



Šolski center Celje

Srednja šola za kemijo, elektrotehniko in računalništvo

Komunikacija LIN Bus z uporabo Arduino Uno in MCP2003

Raziskovalna naloga

Področje: elektrotehnika, elektronika

Mentor:

Gregor Kramer, univ. dipl. inž. el.

Avtor:

Jernej Bizjak, E-4.a

Celje, februar 2024

IZJAVA*

Mentor Gregor Kramer v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Komunikacija LIN Bus z uporabo Arduino Uno in MCP2003, katere avtor je Jernej Bizjak:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalošo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloša nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalošo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloše ob upoštevanju avtorstva in korektnem citiraju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 10.4.2024



Podpis mentorja

Podpis odgovorne osebe
ZA ANITA LAZNIK

*

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloše, pa hrani šola v svojem arhivu.

ZAHVALA

Rad bi se zahvalil mentorju Gregorju Kramerju za čas, trud, podporo, kritiko in nasvete pri izdelavi raziskovalne naloge. Podal mi je veliko dragocenega znanja, ne le pri izdelavi naloge, ampak tudi pri pouku.

Za pomoč bi se rad zahvalil profesoricama Valentini Hrastnik in Klavdiji Špur Jereb za lektoriranje naloge.

Zahvalil bi se rad mojim domačim za neizmerno podporo v času šolanja. V oporo so mi pri boju z boleznijo in brez njih ne bi imel energije, da bi se lotil tudi najtežjih izzivov.

Hvala tudi drugim profesorjem, ki so mi pri izdelavi te naloge podali kakšen nasvet.

POVZETEK

V raziskovalni nalogi sem se odločil raziskati LIN Bus (Local Interconnect Network – Lokalno povezovalno omrežje) komunikacijo, ki se uporablja predvsem v avtomobilski industriji in je nadomestila uporabo dražjega CAN Bus sistema. LIN konzorcij je v devetdesetih letih ustvaril pet avtomobilskih znamk, danes pa je LIN Bus standardiziran. Cilj, ki sem si ga zastavil, je raziskati uporabo oddajniškega čipa na mikrokrmlilniku.

Nalogo sem začel s tem, da sem raziskal potrebne komponente za izdelavo naloge. Izbrane komponente sem naročil in jih začel testirati z različnimi nalogami, ki sem si jih zastavil, vsaka malo zahtevnejša od druge. Naloge so vključevale programiranje mikrokrmlilnika in oddajniškega čipa.

Namen je ugotoviti, kako poteka komunikacija v modernih vozilih, v katerih je vse več potrebe po branju senzorjev in krmiljenju naprav.

Ključne besede: Komunikacija LIN Bus, LIN, Arduino Uno, MCP2003

ABSTRACT

In my research project, I chose to delve into LIN Bus (Local Interconnect Network) communication, predominantly used in the automotive industry as a cost-effective alternative to the more expensive CAN Bus system. The LIN consortium, formed in the nineties by five automotive brands, has since standardized LIN Bus. My aim is to investigate the application of the transmitter chip on a microcontroller.

I began by researching the necessary components for the task. After selecting the components, I ordered them and commenced testing with various tasks, each progressively more difficult. These tasks involved programming the microcontroller and the transmitter chip.

The goal is to understand the communication process in modern vehicles, where there is an increasing need for sensor reading and device control.

Keywords: LIN Bus communication, LIN, Arduino Uno, MCP2003

KAZALO

1	UVOD	7
1.1	Predstavitev raziskovalnega problema	7
1.2	Delovanje komunikacije	7
1.3	Načrt za delo	8
1.4	Hipotezi	8
1.5	Raziskovalne metode	9
2	OSREDNJI DEL NALOGE	10
2.1	Začetek dela	10
2.2	Opis komponent	10
2.2.1	Arduino Uno	10
2.2.2	MCP2003	10
2.2.3	Adafruit OLED zaslon	12
2.3	Prvo vezje	12
2.4	Drugo vezje	14
2.5	Tretje vezje	16
3	RAZPRAVA	19
4	ZAKLJUČEK	20
5	VIRI IN LITERATURA	21

KAZALO SLIK

Slika 1:	Prikaz uporabe LIN Bus kot podomrežje CAN Bus	8
Slika 2:	Uporaba več MCP200X oddajniških čipov	11
Slika 3:	Podnožje MCP2003A IN MCP2004A	11
Slika 4:	Vezje, izdelano na preizkusni ploščici	13
Slika 5:	Načrt za prvo vezje	14
Slika 6:	Vezje, izdelano na preizkusni ploščici	15
Slika 7:	Načrt za drugo vezje	16
Slika 8:	Tretje vezje	17
Slika 9:	Načrt za tretje vezje	18
Slika 10:	Signal na vodilu s Picoscopom	18

1 UVOD

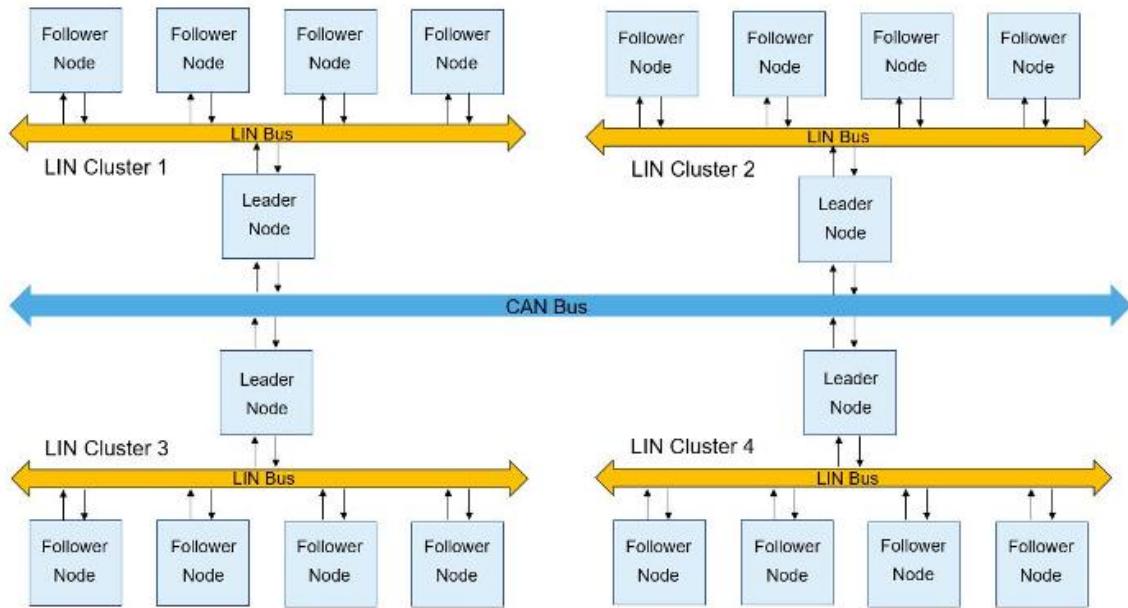
1.1 Predstavitev raziskovalnega problema

