

**SISTEM ZA ZAJEM
OCEANOGRFSKIH PODATKOV
NA MORSKI BIOLOŠKI POSTAJI
V PIRANU**

**POROČILO O POSODOBITVI
KRMILNE ENOTE IN
ZAMENJAVI TOKOMERA
NA OCEANOGRFSKI BOJI**

Naziv projekta:	Interreg III
Listina:	POROCILO_POSODOBITEV.doc
Različica:	1.0
Zadnja sprememba:	Marec 2006
Tiskano:	09.03.2006
Izvod:	1/1
Obseg:	Poročilo
Avtor:	mag. Damjan Šonc univ.dipl.inž.

KAZALO

UVOD.....	3
OPIS STROJNE OPREME	3
OPIS PROGRAMSKE OPREME NA BOJI.....	4
OPIS SPREMEMB PROGRAMSKE OPREME NA KOPNEM	5
ZAMENJAVA TOKOMERA	6
PRILOGA A – NASTAVITVE TOKOMERA.....	7
PRILOGA B – STRUKTURA NOVE MYSQL BAZE PODATKOV ZA MERITVE Z BOJE	8

UVOD

V okviru krovnega projekta Interreg III je v drugi polovici leta 2005 stekel projekt posodobitve krmilne enote na oceanografski boji Morske biološke postaje v Piranu.

Poleg zamenjave krmilne enote, se je izvedla tudi zamenjava tokomera. Starejši tip tokomera ADP je nadomestil novejši tip AWAC, ki poleg morskih tokov meri tudi valove. Oba instrumenta sta izdelek norveškega podjetja Nortek.

V nadaljevanju je podan kratek opis strojne in programske opreme nove krmilne enote ter seznam sprememb v programski opremi na Morski biološki postaji v Piranu. Dodatne listine s podrobnejšimi opisi so v elektronski obliki skupaj z izvorno kodo na posebni DVD zgoščenci.

OPIS STROJNE OPREME

Nova krmilna enota je plod domačega znanja in je sestavljena iz več modulov. Glavni procesni modul je sestavljen okoli sodobnega 32-bitnega mikrokrmilnika AT91RM9200 proizvajalca Atmel, katerega jedro deluje s frekvenco 180 Mhz in doseže 200 MIPS, obenem pa v aktivnem stanju porabi le okrog 30 mA toka pri 1.8V. Poleg mikrokrmilnika se na glavnem procesnem modulu nahajajo še:

- 32 MB SDRAM pomnilnika,
- 4 MB bliskovnega (flash) ROM pomnilnika,
- 100 Mbit/s Ethernet vmesnik,
- 400 Kbit/s I2C vmesnik,
- USB 2.0 vmesnik s hitrostjo prenosa 12 Mbit/s in sicer z vrati za način delovanja kot gostitelj in z vrati za način delovanja kot naprava,
- štiri zaporedni USART (Universal Synchronous Asynchronous Receiver Transmitter) vmesniki, pri čemer eden deluje kot IrDA vmesnik,
- zaporedni SSC (Synchronous Serial Controller) vmesnik,
- dvožični zaporedni vmesnik UART (Universal Asynchronous Receiver Transmitter), ki služi kot konzola operacijskega sistema, lahko pa tudi kot vmesnik za razhroščevanje,
- IDE vmesnik za CF (Compact Flash) pomnilniške kartice,
- MCI vmesnik za MMC/SD pomnilniške kartice,
- 8-kanalni 12-bitni A/D pretvornik,
- tipalo za temperaturo in
- tipalo za zračno vlago.

Na glavni procesni modul je priključeno vodilo, ki je lahko sestavljeno iz več zaporedno vezanih nosilnih plošč, pri čemer ima vsaka plošča 12 prostih rež za priključitev V/I modulov. V/I moduli preko vodila komunicirajo z glavnim procesnim modulom s pomočjo SSC in/ali UART vmesnika (RS422 napetostni nivoji), preko vodila pa dobivajo tudi potrebno napajalno napetost.

