

UNIVERZA V LJUBLJANI
FAKULTETA ZA ELEKTROTEHNIKO

POROČILO PRI PREDMETU SEMINAR IZ MEHATRONIKE

DIGITALNI MERILNIK VRILNE HITROSTI

Mitja Alič

Profesor:
prof. dr. Danijel Vončina

Mentor:
doc. dr. Mitja Nemec

Ljubljana, 2017

Kazalo

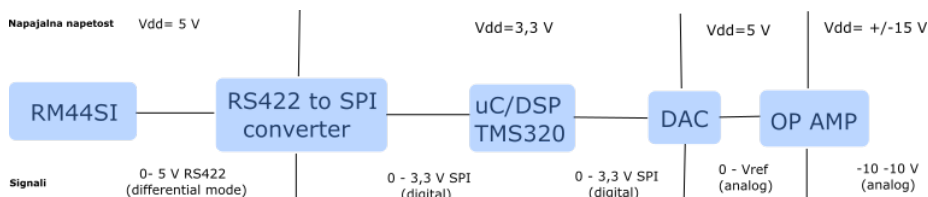
1	Uvod	2
2	Zahteve	2
3	Sestava sheme	3
4	Izdelava tiskanine	5
5	Programiranje	5
5.1	PWM enota	6
5.2	Povezava s senzorjem zasuka	6
5.3	Algoritem za izračun hitrosti	7
5.4	Povezava z digitalno analognim pretvornikom	7
5.5	Ostale težave	7
6	Pisanje poročila	8
7	Napake projekta	8
7.1	Moški "footprint" za ženski konektor	8
7.2	Priključek za napajanje	8
7.3	Pull-down upori na gate-u MOS-FETa	9
7.4	Napaka v povezavi senzorja z vezjem	9
7.5	Postavitev potenciometra poleg JTAG konektorja	9
7.6	Zamenjana vhoda na odštevalniku	10
7.7	Narobe konfigurirani pini na DAC-u	10
7.8	Napačna vrednost referenčne napetosti	10
8	Zaključek	10
9	Dodatek 1	11
10	Dodatek 2	14

1 Uvod

Pri predmetu seminar iz mehatronike sem si izbral projekt izdelave digitalnega merilnika vrtilne hitrosti. Pri delu na projektu, bom osvojil znanje risanje sheme ožičenja, risanja tiskanega vezja, izdelava vezja in testiranje. V povezavi s predmetom digitalno procesiranje v mehatroniki 2 (v nadaljevanju DP2), bom napisal program za določanje vrtilne hitrosti na podlagi poznavanja trenutnega kota zasuka. Cilj izdelanega vezja, je analogni prikaz vrtilne hitrosti na izhodu, katero bi lahko študentje opazovali na laboratorijskih vajah. Podobno seminarsko nalogo je imel Žiga Prajndl¹ leto prej, kjer je uporabil F28069 Piccolo controlSTICK. Piccolo se uporablja pri laboratorijskih vajah predmeta DP2. Na tem čipu naj izdelal salgoritem, nato pa ga prenesel na svojo tiskanino, kjer bi uporabil drug mikrokrmilnik.

2 Zahteve

Na začetku sem si prebral poročilo mojega predhodnika in njegove zahteve prilagodil. Osnovne zahteve so bile enake. Blokovna shema je na sliki 1. Tiskanina bo priključena na 5 V napajalni vir. Ostale napetosti so generirane s DC-DC pretvornikom. Za znane komponente sem pregledal podatkovne liste in začel graditi shemo.



Slika 1: Blokovna shema merilnika z vrednostmi napajalnih napetosti in velikostjo napetostnih signalov

Za preračunavanje in komunikacijo sem uporabil mikro krmilni TMS320F28027. Izvorno informacijo o kotu zasuka gredi elektromotorskega sklopa prejemo iz absolutnega dajalnika z zasuka podjetja RLS z oznako RM44SI0012B30F2E10. Izhod senzorja je lahko inkrementalni z referenčno točko, v obliki analognih potekov sinusa in kosinusa, ali preko serijske komunikacije. Komunikacija deluje v diferencialnem načinu, kar poveča zanesljivost prenosa. Mikrokrmilnik podpira komunikacijo SPI, zato je potrebno uporabiti gonilnik ki pretvori diferencialna signala v polariziran signal. Uporabil sem integrirano vezje SN65HVD73.

