



Назив студијског програма: Машинско инжењерство

Ниво студија: Основне академске студије

Модул: Информатика у инжењерству

Предмет: Програмски језици

Број индекса: 73/2012

Александар В. Синђелић

Развој програмабилних уређаја на Ардуино платформи

- Мултифункционална аутоматска минипекара - завршни рад

Комисија за преглед и одбрану:

1. Проф. др. Ненад Грујовић - ментор

2. _____

3. _____

4. _____

Датум одбране: _____

Оцена: _____

У оквиру овог завршног рада кандидат треба да презентује процес реализације и пројектовања аутоматског система управљања, изврши унапређење постојећег уређаја у смислу проширења његове употребе, одради комплетан програмски код за непрекорно функционисање уређаја и његово примењивање у процесу контроле уз коришћење Ардуино платформе.

Препоручена литература:

- [1] Хенсен А.: Програмирање на језику С, Микрокњига, Београд, 1991.
- [2] Милићев Д.: Објектно оријентисано програмирање на језику С++, Микрокњига, Београд, 1991.
- [3] Massimo Banzi „Getting Started with Arduino, 2nd Edition“, 2011.
- [4] Gordon McComb, Earl Boysen, Електроника за неупућене, Микрокњига, 2007.
- [5] Earl Boysen, Nancy Muir, Електронски пројекти за неупућене, Микрокњига, 2010.
- [6] Јелена Бубало, Радослав Илић, Видојко Вељковић, Оптоелектроника, 1999.

Крагујевац, 2015.

ментор:

др. Ненад Грујовић, ред. проф.

Резиме рада:

Завршни рад „Развој програмабилних уређаја на Ардуино платформи – Мултифункционална аутоматска минипекара“ претставља модификацију и унапређење већ постојећег кућног уређаја за прављење хране. Приказани су процеси дизајнирања и израде штампаних плоча као и начин примене модерних електронских компоненти. Извршена је софтверска симулација дизајнираних електричних кола, ради утврђивања електричних параметара који могу да утичу на исправност уређаја. Имплементирана је Ардуино платформа која се програмира у програмском језику C, чији је задатак да обавља потпуно аутоматизовану контролу уређаја.

Кључне речи:

Ардуино, штампана плоча, објектно оријентисано програмирање, прављење хране.

Summary of work:

Final thesis “Programmable device development based on Arduino platform – Multifunctional automated bread maker” represents modification and upgrade of already existing house device for food making. Design and production processes are shown as well as application of modern electric components. Software simulation for designed electric circuits has been done, in order to determine electric parameters which can affect device operation. An Arduino platform, which is programmed in C programming language, has been implemented to provide fully automated control of the device.

Key words:

Arduino, printed circuit board, object-oriented programming, food making.

Садржај

1. УВОД	1
1.1 Опис задатка	1
1.2 Опис Ардуино платформе	1
1.2.1 Опште	1
1.2.2 Хардвер	2
1.2.3 Софтвер	2
1.2.4 Коришћена Ардуино платформа	3
1.3 Опис мини пекаре	4
2. МОДИФИКАЦИЈЕ	5
2.1 Анализа постојеће (оригиналне) електронике	5
2.2 Повезивање Ардуино платформе	7
2.3 Повезивање и калибрисање температурних сензора	8
2.4 Израда штампаних плоча (PCB)	11
2.4.1 Опште	11
2.4.2 „Фото поступак“ метода	12
2.4.3 Плочица напајања и прекидача	20
2.4.4 Међуплочица	28
2.4.5 Плочица са тастерима	32
2.5 LCD (Liquid Crystal) дисплеј	34
2.6 Шема повезивања свих компоненти	35
2.7 Рам	36
2.8 Пертинакс плоча	40
2.9 Склоп	41
2.10 Модификовање предњег панела	50
3. СОФТВЕР	58
3.1 Библиотеке, дефинисање пинова, тастера и константних вредности	58
3.2 Структура „PROGRAM“	59
3.3 Помоћне променљиве	61
3.4 Метода „периодично паљење“	62
3.5 Метода за мерење температуре	63
3.6 Методе за контролу компоненти	65
3.6.1 Метода „грејање“	65
3.6.2 Метода „мешање“	67
3.6.3 Метода „свеИскључено“	67
3.7 Метода „одбројавање“	68
3.8 Метода „штампање“	69
3.9 Специјални симболи	70
3.10 Метода „ресет“	72

3.11 Метода „звучно упозорење“	72
3.12 Функција „setup“	73
3.13 Функција „loop“	75
4. УКУПНИ ТРОШКОВИ	79
5. РЕЦЕПТИ	80
5.1 Бели хлеб (900г)	80
5.2 Шпанска пита (480г)	81
5.3 Цем од шљиве	81
5.4 Качамак	82
5.5 Пудинг	83
6. ЗАКЉУЧАК	84
7. ЛИТЕРАТУРА	85

1. УВОД

1.1. Опис задатка

Задатак овог завршног рада као што и сам назив теме говори, јесте да се искористи Ардуино платформа за контролу уређаја, односно у овом случају мини пекаре. Наиме сваки уређај контролише, жаргонски речено „мозак“, тј. скуп електронских компоненти и чипова који врше контролу електричних, као и електро-механичких делова уређаја. Поред контроле они врше још и мерење променљивих параметара, у овом случају тај параметар јесте температура, и на основу промене тог параметра врше гашење и укључивање грејача када је то потребно.

Тај скуп електронских компоненти и чипова, односно „мозак“ потребно је заменити другим „мозгом“, тј. Ардуино платформом. Након замене потребно је програмирати ту платформу да може да обавља исте функције као и претходно извађена електроника, без икаквих недостатака, али и извршити нека додатна унапређења. Заправо, претходно извађена електроника била је програмирана да једино врши припрему хлеба што и јесте главна функција мини пекаре, без могућности да корисник креира сопствене програме. Главни циљ јесте да се уз помоћ Ардуино платформе омогући припрема хлеба, као и неких других јела и да се омогући да корисник уколико жели да уз помоћ сопственог рачунара, преко USB прикључка, уз неколико једноставних корака убаци сопствене програме за припрему јела.

1.2. Опис Ардуино платформе

1.2.1. Опште

Ардуино је физичко-рачунарска платформа (развојни систем) отвореног кода. Хардвер се састоји од једноставног отвореног хардверског дизајна Ардуино плоче са Atmel AVR процесором и пратећим улазно-излазним елементима. Софтвер се састоји од развојног окружења које чине стандардни компајлер и боотлоадер који се налази на самој плочи. Програми за Ардуино платформу пишу се у C/C++ програмском језику са додатном софтверском библиотеком званом „Wiring“, која многе улазно/излазне операције чини једноставнијим. [7]



Слика 1. Ардуино лого [7]

1.2.2. Хардвер

Ардуино плочу чине 8-,16- или 32-битни Atmel AVR микроконтролер са припадајућим компонентама које омогућавају програмирање и повезивање са другом електроником. Битан аспект Ардуино пројекта је стандардизован распоред конектора који омогућава лако повезивање са додатним модулима, познатијим као штитови. Ове додатне модуле, штитове, поизводе разни произвођачи широм света. Званичне Ардуино плоче углавном користе megaAvr серију чипова, конкретно ATmega8, ATmega168, ATmega328, ATmega1280 и ATmega2560. Већина плоча поседује 5V линеарни напонски регулатор и 16MHz кристални осцилатор (или керамички резонатор у неким верзијама). Ардуино микроконтролери се испоручују са програмираним bootloader-ом који поједностављује поступак пребацивања преведеног кода у флеш меморију на чипу. Други микроконтролери обично захтевају засебан програматор. [7]

1.2.3. Софтвер

Ардуино интегрисано развојно окружење је апликација написана у „Java“ програмском језику. Креирано је тако да уведе у програмирање ученике, студенте и остале почетнике који нису упознати са начином развоја софтвера. Састоји се од уређивача кода са могућностима као што су означавање кода, упаривање заграда, аутоматско увлачење линија. Овај уређивач може да преведе код а затим га и пребаци у чип једном командом. У овом случају није потребно подашавати параметре превођења кода или покретати програме из командне линије. Ардуино интегрисано развојно окружење долази са софтверском библиотеком званом „Wiring” која чини уобичајене улазно-излазне операције веома једноставним. Ардуино програми се пишу у C/C++ програмском језику, мада корисници морају да дефинишу само две функције како би направили извршни програм. [7]

Те функције су:

- `setup()` – функција која се извршава једном на почетку и служи за почетна подешавања;
- `loop()` – функција која се извршава у петљи све време док се плоча не искључи.

Испод је дат пример једноставног Ардуино програма који пали и гаси LED диоду са међусобним чекањем од једне секунде:

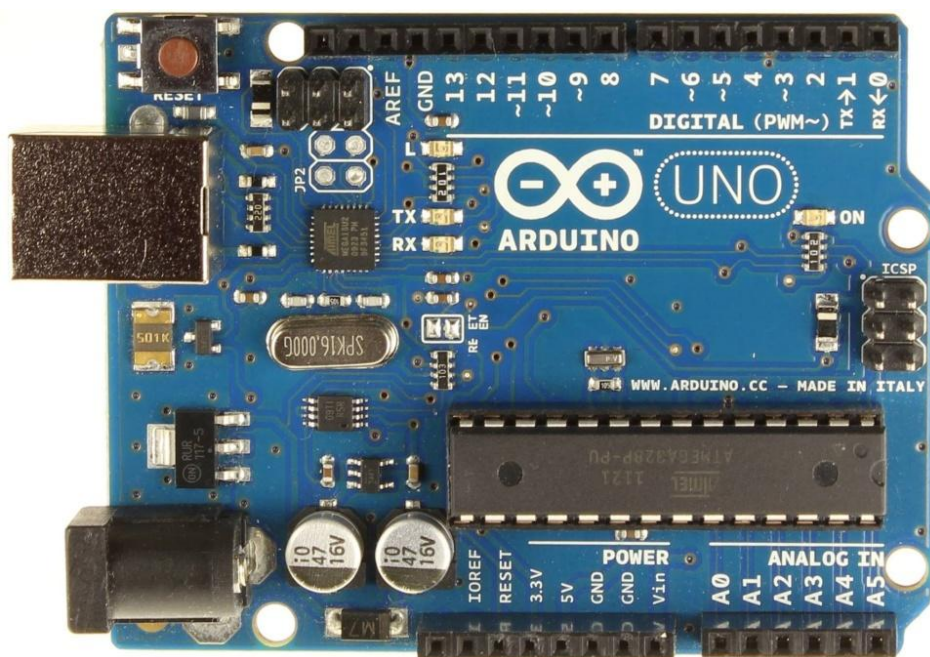
```
#define LED_PIN 13

void setup () {
    pinMode (LED_PIN, OUTPUT);      // дефиниши pin 13 као digitalni izlaz
}

void loop () {
    digitalWrite (LED_PIN, HIGH);   // укључи LED
    delay (1000);                   // сачекај један секунд (1000 milisekundi)
    digitalWrite (LED_PIN, LOW);    // искључи LED
    delay (1000);                   // сачекај један секунд
}
```

1.2.4. Коришћена Ардуино платформа

Постоји више Ардуино платформи које се међусобно највише разликују по томе који управљачки чип користе. За овај пројекат изабрана је „Arduino UNO Rev3“ платформа. Ардуино Уно је микроконтролерска плоча базирана на ATmega328 микроконтролеру. Поседује 14 дигиталних улаза/излаза (од којих 6 могу служити како PWM излази), 6 аналогних улаза, 16MHz кристал осцилатор, USB конектор за програмирање и везу с рачунаром, конектор за екстерно напајање (уколико се не напаја са USB-а), ICSP конектор и тастер ресет. За комуникацију са рачунаром Уно, за разлику од претходних плоча не користи FTDI чип за USB-to-Serial комуникацију. Уместо њега сада се у ове сврхе користи ATmega16U2 (R3) или ATmega8U2 (R2). [8][9]



Слика 2. „Arduino UNO Rev3“ платформа

Техничка спецификација

Микроконтролер:	ATmega328
Радни напон:	5V
Напон напајања (препоручено):	7-12V
Максимални напон напајања (не препоручује се):	20V
Дигитални У/И пинови:	14 (од којих 6 обезбеђује PWM излаз)
Аналогни улазни пинови:	6
Струја по У/И пину (5V):	40mA
Струја по У/И пину (3.3V):	50mA
Флеш меморија:	32KB (Atmega328) од којих је 0.5KB искоришћено за bootloader
SRAM:	2KB (Atmega328)
EEPROM:	1KB (Atmega328)
Радни такт:	16MHz

1.3. Опис мини пекаре

Мини пекара јесте заправо мали кућни апарат за аутоматску припрему хлеба. Њен задатак јесте да без надзора корисника аутоматски изврши припрему теста и да испече хлеб. Једини задатак корисника јесте да убаци састојке који чине хлеб (брашно, квасац, вода, со, итд.) у посуду уређаја у којој се одвија целокупни процес припреме хлеба (мешање, раст теста и печење), а затим одабере програм за који важи одабрана рецептура и покрене уређај. Процес припреме хлеба почиње мешањем убачених састојака у посуду од којих се ствара тесто и који траје неколико минута, са могућим паузама у зависности од програма. Након тога следи ферментација (раст) теста на одређеној температури и одређеном временском интервалу. Промена температуре се констатно прати преко сензора за мерење температуре и на тај начин одржава у границама подешене температуре, односно одступање стварне од подешене температуре се одржава минималним. Затим следи процес печења теста такође на одређеној температури и одређеном временском интервалу.

На слици 3. може се видети мини пекара коришћена у овом раду пре модификације.

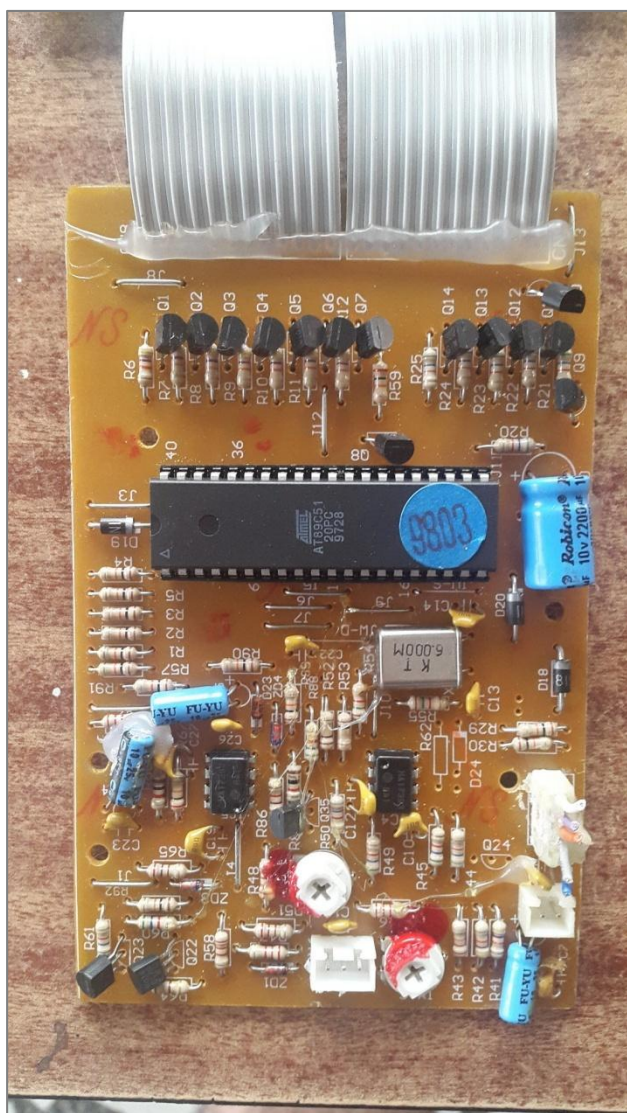


Слика 3. „Heim Haustechnik“ мини пекара

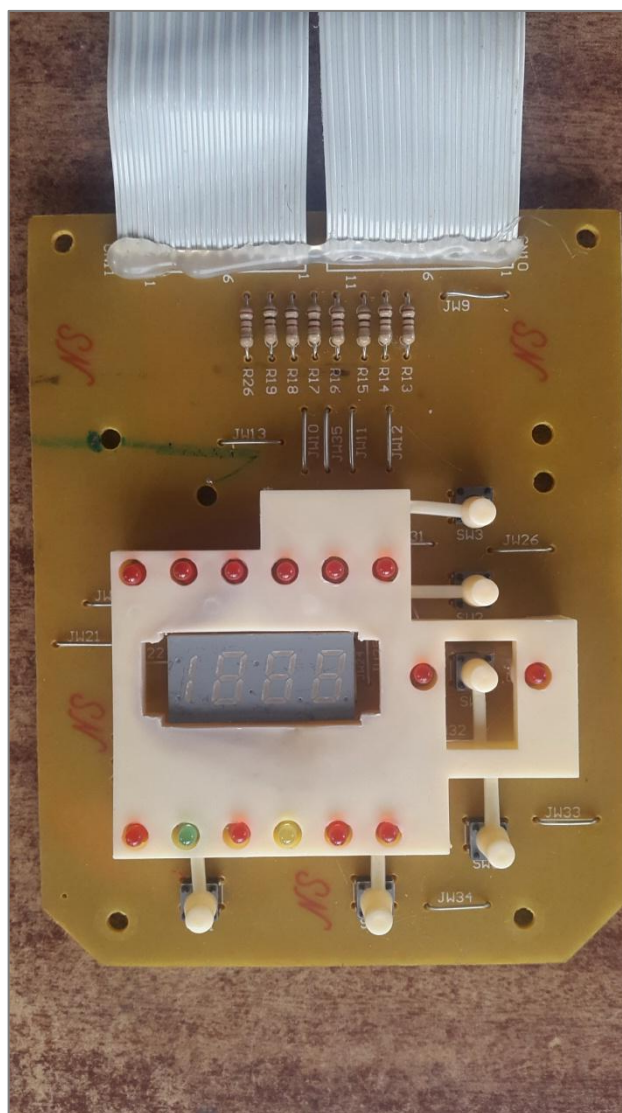
2. МОДИФИКАЦИЈЕ

2.1. Анализа постојеће (оригиналне) електронике

Пре било какве замене компоненти потребно је анализирати постојећу електронику и утврдити да ли је и на који начин могуће извршити замену новим компонентама. Постојећа (оригинална) електроника састоји се из два дела. Први део слика 4. претставља „мозак“ уређаја, односно микроконтролер са свим пратећим компонентама које контролишу рад уређаја (грејач и електромотор) и који је потребно заменити Ардуино платформом. Ту се такође налазе и сви тастери слика 5. помоћу којих се врши одабир програма, подешавање тајмера као и покретање и заустављање рада уређаја. У овом делу налази се још и 4-цифрени дисплеј слика 5. који показује преостало време до завршетка изабраног програма, као и LED диоде које показују који је програм изабран и који се део процеса производње обавља.



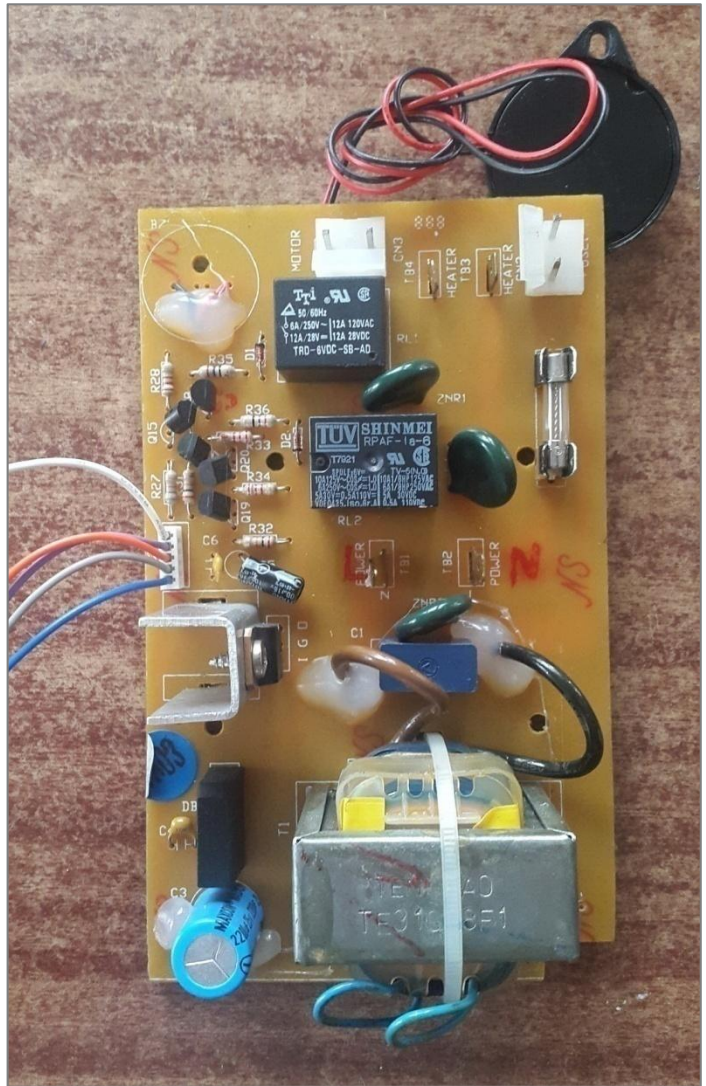
Слика 4. „Мозак“ уређаја



Слика 5. Дисплеј са тастерима и LED диодама

Други део слика 6. претставља прекидачка електроника и регулација напона. Пошто микроконтролери раде на ниском напону (5V) и са малом једносмерном струјом (40-50mA) као што је претходно наведено под тачком 1.2.4. разумљиво је да они не могу директно вршити укључивање и искључивање грејача и електромотора, као и да не могу бити директно повезани на кућну наизменичну струју. Из тог разлога користи се трансформатор који смањује кућни напон на доста нижу вредност, односно у овом случају је то око 9V.

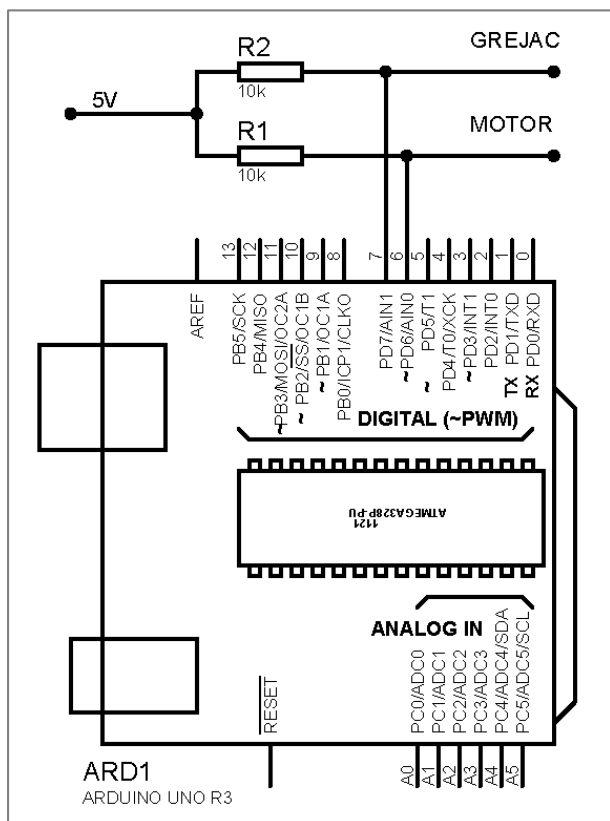
За добијање једносмерне струје користи се „Грецов“ спој диода у комбинацији са одговарајућим кондензаторима који служе за стабилизацију напона. Као што је претходно речено да микроконтролери не могу директно вршити паљење и гашење грејача и електромотора из тог разлога користе се додатне електронске компоненте, у овом случају то су релеји. Релеји коришћени у овом уређају, за укључивање магнета везују се на напон од 5-6V, а по контакту могу да проводе наизменичну струју јачине и до 10A. Упркос томе што се за укључивање магнета у релеју користи иста вредност напона (5V) као на излазним пиновима микроконтролера и даље их је није могуће везати директно, јер јачина електричне струје потребна за укључивање магнета премашује максималну могућу вредност коју пинови микроконтролера могу да издрже. Из тог разлога користи се група транзистора који се могу уочити покрај релеја на слици 6. како не би дошло до квара микроконтролера. Ови транзистори спајају један крај магнета релеја са нулом, док је други крај магнета, преко отпорника спојен на исправљен напон са излаза трансформатора. У овом делу налазе се још и прикључци за грејач и електромотор, али и прикључак за топлотни осигурач који је намонтиран у простору где се пече хлеб. Овде се такође налази и жичани осигурач који издржава максималну јачину струје до 5A. На овом делу налази се и ограничавач напона на 5V, јер се микроконтролер напаја тим напоном. Овде је још смештен и пиезо звучник који емитује одређене звуке када је то потребно. Са леве стране као што се може и приметити на слици 6. налазе се жице које спајају овај са првим делом електронике који је претходно описан и које је било потребно утврдити за шта која служи. На крају је установљено да је повезивање Ардуино платформе са овим делом могуће и безбедно по саму платформу, с тим што је напон потребно довести директно са „Грецовог“ споја, а не са излаза ограничавача напона од 5V (плава жица), јер препоручени напон за напајање Ардуино платформе износи 7-12V. [4][5]



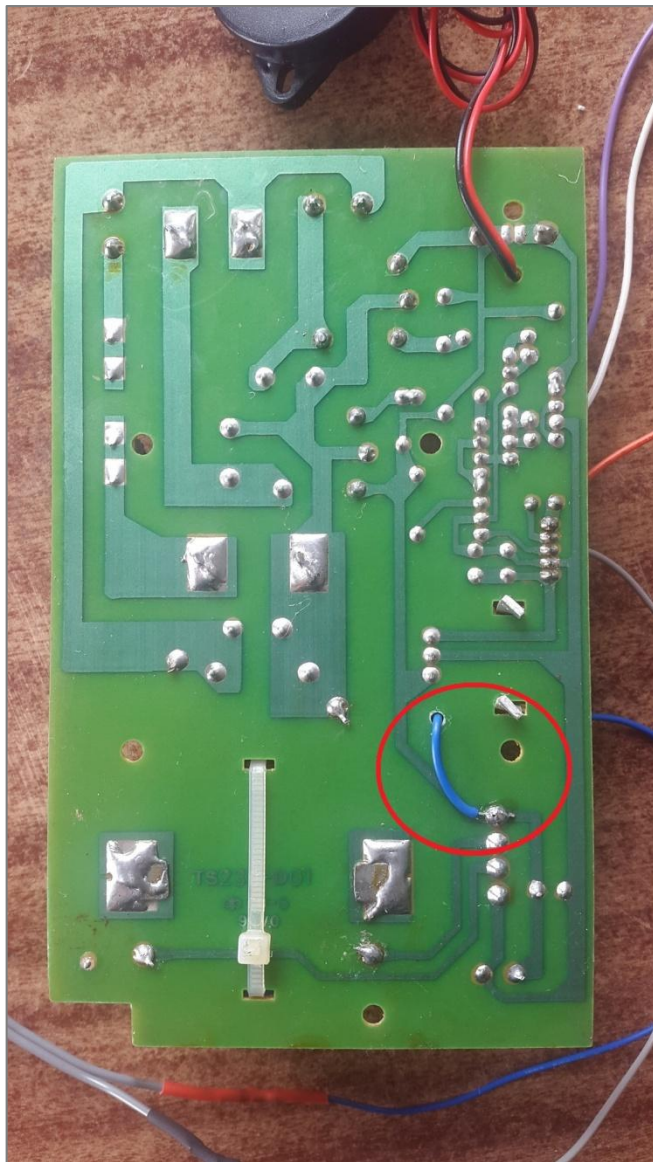
Слика 6. Прекидачка електроника, регулатори напона, пиезо звучник и прикључци

2.2. Повезивање Ардуино платформе

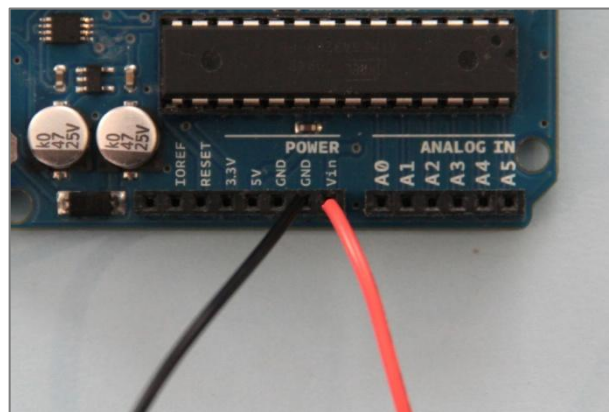
Као што је у претходној тачки речено, Ардуино платформу потребно је повезати директно на „Грецовом“ споју јер је на већ постојећој електроници изведена само жица за напон од 5V. Да би се то одрадило потребно је залемити жицу на (+) крају „Грецовог“ споја као што је то одрађено на слици 7. у црвеном кругу. Други крај те жице прикључује се на „Vin“ пину на Ардуино платформи, док се нула (сива жица на слици 6.) прикључује на једном од „GND“ пинова и на тај начин је доведено напајање овој платформи. Напајање Ардуино платформе преко ових пинова може се видети на слици 9. За почетак повезани су само електромотор и грејач како би се испробао рад са Ардуином и како би се калибрисали сензори са мерење температуре. Начин повезивања ових компоненти може се видети на слици 8. на којој је приказана шема повезивања.



