

Šolski center Celje

Srednja šola za strojništvo, mehatroniko in medije

# **DODATNI VARNOSTNI SISTEM AVTA**

## **RAZISKOVALNA NALOGA**

Avtor:

Nejc TOVORNIK, M-2. f

Mentor:

Robert Ojsteršek, mag. inž. meh.

Mestna občina Celje, Mladi za Celje

Celje 2016

## **POVZETEK**

V današnjem času in v današnjih avtomobilih se vedno pogosteje srečamo z operacijskimi sistemi, ki nudijo skoraj popolno nastavitve avtomobila le na dotik. Nastavlja se lahko vse, od obratovanja motorja do vizualnih učinkov na armaturi. Takšnih sposobnosti starejša vozila niso imela na voljo, zato sem se odločil, da raziščem in zgradim sistem, ki omogoča vsaj nekaj teh možnosti.

## Kazalo vsebine

<b>POVZETEK .....</b>	<b>2</b>
<b>1 UVOD.....</b>	<b>7</b>
<b>2 HIPOTEZE .....</b>	<b>8</b>
<b>3 ELEKTRIČNI DEL.....</b>	<b>9</b>
3.1 PROCESNI DEL SISTEMA .....	9
3.2 KRMILJENJE RGB LED-DIOD.....	10
3.2.1 PULZNO ŠIRINSKA MODULACIJA .....	11
3.3 PRIDOBIVANJE ČASA .....	12
3.4 BLUETOOTH POVEZAVA.....	13
3.4.1 SERIJSKA KOMUNIKACIJA .....	14
3.5 MERJENJE NAPETOSTI.....	15
3.6 MERJENJE TOKA .....	16
3.7 SESTAVA ELEKTRIČNEGA NAČRTA.....	17
<b>4 IZDELAVA TISKANINE .....</b>	<b>20</b>
4.1 RAZVIJANJE TISKANIN .....	21
4.2 JEDKANJE .....	22
<b>5 PROGRAMIRANJE .....</b>	<b>25</b>
5.1 DOMAČI ZASLON.....	27
5.2 GLAVNI MENI .....	28
5.3 PODMENI NASTAVITVE .....	29
5.4 PODMENI AVTO .....	31
5.5 PODMENI SVETLOST .....	32
5.6 PODMENI BLUETOOTH .....	33
5.7 PODMENI KOLEDAR .....	34
<b>6 VGRADNJA SISTEMA .....</b>	<b>35</b>
6.1 OHIŠJA ZA TISKANINE.....	35
6.2 VGRADNJA SISTEMA V AVTO .....	36
<b>7 ZAKLJUČEK .....</b>	<b>39</b>
<b>8 VIRI.....</b>	<b>40</b>

## Kazalo slik

Slika 1: Arduino DUE .....	9
Slika 2: Aplikacija za povezavo .....	9
Slika 3: SMDRGB LED-dioda .....	10
Slika 4: Graf valovne dolžine svetlobe RGB LED .....	10
Slika 5: Vezava MOSFET-tranzistorjev .....	10
Slika 6: Merjenje pulzno moduliranega signala .....	11
Slika 7: Pulzna širina pri 50 % s periodo 54 $\mu$ s .....	11
Slika 8: Pulzna širina pri 25% s periodo 27 $\mu$ s .....	11
Slika 9: I <sup>2</sup> C-vodilo .....	12
Slika 10: Vezava DS3231 in AT24C02 čipa .....	12
Slika 11: Bluetooth modul HC-05 .....	13
Slika 12: Serijska TTL-komunikacija med dvema Arduino ploščama .....	14
Slika 13: Serijska UART-komunikacija med Arduinom in PC .....	14
Slika 14: Delilnik napetosti, prilagojen za merjenje napetosti do 20 V .....	15
Slika 15: Priklop delilnika napetosti .....	15
Slika 16: Priklop delilnika na tiskanini .....	15
Slika 17: ACS758ECB-200B .....	16
Slika 18: Notranja zgradba senzorja .....	16
Slika 19: Električna shema tiskanine, projektirana v programu Eagle CAD.....	17
Slika 20: Končna tiskanina, projektirana v programu Eagle CAD .....	18
Slika 21: Spodnja plast tiskanine .....	18
Slika 22: Zgornja plast tiskanine .....	18
Slika 23: I <sup>2</sup> C »pull up resistors« .....	19
Slika 24: Tiskanina s tokovnim in napetostnim merilcem .....	19

Slika 25: NaOH (lug) .....	20
Slika 26: HCL (solna kislina) .....	20
Slika 27: Vodikov peroksid .....	20
Slika 28: Prosojnice vezja .....	21
Slika 29: Žep za ploščo .....	21
Slika 30: Osvetljevanje tiskanine .....	21
Slika 31: Razvijanje tiskanine .....	21
Slika 32: Konec razvoja .....	21
Slika 33: Jedkanje tiskanine .....	22
Slika 34: Zjedkana tiskanina .....	22
Slika 35: Priprava za postavitve elementov .....	22
Slika 36: Zjedkana tiskanina .....	23
Slika 37: Priprava tiskanine na vrtanje .....	23
Slika 38: Populacija tiskanine .....	23
Slika 39: Populacija tiskanine .....	23
Slika 40: Zadnja stran ekrana .....	24
Slika 41: Preizkus izven vozila .....	24
Slika 42: Izsek kode izrisa drsnika svetlosti .....	26
Slika 43: Domači zaslon .....	27
Slika 44: Glavni meni .....	28
Slika 45: Nastavitve ure .....	29
Slika 46: Nastavitve dneva (izbran jezik: ANG) .....	29
Slika 47: Barvna paleta .....	30
Slika 48: Podmeni: Nastavitve .....	30
Slika 49: Podmeni: Svetlost .....	32
Slika 50: Podmeni: Svetlost .....	32

Slika 51: Podmeni: Bluetooth .....	33
Slika 52: Povezovanje nove naprave preko Bluetooth .....	33
Slika 53: Podmeni: Koledar .....	34
Slika 54: Qwerty tipkovnica .....	34
Slika 55: Priprava tiskanin za v ohišje .....	35
Slika 56: Procesorsko ohišje .....	35
Slika 57: Prvi korak razstavljanja.....	36
Slika 58: Drugi korak razstavljanja .....	36
Slika 59: Tretji korak razstavljanja .....	37
Slika 60: Ekran v zeleni barvi .....	37
Slika 61: Ekran in instrumenti v zeleni barvi .....	37
Slika 62: Ekran in instrumenti v modri barvi .....	38
Slika 63: Nastavitev svetlosti na ekranu in odziv svetlosti na instrumentih .....	38

## 1 UVOD

Cilj naloge je sprogramirati in zgraditi operacijski sistem, ki bo na podlagi različnih senzorjev vozniku javil napake ali opozorila. Vse, kar je prikazano na zaslonu, je lahko po izbiri v slovenskem ali angleškem jeziku.

Kot domači zaslon je prikazana ura, datum, dan, temperatura znotraj vozila, temperatura zunaj vozila in stanje povezave bluetooth. Ob dotiku na katero koli točko na zaslonu nas pripelje v glavni meni, kjer imamo na izbiro več možnosti, vse od osnovnih nastavitvev jezika do nastavitvev ure in tudi izbire barv na barvni paleti. Prikazano imamo električno porabo avtomobila, stanje akumulatorja, stanje alternatorja in še nekaj informacij.

Moja naloga je vpeljati čim več možnosti in izbire za uporabnika in mu omogočiti najlažjo in skoraj samoumevno uporabo naprave. Izdelava sistema mora biti cenovno nizka oz. nižja od že obstoječih podobnih sistemov. Zadostiti je potrebno tudi estetika v samem programu kot tudi v avtomobilu.

## 2 HIPOTEZE

Jedro raziskovalne naloge bo operacijski sistem. Različni ponudniki notranje opreme v avtomobilih nudijo omenjeni sistem, a takšna dodatna oprema lahko presega 1000 evrov. Ne smemo pa pozabiti, da tovrstni sistemi nudijo tudi navigacijo in številne druge možnosti. Za sistem sem v ta namen uporabil nizkocenovni krmilnik SAM3x8e proizvajalca Atmel in sestavil vezje, ki bo ugodilo naslednjim zahtevam.

Zahteve:

- Zgraditi sistem grafičnega prikazovanja.
- Omogočati urejanje na dotik.
- Urejanje ure, datuma, teme itd.
- Izbira jezika.
- Zajemanje in prikazovanje različnih senzorjev.
- Opozorilni sistem ob napaki ali novem dogodku.
- Povezava s pametnim telefonom.
- Izdelava aplikacije za pametni telefon.
- Spreminjanje barve na ekranu in armaturi.
- Vnos novih dogodkov na koledar in opomnik.



### 3 ELEKTRIČNI DEL

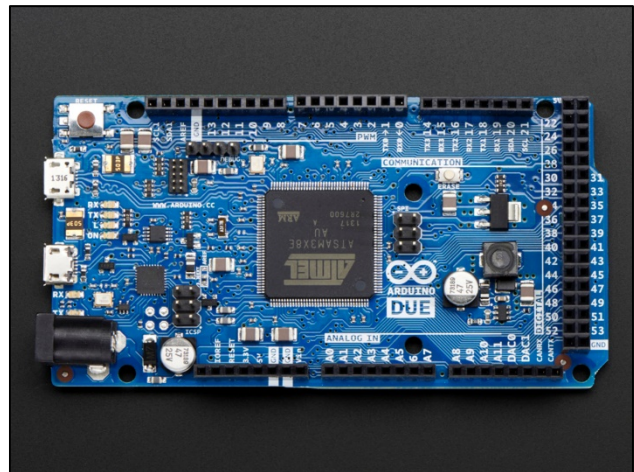
#### 3.1 PROCESNI DEL SISTEMA

Kot glavni pogon sistema sem uporabil razvojno ploščo Arduino DUE, ki jo poganja Atmelov 32-bitni procesor ATSAM3X8E. S 512 KB spomina nam ravno zadostuje za ta projekt in s frekvenco 84 MHz dovolj hitro postavlja vhode in izhode. Na voljo imamo 54 vhodov in izhodov, od tega je 12 sposobnih pulzno širinske modulacije. Obratovalna napetost procesorja je 3.3 V in dosežemo jo s pomočjo linearnega stabilizatorja. Vse ostale zunanje komponente obratujejo na napetosti 5 V, kar za nas predstavlja težavo pri komunikaciji. Če želimo z zunanjo napravo pošiljati podatke oz. signale v procesor, je potrebno uporabiti delilnik napetosti. Ta nam ščiti procesor tako, da signal, ki je v tem primeru od 0 do 5 V zmanjša in spusti na raven od 0 do 3.3 V.

Omejeni smo tudi s SRAM-spominom, natančneje s 96 KB. Tega s celotnim programom zasedamo približno 90 %. V tem spominu hranimo že vnaprej določene globalne spremenljivke in določila za program.

Izbira tega procesorja nam posebej koristi, ker ima sposobnost več serijskih komunikacij hkrati, kot tudi PC in še nekaj ostalih.

Samo programiranje je precej olajšano s programom Arduino IDE.



Slika 1: *Arduino DUE*



Slika 2: *Aplikacija za povezavo*

Ne smemo pozabiti tudi na mobilno napravo, na kateri je aplikacija za povezavo z našim sistemom. Sistem nam omogoča popolni nadzor in upravljanje vseh nalog, ki jih nudi preko pametnega telefona z operacijskim sistemom Android. Sestavil sem aplikacijo, ki se poveže z bluetooth modulom in preko serijske komunikacije komunicira z našim procesorjem. Pri takšnem načinu upravljanja mora obstajati tudi varnostni protokol. Glavno besedo imata še zmeraj procesor in ročno vnašanje ukazov. V primeru neželenih dogodkov lahko v nastavitvah sistema ročno izklopimo možnost uporabe bluetooth povezave.