Obstoječi V/I moduli so zgrajeni okoli mikrokrmilnikov serije AT91SAM7S32 proizvajalca Atmel in so namenjeni zaporedni povezavi z merilnimi instrumenti po RS232, RS422 ali RS485 standardu. Vsak modul omogoča priključitev do štirih naprav po enem izmed omenjenih standardov, pri čemer je izbira standarda poljubna in nastavljiva, naprave pa so galvansko ločene od ostalega sistema. Celotna zasnova sistema omogoča tudi priključitev dodatnih V/I modulov, ki izmenjujejo podatke z merilnimi instrumenti po CAN zaporednem vmesniku, ali V/I modulov z analognimi (do 24 bitni A/D, do 24 bitni D/A), digitalnimi in frekvenčnimi vhodi ter izhodi.

OPIS PROGRAMSKE OPREME NA BOJI

Sistemska programska oprema na boji predstavlja operacijski sistem Linux z različico jedra 2.6.14. Za nalaganje operacijskega sistema poskrbi dvostopenjski zagonski nalagalnik, ki je shranjen v bliskovnem pomnilniku na matični plošči glavnega procesnega modula. Datotečni sistem se nahaja na zunanjem polprevodniškem disku. Začasno je to USB ključ kapacitete 256MB, pri naslednji posodobitvi pa bo to CF oziroma MMC pomnilniška kartica kapacitete vsaj 1GB.

Uporabniška programska oprema je sestavljena iz naslednjih programskih modulov:

1. Program za sinhronizacijo sistemske ure se ob zagonu sistema ter nato še v 24 urnih časovnih presledkih poveže z GPS modulom in nastavi sistemske uro na točen UTC čas, ki ga prejme z atomskih ur na satelitih sistema GPS.
2. Program za zajem podatkov ob zagonu najprej vzpostavi povezavo z V/I moduli in uskladi njihovo uro s sistemsko. Ko so vsi V/I moduli usklajeni in merilni instrumenti, ki so priključeni na njih, pripravljeni za delo, izda ukaz za vzorčenje in prične z zbiranjem podatkov. Podatke zbira v pakete, ki vsebujejo za 30 min meritev in jih zapiše v ustrezne datoteke.
3. Programi na V/I modulih delujejo v neskončni zanki (brez operacijskega sistema) in komunicirajo z merilnimi napravami po različnih protokolih, vršijo diagnostiko (poraba toka, prekoračenje mej merilnih območij na instrumentih, ...), meritve združijo v pakete in jih pošiljajo preko vodila glavnemu procesnemu modulu.
4. Program za školjko operacijskega sistema, ki stiska datoteke z meritvami s pomočjo sistemskega programa bzip2.
5. Program za modemske komunikacije s kopnim, ki je nekoliko spremenjena izvedba preprostega ARQ protokola, ki je deloval v prejšnji krmilni enoti. Glavna sprememba se nanaša na prenos datotek z meritvami. Program preko deljenega pomnilnika (shared memory) izmenjuje tudi podatke s programom za zajem podatkov, ki neposredno komunicira z V/I moduli in sicer preko deljenega pomnilnika posreduje podatke, ki se neposredno izmenjujejo z V/I moduli in programske opreme na kopnem za potrebe sistemske administracije.

Zaradi drugačne strojne opreme in popolnoma drugačne zasnove sistema je vsa programska oprema na boji napisana na novo in so vanjo vključeni le manjši odseki kode starega sistema. Programi za V/I module so razviti v delovnem namizju podjetja IAR Systems – različica 4.11 in prevedeni s prevajalnikom, ki je vključen v to delovno namizje. Operacijski sistem in ostalo programje na glavnem procesnem modulu je prevedeno s križnim GNU C

prevajalnikom - različica 3.4.3., sistemska programska knjižnica pa je uClib, ki je prav tako pod GNU licenco.

Pri preizkušanju novega sistema smo ugotovili, da obstoječi GPS modul ne daje vedno pravih podatkov za UTC čas, temveč občasno vrača satelitski čas, ki se od UTC časa razlikuje za 13 sekund. Zaradi tega se časovna sinhronizacija sistemske ure na glavni procesorski plošči preko GPS modula začasno vrši le ob zagonu sistema. Do zamenjave GPS modula in ustrezne prilagoditve programske opreme na nov modul se beležijo časovna neskladja med uro na glavni procesorski plošči in satelitsko uro ter med uro na glavni procesorski plošči in urami na V/I modulih. Ker se med delovanjem sistema beleži tudi temperatura okolja, kateri je elektronika izpostavljena, bo na ta način zajeta tudi informacija o temperaturni odvisnosti posameznih ur v sistemu in se bo upoštevala pri postopku časovnih popravkov v posodobljeni različici programja.