Слика 8. Повезивање електромотора и грејача са Ардуино платформом



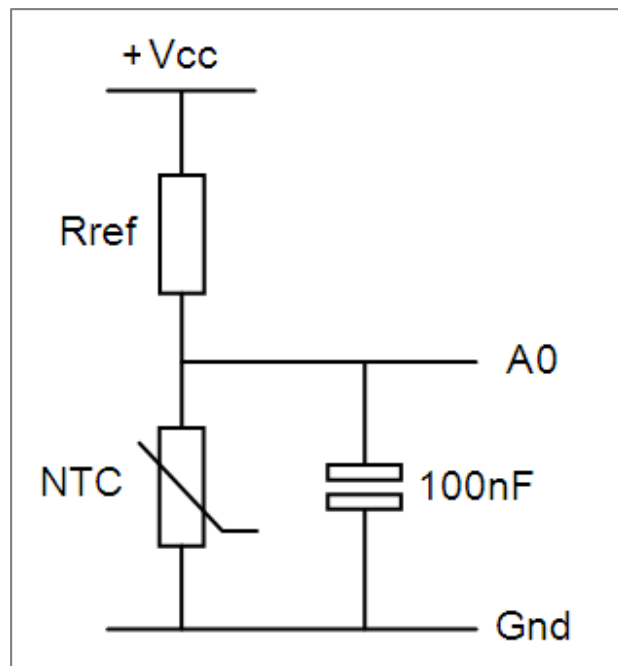
Слика 7. Довођење напона са „Грецовог“ споја



Слика 9. Довођење напајања Ардуину

2.3. Повезивање и калибрисање температурних сензора

Први корак пре повезивања ових сензора јесте био да се открије о ком типу сензора се заправо ради. Мерењем напона и отпорности на крајевима сензора, закључено је да се ради о NTC термисторима. NTC јесте скраћеница од енг. „Negative temperature coefficient“, што значи да се ради о типу сензора чија се отпорност смањује са порастом температуре. Шема повезивања ових сензора са микроконтролером дата је на слици 10. где се за „Rref“ отпорник узима приближна вредност коју термистор има на собној температури. Напон +Vcc јесте напон од 5V. Пошто се са дате шеме очитава напон на термистору „A0“ који се мења са променом отпорности термистора, а пошто је „Rref“ константна вредност, познавањем Омовог закона лако се може извести једначина за израчунавање отпорности на термистору:



Слика 10. Шема повезивања NTC термистора са микроконтролером [4]

$$NTC = R_{ref} \cdot \frac{A0}{V_{cc} - A0}$$

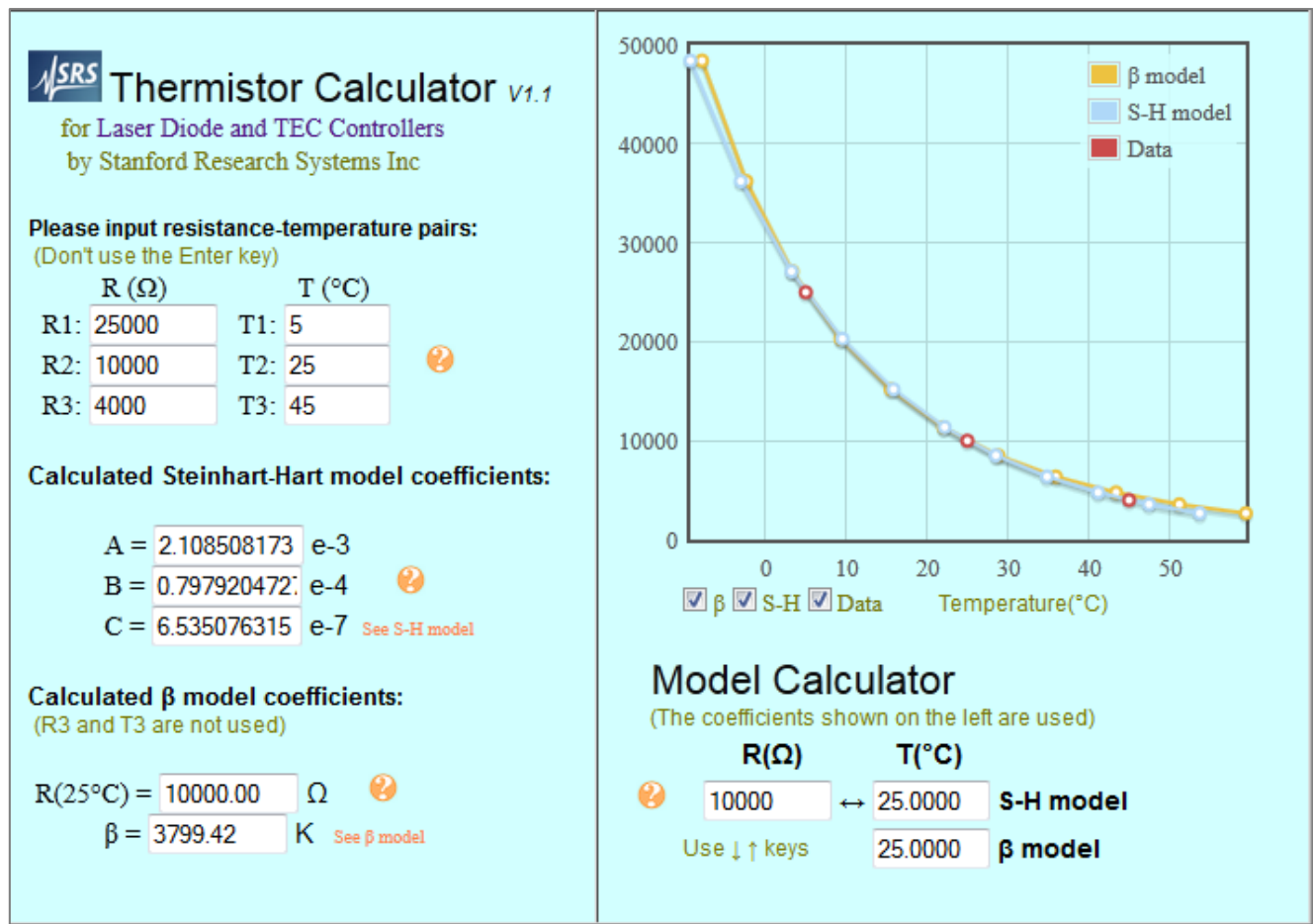
Конверзија отпорности у температуру врши се преко „Штаинхарт-Хартове“ једначине:

$$\frac{1}{T} = A + B \ln(R) + C[\ln(R)]^3,$$

где су:

- T – температура изражена у келвинима;
- R – електрична отпорност сензора у омима (Ω);
- A, B и C – Штаинхарт-Хартови коефицијенти који варирају од сензора до сензора. [11]

Приликом куповине овог типа сензора, произвођач је дужан да достави вредност ових коефицијената. Пошто се ради о половном уређају вредности ових коефицијената потребно је израчунати. Постоје једначине преко којих се ови коефицијенти израчунавају, али да би се процес калибрисања олакшао и са сигурношћу одрадио без грешака искоришћен је интернет калкулатор ових коефицијената слика 11., компаније „Stanford Research Systems Inc“ са следеће веб адресе: <http://www.thinksrs.com/downloads/programs/Therm%20Calc/NTCCalibrator/NTCcalculator.htm>.



Слика 11. Интернет калкулатор коефицијената NTC термистора

За израчунавање ових коефицијената потребно је да се измери електрична отпорност сензора на три различите температуре, а затим да се те вредности унесу у одговарајућа поља као и температурне вредности на којима су отпорности измерене. За најпрецизније израчунавање ових коефицијената препоручује се да се за прву вредност електричне отпорности узме вредност коју сензор има на собној температури (20-25°C). За другу вредност електричне отпорности препоручује се она вредност коју сензор има на средини свог температурног мерног опсега, а за трећу она коју има на крају свог температурног мерног опсега. Загревањем простора у коме се налазе сензори до температуре од 200°C (јер се хлеб пече на тој температури) и мерењем електричне отпорности на сензорима, утврђено је да су ови сензори намењени за мерење температуре до 220-230°C. Из тог разлога, друга вредност електричног отпора са сензора узета је на 100°C, а трећа на 200°C. Мерење температуре за време калибрације ових сензора слика 12. вршено је помоћу сонде слика 13. која служи за мерење температуре на пекарским пећима. Вредности измерених коефицијената као и референтних отпорности (R_{ref}) за оба сензора могу се видети у табели испод:

Коефицијент	T (°C)	Сензор			
		Црвени ($R_{ref} = 180k\Omega$)		Бели ($R_{ref} = 47k\Omega$)	
		Ел.отпор (Ω)	Вредност кое.	Ел.отпор (Ω)	Вредност кое.
A	22	190000	0,0004357140911	52000	0,001097204834
B	100	25000	0,0001733009365	8500	0,00009308562133
C	200	3600	0,0000004710945505	1200	0,0000009997003632



Слика 12. Мерење температуре за време калибрације температурних сензора

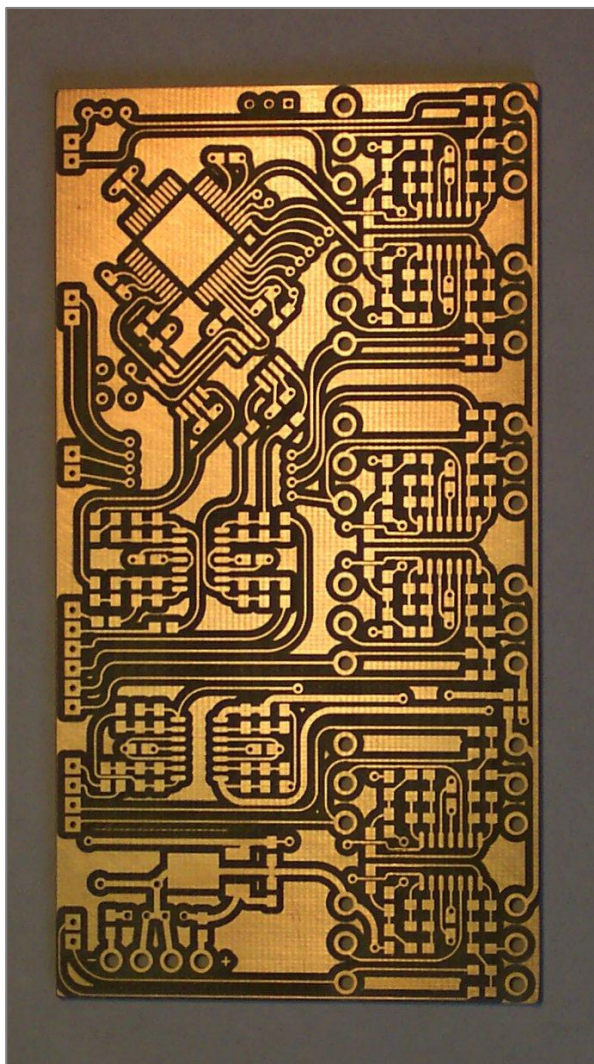


Слика 13. Сonda за мерење температуре на пекарским пећима

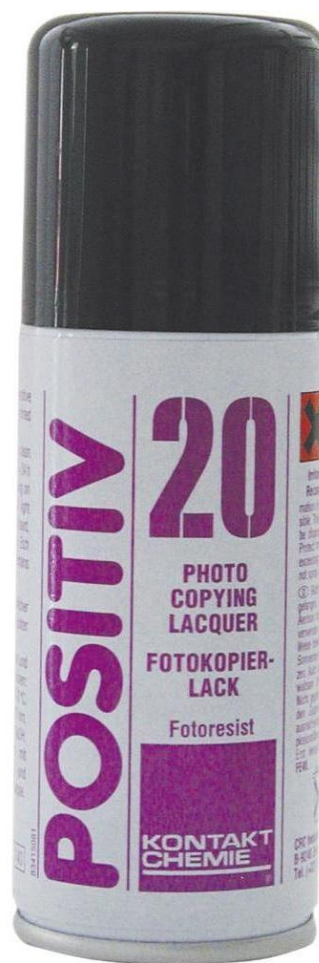
2.4. Израда штампаних плоча (PCB)

2.4.1. Опште

Под изразом „штампане плоче“ (PCB, eng. „Printed Circuit Board“) подразумевају се плоче од перфинакса или витропласта на којима се налази танак слој бакра, који се одређеним методама обликује и тако настају линије којима су спојене електричне компоненте (скуп линија и компоненти чини електрично коло). Пример штампане плоче може се видети на слици 14. Постоје једностране и двостране штампане плоче што претставља на којим се странама налази бакар. У овом раду коришћене су једностране штампане плоче. Као што је претходно напоменуто постоји више метода за обликовање бакра, заправо за цртање линија електричног кола. Најчешће коришћене јесу ручна метода, што значи да се линије цртају слободном руком уз помоћ маркера и „Фото поступак“ метода. За овај рад употребљена је „Фото поступак“ метода која стоји за скупљу методу због цене фото лака слика 15., али резултат јесте израда доста компликованијег електричног кола, као и велика прецизност и квалитет линија. Дизајнирање и цртање електричних шема обављено је у бесплатном софтверу „PCB Artist“ који се може преузети са следеће веб адресе: <http://www.4pcb.com/free-pcb-layout-software/>.



Слика 14. Пример штампане плоче



Слика 15. Фото лак „Позитив 20“ у спреју

2.4.2. „Фото поступак“ метода

Ова метода заправо претставља преношење претходно оштампане слике на бакар, на којем је већ нанет фото лак. За обављање ове методе потребне су следеће ставке:

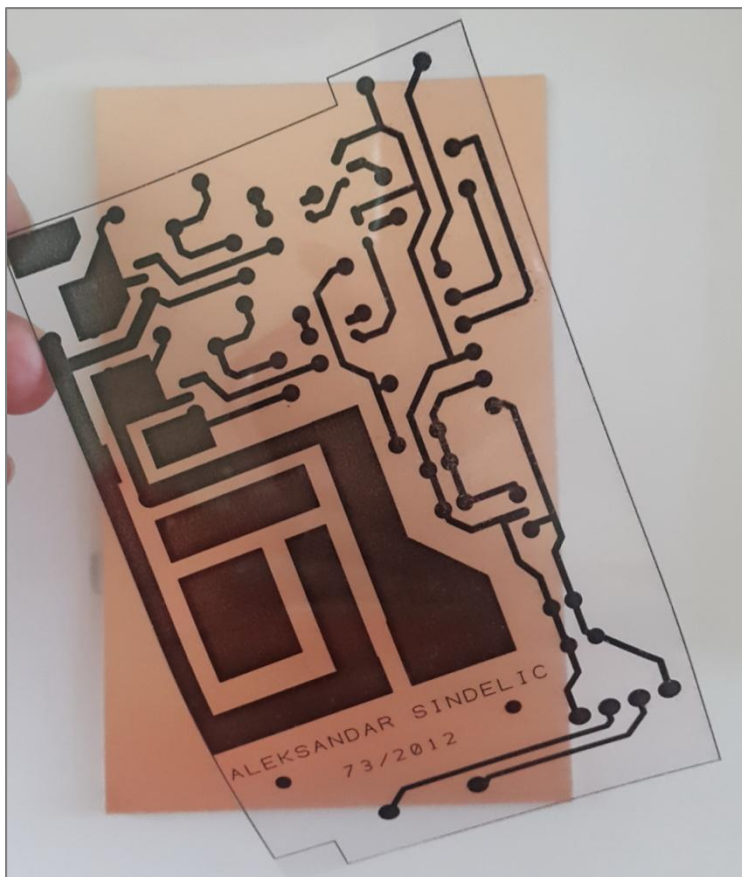
- Хидроген 30%,
- Сона киселина 18%,
- фото лак „Positiv 20“,
- Натријум хидроксид,
- живина сијалица или рефлектор 125-150W,
- разне посуде.

Први корак јесте да се очисти страна плоче на којој се налази бакар, на коју се затим наноси фото позитив лак. Врло је битно да се приликом наношења, као и после наношења фото лака плоча не излаже никаквој светлости осим црвене, што значи да је наношење фото лака најбоље извести у тамној просторији, а затим плочу ставити у некој кутији која не пропушта светлост слика 16. Произвођач каже да је лак потребно сушити око 20 минута на температури од 65-70°C или 24 сата на собној температури. Из сигурносног разлога и првог неуспелог покушаја све плочице израђене у овом раду сушене су 30 минута у рерни кућног шпорета на отприлике 60°C (није могуће установити тачну температуру, јер шпорет не поседује дигитално штеловање, нити мерач температуре), а затим су остављене 24 сата на собној температури да би се комплетно и сигурно осушиле. [12][13]

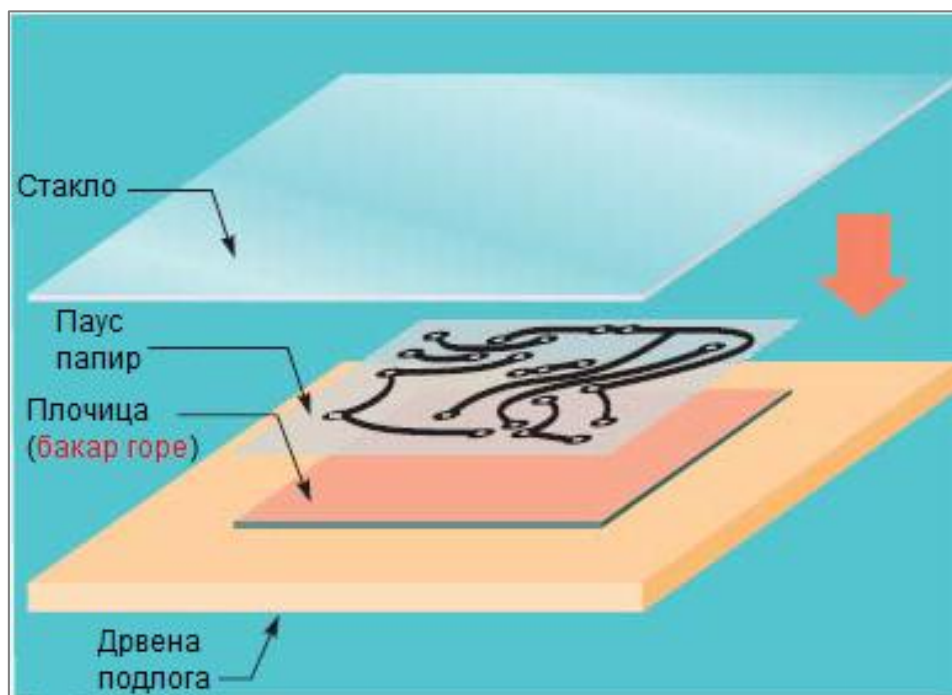


Слика 16. Картонска кутија у којој су смештане плочице

Након сушења лака потребно је отштампати на провидном папиру (паус папир) жељено електрично коло које ће бити израђено. У овом раду уместо паус папира коришћена је провидна термо фолија која се користи за пројекторе. Штампаче електричног кола одрађено је на обичном кућном ласерском штампачу, а пример отштампаног електричног кола на провидној фолији може се видети на слици 17. Након тога прави се тзв. „Сендвич“ слика 18. тако што се прво постави плоча са осушеним фото позитив лаком, затим се изнад ње постави паус папир (или провидна термо фолија) на који је отштампано електрично коло и на крају се постави стакло које је претходно обрисано. Процес прављења „сендвича“ такође је потребно обавити у тамној просторији. [12][13]

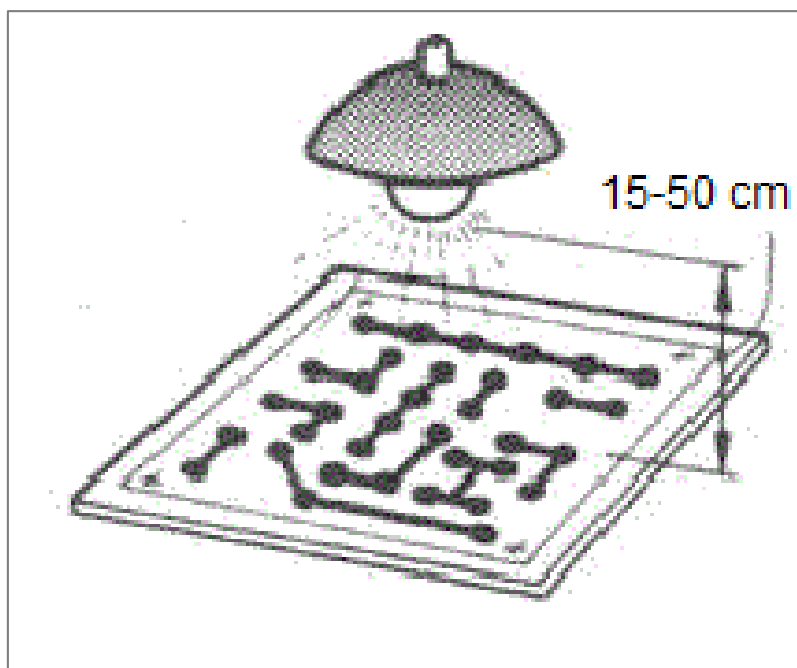


Слика 17. Отштампано електрично коло на провидној термо фолији



Слика 18. Пример „Сендвича“ [14]

Следећи корак јесте паљење светла које се налази изнад постављене плочице слика 19. Као што је већ речено потребно је користити живину сијалицу јачине 125-150W, међутим у овом раду коришћен је рефлектор јачине 1000W. Пошто се у разним упутствима наводи да је плочицу потребно осветлити неколико минута са претходно споменутом сијалицом, то време се доста смањује због доста веће јачине рефлектора од живине сијалице. Време за које је рефлектор био укључен износи 50-55 секунди. Процес осветљавања може се видети на слици 20. Након осветљавања плочица се одмах убацује у посуду (дољни десни угао на слици 20.) у којој се налази развијач (мешавина Натријум хидроксида и воде, а њихов однос дат је од стране произвођача фото лака), где се затим благим мућкањем скидају претходно осветљени делови фото лака. Када се тај процес заврши онда се плочица са исцртаним линијама слика 21. убацује у посуду у којој се налази раствор хидрогена и соне киселине слика 22. Однос хидрогена и киселине утврђује се експериментално. **Напомена:** пре убацивања плочице у раствор киселине и хидрогена проверити да ли су све линије добро исцртане, уколико нису подебљати их маркером. Док се плочица налази у раствору хидрогена и киселине потребно је на сваких 2-3 минута лагано промешати раствор и уколико је потребно досипати хидроген. Након уклањања вишка бакра плочицу је потребно испрати водом. Након испирања исцртане линије уклонити уз помоћ крпе и ацетона. [12][13]



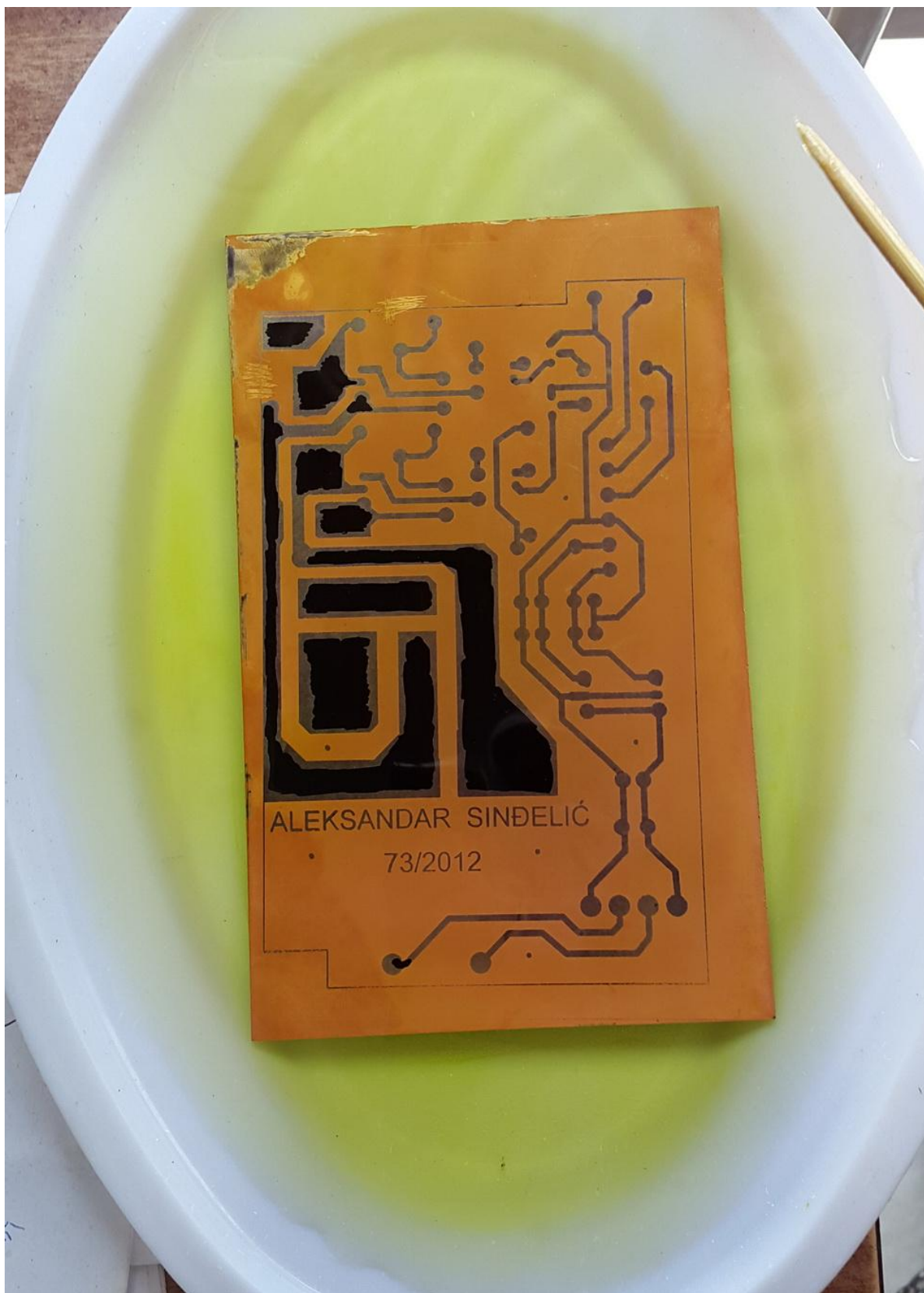
Слика 19. Положај сијалице [13]