### 3.2 KRMILJENJE RGB LED-DIOD

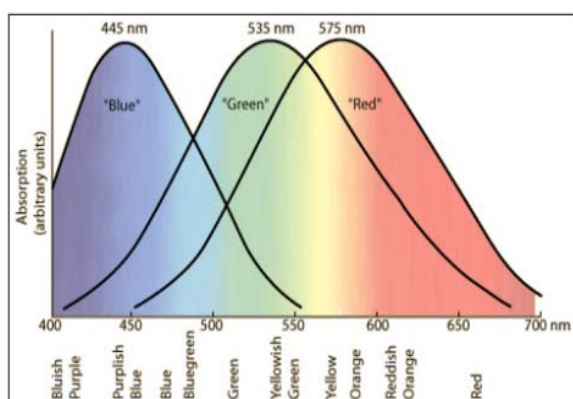
RGB LED-dioda je skupek treh LED-diod v barvni kombinaciji rdeča, zelena in modra. Če posamično dodajamo napajanje od 2 V do 3 V dobimo te tri osnovne barve. Če uporabimo pulzno širinsko modulacijo ali PWM, pa lahko vsaki LED-diodi natančno določimo, s kolikšno močjo bo svutila. Če združimo dve barvi in vsaki določimo neko moč svetilnosti, lahko pridobimo zmes teh dveh barv.

Če na primer vzamemo rdečo barvo in ji nastavimo 50 % pulzno širinske modulacije in če dodamo še zeleno barvo, prav tako s 50 % pulzno širinske modulacije, dobimo nekakšen približek rumenega odtenka. S tem načinom in z RGB-grafom sedaj lahko nastavimo formulo, po kateri bo naš mikroprocesor krmilil in prikazoval barve, ki jih želimo.

LED-diode v avtomobilu vgradimo pod armaturno ploščo, in sicer na mesto, kjer so bile prej navadne rumene žarnice. Za popolno osvetlitev armaturne plošče in vseh inštrumentov je bilo potrebno kar nekaj LED-diod. Vezane so v zaporedno vezavo po tri in napajamo jih z 12 V enosmerne napetosti.



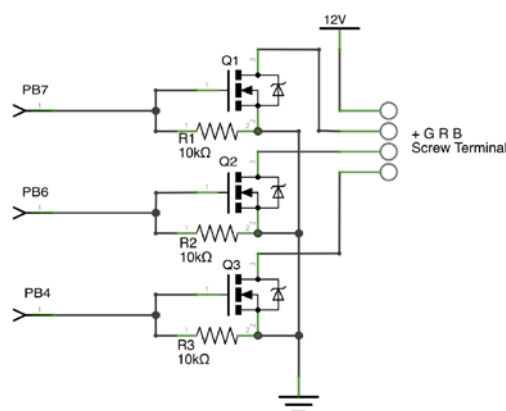
Slika 3: SMD RGB LED-dioda



Slika 4: Graf valovne dolžine svetlobe RGB LED

Na peti sliki lahko vidimo vezavo MOSFET-tranzistorjev, ki so označeni z Q1, Q2 in Q3. Potrebno je dodati upore (R1, R2 in R3), ki nam grobo rečeno določijo reakcijski čas odpiranja MOSFET-a. PB4, PB6 in PB7 so priklopi na procesor, ki odpirajo pot skozi MOSFET, da se lahko tokokrog konča.

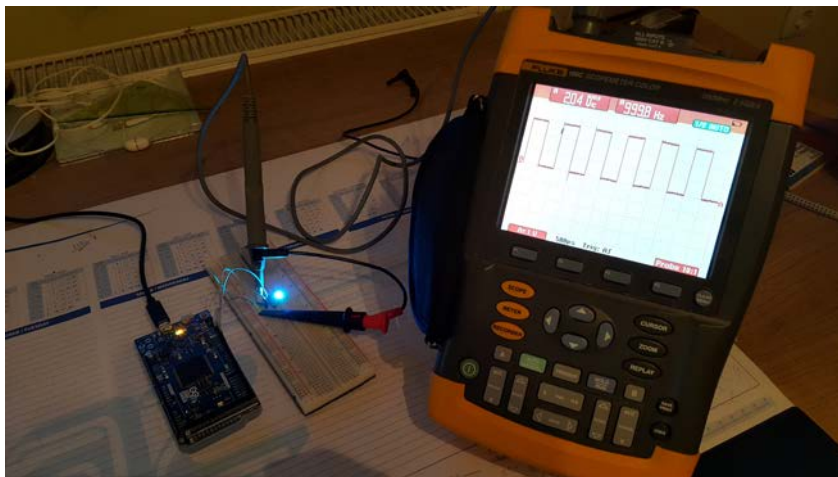
Za takšno vezavo potrebujemo MOSFET-tranzistorje, ki nam bodo krmilili LED-diode. Če bi LED-diode vezali direktno na procesor, bi zaradi preobremenitve stekel prevelik tok in bi ga uničilo. V ta namen sem uporabil IRFZ44NSPBF MOSFET, ki zdrži 50 A toka in maksimalno 55 V napetosti. To je več kot dovolj za mojo uporabo.



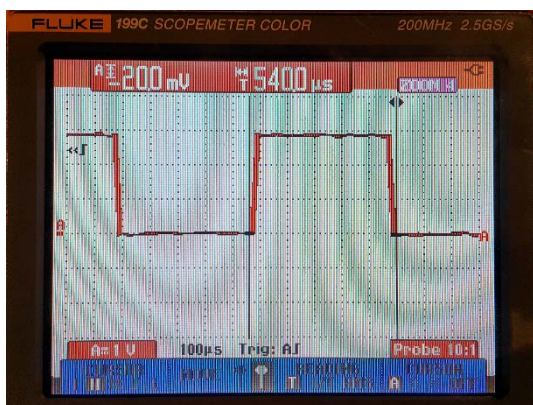
Slika 5: Vezava MOSFET-tranzistorjev

### 3.2.1 PULZNO ŠIRINSKA MODULACIJA

Princip te modulacije je prikazan na slikah 6, 7 in 8. Frekvenca, čas in višina impulzov ostajajo konstantni, spreminja se samo trajanje oz. širina periode. Če poenostavimo in vzamemo primer LED-diode, ki bi bila priklopljena na modulacijo s 50 % širino, bi ta oddajala le polovico svetlobne moči. Bolj ko se bližamo 100 % širini, bolj bo naša LED-dioda svetila. Tako ugotovimo, če je napetost od 0 do 5 V, lahko s širino dosežemo katerokoli vmesno napetost želimo. Vmesna napetost je povprečje med ON in OFF signalom.



Slika 6: Merjenje pulzno moduliranega signala



Slika 7: Pulzna širina pri 50 % s periodo 54  $\mu$ s



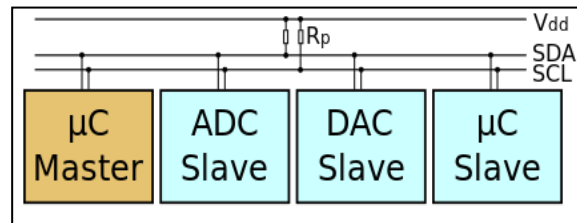
Slika 8: Pulzna širina pri 25 % s periodo 27  $\mu$ s

### 3.3 PRIDOBIVANJE ČASA

Za branje in beleženje časa, datuma, dneva in leta sem uporabil DS3231 čip, ki je narejen prav za to nalogo. Poleg sem dodal še AT24C02, ki je EEPROM spominski čip z 2 KB spomina za dodatno shrambo. Na takšen čip lahko program shranjuje ključne podatke in tudi če izklopimo napajanje, se ti podatki shranijo. Ko povrnemo napajanje, lahko procesor nadaljuje s prejšnjimi podatki, ki ostanejo nepoškodovani.

Ta dva čipa sta na procesor povezana z I<sup>2</sup>C-povezavo ali WIRE-povezavo. Procesor deluje kot gospodar in čipa kot slugi (master read – slave write). To pomeni, da kadar procesor zahteva podatke za čas, datum ali dan na naslovu 0x68 (DS3231), mu čip vrne z odgovorom. Če pa procesor piše na EEPROM-čip, bo poslal zahtevo na I<sup>2</sup>C BUS z naslovom 0x67 (AT24C02) in če mu čip odvrne, da lahko beleži, mu procesor pošlje podatke v shrambo.

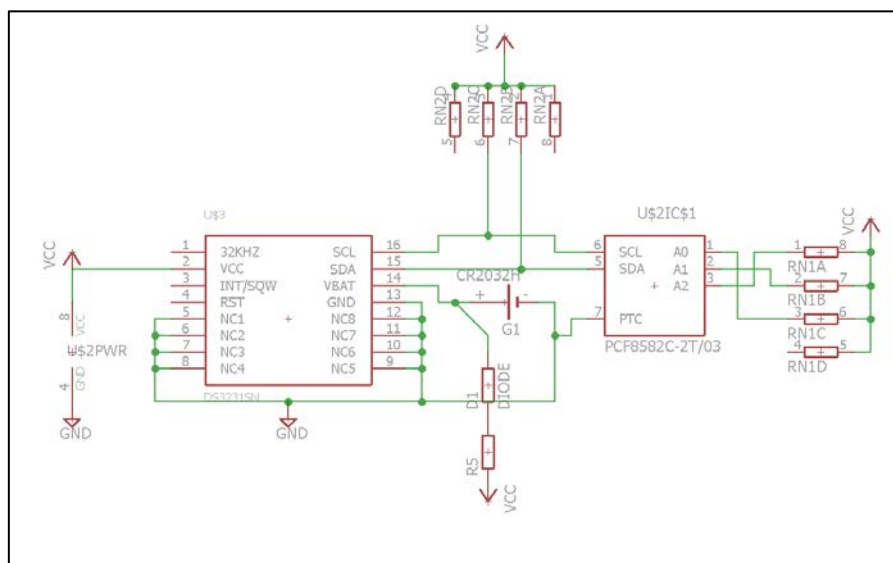
Takšna komunikacija je zelo počasna, ker si mora najprej master poiskati slava na določenem naslovu in nato iti čez postopek handshake, ki ju inicializira za nadaljnjo komunikacijo. Vsak takšen korak obremeni procesor in ga malo upočasni. Če je teh korakov zelo veliko, se lahko kmalu tudi vidijo posledice upočasnitve.



Slika 9:

I<sup>2</sup>C-vodilo

I<sup>2</sup>C BUS je vezava dveh žic SDA, ki je naša podatkovna pot in SCL, ki je skupna ura oziroma takt, da se lahko na SDA-žici uskladijo podatki. Na vsakih nekaj 10 cm dolžine pa je priporočeno uporabiti upore, vezane na napajanje.



