

UNIVERZA NA PRIMORSKEM
Fakulteta za matematiko, naravoslovje
in informacijske tehnologije

Projektna dokumentacija
RAFAL

Predmet: projektni seminar, podiplomski študij,
magistrski program računalništvo in informatika,
štud. l. 2009/10

Avtorji:
Boštjan Titti
Damjan Krivicic
Jakob Bartolj

1. Uvod	2
2. Opis naprav	3
2.1 Opis merilne naprave SEQUENTIAL SAMPLER SEQ47/50	3
2.1.1 Trdni delci PM 10 in PM2.5.....	4
2.2 Opis industrijskega računalnika MOXA ThinkCore W325-LX	5
2.2.1 Specifikacija strojne opreme	6
2.2.2 Specifikacija programske opreme	7
2.2.3 Blok diagram strojne opreme	9
2.2.4 PIN razporeditev	9
2.2.5 Serijski konzolni port	9
3 Spletne storitve	10
3.1 Arhitektura spletnih storitev	10
3.2 Specifikacije spletne storitve.....	11
3.3 SOAP.....	12
3.4 WSDL.....	12
3.5 UDDI.....	12
3.6 SOA.....	13
3.7 Razvoj SOA v prihodnje	14
3.8 Arhitektura REST.....	14
3.9 WOA	15
3.10 MySQL.....	15
3.11 SSH.....	16
3.12 PPP	17
4 GPRS prenos podatkov	17
5 Prenos podatkov iz merilne naprave do uporabnikov	20
5.1 Diagram poteka prenosa podatkov iz merilne naprave do strežnika.....	23
6 Postavitev sistema	24
6.1 Povezava Moxa W325 z računalnikom.....	24
6.2 Konfiguracija Ethernet vmesnika.....	26
6.3 GPRS omrežja	27
6.4 Omogočanje in onemogočanje programov, ki se izvajajo v ozadju (Daemons).....	28
6.5 Avtomatski zagon programa v Run nivoju (Run-Level).....	30
6.6 Nastavitev časa v sistemu.....	31
6.7 Cron-Program (Daemon) za izvajanje načrtovanih ukazov	33
6.8 Upravljanje z komunikacijami	33
6.8.1 Telnet/FTP.....	33
6.8.2 DNS.....	34
6.8.3 Spletna storitev-Apache	34
6.8.4 Inštalacija PHP-ja za Apache spletni strežnik.....	35
PRILOGE	38

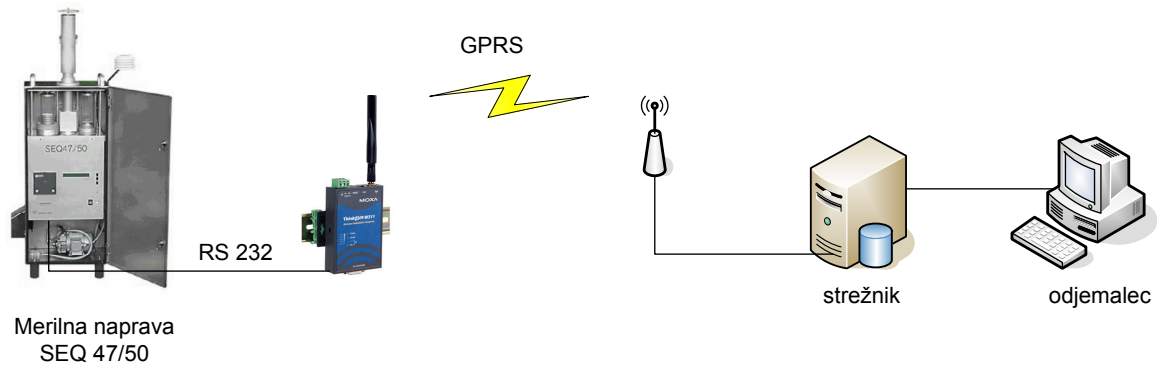
1. Uvod

Projekt „RAFAL“ (*Remote Access For Advanced Leckle*) je nastal v okviru predmeta *Projektni seminar*, na podiplomskem študiju Fakultete za matematiko, naravoslovje in informacijske tehnologije (UP FAMNIT). Razdeljen je na naslednje zahteve:

- Projektni načrt,
- Funkcionalna specifikacija,
- Nefunkcionalna specifikacija,
- Arhitektura sistema,
- Predlog strojne opreme,
- Predlog programske opreme,
- Razdelitev na module,
- Moduli,
- Implementacija modulov,
- Ovrednotenje investicije,
- Tehnična dokumentacija,
- Testiranje modulov,
- Integracija sistema,
- Testiranje celotnega sistema,
- Testna dokumentacija,
- Uporabniška dokumentacija,
- Dokumentacija.

Namen projektnega seminarja je, da s potrebno IKT podporo, merilnik SEQ47/50 podjetja Leckle spremeni v ponudnika spletnih storitev (SOA (*angl. Service-Oriented Architecture*)), ki bo omogočal odjemalcem prebiranje vrednosti iz merilne naprave. Skupaj z merilno napravo smo dobili programsko opremo, ki omogoča branje podatkov iz naprave in nato oblikovanje izpisa. V ta namen je potrebno, da smo fizično poleg naprave. Ker je merilnik nameščen v Luki Koper, kjer ni neposredne ožičene povezave z internetom, na voljo je le napajanje 230V, zato je seveda takšno branje nepraktično. Število podprtih odjemalcev ni omejeno, vendar je potrebno paziti na sklabilnost (razširljivost) in na problem hkratnega popravljanja vrednosti na napravi.

Na sliki 1 lahko vidimo poenostavljeno arhitekturo sistema, kjer merilna naprava SEQ 47/50 opravlja meritve delcev v zraku. Ta je preko RS232 vmesnika povezan z mikro računalnikom, ki je industrijske izvedbe predvsem zaradi okoljskih pogojev (prah, temperatura, ...) v katerih obratuje merilna naprava. Mikro računalnik z Linux OS je pritrjen v notranjosti merilne naprave in nima gibljivih delov ali zaslona. Podatke meritev zajema iz merilne naprave in jih shranjuje, ter pošlje v določenem časovnem intervalu preko GPRS (*angl. General Packet Radio Service*) modema do strežnika.



Slika 1: Poenostavljena arhitektura sistema

2. Opis naprav

2.1 Opis merilne naprave SEQUENTIAL SAMPLER SEQ47/50

Merilnik Leckel je sestavljen iz ohišja iz nerjavečega jekla v katerem je nameščena vakuumška črpalka z nastavljenim pretokom, magazinom z nastavki za čiste filtre in magazinom, kjer se zbirajo že vzorčeni filtri. Zunaj ohišja sta nameščena senzorja za merjenje temperature in atmosferskega tlaka. Vzorčevalna glava za PM₁₀/PM_{2,5} dimenzijsko in oblikovno ustreza zahtevam standarda SIST EN 12341/2000 in SIST EN 14907/2005. To je merilnik z nizkim volumskim pretokom (LVS (*angl. Low Volume Sampler*)) in je namenjen zunanji meritvi zraka pri vseh temperaturah in pogojih okolja. V magazinu je 17 filtrov, ki jih vzorčevalnik sam menja ob datumu in uri, ki ju nastavimo. Čas vzorčenja je 24 ur.

Pretok zraka skozi vzorčevalnik je majhen, da praktično ni izgube lahkih snovi. Celoten vzorčevalni sistem se hladi s tokom zraka. Kot medij za zbiranje delcev uporabljamo filtre.

Pretok skozi vzorčevalnik je kontroliran skladno po prvem fizikalnem zakonu. Temperatura pretoka zraka se meri direktno za filtrom, ko je le-ta v poziciji vzorčenja. Če temperatura v zimskem obdobju pade pod točko rosišča, se bo temperatura filtra dvignila, da ne pride do kondenzacije znotraj merilnika.

- Pretok zraka skozi vzorčevalnik $2,3 \text{ m}^3 / \text{h}$
- Minimalni čas vzorčenja na enem filtru je 1 ura, maksimalni čas pa 168 ur.

Prednosti

- Regulirano in kontrolirano z integriranim LVS3 ali MVS6 napravo za vzorčenje,
- Kontroliranje delovne m^3/h in standardne m^3/h ($0^{\circ}C$ ali $20^{\circ}C$, 760 mm Hg) skozi napravo za merjenje pretoka zraka,
- Ohišje iz nerjavečega jekla za zunanjo uporabo; 19 inčna samostoječa omara za vgradnjo v merilne naprave,
- 8 različnih vhodov za PM10-PM4,0-PM2,5-PM1,0
- Vhod za TSPM,
- Dolgi vzdrževalni intervali frakcionalnih vhodov,
- Meritev temperature filtra, filter se avtomatično segreje v zimskem času,
- Hlajeni vhodni sistem in zaprti vzorčni filtri v magazinu, nobenih izgub določenih volatilnih materialov,
- Sesalna cev z velikim notranjim premerom, ni izgub delcev na notranji steni,
- Uporaba filtrov z premerom 47 in 50mm.



PM10 referenčni vhod po CEN EN 12341 standardu,
PM2,5 standardni vzorčevalnik po CEN EN 14907 standardu.

2.1.1 Trdni delci PM 10 in PM2.5

Trdni delec (PM) je izraz za prah, ki je prisoten v zraku v določenem obdobju. Kot aerosol je v obliki vodne kapljice, v kateri je ujet trden ali tekoč delec. V veliki večini delcev je glavna komponenta ogljik, na tega pa se lahko vežejo primesi kot so kovine, organska topila ali ozon. Najpogosteje se izvajajo v zadnjih letih meritve delcev premera 10 (PM_{10}) in 2,5 ($PM_{2,5}$) μm , ki so zdravju najbolj škodljive.

Sestava delcev je odvisna od izvora delcev. Velja, da se manjši in svetlejši delci zadržujejo v zraku dalj časa. Večji delci se zadržujejo v atmosferi nekaj ur, medtem ko lahko manjši delci ostanejo v atmosferi več tednov in se navadno »sperejo« iz atmosfere šele s padavinami. Delci so naravnega (dim gozdnih požarov, vulkanski pepel) ali antropogenega izvora (energetski objekti, promet, industrija, individualna kurišča). Delci vplivajo na zdravje ljudi, kakor tudi na klimo, vidnost in podobno.

V zimskih mesecih pomembno prispevajo k emisiji trdnih delcev individualna kurišča na les in fosilna goriva. V preteklih letih so največji delež k izpustom skupnega prahu prispevala individualna kurišča. Promet predstavlja velik vir onesnaženja z najmanjšimi delci, predvsem na območjih z veliko gostoto prometa.

Urna mejna koncentracija za varovanje zdravja ljudi je $20 \mu\text{g}/\text{m}^3$ - PM_{10} .

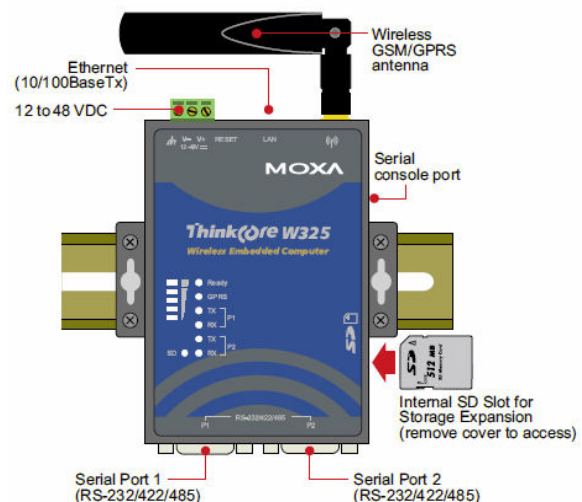
Delci povečajo umrljivost za boleznimi dihal, srca in ožilja. Predvsem so ogroženi starejši in bolniki z obstoječimi boleznimi dihal. Če delci vsebujejo težke kovine, je njihova strupenost še večja. Dokazali so, da je prisotnost cinka v delcih poveča moč vnetja, stopnjo odmiranja tkiv in preobčutljivosti pljuč.

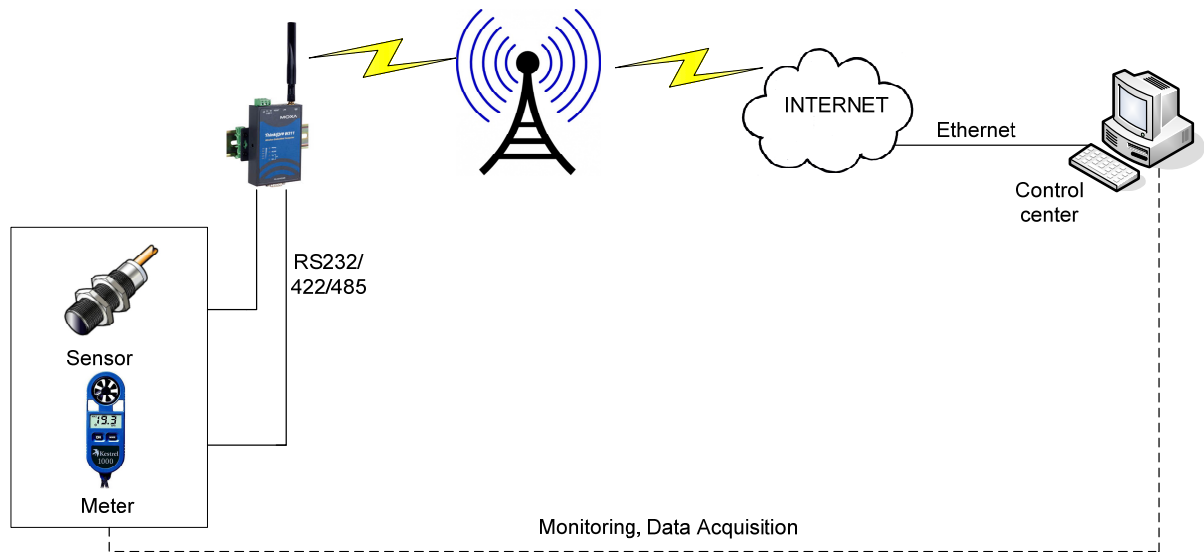
2.2 Opis industrijskega računalnika MOXA ThinkCore W325-LX

ThinkCore W325-LX je računalnik na wireless RISC (*angl. Reduced Instruction Set Computing*) osnovi z GSM/GPRS (*angl. Global System for Mobile Communications/General Packet Radio Service*) modulom, RS-232/422/485 serijskim portom in Ethernet portom, ki v našem primeru ne pride v upoštevanje. Dodano ima tudi SD (*angl. Secure Digital*) odprtino za dodatni spominski prostor. Idealen je za različne strojne aplikacije in omogoča wireless operacije kot tradicionalno ožičeno omrežje ter serijske naprave. Lahko obravnava transparentne podatkovne transferje, numerično računanje, protokolno zamenjavo, podatkovno procesiranje in celo podatkovno enkripcijo.