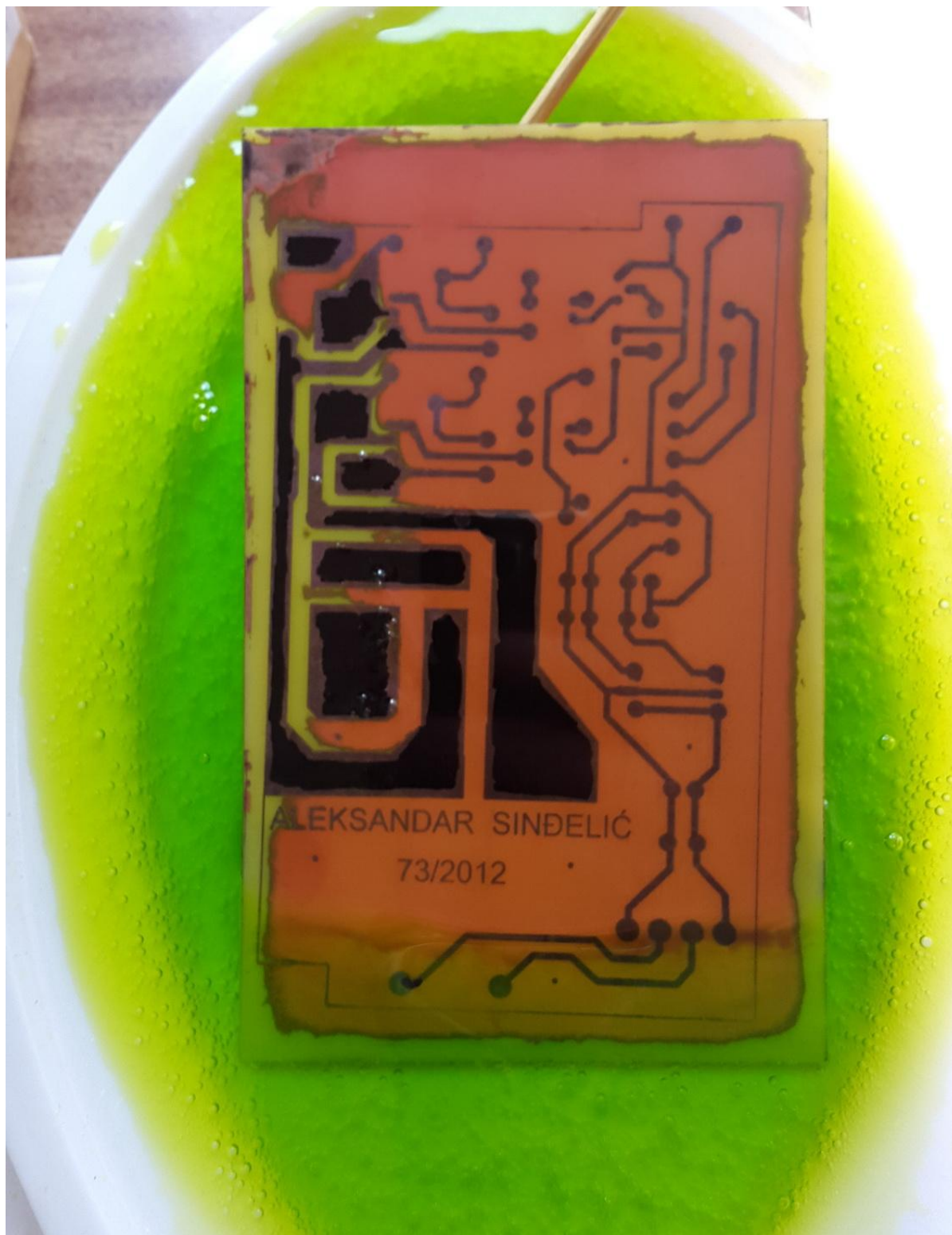
Слика 20. Осветљавање плочице рефлектором



Слика 21. *Плочица са исцртаним линијама*



Слика 22. Убачена плочица у раствор соне киселине и хидрогена



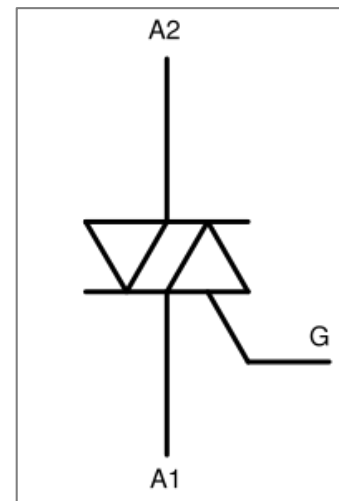
Слика 23. Изглед плочице након неколико минута пошто је потопљена у раствору соне киселине и хидрогена



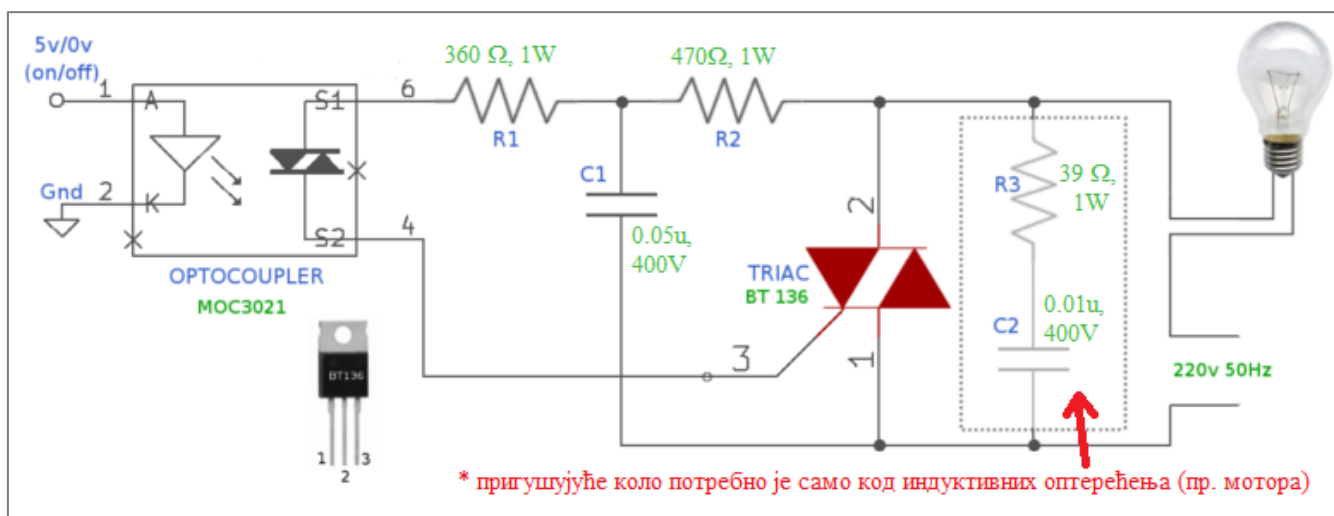
Слика 24. Изглед плочице након отклањања вишка бакра

2.4.3. Пластица напајања и прекидача

Ова пластица заправо јесте модификација оригиналне пластице на којој се налазе релеји који врше укључивање електромотора и грејача слика 6. Пошто релеји јесу електро-механичке компоненте за време њиховог укључивања/искључивања чује се звук лупања контаката који у неким тренуцима може да смета кориснику. Такође се услед тих удараца контакти механички троше. Из тих разлога извршена је модификација старе плоче, заправо направљена је нова с тим што су релеји замењени полупроводничким компонентама, односно триацима. Нова пластица без компоненти приказана је на слици 24. у претходној тачки где је описан фото поступак. Да би се разумео рад триака потребно је видети његов електрични симбол на слици 25. На крајевима A1 и A2 прикључује се електрично коло у коме желимо да се триак понаша као прекидач, односно у коме се врши прекид протока струје. На крају означеном словом „G“ што значи „гејт“ доводи се сигнал који врши укључивање триака. На слици 26. може се видети пример електричне шеме у којој се триак користи као прекидач за наизменичну струју. [4][5]



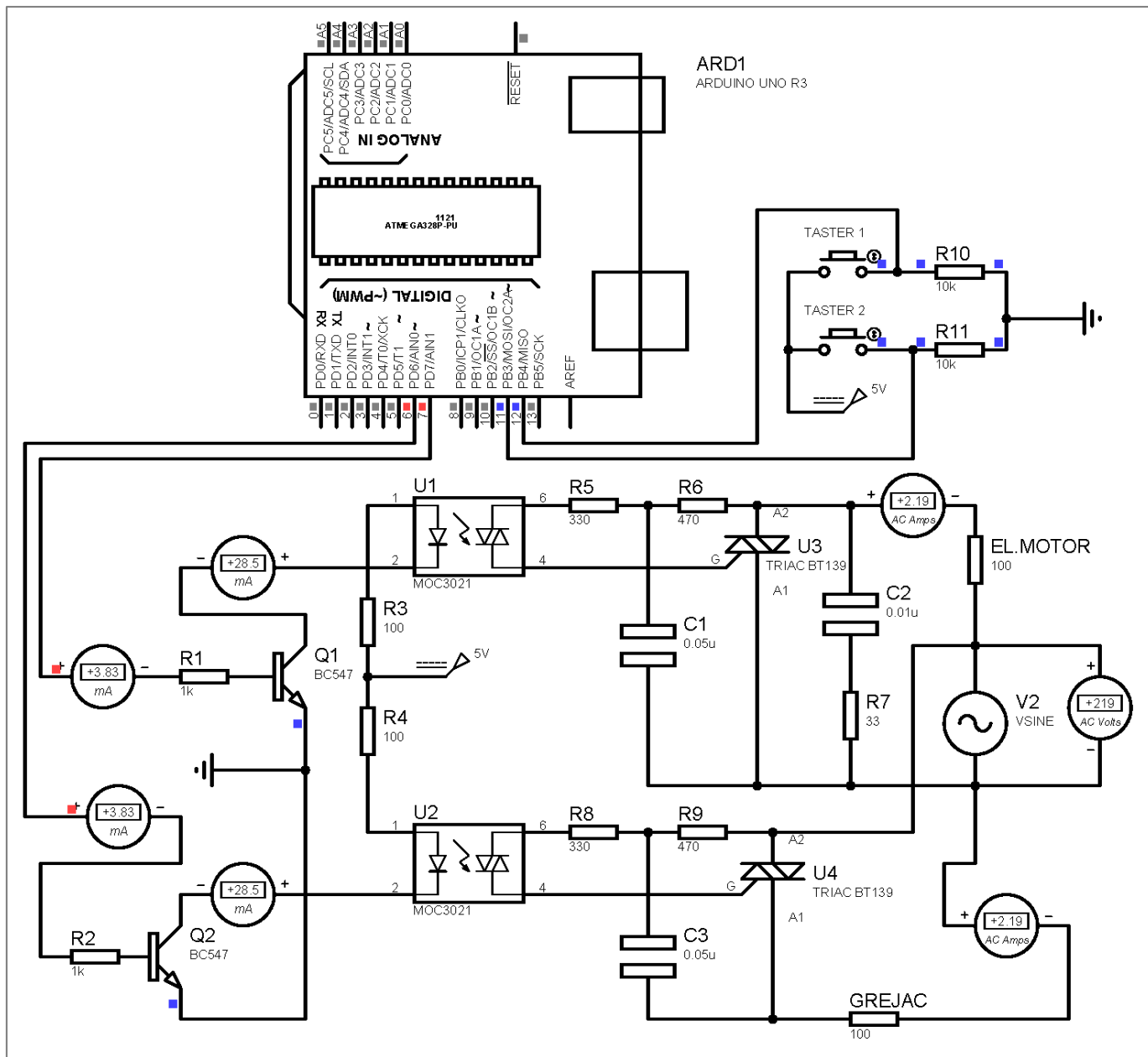
Слика 25. Електрични симбол триака [15]



Слика 26. Електрична шема у којој се триак користи као прекидач наизменичне струје [16]

У колу се може приметити и електрична компонента звана оптокаплер (Optocoupler MOC3021) која се користи за изолацију једног електричног кола од другог. Њихова највећа примена управо јесте у комбинацији са микроконтролерима, из разлога да би се микроконтролери изоловали од електричног кола у коме протиче наизменична струја. Рад оптокаплера заснива се емитовању светлости. Наиме, на пиновима 1 и 2 прикључена је диода која емитује светлост када се на пин 1 доведе напон од 5V. Тада полупроводничка компонента која се налази у оптокаплеру проводи струју. У овом случају полупроводничка компонента која се налази унутар оптокаплера јесте триак, као што се и може видети на горе приказаној шеми. [6]

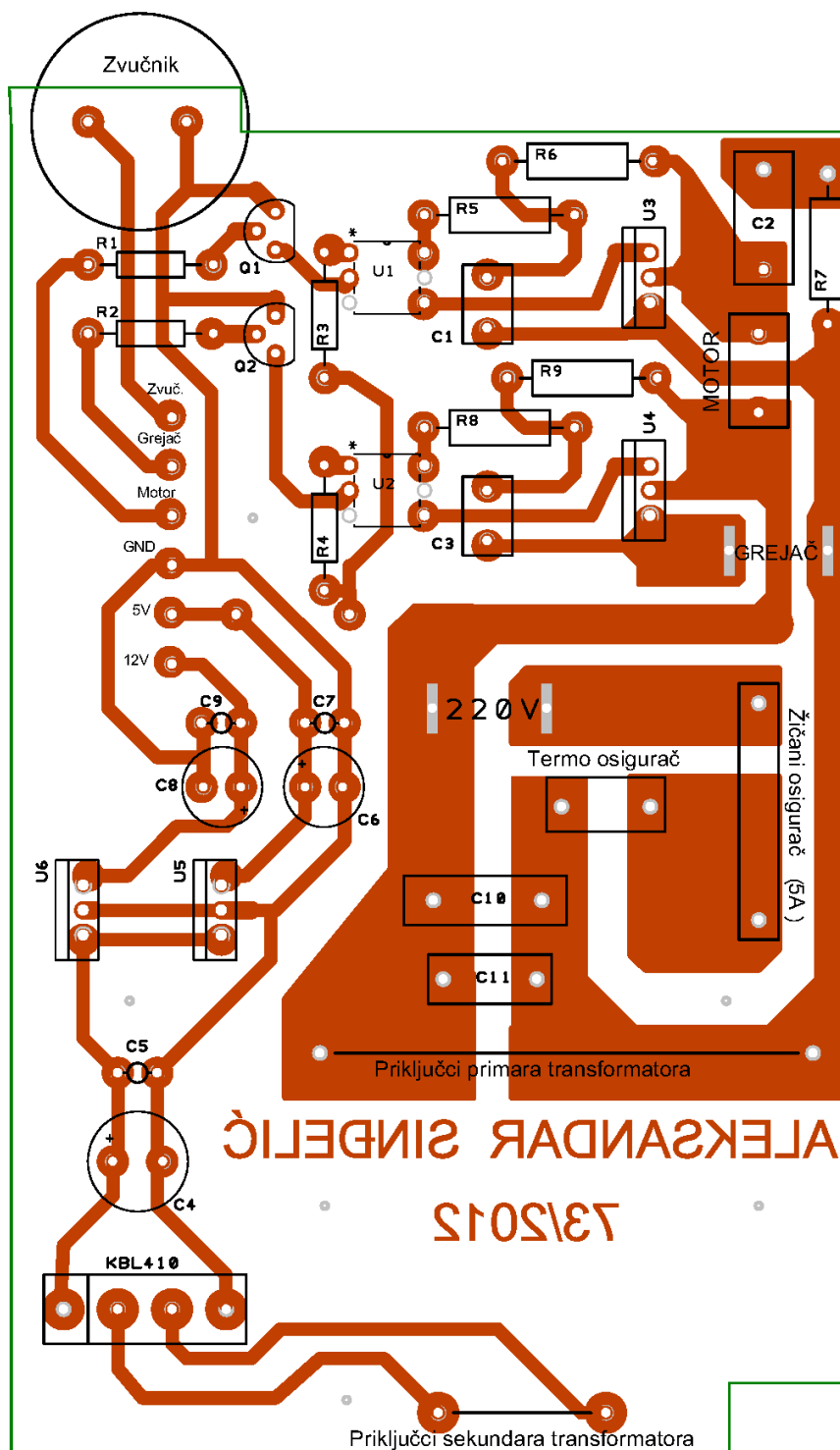
Пре самог цртања електричне шеме која ће бити израђена, било је потребно проверити у симулатору да ли ће та шема уопште радити. У софтверу „Proteus 8 Professional“ извршена је симулација паљења грејача и електромотора уз помоћ Ардуино платформе слика 27., где је мерена јачина електричне струје у свим гранама електричног кола, како са сигурношћу не би дошло до кvara Ардуино платформе.



Слика 27. Електрична шема тестирана у симулатору „Proteus“

На тестираној шеми електромотор се укључује/искључује на тастеру 1, а грејач на тастеру 2. На слици 27. може да се примети да јачина електричне струје на пиновима 6 и 7 на Ардуино платформи износи 3,83mA. Пошто максимална дозвољена јачина струје по пину износи 40mA, то значи да је Ардуино безбедан. Такође за транзисторе Q1 и Q2 максимална дозвољена струја на колектору јесте јачине од 100mA, па пошто је стварна 28,5mA, што се такође може видети на слици 27. указује да су и транзистори безбедни.

Треба напоменути да триаци у шеми која је тестирана слика 27. и коју су употребљени у изради ове плочице носе ознаку „BT139“, а не „BT136“ као у шеми на слици 26. То је измењено зато што триак „BT136“ издржава максималну јачину електричне струје до 4А, а „BT139“ до 16А. Након успешног тестирања одрађен је дизајн ове плочице у „PCB Artist-u“ који се може видети на слици 28. Ознаке компоненти на слици 28. потпуно су идентичне онима на тестираној шеми на слици 27.



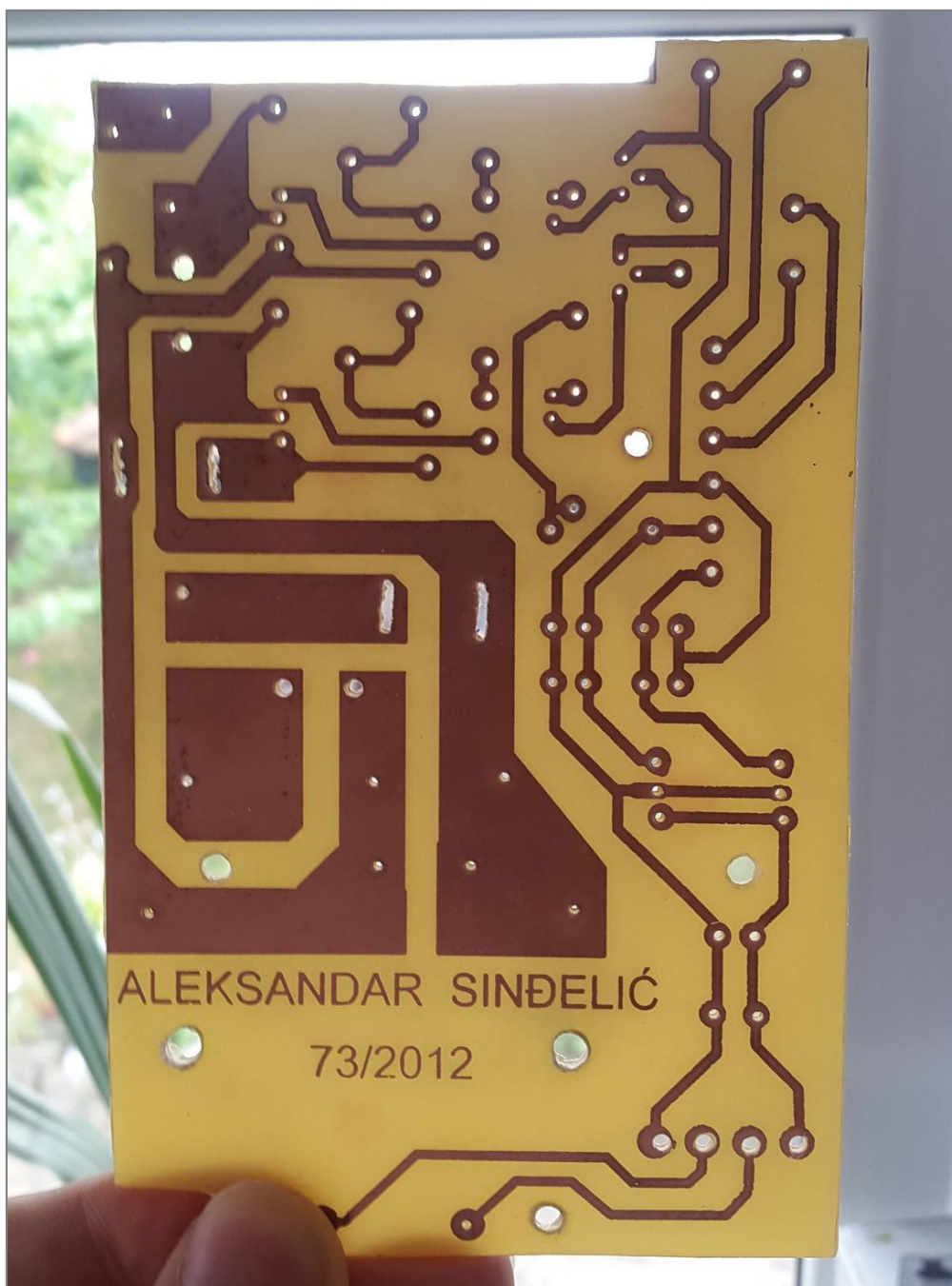
Слика 28. Дизајн плочице напајања и прекидача

У табели испод дат је списак свих компоненти са плочице са додатним информацијама:

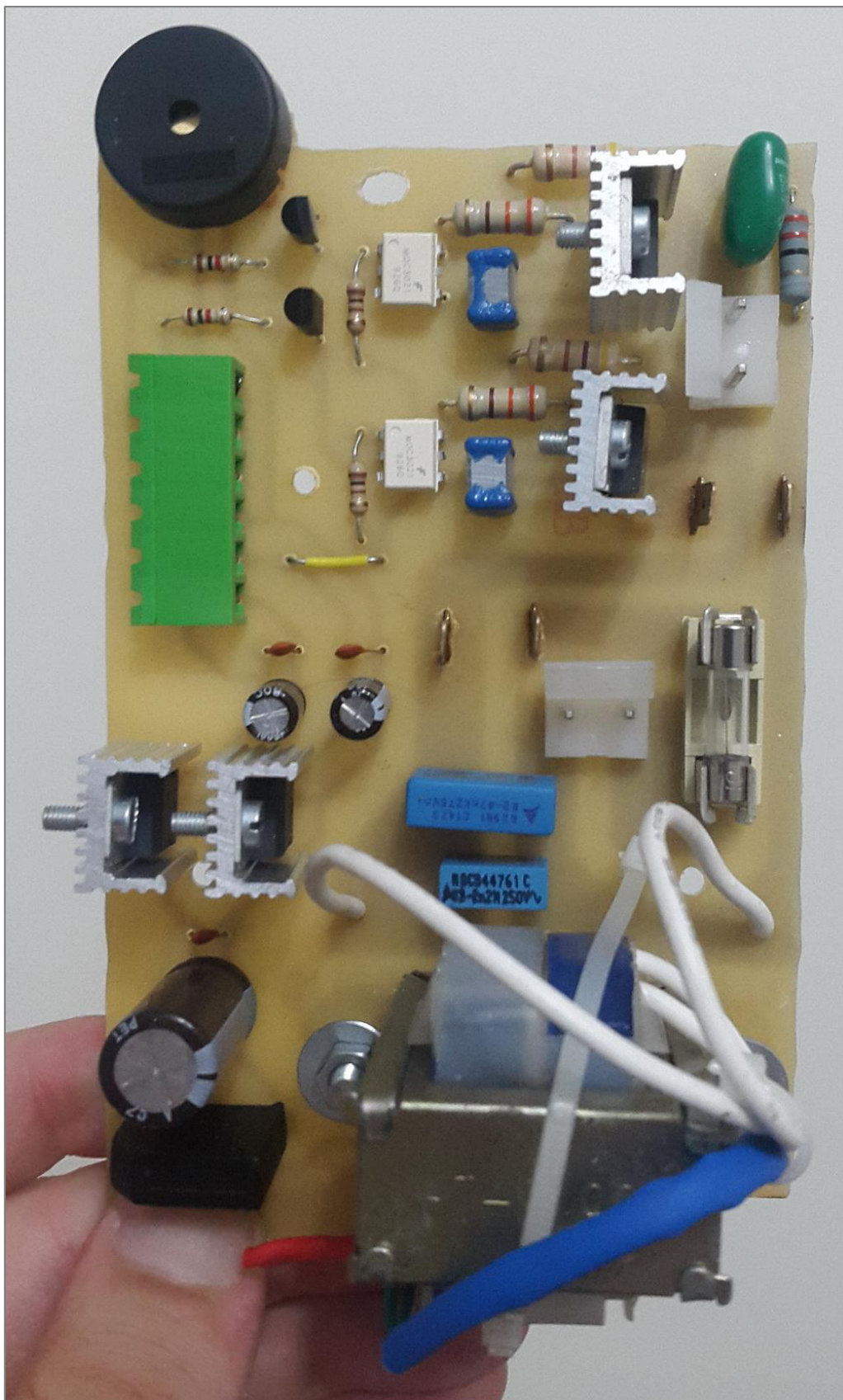
Ознака	Компонента	Опис	Комада	Цена по komadu (RSD)
R1 = R2	Карбонфилм отпорник	1kΩ, 0.25W	2	2,00
R3 = R4	Карбонфилм отпорник	100Ω, 0.25W	2	2,00
R5 = R8	Карбонфилм отпорник	330Ω, 1W	2	6,00
R6 = R9	Карбонфилм отпорник	470Ω, 1W	2	6,00
R7	Металфилм отпорник	33Ω, 1W	1	8,00
C1 = C3	МКП кондензатор	47nF, 400VAC	2	25,00
C2	Полиестер кондензатор	10nF, 400VAC	1	29,00
C4	Електролитски кондензатор	2200uF, 25V	1	89,00
C6 = C8	Електролитски кондензатор	100uF, 16V	2	6,00
C5 = C7 = C9	Керамички кондензатор	103 (10000pF), 50V	3	3,00
C10	МКП кондензатор	47nF, 275VAC	1	20,00
C11	МКП кондензатор	2.2nF, 250VAC	1	15,00
Q1 = Q2	НПН транзистор	BC547	2	4,00
U1 = U2	Оптокаплер	МОС3021	2	45,00
U3 = U4	Триак	BT139	2	91,00
U5	Стабилизатор напона	UA7805 (5V)	1	39,00
U6	Стабилизатор напона	UA7812 (12V)	1	39,00
KBL410	Грецов спој	4A/1000V	1	39,00
Zvučnik	Пиезо звучник		1	/
Žičani osigurač (5A)	Кућиште + осигурач	5A	1	25,00 + 15,00
Termo osigurač*	Прикључак	2-пински	1	/
MOTOR*	Прикључак	2-пински	1	/
GREJAČ*	Прикључак	/	1	/
/	Трансформатор	220/2x6V, 10VA	1	495,00
/	Хладњак	/	4	30,00
УКУПНО:				1316,00

* - означава да је компонента скинута са оригиналне плочице.