Na izhodni strani aplikacije želimo podatek o poziciji in posledično hitrosti našega pogona pošiljati naprej v analogni obliki za nadaljno regulacijo. Potrebna sta dva analogna izhoda zato sem uporabil TLV5618A. Gre za dvokanalni 12-bitni digitalno analogni pretvornik (DAC). Za višino referenčne napetosti Vref sem izbral napetost 4,096 V. Napetost sem generiral s LM4132BMF-4.1.

Tako komunikacija s senzorjem kot tudi z DAC-om zahteva SPI povezavo. Mikrokrmilnik ima le eno SPI enoto. Predhodnik je ugotovil, da lahko senzor in DAC poganja le z eno SPI enoto. Master-input slave-output (MISO) signal služi za prejetje podatkov s senzorja. Master-output slave-input (MOSI) signal pošlje podatke na DAC. Uro komunikacijskega modula je povezana na obe enoti.

¹Digitalni merilnik hitrosti, Žiga Prajndl, LRTME, Ljubljana, 2016

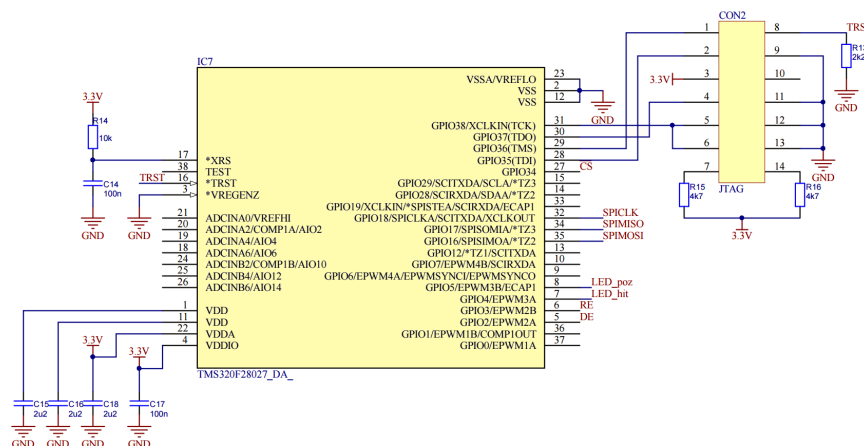
Za zagotovitev željenih izhodnih napetosti izhoda DAC-a ojačam s operacijskim ojačevalnikom. Zahteva mentorja je bila, da ima operacijski ojačevalnik čim manjšo sofazno ojačenje ter naj se vsi nahajajo v enem ohišju. Uporabil sem ojačevalnik OPA4192IPW. Potreboval sem bipolarno napajanje ± 15 V, ki sem ga generiral s pretvornikom TMH515D.

Prisotnost napajanja prikazujem na rumeni LED diodi, priključeni na osnovno napajalno napetost 5 V. Z zeleno LED diodo sem prikazoval trenutno vrednost senzorja. S spreminjanjem pozicija se je spreminjala svetilnost LED diode. To sem izvedel s PWM enoto in tranzistorjem serijsko vezanega s LED diodo. Svetilnost LED diode je tako odvisna od preklopnega razmerja PWM enote. Enako sem storil za rdečo led diodo, ki je predstavljala absolutno vrednost hitrosti.

3 Sestava sheme

Shemo sem delal v programu Altium Designer 16.1. Celotna shema se nahaja v Dodatek 1.

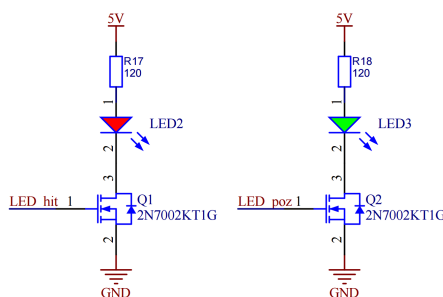
Mentor mi je poslal svoje starejše projekte, po katerih sem nato povezal pine mikrokrmilnika (Slika 2). Ob njemu je postavljen JTAG konektor. GPIO in PWM pine sem po mikrokrmilniku razporedil tako, da so mi na tiskanini čim boljše ustrezali.