Prednosti

- MOXA ART ARM9 32-bit 192 MHz procesor,
- On-board 32 MB RAM, 16 MB flash disk,
- Vgrajen quad band GSM/GPRS 850/900/1800/1900 MHz,
- GPRS razred 10, poprta kodirna shema iz CS1 v CS4,
- Možnost izbire med dvema programskima oprema za RS-232/422/485 serijske porte,
- 10/100 Mbps Ethernet za omrežno redundanco,
- Konstruirano, da vzdrži konstantne 5G vibracije in 50G trk,
- SD odprtino za dodatni spominski prostor,
- Ready-to-run Linux Kernel 2.6 platforma,
- DIN-tračnica ali možnost montaže na steno,
- Robustno, brez ventilatorja.





Slika 2: Tipična aplikacija

2.2.1 Specifikacija strojne opreme

Sistem:

CPU: MOXA ART ARM9 32-bit RISC CPU, 192 MHz

DRAM: 32 MB

Flash: 16 MB

Razširitev spomina: 1 x SD odprtina

Conzolni Port: 1 x RS-232 (TxD, RxD, GND), 4-pin header izhod

Gumb: 1 x Reset gumb, podpora "Reset to Factory Default"

Ostalo: RTC, buzzer, Watchdog Timer

OS: Vgrajen Linux z MMU podporo

Celične komunikacije:

Celični način: GSM, GPRS

Radio frekvenčni pas: 850/900/1800/1900 MHz

GPRS razred: 10

Kodirne sheme: CS1 do CS4

Omrežna komunikacija:

LAN: Automatsko zaznavanje 1 x 10/100Mbps, RJ45

Zaščita: Vgrajena 1.5KV magnetna izolacijska zaščita

Serijska komunikacija:

Serijski port: RS-232/422/485 port z dvojno izbiro programske opreme , DB9 moški konektor
Zaščita: Vgrajena 15KV ESD zaščita za vse signale

Podatkovni biti: 5, 6, 7, 8

Stop biti: 1, 1.5, 2

Parity: None, Even, Odd, Space, Mark

Flow kontrola: RTS/CTS, XON/XOFF, RS-485 ADDCTM

Hitrost: 50 bps do 921.6 Kbps, podpira vse Baud Rate

LED:

Sistem: V pripravljenosti, SD

Celično: GPRS Omogočen, Jakost GSM Signala

LAN: 10M/povezava (na konektorju), 100M/povezava (na konektorju)

Serijsko: TxD, RxD

Napajanje:

Napajalna napetost: 12 do 48 V DC

Mehanične lastnosti:

Dimenzije: (ŠxDxV): 77 x 111 x 26 mm (brez nastavka za montažo in antene)

Teža: 195 g

Antena: 110 mm

Konstruktivni Material: aluminij, 1 mm

Montaža: Stenska, DIN-tračnica (opcijsko, oprema za DK35A DIN-tračnice)

Okolje:

Obratovalna Temperatura: -10 to 60°C, 5 to 95% RH

Temperatura shranjevanja: -20 to 80°C, 5 to 95% RH

Anti-Vibracije: 5G @ IEC-68-2-6, sinusni val, 5-500 Hz, 1 Oct./min, 1 hr/axis.

Anti-Shock: 50G @ IEC-68-2-27, polovica sinusnega vala, 30 ms

Regulatorska odobritev:

EMC: FCC, CE (Razred A)

Varnost: TUV, UL, cUL

Ostali: RoHS, CRoHS, WEEE

Garancija: 5 let

2.2.2 Specifikacija programske opreme

Linux:

Verzija jedra: 2.6.9

Boot Loader: Redboot

Protokolni skladi: TCP, UDP, IPv4, SNMP V1/V3, ICMP, IGMP, ARP, HTTP, CHAP, PAP, SSH 1.0/ 2.0, SSL, DHCP, NTP, NFS,SMTP, Telnet, FTP, PPP, PPPoE

Datotečni Sistem: JFFS2 (on-board flash)

Sistem Utilities: bash, busybox, tinylogin, telnet, ftp, scp

Podprte storitve in Daemons: telnetd: Telnet Server daemon

ftpd: FTP server daemon

sshd: Secure shell server

Apache: Web server daemon, podpora PHP in XML

OdprtiVPN: Virtualni privatni servisno omrežni manager

iptables: manager storitve požarni zid

pppd: kliči in/out preko serijskega porta daemon & PPPoE

snmpd: snmpd agent daemon

inetd: TCP server manager program

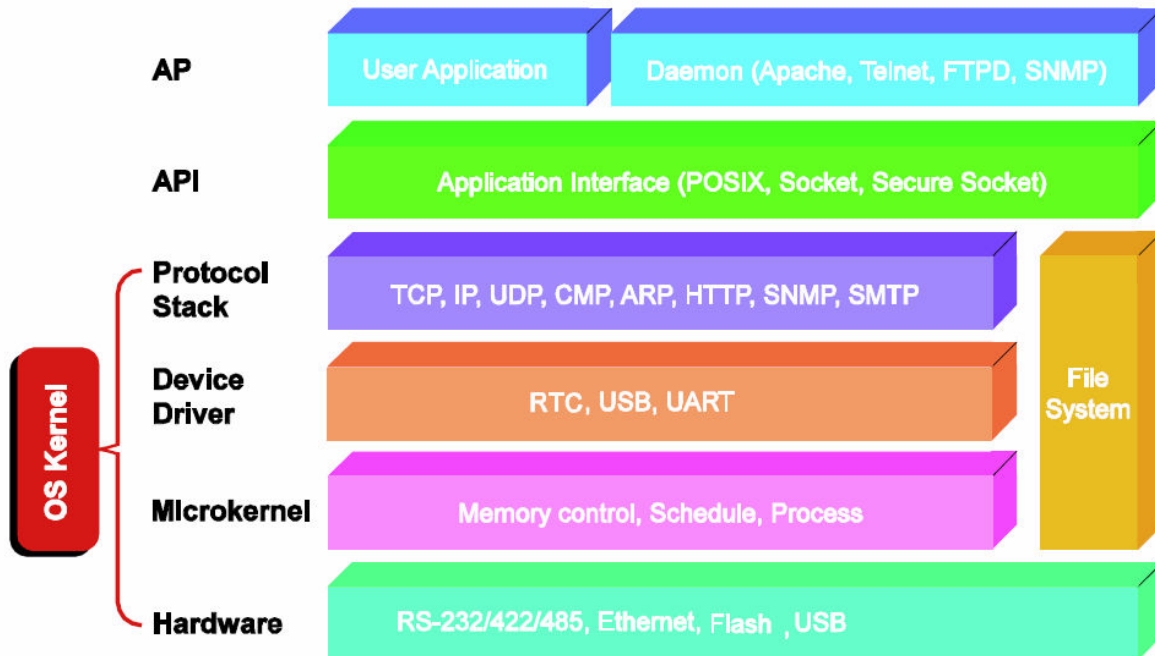
Application Development Environment: MOXA Linux API knjižnjica

Linux Tool Chain: Gcc, Glibc, GDB

Windows Tool Chain: Gcc, Glibc, Insight (Windows-based source level debug utility)

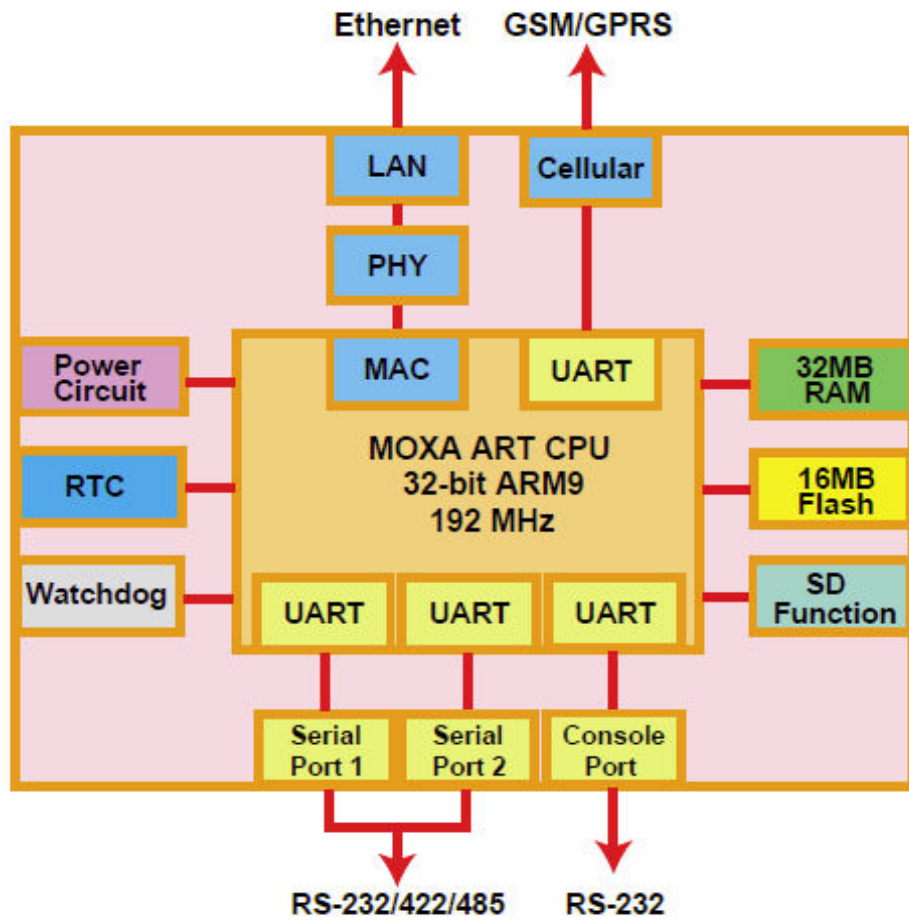
BINEncryptor: Enkripcijsko orodje za binarne datoteke; na osnovi MOXA Intelktualne zaščitne tehnologije (patentirano)

Gonilniki: SD Card, UART, RTC, Buzzer

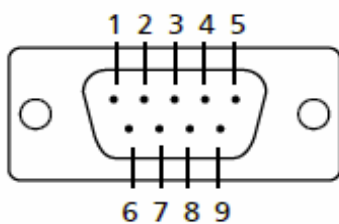


Slika 3: Arhitektura programske opreme

2.2.3 Blok diagram strojne opreme

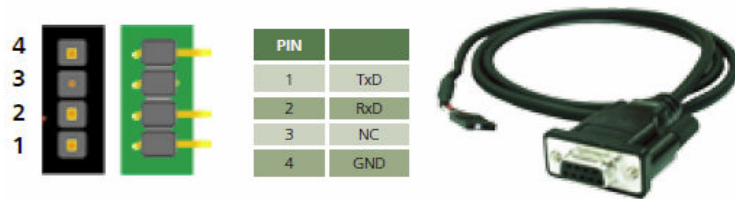


2.2.4 PIN razporeditev



PIN	RS-232	RS-422	RS-485 (4-wire)	RS-485 (2-wire)
1	DCD	TxDA(-)	TxDA(-)	---
2	RxD	TxDB(+)	TxDB(+)	---
3	TxD	RxDB(+)	RxDB(+)	DataB(+)
4	DTR	RxDA(-)	RxDA(-)	DataA(-)
5	GND	GND	GND	GND
6	DSR	---	---	---
7	RTS	---	---	---
8	CTS	---	---	---

2.2.5 Serijski konzolni port



3 Spletne storitve

Spletne storitve so programska oprema, narejena za podporo povezave med računalniki preko omrežja. Spletne storitve so navadni spletni aplikacijski programski vmesnik, do katerega dostopamo preko omrežja, kot npr. spleta, kjer se izvaja na oddaljenem računalniku.

Kot začetek spletnih storitev lahko omenimo 26. januar 2002, ko so bili predstavljeni protokoli XML – *Extensible Markup Language*, ter arhitektura SOAP – *Simple Object Access Protocol*, verzija 1.2. Naslednji zgodovinski datum za spletne storitve je 9. julij 2002, izid WDSL 1.1 – *Web Service Description Language*.

Zgodovina spletnih storitev je podrobneje predstavljena na spletni strani <http://www.w3.org/2002/ws/history.html>.

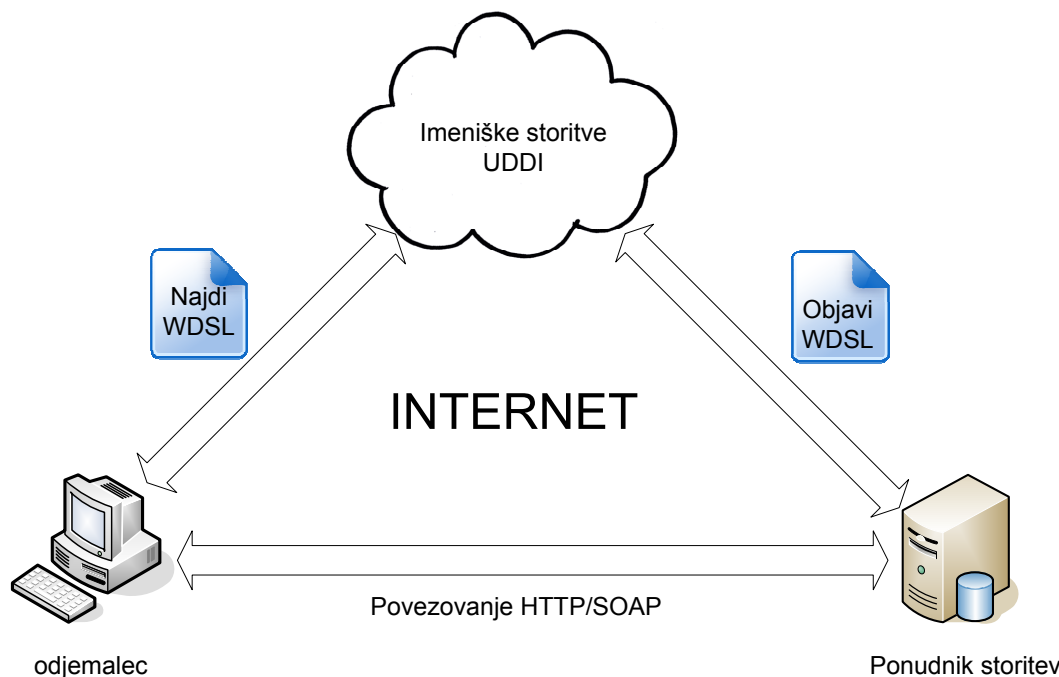
Spletne storitve so odvisne od različnih principov in arhitektur. Najpomembnejše so spletno orientirane arhitekture (*Web Oriented Architecture*), storitveno orientirane arhitekture (*Service Oriented Architecture*) in programska arhitektura (*REpresentational State Transfer*).