V okviru projekta Interreg III je predvidena tudi nadgradnja komunikacijske povezave med kopnim in bojo. Prenosna hitrost se bo bistveno povečala, s tem pa bo postavljena osnova za uporabo TCP, UDP in drugih Internetnih protokolov. Zaradi tega bo potrebno tudi spremeniti programsko opremo za komunikacijo s kopnim in sistemsko administracijo. Omogočena bo uporaba nekaterih rešitev, ki so že vgrajene v Linux operacijski sistem, vsekakor pa bo potrebno nadgraditi programje za sistemsko administracijo V/I modulov, ki bo potem omogočala tudi daljinsko nadgradnjo programske opreme.

OPIS SPREMEMB PROGRAMSKE OPREME NA KOPNEM

Pošiljanje stisnjenih datotek z meritvami in zamenjava tokomera na boji je botrovala tudi spremembam v programski opremi na kopnem. Dodan je programski modul, ki skupaj s programom za školjko operacijskega sistema skrbi za razširitev stisnjenih datotek s podatki. Zaradi drugačnega načina zajema podatkov in novih podatkov s tokomera je bila izdelana nova struktura MySQL podatkovne baze. Ker nova struktura baze zahteva tudi spremembe v programih za naknadno procesiranje podatkov in ker je potrebno delovanje in uporabnost nove baze tudi preizkusiti in po potrebi dopolniti, se podatki začasno vpisujejo hkrati v dve podatkovni bazi: bazo s staro strukturo podatkov in bazo z novo strukturo podatkov. Program za vpis podatkov je bil ustrezno dopolnjen in spremenjen, da ta dvojni vpis omogoča. Ko bo nova struktura podatkovne baze dokončno potrjena in preizkušena in bodo napisani tudi programi za naknadno procesiranje podatkov iz nove baze, bo stara podatkovna baza ukinjena.

Začasna struktura nove MySQL podatkovne baze je v obliki SQL ukazov prikazana v prilogi B.

Obstoječi program za administracijo sistema na boji je ukinjen. Namesto njega sta začasno (do vzpostavitve nove širše komunikacijske povezave z bojo) nameščena dva krajša programa, ki omogočata vklop/izklop in preprost terminalski dostop do posameznih merilnih instrumentov, ki so priključeni na V/I modulih na boji.

ZAMENJAVA TOKOMERA

Poleg zamenjave krmilne enote se je izvedla tudi zamenjava tokomera. Novi tokomer poleg podatkov o morskih tokovih pošilja tudi podatke o morskih valovih. Vzorčenje morskih tokov poteka 5 minut vsake pol ure, vzorčenje valov pa 1024 sekund (hitrost vzorčenja je en vzorec na sekundo) vsake pol ure. Skupno tako tokomer pošilja en paket meritev morskih tokov in en paket meritev morskih valov na pol ure. Celotne trenutne nastavitve tokomera so podane v prilogi A.

Zaradi napake v programski opremi tokomera se podatki posredujejo v ASCII obliki, ki tudi ne omogoča prenos vseh podatkov o morskih valovih. Trenutna različica programske opreme v tokomeru zahteva ročni poseg pri nadgradnji programske opreme in ta na daljavo zato ni izvedljiva. Ko bo podjetje Nortec dostavilo popravljene različice "zapečene" programske opreme (firmware), bo potrebno ustrezno popraviti in dopolniti tudi program na V/I modulu, da bo omogočal zajem podatkov v novi dvojiški obliki in dopolniti strukturo MySQL baze podatkov na Morski biološki postaji z novimi meritvami, ki bodo v tej novi obliki na voljo.