Odločil sem se raziskati LIN Bus komunikacijo, ki se uporablja v avtomobilski industriji za namene branja senzorjev in krmiljenja naprav, to sem želel narediti s komponentami, ki jih lahko priključim na mikrokrmilnik, kot je Arduino Uno. Ideja za nalogo se je porodila pri eni izmed šolskih ur predmeta, pri katerem obravnavamo programiranje. Pri izdelavi sem razmišljal o tem, kako bi lahko s komunikacijo krmilil element ter kako bi lahko bral neko vrednost, tako kot bi to počeli v avtomobilu. Glavni namen je bil narediti koncept, kako izgleda uporaba te komunikacije v avtomobilih. Delo raziskovalne naloge je zajemalo zbiranje idej, raziskovanje komponent, programiranje in izdelavo vezij.

1.2 Delovanje komunikacije

LIN Bus je komunikacija, ki se večinoma uporablja v avtomobilski industriji, za katero je bil tudi narejen za manjšanje stroškov zaradi vse večje rasti števila elementov, ki so morali biti krmiljeni. LIN se v avtomobilih uporablja za krmiljenje tistih segmentov, kjer lahko z enim mikrokrmilnikom (t.i. master) krmilimo do 15 naprav (t.i. slave). LIN segmente lahko povezujemo tudi v večja omrežja z uporabo CAN Bus. Slika 1 prikazuje, kako izgleda komunikacijska struktura v avtomobilu.

LIN bus je deluje z najvišjo hitrostjo 20 kbit/s, zato je najbolj primeren za uporabo v primerih, ko ni potrebno imeti velike pasovne širine in hitrosti, to pa so ponavadi nekritične funkcije v vozilih. Uporablja se tako, da je dominantni mikrokrmilnik s podrejenimi mikrokrmilniki povezan z enim vodnikom, kar nam olajša kompleksnost vodenja. Glavni problem z LIN Bus je, da če imamo težave z dominantnim mikrokrmilnikom, nam je delovanje celotne komunikacije onemogočeno.



Slika 1: prikaz uporabe LIN Bus kot podomrežje CAN Bus-u

(Vir: <https://community.nxp.com/t5/Blog/101-Local-Interconnect-Network-LIN/ba-p/1284877>)

1.3 Načrt za delo

Pri LIN Bus gre za half-duplex komunikacijo, zato sem potreboval dominantni mikrokrmlilnik (t.i master) in podrejeni mikrokrmlilnikom (t.i. slave). Ideja je, da s tipko, ki je priključena na dominantni krmilnik, izvedemo ukaz, s katerim zahtevamo, da podrejeni krmilnik izvede želen ukaz; ta lahko vklopi element ali pa prebere vrednost in nam pošlje nazaj povratno informacijo, ki se nam prikaže na dominantnem krmilniku.

1.4 Hipotezi

- Z dominantnim krmilnikom vključimo diodo na podrejenem krmilniku preko LIN Bus.
- Z dominantnim krmilnikom preberemo vrednost na senzorju na podrejenem krmilniku preko LIN Bus.

1.5 Raziskovalne metode

Raziskovalna naloga se je začela z iskanjem potrebnih elementov za izvedbo naloge. Začel sem raziskovati, kako deluje MCP2003 čip, ki sem ga potreboval za LIN Bus signal, ter kako ga dejansko lahko uporabim z mikrokrmlnikom Arduino UNO. Ko sem ugotovil, kako narediti vezje, sem se lotil programiranja v Arduino IDE okolju, kjer sem vzpostavil komunikacijo med dominantnim in podrejenim krmilnikom. Z napredovanjem pri izdelavi raziskovalne naloge sem uporabil osciloskop, da sem lažje razumel delovanje vezij in programov.

2 OSREDNJI DEL NALOGE

2.1 Začetek dela

Začetek naloge je bilo načrtovanje, kako izvesti zadane naloge. Najprej sem naredil dve vezji, vendar sem moral poiskati način, kako napisati program za krmilnika, da bom lahko pošiljal signal na oddajniški čip in s katerim bom lahko izvajal zastavljene naloge. Tega izziva sem se lotil z iskanjem knjižnic, vendar sem ugotovil, da primerne knjižnice za ta projekt ni, zato sem se moral poslužiti forumov.

2.2 Opis komponent

2.2.1 Arduino Uno

Odločil sem se, da je najbolje uporabiti mikrokrmlnik Arduino Uno, saj je cenovno dostopen in sem ga spoznal v šoli.

Arduino Uno je mikrokrmlnik z vgrajenim s čipom ATMega328p, ki deluje s hitrostjo 16 MHz. Na mikrokrmlniku imamo 14 digitalnih vhodov in izhodov z napetostjo 5 V in enosmernim tokom 20 mA, 6 od teh pa lahko uporabimo kot izhode s pulzno širinsko modulacijo. Ima tudi 1 kb velik EEPROM, ki ne izgubi vsebine, če je mikrokrmlnik ugasnjen. Uporabimo ga tako, da se z USB tipa B Arduino priključimo v računalnik in nanj naložimo kodo, nato ga lahko uporabljamemo tako, da ga napajamo z računalnikom ali ga priključimo na napajalnik (7 – 12 V).

Za izvajanje komunikacije LIN Bus sem uporabil UART, ki ga Arduino Uno ponuja.

Za prikazovanje vrednosti na zaslonu sem uporabil I2C, ki jo Arduino Uno ponuja.

