus) {
    relee1juhtus = true;
    termopaar1Elapsed = 0;
```

```
   moodatemp1 = thermocouple.readCelsius();}
  if ((termopaar1Elapsed) / 1000 > 59 && 61 > (termopaar1Elapsed) / 1000 && thermocouple.readCelsius() - moodatemp1 < 10 &&
!termopaar1juhtus && thermocouple.readCelsius() < 120) {
   termopaar1juhtus = true;                                          //kui 1 minuti jooksul pole välistemp tõusnud 5 kraadi
   releestate = LOW;
   lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("T1 error #1");
   lcd.setCursor(0, 1);
   lcd.print("v2listermopaar");
   lcd.setCursor(0, 2);
   lcd.print("pole yhendatud -");
   lcd.setCursor(0, 3);
   lcd.print("taask2ivita seade");}}

void termopaari2kaitse() {                                           //juhul kui sisemine termopaar pole ühendatud, siis antakse pidev
helisignaal
  if (!relee2juhtus) {
   relee2juhtus = true;
   termopaar2Elapsed = 0;
   moodatemp2 = thermocouple2.readCelsius();}
  if ((termopaar2Elapsed) / 1000 > 300 &&  thermocouple2.readCelsius() - moodatemp2 < 10 && !termopaar2juhtus &&
thermocouple2.readCelsius() < 65 ) {
   termopaar2juhtus = true;                                          //kui sisese termopaari näit 5min jooksul pole temp tõusnud 10 kraadi
   releestate = LOW;
   lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("T2 error #1");
   lcd.setCursor(0, 1);
   lcd.print("sisetermopaar");
   lcd.setCursor(0, 2);
   lcd.print("pole yhendatud -");
   lcd.setCursor(0, 3);
   lcd.print("taask2ivita seade");}}

void termopaarnull() {                                               //kui mikrokontroller ei saa ühendust ADC MAX6675-ga.Väljastatakse 0-bitti.
  if  (0 >= thermocouple.readCelsius()) {
   if (!termopaar1juhtus2) {
     lcd.clear();}
   termopaar1juhtus2 = true;
   lcd.setCursor(2, 0);
   lcd.print("T1 error #2");
   lcd.setCursor(3, 1);
   lcd.print("kyte v2ljas");
   analogWrite(PIN_OUTPUT, LOW);}
  if ( 0 < thermocouple2.readCelsius()  ) {
   if (termopaar2juhtus2) {
     lcd.clear();
     myPID.SetMode(MANUAL);
   myPID.SetMode(AUTOMATIC); }
   termopaar2juhtus2 = false; }
  if ( 0 < thermocouple.readCelsius()) {                             //juhul, kui edastati ekslik termopaari näit, siis taastatakse
   if (termopaar1juhtus2) {//kütmise seadistus
     lcd.clear();
   myPID.SetMode(MANUAL);
   myPID.SetMode(AUTOMATIC); }
   termopaar1juhtus2 = false; }
  if ( thermocouple2.readCelsius() <=   0) {                         //juhul, kui edastati ekslik termopaari näit, siis taastatakse
   if (!termopaar2juhtus2) {                                         //kütmise seadistus
     lcd.clear(); }
   termopaar2juhtus2 = true;
   lcd.setCursor(2, 0);
   lcd.print("T2 error #2");
   lcd.setCursor(3, 1);
   lcd.print("kyte v2ljas");
   analogWrite(PIN_OUTPUT, LOW); }}

void termopaarisnan() {                                              //termopaar väljastab NaN-Not a Number väärtuse, kui termopaar on katki, või...
  if  (isnan(thermocouple.readCelsius())) {                          //lahti ühendatud.
   if (!termopaar1juhtus3) {
     lcd.clear(); }
   termopaar1juhtus3 = true;
   lcd.setCursor(2, 0);
   lcd.print("T1 error #3");
   lcd.setCursor(3, 1);
   lcd.print("kyte v2ljas");
```

