

ROBOTIZIRAN ROČNI MENJALNIK IN
PROGRAMIRANJE TURBINE

STROJNIŠTVO

RAZISKOVALNA NALOGA

DAVID KRANJC, S4D

MENTOR: NEJC MARINŠEK

2023/2024

ŠOLSKI CENTER LJUBLJANA
SREDNJA ŠOLA ZA STROJNIŠTVO,
KEMIJO IN VAROVANJE

Ključne besede: Menjalnik
Robotiziran menjalnik
Pnevmatika
Krmilna enota
Turbina

POVZETEK

Menjalnik je v vozilih zelo pomembna naprava, ki omogoča učinkovito vožnjo pri različnih hitrostih. Ročne menjalnike je možno robotizirati, kar pomeni, da vozniku ni potrebno menjati prestav s prestavno ročico, ampak z obvolanskimi stikali in to precej hitreje. V prvem delu naloge je opisana robotizacija menjalnika, ki ga upravljamo s pomočjo pnevmatike in elektronike. Celoten sistem je sestavljen iz mehanskega dela, krmilja in računalniškega programa. K povečanju moči in navora motorja z notranjim izgorevanjem pripomore tudi turbopolnilnik, ki mu prisilno vpihuje dodaten zrak. V drugem delu naloge je opisano krmiljenje turbopolnilnika in ostalih elektronskih komponent pogonskega sistema.

KAZALO

1	UVOD	1
1.1	Robotiziran menjalnik	1
1.2	Krmiljenje turbo polnilnika	1
2	HIPOTEZE	2
3	MENJALNIK	3
3.1	Delovanje in tipi	3
3.2	VW/AUDI 5 stopenjski ročni menjalnik	3
4	SISTEM ROBOTIZIRANEGA MENJALNIKA	5
4.1	Pridobivanje stisnjenega zraka	5
4.2	Modul na menjalniku	6
4.2.1	Način gibanja	6
4.2.2	Izbira valjev	10
4.2.3	Krmiljenje	12
4.3	Modul na sklopki	13
4.3.1	Zaznavanje stopala	15
4.3.2	Varnostni sistem	16
4.4	Obvolanske ročice	16
5	ELEKTRONSKI SISTEM MENJALNIKA	17
5.1	Mikrokrmilnik	17
5.1.1	Razširitev vezja	17
5.1.2	Releji	18
5.2	Induktivno stikalo	19
5.3	Zaslon	19
5.4	Program	21
5.5	Varnost	22
6	KRMILJENJE MOTORJA IN TURBINE	23

6.1	Računalnik motorja	23
6.1.1	Senzorji	23
6.2	Nadzor turbine.....	24
6.2.1	MAP senzor.....	24
6.2.2	Ventil za prepihanje N75.....	24
6.2.3	Izpihovalni ventil.....	25
6.3	Integracija prestav z motorjem.....	25
7	REZULTATI.....	26
8	ZAKLJUČEK.....	28
9	VIRI.....	29
10	PRILOGE	31

KAZALO SLIK

Slika 3.1: Prestavna os	4
Slika 3.2: Razstavljen kompresor klime	5
Slika 4.1: Matrica prestavne osi	6
Slika 4.2: Shema pnevmatskih valjev [3]	7
Slika 4.3: 3D model rotacijskega mehanizma	7
Slika 4.4: Rotacijski mehanizem	8
Slika 4.5: 3D model linearnega mehanizma	8
Slika 4.6: Modul na menjalniku	9
Slika 4.7: Gumijasti blažilec [4]	10
Slika 4.8: Predelava pnevmatskih cilindrov	11
Slika 4.9: Merjenje hitrosti valjev	11
Slika 4.10: Graf hitrosti cilindrov	12
Slika 4.11: Graf odziva krmilnega vezja	12
Slika 4.12: Končan prestavni modul I	13
Slika 4.13: Končan prestavni modul II	13
Slika 4.14: 3D model modula na sklopki	14
Slika 4.15: Dušilni ventil	15
Slika 4.16: Prototip stikala na stopalki	15
Slika 4.17: Obvolanske ročice (F1 prestave)	16
Slika 5.1: Arduino Uno R4	17
Slika 5.2: Integrirano vezje MCP23017	18
Slika 5.3: Modul mehanskih relejev	18
Slika 5.4: Induktivno stikalo	19
Slika 5.5: Zaslon	20
Slika 5.6: Krmilnik zaslona z upori	20
Slika 5.7: Diagram delovanja programa [5]	21
Slika 6.1: Krmilni računalnik motorja (zaščiten)	23
Slika 6.2: Izpihovalni ventil	25

SEZNAM UPORABLJENIH OZNAK IN KRATIC

CVT	(Continuously Variable Transmission) – brezstopenjski menjalnik
VAG	Volkswagen & Audi Group
I2C	Način žične komunikacije
SSR	(Solid State Relay) – Polprevodniški rele
LED	(Light Emmiting Diode) – LED sijalka
PWM	(Pulse Width Modulation) – Pulzno širinska modulacija
IDE	(Integrated Development Environment) – Integrirano razvojno okolje

KAZALO ENAČB

Enačba 4.1: Izračun površine valja.....	10
Enačba 4.2: Izračun premera valja	10
Enačba 5.1: Izračun potrebne napetosti LED sijalk	19
Enačba 5.2: Izračun upornika	20

UPORABLJENI SIMBOLI

A	mm^2	Površina
F	N	Sila
p	MPa	Tlak
d	mm	Premer
U_s	V	Skupni padec napetosti
U_1	V	Padec napetosti ene diode
U	V	Napetost
I	A	Električni tok
R	Ω	Električni upor

SEZNAM PRILOG

Priloga 1: Program robotiziranega menjalnika.....	31
Priloga 2: Tabele krmilnega računalnika motorja.....	36

1 UVOD

1.1 Robotiziran menjalnik

V prvih poglavjih raziskovalne naloge je predstavljeno delovanje avtomobilskih menjalnikov ter bolj podrobno ročnega menjalnika. Tega se je predelalo v robotiziran ročni menjalnik s pomočjo pnevmatskih valjev. Skonstruirati je bilo potrebno celoten sistem mehanizmov, krmilnih enot, računalniških vezji, programa in varnostnih sistemov. Predstavljeni bodo problemi na katere naletimo med konstruiranjem in ustrezne rešitve oz. optimizacije samega sistema.

Cilj prvega sklopa je uspešno in varno krmiliti menjavo prestavnega razmerja v avtomobilu.

1.2 Krmiljenje turbo polnilnika

Drugi del naloge prikazuje krmiljene sodobnega računalnika za nadzor motorja z notranjim izgorevanjem. Del sistema je tudi turbina, ki motorju s pomočjo prisilnega vpihovanja zraka znatno poveča moč. To je potrebno skrbno nadzorovati in krmiliti, saj lahko z nepravilnim in nepredvidljivim delovanjem pride do katastrofalnih posledic za motor in voznika.

Cilj drugega sklopa je s pomočjo avtomobilskega turbopolnilnika motorju zanesljivo in predvsem varno povečati končno moč. Pri tem učinkujejo različni krmilni ventili in senzorji, ki v povezavi z računalnikom motorja pripomorejo k bolj učinkovitem delovanju celotnega sistema. Potrebno je najti pravilno razmerje med željeno močjo motorja in mejo varnosti.

Na koncu je predstavljena še povezava robotiziranega menjalnika z računalnikom motorja, ki pripomore k mirnejši vožnji, manjši obrabi sklopke in zobnikov v menjalniku ter hitrejšemu menjanju prestav.

2 HIPOTEZE

- I. Za nizko ceno lahko ročni menjalnik v avtomobilu sami robotiziramo.
- II. Menjanje prestav z robotiziranim pnevmatskim menjalnikom je hitrejše od menjanja prestav z ročnim menjalnikom.
- III. Sistem robotiziranih prestav je primeren za vsakdanjo rabo.
- IV. Nadzorovanje turbopolnilnika je s pomočjo krmilne enote motorja enostavno in zanesljivo.
- V. Krmilna sistema menjalnika in motorja lahko povežemo in dosežemo boljše rezultate.

3 MENJALNIK

3.1 Delovanje in tipi

Menjalnik je mehanska naprava s katero spreminjamo prestavna razmerja strojem in motorjem. S tem stroju omogočimo regulacijo izhodne hitrosti in navora ter smeri vrtenja.

V avtomobilskem svetu poznamo:

- Ročne menjalnike: prestave spreminjamo s fizičnim potiskom prestavne ročice,
- robotizirane menjalnike: na željo voznika, sistem z mehanizmi sam spremeni prestavno razmerje,
- avtomatske menjalnike: sistem sam določa, kdaj bo spremenil prestavno razmerje,
- CVT menjalnike: s posebno obliko omogoča brezstopenjsko regulacijo hitrosti in navora [1].

V Sloveniji tudi v današnjih časih še vedno prevladujejo ročni menjalniki, predvsem zaradi nižje cene in nižjih stroškov popravil. Kljub najnovejšim tehnologijam, ki avtomatske menjalnike nudijo po zelo ugodnih cenah, je na trgu približno polovica novih vozil še vedno na voljo z nameščenimi ročnimi menjalniki. Po vsej verjetnosti je hitrost naraščanja števila avtomatskih menjalnikov padla, tudi zaradi električnih vozil, kjer so se proizvajalci menjalnika povsem znebili.

3.2 VW/AUDI 5 stopenjski ročni menjalnik

Za ročni menjalnik, ki ga namenimo pnevmatski robotizaciji si izberemo VAG 5 stopenjski menjalnik s kodo DHW [2]. Narejen je bil leta 1998 in zaradi redne menjave olja še danes deluje brezhibno. Vgrajen ima senzor za vzvratno vožnjo, ki ob zaznavi vzratne prestave prižge belo luč na zadku avtomobila. Poleg tega, ima vgrajen mehanizem, ki preprečuje vozniku, da bi po nesreči iz pete prestave menjal direktno v vzvratno.