V današnjem času so spletne storitve malo več kot komponente programske opreme, ovite v SOAP-vmesnik. In tako lahko razvijalci dodajo preprosto XML-podporo k njihovim produktom in te tako lahko ponujajo testiranje spletnih storitev uporabnikom.

Zato se pri testiranju spletnih storitev pogosto sprašujemo, katero orodje je najboljše za testiranje spletnih storitev, kajti veliko je komercialnih produktov in njihova kakovost oz. uporabnost je slaba.

3.1 Arhitektura spletnih storitev

Slika 3 prikazuje pošiljanje zahtev in prejemanje odgovorov. Pri tem se uporabljajo jezik WSDL-protokola HTTP in SOAP ter UDDI-register. API (ang. *Application Programming Interface*) – je aplikacijski programski vmesnik, ki vsebuje nabor funkcij, ki jih storitev omogoča, da lahko sprejema zahteve, dane s strani odjemalca.



Slika 4: Prikaz arhitekture spletnih storitev, ter pošiljanja zahtev in sprejemanja odgovorov v spletnih storitvah

Strežnik

Strežnik je aplikacija ali sistem, ki izvaja storitve s povezanim odjemalcem, v smislu odjemalec-strežnik.

Strežniška aplikacija je bila definirana na podlagi HTTP-ja, ki deluje tako, da sprejme zahteve in pošilja odgovore. Strežniki so računalniki, narejeni, da se na njih izvajajo aplikacije, z daljšo časovno periodo s čim manjšim človeškim posegom v delovanje strežnika.

Odjemalec

Je aplikacija ali sistem, ki dostopa do oddaljenega strežnika na drugem računalniškem sistemu, ki je poznan kot strežnik, vse to poteka preko omrežja. Model odjemalec-strežnik se še vedno uporablja na spletu, kjer se lahko uporabnik poveže do storitve preko SOAP- protokola. Spletni iskalniki so odjemalci, ki se povežejo do spletne storitve in prikažejo vsebino spletne strani.

XML

XML-jezik je obravnavan kot "razširljiv" označevalni jezik, ker dovoljuje uporabo svojih oznak (tags). XML je zelo preprost in podoben HTML-ju. Njegov glavni namen je, da olajšuje razdeljevanje podatkovne strukture preko različnih informacijskih sistemov, zlasti preko spleta. XML-jezik je priporočen s strani W3C (mednarodna organizacija za standarde) konzorcija in spada med odprte standarde.

XML je razdeljen na tri dele:

- *podatkovni* (tukaj shranjujemo podatke, ki lahko imajo različne oznake)
- *deklarativni* (skrbi za pomen oznak pri dodajanju novih podatkov)
- *predstavitveni* (z njim oblikujemo izpis podatkov)

3.2 Specifikacije spletne storitve

Glavne specifikacije iz tehnološkega vidika spletnih storitev so:

- *SOAP* (ang. *Simple Object Access Protocol*) – protokol

- *WSDL* (ang. *Web Service Descripton Language*) – programski jezik
- *UDDI* (ang. *Universal Descripton, Discoveryand Integretion*) – register

3.3 SOAP

SOAP je na začetku pomenil "enostavni objektni vstopni protokol", vendar ga zdaj enostavno imenujemo samo SOAP. SOAP-verzija 1.1 je bila izdana leta 2000 s podporo W3C-organizacije. Zadnja verzija SOAP je 1.2, izdana je bila 2007, prav tako ob podpori W3C-organizacije.

SOAP je protokol za izmenjavo XML-sporočil preko računalniškega omrežja, ponavadi z navezo HTTP/HTTPS. Obstaja nekaj različnih tipov sporočil pri SOAP-u, ampak med najbolj pogostimi se uporablja RPC (oddaljeno obravnavanje klicev). Deluje na ta način, da omrežje (odjemalec) pošilja sporočila v obliki zahtev na druga vozlišča (strežnike) in potem ti strežniki pošiljajo odzivna sporočila nazaj k odjemalcu. To je protokol, ki skrbi za komunikacijo med aplikacijami. HTTP (angl. *HypetText Transfer Protocol*) je glavna metoda prenosa informacij na spletu. SMTP (angl. *Simple Message Transport Protocol*) in HTTP sta veljavna aplikacijska protokola, ki se uporabljata kot "prevozno sredstvo" za SOAP. SOAP se lahko prav tako uporablja preko HTTPS-ja, ki uporablja kriptiranje podatkov oz. sporočil.

Glavna prednost pri uporabi SOAP-a preko HTTP-ja je, da omogoča lažje komuniciranje preko posrednikov (ang. *proxy-ev*) in požarnih zidov. Kot slabost lahko navedemo obsežnost XML-jezika, ki upočasni SOAP-protokol, in veliko je takšnih implementacij SOAP-a, ki omejujejo velikost poslanih podatkov.

Nekaj največjih razlik med SOAP 1.1 in SOAP 1.2:

Dokument verzije SOAP 1.2 je sestavljen iz treh delov, medtem ko je SOAP 1.1 le samostojen dokument. SOAP 1.1 temelji na XML 1.0, SOAP 1.2 na XML-infoset. Tudi domenski prostor se je pri SOAP 1.2 spremenil.

3.4 WSDL

WSDL je jezik za opisovanje spletnih storitev, ki temelji na XML-jeziku. Primarna protokola, preko katerih deluje, sta HTTP in HTTPS.

Trenutna verzija WSDL je 2.0, ki je v bistvu preimenovana verzija WSDL-ja 1.2, katero podpira konzorcij W3C. Prejšnjo verzijo WSDL-jezika 1.1 konzorcij W3C ni podpiral. Glavne razlike med verzijama so, da WSDL 2.0 ne podpira samo *get* in *post* kot verzija WSDL 1.1, ampak podpira tudi ostale druge zahteve.

HTTP je glavna metoda za prenos informacij na splet. Protokol je bil prvotno namenjen objavljanju in prejemanju HTML-strani.

HTTPS je "zavarovana" različica HTTP-ja. Uporablja SSL (kriptografski medmrežni protokol) in TLS (naslednik SSL-ja in deluje podobno kot SSL), zato da zakodira in s tem zaščiti promet pred vmesnimi opazovalci.

Jezik WSDL se največkrat uporablja v navezi s SOAP-om in XML-shemo – za oskrbovanje spletnih storitev preko spleta. Odjemalčev program se poveže na spletno storitev in prebere WSDL-dokument, da določi katere funkcionalnosti so razpoložljive na strežniku. Potem lahko odjemalec uporabi SOAP-protokol, da dejansko pokliče eno od funkcionalnosti, ki je v WSDL-dokumentu.

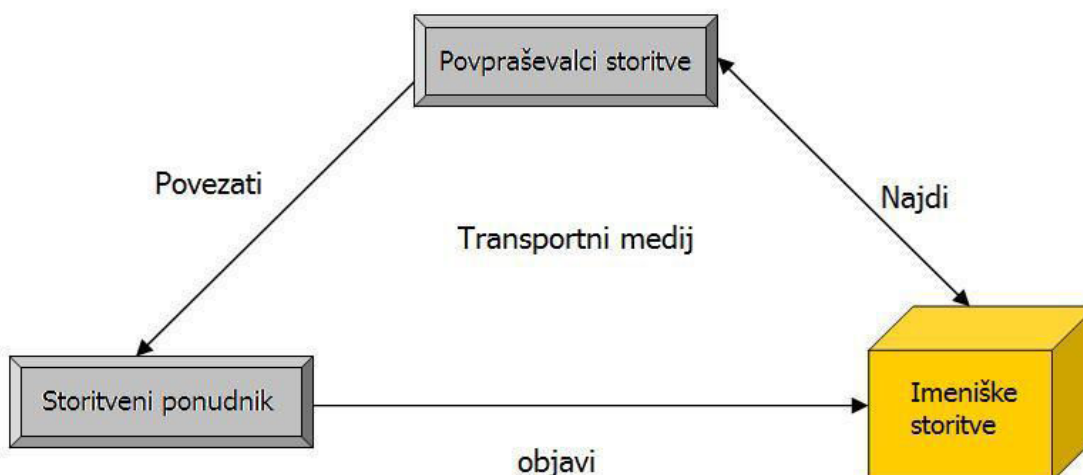
3.5 UDDI

Je protokol za objavljanje in raziskovanje podatkov o spletnih storitvah, ki omogoča, da aplikacija najde te podatke. Temelji na jeziku XML, v njem pa hrani dokument WSDL in ostale pomembne dokumente, ki opisujejo spletno storitev in so pomembni za odjemalca.

Odjemalec mora poznati način, kako uporabiti storitve, ki jih določena spletna storitev ponuja. UDDI ni repozitorij, kar pomeni, da ni podatkovna shramba, pač pa register, ki usmerja uporabnika na ustrezne vire s podatki. Zadnja verzija UDDI-registra 3.0.2 je izšla leta 2006.

3.6 SOA

V računalniškem svetu izraz "*Service Oriented Architecture – SOA*" ni nov in predstavlja arhitekturo, v katero je združenih več neodvisnih programskih paketov. Programske storitve niso med seboj tesno povezane in so praviloma tudi od različnih proizvajalcev, vendar komunicirajo med seboj po vnaprej predpisanem pravilu. Pri SOA ne gre za novo tehnologijo, saj so lahko posamezne storitve implementirane na različnih tehnologijah, bistveno pa je, da so dostopni vsem uporabnikom ali drugim storitvam. V smislu povezovanja gre torej za zelo odprto strukturo. SOA je arhitektura, ki se nanaša na storitveno orientirani princip oblikovanja. SOA predstavlja model, ki je funkcionalno razgrajen na majhne, jasne enote (storitve), ki jih lahko razvrstimo preko omrežja in ki jih sestavimo skupaj, da tvorijo poslovno aplikacijo. Storitve komunicira z vsakim na ta način, da pošilja podatke od ene storitve do druge. Arhitektura SOA:



Slika 5: Arhitektura SOA

SOA je povezana z meta podatki, ki so primerni za opisovanje ne samo značilnosti storitev, ampak tudi opisujejo podatke, s katerimi so povezani. Spletne storitve so tipično napisane v WSDL-jeziku in komunicirajo po SOAP-protokolu. Za SOAP lahko rečemo, da je odvisen od podatkov in storitev, ki so opisani v najrazličnejših implementacijah meta podatkov (*meta podatek* = podatek o podatku).

Glavni namen SOA je povezovanje poslovnih in proračunskih sredstev (zlasti organizacija, aplikacije in podatki) na zahtevo doseganja željenih rezultatov za storitvene potrošnike. Tudi nove aplikacije, narejene iz različnih storitev, omogočajo večjo prilagodljivost in poenotenost.

Najpomembnejša obljuba storitvene arhitekture (SOA) je zmožnost razvoja takega informacijskega sistema, ki direktno podpira poslovne procese podjetja in je veliko bolj prilagodljiv od obstoječih rešitev. Zato SOA omogoča dejansko, merljivo povečanje učinkovitosti informatike ter omogoča, da prihranke namenimo razvoju novih, inovativnih storitev.

SOA je zgrajen iz treh glavnih delov:

- storitveni ponudnik (ang. *Service provider*)

Storitveni ponudnik naredi spletno storitev in morda objavi vmesnik in dostopne informacije na storitvenem registru. Vsak ponudnik mora pri tem narediti več odločitev: od varnosti, hitrosti dostopnosti, ceni itd.

- storitveni posrednik (ang. *Service broker*)

Znan kot storitveni register, odgovoren je za vmesnik upraviteljskega modula in implementacijski dostop ter zagotavlja dostopnost vsem storitvenim povpraševalcem. Javni storitveni posredniki so dostopni preko spleta, medtem ko so privatni dostopni le redkim, npr. dostop preko notranjega omrežja v podjetju. UDDI določa, na kakšen način se naj objavljajo in odkrivajo informacije o spletnih storitvah.

- storitveni povpraševalci (ang. *Service requestor*)

Storitveni register uporabljajo za iskanje primernih storitev, ki ustrezajo njihovim zahtevam.

3.7 Razvoj SOA v prihodnje

Medtem ko se industrija še vedno ukvarja s SOA, Oracle že govori o novi verziji SOA 2.0, ki bo kombinirana kot storitveno-orientirana arhitektura (SOA) in dogodkovno-vodena arhitektura (EDA) in ki kategorizira SOA kot nekakšno vodenje odjemalec-strežnik.

EDA (ang. *Event Driven Architecture*) je arhitektura programske opreme, ki bo delovala kot predloga k pospeševanju produktivnosti, odkritosti, manjši porabi in bo pozitivno reagirala na dogodke. Dogodek je lahko definiran kot pomemben prehod med stanji.

Večina atributov, ki jih obljublja SOA 2.0, že obstaja oz. jih že uporabljajo nekatera podjetja (Tibco podjetje ...), zato je možno, da je SOA 2.0 le marketinška oz. tržna poteza.

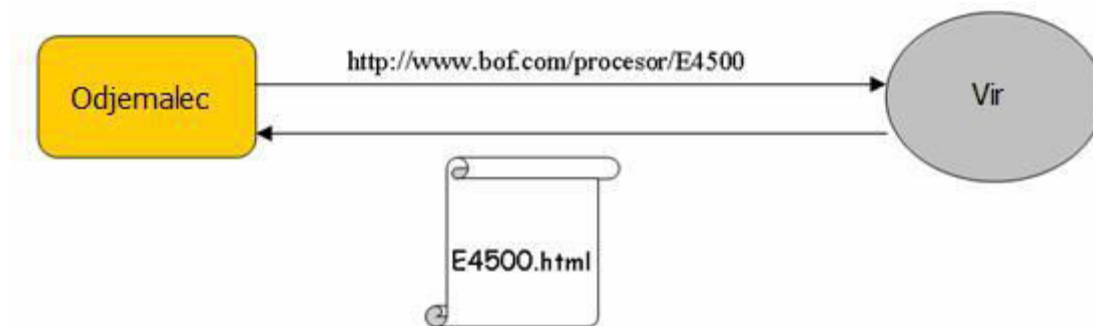
Oracle vidi združitev (java platforme, Jave EE5, SOA 2.0 in Web 2.0) zaradi čim večje produktivnosti pri aplikacijskih platformah. Zato je veliko industrij, ki pravijo, da bi morala naslednja generacija SOA (SOA 2.0) biti bolj usmerjena na optimizacijo IT (informacijskih tehnologij) pri poslovnem razvoju.