```
   analogWrite(PIN_OUTPUT, LOW); }
  if (!isnan(thermocouple.readCelsius())) {                    //kui esineb ekslik näit, siis programmikäik taastatakse.
   if (termopaar1juhtus3) {
   lcd.clear();
   myPID.SetMode(MANUAL);
   myPID.SetMode(AUTOMATIC);
   if (setpoindinijoutud) {
    digitalWrite(marguandetuli, HIGH); }                       //led jääb põlema
   if (!setpoindinijoutud) {
    digitalWrite(marguandetuli, LOW);}}
  termopaar1juhtus3 = false; }
  if (isnan(thermocouple2.readCelsius())) {
   if (!termopaar2juhtus3) {
    lcd.clear(); }
   termopaar2juhtus2 = true;
   lcd.setCursor(2, 0);
   lcd.print("T2 error #3");
   lcd.setCursor(3, 1);
   lcd.print("kyte v2ljas");
   analogWrite(PIN_OUTPUT, LOW);
  }
  if (!isnan(thermocouple2.readCelsius())) {
   if (termopaar2juhtus3) {
    lcd.clear();
   myPID.SetMode(MANUAL);
   myPID.SetMode(AUTOMATIC);
     if (setpoindinijoutud) {
    digitalWrite(marguandetuli, HIGH); }
     if (!setpoindinijoutud) {
    digitalWrite(marguandetuli, LOW); }
  termopaar2juhtus3 = false; } } }

void makstempkaitse() {                                        //piiratakse maksimaalseid temperatuure reaktori sees ja küttekehal
  if (thermocouple.readCelsius() > 235 || thermocouple2.readCelsius() > 205) {
   if (!makstempyletatud) {
    lcd.clear(); }
   makstempyletatud = true;
   lcd.setCursor(0, 0);                                        //max tempi ületamisel kuvatakse temperatuure.
   lcd.print("T1=");                                           //Küttekeha TEMP
   lcd.setCursor(4, 0);
   lcd.print(thermocouple.readCelsius());
   lcd.print("C");
   lcd.setCursor(0, 1);
   lcd.print("T2=");                                           //Rreaktorisisene TEMP
   lcd.setCursor(4, 1);
   lcd.print(thermocouple2.readCelsius());
   lcd.print("C");
   lcd.setCursor(5, 2);
   lcd.print("temp yletas");
   lcd.setCursor(5, 3);
   lcd.print("max piiri");
   analogWrite(PIN_OUTPUT, LOW); }
  if ( 215 > thermocouple.readCelsius()  && 205 > thermocouple2.readCelsius() ) {
   if (makstempyletatud) {                                     //vanade kuvatavate väärtuste taastamine
    lcd.clear();
   if (setpoindinijoutud) {
    digitalWrite(marguandetuli, HIGH); }
   if (!setpoindinijoutud) {
    digitalWrite(marguandetuli, LOW); }}
  makstempyletatud = false; }}

void vahetajanupp(){                                           //lüliti esipaneelil: ülemine, vasakul.  vajutamisel liigutab +  lcd ridade vahel 2,3
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval2) {
   previousMillis = currentMillis;
   buttonState2 = digitalRead(vahetaja);                       //mis loendab 1-st kuni 2-ni, st kolme juures resetitakse antud loendur.
   buttonState4 = digitalRead(fiksaator);                      //salvestatakse näit
   if (buttonState2 != lastButtonState2) {          // paaritu arvuga muudetakse setpoindi väärtust, paarisarvuga muudetakse
inkubatsiooniaega.
     if (buttonState2 == HIGH) {
      counter++; }
     lastButtonState2 = buttonState2; }
   if (counter == 3) {                                         //resetib loenduri
    counter = 1; }

   if (counter == 1 /*paaritu*/) {                             //inkubatsiooniaeg
```

```
      lcd.setCursor(11, 2);
      lcd.print(" ");
      lcd.setCursor(11, 3);
      lcd.print("+");
      val2 = analogRead(A0);
      val22 = map(val2, 50, 990, 1, 60);                    //1...60min
      val22 = constrain(val22, 1, 60);
      if (val22 < 1) {
        val22 = 1; }
      lcd.setCursor(6, 3);
      lcd.print("  ");
      lcd.setCursor(6, 3);
      lcd.print(val22);
      lcd.setCursor(8, 3);
      lcd.print("min");
      if (buttonState4 == HIGH && counter != 2) {
        Inkubatsiooniaeg = val22;
        EEPROM.write(aadress1, val22);                      //salvestatakse eeprommi
        lcd.setCursor(12, 3);
        lcd.print(Inkubatsiooniaeg);                        //prinditakse salvestatud inkubatsiooniaeg
        lcd.print("min"); }}