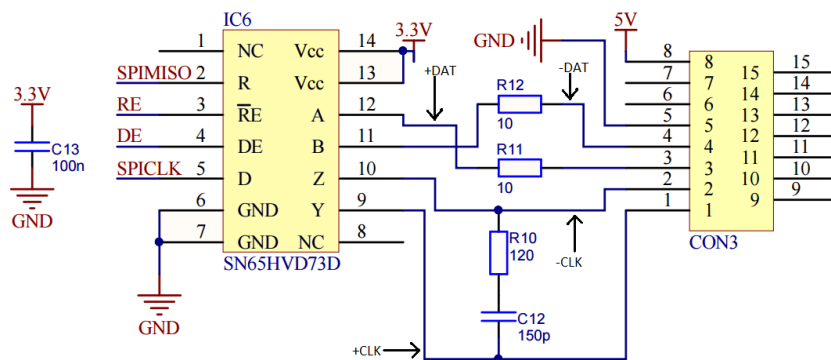
Slika 2: Shema mikrokrmilnika

PWM pina sem povezal s tranzistorjema (Slika 3), kjer nisem povezal upora na maso ("pull-down"). Tako je na gate-u MOS-FET-a lahko ob nepričakovanem dogodku (nedefiniran izhod PWM enote) ostal naboj in posledično je LED dioda svetila.

Pri povezavi pretvornika komunikacije je predhodnik naredil lapsus (zamenjal para ure in podatka na SN65HVD73). To sem zamenjal, nisem pa zamenjal tudi elemntov. Med izhodom senzorja zasuka mora biti paralelno vezan 120 Ω upor in manjši kondenzator. To pa sem obdržal med povezavama +CLK -CLK. Tekom projekta sem ugotovil, da je to napaka. Upor R10 in kondenzator C12 (Slika 4), bi se morala nahajati med +DAT in -DAT. Upora R11 in R12 bi morala biti na povezavi +CLK oz - CLK.

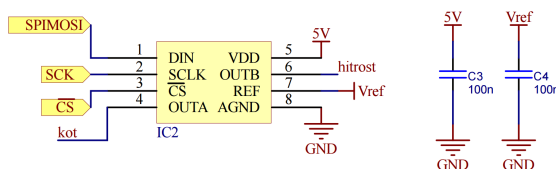


Slika 3: Shema krmiljenja LED diod



Slika 4: Shema komunikacije s senzorjem zasuka

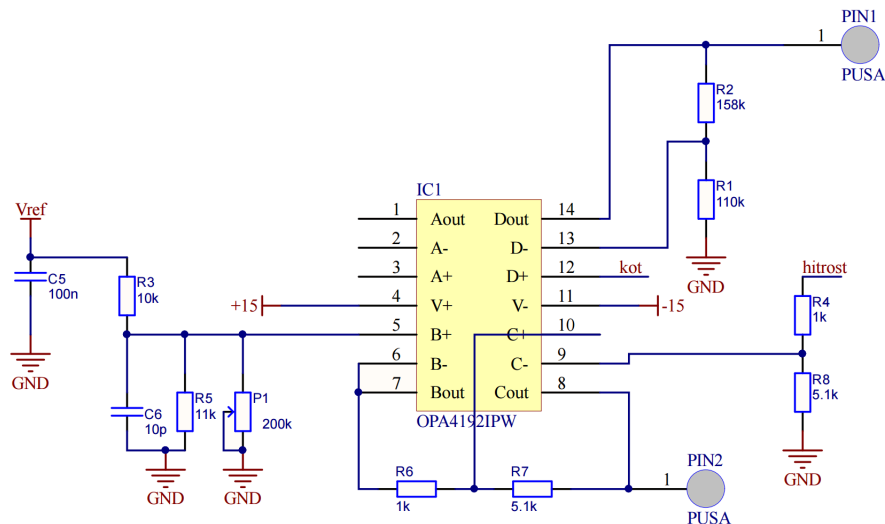
Vezava DAC-a je na sliki 5. Takoj na sliki se vidi, da sem pine označeval na levi strani od zgoraj navzdol in na desni prav tako. Ob izbiri "footprint-a" sem upošteval razvrščenost pinov na levi od zgoraj navzdol na desni pa od spodaj navzgor. Tako je bil "footprint" leve strani DAC-a v obratnem vrstnem redu. Huda napaka je predvsem to, da je masa DAC-a vezna na 5 V. Pini, vezani z ostalo periferijo lahko tako deformirajo druge dele vezja.