Sestavljen iz 10 poševnih zobnikov (en par za vsako od petih prestav), ki imajo ob vrtenju nizek nivo hrupa in 3 ravnih valjastih zobnikov, ki so namenjeni vzratni vožnji. Poševni zobniki so precej močnejši od ravnih, saj se površina naleganja zobov močno poveča. Ker je bil izbran menjalnik uporabljen v velikem številu vozil z različnimi močmi smo lahko prepričani, da bo prenesel vse napore s katerimi ga bomo s povečanjem moči obremenili.

Gibanje zobnikov, posledično izbranih prestav, omogoča ena os, locirana na zadnjem delu menjalnika.



Slika 3.1: Prestavna os.

Prestavna os je običajno neposredno povezana s prestavno ročico v kabini avtomobila. Njeno premikanje je omejeno na 3 pozicije v smeri notranjosti in izven menjalnika ter na 3 pozicije rotacije okoli središčne osi. Kot je razvidno s slike 3.1 se omenjena os nahaja zelo blizu ohišja menjalnika, zato bo njeno krmiljenje precej zahtevno.

4 SISTEM ROBOTIZIRANEGA MENJALNIKA

4.1 Pridobivanje stisnjenega zraka

Predstavljeni sistem za svoje delovanje potrebuje precejšnje količine stisnjenega zraka. Pri tem pomaga avtomobilski kompresor za klimatsko napravo, ki ga je potrebno najprej predelati. Po izpraznitvi plina klimatske naprave odstranimo gonilni jermen in kompresor odvijamo od motorja. Na čisti površini se je kompresor razstavilo in raziskalo kako deluje. Cilj je zatesniti prostor v katerem se nahajajo pogonski mehanizmi in ležaji, od prostorov, kjer zrak stiska 5 okroglih batov.



Slika 3.2: Razstavljen kompresor klime.

Po zatesnitvi štirih zelo majhnih prekatov, ki povezujejo pogonsko in kompresijsko komoro, se je previdno sestavilo celotno enoto in pri tem pazilo, da so pravilno nameščena vsa potrebna tesnila. Pred dokončnim privitjem pogonskega dela kompresorja, se je vanj vbrizgalo mast, ki se bo pri obratovalni temperaturi utekočinila in skrbela, da bodo premikajoče se komponente ustrezno podmazane. Nekaj te masti sicer prodre skozi tesnila batov kompresorja, vendar je to za sistem zdravo, saj ta pri potovanju skozi cevi podmazuje krmilne ventile in valje.

Stisnjen zrak je shranjen v dveh zalogovnikih narejenih iz uporabljenih gasilnih aparatov s certifikatom, ki potrjuje, da lahko posodi zadržita ustrezen tlak (največ 10 bar). Večja posoda se nahaja tik ob kompresorju zraka (zadaj), manjša pa v sprednjem delu avtomobila, saj preprečuje prevelike padce tlaka v sistemu. S tem je dosežen boljši in predvsem hitrejši

pretok zraka tudi komponentam, ki so precej oddaljene od glavnega zalogovnika.

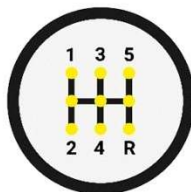
Na obeh tlačnih posodah sta nameščena varnostna ventila, ki se pri tlaku večjem od 10 bar odpreta in sprostita nepotreben zrak v ozračje. Ta izpust je precej glasen, zato lahko voznik takoj prepozna napako, ugasne kompresor in varno izstopi iz vozila.

Tlak nadzoruje tlačno stikalo, ki se vklopi ob padcu tlaka pod 6 bar in izklopi, ko ta doseže 8 bar. Takoj, ko to zazna, pošlje 12 V signal skozi nadzorno stikalo locirano pri vozniku v elektromagnetno sklopko, ki vklopi kompresor. Ta deluje in stiska zrak vse do trenutka, ko senzor zazna zadosten tlak in prekine s pošiljanjem signala, kar izklopi elektromagnet.

4.2 Modul na menjalniku

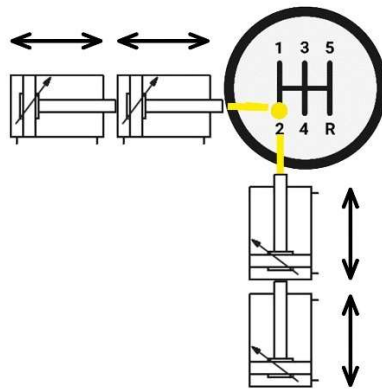
4.2.1 Način gibanja

Cilj modula na menjalniku je s čim manjšim številom pnevmatskih cilindrov/valjev nadzorovati vse možne pozicije prestavne osi. Zaradi ugodne postavitve prestav izbranega menjalnika si lahko polje, ki ga nadzorujemo, predstavljamo v 3x3 matrici:



Slika 4.1: Matrica prestavne osi.

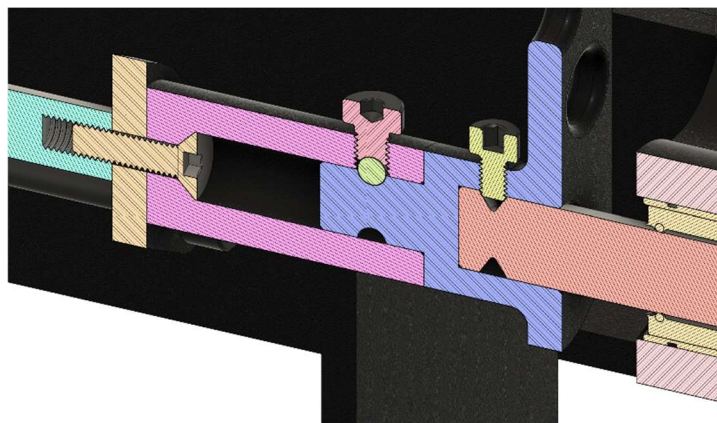
Gibanje je lahko torej razčlenjeno na vodoravno in navpično os, za kateri je potrebno zagotoviti tri možne pozicije. Za vsako os se to izvede tako, da se zaporedno veže dva dvosmerna pnevmatska cilindra. Vodi se jih z gladkimi vijaki v kotih ohišja, ti pa služijo tudi kot omejevalci hoda, ki omogočijo fine nastavitve pozicije osi menjalnika. Shema je narisana v dvodimenzionalnem prostoru, kot prikazuje slika 4.2.



Slika 4.2: Shema pnevmatskih valjev. [3]

Prenos teh dveh gibanj na eno samo gred je precej zapleten. Zagotoviti je potrebno, da bo gibanje osi izven in proti menjalniku popolnoma neodvisno od rotacije iste osi. To se reši z dvema posebnima mehanizmoma:

- Prvi omogoča vzdolžni premik ne glede na rotacijo izhodne komponente. To se doseže s pomočjo stružnice - strojem, ki omogoča izdelovanje okroglih izdelkov z nožem za rezanje kovin. V jeklen surovec premera 10 mm s posebnim radijskim (krožnim) nožem izrežemo utor, kamor se natančno prilega kaljena jeklena kroglica premera 4 mm. Ta je varovana z vijakom in omogoča prosto vrtenje izhodnega elementa,

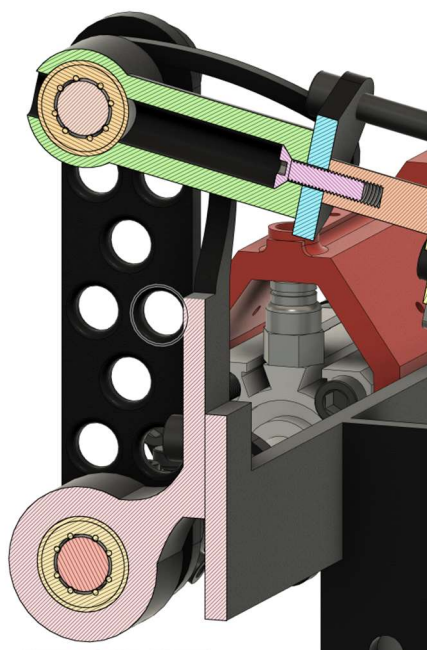


Slika 4.3: 3D model rotacijskega mehanizma.



Slika 4.4: Rotacijski mehanizem.

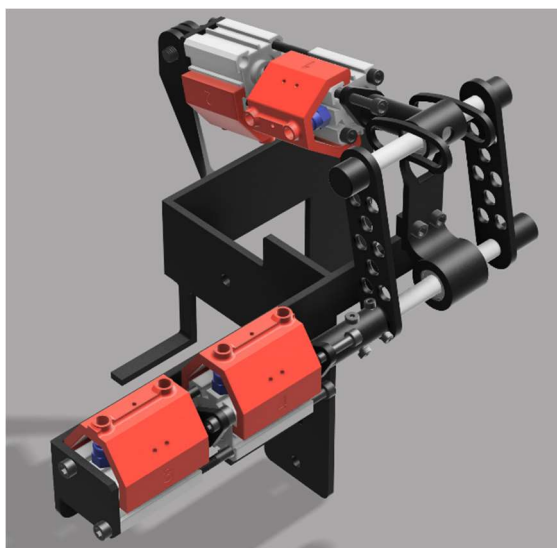
- Drugi mehanizem omogoča nadzorovano rotacijo, ki je neodvisna od linearnega premika. Izdelan je s pomočjo dveh poliranih gredi in linearnih ležajev. Ti omogočajo nadzorovano rotacijo prestavne osi, ki je neodvisna od vzdolžnega pomika.



Slika 4.5: 3D model linearnega mehanizma.

