

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Peter Mlakar

GLOBALNI NAVIGACIJSKI SATELITSKI SISTEMI

DIPLOMSKO DELO
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: doc. dr. Patricio Bulić

Ljubljana, 2010



Št. naloge: 00469/2009

Datum: 01.09.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **PETER MLAKAR**

Naslov: **GLOBALNI NAVIGACIJSKI SATELITSKI SISTEMI**
GLOBAL NAVIGATION SATELLITE SYSTEMS

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Preučite delovanje modernih globalnih navigacijskih satelitskih sistemov (GNSS). Posebno pozornost posvetite strojni in programski opremi modernih GNSS. V okviru naloge pokažite možnosti realizacije lastne programske aplikacije v napravah GNSS ter preučite uporabo komponent GIS.NET in Geodesy.NET za tvorjenje lastnih kart. Posebej še preučite vgrajene sisteme za realizacijo modulov GPS ter predlagajte preprosto realizacijo z izbranim vgrajenim sistemom.

Mentor:

doc. dr. Patricio Bulić



Dekan:

prof. dr. Franc Solina

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si



Št. naloge: 00469/2009

Datum: 01.09.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **PETER MLAKAR**

Naslov: **GLOBALNI NAVIGACIJSKI SATELITSKI SISTEMI**
GLOBAL NAVIGATION SATELLITE SYSTEMS

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Preučite delovanje modernih globalnih navigacijskih satelitskih sistemov (GNSS). Posebno pozornost posvetite strojni in programski opremi modernih GNSS. V okviru naloge pokažite možnosti realizacije lastne programske aplikacije v napravah GNSS ter preučite uporabo komponent GIS.NET in Geodesy.NET za tvorjenje lastnih kart. Posebej še preučite vgrajene sisteme za realizacijo modulov GPS ter predlagajte preprosto realizacijo z izbranim vgrajenim sistemom.

Mentor:

doc. dr. Patricio Bulić



Dekan:

prof. dr. Franc Solina

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Peter Mlakar,

z vpisno številko 63040251,

sem avtor/-ica diplomskega dela z naslovom:

GLOBALNI NAVIGACIJSKI SATELITSKI SISTEMI

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek) doc. dr. Patricio Bulić
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 12.3.2010

Podpis avtorja:

Zahvala

Zahvalil bi se svojemu mentorju, doc. dr. Patriciu Buliću, za vsestransko podporo, strokovno pomoč in nasvete pri izdelavi diplomske naloge.

Še posebej bi se zahvalil tudi svoji družini, ki mi je na celotni študijski poti stala ob strani ter mi le-to tudi omogočila.

Posebna zahvala pa gre tudi Roku Horjaku in Andreju Peterki, ki sta mi z raznimi komentarji pomagala, da ima diplomsko delo vsebino kot jo ima ter mi tudi sicer skozi celotni študij pomagala po najboljših močeh.

KAZALO VSEBINE

KAZALO VSEBINE	I
SEZNAM UPORABLJENIH KRATIC.....	IV
POVZETEK	1
ABSTRACT	3
1. UVOD.....	4
2. GNSS	6
2.1 Zgodovina in teorija.....	6
2.2 Osnove delovanja	7
2.2.1 Kako določimo položaj z GNSS	7
2.2.2 Ugotavljanje položaja satelita	8
2.2.3 Almanah	8
2.2.4 Podatki o efemeridah	9
2.2.5 Merjenje časa potovanja signalov.....	10
2.2.6 Izračun lokacije sprejemnika	10
2.3 Izvor napak	13
2.3.1 Atmosferske motnje v troposferi in ionosferi	14
2.3.2 Napake pri sinhronizaciji ur v satelitih	15
2.3.3 Električni šum v signalu in v sprejemnikovi elektroniki.....	15
2.3.4 Napaka mnogih poti signala	15
2.3.5 Netočno poznavanje lokacije satelita.....	16
2.3.6 Selektivna dostopnost (S/A)	16
2.3.7 Geometrijska postavitev satelitov.....	16
2.3.8 Izboljšanje točnosti – diferencialni GPS	18
3. TRENUTNO AKTUALNI GNSS	23
3.1 Globalni navigacijski sistem – GPS	23
3.1.1 Kratka zgodovina GPS.....	23
3.1.2 Karakteristike GPS signala	23
3.1.3 GPS navigacijsko sporočilo	24
3.2 GLONASS	26
3.2.1 Kratka zgodovina GLONASS.....	26

3.2.2	Signali.....	26
3.2.3	Sateliti.....	27
3.3	Beidou 1/COMPASS.....	28
3.4	Galileo.....	29
3.4.1	Zgodovina in razvoj.....	29
4.	UPORABA GNSS IN KONCEPTI UPORABE.....	32
4.1	Vojaški nameni.....	32
4.1.1	Navigacija.....	32
4.1.2	Sledenje.....	32
4.1.3	Vodenje raket in pametnih bomb.....	32
4.1.4	Reševanje.....	34
4.2	Polijski in gasilski nameni.....	34
4.3	Civilni nameni.....	36
4.3.1	Pomorska navigacija.....	36
4.3.2	Cestna navigacija.....	37
4.3.3	Letalska navigacija.....	38
4.3.4	Uporaba v športih.....	40
5.	STROJNA IN PROGRAMSKA OPREMA.....	42
5.1	Programska oprema – operacijski sistemi.....	42
5.1.1	Windows CE.....	42
5.1.2	Symbian.....	43
5.1.3	Linux.....	44
5.2	Razvoj osnovne lastne aplikacije.....	45
5.2.1	Osnovni stavek.....	45
5.2.2	Jedro interpreterja.....	45
5.2.3	Čiščenje napak.....	47
5.2.4	Brezžična atomska ura.....	49
5.2.5	Smer in opozorila prekoračitve hitrosti.....	51
5.2.6	Ali je pozicija določena?.....	53
5.2.7	Spremljanje satelitov v realnem času.....	56
5.2.8	Interpreter, ki deluje po celem svetu.....	59
5.2.9	Ugotavljanje potrebne natančnosti.....	61
5.2.10	Uveljavljanje točnosti.....	62
5.3	Razvoj GIS in kartografske aplikacije za .NET.....	68

5.3.1	Geografski, projekcijski in točkovni koordinatni sistem.....	68
5.3.2	Ravna Zemlja je boljša za interpretacijo.....	68
5.3.3	Matematične enačbe projekcij.....	69
5.3.4	Aktivni del karte	71
5.3.5	Razmerje objektov.....	72
5.3.6	GIS	74
5.4	Praktični primer z uporabo Arduina	76
5.4.1	Arduino	76
5.4.2	EM-406 GPS modul.....	77
5.4.3	Določanje grobega položaja sprejemnika	78
5.4.4	Trenutna hitrost, razdalja do cilja in smer do cilja	80
6.	ZAKLJUČKI IN MOŽNE RAZŠIRITVE.....	83
	SEZNAM SLIK	85
	SEZNAM TABEL.....	86
	SEZNAM LITERATURE IN VIROV.....	87
6.1	Publikacije	87
6.2	Članki	87
6.3	Spletni viri (dostopno Februarja 2010)	87

SEZNAM UPORABLJENIH KRATIC

AIS	ang. Automated Information System – Avtomatski informacijski sistem
ASCII	ang. American Standard Code for Information Interchange – Ameriški standardni nabor znakov za izmenjavo informacij
C/A	ang. Coarse/Acquisition – Koda, ki se uporablja za pridobitev in dekodiranje L1 signala
CSEL	ang. Combat Survivor Evader Locator – Prenosni sistem ameriške vojske za hitrejše lociranje in posledično reševanje
DGPS	ang. Differential Global Positioning System – Diferencialni globalni navigacijski sistem
DOD	ang. Department of Defense – Obrambni oddelek Združenih držav Amerike
DOP	ang. Dilution of Precision – Merilo vpliva geometrijske postavitve satelitov na nebu na natančnost določanja položaja
ESA	ang. European Space Agency – Evropska vesoljska agencija
FAA	ang. Federal Aviation Administration – Ameriška zvezna uprava za letalstvo
FM	ang. Frequency modulation – Frekvenčna modulacija
GDOP	ang. Geometric Dilution Of Precision – Manjša točnost v lociranju v 3D prostoru in času
GLONASS	ang. GLObal Navigation Satellite System – Kratica imena ruskega globalnega navigacijskega satelitskega sistema
GNSS	ang. Global Navigation Satellite System – Globalni navigacijski satelitski sistem
GPS	ang. Global Positioning System – Ameriški sistem za globalno določanje položaja
HDOP	ang. Horizontal Dilution Of Precision – Manjša točnost v lociranju po 2D prostoru na tleh
HP	ang. High Precision – Visoka točnost
LAAS	ang. Local Area Augmentation System – Kratica imena ameriškega satelitskega sistema za zagotavljanje popravkov opazovanj na lokalnem področju
LORAN	ang. Long Range Navigation – Navigacija dolgega dosega
MCS	ang. Master Control Station – Glavna nadzorna postaja
MLRS	ang. Multiple Launch Rocket System – Mobilni sistem ameriške vojske za izstrelitev raket
NAVSTAR	ang. NAVigation System with Time And Ranging – Uradna kratica ameriškega programa oziroma sistema GPS satelitov
NMEA	ang. National Marine Electronics Associations – Kratica za standardizirana sporočila, ki jih oddajajo tipični GPS sprejemniki
PDOP	ang. Position Dilution of Precision – Vrednost, ki se nanaša na kakovost določitve položaja

POI	ang. Points Of Interest – Zanimive točke v cestni kartografiji
ROM	ang. Read Only Memory – Pomnilnik namenjen samo za branje
S/A	ang. Selective Availability – Selektivna dostopnost
SP	ang. Standard Precision – Standardna točnost
TDARDS	ang. Truth, Data Acquisition, Recording, and Display System - Kompaktna, lahka, poceni in izjemno prenosljiva naprava, ki uporablja GPS podatke v realnem času, radijsko komunikacijo in računalniško tehnologijo in s tem zagotavlja zelo točno realno-časovno, čas-prostor informacijo
TSPI	ang. Time-Space-Position Information – Realnočasovna, čas-prostor informacija
VDOP	ang. Vertical Dilution Of Precision – Manjša točnost v lociranju višine sprejemnika
VHF	ang. Very High Frequency - Radijska frekvenca med 30MHz in 300 Mhz
WAAS	ang. Wide Area Augmentation System – Kratica imena ameriškega satelitskega sistema za zagotavljanje popravkov opazovanj na širšem področju
WMS	ang. Wide-area Master Station – Glavna postaja pri sistemu zagotavljanja popravkov opazovanj na širšem področju

POVZETEK

Živimo v času, ki je prepreden z raznoraznimi tehnologijami. Te tehnologije imamo zato, da nam olajšajo vsakodnevna opravila ali pa povečajo produktivnost na poslovnem področju. V zadnjih 25 letih smo priča izjemno hitremu razvoju računalništva, integriranih vezij, programov in tehnologij, ki so s temi področji povezana. Skoraj si ne moremo več predstavljati gospodinjstva, ki ne bi imelo vsaj enega računalnika, digitalne televizije, mobilnega telefona ali navigacijske naprave. Še večja odvisnost od teh naprav pa se kaže v poslovnem svetu, kjer si ne moremo več predstavljati, kako poteka beleženje zalog in izdajanje računov s papirjem in pisalom. Tudi sledenje množici avtobusov bi bilo s telefonsko komunikacijo sila nepraktično, z uporabo navigacijskih naprav pa je ta postopek relativno preprost.

Globalni navigacijski satelitski sistemi so rdeča nit te diplomske naloge. Za to področje sem se odločil, ker se večina ljudi ne zaveda, da je satelitski navigacijski sistem še vse kaj drugega kot le tista mala škatla, ki nas usmerja, da z avtomobilom najdemo pravi naslov brez iskanja in nepotrebne zapravljanja časa. Prav tako večina ljudi ne ve, koliko truda in zapletene tehnologije je potrebne, preden nam tista mala škatla v avtomobilu postreže s pravilnim rezultatom iskanja in planom poti.

V sklopu diplomskega dela bom opisal globalne navigacijske satelitske sisteme, ki so že delujoči, bodo delujoči v kratkem ali pa so v zadnjih fazah razvoja. Poleg tega bom opisal teorijo pozicioniranja s pomočjo satelitov, napake, ki se pri tem lahko pojavijo, postopke za njihovo odpravo in postopke za izboljšanje natančnosti. Da pa bo diplomsko delo poleg nezahtevnim uporabnikom lahko prišlo prav tudi nadebudnim razvijalcem, ki bi radi v svoje projekte vključili satelitsko navigacijo, se bom v drugem delu posvetil programerskemu in strojnemu vidiku snovanja naprav ter izdelavi lastne kartografije.

ABSTRACT

Our time is a time of numerous technologies that allow us to spend less energy in doing everyday chores, enhance business productivity, etc. In the past 25 years we have witnessed an incredibly rapid development of computing, integrated circuits, programs and other similar technologies. One can barely imagine a home that would not have at least one computer, one digital television, one mobile phone and one navigation device. The world of business and economy shows an even greater need of the mentioned devices – we cannot imagine keeping records of stocks of goods or issuing invoices with pen and paper. Staying informed about the locations of numerous buses with, for example, the help of telephones would also be extremely impractical, were it not for navigation devices.

Global navigation satellite systems are the main theme of the following thesis. I have decided to explore this topic as most people do not realize that a satellite navigation system is actually much more than that small box in the car which effortlessly guides us to a certain address. Furthermore, not many of us are aware of the quantity of effort and the number of complex technologies that are needed for that small device to start working properly and giving us the right instructions.

In my thesis I will concentrate on global navigation satellite systems that are already in use or will be shortly and also on those soon to be developed. Additionally, I will describe satellite positioning theory, mistakes that can occur therein and processes for their removal as well as processes for the improvement of accuracy. Since I would like my thesis to aid not only everyday users but also eager researchers who want to include satellite navigation as part of their projects, I will concentrate on the programming and designing aspects needed to make the aforementioned devices as well as the preparation of one's own cartography.

1. UVOD

Med brskanjem za primerno temo diplomske naloge sem našel predlog naslova v zvezi s sistemom za globalno določanje položaja (ang. Global Positioning System, krat. GPS) in takoj pomislil na to, da je naslov morda napačen, saj bi bilo bolje spisati delo v katerem bi zajel vse globalne navigacijske satelitske sisteme (ang. Global Navigation Satellite System, krat. GNSS), ne pa samo tega, ki ga upravlja ameriška vojska. Prav tako sem hotel, da diplomsko delo ne bi romalo v polico, kjer bi se na njem nabiral prah, temveč bi bilo primerno za vse tiste, ki jih ta tema zanima in bi se radi poučili bodisi o osnovah delovanja in možnih napakah, bodisi o postopkih programiranja in načrtovanja neke aplikacije ali strojnega modula. Vse preveč ljudi se namreč ne zaveda, kako pomembna je satelitska navigacija v današnjem življenju in da se nahaja tudi tam, kjer na to ne bi pomislili.

Prvi del diplomske naloge je teoretične narave. Predstavljena je zgodovina GNSS in teorija uporabe takih sistemov. Bralec se v tem delu seznanja z osnovami delovanja satelitskih navigacijskih sistemov, ki obsegajo razlage o tem, kako določimo položaj sprejemnika z uporabo GNSS tehnologije. Pri tem se seznanja tudi s postopki ugotavljanja lokacije satelitov in o podatkih, ki jih ti sateliti pošiljajo na Zemljo. Poleg surovih podatkov je v tem delu opisana tudi vloga almanaha in efemerid pri določanju lokacije satelita. Pojasnjen je tudi postopek merjenja časa potovanja signalov in pa izračun točne lokacije sprejemnika. Pomemben del tega prvega dela diplomske naloge je tudi del o napakah oziroma izvoru napak. Predstavljene so atmosferske motnje, napake pri sinhronizaciji ur v satelitih, vpliv električnih šumov v samih napravah, napake odbojev in mnogih poti, napake, ki izvirajo iz netočnega poznavanja položaja satelitov, namenske napake, povzročene s strani oblasti in pa napake zaradi same geometrijske postavitve satelitov. V zadnjem delu tega sklopa je opisan še aktualni diferencialni GPS, ki s svojimi izboljšavami lokalnega ali širšega tipa močno izboljša točnost celotnega sistema.

V drugem delu je bralec seznanjen z delujočimi GNSS in pa tistimi, ki so šele v fazi izgradnje ali razvoja. Pomembno je, da bralec izve, da ne obstaja samo GPS ameriške vojske temveč tudi konkurenčna tehnologija Rusije in Kitajske, ter da se seznanja z evropskim projektom Galileo, ki bo slej ko prej zaživel in Evropi prinesel lastni satelitski navigacijski sistem. Ta drugi del je opisan veliko krajše, kot bi bil lahko, a je kljub temu razloženo, zakaj

hoče toliko držav zgraditi lastno satelitsko postavitvev in pa kako je izgradnja že delujočih sistemov potekala. Zanimivi podatki so tudi kar se tiče karakteristik signalov, podatkov, ki so posredovani na Zemljo in razvoj satelitov.

Tretji del je osredotočen na samo uporabo in pa koncept uporabe GNSS tehnologije in tehnologij, povezanih s tem področjem. Bralcu predstavim vlogo satelitske navigacije za vojaške, policijske in civilne namene. Tu je opisana raba, ki sega od sledenja, vodenja raket in pametnih bomb, do policijskega merjenja hitrosti in povečanja učinkovitosti. Prav tako je opisana raba na področju pomorske, cestne in letalske navigacije ter uporabe GNSS v športu.

V četrtem delu se posvetimo programski in strojni opre. Opisani so operacijski sistemi, na katerih tečejo navigacijski programi, razlike med njimi in praktičnost. V nadaljevanju bo opis razvoja osnovne lastne aplikacije, ki sega od poznavanja osnovnih stavkov do kreiranja samega jedra interpreterja. Postopek razlage je v tem delu malce daljši, a bralec tako lahko lažje absorbira znanje, saj gre za učenje korak za korakom. Tekom snovanja so prikazane osnovne funkcije kot so razbijanje NMEA (ang. National Marine Electronics Associations) stavka, čiščenje napak, ugotavljanje smeri in prekoračitve hitrosti ter ugotavljanje točnosti določitve lokacije. Prav tako pa v tem opisu prikažem, kako kreiramo interpreter, ki bo deloval po celem svetu, kako spremljamo satelite v realnem času in pa razlago o točnosti. Pomembno je namreč, da se pisec programa zaveda, koliko točnosti potrebuje za svojo nalogo, saj lahko sicer preveč časa posveti izboljšanju točnosti, ki niti ni tako pomembna, zanemari pa recimo opozorila o prekoračitvi hitrosti ali kaj podobno pomembnega. V tem delu bralcu predstavim tudi osnovne gradnike, ki uporabljajo tehnologijo GNSS. Z uporabo mikrokontrolerja Arduino, ki je ena najboljših in najcenejših ter najbolj odprtih platform za razvoj strojne opreme, pokažemo, kako izgleda shema vezave in pa kateri gradniki so potrebni za izdelavo osnovne GNSS naprave, na katero lahko naložimo svoj navigacijski program, ki je dopolnjen s funkcionalnostjo, ki smo si jo zadali.

Kot enega pglavitnih ciljev pri diplomski nalogi sem si zadal, da znanje, pridobljeno v času pisanja te naloge, strnim v obliko, ki bo primerna tudi za ostale, ki jih to področje zanima. Bralec z znanjem programiranja in željo po ustvarjanju, bi moral biti po končanem branju naloge seznanjen s podrobnim delovanjem GNSS in vseh tegob, ki se lahko pojavijo. Prav tako bi jih moral znati vsaj deloma odpraviti. Poleg tega pa bi moral biti pripravljen na

programerske naloge iz tega področja in pa imeti predstavo o tem, katere strojne gradnike bo potreboval in kako se povežejo skupaj v delujočo celoto.

2. GNSS

2.1 Zgodovina in teorija

Globalni navigacijski satelitski sistem je izraz, ki se uporablja za vse satelitske navigacijske sisteme, ki zagotavljajo pozicioniranje na globalnem nivoju. GNSS omogoča, da majhni elektronski sprejemniki izračunajo svojo geometrijsko širino in višino ter nadmorsko višino z natančnostjo nekaj metrov z uporabo radijskih signalov, ki jih na Zemljo pošiljajo sateliti v zemeljski orbiti.

V letu 2010 je ameriški NAVSTAR (ang. Navigation System with Time And Ranging) edini polno delujoči GNSS. Drugi v vrsti je ruski GLONASS (ang. Global Navigation Satellite System), ki bo v bližnji prihodnosti spet postal polno funkcionalen. Na tretje mesto lahko postavimo kitajski Compass, ki ga bo Kitajska razširila do globalne ravni. Tudi Evropa razvija svoj sistem Galileo, ki pa je zaradi velikih težav s financiranjem še vedno v začetni fazi.

Zgodovina GNSS sega v čase sistemov DECCA, LORAN (ang. Long Range Navigation) in OMEGA, ki so uporabljali zemeljske oddajnike radijskih frekvenc namesto satelitov [22]. Ti sistemi so oddajali radijske impulze iz znane glavne postaje, čemur so sledili kratki radijski pulzi iz sekundarnih postaj. Zamik med sprejetim in oddajnim signalom v sekundarnih postajah je bil natančno izmerjen, kar je omogočilo sprejemnikom, da so primerjali zamik pri sprejemanju in zamik pri oddajanju. Iz teh podatkov so lahko izračunali razdaljo do sekundarnih postaj in posledično svojo lokacijo.

Prvi satelitski navigacijski sistem se je imenoval Transit, ki ga je ameriška vojska zgradila v šestdesetih letih prejšnjega stoletja. Transit je deloval po principu Doppler efekta. Sateliti so potovali po znanih orbitah in oddajali podatke na znanih frekvencah. Glede na Dopplerjev efekt in z nekaj več meritvami je lahko sprejemnik izračunal svojo razdaljo do satelita. Z več sateliti in meritvami do njih pa je sprejemnik lahko izračunal svojo bolj ali manj točno lokacijo.

Moderni sistemi so bolj direktni. Sateliti oddajajo radijski signal, ki že vključuje orbitalne podatke (iz česar lahko izračunamo pozicijo satelita na nebu) in pa točno časovno oznako kdaj je bil signal poslan. Orbitalni podatki so oddani v podatkovnem sporočilu, ki je frekvenčno moduliran (ang. Frequency Modulation, krat. FM) na kodo, ki služi kot časovna znamka. Sprejemnik primerja čas oddajanja signala s časom, ko je bil signal prejet, in s tem izračuna čas potovanja signala. Naenkrat se lahko izvede kar nekaj takih meritev do različnih satelitov, kar ima za posledico navigacijo in pozicioniranje v realnem času.

2.2 Osnove delovanja

2.2.1 Kako določimo položaj z GNSS

Sprejemnik določi GNSS položaj s tehniko, imenovano geometrijska trilateracija (ang. Geometric Trilateration). Tehnika deluje tako, da pri izračunu neznane lokacije uporabimo razdalje do več znanih lokacij. V tem primeru so znane lokacije satelitov v orbiti, neznana lokacija pa je sprejemnik. Za izračun svoje pozicije sprejemnik tako potrebuje točne lokacije nekaj vidnih satelitov, nato pa izmeri razdaljo do vsakega od njih z merjenjem časa, ki ga signal porabi, da pride od satelita k sprejemniku.

Kako točna pa mora biti ta informacija? Če vemo, kje se satelit nahaja na kilometer natančno, potem lahko tudi svojo lokacijo izračunamo na kilometer natančno [17]. Točna informacija o tem, kje se satelit nahaja, je torej ključna za pravilen izračun. Za primer; GPS sistem deluje tako, da sprejemnik ve, kje se nahaja na 1 meter natančno. Najbrž ni potrebno posebej razlagati, kako zapletena tehnologija stoji za tem, saj sateliti krožijo v orbiti 20.000 kilometrov visoko in se gibljejo s hitrostjo 3.900 m/s, sprejemnik pa je vseeno lociran na 1 meter natančno.

Poznavanje točne pozicije satelita pa ni edino precizno merjenje. Ključno je tudi pravilno merjenje časa, ki ga signal potrebuje, da pride od satelita k sprejemniku. Če pride do 1 milisekunde zamika, bo napaka pri lociranju približno 300 kilometrov. Da bi uresničili natančnost 1 metra, mora biti torej meritev natančna na približno tri nanosekunde, kot kaže (1).

$$\begin{aligned}
 \text{Napaka lokacije} &= \text{Hitrost signala} * \text{Napaka meritve} & (1) \\
 300\text{km} &= 300.000.000\text{m/s} * 1 * 10^{(-3)}\text{s} \\
 1\text{m} &= 300.000.000\text{m/s} * 3 * 10^{(-9)}\text{s}
 \end{aligned}$$

Ključno pri izračunu sprejemnikove lokacije je torej dobro poznavanje pozicij satelitov in pa zelo točno merjenje časa. Preden pa preidemo na razlago, kako se računa pozicija sprejemnika, si moramo ogledati še, kako določimo točno lokacijo satelita in pa kako se meri čas potovanja signala.

2.2.2 Ugotavljanje položaja satelita

Pot vsakega satelita bi lahko v teoriji zelo natančno določili s pomočjo treh Keplerjevih zakonov o planetarnih gibanjih (ki je izdal v letih 1609 in 1619). Napovedana pot je v teoriji rezultat gibanja satelita, na katerega vpliva le Zemljina gravitacijska sila, ob tem pa je Zemlja povsem okrogla in ima povsod enako gostoto [18]. To pa je seveda le teorija, v praksi so stvari povsem drugačne. Kot prvo, Zemlja ni popolna krogla ampak je rahlo izbočena ob ekvatorju. Kot drugo seveda nima povsod enake gostote in kot tretje imajo ostala nebesna telesa, še posebej Luna in Sonce, svoj gravitacijski privlak, ki vpliva na gibanje satelitov.

Ostale težave nastanejo zaradi dejstva, da sateliti ne potujejo v čistem vakuumu, zaradi česar občutijo rahlo zaviralno silo. Še bolj pomembna kot to pa je sila zaradi sevanja fotonov iz sonca in odbitih fotonov iz Zemlje, čemur pravimo solarni radiacijski pritisk (ang. Solar radiation pressure). Pritisk je odvisen od velikosti in orientacije satelita, oddaljenosti od Sonca itd. Solarna radiacija upočasnjuje satelite, ki se gibljejo proti Soncu in jih pospešuje, ko se od njega oddaljujejo. Ta pojav pa predstavlja tudi največjo napako pri izračunih, saj ga je izjemno težko predvideti.

Ker ni mogoče natančno napovedati poti satelitov, moramo njihovo lokacijo in pot periodično izmeriti. V ta namen se uporabljajo štiri avtomatizirane bazne postaje na Zemlji. Z njihovo pomočjo napaka pri poziciji satelita znaša manj kot 1 meter.

V primeru GPS se vsake toliko vsi ti signali iz baznih postaj pošljejo v glavno bazno postajo v Koloradu, kjer vzdržujejo spisek satelitov in njihove položaje. Ker pa morajo biti ti podatki dostopni vsem uporabnikom GPS tehnologije, se iz glavne postaje ti podatki periodično pošljejo v satelite, od koder so potem dostopni vsem uporabnikom [6].

Podatki o lokaciji so razdeljeni na dva dela: almanah in podatki o efemeridah. V nadaljevanju sledi, kaj ti podatki vsebujejo.

2.2.3 Almanah

Almanah vsebuje približne orbite satelitov in njihovih poti čez daljša časovna obdobja. V nekaterih primerih so ti podatki lahko nespremenjeni mesece ali celo več. Vsak satelit vsebuje almanah za vso konstelacijo satelitov. Posledično mora sprejemnik pridobiti samo almanah enega satelita, da dobi vpogled v približne orbite vseh ostalih.

Almanah je oddan vsake 12,5 minut, prav toliko pa traja tudi njegov prenos. Ko je torej sprejemnik vključen in so potrebni podatki iz almanaha, bo trajalo vsaj 12,5 minut, da bodo ti podatki osveženi, približne orbite satelitov pa znane. Vsi sprejemniki shranjujejo almanah na interni spomin, tako da ga je potrebno osvežiti, šele ko je zastarel. Podatki iz sprejemnikovega almanaha so neuporabni tudi v primeru, če je bil sprejemnik prestavljen za nekaj sto kilometrov.