Slika 5: Shema DAC-a

Operacijske ojačevalnike bi moral le preisati, ojačenje pa zmanjšati za polovico. Sam sem sestavil le blok kot skupek 4 ojačevalnikov. Ob povezovanju sem naredil napako, da sem na ojačevalniku C zamenjal vhoda. Tako sem naredil pozitivno po-

vratno zanko in posledično vezje ni ustrezalo odštevalniku (Slika 6).



Slika 6: Shema operacijskega ojačevalnika

4 Izdelava tiskanine

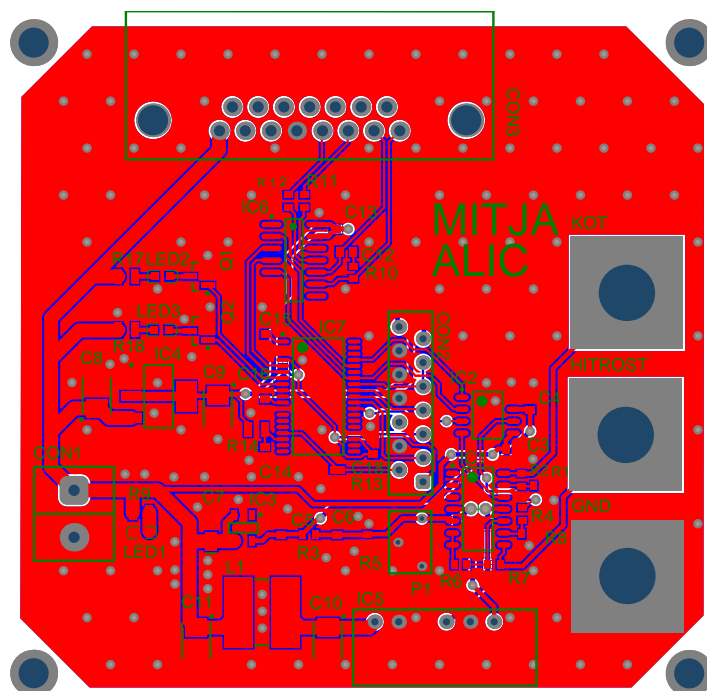
Ko sem shemo, po mojem mnenju zaključil, sem se lotil postavitve elementov na dvoplastno tiskano. Rezultat je na sliki 7.

"Footprint-e" sem nastavljal že na shemi, zato se s tem tu nisem ukvarjal. Vse napake na tiskani, so posledica nenatančnosti pri načrtovanju sheme. Konektor za napajanje sem izbral tako, da je bilo potrebno vanj vijačiti napajalni žici. Naslednik naj uporabi konektor, katerega lahko izklopi brez kakšnih potrebnih pripomočkov. Sam sem postavil na tiskano upore in kondenzatorje velikosti 0603². V laboratoriju LRTME ponavadi uporabljajo komponente velikosti 0805, zato jih je bilo potrebno na Farnell-u naročiti. Najprej sem s pomočjo Aleksandra Abramoviča naspajkal mikrokrmilnik, nato pa nadeljeval z upori ter kondenzatorji okoli mikrokrmilnika. Naspajkal sem vse razen napajalnika za napetost $\pm 15V$ in DAC-a. DAC-a ob naročanju materiala na Farnell-u ni bilo več na zalogi, zato sem ga naročil na Mouser-ju. Ko sem prvič priklopil vezje na napetostni vir, sem napajalno napetost počasi višal in opazoval vhodni tok. Še preden sem začel vezje programirati sem prevezal pina na ojačevalniku C, katera sem v shemi zamenjal. Preko Texas Instruments XDS100v2 USB Debug Probe sem priklopil vezje na računalnik in lahko začel programirati v Code Composer Studio.

