

UNIVERZA V LJUBLJANI
FAKULTETA ZA ELEKTROTEHNIKO,
FAKULTETA ZA MATEMATIKO IN FIZIKO,
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO TER
ISYSTEM LABS d.o.o.

SISTEM ZA ZAZNAVANJE NEVARNOSTI V PROMETU

Mentor:
izr. prof. dr. MOŽEK MATEJ, univ.
dipl. inž. el.

Avtorji:
HEDŽET KOSTAJNŠEK ŽAN
VIŠNAR DOMEN
VUHERER MATJAŽ

Ljubljana, april, 2021

Razvojno poročilo	3
Zastavljena rešitev	3
Implementacija rešitve	6
Testno poročilo	9
Unit testi	9
Integracijski testi	10
Projektno poročilo	10

Razvojno poročilo

V času, ko eksponentna rast zanimanja za alternativne načine prevoza postaja standard, se vzporedno z napredkom, na tem področju, odpirajo nova vprašanja o varnosti udeležencev v prometu. Razvoj samovoznih tehnologij in njihov prodor v realnost, množična uporaba e-koles, električnih skirojev ter drugih alternativnih prevoznih sredstev, posameznika vzpodbujajo k prilagajanju spremembam, ki v celoti tvorijo povsem novo realnost. Zaradi človekove nezmožnosti za –dovolj– hitro prilagajanje le-tem, pa je ta tranzicija iz že tako nevarnega prometa, za mnoge težavna in predstavlja povečano nevarnost v prometu. Nedvomno je “zvezda” tako mestne e-mobilnosti, kot tudi e-rekreacije, električno kolo. Slednje pa, zaradi enostavne konstrukcije in z e-mobilnostjo povezanimi povečanimi hitrostmi, predstavlja nevarnost hujših nesreč v prometu. Pogosti primeri rizičnih situacij so nepregledna križišča, prehitvajajoča vozila, sekanje ovinkov ipd.

Pri podrobnejši analizi teh situacij lahko pridemo do zanimivega zaključka, da sta edina podatka, ki bi lahko poljubna dva udeležence v prometu obvarovala pred nesrečo, hitrost in lokacija relativno drug drugega.

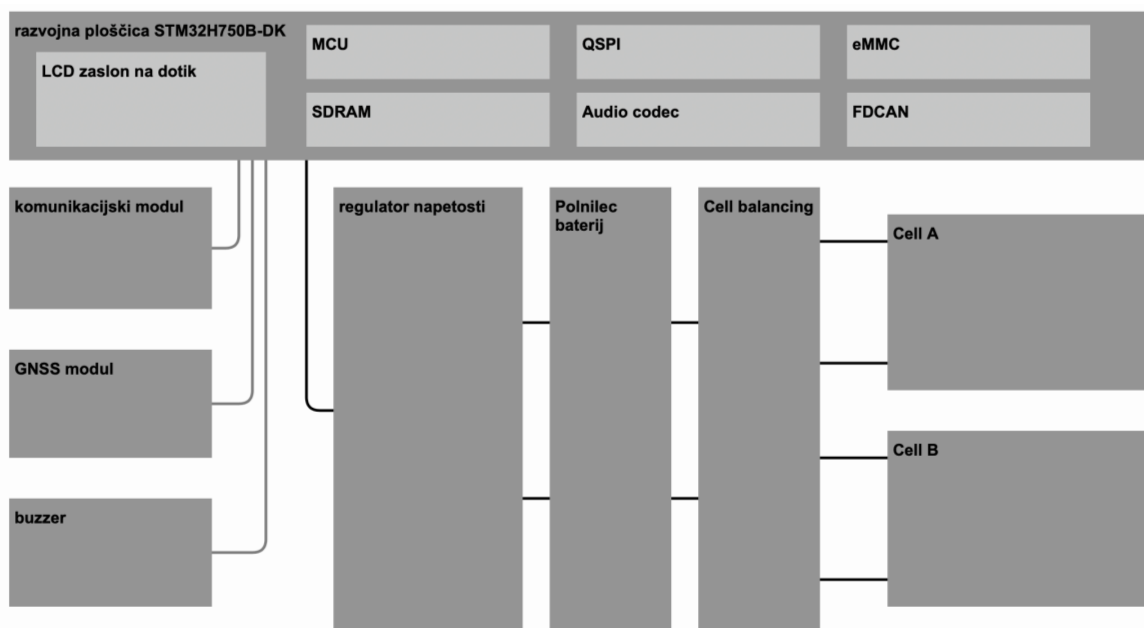
Zastavljena rešitev

Zdi se, kot, da je logičen odrešitelj slednje problematike razvoj avtonomne vožnje. Mar je ta avtonomnost zgrajena na osnovi računalniškega vida ali lidar tehnologije, nas težave v implementaciji, katerim se v tem poročilu na žalost ne bomo posvečali, opozarjajo na časovni obseg, ki ga bo zajel razvoj pred širšim lansiranjem tehnologije.