    if (counter == 2 /*paaris*/) {                          //setpoint
      lcd.setCursor(11, 3);
      lcd.print(" ");
      lcd.setCursor(11, 2);
      lcd.print("+");
      int val = analogRead(A0);
      val1 = map(val, 50, 990, 1, 200);                     //Setpoint temperatuuri vahemik 1...200.
      val1 = constrain(val1, 1, 200);
      if (val1 < 1) {
        val1 = 1;}
      lcd.setCursor(5, 2);
      lcd.print("   ");                                     //et ei tekiks sajast alla minnes viga.
      lcd.setCursor(5, 2);
      lcd.print(val1);                                      //prinditakse setpoindi väärtus 1st-200ni
      lcd.setCursor(8, 2);
      lcd.print("C");
      if (buttonState4 == HIGH && counter != 1) {
        SP2 = val1;                                   //SP2 on reaktorisisene sättetemperatuur, mida kasutatakse PID valemis
väljundväärtusena.
        EEPROM.write(aadress0, SP2);
        lcd.setCursor(12, 2);
        lcd.print(SP2);
        lcd.print("C"); }}}}

void esimeneekraan() {                                      //algusekraan, kus valitakse sp ja inkubatsiooniaeg.
  lcd.setCursor(0, 0);
  lcd.print("Vali setpoint ja");
  lcd.setCursor(0, 1);
  lcd.print("inkubatsiooniaeg");
  lcd.setCursor(0, 2);
  lcd.print("Tset=");                                       //SETPOINT
  lcd.setCursor(0, 3);
  lcd.print("t-ink=");                                      //INKUBATSIOONIAEG
  unsigned long currentMillis7 = millis();                  //termopaaridele  1 sekundiline värskendusinterval
  if (currentMillis7 - previousMillis7 >= interval) {
    previousMillis7 = currentMillis7;
    lcd.setCursor(17, 2);
    lcd.print(thermocouple.readCelsius(),0);                //siin kuvatakse hetketemperatuurid
    lcd.setCursor(17, 3);
    lcd.print(thermocouple2.readCelsius(), 0);
    if (100 > thermocouple2.readCelsius()) {
      lcd.setCursor(19, 3);
      lcd.print(" "); }
    if (100 > thermocouple.readCelsius()) {
      lcd.setCursor(19, 2);
      lcd.print(" "); }}}

void teineekraan() {                                        //kütmise ajal kuvatavad parameetrid
  unsigned long currentMillis2 = millis();                  //kütmise ajal 300ms andmete värskendussagedus
  if (currentMillis2 - previousMillis2 >= interval) {
    previousMillis2 = currentMillis2;
    lcd.setCursor(0, 0);
    lcd.print("T1= ");                                     //ESIMENE TEMP
    lcd.setCursor(4, 0);
```

```
  lcd.print(thermocouple.readCelsius());
  lcd.print(" C");
  lcd.setCursor(0, 1);
  lcd.print("T2= ");                                  //TEINE TEMP
  lcd.setCursor(4, 1);
  lcd.print(thermocouple2.readCelsius());
  lcd.print(" C");
  lcd.setCursor(0, 2);
  lcd.print("Tset= ");                                //SETPOINT
  lcd.setCursor(6, 2);
  lcd.print(SP2);                                     //näitab valitud sättetemperatuuri.
  lcd.print(" C");
  lcd.setCursor(0, 3);
  lcd.print("t-ink= ");
  if(!setpoindinijoutud){
   lcd.setCursor(7, 3);
  lcd.print(Inkubatsiooniaeg);                        //näitab valitud inkubatsiooniaega.
  lcd.print(" min"); }
  if (setpoindinijoutud && !puhastamine) { //Kui inkubatsiooniaega alustatakse
   lcd.setCursor(7, 3);                               //hakatakse minuteid ekraanil loendama.
   int ii = Inkubatsiooniaeg;
   aeg = (((ii * 60 ) - (timeElapsed / 1000))) / 60 ;
   if ( 1 > aeg ) {
    lcd.print(">"); }
   lcd.print(aeg);
   lcd.print(" min"); }
   if(kutminelabi && !puhastamine){
    puhastamine = true;
    lcd.clear();
    lcd.setCursor(7, 3);
    lcd.print(" lopp");}}}                            //kui kütmine on läbi kuvatakse inkubatsiooniaja kohale lopp.

void kolmasekraan() {                          //kütmise ajal kuvatakse ka teisi väärtuseid:
 if (kutmistalustati && !setpoindinijoudmas && !setpoindinijoutud ) {
  if (!lcdpuhastamine4) {
   lcdpuhastamine4 = true;
   lcdpuhastamine5 = false;
   lcd.clear(); }
  lcd.setCursor(3, 2);                             //kütmise alguses kasutusel olevad võimendustegurite väärtused
  lcd.print(algusKp, 1);
  lcd.setCursor(10, 2);
  lcd.print(algusKi, 1);
  lcd.setCursor(17, 2);
  lcd.print(algusKd, 1);
  lcd.setCursor(0, 1);
  lcd.print("P-kutmine");}                          //toimub vaid proportsionaalne kütmine