PRILOGA A – NASTAVITVE TOKOMERA

NCELLS : 21
CELLSIZE : 1.0
BLANKING : 1.00
AVGINTERVAL : 600
MEASINTERVAL : 1800
COORDINATESYSTEM : ENU
WAVEPROCMETHOD : MLMST
NWAVESAMPLES : 1024
WAVESAMPLERATE : 1
WAVEINTERVAL : 1800
SENSORFORMAT : DT:17 TP:5:2 PR:8:3 PI:6:2 RO:6:2 HE:5:1 BT:4:1 SS:4:0 O1:3
O2:3 ER:6
CURRENTFORMAT : DT:17 CN:3 V1:6:3 V2:6:3 V3:6:3 A1:3 A2:3 A3:3 SP:6:3
DR:5:1 ER:6
WAVEFORMAT : DT:17 MZ:5:2 PP:5:2 PD:5:1 PS:4:1 WD:5:1 UI:4:2 HS:5:2 H1:5:2
HX:5:2 TZ:5:2 TM:5:2 MP:6:2 BS:4 ER:6
ERRORFORMATMARKER : E
SENSORFORMATMARKER : S
CURRENTFORMATMARKER : C
WAVEFORMATMARKER : W
CURRENTCELLS : 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21
OUTPUTDELIMITER : ,
ENDOFDATAMARKER : \r\n
NANINDICATOR : NUL
DATAOUTPUT : SENSOR CURRENT WAVE
OUTPUTMODE : ASCII
MOUNTINGHEIGHT : 0.50
PRESSUREOFFSET : 0.00
DEPTHESTIMATE : 20.00
REPORTMODE : LONG
CHECKSUM : NORTEK
ANALOGINPUT1 : OFF
ANALOGINPUT2 : OFF
POWEROUT : OFF
SALINITY : 37.0
WAVEENABLED : ON
POWERLEVEL : HIGH+
FEEDCONTROL : STREAMED

PRILOGA B – STRUKTURA NOVE MySQL BAZE PODATKOV ZA MERITVE Z BOJE

```
#
# Predpostavimo, da ustrezna baza podatkov ze obstaja
#
USE buoy??_new          # buoy04 za 2004, buoy05 za 2005 ...

#
# Struktura tabele 'profile'
#
DROP TABLE IF EXISTS profile;
CREATE TABLE `profile` (
  `pid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,      # ID
  `datestart` DATETIME NOT NULL,                       # datum/ura zacetka meritev
  `sec` INT(10) UNSIGNED NOT NULL,                     # zacetek meritev v sekundah od 1.1.1970 - UNIX cas
  `srvdatestart` DATETIME NOT NULL,                   # datum/ura zacetka sprejemanja podatkov (ura na strezniku)
  `srvdateend` DATETIME NOT NULL,                     # datum/ura zakljucka sprejemanja podatkov (ura na strezniku)
  PRIMARY KEY (`pid`),                                 # PRIMARY KEY
  UNIQUE KEY `pid` (`pid`),                            # UNIQUE KEY
  KEY `datestart` (`datestart`)                       # KEY
) TYPE = MyISAM COMMENT = 'Tabela vsebuje globalne podatke o polurnih profilih';

#
# Struktura tabele 'awacsensors'
#
DROP TABLE IF EXISTS awacsensors;
CREATE TABLE `awacsensors` (
  `asid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,    # ID
  `acnfid` SMALLINT(5) UNSIGNED NOT NULL,            # povezava z awacconfig tabelo
  `datetime` DATETIME NOT NULL,                      # datum in cas meritve
  `heading` FLOAT NOT NULL,                           # smer - v stopinjah
  `pitch` FLOAT NOT NULL,                             # naklon - v stopinjah
  `roll` FLOAT NOT NULL,                              # naklon - v stopinjah
  `temperature` FLOAT NOT NULL,                      # temperatura - v stopinjah Celzija
  `pressure` FLOAT NOT NULL,                          # tlak - v barih
  `sndspeed` FLOAT NOT NULL,                          # hitrost zvoka - v m/s
  `voltage` FLOAT NOT NULL,                           # napetost napajanja - v voltih
  `errcode` SMALLINT(5) UNSIGNED NOT NULL,           # koda napake
```