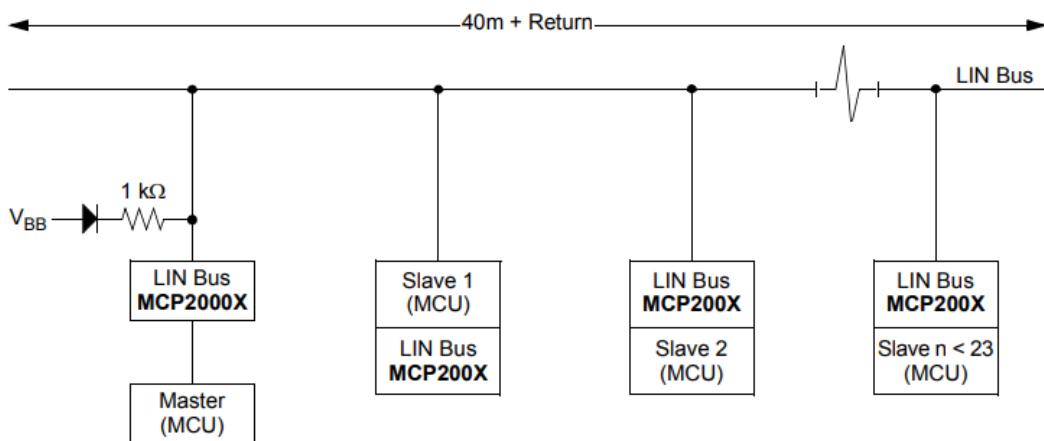
2.2.2 MCP2003

MCP2003 je čip, ki ima funkcijo oddajnika LIN signala. Čip se uporablja kot dvosmerno half-duplex fizični vmesni. Proizvajalec ga promovira kot komponento, ki je specifično oblikovana za uporabo v avtomobilski industriji. Za izdelavo te naloge je bil uporabljen MCP2003A, ki ga proizvajalec odsvetuje za izdelavo novih izdelkov, saj ga je nadomestil MCP2003B.

Na čipu so bili uporabljeni:

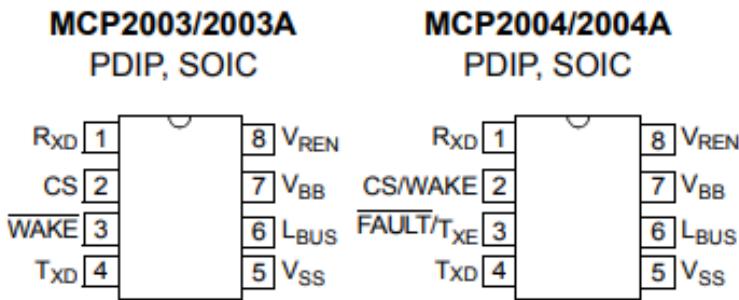
- Rx, ki je izhod za prejem podatkov.

- Tx, ki je vhod za oddajo podatkov.
- CS, ki je namenjen vklopu delovanja čipa. V primeru, da je na njem logična ničla je čip v stanju pripravljenosti.
- V_{SS} je namenjen minusu.
- V_{BB} je namenjen plusu.
- L_{BUS} je izhod z LIN Bus. Na vodilo oddajniških čipov v vezju je potrebno priklopiti $1\text{ k}\Omega$ upor.



Slika 2: primer uporabe več MCP200X oddajniških čipov

(Vir: <https://www.microchip.com/en-us/product/MCP2003A#>)



Slika 3: podnožje MCP2003A IN MCP2004A

(Vir: <https://www.microchip.com/en-us/product/MCP2003A#>)

Čip je zaščiten pred kratkimi stiki in previsokimi temperaturami z notranjim vezjem. Čip lahko uporabljamo do dolžine 40 m.

2.2.3 Adafruit OLED zaslon

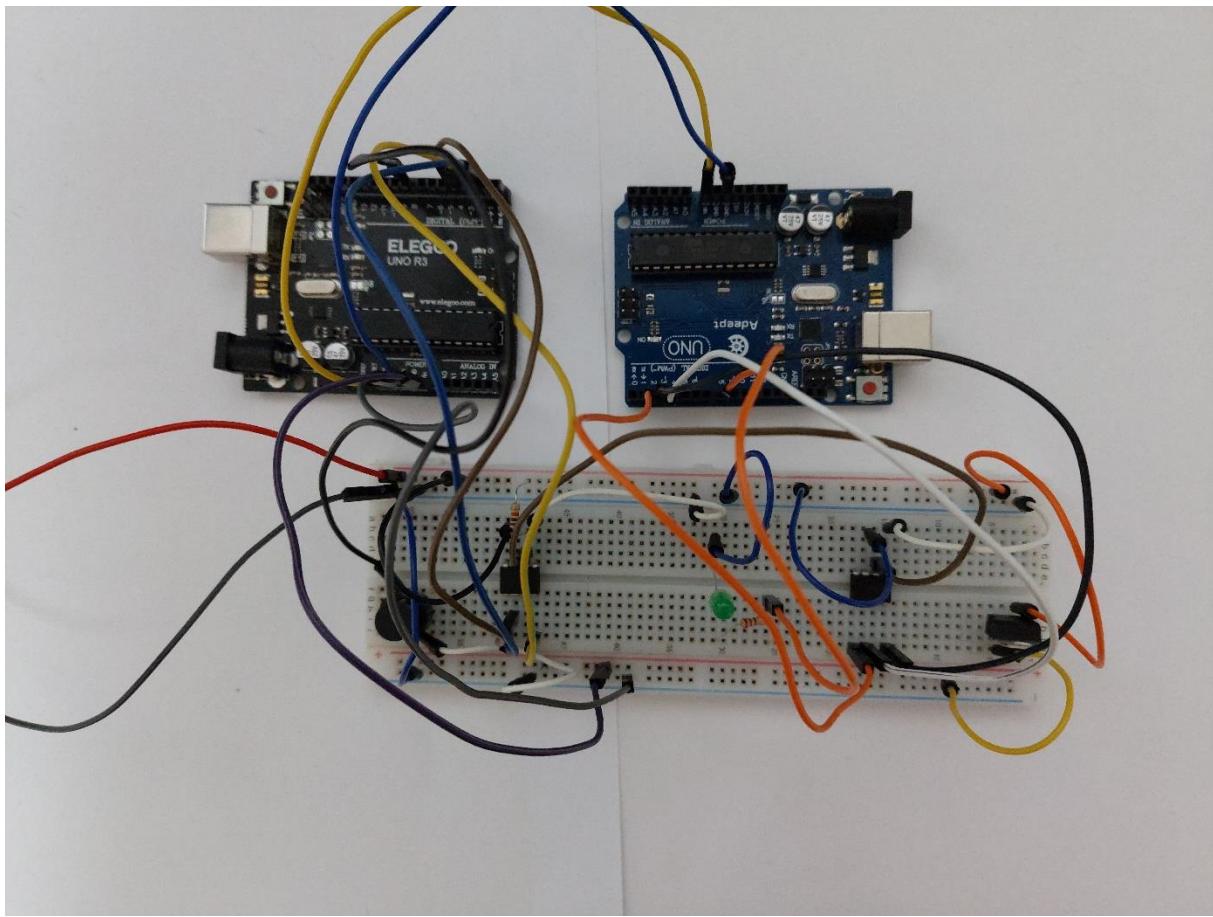
Za prikazovanje podatkov iz dominantnega mikrokrmlnika sem se odločil za uporabo OLED zaslona proizvajalca Adafruit. Zaslon ima diagonalo veliko 0,96 palca in ima 128 x 64 slikovnih pik. Zaslon za komuniciranje z mikrokrmlnikom uporablja I2C komunikacijo.