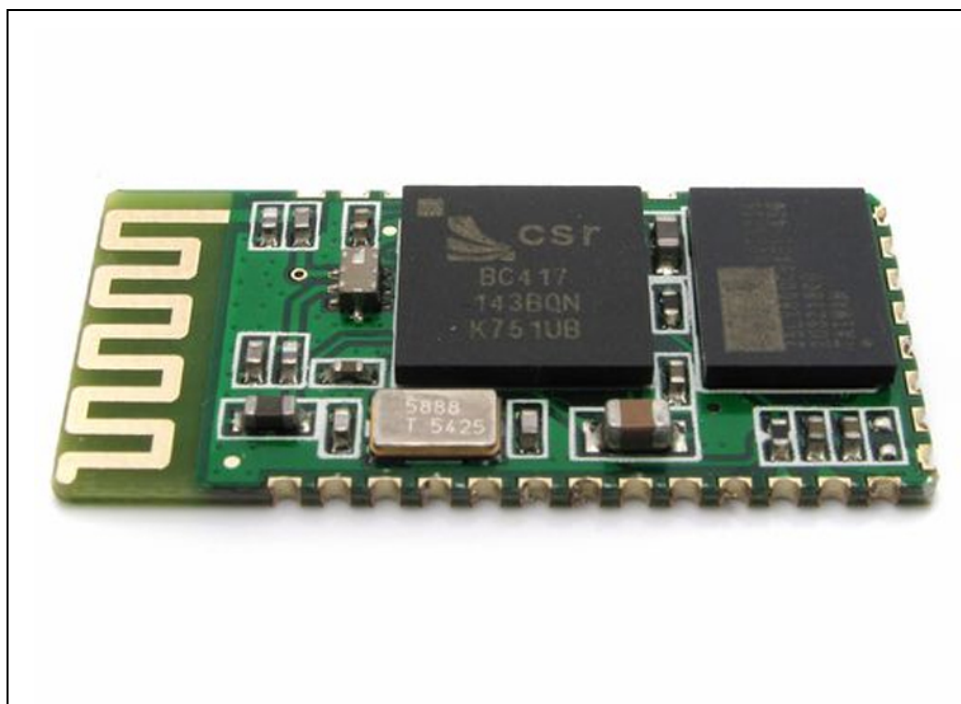
Slika 10: Vezava DS3231 in AT24C02 čipa

### 3.4 BLUETOOTH POVEZAVA

Za povezavo z mobilnim aparatom sem uporabil bluetooth modul HC-05. S procesorjem komunicira preko serijske komunikacije TTL ( transistor, transistor logic). Vezan je na Serial 1 z namenom, da je ločen od glavne serijske komunikacije, ki jo procesor uporablja za komunikacijo z računalnikom med programiranjem.

Prednost te komunikacije je hitrost, a slabost je v tem, da na sprejemni strani procesor ves čas preverja nove spremembe. Slabost se pojavi pri sprejemu novih podatkov, lahko se zgodi, da ne prejme polnega podatka oz. ni nikakršne funkcije, ki bi preverila, ali je prispel popolnoma ohranjen podatek.

Sam modul je zelo preprost za vezavo in uporabo, zahteva le napajanje od 2.5 V do 5 V. Ima tudi funkcijo key, ki ga postavi v AT-mode, kjer lahko nastavljamo ime naprave, geslo, podatkovno hitrost (baud rate) itd.

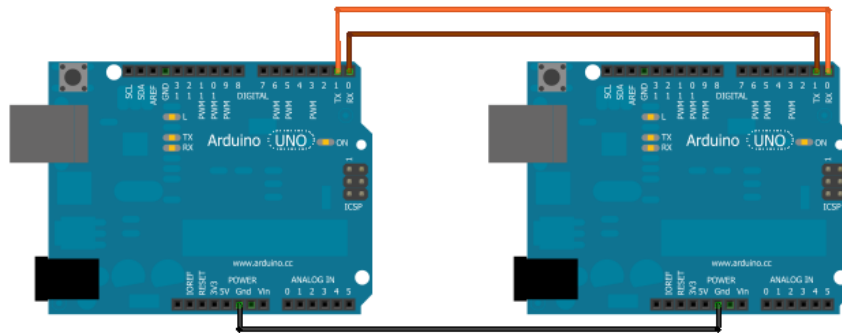


Slika 11: *Bluetooth modul HC-05*

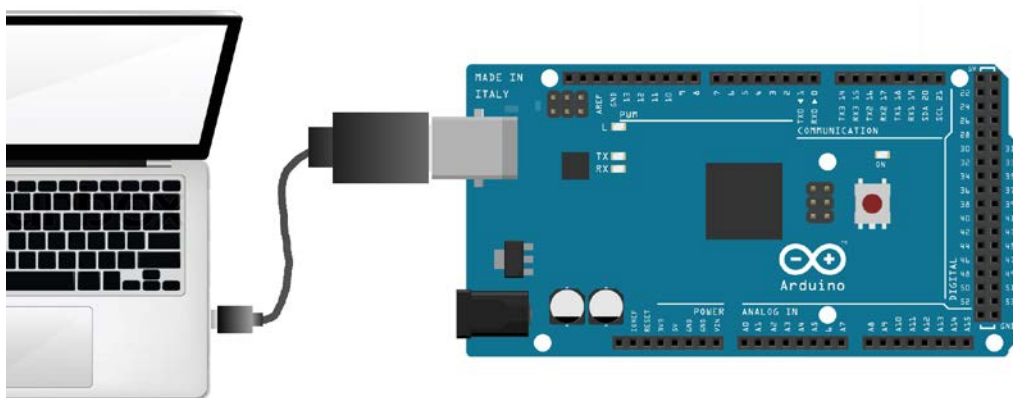
### 3.4.1 SERIJSKA KOMUNIKACIJA

Serijska komunikacija UART ali USART se uporablja za komunikacijo z razvojno ploščo in računalnikom. S tem postopkom tudi računalnik naloži program na procesor ali posega v obstoječ program prek serijskega monitorja, vgrajenega v Arduino okolje. Tovrstno komuniciranje je zelo podobno USB-komunikaciji, ki jo danes uporablja že skoraj vsa digitalna elektronika, ki ima sposobnost povezave z računalnikom.

Za pretvorbo iz USB v UART se na razvojni plošči nahaja čip ATMEGA16u2, ki je zadolžen za prevajanje v TTL-komunikacijo, ki jo procesor sprejme. Arduino DUE ima poleg programskega USB-vhoda tudi USB direktni vhod, ki je vezan neposredno na procesor SAM3X8E. Poleg tega pa ima DUE še 3 serijske TTL vhodno-izhodne enote za komunikacijo z zunanjimi TTL sposobnimi napravami.



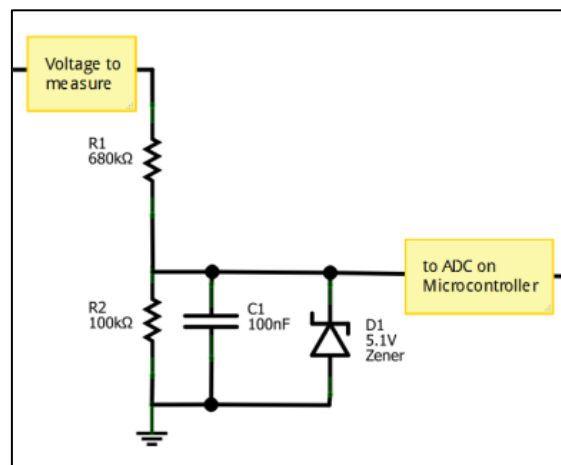
Slika 12: *Serijska TTL-komunikacija med dvema Arduino ploščama*



Slika 13: *Serijska UART-komunikacija med Arduinom in PC*

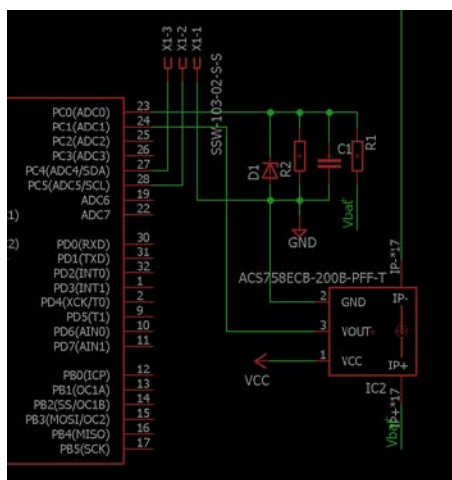
### 3.5 MERJENJE NAPETOSTI

Kot dodatek za sistem sem dodal funkcijo merjenja napetosti akumulatorja. Delilnik napetosti je potreben za varno merjenje nad obratovalno napetostjo samega procesorja. Na sliki 13 je konfiguracija pripravljena za 20 V vhod. Takšna izvedba merjenja napetosti je preprosta in poceni, ampak nevarna, če pride do previsoke napetosti na merilni strani. V vezavi je dodan C1 kondenzator za izničevanje visokih frekvenc, ki bi ovirale meritev in D1 zener dioda, ki ščiti vsa vezja na poti do procesorja.

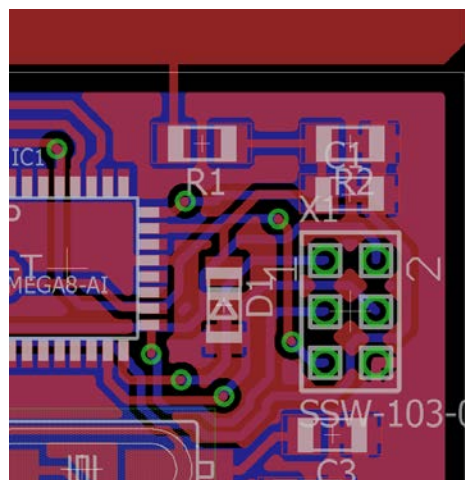


Slika 14: Delilnik napetosti, prilagojen za merjenje napetosti do 20 V

Pri preizkusu sem opazil, da se vezje kar dobro obnese. Z dobrim preračunom sem uspešno dosegel iste rezultate kot digitalni multimeter.



Slika 15: Priklop delilnika napetosti



Slika 16: Priklop delilnika na tiskanini

### 3.6 MERJENJE TOKA

Da bi učinkovito meril porabo električne napetosti, sem v ta namen uporabil HALL efekt sondo, ki je galvansko ločena od napetosti akumulatorja. To dosežemo tako, da jo pritrdimo na feridni obroč, skozi katerega poteka napajalna žica. V moji izvedbi sta obroč in sonda vgrajena v ohišje, skozi katero poteka glavni vod (+12 V akumulatorja). Ta se imenuje ACS758ECB-200B in prenese 200 A maksimalnega toka.

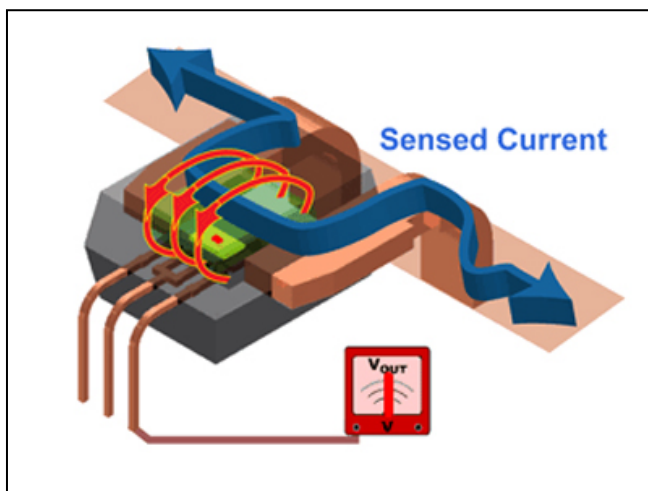
Na sliki 15 lahko vidimo prav ta senzor. Na procesni strani ima 3 nožice, na katere dodamo napajanje in povratno prejmemo analogni signal, ki ga lahko s preračunom izpišemo v amperih.

Dve veliki bakreni peresi sta namenjeni med glavni vod napajanja porabnika, ki ga merimo. Potrebno je tudi preveriti, kako je senzor priklopljen oz. ali bere pretok napetosti v obe smeri ali samo v eno.

Za dodatno varnost sem dodal tudi možnost odstranljive vezi, ki bi jo vstavil med obe peresi v primeru, da ob prenapetosti zadeva pregori.