Končani modul se z vsemi elementi najprej konstruira v programu za prostorsko modeliranje ter nato izdelava iz ploščatega jekla, polne okrogle jeklene palice in vijakov.

Za izdelavo nekaj okroglih delov se uporabi stružnico, za vse ostale dele pa kotno brusilko, vrtalni stroj, varilni stroj in različne pile. Čim več kosov privijamo in s tem omogočimo hitrejše in predvsem lažje sestavljanje in razstavljanje. Ker v modulu nastopajo precej veliki momenti in izmenične sile (tudi nad 300 N), mora biti konstrukcija precej toga in vzdržljiva, hkrati morajo imeti premikajoče se komponente, zaradi vztrajnostne sile, čim manj mase. To se doseže tako, da se v dele, ki ne prenašajo velikih sil in momentov, zvrta čim več lukenj in se s tem znebi nepotrebnega materiala.



Slika 4.6: Modul na menjalniku.

Celoten modul je potrebno trdno pričvrstiti na menjalnik, saj bo sila pnevmatičnega valja celoten mehanizem odpravila od menjalnika. Pri običajni zvezi z vijaki se pojavi težava zaradi ekstremnih vibracij celotnega motorja, ki bi po vsej verjetnosti precej hitro uničile celoten sklop robotiziranih prestav. Vijake se nadomesti z gumijastimi blažilci (slika 4.7), ki močno zmanjšajo količino vibracij prenesenih na prestavni modul.



Slika 4.7: Gumijasti blažilec. [4]

4.2.2 Izbira valjev

Z merilnikom sile je bilo izmerjeno, da je za premik prestavne osi v povprečju potrebna sila 40 N. Izbrati je potrebno premer pnevmatskega cilindra, kateri bo zadostoval moči upora osi. Cilj je izbrati pravilno ravnotežje med silo in hitrostjo batnice. S silo lahko na začetku pretiravamo, ter kasneje, glede na praktične rezultate, z regulatorjem znižamo tlak v sistemu. Predvidi se, da se potrebuje 5-krat večjo silo, torej 200 N. Izračun potrebne površine bata cilindra pri tlaku 6 bar:

$$A = \frac{F}{p}$$

F – sila [N]

p – tlak [MPa]

A – površina [mm²]

Enačba 4.1: Izračun površine valja

S pomočjo enačbe 4.1 je bil dobljen rezultat; površino 333,3 mm², katerega nato uporabimo za izračun premera valja.

$$d = \sqrt{\frac{4A}{\pi}}$$

d – premer valja [mm]

π – Arhimedova konstanta (3,142)

Enačba 4.2: Izračun premera valja

Takoj za preračunom potrebnega premera valja (enačba 4.2), ki znaša 20,60 mm, se izbere

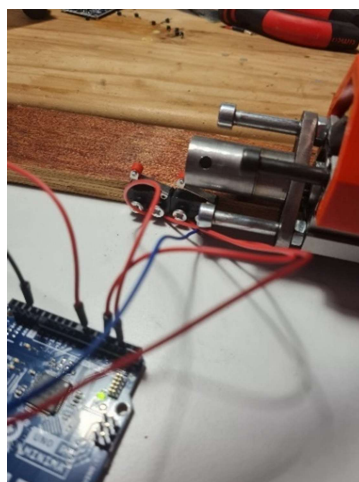
prvi večji standardni premer cilindra 25 mm, ki nam zagotovi potrebno silo.

Problem se pojavi pri hitrosti iztega, na kar zelo vpliva premer cevi, po katerih teče zrak. Ta se skozi celoten sistem giblje med 4 in 5 mm, vendar se pri vходу cilindra pojavi ozko grlo celotnega sistema; premer 1 mm. To drastično upočasni gib cilindra. Izvrtino je bilo potrebno povečati, vendar je bilo zaradi tanke stene ogrodja nemogoče vrezati nove navoje za priključke. Problem se reši tako, da se priključke na cilindru prilepi z dvokomponentnim lepilom ter dodatno zavaruje s 3D tiskanimi ojačitvami, ki pritrdijo cevi.

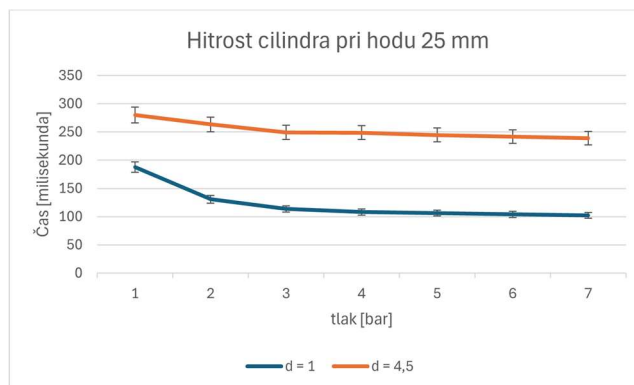


Slika 4.8 : Predelava pnevmatskih cilindrov.

Hitrosti aktuatorja z majhno in povečano izvrtino se testira s preprosto pripravo, zgrajeno iz dveh stikal (na začetku in na koncu) ter mikrokrmilnikom Arduino® tako, da se meri čas med pritiskom obeh stikal. Na konec cilindra se doda nekaj mase, ki simulira upor osi na menjalniku.



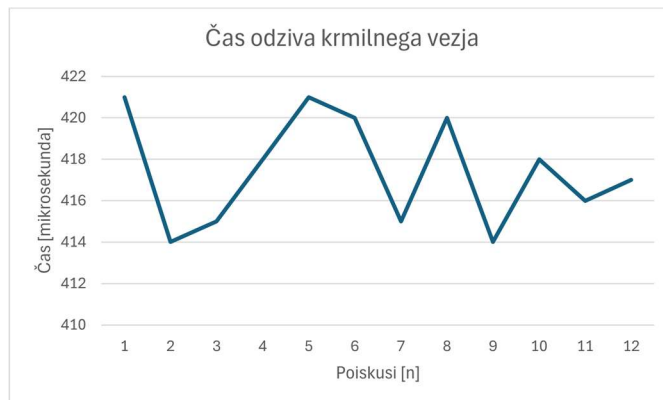
Slika 4.9: Merjenje hitrosti valjev.



Slika 4.10: Graf hitrosti cilindrov.

4.2.3 Krmiljenje

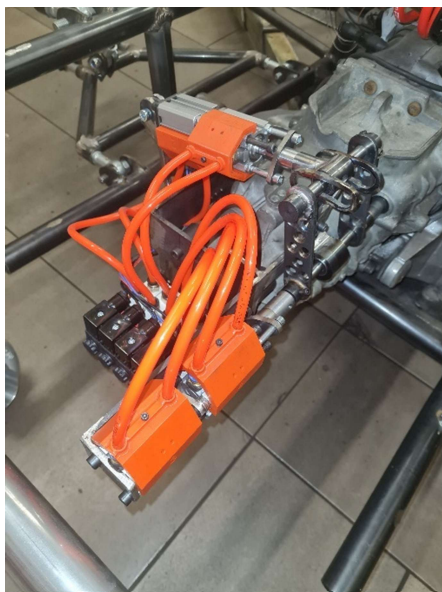
Vsak cilinder nadzoruje po en 5/2 monostabilni elektromagnetni ventil (slika 4.13), ki ob prejetem signalu 12 V spremeni svoje stanje in posledično pozicijo cilindra. Čas menjave stanja se izmeri s podobnim testom kot v prejšnji točki, le da se začne šteti čas, ko je iz računalnika poslan signal in meritev se ustavi takrat, ko se cilinder začne premikati in povezi prvo stikalo. Izmerjene količine so prikazane na grafu (slika 13).



Slika 4.11: Graf odziva krmilnega vezja.

Iz podatkov z grafa se dobi količino časa potrebnega za to, da lahko mikrokrmilnik pošlje signal, fizično preklopi rele, energizira tuljavo na ventilu, fizično premakne ventil ter po 20 cm dolgih ceveh pošlje zrak v cilinder. Ta v povprečju znaša zgolj 417 mikrosekund ($\sim 0,0004$ sekunde) z odstopanjem le 7 mikrosekund. Obe vrednosti sta v primerjavi z ostalimi zamudami astronomsko majhni, zato sta zanemarljivi.

Vsi štirje krmilni ventili so priviti na modul, kjer so nameščeni cilindri, dodane so pnevmatične cevi in vse skupaj je z blažilci pritrjeno na menjalnik.



Slika 4.12: Končan prestavni modul I.



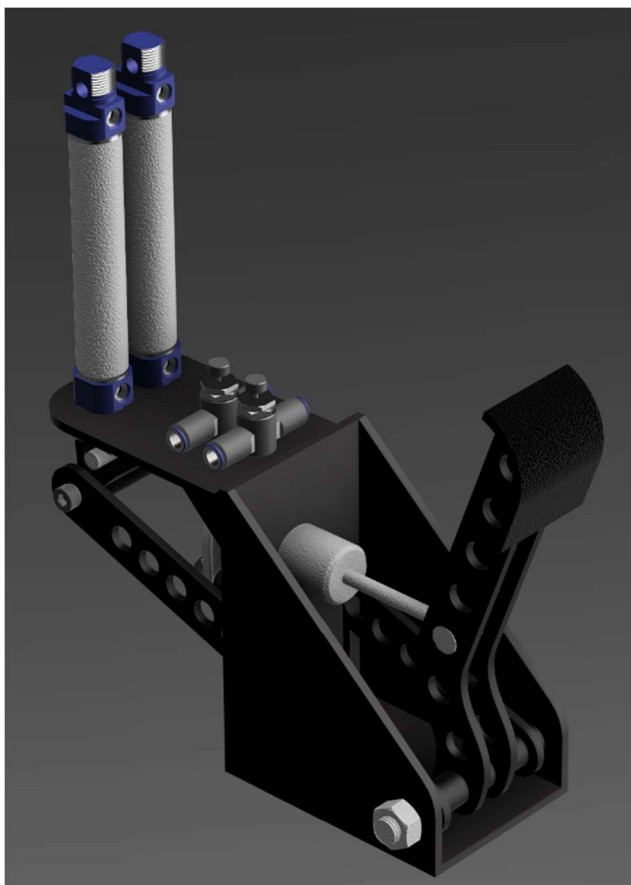
Slika 4.13: Končan prestavni modul II.