2.3 Prvo vezje

Prvo vezje sem izdelal na preizkusni plošči (Slika 2), s katerim sem želel prižgati element, kot je led dioda, ki bi ob pritisku gorela v programu določen čas. S tem sem želel simulirati, kako bi deloval prižig nekih svetil. Ta del je ključen, saj sem v tej fazi prvič uspel vzpostaviti povezavo med mikrokrmlnikoma in s tem pridobil znanje za nadaljnje naloge.

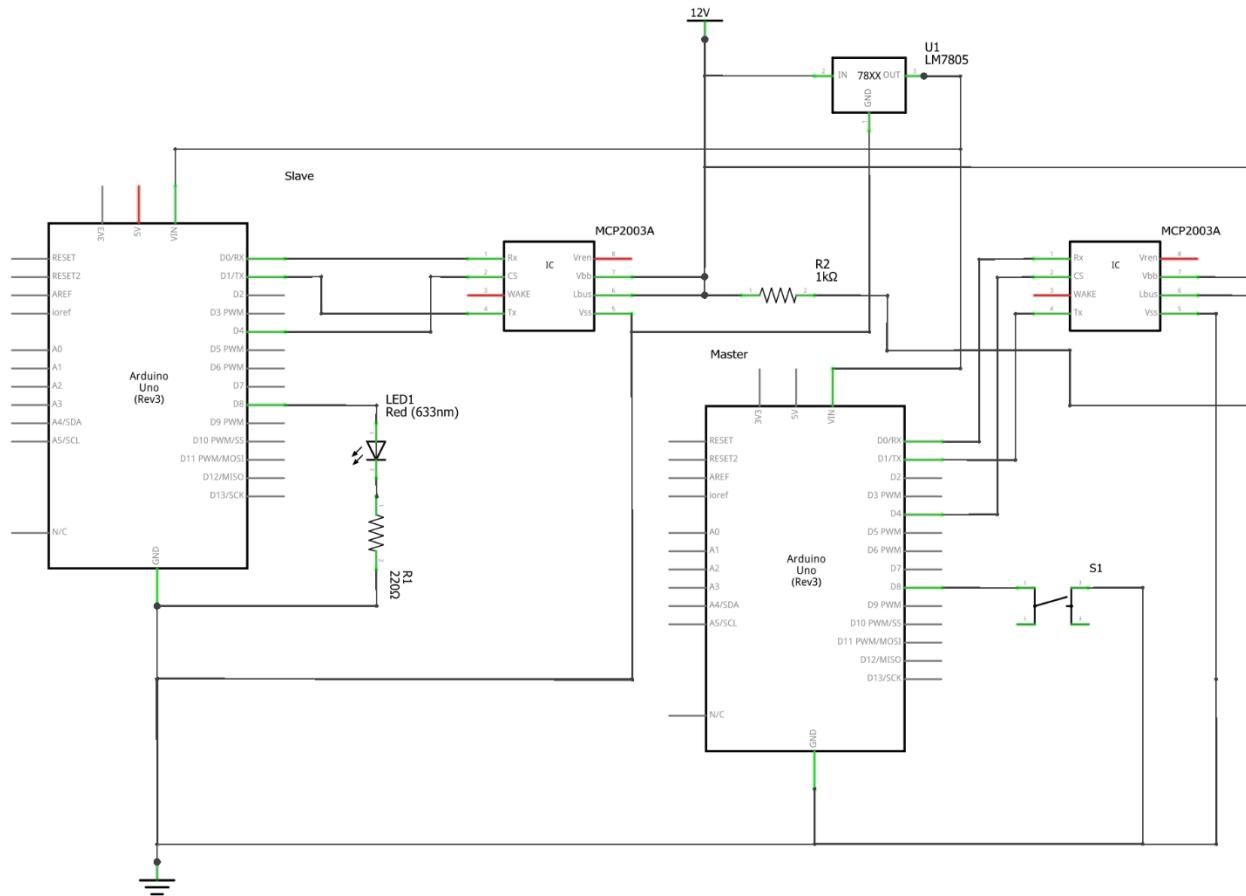
Pri izdelavi programa za prvi del sem se soočal s kar nekaj težavami, saj nisem našel primerne knjižnice, ki bi jo lahko uporabil za vzpostavitev komunikacije. Prve rešitve sem našel na forumih za Arduino okolje.

Program je narejen tako, da ko je tipka enkrat pritisnjena, bo program v mikrokrmlniku poslal zahtevo na podrejeni mikrokrmlnik, naj ta prižge led diodo za eno sekundo.