Četudi je avtonomna vožnja zgolj stvar prihodnosti, člani ekipe odločno zavračamo mišljenje, da je čas, ki nas loči od avtonomnosti, obdobje rastoče nevarnosti v prometu. Prav tako zavračamo idejo, privilegizacije varnosti določenih skupin udeležencev prometa kar tolmačimo kot možnost uprabe aktivnih varnostnih sistemov zgolj v novejših vozilih in ali prestižnejših vozilih. Kljub direktivam, ki novejše aktivne varnostne sisteme uvajajo v vsa novoizdelana vozila, čutimo pomankanje osredotočenosti na področju varnosti drugih udeležencev prometa kot so kolesarji, pešci ipd.

Stremimo torej k napravi, ki je lahko varnostni vmesnik za vse udeležence prometa. V razmisleku smo se spomnili na letalstvo in sicer predvsem na manjša plovila s katerimi imamo nekaj izkušenj tudi sami. V tej kategoriji letalstva je namreč značilno, da plovila letijo z majhno medsebojno razdaljo. Kljub slednjemu dejstvu pa je nevarnost za trk, zaradi obvezne uporabe tehnologije FLARM oz. “Flight Alarm”, precej majhna. Sistem posamezni dve plovili, v primeru bližjega srečanja, opozarja na razdaljo med njima. Podobna implementacija varnostnega sistema bi ustrezala našim zahtevam, le da smo morali implementacijo iz treh dimenzij, prenesti na dve.

Dejstvo je, da bomo zaradi njegove preciznosti in zanesljivosti, za namene pridobivanja lokacije in vektorjev smeri ter pospeška, uporabili sistem GNSS. Poleg sistema GNSS pa bomo za voljo povečane zanesljivosti po potrebi uporabili tudi senzor tlaka. Povsem drugače pa je potekala izbira komunikacijskega modula, kjer smo naleteli na prvo težavo. Zavedali smo se, da mora komunikacijski modul brezhibno skrbeti za neprekinjeno komunikacijo med poljubnima dvema uporabnikoma. Za začetek smo način pretoka informacij med napravami (šlo je torej za vprašanje o tem ali bo komunikacija potekala preko strežnika ali ne) prepustili usodi. Dejstvo je, da mora delovati v tako urbanem okolju z visoko onesnaženostjo s signalom in možnostjo izpada signala, kot tudi na gorskih cestah, kjer obstaja možnost, da se med dvema voziloma znajde večja količina materiala, kar bi lahko oviralo pretok podatkov.