2.2.4 Podatki o efemeridah

Ti podatki vključujejo točno pozicijo vsakega satelita in parametre, ki so potrebni za njegovo lociranje v bližnji prihodnosti. Z razliko od almanaha vsak satelit odda samo svoje podatke in ne za vse satelite skupaj. Posledično mora sprejemnik pridobiti te podatke od vsakega vidnega satelita posebej. Vsak satelit jih odda v 30 sekundnem intervalu, prenos pa traja 12 sekund. Ti podatki so veljavni 4 do 6 ur, satelit pa pri oddaji efemerid odda tudi informacijo o tem, koliko časa bodo veljavne. Sprejemnik je zadolžen za zbiranje novih posodobljenih efemerid po potrebi in njihovo shranjevanje v interni spomin. Nekateri proizvajalci se odločijo, da bodo efemeride osveževali periodično na 30 ali 60 minut, namesto da bi spremljali ažurnost trenutno shranjenih podatkov za vsak satelit posebej. Kakorkoli pa že pridemo do njih, so efemeride satelitov ključne za točno določanje položaja sprejemnika. »Slika 1« prikazuje, kako se v praksi odraža napačen podatek shranjenih efemerid.



Slika 1: Napaka v shranjenih efemeridah se lahko odraža v napačnih izračunih lokacije [16]

2.2.5 Merjenje časa potovanja signalov

Ko ima GNSS sprejemnik sveže efemeride in almanah, pozna točno lokacijo vseh vidnih satelitov na nebu, vendar so ti podatki neuporabni, če sprejemnik ne pozna razdalje do teh satelitov. Razdalja je definirana z merjenjem časa, ki ga radijski signal porabi, da pride od satelita k sprejemniku. Sateliti imajo vgrajenih nekaj zelo točnih atomskih ur, sprejemniki pa navadno digitalno, ki jo žene manj točni kvarčni kristal. Kako točno se izmeri ta časovni zamik je najbolje ponazoriti z naslednjim primerom.

Recimo, da sta satelit in sprejemnik nastavljeni tako, da ob 13:00 začeta istočasno igrati neko skladbo. Satelit začne oddajati to skladbo preko radijskih valov ki potujejo skoraj s svetlobno hitrostjo t.j. 300.000.000 m/s. Sprejemnik prejme to skladbo in jo primerja s tisto, ki jo predvaja sam. Če bi bili obe uri popolnoma sinhronizirani, bi tista skladba s satelita rahlo zaostajala, ker je potrebovala nekaj časa, da je prišla od satelita k sprejemniku. Čas, za katerega mora sprejemnik »premakniti« predvajanje svoje skladbe nazaj, da bi se popolnoma skladala s tisto, oddano s satelita, je enak času potovanja signala. Ker poznamo hitrost signala, lahko enostavno izračunamo, kako daleč je satelit z (2).

$$\text{Razdalja do satelita} = \text{čas potovanja signala} * \text{hitrost signala} \quad (2)$$

Zgornje pa drži samo v primeru, da sprejemnik začne igrati skladbo točno ob dogovorjenem času (13:00). Če začne s predvajanjem samo eno milisekundo prepozno, bo izračunana razdalja do satelita napačna za 300 kilometrov, kot to kaže (1). Ker ure v sprejemnikih niso

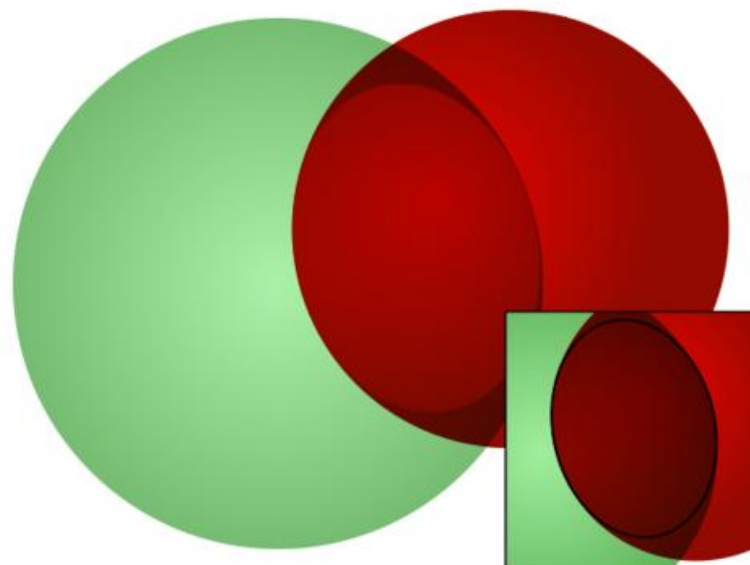
dovolj točne zaradi cene in prenosljivosti, samo na podlagi tega ne moremo izračunati pravilne razdalje do satelitov.

Ključ za rešitev tega problema se skriva v tem, da sprejemnik istočasno posluša oddane skladbe z več satelitov. Ti sateliti pa seveda ne predvajajo skladb ampak digitalne kode, 1000-krat vsako sekundo. Z večjim številom satelitov in unikatno kodo vsakega lahko GNSS sprejemnik popravi napake svoje ure in izračuna realne čase poti – razdalj do satelitov. Kako sprejemnik to opravi, je opisano v poglavju 4.2.6.

2.2.6 Izračun lokacije sprejemnika

Ko enkrat poznamo točen položaj vseh vidnih satelitov in razdalje do njih, lahko začnemo s preračunavanjem sprejemnikove lokacije. Možne napake na satelitovi ali sprejemnikovi strani so v tem delu zaradi lažjega razumevanja procesa zanemarjene.

Položaj satelita in razdalja do sprejemnika definirata sfero, ki ima v svojem središču satelit. Sprejemnik se nahaja nekje na površini te sfere. Ker je ta informacija neuporabna za potrebo navigacije, je potrebno dodati še en satelit. Dva satelita pomenita dve sferi, ki se lahko sekata v eni točki ali pa, kot je najbolj običajno, v krožnici, kot kaže »Slika 2«.

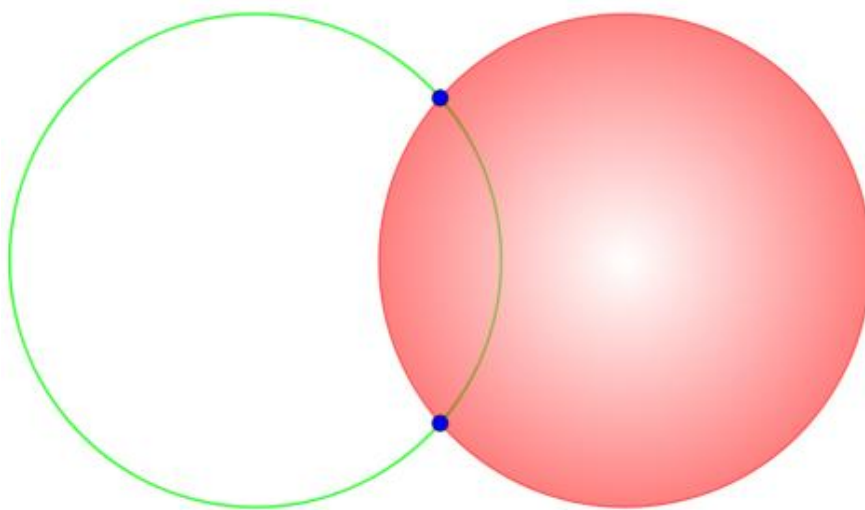


Slika 2: Dve sferi, ki se sekata v krožnici

Tudi to še ni dovolj dobro, saj krožnica predstavlja preveč možnih položajev sprejemnika, ki se glede na krožnico lahko nahaja nekje na Zemlji, v zraku ali pa v Vesolju. Da dobimo boljši rezultat, moramo dodati še en satelit, kot kaže »Slika 3«. Presečišče tretje sfere s prvima

dvema, je kar presečišče tretje sfere s krožnico iz »Slike 2«, v vseh praktičnih primerih pa za rezultat dobimo dve točki kot je razvidno iz »Slike 3«.

Za potrebe avtomobilske navigacije in navigacije, uporabljene v objektih blizu površja Zemlje, je pravilnen položaj sprejemnika točka, ki se nahaja najbližje površini Zemlje. Pravilni položaj sprejemnika je tudi presečišče, ki je najbližje površini sfere četrtega satelita. Za vesoljska plovila pa praviloma velja ravno obratno, razen v primerih vzletanja in pristajanja, pri čemer pa si lahko pomagamo z izboljšavami in nadgradnjami, ki bodo opisane v poglavju 4.3.8.

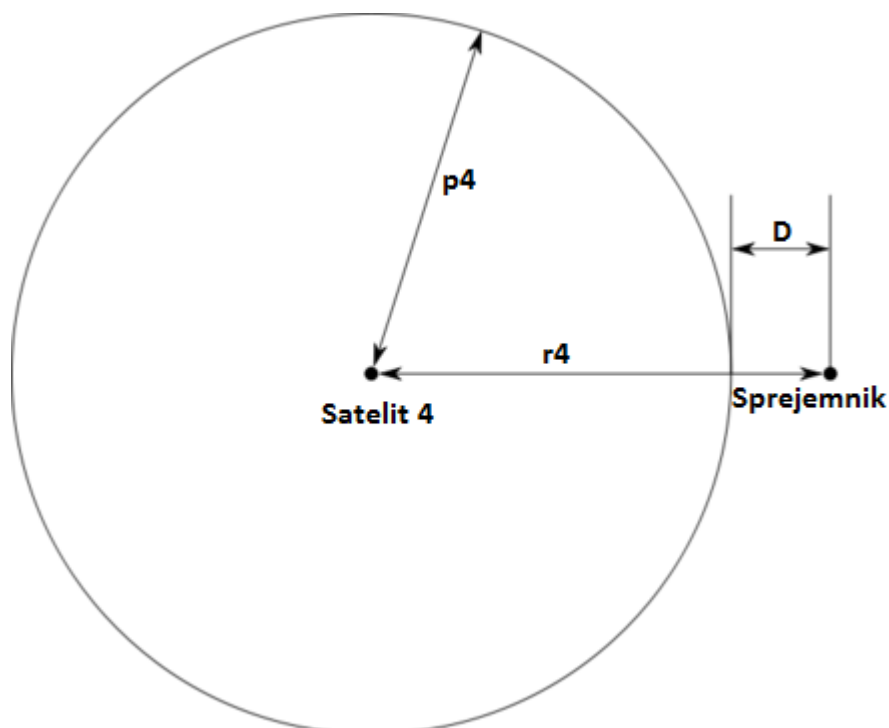


Slika 3: Površina sfere seka krožnico v dveh točkah

Kot rečeno pa zgornja razlaga drži samo v idealnih okoliščinah, kjer kakršnekoli napake niso prisotne. Ena izmed najbolj pomembnih napak je napaka sprejemnikove ure. Zaradi izjemne hitrosti signala, ki je blizu svetlobne hitrosti, so ocenjene razdalje sprejemnika do satelitov, imenovane psevdo meritve, močno podvržene napakam sprejemnikove ure. Za pravilno delovanje bi potemtakem potrebovali izjemno natančno in drago uro v vsakem GPS sprejemniku. Po drugi strani pa proizvajalci stremijo k temu, da bi bile naprave čim cenejše in dostopne kar najširšemu krogu ljudi. Rešitev te napake izvira iz tega, kako se površine sfer satelitov sekajo.

Zelo verjetno je, da se bodo tri sfere med seboj sekale v dveh točkah, saj je krožnica pri presečišču dveh sfer zelo velika, posledično pa obstaja velika verjetnost, da bo tretja sfera sekala to krožnico. Zelo malo verjetno pa je, da bo četrta sfera sekala katerokoli

izmed dveh presečišč prvih treh sfer saj bi že najmanjša napaka ure pomenila, da bi sfera ti točki zgrešila. Razdalja med ocenjenim položajem GPS sprejemnika do površine sfere četrtega satelita pa lahko služi kot popravek uri. »Slika 4« vsebuje r_4 , ki označuje ocenjeno razdaljo med sprejemnikom in satelitom, p_4 pa označuje psevdo meritev razdalje do četrtega satelita. D pomeni razliko razdalj $r_4 - p_4$.



Slika 4: Sfera satelita 4, r_4 , p_4 , D in položaj sprejemnika

Vidimo, da D pravzaprav pomeni razdaljo med izračunanim položajem GPS sprejemnika do površine sfere četrtega satelita. (3) Tako opisuje kvocient b , ki poda razliko med pravim časom in časom, ki ga uporablja sprejemnikovo ura. Če je b pozitiven, potem mora sprejemnik svojo uro premakniti naprej, če pa je negativen, jo mora premakniti nazaj.

$$b = D / \text{hitrost svetlobe } (c) \quad (3)$$

2.3 Izvor napak

Glavni izvori napak pri GNSS so: [1,6,16]

- Atmosferske motnje v troposferi in ionosferi
- Napake pri sinhronizaciji ur v satelitih
- Električni šum v signalu in v sprejemnikovi elektroniki

- Napaka mnogih poti signala
- Netočno poznavanje lokacije satelita
- Geometrijska postavitev satelitov

»Tabela 1« prikazuje možne napake in tipične realne napake za sistem GPS.

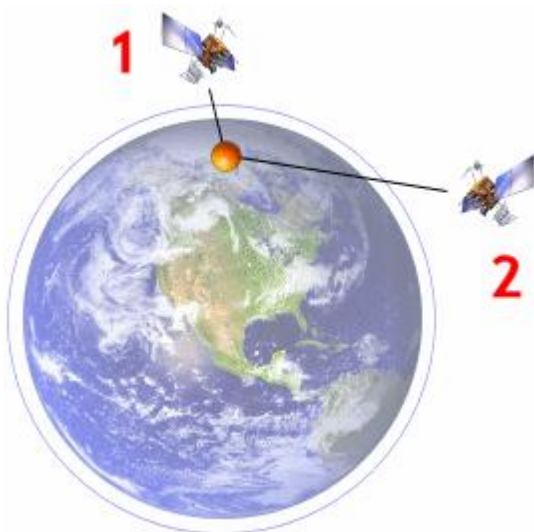
Izvor napake	Možna napaka	Tipična napaka
Ionosfera	5m	0,4m
Troposfera	0,5m	0,2m
Napake efemerid	2,5m	0m
Napačna sinhronizacija ur	1,5m	0m
Napaka večih poti	0,6m	0,6m
Električni šum	0,3m	0,3m
Skupaj	~15m	~10m

Tabela 1: Možne in tipične napake z vrednostmi

2.3.1 Atmosferske motnje v troposferi in ionosferi

GNSS literatura [1,6,16] razdeli zemeljsko atmosfero na dva dela: troposfero in ionosfero. Troposfera zaradi te razdelitve vsebuje tudi stratosfero in mezosfero in se razteza od površja Zemlje pa do 80 kilometrov nadmorske višine. To področje vsebuje nekje 99,99% vse zemeljske atmosfere. Ionosfera se razteza od 80 kilometrov nadmorske višine pa vse do 1000 kilometrov višine. Ionosfera se imenuje zato, ker v njej solarna in kozmična radiacija odstraniti vse elektrone z molekul in ionizirata pline.

Hitrost svetlobe se zmanjša v troposferi in ionosferi, tako da je izmerjena razdalja do satelitov daljša od resnične. V ionosferi se hitrost svetlobe zmanjša glede na to, koliko



Slika 5: Razlika v dolžini poti signala je lahko ključnega pomena.

elektronov se nahaja na njeni poti, gostota elektronov pa je odvisna od zemljepisne širine satelita, dela dneva in pa kota, pod katerim se satelit nahaja glede na površje Zemlje. Bistveno večji zamiki se pojavijo pri satelutih, ki se nahajajo v nižjih kotih, kot pri tistih, ki so bolj navpično nad sprejemnikom, saj prvi potujejo čez več ionosfere, kar je razvidno iz »Slike 5«. Napake pa so tudi dosti večje podnevi kot ponoči zaradi solarne radiacije. Večje napake so tudi blizu ekvatorja in pa blizu magnetnega severnega

in južnega pola.

Vsi GNSS sprejemniki vsebujejo matematične modele za ionosfero. Kot del navigacijskih podatkov GNSS sateliti oddajajo parametre (GPS datum in čas, status in stanje satelita, efemeride, almanah), ki jih sprejemnik vključi v ta matematični model. Ti parametri so osveženi enkrat ali dvakrat dnevno, ko so podatki zbrani in procesirani v glavni nadzorni bazi.

GNSS sprejemnik uporabi parametre, poslane s satelita, in ostale faktorje, ki temeljijo na uri v dnevu, in kot satelita glede na površje Zemlje, za izračun ionosferskega popravka za vsak satelit posebej. Ker pa so ti parametri samo neko povprečje parametrov z vseh satelitov, je tudi rezultat samo približek realnega stanja atmosfere v določenem trenutku. GPS specifikacije določajo, da je lahko pri teh parametrih le 50% točnost [2]. Ti približki pa privedejo do 5 metrov napake ponoči in do 30 metrov napake podnevi za satelite pod nizkim kotom in 3 do 5 metrov za satelite pod visokim kotom. V »Tabeli 1« je napisana možna napaka ionosfere 5 metrov, saj je to mednarodno sprejeta povprečna možna napaka. Napake troposfere so bistveno manjše od ionosferskih in le redko presegajo 1 meter.

Boljši komercialni sprejemniki se zanašajo na dejstvo, da se ionosferske motnje razlikujejo glede na frekvenco radijskih valov, ki gredo čez njo. Ionosferske motnje so tako lahko skoraj popolnoma izničene, če sprejemnik uporablja L1 in L2 frekvence, ki jih oddajajo sateliti. Ta korekcija se nanaša točno na radijske valove in ne na njegovo vsebino. To pomeni, da lahko sprejemnik uporabi podatke iz L2 frekvenčnega pasu, brez da bi moral dešifrirati njegovo vsebino. Ta pristop se imenuje korekcija z dvema frekvencama (ang. Two-frequency-correction). Prednost ima v tem, da neposredno meri efekt ionosfere med potjo signala in se ne zanaša na povprečja, ki jih oddajajo sateliti.

2.3.2 Napake pri sinhronizaciji ur v satelitih

Za idealne izračune bi morale biti ure v satelitih povsem sinhronizirane. V praksi so umerjene na 3 nanosekunde natančno [8]. To se na prvi pogled zdi majhna številka a se zaradi velike hitrosti svetlobe odraža v napaki lociranja, veliki 1 meter.

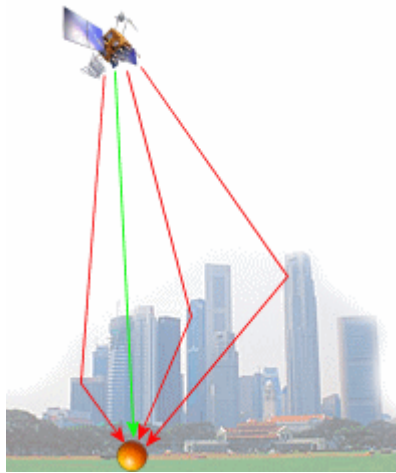
2.3.3 Električni šum v signalu in v sprejemnikovi elektroniki

Električni šum signala in sprejemnikovih komponent lahko doda še nekaj napake k izračunani lokaciji. Sprejemnikov šum je odvisen od kvalitete in načrtovanja sprejemnika, interference z

ostalimi radijskimi signali, temperature in ostalih spremenljivk. Napaka zaradi električnega šuma se odraža v napaki izračunane lokacije za do 2 metra [7].

2.3.4 Napaka mnogih poti signala

Napaka mnogih poti je povzročena, ker se GNSS signal odbije od površine ali nekega objekta, preden sprejemnik zazna prvotni signal. V ekstremnih primerih je ta napaka lahko velika 15-20 metrov. Študije so pokazale, da so velike vodne površine najbolj odgovorne za večje napake mnogih poti, medtem ko so za to napako najmanj dovzetna peščena tla [4,5]. Stavbe ali večje skale so prav tako lahko izvor napake mnogih poti, kot kaže »Slika 6«.



Slika 6: Zelena puščica predstavlja neoviran signal, rdeče pa popačen zaradi odbojev

2.3.5 Netočno poznavanje lokacije satelita

Kot sem opisal na začetku tega poglavja, poznamo lokacijo satelita na približno 1 meter natančno. Posledično je tudi napaka pri izračunu lokacije sprejemnika lahko le toliko natančna.

2.3.6 Selektivna dostopnost (S/A)

V prvotnem GPS je vojska uvedla naključne napake, da neavtorizirane osebe ne bi imele tako točnih podatkov kot oni. Podatki so vsebovali namenske napake o poziciji satelitov in/ali spreminjanje frekvence ure.

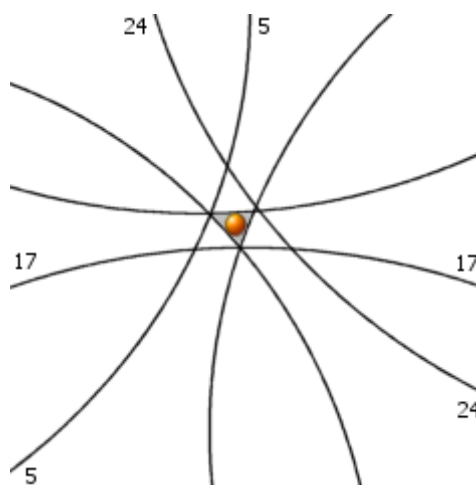
Ta namenska napaka se imenuje selektivna dostopnost (ang. Selective Ability, krat. S/A). Napaka je povzročila, da je bilo v civilni rabi mogoče locirati sprejemnik le na nekje 100 metrov natančno. Za primer: če bi sedeli v čolnu, privezanem na pomol, bi na GPS napravi vseeno videli spremembe v hitrosti, smeri in lokaciji, četudi bi mirovali. S/A način je bil ukinjen maja leta 2000 in od takrat še ni bil vključen [17].

2.3.7 Geometrijska postavitve satelitov

Efekt geometrijske postavitve satelitov je ovrednoten z redčenjem natančnosti (ang. Dilution Of Precision, krat. DOP) . »Slika 7« in »Slika 8« prikazujeta vpliv geometrijske postavitve satelitov na točnost izmerjene lokacije sprejemnika.



Slika 7: Velik DOP pomeni večje področje možne lokacije kar je slabše za točno lociranje



Slika 8: Če v »Slika7« dodamo še tri bolj enakomerno razporejene satelite dobimo manjši DOP oz. manjše področje možne lokacije

Najboljši pristop k zmanjševanju redčenja natančnosti je, da opazujemo kar se da veliko satelitov. Kot pa je opisano v poglavju o ionosferi, je čas potovanja signala iz tega območja daljši, prav tako pa je napaka zaradi ionosferskih motenj bistveno povečana. Iz tega sledi, da je najbolje prezreti satelite, ki se nahajajo pod 15 stopinjskim kotom glede na Zemljino površje. Ta prag se po navadi lahko nastavi na sprejemniku.

Različne napake redčenja natančnosti so lahko javljene na GNSS sprejemniku, vse pa imajo isto osnovo napake:

- manjša točnost v lociranju višine sprejemnika (ang. Vertical Dilution Of Precision, krat. VDOP)
- manjša točnost v lociranju po 2D prostoru na tleh (ang. Horizontal Dilution Of Precision, krat. HDOP)
- manjša točnost v lociranju v 3D prostoru (ang. Positional Dilution Of Precision, krat. PDOP)
- manjša točnost v lociranju v 3D prostoru in času (ang. Geometric Dilution Of Precision, krat. GDOP)

DOP vrednosti so predstavljene s številko od 1 do 50. »Tabela 2« prikazuje, kako numerično razdelimo vpliv DOP.[23]

DOP	Vrednost	Opis
1	Idealna	To je najboljše možno stanje, ki ga aplikacije uporabljajo kadar se v vsakem trenutku zahteva najvišja točnost
2-3	Odlična	S to točnostjo se lahko zadovoljijo vse razen najbolj občutljivih aplikacij
4-6	Dobra	To je najnižja vrednost, ki še ustreza poslovnim aplikacijam.
7-8	Zadovoljiva	Pri tej vrednosti lahko opravljamo meritve a točna določitev bi bila lahko še bistveno boljša
9-20	Slaba	Ta vrednost nam daje malo zaupanja. Meritev se naj uporabi le kadar rabimo približek, ki je lahko bolj ali manj netočen.
21-50	Izjemno slaba	Pri tej vrednosti izmerjena lokacija lahko odstopa tudi 100 metrov in več in je kot taka zavržena

Tabela 2: DOP vrednosti in njihov pomen

2.3.8 Izboljšanje točnosti – diferencialni GPS

Točnost izračunane lokacije lahko bistveno izboljšamo z uporabo tehnike, imenovane diferencialni GPS (ang. Differential GPS, krat. DGPS). DGPS tehnika temelji na uporabi vsaj dveh GPS sprejemnikov. En sprejemnik se nahaja na znani lokaciji, ki se ne spreminja in ki je bila v naprej zabeležena s tradicionalnimi merskimi metodami. Ta sprejemnik se imenuje bazna postaja in med drugim vsebuje tudi računalnik.

Bazna postaja neprestano sprejema GPS signale in izračunava lokacijo glede na dobljene podatke s satelitov. Računalnik potem primerja izračunano lokacijo s tisto, ki je bila že v naprej izmerjena in je najbolj točna izmerjena lokacija te bazne postaje. Razlika med izračunano lokacijo glede na GPS podatke in tisto lokacijo, ki je bila vnaprej izmerjena, je napaka GPS-a za dotični primer. Vsaka meritev je označena tudi s časovno znamko, tako da vemo, za kateri del dneva ta napaka velja.

Cenejše GPS naprave so na ta način sposobne izračunati lokacijo na manj kot 1 meter natančno, še posebej, če je bazna postaja oddaljena samo nekaj kilometrov in če ta bazna postaja ter sprejemnik spremljata iste satelite. Dražje DGPS naprave so sposobne izračunati lokacijo na manj kot centimeter natančno [12].

Obstajata dva DGPS sistema: DGPS v realnem času in pa DGPS s poobdelavo podatkov. DGPS s poobdelavo podatkov zbira podatke v bazni postaji za vse instance signalov posebej, skupaj z napakami meritev. Na koncu dneva se vse te meritve primerjajo z meritvami sprejemnikov na terenu in po potrebi popravijo. Očitno je, da je tak sistem primeren le za sprejemnike, ki ne rabijo točnih podatkov, medtem ko se premikajo. Tak sistem se recimo uporablja v nadzemnih rudnikih zlata, diamantov, premoga itd. Avtomobila z GPS sprejemnikom dejansko ne zanima točna pozicija, medtem ko se vozi po novi trasi rudnika. V nadzornem centru pa konec dneva ali tedna morajo vedeti, kako napredujejo dela, kakšen je ugrez, kakšni so premiki Zemlje in tako naprej.