Poročilo o posodobitvi krmilne enote in zamenjavi tokomera

```
PRIMARY KEY (`acid`), # PRIMARY KEY
UNIQUE KEY `acid` (`acid`) # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'Tabela vsebuje zaglavje posameznih profilov ADP-ja';

#
# Struktura tabele 'awaccurrents'
#
DROP TABLE IF EXISTS awaccurrents;
CREATE TABLE `awaccurrents` (
  `acid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, # ID
  `asid` INT(10) UNSIGNED NOT NULL, # konfiguracija - povezava z awacsensors datoteko
  `cellno` TINYINT UNSIGNED NOT NULL, # st. celice nad tokomerom
  `curE` FLOAT NOT NULL, # E komponenta hitrosti toka v cm/s
  `curN` FLOAT NOT NULL, # N komponenta hitrosti toka v cm/s
  `curW` FLOAT NOT NULL, # W (up) komponenta hitrosti toka v cm/s
  `amp1` TINYINT(3) UNSIGNED NOT NULL, # jakost odziva 1. curka
  `amp2` TINYINT(3) UNSIGNED NOT NULL, # jakost odziva 2. curka
  `amp3` TINYINT(3) UNSIGNED NOT NULL, # jakost odziva 3. curka
  `errcode` SMALLINT(5) UNSIGNED NOT NULL, # koda napake
  PRIMARY KEY (`acid`), # PRIMARY KEY
  UNIQUE KEY `acid` (`acid`) # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'AWAC izmerjeni podatki o morskih tokovih';

#
# Struktura tabele 'awacwaves'
#
DROP TABLE IF EXISTS awacwaves;
CREATE TABLE `awacwaves` (
  `awid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, # ID
  `acnfid` SMALLINT(5) UNSIGNED NOT NULL, # konfiguracija
  `datetime` DATETIME NOT NULL, # datum in cas meritve
  `mzcp` FLOAT NOT NULL, # srednja vrednost periode preskokov cez 0 v s
  `peakp` FLOAT NOT NULL, # vrsna vrednost periode v s
  `dirpp` FLOAT NOT NULL, # smer valov pri vrsni periodi v stopinjah
  `spreadpp` FLOAT NOT NULL, # razprsenost valov pri vrsni periodi v stopinjah
  `maindir` FLOAT NOT NULL, # glavna smer valov v stopinjah
  `UI` FLOAT NOT NULL, # smerni indeks
  `wheight` FLOAT NOT NULL, # visina valov (iz spektra) v m
  `wh10` FLOAT NOT NULL, # 1/10 visine valov
  `whmax` FLOAT NOT NULL, # maksimalna visina valov
```

Poročilo o posodobitvi krmilne enote in zamenjavi tokomera

```
`mwaveper` FLOAT NOT NULL,           # srednja vrednost periode valov v s
`bestwaveper` FLOAT NOT NULL,        # najboljsi priblizek za periodo valov v s
`sensdepth` FLOAT NOT NULL,          # tipalo je v globini ? m (podatek o tlaku)
`baddet` SMALLINT UNSIGNED NOT NULL, # stevilo AST slabih zaznav
`errcode` INT UNSIGNED NOT NULL,     # koda napake
PRIMARY KEY (`awid`),                 # PRIMARY KEY
UNIQUE KEY `awid` (`awid`)           # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'AWAC izmerjeni podatki o morskih valovih';

#
# Struktura tabele 'awacconfig'
#
DROP TABLE IF EXISTS awacconfig;
CREATE TABLE `awacconfig` (
  `acnfid` SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT, # adpcnf ID
  `datetime` DATETIME NOT NULL,                          # datum in cas od katerega dalje velja
  `noCells` TINYINT(3) UNSIGNED NOT NULL,                 # number of cells
  `cellSize` SMALLINT(5) UNSIGNED NOT NULL,               # one cell size (cm)
  `blankDist` SMALLINT(5) UNSIGNED NOT NULL,              # blank distance (cm)
  `avgTime` SMALLINT(5) UNSIGNED NOT NULL,                # average time in s
  `FFTsize` SMALLINT(5) UNSIGNED NOT NULL,                # dolzina FFT
  PRIMARY KEY (`acnfid`),                                  # PRIMARY KEY
  UNIQUE KEY `acnfid` (`acnfid`)                          # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'konfiguracija AWAC-a ... tu pregledamo ce je z AWAC-om vse OK';

#
# Struktura tabele 'gps'
#
DROP TABLE IF EXISTS gps;
CREATE TABLE `gps` (
  `gid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,        # ID
  `sec` INT(10) UNSIGNED NOT NULL,                       # UNIX cas meritve
  `lon` FLOAT NOT NULL,                                  # zemljepisna sirina v stopinjah
  `lat` FLOAT NOT NULL,                                  # zemljepisna dolzina v stopinjah
  `alt` FLOAT NOT NULL,                                   # nadmorska visina v m
  `satno` SMALLINT NOT NULL,                             # stevilo satelitov
  `errcode` TINYINT UNSIGNED NOT NULL,                  # err. code
  PRIMARY KEY (`gid`),                                    # PRIMARY KEY
  UNIQUE KEY `cid` (`gid`)                               # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'GPS podatki';
```