3.8 Arhitektura REST

REST (angl. *Representational State Transfer*) je arhitektura programske opreme za hipermedijske (angl. *hypermedia*) sisteme, kot so svetovni splet (WWW). REST se uporablja kot opis za preprosti vmesnik, ki prenaša specifične domenske podatke preko HTTP-ja ter deluje brez dodatnega sporočilnega sloja, kot je SOAP-protokol. Arhitektura REST je bila predstavljena leta 2000 s strani Royja Fieldinga, ki je tudi eden od glavnih avtorjev protokola HTTP. REST nam omogoča oblikovanje velikih sistemov programske opreme, brez uporabe HTTP ali WWW. Z njim je možno oblikovati preproste XML + HTTP vmesnike, ki niso skladni z REST-principom. Definicijo REST-a lahko razumemo kot nasprotje SOAP-a. Na začetku je bil definiran kot koncept, ki ima dostop do informacij in medijev. REST je uporaben z zahtevo *get*, ampak nepripravljen oz. neprimeren za druge kot npr. za zahtevo *put*. Razviti so bili tudi nekateri REST-sistemi, ki so podpirali *get*, *post*, prav tako tudi HTTP-metode *head*, *delete*, *put*, *copy*, *move* itd.. Glavni problem sistema REST je ta, da se preveč zanaša na spletne brskalnike. Prehodi, strežniki, vrata itd., ki se nenehno spreminjajo, so glavne kritične komponente, ki slabijo sistem REST. Moč in fleksibilnost sistema REST temeljita na uporabi URI-jev. URI-ji so samo imena in naslovi virov. Na začetku je splet imel tri glavne komponente, in sicer HTML- jezik, protokol HTTP in URI, ki se je pozneje preimenoval v URL.

REST zahteva, da si ustvarimo omrežni vmesnik s povezavo z URI in XML-raziskavami in da določimo prekinitev med vmesnikom (kateri je REST-orientirana) in implementacijo (katera je objektno-orientirana).

Delovanja REST:



Slika 6: Arhitektura REST

REST ni standard, kajti W3C ni izdalo nobene specifikacije o tem. Tudi nobena orodja se ne prodajajo s strani Microsofta, IBM-a, Sun-a itd; to pa zato, ker je REST le arhitektura. Kljub temu da REST ni standard, predpisuje uporabo standardov, kot so HTTP, URL, XML, HTML, JPG, TEXT/HTML itd..

Splet je primer sistema REST. Vse te spletne storitve, ki smo jih uporabljali leta nazaj (storitev naročanja knjig, iskalne storitve ...), temeljijo na REST arhitekturi. Ko govorimo o REST, lahko rečemo, da ima njegov pristop prednost v ceni same arhitekture REST-a in implementaciji storitev.

3.9 WOA

WOA opisuje jedro določenih spletnih protokolov kot HTTP in planira XML kot najbolj dinamičen, stopenjski pristop k spletnim storitvam. Edina prava razlika med SOA in konceptom WOA je v tem, da WOA zagovarja REST-arhitekturo in je zelo popularen in preprost protokol HTTP. Eden od glavnih ustvarjalcev WOA je tudi Roy Fielding.

Generalno WOA-odjemalec temelji na preizkušenih stvareh, ki temeljijo na tako imenovani dobri praksi. Kar pomeni izogibanje avtomatičnemu ustvarjanju, zmanjševanje serijskih abstrakcij med storitvami in odjemalci, periodično razumevanje programiranja in zelo dobro razumevanje dogajanja preko omrežja. V največ primerih ugotavljamo, da WOA najbolje deluje, ko uporablja REST s HTTP-jem.

3.10 MySQL

MySQL je sistem za upravljanje s podatkovnimi bazami. MySQL je odprtokodna implementacija relacijske podatkovne baze, ki za delo s podatki uporablja jezik SQL.

MySQL deluje na principu odjemalec - strežnik, pri čemer lahko strežnik namestimo kot sistem, porazdeljen na več strežnikov. Obstaja veliko število odjemalcev, zbirk ukazov in programskih vmesnikov za dostop do podatkovne baze MySQL.

Lastnosti:

- Napisan je v jezikih C in C++.
- Deluje na več različnih operacijskih sistemih (Windows, Linux,...).
- Za prenosljivost med operacijskimi sistemi uporablja GNU Automake, Autoconf in Libtool.
- Obstojajo programski vmesniki za C, C++, Eiffel, Java, Perl, PHP, Python, Ruby in Tcl.

- Bistvene programske niti mreže so večopravilne. Zna izkoristiti toliko centralnih procesnih enot, kolikor jih je na voljo.
- Uporablja zelo hitre diskovne tabele MyISAM, s stiskanjem indeksov.
- Programska koda je preizkušena z aplikacijama Purify in Valgrind.

3.11 SSH

Secure Shell (SSH) je omrežni protokol za varno podatkovno komunikacijo, oddaljene shell storitve ali ukazne izvršitve in ostale varne omrežne storitve med dvema v mrežo povezanima računalnikoma, ki se povežeta preko varnega kanala preko nezaščitenega omrežja: strežnik in odjemalec (na njih sta SSH strežnik in SSH odjemalec programska oprema). Specifikacije protokola ločijo dve glavni verziji: SSH-1 in SSH-2.

Najbolj znana aplikacija protokola je za dostop do shell računov na Unix operacijskih sistemih. SSH je bil načrtovan kot zamenjava za Telnet in ostale nezaščitene oddaljena shell protokole, kot so to Berkley rsh in rexec protokola, ki pošiljata informacije, pomembna gesla v enostavni tekstovni obliki tako, da so dovzetni za prestrezanje in razkritje pri uporabi paketne analize. Enkripcija, ki se uporablja pri SSH je namenjena, da omogoči zaupnost in integriteto podatkov skozi nezaščiteno omrežje kot je to Internet.

SSH uporablja kriptografijo z javnim ključem, da ugotovi pristnost oddaljenega računalnika in uporabnika, če je to potrebno. Vsakdo lahko naredi par različnih ključev, ki se ujemata (javni in zasebni). Javni ključ se nahaja na vseh računalnikih, ki morajo dovoliti dostop lastniku privatnega ključa, ki se z njim ujema (privatni ključ lastnika je skrivnost). Medtem ko je pristnost določena na osnovi privatnega ključa, se ključ sam nikoli ne prenese preko omrežja med postopkom identifikacije.

SSH samo preveri ali ista oseba, ki ponuja javni ključ tudi lastnik privatnega ključa, ki se ujemata. V vseh verzijah SSH-ja je pomembna resničnost neznanih javnih ključev, združitve javnih ključev z identitetami preden jih sprejme kot veljavne. Če bi sprejel javni ključ brez veljavnosti, bi nepooblašчени uporabniki izpadli kot pooblašчени.

SSH se tipično uporablja za dostop do oddaljene naprave in izvrševanje ukazov, podpira pa tudi tuneliranje, forwardiranje TCP portov in X11 povezav; lahko prenaša datoteke z uporabo povezanih SSH datotek (SFTP) ali protokolov varnih kopij (SCP). SSH uporablja strežnik-odjemalec model.

Standardni TCP port 22 je bil dodeljen za kontaktiranje SSH strežnikov, čeprav ga administratorji pogosto spremenijo v ne standardni port, kot dodatna mera varnosti.

Program SSH odjemalca se tipično uporablja za vzpostavitev povezav do SSH daemon, da sprejmejo oddaljene povezave. Oba sta pogosto prisotna na večini modernih operacijskih sistemov: Mac OS X, večini GNU/Linux, OpenBSD, FreeBSD, Solaris in OpenVMS.

SSH je pomemben pri računanju v oblaku za reševanje problemov povezovanja, izogibanju varnostnih zadev razkritja virtualnih naprav na osnovi oblaka direktno na Internet. SSH tunel lahko omogoča varno pot skozi Internet, skozi požarni zid do virtualne naprave.

3.12 PPP

Point-to-Point Protocol (PPP) je protokol podatkovne povezave, ki se pogosto uporablja pri vzpostavljanju direktne povezave med dvema omrežnima točkama. Lahko omogoča pristnost povezave, enkripcijo prenosa (z uporabo ECP, RFC 1968), in kompresijo.

PPP se uporablja pri mnogih tipih fizičnih omrežji vključno z serijskim kablom, telefonsko linijo, mobilno telefonijo, specializiranimi radijskimi povezavami in optičnimi povezavami, kot je to SONET. PPP se tudi uporablja pri povezavah dostopa do Interneta (sedaj se reklamira kot "široka povezava"). Ponudniki internetnih storitev najpogosteje uporabljajo dve obliki PPP za vzpostavitev DSL povezave med uporabniki in Internet omrežjem:

- Point-to-Point Protocol over Ethernet (PPPoE) IN
- Point-to-Point Protocol over ATM (PPPoA).

PPP se pogosto uporablja kot sloj protokola podatkovne povezave za povezavo skozi sinhrona in asinhrona omrežja. PPP je bil načrtovan za sodelovanje z mnogimi sloji omrežnih protokolov kot so to: Internet Protokol (IP), TRILL, Novell's Internetwork Packet Exchange (IPX), NBF in AppleTalk.

Link Control Protocol (LCP) elegantno inicializira in zaključuje povezave, dovoljuje gostiteljem, da se pogajajo o možnostih povezave. LCP je integralen del PPP in je definiran v isti specifikaciji standarda. LCP omogoča automatico konfiguracijo vmesnikov na vsakem koncu in za izbiro pristnosti. LCP protokol deluje na vrhu PPP in zato mora biti osnovna PPP povezava vzpostavljena preden se lahko LCP skonfigurira.

Ko je povezava vzpostavljena se lahko dodatna omrežna konfiguracija pojavi. Najpogosteje se uporablja Internet Protocol Control Protocol (IPCP), čeprav so bili Internetwork Packet Exchange Control Protocol (IPXCP) in AppleTalk Control Protocol (ATCP) enkrat popularni. Internet Protocol Version 6 Control Protocol (IPv6CP) bo videl razširjeno uporabo v prihodnosti, ko bo zamenjal IPv4 kot dominantni layer-3 protokol.

4 GPRS prenos podatkov

GPRS je nadgradnja GSM-omrežja. Sistem GPRS je neke vrste dodatek k klasičnemu GSM-sistemu. GPRS je infrastrukturna storitev, ki omogoča prenos podatkov. Prenos podatkov poteka drugače – preko paketno-komutiranega omrežja. Nastal je iz želje povezati Internet in obstoječe mobilno omrežje.

Razvoj GSM-omrežja do 2,5 G (angl. *Second Generation*) je pripeljal do predstavitve GPRS-tehnologije. GPRS preskrbuje GSM s podatkovnim oddajnim vmesnikom in IP, ki služi kot podlaga omrežja. Nadalje se je GPRS razvil v EDGE, ki zmore prenesti trikrat več podatkov, a pri satelitskem sledenju ni potrebe po tolikšnih kapacitetah prenosa. Standardizacija GPRS se je začela leta 1994, ETSI je GPRS leta 1997 potrdil in postavil na trg – izdaja '97. Nekaj popravkov pa je doživela v izdaji '98.

Zakaj GPRS? GPRS sistem ima zelo kratek dostop časa v omrežje in se obračunava zgolj po številu prenesenih podatkov. Uporablja od enega do osmih kanalov odvisno od mobilne postaje. Avtomatično so dodeljeni mobilni postaji, kadar gre bodisi za pošiljanje ali prejemanje podatkov. Kanali se dodeljujejo sproti glede na potrebe uporabnika in zmožnosti omrežja. Omrežje GPRS ima posebej rezervirane kanale za nalaganje in pošiljanje podatkov, kar omogoča različne

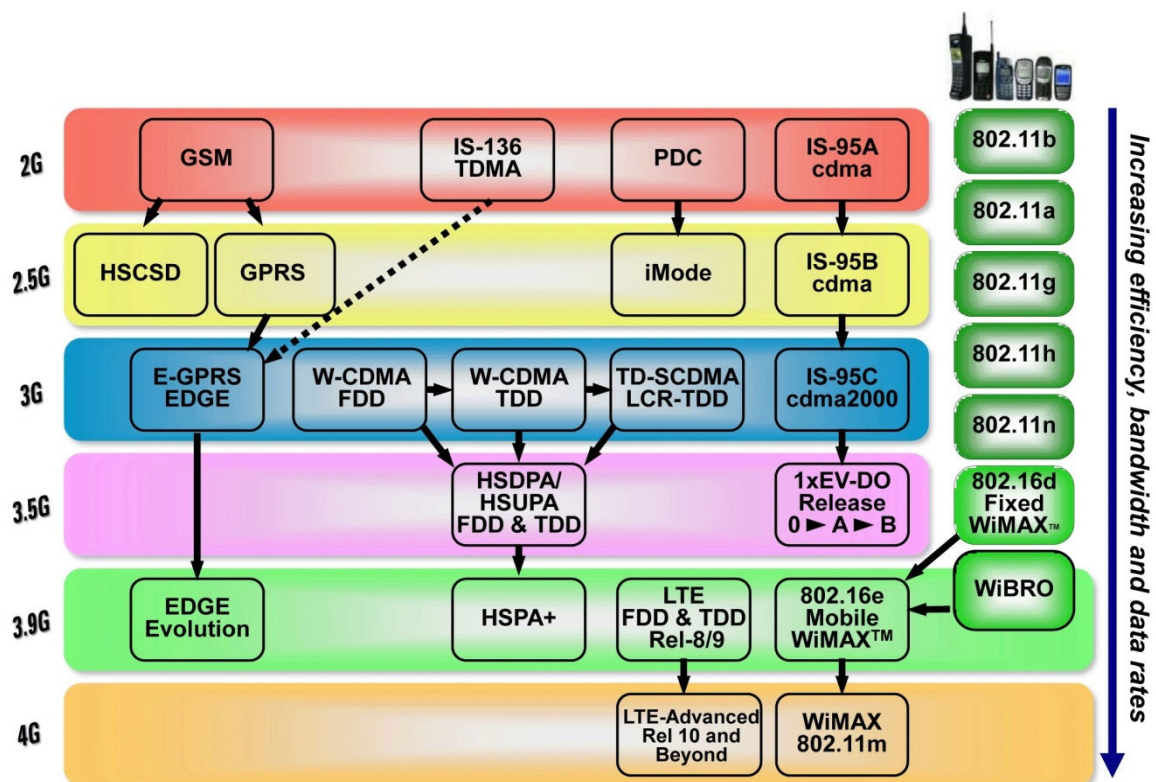
kapacitete prenosov. Seveda mobilna postaja omogoča tudi pogovor med samim prejemanjem ali pošiljanjem podatkov. GPRS podpira tudi SMS.