4.3 Modul na sklopki

Celoten sistem bo ima možnost avtomatskega pritiskanja in spuščanja sklopke. Ker bo to možnost moč izbrati (s stikalom na armaturni plošči je možno izbrati tudi način, kjer voznik sam pritiska sklopko), je potrebno obdržati tretje pedalo in avtomatski sistem namestiti tako, da ne ovira voznika.

Hitrost premika levega pedala je zelo delikatna operacija. Pri menjavi v višjo prestavo je po navadi mnogo večja kot pri prestavljanju v nižjo prestavo, saj je potrebno počakati nekaj

časa, da se motorju povzpne število obratov. Za čim hitrejšo potiskanje stopalke se uporabi dva valja, ki pri 6 atmosferah tlaka potiskata s kar 600 N. Vsakega nadzoruje svoj 5/2 elektromagnetni ventil, ki je povezan neposredno na zalogovnik stisnjenega zraka. To omogoči čim hitrejši pretok zraka po sistemu, saj sta, sicer enaka sistema, med seboj popolnoma neodvisna in brez skupnih ozkih grl. Hitrost popuščanja stopalke se nadzoruje tako, da se zraka, ki izhaja iz valjev, ne spusti v ozračje, vendar se ga duši z različnimi dušilnimi ventili. S tem se omogoči prilagodljivo in nadzorovano hitrost gibanja.



Slika 4.14: 3D model modula na sklopki.



Slika 4.15: Dušilni ventil.

4.3.1 Zaznavanje stopala

Pri speljevanju in začetku vožnje vzvratno, zaradi različnih razmer računalnik ne nudi dovolj natančnega nadzora spuščanja sklopke. Voznik mora torej pred vsakim speljevanjem svojo nogo položiti na pedalo. To mora računalnik zaznati, zato je na samem koncu stopalke nameščen senzor za pritisk, narejen iz plasti prevodnega traku, vzmetne pene in prevodnega traku nalepljenega na plast gume (slika 4.16). Ko voznik pohodi stopalko se prevodni plasti dotakneta, to pošlje signal računalniku ter mu tako sporoči, da spusti avtomatsko sklopko in čaka na nadaljnji ukaz.



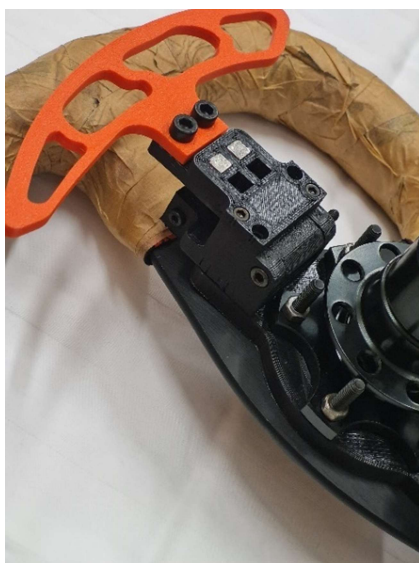
Slika 4.16: Prototip stikala na stopalki.

4.3.2 Varnostni sistem

V kateremkoli primeru napačnega delovanja prestav, motorja ali celo zaviranja v sili je sistem zasnovan tako, da stopalka in pnevmatična cilindra nista permanentno povezana. To pomeni, da lahko voznik v vsaki situaciji, ne glede na pozicijo batnice kadarkoli pritisne sklopko in s tem prekini povezavo motorja z menjalnikom.

4.4 Obvolanske ročice

Prestave je možno menjati z obvolanskimi ročicami (F1 prestave). Celoten sistem je pritrjen med zadnjo stranjo volana in sistemom za hiter odklop (ki omogoča lažje, hitrejšo in varnejše izstopanje iz vozila). Ročici ne vsebujeta nobene vzmeti, ampak magnet, ki varujejo pred zmotnim menjanjem prestav, saj zagotavljajo potrebo po precej veliki sili za sprožitev. Takoj ko se magneta »odlepita« se sila znatno zmanjša in sproži se preprosto mehansko stikalo. Signal obeh stikal nadaljuje po zaščiteni žici do krmilne enote. Omenjena zaščita preprečuje zunanji motnjam (npr.: visoka napetost svečk v motorju), da bi motile signale, ki potujejo po prevodniku. Takoj za tem, ko je celoten sistem narisano v računalniškem programu, se ga pošlje 3D tiskalniku, ki ga natisne iz ustreznega polimera. Vse komponente so privijačene na volanski obroč in testirane.



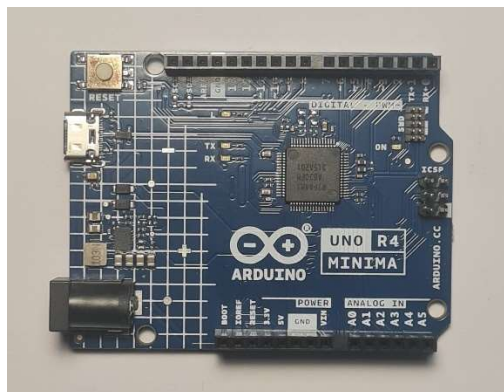
Slika 4.17: Obvolanske ročice (F1 prestave).

5 ELEKTRONSKI SISTEM MENJALNIKA

5.1 Mikrokrmilnik

Italijansko podjetje Arduino® izdeluje odprtokodne mikrokrmilnike, ki omogočajo amatersko in tudi do neke mere profesionalno izdelavo prototipov ali celo končnih izdelkov na področju elektrotehnike ali mehatronike. Krmilniki so zelo poceni in široko dostopni, kajti o njih obstaja ogromno dokumentacije po svetovnem spletu.

Za formulo je bil izbran krmilnik Arduino® Uno R4; čip najnovejše generacije se ponaša s precej večjimi hitrostmi, zanesljivostjo in odpornostjo proti zunanjim motnjam. Deluje na napetosti 5 voltov, zato je potrebno del avtomobilske napetosti 14,5 voltov znižati s posebnimi transformatorji, ki nudijo tudi zaščito pred sunki velikih tokov in napetosti, še posebno pri zagonu motorja.

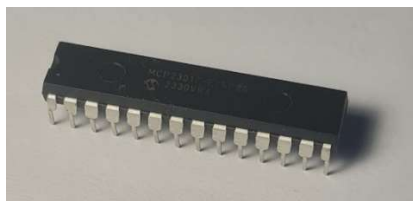


Slika 5.1: Arduino Uno R4.

5.1.1 Razširitev vezja

Mikrokrmilnik ima 5 analognih vhodov (niso v rabi) in 13 digitalnih vhodov in izhodov. Na žalost je to za celoten sistem prestav premalo. Lahko bi se odločili za večji krmilnik Arduino® Mega, ki ponuja 56 krmilnih kontaktov, a ima na žalost vgrajen prepočasen procesor (3x počasnejši od Arduino® Uno R4). Na voljo ostane samo še rešitev z razširitvenim vezjem.

Najbolj primeren razširitveni modul je čip MPC23017, ki se preko I2C komunikacijskega protokola »pogovarja« z Arduinom preko treh žic. Zaradi vseh vmesnih sprememb signalov, se z vsakim ukazom izgubi nekaj sto mikrosekund, zato se preko tega modula pošilja signale le komponentam, katerih hitrosti niso ključne npr.: zaslon, stikala pri vozniku, itd.



Slika 5.2: Integrirano vezje MCP23017.

5.1.2 Releji

Krmilni ventili za svoj vklop potrebujejo precej več toka, kot ga lahko Arduino® zagotovi. To pomeni, da je potrebno v sistem umestiti tudi releje. Ti lahko z malo toka vklopijo kontakt, ki prenaša tudi do 100x več toka in napetosti.

Obstajata dve vrsti relejev in sicer:

- Mehanski rele: z majhnim tokom tuljava fizično premakne kotvo, ki vklopi stik, čez katerega lahko tečejo veliki tokovi. Zaznamuje ga relativno počasno delovanje, zaradi fizičnih premikov hitra obraba, glasnost med preklapljanjem in nizka cena,
- SSR rele (solid state relay): v tem primeru z zelo majhnim tokom vklopi tranzistor (mosfet), ki v povprečno 50 nanosekundah s svojimi polprevodniškimi lastnostmi vklopi kontakt, ki prenaša večje tokove. Poleg hitrejšega vklopa so popolnoma tihi, vendar precej dražji.

Za aplikacijo SSR releji niso potrebni, saj take hitrosti preklapljanja niso zahtevane, zvok preklapljanja pa pomaga pri diagnosticiranju težav v sistemu. Za pravilno delovanje sistem potrebuje skupno 7 relejev (enega za vsak pnevmatični krmilni ventil), ki se jih namesti v bližino krmilne enote.



Slika 5.3: Modul mehanskih relejev.

5.2 Induktivno stikalo

Sistem mora natančno vedeti, kdaj je sklopka dokončno pritisnjena in to čim hitreje poslati krmilni enoti. Namesto uporabe običajnega (mehanskega) stikala, se uporabi induktivno stikalo, ki z oddajanjem in spremljanjem elektromagnetnega polja zazna feromagnetne (kovinske) predmete na razdalji manj od 8 mm [6]. Ta senzor je izbran zato, ker za zaznavo ne potrebuje fizičnega kontakta, kar omogoča natančno in zanesljivo zaznavanje tudi ob možni manjši deformaciji pedala zaradi hitrih in sunkovitih obremenitev.