Slika 4: slika prikazuje vezje sestavljeni na preizkusno ploščico

(Osebni vir)



Slika 5: slika prikazuje načrt za vezje

(Osebni vir)

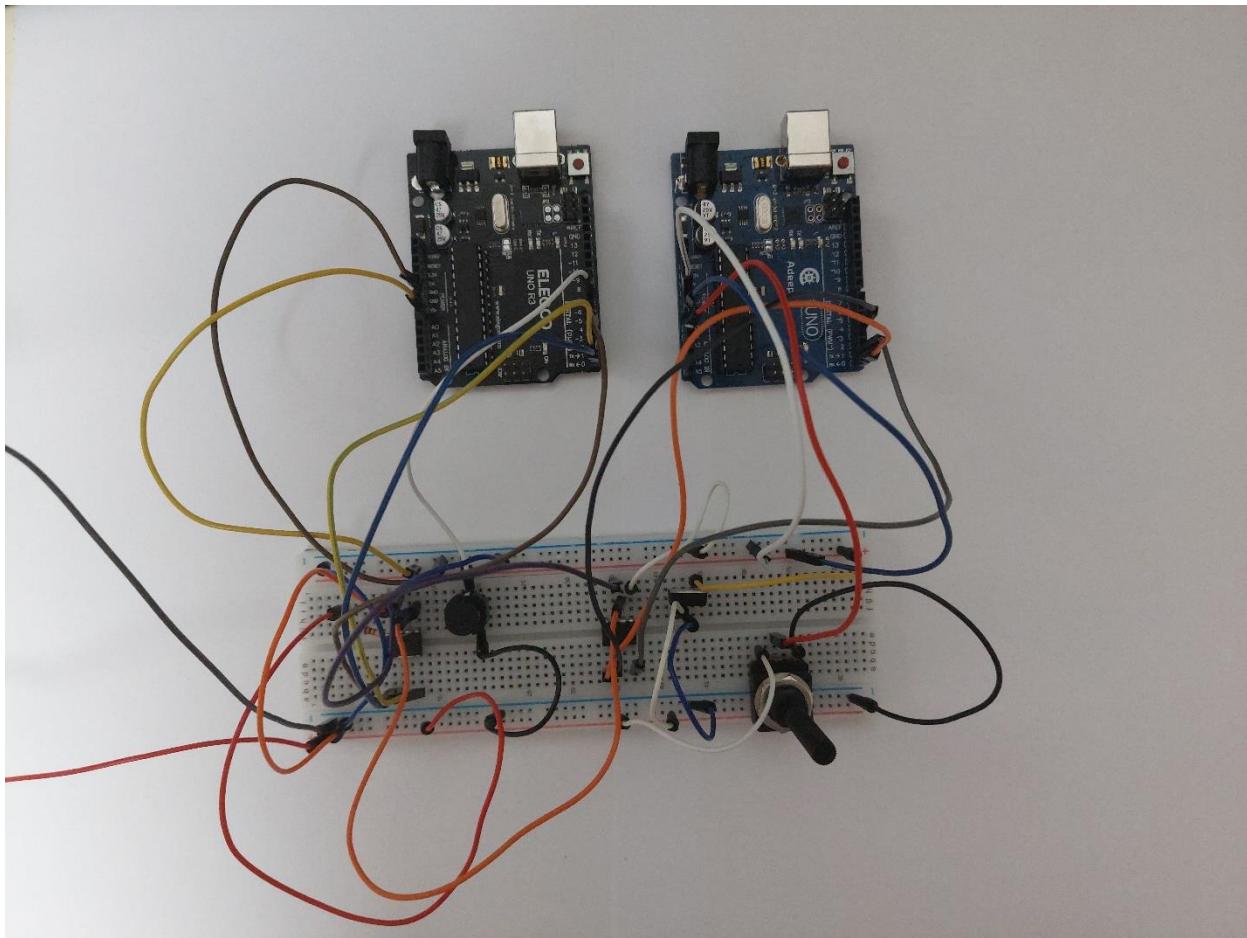
2.4 Drugo vezje

S pridobljenim znanjem iz prve naloge sem se lotil drugega dela naloge, in sicer branje senzorja.

Drugo vezje, ki sem ga prav tako izdelal na preizkusni plošči, je bilo takšno, da sem na podrejeni krmilnik priključil potenciometer, namenjen simulaciji nekega senzorja, ki bi ga našli v avtomobilu, npr. temperaturni senzor za merjenje temperature v vozilu. Pri tem sem moral upoštevati, da ne bo mogoče pošiljati ukazov preko primarnih UART vhodov, ki jih ima Arduino Uno in istočasno brati vrednosti, ki bi jih podrejeni krmilnik pošiljal na dominantnega. Rešitev za ta problem je bila uporaba knjižnice Software Serial, ki je omogočila, da sem lahko brez motenj na dominantnem krmilniku bral vrednosti, ki so bile merjene na podrejenem Arduino. Uporaba