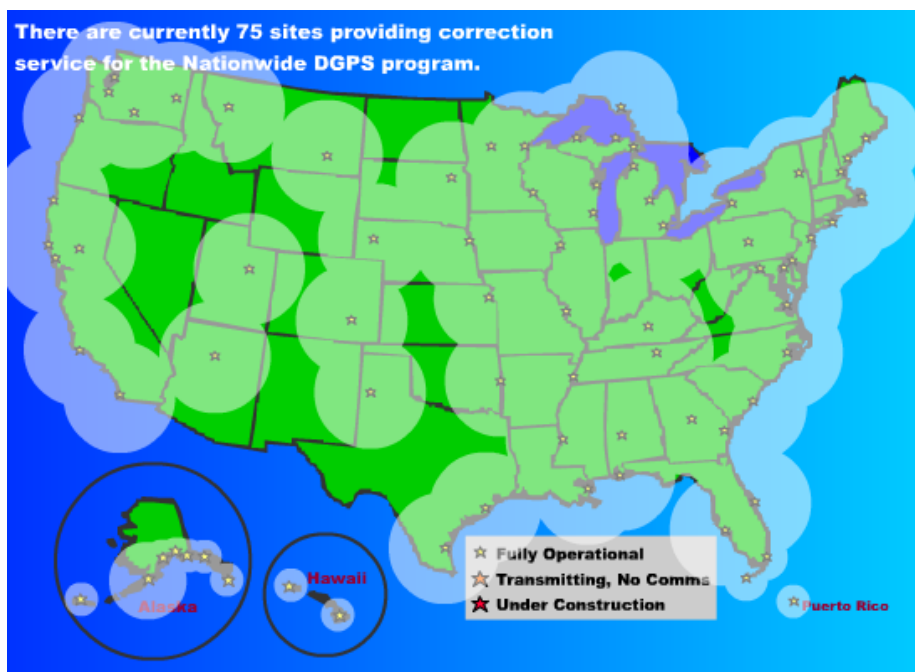
Zgornji pristop pa je neprimeren za naprave, ki morajo vedeti svojo lokacijo v realnem času kot je to recimo letalo pri pristajanju ali avtomobil, ki išče svoj cilj. Realnočasovni DGPS sestoji tudi iz radijskega oddajnika v bazni postaji. Takoj po sprejetem signalu iz satelita bazna postaja izračuna napako in ta podatek preko svojega radijskega oddajnika sporoči vsem ostalim GPS sprejemnikom v dometu. GPS sprejemniki, ki hočejo izkoristiti DGPS, morajo biti opremljeni z ustreznim sprejemnikom in imeti ustrezno računsko moč, da lahko v realnem času procesirajo vse te podatke. Prikazana lokacija na zaslonu naprave ima tako vse napake meritev že izključene in nam postreže z najboljšo možno točnostjo lokacije.

Za pridobitev kar se da velike točnosti na lokalnem nivoju je DGPS najboljši pristop. Ameriška vlada se je med prvimi zavedala tega in vzpostavila sistem DGPS baznih postaj in radijskih oddajnikov. Še posebej veliko jih je ob obalah in pa ob letališčih. Inštalacija DGPS baznih postaj in zagotavljanje njihovih signalov zastonj ali proti plačilu se imenuje GPS

nadgradnja (ang. GPS Augmentation). Poznamo tri glavne DGPS nadgrajene sisteme: svetilnik, WAAS in LAAS.

Svetilnik

Ameriška obalna straža operira z mrežo 75 baznih postaj, ki so postavljene okoli pristanišč in morskih poti z večjo obiskanostjo. Podatki o korekciji so posredovani preko oddajnikov na kopnem. Uradni podatek o točnosti sistema Svetilnik je 10 metrov. Večina proizvajalcev pomorskih GPS navigacijskih naprav ima vgrajeno možnost sprejemanja teh signalov, kar je označeno v specifikacijah sprejemnika. Poleg 75 DGPS svetilnikov, ki jih upravlja ameriška obalna straža, pa je po svetu postavljenih še 124 drugih [12]. »Slika 9« prikazuje položaje v ZDA kjer so prisotne svetilniške postaje.

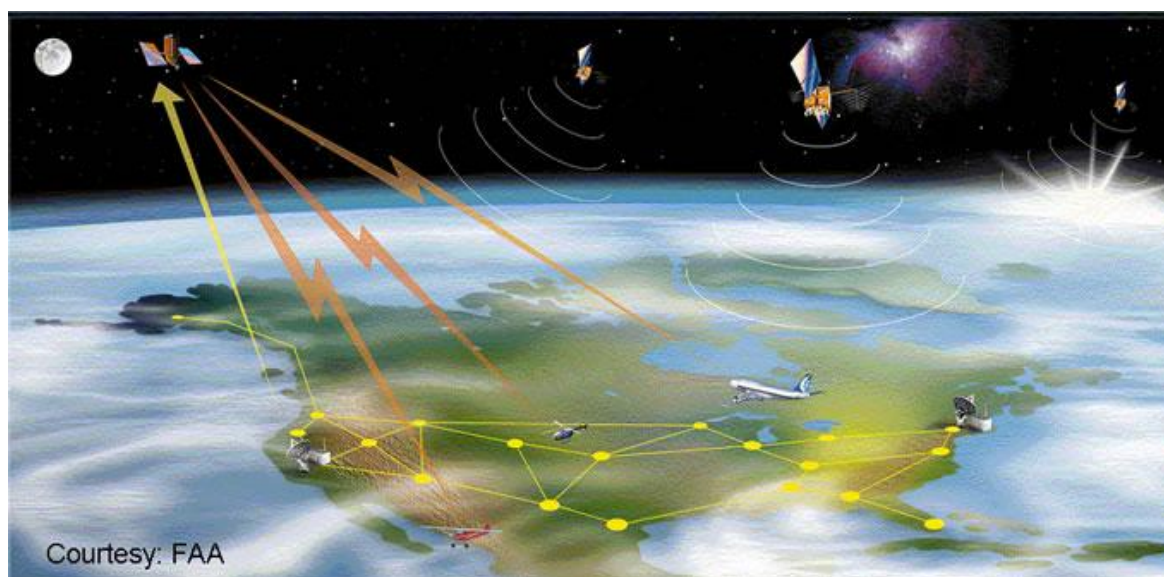


Slika 9: Svetilniške postaje v ZDA [12]

WAAS – nadgrajeni sistem za širša področja

Ameriška Zvezna uprava za letalstvo (ang. Federal Aviation Administration, krat. FAA) ima pod nadzorom 25 WAAS (ang. Wide-Area Augmentation Sstem) DGPS baznih postaj, ki pokrivajo celotno severno Ameriko s Kanado in pa delom Mehike. Za vsako od baznih postaj so skrbno izbrali lokacijo, prav tako pa vsaka vsebuje GPS sprejemnik. Vse bazne postaje posredujejo napake in korekcije v glavno bazno postajo (ang. Wide-area Master Station, krat. WMS). WMS ima algoritem, ki izračuna napake in korekcije ter poda, kakšna je integriteta sistema. Ta informacija je potem v rednih intervalih posredovana v geostacionarni

komunikacijski satelit. Ta satelit nato pošlje sporočilo o napaki in korekciji vsem sprejemnikom na letalih v zraku in tistim na tleh. To sporočilo o napaki in korekciji je brezplačno dostopno vsem sprejemnikom z opcijo WAAS. Veliko GPS sprejemnikov ima vgrajeno podporo WAAS, s čimer lahko uporabijo te dodatne podatke za večjo točnost. WAAS izboljša nazivno točnost GPS sistema na 7 metrov [12]. »Slika 10« prikazuje delovanje WAAS DGPS.



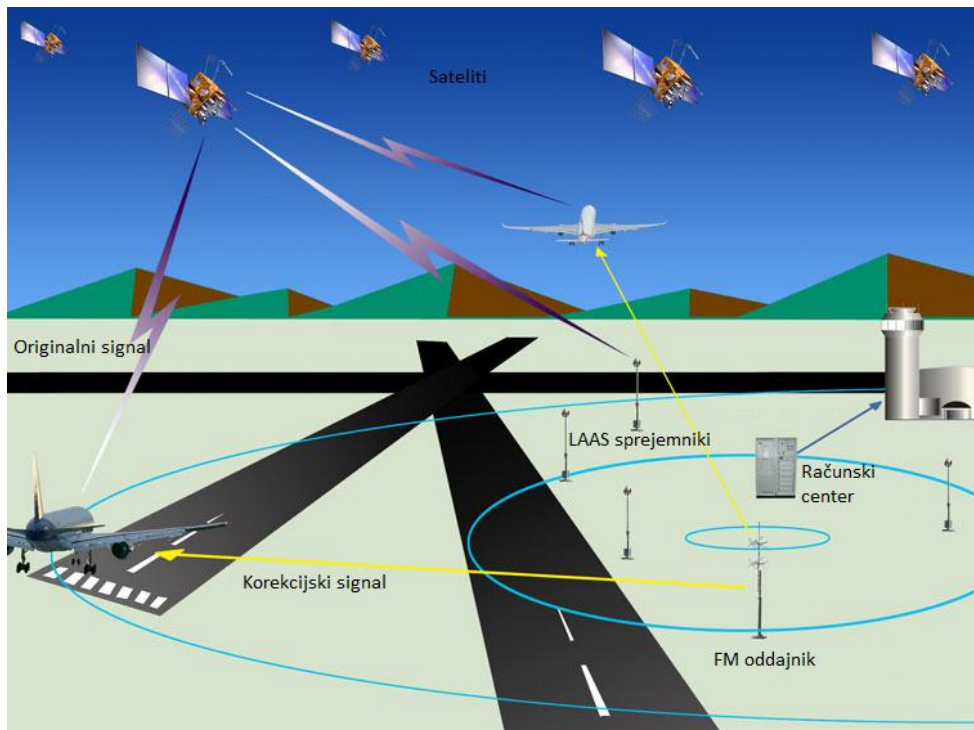
Slika 10: Komponente sistema WAAS: geostacionarni komunikacijski satelit, GPS sateliti, GPS sprejemniki, FM oddajniki in sprejemniki [12]

LASS – lokalno nadgrajeni sistem.

FAA uporablja tudi izboljšave GPS, ki so fokusirane na letališča in pa območja okoli letališč, s premerom od 30 do 50 kilometrov glede na letališče. Sistem uporablja več zelo točno postavljenih baznih postaj na ali ob letališču, s katerimi določi napake in morebitne korekcije signalov z GPS satelitov. Sporočilo o korekciji je oddano po VHF radijski zvezi in ga lahko sprejme vsak sprejemnik z možnostjo LAAS (ang. Local-Area Augmentation System). Točnost lociranja s tem pristopom je en meter ali manj [12]. Cilj LAAS je visoka točnost, velika zanesljivost in dostopnost za vsa letala v pristajanju. Napoveduje se, da bo sčasoma ta sistem omogočil pristajanje letal v ničli vidljivosti. »Slika 11« prikazuje postavitev in delovanje LAAS GPS.

Tudi privatni sektor se je spustil v vode nadgrajenih DGPS sistemov. Uporabniki se registrirajo za storitev nadgrajenega DGPS in sprejemajo popravke k svojim izračunom preko FM radia ali preko satelitskih povezav. Velikost območja, ki je pokrit s tem DGPS, in pa

točnost se razlikujeta. Nekaj komercialnih podjetij ponuja storitev, ki zagotavlja, da bo točnost lociranja najmanj 1 meter. Storitve imajo podprto z 90 baznimi postajami, korekcije signalov pa pošiljajo preko komercialnih satelitov v geostacionarnih orbitah.



Slika 11: Konfiguracija LAAS na letališčih

3. TRENUTNO AKTUALNI GNSS

3.1 Globalni navigacijski sistem – GPS

3.1.1 Kratka zgodovina GPS

Vojska, vlada in civilni uporabniki po celem svetu se zanašajo na GPS za potrebe navigacije in pozicioniranja, radijski valovi pa so bili uporabljeni za navigacijo že od davnega leta 1920 naprej. LORAN (Long Range Aid to Navigation) je sistem, ki je bil razvit med drugo svetovno vojno, z merjenjem časovnih razlik med prihajajočimi radijskimi signali pa je bolj ocenjeval kot meril razdalje in posledično lokacijo sprejemnika.

Prvi znaki GPS segajo v leto 1957 ko so Rusi izstrelili satelit Sputnik, ki je hkrati tudi prvi satelit, ki se je utiril v Zemljino orbito. Ameriški znanstveniki so pri oddajanju signalov s Sputnika odkrili Dopplerjev efekt, kar se je pozneje izkazalo za zlato jamo. Dopplerjev efekt lahko najlaže predstavljamo tako, da se spomnimo policijskega ali reševalnega vozila, ki se je peljalo mimo nas. Ton sirene se viša, ko se nam vozilo približuje in se hitro niža, ko se vozilo oddaljuje. Po istem principu lahko izmerimo Dopplerjev efekt tudi pri radijskih signalih.

Znanstveniki so ugotovili, da bi, če bi poznali točno pozicijo satelita, lahko izračunali svojo točno lokacijo na Zemlji s poslušanjem signalov s satelitov in merjenjem Dopplerjevega zamika teh signalov. Sateliti kot pomoč pri navigaciji so predstavljali dobro rešitev in obrambni oddelek ZDA (ang. Department of Defense, krat. DOD) je takoj pograbil idejo.

Do leta 1960 je obstajalo nekaj različnih neodvisnih navigacijskih sistemov. Ameriška vojska, mornarica in letalstvo so vsak zase razvijali satelitski navigacijski sistem, ki bi jim omogočil pozicioniranje v vseh vremenskih pogojih in to 24 ur na dan. Leta 1973 so bile ameriške letalske sile izbrane kot vodilna organizacija za razvoj in postavitve satelitskega navigacijskega sistema. Ta odločitev je kasneje pomenila razvoj NAVSTAR GPS (ang. Navigation Satellite Timing and Ranging Global Positioning System), kar je tudi uradni naziv za ameriški GPS [17].

Ameriška vojska pa ni bila navdušena nad GPS le zaradi pozicioniranja in navigacije temveč tudi zaradi učinkovitejšega namerjanja orožja. Pametne rakete, kot so Tomahawk, uporabljajo GPS za svoje usmerjanje. Zaradi možnosti zlorabe sistema in napada na Ameriko,

so letalske sile pozneje uvedle manj točen sistem za civilno rabo, ki pa je bil nekaj let zatem zopet izključen.

Prvi NAVSTAR satelit je bil izstreljen leta 1974 za potrebe testiranja koncepta. Do sredine leta 1980 je bilo dodanih dovolj satelitov, da je sistem postal funkcionalen. Leta 1994 je bilo dodanih vseh 24 načrtovanih satelitov, kmalu zatem pa je vojska razglasila sistem za polno funkcionalnega [2]. Projekt je bil in je še vedno zelo uspešen na vseh področjih, v celoti pa ga še vedno financira ameriški DOD.

3.1.2 Karakteristike GPS signala

Vsak GPS satelit oddaja šibak radijski signal z močjo 20 do 50 W. Zanimiva je primerjava s klasičnimi radijskimi valovi, ki jih oddaja lokalna radijska postaja. Moč GPS 50 W signala na površju Zemlje, ko prepotuje 18.000 kilometrov, je zanemarljiva v primerjavi z močjo signala te lokalne radijske postaje, ki oddaja signal z močjo 100.000 W in to s 50 ali več metrske antene nekaj kilometrov proč [1].

Vsak satelit oddaja informacije na dveh frekvencah, imenovanih L1 (1575,42 MHz) in L2 (1227,6 MHz). Ta princip je identičen tistemu, ki ga uporabljajo radijske postaje. Ko radijski sprejemnik nastavimo na recimo 101,1 FM, sprejemamo čisti radijski signal. Glasba, ki jo poslušamo, je modulirana na ta nosilni signal s pomočjo majhnih odmikov osnovne frekvence, sinhronizirane z uro. Ta tehnika je znana kot frekvenčna modulacija. Druga tehnika je rahlo spreminjanje amplitude signala, sinhroniziranega z uro. Ta tehnika se imenuje amplitudna modulacija (ang. Amplitude Modulation, krat. AM).

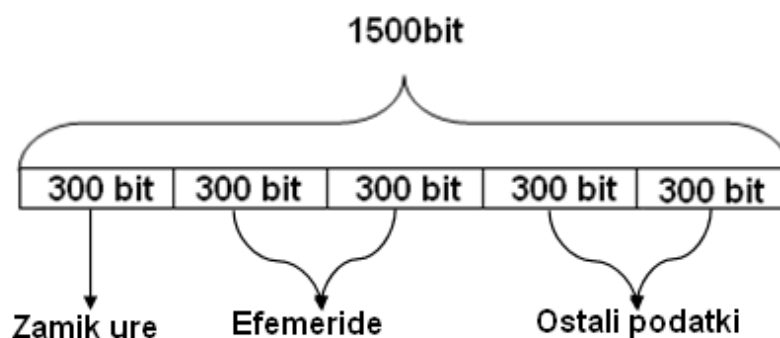
Podatki z GPS satelitov so povsem digitalni in v veliko primerih je to informacijo bistveno lažje modulirati na nosilni signal kot recimo klasično glasbo, kar se odraža v bolj kakovostnem prenosu z manj napakami. Ta tip podatkov pa je tudi manj dovzeten za namenske motnje.

GPS signal lahko tako kot svetloba in ostali elektromagnetni valovi potuje le po vidni liniji. Izogne se lahko le oviram, katerih velikost je odvisna od valovne dolžine signala.

3.1.3 GPS navigacijsko sporočilo

Navigacijsko sporočilo se neprestano oddaja s satelita na moduliranem nosilnem signalu s hitrostjo 50 b/s. To je telemetrijsko sporočilo, podatki pa so razdeljeni v okvirje. Okvir sporočila je velik 1500 bitov, kar pomeni, da se prenese v 30 sekundah. Vsak satelit ga začne oddajati natanko ob vsaki polni minuti in polovični minuti glede na interno uro. Vsak okvir je

razdeljen na 5 pod-okvirjev, ki so dolgi 300 bitov. Pod-okvirji 1,2 in 3 vsebujejo podatke visoke točnosti, kot so efemeride in zamik ure. Podatkovni del teh treh pod-okvirjev je pri satelitu enak za vse zaporedne okvirje za obdobje, dolgo do dveh ur. Novi pod-okvirji se nato začnejo pošiljati točno ob polni uri glede na interno uro. Pod-okvir 1 vsebuje koeficiente polinomov druge stopnje, ki se uporabljajo za izračun zamika ure. Pod-okvirja 2 in 3 vsebujeta orbitalne podatke, pod-okvirja 4 in 5 pa vsebujeta druge podatke, ki se razlikujejo v vsakem okvirju. Ti podatki se sicer ponavljajo, a sprejemnik mora prejeti 25 zaporednih okvirjev s pod-okvirji 4 in 5, da dobi vse podatke, ki jih satelit oddaja. Satelit oddaja iste pod-okvirje 4 in 5, dokler niso v satelit naloženi novi podatki, kar po navadi traja 24 ur. Pod-okvirja 4 in 5 sicer vsebujeta almanah in podatke povezane s stanjem in konfiguracijo [4]. »Slika 12« prikazuje okvir in razdelitev na pod-okvirje.



Slika 12: Okvir navigacijskega sporočila in podatki, ki jih vsebuje

Vsak pod-okvir je razdeljen na 10 besed, dolgih 30 bitov, in v vsaki besedi 6 bitov služi za pariteto.

Nalaganje podatkov na vsak satelit se običajno opravi enkrat na dan. V tem nalaganju glavna nadzorna postaja (ang. Master Control Station, krat. MCS) satelitu pošlje podatke, ki jih bo satelit oddajal naslednjih 24 ur, prav tako pa tudi podatke, ki jih bo satelit oddajal naslednji teden, za primer, če bi šlo pri nalaganju podatkov kaj narobe.

Vsi sateliti oddajajo na dveh frekvencah: 1575,42 MHz (L1) in 1227,6 MHz (L2). Sprejemnik lahko loči med signali z različnih satelitov, ker uporablja kodno tehniko preklapljanja (ang. Code Division Multiple Access, krat. CDMA), ki za razliko od časovnega preklapljanja (ang. Time Division Multiple Access, krat. TDMA) vsakemu satelitu doda posebno kodo za identifikacijo PRN. Te kode so uporabljene pri kodiranju navigacijskih sporočil, ker pa jih sprejemnik pozna vnaprej, lahko na podlagi tega dekodira sporočilo. Pri

kodiranju signala sta uporabljeni dve različni tehniki CDMA kodiranja: C/A (ang. Coarse/Acquisition) in P (ang. Precision). L1 je kodiran z obema, L2 pa samo s P tehniko. Medtem ko je C/A koda znana vsem in jo lahko vsi uporabljajo v civilne namene, pa je P koda dostopna le vojaškim napravam z ustreznim šifrirnim ključem.

Poleg L1 in L2 pa poznamo še tri frekvence, na katerih lahko oddajajo sateliti:

- 1381,05 MHz (L3) – Uporablja se za javljanje jedrskih in drugih močnih eksplozij.
- 1379,913MHz (L4) – Trenutno se testira za morebitne nadaljnje popravke ionosferskih motenj.
- 1176,45MHz (L5) – Ta frekvenca je predvidena za civilno rabo, še posebej za javljanje položaja v kritični situaciji. Frekvenca je za to nalogo primerna, saj spada v mednarodno zavarovano področje, kar ji zagotavlja malo oziroma nič motenj. Prvi satelit, ki bo zmožen oddajati ta signal, je ruski Block IIF, ki naj bi bil v tirnico okoli Zemlje postavljen v letu 2010.

3.2 GLONASS

3.2.1 Kratka zgodovina GLONASS

GLONASS je satelitski navigacijski sistem, ki prav tako kot ameriški GPS bazira na radijskih signalih, zgrajen pa je bil v nekdanji Sovjetski zvezi. Od razpada Sovjetske zveze z njim upravlja ruska vlada oziroma ruske vesoljske sile. Je alternativa in dodatek ameriškemu GPS satelitskemu sistemu, kitajskemu Compass in evropskemu Galileo.

Gradnja satelitskega sistema GLONASS se je pričela leta 1976 s ciljem pokriti vso Zemljino površje do leta 1991. 12. oktobra 1982 so pričele leteti prve rakete s sateliti in jih dodajati v konstelacijo, ki je bila dokončana leta 1995. Na žalost pa je sistem začel propadati takoj po začetku delovanja, saj ni bilo denarja za vzdrževanje – Rusija se je spopadala z veliko ekonomsko krizo. Leta 2001 se je Rusija zavezala, da bo sistem ponovno vzpostavila. Z nekaj popravki naj bi bil sistem spet polno funkcionalen konec leta 2010 oz. takoj v letu 2011, kar pa je morda malo preveč optimistična napoved glede na trenutne ekonomske razmere.

GLONASS je bil zgrajen z namenom priskrbeti realno-časovne podatke o poziciji in hitrosti, ki bi ga uporabljala ruska vojska za potrebe usmerjanja in namerjanja svojih balističnih raket. To je bila druga generacija satelitske navigacije Sovjetske zveze. Prva

generacija je bila Tsiklon, v podrobnosti katere se ne bomo spuščali. Kot zanimivost naj služi dejstvo, da je ta sistem za izračun pozicije potreboval od ene do dveh ur procesiranja signalov. Za primerjavo: današnji GLONASS ali GPS sprejemniki lahko izračunajo pozicijo praktično takoj ob sprejetju ustreznih signalov. V optimalnih razmerah naj bi sistem GLONASS imel ločljivost 50-70 metrov in izmeril hitrost na 15 centimetrov na sekundo natančno. Vse to opravi z 99,7% verjetnostjo [15].

3.2.2 Signali

GLONASS satelitski sistem oddaja dve vrsti signalov: standardna točnost (ang. Standard Precision krat. SP) in visoka točnost (ang. High Precision, krat. HP). Vsi sateliti oddajajo enak signal, ki je SP, a vsak satelit oddaja na svoji frekvenci. Pri tem sistem uporablja 15-kanalno frekvenčno tehniko preklapljanja (ang. Frequency Division Multiple Access, krat. FDMA) tehniko. Osnovna frekvenca znaša $1602,0 \text{ MHz} + n * 0,5625 \text{ MHz}$, pri čemer je n številka satelitovega frekvenčnega kanala [15]. Na prvi pogled se zdi 15 kanalov premalo za 24 satelitov, a ker ti potujejo okoli Zemlje in imajo med seboj enake razmike, se ne more zgoditi, da bi naenkrat potrebovali vseh 24 satelitov. Posledično je 15 kanalov povsem dovolj.

3.2.3 Sateliti

V treh desetletjih so sateliti v konstelaciji GLONASS doživeli kar nekaj sprememb in nadgradenj, vsi pa so se in se še vedno imenujejo Uragan, čemur sledi številka ali črka.

Prvi GLONASS poizkusni sateliti so bili izstreljeni med leti 1982 in 1985. 18 takih satelitov je bilo izstreljenih z načrtovano življenjsko dobo enega leta, v povprečju pa so delovali 14 mesecev.

Sateliti prve generacije so bili že zelo dobri sateliti, saj so imeli skromne manevrske sposobnosti za premeščanje znotraj konstelacije. Težki so bili 1250 kilogramov. Med leti 1985 in 1986 je bilo izstreljenih 6 satelitov s povprečno življenjsko dobo 16 mesecev. Leta 1987 je bilo izstreljenih 12 nadgrajenih satelitov s pričakovano življenjsko dobo 24 mesecev. 6 jih je bilo izgubljenih pri nesrečah ob izstrelitvi, ostalih 6 pa je delovalo v povprečju 22 mesecev. Med leti 1988 in 2000 (nekaj celo leta 2005) je bilo izstreljenih 25 novih satelitov s pričakovano življenjsko dobo treh let, a so to prekoračili skoraj vsi. Eden od njih je deloval kar 68 mesecev. Z raketami Proton so naenkrat izstrelili po tri satelite.

Drugo generacijo satelitov, Uragan-M, so izdelovali med leti 1990 in 2007. Ti sateliti so pridobili na življenjski dobi, saj je bila le-ta 7 let. Teža satelitov se je povečala na 1480

kilogramov, vse skupaj pa so izstrelili 14 satelitov. Prav tako kot pri prejšnji generaciji pa so tudi tokrat uporabljali rakete Proton.

Tretja generacija satelitov, znana pod imeni Uragan-K, je bila zasnovana za 10-12 letno dobo delovanja, poleg te izboljšave pa imajo novi sateliti tudi zmanjšano težo – samo 750 kilogramov. Ti sateliti bodo prvič izstreljeni v letu 2010. Zaradi zmanjšane teže bo lahko ena raketa Proton naenkrat v orbito ponesla 6 takih satelitov [17].

3.3 Beidou 1/COMPASS

Beidou 1 je satelitski navigacijski sistem Ljudske republike Kitajske. Do leta 2010 naj bi ga dokončali, gradnja pa bo potekala v dveh fazah. S tem sistemom se Kitajska zavzema za izgradnjo svojega lastnega neodvisnega navigacijskega sistema, ki pa bo vseeno kompatibilen z GPS, GLONASS in Galileo satelitskimi navigacijskimi sistemi.

Prvi predlogi za izgradnjo so se pojavili leta 1983. Ta predlog je vseboval le dva geostacionarna satelita. Koncept so potrdili leta 1989 z dvema komunikacijskima satelitoma, ki sta že bila v orbiti. Test je prav tako pokazal, da bi bila točnost novega sistema primerljiva z ameriškim GPS. Leta 1993 se je Beidou 1 tudi uradno začel načrtovati, šele leta 2000 pa so izstrelili prva dva satelita. Končni sistem v prvi fazi vsebuje štiri geostacionarne satelite: dva operativna in pa dva v rezervi, če se kateri od prvih dveh pokvari.

Prva dva satelita uporabljata iterativno metodo za določanje višine sprejemnika. Vsak satelit neprestano oddaja signal na celotno vidno področje, ki je obsega celotno Azijo. Sprejemnik vzame del tega signala in ga pošlje nazaj satelitu. Nato satelit posreduje sprejeti signal od sprejemnika k nadzornemu centru sistema Beidou 1. Računalnik v tem nadzornem centru potem določi razdaljo med satelitom in sprejemnikom na podlagi časovnih oznak, ki jih nosi novi signal, ki ga je oddal sprejemnik in pa originalnega signala, poslanega s satelita. Nato nadzorni center združi podatek o približni nadmorski višini sprejemnika (in posledično podatek o oddaljenosti od središča Zemlje) s prejšnjim podatkom o višini ali pa glede na neko približno ugibanje glede na teren. Skupaj s tem podatkom o razdalji sprejemnika do obeh satelitov dobi prvi približek o poziciji sprejemnika. Ko je to opravljeno, se podatek primerja s shranjeno karto področja v sistemu in naredi nov izračun o nadmorski višini. Ko je to

izračunano, se postopek zopet ponovi še za pozicioniranje. Posledično se z vsakim takim merjenjem in prilagajanjem izračuna bolj točno lokacijo in višino sprejemnika.