Slika 5.4: Induktivno stikalo.

5.3 Zaslون

Prestava v kateri je menjalnik, bo v vsakem trenutku (realtime) izpisana na LED zaslonu na armaturni plošči. Ta je sestavljen iz 7 segmentov, v katerih je po 5 zaporedno vezanih LED diod barve »super red«, za katere je značilno, da vsaka potrebuje 1.85 V in približno 20 mA. Izračunamo (enačba 5.1), da en segment s 5 diodami potrebuje napetost 9,25 V.

$$U_s = U_1 \times n$$

U_s – skupni padec napetosti [V]

U_1 – padec napetosti ene LED diode [V]

n – število diod v enem segmentu

Enačba 5.1: Izračun potrebne napetosti LED sijalk.

Tako moramo za napajanje zaslona napetost najprej zmanjšati napetost iz 14,5 V na 9,25 V, torej potrebujemo upor, ki bo napetost zmanjšal za 5,25 V pri 20 mA toka. Njegovo vrednost se izračuna po enačbi Ohmovega zakona:

$$R = \frac{U}{I}$$

R – upor [ohm, Ω]

U – napetost [V]

I – električni tok [A]

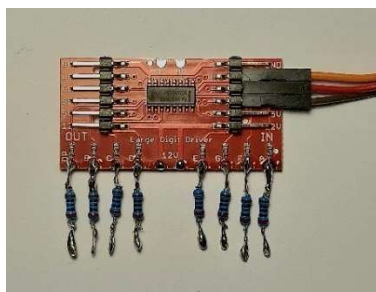
Enačba 5.2: Izračun upornosti.

Izračunana potrebno upornost (enačba 5.2), ki znaša 262,5 Ω . Za vsak segment zaslona se izbere najbližjo standardno vrednost upora, ki v tem primeru znaša 270 ohm-ov.



Slika 5.5: Zaslona.

Zaradi lažjega in hitrejšega nadzora vseh delov zaslona hkrati, se uporabi pomični register (ang.: shift register), ki omogoča, da se z zgolj tremi žicami lahko nadzoruje do osem segmentov zaslona. Najbolj primeren je čip 6C596G4, ki sprejeme 5 V signale neposredno iz krmilnika prestav. Pogosto je na matični plošči z že vnaprej vgrajenimi uporniki, vendar na žalost največ 150 ohmov za vsak segment. Tej vrednosti je potrebno torej na vsaki žici dodati še vsaj 120 Ohmov upora, za pravilno in varno delovanje zaslona.

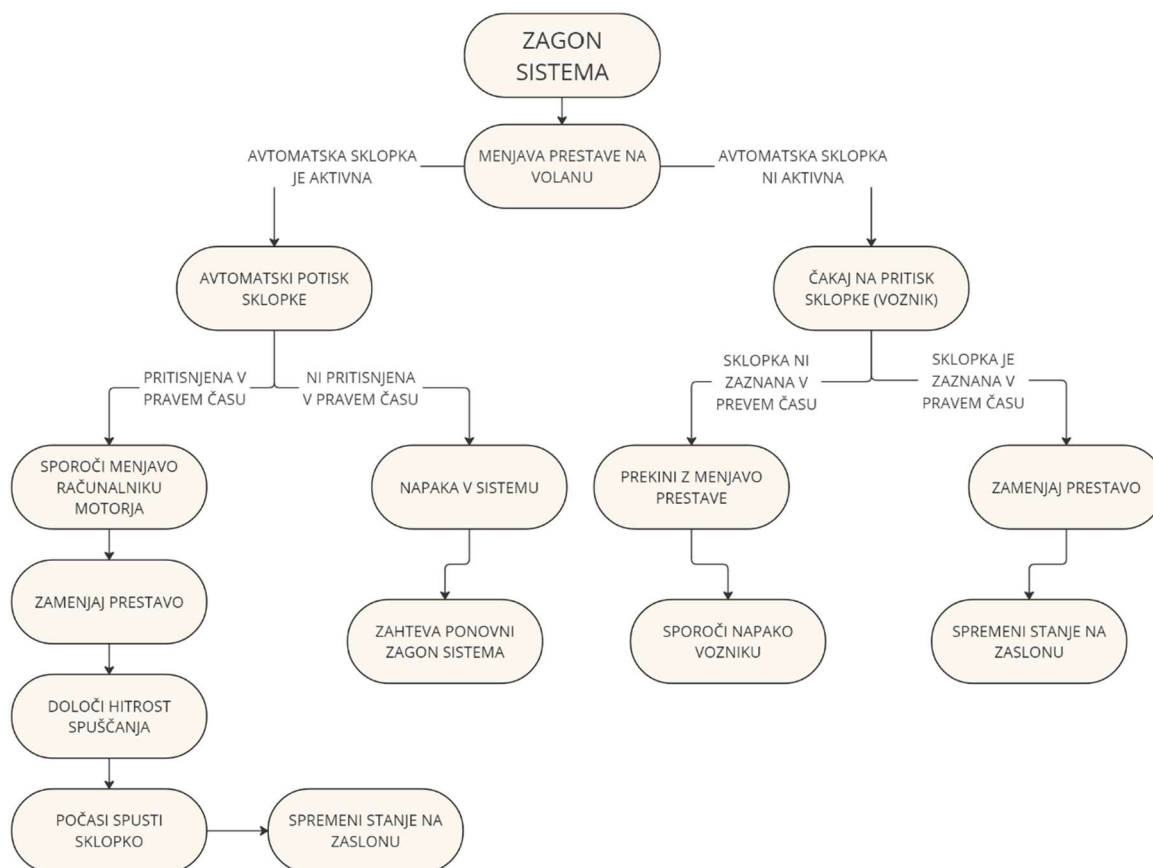


Slika 5.6: Krmilnik zaslona z upori.

5.4 Program

Izumitelji Arduina® so razvili svoj programski jezik, ki bazira na jeziku C++. Program za menjanje prestav je napisan v njihovem virtualnem razvojnem okolju Arduino IDE.

Logično delovanje je precej kompleksno, predvsem zaradi velikega števila možnosti (stikal na armaturni plošči), vendar ga je možno poenostaviti z diagramom.



Slika 5.7: Diagram delovanja programa. [5]

Povzetek delovanje: Sistem čaka na menjavo prestave na obvolanskih ročicah. Takoj, ko dobi signal, preveri ali je vključena funkcija avtomatske sklopke - če je vključena nemudoma pošlje signal releju in krmilnemu ventilu naj jo pritisne. Stopalko mora pritisniti v točno določenem časovnem intervalu (največ 0,4 sekunde). V nasprotnem primeru računalnik vozniku sporoči, da je nekje v sistemu napaka. Enak protokol velja, če način avtomatske sklopke ni vključen; voznik mora stopalko pritisniti dovolj hitro. Ob sprejetju signala z induktivnega senzorja na stopalki, se sproži podprogram menjanja prestav. Ker se morajo pnevmatični valji po 3 x 3 prestavnem polju premikati s točno določenim zaporedjem,

računalnik zbere podatke o prestavi v kateri se trenutno nahaja in kamor se hoče premakniti. Tako torej določi pravilen vrstni red sprožitve ventilov. V primeru menjave z avtomatsko sklopko, nato določi stopnjo dušenja izstopnega zraka in s primerno hitrostjo spusti sklopko. Na koncu s pomočjo logičnega vezja spremeni stanje na zaslonu ter s tem vozniku sporoči, da je bila menjava prestave uspešna. Celotna zanka se zopet ponovi. [Priloga 1].

5.5 Varnost

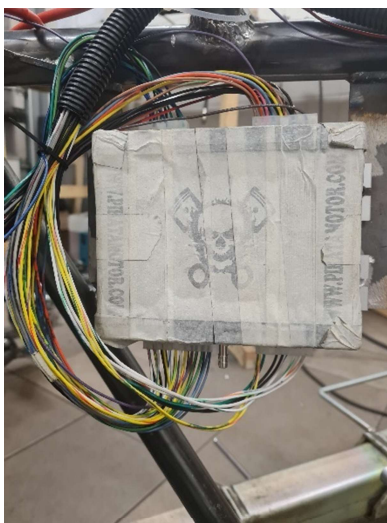
Varnost sistema je pri vozilih z veliko zmogljivostjo in hitrostjo zelo pomembna. Vse elektronske in pnevmatske komponente so zvezane tako, da ob izpadu elektrike oz. kratkem stiku, cilindri menjalnik na silo (brez uporabe sklopke) potisnejo v prosti tek, kjer ostanejo do ponovnega zagona sistema.

Vozilo je opremljeno tudi s tipko v sili, ki kot že prej omenjeno, menjalnik čim prej potisne v prosti tek in s tem zaščiti vse mehanske in električne komponente ter najbolj pomembno, zavaruje voznika.

6 KRMILJENJE MOTORJA IN TURBINE

6.1 Računalnik motorja

Motor je krmiljen preko računalnika Megasquirt MS3 Competition, ki je eden od bolj cenovno ugodnih enot za popoln nadzor kateregakoli motorja z notranjim izgorevanjem. Obdaja ga 80 žic, vsaka s svojo nalogo. Nekatere sprejemajo podatke s senzorjev, nekatere oddajajo signale v izvršilne enote, nekatere pa so prosto programljive s pomočjo krmilnega programa. Računalnik je potrebno konfigurirati z vsemi podatki o pogonskem agregatu. Kalibrirati je potrebno vse senzorje in vanj vnesti tabele [Priloga 2], ki nadzorujejo količino vbrizganega goriva in drugih veličin. Zaradi najnovejših tehnologij je računalnik s pomočjo senzorjev sposoben sam spreminjati tabele goriva in tako znatno povečati moč. S pomočjo večje turbine in novih brizgalnikov goriva je to v namenu zvišati iz 150 na 250 konjskih moči.