Poročilo o posodobitvi krmilne enote in zamenjavi tokomera

```
#
# Struktura tabele 'procunit'
#
DROP TABLE IF EXISTS procunit;
CREATE TABLE `procunit` (
  `puid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, # ID
  `sec` INT(10) UNSIGNED NOT NULL, # UNIX cas meritve
  `voltage` FLOAT NOT NULL, # el.napetost baterij v voltih (vzorci se na 10 sekund)
  `currentio` FLOAT NOT NULL, # skupna poraba el.toka v amperih (vzorci se na 10 sekund)
  `currentcpu` FLOAT NOT NULL, # poraba el.toka procesorske plosce v amperih (vzorci se na 10
  # sekund)
  `humid` TINYINT(3) NOT NULL, # relatina vlaznost v glavni omarici v %(vzorci se na 10sekund)
  `temperature` FLOAT NOT NULL, # temperatura v glavni omarici v stopinjah Celzija (vzorci se na
  # 10 sekund, natančnost na 0.5 stopinje)
  `errcode` TINYINT UNSIGNED NOT NULL, # err. code
  PRIMARY KEY (`puid`), # PRIMARY KEY
  UNIQUE KEY `puid` (`puid`) # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'podatki z glavne procesorske plosce v krmilni omarici';

#
# Struktura tabele 'instruments'
#
DROP TABLE IF EXISTS instruments;
CREATE TABLE `instruments` (
  `inid` SMALLINT(5) UNSIGNED NOT NULL, # oznaka instrumenta
  `board` TINYINT(3) UNSIGNED NOT NULL, # oznaka plosce (ID plosce od 0..15)
  `port` TINYINT(3) UNSIGNED NOT NULL, # stevilka vrat (0..3) oziroma kasneje (0..1)
  `name` VARCHAR(50) NOT NULL, # ime/oznaka instrumenta
  `description` VARCHAR(200) NOT NULL, # dodaten opis
  PRIMARY KEY (`inid`), # PRIMARY KEY
  UNIQUE KEY `inid` (`inid`), # UNIQUE KEY
  KEY `name` (`name`) # KEY
) TYPE = MyISAM COMMENT = 'Podatki o instrumentih';

INSERT instruments VALUES (1,2,0,"Vlagomer","Proizvajalec Vaisala, model ...");
INSERT instruments VALUES (2,2,1,"Slanomer","Proizvajalec Idronaut, model 316");
INSERT instruments VALUES (3,3,0,"Vetromer","Proizvajalec Vaisala, model ...");
INSERT instruments VALUES (4,3,1,"Kompas","Proizvajalec Vaisala, model ...");
INSERT instruments VALUES (5,4,0,"Tokomer","Proizvajalec Vaisala, model ...");
INSERT instruments VALUES (6,0,0,"Procesorska plošča","Proizvajalec Ditel, model AT91RM9200");
```