5 Programiranje

Programirati sem začel v koncu februarja 2017. Projekt z začetnimi header datotekami, osnovnimi inicializacijami mikrokrmilnika in prekinitvijo mi je pripravil mentor. Potek programiranja sem si zadal po naslednjem seznamu:

²Imperialne enote, v metričnem sistemu je to 1608



Slika 7: Izgled tiskanine katera je šla v izdelavo

- Zaženi PWM enoto za krmiljenje LED diod
- Uredi komunikacijo s senzorjem zasuka
- Dodaj algoritem alfa beta filtra za izračun hitrosti
- Uredi komunikacijo z DAC-om

5.1 PWM enota

Za PWM enoti sem moral pina GPIO 4 in GPIO 5 nastaviti za PWM izhoda. Nastavil sem registre tretjega modula PWM, kateremu pripadata izhodna pina. Ob reset-u programa sta prikazovalni LED diodi prikazovali naključna stanja saj se je izhod PWM enote postavil v visoko impedančno stanje, pull-down upora na gate-u tranzistorja pa ni bilo.

5.2 Povezava s senzorjem zasuka

Pri predmetu DP2 sem dobil že napisano inicializacijo in zajem kota po SPI komunikaciji. Potrebni datoteki (SPI_dajalnik.h in SPI_dajalnik.c) sem dodal v projekt. Nastaviti sem moral še pina za krmiljenje čipa za spremembo komunikacije (DE na GPIO 2 in RE na GPIO 3). Nastavil sem ju na vrednosti tako, da je bila komunikacija ves čas mogoča. Ko sem želel preveriti delovanje se je hitro pokazalo nedelovanje. Za vrnjen podatek sem dobil le invertiran urin signal. To je bila posledica moškega "footprint-a" konektorja za povezavo s senzorjem. Zaradi zrcalne slike je bila žica

katera napaja senzor vezana na pin CLK+. Senzor se je s tem signalom le prižigal in ugašal. To napako sem rešil tako, da sem konektor odstranil z zgornje strani tiskanine in ga prispajkal na spodnjo stran. S tem posegom sem izhodne pine konektorja spravil na pravo mesto in komunikacija je delovala.

5.3 Algoritem za izračun hitrosti

Kodo za izračun vrtilne hitrosti sem napisal že pri vajah predmeta DP2. Vrnjen rezultat je bila frekvenca in ne vrtilna hitrost, ki je le za faktor 2π skalirana. V programu sem računal s plavajočo vejico (na procesorju TMS320f28069). Ko sem dodal to kodo v moj mikrokrmilnik, se kode ni uspela izvajati dovolj hitro. Algoritem sem zato pretvoril na računanje s fiksno vejico³. Algoritem se nahaja v Dodatek 2. Uporabil sem knjižnico IQ_math. Algoritem se tako izvede v dovolj kratkem času, časa pa nisem pomeril.

5.4 Povezava z digitalno analognim pretvornikom

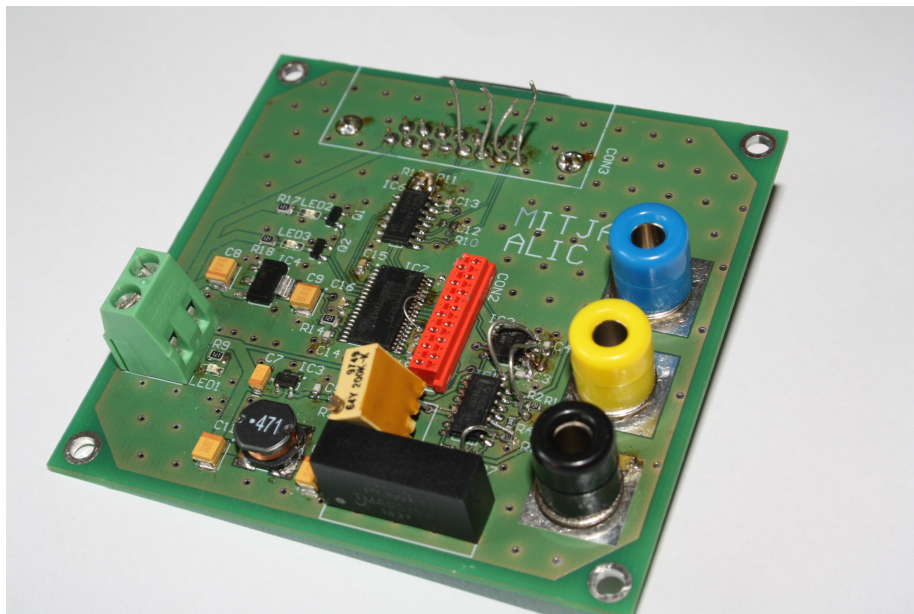
Ko sem dobil DAC in je bil algoritem za izračun hitrosti dokončan, sem naspajkal pretvornik na tiskanino. Ob priključitvi na napetostni vir, tiskanina ni izkazovala nobene napake. Zato sem se lotil nastavljanja izhodov za postavitev komunikacije. Za vzpostavitev komunikacije (chip enable), sem uporabil GPIO 34, kateri ima le to funkcijo. V "datasheet-u" nikjer drugje ni bil nič opisan. Vsi ostali pini imajo več funkcij, zato je bil vsak pin podrobneje opisan, GPIO 34 pa ima le to funkcijo in ne potrebuje drugega opisa. Ko bi moral postaviti GPIO 34 na 0 V je imel ta konstantno 5 V. Naj omenim, da je mikrokrmilnik napajan s 3,3 V in v razponu, od 0 V do napajalne napetosti, je lahko izhod. To se mi tu še ni posvetilo. Dvomil sem v pin GPIO 34 kateri ni bil podrobneje opisan v "datasheet-u". Zato sem na mikrokrmilniku prevezal tako, da sem chip enable pin DAC-a povezal z GPIO 19. Ta pin ima že funkcijo, da se postavi v želejeno stanje ob začetku SPI komuniciranja. Vendar ta prevezava ni pomagala. Mentor je nato ugotovil napako, ki je bila ustvarjena v shemi, da so pini DAC-a na levi strani zamenjani. Na tej točki je mentor predlagal, da s projektom končam in bo zapisan kot neuspešno izdelan. Nato sva ugotovila, da se lahko DAC še preveže v delujoče stanje.