Kodne sheme omogočajo največje podatkovne hitrosti v enem časovnem oknu 20 kbit/s (pred kanalskim kodiranjem). 20 kbit/s na časovno okno in osem časovnih oken nam da skupno največjo hitrost prenosa podatkov. Teoretični maksimum prenosa v sistemu je torej 160 kbit/s ob uporabi vseh osmih kanalov, brez popravljanja napak. Seveda se število kanalov prilagaja potrebi po prenosu podatkov, tako da je lahko število kanalov za prejetje in nalaganje različno. Po navadi je tudi asimetrično, kar v praksi pomeni, da več prejemamo, kot nalagamo.

Za primer hitrosti prenosov bomo prikazali največjega slovenskega ponudnika mobilne telefonije Mobitel, d. d, ki je ob enem bil tudi prvi tovrstni ponudnik. Storitve Paketni prenos podatkov omogoča stalno povezavo z internetom s hitrostjo prenosa do 7,2 Mb/s (HSDPA), do 1,4 Mb/s (HSUPA), 384 kb/s (UMTS), do 236,8 kb/s (EDGE) oziroma do 82,4 kb/s (GPRS). Prednosti paketnega prenosa podatkov v sistemih UMTS in GPRS so ("Zasebni uporabniki" :

- storitve se obračunavajo glede na količino prenešenih podatkov (plača se informacije, ne čas);
- uporabnik je na omrežje neprestano priključen, kar pomeni hitrejši prenos;
- enostaven, hiter in varen dostop do informacijskih aplikacij;
- izboljšana produktivnost in kakovost dela;
- Mobitel naročniku pomaga pri vseh fazah priprave podatkovnega priklopa.

Družba Mobitel je nacionalni operater mobilnih telekomunikacij, ki upravlja in tudi sam izgrajuje mobilna omrežja. Mobitel je svoje delovanje začel z analognim omrežjem NMT leta 1991, pet let kasneje pa je uvedel digitalni sistem GSM. Leta 2000 je sledila uvedba tehnologije HSCSD z možnostjo hitrega prenosa podatkov, ki je že napovedovala prihod podatkovnega prenosa podatkov GPRS leta 2001. Mobitel je svojim uporabnikom med prvimi operaterji leta 2003 predstavil tretjo generacijo mobilnih telekomunikacij UMTS, ki evolucijsko predstavlja pomemben mejnik razvoja sodobnih tehnologij ("Mobilne generacije".



Slika 7: Razvoj mobilnih sistemov 1990-2011

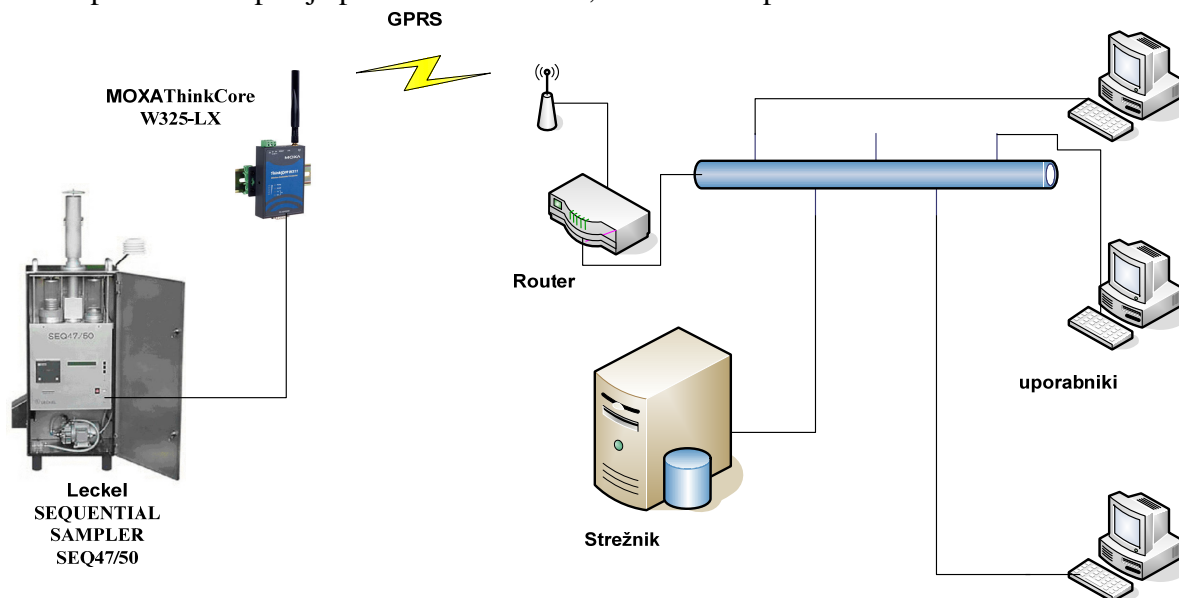
Paketni prenos podatkov oz. GPRS je v primerjavi s HSCSD, ki je še vedno klasična telekomunikacijska storitev, s katero imamo med dvema točkama vzpostavljeno stalno fizično povezavo, tudi ko ni dejanskega prenosa podatkov, bolj internetno usmerjena storitev. V GPRS je uporabljena vrsta internetnih standardov. Pomembna prednost sistema GPRS je prenašanje podatkov v paketih, saj je tako omrežje obremenjeno samo med prenašanjem podatkov. Radijske vire si deli več uporabnikov. Vedno, ko so na voljo prosta časovna okna (ki jih ne zaseda tokokrogovno komutirana zveza), lahko GPRS le-ta uporablja za prenos paketov. Prepletanje z globino 4 pomeni manj zakasnitev v primerjavi z GSM, kjer je globina prepletanja 19. Storitve se obračunava na podlagi dejanske količine prenesenih podatkov in ne na podlagi časa trajanja podatkovne zveze.

GPRS je pomemben korak pri združevanju mobilne telefonije in Interneta, saj je v GPRS uporabljena vrsta internetnih standardov. Ena izmed podobnosti je že v paketnem načinu prenosa podatkov. Poleg tega dobi tudi pri GPRS tehnologiji vsak uporabnik s telefonsko številko še posebno začasno IP-številko, ki je značilna za internetni protokol. Uporabnik se prijavi v omrežje in ostane ves čas priključen (angl. *always on-line*), kar omogoča takojšnji dostop do storitev in s tem varčevanje s časom ("GPRS").

GPRS-terminali se delijo v tri razrede: A, B in C. Terminali razreda A zagotavljajo GPRS- in ostale GSM-storitve sočasno, kar pomeni, da lahko poteka glasovna zveza simultano s paketnim prenosom podatkov. Druga možnost so terminali razreda B, ki omogočajo le eno aktivnost v določeni časovni enoti – prenos podatkov se mora recimo med pogovorom ustaviti in obratno. Razred C pa dopušča samo prenos podatkov prek GPRS ali samo prenos govora.

5 Prenos podatkov iz merilne naprave do uporabnikov

Podatki iz merilne postaje Leckel SEQUENTIAL SAMPLER SEQ47/50 se enkrat dnevno prenesejo na SD kartico MOXA ThinkCore W325 LX računalnika. Moxa nato vzpostavi GPRS povezavo in pošlje podatke na strežnik, ki so vidni uporabnikom v HTML obliki.



Slika 8: Arhitektura prenosa podatkov iz merilne naprave do uporabnikov

Moxa je preko RS232 serijskega porta povezana z Leckel merilno napravo, preko katerega se prenesejo merilni podatki v sledeči obliki:

```
10/0057;10/0027;2_0_0;5_0_1;
9; 2.3m3/h; 55.13; 51.75; ; +8.5; +14; +18; 981; 23h58min;23:58;28.03.10
10; 2.3m3/h; 55.14; 51.43; ; +10.3;+15.2;+17.4; 982; 23h58min;23:58;29.03.10
11; 2.3m3/h; 55.14; 51.17; ; +10.3;+12.9;+15.6; 976; 23h58min;23:58;30.03.10
12; 2.3m3/h; 55.13; 51.69; ; +7.2;+12.5;+16.4; 974; 23h58min;23:58;31.03.10
13; 2.3m3/h; 55.13; 52.22; ; +6.3;+12.7;+16.4; 981; 23h58min;23:58; 1.04.10
14; 2.3m3/h; 50.41; 48.31; ; +4.5; +12;+15.8; 986; 21h55min;23:58; 2.04.10
1; 2.3m3/h; 55.13; 52.59; ; +5.8;+11.9;+15.7; 988; 23h58min;23:58; 3.04.10
2; 2.3m3/h; 55.13; 51.74; ; +10;+13.3;+16.4; 986; 23h58min;23:58; 4.04.10
3; 2.3m3/h; 55.13; 52.1; ; +8.3;+12.2;+16.2; 986; 23h58min;23:58; 5.04.10
1; 2.3m3/h; 55.17; 52.12; ; +9.9;+12.9; +15; 992; 23h59min;23:58;20.03.10
2; 2.3m3/h; 55.13; 51.9; ; +10.4;+12.4;+15.2; 990; 23h58min;23:58;21.03.10
3; 2.3m3/h; 55.13; 52.06; ; +9.8;+11.6;+14.8; 991; 23h58min;23:58;22.03.10
4; 2.3m3/h; 55.13; 51.74; ; +10.7;+13.6;+17.8; 988; 23h58min;23:58;23.03.10
5; 2.3m3/h; 55.14; 51.76; ; +10.3;+14.6;+17.6; 987; 23h58min;23:58;24.03.10
6; 2.3m3/h; 55.13; 51.36; ; +12.3;+14.4;+16.4; 987; 23h58min;23:58;25.03.10
7; 2.3m3/h; 55.14; 51.35; ; +11.9;+13.5;+15.1; 985; 23h58min;23:58;26.03.10
8; 2.3m3/h; 55.13; 51.52; ; +9.5;+13.2;+16.1; 979; 23h58min;23:58;27.03.10
---;-----;-----;-----;-----;-----;-----;-----;-----;-----0
```

Te podatke Moxa pretvori v sledečo XML obliko:

```
<?xml version="1.0"?>
<resultset statement="select * from measurement order by stopDate DESC LIMIT 1
" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <row>
    <field name="MeasurementId">9</field>
    <field name="Filter">12</field>
    <field name="FlowRate">2.3</field>
    <field name="SamplingTime">23:58:00</field>
    <field name="StopTime">23:58:00</field>
    <field name="StopDate">2031-03-10</field>
    <field name="Standard">51.69</field>
    <field name="Operating">55.13</field>
    <field name="Tam">7.2</field>
    <field name="Tfm">12.5</field>
    <field name="Tim">16.4</field>
    <field name="Pam">974</field>
    <field name="CloggedFilter">0</field>
    <field name="WriteDateTime">2011-06-28 20:32:10</field>
  </row>
</resultset>
```

Merilni podatki vsebujejo sledeče informacije:

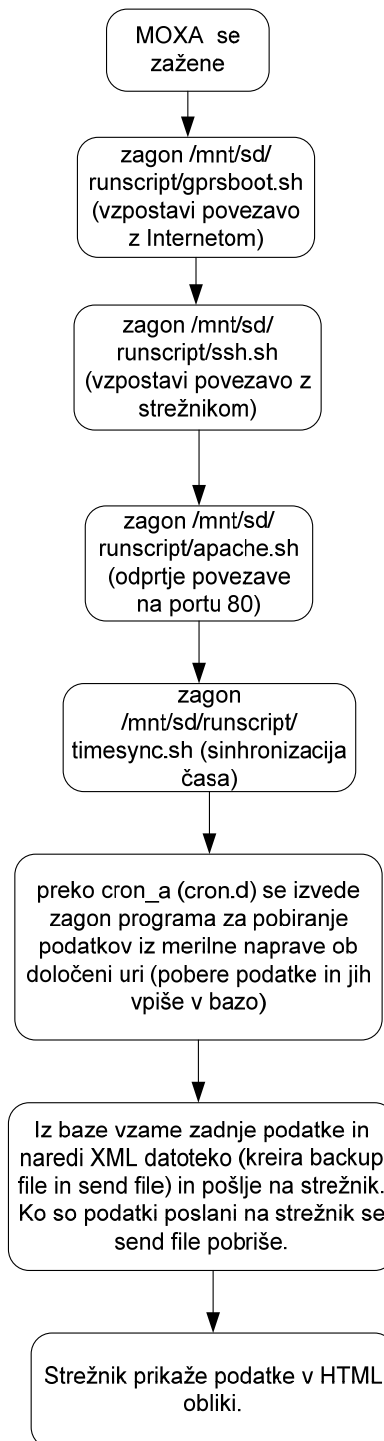
- Identifikator meritve,
- Številka filtra,
- Pretok zraka skozi vzorec,
- Čas vzorčenja,
- Čas končanja vzorčenja,
- Datum končanja vzorčenja,
- Preračunana vrednost volumenskega pretoka,
- Volumenski pretok m^3 / h ,
- Povprečna temperatura ambienta med vzorčenjem (Tam),
- Povprečna temperatura filtra med vzorčenjem (Tfm),
- Povprečna temperatura znotraj vzorčevalnika med vzorčenjem (Tim),
- Povprečen tlak ambienta med vzorčenjem (Pam),

XML datoteka se nato pošlje preko GPRS modema na strežnik in je vidna uporabnikom v HTML obliki (slika spodaj).



Slika 9: RAFAL spletna stran

5.1 Diagram poteka prenosa podatkov iz merilne naprave do strežnika



6 Postavitev sistema

Ko priklopimo Moxa W325 na napajanje, potrebuje 30-60 sekund, da se operacijski sistem zažene. Zelena LED lučka se ne vklopi dokler ni operacijski sistem pripravljen.

6.1 Povezava Moxa W325 z računalnikom

Obstajata dva načina povezave z osebnim računalnikom; preko serijskega konzolnega porta ali z Telnet-om preko omrežja.