Slika 6.1: Krmilni računalnik motorja (zaščiten).

6.1.1 Senzorji

Pogonski agregat deluje na principu zbiranja podatkov na vhodnih senzorjih, obdelave zbranih podatkov in posredovanje le teh izvršilnim enotam, kot so vžigalne tuljave in brizgalniki goriva.

Med glavne in izredno pomembne vhodne senzorje spadajo:

- CKP (Crankshaft position sensor) oz. senzor pozicije glavne gredi, ki deluje na principu zaznavanja magnetnega polja zobnika na glavni gredi. Senzor računalniku posreduje zelo natančno pozicijo in hitrost glavne gredi,
- TPS (Throttle position sensor) oz. senzor lopute za plin, ki računalniku pove v katerem položaju je zračna loputa in kakšen pospešek voznik zahteva od motorja,
- IAT (Intake air temperature) oz. senzor za temperaturo vhodnega zraka, ki ves čas spremlja spremembe temperature in jih sporoča računalniku.

Ti in še ostali, manj pomembni senzorji, konstantno spremljajo delovanje motorja in stanje vhodnih medijev ter pripomorejo k učinkovitosti delovanja.

6.2 Nadzor turbine

6.2.1 MAP senzor

V modulu za nadzor motorja je tovarniško vgrajen senzor MAP (mass air pressure), ki ves čas meri tlak zraka tik pred vhomom v cilindre. Ta tlak je ključnega pomena za delovanje motorja, saj se nanj orientirajo skoraj vse veličine, ki skrbijo za optimalno moč, kot so:

- količina vbrizganega goriva,
- začetek vbrizgavanja goriva,
- predvžig vžigalnih svečk,
- predvidena moč turbine.

6.2.2 Ventil za prepihanje N75

Ko voznik stopi iz pedala za plin, turbopolnilnik nima proste poti za vpihovanje zraka, zato na samem turbinskem kolesu nastane ogromna količina upora, ki poleg povzročanja obrabe ležajev, znatno poveča upor izpušnih plinov in posledično zmanjša učinkovitost celotnega motorja. Zaradi tega je v samem ohišju turbopolnilnika vgrajena loputa, ki omogoči izpušnim plinom, da v celoti obidejo turbino in se sprostijo neposredno v izpušni sistem. To loputo nadzoruje N75 ventil, ki ga s pomočjo PWM signala krmili glavni računalnik. S tem ventilom in MAP senzorjem dosežemo delovanje turbine v zaprti zanki, kar pomeni, da računalnik konstantno spremlja tlak vhodnega zraka in ga po potrebi manjša. Posledično omogoča bolj konstantno moč in še pomembneje, zaščito pred prevelikimi tlaki zraka, ki lahko privedejo do katastrofalnega uničenja motorja.

6.2.3 Izpihivalni ventil

Izpihivalni ventil (ang. blowoff valve) je še dodatna varnostna loputa, ki ščiti sistem pred previsokimi tlaki. Kot že prej omenjeno, se pri prenehanju dodajanja plina turbina še naprej vrti in črpa zrak skozi sistem vmesnih hladilnikov. Takoj, ko izpihivalni ventil to zazna, se odpre loputa, ki izpusti ves nepotreben zrak pod tlakom v okolico. Nemudoma, ko se loputa za plin ponovno začne odpirati, se ta ventil zatesni in zrak lahko ponovno s povišanim tlakom vstopa v motor.



Slika 6.2: Izpihivalni ventil.

6.3 Integracija prestav z motorjem

Zaradi odprtokodne narave krmilnika, lahko motor in prestave povežemo ter s tem dosežemo zelo učinkovito in gladko menjanje prestav. Arduino® tik pred menjavo prestave pošlje signal računalniku motorja, ki za približno 0,1 sekunde (odvisno od pozicije zračne lopute in obratov motorja) prekine dodajanje goriva, kar izniči moč motorja in omogoči boljšo sinhronizacijo zobnikov v menjalniku.

S tem principom je mogoče menjati prestave tudi brez pritiskanja na sklopko tako, da za dlje časa (0,2 – 0,3 sekunde) izklopimo dovod goriva, vendar je ni ugodno za zobnike in sinhronizacijske obročke v menjalniku. Zaradi tega razloga, v predstavljenem sistemu, ta funkcija ni uporabljena, saj je tveganje za popolno uničenje celotnega pogonskega sistema preveliko.

7 REZULTATI

Vse postavljene hipoteze je mogoče potrditi ali zavrniti, vendar samo odgovor DA ali NE ne zadostuje pri nobeni od njih:

- I. Za nizko ceno lahko ročni menjalnik v avtomobilu sami robotiziramo.

Hipotezo **POTRDIMO**. Cena je vedno relativna, vendar smo v primerjavi z avtomobili, za katere proizvajalci nudijo predelavo menjalnikov v robotizirane, za celoten sistem plačali v povprečju kar 10-krat manj.

- II. Menjanje prestav z robotiziranim pnevmatskim menjalnikom je hitrejšo od menjanja prestav z ročnim menjalnikom.

Hipotezo **POTRDIMO**. Z nekaj vaje in popolnim razumevanjem sistema lahko voznik izjemno hitro menja prestave; v popolnih pogojih lahko prestavo zamenja v zgolj 380 milisekundah (0,38 sekunde), kar je precej hitreje od povprečnega izmerjenega časa menjave prestave, ki znaša 1,62 sekunde.

- III. Sistem robotiziranih prestav je primeren za vsakdanjo rabo.

Hipotezo **OVRŽEMO**. Sistem deluje odlično na vozilih predelanih za privatno dirkaško stezo. V vsakdanjem svetu bi bila modifikacija nesmiselna, saj poleg nelegalnosti predelave avtomobila, vožnja v veliko primerih postane precej neudobna. Poleg tega izgubimo klimatsko napravo in ob vsakem menjanju prestave zaslišimo glasen in moteč šum zraka.

- IV. Nadzorovanje turbopolnilnika je s pomočjo krmilne enote motorja enostavno in zanesljivo.

Hipotezo **OVRŽEMO**. Princip nadzorovanja v teoriji ni zahteven, vendar se v praksi pojavi nešteto spremenljivk, ki otežijo popoln nadzor prisilnega polnjenja zraka. Največjo vlogo za pogoste napake predstavlja dolga pot zraka od turbo polnilnika do samega vhoda v motor, saj se tlak in temperatura skozi sistem močno spreminjata in povzročata precejšnje zaostanke signalov.

- V. Krmilna sistema menjalnika in motorja lahko povežemo in dosežemo boljše

rezultate.

Hipotezo **POTRDIMO**. S povezovanjem obeh krmilnih sistemov privedemo predvsem do udobnejše vožnje in manjše obrabe pogonskih elementov.

8 ZAKLJUČEK

Predelava menjalnika je bila torej uspešna. Dosegli smo visoko stopnjo avtomatizacije celotnega sistema in s tem presegli naša začetna pričakovanja. Kljub temu, da upravljalna enota zaradi občasne neudobnosti vožnje in hrupnega menjanja prestav ni primerna za vsakdanjo vožnjo, je varnosti voznika povečana. Obe roki ves čas ostaneta na volanu in s tem pripomoreta boljšemu nadzoru vozila in večji koncentraciji voznika. Kljub današnji tehnologiji robotizirani menjalniki ne predstavljajo konkurence avtomatskim menjalnikom, ki so veliko bolj zmogljivi in zanesljivi, ter vse cenejši in dostopni povprečnemu vozniku. Krmiljenje motorja in turbine je nekoliko bolj zapleteno, še posebno zaradi različnih načinov vožnje in zunanjih vplivov, kot je na primer temperatura zraka. Večino težav je mogoče rešiti, vendar je popoln nadzor turbine brez posebnih priprav nemogoč. Kljub temu smo tudi pri raziskovanju krmiljenja turbine presegli pričakovanja in uspešno povečali moč motorju.

9 VIRI

- [1] CAR AND DRIVER. *What is cvt transmission.* [na spletu]. 2024. Dostopno: <https://www.caranddriver.com/research/a30822055/what-is-a-cvt-transmission/>

- [2] MANUA. *Audi manuals.* [na spletu]. 2024. Dostopno: <https://www.manua.ls/cars/audi>

- [3] FESTO. *Fluidsim.* [na spletu]. 2024. Dostopno: https://www.festo.com/si/sl/e/tehnico-izobrazevanje/digitalno-ucenje/virtualna-simulacija-in-modeliranje/fluidsim-id_1663056/

- [4] M&M INTERCOM. *Gumijasti blažilci.* [na spletu]. 2024. Dostopno: <https://mm-eshop.si/blazilec-st-2-30x20-m8x23-gumi-blazilec>

- [5] MIRO. *Visual project manager.* [na spletu]. 2024. Dostopno: <https://miro.com/index/>

- [6] MOJMOJSTER. *Približevalni senzorji in stikala.* [na spletu]. 2024. Dostopno: https://www.mojmojster.net/clanek/1384/priblizevalni_senzorji_in_stikala

ZAHVALA

Zahvalil bi se mentorju Nejcju Marinšku, za vso pomoč in podporo skozi celoten projekt. Zahvaljujem se tudi Šolskemu centru Ljubljana, Srednji šoli za strojništvo, kemijo in varovanje, ki je omogočila projekta. Posebno bi se zahvalil Avtocentru Šubelj, ki nam je omogočil prostore skozi celotno ustvarjanje in nudil vso možno podporo, Akrapovič d. d. za izdelavo izpušnega sistema, podjetju Prit d. o. o. za hidravlične in pnevmatične cevi ter priključke, podjetju OGM d. o. o. za toplotno izolacijo in podjetju Dts d. o. o. za potisk majic in tiskanje nalog.