Čeprav ima ta koncept v teoriji enake ali vsaj podobne karakteristike kot GPS, ima v praksi nekaj omejitev. Uporabnik mora oddajati signal, če hoče izvedeti svojo lokacijo. Poleg tega tudi nadzorni center predstavlja lahko tarčo v primeru napada, saj je samo en. Kljub temu pa je sistem več kot dovolj dober za splošno rabo, vse to pa je omogočeno samo z dvema satelitoma. V primerjavi z GPS, GLONASS ali Galileo programom, ki za delovanje potrebujejo nekaj deset satelitov, je to vsekakor cenejša in enostavnejša rešitev.

Ker je največja omejitev sistema pokritost (z dvema geostacionarnima satelitoma si ne morejo zagotoviti popolne pokritosti,) načrtuje Kitajska izgradnjo COMPASS satelitskega sistema. Končni sistem bo vseboval 35 satelitov, med katerimi bo tudi 5 geostacionarnih za kompatibilnost z Beidou 1. Novi sistem bo pokrival celotno Zemljo, za uporabo pa bosta na voljo dve različici: komercialna, ki jo bodo uporabljali civilisti in komercialne družbe, in pa vojaška različica, namenjena potrebam vojske. Kitajska načrtuje, da bo sistem s polno funkcionalnostjo zaživel leta 2015 [11].

3.4 Galileo

Galileo je globalni navigacijski satelitski sistem, ki ga gradita Evropska unija (ang. European Union, krat. EU) in Evropska vesoljska agencija (ang. European Space Agency, krat. ESA). Projekt, vreden 3,4 milijarde evrov, je alternativa in dodatek ameriškemu GPS in ruskemu GLONASS. Sistem naj bi začel delovati leta 2013.

Ko bo enkrat deloval, bo imel Galileo dva nadzorna centra na Zemlji. Enega blizu Munchna v Nemčiji in drugega v Fucinu blizu Rima [2]. Galileo naj bi imel bolj natančne meritve kot GPS ali GLONASS, saj naj bi imel ločljivost manj kot 1 meter, prav tako pa bo omogočal merjenje višine nad gladino morja. Izboljšana bo tudi točnost lociranja v višinah. Politični cilj projekta je izdelava navigacijskega sistema, ki bo neodvisen od ostalih in bo dostopen državam EU tudi v času morebitne vojne ali političnih nesoglasij, saj bi lahko Amerika ali Rusija s pomočjo kodiranja svoje satelite zaklenila pred neavtorizirano uporabo.

Tako kot GPS bo imel tudi Galileo dve možnosti uporabe. Brezplačna z manjšo točnostjo bo na voljo vsem, plačljiva oz. licencirana uporaba pa bo na voljo vojski in

pogodbenikom, ki bodo morali za pridobitev večje natančnosti predložiti ustrezne dokumente o nameri in storitve plačati.

Končni sistem bo sestavljen iz dveh zemeljskih nadzornih centrov in 30 satelitov v srednji zemeljski orbiti. Vsak satelit bo težak 675 kilogramov, deloval pa naj bi 12 let [13].

3.4.1 Zgodovina in razvoj

Leta 1999 so bili primerjani različni koncepti iz Nemčije, Francije, Italije in Anglije in na koncu je bil izbran kompromis, ki ga je izbrala ekipa inženirjev iz vseh štirih držav. Prva faza programa Galileo je bila s strani Evropske unije in Evropske vesoljske agencije potrjena maja 2003. Sistem naj bi bil primarno zgrajen za potrebe civilistov, kar je v nasprotju z ameriškim GPS, ki ga upravlja vojska. Po drugi strani pa si evropska skupnost prav tako pridržuje pravico do ekskluzivne uporabe Galileo sistema za svojo vojsko, v kolikor bi prišlo do take potrebe.

EU je imela kar nekaj finančnih težav pri pridobivanju sredstev za projekt Galileo. Tudi mednarodna politika projektu ni bila naklonjena. Po 11. septembru 2001 so ZDA poslale evropski skupnosti pismo, v katerem so izrazile svoje nasprotovanje projektu, saj naj bi s tem onemogočili možnost ZDA, da v vojnih časih uporabijo GPS izključno zase in za svoje vojaške namene. Pritiski so se vršili do te mere, da so ključni ljudje pri projektu Galileo januarja 2002 izjavili, da je projekt praktično mrtev. Na srečo se je situacija nekaj mesecev zatem drastično spremenila. Deloma tudi za to, ker je EU na podlagi protestnih pisem ZDA ugotovila, da dejansko potrebuje svoj neodvisni satelitski navigacijski sistem in da se pri tako delikatni stvari ne morejo zanašati na tujo velesilo [13].

EU in ESA sta se marca 2002 dogovorili glede plačevanja projekta. Ocena stroškov se je gibala okoli 1,1 milijarde evrov. Sateliti, potrebnih bi bilo 30, naj bi bili izstreljeni med leti 2006 in 2010. Prav tako naj bi leta 2010 celotni sistem postal polno operativen [13]. Stroški pa so se kopičili in popravljena ocena potrebnih finančnih sredstev se je dvignila na 3 milijarde evrov, v tej ceni pa so bile zajete tudi postaje na Zemlji, ki bi bile zgrajene med leti 2006 in 2007. Načrt je prav tako predvideval, da bi zasebni vlagatelji prispevali dve tretjini potrebnega denarja, EU in ESA pa ostalo tretjino. Osnovna natančnost bi bila za uporabo na voljo brezplačno, komercialna storitev z izboljšano natančnostjo pa proti plačilu.

Projekt je bil ponovno označen za mrtvega v letu 2007, saj ni bilo denarja od zasebnih investorjev, prav tako pa je imela EU finančne težave. Glede na to, da je bil do leta 2007 izstreljen le en sam satelit, je veliko ljudi nad projektom Galileo obupalo.

Celotno leto 2007 je minilo v znamenju iskanja sredstev. Iz rebalansa proračuna so sprva dobili 548 milijonov evrov, pozneje pa še malo večji znesek iz proračuna za agrikulturo in nekaj iz administrativnih stroškov. Prav tako so olajšali proces priključevanja zasebnih investorjev, kar je na koncu pomenilo, da so uspeli zbrati dovolj sredstev za ponovni zagon programa.

Do aprila 2008 je bilo zbranih 3,4 milijarde evrov, kar pa je po novih ocenah zadosti le za razvoj in izgradnjo nekaj satelitov in baznih postaj na Zemlji. Glede na trenutno globalno finančno krizo pa je končni datum še vedno pod vprašajem in se vztrajno odmika od načrtovanega leta 2013.

4. UPORABA GNSS IN KONCEPTI UPORABE

4.1 Vojaški nameni

GNSS je bil že od nekdaj nekaj, kar si je vsaka vojaška sila želela. Zato tudi ni sporno, da je prav vojska bila prva, ki je začela razvijati satelitsko navigacijo. V nadaljevanju si ogledamo nekaj tipičnih nalog, ki jih GNSS opravlja v službi vojske.

4.1.1 Navigacija

Za vojaka, ki na sovražnem ozemlju izvaja akcije ponoči, je eden največjih izzivov navigacija, predvsem zaradi nepoznanega terena in zaradi pomanjkanja prepoznavnih zemeljskih točk. Vojaki so skozi celotno zgodovino uporabljali nočno nebo za orientacijo, a določanje točne lokacije je bilo vedno nemogoče. Da vojaki resnično potrebujejo napravo, ki bi jim omogočala določitev svojega položaja in pa lokacijo sovražnika, se je pokazalo v Zalivski vojni leta 1990. Ob pričetku vojne je bilo vojakom danih 1000 ročnih satelitskih navigacijskih naprav, ob koncu vojne pa se je to število povzpelo na 9000 [3]. GPS sprejemniki zelo hitro nadomeščajo običajne kompase v osnovni opremi vojakov. Specialne enote se poslužujejo GPS naprav tudi pri uničenju ključnih tarč. Take ekipe uporabijo artilerijo ali podporo iz zraka, ki je relativno zelo točna, predvsem zaradi posredovanja natančnih podatkov o položaju tarče.

Sodobna artilerija se mora premikati zelo hitro, da dohaja napadalni val ali pa da se umakne sovražnemu ognju. To je mogoče doseči bistveno hitreje z uporabo GPS tehnologije, saj se naslednja primerna lokacija bodisi za streljanje bodisi za skrivanje lahko določi v naprej oziroma zelo hitro, ko se pojavi potreba po tem. Na ta način prihranimo dragoceni čas, ki ga v boju zagotovo ni. Namesto da bi se ukvarjali z načrtovanjem poti in branjem zemljevidov, se tako vojaki lahko osredotočijo na boj.

4.1.2 Sledenje

V vojaškem scenariju je potrebno potencialne tarče konstantno spremljati in jim slediti, preden so označene za sovražne, in se jih lotiti z različnimi orožji. Ti podatki o sledenju se prenašajo v sodobna orožja, kot so razne rakete in pametne bombe. ZDA so razvile GPS TDARDS (ang. Global Positioning System Truth Data Acquisition, Recording and Display

System). To je kompaktna, lahka, poceni in izjemno prenosljiva naprava, ki uporablja GPS podatke v realnem času, radijsko komunikacijo in računalniško tehnologijo, in s tem zagotavlja zelo točno realno-časovno, čas-prostor informacijo (ang. Time-Space Position Information, krat. TSPI) nad desetimi objekti, kot so vozila, helikopterji ali letala [3]. Sistem je zelo modularen in se ga da dokaj enostavno prilagoditi spremenjenim razmeram ali posebnim zahtevam.

4.1.3 Vodenje raket in pametnih bomb

Moderni sistemi oborožitve so zasnovani tako, da uporabljajo podatke iz GPS sistema za sledenje in namerjanje. Ameriška vojska ima rakete dolgega in srednjega dosega, ki svojo trenutno pozicijo v realnem času določajo glede na GPS podatke in z njimi popravljajo svojo smer leta, da svoj cilj zadenejo kar se da točno. MLRS vozilo ameriške vojske (ang. Multiple Launched Rocket System), prikazano na »Sliki 13«, uporablja GPS signal za izredno hitro določanje optimalnega položaja vozila in njegove izstrelitvene ploščadi, kar mu omogoča, da lahko hitro pripelje na mesto izstrelitve, že pred tem usmeri izstrelitveno ploščad, izstreliti rakete in pobegne na naprej izračunano optimalno lokacijo, še preden sovražnik lahko reagira [3].



Slika 13: MLRS vozilo ameriške vojske [23]

4.1.4 Reševanje

Reševanje in hitro odziv v krizi je še eno področje, kjer se GNSS tehnologija izkaže za neprecenljiv pripomoček. Vojska ZDA uporablja sistem CSEL (ang. Combat Survivor Evader Locator) na »Sliki 14«. V osnovi je sistem sestavljen iz radia in GPS sprejemnika, kar reševalcem omogoča izredno hitro lociranje in posledično reševanje. CSEL preko več satelitov odda podatek o svoji lokaciji in omogoči varno dvostransko komunikacijo med pogrešanim in bazo [3].



Slika 14: CSEL postaja [23]

4.2 Policijski in gasilski nameni

Nadzorni center policijske uprave mora v vsakem trenutku vedeti, kje se nahaja neko policijsko vozilo. Pred uvedbo GPS sledenja je na klic v sili odhitelo vozilo, ki je bilo zadolženo za to področje, sedaj pa z uvedbo GPS sledenja nadzorni center pošlje vozilo, ki je najbližje kličočemu, četudi to vozilo ni zadolženo za tisto področje, zaradi česar je prihranjenega veliko časa. Policisti, ki imajo to tehnologijo, preko iste naprave dobivajo tudi napotke in fotografije kraja, kamor se peljejo. Te informacije vsebujejo podatek o tem ali gre za eno ali večstanovanjsko hišo, koliko nadstropij ima in kaj je na drugi strani objekta.

Poleg teh urgentnih zadev pa se glede na število patroljnih vozil uvedba GPS zdi smiselna tudi stroškovno. Kadar se predvidene poti patroljiranja določijo avtomatsko v bazi je manj možnosti, da bi policijska vozila patroljirala preblizu, preveč narazen ali pa povsem

po nepotrebnem. V ameriškem mestu Franklin so se v praksi pokazali konkretni prihranki pri gorivu in zahtevanem številu policistov [9].

Podobne naprave kot policisti imajo tudi gasilci, ki prav tako potrebujejo podatke o tem kje točno se nahaja določena zgradba, kakšna je glede na izdelavo in kaj se nahaja v okolici. Izračun najkrajše poti pa je seveda nujen, da je na objektu narejeno čim manj škode ob prihodu gasilcev.

4.3 Civilni nameni

4.3.1 Pomorska navigacija

Kmalu po uvedbi GPS naprav se je pojavila potreba po GPS navigaciji za večja in manjša vodna plovila. Pomorščaki so hitro uvideli prednost teh naprav, saj se niso hoteli pretirano ubadati z branjem pomorskih kart in kompasa. Pomorska GPS navigacija vse to opravi namesto njih in kar je najpomembneje: hitreje in bolj natančno. Naprava ima v sebi shranjeno digitalizirano bazo pomorskih kart, na katere pa lahko pomorščaki dodajajo svoje opombe. Skupaj s sonarjem se izjemno težko zgodi, da bi se ladja z delujočim GPS sprejemnikom zaletela v drugo ladjo ali pa v čeri. Velika prednost uporabe teh naprav je navigacija v pristaniščih in v prometnih vodnih poteh. Ladje, ki so opremljene z GPS sprejemnikom in avtomatskim informacijskim sistemom (ang. Automated Information System, krat. AIS), kot je ta na »Sliki 15«, si med seboj izmenjujejo podatke in se usklajujejo. Še posebno to pride prav pri večjih tankerjih, ki se ne morejo ustaviti prav hitro, vsak najmanjši manever pa pomeni tudi izgubo časa in izgubo dragocenega goriva. Odličen primer je Hamburg Express - ena največjih ladij za prevoz kontejnerjev na svetu. Ta 320 metrov dolga ladja s porabo goriva 200 ton dnevno se enostavno ne more ustaviti ali obračati za vsako nepredvideno stvar, ki jo z GPS tehnologijo zlahka eliminiramo, niti si tega ne more privoščiti.



Slika 15: advanSea T.56 ladijski GPS sprejemnik s prikazom pomorske karte, kompasa, hitrosti vetra in temperature motorja

4.3.2 Cestna navigacija

Najbolj poznana komponenta GPS navigacije je cestna navigacija. Naprave, ki jih imamo v avtu bodisi vgrajene s strani proizvajalca bodisi kupljene posebej, nam javljajo, kje se nahajamo, preračunavajo pot, po kateri moramo iti, prikazujejo hitrost, čas, nadmorsko višino in še cel kup potrebnih in malo manj potrebnih podatkov. Novejše naprave nam tako grafično prikažejo odcep, po katerem moramo peljati, pokažejo nam realno sliko mesta, po katerem se vozimo, »Slika 16«, in nam dajo na izbiro več možnih poti; najkrajša, najhitrejša, najbolj ekonomična. Voznik lahko prav tako izbere pot, na kateri ne bo cestninskih postaj ali pa opcijo, da se pelje čez čim več različnih držav. Večina sodobnih naprav ima prav tako možnost spremljanja tako GPS kot GLONASS satelitov, kar pomeni, da ima naprava v danem trenutku na voljo preko 30 satelitov, kar je povsem dovolj za nemoteno navigacijo v vseh vremenskih pogojih.

Ker se bomo cestni navigaciji bolj posvetili v nadaljevanju, bi si v tem delu ogledali še koncept uporabe GPS sprejemnikov v prihodnosti. Veliko je namreč govora o mrežnem modelu razporejanja prometa, nekakšni mrežni inteligenci, s katero bi razbremenili najbolj obremenjene ceste. Ideja je, da bi vsak avtomobil imel v sebi integriran GPS sprejemnik in FM oddajnik. Vozila bi bila v nekem centralnem računalniku zbrana na mrežo, ki predstavlja ceste. Glede na zasedenost ene bi se vozila preusmerila na drugo oz. če bi prišlo do prometne nesreče, bi vsa vozila, ki se jih to tiče, sprejela informacijo in se prerazporedila drugam za nemoten potek. Izboljšan GPS z dobrimi LAAS postajami igra pri tem, sicer futurističnem modelu, ključno vlogo.



Slika 16: Najboljše kar trenutno premore cestna navigacija. Navigon 8410 pokaže celo sliko mesta po katerem se vozimo.

4.3.3 Letalska navigacija

GPS je v letalstvu prisoten že od svojega začetka in pilotom nudi obilo koristnih informacij, ki jih potrebujejo, da si olajšajo delo ter omogočijo varnejše polete, kot je prikazano na »Sliki 17«. GPS naprava v letalu preračunava svojo lokacijo izjemno natančno, saj ima poleg navadnega GPS signala na voljo tudi WAAS signal ter v okolici letališč še LAAS. Z vsemi temi podatki ter radarjem je možno vsak morebiten trk predvideti dovolj zgodaj, da se mu letalo lahko izogne. Prav tako je GPS v navezi z vremenskimi poročili nenadomestljiv pripomoček za zmanjšanje porabe goriva, saj lahko pred in med potjo letalski računalnik preračuna optimalno pot glede na trenutne vremenske razmere. Morda se sliši neumno, a če vzamemo za primer Airbus A346, ki porabi 1500 litrov goriva na 100 kilometrov, hitro vidimo, da lahko z 10% boljším izkoristkom pri daljšem letu in ob trenutnih cenah goriva letalska družba prihrani ogromno denarja, če let optimizira. Svetla prihodnost se GPS sistemu obeta tudi pri samem pristajanju letal na letališčih. Trenutna ločljivost še ni dovolj dobra za avtomatizira pristaneke, tako da je ta del še vedno prepuščen pilotom, a v ne tako oddaljeni prihodnosti se pričakuje, da bodo letala lahko varno pristala brez človeškega posredovanja. Trenutno sicer letala na dolgih letih usmerja in kontrolira avtomatski pilot, ki je program, ki upošteva vse relevantne podatke s senzorjev in GPS sprejemnika, da prispe do svojega cilja praktično brez posredovanja pilota ali kopilota.



Slika 17: Letalska navigacija Garmin G500

Pri vojaških letalih je GPS seveda še toliko bolj pomemben, saj piloti ne poznajo terena, prav tako pa se tudi večina današnjih izstrelkov tako ali drugače orientira glede na podatke internega GPS sprejemnika.

Tretja svetla stran uporabe GPS v letalstvu pa so letala brez posadke, kot je ta na »Sliki 18«. Trenutno gre le za vojaška letala, ki se uporabljajo ali za iskanje in markiranje cilja ali pa za izvidniška letala. Ta letala iz kontrolne sobe spremlja računalniški tehnik, njihovo upravljanje pa je podobno kot v kakšni računalniški igri. Največkrat imajo ta letala v svojem programu vnaprej definirano pot, katere se držijo prav zaradi integriranega GPS sprejemnika. Glede na uspehe, ki jih dosegajo ta letala v zadnjih letih, je realno pričakovati tudi komercialna letala brez posadke, ki pa bodo za svoje delovanje nujno potrebovala boljše GPS sprejemnike in LAAS enote, da ne bo prihajalo do nepotrebnih zapletov.



Slika 18: Letalo brez posadke MQ-9 Reaper

4.3.4 Uporaba v športih

GPS navigacija je zelo uporabna tudi pri športnih aktivnostih. V zadnjem času je nepogrešljivi pripomoček tekačev, triatloncev, kolesarjev ipd. Garmin Forerunner, kot je ta na »Sliki 19«. To je športna ura z GPS sprejemnikom, ki meri razdaljo, hitrost, čas, višino in ritem, ti podatki pa profesionalnim in boljšim amaterskim športnikom pridejo še kako prav bodisi pri zapisovanju rezultatov ali pa za načrtovanje novih treningov. S tem pripomočkom lahko športniki po končani vadbi pogledajo, kje točno so se gibali, kakšna je bila njihova hitrost, srčni utrip in poraba kalorij na določenem delu. S primerjanjem rezultatov večjega števila treningov se tako lahko oblikuje slika zmogljivosti posameznika in šibke točke, vse pa to po navadi ponuja programska oprema, ki jo priloži proizvajalec.



Slika 19: Garmin Forerunner 205 s prikazano razdaljo in nadmorsko višino treninga

5. STROJNA IN PROGRAMSKA OPREMA

5.1 Programska oprema – operacijski sistemi

Še tako dobra strojna oprema ni vredna kaj veliko, če na njej ni nameščene dobre programske opreme. Poznamo kar nekaj večjih proizvajalcev cestnih navigacijskih naprav ter množico manjših oz. OEM proizvajalcev, ki se pretežno nahajajo na Kitajskem. Večina jih uporablja neko verzijo Windows ali Linux operacijskega sistema, nekaj pa jih je razvilo tudi svojega. Glede na množičnost teh sistemov si bomo ogledali samo večje izmed njih.

5.1.1 Windows CE

Najbolj uporabljan operacijski sistem v cestni navigaciji je zagotovo Windows CE (ang. Windows Embedded Compact). Operacijski sistem je narejen za manj zmogljive naprave, predvsem dlančnike, POS terminale, GPS navigacije... Windows CE je lahko naložen na napravah, ki vsebujejo Intel x86, MIPS, ARM ali Hitachi SuperH procesorje [25].

Windows CE je optimiziran za naprave, ki imajo minimalno količino internega spomina. Samo jedro tega operacijskega sistema lahko teče na sistemu z manj kot 1 MB pomnilnika. Naprave so velikokrat zgrajene brez trdega diska ali drugega večjega pomnilnika in so po navadi zaklenjene, tako da uporabnik ne more dodati kakšnih svojih dodatkov. Vse skupaj je namreč naloženo v bralni pomnilnik (ang. Read Only Memory, krat. ROM). Windows CE deluje v realnem času in za svoje nemoteno izvajanje uporablja sistem niti (ang. Threads) ter 256 nivojev prekinitvev (ang. Interrupt). Posledica takega pristopa je nizek odzivni čas in transparentnejše delovanje [25].

Kot rečeno je Windows CE zelo razširjen operacijski sistem, čeprav se morda tega večina ljudi niti ne zaveda. Iz osnovne Windows CE verzije smo za različne naprave dobili različne verzije, ki so: AutoPC, Pocket PC 2000, Pocket PC 2002, Windows Mobile 2003, Windows Mobile 2003 SE, Windows Mobile 5.0, Windows Mobile 6.0, Windows Mobile 7.0, Smartphone 2002, Smartphone 2003 in Portable Media Center. Windows CE je bil celo prisoten pri legendarni Sega igralni konzoli Dreamcast [25].

Velika razlika med Windows CE in ostalimi Microsoft operacijskimi sistemi je ta, da je velik del Windows CE ponujen v izvorni kodi. Za ta korak so se odločili zato, da si lahko

proizvajalci sami prilagodijo sistem, ki bo kar najbolje služil njihovi napravi. Tak primer je viden na »Sliki 20«, kjer je prikazana naprava kitajskega OEM proizvajalca in pa Windows CE, ki teče v ozadju.



Slika 20: JAT Technology GPS-3503. V ozadju teče navigacijski program, spredaj pa je viden Windows CE 4.2

5.1.2 Symbian

Symbian je operacijski sistem, razvit za potrebe mobilnih naprav in pametnih telefonov. Za razliko do Windows CE lahko Symbian teče le na ARM procesorjih. Ker je Symbian razvit za potrebe počasnejših naprav, razvijalci že od vsega začetka poudarjajo, da mora biti vsaka izdaja tega operacijskega sistema podvržena določenim omejitvam. Pri vsaki izdaji se upoštevajo trije glavni principi: integriteta in varnost uporabnikovih podatkov ima najvišjo prioriteto, uporabnikov čas je dragocen in vsi resursi v sistemu so omejeni [24].

Da bi se teh načel kar najbolje držali, so za Symbian razvili mikrojedro, ki deluje po principu strežnika. Operacijski sistem je optimiziran za naprave, ki za svoje delovanje potrebujejo malo energije in ki imajo praktično vse potrebne podatke shranjene na ROMu. Aplikacije in celotni OS so objektno orientirani, tipični uporabniški vmesnik pa je prikazan na »Sliki 21«.



Slika 21: Grafični vmesnik Symbian naprav

5.1.3 Linux

Linux je brezplačna, odprtokodna alternativa Symbian, Windows CE, PalmOS itd. operacijskim sistemom. Za razvijalce je ta opcija izjemno zanimiva, saj je kot rečeno brezplačna v nasprotju z Windows CE, največja prednost pa je to, da je izvorna koda v celoti dostopna. V praksi to pomeni, da lahko razvijalci manjših naprav prilagodijo operacijski sistem potrebam, ki jih njihova naprava ima, brez da bi le-tega razvijali na novo od začetka. Ker je ta pristop bistveno lažji in cenejši, kot ga ponujajo ostali, ima Linux sorazmerno velik delež pri pametnih telefonih, dlančnikih in GPS navigacijah. Zaradi svoje odprtokodnosti in možnosti popravljanja ter dopolnjevanja funkcionalnosti je Linux primeren za širok spekter uporabe, ki sega od strežnikov in namiznih računalnikov pa vse do »operacijskega sistema«, ki žene DVD predvajalnik.

Različice Linuxa, ki so primerne za razvijalce GPS programov, so predvsem tiste, ki tečejo na pametnih telefonih. Trenutno najpopularnejši različici sta Android in Maemo, za kateri lahko razvijalec dobi vso izvorno kodo in tako svoje aplikacije dodatno optimizira za delovanje na teh napravah oziroma nanju priključi svoj GPS modul in po potrebi napiše tudi ustrezne gonilnike in protokole prenosa informacij [19].

5.2 Razvoj osnovne lastne aplikacije

V tem poglavju si bomo ogledali potek programiranja in možne probleme, na katere lahko naletimo pri pisanju navigacijske programske opreme.

5.2.1 Osnovni stavek

Za lažjo predstavo surovih GPS podatkov je poskrbel ameriški National Marine Electronics Association (krat. NMEA), ki je v industrijo vpeljal standard, ki se sedaj uporablja v večini GPS naprav. NMEA podatki so poslani kot stavek v katerem so podatki ločeni z vejico, vsebina pa predstavlja podatke, ki so povezani s prvo besedo podatkovnega stavka. Obstaja več kot 50 različnih krmilnih besed, običajni interpreter pa jih potrebuje le nekaj, da opravi svojo funkcijo. Najbolj razširjena krmilna beseda od vseh je priporočeni minimum (ang. Recommended Minimum), ki se začne z \$GPRMC. Celotna koda \$GPRMC stavka je vidna na »Sliki 22«.

```
$GPRMC,040302.663,A,3939.7,N,10506.6,W,0.27,358.86,200804,,*1A
```

Slika 22: \$GPRMC stavek

En tak stavek vsebuje skoraj vse podatke, ki jih GPS aplikacija potrebuje: zemljepisno višino in širino, hitrost, smer, čas v satelitu, status lociranja in magnetno variacijo.

5.2.2 Jedro interpreterja

Programiranje navigacijskega programa je močno olajšano, če za svoj projekt uporabljamo GPS sprejemnik, ki na izhod pošilja standardiziran NMEA format podatkov. S tem pristopom se manj truda posveča samemu branju podatkov iz sprejemnika, več pa ga lahko posvetimo optimizaciji kode in pa grafičnemu prikazu podatkov. Koda, ki je predstavljena v tem poglavju, je v prvem delu spisana v programskem jeziku Visual Basic, v drugem delu pa v C#. Ta dva programska jezika sta nekoliko bolj primerna za pisanje navigacijskih programov, saj imata boljšo podporo za serijsko komunikacijo in vgrajene sisteme (ang. Embedded systems) na splošno. Seveda pa to še zdaleč nista edina programska jezika, v katerih lahko pišemo navigacijske programe. Programerji, izkušeni v Javi, lahko izkoristijo Java Communications 3.0 API, znan tudi kot javax.comm, ki je dodatek Javi in omogoča mnogo dodatnih pristopov pri programiranju vgrajenih sistemov.