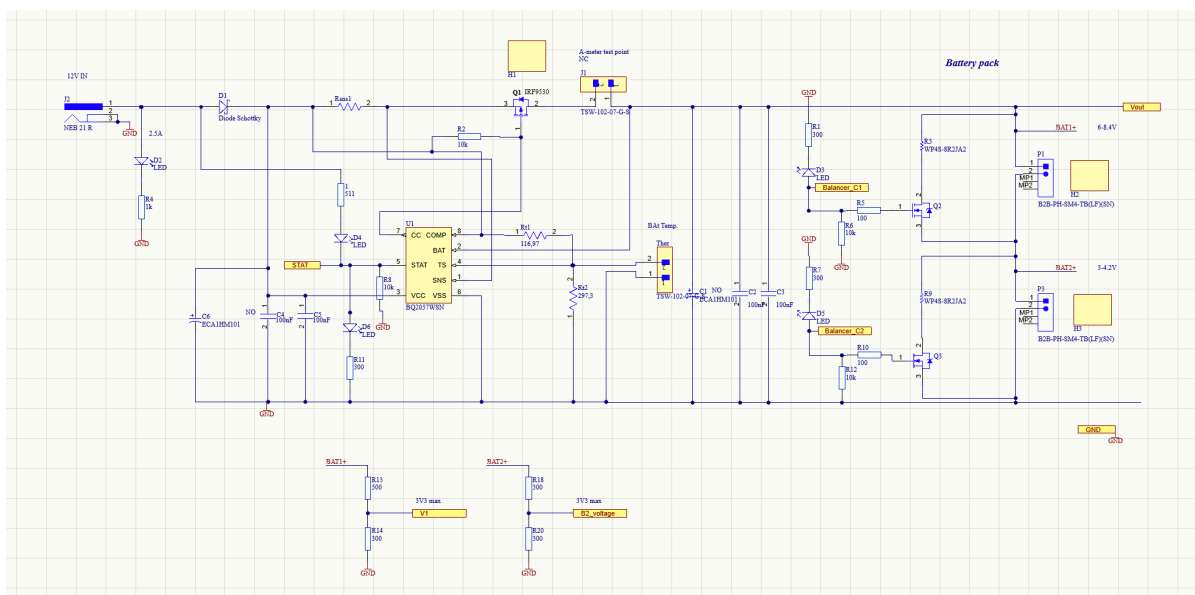


Slika 1: blokovna shema koncepta naprave

Zaradi počasnosti, nepokritosti s signalom in visoke porabe moči telefonskih standardov komunikacije, smo najprej ovrgli uporabo le-teh. Zaradi nezanesljivosti in majhnega dometa, smo za tem ovrgli še uporabo 433MHz komunikacijskih protokolov. Nazadnje smo, odločeni, da je ta zaradi dobrega obnašanje v zastavljenih urbanih in gorskih okolica najboljše, svojo pozornost posvetili drugim standardom komunikacije zgrajenih na osnovi frekvence 2,4GHz.

Zaradi uporabe ARM arhitekture linije procesorjev STM32 smo na naslednjo težavo naleteli že pri implementaciji map. Zavedali smo se namreč, da je poleg uporabe lokacije in preračunanih vektorjev, zaradi izjemno majhnih razdalj med cestami, nujna tudi uporaba zemljevidov. S pomočjo zemljevidov bomo lahko iz cest matematično konstruirali krivulje na katere bomo umestili udeležence prometa. Ker se zavedamo omejenosti procesorske moči takšnih procesorjev smo se odločili, da bomo procesorju dovedli layer zemljevida, ki vsebuje zgolj ceste. Takšne datoteke smo si priskrbeli na TomTom Developers vmesniku, ki umogoča tako prenos le-teh preko LTE omrežja kot tudi prenos datotek. Kljub pomisleku, da bi uporaba 4G M2M komunikacijskega modula, ki ga v sodelovanju ponujata TomTom in STM, omogočala dobro uporabniško izkušnjo, smo idejo uporabe telekomunikacijskih standardov zaradi prej navedenih razlogov ponovno zavrnil in odločili, da bodo mape posnete na pomnilnik procesorja ali pomnilniški kartici (v primeru, da bodo le-te obsežnejše).

Z zadnjo konceptualno težavo so se srečali pri samem algoritmu. Določili smo, da bomo iz cest konstruirali krivulje, posamezen udeleženec prometa pa bo zahteval določene podatke kot so lokacija, vektor hitrosti in pospeška od vsakega udeleženca prometa, ki se relativno na njegovo hitrost, nahaja v neki delta okolici. Hkrati bo ta udeleženec prometa svoje parametre poslal vsem drugim udeležencem, ki bodo to zahtevali. Naprava posameznega udeleženca prometa bo nato procesirala pridobljene podatke in s pomočjo algoritma po potrebi opozorila na nevarnost trka. Ker gre za varnostni sistem smo pri programiranju morali določiti prioritete naprave. Za voljo izvršitve algoritma mora naprava izvrševati pet procesov: branje poslanega GPS, branje podatkov prejetih preko komunikacijskega modula XBEE PRO, branje analognih vrednosti kar potrebujemo za nastavljanje zaslona in ponastavljanje GPS modula. Pri tem je potrebno GPS modul ponovno ponastavljati, ker le-ta nima svojega spomina. GPS modul pri tem ponastavljamo s pošiljanjem