Poročilo o posodobitvi krmilne enote in zamenjavi tokomera

```
INSERT instruments VALUES (7,0,1,"V/I plošča - oznaka 2","Proizvajalec Ditel, model AT7SAM32");
INSERT instruments VALUES (8,0,1,"V/I plošča - oznaka 3","Proizvajalec Ditel, model AT7SAM32");
INSERT instruments VALUES (9,0,1,"V/I plošča - oznaka 4","Proizvajalec Ditel, model AT7SAM32");
INSERT instruments VALUES (10,0,2,"Modem","Proizvajalec , model ...");
INSERT instruments VALUES (11,0,4,"GPS","Proizvajalec Ashtek, model ...");
```

```
#
# Struktura tabele 'samptimes'
#
DROP TABLE IF EXISTS samptimes;
CREATE TABLE `samptimes` (
  `smpid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, # ID
  `inid` SMALLINT(5) UNSIGNED NOT NULL, # oznaka instrumenta
  `datetime` DATETIME NOT NULL, # datum in cas od katerega dalje velja
  `samptime` INT(10) NOT NULL, # cas vzorčenja v us (mikrosekundah)
  PRIMARY KEY (`smpid`), # PRIMARY KEY
  UNIQUE KEY `smpid` (`smpid`) # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'Podatki o vzorcevalnih casih instrumentov';
```

```
INSERT samptimes (inid,datetime,samptime) VALUES (1,NOW(),10000000);
INSERT samptimes (inid,datetime,samptime) VALUES (2,NOW(),5000000);
INSERT samptimes (inid,datetime,samptime) VALUES (3,NOW(),250000);
INSERT samptimes (inid,datetime,samptime) VALUES (4,NOW(),250000);
INSERT samptimes (inid,datetime,samptime) VALUES (6,NOW(),10000000);
INSERT samptimes (inid,datetime,samptime) VALUES (7,NOW(),60000000);
INSERT samptimes (inid,datetime,samptime) VALUES (8,NOW(),60000000);
INSERT samptimes (inid,datetime,samptime) VALUES (9,NOW(),60000000);
INSERT samptimes (inid,datetime,samptime) VALUES (11,NOW(),1000000);
```

```
#
# Struktura tabele 'powerconsumption'
#
DROP TABLE IF EXISTS powerconsumption;
CREATE TABLE `powerconsumption` (
  `pcid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, # ID
  `inid` SMALLINT(5) UNSIGNED NOT NULL, # oznaka instrumenta
  `sec` INT(10) UNSIGNED NOT NULL, # UNIX cas meritve
  `current` FLOAT NOT NULL, # el.tok v amperih (vzorci se na 60 sekund)
```

Poročilo o posodobitvi krmilne enote in zamenjavi tokomera

```
PRIMARY KEY (`pcid`),                                # PRIMARY KEY
UNIQUE KEY `pcid` (`pcid`)                          # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'Podatki o porabi toka posameznih instrumentov';

#
# Struktura tabele 'wind'
#
DROP TABLE IF EXISTS wind;
CREATE TABLE `wind` (
  `wid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,    # wind ID
  `sec` INT(10) UNSIGNED NOT NULL,                  # UNIX cas meritve
  `msec` SMALLINT(4) UNSIGNED NOT NULL,            # milisekunda meritve
  `wndU` FLOAT NOT NULL,                            # U komponenta vetra v m/s
  `wndV` FLOAT NOT NULL,                            # V komponenta vetra v m/s
  `wndW` FLOAT NOT NULL,                            # W komponenta vetra v m/s
  `sndSpeed` FLOAT NOT NULL,                        # hitrost zvoka v m/s
  `instrcode` TINYINT UNSIGNED NOT NULL,           # instrument err. code
  `errcode` TINYINT UNSIGNED NOT NULL,              # err. code
  PRIMARY KEY (`wid`),                              # PRIMARY KEY
  UNIQUE KEY `wid` (`wid`)                          # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'Tabela vsebuje trenutne vrednosti hitrosti vetra';

#
# Struktura tabele 'compass'
#
DROP TABLE IF EXISTS compass;
CREATE TABLE `compass` (
  `cid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,   # compass ID
  `sec` INT(10) UNSIGNED NOT NULL,                  # UNIX cas meritve
  `msec` SMALLINT(4) UNSIGNED NOT NULL,            # milisekunda meritve
  `dir` FLOAT NOT NULL,                             # smer v stopinjah
  `roll` FLOAT NOT NULL,                            # 'roll' v stopinjah
  `pitch` FLOAT NOT NULL,                           # 'pitch' v stopinjah
  `errcoded` TINYINT(3) UNSIGNED NOT NULL,         # koda napake za smer
  `errcoder` TINYINT(3) UNSIGNED NOT NULL,         # koda napake za 'roll'
  `errcodep` TINYINT(3) UNSIGNED NOT NULL,         # koda napake za 'pitch'
  PRIMARY KEY (`cid`),                              # PRIMARY KEY
  UNIQUE KEY `cid` (`cid`)                          # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'Trenutni podatki kompasa';
```