- preko serijskega konzolnega porta

Serijski konzolni port daje uporabnikom pripraven način povezave do W325. Ta metoda je predvsem uporabna kadar uporabljamo računalnik prvič in kadar ne poznamo katerega od IP naslovov. Uporabite nastavitve serijskega konzolnega porta kot so zapisani v tabeli spodaj:

Baudrate	115200 bps
Parity	None
Data bits	8
Stop bit	1
Flow Control	None
Terminal	VT100

Enkrat ko je povezava vzpostavljena se pojavi naslednje okno:

```

PComm Terminal Emulator - COM1,115200,None,8,1,VT100
Profile Edit Port Manager Window Help
714 root      428 S    /sbin/getty 115200 ttyS1
715 root      628 R    ps
root@Moxa: /#
Moxa login: root
Password:

#####          #####          #####          #####          #####          ##
###          #####          ###          ###          #####          #####          ###
###          ###          ###          ###          ###          ##          ###
###          #####          ##          ##          ###          #          #####
#####          #          ##          ##          ##          ##          ##          ##
###          #          ##          ##          ##          ##          ##          ##
##          #          ##          ##          ##          ##          ##          ##
##          #          ##          ##          ##          ##          ##          ##
##          #          ##          ##          ##          ##          ##          ##
##          #          ##          ##          ##          ##          ##          ##
##          #          ##          ##          ##          ##          ##          ##
#####          #          #####          #####          #####          #####          #####
#####          #          #####          #####          #####          #####          #####

For further information check:
http://www.moxa.com/

root@Moxa: ~# |
State: OPEN  CTS DSR RT DCD Got Break Signal
    
```

Za prijavo vtipkajte Login name in password kot je zahtevano. Privzeto sta obe vrednosti **root**.

Login: root
Password:root

- **Telnet konzola**

Če poznate vsaj eden izmed IP naslovov in Netmask lahko uporabite Telnet, za povezavo do W325 konzolnega programa. Privzeto nastavljeni IP naslova in Netmaski za vsak izmed dveh portov sta podana spodaj:

	Default IP Address	Netmask
LAN	192.168.3.127	255.255.255.0

Uporabite prekrižani Ethernet kabel za direktno povezavo med vašim osebnim računalnikom in W325. Najprej morate spremeniti IP naslov in Netmask na svojem računalniku tako, da je vaš računalnik na istem subnet-u kot W325. Na primer, če se želite povezati na LAN lahko nastavite svoj IP naslov računalnika na 192.168.3.126 in Netmask na 255.255.255.0.

Za povezavo na hub ali switch, ki sta povezana na vaš lokalni LAN uporabite naravnost-skozi Ethernet kabel. Za prijavo vtipkajte Login name in password kot je zahtevano. Privzeto sta obe vrednosti **root**.

Login: root
Password:root



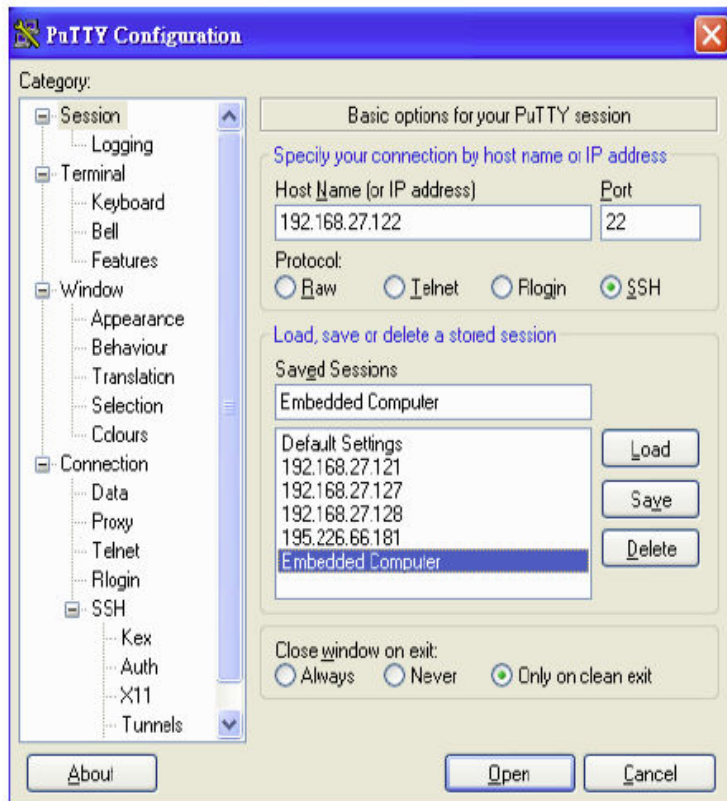
Sedaj lahko nadaljujete z konfiguracijo omrežnih nastavitev ciljnega računalnika ko pridete do ukazne vrstice.

- **SSH konzola**

W325 podpira SSH konzolo, da omogoči uporabnikom boljše varnostne pogoje.

- **Windows uporabniki**

Kliknite na povezavo: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>, da dobite brezplačno programsko opremo PuTTY za vzpostavitev SSH konzole za W325 v Windows okolju. Sledeča slika prikazuje preprost primer konfiguracije, ki je potrebna.



- **Linux uporabniki**

Na Linux računalniku uporabite »ssh« ukaz za dostop do W325 ukazuje vrstice preko SSH.
#ssh 192.168.3.127

Izberite Yes, da zaključite z povezavo.

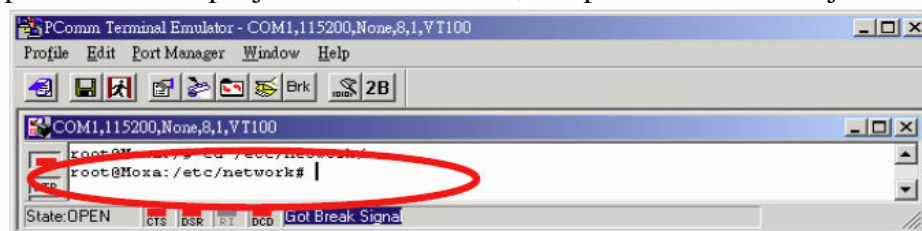
```
[root@localhost root]# ssh 192.168.3.127
The authenticity of host '192.168.3.127 (192.168.3.127)' can't be established.
RSA key fingerprint is 8b:ee:ff:84:41:25:fc:cd:2a:f2:92:8f:cb:1f:6b:2f.
Are you sure you want to continue connection (yes/no)? yes
```

6.2 Konfiguracija Ethernet vmesnika

Omrežne nastavitve W325 lahko spreminjamo z serijsko konzolo ali online preko mreže.

- **Modificiranje omrežnih nastavitvev preko serijske konzole**

Sledite navodilom iz prejšnjega poglavja za dostop do konzolnega programa preko serijskega porta in nato vtipkajte #cd /etc/network , da spremenite direktorije.



Vtipkajte #vi interfaces, da uporabite vi editor za nastavitvev omrežne datoteke Lahko konfigurirate Ethernet porte W325 za statične ali dinamične (DHCP) IP naslove.

Statični IP naslovi:

Kot je prikazano spodaj, je potrebno spremeniti 4 omrežne naslove: **address**, **network**, **netmask**, in **broadcast**. Privzeti IP naslovi so: 192.168.3.127 za LAN z netmaskami 255.255.255.0.

Dinamični IP naslovi:

Privzeto, je W325 nastavljen na statične IP naslove. Za nastavitve enega ali obeh LAN portov po dinamični zahtevi IP naslova zamenjajte **static** z **dhcp** in nato zbrisite **address**, **network**, **netmask**, in **broadcast** vrstice.

Default Setting for LAN	Dynamic Setting using DHCP
<pre>iface eth0 inet static address 192.168.3.127 network: 192.168.3.0 netmask 255.255.255.0 broadcast 192.168.3.255</pre>	<pre>iface eth0 inet dhcp</pre>

Ko so boot nastavitve LAN vmesnika spremenjene, vpišite naslednji ukaz, da takoj sktivate LAN nastavitve: `#/etc/init.d/networking restart`.

- **Modificiranje Ethernet nastavitve preko mreže**

IP nastavitve lahko aktiviramo preko mreže, ampak nove nastavitve se ne bodo shranile v flash ROM brez spremembe datoteke **file/etc/network/interfaces**.

Za primer, vtipkajte ukaz **#ifconfig eth0 192.168.1.1**, da spremenite IP naslov LAN-a na 192.168.1.1.

```
root@Moxa:~# ifconfig eth0 192.168.27.125
root@Moxa:~#
```

6.3 GPRS omrežja

W325 vključuje GSM/GPRS modul za brezžično komunikacijo. Modul lahko uporabimo za prenos podatkov preko GPRS omrežja.

- **Povezava na GPRS omrežje**

Preden se povežete na GPRS omrežje preverite ali je SIM kartica pravilno vstavljena in, da je antena priključena. SIM kartico vstavljamo kadar W325 ni priključen na el. Napajanje. W325 omogoča skriptne ukaze za hitro GPRS povezavo.

1. Za vzpostavitev povezave uporabite naslednji ukaz: **# gprscmd connect**
2. Povezava na GPRS traja samo nekaj sekund. Medtem ko se povezuje se v ukaznem oknu pojavi sledeči napis: **Connecting... Please wait a few seconds.**
3. Ko je W325 uspešno priključen na GPRS omrežje se pojavi naslednji napis: **Connected.**
4. Da preverite povezavo vpišite naslednje: **#ifconfig**

Pojaviti bi se moral ppp (Point-to-Point Protocol) vmesnik, ki izgleda takole:

```
ppp0  Link encap:Point-Point Protocol
      inet addr:221.120.21.171  P-t-P:192.168.111.111  Mask:255.255.255.255
      UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
      RX packets:20 errors:0 dropped:0 overruns:0 frame:0
      TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:3
      RX bytes:954 (954.0 B)  TX bytes:779 (779.0 B)
```

- **Prekinitev GPRS povezave**

Za končanje GPRS povezave vtipkajte sledeče: **#gprscmd disconnect**

6.4 Omogočanje in onemogočanje programov, ki se izvajajo v ozadju (Daemons)

Ponavadi ti programi ponujajo določene storitve kot so to spletni dostop, FTP in email storitve, itd. Sledeči programi se omogočijo ob zagonu W325:

snmpdSNMP Agent daemon
telnetdTelnet Server / Client daemon
inetdInternet Daemons
ftpd.....FTP Server / Client daemon
sshdSecure Shell Server daemon
httpdApache WWW Server daemon

Vtipkajte ukaz »ps«, da vidite kateri programi se trenutno izvajajo.


```

192.168.3.127 - PuTTY
root@Moxa:~# cd /etc
root@Moxa:/etc# ps
  PID  Uid      VmSize  Stat  Command
   1  root        532  S    init [3]
   2  root          SWN  [ksoftirqd/0]
   3  root          SW<  [events/0]
   4  root          SW<  [khelper]
  13  root          SW<  [kblockd/0]
  14  root          SW   [khubd]
  24  root          SW   [pdflush]
  25  root          SW   [pdflush]
  27  root          SW<  [aio/0]
  26  root          SW   [kswapd0]
 604  root          SW   [mtdblockd]
 609  root          SW   [pccardd]
 611  root          SW   [pccardd]
 625  root          SWN  [jffs2 gcd mtd3]
 673  root        500  S    /bin/inetd
 679  root       3004  S    /usr/bin/httpd -k start -d /etc/apache
 682  bin        380  S    /bin/portmap
 685  root       1176  S    /bin/sh --login
 690  root        464  S    /bin/snmpd
 694  nobody    3012  S    /usr/bin/httpd -k start -d /etc/apache
 695  nobody    3012  S    /usr/bin/httpd -k start -d /etc/apache
 696  nobody    3012  S    /usr/bin/httpd -k start -d /etc/apache
 697  nobody    3012  S    /usr/bin/httpd -k start -d /etc/apache
 698  nobody    3012  S    /usr/bin/httpd -k start -d /etc/apache
 701  root        352  S    /bin/reportip
 714  root       1176  S    -bash
 726  root        436  S    /bin/telnetd
 727  root       1180  S    -bash
 783  root        628  R    ps -ef
root@Moxa:/ect# █

```

Če želite videti vse programe, ki se izvajajo ob zagonu vtipkajte sledeče:

```
#cd /etc/rc.d/rc3.d
```

```
#ls
```

```

192.168.3.127 - PuTTY
root@Moxa:/etc/rc.d/rc3.d# ls
S19nfs-common S25nfs-user-serverS99showreadyled
S20snmpd      S55ssh
S24pcmcia    S99rmnlogin
root@Moxa:/etc/rc.d/rc3.d# █

```

```
#cd /etc/rc.d/init.d
```

Uredite skripto, da se izvede `/root/tcps2-release` in shranite v `tcps2` kot primer.

```
#cd /etc/rc.d/rc3.d
```

```
#ln -s /etc/rc.d/init.d/tcps2 S60tcps2
```

SxxRUNFILE pomeni sledeče

S: start the run file while linux boots up.

xx: število med 00-99. Manjše številke imajo večjo prioriteto.

RUNFILE: ime datoteke.

```

192.168.3.127 - PuTTY
root@Moxa:/etc/rc.d/rc3.d# ls
S19nfs-common S25nfs-user-serverS99showreadyled
S20snmpd      S55ssh
S24pcmcia    S99rmnologin
root@Moxa:/etc/rc.d/rc3.d# ln -s /root/tcps2-release S60tcps2
root@Moxa:/etc/rc.d/rc3.d# ls
S19nfs-common S25nfs-user-serverS99rmnologin
S20snmpd      S55ssh          S99showreadyled
S24pcmcia    S60tcps2
root@Moxa:/etc/rc.d/rc3.d# █

```

KxxRUNFILE pomeni sledeče

K: start the run file while linux shuts down or halts.

xx: število med 00-99. Manjše številke imajo večjo prioriteto.

RUNFILE: ime datoteke.