 if (setpoindinijoudmas && kutmistalustati && !setpoindinijoutud) {
  if (!lcdpuhastamine5) {
   lcdpuhastamine4 = false;
   lcdpuhastamine5 = true;
   lcd.clear(); }
  lcd.setCursor(3, 2);
  lcd.print(enneKp, 1);
  lcd.setCursor(10, 2);
  lcd.print(enneKi, 1);
  lcd.setCursor(17, 2);
  lcd.print(enneKd, 1);
  lcd.setCursor(0, 1);
  lcd.print("PI-kutmine"); }                         //toimub proportsionaalselt integreeriv kütmine

 if (setpoindinijoutud && kutmistalustati && setpoindinijoudmas && !kutminelabi && releestate == HIGH) {
  if (!lcdpuhastamine4) {
   lcdpuhastamine4 = true;
   lcdpuhastamine5 = false;
   lcd.clear(); }
  lcd.setCursor(3, 2);
  lcd.print(parastKp, 1);
  lcd.setCursor(10, 2);
  lcd.print(parastKi, 1);
  lcd.setCursor(17, 2);
  lcd.print(parastKd, 1);
  lcd.setCursor(0, 1);
  lcd.print("Setpoint saavutatud");  }              //alustatakse proportsionaalset diferentseerivat kütmist.

 if (kutminelabi && setpoindinijoutud && kutmistalustati && setpoindinijoudmas && !plahvatus) {
```

```
  if (!lcdpuhastamine5) {
    lcdpuhastamine4 = false;
    lcdpuhastamine5 = true;
    lcd.clear(); }
  lcd.setCursor(0, 1);
  lcd.print("kutmine lopetatud");  }                    //pärast kütmist kuvatakse, et kütmine on läbi

 if (plahvatus && kutminelabi && setpoindinijoutud && kutmistalustati && setpoindinijoudmas) {
  if (!lcdpuhastamine4) {
    lcdpuhastamine4 = true;
    lcdpuhastamine5 = false;
    lcd.clear(); }
  lcd.setCursor(0, 1);
  lcd.print("plahvatuseks valmis");  }                  //kui temp langeb alla 80C

  lcd.setCursor(0, 0);//kuvatakse paikne tekst
  lcd.print("Tseade=");
  lcd.setCursor(0, 2);
  lcd.print("Kp=");
  lcd.print("");
  lcd.setCursor(7, 2);
  lcd.print("Ki=");
  lcd.print("");
  lcd.setCursor(14, 2);
  lcd.print("Kd=");
  lcd.print("");
  unsigned long currentMillis5 = millis();
  if (currentMillis5 - previousMillis5 >= interval) {          //1sek interval
    previousMillis5 = currentMillis5;
    lcd.setCursor(7, 0);
    lcd.print(String(lm35.readCelsius(),1));                   //kuvatakse temperatuur korpuses
    lcd.print(" C");
    lcd.setCursor(0, 3);
    lcd.print("OUTPUT=");
    lcd.setCursor(7, 3);
    lcd.print(Output, 0);                                      //kuvatakse hetkel rakendatav juhttoime väärtus
    if (10 > Output) {
      lcd.setCursor(8, 3);
      lcd.print(" "); }
    if (100 > Output) {
      lcd.setCursor(9, 3);
      lcd.print(" "); }}}