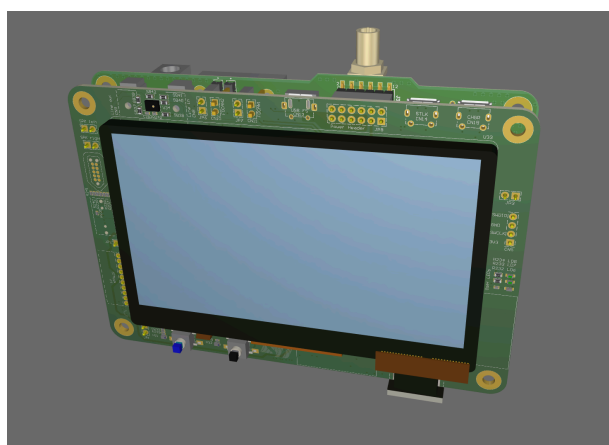


Slika 2: shema baterijskega vezja narisana v programu Altium Designer (Vir: posnetek zaslona)

nastavitvene datoteke preko serijskega vmesnika, po končanem procesu pa se sprostijo vsi semaforji za delovanje ostalih procesov.

Pri prej omenjenih procesih imata največjo pomembnost procesa, ki se ukvarjata z branjem in pridelavo podatkov za tako GPS kot tudi XBEE modul. Pri tem se posamezen proces sproži ko je na serijskem vhodu "mir" (torej USART IDLE). Takrat se odpre semafor, ki opravlja proces za obdelavo prejetih podatkov (kar smatramo kot GPS in XBEE). Pridelani podatki se shranijo v lastno strukturo, ki je potrebna za za posredovanje procesu, ki skrbi za delovanje zaslona. Za omenjenimi procesi se izvede proces, ki skrbi za prikazovanje podatkov na zaslonu. Najmanjšo prioriteto ima proces branja analognih vrednosti, ki skrbijo za uravnavanje napetosti na baterijah.

Marsikdo se mogoče sprašuje kakšna je razlika med zgoraj opisanim konceptom naprave in mobilnim telefonom. Že res, telefon omogoča tako način komunikacije med drugimi napravami kot tudi GNSS podatke a uporabo telkefona za varnostni sistem v prometu ne vidimo kot zadosti zanesljivo rešitev. Prepričani smo namreč, da mora naprava takšnega tipa neprestano brezhibno delovati. Ker se pri mobilnem telefonu ne moremo zanesti na napolnjenost naprave, brezhibnost procesiranja in komunikacije, takšne rešitve ne smatramo kot ustrezne.

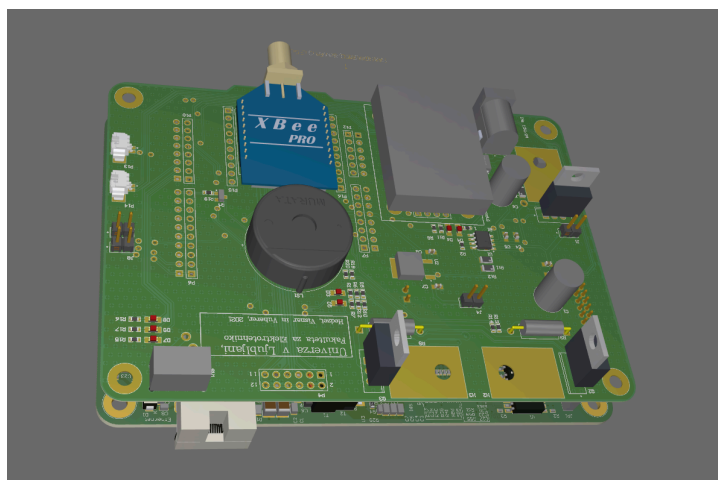


Slika 3: tiskarina in razvojna ploščica (Vir: posnetek zaslona)

Poleg prej naštetega vidimo v prihodnosti možnosti za izboljšanje naprave, ki jim uporaba telefona ne zadošča. Možnost integracije naprave v vozilo ali povezovanje naprave z vozilom s pomočjo CAN komunikacijskega protokola, predstavljajo možnost za tako integracijo naprave v vozilo, kot tudi možnost za pocenitev naprave same. Z napravo povezano na vozilo bi namreč lahko nekatere dele neše naprave kot so GNSS modul in napajanje nadomestili z že obstoječimi v vozilu.