Prevezal sem DAC in začelo je delovati. Po tem je bilo potrebno samo še skalirati in nastaviti vrednosti, da bo končni izhod napetosti hitrosti in kota med $\pm 10V$. Pojavila se je težava, da je izhod B DAC-a upadel za približno 1 V ko je potekala komunikacija. Tega nisem znal rešiti, slutim pa, da je bila napaka povzročena z dodatnimi prevezavami na tiskanini.

5.5 Ostale težave

Nepravilno se je začel obnašati tudi komunikacija s senzorjem zasuka. Kljub temu, da je senzor prejemal urin signal, ni vračal podatka. Podatek je vrnil če sem ob pravem trenutku spremenil vrednost DE pina na komunikaciji iz 0 na 1. To sem ugotovil, ko sem ponesreči povezal pin DE s pinom SPICLK. Vrednost na pinu DE je tako preklapljala med 0 in 1 in komunikacija je nato začela delovati. To sem poskušal nato doseči tudi s programom, tako da je preklapljal vrednost na pinu DE vendar se ni odzval vedno.

³Uporabil sem format f8.24



Slika 8: Tiskanina na koncu projekta

Po tednu iskanja rešitve za nastale težave sem se odločil in mentorju sporočil, da bi projekt raje zaključil saj traja že predolgo. Projekt ni bil tako kompleksen, vendar sem si ga z napakami zakompliciral sam.

6 Pisanje poročila

V poročilu sem morala zbrati vse napake, ki sem jih naredil pri projektu. Hotel sem posneti oscilogramе vendar sem ob tem skuril mikrokrmilnik. Zato oscilogramov v tem poročilu ne boste opazili.