void voimendustegur() {                                        //kuvatakse ekraanile võimendustegurid ning neid saab muuta.
  unsigned long currentMillis8 = millis();
  if (currentMillis8 - previousMillis8 >= interval2) {
    previousMillis8 = currentMillis8;
    int lugema = digitalRead(luliti);
    buttonState2 = digitalRead(vahetaja);
    buttonState4 = digitalRead(fiksaator);
    if (lugema != lastButtonState) {
      lastDebounceTime = millis();}
    if ((millis() - lastDebounceTime) > debounceDelay) {
      if (lugema != buttonState) {
        buttonState = lugema;
        if (buttonState == HIGH) {
          tehtud = !tehtud; }}}
    lastButtonState = lugema;
    if (buttonState2 != lastButtonState2) {
      if (buttonState2 == HIGH) {//counter
        counter2++;}
      lastButtonState2 = buttonState2;}
    if (counter2 == 8) {  //counter 1...7
      counter2 = 1; }
    voimendustegr = analogRead(A0);
    voim = map(voimendustegr, 50, 990, 0, 1000);
    int protsent = voim / 10;                                  //võimendustegurite väärtused 0...100, pärast jagatakse veel 10-ga.
    if (protsent < 0) {
      protsent = 0;}
    if (protsent < 10) {
      lcd.setCursor(7, 0);                                     //kustutatakse üleliigsed nullid ära
      lcd.print(" ");
      lcd.setCursor(7, 1);
      lcd.print(" ");
      lcd.setCursor(18, 0);
      lcd.print(" ");
```

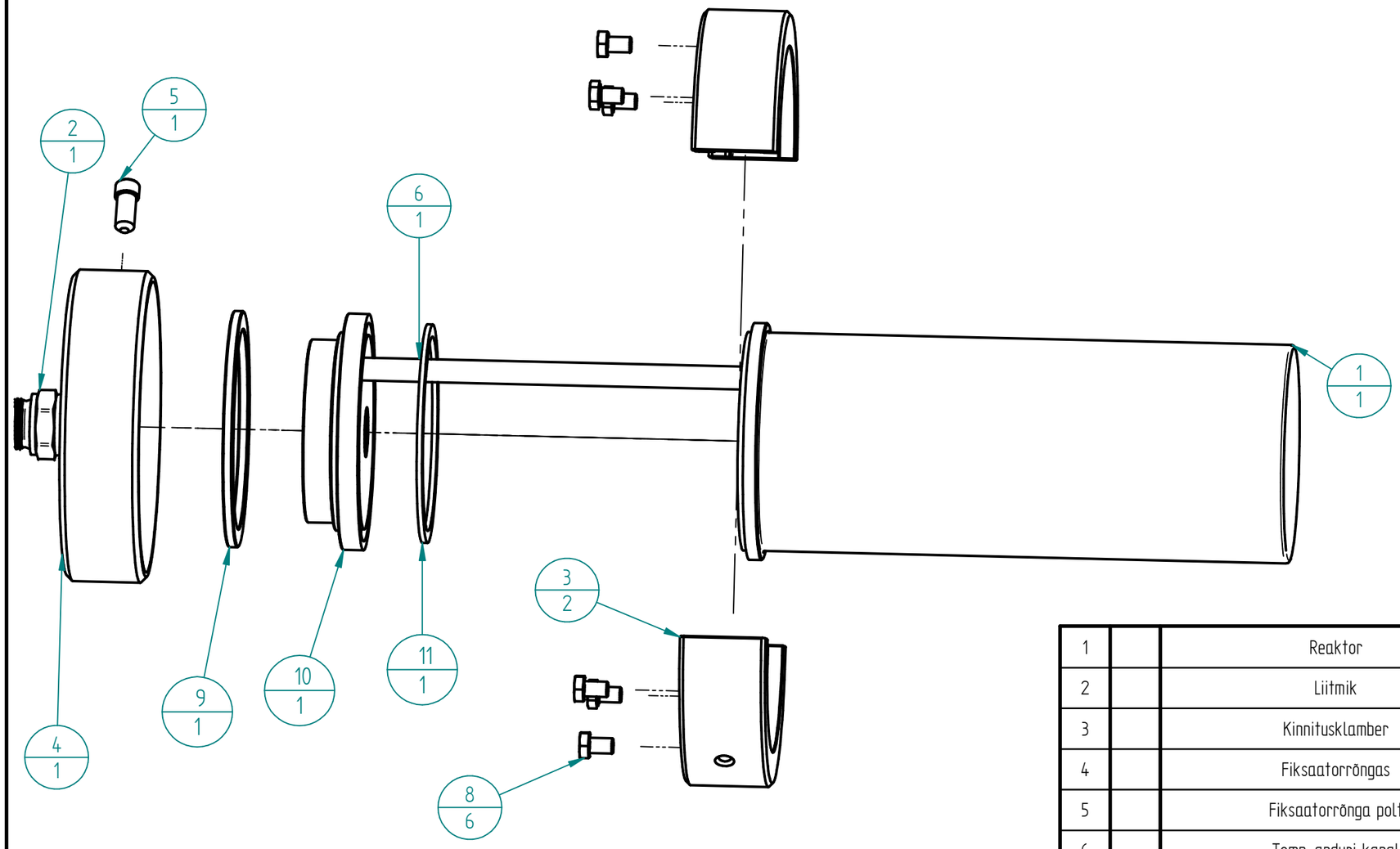
```
  lcd.setCursor(18, 1);
  lcd.print(" ");}