Prvi korak pri kreiranju NMEA interpreterja je pisanje metode, ki naredi dve stvari: razdeli posamezni stavek na posamezne besede in pa ugotovi pomen prve besede. Potrebna koda je vidna na »Sliki 23«.

```
'Jedro NMEA interpreterja

Public Class NmeaInterpreter
    ' Procesira informacija z GPS sprejemnika
    Public Function Parse(ByVal sentence As String) As Boolean
        ' Stavek razdeli na besede
        Dim Words() As String = GetWords(sentence)
        ' Pogleda prvo besedo, da izve kam mora iti naprej
        Select Case Words(0)
            Case "$GPRMC" ' "Recommended Minimum" stavek je bil najden!
                ' Stavek je bil prepoznan
                Return True
            Case Else
                ' Stavek ni bil prepoznan
                Return False
        End Select
    End Function
    ' Razdeli stavek na posamezne besede
    Public Function GetWords(ByVal sentence As String) As String()
        Return sentence.Split(",")
    End Function
End Class
```

Slika 23: Ugotavljanje pomena prve besede v VB.NET programskem jeziku

V naslednjem koraku izluščimo informacije, začnemo pa z zemljepisno višino in širino. Višina in širina sta shranjeni v formatu »DDD°MM'SS.S«, kjer D predstavlja ure, M minute in S sekunde. Koordinate so lahko zapisane tudi v krajšem formatu recimo »DD°MM.M« ali celo samo »DD°«. Četrta beseda v stavku, »3939.7«, predstavlja trenutno zemljepisno višino kot ure in minute (39°39.7'), le da so številke napisane skupaj. Prve dve številki predstavljata ure ostale tri pa minute. Zemljepisna širina je predstavljena podobno, le da so ure predstavljene s tremi številkami. Peta in sedma beseda predstavljata hemisfero kjer N pomeni sever in W pomeni zahod. Podatek o hemisferi je dodan numeričnemu delu in s tem dobimo popolno meritev. NMEA interpreterji so zelo pregledni in relativno lahki za uporabo, saj so proženi dogodkovno. To pa zato, ker podatki prihajajo v povsem naključnem vrstnem redu. Dogodkovno prožen razred daje interpreterju največ fleksibilnosti in odzivnosti, ki jo potrebuje za izvajanje aplikacij. V primeru iz »Slike 24« se bo prvi dogodek, PositionReceived, sprožil vedno, kadar sta prejeti zemljepisna višina in širina.

```
'Jedro NMEA interpreterja

Public Class NmeaInterpreter
    ' Procesira informacija z GPS sprejemnika
    Public Function Parse(ByVal sentence As String) As Boolean
        ' Stavek razdeli na besede
```

```

Dim Words() As String = GetWords(sentence)
' Pogleda prvo besedo, da izve kam mora iti naprej
Select Case Words(0)
    Case "$GPRMC"
        ' "Recommended Minimum" stavek je bil najden!
        ' Stavek je bil prepoznan
        Return True
    Case Else
        ' Stavek ni bil prepoznan
        Return False
End Select
End Function
' Razdeli stavek na posamezne besede
Public Function GetWords(ByVal sentence As String) As String()
    Return sentence.Split(",")
End Function
' Interpretira $GPRMC sporočilo
Public Function ParseGPRMC(ByVal sentence As String) As Boolean
    ' Razdeli stavek na posamezne besede
    Dim Words() As String = GetWords(sentence)
    ' Imamo dovolj podatkov za opis lokacije?
    If Words(3) <> "" And Words(4) <> "" And Words(5) <> "" And _
        Words(6) <> "" Then
        ' Da. Razberi podatke o zemljepisni višini in širini
        Dim Latitude As String = Words(3).Substring(0, 2) & "°" ' Pripni ure
        Latitude = Latitude & Words(3).Substring(2) & "" ' Pripni minute
        Latitude = Latitude & Words(4) ' Pripni hemisfero
        Dim Longitude As String = Words(5).Substring(0, 3) & "°" ' Pripni ure
        Longitude = Longitude & Words(5).Substring(3) & "" ' Pripni minute
        Longitude = Longitude & Words(6) ' Pripni hemisfero
        ' Klicočo aplikacijo obvesti o spremembi
        RaiseEvent PositionReceived(Latitude, Longitude)
    End If
    ' Obvesti, da je bil stavek prepoznan
    Return True
End Function
End Class

```

Slika 24: Dogodkovno prožen interpreter v VB.NET programskem jeziku

Nekatere GPS naprave bodo za podatke, za katere nimajo vrednosti, podale prazno polje, zato je dobro, da vsako besedo testiramo preden jo prevedemo.

5.2.3 Čiščenje napak

Kontrolna vsota je izračunana kot logična funkcija XOR med bajti med znakoma \$ in *. Ta vsota se potem primerja z vsoto s stavka. Če se vsoti ne ujemata, je stavek največkrat zavržen, kar pa je načeloma v redu, saj se ti podatki pošljejo vsakih nekaj sekund. Z možnostjo preverjanja kontrolnih vsot pa interpreter lahko zavrže tudi stavke z neveljavno kontrolno vsoto, kot je to prikazano na »Sliki 25«.

```

'Jedro NMEA interpreterja

Public Class NmeaInterpreter
    ' Aktivira se kadar je trenutna lokacija spremenjena
    Public Event PositionReceived(ByVal latitude As String, _
        ByVal longitude As String)
    ' Procesira informacije z GPS sprejemnika
    Public Function Parse(ByVal sentence As String) As Boolean
        ' Zavrže stavek kadar se kontrolna vsota ne ujema z izračunano kontrolno vsoto
        If Not IsValid(sentence) Then Return False
    End Function
End Class

```

```

' Pogleda prvo besedo, da izve kam mora iti naprej
Select Case Words(0)
Case "$GPRMC" ' "Recommended Minimum" stavek je bil najden!
' Stavek je bil prepoznan
Return True
Case Else
' Stavek ni bil prepoznan
Return False
End Select
End Function
' Razdeli stavek na posamezne besede
Public Function GetWords(ByVal sentence As String) As String()
Return sentence.Split(",","c")
End Function
' Interpretira $GPRMC sporočilo
Public Function ParseGPRMC(ByVal sentence As String) As Boolean
' Razdeli stavek na posamezne besede
Dim Words() As String = GetWords(sentence)
' Imamo dovolj podatkov za opis lokacije?
If Words(3) <> "" And Words(4) <> "" And Words(5) <> "" And _
Words(6) <> "" Then
' Da. Razberi podatke o zemljepisni višini in širini
Dim Latitude As String = Words(3).Substring(0, 2) & "°" ' Pripni ure
Latitude = Latitude & Words(3).Substring(2) & "" ' Pripni minute
Latitude = Latitude & Words(4) ' Pripni hemisfero
Dim Longitude As String = Words(5).Substring(0, 3) & "°" ' Pripni ure
Longitude = Longitude & Words(5).Substring(3) & "" ' Pripni minute
Longitude = Longitude & Words(6) ' Pripni hemisfero
' Kličočo aplikacijo obvesti o spremembi
RaiseEvent PositionReceived(Latitude, Longitude)
End If
' Obvesti, da je bil stavek prepoznan
Return True
End Function
' Vrne True če se kontrolna vsota stavka ujema z izračunano kontrolno vsoto
Public Function IsValid(ByVal sentence As String) As Boolean
' Primerja znake za zvezdico z izračunom
Return sentence.Substring(sentence.IndexOf("*") + 1) = GetChecksum(sentence)
End Function
' Izračuna kontrolno vsoto za stavek
Public Function GetChecksum(ByVal sentence As String) As String
' Zanka čez vse znake, da dobimo vsoto
Dim Character As Char
Dim Checksum As Integer
For Each Character In sentence
Select Case Character
Case "$"c
' Ignoriraj znak za dolar
Case "*"c
' Ustavi operacijo pred znakom zvezdica
Exit For
Case Else
' Je to prva vrednost kontrolne vsote
If Checksum = 0 Then
' Da. Nastavi kontrolno vsoto glede na to vrednost
Checksum = Convert.ToByte(Character)
Else
' Ne. Izvedi XOR nad kontrolno vsoto in vrednostjo znaka
Checksum = Checksum Xor Convert.ToByte(Character)
End If
End Select
Next
' Vrni kontrolno vsoto v šestnajstičnem formatu z dvema znakoma
Return Checksum.ToString("X2")
End Function
End Class

```

Slika 25: Izračun kontrolne vsote v VB.NET programskem jeziku

5.2.4 Brezžična atomska ura

Čas je odločilnega pomena pri GPS tehnologiji, ker se razdalje merijo blizu svetlobne hitrosti. Vsak satelit vsebuje štiri atomske ure, ki jih uporabi za časovno označevanje radijskih sporočil, ki so med seboj ločena le nekaj nanosekund. Zanimivo je, da lahko z le nekaj vrsticami kode te atomske čase uporabimo za sinhronizacijo z računalniško uro z le nekaj milisekund dolgim zamikom. Druga beseda \$GPRMC stavka, »040302.663«, vsebuje satelitov čas v stisnjem formatu. Prva dva znaka predstavljata uro, druga dva minute in tretja dva sekunde. Vse, kar pride za tem, so milisekunde. Čas iz primera je tako 4:03:02.663 dopoldne. Treba pa je vedeti, da sateliti oddajajo čas v GMT+0 formatu, tako da moramo ta čas prilagoditi lokalnemu času. V kodi spodaj imamo tudi metodo *DateTime.ToLocalTime*, ki satelitov čas pretvori v lokalni čas.

```
'Jedro NMEA interpreterja

Public Class NmeaInterpreter
    ' Aktivira se kadar je trenutna lokacija spremenjena
    Public Event PositionReceived(ByVal latitude As String, _
        ByVal longitude As String)
    ' Procesira informacije z GPS sprejemnika
    Public Function Parse(ByVal sentence As String) As Boolean
        ' Zavrže stavek kadar se kontrolna vsota ne ujema z izračunano kontrolno vsoto
        If Not IsValid(sentence) Then Return False
        ' Pogleda prvo besedo, da izve kam mora iti naprej
        Select Case Words(0)
            Case "$GPRMC" ' "Recommended Minimum" stavek je bil najden!
                ' Stavek je bil prepoznan
                Return True
            Case Else
                ' Stavek ni bil prepoznan
                Return False
        End Select
    End Function
    ' Razdeli stavek na posamezne besede
    Public Function GetWords(ByVal sentence As String) As String()
        Return sentence.Split(",")
    End Function
    ' Interpretira $GPRMC sporočilo
    Public Function ParseGPRMC(ByVal sentence As String) As Boolean
        ' Razdeli stavek na posamezne besede
        Dim Words() As String = GetWords(sentence)
        ' Imamo dovolj podatkov za opis lokacije?
        If Words(3) <> "" And Words(4) <> "" And Words(5) <> "" And _
            Words(6) <> "" Then
            ' Da. Razberi podatke o zemljepisni višini in širini
            Dim Latitude As String = Words(3).Substring(0, 2) & "°" ' Pripni ure
            Latitude = Latitude & Words(3).Substring(2) & "" ' Pripni minute
            Latitude = Latitude & Words(4) ' Pripni hemisfero
            Dim Longitude As String = Words(5).Substring(0, 3) & "°" ' Pripni ure
            Longitude = Longitude & Words(5).Substring(3) & "" ' Pripni minute
            Longitude = Longitude & Words(6) ' Pripni hemisfero
            ' Kličočo aplikacijo obvesti o spremembi
            RaiseEvent PositionReceived(Latitude, Longitude)
        End If
        ' Imamo dovolj vrednosti, da razčlenimo satelitov čas?
        If Words(1) <> "" Then
```

```

' Da. Izlušči ure, minute, sekunde in milisekunde
Dim UtcHours As Integer = CType(Words(1).Substring(0, 2), Integer)
Dim UtcMinutes As Integer = CType(Words(1).Substring(2, 2), Integer)
Dim UtcSeconds As Integer = CType(Words(1).Substring(4, 2), Integer)
Dim UtcMilliseconds As Integer
' Izlušči milisekunde, če so na voljo
If Words(1).Length > 7 Then
    UtcMilliseconds = CType(Single.Parse(Words(1).Substring(6), _
        CultureInfo.InvariantCulture) * 1000, Integer)
End If
' Kreiraj DateTime objekt z vsemi vrednostmi
Dim Today As DateTime = System.DateTime.Now.ToUniversalTime
Dim SatelliteTime As New System.DateTime(Today.Year, Today.Month, _
    Today.Day, UtcHours, UtcMinutes, UtcSeconds, UtcMilliseconds)
' Obvesti o novem času prilagojenem lokalnemu času
RaiseEvent DateTimeChanged(SatelliteTime.ToLocalTime)
End If
' Obvesti, da je bil stavek prepoznan
Return True
End Function
' Vrne True če se kontrolna vsota stavka ujema z izračunano kontrolno vsoto
Public Function IsValid(ByVal sentence As String) As Boolean
    ' Primerja znake za zvezdico z izračunom
    Return sentence.Substring(sentence.IndexOf("*") + 1) = GetChecksum(sentence)
End Function
' Izračuna kontrolno vsoto za stavek
Public Function GetChecksum(ByVal sentence As String) As String
    ' Zanka čez vse znake, da dobimo vsoto
    Dim Character As Char
    Dim Checksum As Integer
    For Each Character In sentence
        Select Case Character
            Case "$"c
                ' Ignoriraj znak za dolar
            Case "*"c
                ' Ustavi operacijo pred znakom zvezdica
                Exit For
            Case Else
                ' Je to prva vrednost kontrolne vsote
                If Checksum = 0 Then
                    ' Da. Nastavi kontrolno vsoto glede na to vrednost
                    Checksum = Convert.ToByte(Character)
                Else
                    ' Ne. Izvedi XOR nad kontrolno vsoto in vrednostjo znaka
                    Checksum = Checksum Xor Convert.ToByte(Character)
                End If
            End Select
        Next
    ' Vrni kontrolno vsoto v šestnajstičnem formatu z dvema znakoma
    Return Checksum.ToString("X2")
End Function
End Class

```

Slika 26: Prebiranje satelitovega časa in prilagoditev na lokalni čas v VB.NET programskem jeziku

5.2.5 Smer in opozorila prekoračitve hitrosti

GPS naprave analizirajo pozicijo sprejemnika skozi čas in na podlagi tega izračunajo hitrost in smer. \$GPRMC stavek vsebuje tudi te podatke. Hitrost je vedno podana v vozlih, smer pa v azimutih. Z nekaj matematike vozle pretvorimo v poljubno enoto in razberemo smer.

```
'Jedro NMEA interpreterja

Public Class NmeaInterpreter
    ' Aktivira se kadar je trenutna lokacija spremenjena
    Public Event PositionReceived(ByVal latitude As String, _
        ByVal longitude As String)
    Public Event DateTimeChanged(ByVal dateTime As DateTime)
    Public Event BearingReceived(ByVal bearing As Double)
    Public Event SpeedReceived(ByVal speed As Double)
    Public Event SpeedLimitReached()
    ' Procesira informacije z GPS sprejemnika
    Public Function Parse(ByVal sentence As String) As Boolean
        ' Zavrže stavek kadar se kontrolna vsota ne ujema z izračunano kontrolno vsoto
        If Not IsValid(sentence) Then Return False
        ' Pogleda prvo besedo, da izve kam mora iti naprej
        Select Case GetWords(sentence)(0)
            Case "$GPRMC"
                ' "Recommended Minimum" stavek je bil najden!
                Return ParseGPRMC(sentence)
            Case Else
                ' Stavek ni bil prepoznan
                Return False
        End Select
    End Function
    ' Razdeli stavek na posamezne besede
    Public Function GetWords(ByVal sentence As String) As String()
        Return sentence.Split(",")
    End Function
    ' Interpretira $GPRMC sporočilo
    Public Function ParseGPRMC(ByVal sentence As String) As Boolean
        ' Razdeli stavek na posamezne besede
        Dim Words() As String = GetWords(sentence)
        ' Imamo dovolj podatkov za opis lokacije?
        If Words(3) <> "" And Words(4) <> "" And Words(5) <> "" And _
            Words(6) <> "" Then
            ' Da. Razberi podatke o zemljepisni višini in širini
            Dim Latitude As String = Words(3).Substring(0, 2) & "°" ' Pripni ure
            Latitude = Latitude & Words(3).Substring(2) & "" ' Pripni minute
            Latitude = Latitude & Words(4) ' Pripni hemisfero
            Dim Longitude As String = Words(5).Substring(0, 3) & "°" ' Pripni ure
            Longitude = Longitude & Words(5).Substring(3) & "" ' Pripni minute
            Longitude = Longitude & Words(6) ' Pripni hemisfero
            ' Klicočo aplikacijo obvesti o spremembi
            RaiseEvent PositionReceived(Latitude, Longitude)
        End If
        ' Imamo dovolj vrednosti, da razčlenimo satelitov čas?
        If Words(1) <> "" Then
            ' Da. Izlušči ure, minute, sekunde in milisekunde
            Dim UtcHours As Integer = CType(Words(1).Substring(0, 2), Integer)
            Dim UtcMinutes As Integer = CType(Words(1).Substring(2, 2), Integer)
            Dim UtcSeconds As Integer = CType(Words(1).Substring(4, 2), Integer)
            Dim UtcMilliseconds As Integer
            ' Izlušči milisekunde, če so na voljo
            If Words(1).Length > 7 Then
                UtcMilliseconds = CType(Single.Parse(Words(1).Substring(6), _
                    CultureInfo.InvariantCulture) * 1000, Integer)
            End If
            ' Kreiraj DateTime objekt z vsemi vrednostmi
            Dim Today As DateTime = System.DateTime.Now.ToUniversalTime
```

```

    Dim SatelliteTime As New System.DateTime(Today.Year, Today.Month, _
        Today.Day, UtcHours, UtcMinutes, UtcSeconds, UtcMilliseconds)
    ' Obvesti o novem času prilagojenem lokalnemu času
    RaiseEvent DateTimeChanged(SatelliteTime.ToLocalTime)
End If
' Imamo dovolj informacij, da izluščimo trenutno hitrost
If Words(7) <> "" Then
    ' Da. Pretvori v MPH
    Dim Speed As Double = CType(Words(7), Double) * 1.150779
    ' Če smo hitrejši od 55MPH sproži hitrostno opozorilo
    If Speed > 55 Then RaiseEvent SpeedLimitReached()
    ' Obvesti o trenutni hitrosti
    RaiseEvent SpeedReceived(Speed)
End If
' Imamo dovolj informacij da izluščimo smer?
If Words(8) <> "" Then
    ' Obvesti, da je bil stavek prepoznan
    Dim Bearing As Double = CType(Words(8), Double)
    RaiseEvent BearingReceived(Bearing)
End If
' Obvesti, da je bil stavek prepoznan
Return True
End Function

' Vrne True če se kontrolna vsota stavka ujema z izračunano kontrolno vsoto
Public Function IsValid(ByVal sentence As String) As Boolean
    ' Primerja znake za zvezdico z izračunom
    Return sentence.Substring(sentence.IndexOf("*") + 1) = GetChecksum(sentence)
End Function

' Izračuna kontrolno vsoto za stavek
Public Function GetChecksum(ByVal sentence As String) As String
    ' Zanka čez vse znake, da dobimo vsoto
    Dim Character As Char
    Dim Checksum As Integer
    For Each Character In sentence
        Select Case Character
            Case "$"c
                ' Ignoriraj znak za dolar
            Case "*"c
                ' Ustavi operacijo pred znakom zvezdica
                Exit For
            Case Else
                ' Je to prva vrednost kontrolne vsote
                If Checksum = 0 Then
                    ' Da. Nastavi kontrolno vsoto glede na to vrednost
                    Checksum = Convert.ToByte(Character)
                Else
                    ' Ne. Izvedi XOR nad kontrolno vsoto in vrednostjo znaka
                    Checksum = Checksum Xor Convert.ToByte(Character)
                End If
            End Select
        Next
    ' Vrni kontrolno vsoto v šestnajstičnem formatu z dvema znakoma
    Return Checksum.ToString("X2")
End Function
End Class

```

Slika 27: Določanje hitrosti in smeri v VB.NET programskem jeziku

5.2.6 Ali je pozicija določena?

\$GPRMC stavek vključuje tudi besedo, ki nam pove, ali je pozicija GPS sprejemnika določena. Ta določitev se zgodi, ko sprejemnik dobiva dovolj močan signal z vsaj treh satelitov. Če je vpleten še četrti satelit, lahko izračunamo tudi nadmorsko višino. Tretja beseda \$GPRMC stavka sestoji iz ene od dveh črk: »A« za aktivno povezavo in »V« za neveljavno, kadar določitev ni možna. V »Sliki 28« je dodan del, ki preveri ta znak in sporoči, ali je določitev aktivna.

'Jedro NMEA interpreterja

```
Public Class NmeaInterpreter
    ' Aktivira se kadar je trenutna lokacija spremenjena
    Public Event PositionReceived(ByVal latitude As String, _
        ByVal longitude As String)
    Public Event DateTimeChanged(ByVal dateTime As DateTime)
    Public Event BearingReceived(ByVal bearing As Double)
    Public Event SpeedReceived(ByVal speed As Double)
    Public Event SpeedLimitReached()
    Public Event FixObtained()
    Public Event FixLost()
    ' Procesira informacije z GPS sprejemnika
    Public Function Parse(ByVal sentence As String) As Boolean
        ' Zavrže stavek kadar se kontrolna vsota ne ujema z izračunano kontrolno vsoto
        If Not IsValid(sentence) Then Return False
        ' Pogleda prvo besedo, da izve kam mora iti naprej
        Select Case GetWords(sentence)(0)
            Case "$GPRMC"
                ' "Recommended Minimum" stavek je bil najden!
                Return ParseGPRMC(sentence)
            Case Else
                ' Stavek ni bil prepoznan
                Return False
        End Select
    End Function
    ' Razdeli stavek na posamezne besede
    Public Function GetWords(ByVal sentence As String) As String()
        Return sentence.Split(",")
    End Function
    ' Interpretira $GPRMC sporočilo
    Public Function ParseGPRMC(ByVal sentence As String) As Boolean
        ' Razdeli stavek na posamezne besede
        Dim Words() As String = GetWords(sentence)
        ' Imamo dovolj podatkov za opis lokacije?
        If Words(3) <> "" And Words(4) <> "" And Words(5) <> "" And _
            Words(6) <> "" Then
            ' Da. Razberi podatke o zemljepisni višini in širini
            Dim Latitude As String = Words(3).Substring(0, 2) & "°" ' Pripni ure
            Latitude = Latitude & Words(3).Substring(2) & "" ' Pripni minute
            Latitude = Latitude & Words(4) ' Pripni hemisfero
            Dim Longitude As String = Words(5).Substring(0, 3) & "°" ' Pripni ure
            Longitude = Longitude & Words(5).Substring(3) & "" ' Pripni minute
            Longitude = Longitude & Words(6) ' Pripni hemisfero
            ' Klicočo aplikacijo obvesti o spremembi
            RaiseEvent PositionReceived(Latitude, Longitude)
        End If
        ' Do we have enough values to parse satellite-derived time?
        If Words(1) <> "" Then
            ' Imamo dovolj vrednosti, da razčlenimo satelitov čas?
            If Words(1) <> "" Then
                ' Da. Izlušči ure, minute, sekunde in milisekunde
```

```

Dim UtcHours As Integer = CType(Words(1).Substring(0, 2), Integer)
Dim UtcMinutes As Integer = CType(Words(1).Substring(2, 2), Integer)
Dim UtcSeconds As Integer = CType(Words(1).Substring(4, 2), Integer)
Dim UtcMilliseconds As Integer
' Izlušči milisekunde, če so na voljo
If Words(1).Length > 7 Then
    UtcMilliseconds = CType(Single.Parse(Words(1).Substring(6), _
        CultureInfo.InvariantCulture) * 1000, Integer)
End If
' Kreiraj DateTime objekt z vsemi vrednostmi
Dim Today As DateTime = System.DateTime.Now.ToUniversalTime
Dim SatelliteTime As New System.DateTime(Today.Year, Today.Month, _
    Today.Day, UtcHours, UtcMinutes, UtcSeconds, UtcMilliseconds)
' Obvesti o novem času prilagojenem lokalnemu času
RaiseEvent DateTimeChanged(SatelliteTime.ToLocalTime)
End If
' Imamo dovolj informacij, da izluščimo trenutno hitrost
If Words(7) <> "" Then
    ' Da. Pretvori v MPH
    Dim Speed As Double = CType(Words(7), Double) * 1.150779
    ' Če smo hitrejši od 55MPH sproži hitrostno opozorilo
    If Speed > 55 Then RaiseEvent SpeedLimitReached()
    ' Obvesti o trenutni hitrosti
    RaiseEvent SpeedReceived(Speed)
End If
' Imamo dovolj informacij da izluščimo smer?
If Words(8) <> "" Then
    ' Obvesti, da je bil stavek prepoznan
    Dim Bearing As Double = CType(Words(8), Double)
    RaiseEvent BearingReceived(Bearing)
End If
'Ima naprava »fix« s satelitom?
If Words(2) <> "" Then
    Select Case Words(2)
        Case "A"
            RaiseEvent FixObtained()
        Case "V"
            RaiseEvent FixLost()
    End Select
End If
' Obvesti, da je bil stavek prepoznan
Return True
End Function

' Vrne True če se kontrolna vsota stavka ujema z izračunano kontrolno vsoto
Public Function IsValid(ByVal sentence As String) As Boolean
    ' Primerja znake za zvezdico z izračunom
    Return sentence.Substring(sentence.IndexOf("**") + 1) = GetChecksum(sentence)
End Function

' Izračuna kontrolno vsoto za stavek
Public Function GetChecksum(ByVal sentence As String) As String
    ' Zanka čez vse znake, da dobimo vsoto
    Dim Character As Char
    Dim Checksum As Integer
    For Each Character In sentence
        Select Case Character
            Case "$"c
                ' Ignoriraj znak za dolar
            Case "*"c
                ' Ustavi operacijo pred znakom zvezdica
                Exit For
            Case Else
                ' Je to prva vrednost kontrolne vsote
                If Checksum = 0 Then
                    ' Da. Nastavi kontrolno vsoto glede na to vrednost
                    Checksum = Convert.ToByte(Character)
                Else
                    ' Ne. Izvedi XOR nad kontrolno vsoto in vrednostjo znaka

```

```

        Checksum = Checksum Xor Convert.ToByte(Character)
    End If
End Select
Next
' Vrni kontrolno vsoto v šestnajstiškem formatu z dvema znakoma
Return Checksum.ToString("X2")
End Function
End Class

```

Slika 28: Koda z dodanim preverjanjem aktivne povezave v VB.NET programskem jeziku

Kot lahko vidimo je v en NMEA stavek stisnjeno kar nekaj informacij. Naš interpreter pa je lahko nadgrajen s podporo drugemu NMEA stavku: \$GPGSV (ang. GPS Satellites in view). Ta stavek opisuje konfiguracijo satelitov nad sprejemnikom v realnem času. Celotna koda tega stavka je predstavljena v »Sliki 29«.

```
$GPGSV,3,1,11,03,03,111,00,04,15,270,00,06,01,010,00,13,06,292,00*74
```

Slika 29: \$GPGSV stavek

Polja \$GPGSV stavka pomenijo sledeče:

- 1 – Skupno število sporočil tega tipa v tem ciklu
- 2 – Številka sporočila
- 3 – Skupno število vidnih satelitov
- 4 – Satelitova PRN koda
- 5 – Naklon v stopinjah, največ 90°
- 6 – Azimut, stopinje od severnega pola, vrednost od 000 do 359
- 7 – Razmerje signal-šum (ang. Signal-to-Noise ratio, krat. SNR), 00-99 dB
- 8-11 - Informacija o drugem vidnem satelitu, isto kot polja 4-7
- 12-15 - Informacija o tretjem vidnem satelitu, isto kot polja 4-7
- 16-19 - Informacija o četrtem vidnem satelitu, isto kot polja 4-7



Slika 30: Grafična predstavitev \$GPGSV stavka, kjer središče kroga predstavlja trenutno lokacijo, robovi pa predstavljajo obzorje

5.2.7 Spremljanje satelitov v realnem času

Poznavanje lokacije satelitov je zelo pomembno, ko ugotavljamo, kako točne so meritve in kako stabilno je naše določevanje pozicije sprejemnika. V poglavju 5.2.9 bo opisana točnost GPS, sedaj pa bo v interpreter dodana koda, ki spremlja lokacijo satelitov in moč signala, ki ga oddajajo. GPS sistem ima na nebu 24 satelitov, ki so razporejeni tako, da je z vsakega dela Zemlje možno spremljati vsaj 6 satelitov. Sateliti se neprestano premikajo, kar je dobro, saj s tem izničimo morebitne cone brez signala. Lokacija satelitov je podana kot azimut in kot kot glede na površje Zemlje. Če naprava torej pravi, da je azimut satelita 45° in da je kot glede na površje Zemlje 45° , to pomeni, da se satelit nahaja na pol poti proti obzorju v smeri severovzhod. Poleg lokacije pa vsak satelit oddaja tudi svojo PRN, ki se uporablja za identifikacijo posameznega satelita.