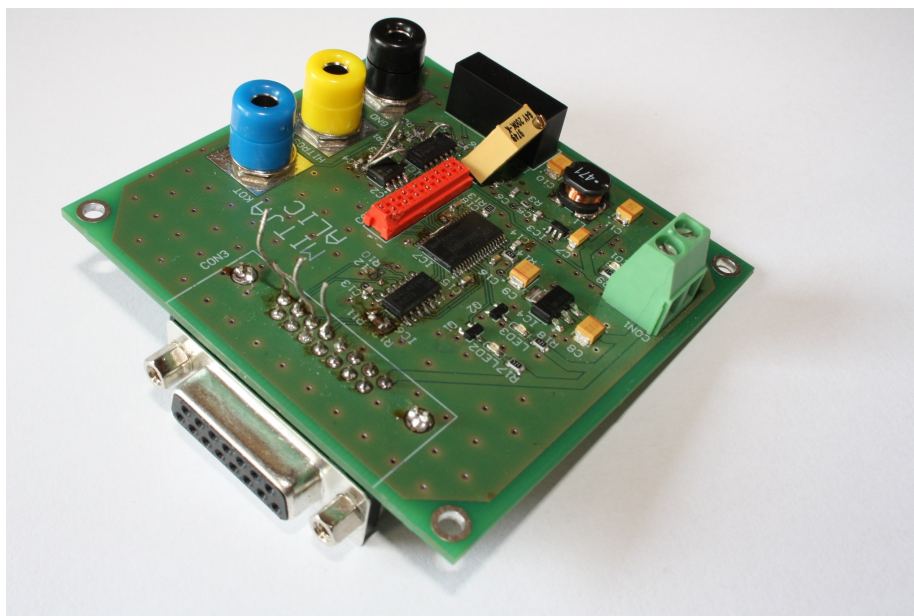
7 Napake projekta

7.1 Moški "footprint" za ženski konektor

Mentor me je velikokrat opozoril ali sem resnično uporabil pravi "footprint" za konektor, vendar tega ob sestavljanju sheme nisem nikoli preveril. Napaka se je pokazala ob vzpostavljanju komunikacije s senzorjem. Napako sem rešil tako, da sem konektor prispajkal s spodnje strani tiskanine(slika 9).

7.2 Priključek za napajanje

Ta napaka niti ni tako huda, je pa priporočljivo da se uporabi konektor katerega lahko odklopiš, brez izvijača. Konektor katerega sem uporabil se vidi na desni strani slike 9.



Slika 9: Konektor prispajkan na spodnji strani tiskanine zaradi anpake v "footprintu"

7.3 Pull-down upori na gate-u MOS-FETa

Pin mikrokrmilnika je bil direktno vezan na gate tranzistorja. Ob priključitvi napajalne napetosti so izhodi mikrokrmilnika v visoko impedančnem stanju. Tranzistor je v tem stanju lahko prevajal, lahko ne. To bi rešil upor vezan na maso in tako definiral stanje v primeru visoko impedančnega stanja pina na mikrokrmilniku. Prajndl je upore uporabil, jaz sem jih spustil.

7.4 Napaka v povezavi senzorja z vezjem

V poglavju 3, sem že opisal zamenjavo impedanc med paroma ure (CLK) in podatka (DAT). To napako sem našel na koncu, ko mi komunikacija s senzorjem ni delovala in sem se posvetil branju "datasheet-a" senzorja RM44SI. Opazil sem, da je potrebno pri izhodnih signalih senzorja vezati 120Ω upor in kondenzator, kar sem popravil. Še ena napaka, ki me je naučila naj dobro berem vse podatkovne listine.

7.5 Postavitev potenciometra poleg JTAG konektorja

Na sliki 8, prav tako na sliki 9 se vidi da potenciometer ni lepo prispajkan. Razlog je v tem, ker sem ga na tiskanini postavil preblizu konektorja JTAG. Ko sem želel povezati tiskanino z računalnikom, ni bilo prostora za nasaditev priključka. Potenciometer sem moral tako zamakniti, da sem lahko uporabil konektor. Ob končnem izdelku to niti ni napaka, saj konektor služi le za povezavo z računalnikom v času razvijanja programa. Ja pa to šola za naprej, da si ob konektorjih pustim dovolj prostora. Napaka pa je bila popolnoma nepotrebna, če bi bil dosleden.

7.6 Zamenjana vhoda na odštevalniku

Napaka, ki bi jo moral hitro opazi že na shemi. Povezavi na vhodu ojačevalnika C sem obrnil. S tem sem naredil pozitivno povratno zanko. To se je dalo popraviti s prevezavo na tiskanini.