Slika 17: ACS758ECB-200B



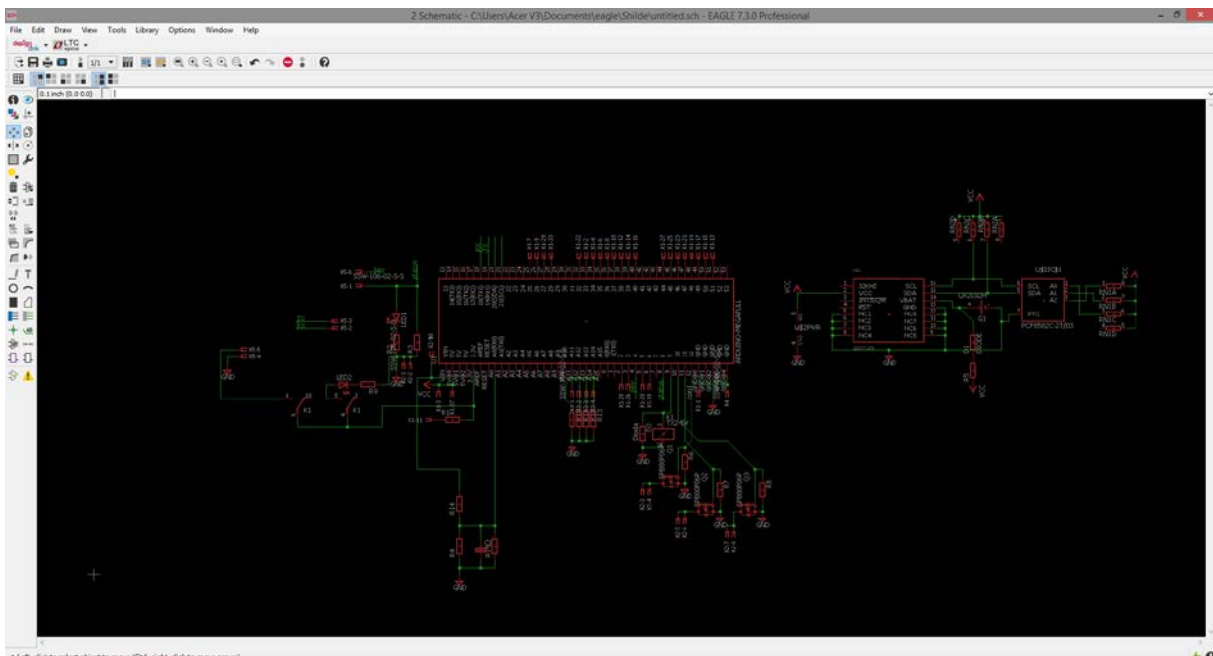
Slika 18: Notranja zgradba senzorja



### 3.7 SESTAVA ELEKTRIČNEGA NAČRTA

Električni načrt za celotni projekt sem zrisal v programu Eagle CAD. Ta program je odličen pripomoček za izdelavo vrhunskih tiskanic. Nudi veliko izbiro električnih komponent in je dokaj preprost za uporabo. Samo vezje začnemo najprej izdelovati z načrtom in povezavo elementov, nato pa sledi risanje vezi in postavitvev. Z malo vaje lahko proizvedemo odlično tiskanino.

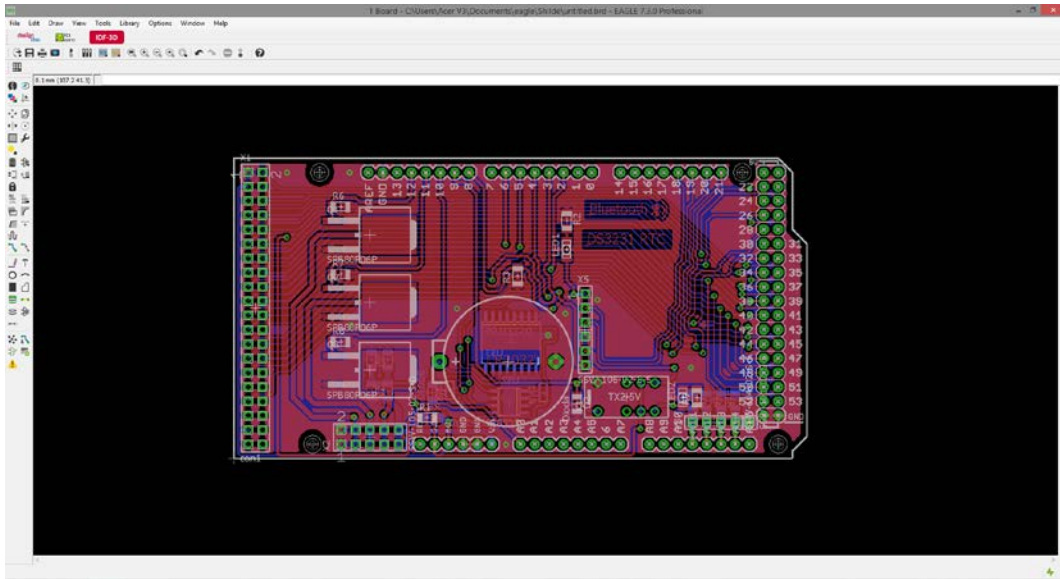
Odločil sem se, da se bo tiskanina nahajala nad samim Arduinoom kot nekakšen priklop ali "Shield". Spodnji procesor se skozi priklope ob straneh poveže z zgornjo ploščo, ki pa je naše vezje. S to postavitvijo zmanjšamo prostornino in velikost, ki bi jo sistem zavzemal.



Slika 19: Električna shema tiskanine, projektirana v programu Eagle CAD

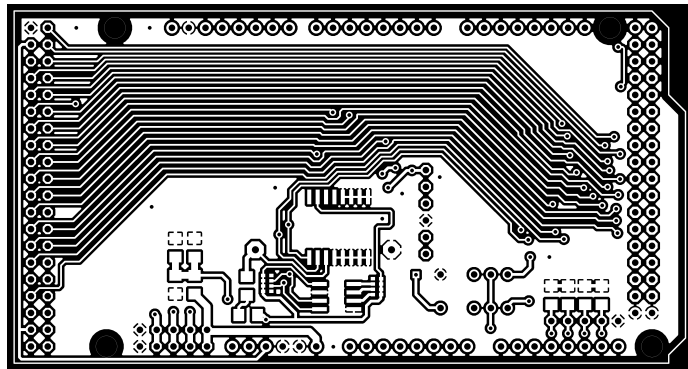
Na zgornji sliki je prikazan električni vezalni načrt, ki se razdeli na 4 glavne dele:

1. BUS-povezava med procesorjem in ekranom,
2. DS3231 ura oz. čip za izpis časa, datuma in dneva,
3. Bluetooth povezava,
4. Vezava MOSFET-tranzistorjev za RGB LED-diode.

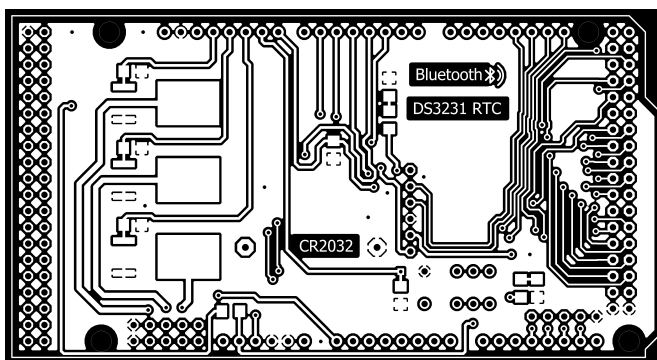


Slika 20: Končna tiskanina, projektirana v programu Eagle CAD

Slika 20 prikazuje zgornji pogled nad vezje. Vse vezi, ki so v rdeči barvi, so na zgornji strani tiskanine, vse vezi v modri pa na spodnji strani. DS3231, EEPROM, MOSFET-tranzistorji, razni upori, kondenzatorji in signalne LED-diode so v SMD-obliki. To pomeni, da so elementi pritrjeni na površini tiskanine in njihove nogice ne gledajo skozi vezje. SMD-oblika je tudi manjša in kompaktnjša, kar je v mojem primeru koristno, saj smo zelo omejeni s prostorom.



Slika 21: Spodnja plast tiskanine



Slika 22: Zgornja plast tiskanine

Ob zaključku risanja tiskanine se izbere zgornja ali spodnja stran in se iztiska na prosojno folijo ali folijo za grafoskope. Tako dobimo negativ, ki se v procesu izdelave fizične tiskanine prenese na bakreno podlago.

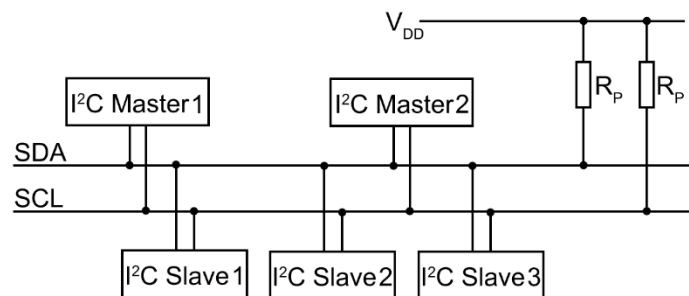
Dodatno sem zrisal še eno tiskanino, ki drži tokovni merilnik ACS758ECB-200B in napetostni delilnik. Povezujeta se na Atmelov čip ATMEGA328p, ki z I<sup>2</sup>C-komunikacijo na ukaz pošlje pridobljene vrednosti na glavni procesor SAM3X8E.

Tokovni senzor in napetostni delilnik konstantno zajemata podatke, kar bi preobremenilo glavni procesor, ki že tako obdeluje velike količine kode. ATMEGA328p-čip je v bistvu nosilec dodatnega bremena, saj lahko zelo hitro zavzema podatke o toku in napetosti. Ob ukazu SAM3X8E-procesorja pošlje zajete informacije po I<sup>2</sup>C-komunikaciji.

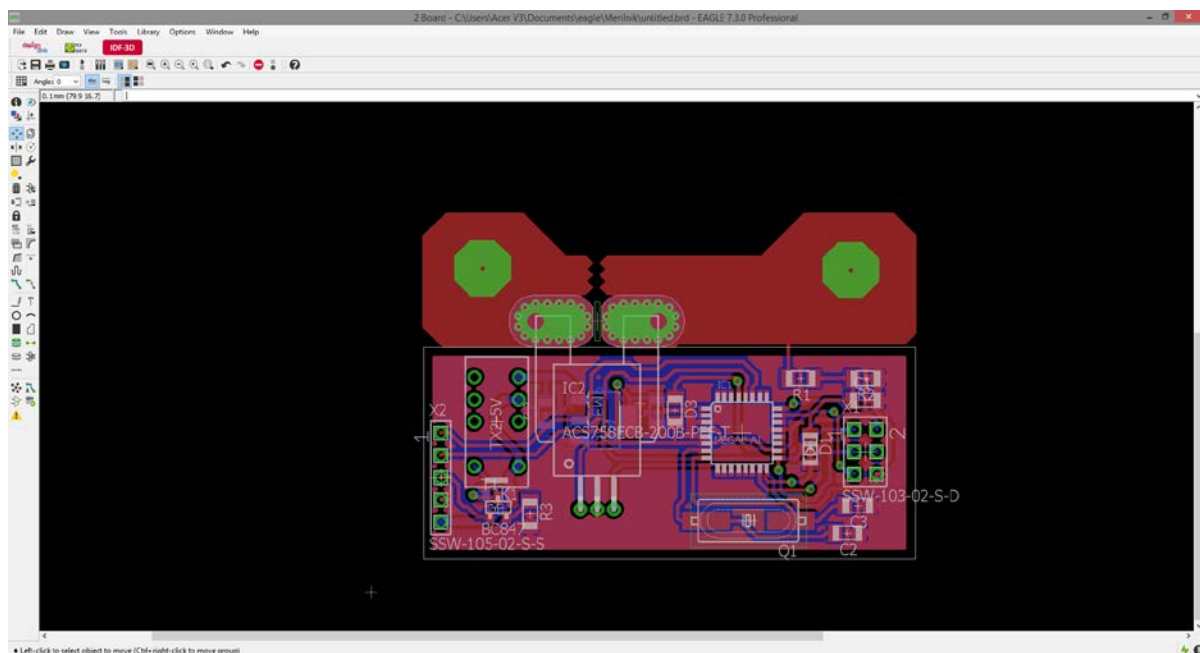
Zato imamo dodatni procesor (ATMEGA328p), da meri višje obratovalne napetosti, kot je sama napajalna napetost glavnega procesorja (SAM3X8E) in z njim ščitimo pred nenamerno napako in uničenjem.