Implementacija rešitve

Pri izbiri procesorja smo odločili, da bomo uporabili razvojno ploščico STM32H750B-DK, ki nam poleg samega procesorja zagotavlja tudi LCD zaslon na dotik, ki ga bomo uporabili kot uporabniški vmesnik, FDCAN s pomočjo katerega bi lahko napravo povezali na vozilo in ustrezno velik



Slika 4: tiskanina (Vir: posnetek zaslona)

pomnilnik, ki nam omogoča shranjevanje večjih map. Neposredno na razvojno ploščico smo priključili še Ublox NEO M8 GNSS modul in modul za 2,4GHz komunikacijo XBEE PRO..

Za voljo kompaktnosti in enostavnosti smo s programsko opremo Altium Designer 21, izdelali tiskanino, ki se vstavi neposredno na spodnjo stran razvojne ploščice. S tem smo omogočili ustrezne stike med ploščicama in zagotovili kompaktnost same naprave. Pri risanju tiskanine smo si pomagali s programsko opremo Saturn PCB Toolkit. Napajanje ploščic smo zagotovili na podlagi dveh zaporedno vezanih Li-Ion baterijskih celic, vsake s kapaciteto 2Ah. Ker se je izkazalo, da zaščitno vezje, integrirano v baterije, ne omogoča polnenja z napravo, smo baterije prevezali tako, da smo zaščitno vezje "obvozili". Kljub temu, da je uporaba dveh celic zapletla upravljanje bateri, saj smo morali urediti uravnavanje napetosti na celicah (cell balancing), nam je višja napetost 8,4V namesto 4,2V, precej poenostavila kasnejše uravnavanje napetosti na 5V in 3,3V.

Za napajanje dvocelične baterije smo izbrali čip BQ2057WSN, ki omogoča polnenje dveceličnih Li-Ion baterij s tokom do 2A. Polnenje naprave pa smo omogočili preko 11mm konektorja in 12V z varovanjem pred nasprotno napetostjo.

Na ploščico smo dodali še dodatno uporabniško signalizacijo z piezom in par dodatnih konektorjev z I2C in SPI kontakti, ki bi nam potencialno lahko prišli prav v nadaljnjem razvoju.

	Opis	Količina
--	------	----------

BQ2057WSN	Linearni polnilec dvoceličnih baterij	1
IRF9530	P-channel mosfet	1
IRFB31N20DPBF	N-channel mosfet	2
WP4S-8R2JA2	4W upornik	2
LD29150DT50R	1,5A 5V low drop voltage regulator	1
STM32H750B-DK	Razvojna ploščica	1
Piezo	Piskač	1
Ublox NEO M8	GNSS modul	1
XBEE PRO	Komunikacijski modul	1

Tabela 1: kosovnica pomembnejših elektronskih komponent (količina na eno napravo)

Pri nakupu in pridobitvi opreme ter delov, smo uporabili več virov. Na ponudbo iSystem izziva smo si izbrali omenjeno razvojno ploščico, ponujeno vrednost pri trgovcu IC Elektronika, pa smo porabili za nakup materiala za spajkanje in del komponent kasneje uporabljenih pri razvoju tiskanine. S pomočjo profesorja dr. Mateja Možeka, ki nam je omogočil tudi kvalitetno mentoriranje, in Fakultete za elektrotehniko, Univerze v Ljubljani, smo si zagotovili dva kosa tiskanine izdelane pri podjetji Svet elektronike, preostale komponente potrebne za “naselitev” tiskanin, naročene pri podjetji Farnell ter dva kosa 3D sprintanega ohišja, izdelanega na fakulteti. Preostale dele naprave, torej pet kosov baterij naročenih pri podjetju Hobbyking, dva kosa XBEE PRO modulov, naročenih pri podjetju Farnell in dva kosa GNSS modulov. Vso ostalo opremo, ki smo jo uporabljali za spajkanje in razvoj, kot so spajkalnik, osciloskop, laboratorijski napajalnik ipd., smo si lastili že pred spopadom z izzivom.