10 PRILOGE

Priloga 1: Program robotiziranega menjalnika

```
#include <Bounce2.h>
#include <Servo.h>

const byte SEGMENT_LOOKUP[] = {
  B01110111, // 0
  B00100100, // 1
  B11100011, // 2
  B11100110, // 3
  B10110100, // 4
  B11010110, // 5
  B11010111, // 6
  B01100100, // 7
  B11110111, // 8
  B11110110, // 9
  B00001000, // 10, .
  B01110000, // 11, R
  B11010011, // 12, E
  B00000000 // 13, blank
};

int cdel = 30;
int bdel = 30;

#define clutchSwitchEnable 7
#define autoClutchEnable 12
#define clutchSensor 8
#define footSensor 6

#define gearUpPin 2
#define gearDownPin 4

#define cil_1 3
#define cil_2 5
#define cil_3 13
#define cil_4 9
#define clu 11
#define ger 10

#define latchPin 15
#define dataPin 16
#define clockPin 17

bool blip = false;

int cdel = 20;
int bdel = 70;
int dispDelay = 100;

int gear = 0;
int gearN;
int gear0;
bool standby;
bool ready = false;
bool done = false;

unsigned long startTime = 0;
unsigned long timeWindow = 2000;
unsigned long prevTime = millis();

Bounce2::Button gearUp =
  Bounce2::Button();
Bounce2::Button gearDown =
  Bounce2::Button();

Servo clutchValve;

void setup() {
  Serial.begin(9600);
  Serial.println("Start");

  clutchValve.attach(3);

  pinMode(cil_1, OUTPUT);
  pinMode(cil_2, OUTPUT);
  pinMode(cil_3, OUTPUT);
  pinMode(cil_4, OUTPUT);
  pinMode(clu, OUTPUT);
  pinMode(ger, OUTPUT);

  gearUp.attach(gearUpPin, INPUT);
  gearDown.attach(gearDownPin, INPUT);
  gearUp.interval(5); //debounce time ms
  gearDown.interval(5);
  gearUp.setPressedState(HIGH);
  gearDown.setPressedState(HIGH);

  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(latchPin, OUTPUT);

  pinMode(clutchSwitchEnable, INPUT);
  pinMode(autoClutchEnable, INPUT);
  pinMode(clutchSensor, INPUT);
  pinMode(footSensor, INPUT);

  standby = true;
}
```

```

Serial.println("Standby");
display(0);
gear = 0;
changeGear();
}

void loop() {

  ready = false;
  unsigned long currentTime = millis();

  if(currentTime - prevTime > 1000){
    Serial.println(gear);
    prevTime = currentTime;
  }

  gearUp.update();
  gearDown.update();

  if (gearUp.pressed() && gear <= 4) {
    gear0 = gear;
    gearN = gear + 1;
    delay(25);
    Serial.println("UP presses");
    Serial.println(gear);
    Serial.println(gear0);
    Serial.println(gearN);
    Serial.println(gearCheck);
  }else if (gearDown.pressed() && gear >=
0) {
    gear0 = gear;
    gearN = gear - 1;
    delay(25);
    Serial.println("DOWN presses");
    Serial.println(gear);
    Serial.println(gear0);
    Serial.println(gearN);
    Serial.println(gearCheck);
  }

  if (gearN != gear){

    if(digitalRead(autoClutchEnable) ==
HIGH){
      Serial.println("Automatic clutch
ENABLED");
      if(gear < gearN && gearN != -1 &&
gearN != 0 && gearN != 1){
        clutchValve.write(60);
        Serial.println("Servo 50 odprt");
      }
      if(gear > gearN && gearN != -1 &&
gearN != 0 && gearN != 1){
        clutchValve.write(120);
        Serial.println("Servo 20 odprt");
      }
    }
    if(gearN == -1 || gearN == 0 ||
gearN == 1 && gear0 == 0){
      clutchValve.write(0);
      Serial.println("Servo čisto
odprt");
    }
    if(gearN == 1 && gear0 == 2){
      clutchValve.write(30);
      Serial.println("Servo 10 odprt");
    }
    clutchPre(); // tudi v primeru -1,
0, 1 ostane pritiskanje enako in
vedno...spremeni se samo spuskanje
  }
  gear = gearN;
  check(); // -> ready ok/not
}

if(ready == true){
  Serial.println("READY");
  changeGear();
  ready = false;
  delay(500);
}
}

void clutchPre(){
  done = false;
  digitalWrite(clu, HIGH);
  startTime = millis();
  while(millis() - startTime <
timeWindow){
    if (digitalRead(clutchSensor) == HIGH)
{
      Serial.println("automatic clutch
pressed in time");
      return;
      done = true;
    }
  }
  if(done != true){
    Serial.println("automatic clutch NOT
pressed in time");
    error();
    done = false;
  }
}

void check(){
  Serial.println("Check");
  if (digitalRead(clutchSwitchEnable) ==
HIGH && done == false) {
    done = false;
    startTime = millis();
  }
}

```

```

        while(millis() - startTime <
timeWindow){
    if (digitalRead(clutchSensor) ==
HIGH) {
        Serial.println("pressed in time");
        ready = true;
        return;
    }
}
if(done == false){
    Serial.println("not pressed in
time");
    if(digitalRead(autoClutchEnable) ==
HIGH){
        error();
    }else{
        returnGear();
    }
    return;
}
}
}

void clutchRel(){
    if(digitalRead(autoClutchEnable) ==
HIGH){
        if(gear == 2 || gear == 3 || gear == 4
|| gear == 5){
            Serial.println("Automatic clutch
releaed");
            digitalWrite(clu, LOW);
        }
        if(gear == 1 && gear0 == 2){
            Serial.println("Automatic clutch
releaed slowly 2 -> 1");
            digitalWrite(clu, LOW);
        }
        if(gear == -1 || gear == 1 && gear0 !=
2){;
            int ch = 0;
            while(ch == 0){
                if(digitalRead(footSensor) ==
LOW){
                    ch = 1;
                }
                delay(10);
            }
            Serial.println("Foot detected,
automatic clutch releaed");
            digitalWrite(clu, LOW);
        }
    }

    if(gear == 0){
        Serial.println("Automatic clutch
still pressed");
    }
}

void error(){
    Serial.println("ERROR"); //Prisilni N
delay(100000);
}

void returnGear(){
    Serial.println("Return gear");
    /*Serial.println((String) "O = " +
gear0);
    Serial.println((String) "N = " + gearN);
    Serial.println((String) "G = " + gear);
    Serial.println((String) "C = " +
gearCheck);*/
    gear = gear0;
    gearN = gear0;
    return;
}

void changeGear(){
    digitalWrite(ger, HIGH);
    switch(gear){
        case -1:
            if(gear0 == 0){
                digitalWrite(cil_1, HIGH);
                digitalWrite(cil_4, LOW);
                delay(cdel);
                digitalWrite(cil_2, HIGH);
                delay(bdel);
                digitalWrite(cil_3, LOW);
                Serial.println("N -> R");
                display(6);
            }
            break;

        case 0:
            if(standby == true){
                standby = false;
                digitalWrite(cil_2, HIGH);
                delay(cdel);
                digitalWrite(cil_1, LOW);
                digitalWrite(cil_3, LOW);
                digitalWrite(cil_4, HIGH);
                Serial.println("any -> N");
                display(0);
            }
            else if(gear0 == -1 && standby ==
false){

```

```

        digitalWrite(cil_2, HIGH);
        delay(bdel);
        digitalWrite(cil_4, LOW);
        digitalWrite(cil_3, HIGH);
        delay(cdel);
        digitalWrite(cil_1, LOW);
        Serial.println("R -> N");
        display(0);
    }else if(gear0 == 1 && standby ==
false){

        digitalWrite(cil_1, LOW);
        digitalWrite(cil_3, LOW);
        delay(cdel);
        digitalWrite(cil_2, HIGH);
        digitalWrite(cil_4, HIGH);
        Serial.println("1 -> N");
        display(0);
    }
break;
case 1:
    if(gear0 == 0){

        digitalWrite(cil_1, HIGH);
        delay(cdel);
        digitalWrite(cil_2, LOW);
        digitalWrite(cil_3, LOW);
        digitalWrite(cil_4, LOW);
        Serial.println("N -> 1");
        display(1);

    }else if(gear0 == 2){

        digitalWrite(cil_1, HIGH);
        delay(cdel);
        digitalWrite(cil_2, LOW);
        digitalWrite(cil_4, LOW);
        delay(bdel);
        digitalWrite(cil_3, HIGH);
        Serial.println("2 -> 1");
        display(1);

    }
break;
case 2:
    if(gear0 == 1){

        digitalWrite(cil_3, LOW);
        digitalWrite(cil_2, LOW);
        digitalWrite(cil_1, HIGH);
        delay(cdel);
        digitalWrite(cil_4, HIGH);

        Serial.println("1 -> 2");
        display(2);
    }else if(gear0 == 3){

        digitalWrite(cil_1, LOW);
        digitalWrite(cil_4, HIGH);
        delay(bdel);
        digitalWrite(cil_3, HIGH);
        digitalWrite(cil_2, LOW);
        Serial.println("3 -> 2");
        display(2);

    }
break;
case 3:
    if(gear0 == 2){

        digitalWrite(cil_1, HIGH);
        delay(cdel);
        digitalWrite(cil_2, LOW);
        digitalWrite(cil_4, HIGH);
        digitalWrite(cil_3, LOW);
        Serial.println("2 -> 3");
        display(3);

    }else if(gear0 == 4){

        digitalWrite(cil_1, LOW);
        digitalWrite(cil_3, LOW);
        delay(bdel);
        digitalWrite(cil_4, HIGH);
        digitalWrite(cil_2, HIGH);
        Serial.println("4 -> 3");
        display(3);

    }
break;
case 4:
    if(gear0 == 3){

        digitalWrite(cil_1, HIGH);
        digitalWrite(cil_4, HIGH);
        digitalWrite(cil_3, LOW);
        delay(cdel);
        digitalWrite(cil_2, LOW);
        Serial.println("3 -> 4");
        display(4);

    }else if(gear0 == 5){

        digitalWrite(cil_4, HIGH);
        delay(bdel);
        digitalWrite(cil_2, LOW);