Za odstranitev programa, odstranite run datoteko iz **/etc/rc.d/rc3.d** datoteke z uporabo sledečega ukaza:

```
#rm -f /etc/rc.d/rc3.d/S60tcps2
```

6.5 Avtomatski zagon programa v Run nivoju (Run-Level)

Če želite, da se program začne izvajati pri Run nivoju spremenite **rc.local** datoteko na sledeč način:

```
#cd /etc/rc.d
```

```
#vi rc.local
```

```

192.168.3.127 - PuTTY
root@Moxa:~# cd /etc/rc.d
root@Moxa:/etc/rc.d# vi rc.local █

```

Nato uporabite vi, da odprete svojo aplikacijo. Uporabimo primer programa **tcps2-release** in ga dajte da se izvaja v ozadju.

```

192.168.3.127 - PuTTY
# !/bin/sh
# Add you want to run daemon
/home/tcps2-release &~

```

Omogočeni programi iz ozadja (daemons) bodo na voljo ob ponovnem zagonu sistema.

```

192.168.3.127 - PuTTY
root@Moxa:~# ps
  PID  Uid    VmSize  Stat Command
    1  root      532 S    init [3]
    2  root          SWN  [ksoftirqd/0]
    3  root          SW<  [events/0]
    4  root          SW<  [khelper]
   13  root          SW<  [kblockd/0]
   14  root          SW   [khubd]
   24  root          SW   [pdflush]
   25  root          SW   [pdflush]
   27  root          SW<  [aio/0]
   26  root          SW   [kswapd0]
  604  root          SW   [mtdblockd]
  609  root          SW   [pccardd]
  611  root          SW   [pccardd]
  625  root          SWN  [jffs2_gcd_mtd3]
  673  root      500 S    /bin/inetd
  674  root     1264 S    /root/tcps2-release
  679  root     3004 S    /usr/bin/httpd -k start -d /etc/apache
  682  bin       380 S    /bin/portmap
  685  root     1176 S    /bin/sh --login
  690  root      464 S    /bin/snmpd
  694  nobody   3012 S    /usr/bin/httpd -k start -d /etc/apache
  695  nobody   3012 S    /usr/bin/httpd -k start -d /etc/apache
  696  nobody   3012 S    /usr/bin/httpd -k start -d /etc/apache
  697  nobody   3012 S    /usr/bin/httpd -k start -d /etc/apache
  698  nobody   3012 S    /usr/bin/httpd -k start -d /etc/apache
  701  root      352 S    /bin/reportip
  714  root     1176 S    -bash
  726  root      436 S    /bin/telnetd
  727  root     1180 S    -bash
  783  root      628 R    ps -ef
root@Moxa:~#

```

6.6 Nastavitev časa v sistemu

- **Ročna nastavitev časa**

W325 ima dve časovni nastavitvi. Ena je sistemski ura in druga je RTC (Real Time Clock) ura katero sinhronizira strojna oprema W325. Uporabite `#date` ukaz za poizvedbo o trenutnem sistemskem času ali ga nastavite na novo. Uporabite `#hwclock` ukaz za poizvedbo trenutnega RTC časa ali ga nastavite na novo. Sledeči ukaz uporabite za nastavitev sistema časa: **`#date MMDDhhmmYYYY`**

MM = Month

DD = Date

hhmm = hour and minute

YYYY = Year

Za nastavitev RTC časa uporabite sledeči ukaz: **`#hwclock -w`**

Sledeča slika prikazuje kako posodobiti sistemski čas in RTC čas.


```

192.168.3.127 - PuTTY
root@Moxa:~# date
Fri Jun 23 23:30:31 CST 2000
root@Moxa:~# hwclock
Fri Jun 23 23:30:35 2000 -0.557748 seconds
root@Moxa:~# date 120910002004
Thu Dec 9 10:00:00 CST 2004
root@Moxa:~# hwclock -w
root@Moxa:~# date ; hwclock
Thu Dec 9 10:01:07 CST 2004
Thu Dec 9 10:01:08 2004 -0.933547 seconds
root@Moxa:~# █

```

- **NTP klient**

W325 ima vgrajen NTP (Network Time Protocol) klient, ki se uporablja za inicializacijo časovne zahteve do oddaljenega NTP strežnika. Uporabite **#ntpdate <this client utility>** za posodobitev systemskega časa.

```
#ntpdate time.stdtime.gov.tw
#hwclock -w
```

```

10.120.53.100 - PuTTY
root@Moxa:~# date ; hwclock
Sat Jan 1 00:00:36 CST 2000
Sat Jan 1 00:00:37 2000 -0.772941 seconds
root@Moxa:~# ntpdate time.stdtime.gov.tw
9 Dec 10:58:53 ntpdate[207]: step time server 220.130.158.52 offset 155905087.9
84256 sec
root@Moxa:~# hwclock -w
root@Moxa:~# date ; hwclock
Thu Dec 9 10:59:11 CST 2004
Thu Dec 9 10:59:12 2004 -0.844076 seconds
root@Moxa:~# █

```

- **avtomatska nastavitve časa**

Primer skripte za periodično posodobitev systemskega časa

```
#!/bin/sh
ntpdate time.nist.gov # You can use the time server's ip address or domain
# name directly. If you use domain name, you must
# enable the domain client on the system by updating
# /etc/resolv.conf file.
hwclock --systohc
sleep 100 # Updates every 100 seconds. The sleeping time is 100 seconds. Change
# 100 to a larger number to update RTC less often.
```

Shranite skripto z uporabo kateregakoli imena datoteke, npr. fixtime.

Kako avtomatsko zagnati skripto ko se kernel ponovno zažene

Kopirajte primer skripte fixtime v direktorij **/etc/init.d**, in nato uporabite **chmod 755 fixtime**, da spremenite obliko skripte. Nato uporabite vi editor za spremembo datoteke **/etc/inittab**. Dodajte sledečo vrstico na dno datoteke:

```
ntp : 2345 : respawn : /etc/init.d/fixtime
```

Uporabite ukaz **#init q** za reinicializacijo kernela.

6.7 Cron-Program (Daemon) za izvajanje načrtovanih ukazov

Cron je storitev časovnega razvrščanja v Linux-u. Cron se prebudi vsako minuto in preveri konfiguracijsko datoteko crontab, da preveri ali je potrebno zagnati kakšen načrtovan ukaz v trenutni minuti. Crontab se nahaja v `/etc/cron.d` datoteki. Modificirajte datoteko `/etc/cron.d/crontab`, če želite ob določenem času zagnati svoje aplikacije. Crontab ima sledeči format:

mm	h	dom	mon	dow	user	command
min	hour	date	month	week	user	command
0-59	0-23	1-31	1-12	0-6 (0 is Sunday)		

Sledeči primer prikazuje kako uporabljati Cron:

Kako uporabiti Cron za posodobitev systemskega časa in RTC časa ob 8 uri vsak dan.

STEP1: Write a shell script named fixtime.sh and save it to /home/.

```
#!/bin/sh
ntpdate time.nist.gov
hwclock --systohc
exit 0
```

STEP2: Change mode of fixtime.sh

```
#chmod 755 fixtime.sh
```

STEP3: Modify /etc/cron.d/crontab file to run fixtime.sh at 8:00 every day.

Add the following line to the end of crontab:

```
* 8 * * * root /home/fixtime.sh
```

STEP4: Enable the cron daemon manually.

```
#/etc/init.d/cron start
```

STEP5: Enable cron when the system boots up.

Po privzetem načinu je Cron storitev onemogočena ob zagonu naprave.

6.8 Upravljanje z komunikacijami

6.8.1 Telnet/FTP

Poleg podpore Telnet klient/strežnik in FTP klient/strežnik, W325 tudi podpira SSH in sftp klient/strežnik. Za omogočanje ali onemogočanje Telnet/ftp strežnika morate najprej spremeniti datoteko `/etc/inetd.conf`.

- **omogočanje Telnet/ftp strežnika**

Sledeči primer dokazuje privzeto vsebino datoteke `/etc/inetd.conf`. Privzeto je omogočanje Telnet/ftp strežnika:

```
discard dgram udp wait root /bin/discard
discard stream tcp nowait root /bin/discard
telnet stream tcp nowait root /bin/telnetd
ftp stream tcp nowait root /bin/ftpd -l
```

- **onemogočanje Telnet/ftp strežnika**

Program onemogočimo tako, da pred prvi znak v vrstici dodamo »#«.

6.8.2 DNS

W325 podpira DNS klient (vendar ne podpira DNS strežnik). Da bi nastavili DNS klient je potrebno spremeniti sledeče tri konfiguracijske datoteke: **/etc/hosts**, **/etc/resolv.conf**, in **/etc/nsswitch.conf**.

/etc/hosts

To je prva datoteka, ki jo Linux sistem prebere, da razreši ime gostitelja in IP naslov.

/etc/resolv.conf

To je najpomembnejša datoteka, ki jo je potrebno urediti kadar uporabljate DNS za druge programe. Na primer, preden uporabite **#ntpdate time.nist.gov** za posodobitev systemske ure je potrebno dodati naslov DNS strežnika v datoteko. Vprašajte svojega omrežnega administratorja kateri DNS strežniški naslov bi uporabili. DNS strežniški IP naslov je specificiran z ukazom »imestrežnik«. Za primer, dodajte sledečo vrstico v **/etc/resolv.conf**, če je IP naslov DNS strežnika 168.95.1.1:

nameserver 168.95.1.1

```

10.120.53.100 - PuTTY
root@Moxa:/etc# cat resolv.conf
#
# resolv.conf This file is the resolver configuration file
# See resolver(5) .
#
#nameserver 192.168.1.16
nameserver 168.95.1.1
nameserver 140.115.1.31
nameserver 140.115.236.10
root@Moxa:/etc#

```

/etc/nsswitch.conf

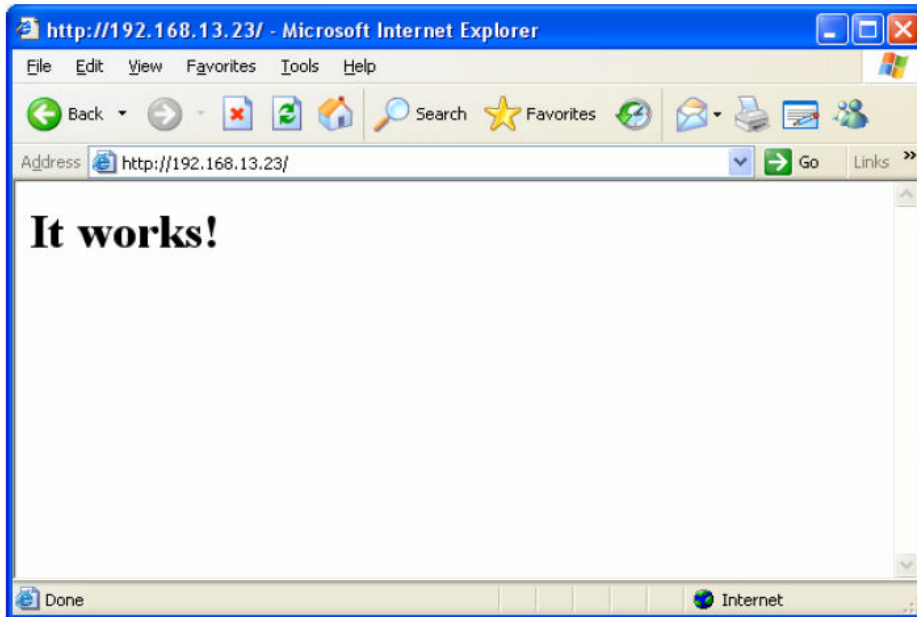
Ta datoteka definira sekvenco, da razreši IP naslov z uporabo **/etc/hosts** ali **/etc/resolv.conf**.

6.8.3 Spletna storitev-Apache

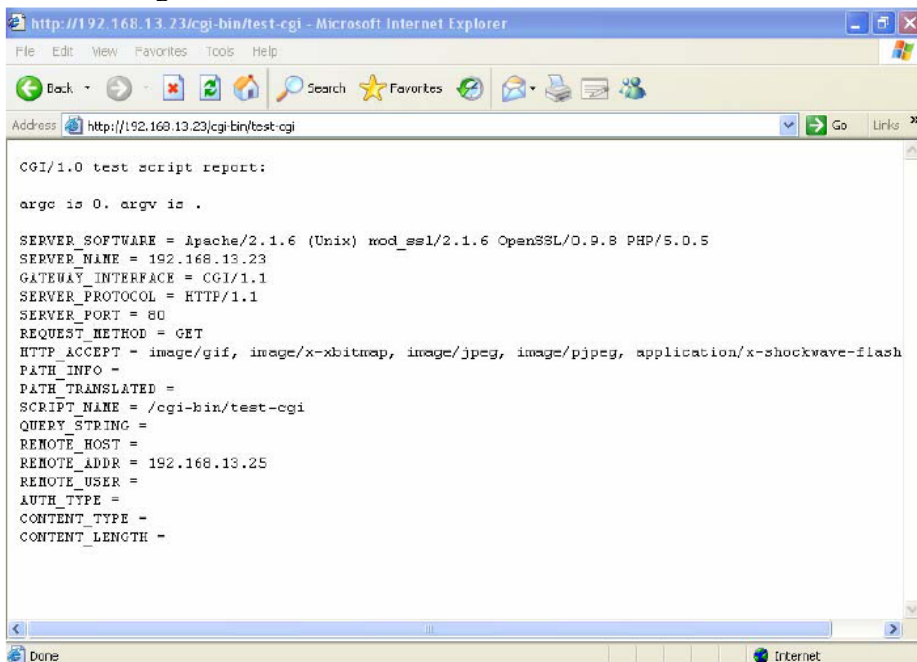
Glavna datoteka Apache spletnega strežnika je **/etc/apache/conf/httpd.conf**, z privzeto domačo stranjo locirano na **/home/httpd/htdocs/index.html**. Shranite svojo domačo stran v sledeči direktoriji:

/home/httpd/cgi-bin/

Predn modificirate domačo stran, uporabite brskalnik (kot sta to Microsoft Internet Explore ali Mozilla Firefox) na vašem osebnem računalniku, da preverite ali deluje Apache spletni strežnik. Vtipkajte LAN IP naslov v brskalnikovo naslovno vrstico, da odprete domačo stran. Če je privzeti IP naslov še kar aktiven vtipkajte **http://192.168.3.127** v naslovno vrstico brskalnika.



Za odprtje privzete CGI strani vtipkajte <http://192.168.3.127/cgi-bin/test-cgi> v naslovno vrstico vašega brskalnika.



6.8.4 Inštalacija PHP-ja za Apache spletni strežnik

W325 podpira PHP opcijo. Čeprav je PHP datoteka velikosti 3 MB ni inštalirana na W325. Če želite sami namestiti PHP na W325 najprej preverite, če je na flash ROM-u dovolj prostora.