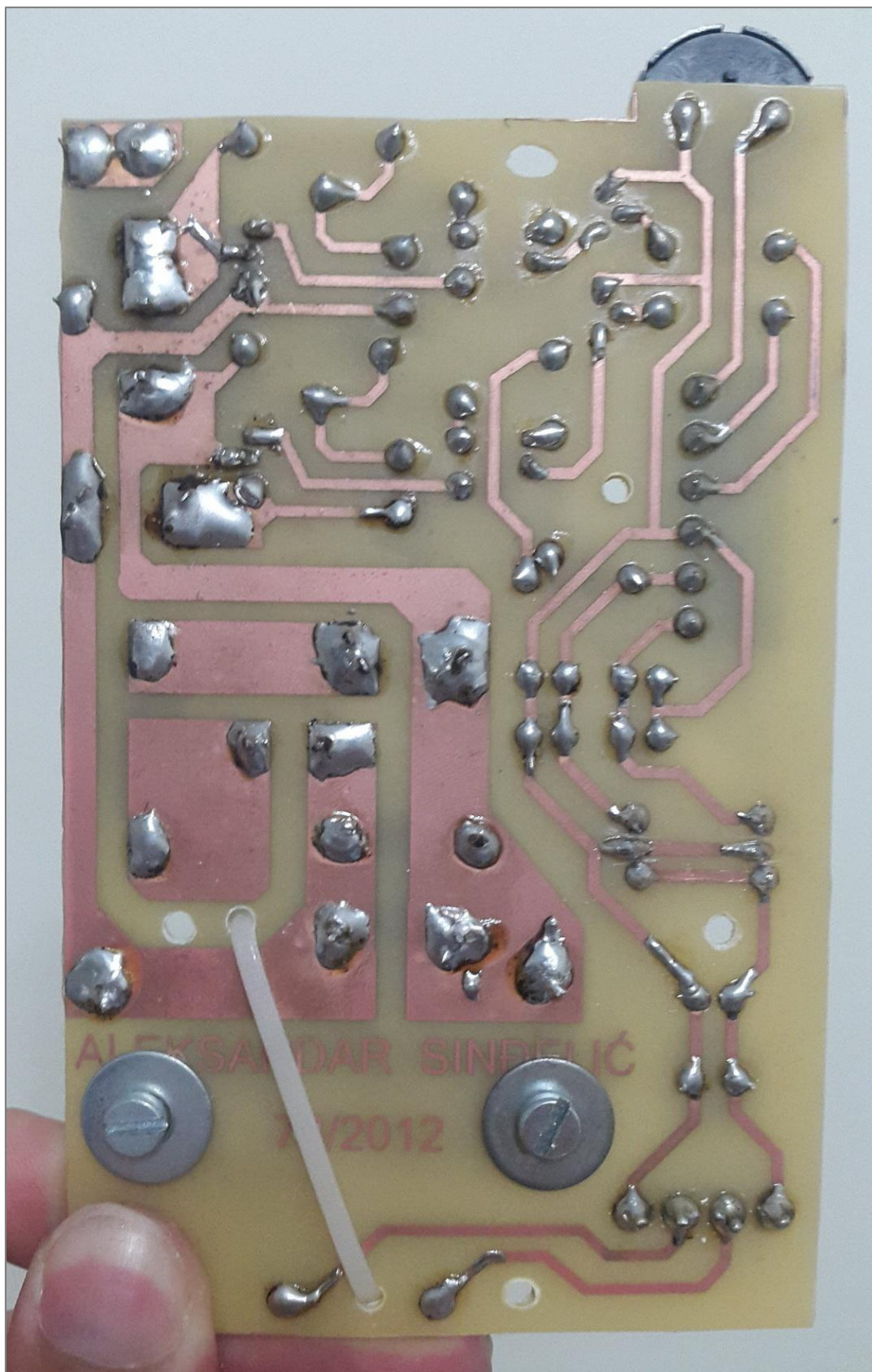
У тачки 2.4.2. приказан је целокупан фото поступак израде ове плочице. На слици 24. може се видети ова плочица пре бушења рупа и прикачивања свих компоненти. На слици 29. приказана је ова плочица након бушења рупи, док је на сликама 30. и 31. ова плочица приказана након прикачивања (летовања) свих компоненети. На слици 32. може се видети поређење између старе и нове плочице напајања и прекидача.



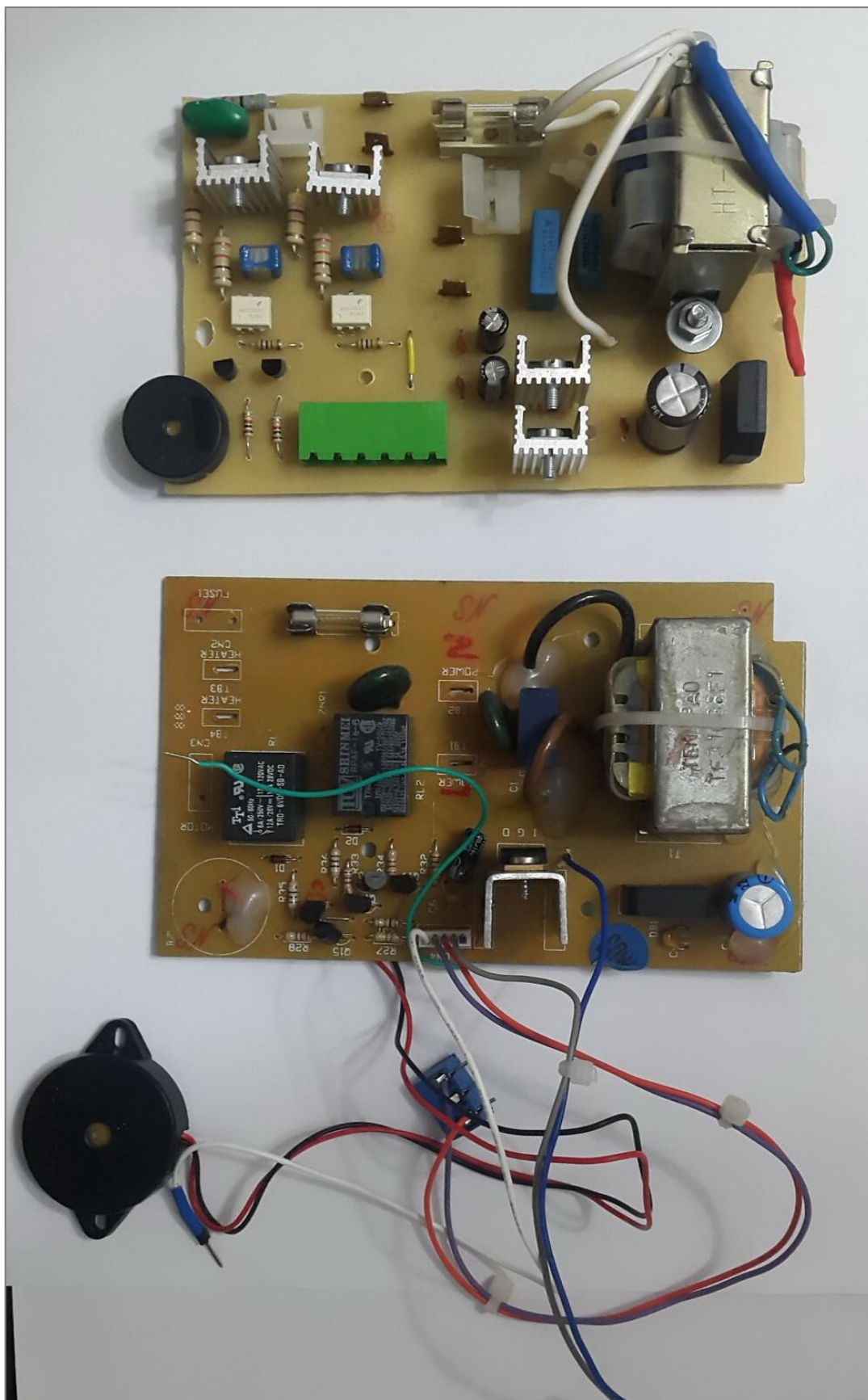
Слика 29. Плочица напајања и прекидача након бушења рупи



Слика 30. Платица напајања и прекидача са предње стране након прикачивања свих компоненти



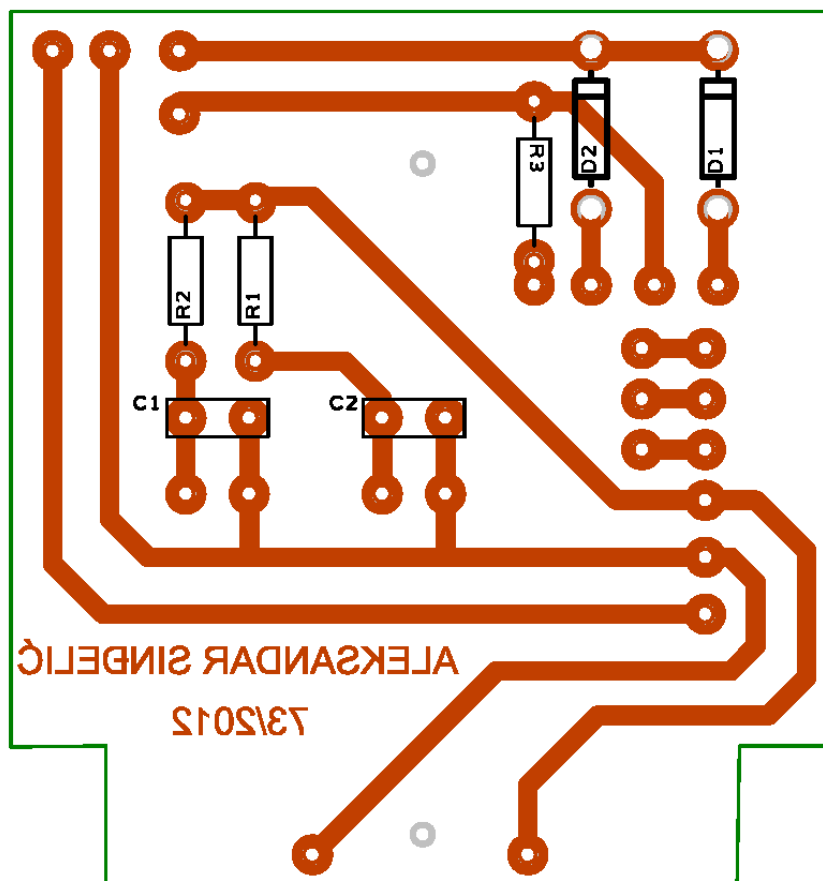
Слика 31. Пластица напајања и прекидача са задње стране након прикачивања свих компоненти



Слика 32. Поређење старе и нове плочице напајња и прекидача

2.4.4. Међуплочица

Преко ове плочице пре свега доводи се напајање Ардуино платформи, а такође су прикачене и све остале компоненте које се користе. У те компоненте спадају сензори за читавање температуре којима се доводи V_{cc} (5V) напон и дисплеј. Дизајн ове плочице израђен у „PCB Artist-u“ може се видети на слици 33.

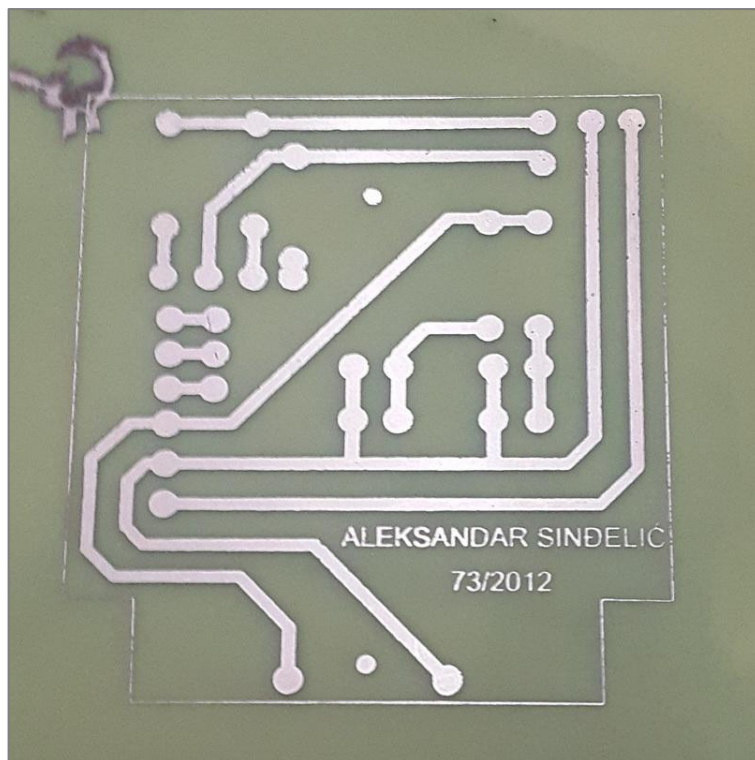


Слика 33. Дизајн међуплочице

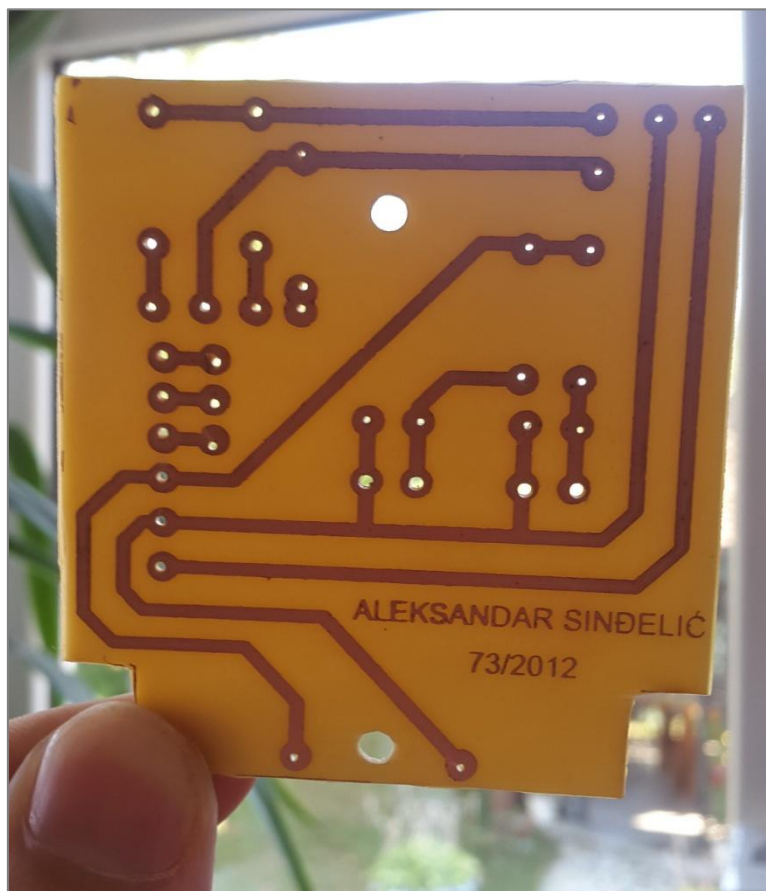
У табели испод дат је списак свих компоненти са плочице са додатним информацијама:

Ознака	Компонента	Опис	Комада	Цена по комаду (RSD)
R1	Карбонфилм отпорник	47k Ω , 0.25W	1	2,00
R2	Карбонфилм отпорник	180k Ω , 0.25W	1	
R3	Карбонфилм отпорник	220 Ω , 0.25W	1	
C1 = C2	Полиестер кондензатор	100nF	2	10,00
D1 = D2	Диода	1N4007	2	2,00
УКУПНО:				30,00

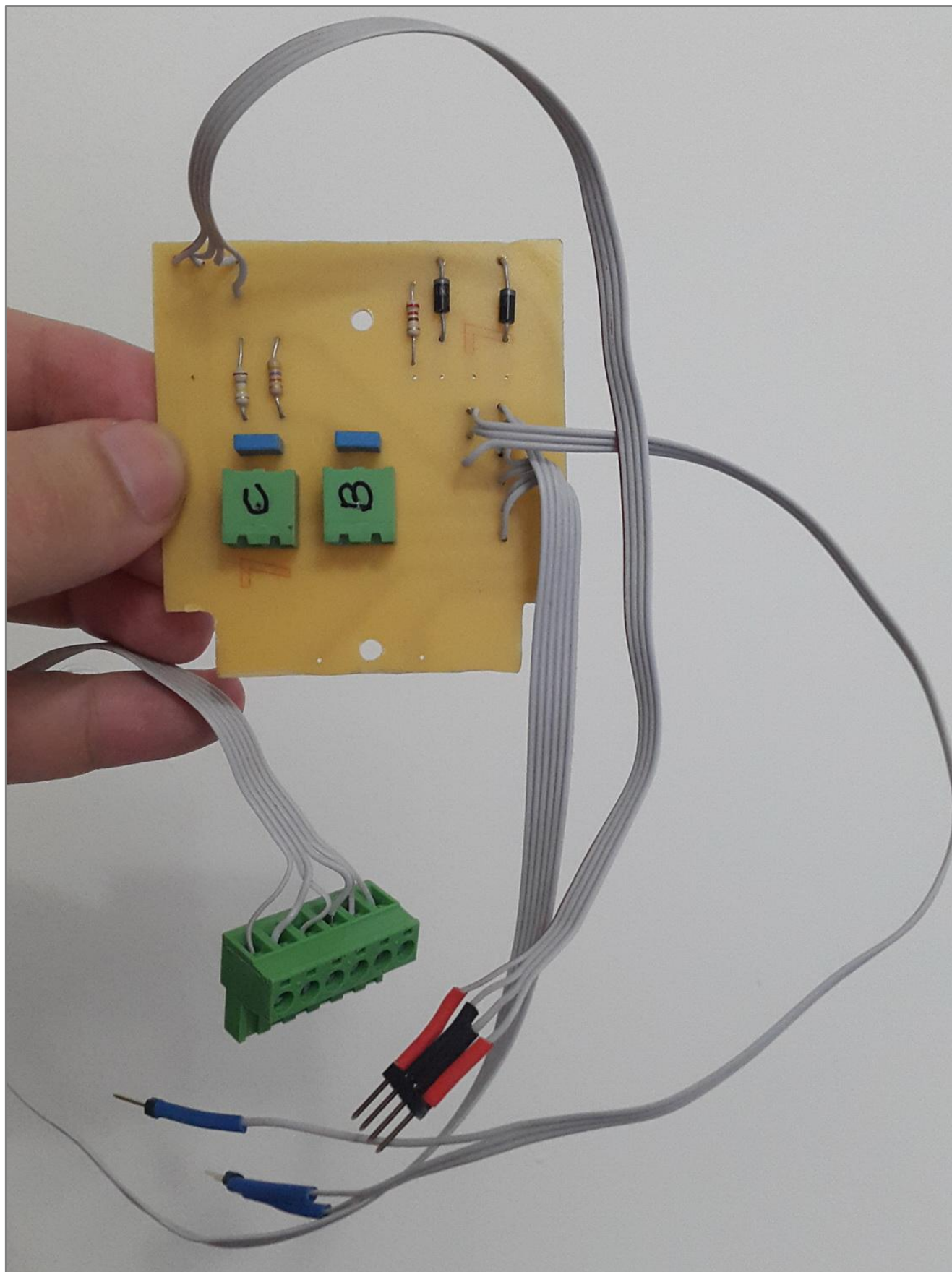
На слици 34. може се видети ова плочица пре бушења рупа, на слици 35. након бушења, док се на сликама 36. и 37. ова плочица може видети након прикључивања свих жица и компоненти, осим дисплеја.



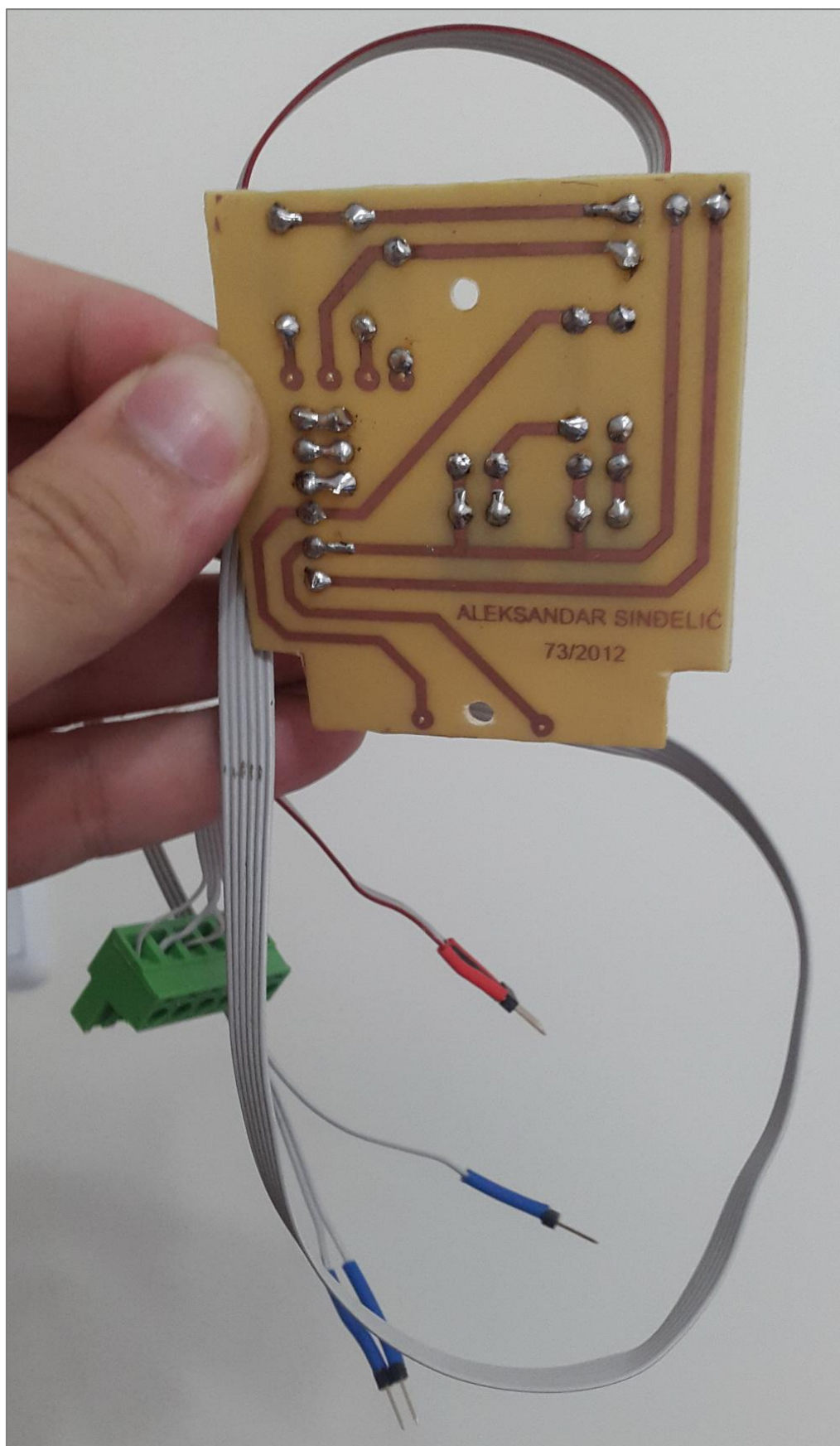
Слика 34. Међуплочица пре обликовања и бушења отвора



Слика 35. Међуплочица након обликовања и бушења отвора



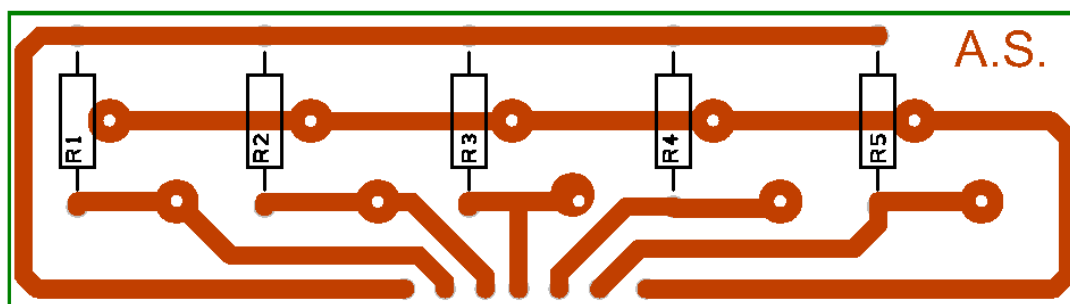
Слика 36. Међуплочица са предње стране након постављања компоненти



Слика 37. Међуплочица са задње стране након постављања компоненти

2.4.5. Пластица са тастерима

Као што и сам назив каже на ово пластици налазе се сви тастери који су потребни за функционисање овог уређаја. Дизајн ове пластице израђен у „PCB Artist-u“ може се видети на слици 38.

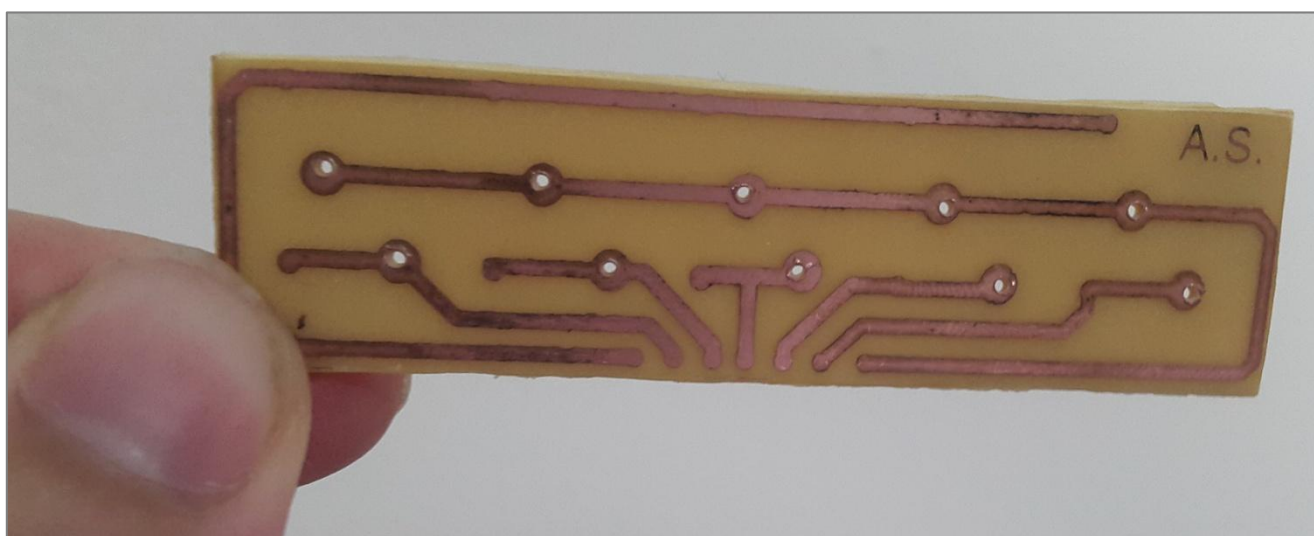


Слика 38. Дизајн пластице са тастерима

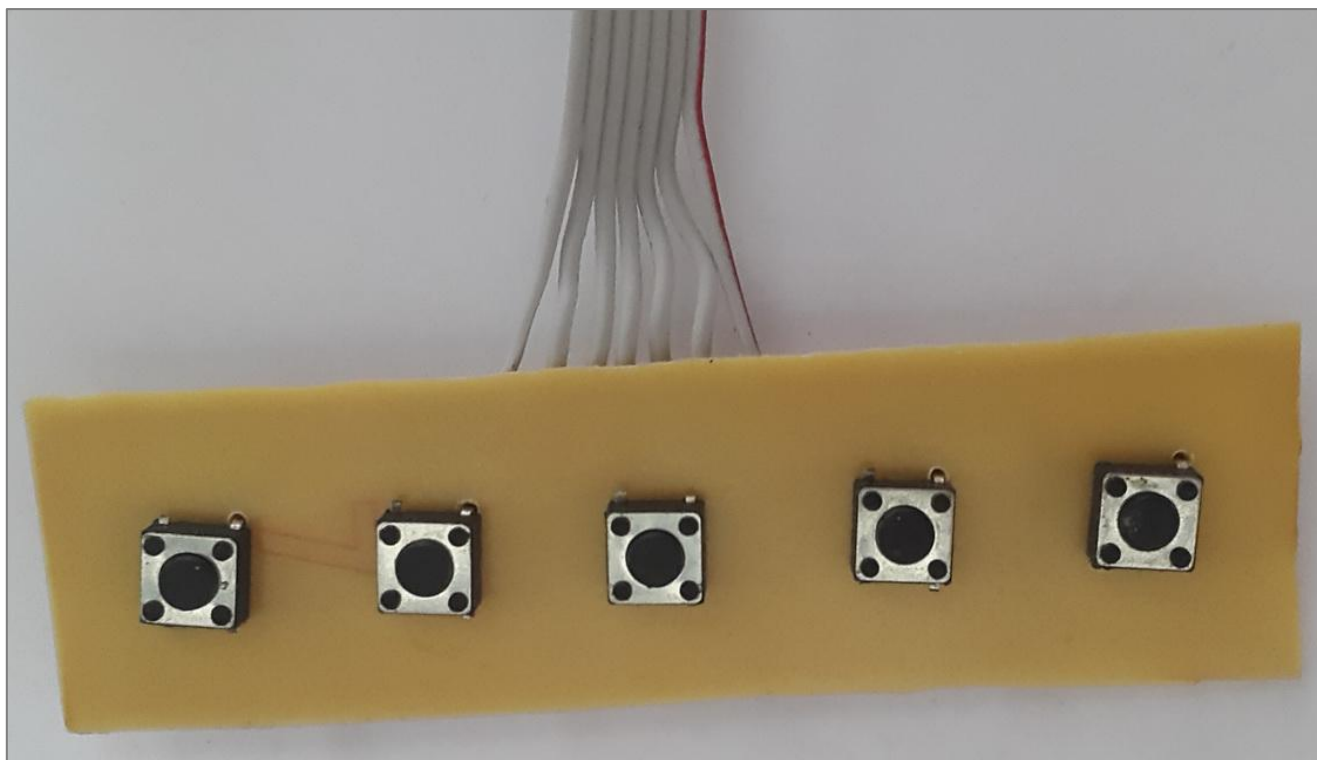
У табели испод дат је списак свих компоненти са пластице са додатним информацијама:

Ознака	Компонента	Опис	Комада	Цена по комаду (RSD)
R1= R2= R3= R4= R5	Карбонфилм отпорник	10kΩ, 0.25W	5	2,00
/	Тастер	/	5	10,00
УКУПНО:				60,00

На слици 39. може се видети ова пластица након бушења отвора, док се на сликама 40. и 41. ова пластица може видети након постављања свих компоненти и жица.