7.7 Narobe konfigurirani pini na DAC-u

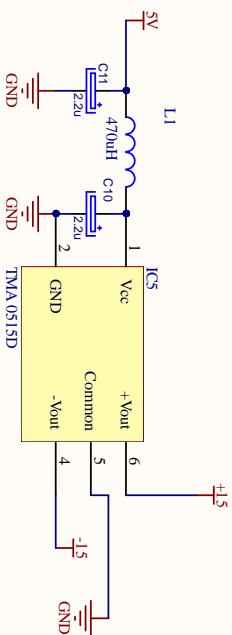
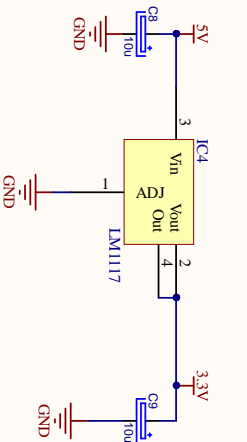
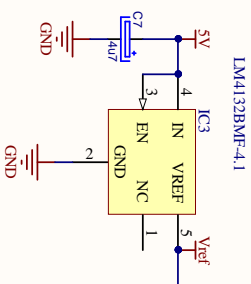
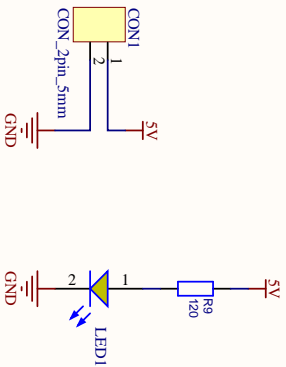
Na DAC-u sem levi strani pinov pripisal napačne "designatorje". Na levi strani sem jih oštevilčil od zgoraj navzol. Na "footprint-u" pa oštevilčenje pinov na levi strani, poteka od spodaj navzgor. To napako bi lahko popravil če bi spremenil oznake "designatorjev", ali še lažje v nastavitvah Pin Map bi lahko posameznemu pinu na shemi priredil pin na "footprint-u".

7.8 Napačna vrednost referenčne napetosti

Ob koncu projekta sem ugotovil, da moja referenčna napetost (4,096 V), ni bila najprimernejša. V "datasheet-u" DAC-a sem opazil kasneje, da je najprimerneje uporabiti vrednost referenčne napetosti 2,048V. Najvišja referenčna napetost pa naj nebi bila višja od $V_{DD} - 1,5$ V. Ni ključna napaka je pa prav, da jo omenim.

8 Zaključek

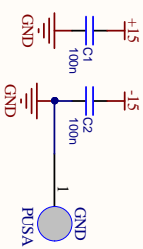
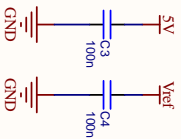
V projektu sem ustvaril lepo zbirko napak. Več ali manj jih je bilo proizvedenih med izdelavo sheme, katere sem se lotil zelo lahkotno. Ta lahkotna misel pri sestavi sheme, me je pripeljala do neuspešnega zaključka. S tem projektom sem se po mojem mnenju na nekoliko boleč način npridobil dosti več novega znanja in bom zato v prihodnosti bolj dosleden.



9 Dodatek 1

* Title				*
Size: A4	Number:*	Revision:*	* * *	
Date: 3/14/2017	Time: 1:00:00 PM	Sheet * of *	* * *	
File: C:\Users\Milica\Documents\5_tehniki\Seminarij\Meritnik_hirosfi\Napajanje_SchDoc				





10 Dodatek 2

Algoritem alfa beta filtra sem sestavljal po predlogi iz magistrskega dela Repetitivna regulacija hitrosti sinhronskega stroja s trajnimi magneti, Denisa Sušina, Univerza v Ljubljani, 2016 stran 39,40.

```
kot_abf+=_IQmpy(f_abf,dt); //kot_abf je spremenljivka ki ima vrednost
                           //ze iz prejsnje prekinitve
if(kot_abf>_IQ(1.0)) // kot_abf je v p.u. zato se nahaja med 0.0 in 1.0
{
    kot_abf-=_IQ(1.0);
}
if(kot_abf < 0 )
{
    kot_abf +=_IQ(1.0);
}

epsilon= (kot_iz_senzorja<<12)- kot_abf; //kot_iz_senzorja je 12 biten
                                           //pretvorim v f8.24 p.u.
if(epsilon >= _IQ(0.5)) // odprava prehoda iz 0.9999 na 0
{
    epsilon-=_IQ(1.0);
}
if(epsilon <= _IQ(-0.5) )
{
    epsilon +=_IQ(1.0);
}

kot_abf+=_IQmpy(alpha,epsilon); //definicija popravljivega kot_abf
                                //za naslednjo prekinitvev

f_abf+=_IQmpy(epsilon,_IQdiv(betha,dt)); //definicija popravljivega
                                           // f_abf za naslednjo prekinitvev
```