```



```

        digitalWrite(cil_1, LOW);
        delay(cdel);
        digitalWrite(cil_3, HIGH);
        Serial.println("5 -> 4");
        display(4);
    }
    break;

    case 5:
        if(gear0 == 4){
            digitalWrite(cil_1, HIGH);
            delay(cdel);
            digitalWrite(cil_3, HIGH);
            digitalWrite(cil_2, HIGH);
            digitalWrite(cil_4, HIGH);
            Serial.println("4 -> 5");
            display(5);
        }
        break;
    }
    digitalWrite(ger, LOW);
    clutchRel();
}

void display(int digit) {
    if (digit < 0 || digit > 13) {
        return; // Exit
    }
    byte segments = SEGMENT_LOOKUP[digit];
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST,
segments);
    digitalWrite(latchPin, HIGH);
    Serial.println((String) "Display: " +
digit);
}

```

Priloga 2: Tabele krmilnega računalnika motorja

The screenshot displays the TunerStudio MS Ultra v3.1.08-3 software interface. The main window shows the 'Fuel VE Table 1' and the 'Ignition Table 1 (Spark Advance)'. The status bar at the bottom indicates '3 Ready' and 'CurrentTune.msq'.

Fuel VE Table 1

220.0	89.5	90.3	91.0	91.8	93.0	94.1	96.1	98.0	100.0	101.9	103.8	105.8	107.7	109.6	111.5
220.0	89.5	90.3	91.0	91.8	93.0	94.1	96.1	98.0	100.0	101.9	103.8	105.8	107.7	109.6	111.5
190.0	86.5	87.1	87.8	88.4	89.4	90.4	92.0	93.6	95.2	96.8	98.4	100.0	101.7	103.3	105.0
160.0	77.5	77.9	78.3	78.5	79.7	80.8	82.7	84.5	86.3	88.1	89.6	91.0	92.9	94.4	96.0
130.0	66.0	66.9	67.7	68.6	69.9	71.2	73.3	75.3	77.4	79.4	80.7	81.9	84.0	85.5	87.0
100.0	59.5	60.8	62.1	63.4	65.3	67.2	68.4	69.6	70.7	71.9	73.6	74.7	76.1	77.2	78.5
90.0	56.5	58.1	59.6	61.1	63.5	65.8	66.9	68.0	69.0	70.1	71.2	72.3	73.4	74.4	75.5
80.0	54.8	56.3	57.8	59.3	61.5	63.7	64.7	65.8	66.8	67.9	68.9	69.9	71.0	72.0	73.0
70.0	53.1	55.3	58.5	59.1	59.4	59.6	59.8	60.9	62.3	64.4	65.8	66.3	67.3	68.4	69.5
60.0	51.3	54.2	57.0	57.5	57.3	57.1	57.2	58.6	60.0	61.9	63.3	63.9	65.0	66.0	67.0
55.0	50.5	53.6	56.7	57.3	56.6	55.9	56.1	57.5	58.8	60.6	62.0	62.7	63.8	64.9	66.0
50.0	49.6	53.1	56.5	57.0	55.8	54.6	54.9	56.3	57.7	59.4	60.8	61.5	62.6	63.7	64.8
40.0	47.2	50.0	52.9	55.7	54.0	52.2	52.7	54.0	55.3	56.9	58.2	59.1	60.3	61.4	62.5
30.0	44.8	47.0	49.2	51.4	50.6	49.7	50.4	51.7	53.0	54.5	55.7	56.8	57.9	59.1	60.2
20.0	42.4	43.8	45.5	47.1	47.2	47.2	48.2	49.4	50.6	52.0	53.2	54.4	55.6	56.7	57.8
10.0	40.0	40.9	41.9	42.8	43.7	44.7	45.9	47.1	48.3	49.5	50.7	52.0	53.2	54.4	55.6
	800	1000	1200	1400	1700	2000	2500	3000	3500	4000	4500	5000	5500	6000	6500

Ignition Table 1 (Spark Advance)

220.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
220.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0
190.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
160.0	10.0	10.0	11.3	11.5	11.7	12.0	12.4	12.8	13.2	13.6	14.0	14.4	14.8	15.2	15.6
130.0	10.0	10.0	12.7	13.0	13.5	13.9	14.7	15.5	16.4	17.2	18.0	18.8	19.6	20.4	21.2
100.0	10.0	10.0	14.0	14.5	15.2	15.9	17.1	18.3	19.6	20.8	22.0	23.2	24.4	25.6	26.8
90.0	10.0	10.0	14.7	15.7	17.2	18.8	21.1	23.6	26.0	27.0	28.5	29.5	28.5	28.5	28.5
80.0	13.0	10.0	14.7	15.7	17.3	18.8	21.3	23.9	26.4	27.7	29.0	29.0	29.0	29.0	29.0
70.0	13.0	10.0	15.0	16.0	17.6	19.1	21.7	24.3	26.9	28.4	29.5	29.5	29.5	29.5	29.5
65.0	13.0	10.0	15.2	16.2	17.8	19.4	22.0	24.6	27.2	28.8	29.8	29.8	29.8	29.8	29.8
60.0	13.0	10.0	15.4	16.4	18.0	19.6	22.2	24.8	27.4	29.1	30.0	30.0	30.0	30.0	30.0
55.0	13.0	10.0	15.5	16.6	18.1	19.7	22.4	25.0	27.7	29.5	30.3	30.3	30.3	30.3	30.3
50.0	13.0	10.0	15.7	16.8	18.3	19.9	22.6	25.2	27.9	29.9	30.5	30.5	30.5	30.5	30.5
40.0	13.0	10.0	16.1	17.2	18.8	20.4	23.0	25.7	28.3	30.6	31.0	31.0	31.0	31.0	31.0
30.0	13.0	10.0	16.4	17.5	19.1	20.7	23.4	26.1	28.8	31.3	31.5	31.5	31.5	31.5	31.5
20.0	13.0	10.0	16.8	17.9	19.5	21.1	23.9	26.6	29.3	32.0	32.0	32.0	32.0	32.0	32.0
	800	1000	1200	1400	1700	2000	2500	3000	3500	4000	4500	5000	5500	6000	6500

Status Dashboard

Not Ready	Not Cranking	ASE OFF	WUE OFF	TPS Accel	MAP Accel	TPS Decel	MAP Decel	Need Burn	Data Lost	Config Error	Not Ready	VE1/2	SPK1/2
Not Ready	N2O 1 off	N2O 2 off	No Launch	No Flat shft	Spark out	Over boost	CL Idle off	No Fuel out	T-log	3 step off	Test mode	No 3 step	No soft limit
No seq shft	Launch Off	EGT warn	EGT shutdown	AFR warn	AFR shutdown	Idle VE	Idle Adv	Fan	MAP/Throttle error	WT1 err	WT2 err	WT3 err	WT4 err
No Knock	AC	Check Engine	Limp mode	W.I. low	VSS launch LO	VSS launch on	DBW fault	Oil shutdown	Not Ready	Overman fuel out	Brake switching	Flap/val switching	SD Warn
Brake switching	Safety shutdown	CAN RX	CAN TX	IObox1	IObox2	IObox3	L.T.Trim	No SD	SD ready	SD Log	SD Err		MAP fault
MAT fault	CLT fault	TPS fault	BATT fault	AFR0 fault	SYNC fault	EGT fault	Flex fault	MAF fault	Knock fault	Cam fault	Oil fault	Fuel press.	CEL/EGT shutdown
CEL/AFR shutdown	CEL/Wiring low	Cruise	Accel	Decel	Overrun	Idle	WOT	Not Logging	Not Connected	Not Ready			

Main Dashboard | Acceleration Performance | a |

3 Ready | | CurrentTune.msq |

File Options Data Logging Communications Tools Help

Basic/L Settin Fuel Settin Igni Sett Start... Accel Enrich Boo... Tab choik Advan Engin 3D Tuning Maps CAN Testr

Graphing & Logging

Color Shade White Color Theme Even Spacing

rpm: 1000
ignload: 30.0
Selected deg: 10.0

ed

Color Theme Even Spacing

Standard Histogram

Burn Close

Half RPM sync	N2O 1 off	N2O 2 off	No Launch	No Flat shift	Spark cut	Over boost	CL Idle off	No Fuel cut	T-log	3 step off	Test mode	No 3 step	No soft limit
No seq shift	Launch Off	EGT warn	EGT shutdown	AFR warn	AFR shutdown	Idle VE	Idle Adv	Fan	RPM/sample error	WT1 err	WT2 err	WT3 err	WT4 err
No Knock	AC	Check Engine	Limp mode	W.I. low	VSS launch LO	VSS launch HO	DBW fault	Oil shutdown	Overrun	Overrun fuel cut	Stuck switching	Fuel cut switching	SD Warn
Boost switching	Safety shutdown	CAN RX	CAN TX	IObox1	IObox2	IObox3	L.T.Trim	No SD	SD ready	SD Log	SD Err		MAP fault
MAT fault	CLT fault	TPS fault	BATT fault	AFR0 fault	SYNC fault	EGT fault	Flex fault	MAF fault	Knock fault	Cam fault	Oil fault	Fuel press.	CO2/GST misread
CO2 AFR shutdown	CEL W. Inj low	Cruise	Accel	Decel	Overrun	Idle	WOT	Temp Logging					

Main Dashboard Acceleration Performance a

3 Ready CurrentTune.msq