Za komunikacijo je potrebno, da ne presežemo meje 3.3 V, kar lahko pomeni uničenje našega procesorja. Zaradi tega sem spustil napajalno napetost ATMEGA328p s 5 V na 3.3 V in s tem zagotovil, da ne more preseči te meje.

Na tiskanino dovajamo napetost +12 V in jo z linearnim stabilizatorjem spustimo na 3.3 V. Kot izhod imamo SDA, SCL (I<sup>2</sup>C) in skupni minus. Ker tovrstna komunikacija ni najidealnejša za daljšo razdaljo, je potrebno dodati upore, ki povlečejo signal navzgor proti plusu (pull up resistors).



Slika 23: I<sup>2</sup>C »pull up resistors«



Slika 24: Tiskanina s tokovnim in napetostnim merilcem

## 4 IZDELAVA TISKANINE

Izdelava tiskanega vezja je lahko zaradi kemikalij zelo nevaren postopek. Uporaba zaščitnih sredstev je nujna, prav tako dobro prezračen prostor, če se postopki izvajajo v zaprtem prostoru. Poznamo več postopkov izdelave tiskanin, sam sem se odločil za fotopostopek.

Začne se s tiskanjem prosojnice z narisanim vezjem. Za dober razvoj tiskanine je potrebno, da skozi zatemnjene vezi ne pride skoraj nič UV-svetlobe. Za to lahko poskrbimo tako, da natisnemo dvojno enako prosojnico in jo natančno položimo na prvo. V primeru, da je tiskanina dvostranska, iz prosojnic naredimo žep, v katerega položimo dvostransko ploščico s fotolakom.

Ploščico in prosojnico položimo v UV osvetljevalno mizo in ju osvetljujemo od 1 do 2 minuti. Po osvetljevanju je potrebno tiskanino razviti v natrijevem hidroksidu (NaOH). V tej fazi se osvetljeni del tiskanine uniči in NaOH ga odstrani. Med vsakim kemičnim postopkom je potrebno tiskanino sprati s tekočo vodo.

Za jedkanje pripravimo zmes kisline in vode (1 : 1), klorovodikove kisline (HCl) in vodikovega peroksida (H<sub>2</sub>O<sub>2</sub>). Tako dobimo jedko raztopino, s pomočjo katere odstranimo odvečni baker (Cu) z osvetljenih delov vezja. Reakcija kisline z bakrom proizvede veliko strupenih in zdravju škodljivih plinov in je zelo eksotermna. Pri tem postopku je nujna uporaba zaščitnih rokavic, očal in maske. Potreben je dobro prezračen prostor in podlaga iz umetnega materiala.

Ob končanem jedkanju tiskanino speremo pod vodo in jo obrišemo s tehničnim alkoholom. Preostane nam kislina za jedkanje in razvijalec. Kislino shranimo v plastično posodo in na vrhu naredimo luknjo. Razvijalec pa shranimo v stekleno zatemnjeno posodo, da ga svetloba ne uniči.



Slika 25: NaOH (lug)



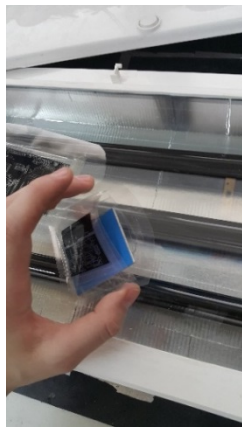
Slika 26: HCL (solna kislina)



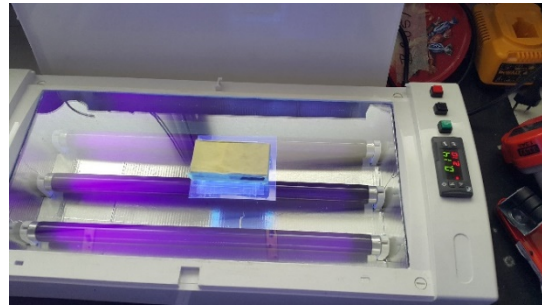
Slika 27: Vodikov peroksid

## 4.1 RAZVIJANJE TISKANIN

Iz programa Eagle CAD sem natisnil vezja na prosojnice in naredil žep za vstavitve dvostranske plošče s fotolakom. Vezje sem osvetljeval na doma narejeni mizi za osvetljevanje, ki sem jo sestavil zaradi velike količine tiskanin, ki jih proizvedem. Miza vsebuje tri 15 W UV neonske luči, ki se vklopijo in ugasnejo na časovnik. Tiskanino sem z vsake strani osvetljeval po 2 min in 10 s. Ob zaključku sem tiskanino pustil v zatemnjenem prostoru dokler razvijalec ni pripravljen.



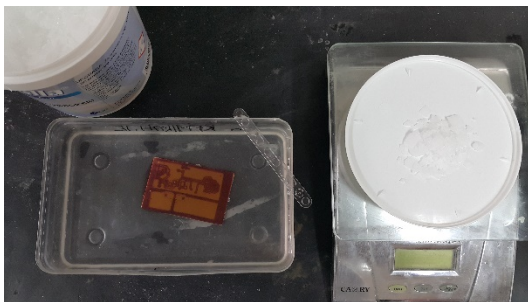
Slika 29: Žep za ploščo



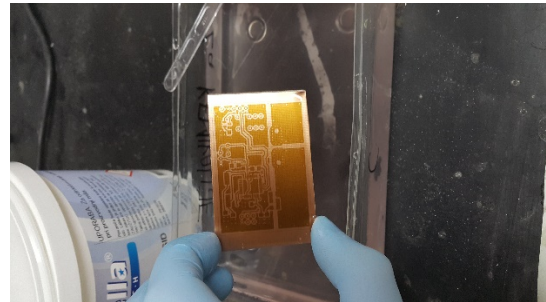
Slika 30: Osvetljevanje tiskanine

Slika 28: Prosojnice vezja

Razvijalec (NaOH) v obliki lusk (lug) sem dodal vodi v razmerju 7 g NaOH na 1 l H<sub>2</sub>O. Reakcija z vodo je eksotermična, kar pospeši in pripomore k razvoju tiskanine. Ko je NaOH popolnoma raztopljen, se v raztopino potopi tiskanina. Po nekaj sekundah opazimo rdečkasto plast na površini, kjer je bilo vezje osvetljeno. Tako počasi odstranimo uničeno plast fotolaka. Ostane nam zaščita, ki je pripravljena za jedkanje. Tiskanino speremo pod vodo in pripravimo za sledeči korak.



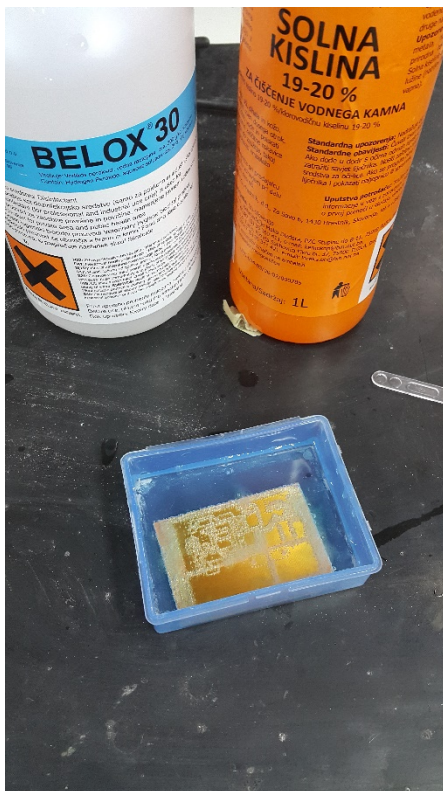
Slika 31: Razvijanje tiskanine



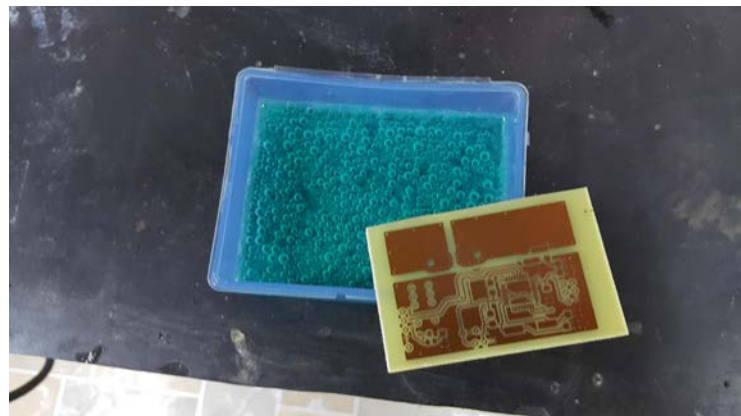
Slika 32: Konec razvoja

## 4.2 JEDKANJE

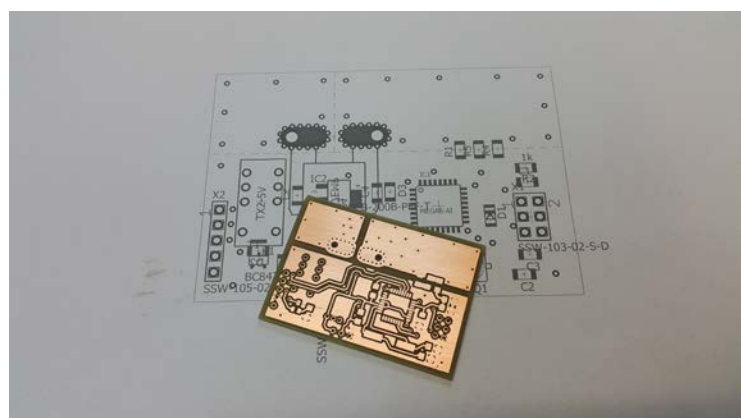
Ko je tiskanina razvita in pripravljena, sledi korak jedkanja. Pripravimo raztopino vode, klorovodikove kisline (HCl) in vodikovega peroksida ( $H_2O_2$ ) v razmerju 1 : 1 : 1. Raztopino pripravimo v posodi iz umetne mase ali stekla. Paziti moramo, da ne izpostavimo kakšnega orodja iz kovine. Ko je vse pripravljeno, potopimo tiskanino v raztopino. Če je reakcija preveč agresivna in eksotermna, dodamo vodo. Obvezno si zaščitimo roke in oblačila, saj je ta raztopina zelo jedka in lahko pusti opekline ter uničeno blago. Ob končanem jedkanju vzamemo tiskanino iz raztopine in jo speremo pod tekočo vodo. Preostalo je še samo vrtnanje lukenj, populacija tiskanine in brušenje robov.