V celem \$GPGSV stavku je ena najboljših informacij razmerje signal/šum. Številka nam pove, kako dobro sprejemamo satelitov radijski signal. Vsi sateliti sicer oddajajo enako močan signal, a razne prepreke, kot so drevesa, stene itd., lahko signal popačijo preko meje razpoznavnosti. Tipična SNR vrednost je nekje med 0 in 50, kjer 50 pomeni odličen signal (številka je lahko visoka do 99, a tudi v jasnem vremenu in brez preprek le malokdaj pride nad 50). Na »Sliki 30« zelena satelita pomenita močan signal, rumeni nekaj srednjega, signal s sivega satelita pa je preveč popačen za relevantne meritve. V »Sliki 31« je v interpreter dodan del, ki bere te satelitske podatke.

```
'Jedro NMEA interpreterja
```

```
Public Class NmeaInterpreter
    ' Aktivira se kadar je trenutna lokacija spremenjena
    Public Event PositionReceived(ByVal latitude As String, _
        ByVal longitude As String)
    Public Event DateTimeChanged(ByVal dateTime As DateTime)
    Public Event BearingReceived(ByVal bearing As Double)
    Public Event SpeedReceived(ByVal speed As Double)
    Public Event SpeedLimitReached()
    Public Event FixObtained()
    Public Event FixLost()
    Public Event SatelliteReceived(ByVal pseudoRandomCode As Integer, _
        ByVal azimuth As Integer, _
        ByVal elevation As Integer, _
        ByVal signalToNoiseRatio As Integer)
    ' Procesira informacije z GPS sprejemnika
    Public Function Parse(ByVal sentence As String) As Boolean
        ' Zavrže stavek kadar se kontrolna vsota ne ujema z izračunano kontrolno vsoto
        If Not IsValid(sentence) Then Return False
        ' Pogleda prvo besedo, da izve kam mora iti naprej
        Select Case GetWords(sentence)(0)
            Case "$GPRMC"
                ' "Recommended Minimum" stavek je bil najden!
                Return ParseGPRMC(sentence)
            Case "$GPGSV"
                ' Stavek o vidnih satelitih je bil najden!
                Return ParseGPGSV(sentence)
        End Select
    End Function
End Class
```

```

    Case Else
        ' Stavek ni bil prepoznan
        Return False
    End Select
End Function
' Razdeli stavek na posamezne besede
Public Function GetWords(ByVal sentence As String) As String()
    Return sentence.Split(",")
End Function
' Interpretira $GPRMC sporočilo
Public Function ParseGPRMC(ByVal sentence As String) As Boolean
    ' Razdeli stavek na posamezne besede
    Dim Words() As String = GetWords(sentence)
    ' Imamo dovolj podatkov za opis lokacije?
    If Words(3) <> "" And Words(4) <> "" And Words(5) <> "" And _
        Words(6) <> "" Then
        ' Da. Razberi podatke o zemljepisni višini in širini
        Dim Latitude As String = Words(3).Substring(0, 2) & "°" ' Pripni ure
        Latitude = Latitude & Words(3).Substring(2) & "" ' Pripni minute
        Latitude = Latitude & Words(4) ' Pripni hemisfero
        Dim Longitude As String = Words(5).Substring(0, 3) & "°" ' Pripni ure
        Longitude = Longitude & Words(5).Substring(3) & "" ' Pripni minute
        Longitude = Longitude & Words(6) ' Pripni hemisfero
        ' Klicočo aplikacijo obvesti o spremembi
        RaiseEvent PositionReceived(Latitude, Longitude)
    End If
    ' Imamo dovolj vrednosti, da razčlenimo satelitov čas?
    If Words(1) <> "" Then
        ' Da. Izlušči ure, minute, sekunde in milisekunde
        Dim UtcHours As Integer = CType(Words(1).Substring(0, 2), Integer)
        Dim UtcMinutes As Integer = CType(Words(1).Substring(2, 2), Integer)
        Dim UtcSeconds As Integer = CType(Words(1).Substring(4, 2), Integer)
        Dim UtcMilliseconds As Integer
        ' Izlušči milisekunde, če so na voljo
        If Words(1).Length > 7 Then UtcMilliseconds = _
            CType(Single.Parse(Words(1).Substring(6), _
                CultureInfo.InvariantCulture) * 1000, Integer)
        ' Kreiraj DateTime objekt z vsemi vrednostmi
        Dim Today As DateTime = System.DateTime.Now.ToUniversalTime
        Dim SatelliteTime As New System.DateTime(Today.Year, Today.Month, _
            Today.Day, UtcHours, UtcMinutes, UtcSeconds, UtcMilliseconds)
        ' Obvesti o novem času prilagojenem lokalnemu času
        RaiseEvent DateTimeChanged(SatelliteTime.ToLocalTime)
    End If
    ' Imamo dovolj informacij, da izluščimo trenutno hitrost
    If Words(7) <> "" Then
        ' Da. Pretvori v MPH
        Dim Speed As Double = CType(Words(7), Double) * 1.150779
        ' Če smo hitrejši od 55MPH sproži hitrostno opozorilo
        If Speed > 55 Then RaiseEvent SpeedLimitReached()
        ' Obvesti o trenutni hitrosti
        RaiseEvent SpeedReceived(Speed)
    End If
    ' Imamo dovolj informacij da izluščimo smer?
    If Words(8) <> "" Then
        ' Obvesti, da je bil stavek prepoznan
        Dim Bearing As Double = CType(Words(8), Double)
        RaiseEvent BearingReceived(Bearing)
    End If
    ' Ima naprava »fix« s satelitom?
    If Words(2) <> "" Then
        Select Case Words(2)
            Case "A"
                RaiseEvent FixObtained()
            Case "V"
                RaiseEvent FixLost()
        End Select
    End If
End If

```

```

    ' Obvesti, da je bil stavek prepoznan
    Return True
End Function
' Interpretira $GPGSV NMEA stavek
Public Function ParseGPGSV(ByVal sentence As String) As Boolean
    Dim PseudoRandomCode As Integer
    Dim Azimuth As Integer
    Dim Elevation As Integer
    Dim SignalToNoiseRatio As Integer
    ' Razčleni stavek v besede
    Dim Words() As String = GetWords(sentence)
    ' Vsak stavek vsebuje štiri bloke podatkov s satelita
    ' Prebere vsak blok in javi informacijo glede vsakega satelita
    Dim Count As Integer
    For Count = 1 To 4
        ' Ima stavek dovolj besed za analizo?
        If (Words.Length - 1) >= (Count * 4 + 3) Then
            ' Da. Nadaljuj z analizo bloka. Vsebuje kakšno informacijo?
            If Words(Count * 4) <> "" And Words(Count * 4 + 1) <> ""
                And Words(Count * 4 + 2) <> "" And Words(Count * 4 + 3) <> "" Then
                ' Da. Izlušči informacije o satelitih in jih javi
                PseudoRandomCode = CType(Words(Count * 4), Integer)
                Elevation = CType(Words(Count * 4 + 1), Integer)
                Azimuth = CType(Words(Count * 4 + 2), Integer)
                SignalToNoiseRatio = CType(Words(Count * 4 + 3), Integer)
                ' Javi informacije tega satelita
                RaiseEvent SatelliteReceived(PseudoRandomCode, Azimuth, Elevation, _
                    SignalToNoiseRatio)
            End If
        End If
    Next
    ' Obvesti, da je bil stavek prepoznan
    Return True
End Function
' Vrne True če se kontrolna vsota stavka ujema z izračunano kontrolno vsoto
Public Function IsValid(ByVal sentence As String) As Boolean
    ' Primerja znake za zvezdico z izračunom
    Return sentence.Substring(sentence.IndexOf("*") + 1) = GetChecksum(sentence)
End Function
' Izračuna kontrolno vsoto za stavek
Public Function GetChecksum(ByVal sentence As String) As String
    ' Zanka čez vse znake, da dobimo vsoto
    Dim Character As Char
    Dim Checksum As Integer
    For Each Character In sentence
        Select Case Character
            Case "$"c
                ' Ignoriraj znak za dolar
            Case "*"c
                ' Ustavi operacijo pred znakom zvezdica
                Exit For
            Case Else
                ' Je to prva vrednost kontrolne vsote
                If Checksum = 0 Then
                    ' Da. Nastavi kontrolno vsoto glede na to vrednost
                    Checksum = Convert.ToByte(Character)
                Else
                    ' Ne. Izvedi XOR nad kontrolno vsoto in vrednostjo znaka
                    Checksum = Checksum Xor Convert.ToByte(Character)
                End If
            End Select
    Next
    ' Vrni kontrolno vsoto v šestnajstiškem formatu z dvema znakoma
    Return Checksum.ToString("X2")
End Function
End Class

```

Slika 31: Prebiranje \$GPGSV podatkov v VB.NET programskem jeziku

5.2.8 Interpreter, ki deluje po celem svetu.

Vsa zgornja koda je primerna le, če živimo na področju Združenih Držav Amerike. Vsi zapisi so zapisani v numeričnem formatu, kot ga uporabljajo ZDA. Evropske države in še veliko drugih uporabljajo drugačen format zapisa, zaradi česar moramo v naš interpreter dodati še kodo, ki bo te formate uredila. Na srečo .NET vključuje sistem za pretvorbo formata med različnimi državami, tako da je rešitev problema dokaj enostavna. V našem interpreterju je edino necelo število tisto, ki opisuje hitrost, tako da moramo spremeniti le to. NmeaCultureInfo spremenljivka predstavlja področje, na podlagi katerega se po potrebi konvertirajo števila iz NMEA stavkov. Double.Parse metoda se potem uporabi, da spremenimo hitrost v tak format, kot ga uporablja sprejemnik. Dopolnjeni interpreter na »Sliki 32« je sedaj pripravljen, da se ga uporablja po vsem svetu.

```
'Jedro NMEA interpreterja
Imports System.Globalization
Public Class NmeaInterpreter
    ' Predstavlja EN-US področje, ki se potem uporablja za številke v NMEA stavkih
    Private NmeaCultureInfo As New CultureInfo("en-US")
    ' Pretvarja vozle v MPH
    Private MPHPerKnot As Double = Double.Parse("1.150779", NmeaCultureInfo)
    ' Aktivira se kadar je trenutna lokacija spremenjena
    Public Event PositionReceived(ByVal latitude As String, _
        ByVal longitude As String)
    Public Event DateTimeChanged(ByVal dateTime As DateTime)
    Public Event BearingReceived(ByVal bearing As Double)
    Public Event SpeedReceived(ByVal speed As Double)
    Public Event SpeedLimitReached()
    Public Event FixObtained()
    Public Event FixLost()
    Public Event SatelliteReceived(ByVal pseudoRandomCode As Integer, _
        ByVal azimuth As Integer, _
        ByVal elevation As Integer, _
        ByVal signalToNoiseRatio As Integer)
    ' Procesira informacije z GPS sprejemnika
    Public Function Parse(ByVal sentence As String) As Boolean
        ' Zavrne stavek kadar se kontrolna vsota ne ujema z izračunano kontrolno vsoto
        If Not IsValid(sentence) Then Return False
        ' Pogleda prvo besedo, da izve kam mora iti naprej
        Select Case GetWords(sentence)(0)
            Case "$GPRMC" ' "Recommended Minimum" stavek je bil najden!
                Return ParseGPRMC(sentence)
            Case "$GPGSV" ' Stavek o vidnih satelitih je bil najden!
                Return ParseGPGSV(sentence)
            Case Else
                ' Stavek ni bil prepoznan
                Return False
        End Select
    End Function
    ' Razdeli stavek na posamezne besede
    Public Function GetWords(ByVal sentence As String) As String()
        Return sentence.Split(",")
    End Function
    ' Interpretira $GPRMC sporočilo
    Public Function ParseGPRMC(ByVal sentence As String) As Boolean
        ' Razdeli stavek na posamezne besede
        Dim Words() As String = GetWords(sentence)
```

```

' Imamo dovolj podatkov za opis lokacije?
If Words(3) <> "" And Words(4) <> "" _
And Words(5) <> "" And Words(6) <> "" Then
' Da. Razberi podatke o zemljepisni višini in širini
Dim Latitude As String = Words(3).Substring(0, 2) & "°" ' Pripni ure
Latitude = Latitude & Words(3).Substring(2) & "" ' Pripni minute
Latitude = Latitude & Words(4) ' Pripni hemisfero
Dim Longitude As String = Words(5).Substring(0, 3) & "°" ' Pripni ure
Longitude = Longitude & Words(5).Substring(3) & "" ' Pripni minute
Longitude = Longitude & Words(6) ' Pripni hemisfero
' Klicočo aplikacijo obvesti o spremembi
RaiseEvent PositionReceived(Latitude, Longitude)
End If
' Imamo dovolj vrednosti, da razčlenimo satelitov čas?
If Words(1) <> "" Then
' Da. Izlušči ure, minute, sekunde in milisekunde
Dim UtcHours As Integer = CType(Words(1).Substring(0, 2), Integer)
Dim UtcMinutes As Integer = CType(Words(1).Substring(2, 2), Integer)
Dim UtcSeconds As Integer = CType(Words(1).Substring(4, 2), Integer)
Dim UtcMilliseconds As Integer
' Izlušči milisekunde, če so na voljo
If Words(1).Length > 7 Then
UtcMilliseconds = CType(Single.Parse(Words(1).Substring(6), _
CultureInfo.InvariantCulture) * 1000, Integer)
End If
' Kreiraj DateTime objekt z vsemi vrednostmi
Dim Today As DateTime = System.DateTime.Now.ToUniversalTime
Dim SatelliteTime As New System.DateTime(Today.Year, Today.Month, _
Today.Day, UtcHours, UtcMinutes, UtcSeconds, UtcMilliseconds)
' Obvesti o novem času prilagojenem lokalnemu času
RaiseEvent DateTimeChanged(SatelliteTime.ToLocalTime)
End If
' Imamo dovolj informacij, da izluščimo trenutno hitrost
If Words(7) <> "" Then
' Da. Pretvori v MPH
Dim Speed As Double = Double.Parse(Words(7), NmeaCultureInfo) _
* MPHPerKnot
' Obvesti o trenutni hitrosti
RaiseEvent SpeedReceived(Speed)
' Če smo hitrejši od 55MPH sproži hitrostno opozorilo
If Speed > 55 Then RaiseEvent SpeedLimitReached()
End If
' Imamo dovolj informacij da izluščimo smer?
If Words(8) <> "" Then
' Obvesti, da je bil stavek prepoznan
Dim Bearing As Double = CType(Words(8), Double)
RaiseEvent BearingReceived(Bearing)
End If
'Ima naprava »fix« s satelitom?
If Words(2) <> "" Then
Select Case Words(2)
Case "A"
RaiseEvent FixObtained()
Case "V"
RaiseEvent FixLost()
End Select
End If
' Obvesti, da je bil stavek prepoznan
Return True
End Function
' Interpretira $GPGSV NMEA stavek
Public Function ParseGPGSV(ByVal sentence As String) As Boolean
Dim PseudoRandomCode As Integer
Dim Azimuth As Integer
Dim Elevation As Integer
Dim SignalToNoiseRatio As Integer
' Razčleni stavek v besede
Dim Words() As String = GetWords(sentence)

```

```

' Vsak stavek vsebuje štiri bloke podatkov s satelita
' Prebere vsak blok in javi informacijo glede vsakega satelita
Dim Count As Integer
For Count = 1 To 4
    ' Ima stavek dovolj besed za analizo?
    If (Words.Length - 1) >= (Count * 4 + 3) Then
        ' Da. Nadaljuj z analizo bloka. Vsebuje kakšno informacijo?
        If Words(Count * 4) <> "" And Words(Count * 4 + 1) <> ""
            And Words(Count * 4 + 2) <> "" And Words(Count * 4 + 3) <> "" Then
                ' Da. Izlušči informacije o satelitih in jih javi
                PseudoRandomCode = CType(Words(Count * 4), Integer)
                Elevation = CType(Words(Count * 4 + 1), Integer)
                Azimuth = CType(Words(Count * 4 + 2), Integer)
                SignalToNoiseRatio = CType(Words(Count * 4 + 3), Integer)
                ' Javi informacije tega satelita
                RaiseEvent SatelliteReceived(PseudoRandomCode, Azimuth, Elevation, _
                    SignalToNoiseRatio)
            End If
        End If
    End If
Next
' Obvesti, da je bil stavek prepoznan
Return True
End Function
' Vrne True če se kontrolna vsota stavka ujema z izračunano kontrolno vsoto
Public Function IsValid(ByVal sentence As String) As Boolean
    ' Primerja znake za zvezdico z izračunom
    Return sentence.Substring(sentence.IndexOf("*") + 1) = GetChecksum(sentence)
' Izračuna kontrolno vsoto za stavek
Public Function GetChecksum(ByVal sentence As String) As String
    ' Zanka čez vse znake, da dobimo vsoto
    Dim Character As Char
    Dim Checksum As Integer
    For Each Character In sentence
        Select Case Character
            Case "$"c
                ' Ignoriraj znak za dolar
            Case "*"c
                ' Ustavi operacijo pred znakom zvezdica
                Exit For
            Case Else
                ' Je to prva vrednost kontrolne vsote
                If Checksum = 0 Then
                    ' Da. Nastavi kontrolno vsoto glede na to vrednost
                    Checksum = Convert.ToByte(Character)
                Else
                    ' Ne. Izvedi XOR nad kontrolno vsoto in vrednostjo znaka
                    Checksum = Checksum Xor Convert.ToByte(Character)
                End If
            End Select
        End Select
    Next
    ' Vrni kontrolno vsoto v šestnajstiškem formatu z dvema znakoma
    Return Checksum.ToString("X2")
End Function
End Class

```

Slika 32: Urejanje lokalnega formata v VB.NET programskem jeziku

Z vso zgornjo kodo smo izdelali interpreter, ki uspešno prepozna NMEA stavke. Prav tako lahko z njim ugotovimo svojo lokacijo, sinhroniziramo interno uro, najdemo svoj cilj, pazimo na hitrost in ugotovimo, kje na nebu se nahaja določen satelit. A z vsem tem še ne dobimo naprave, ki bi jo lahko uporabljali za uspešno navigacijo, saj nam manjka ključna komponenta – točnost. GPS aplikacije so načrtovane tako, da javijo vse informacije, ki jih

najdejo, četudi so te informacije napačne. Izračun lokacije je lahko napačen tudi za več kot 100 metrov, pa čeprav uporabljamo DGPS ali WAAS korekcijske tehnologije. V nadaljevanju bomo zgornji interpreter dopolnili tako, da bomo dobili kar najboljšo točnost, primerno za uporabo v pogojih, kjer je le-ta ključnega pomena.

5.2.9 Ugotavljanje potrebne natančnosti

V poglavju o geometrijski postavitvi satelitov smo razložili vpliv napak in vrednosti, ki jih te napake lahko zavzamejo. Velja neka splošna ocena, da je DOP vrednost 6 ali manj povsem dovolj za avtomobilsko navigacijo. Toda ali je 6 res sprejemljivo in zakaj je temu tako? Splošna enačba je opisana s (4).

$$\text{Točnost GPS naprave} * \text{DOP} = \text{največja dovoljena napaka} \quad (4)$$

Enačba uporablja DOP kot faktor pri kalkulaciji največje možne napake, ki se odraža v merljivi količini. Če privzamemo, da je tipična napaka GPS naprave nekje od 5-7 metrov, torej povprečno 6, dobimo največjo dovoljeno napako po formuli: $6\text{m} * 6 = 36$ metrov. Glede na povprečne dolžine ulic je ta vrednost zadovoljiva, saj voznik lahko pravilno izbere svoj izvoz. Če bi imeli DOP enak 12, potem bi po formuli dobili $6\text{m} * 12 = 72$ metrov, kar pa ni več zadovoljiva natančnost, saj voznik lahko zgreši svoj izvoz. DOP je torej lahko tudi večji od 6, a ne preveč.

Trik pri tej formuli je, da je potrebno za vsako namembnost posebej poiskati, kakšna je največja možna napaka. Pri igri golfa recimo obstajajo GPS pripomočki, ki igralcu svetujejo, kakšno palico naj uporabi glede na njihovo lokacijo in pa oddaljenost od luknje. Raziskave so pokazale, da je razlika v dveh palicah ponavadi nekje od 10 pa do 15 metrov. Posledično ima lahko GPS pripomoček za igranje golfa največjo dovoljeno napako 10 metrov, da lahko z neko stopnjo suverenosti predlaga pravilno izbiro palice. Če uporabimo zgornjo formulo, izračunamo, da je lahko največji DOP enak 3. GPS pripomočki za igro golfa morajo tako imeti enkrat večjo natančnost od tistih v avtomobilski navigaciji.

Zakaj se sploh ubadamo s formulami in enostavno ne rečemo, da mora biti DOP enak 1? Zdi se kot idealna rešitev, a večja natančnost zahteva bolj jasen sprejem signala s satelitov. Avtomobilski GPS sprejemnik najbrž ne bo nikdar dobil DOP enak 1 (še posebej v središču mesta, kjer je signal moten zaradi stavb in drugih preprek), če pa je sprejemljiv DOP prevelik, pa bomo dobili preveč meritev, ki ne bodo točne, prav tako pa se bo navigacija

konstantno oglašala in predlagala spremembo smeri, tudi če voznik miruje. Aplikacije za golf lahko realno pričakujejo majhen DOP, saj operirajo na odprtih, zunanjih terenih, kjer sprejem signala načeloma ni moten, razen v primeru, ko žogica pristane nekeje v gozdu.

Če povzamemo, moramo za dobro GPS aplikacijo vedno upoštevati razmere, v katerih se bo uporabljala, in pa največji DOP, ki je še sprejemljiv. To nam bo omogočilo, da aplikacija odpravi večino problemov zaradi netočnosti in da deluje tudi v slabših pogojih.

5.2.10 Uveljavljanje točnosti

Sedaj, ko poznamo zahteve naše GPS aplikacije, si lahko ogledamo, kakšno kodo potrebujemo, da izvemo DOP vrednosti. Vse DOP vrednosti so zapisane v NMEA \$GPGSA čigar koda je prikazan v »Sliki 33«.

```
$GPGSA,A,3,11,29,07,08,5,17,24,,,,,,,,,2.3,1.2,2.0*30
```

Slika 33: \$GPGSA stavek

Izkušen programer lahko že na pogled prepozna, če so DOP vrednosti zadovoljive za natančno merjenje. Najboljši DOP je zagotovljen, kadar je v določanje lokacije vpletenih več satelitov, ki so enakomerno razporejeni po nebu in pod različnimi koti, tako da je GPS sprejemnik z vseh strani bombardiran s signali. Polja v \$GPGSA stavku sicer pomenijo sledeče:

- 1 – Način: M=ročni, A=Avtomatski
- 2 – Način: 1= Točna določitev ni možna, 2= 2D določitev, 3= 3D določitev
- 3-14 – ID številke vidnih satelitov (brez vsebine za neuporabljena polja)
- 15 – PDOP
- 16 – HDOP
- 17 - VDOP

Končna koda interpreterja na »Sliki 34« z vgrajeno metodo ParseGPGSA izlušči DOP vrednosti in jih javi preko HDOPReceived, VDOPReceived in PDOPReceived.

```
'Jedro NMEA interpreterja
using System;
using System.Globalization;
public class NmeaInterpreter
{
```

```

// Predstavlja EN-US področje, ki se potem uporablja za številke v NMEA stavkih
public static CultureInfo NmeaCultureInfo = new CultureInfo("en-US");
// Pretvarja vozle v MPH
public static double MPHPerKnot = double.Parse("1.150779",
    NmeaCultureInfo);
#region Delegates
    public delegate void PositionReceivedEventHandler(string latitude,
        string longitude);
    public delegate void DateTimeChangedEventHandler(System.DateTime dateTime);
    public delegate void BearingReceivedEventHandler(double bearing);
    public delegate void SpeedReceivedEventHandler(double speed);
    public delegate void SpeedLimitReachedEventHandler();
    public delegate void FixObtainedEventHandler();
    public delegate void FixLostEventHandler();
    public delegate void SatelliteReceivedEventHandler(
        int pseudoRandomCode, int azimuth, int elevation,
        int signalToNoiseRatio);
    public delegate void HDOPReceivedEventHandler(double value);
    public delegate void VDOPReceivedEventHandler(double value);
    public delegate void PDOPReceivedEventHandler(double value);
#endregion
#region Events
    public event PositionReceivedEventHandler PositionReceived;
    public event DateTimeChangedEventHandler DateTimeChanged;
    public event BearingReceivedEventHandler BearingReceived;
    public event SpeedReceivedEventHandler SpeedReceived;
    public event SpeedLimitReachedEventHandler SpeedLimitReached;
    public event FixObtainedEventHandler FixObtained;
    public event FixLostEventHandler FixLost;
    public event SatelliteReceivedEventHandler SatelliteReceived;
    public event HDOPReceivedEventHandler HDOPReceived;
    public event VDOPReceivedEventHandler VDOPReceived;
    public event PDOPReceivedEventHandler PDOPReceived;
#endregion
// Procesira informacije z GPS sprejemnika
public bool Parse(string sentence)
{
    // Zavrže stavek kadar se kontrolna vsota ne ujema z izračunano kontrolno vsoto
    if (!IsValid(sentence)) return false;
    // Pogleda prvo besedo, da izve kam mora iti naprej
    switch (GetWords(sentence)[0])
    {
        case "$GPRMC":
            // "Recommended Minimum" stavek je bil najden!
            return ParseGPRMC(sentence);
        case "$GPGSV":
            // Stavek o vidnih satelitih je bil najden!
            return ParseGPGSV(sentence);
        case "$GPGSA":
            return ParseGPGSA(sentence);
        default:
            // Stavek ni bil prepoznan
            return false;
    }
}

// Razdeli stavek na posamezne besede
public string[] GetWords(string sentence)
{
    return sentence.Split(',');
}
// Interpretira $GPRMC sporočilo
public bool ParseGPRMC(string sentence)
{
    // Razdeli stavek na posamezne besede
    string[] Words = GetWords(sentence);
    // Imamo dovolj podatkov za opis lokacije?
    if (Words[3] != "" && Words[4] != "" &&

```

```

Words[5] != "" && Words[6] != "")
{
    // Da. Izlušči podatke o zemljepisni višini in širini
    // Pripni ure
    string Latitude = Words[3].Substring(0, 2) + "°";
    // Pripni minute
    Latitude = Latitude + Words[3].Substring(2) + "\"";
    // Pripni ure
    Latitude = Latitude + Words[4]; // Pripni hemisfero
    string Longitude = Words[5].Substring(0, 3) + "°";
    // Pripni minute
    Longitude = Longitude + Words[5].Substring(3) + "\"";
    // Pripni hemisfero
    Longitude = Longitude + Words[6];
    // Klicočo aplikacijo obvesti o spremembi
    if(PositionReceived != null)
        PositionReceived(Latitude, Longitude);
}
// Imamo dovolj vrednosti, da razčlenimo satelitov čas?
if (Words[1] != "")
{
    // Da. Izlušči ure, minute, sekunde in milisekunde
    int UtcHours = Convert.ToInt32(Words[1].Substring(0, 2));
    int UtcMinutes = Convert.ToInt32(Words[1].Substring(2, 2));
    int UtcSeconds = Convert.ToInt32(Words[1].Substring(4, 2));
    int UtcMilliseconds = 0;
    // Izlušči milisekunde, če so na voljo
    if (Words[1].Length > 7)
    {
        UtcMilliseconds = Convert.ToInt32(
            float.Parse(Words[1].Substring(6), NmeaCultureInfo) * 1000);
    }
    // Kreiraj DateTime objekt z vsemi vrednostmi
    System.DateTime Today = System.DateTime.Now.ToUniversalTime();
    System.DateTime SatelliteTime = new System.DateTime(Today.Year,
        Today.Month, Today.Day, UtcHours, UtcMinutes, UtcSeconds,
        UtcMilliseconds);
    // Obvesti o novem času prilagojenem lokalnemu času
    if(DateTimeChanged != null)
        DateTimeChanged(SatelliteTime.ToLocalTime());
}
// Imamo dovolj informacij, da izluščimo trenutno hitrost
if (Words[7] != "")
{
    // Da. Pretvori v MPH
    double Speed = double.Parse(Words[7], NmeaCultureInfo) *
        MPHPerKnot;
    // Obvesti o trenutni hitrosti
    if(SpeedReceived != null)
        SpeedReceived(Speed);
    // Če smo hitrejši od 55MPH sproži hitrostno opozorilo
    if (Speed > 55)
        if(SpeedLimitReached != null)
            SpeedLimitReached();
}
// Imamo dovolj informacij da izluščimo smer?
if (Words[8] != "")
{
    // Obvesti, da je bil stavek prepoznani
    double Bearing = double.Parse(Words[8], NmeaCultureInfo);
    if(BearingReceived != null)
        BearingReceived(Bearing);
}
// Ima naprava »fix« s satelitom?
if (Words[2] != "")
{
    switch (Words[2])