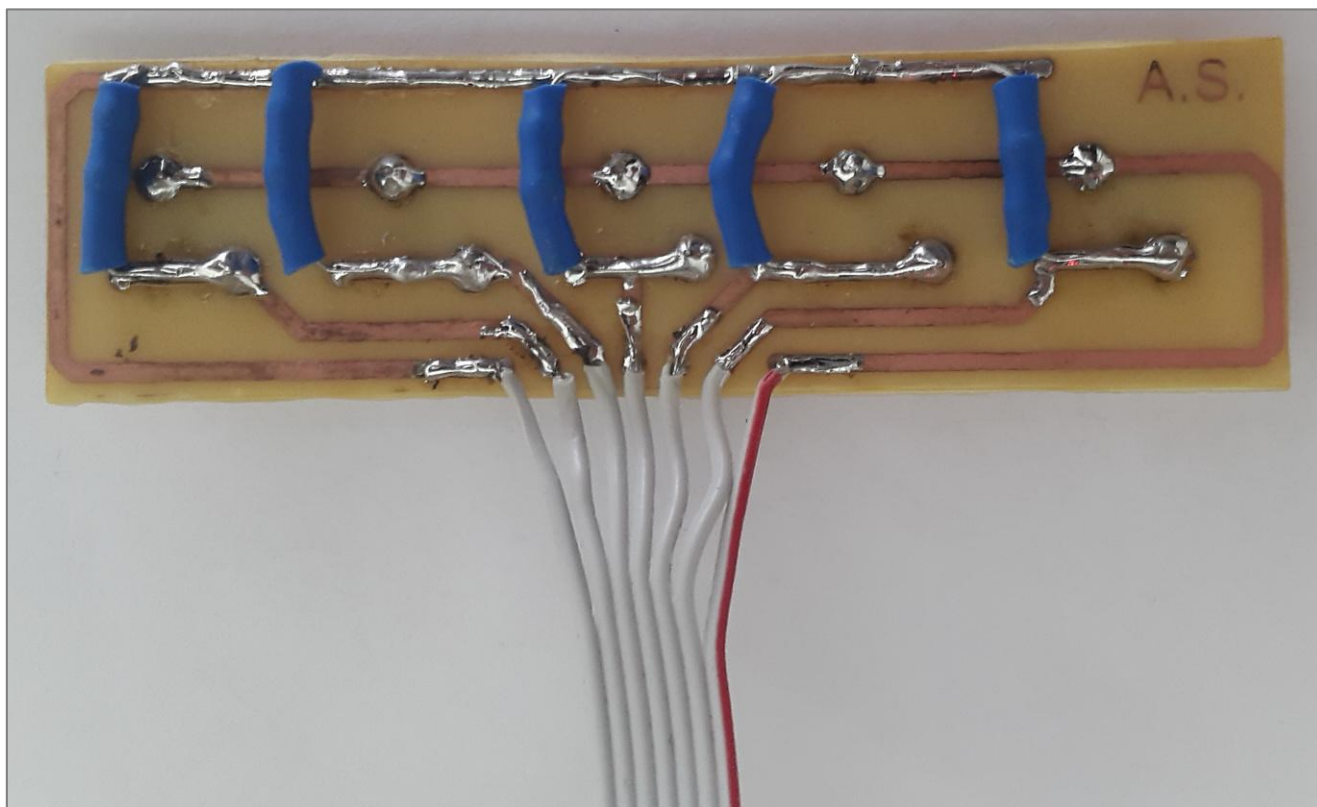


Слика 39. Пластица са тастерима након бушења отвора



Слика 40. *Плочица са тастерима са предње стране након постављања компоненти*

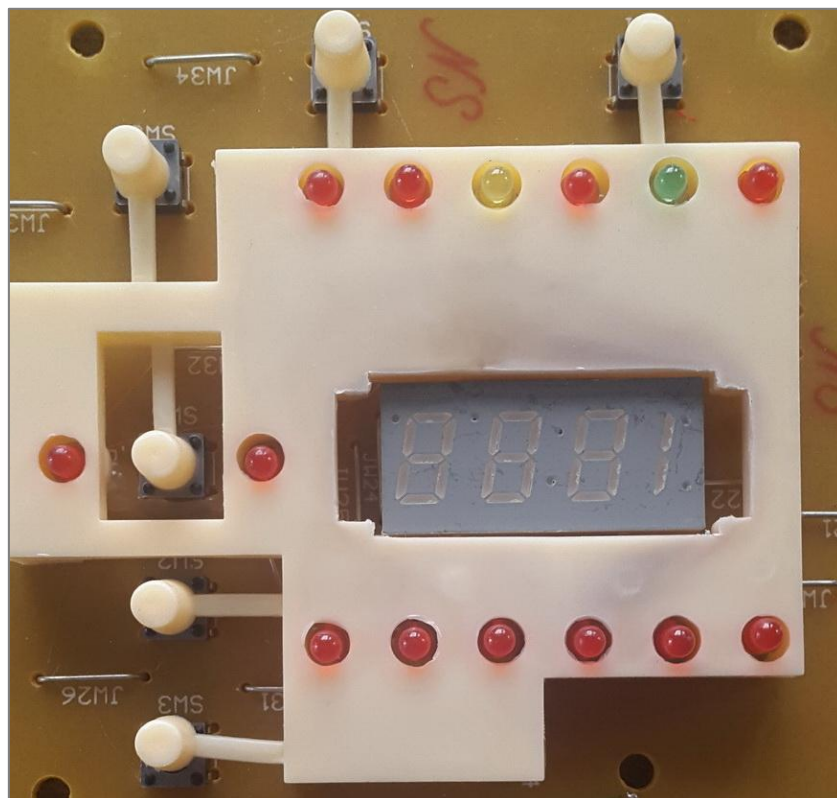
На слици 41. може се видети да је на отпорницима постављен термоскупљајући бужир, како би се кућиште отпорника изоловало од електричних водова.



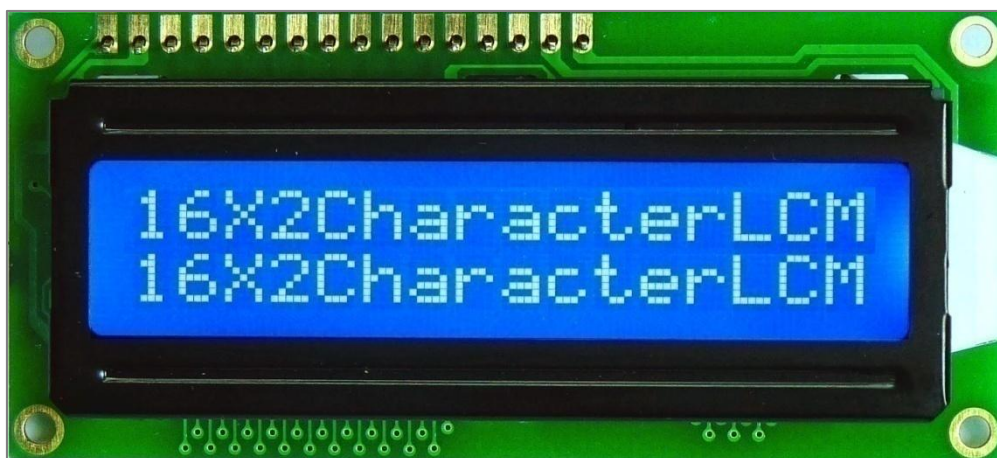
Слика 41. *Плочица са тастерима са задње стране након постављања компоненти*

2.5. LCD (Liquid Crystal) дисплеј

Претходни показивач информација слика 42. кориснику претстављао је 4-цифрени дисплеј који је показивао преостало време до завршетка изабраног програма, као и скуп LED диода које су указивале на изабран програм и тренутну операцију коју уређај обавља. У жељи да се унапреди систем који пружа информације кориснику убачен је LCD (Liquid Crystal) 16x2 дисплеј слика 43. на коме се може прочитати изабран програм, преостало време до завршетка програма као и температура унутар простора у коме се пече храна.



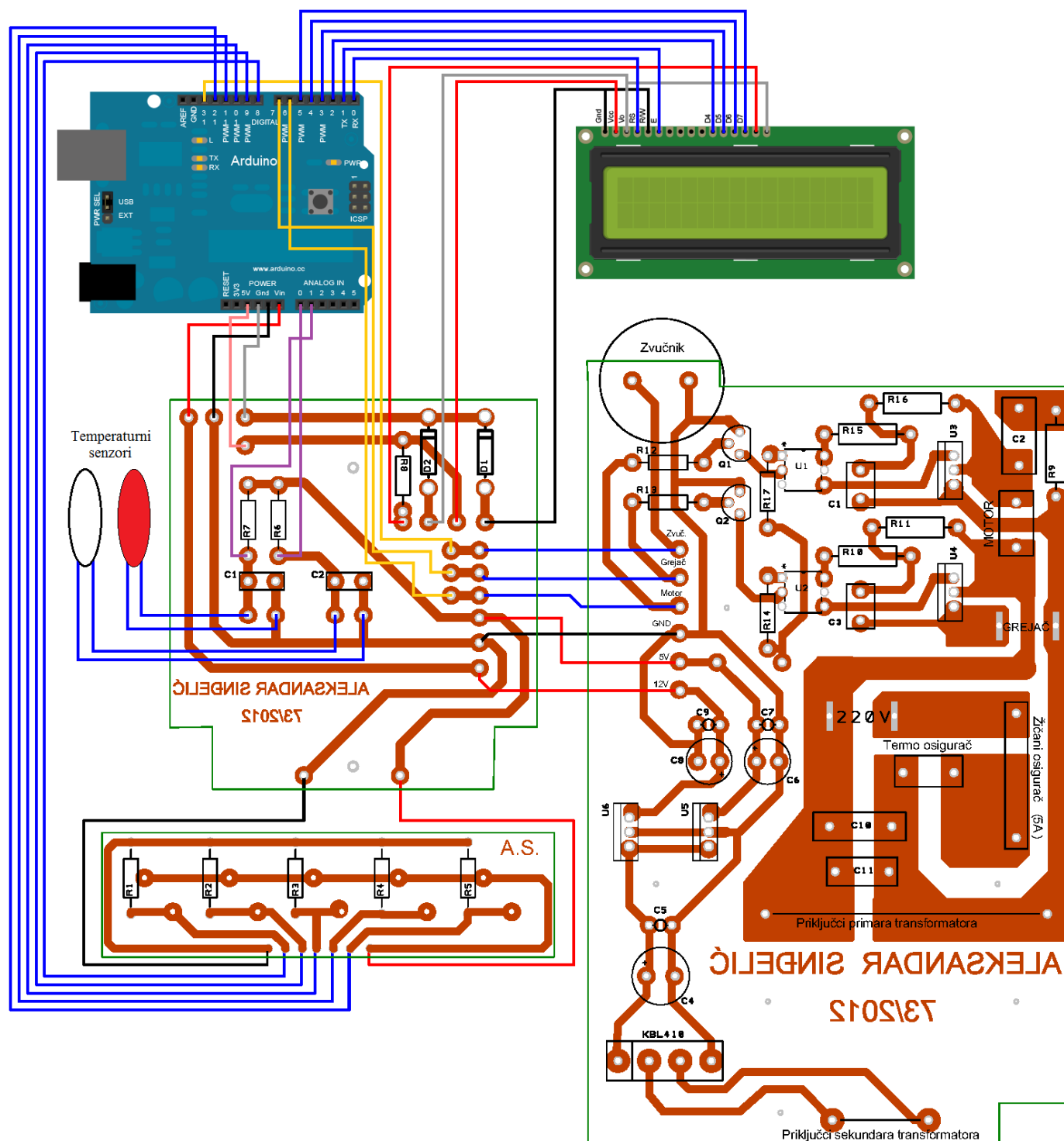
Слика 42. Претходни показивач информација



Слика 43. LCD (Liquid Crystal) 16x2 дисплеј

2.6. Шема повезивања свих компоненти

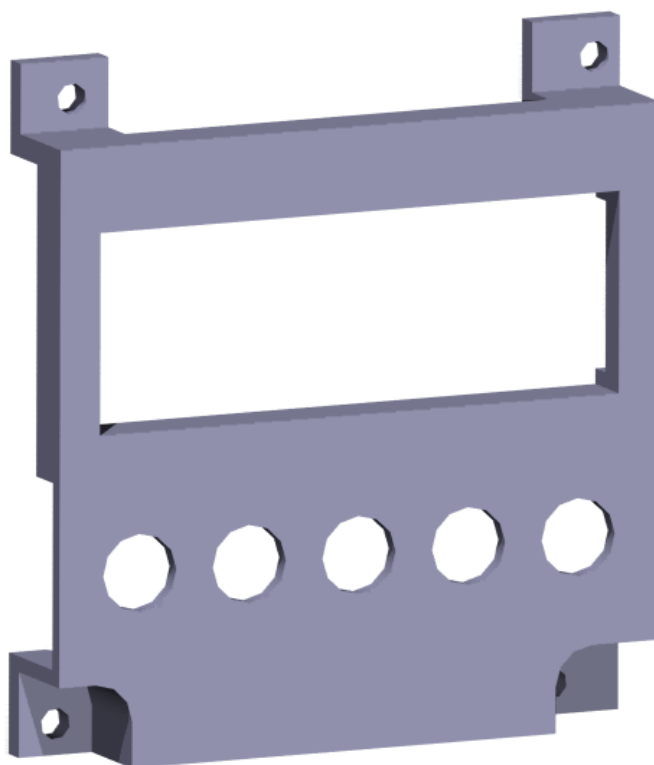
Након израде свих потребних плочица потребно их је правилно повезати како би све исправно функционисало. Шема повезивања дата је на слици 44. где се може тачно видети како је потребно повезати све компоненте.



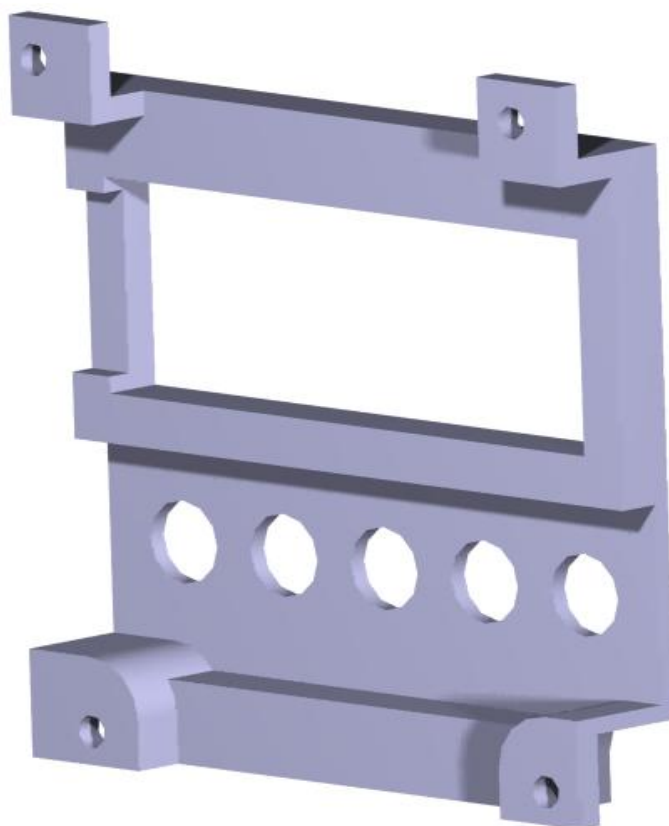
Слика 44. Шема повезивања свих компоненти

2.7. Рам

Пошто је извршена измена дисплеја, као и положај тастера на предњем панелу, било је потребно израдити тзв. рам како би монтирање претходно израђених и измењених компоненти било могуће. Пошто је овај рам било потребно израдити од пластике он је израђен на 3D штампачу који се налази на „Факултету инжењерских наука“ у „Центру информационих технологија (СИТ)“. Пре израде било је потребно одрадити 3D модел овог рама. Модел слика 45. и 46. одрађен је у софтверу „Catia R5“.

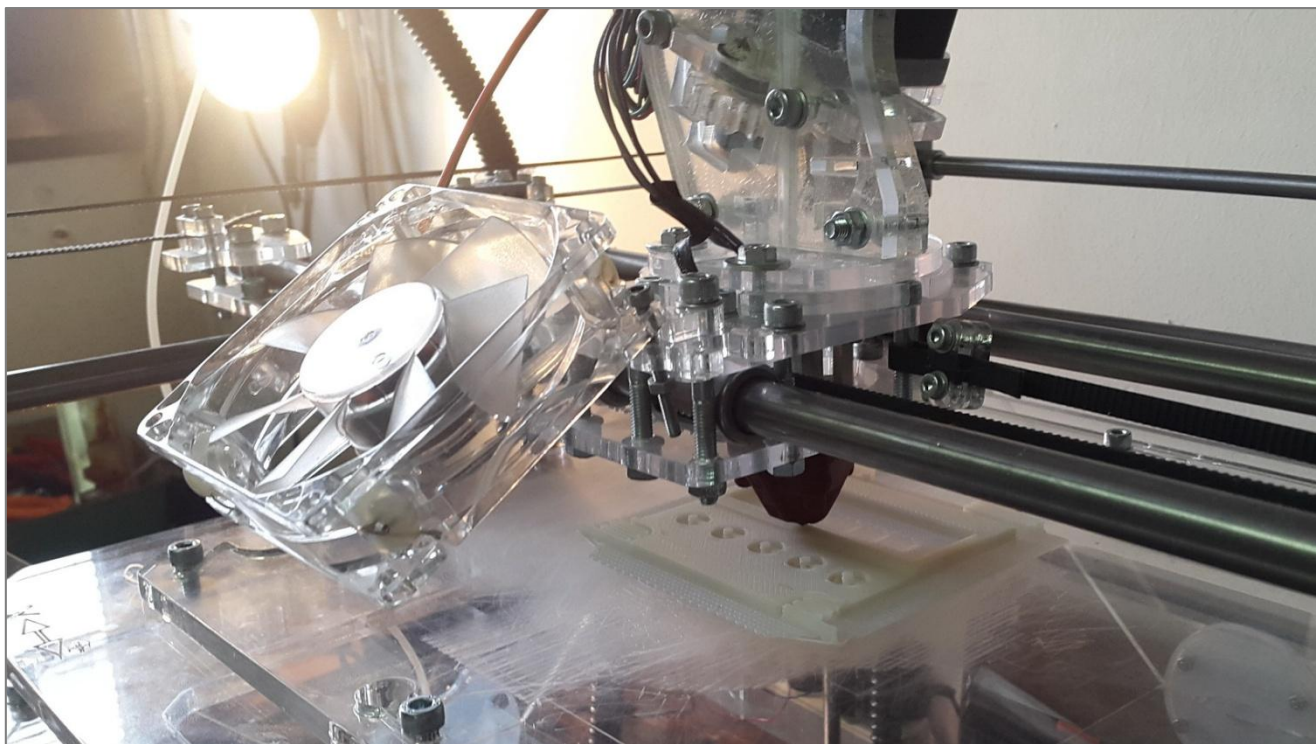


Слика 45. Изглед рама са предње стране

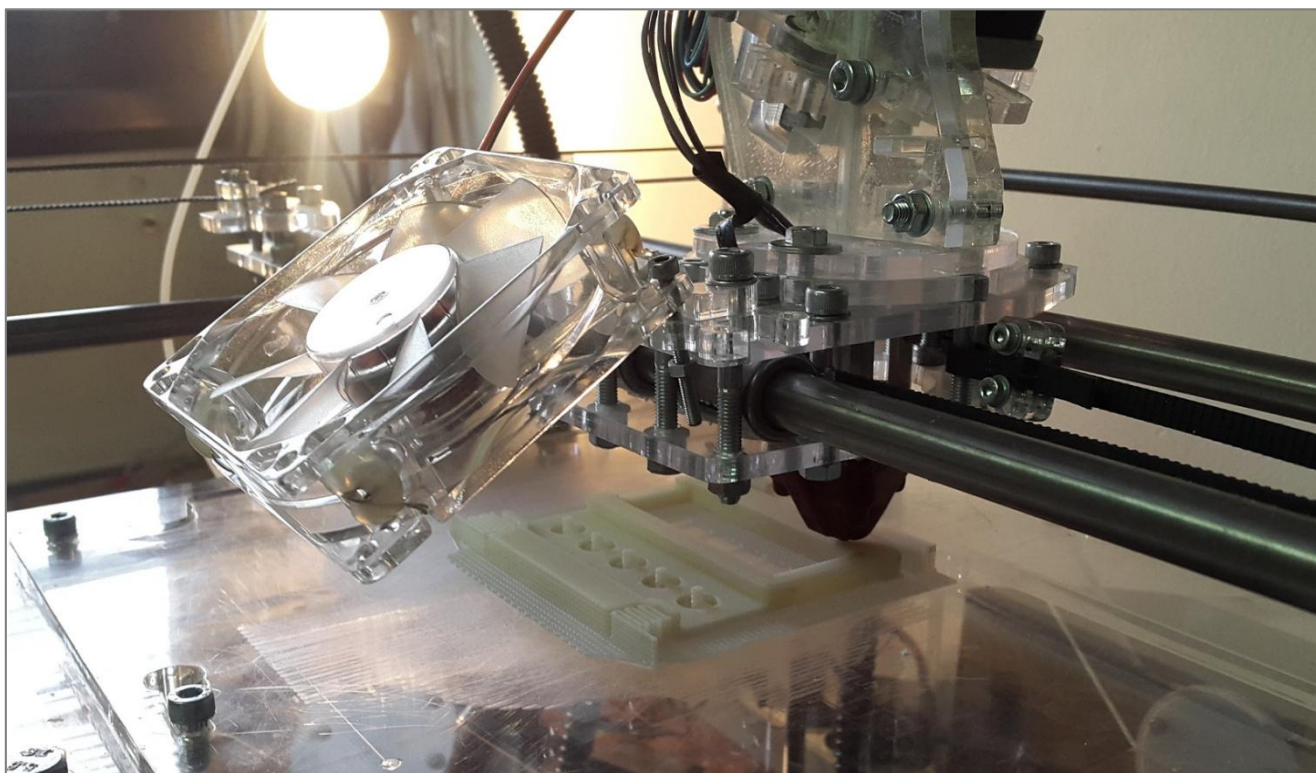


Слика 46. Изглед рама са задње стране

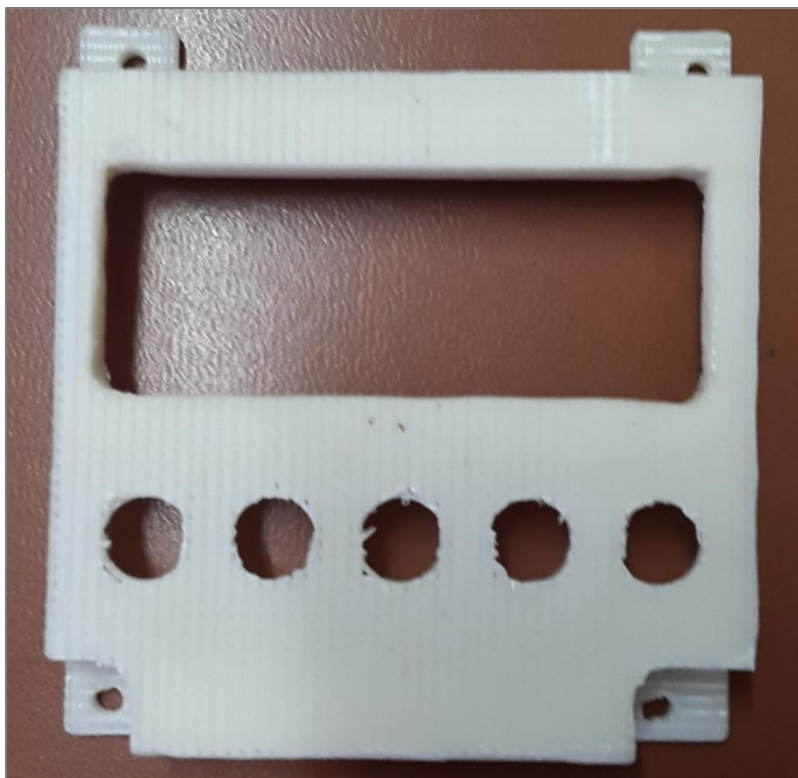
Процес 3D штампања може се видети на сликама 47. и 48. На сликама 49. и 50. може видети израђен рам, док се на сликама 51. и 52. може видети исти рам након фарбања.