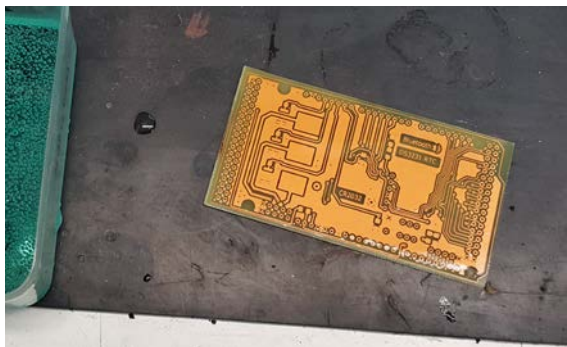
Slika 33: Jedkanje tiskanine



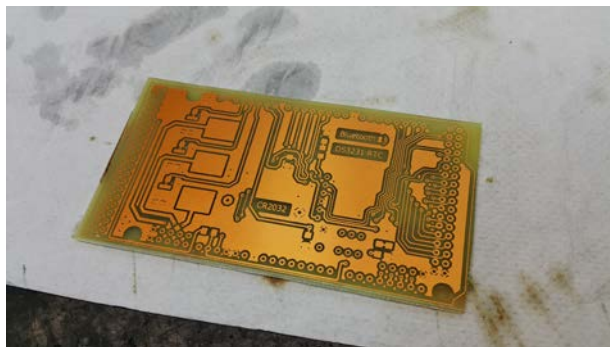
Slika 34: Zjedkana tiskanina



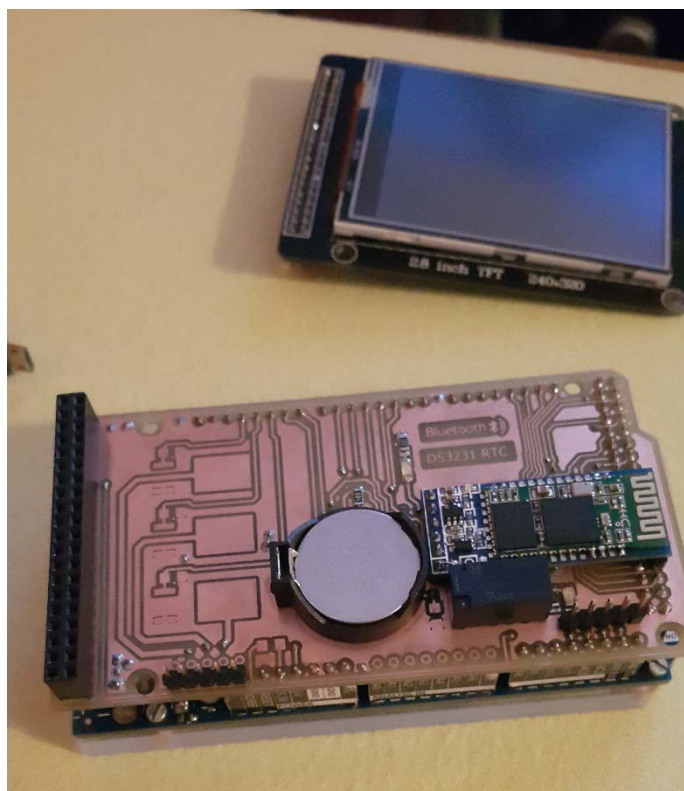
Slika 35: Priprava za postavitev elementov



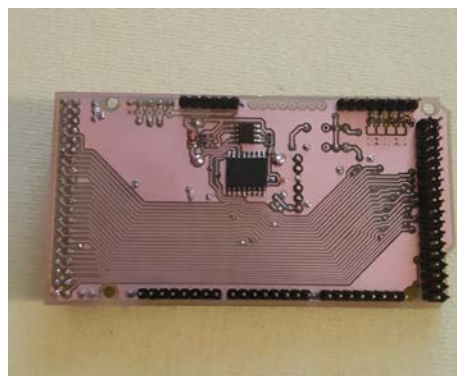
Slika 36: Zjedkana tiskanina



Slika 37: Priprava tiskanine na vrtanje



Slika 39: Populacija tiskanine



Slika 38: Populacija tiskanine

Popolnoma isti postopek izdelave sem uporabil za izdelovanje glavne tiskanine, ki povezuje vse module in MOSFET-tranzistorje.



Slika 40: Zadnja stran ekrana



Slika 41: Preizkus izven vozila



## 5 PROGRAMIRANJE

Program sem napisal v Arduino IDE-okolju, ki uporablja C in C++ programski jezik. Ta program Arduino prevede iz C-kode v strojno kodo in to pošlje preko USB-vmesnika na SAM3X8E procesor.

Za program je bilo potrebno uporabiti precej knjižnic in neuspešnih poizkusov. Ugotovil sem, če si izrišem načrt delovanja, je potem lažje sestaviti modularni program. To pomeni, da lahko kadar koli pozneje na zelo olajšan način dodam ali nadgradim sistemu funkcijo.

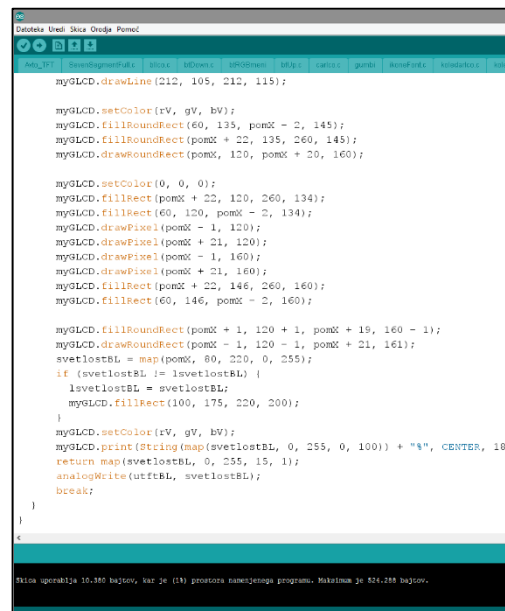
### Delitev kode:

- **Domači zaslon**
  - Prikaz ure
  - Prikaz dneva
  - Prikaz datuma
  - Prikaz leta
  - Prikaz temperature znotraj in zunaj vozila
  - Stanje povezave Bluetooth
  
- **Glavni meni**
  - Prikaz ure, temperature
  - Prikaz podmenija: Nastavitve
  - Prikaz podmenija: Avto
  - Prikaz podmenija: Svetlost
  - Prikaz podmenija: Bluetooth
  - Prikaz podmenija: Koledar
  
- **Podmeni: Nastavitve**
  - Urejanje jezika (SLO ali ANG)
  - Urejanje barve (selektivna barvna paleta)
  - Nastavitev izpisa temperature (°C ali °F)
  - Nastavitev ure
  - Nastavitev datuma
  - Nastavitev dneva
  
- **Podmeni: Avto**
  - Prikaz in izpis porabe el. toka
  - Prikaz in izpis el. napetosti v akumulatorju
  - Prikaz, ali alternator polni
  - Javljanje napak o el. napetosti v avtomobilu
  - Komunikacija z ATMEGA328p, ki meri tok, napetost in temperaturo
  - Prikaz in izpis porabe el. toka bas ojačevalnika in avdio sistema

- **Podmeni: Svetlost**
  - Prikaz drsnika za uravnavanje svetlosti
  - Nastavljanje svetlosti RGB LED-diod
  
- **Podmeni: Bluetooth**
  - Izklop ali vklop funkcije bluetooth
  - Prikaz stanja povezave
  - Prikaz uporabnika oz. naprave
  - Možnost prekinitve povezave z uporabnikom
  - Črni seznam uporabnikov
  
- **Podmeni: Koledar**
  - Izris koledarja po mesecih za tekoče leto
  - Ob kliku na mesec pokaže vnos opomnika
  - Možnost dodajanja novega vnosa opomnika
  - Možnost odstranjevanja opomnika
  - Na domači zaslon javi opomin za tekoči dan in uro

Pri tako obsežnem programiranju je potrebno ločevati kodo ali po listih ali po vnosih. Zlahka se ob pogledu v program izgubimo ali ne najdemo želene kode, ki bi jo spremenili oz. popravili.

Programu ne bi bilo konca, če ne bi bil omejen s prostorom. Večkrat sem ga začel ponovno pisati zaradi neurejenosti ali pa zaradi slabe modularnosti. Zdaj je v fazi, ko ga je enostavno dopolnjevati in dodajati funkcije. V naslednji fazi razvoja imam v načrtu dodati GPS- in GSM-povezavo. Tako bo lahko voznik s SMS-sporočilom zaklenil vozilo, ugotovil, kje je, če je vozilo ukradeno, in prejemal nujna SMS-sporočila, ko je potreben servis.



```

myGLCD.drawLine(212, 105, 212, 115);

myGLCD.setColor(rV, gV, bV);
myGLCD.fillRect(60, 135, pomX - 2, 145);
myGLCD.fillRect(pomX + 22, 135, 260, 145);
myGLCD.drawRoundRect(pomX, 120, pomX + 20, 160);

myGLCD.setColor(0, 0, 0);
myGLCD.fillRect(pomX + 22, 120, 260, 134);
myGLCD.fillRect(60, 120, pomX - 2, 134);
myGLCD.drawPixel(pomX - 1, 120);
myGLCD.drawPixel(pomX + 21, 120);
myGLCD.drawPixel(pomX - 1, 160);
myGLCD.drawPixel(pomX + 21, 160);
myGLCD.fillRect(pomX + 22, 146, 260, 160);
myGLCD.fillRect(60, 146, pomX - 2, 160);

myGLCD.fillRect(pomX + 1, 120 + 1, pomX + 19, 160 - 1);
myGLCD.drawRoundRect(pomX - 1, 120 - 1, pomX + 21, 161);
svetlostBL = map(pomX, 80, 220, 0, 255);
if (svetlostBL != lsvetlostBL) {
  lsvetlostBL = svetlostBL;
  myGLCD.fillRect(100, 175, 220, 200);
}
myGLCD.setColor(rV, gV, bV);
myGLCD.print(String(map(svetlostBL, 0, 255, 0, 100)) + "%", CENTER, 18);
return map(svetlostBL, 0, 255, 15, 1);
analogWrite(ufftBL, svetlostBL);
break;
}
}

```

Slika 42: Izsek kode izrisa drsnika svetlosti

## 5.1 DOMAČI ZASLON

Domači zaslon prikazuje trenutni čas, datum, dan in temperaturo ter je pripravljen, da nas popelje v glavni meni ob dotiku katere koli točke na zaslonu. Ozadje zaslona je temno sivo oz. črno, da se zlije z interierjem avtomobila.

Opozorila, s katerimi se srečamo na domačem zaslonu:

- Opozorilo poledice, ko je zunanja temperatura manjša od 3 °C.
- Opozorilo okvare alternatorja, ko napajalna napetost pade pod 13,5 V.
- Opozorilo okvare akumulatorja, če ob vžigu motorja napetost pade pod 9 V.
- Opozorilo nove prijave preko Bluetootha.
- Opozorilo opomnika ob določeni uri ali ob vžigu avtomobila.