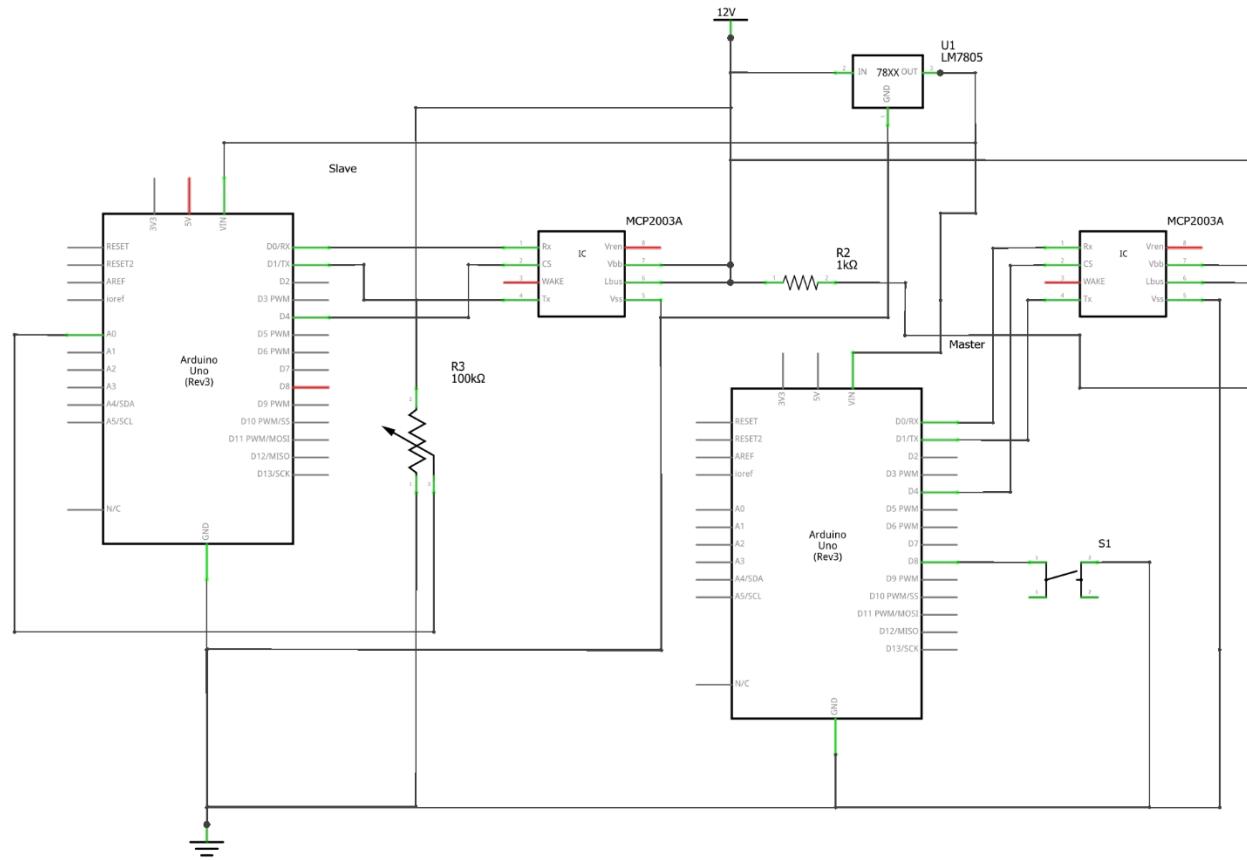
knjižnice Software Serial je pomenila le nekaj dodatnih vrstic v programu, pina Rx in Tx pa sta sedaj postala vhoda 2 in 3 na mikrokrmlniku.

Program za to vezje deluje tako, da ob pritisku tipke, ki je priključena na dominantni mikrokrmlnik, ta pošlje zahtevo na dominantni mikrokrmlnik, naj izmeri vrednost, ki je tisti trenutek na potenciometru. Ko podrejeni mikrokrmlnik izmeri vrednost, jo nato pošlje dominantnemu mikrokrmlniku nazaj, ta pa bo vrednost izpisal v serijskem monitorju v Arduino IDE okolju.



Slika 6: slika vezja izdelanega na preizkusni ploščici

(Osebni vir)



Slika 7: slika načrta za drugo vezje

(Osebni vir)