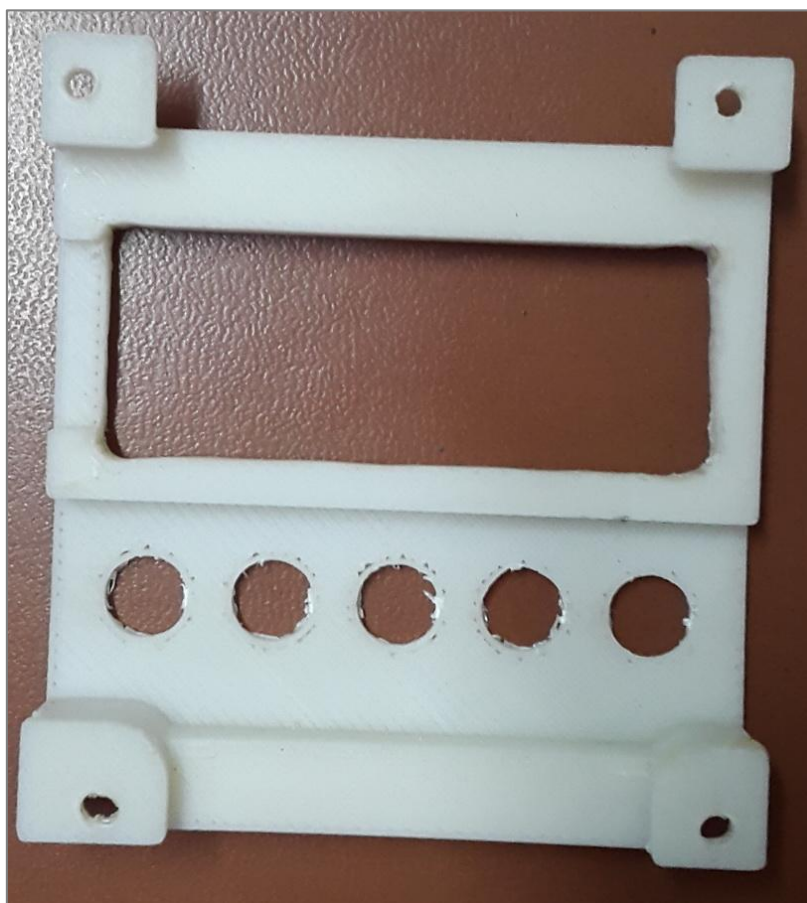
Слика 47. Процес 3D штампања, прва слика



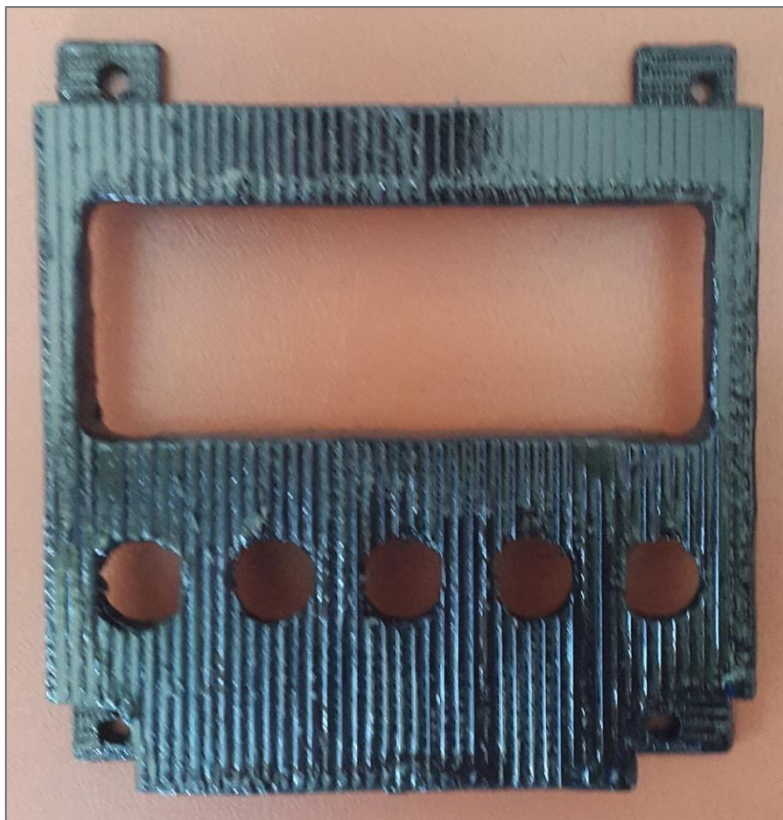
Слика 48. Процес 3D штампања, друга слика



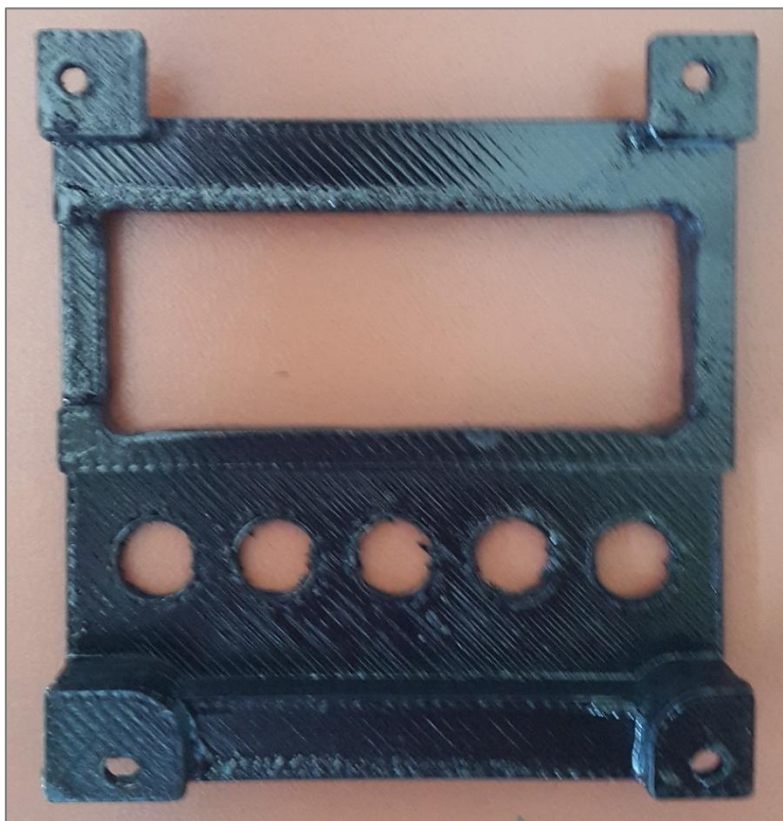
Слика 49. Рам након израде са предње стране



Слика 50. Рам након израде са задње стране



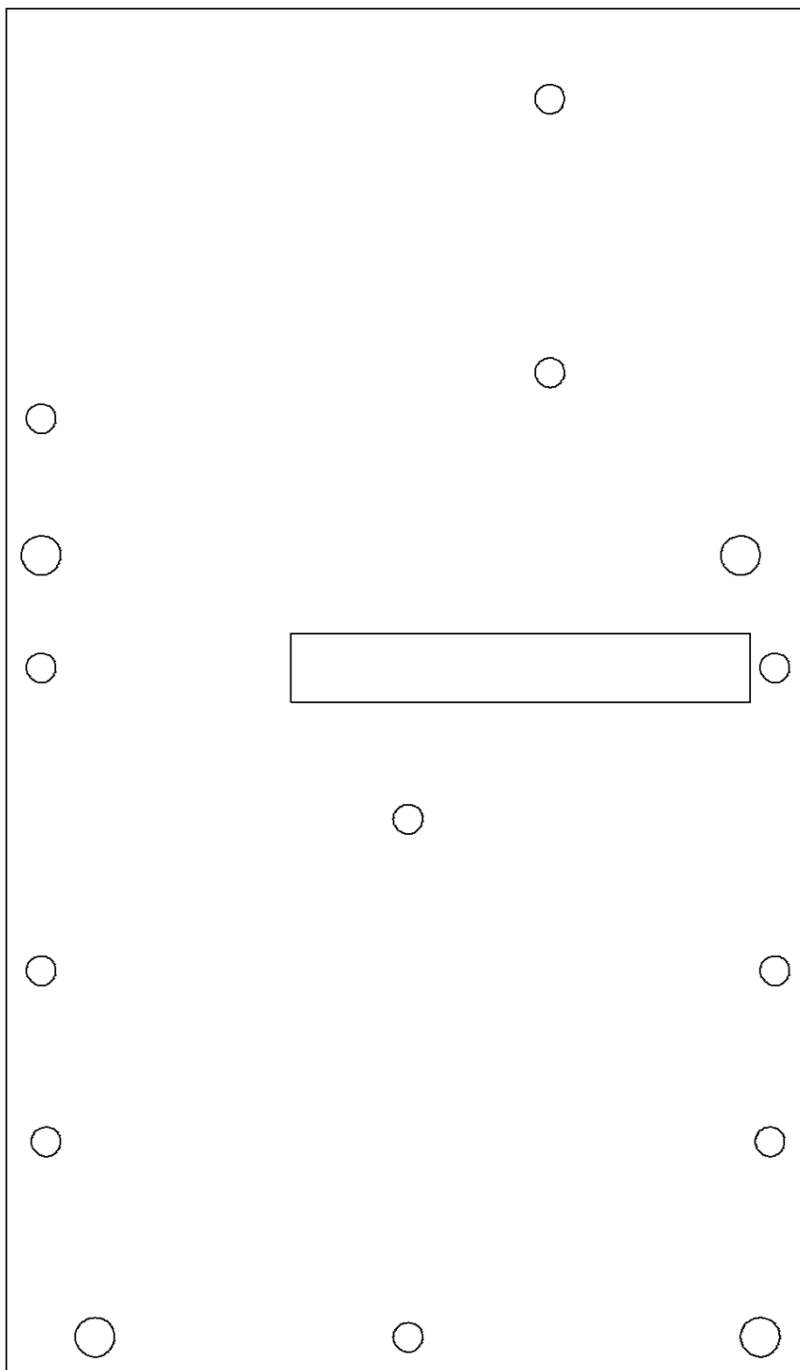
Слика 51. Рам након фарбања са предње стране



Слика 52. Рам након фарбања са задње стране

2.8. Пертинакс плоча

Да би се све склопило како треба исечена је плоча од пертинакса дебљине 1mm. Дизајн плоче такође је као и рам израђен у софтверу „Catia R5“. На слици 53. може се видети пројекција ове плоче са предње стране.

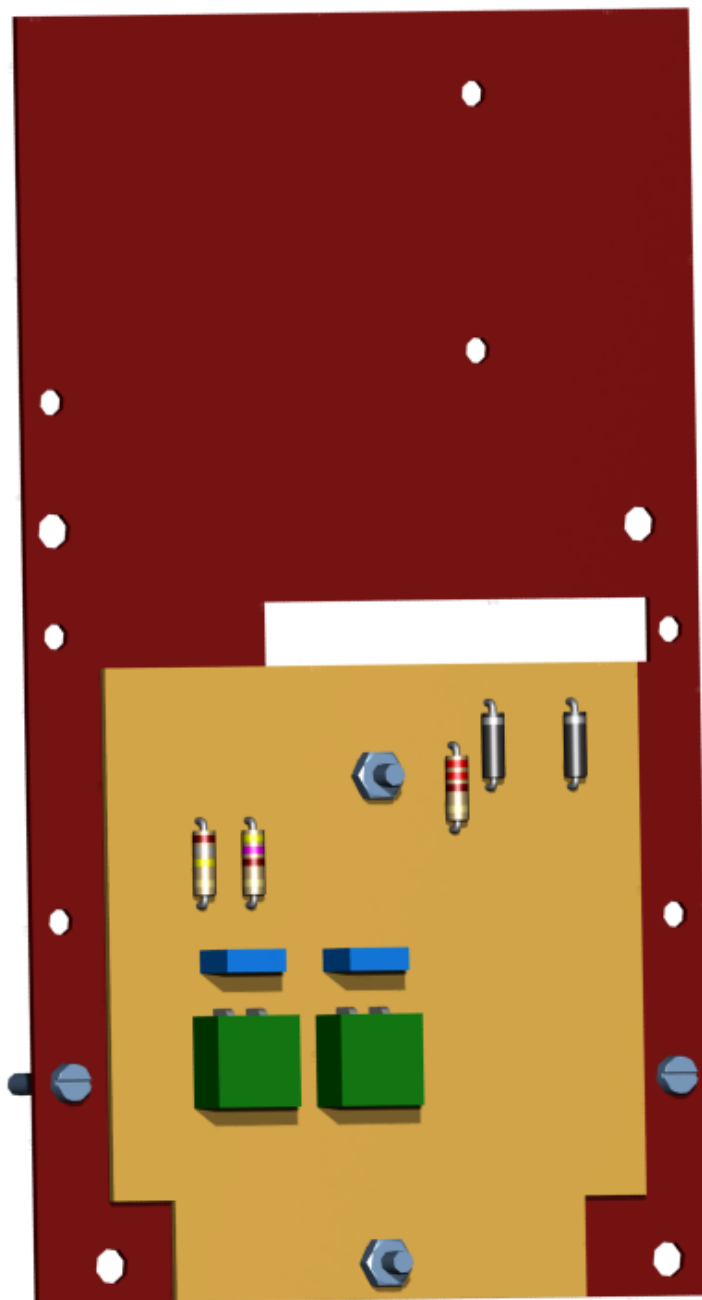


Слика 53. Пертинакс плоча

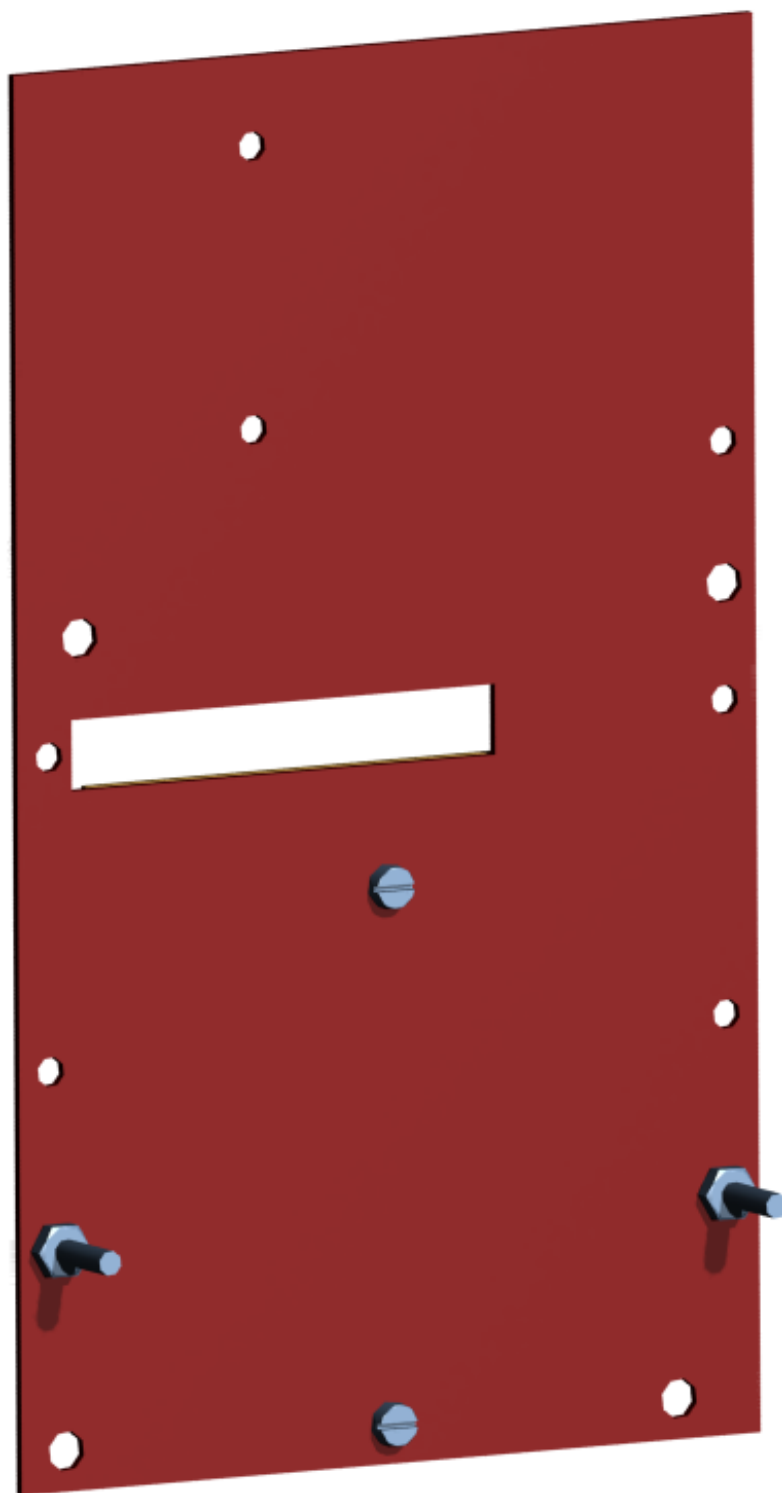
2.9. Склоп

Пошто су израђени сви потребни делови, потребно их је правилно спојити. На који начин и како су спојени сви делови овог пројекта може се видети на сликама испод, на којима је приказан модел склопа који је одрађен у софтверу „Catia R5“.

Први корак јесте причвршћивање међуплочице и пертинакс плоче, као и причвршћивање два шрафа која ће држасти плочицу са тастерима што се и може видети на слици 54. и 55.

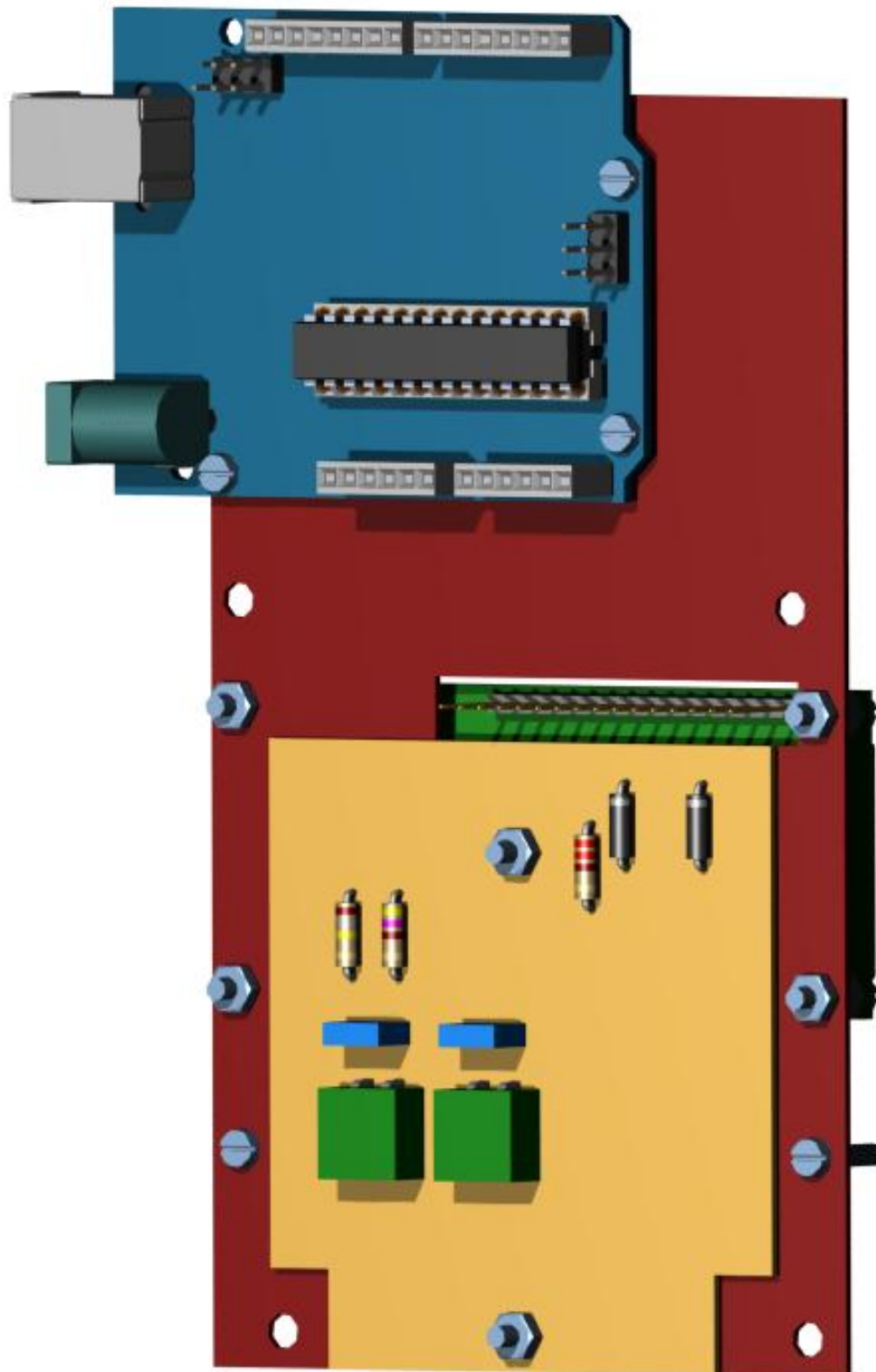


Слика 54. Причвршћивање међуплочице и пертинакс плоче,
поглед са задње стране

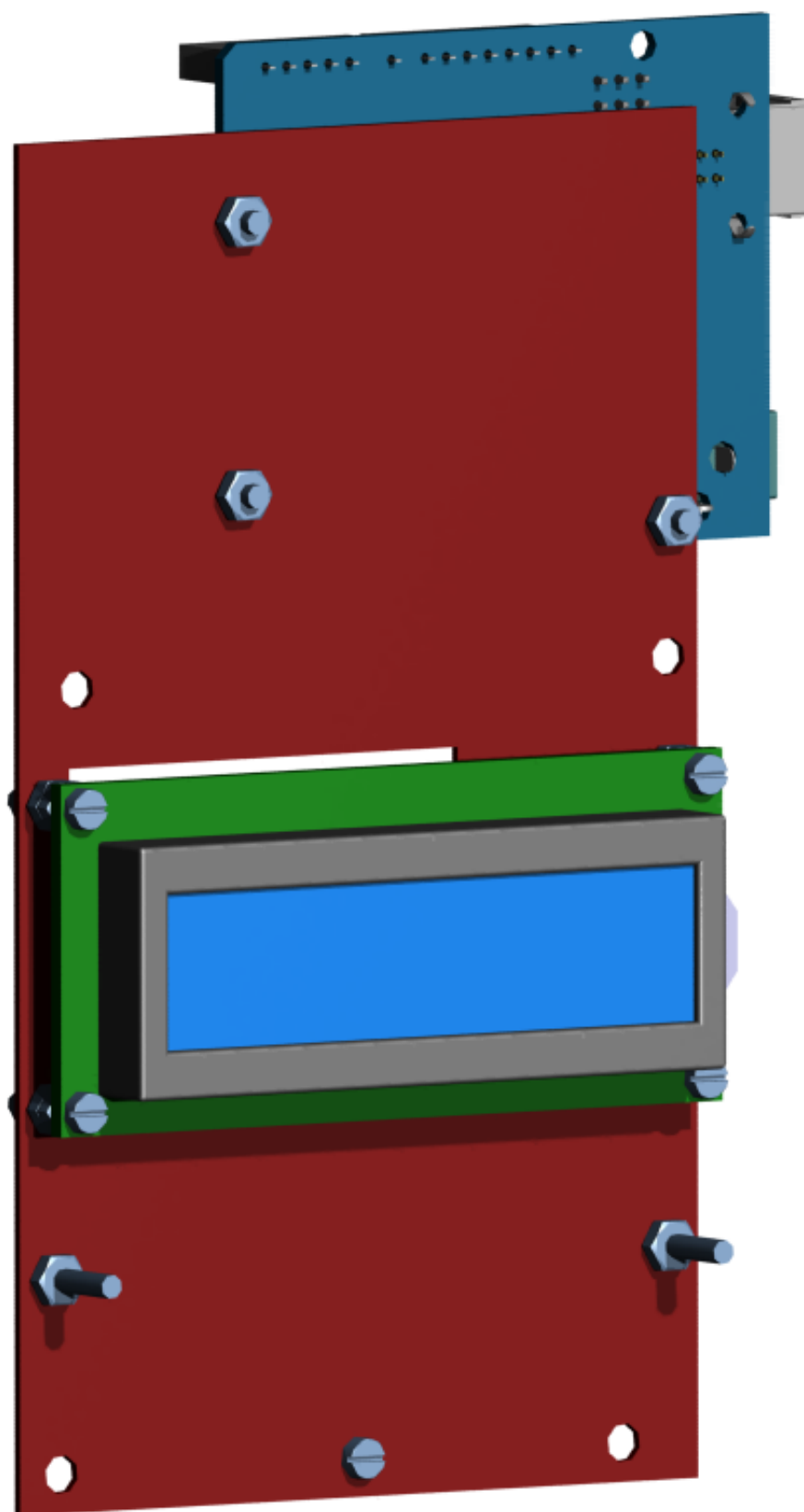


Слика 55. Причвршћивање међуплочице и пертинакс плоче,
поглед са предње стране

Након тога на пертинакс плочу причвршћују се дисплеј и Ардуино платформа што се може видети на сликама 56. и 57. Како би се убрзао процес моделирања склопа, модел Ардуино платформе преузет је са следеће веб адресе: <https://networkbot.wordpress.com/2011/12/13/arduino-cad-design/>.

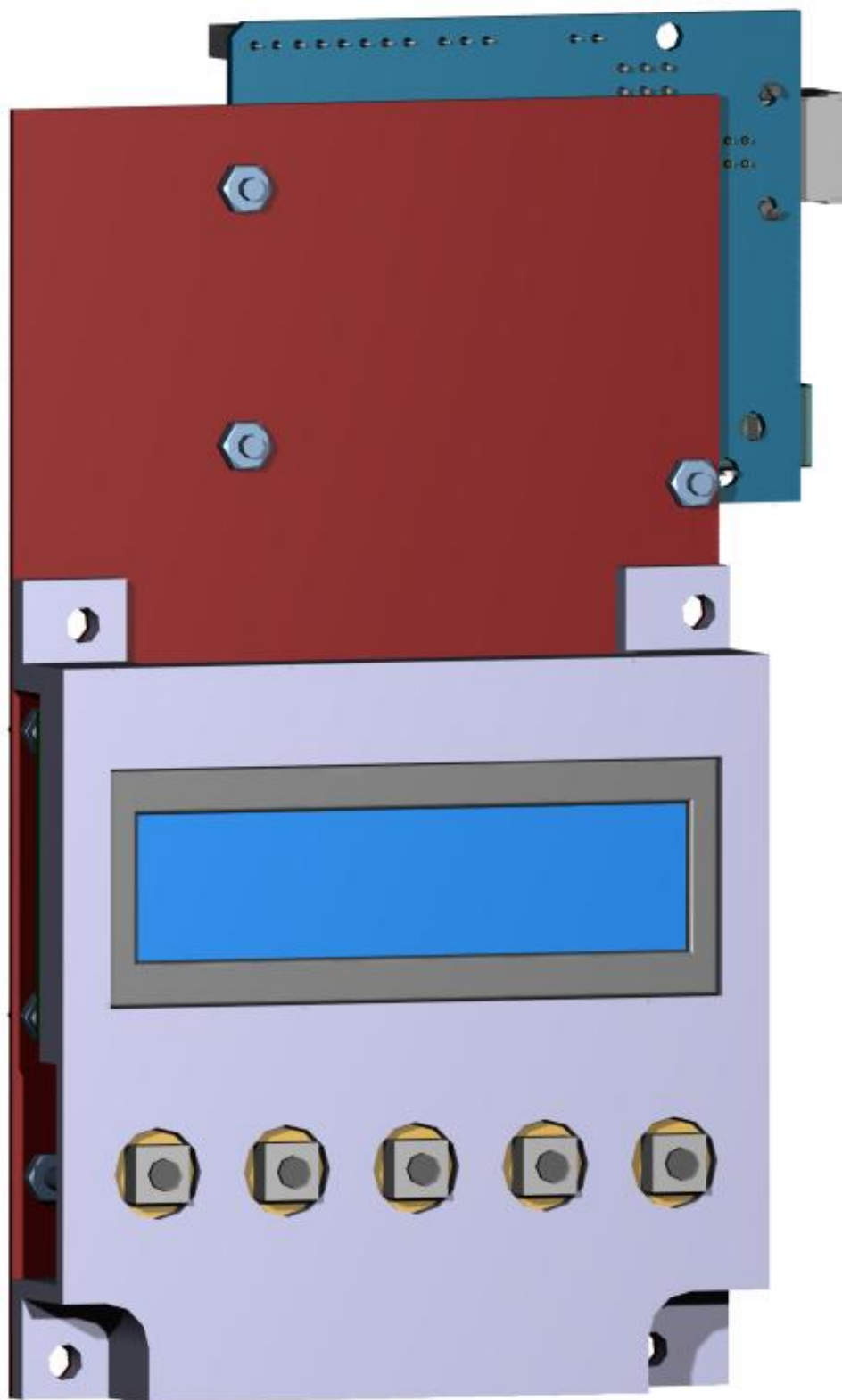


Слика 56. Склоп након монтирања дисплеја и Ардуино платформе,
поглед са задње стране

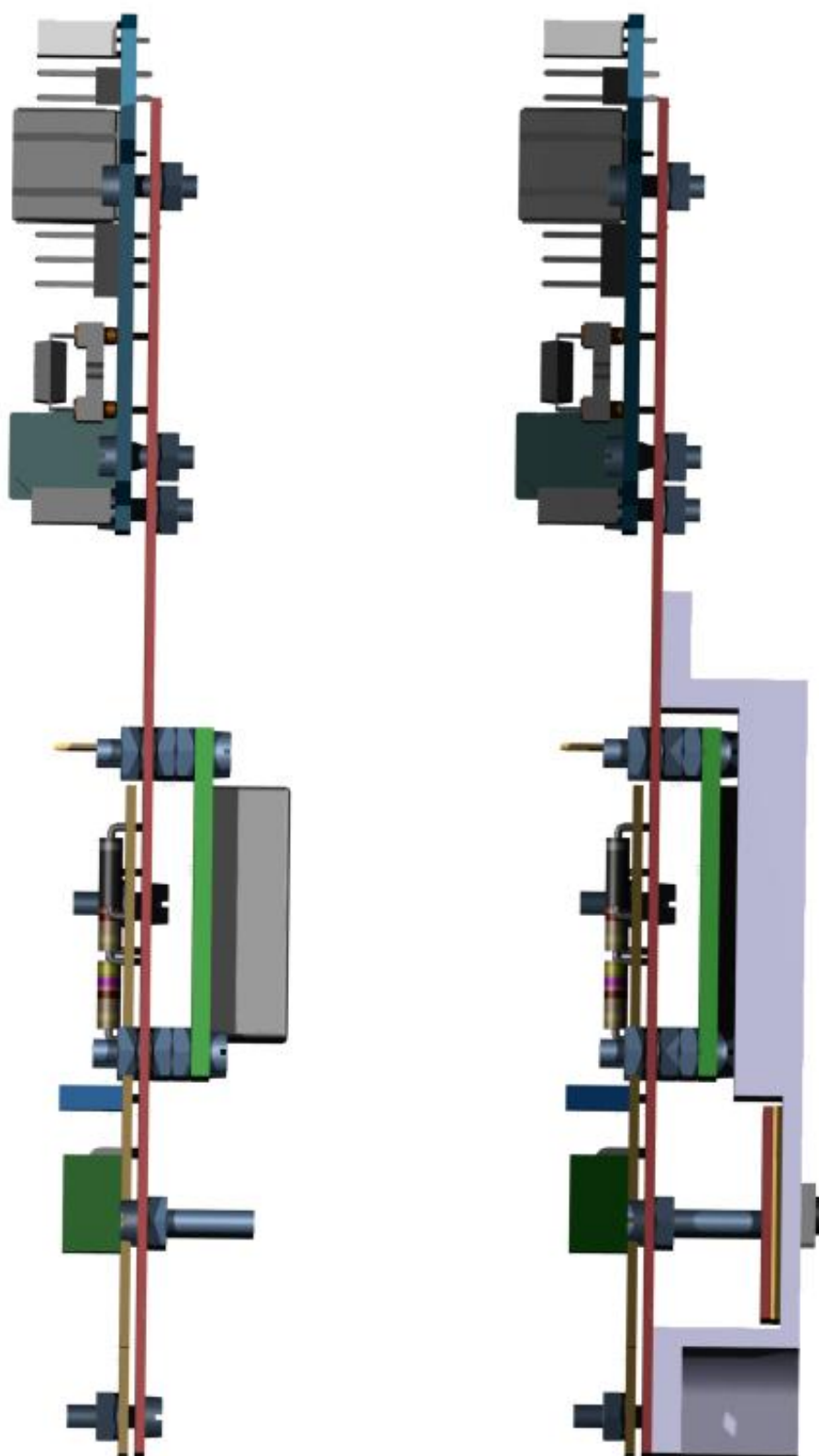


Слика 57. Склоп након монтирања дисплеја и Ардуино платформе,
поглед предње стране

Након тога поставља се рам на коме је претходно на свом месту позициониран панел са тастерима.