Slika 43: Domači zaslon

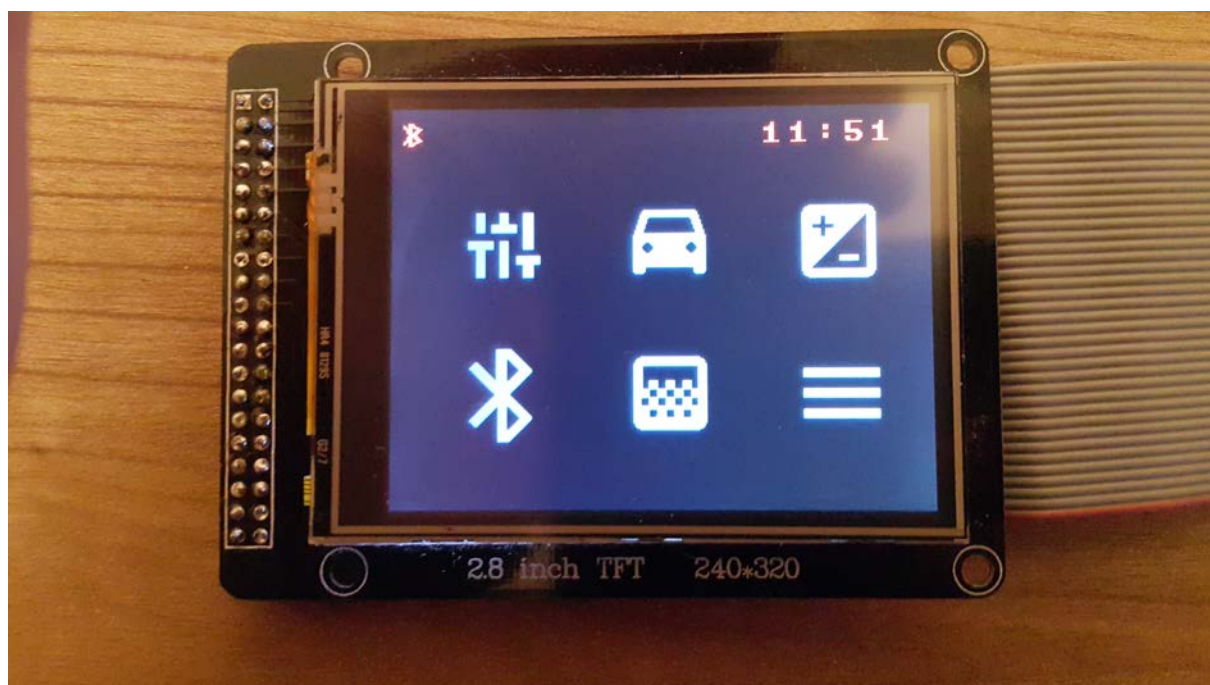
## 5.2 GLAVNI MENI

V glavnem meniju se nahajajo ikone z različnimi nastavitvami ali prikazi. V zgornji vrstici imamo stanje povezave Bluetooth in uro. Če se ne dotaknemo ali ne uporabimo zaslona, se po 1 minuti povrne na domači zaslon.

### Ikone:

- Nastavitve
- Avto
- Svetlost
- Bluetooth
- Koledar
- Vrnitev na domači zaslon

Koda menijev je pisana v switch casu, kar pomeni, da lahko preprosto dodajamo in odvezujemo menije. Ikone so slike, narisane s pomočjo GIMP2-programa (v .png obliki) in nato pretvorjene v C-kodo s pomočjo spletnega pretvornika Rinky Dink Electronics, kjer je tudi osnovna koda za knjižnico takšnega ekrana. Ikone so zapisane na spomin procesorja in ne na SD-kartico (kompatibilni problemi), kar zaseda 40 % celotne kode.



Slika 44: Glavni meni

### 5.3 PODMENI NASTAVITVE

V nastavitvah najdemo osnovne podatke, kot so: jezik, barva, ura, datum, dan in tudi nastavev pretvorbe temperature v °F. V zgornji vrstici vidimo razpoložljivost povezave bluetootha in ure.

Jezik je nastavljen v slovenščino ali angleščino, v prihodnosti pa tudi nemščino. Osvetljeni tekst nam pove, na kateri izbiri smo. Ob pritisku na tekst s temnim ozadjem se nastavev spremeni.

Temperatura uporablja preprosto formulo  $T(^{\circ}\text{F}) = T(^{\circ}\text{C}) \times 9/5 + 32$ , ki jo uporabi le ob pritisku na gumb °F. Temperaturo s sonde DS1820 prejemamo v °C na ATMEGA328p, potem jo posreduje glavnemu procesorju, ki jo objavi na ekranu.

Ob nastavljanju ure ali datuma se nam odpre okno, na katerem so prikazane puščice, ki prištevajo ali odštevajo čas oz. dan. Ob kliku na rob ekrana se nastavev shrani in pošlje na RTC-modul.

Ura, datum in jezik se prav tako lahko nastavljajo preko bluetooth povezave.



Slika 45: Nastavev ure



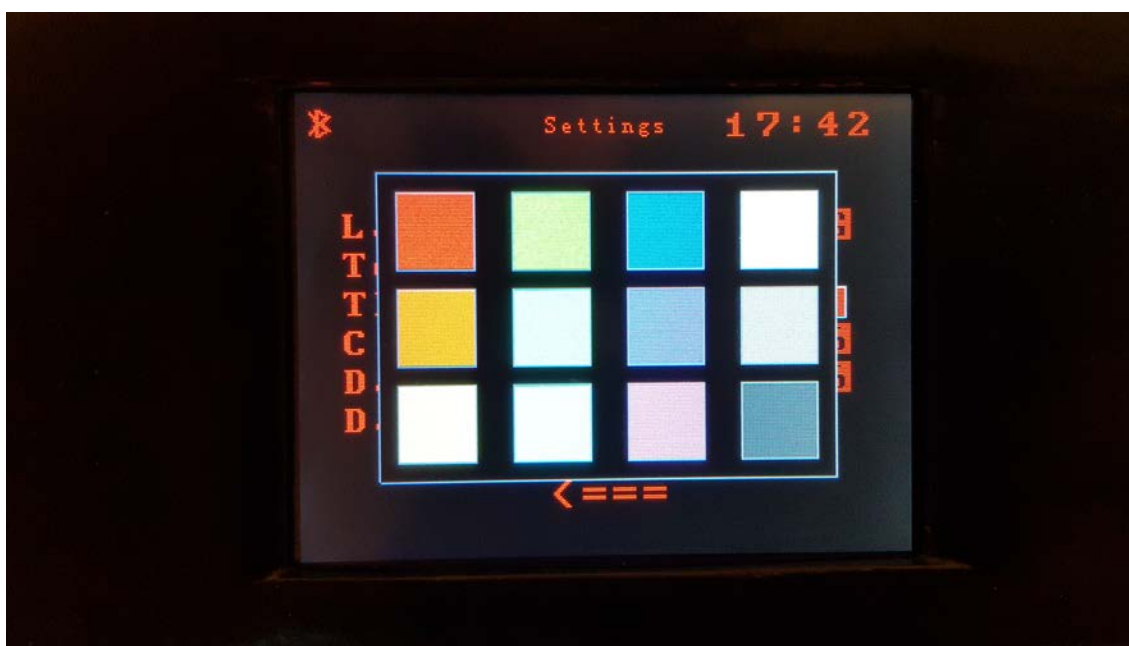
Slika 46: Nastavev dneva (izbran jezik: ANG)

Popolnoma enako je za nastavev dneva. Izberemo le želeni dan in pritisnemo na rob ekrana. To nas popelje nazaj v podmeni nastavitve in shrani dan na RTC-modul.

Na dnu ekrana se nahaja puščica, ki nas popelje nazaj na glavni meni. Lahko pa počakamo 1 minuto in sistem se povrne na domači zaslon.

V primeru, da nastavljam uro ali datum, pa procesor ne more sprožiti opozorila zaradi morebitnih dodatnih napak, ki bi jih povzročili oz. bi bilo moteče za uporabnika.

V naslednji fazi razvoja se nastavitvam doda še stranski drsnik, saj bo na voljo mnogo dodatnih nastavitvev.



Slika 47: Barvna paleta



Slika 48: Podmeni: Nastavitve

## 5.4 PODMENI AVTO

V podmeniju avto se nahajajo informacije o porabi električnega toka in o napetosti na akumulatorju. Tukaj poteka komunikacija z ATMEGO328p, ki se nahaja pri motorju, kjer meri zunanjo temperaturo, tok, ki ga porablja celoten avto, in napetost na akumulatorju. Če vemo, da se normalna napetost akumulatorja giblje od 11.5 V do 12 V, lahko preverjamo zdravje samega akumulatorja. To lahko preverimo tako, da ob zagonu motorja izmerimo padec napetosti in če je ta pod 9 V, pomeni, da je akumulator že preslab. Ob takšnem primeru bo tudi glavni procesor javil napako akumulatorja.

Kadar motor žene alternator, je napetost na akumulatorju od 13,8 V do 14,3 V. Če dosegamo to stanje, se nam izpiše na ekranu, da je alternator zdrav. Če pa pride do okvare na diodnem bloku alternatorja, pa napajalna napetost pade in sproži opozorilo. Takšne vrste opozorilo tudi že imamo na armaturi. Kadar je okvara na alternatorju na armaturi, zagori rdeča luč s simbolom akumulatorja.

Tok, ki ga merimo na akumulatorju, ob zagonu motorja naraste z 1,5 A na 80 A ali 90 A. Zaradi tega je potrebno imeti pravilno izbrano merilno sondo, ki je takšen tok ne bo okvaril.

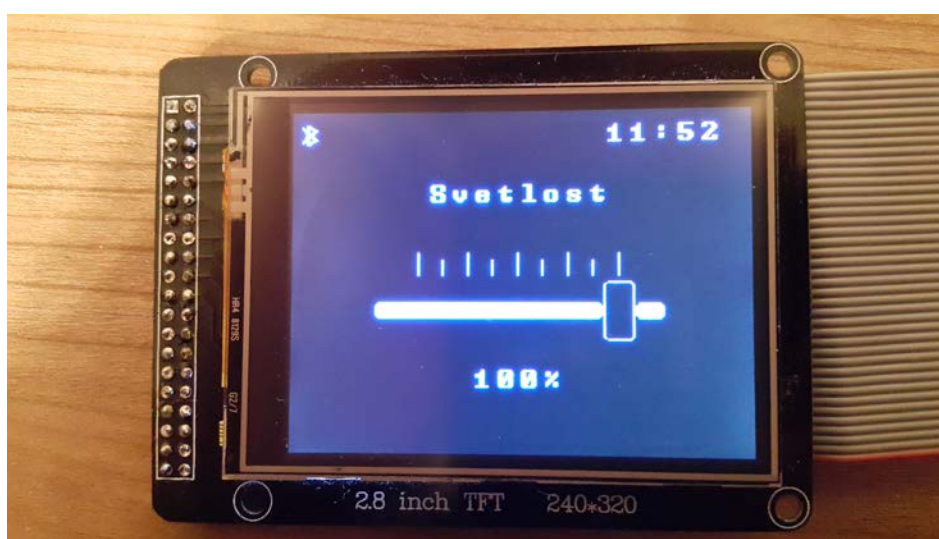
V prihodnosti se bo na tem mestu pojavil tudi merilnik življenjske dobe žarnice, ki nas bo opozoril, če bodo odpovedali žarometi.

## 5.5 PODMENI SVETLOST

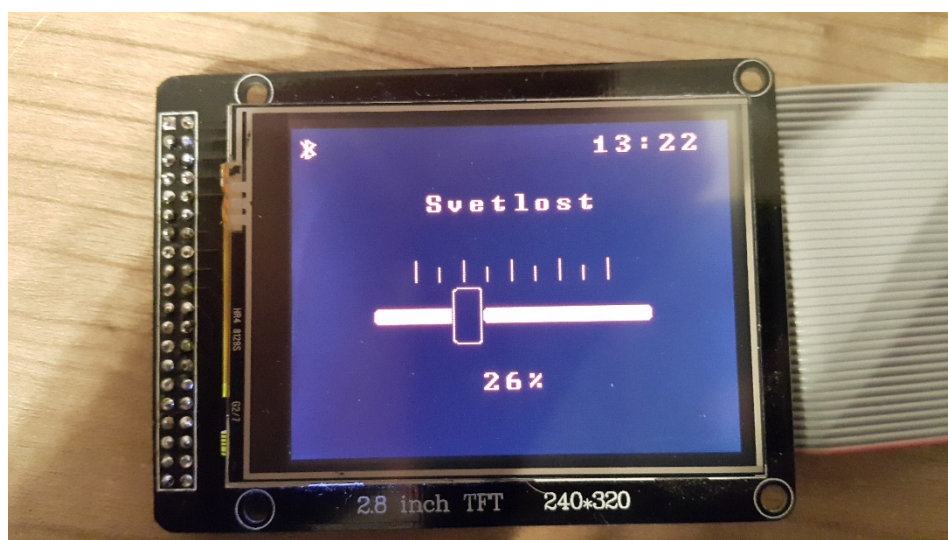
V podmeniju svetlost nastavljam svetlost RGB LED-diod na armaturi. Ko je uporabnik povezan na sistem s pametnim telefonom in preko aplikacije nastavlja svetlost, se tudi na ekranu drsnik posodablja glede na želeno svetlost.