if (counter2 == 1) {
  lcd.setCursor(7, 3);
  lcd.print(" ");
  lcd.setCursor(9, 0);
  lcd.print(" ");
  lcd.setCursor(9, 0);
  lcd.print("+");
  if (tehtud == HIGH) {
    lcd.setCursor(4, 0);
    lcd.print(protsent / 10.0, 1);
    EEPROM.update(aadress2, protsent);                    //salvestatakse proportsionaalne Võimendustegur
    EEPROM.update(aadress3, protsent);                    //update uuendab vt-t, kui väärtus muutub
    algusKp = (EEPROM.read(aadress2)) / 10.0;
    enneKp = (EEPROM.read(aadress3)) / 10.0;  }
    }
if (counter2 == 2) {
  lcd.setCursor(9, 0);
  lcd.print(" ");
  lcd.setCursor(19, 0);
  lcd.print(" ");
  lcd.setCursor(19, 0);
  lcd.print("+");
  if (tehtud == HIGH) {
    lcd.setCursor(15, 0);
    lcd.print(protsent / 10.0, 1);
    EEPROM.update(aadress4, protsent);
    enneKi = (EEPROM.read(aadress4)) / 10.0;  }
    }
if (counter2 == 3) {
  lcd.setCursor(19, 0);
  lcd.print(" ");
  lcd.setCursor(9, 1);
  lcd.print(" ");
  lcd.setCursor(9, 1);
  lcd.print("+");
  if (tehtud == HIGH) {
    lcd.setCursor(4, 1);
    lcd.print(protsent / 10.0, 1);
    EEPROM.update(aadress5, protsent);
    parastKp = (EEPROM.read(aadress5)) / 10.0; }
}
if (counter2 == 4) {
  lcd.setCursor(9, 1);
  lcd.print(" ");
  lcd.setCursor(19, 1);
  lcd.print(" ");
  lcd.setCursor(19, 1);
  lcd.print("+");
  if (tehtud == HIGH) {
    lcd.setCursor(15, 1);
    lcd.print(protsent / 10.0, 1);
    EEPROM.update(aadress6, protsent);
    parastKd = (EEPROM.read(aadress6)) / 10.0; }
}
if (counter2 == 5) { //välise ülekütmise väärtus 1...40
  temp = map(voimendustegr, 50, 990, 0, 40);
  if (temp < 1) {
    temp = 1;}
  lcd.setCursor(19, 1);
  lcd.print(" ");
  lcd.setCursor(13, 2);
  lcd.print(" ");
  lcd.setCursor(13, 2);
  lcd.print("+");
  if (tehtud == HIGH) {
    lcd.setCursor(8, 2);
    lcd.print("   ");
    lcd.setCursor(8, 2);
    lcd.print(temp, 1);
    EEPROM.update(aadress7, temp); }
}
if (counter2 == 6) {//juhttoime piirangu väärtus PWM 0...255.
  voim = map(voimendustegr, 40, 990, 0, 255);
```

```
    if (voim < 1) {
      voim = 1; }
    lcd.setCursor(13, 2);
    lcd.print(" ");
    lcd.setCursor(7, 3);
    lcd.print(" ");
    lcd.setCursor(7, 3);
    lcd.print("+");
    if (tehtud == HIGH) {
      lcd.setCursor(3, 3);
      lcd.print("   ");
      lcd.setCursor(3, 3);
      lcd.print(voim);
      max1 = voim;
      EEPROM.update(aadress8, max1); }}
  }
  lcd.setCursor(0, 0);//paikne tekst
  lcd.print("1kp=");
  lcd.setCursor(11, 0);
  lcd.print("1ki=");
  lcd.setCursor(0, 1);
  lcd.print("2kp=");
  lcd.setCursor(11, 1);
  lcd.print("2kd=");
  lcd.setCursor(0, 2);
  lcd.print("yletemp=");
  lcd.setCursor(0, 3);
  lcd.print("Op=");

  if (!naitakorra) {                                          //kui ekraanivahetuslüliti muutub, kuvatakse väärtused uuesti
    naitakorra = true;
    lcd.setCursor(4, 0);
    lcd.print(algusKp, 1);
    lcd.setCursor(15, 0);
    lcd.print(enneKi, 1);
    lcd.setCursor(4, 1);
    lcd.print(parastKp, 1);
    lcd.setCursor(15, 1);
    lcd.print(parastKd, 1);
    lcd.setCursor(8, 2);
    lcd.print(temp);
    lcd.setCursor(3, 3);
    lcd.print(max1);
}}