Слика 58. *Комплетан склоп, поглед са предње стране*

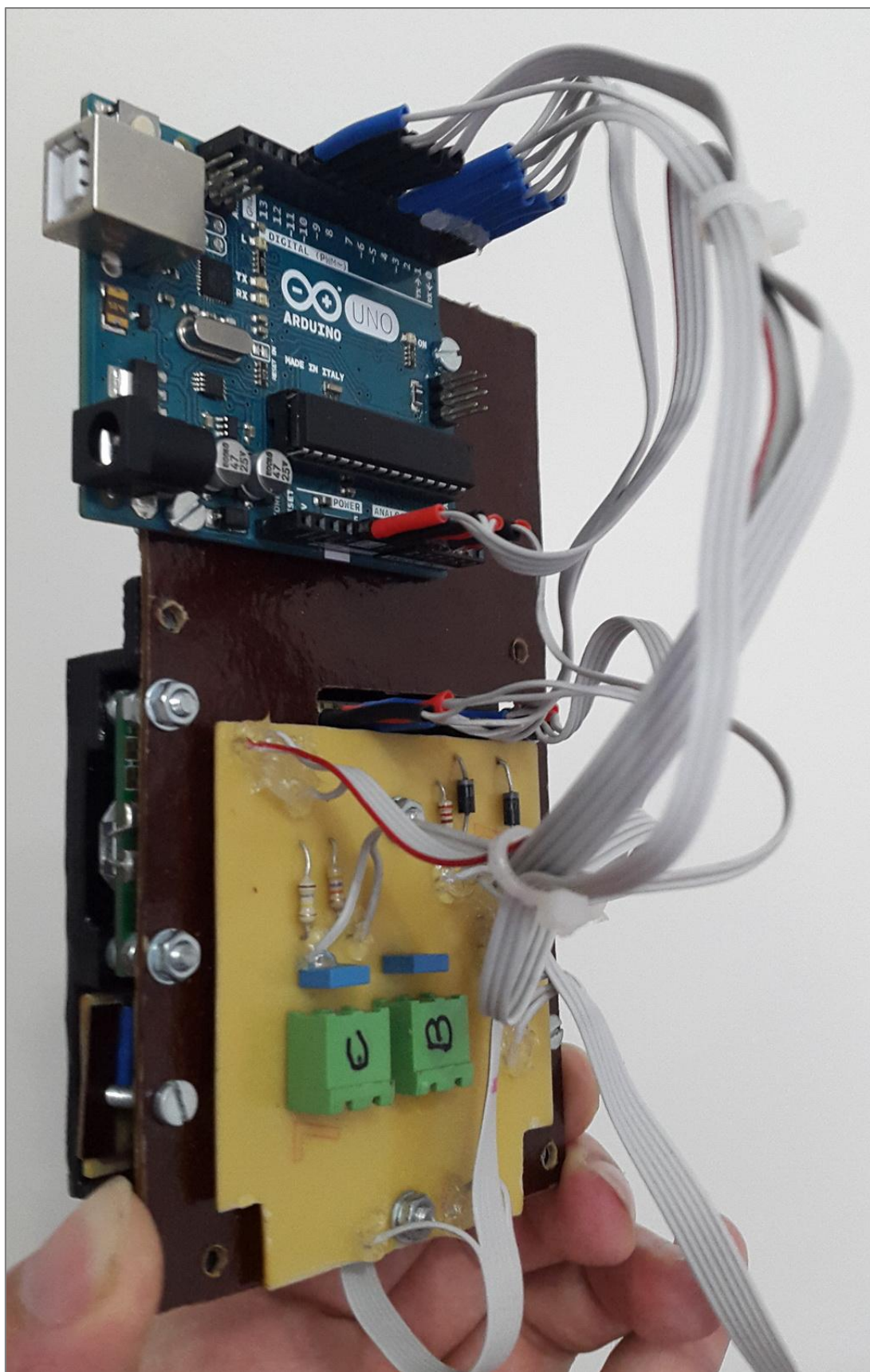


Слика 59. *Склоп пре монтирања рама и плочице са тастерима (лево) и
комплетан склоп (десно)*

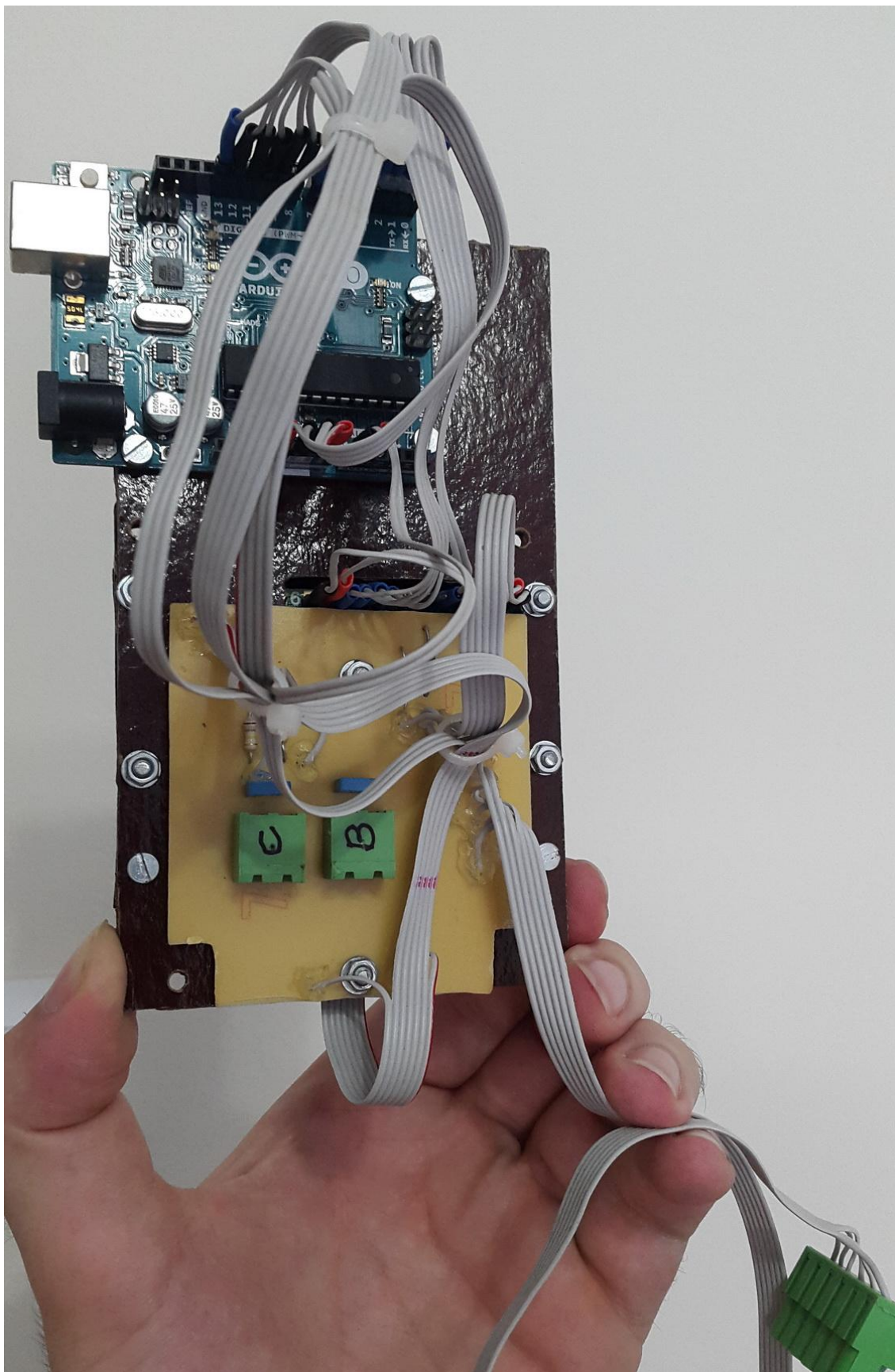
На сликама 60., 61. и 62. може се видети реализација овог склопа са свим потребним жицама.



Слика 60. Реализован склоп, поглед са предње стране



Слика 61. Реализован склоп, поглед са стране



Слика 62. Реализован склоп, поглед са задње стране

2.10. Модификовање предњег панела

Предњи панел претставља пластика на којој се описани склоп у прошлој тачки причвршћује. На овом панелу такође се налази налепница на којој су исписане функције свих тастера, као и неке додатне информације. Пошто је измењен дисплеј, као и положај тастера, било је потребно модификовати налепницу. Првобитни изглед овог панела може се видети на слици 63., док се на слици 64. може видети овај панел са прикаченим склопом.



Слика 63. Првобитни изглед предњег панела



Слика 64. Предњи панел са прикаченим склопом, поглед са предње стране

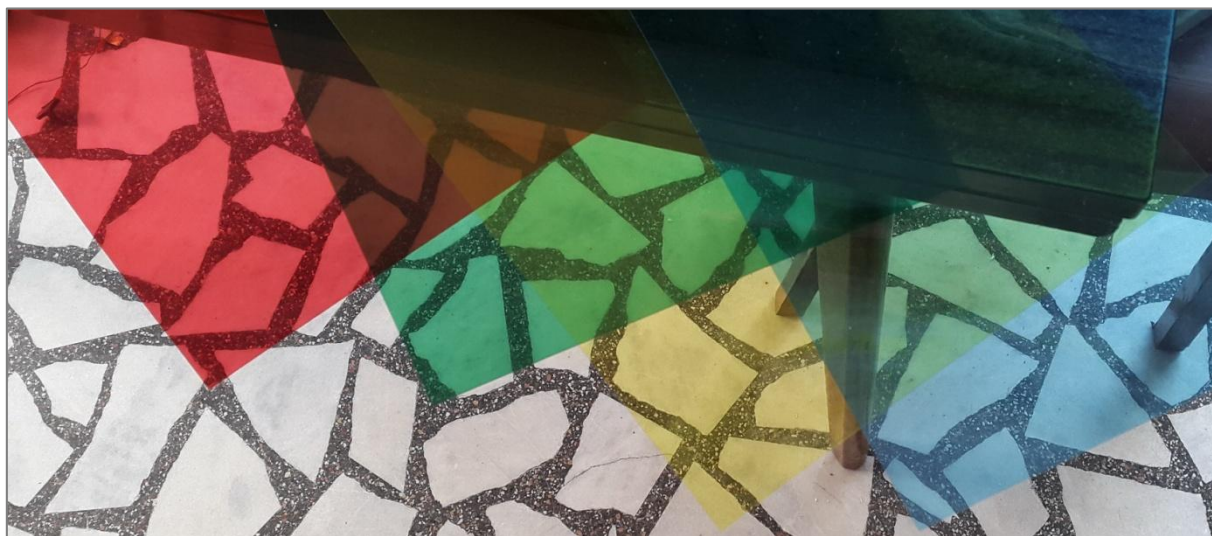


Слика 65. Предњи панел са прикаченим склопом, поглед са задње стране

Након тога било је потребно израдити одговарајућу налепницу. Она је израђена од провидне термо фолије (иста фолија која је коришћена за израду штампаних плочица) и фото папира (сјајан папир на коме се штампају фотографије). То је одрађено тако што је на фолији и на фото папиру отштампан изглед налепнице па су та два дела спојена лепком за папир, с тим што су претходно на местима где се налазе тастери на фото папиру залепљени мали делови обојених термо фолија које су такође провидне. Дизајн налепнице слика 66. одрађен је у софтверу „Catia R5“.



Слика 66. Дизајн налепнице



Слика 67. Провидне термо фолије у боји



Слика 68. Дизајн налепнице отштампан на провидној термо фолији



Слика 69. Израђена налепница



Слика 70. Налепница залепљена на предњи панел, поглед са задње стране



Слика 71. Налепница залепљена на предњи панел, поглед са предње стране



Слика 72. Изглед уређаја након свих хардверских промена

3. СОФТВЕР

3.1. Библиотеке, дефинисање пинова, тастера и константних вредности [3][17]

Пре почетка преограмирања потребно је имплементирати одговарајуће програмске библиотеке, дефинисати све пинове, тастере и константне вредности. Тастери су у овом пројекту дефинисани као „Bounce“ објекти јер је њих лакше користити, што ће касније бити објашњено. Програмски код у коме се дефинишу и имплементирају претходно поменуте компоненте и библиотеке приказан је испод:

```
//Implementiranje potrebnih biblioteka
#include <math.h>
#include<Bounce2.h>
#include <LiquidCrystal.h>

//Definisanje pinova
#define motor 6
#define grejac 7
#define dugmeStop 8
#define dugmeStart 9
#define dugmeIzbor 10
#define dugmeTajmerM 11
#define dugmeTajmerP 12
#define zvucnik 13

//Definisanje displeja
LiquidCrystal lcd(0,1,2,3,4,5);

//Pomoćne konstantne vrednosti radi lakšeg programiranja
const int ukljucen = HIGH;
const int iskljucen = LOW;

//Definisanje pinova za očitavanje temperature sa senzora
const int senzorBE = A0;
const int senzorCR = A1;

//Definisanje "Bounce" objekata za sve tastere
Bounce stop_b = Bounce();
Bounce start_b = Bounce();
Bounce izbor_b = Bounce();
Bounce tajmerM_b = Bounce();
Bounce tajmerP_b = Bounce();
```

3.2. Структура „PROGRAM“

Пошто је потребно да уређај може да има више програма, односно начина на који припрема храну, било је потребно креирати нови тип података који садржи све потребне информације за рад уређај. Заправо, ова структура садржи информације о процесу производње који се обавља, односно коју функцију (грејање, мешање или оба) обавља уређај у тренутној периоди. Затим ова структура садржи и информацију колико дуго, односно колико минута треба да се обавља тренутна периода. Уколико се обавља грејање (печење) потребно је знати на којој се температури обавља, па ова структура садржи још и вредности температура. Програмски код за ову структуру дат је испод:

```
//Konstantne vrednosti potrebne za kreiranje programa
const int brojPerioda = 10;
const int brojPrograma = 5;

//Struktura "PROGRAM"
struct PROGRAM{
    //Informacija o kojoj se radnji radi u zavisnosti od periode
    int radnja[brojPerioda];
    //Vreme trajanja perioda
    float vremeTrajanja[brojPerioda];
    //Temperatura za periode u kojima se vrši grejanje(pečenje)
    int temperatura[brojPerioda];

    //Metoda koja vraća ukupno vreme izabranog programa
    float ukupnoVreme(){
        float ukupno=0;
        for(int i=0;i<brojPerioda;i++){
            ukupno+=vremeTrajanja[i];
        }
        return ukupno;
    }
};
```

//Kreirani programi

PROGRAM programi[brojPrograma] = {

//Program 1 - Beli hleb (duži program)

{ {1,0,1,0,1,0,2,2} , {10,20,15,20,0.03,10,40,45} , {0,0,0,0,0,0,40,200} },

//Program 2 - Beli hleb (kraći program)

{ {1,2,2} , {10,35,45} , {0,40,200} },

//Program 3 - Španska pita

{ {1,0,2} , {5,5,60} , {0,0,200} },

//Program 4 - Džem od šljive

{ {4} , {70} , {200} },

//Program 5 - Puding i kačamak

{ {3,4} , {3,20} , {200,200} }

};

На први поглед делују компликовано, али после краћег објашњења закључује се да заправо и није. Пре свега потребно је напоменути да сви бројеви уписани између две витичасте заграде које су обојене црвеном бојом претстављају податке о једном програму. Бројеви осенчени плавом бојом претстављају функцију коју обавља уређај. Број 0 значи да је реч о паузи и тада уређај мирује, број 1 значи да је реч о мешању, број 2 да је реч о грејању (печењу), број 3 значи да се грејање и мешање обавља истовремено, док број 4 значи да је укључен грејач и да се симулира ручно мешање. Бројеви осенчени црвеном бојом претстављају временске интервале изражене у минутима. Сваки број означава временски интервал за једну периоду. Бројеви осенчени жутом бојом претстављају температуру на којој се обавља грејање (печење). Уколико се обавља само мешање или је пауза тада уређај не узима овај податак, али је битно да се унесе неки број, па се уноси број 0. Да би се лакше разумело како се обавља програм погледати табелу испод у којој је за пример узет програм 1 из претходно написаног програмског кода.

ПРОГРАМ 1								
Периода	1	2	3	4	5	6	7	8
Функција	1 (мешање)	0 (пауза)	1 (мешање)	0 (пауза)	1 (мешање)	0 (пауза)	2 (грејање)	2 (грејање)
Време трајања (min)	10	20	15	20	0.03 (1.8s)	10	40	45
T (°C)	0	0	0	0	0	0	40	200

У табели на претходној страни јасно је претстављен пример једног од програма. Може се приметити да програм има 8 од максималних 10 периода. Податке је потребно унети само за периоде које се користе, није потребно уношење за свих 10. Разлог томе јесте уколико се не унесу подаци за остале периоде уређај им сам додељује вредност 0, а он је програмиран тако да када наиђе на временски интервал који је једнак 0-ли сам прекида извршење програма, односно зна да је програм завршен. Тада стомира све функције (мешање и грејање) и ресетује, односно додељује почетне вредности свим параметрима који се користе приликом извршења програма. У датој табели се такође може приметити да је за првих шест периода постављена температура на 0°C. Без обзира и да је унета нека друга вредност температуре грејач се не би укључивао, јер је функција постављена на мешање, односно паузу. Пошто су вредности свих параметара смештени у низове бројева и којима се на основу индекса периоде приступа, врло је битно унети и вредности температуре за периоде у којима се не користи грејач као што је то урађено у табели на претходној страни. Да би програм функционисао по жељи корисника и да не би дошло до грешке, потребно је пратити једноставно правило које гласи: „Онолико функција колико садржи један програм, тај исти програм потребно је да садржи исто толико временских интервала, као и исто толико температурних вредности“.

3.3. Помоћне променљиве

Све помоћне променљиве дате су у коду испод:

```
//Помоћне променљиве
boolean ukljucenProgram = false;
boolean ukljucenTajmer = false;
int motorStanje = LOW;
int izabranPr = 0, prosliPr = 1;
int trenutnaPerioda = 0;

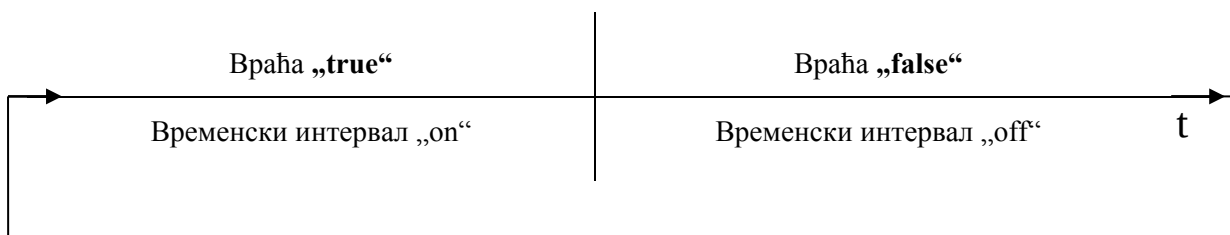
int temp = 0, proslaTemp = 0;
int sat = 0, prosliSat = 0, minut = 0, prosliMinut = 0;
int tajmer = 0;

unsigned long ukupnoVreme = 0;
unsigned long vremePocetkaPrograma = 0;
unsigned long vremePocetkaPeriode = 0;
unsigned long trenutnoVreme = 0;
unsigned long prosloVreme = 0;
unsigned long promenaStanja = 0;
long preostaloVreme = 0;
```

Када и где се користе све променљиве биће претстављено у даљем описивању софтвера.

3.4. Метода „периодично паљење“

Ова метода јесте логичка метода, то значи да она враћа логичку вредност („true“ или „false“). Приликом позивања ове методе потребно је унети временске интервале у милисекундама који претстављају колико дуго ова метода враћа вредност „true“, а колико „false“. Ова метода користи се код паљења грејача, јер је у неким условима потребно да његов рад буде периодичан. Да би се лакше разумео рад ове методе погледати слику 73.



Слика 73. Начин рада методе „периодично паљење“

Програмски код за ову методу дат је испод:

```
/*  
Metoda koja vraća vrednost "true" za vremenski interval "on",  
odnosno vrednost "false" za vremenski interval "off"  
*/  
boolean periodicnoPaljenje(long on, long off){  
  
    if(trenutnoVreme-prosloVreme>on+off){  
        prosloVreme = trenutnoVreme;  
        return true;}  
  
    else if(trenutnoVreme-prosloVreme>on){  
        return false;}  
  
    else{return true;}  
}
```

3.5. Методе за мерење температуре

Као што и сам назив каже ова метода користи се за очитавање температуре са температурних сензора. Програмски код за ову методу дат је испод:

```
void merenjeTemperature(){

    //Koeficijenti su abecednim redom (A,B,C)
    const double koeCR[3] = {0.0004357140911, 0.0001733009365, 0.0000004710945505};
    const double koeBE[3] = {0.001097204834, 0.00009308562133, 0.0000009997003632};

    /*Otpornosti izražene u omima koje su vezane na senzorima:
       Prva vrednost je za crveni(gornji) senzor, a druga za beli(donji)
    */
    const long otporConst[2] = {177600, 46500};

    //Promenljive
    float tempSenzora[2];
    int i;
    long otpor[2];
    float prosek[2];

    //Konstanta koja pokazuje koliko puta se vrši očitavanje sa senzora
    const int m=10;

    //Prvo se prosečnim vrednostima dodeljuje vrednost 0
    for(i=0;i<2;i++){
        prosek[i]=0;
    }

    /* Zatim se vrši očitavanje napona sa senzora onoliko puta za vrednost parametra "m"
       i dodaje promenljivoj "prosek". Ovo se radi radi dobijanja stabilnije vrednosti
    */
    for(i=0;i<m;i++){
        delay(2);
        prosek[0]+=analogRead(senzorCR);
        delay(2);
        prosek[1]+=analogRead(senzorBE);
    }
}
```

Наставак ове методе дат је на следећој страни.

```
/* Zatim se izračunavaju prosečne vrednosti i na osnovu njih izračunavaju trenutne
   otpornosti na senzorima
*/
for(i=0;i<2;i++){
    prosek[i]/=m;
    otpor[i]=otporConst[i]*prosek[i]/(1024.0-prosek[i]);
}

//Na osnovu otpornosti na senzorima izračunava se temperatura u °C koju oni mere
tempSenzora[0]=1/(koeCR[0]+koeCR[1]*log(otpor[0])+koeCR[2]*pow(log(otpor[0]),3))-273;
tempSenzora[1]=1/(koeBE[0]+koeBE[1]*log(otpor[1])+koeBE[2]*pow(log(otpor[1]),3))-273;

//Pošto postoje dva senzora, za konačnu vrednost temperature uzima se njihova prosečna
temp=(tempSenzora[0]+tempSenzora[1])/2;
}
```

Као што се у коду може видети прво се мере вредности напона на крајевим температурних сензора. Пошто је очитавање са сензора нестабилно, заправо разлике у очитавањима су неколико хиљадитих делова једног волта (1V) врше се неколико очитавања на сваке две милисекунде, односно у овом случају по 10 очитавања за сваки сензор. Свака очитана вредност додаје се оној променљивој „просек“ која одговара за сензор са кога се врши очитавање. Након тога у следећем кораку рачуна се просечна вредност збира свих очитавања за сваки сензор и на основу тих просечних вредности рачунају се тренутне отпорности оба сензора. Када се израчунају отпорности оба сензора, тада се преко „Штаинхарт-Хартове“ једначине која је објашњена у тачки 2.3. овог рада израчунавају температуре у келвинима (K) коју ови сензори мере. Као што се може видети на крајевима једначина у програмском коду додато је (-273), чега нема у „Штаинхарт-Хартовој“ једначини и то је урађено да би се вредност температуре добила у степене целзијусе (°C). На крају се за температуру у простору где се врши грејање (печење), узима просечна вредност збира температурних вредности са оба сензора.