Relativno visokim stroškom navkljub, menimo, da je, glede na zmožnosti in kvaliteto izdelave, vsak evro, prispevan tako iz naših žepov, kot tudi tisti prispevani iz strani fakultete in podjetja iSystems, upravičen, saj ima izdelana naprava potencial za nadaljno raziskovanje in možnost uporabe naprave v realnem okolju. Še več, menimo, da bi bila implementacija takšne naprave v primeru večje omejenosti s stroški, nemogoča. Na koncu koncev bi bili stroški razvoja takšnega prototipa v podjetju, znatno večji.

iSystem		Trgovina	Število kosov	Cena (brez DDV)
	Razvojna ploščica STM32H750B-DK	/	2	/
IC Elektronika, iSystem				
	Izbrana oprema za spajkanje in komponenti	ICelektronika	1	36,14
Univerza v Ljubljani				

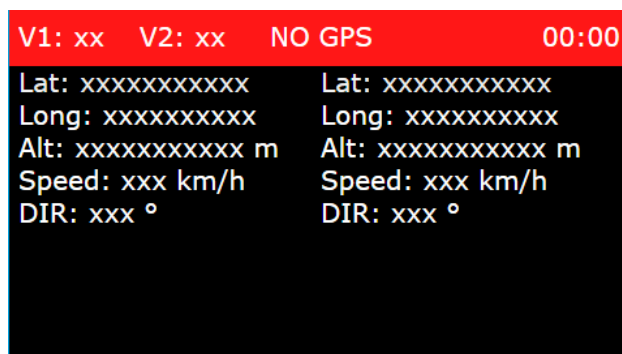
	Komponente za naselitev tiskanine vključno z rezervnimi deli	Farnell	1	81,20
	Tiskanina	Svet elektronike	2	26,77
	3D tisk	Fakulteta za elektrotehniko, Univerza v Ljubljani	2 /	
Nakup na lastne stroške				
	XBEE PRO	Farnell	2	38,88
	Lipo baterija	Hobbyking	5	4,34
SKUPAJ:				270,34

Tabela 2: poročilo o proračunu izvedenega projekta

Tekom implementacije naprave smo ustvarili nekaj pomembnejših zaključkov, ki nam bodo pomagali pri nadaljnjem razvoju tiskanine:

- Uporaba “rules” s programsko opremo Altium Designer 21: kljub temu, da smo si v *Rule Wizardu* nastavili vse ustrezne parametre, pred izdelavo *gerber* datotek nismo bili pozorni na opozorila za nepovezane “net-e”. Zaradi te napake smo bili, na tiskanini, primorani naknadno prevezati dva nepovezana stika.
- *Vij* nismo pokrili z silkscreeniom: misleč, da nam bodo odprti stiki na *vijah* omogočali lažje ugotavljanje napak na tiskanini, smo le-te nalašč pustili odprte, kar se je izkazalo kot povečana verjetnost za kratke stike.
- Kljub relativno kratkim analognim linijam se je izkazalo, da so te precej nestabilne. V prihodnje bi med analognim linijami in 0V dodali kondenzatorje.

Kljub nekaterim napakam nam je uspelo ustvariti robustno napravo, za dodatno zaščito pa smo celotno elektroniko vmestili v 3D sprintano ohišje, ki upotabniku omogoča pritrditev naprave na ustrezno mesto v vozilu.

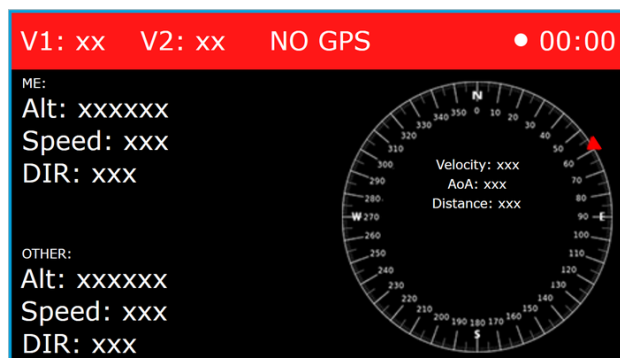


Slika 5: zaslon naprave

Preden smo se lotili programiranja naprave smo poiskali vse potrebne in kasneje uporabljene programe kot so CubeIDE (primarno programsko okolje s katerim smo programirali napravo),

CubeMX (ki smo ga uporabljali za netavitve procesorja, vhodov on izhodov), Touchgfx (uporabljen za nastavljanje grafičnega vmesnika) in Termite (serijski terminal, ki nam je omogoč prikaz podatkov).

V začetku programiranje smo za osnovo uporabili kar prazen demo program, ki nam je omogočal enostavno nastavitve LCD zaslona. V nadaljevanju programiranja smo nato gradili končno verzijo programa. Pri tem smo, za bolj učinkovito uporabo procesorske moči, uporabili FreeRTOS.