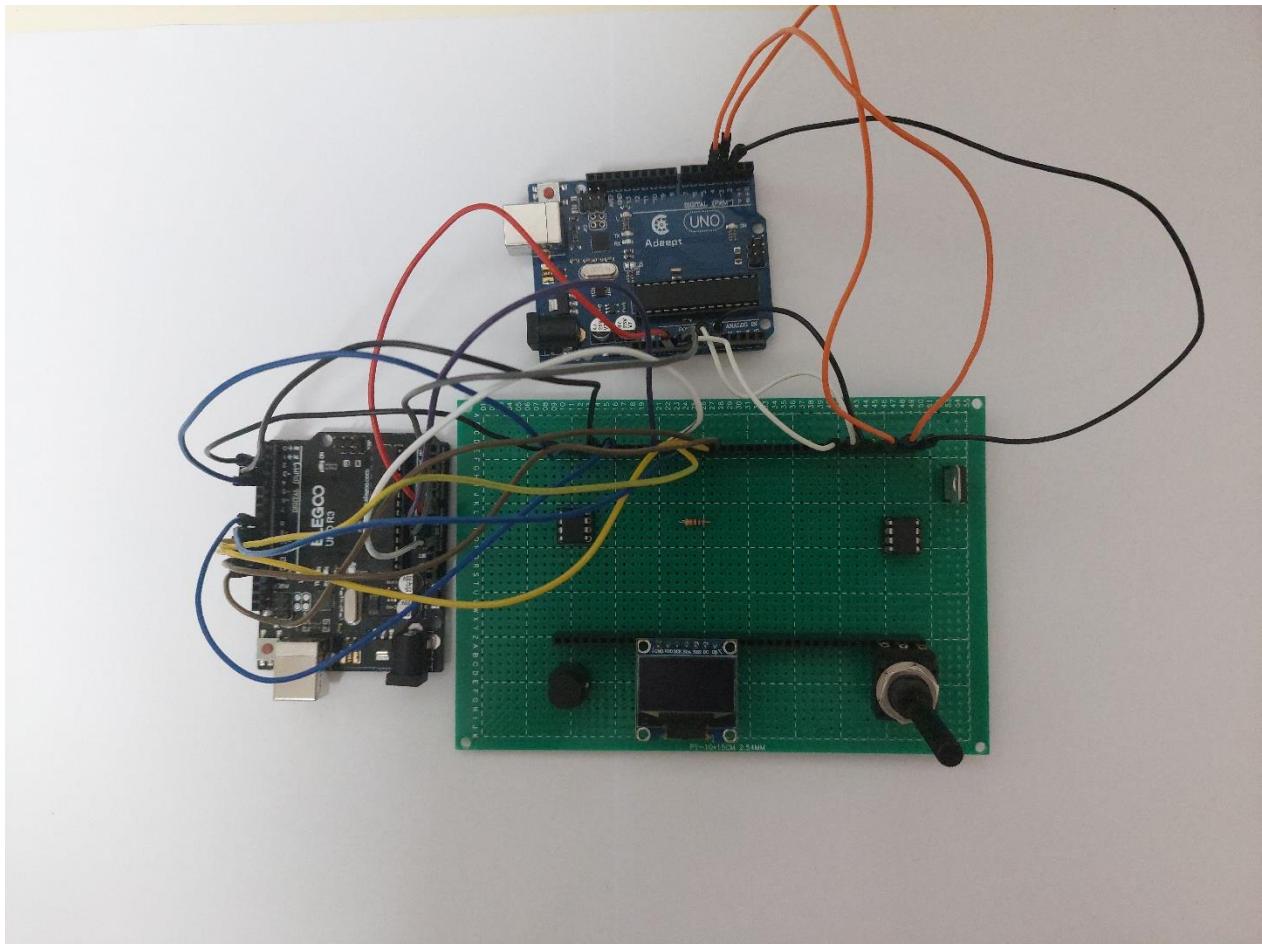
2.5 Tretje vezje

S končanim drugim vezjem sem se lotil tretjega in končnega vezja, tj. izdelka, ki prikazuje delovanje LIN komunikacije. Za vezje sem uporabil enake komponente kot prej, le da sem tokrat na vezje dodal še LM7805 regulator in OLED zaslon, proizvajalca Adafruit. Komponente sem spajkal na vezje in s tem pripravil bolj prenosno in robustno vezje.

Izdelano vezje se uporablja tako, da ob pritisku tipke dominantni krmilnik pošlje zahtevo na podrejeni krmilnik, ta pa dominantnemu krmilniku vrne povratno informacije glede potenciometra. Dominantni krmilnik bo informacijo glede potenciometra izpisal na Adafruit zaslon.

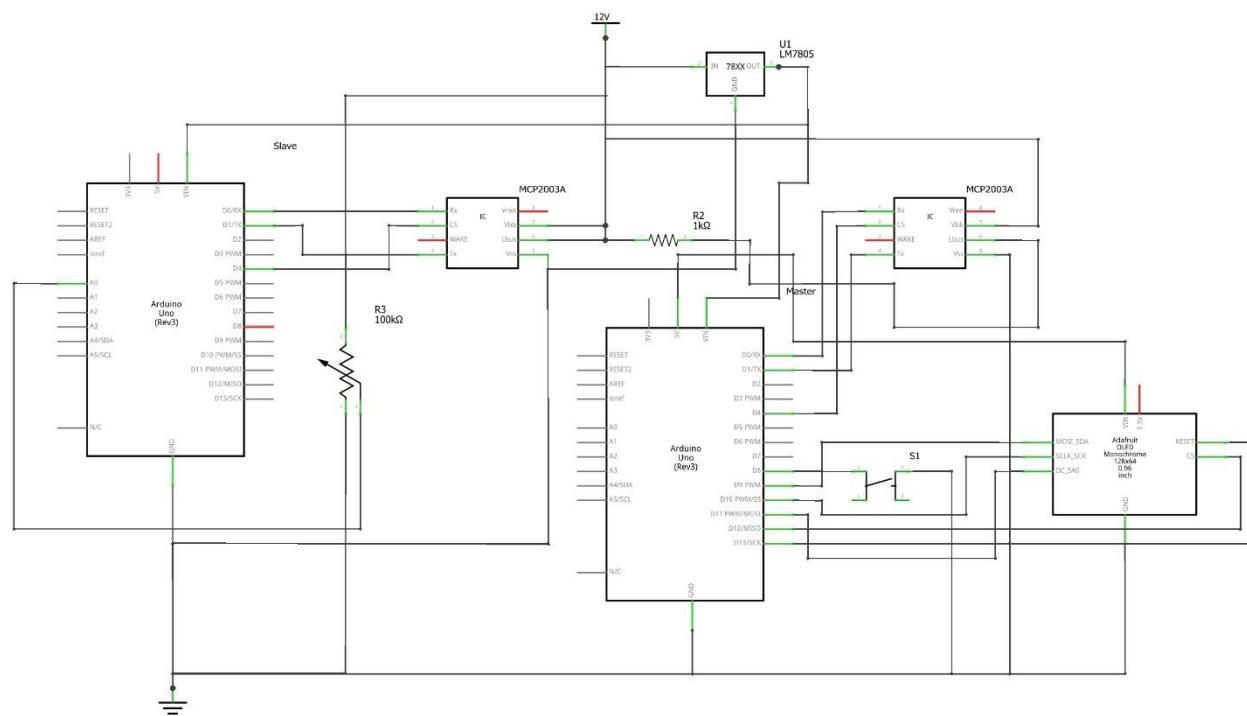
Program, ki sem ga napisal za to vezje, je isti, kot je bil za drugo vezje, le da je bil dodan del za izpisovanje vrednosti na zaslon namesto v serijski monitor v Arduino IDE.

Pri tem vezju sem z osciloskopom (Slika 8) izmeril tudi signal, ki je potekal med oddajniškima čipoma dominantnega in podrejenega krmilnika. LIN Bus signal je sestavljen iz okvirja, ki je sestavljen iz dveh delov glave in odziva. Glava se prav tako razdeli ne več delov, in sicer pavzo, sinhroniziran bajt in ID okvir. Na več delov se razdeli tudi glava, in sicer podatek in kontrolna vsota.