3.6. Методе за контролу компоненти

3.6.1. Метода „грејање“

Ова метода за контролу компоненти јесте и најсложенија јер је одржавање температуре компликовано. Наиме, из разлога што у простору у коме се врши грејање/печење не постоји вентилатор који би узроковао циркулацију ваздуха у том простору, читавање сензора није довољно брзо. Једноставан алгоритам који би искључивао грејач када сензор очита вредност која је већа или једнака задатој, односно алгоритам који би укључивао грејач када сензор очита температуру мању од задате није довољно добар. Коришћењем оваквог алгоритма приликом експериментисања установило се да вредност температуре варира и до $\pm 30\%$ од вредности жељене. Из тог разлога користи се алгоритам који врши периодично паљење грејача помоћу методе „периодично паљење“ која је описана у тачки 3.4. Комплетан програмски код за паљење грејача дат је испод:

```
void grejanje(int temperatura){

    int koeA=0,koeB,koeC=0,razlika=0;
    long koeD=0;
    float procenat=0.0;

    //Zadavanje procenta i koeficijenta B u zavisnosti od zadate temperature
    if(temperatura<80){procenat=0.65;}
    else if(temperatura<150){procenat=0.85;}
    else{procenat=0.95;}

    if(temperatura<60){koeB=10000;} else{koeB=5000;}

    //Zadavanje ostalih koeficijenata u zavisnosti od zadate temperature
    if(temperatura<60){
        koeA=koeC=1000;
        koeD=60000;
    }
    else if(temperatura<=80){
        koeA=koeC=1500;
        koeD=30000;
    }
    else if(temperatura<=100){
        razlika=temperatura-80;
        koeA=koeC=1500+razlika*50;
        koeD=30000;
    }
}
```

Наставак кода налази се на следећој страни.

```
else if(temperatura<=120){
    razlika=temperatura-100;
    koeA=koeC=2500+razlika*50;
    koeD=30000-razlika*500;
}
else if(temperatura<=140){
    razlika=temperatura-120;
    koeA=3500+razlika*100;
    koeC=3500;
    koeD=20000-razlika*500;
}
else if(temperatura<=160){
    razlika=temperatura-140;
    koeA=5500;
    koeC=3500;
    koeD=10000-razlika*100;
}
else if(temperatura<=180){
    razlika=temperatura-160;
    koeA=5500+razlika*100;
    koeC=3500;
    koeD=8000-razlika*150;
}
else if(temperatura<=200){
    razlika=temperatura-180;
    koeA=7500+razlika*50;
    koeC=3500+razlika*50;
    koeD=5000;
}

//Paljenje grejača
if(temp<temperatura*procenat){digitalWrite(grejac,ucljucen);}
else if(temp<temperatura){
    if(periodicnoPaljenje(koeA,koeB)){digitalWrite(grejac,ucljucen);}
    else{digitalWrite(grejac,iskljucen);}
}
else if(temp>temperatura*1.05){digitalWrite(grejac,iskljucen);}
else{
    if(periodicnoPaljenje(koeC,koeD)){digitalWrite(grejac,ucljucen);}
    else{digitalWrite(grejac,iskljucen);}
}
}
```

Као што се може и видети у коду постоје коефицијенти чије су вредности изражене у милисекундама. Ови коефицијенти користе се за периодично паљење грејача и они зависе у односу на задату температуру. Сви коефицијенти добијени су експериментисањем. Након израчунавања свих коефицијената врши се укључивање грејача. Прво се проверава да ли је тренутна вредност температуре мања од одговарајуће процентуалне вредности жељене температуре, и то уколико је жељена температура мања од 80°C процентуална вредност је 65%, уколико је жељена температура мања од 150°C процентуална је 85%, док је за температуре веће од 150°C процентуална 95%. Уколико је тај услов испуњен грејач је укључен без периодичног паљења, односно не врши се његово искључивање. Уколико није испуњен проверава се да ли тренутна температура мања од жељене температуре, уколико јесте врши се периодично укључивање грејача на основу коефицијената „koeA“ и „koeB“. Када вредност тренутне температуре премашује вредност жељене тада се проверава да ли је премашила жељену за 105% њене вредности. Уколико јесте грејач се искључује и тада се чека да температура опадне, уколико то није случај тада се врши периодично паљење грејача на основу коефицијената „koeC“ и „koeD“. Коришћењем овакве методе знатно се смањило одступање од жељене вредности температуре и оно износи око $\pm 10\%$ од жељене вредности, што значи да се одступање троструко смањило.

3.6.2. Метода „мешање“

Сам назив методе говори да је реч о укључивању електромотора који врши мешање у посуди. Да би се поједноставило писање програма направљена је ова метода која врши укључивање електромотора. Програмски код ове методе може да се види испод:

```
void mesanje(int stanje){  
    digitalWrite(motor,uključen);  
    motorStanje=stanje;  
}
```

3.6.3. Метода „свеИскључено“

Потребно је краирати и методу која врши искључивање свих компоненти, односно грејача и електромотора јер постоје програми који у одређеном временском интервалу праве паузу. Ова метода такође се позива након завршетка програма како би се угасиле све компоненте. Програмски код за ову методу може се видети испод:

```
void sveIskljuceno(){  
    digitalWrite(grejac,isključen);  
    mesanje(isključen);  
}
```

3.7. Метода „одбројавање“

Ова метода врши рачунање преосталог времена до завршетка програма. Након тога, пошто микроконтролер време рачуна у милисекундама ова метода то време претвара у преостале сате и минуте, како би кориснику преостало време било разумљивије. Програмски код за ову методу може се видети испод:

```
void odbrojavanje(){
    //Pretvaranje minuta u milisekunde
    ukupnoVreme=60000*(programi[izabranPr].ukupnoVreme()+tajmer);

    //Računanje preostalog vremena u zavisnosti da li je uključen program ili tajmer
    if(uključenProgram || uključenTajmer){
        preostaloVreme=ukupnoVreme-(trenutnoVreme-vremePocetkaPrograma);
    }
    else{
        preostaloVreme=ukupnoVreme;
    }

    //Pretvaranje preostalog vremena u minute
    int vr=preostaloVreme/60000;
    sat=minut=0;

    //Računanje ukupnog broja sati i minuta
    while(vr>=60){
        vr-=60;
        sat++;
    }
    minut=vr;
}
```

Као што се и може видети у горе прогамском коду, прво се врши претварање минута у милисекунде, односно пошто минут има 60000 милисекунди све се множи тим бројем. Затим се проверава да ли је укључен програм или тајмер како би се рачунало преостало време, уколико то није испуњено преостало време заправо јесте укупно време. Затим се опет врши претварање, с тим што се милисекунде претварају у минуте, па се сада врши дељење. Онда се променљиве које мере сате и минуте постављају на нулу. Након тога рачунају се сати и минути преостали до завршетка програма који се уједно и приказују на дисплеју.

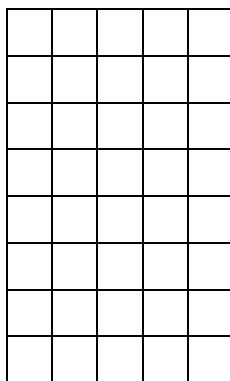
3.8. Метода „штампање“

Ова метода врши штампање свих потребних информација на дисплеј уређаја. Ова метода је врло једноставна и све функције су прилично једноставне. Програмски код за ову методу може се видети испод:

```
void stampanje(){
    if(izabranPr!=prosliPr){
        lcd.setCursor(9,0);
        lcd.print(" ");
        lcd.setCursor(9,0);
        /* Štampanje indeksa izabranog programa na koji se dodaje vrednost 1,
         * iz razloga da bi brojanje programa počelo od broja 1
         */
        lcd.print(izabranPr+1);
        prosliPr=izabranPr;
    }
    lcd.setCursor(15,0);
    //Štampanje "play", odnosno "stop" simbola u zavisnosti da li je uključen program ili tajmer
    if(uključenProgram || uključenTajmer){
        lcd.write(3);
    }
    else{
        lcd.write(2);
    }
    if(temp!=proslaTemp){
        lcd.setCursor(2,1);
        lcd.print(" ");
        lcd.setCursor(2,1);
        lcd.print(temp);
        lcd.write(1);
        lcd.print("C");
        proslaTemp=temp;
    }
    if(sat!=prosliSat || minut!=prosliMinut){
        lcd.setCursor(11,1);
        lcd.print(" ");
        lcd.setCursor(11,1);
        lcd.print(sat);
        lcd.print(":");
        if(minut<10){lcd.print(0);}
        lcd.print(minut);
        prosliSat=sat;
        prosliMinut=minut;
    }
}
```

3.9. Специјални симболи [18]

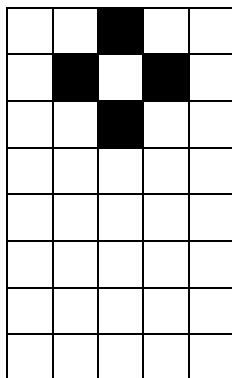
Приликом коришћења дисплеја потребно је употреби неке одређене симболе којих нема у Ардуиновим библиотекама. Тада се креирају посебни симболи по жељи корисника. Наиме, сваки карактер на дисплеју приказан је одговарајућим бројем пиксела који могу максимално бити димензија 5x8. На слици 74. приказан је број пиксела за један карактер.



Слика 74. Број пиксела по карактеру

Било је потребно креирати симбол за означавање степена (°), као и традиционалне симболе за „play“ (►) и „stop“ (■) функцију који означавају да ли је програм покренут или није. Програмски кодови за креирање ових симбола дати су испод:

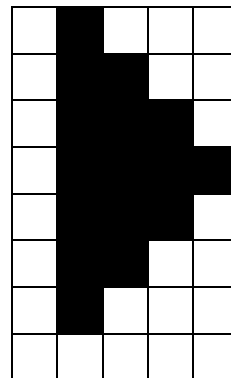
```
byte stepenSimbol[8] = {
    0b00100,
    0b01010,
    0b00100,
    0b00000,
    0b00000,
    0b00000,
    0b00000,
    0b00000
};
```



```
byte stopSimbol[8] = {
    0b00000,
    0b11111,
    0b11111,
    0b11111,
    0b11111,
    0b11111,
    0b00000,
    0b00000
};
```



```
byte playSimbol[8] = {
    0b01000,
    0b01100,
    0b01110,
    0b01111,
    0b01110,
    0b01100,
    0b01000,
    0b00000
};
```



Слика 75. Приказ креираних симбола на основу изнад датих програмских кодова

Као што се може видети у програмском коду на претходној страни и на слици 75. пиксели којима је додељена вредност „1“ они светле, а којима је додељена вредност „0“ они не светле. На сликама 76. и 77. може се видети како ови симболи изгледају на дисплеју уређаја.



Слика 76. Дисплеј уређаја, прва слика



Слика 77. Дисплеј уређаја, друга слика

3.10. Метода „ресет“

Након завршетка програма или уколико је програм стопиран, потребно је ресетовати, односно доделити почетне вредности променљивема које се користе. То је потребно учинити из разлога уколико корисник жели да након завршетка неког програма поново користи уређај без његовог искључивања. Програмски код за ову методу може се видети испод:

```
void reset(){
    trenutnaPerioda=0;
    prosloVreme=0;
    ukljucenProgram=false;
    ukljucenTajmer=false;
    tajmer=0;
    sveIskljuceno();
}
```

3.11. Метода „звучно упозорење“

Пошто је убачен пиезо елемент потребно је креирати звучна упозорења, која одговарајућим звучним сигнаlima обавештавају корисника шта се дешава са уређајем. Ова метода заправо садржи два типа обавештења. Приликом позивања ове методе, уноси се одговарајући број који означава који тип упозорења је потребно користити. Први тип упозорења претставља звучни сигнал који наглашава да је програм завршен и он је претстављен са више узастопних бипова (eng. „beep“). Други тип звучног упозорења сачињен је од једног бипа и он означава почетак програма, али и промену периода. Програмски код за ову методу претстављен је испод:

```
void zvucnoUpozorenje(int tip){
    if(tip==1){
        for(int i=0;i<10;i++){
            delay(350);
            tone(zvucnik,500);
            delay(350);
            noTone(zvucnik);
        }
    }
    else if(tip==2){
        tone(zvucnik,500);
        delay(350);
        noTone(zvucnik);
    }
}
```

3.12. Функција „setup“

У овој функцији као што је објашњено у тачки 1.2.3. наводе се операције које се извршавају само једном и то приликом покретања Ардуино платформе. У овој функцији обављају се и сва подешавања потребна за правилан рад електронике. Програмски код ове функције дат је испод:

```
void setup() {  
    //Kreiranje specijalnih simbola i definisanje displeja  
    lcd.createChar(1,stepenSimbol);  
    lcd.createChar(2,stopSimbol);  
    lcd.createChar(3,playSimbol);  
    lcd.begin(16,2);  
  
    /* Na ekranu se vrsi štampanje informacija o fakultetu, smeru studija,  
       naziva teme završnog rada, podaci o studentu kao i podaci o mentoru */  
    lcd.print("FIN");  
    lcd.setCursor(0,1);  
    lcd.print("Kragujevac");  
    delay(3000);  
    lcd.clear();  
    lcd.print("OAS - Masinsko");  
    lcd.setCursor(0,1);  
    lcd.print("inzenjerstvo");  
    delay(3000);  
    lcd.clear();  
    lcd.print("Zavrsni rad");  
    lcd.setCursor(0,1);  
    lcd.print("Arduino pekara");  
    delay(3000);  
    lcd.clear();  
    lcd.print("Aleksandar");  
    lcd.setCursor(0,1);  
    lcd.print("Sindjelic");  
    delay(3000);  
    lcd.clear();  
    lcd.print("Br. indeksa:");  
    lcd.setCursor(0,1);  
    lcd.print("73/2012");  
    delay(3000);  
    lcd.clear();  
    lcd.print("Mentor: dr.Nenad");  
    lcd.setCursor(0,1);  
    lcd.print("Grujovic");  
    delay(3000);  
    lcd.clear();  
}
```

Наставак кода дат је на слећој страни.

```
//Na ekranu se štampaju potrebne informacije
lcd.print("Program:");
lcd.setCursor(0,1);
lcd.print("T:");
lcd.setCursor(9,1);
lcd.print("V:");

//Definisanje funkcije pinova
pinMode(dugmeStop,INPUT_PULLUP);
pinMode(dugmeStart,INPUT_PULLUP);
pinMode(dugmeIzbor,INPUT_PULLUP);
pinMode(dugmeTajmerM,INPUT_PULLUP);
pinMode(dugmeTajmerP,INPUT_PULLUP);
pinMode(grejac, OUTPUT);
pinMode(motor,OUTPUT);

//Dodeljivanje tastera odgovarajućim "bounce" elementima
stop_b.attach(dugmeStop);
stop_b.interval(5);

start_b.attach(dugmeStart);
start_b.interval(5);

izbor_b.attach(dugmeIzbor);
izbor_b.interval(5);

tajmerM_b.attach(dugmeTajmerM);
tajmerM_b.interval(5);

tajmerP_b.attach(dugmeTajmerP);
tajmerP_b.interval(5);

//Početno stanje motora, grejača i zvučnika
digitalWrite(grejac,iskljucen);
digitalWrite(motor,iskljucen);
noTone(zvucnik);
}
```

У овој функцији не треба ништа претерано објашњавати осим дефинисања тастера преко „bounce“ објеката. Као што се може видети у дефинисању пинова за тастере узима се „INPUT_PULLUP“, а не само „INPUT“. Такође се приликом дефинисања „bounce“ објеката за интервал узима 5ms. То значи да када се на тастеру промени стање (притисне или пусти) да се након тог интервала то стање прочита. То се ради из разлога што приликом промене стања тастера може доћи до пулсирајућег напона који је врло кратак (до 3ms), па може доћи до погрешног читавања. Из тог разлога за интервал је узета вредност од 5ms.

3.13. Функција „loop“

Ова функција заправо јесте главни програм где се дешава цео рад овог уређаја. Да би се разумео цео рад уређаја најпре је потребно погледати програмски код ове функције који је дат испод:

```
void loop() {

    /*Promenljivoj "trenutnoVreme" dodeljuje se vrednost koju mikrokontroler
    broji od svog uključivanja */
    trenutnoVreme=millis();
    noTone(zvucnik);

    //Očitavanje stanja sa svih tastera
    stop_b.update();
    start_b.update();
    izbor_b.update();
    tajmerM_b.update();
    tajmerP_b.update();

    /*Ukoliko je uključen tajmer ili program nije moguće koristiti nijedan
    taster osim tastera "stop" */
    if(!(ukljucenProgram || ukljucenTajmer)){
        //Pritiskom na taster "izbor"
        if(izbor_b.fell()){
            izabranPr++;
            tajmer=0;
            if(izabranPr>=brojPrograma){
                izabranPr=0;
            }
        }
        //Pritiskom na taster za povećavanje tajmera
        if(tajmerP_b.fell()){
            tajmer+=10;
            if(tajmer>720){
                tajmer=0;
            }
        }
        //Pritiskom na taster za smenjenje tajmera
        if(tajmerM_b.fell()){
            tajmer-=10;
            if(tajmer<0){
                tajmer=0;
            }
        }
    }
}
```

Наставак програмског кода дат је на следећој страни.

```
//Pritiskom na taster "start"
if(start_b.fell()){
    ukljucenTajmer=true;
    vremePocetkaPrograma=trenutnoVreme;
    zvucnoUpozorenje(2);
}
}

//Pritiskom na taster "stop"
if(stop_b.fell()){
    reset();
}

//Ukoliko je uključen tajmer
if(ukljucenTajmer){
    sveIskljuceno();
    if(60000*tajmer<=(trenutnoVreme-vremePocetkaPrograma)){
        ukljucenProgram=true;
        ukljucenTajmer=false;
        vremePocetkaPeriode=trenutnoVreme;
        if(tajmer!=0){
            zvucnoUpozorenje(2);
        }
    }
}

//Ukoliko je uključen program
else if(ukljucenProgram){

    //Proverava se da li se prelazi na sledeću periodu na osnovu proteklog vremena
    if(60000*programi[izabranPr].vremeTrajanja[trenutnaPerioda]<=(trenutnoVreme-vremePocetkaPeriode)){
        trenutnaPerioda++;
        vremePocetkaPeriode=trenutnoVreme;
        zvucnoUpozorenje(2);
    }

    /*Gašenje programa ukoliko se dođe do maksimalnog broja perioda ili ukoliko je
    vreme trajanja periode jednako 0-li
    */
    if(trenutnaPerioda>=brojPerioda || programi[izabranPr].vremeTrajanja[trenutnaPerioda]==0){
        reset();
        zvucnoUpozorenje(1);
    }
}
```

Наставак програмског кода да је на следећој страни.

```
//Ukoliko je funkcija broj 0, tada je pauza, odnosno svi elementi su isključeni
else if(programi[izabranPr].radnja[trenutnaPerioda]==0){
    sveIskljuceno();
}

//Ukoliko je funkcija broj 1, tada se vrši samo mešanje
else if(programi[izabranPr].radnja[trenutnaPerioda]==1){
    mesanje();
    digitalWrite(grejac,iskljucen);
}

//Ukoliko je funkcija broj 2, tada se vrši samo grejanje/pečenje
else if(programi[izabranPr].radnja[trenutnaPerioda]==2){
    grejanje(programi[izabranPr].temperatura[trenutnaPerioda]);
    digitalWrite(motor,iskljucen);
}

//Ukoliko je funkcija broj 3, tada se vrši i mešanje i grejanje/pečenje
else if(programi[izabranPr].radnja[trenutnaPerioda]==3){
    mesanje();
    grejanje(programi[izabranPr].temperatura[trenutnaPerioda]);
}

//Ukoliko je funkcija broj 4, tada se vrši grejanje/pečenje sa simulaciom ručnog mešanja

else if(programi[izabranPr].radnja[trenutnaPerioda]==4){
    grejanje(programi[izabranPr].temperatura[trenutnaPerioda]);

    //Ne meša 6 sekundi
    if(trenutnoVreme>(promenaStanja+6000) && motorStanje==iskljucen){
        mesanje(ukljucen);
        promenaStanja=trenutnoVreme;
    }

    //Meša 1,5 sekundu
    else if(trenutnoVreme>(promenaStanja+1500) && motorStanje==ukljucen){
        mesanje(iskljucen);
        promenaStanja=trenutnoVreme;
    }
}

//Pozivanje svih potrebnih metoda
merenjeTemperature();
odbrojavanje();
stampanje();
delay(10);
}
```

Као што се и може видети на почетку програмског кода на страни 75. први корак јесте да се променљивој „trenutnoVreme“ додели тренутна вредност до које је стигао бројач микроконтролера. Тај бројач почиње да броји приликом укључења микроконтролера и бројање врши у микросекундама. Након тога врши се читавање са тастера, односно проверава се да ли је неки од њих променио стање. Затим се извршавају функције тастера, с тим што уколико је покренут програм или тајмер њихове функције се не извршавају. Једино функција тастера „стоп“ може да се изврши када је покренут програм или тајмер. Притиском на тастер „старт“ покреће се рад уређаја. Без обзира да ли је тајмер подешен или није, први пут приликом покретања програма пролази се кроз функцију тајмера. Када протекне време на које је постављен тајмер или уколико он није постављен прелази се на почетак извршења програма. Прва провера јесте да ли је протекло подешено време за тренутну периоду, уколико јесте прелази се на следећу, а уколико није остаје се на исту периоду. Након тога проверава се да ли је индекс тренутне периоде већи од максималног броја периода по програму, уколико јесте или уколико је време трајања периоде једнако нули, програм се гаси и стопа се уређај. Као што је раније поменуто у овом раду, уколико неки програм садржи мањи број периода од максималног, за преостале могуће периоде време трајања се аутоматски поставља на нулу, из тог разлога је уређај програмиран да заврши програм када наиђе на периоду чије је временско трајање једнако нули. Нпр. креира се програм који садржи четири периоде, пошто је максимални број периода десет за остале шест периоде сви параметри се аутоматски постављају на нулу. Приликом извршења програма како уређај не би бројао до максималног броја периода и емитовао звуке упозорења за промену периода, он је програмиран да заврши програм када наиђе на прву периоду чији је временски интервал једнак нули. Након тога преко индекса периоде приступа се одговарајућој функцији коју уређај треба да обавља приликом трајања те периоде, односно броју који означава о којој функцији је реч. На основу тог броја обавља се одређена функција. Број „0“ претставља паузу и тада уређај мирује, број „1“ претставља мешање, број „2“ претставља грејање/печење, број „3“ комбинацију претходне две функције (мешање и грејање/печење) и број „4“ претставља грејање па симулацијом ручног мешања. Након тога позивају се све методе потребне за рад уређаја, односно метода за мерење температуре, метода која врши одбројавање времена као и метода која врши штампање свих информација на дисплеј.

4. УКУПНИ ТРОШКОВИ

Компонента	Комада	Цена по комаду (RSD)
Мини пекара	1	2700,00
Ардуино УНО	1	2990,00
Дисплеј 16x2	1	1290,00
Компоненте за плочицу напајања и прекидача	1	1316,00
Компоненте за међуплочицу	1	30,00
Компоненте за плочицу са тастерима	1	60,00
Пертинакс једностран 100x160	2	120,00
УКУПНО:		8626,00

5. РЕЦЕПТИ

5.1. Бели хлеб (900г)

Програм: 2	
Састојци:	Количина:
Бело брашно	560г
Вода	300мл
Млеко	100мл
Зејтин	2 кашике
Квасац	22г
Со	1,5 кашичица
Шећер	2 кашичице



5.2. Шпанска пита (480г)

Програм: 3	
Састојци:	Количина:
Бело брашно	170г
Со	0,5 кашичице
Прашак за пециво	0,5 кесице
Јогурт	0,5 чаше
Шунка*	50г
Кисео краставац*	50г
Зејтин	0,5 чаше
Јаја	2 комада



* - значи ситно исецкано.

5.3. Цем од шљиве

Програм: 4	
Састојци:	Количина:
Шљиве (очишћене)	500г
Шећер	250г



5.4. Качамак

Програм: 5	
Састојци:	Количина:
Вода	500мл
Кукурузно брашно	130г
Со	1 кашичица



5.5. Пудинг

Програм: 5	
Састојци:	Количина:
Пудинг	2 кесе
Млеко	800мл
Шећер	6 кашике



6. ЗАКЉУЧАК

Полазну тачку овог завршног рада претстављао је уређај који је једино могао да врши припрему хлеба. Циљ овог рада јесте модификација уређаја и проширење његове примене, односно да се прошири асортиман хране која се припрема. Да би се то одрадило било је потребно научити неке од основних закона електротехнике, упознати се са електронским компонентама и научити принцип њиховог рада. Такође је било потребно упознати се са техничким карактеристикама свих примењених компоненти, како би се пројектовао систем који ће функционисати без њиховог отказивања.

Највећа предност овог уређаја у односу на остале фабричке уређаје је та што се на њему, помоћу рачунара могу убацивати нови програми за припрему хранљивих производа. На тај начин корисник може поред хлеба припремити и неке друге хранљиве производе. Одрађене модификације на овом уређају допринеле су томе да овај уређај поред аутоматске припреме хлеба, за шта је претходно и био намењен, врши аутоматску припрему још неких јела. Такође је додата и могућност праћења температуре у простору у коме се врши печење/кување и приказивање вредности те температуре на дисплеју уређаја. Све ове модификације говоре да је уређај унапређен.

Данас људи живе брзим животом, имају доста обавеза и понекад немају времена да себи припреме оброк којим би започели дан. Овај уређај је идеалан за такве ситуације јер не захтева присуство корисника, а коришћењем тајмера корисник може да подеси када жели да његова храна буде припремљена. Једина потреба корисника јесте да убаци састојке и изабере одговарајући програм, што не захтева више од пет минута. На тај начин корисник може сваког јутра имати свеже припремљен своји омиљени оброк. Са друге стране овај уређај могу користити и кувари како би убрзали свој рад у кухињи, тако што би уређај припремао једну врсту хране док они то време користе за припрему неке друге.

Пошто је уређај потпуно функционалан и унапређен, закључује се да Ардуино платформа има широку могућност примене и да се може користити за контролу неког другог кућног уређаја али и да се може примењивати и у индустрији. Она такође претставља веома једноставну платформу за едукацију студената који немају предзнање, а имају жељу да се баве програмирањем микроконтролера.

Рад на овом пројекту доноси велико задовољство јер је задатак био да се креира реалан систем који може бити од користи сваком човеку. Такође је стечено доста знања у области програмирања и електронике што може бити од велике користи у будућој каријери машинског инжењера.

Надам се да је овај рад обезбедио довољно предзнање за колеге који желе да се баве у овој области машинског инжењерства, али и да им је показатељ да се уз велики труд и напор сви циљеви могу постићи.

7. ЛИТЕРАТУРА

- [1] Хенсен А.: Програмирање на језику С, Микрокњига, Београд, 1991.
- [2] Милићев Д.: Објектно оријентисано програмирање на језику С++, Микрокњига, Београд, 1991.
- [3] Massimo Banzi „Getting Started with Arduino, 2nd Edition“, 2011.
- [4] Gordon McComb, Earl Boysen, Електроника за неупућене, Микрокњига, 2007.
- [5] Earl Boysen, Nancy Muir, Електронски пројекти за неупућене, Микрокњига, 2010.
- [6] Јелена Бубало, Радослав Илић, Видојко Вељковић, Оптиелектроника, 1999.
- [7] <https://sr.wikipedia.org/wiki/Arduino> (8. август 2015),
- [8] <http://www.elektronika-projekti.com/kupi-online-elektroniku/category/18-arduino-uno-rev3.html> (8. август 2015),
- [9] <https://www.arduino.cc/en/Main/arduinoBoardUno> (8. август 2015)
- [10] <http://billysugger.blogspot.com/2014/05/billysugger-simples-how-to-measure.html> (15. мај 2015),
- [11] https://en.wikipedia.org/wiki/Steinhart%E2%80%93Hart_equation (15. мај 2015),
- [12] <http://www.elitesecurity.org/t23768-33> (2. јун 2015),
- [13] <http://www.ptt.rs/korisnici/e/l/elcomp/PCB.htm> (2. јун 2015),
- [14] <https://prakticanastavaets.files.wordpress.com/2012/01/stamp-pane-ploce.pdf> (2. јун 2015),
- [15] <https://en.wikipedia.org/wiki/TRIAC> (15. јун 2015),
- [16] http://www.sigmatone.com/circuits/electronic_lock/triac_switch_to_control_high_voltage_devices.htm (15. јун 2015),
- [17] <http://playground.arduino.cc/Code/Bounce> (5. август 2015),
- [18] <https://www.arduino.cc/en/Reference/LiquidCrystalCreateChar> (5. август 2015).