Slika 6: druga verzija zaslonske naprave

Testno poročilo

Unit testi

Po zaključenem naseljevanju tiskanine, testiranju osnovnih funkcij naprave in napisanem delujočem programu, smo se lotili testiranja posamezne naprave. Najprej nas je zanimala brezhibnost napajalnega sistema. Samo testiranje (torej polnjenje in praznjenje baterij) je sicer potekalo po zastavljenih ciljih, na kar je zaradi kratkega stika, ki smo ga v napaki dovedli med testiranjem sami, ena od baterij odpovedala. Po zamenjvi baterije smo precenili, da napajalni sistem zadošča našim zahtevam.



Slika 7: fotografija končne verzije obeh naprav v ohišju (Vir: osebni arhiv)

Našim zahtevam je prav tako zadoščal test natančnost sledenja lokacije GNSS modula v premikajočem vozilu. Test smo opravili tako v urbanem okolju Ljubljane, kot tudi na bolj odprtem področju na Barju.

Integracijski testi

Kot prvi test sistema smo ugotavljali domat naprave. Pri tem smo eno od naprav namestili na fiksno mesto v gosto poseljenem delu ljubljane. Na premikajočo napravo smo naložili program, ki je meril razdaljo od statične naprave, v primeru izgube signala, pa je zadnjo izmerjeno razdaljo zabeležil. Po večih ponovitvah smo sklenili, da je doomet naprave v urbanem okolju približno 125m, zunaj urbaega okolja pa vse do 300m. Rezultati dometa naprave so nas razočarali saj smo glede na specifikacije naprave pričakovali znatno večji domet.

V času pisanja poročila smo algoritem razvili do te mere, da lahko napravi zaznavata razdaljo razdaljo med njima in smer v kateri se glede na eno od naprav nahaja druga. S tem smo si za nadaljen razvoj pripravili tudi vse podatke, ki jih morata napravi med seboj posredovati za uspešno opozarjanje o nevarnosti.

Testiranje sledenja napravam smo izvedli tako v centru Ljubljane kjer smo testirali odziv sistema v urbanem okolju, kot tudi na Barju, za voljo testiranja sistema v "line of sight" okoliščinah. V prvem delu testiranja smo eno od naprav namestili v mirujoče vozilo, drugo pa v vozilo s katerim smo se približevali in oddaljevali prvi napravi. Algoritem smo s testiranjem pripravili do te mere, da brezhibno sledi drugi napravi. Za drug del testiranja smo obe napravi namestili v premikajoči vozili pri čemer smo bili z rezultati zadovoljni.

Projektno poročilo

Nazadnje moramo omeniti, da smo s sistemom kot celoto precej zadovoljni. Z izjemo algoritma za opozarjanje o nevarnosti, smo vse zastavljene cilje dosegli pravočasno. Poleg tega naša naprava dosega zanesljivost, ki smo jo pričakovali od samega začetka izziva. V nadaljevanju je potrebno, na podlagi že pripravčene osnove, dodati algoritem za nevarnost in ga ustrezno testirati. Predvsem pa nas zanima kako bi se algoritem odzval v primeru, ko bi bilo v bližji okolici prisotno večje število naprav. Za enkrat namreč menimo, da je sam program, za brezhibno delovanje, dovolj optimiziran, ne moremo pa jamčiti, da bi to držalo tudi v okolici naprave več deset ali celo več sto naprav.

Predvsem pa smo zadovoljni z znanjem, ki smo ga v sklopu izziva pridobili. To zajema tako mehke veščine kot tudi strokovno znanje. Za uspešnost projektnega vodenja in doseganje ciljev v določenih časovnih omejitvah, smo uporabljali github. Slednjo platformo pa smo obogatili tudi z vso izvorno kodo in strojnimi datotekami. Podroben opis projekta, kot je to poročilo, pa je dostopno na naši wiki razdelku githuba. Kljub uspešnem doseganju ciljev pa menimo, da bi bili v prihodnjih projektih, sposobni podobno zahteven projekt, z še bolj organiziranim vodenjem in obogatenim strokovnim znanjem, zaključiti znatno hitreje.

Povezava do Github spletnega mesta: <https://github.com/matek100/iSYSTEM-labs>