```

```

    {
        case "A":
            if(FixObtained != null)
                FixObtained();
            break;
        case "V":
            if(FixLost != null)
                FixLost();
            break;
    }
}
// Obvesti, da je bil stavek prepoznan
return true;
}

// Interpretira $GPGSV NMEA stavek
public bool ParseGPGSV(string sentence)
{
    int PseudoRandomCode = 0;
    int Azimuth = 0;
    int Elevation = 0;
    int SignalToNoiseRatio = 0;
    // Razčleni stavek v besede
    string[] Words = GetWords(sentence);
    // Vsak stavek vsebuje štiri bloke podatkov s satelita
    // Prebere vsak blok in javi informacijo glede vsakega satelita
    int Count = 0;
    for (Count = 1; Count <= 4; Count++)
    {
        // Ima stavek dovolj besed za analizo?
        if ((Words.Length - 1) >= (Count * 4 + 3))
        {
            // Da. Nadaljuj z analizo bloka. Vsebuje kakšno informacijo?
            if (Words[Count * 4] != "" && Words[Count * 4 + 1] != ""

                && Words[Count * 4 + 2] != "" && Words[Count * 4 + 3] != "")
            {
                // Da. Izlušči informacije o satelitih in jih javi
                PseudoRandomCode = System.Convert.ToInt32(Words[Count * 4]);
                Elevation = Convert.ToInt32(Words[Count * 4 + 1]);
                Azimuth = Convert.ToInt32(Words[Count * 4 + 2]);
                SignalToNoiseRatio = Convert.ToInt32(Words[Count * 4 + 3]);
                // Javi informacije tega satelita
                if(SatelliteReceived != null)
                    SatelliteReceived(PseudoRandomCode, Azimuth,
                        Elevation, SignalToNoiseRatio);
            }
        }
    }
}
// Obvesti, da je bil stavek prepoznan
return true;
}

public bool ParseGPGSA(string sentence)
{
    // Stavek razdeli v besede
    string[] Words = GetWords(sentence);
    // Update the DOP values
    if (Words[15] != "")
    {
        if(PDOPReceived != null)
            PDOPReceived(double.Parse(Words[15], NmeaCultureInfo));
    }
    if (Words[16] != "")
    {
        if(HDOPReceived != null)
            HDOPReceived(double.Parse(Words[16], NmeaCultureInfo));
    }
}

```

```

    if (Words[17] != "")
    {
        if (VDOPReceived != null)
            VDOPReceived(double.Parse(Words[17], NmeaCultureInfo));
    }
    return true;
}

// Vrne True če se kontrolna vsota stavka ujema z izračunano kontrolno vsoto
public bool IsValid(string sentence)
{
    // Primerja znake za zvezdico z izračunom
    return sentence.Substring(sentence.IndexOf("*") + 1) ==
        GetChecksum(sentence);
}

// Izračuna kontrolno vsoto za stavek
public string GetChecksum(string sentence)
{
    // Zanka čez vse znake, da dobimo kontrolno vsoto
    int Checksum = 0;
    foreach (char Character in sentence)
    {
        if (Character == '$')
        {
            // Ignoriraj znak za dolar
        }
        else if (Character == '*')
        {
            // Nehaj z izvajanjem pred zvezdico
            break;
        }
        else
        {
            // Je to prva vrednost kontrolne vsote
            if (Checksum == 0)
            {
                // Da. Nastavi kontrolno vsoto glede na to vrednost
                Checksum = Convert.ToByte(Character);
            }
            else
            {
                // Ne. Izvedi XOR nad kontrolno vsoto in vrednostjo znaka
                Checksum = Checksum ^ Convert.ToByte(Character);
            }
        }
    }
    // Vrni kontrolno vsoto v šestnajstiškem formatu z dvema znakoma
    return Checksum.ToString("X2");
}
}

```

Slika 34: Končna koda interpreterja, ki javlja tudi VDOP, HDOP in PDOP vrednosti, spisana v C# programskem jeziku.

5.3 Razvoj GIS in kartografske aplikacije za .NET

V prejšnjem poglavju sem opisal, kako razvijemo osnovno GPS aplikacijo, kateri pa lahko po potrebi dodamo še cel kup funkcionalnosti glede na to, v kakšnem sistemu bo aplikacija uporabljena. V tem poglavju si bomo ogledali, kako s pomočjo GIS.NET kartografske komponente in Geodesy.NET koordinatne knjižnice generiramo svoje karte.

5.3.1 Geografski, projekcijski in točkovni koordinatni sistem

Da bi lahko razumeli tehnologijo, ki je vpletena v kartografijo, moramo poznati tri koordinatne sisteme: geografskega, projekcijskega in točkovnega. Vsak sistem igra pomembno vlogo pri prikazovanju kart, tekoča transformacija iz enega v drugega pa je ključnega pomena. Največkrat začnemo kreirati karte tako, da uporabimo okroglo sliko Zemlje, kjer veljajo meritve v obliki geografske višine in širine. Nato to sferično obliko Zemlje pretvorimo v ravno karto, ki služi kot projekcijski sistem, pri katerem veljata X in Y koordinati. V zadnjem koraku uredimo velikost in razmerja posameznih delov zemljevida, da se ujemajo z našo najbolj natančno točkovno karto. Pri tej uporabljamo točkovne koordinate. Celotni potek je prikazan na »Sliki 35«.



Slika 35: Geografski, projekcijski in točkovni koordinatni sistem

5.3.2 Ravna Zemlja je boljša za interpretacijo

V prejšnjih poglavjih smo opisali, kako GPS naprava izračuna svojo lokacijo kot geografsko višino in dolžino. Ta par informacij imenujemo kar geografske koordinate. Težava v teh geografskih koordinatah pa je v tem, da predstavljajo koordinate na Zemljinem površju, ki pa je sferične oblike. Ker so računalniški zasloni in zasloni GPS navigacij ravni, moramo to sferično Zemljo nekako raztegniti v dvodimenzionalno obliko, preden jo prikažemo na zaslonu. Ta tehnika se imenuje projekcija in je ključna pri vsaki kartografiji. Poznamo več kot 100 različnih tipov projekcij, vsaka pa se uporablja za drug spekter opravil. Mercator

projekcija se recimo uporablja v pomorstvu, saj uporablja karto, na kateri so črte z nespremenjeno smerjo kar ravne črte, kar posledično zelo olajša navigacijo. Neželena posledica tega pa je, da se popači velikost vsega, ko se približujemo severnemu ali južnemu polu, kar karto naredi neuporabno za operacije blizu teh dveh točk. Na »Sliki 36« je vidna razlika med Mercator in Polyconic projekcijo Zemlje.



Slika 36: Mercator projekcija levo in Polyconic projekcija desno

Različne projekcije lahko privedejo do popolnoma različne karte glede na osnovno, kljub temu pa so vse projekcije točne. Ena od projekcij, ki se imenuje ortografska projekcija, je še vedno ravna, a daje občutek 3D slike na računalniškem zaslonu. Ta postopek se velikokrat uporablja v raznih 3D aplikacijah, vključno s 3D igrami. Ne glede na velikost in obliko pa projekcijske koordinate zravnajo 3D koordinate, kar občutno olajša nalogo kartografiranja.

5.3.3 Matematične enačbe projekcij

Za vsako projekcijo stoji matematična enačba, ki je lahko bolj ali manj zapletena. Van der Grinten projekcija recimo za izračun X koordinate uporablja (5), za Y koordinato pa (6) [21].

$$X = \frac{\pm\pi(A(G - P^2) + \sqrt{A^2(G - P^2) - (P^2 + A^2)(G^2 - P^2)}}{P^2 + A^2} \quad (5)$$

$$Y = \frac{\pm\pi(PQ - A\sqrt{A^2 + 1}(P^2 + A^2) - Q^2}{P^2 + A^2} \quad (6)$$

V programu, ki ga bomo spisali v nadaljevanju, je uporabljena koda projekcije Plate Caree zaradi hitrosti in enostavnosti. To je tudi privzeta projekcija v GIS.NET 3.0 komponenti, ki vsebuje 25 različnih projekcij. Enačbi, ki opisujeta X in Y koordinate sta opisani v (7) in (8) [21].

$$X = \lambda \cos(\Phi_1) \quad (7)$$

$$Y = \Phi \quad (8)$$

Z enačbami za projekcijo kart je enostavneje delati, kadar so geografske koordinate predstavljene z radiani. Z njimi je enostavno računati, pripišemo pa jih lahko tako zemljepisni širini kot višini. Pretvarjanje iz stopinj v radiane in obratno je prikazana na »Slika 37«.

```
// 90 stopinj
double degrees = 90.0;
// Pretvori v radiane
double radians = degrees * (Math.Pi / 180.0);
// Pretvori radiane nazaj v stopinje
degrees = radians / (Math.Pi / 180.0);
```

Slika 37: Pretvarjanje iz stopinj v radiane ter obratno v C# programskem jeziku

Vsi programi, ki opravljajo projekcijo, so sestavljeni iz dveh metod. Prva opisuje, kako iz geografskih koordinat dobimo projekcijske koordinate, druga pa je ravno obratna, saj aplikaciji pove, kako iz projekcijskih koordinat dobimo geografske. V praksi to pomeni, da lahko uporabnik poljubno povečuje zemljevid in ima v vsakem povečevalnem ali pomanjševalnem koraku točno informacijo in pravilne koordinacijske podatke. Postopek je prikazan na »Sliki 38«.

```
using System.Drawing;
public class PlateCaree
{
    public PointF Project(PointF geographicCoordinate)
    {
        // Prvo pretvori geografske koordinate v radiane
        double radianX = geographicCoordinate.X * (Math.PI / 180);
        double radianY = geographicCoordinate.Y * (Math.PI / 180);
        // Ustvari nov razred
        PointF result = new PointF();
        // Izračunaj projekcijsko X koordinato
        result.X = (float)(radianX * Math.Cos(0));
        // Izračunaj projekcijsko Y koordinato
        result.Y = (float)radianY;
        // Vrni rezultat
        return result;
    }

    public PointF Deproject(PointF projectedCoordinate)
    {
        // Ustvari nov razred za shranjevanje rezultata
        PointF result = new PointF();
    }
}
```

```

// Izračunaj geografsko X koordinato (zemljepisna dolžina)
result.X = (float)(projectedCoordinate.X / Math.Cos(0) /
              (Math.PI / 180.0));
// Izračunaj geografsko Y koordinato (zemljepisna širina)
result.Y = (float)(projectedCoordinate.Y / (Math.PI / 180.0));
return result;
}
}

```

Slika 38: Pretvarjanje iz geografskih v projekcijske koordinate in obratno v programskem jeziku C#

S tem razredom lahko sedaj kreiramo projekcijske koordinate vsake lokacije na Zemlji, kot je vidno na »Sliki 39«.

```

// Definiraj geografske koordinate, v tem primeru GPS lokacijo
PointF myLocation = new PointF();
myLocation.Y = 39.0; // 39 Severno
myLocation.X = -105.0; // 105 Zahodno
// Sedaj uporabimo projekcijski razred da »izravnamo« te koordinate
PlateCaree projection = new PlateCaree();
PointF myProjectedLocation = projection.Project(myLocation);
myLocation = projection.Deproject(myProjectedLocation);

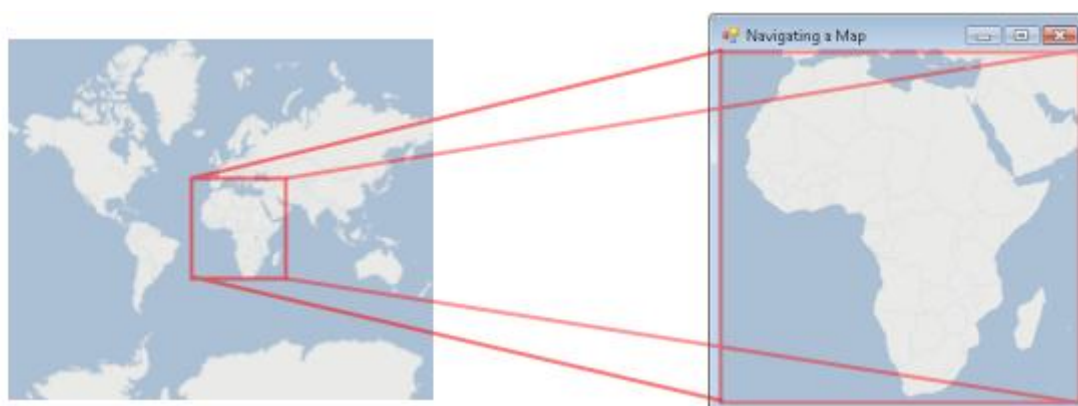
```

Slika 39: Projekcija dveh geografskih koordinat v C# programskem jeziku

Zgornji postopek je ponovljen za vse geografske koordinate, dokler vseh podatkov ne predstavimo s projekcijskimi. Ko naredimo to, nam manjka le še en korak do točkovnih koordinat, ki so lahko prikazane na zaslonu.

5.3.4 Aktivni del karte

Risanje zemljevida na steno je enostavno opravilo, saj ga tja narišemo z vsemi dostopnimi podatki. Navigacijska kartografija pa ni tako enostavna, saj mora biti uporabnikom dana možnost, da karte obračajo in jih povečujejo (celotne ali pa samo določen del). Za doseg tega si moramo predstavljati pravokotnik, ki predstavlja del karte, ki jo hočemo videti bolj podrobno. kot to kaže »Slika 40«. Ko to naredimo, moramo z nekaj matematike spremeniti projekcijske koordinate v točkovne koordinate.



Slika 40: Aktivni del karte pri pretvorbi iz projekcijskih koordinat v točkovne

.NET razvijalci so povečini že seznanjeni s točkovnimi koordinatami. To so iste koordinate, kot jih potrebujemo, da postavimo stvari na *Form*. Kar rabimo, je postopek, s katerim projekcijske koordinate pretvorimo v točkovne. Postopek je tak, da projekcijske koordinate povečamo ali pomanjšamo in pretvorimo, da se novi pravokotnik z detajlno vsebino poravnava z velikostjo točke v *Form*. Pretvarjanje se izvede tako, da zgornjemu levemu kotu pravokotnika dodamo negativno vrednost X in pozitivno vrednost Y koordinate. Horizontalno skalo izračunamo tako, da širino ploskve, na katero bomo izrisovali, delimo s predvideno širino pravokotnika. Enako velja tudi za vertikalno skalo.

Na srečo je pri namiznih računalnikih mogoča uporaba razreda *Matrix*, ki večino tega dela lahko opravi namesto nas. Objekte tipa *Matrix* lahko obračamo, povečujemo in pomanjšujemo. Koda tega dela je prikazana na »Sliki 41«.

```
// Uporabimo razred Matrix za uporabo povečav in obračanja
Matrix transform = new Matrix();
// Najprej pretvorimo projekcijske točke, da se ujemajo s točkami 0,0
transform.Translate(-viewport.X, viewport.Y, MatrixOrder.Append);
// Nato uredimo skalo za vse točke, da pravokotnik z detajli ustreza velikosti form
transform.Scale(this.Width / viewport.Width,
               this.Height / -viewport.Height, MatrixOrder.Append);
```

Slika 41: Razred *Matrix* in njegova uporaba v C# programskem jeziku

Če razvijamo aplikacijo v C/C++/C# programskem jeziku in bo aplikacija bazirala na Windows operacijskem sistemu (avtomobilske GPS navigacijske naprave velikokrat uporabljajo Windows NT), se lahko poslužujemo GDI+ API-ja. V spodnji kodi uporabimo razred *Graphics*, ki se tipično uporablja pri metodi *onPaint*. Na srečo lahko transformacijo in skalo z lahkoto določimo z lastnostjo *Transform* našega *Graphics* objekta. Ko je to opravljeno lahko s projekcijskimi koordinatami in metodo *DrawLine* narišemo in pobarvamo naš objekt oziroma detajlni del zemljevida, »Slika 42«.

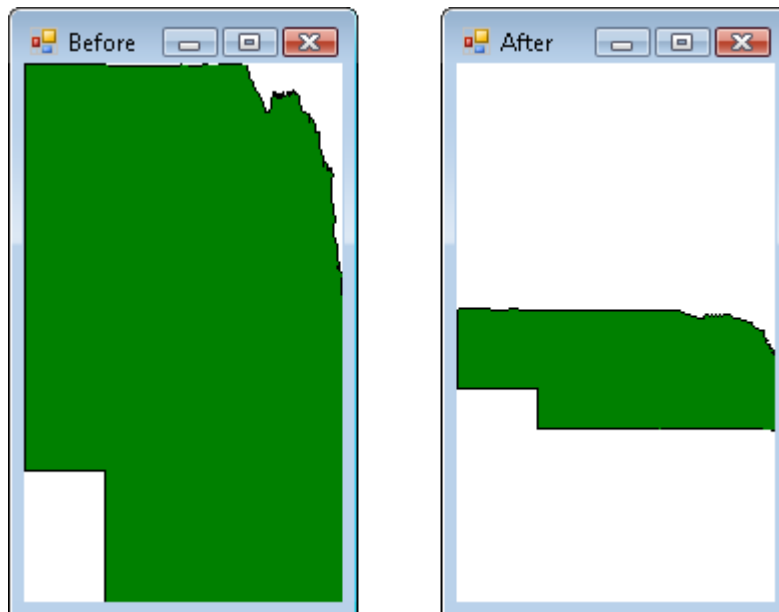
```
protected override void OnPaint(PaintEventArgs e)
{
    // Uporabi to transformacijo za vse grafične opreacije
    e.Graphics.Transform = transform;
    // Nariši objekt s črno obrobo in zelenim polnilom
    e.Graphics.FillPolygon(Brushes.Green, projectedCoordinates);
    e.Graphics.DrawPolygon(Pens.Black, projectedCoordinates);
}
```

Slika 42: Obroba objekta in polnilo v C# programskem jeziku

5.3.5 Razmerje objektov

V tem delu o kreiranju kartografije se srečujemo z dvema pravokotnikoma. Eden predstavlja celotni *Form*, na katerem bo vse predstavljano, drugi pa samo območje, ki bi ga radi

pobarvali. Če se območje, ki bi ga radi pobarvali, po obliki močno razlikuje od *Form*, moramo obliki med seboj uskladiti, da ne pride do napačnih rezultatov.



Slika 43: Objekt brez korekcije (levo) in objekt s korekcijo razmerja (desno)

Razmerje se izračuna z deljenjem širine pravokotnika z njegovo višino. Če bi imeli na primer širino 10 in višino 5 točk, bi to pomenilo, da je razmerje 0,5. Da prilagodimo razmerje objektov moramo primerjati razmerje objekta, ki ga želimo pobarvati z razmerjem celotnega *Form*. Če ima objekt večje razmerje kot *Form*, potem se objektu zmanjša višina, v nasprotnem primeru pa se mu poveča širina. Primer je prikazan na »Sliki 43«, koda pa na »Sliki 44«.

```
//Izračunaj razmerje Form
float pixelAspectRatio = (float)this.Width / this.Height;
//Izračunaj razmerje objekta
float projectedAspectRatio = viewport.Width / viewport.Height;
//Kopiraj objekta, ki jo lahko spreminjamo
RectangleF adjustedViewport = viewport;
//Je razmerje Form večje od razmerja objekta?
if (pixelAspectRatio > projectedAspectRatio)
{
    // Da. Povečaj širino objekta
    adjustedViewport.Inflate(
        (pixelAspectRatio * adjustedViewport.Height - adjustedViewport.Width)
        / 2,
        0);
}
// Je razmerje objekta večje od razmerja Form
else if (pixelAspectRatio < projectedAspectRatio)
{
    // Da. Povečaj višino objekta
    adjustedViewport.Inflate(
        0,
        (adjustedViewport.Width / pixelAspectRatio - adjustedViewport.Height)
        / 2);
}
```

}

Slika 44: Prilagajanje razmerja objekta v C# programskem jeziku

S tem, ko reguliramo razmerja, bodo vsi geografski objekti ohranili svojo velikost in obliko, četudi se bosta velikost in oblika *Form* spreminjali.

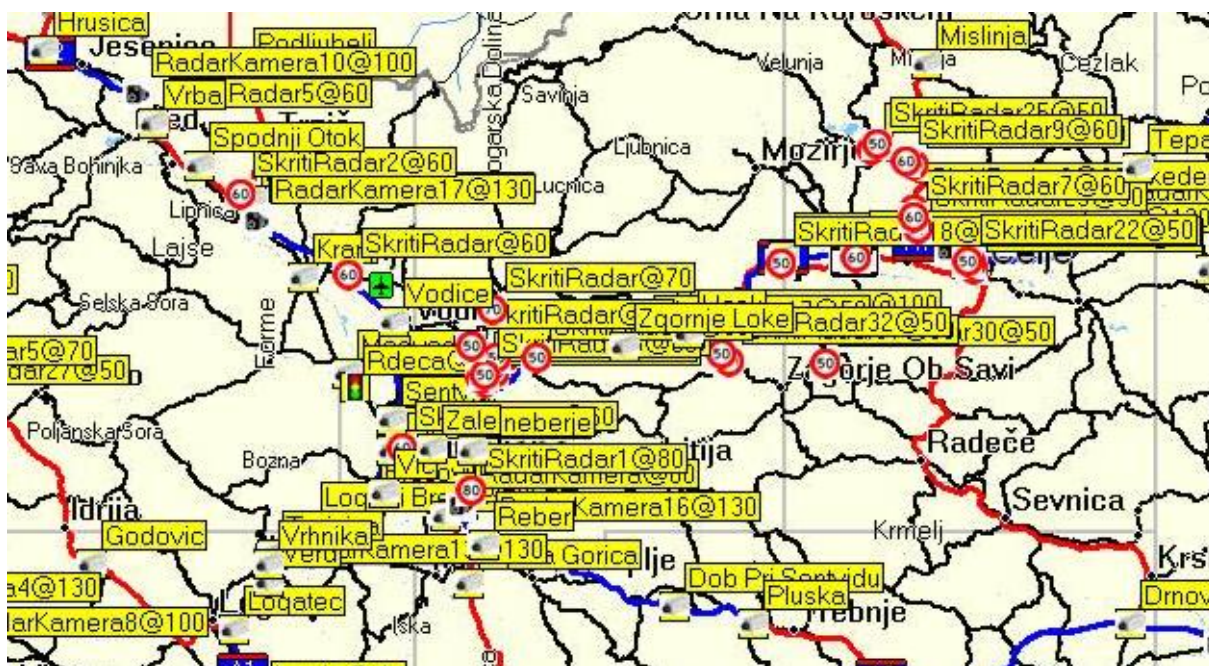
5.3.6 GIS

V poglavju 5.3 so pokazani osnovni gradniki pri kreiranju lastne kartografije. Za bolj podrobni prikaz pa moramo vedeti, za kaj se bo ta naša kartografija uporabljala in katere vse podatke za tako implementacijo potrebujemo. Ker bi bilo merjenje za vsako tako karto prezahtevno, predrago in predolgotrajno, se lahko poslužujemo geografskih informacijskih sistemov (GIS), ki so že v uporabi. Teh sistemov je ogromno, obstajajo pa za najrazličnejše možne namene. Tako poznamo GIS, ki pokaže, kje vse so bili izmerjeni potresi in kako močni so bili, sisteme, ki nam pokažejo gostoto prebivalstva v določenem območju, in tako dalje. Kar pa je pomembno za navigacijske naprave, pa je to, da poznamo geografske informacijske sisteme, ki za določeno področje prikažejo cestne in železniške povezave, pomorske poti in čeri, oziroma pokažejo informacije, povezane z aeronavtiko. Kot rečeno, se glede na potrebe naše aplikacije odločimo za enega (ali več) od teh in ga/jih uporabimo v našem programu.

Da bi zgornji program o navigacijski kartografiji dokončali do funkcionalnosti, ki jo zahteva cestna navigacija, bi morali spisati še kar nekaj kode, večina pa bi se nanašala na upoštevanje podatkov iz GIS. Tako bi morali podatke iz GIS vnesti na našo zgornjo karto in z izračunavanjem lokacije sprejemnika in primerjanjem s karto ugotoviti, po kateri poti se moramo peljati, da pridemo najhitreje do cilja. Bolj zapleten algoritem bi prikazal tudi potrebni čas, da pridemo do te lokacije in porabo goriva, znesek cestnin, upošteval hitrostne omejitve po odsekih in nam predlagal alternativno pot glede na razdaljo, potrebni čas vožnje ali pa glede na stroške, ki jih bomo imeli s to potjo. Praviloma obstaja 5 korakov, ki se jih moramo držati, ko se ukvarjamo z GIS za satelitsko navigacijo [14]:

1. GIS analizo začnimo s tem, da dobro postavimo vprašanje, ki nas zanima:
 - a. Katere ceste so na določenem območju?
 - b. Katere hitrostne omejitve veljajo na teh cestah?
 - c. Koliko pasov imajo te ceste?
 - d. Katere zanimive točke so v bližini teh cest (ang. Points Of Interest, krat. POI)?
 - e. Ali so na teh cestah stacionarni radarji in kje so?
 - f. Drugo.

2. Izmed vseh GIS na voljo izberemo tistega, lahko pa tudi več njih, ki vsebujejo za našo nalogo pomembne podatke.
3. Izvedemo analizo vseh izbranih sistemov, s katero določimo, kateri podatki so relevantni in za nas uporabni, kateri pa neuporabni iz takšnega ali drugačnega razloga.
4. Podatke obdelamo. V primeru razvoja navigacijske karte zabeležimo vse POI. Prav tako zabeležimo vse ceste, nadmorske višine, hitrostne omejitve, varnostne in hitrostne kamere. Vsem tem stvarim pa moramo seveda dodati tudi ustrezne koordinate, s katerimi bo potem naš sprejemnik primerjal svojo lokacijo.
5. Ko je tudi to narejeno, pogledamo končni izdelek in ga testiramo, da vidimo, če res vsebuje vse podatke, ki so za navigacijo potrebni, če je pokritost cestnega omrežja zadovoljiva itd. Končni izdelek v primeru satelitske navigacije vsebuje tudi podatke, ki so prikazani na »Sliki 45«.