1. **Korak:** Preverite ali je dovolj prostora. Spodnja slika prikazuje kako preverite da ima /dev/mtdblock3 vsaj 3MB prostora.

```
192.168.3.127 - PuTTY
root@Moxa:/bin# df -h
Filesystem      Size      Used Available Use% Mounted on
/dev/mtdblock2  8.0M      6.0M      2.0M    75% /
/dev/ram0        499.0k    17.0k     457.0k    4% /var
/dev/mtdblock3  6.0M      488.0k    5.5M     8% /tmp
/dev/mtdblock3  6.0M      488.0k    5.5M     8% /home
/dev/mtdblock3  6.0M      488.0k    5.5M     8% /etc
tmpfs           30.4M      0         30.4M    0% /dev/shm
root@Moxa:/bin#
```

2. **Korak:** Vtipkajte **upramdisk**, da dobite prostor na ram disku za shrambo paketa.

```
192.168.3.127 - PuTTY
root@Moxa:/bin# upramdisk
root@Moxa:/bin# df -h
Filesystem      Size      Used Available Use% Mounted on
/dev/mtdblock2  8.0M      6.0M      2.0M    75% /
/dev/ram0        499.0k    18.0k     456.0k    4% /var
/dev/mtdblock3  6.0M      488.0k    5.5M     8% /tmp
/dev/mtdblock3  6.0M      488.0k    5.5M     8% /home
/dev/mtdblock3  6.0M      488.0k    5.5M     8% /etc
tmpfs           30.4M      0         30.4M    0% /dev/shm
/dev/ram1       16.0M      1.0k     15.1M    0% /var/ramdisk
root@Moxa:/bin#
```

3. **Korak:** Naložite PHP paket iz CD-ROM-a. Paket lahko najdete v **CD-ROM/software/php/php.tar.gz**.

```
192.168.3.127 - PuTTY
root@Moxa:/bin# cd /mnt/ramdisk
root@Moxa:/mnt/ramdisk# ftp 192.168.27.130
Connected to 192.168.27.130.
220 (vsFTPD 2.0.1)
Name (192.168.27.130:root): root
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /tmp
250 Directory successfully changed.
ftp> bin
200 Switching to Binary mode.
ftp> get php.tar.gz
local: php.tar.gz remote: php.tar.gz
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for php.tar.gz (1789032 bytes).
226 File send OK.
1789032 bytes received in 0.66 secs (2.6e+03 Kbytes/sec)
ftp>
```

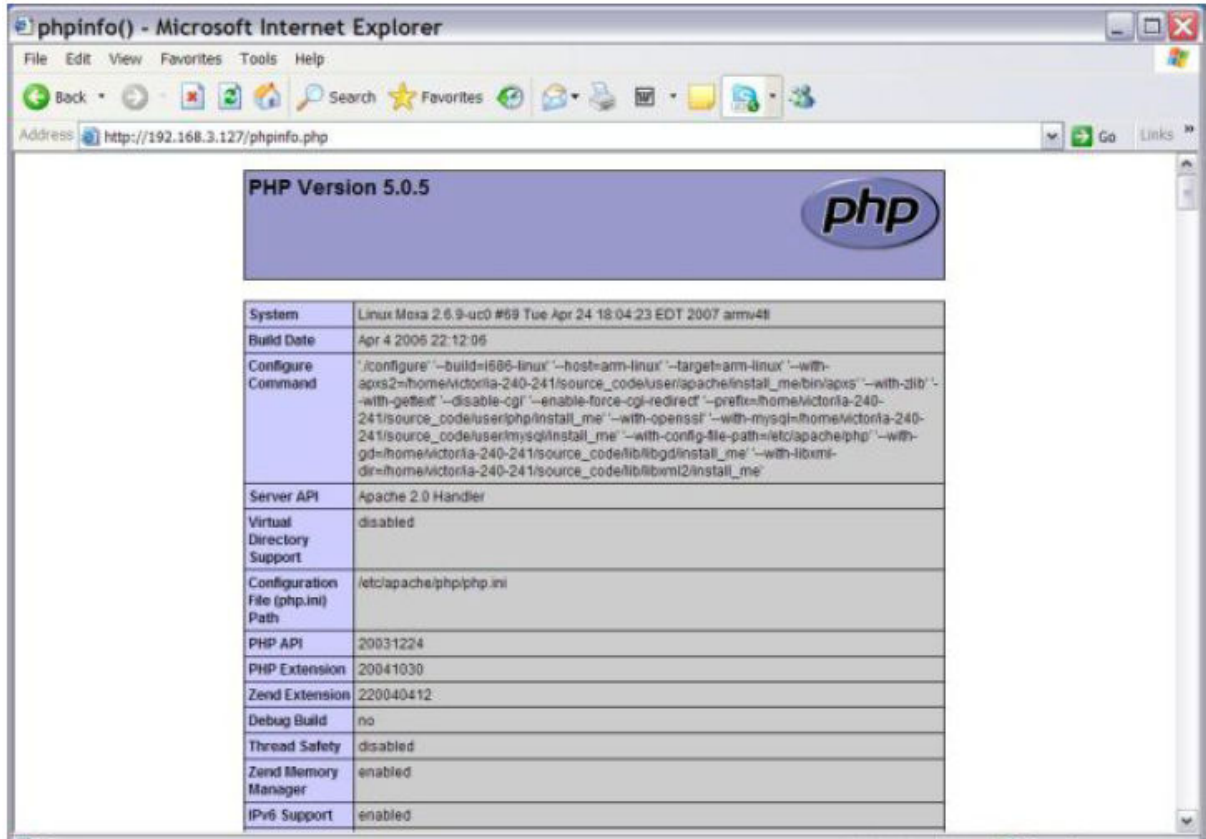
4. **Korak:** Razpakirajte paket. To naredite z sledečim ukazom: **tar xvzf php.tar.gz**.

```
192.168.3.127 - PuTTY
root@Moxa:/mnt/ramdisk# tar xvzf php.tar.gz
envvars
envvars.old
httpd.conf
httpd.conf.old
install.sh
lib
lib/libmysqlclient.so.15
lib/libpng.so.2
lib/libphp5.so
lib/libmysqlclient.so.15.0.0
lib/libgd.so
lib/libxml2.so.2.6.22
lib/libgd.so.2.0.0
lib/libjpeg.so
lib/libxml2.so.2
lib/libgd.so.2
php
php/php.ini
phpinfo.php
root@Moxa:/mnt/ramdisk#
```

5. **Korak:** Zaženite **install.sh** in izberite install php.

```
192.168.3.127 - PuTTY
root@Moxa:/mnt/ramdisk# ./install.sh
Press the number:
1. Install PHP package
2. Uninstall PHP package
3. Exit.
1
Start to install PHP. Please wait ...
Starting web server: apache.
PHP install sucess.
root@Moxa:/mnt/ramdisk#
```

6. **Korak:** Testirajte ga. Uporabite brskalnik za dostop do <http://192.168.3.127/phpinfo.php>.



Če želite izbrisati PHP iz flash ROM-a sledite korakom 2 do 5 vendar izberite uninstall možnost.

PRILOGE

Linux normal command utility collection

File manager

1. **cp** copy file
2. **ls** list file
3. **ln** make symbolic link file
4. **mount** mount and check file system
5. **rm** delete file
6. **chmod** change file owner & group & user
7. **chown** change file owner
8. **chgrp** change file group
9. **sync** sync file system, let system file buffer be saved to hardware
10. **mv** move file
11. **pwd** display now file directly
12. **df** list now file system space
13. **mkdir** make new directory
14. **rmdir** delete directory

Editor

1. **vi** text editor
2. **cat** dump file context
3. **zcat** compress or expand files
4. **grep** search string on file
5. **cut** get string on file
6. **find** find file where are there
7. **more** dump file by one page
8. **test** test file exist or not
9. **sleep** sleep (seconds)
10. **echo** echo string

Network

1. **ping** ping to test network
2. **route** routing table manager
3. **netstat** display network status
4. **ifconfig** set network ip address
5. **tracerout** trace route
6. **telnet** teletype network
7. **ftp** file transfer protocol

Process

1. **kill** kill process
2. **ps** display now running process

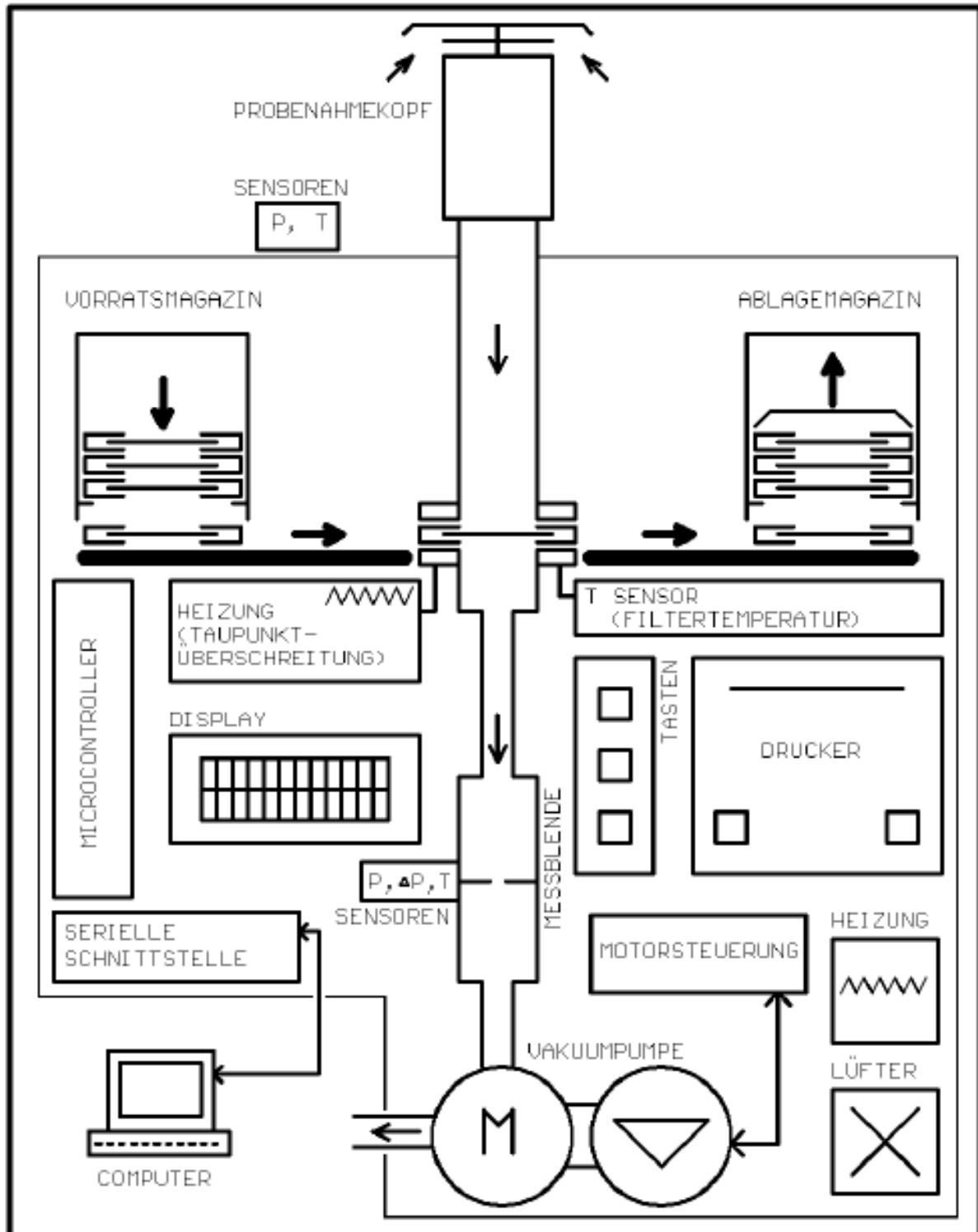
Other

1. **dmesg** dump kernel log message
2. **stty** to set serial port
3. **zcat** dump .gz file context
4. **mknod** make device node
5. **free** display system memory usage
6. **date** print or set the system date and time
7. **env** run a program in a modified environment
8. **clear** clear the terminal screen

- | | |
|-------------------------|---|
| 9. reboot | reboot / power off/on the server |
| 10. halt | halt the server |
| 11. du | estimate file space usage |
| 12. gzip, gunzip | compress or expand files |
| 13. hostname | show system's host name |

Moxa special utilities

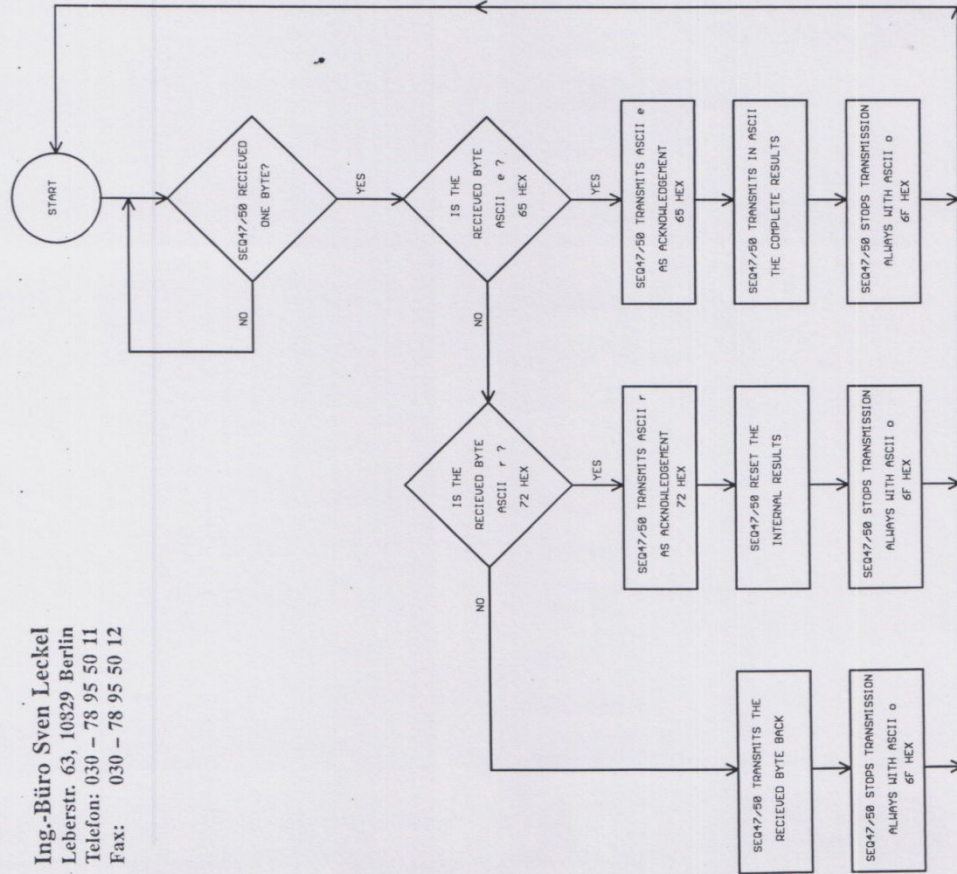
- | | |
|----------------------------|------------------------------------|
| 1. kversion | show kernel version |
| 2. cat /etc/version | show user directory version |
| 3. upramdisk | mount ramdisk |
| 4. downramdisk | unmount ramdisk |



BLOCKSCHALTBILD SEQ47/50

COMMUNICATION PROTOCOL OF THE SEQ47/50

Ing.-Büro Sven Leckel
 Leberstr. 63, 10829 Berlin
 Telefon: 030 - 78 95 50 11
 Fax: 030 - 78 95 50 12



DATA OF THE SERIAL INTERFACE RS232	
FORMAT	1 startbit, 8 databit, 1 stopbit, no parity
BAUDRATE	9600 baud
HANDSHAKE	no handshake