void loop() {                                                 //põhiprogramm
  int valik = digitalRead(valikululiti);
  if (SP2 != 0 && Inkubatsiooniaeg != 0 && releestate == LOW && valik == LOW) {
    int reading = digitalRead(luliti);                        //lüliti, mis käivitab kütmise, kontaktidega
relee.
    if (reading != lastButtonState) {                         //nupu vajutamisel jääb see sisse või välja lülitatud
olek
      lastDebounceTime = millis(); }
    if ((millis() - lastDebounceTime) > debounceDelay) {
      if (reading != buttonState) {
        buttonState = reading;
        if (buttonState == HIGH) {
          releestate = !releestate; } } }
    digitalWrite(relee, releestate);
    lastButtonState = reading; }
  if (releestate == HIGH) {
    releelaks = true; }
  if (!releelaks && valik == LOW && !termopaar1juhtus2 && !termopaar2juhtus2 && !termopaar1juhtus3 && !termopaar2juhtus3) {
    if (!lcdpuhastamine) {
      lcdpuhastamine = true;
      lcdpuhastamine2 = false;
      lcdpuhastamine3 = false;
      naitakorra = false;
      lcd.clear();
      lcd.setCursor(12, 3);
      lcd.print(Inkubatsiooniaeg);                            //fikseeritakse inkubatsiooniaeg
      lcd.setCursor(12, 2);
      lcd.print(SP2);  }                                      //fikseeritakse setpoint, mis läheb pid valemisse
    esimeneekraan();
    vahetajanupp();}
```

```
 if (releelaks && valik == LOW && !termopaar2juhtus && !makstempyletatud && !termopaar1juhtus && !termopaar1juhtus2 &&
!termopaar2juhtus2 && !termopaar1juhtus3 && !termopaar2juhtus3) {
   if (!lcdpuhastamine2) {
     lcdpuhastamine = false;
     lcdpuhastamine2 = true;
     lcdpuhastamine3 = false;
     lcd.clear(); }
   teineekraan(); }

 if (valik == HIGH && !termopaar2juhtus && !makstempyletatud && !termopaar1juhtus && !termopaar1juhtus2 &&
!termopaar2juhtus2 && !termopaar1juhtus3 && !termopaar2juhtus3) {
   if (!lcdpuhastamine3) {
     lcdpuhastamine = false;
     lcdpuhastamine2 = false;
     lcdpuhastamine3 = true;
     lcd.clear();  }
   if (releelaks) {
     kolmasekraan();  }
   if (!releelaks) {
     voimendustegur(); }
 }
 if (releestate == HIGH) {                                          //regulaator lülitatakse sisse
   digitalWrite (led, HIGH);                                        //roheline led läheb põlema
 }
 if (releelaks) {
   kutmistalustati = true;
 }
 unsigned long currentMillis3 = millis();                           //termopaaridele ja PID-le 1sekundiline interval
 if (currentMillis3 - previousMillis3 >= interval) {
   previousMillis3 = currentMillis3;
   if (kutmistalustati && !setpoindinijoudmas) {                    //küttekeha kütab välise termopaari järgi
     Setpoint = SP2 + temp;                                  //1…40 kraadi kõrgem, kui soovitud reaktori sisene temp
     myPID.SetOutputLimits(min1, max1);                             //max output – 1…255 pwm regulaatori juhttoime
     Input = thermocouple.readCelsius();                            //igal koodi läbimisel uuendatakse regul sisendi väärtust
      if( isnan(thermocouple.readCelsius())){
 Input = 0; }
     myPID.SetTunings(algusKp, algusKi, algusKd);                   //Proportsionaalne reguleerimine
     PIDkalk();                                                     //PID kalkulatsioon ja reguleerimine
   }
   if (11 > SP2 - thermocouple.readCelsius() && kutmistalustati) {   //10 kraadi enne küttekeha sättetempi
     setpoindinijoudmas = true;
   }
   if (kutmistalustati && setpoindinijoudmas && !setpoindinijoutud) {   //integreetori sisselülitamine
     Setpoint = SP2 + temp;
     Input = thermocouple.readCelsius();
          if( isnan(thermocouple.readCelsius())){
 Input = 0;
 }
     myPID.SetTunings(enneKp, enneKi, enneKd);
     PIDkalk();
   }
   if (kutmistalustati && !kutminelabi && !setpoindinijoutud &&  thermocouple2.readCelsius() >= SP2 && Setpoint != 0) {
     timeElapsed = 0;                                   //inkubatsiooniaja alustamine
     digitalWrite(marguandetuli, HIGH);                 //inkubatsiooniaja indikaatortuli
     setpoindinijoutud = true;
     Output = 0;                                        //väljundi resettimine
     tone(sumisti, sound, 200);                         //lühike helisignaal
   }
   if (setpoindinijoudmas && setpoindinijoutud && !kutminelabi) {
     Setpoint = SP2;                            //Lülitatakse pid ümber PD režiimi, kus hakatakse inkubatsiooniaja vältel setpointi
hoidma.
     Input = thermocouple2.readCelsius();       //regulaatorile söödetakse ette reaktorisisene temperatuur.
          if( isnan(thermocouple.readCelsius())){
 Input = 0;
 }
     myPID.SetTunings(parastKp, parastKi, parastKd);//KI = 0
     PIDkalk();
   }
   if (timeElapsed / 1000 >  (Inkubatsiooniaeg * 60 )  && setpoindinijoutud && !kutminelabi) { //kontroll, kas inkubatsiooniaeg on läbi
     tone(sumisti, sound, 1000);                //1s helisignaal
     kutminelabi = true;
   }
   if (kutminelabi) {
     releestate = LOW;
   }
   if (kutminelabi && !plahvatus && 81 > thermocouple2.readCelsius()) {     //vaatab, kuna temp reaktori sees langeb alla 80 kraadi.
```