V zgornji vrstici zopet vidimo stanje bluetooth povezave in uro. Drsnik in svetlostni podmeni nimata nobene druge funkcije kot vizualni prikaz drsnika. Pri tem grafičnem programiranju se stvar uteži takoj, ko imamo premikajoče se dele.

Ekran ima notranji spomin, na katerega se slika zapiše po slikovnih točkah in te je treba za novo sliko zlit z ozadjem oz. spremeniti v isto barvo.



Slika 49: Podmeni: Svetlost



Slika 50: Podmeni: Svetlost



## 5.6 PODMENI BLUETOOTH

V podmeniju Bluetooth se nahajajo informacije o uporabniku, ki je povezan in nastavitve za modul. S pritiskom statusa na OFF izklopimo BT-modul in nobena naprava se ne more povezati na naš sistem.

Pod povezavo se izpiše črka U, ki pomeni uporabnik, kadar je kdo povezan v sistem. Poleg črke U se izpiše še ime naprave. S pritiskom na gumb prekini pošlje procesor ukaz pametnemu telefonu, da izklopi povezavo in da se izpiše.

V prihodnje bo dodan še gumb za črni seznam (blokirana imena), časovno pa se bo omejila povezava uporabnika.

Ob prekinitvi signala telefona s sistemom si telefon za varčevanje energije sam izklopi BT-povezavo.



Slika 51: Podmeni: Bluetooth

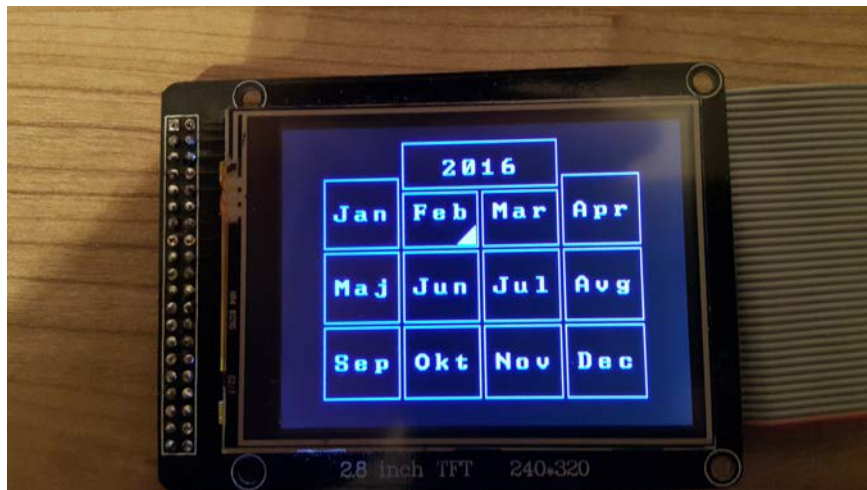


Slika 52: Povezovanje nove naprave preko Bluetooth

## 5.7 PODMENI KOLEDAR

V koledarju se razvija funkcija vnosa opomnika, ki bo uporabnika opozarjal na bodoče dogodke. Uporabnik bo zbral mesec, ki ga bo vnašal, nanj pritisne in doda nov vnos. Odpre se polno delujoča qwerty tipkovnica, v kateri uporabnik napiše opomnik. V naslednjem koraku izpolni datum in čas. Sistem začne opozarjati 24 h pred dogodkom. Na tako majhnem ekranu je aplikacija s tipkovnico malo nesmiselna, a za večji projekt je vsekakor koristna.

V prihodnosti bo ob dodatku GPS- in GSM-modulov lahko uporabnik nastavil še lokacijo, kje se dogodek nahaja. Preko GSM-modula bo prejel SMS-sporočilo o dogodku.



Slika 53: Podmeni: Koledar



Slika 54: Qwerty tipkovnica

## 6 VGRADNJA SISTEMA

### 6.1 OHIŠJA ZA TISKANINE

Tiskanine sem vgradil v vozilo tako, da sem jih najprej vstavil v ABS-škatlo in jih z vijaki pritrdil. Tiskanina, na kateri je ATMEGA328p čip, je zaradi pozicije v vozilu še protivodno zaščitena (IP65). Vstavil sem jo v škatlo z vsemi priklopi in vse luknje zapolnil s prozornim silikonom. To ohišje se nahaja pri akumulatorju, kjer najlažje meri zunanjo temperaturo, tok in napetost.

Procesorsko vezje potrebuje precej večje ohišje zaradi dvojne višine (ena tiskanina nad drugo) in dolžine. Izdelal sem še luknje za mikro USB-priključke za lažji dostop do tiskanine oz. do lažjega programiranja.

Ker je procesorska tiskanina v notranjosti avta, ni potrebno, da je še posebej zaščitena proti vlagi. Pomembno je le, da je ohišje pritrjeno na trdno in nevidno mesto in da je poskrbljeno za dovod el. napajanja.



Slika 55: Priprava tiskanin za v ohišje



Slika 56: Procesorsko ohišje

## 6.2 VGRADNJA SISTEMA V AVTO

Ker sem že vnaprej vedel, kako se razstavi konzola nad ventilacijo v avtu, je vgradnja zopet preprosta. Previdnost ni odveč, da ne izgubimo vijakov, ki držijo plastiko, in da kakšne stvari ne odlomimo. Nekatera vozila imajo varnostni sistem, ki avtomobilu ne pusti zagnati motorja, če ni na mestu vseh komponent. V mojem primeru tega ni bilo.

Procesor z ohišjem bo pritrjen pod armaturo, in sicer nad sovoznikovim predalčkom, tako da bo čisto neviden. V predalčku bo skozi luknjo USB-podaljšek, preko katerega lahko reprogramiramo procesor.



Slika 57: Prvi korak razstavljanja



Slika 58: Drugi korak razstavljanja

Na naslednjih slikah se vidi vgradnja ekrana in prvi test. Ob končni vezavi bo napajanje vzeto iz prejšnjega priklopa. Ta dovaja napajanje, ko odklenemo vozilo, prav tako pa še nekaj sekund po zaklepanju.

Napajanje RGB LED-diod v armaturi je vezano preko procesorja na 12 V, ki jih preklapljammo preko stikala za vžig luči.

Kot vidljivosti je idealen z obeh sedežev, barva pa je lepo vidna tudi podnevi.



Slika 60: Ekran v zeleni barvi



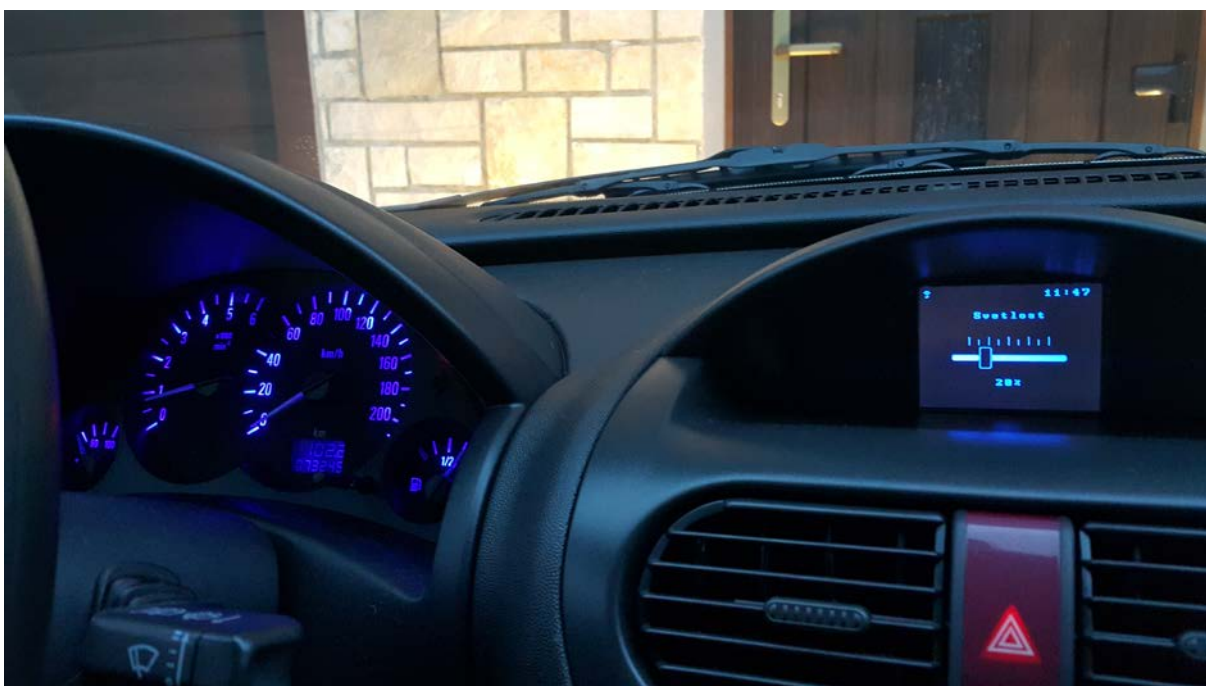
Slika 59: Tretji korak razstavljanja



Slika 61: Ekran in instrumenti v zeleni barvi



Slika 62: Ekran in instrumenti v modri barvi



Slika 63: Nastavitev svetlosti na ekranu in odziv svetlosti na instrumentih

## 7 ZAKLJUČEK

Naloga je zahtevala, da se naučim veliko novih stvari, novih načinov programiranja in tudi dela. Kot prvi model takšnega sistema brez prototipa se je naloga obnesla precej dobro. Nekaj zelenih ciljev nisem dosegel v določenem času, a projekta še ni konec. Sistem bom še nadgrajeval, saj imam zagon, ki žene vsakega radovedneža.

Sistem se v vozilu obnese in do sedaj še nisem imel večjih težav. Celotno nalogo in cel sistem sem izdelal sam, od programa do tiskanin. Cenovno bi svoj izdelek uvrstil zelo nizko, najdražja sta bila samo razvijalna plošča Arduino DUE in ekran na dotik. Takšen sistem bi na trgu segel krepko čez 200 €, sam pa sem ga naredil za šestino te vrednosti.

Da bi dosegel 100 % realizacijo hipotez, bi potreboval več časa za razvoj. V tej nalogi sem ovrigel hipotezo vnosa novih dogodkov na koledar in opomnik.

## 8 VIRI

[1] Arduino okolje. (spletni vir). 2015. (citirano 11. 2. 2016). Dostopno na:

<https://www.arduino.cc/>

[2] Arduino DUE-specifikacije. (spletni vir). 2015. (citirano 11. 2. 2016). Dostopno na:

<https://www.arduino.cc/en/Main/ArduinoBoardDue>

[3] Arduino UTFT-knjižnica. (spletni vir). 2015. (citirano 11. 2. 2016). Dostopno na:

<http://www.rinkydinkelectronics.com/library.php?id=51>

[4] Eagle CAD-program. (spletni vir). 2015. (citirano 11. 2. 2016). Dostopno na:

<http://www.cadsoftusa.com/>

[5] Merjenje napetosti z napetostnim delilnikom. (spletni vir). 2015. (citirano 11. 2. 2016). Dostopno na:

<http://arduinotronics.blogspot.si/2011/03/monitoring-voltage-of-dc-battery-supply.html>

[6] Spletni pretvornik slik v C-kodo. (spletni vir). 2015. (citirano 11. 2. 2016). Dostopno na:

[http://www.rinkydinkelectronics.com/t\\_imageconverter565.php](http://www.rinkydinkelectronics.com/t_imageconverter565.php)