Slika 45: navigacijska karta z vpisanimi cestami in radarji. Pri uporabi povečave bi se seveda izrisale še ostale ceste

5.4 Praktični primer z uporabo Arduina

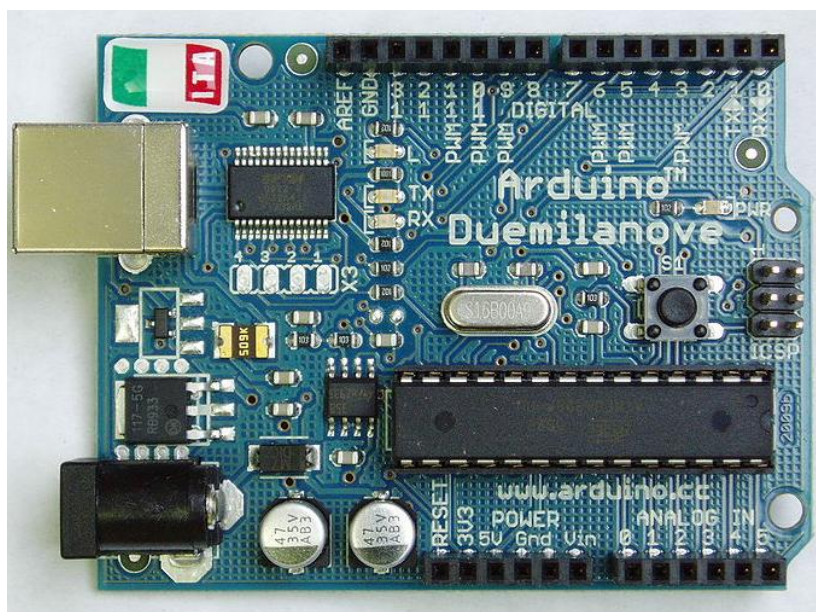
Načrtovanja GPS sistemov se lahko lotimo na mnogo načinov. Že izbira primernege mikrokontrolerja in ostale strojne opreme nam lahko vzame precej časa, saj se je ponudba na tem področju v zadnjem času močno povečala. Tako lahko izbiramo med rešitvami, ki jih ponujajo Motorola, PIC, Arduino in še kopica drugih. Tudi anten in ostalih potrebnih komponent je na trgu veliko, glede na zahteve projekta pa mora vsak sam preučiti, kaj točno potrebuje, da bo naprava delovala kar se da dobro.

Ker sem v svojem študijskem obdobju pa tudi v prostem času preizkusil vse tri zgoraj naštetih vgrajene sisteme, lahko rečem, da je še najlažji za razumevanje in razlago prav Arduino. Primer praktičnega izdelka v nadaljevanju je tako izdelan prav z njim.

5.4.1 Arduino

Arduino je vgrajeni sistem, ki bazira na enostavnem, odprtem strojnem načrtu z vgrajeno podporo vhodno-izhodnim napravam in standardnim programskim jezikom. Programski jezik Arduina je zasnovan na *Wiring* programskem jeziku in je zelo podoben C/C++. Cilj Arduina je približati vgrajene sisteme umetnikom in domačim uporabnikom po dostopni ceni, z relativno enostavnim programskim jezikom in s praktično neomejenimi možnostmi razširitev.

Z Arduinom, »Slika 46«, lahko izdelamo samostojne, interaktivne rešitve, lahko pa ga uporabljamo v povezavi z računalnikom. Tipična plošča Arduina vsebuje mikrokontroler Atmel AVR, nekaj vhodno-izhodnih povezav in serijski ali USB priključek za povezovanje z računalnikom [10]. Ta serijska ali USB povezava se uporablja bodisi za programiranje samega mikrokontrolerja bodisi za interakcijo z Arduinom v realnem času. Arduino je lahko povezan tudi v omrežje z drugimi Arduini, pri čemer pa po navadi med njih vpeljemo računalnik, ki skrbi za vso komunikacijo. Interakcija z računalnikom je enostavna, saj je lahko programska oprema na računalniku spisana v množici programskih jezikov. Najbolj popularna rešitev je Processing, možno pa je pisati tudi v Javi in drugih.



Slika 46: Arduino Duemilanove verzija 2009b. Na levi strani se nahajata USB in napajalni priključek, na spodnji strani analogni in na zgornji strani digitalni priključki.

5.4.2 EM-406 GPS modul

Drugi pomembnejši gradnik sta GPS modul in antena. Kaj nekaj karakteristik je, ki jih moramo upoštevati, nekatere seveda bolj, druge manj, odvisno od potreb in razpoložljivih sredstev. Prva pomembna karakteristika je čas, ki ga modul potrebuje, da pridobi točno lokacijo. Ta točna lokacija pomeni, da modul pridobiva GPS signal z dovolj satelitov, da izračuna svoj položaj. Druga pomembna stvar je, kako točen je modul pri izračunavanju položaja in kako dobro ohranja zagotavljanje točne lokacije v prisotnosti prepek, kot so recimo stavbe v naseljih ali pa gosti gozdovi.

Te karakteristike pa so odvisne od raznih tehnoloških razlik. Število satelitov, ki jih modul lahko istočasno sprejema, je pomemben podatek. Današnji GPS moduli sprejemajo do 20 kanalov, kar se odraža v hitrejšem iskanju satelitov in boljšem izračunu položaja v urbanem okolju. Procesiranje GPS signala v samem modulu je tudi pomemben faktor, saj posledično lahko uporabimo počasnejši mikrokontroler. Novejša vezna čipovja, kot je recimo SiRF III, omogočajo bistveno hitrejša časa pridobivanja signala in močno izboljšano robustnost pri držanju signala v gosto poseljenih območjih kot SiRF II ali starejša. Tudi podpora WAAS in ostalim izboljšavam GPS je pomemben faktor, saj naprava tako lahko položaj izračunava še bolj točno.

Komercialni GPS moduli za projekte, kot je ta, imajo dobro lastnost, da so vsi kompatibilni z NMEA 0831 formatom podatkov, kar pomeni, da lahko vsako sekundo izpišejo

standardno NMEA sporočilo, ki vsebuje hitrost, 3D položaj, število vidnih satelitov, GPS čas in drugo. NMEA se pošilja v ASCII formatu (ang. American Standard Code for Information Interchange) s hitrostjo 4800 baud, 8N1 (8 podatkovnih bitov, nobenega paritetnega, 1 stop bit).

Karakteristike EM-406 SiRF III modula, »Slika47«, z vgrajeno anteno so [10]:

- 20 kanalni sprejemnik
- Zelo občutljiva antena: -159dBm
- Točnost lociranja na 10 metrov oz. 5 metrov z uporabo WAAS
- Mrzli zagon (modul nima shranjenih satelitov): 42 sekund
- Poraba 70mA pri 4,5-6,5V



Slika 47: EM-406 GPS modul. Dimenzije so 30x30x10,5mm. Teža znaša 16 gramov. Odlična izbira za mobilne naprave.

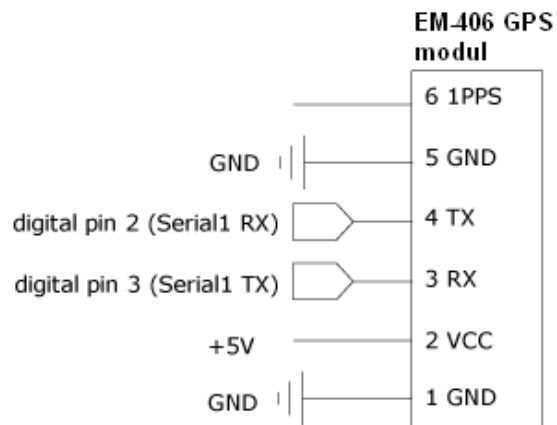
5.4.3 Določanje grobega položaja sprejemnika

V spodnjem primeru je prikazan postopek grobe določitve signala. Za lažjo predstavbo je na »Sliki 49« podana tudi shema vezave in na »Sliki 48« shema izhodov EM-406 modula.

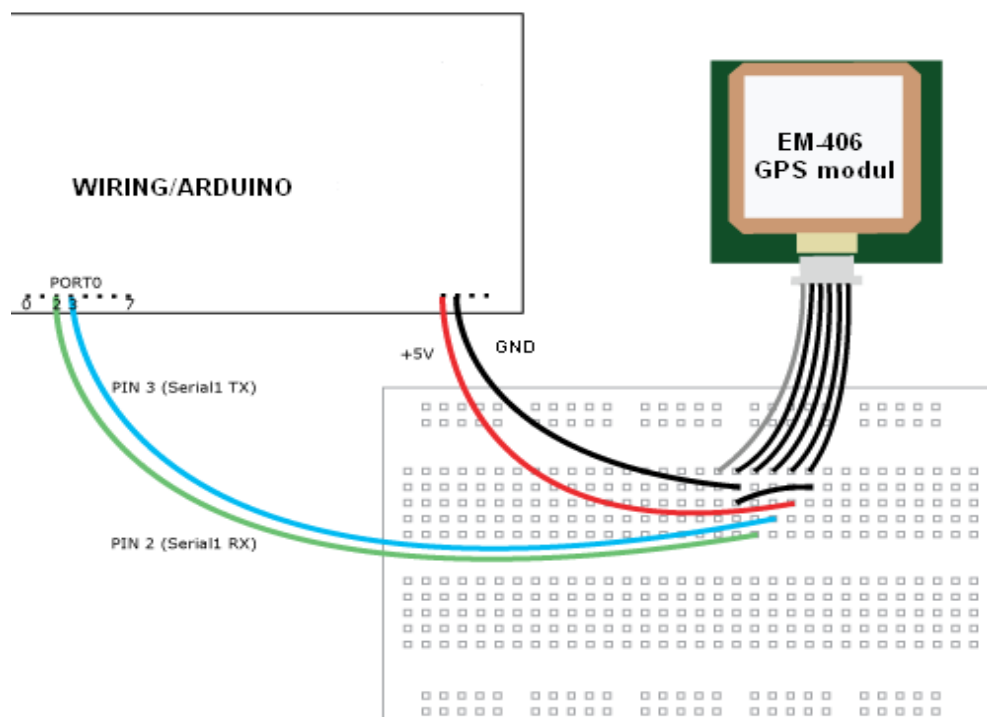
Primer prikazuje osnovne funkcije NMEA knjižnice. Izhodni signal GPS modula je povezan na vhodno linijo *Wiring*/Arduina, hitrost povezave znaša 4800 baud, svetleča dioda pa je povezana na digitalni vhodno/izhodni priključek 0.

Najprej ustvarimo GPS podatkovno povezavo za tip \$GPRMC stavka (priporočeni minimum opisan v poglavju 5.2.1), s katero potem pridobimo podatek o lokaciji. Ko se GPS modul nahaja na območju Slovenije, svetleča dioda zasveti, sicer pa je ugasnjena. »Slika 50« prikazuje potrebno kodo. Zaradi poenostavitve je Slovenija razdeljena na dva kvadrata, za bolj specifično sledenje pa bi seveda morali dodati dodatno kodo. »Slika 51« prikazuje kodo,

ki je potrebna za ugotavljanje smeri gibanja sprejemnika, relativno smer do cilja in hitrost, »Slika 52« pa prikazuje postopek ugotavljanja potrebe po spreminjanju smeri in katera LED naj posveti, v kolikor je sprememba potrebna.



Slika 48: Izhodi EM-406 GPS modula. Povezava poteka s hitrostjo 4800 baud paket pa vsebuje 8 podatkovnih bitov in 1 stop bit ter je brez paritetnih bitov



Slika 49: Shema povezave EM-406 GPS modula z Wiring ali Arduino. Povezave na sliki so vse potrebne povezave, ki jih EM-406 zahteva za delovanje.

```

GPS_EM_406 | Wiring 0024
File Edit Sketch Tools Help
GPS_EM_406 $
NMEA gps(GPRMC);

void setup() {
  Serial.begin(4800);
  pinMode(0, OUTPUT);
}

void loop() {
  if (Serial.available() > 0) {
    // Preberi in shrani podatke, ki jih pošilja GPS modul EM-406
    char c = Serial.read();
    // Preveri, če je stavek veljaven GPS stavek
    if (gps.decode(c)) {
      // Preveri, če so podatki veljavni
      if (gps.gprmc_status() == 'A') {
        // Preveri, če se sprejemnik nahaja v Sloveniji
        boolean vsloveniji = ((gps.gprmc_latitude() > 45.28)
          && (gps.gprmc_latitude() < 46.35)
          && (gps.gprmc_longitude() < 13.31)
          && (gps.gprmc_longitude() > 15.38))
          || ((gps.gprmc_latitude() > 46.20)
          && (gps.gprmc_latitude() > 46.49)
          && (gps.gprmc_longitude() > 15.38)
          && (gps.gprmc_longitude() > 16.24));
        // Če se GPS nahaja v Sloveniji LED prižgemo, sicer jo ugasnemo
        if (vsloveniji) {
          digitalWrite(0, HIGH);
        } else {
          digitalWrite(0, LOW);
        }
      }
    }
  }
}
Done compiling. No syntax errors found
33

```

Slika 50: Z Wiring preberemo podatke in preverimo, če se sprejemnik nahaja na področju Slovenije

5.4.4 Trenutna hitrost, razdalja do cilja in smer do cilja

Knjižnica nmea.h vsebuje nekaj zelo priročnih ukazov za delo z GPS navigacijo. Kot prikazuje »Slika 51«, lahko na dokaj enostaven način določimo smer gibanja, smer cilja in trenutno hitrost.

```

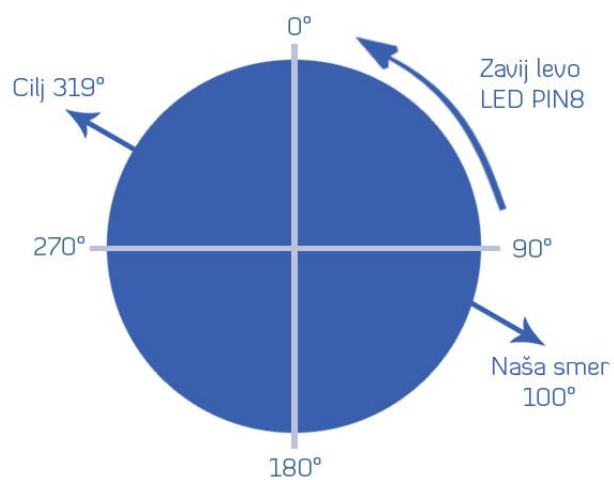
#include <nmea.h>

// Ustvarimo GPS podatkovno povezavo za GPRMC tip stavka
NMEA gps(GPRMC);
// Relativna smer do cilja
float d;

// Koordinate cilja
float dest_latitude = 45.72;
float dest_longitude = 14.28;

void setup() {

```

Slika 52: Smer cilja (319°) – smer sprejemnika (100°) = 219° . Ker je $219^\circ > 180^\circ$ odštejemo 360° kar znese -141° . Ker je -141° manjše od 5 in večje od -5 , prižgemo LED 8, kar pomeni, da moramo zaviti levo

6. ZAKLJUČKI IN MOŽNE RAZŠIRITVE

Diplomsko delo je imelo več ciljev. V prvem delu sem želel predstaviti, kaj GNSS sploh je in kaj pomeni v našem vsakdanjem življenju. Prav tako sem hotel bralcu razložiti osnovne pojme, povezane s satelitskim navigacijskim sistemom – kako le-ta deluje, na kaj je občutljiv, katere napake lahko nastanejo in kako izračun položaja sprejemnika sploh poteka. Hotel sem, da ta prvi del bralcu da osnovni pogled na delovanje GNSS, ki ga bo v naslednjih poglavjih potreboval za razumevanje bolj zapletenih prijemov. Možnih razširitev je pri prvem delu veliko, saj ima vsak problem, ki je obravnavan, še kopico drugih povezanih z njim, za vsakega pa obstaja različna razlaga in rešitev. Ker nisem hotel, da bi bilo diplomsko delo preobsežno in posledično nepregledno, sem se osredotočil na razlago najpomembnejših, s katerimi se tudi največkrat srečamo in ki doprinesejo največji delež k napakam. Z upoštevanjem teh problemov in njihovo odpravo lahko naredimo napravo, ki bo že zelo dobra, z implementacijo dodatnih rešitev pa bi naprava lahko služila že komercialnim namenom.

Ko je bralec seznanjen z osnovami, ki so predstavljene v prvem delu, si v drugem delu ogleda GNSS sisteme, ki so že v celoti postavljeni, deloma postavljeni, v izgradnji ali pa šele na papirju. Pomembno je, da ni opisan samo ameriški GPS, temveč tudi ruski GLONASS, kitajski COMPASS in evropski GALILEO. Vsi ti sistemi niso pomembni le zato, ker večim državam omogočajo satelitsko navigacijo, ampak ker se med seboj dopolnjujejo. Moderni GPS sprejemniki tako največkrat spremljajo tako GPS kot tudi GLONASS satelite, v prihodnosti pa bodo poslušali še GALILEO in COMPASS. S tem bo sledenje pridobilo na točnosti in pa zanesljivosti. O vsakem izmed teh sistemov bi se dalo napisati debelejšo knjigo, saj niso zanimive le tehnične karakteristike signalov ampak tudi zgodovina, razvoj, denarni vložek in tehnologija satelitov, ki pa niso tema tega diplomskega dela in sem jih zato v veliki meri izpustil.

V tretjem delu diplomske naloge so opisani koncepti uporabe GNSS. Medtem ko povprečen uporabnik pozna le avtomobilsko navigacijo, je v tem delu opisan še pomen GNSS v vojski, policiji, pomorstvu, letalstvu in športu. Razširitev tega dela bi lahko obsegala bolj podrobne analize prijemov, ki jih omenjena področja uporabljajo, prihranke zaradi uporabe GNSS in pa bolj futuristične napovedi, ki so zaenkrat še na papirju, z večanjem procesorske

moči in izboljšanjem ter optimizacijo algoritmov pa bi lahko kaj kmalu postale del našega vsakdana.

Četrty del diplomske naloge je namenjen programiranju interpreterja, ki zna izluščiti podatke iz NMEA stavka, ki se tudi največkrat uporablja v navigacijskih napravah, saj so ti podatki standardizirani. V začetku so predstavljeni operacijski sistemi, na katerih največkrat teče navigacijski program, nato pa se bralec seznanja z osnovnim NMEA stavkom in pomenom njegove vsebine. Preko učenja korak-za-korakom so opisani postopki čiščenja napak, določanje položaja, smer vožnje in opozorila zaradi prekoračitve dovoljene hitrosti itd. Predstavljen je tudi pomen geografskih informacijskih sistemov. Možna razširitev tega poglavja bi bila, da bi bil opisan celotni postopek kreiranja kartografije. Zaradi zapletenosti in dolgotrajnosti sem v tej diplomski nalogi pokazal le smernice pri kreiranju kartografije, bolj podrobna razlaga pa bi lahko bila že drugo diplomsko delo.

Zadnji del diplomske naloge opisuje strojno opremo, ki je potrebna za izgradnjo GPS naprave in pa osnovni program, ki ga taka naprava potrebuje za delovanje. Opisan je mikrokontroler Arduino in pa GPS modul EM-406. V tem delu je prikazana shema vezave in pa program, spisan v Wiring okolju, ki prikazuje nekaj osnovnih načinov delovanja GPS naprave, kot so določanje smeri, določanje smeri cilja, razdalja do cilja in hitrost sprejemnika. V razširitvi tega dela bi lahko bil prikazan tudi LCD zaslon, ki bi izpisoval vse relevantne podatke in pa zemljevid poti. Z uporabo GIS pristopov iz poglavja 5.3.6 bi lahko bila narejena polno funkcionalna navigacijska naprava.

SEZNAM SLIK

Slika 1: Napaka v shranjenih efemeridah se lahko odraža v napačnih izračunih lokacije.....	10
Slika 2: Dve sferi, ki se sekata v krožnici.....	11
Slika 3: Površina sfere seka krožnico v dveh točkah	12
Slika 4: Sfera satelita 4, r_4 , p_4 , da in položaj sprejemnika	13
Slika 5: Razlika v dolžini poti signala je lahko ključega pomena.....	14
Slika 6: Zelena puščica predstavlja neoviran signal, rdeče pa popačen zaradi odbojev	16
Slika 7: Velik DOP pomeni večje področje možne lokacije kar je slabše za točno lociranje	17
Slika 8: Če v »Slika7« dodamo še tri bolj enakomerno razporejene satelite dobimo manjši DOP oz. manjše področje možne lokacije.....	17
Slika 9: Svetilniške postaje v ZDA	20
Slika 10: Komponente sistema WAAS: geostacionarni komunikacijski satelit, GPS sateliti, GPS sprejemniki, FM oddajniki in sprejemniki	21
Slika 11: Konfiguracija LAAS na letališčih	22
Slika 12: Okvir navigacijskega sporočila in podatki, ki jih vsebuje	25
Slika 13: MLRS vozilo ameriške vojske	33
Slika 14: CSEL postaja	34
Slika 15: advanSea T.56 ladijski GPS sprejemnik s prikazom pomorske karte, kompasa, hitrosti vetra in temperature motorja	37
Slika 16: Najboljše kar trenutno premore cestna navigacija. Navigon 8410 pokaže celo sliko mesta po katerem se vozimo.	38
Slika 17: Letalska navigacija Garmin G500	39
Slika 18: Letalo brez posadke MQ-9 Reaper	40
Slika 19: Garmin Forerunner 205 s prikazano razdaljo in nadmorsko višino treninga	41
Slika 20: JAT Technology GPS-3503. V ozadju teče navigacijski program, spredaj pa je viden Windows CE 4.2.....	43
Slika 21: Grafični vmesnik Symbian naprav.....	44
Slika 22: \$GPRMC stavek En tak stavek vsebuje skoraj vse podatke, ki jih GPS aplikacija potrebuje: zemljepisno višino in širino, hitrost, smer, čas v satelitu, status lociranja in magnetno variacijo.....	45
Slika 23: Ugotavljanje pomena prve besede v VB.NET programskem jeziku	46
Slika 24: Dogodkovno prožen interpreter v VB.NET programskem jeziku.....	47
Slika 25: Izračun kontrolne vsote v VB.NET programskem jeziku	48
Slika 26: Prebiranje satelitovega časa in prilagoditev na lokalni čas v VB.NET programskem jeziku ...	50
Slika 27: Določanje hitrosti in smeri v VB.NET programskem jeziku.....	52
Slika 28: Koda z dodanim preverjanjem aktivne povezave v VB.NET programskem jeziku.....	55
Slika 29: \$GPGSV stavek	55
Slika 30: Grafična predstavitev \$GPGSV stavka, kjer središče kroga predstavlja trenutno lokacijo, robovi pa predstavljajo obzorje	55
Slika 31: Prebiranje \$GPGSV podatkov v VB.NET programskem jeziku	58
Slika 32: Urejanje lokalnega formata v VB.NET programskem jeziku	61
Slika 33: \$GPGSA stavek	63
Slika 34: Končna koda interpreterja, ki javlja tudi VDOP, HDOP in PDOP vrednosti, spisana v C# programskem jeziku.....	67
Slika 35: Geografski, projekcijski in točkovni koordinatni sistem	68

Slika 36: Mercator projekcija levo in Polyconic projekcija desno.....	69
Slika 37: Pretvarjanje iz stopinj v radiane ter obratno v C# programskem jeziku	70
Slika 38: Pretvarjanje iz geografskih v projekcijske koordinate in obratno v programskem jeziku C# .	71
Slika 39: Projekcija dveh geografskih koordinat v C# programskem jeziku	71
Slika 40: Aktivni del karte pri pretvorbi iz projekcijskih koordinat v točkovne	71
Slika 41: Razred Matrix in njegova uporaba v C# programskem jeziku.....	72
Slika 42: Obroba objekta in polnilo v C# programskem jeziku	72
Slika 43: Objekt brez korekcije (levo) in objekt s korekcijo razmerja (desno).....	73
Slika 44: Prilagajanje razmerja objekta v C# programskem jeziku	74
Slika 45: navigacijska karta z vpisanimi cestami in radarji. Pri uporabi povečave bi se seveda izrisale še ostale ceste.....	75
Slika 46: Arduino Duemilanove verzija 2009b. Na levi strani se nahajata USB in napajalni priključek, na spodnji strani analogni in na zgornji strani digitalni priključki.	77
Slika 47: EM-406 GPS modul. Dimenzije so 30x30x10,5mm. Teža znaša 16 gramov. Odlična izbira za mobilne naprave.....	78
Slika 48: Izhodi EM-406 GPS modula. Povezava poteka s hitrostjo 4800 baud paket pa vsebuje 8 podatkovnih bitov in 1 stop bit ter je brez paritnetnih bitov.....	79
Slika 49: Shema povezave EM-406 GPS modula z Wiring ali Arduino. Povezave na sliki so vse potrebne povezave, ki jih EM-406 zahteva za delovanje.	79
Slika 50: Z Wiring preberemo podatke in preverimo, če se sprejemnik nahaja na področju Slovenije	80
Slika 51: Potrebna koda za določitev smeri vožnje, smeri cilja in hitrosi, prav tako napisana v Wiring	81
Slika 52: Smer cilja (319°) – smer sprejemnika (100°) = 219°. Ker je 219°>180°odštejemo 360°kar znese -141°. Ker je -141°manjše od 5 in večje od -5, prižgemo LED 8, kar pomeni, da moramo zaviti levo	82

SEZNAM TABEL

Tabela 1: Možne in tipične napake z vrednostmi.....	14
Tabela 2: DOP vrednosti in njihov pomen	18

SEZNAM LITERATURE IN VIROV

6.1 Publikacije

- [1] J. Bell, *Basic GPS Navigation: A practical guide to GPS navigation*, 2008
- [2] B. Bhatta, *Global navigation Satellite Systems*, BS Publications, 2010.
- [3] Defense Science Board Task Force, *The Future of the Global Positioning System*, Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics Washington, D.C. , 2005
- [4] E. Kaplan, C. Hegarty, *Understanding GPS: Principles and Applications, Second Edition*, Artech House Publishers; 2 edition, 2005.
- [5] K. Kingsley-Hughes, *Hacking GPS*, Wiley Publshing, Inc., 2005
- [6] J. McNamara, *GPS for Dummies*, Wiley Publishing, Inc., 2004
- [7] R. Prasad, M. Ruggieri, *Applied Sattelite Navigation Using GPS, Galileo, and Augmentation Systems*, Artech House Publishers, 2005.

6.2 Članki

- [8] R. Langley, *The WAAS L5 Signal*, GPS World, May 2009.
- [9] C. Warner, *Police Incorporate GPS to Decrease Response Time and Fuel Consumption: Innovative Department Servng Historic City Embraces Technology*, Franklin media release, www.franklintn.gov, July 2009.

6.3 Spletni viri (dostopno Februarja 2010)

- [10] Arduino, Wiring, EM-406. Dostopno na:
 - <http://www.arduino.cc/>
 - <http://www.wiring.org.co/>
 - http://www.sparkfun.com/commerce/product_info.php?products_id=465
- [11] COMPASS. Dostopno na:
 - http://en.wikipedia.org/wiki/Beidou_navigation_system
- [12] Diferencialni GPS. Dostopno na:
 - <http://www.gpsinformation.org/dale/dgps.htm>
- [13] Galileo. Dostopno na:

- http://en.wikipedia.org/wiki/Galileo_system
 - <http://www.esa.int/esaNA/galileo.html>
- [14] GIS. Dostopno na:
- <http://www.gis.com/content/what-gis>
- [15] GLONASS. Dostopno na:
- <http://en.wikipedia.org/wiki/Glonass>
- [16] GNSS. Dostopno na:
- <http://en.wikipedia.org/wiki/Gnss>
 - http://wapedia.mobi/en/Global_Positioning_System?t=3.#4.
- [17] GPS. Dostopno na:
- <http://en.wikipedia.org/wiki/Gps>
- [18] Kepler. Dostopno na:
- http://en.wikipedia.org/wiki/Kepler's_laws_of_planetary_motion
- [19] Linux, Android, Maemo. Dostopno na:
- <http://en.wikipedia.org/wiki/Linux>
 - <http://en.wikipedia.org/wiki/Android>
 - <http://en.wikipedia.org/wiki/Maemo>
- [20] NMEA. Dostopno na:
- <http://www.gpsinformation.org/dale/nmea.htm>
- [21] Projekcije. Dostopno na:
- http://en.wikipedia.org/wiki/Van_der_Grinten_projection
 - http://en.wikipedia.org/wiki/Plate_carre
- [22] Radijske navigacije. Dostopno na:
- http://en.wikipedia.org/wiki/Radio_navigation
- [23] F. Snider, *GPS: Theory, Practice and Applications*
- <http://www.pdhoneonline.org/>
- [24] Symbian. Dostopno na:
- http://en.wikipedia.org/wiki/Symbian#integrated_Symbian_platform
- [25] Windows CE. Dostopno na:
- http://en.wikipedia.org/wiki/Windows_ce