```
    plahvatus = true;
    tone(sumisti, sound, 3000);                                   //3sek inkubatsiooniaeg
   }
   if (releestate == LOW ) {
    digitalWrite (PIN_OUTPUT, LOW);
    digitalWrite (led, LOW);
    digitalWrite (marguandetuli, LOW);
   }
   if (plahvatus) {
    labi = true;
   }
   if (labi && 50 > thermocouple2.readCelsius()) {               //kõikide booleanite resettimine
    kutmistalustati = false;
    setpoindinijoudmas = false;
    setpoindinijoutud = false;
    kutminelabi = false;
    plahvatus = false;
    releelaks = false;
    labi = false;
    puhastamine = false;
   }
  }
  if (makstempyletatud && releelaks || termopaar1juhtus && releelaks || termopaar2juhtus && releelaks || termopaar1juhtus2 ||
termopaar2juhtus2 || termopaar1juhtus3 || termopaar2juhtus3) {
   digitalWrite(marguandetuli, millis() >> 10 & 1);              // ca. 2 Hz vilkumine, vea tekkimisel, vilgutatakse punast ledi
  }
  unsigned long currentMillis4 = millis();
  if (currentMillis4 - previousMillis4 >= interval3) {          //ohutuslauseed
   previousMillis4 = currentMillis4;
   if (!releelaks) {
    termopaarnull();
    termopaarisnan(); }
   if (releelaks) {
    termopaari1kaitse();//ohutuslause
    termopaari2kaitse();//ohutuslause
    makstempkaitse();//ohutuslause
    termopaarnull();
    termopaarisnan();
   }
   oldInput = Input;
  }}
```

Lisa 7. Kuumutussüsteemi koostejoonis



Osa	Väli	Nimetus, materjal	Tähis	Hulk	Märkus
1		Reaktor		1	
2		Liitmik		1	
3		Kinnitusklamber		2	
4		Fiksaatorrõngas		1	
5		Fiksaatorrõnga polt		1	
6		Temp. anduri kanal		1	
7		Kütteelement		1	
8		M8 klambri kinnituspolt		6	
9		Kaane võru		1	
10		Reaktori kaan		1	
11		PTFE tihend		1	

		Mass	Mõõt 1:2
Teostas	Mihkel Keisk	Nimetus: Lisa 7. Kuumutussüsteem	
Kontrollis	Vahur Rooni, Erkki Jõgi		
Kinnitas	Vahur Rooni, Erkki Jõgi		
EMU		Leht: 1/1	Tähis: TA 18/110585 7 02 01 K

SOLID EDGE ACADEMIC COPY

Lisa 8. Lihtlitsents lõputöö salvestamiseks ja üldsusele kättesaadavaks tegemiseks ning juhendajate kinnitus lõputöö kaitsmisele lubamise kohta

Mina, Mihkel Keisk,

sünniaeg: 31.12.1991,

1. annan Eesti Maaülikoolile tasuta loa (lihtlitsentsi) enda koostatud lõputöö:

   "Temperatuuriregulaator lignotselluloosse biomassi eeltöötlussüsteemile,"

   mille juhendajad on nooremteadur Erkki Jõgi, nooremteadur Vahur Rooni,

1.1. salvestamiseks säilitamise eesmärgil,
1.2. digiarhiivi DSpace lisamiseks ja
1.3. veebikeskkonnas üldsusele kättesaadavaks tegemiseks kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile;
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Lõputöö autor _____

(*allkiri*)

Tartu, _____

(*kuupäev*)

Juhendajate kinnitus lõputöö kaitsmisele lubamise kohta

Luban lõputöö kaitsmisele.

_____ _____

(*juhendaja nimi ja allkiri*) (*kuupäev*)

_____ _____

(*juhendaja nimi ja allkiri*) (*kuupäev)*