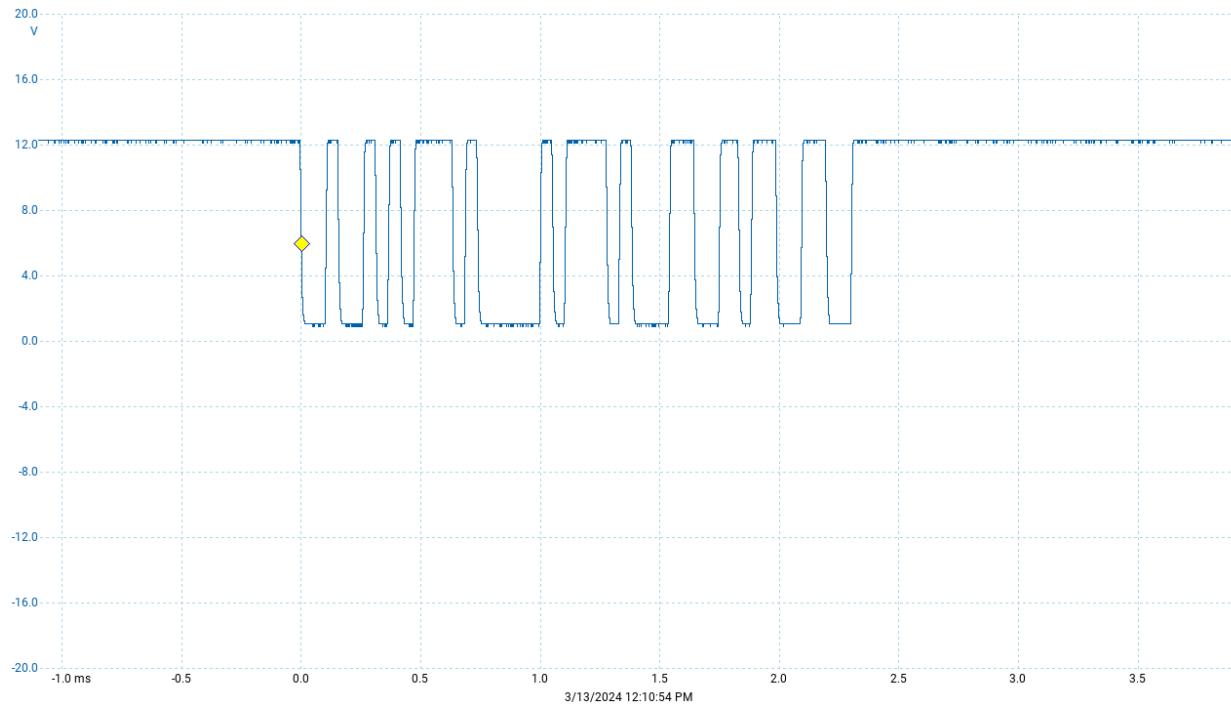
Slika 8: slika tretjega vezja

(Osebni vir)



Slika 9: slika načrta za tretje vezje

(Osebni vir)



Slika 10: slika signala s Picoscopom na tretjem vezju

(Osebni vir)

3 RAZPRAVA

Z nalogo sem želel ugotoviti, kako deluje komunikacija v vozilih in to mi je z zastavljanjem več vezij tudi uspelo. Pri izdelavi je največ časa vzelo prvo vezje, saj nisem vedel, kako bi se lotil izziva, da vzpostavim komunikacijo, zato je bilo na začetku veliko iskanja knjižnic in branja po forumih.

Pri nadalnjem delu večjih težav s programiranjem nisem imel, saj sem že pri drugem vezju bolje razumel delovanje programa. Pri izdelavi vezja so bile manjše spremembe priključitve na podrejeni mikrokrmilnik.

Tretje vezje sem naredil, ker sem želel imeti nekaj, kar je bolj robustno kot vezja, izdelana na preizkusni ploščici in neodvisno od računalnika, da lahko spremljamo vrednosti na dominantnem krmilniku. Prav tako je prikaz, kako deluje LIN Bus.

Vsa tri vezja predstavljajo fizično plast LIN Bus komunikacije, kjer je delovanje komunikacije po eni žici, naslednji korak pa bi bil programiranje LIN protokola, kar predstavlja tudi naslednjo največjo nadgradnjo raziskovalne naloge.

4 ZAKLJUČEK

Med izdelavo te naloge sem veliko naučil o delovanju komunikacije v avtomobilu, ko uporabljam različne funkcije.

Naloga je bila zame prva izkušnja s to temo, zato je bilo iskanje podatkov, preizkušanje različnih programov in programiranje Arduina zelo poučno. Naloga ima potencial, da se jo v prihodnje še razširi z izdelavo kakšnega izdelka, programiranja protokola LIN ali uporabo z CAN Bus.

Kljub zahtevnosti naloge in začetnim težavam z vzpostavitvijo programa za prvo vezje sem uspel potrditi prvo hipotezo, da se z dominantnim krmilnikom (Arduino Uno) da prižigati led diodo na podrejenem krmilniku.

Po uspešno izvedenem prvem vezju sem se lotil drugega izziva, in sicer pošiljanje podatkov med krmilnikoma. Nalogo sem uspešno izvedel in s tem potrdil drugo hipotezo, da lahko z dominantnim krmilnikom preberemo vrednost senzorja na podrejenem krmilniku preko LIN Bus.

5 VIRI IN LITERATURA

- [1] Adafruit OLED zaslon. [Online]. Adafruit Industries. [Pridobljeno 29. feb. 2024] Dostopno na spletnem naslovu: <https://www.adafruit.com/product/326>.
- [2] LIN Bus. [Online]. Wikipedija, prosta enciklopedija. [Pridobljeno 14. feb. 2024] Dostopno na spletnem naslovu: https://en.wikipedia.org/wiki/Local_Interconnect_Network.
- [3] LIN Bus in Picoscope. [Online]. Test & Measurement Tips. [Pridobljeno 1. mar. 2024] Dostopno na spletnem naslovu: <https://www.testandmeasurementtips.com/exploring-lin-bus-oscilloscope/>.
- [4] LIN Bus protokol. [Online]. AutoPi Blog. [Pridobljeno 3. jan. 2024] Dostopno na spletnem naslovu: <https://www.autopi.io/blog/lin-bus-protocol-explained/>.
- [5] LIN Bus signal z osciloskopom. [Online]. Test & Measurement Tips. [Pridobljeno 29. feb. 2024] Dostopno na spletnem naslovu:
<https://www.testandmeasurementtips.com/exploring-lin-bus-oscilloscope/>.
- [6] LIN in CAN Bus. [Online]. Prodigy Techno. [Pridobljeno 29. feb. 2024] Dostopno na spletnem naslovu: <https://www.prodigytechno.com/difference-between-lin-can-and-flexray-protocols>.
- [7] LM7805. [Online]. Fairchild Semiconductor. [Pridobljeno 28. feb. 2024] Dostopno na spletnem naslovu: <https://pdf1.alldatasheet.com/datasheet-pdf/view/82833/FAIRCHILD/LM7805.html>.
- [8] MCP2003. [Online]. Microchip Technology. [Pridobljeno 1. mar. 2024] Dostopno na spletnem naslovu: <https://www.microchip.com/en-us/product/MCP2003A#>.
- [9] Mikrokrumilnik Arduino. [Online]. Arduino. [Pridobljeno 29. feb. 2024] Dostopno na spletnem naslovu: <https://www.arduino.cc>.