Poročilo o posodobitvi krmilne enote in zamenjavi tokomera

```
#
# Struktura tabele 'air'
#
DROP TABLE IF EXISTS air;
CREATE TABLE `air` (
  `arid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, # ID
  `sec` INT(10) UNSIGNED NOT NULL, # UNIX cas meritve
  `humid` TINYINT UNSIGNED NOT NULL, # relativna vlaznost zraka v %
  `temp` FLOAT NOT NULL, # temperatura zraka v stopinjah Celzija
  `errcode` TINYINT UNSIGNED NOT NULL, # koda napake
  PRIMARY KEY (`arid`), # PRIMARY KEY
  UNIQUE KEY `arid` (`arid`) # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'Temperatura un vlaznost zraka';

#
# Struktura tabele 'sea'
#
DROP TABLE IF EXISTS sea;
CREATE TABLE `sea` (
  `seid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, # ID
  `sec` INT(10) UNSIGNED NOT NULL, # UNIX cas meritve
  `temp` FLOAT NOT NULL, # temperatura morja v stopinjah Celzija
  `cond` FLOAT NOT NULL, # prevodnost morske vode v S/m
  `salin` FLOAT NOT NULL, # slanost
  `errcode` TINYINT UNSIGNED NOT NULL, # koda napaka
  PRIMARY KEY (`seid`), # PRIMARY KEY
  UNIQUE KEY `seid` (`seid`) # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'Sonda za merjenje slanosti';

#
# Struktura tabele 'timesync'
#
DROP TABLE IF EXISTS timesync;
CREATE TABLE `timesync` (
  `tid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, # ID
  `datetime` DATETIME NOT NULL, # datum in cas casovne sinhronizacije po trenutni uri na glavni
  # proc. plosci
  `timediff` INT(10) NOT NULL, # razlika med GPS uro in uro na glavni proc. plosci v ms
```

Poročilo o posodobitvi krmilne enote in zamenjavi tokomera

```
PRIMARY KEY (`tid`), # PRIMARY KEY
UNIQUE KEY `tid` (`tid`) # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'Podatki o casovnih sinhronizacijah z GPS uro';

#
# Struktura tabele 'iosync'
#
DROP TABLE IF EXISTS iosync;
CREATE TABLE `iosync` (
  `ioid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, # ID
  `datetime` DATETIME NOT NULL, # datum in cas casovne sinhronizacije po trenutni uri na glavni
  # proc. plosci
  `boardid` TINYINT(3) UNSIGNED NOT NULL, # oznaka v/i plosce
  `timediff` INT(10) NOT NULL, # razlika med uro na glavni proc. plosci in uro na v/i plosci v
  # ms
  PRIMARY KEY (`ioid`), # PRIMARY KEY
  UNIQUE KEY `ioid` (`ioid`) # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'Podatki o casovnih sinhronizacijah V/I plosc z uro na glavni proc. plosci';

#
# Struktura tabele 'instrsync'
#
DROP TABLE IF EXISTS instrsync;
CREATE TABLE `instrsync` (
  `isid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, # ID
  `inid` SMALLINT(5) UNSIGNED NOT NULL, # oznaka instrumenta
  `datetime` DATETIME NOT NULL, # datum in cas casovne sinhronizacije po trenutni uri na glavni
  # proc. plosci
  `timediff` INT(10) NOT NULL, # razlika med uro na glavni proc. plosci in uro instrumenta v ms
  PRIMARY KEY (`isid`), # PRIMARY KEY
  UNIQUE KEY `isid` (`isid`) # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'Podatki o casovnih sinhronizacijah instrumentov z uro na glavni plosci';

#
# Struktura tabele 'alarms'
#
DROP TABLE IF EXISTS alarms;
CREATE TABLE `alarms` (
  `almid` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, # ID
  `datetime` DATETIME NOT NULL, # datum in cas alarma
```

Poročilo o posodobitvi krmilne enote in zamenjavi tokomera

```
`text` VARCHAR(255) NOT NULL,           # besedilo alarma
PRIMARY KEY (`almid`),                  # PRIMARY KEY
UNIQUE KEY `almid` (`almid`)           # UNIQUE KEY
) TYPE = MyISAM COMMENT = 'Podatki o alarmih v sistemu';

#
# Struktura tabele 'current_alarms'
#
DROP TABLE IF EXISTS current_alarms;
CREATE TABLE `current_alarms` (
  `battery` TINYINT UNSIGNED NOT NULL,  # stanje baterije
  `lbalmid` INT(10) UNSIGNED NOT NULL,  # last battery alarm ID
  `humidity` TINYINT UNSIGNED NOT NULL, # vlaga v omarici
  `lhalmid` INT(10) UNSIGNED NOT NULL,  # last humidity alarm ID
  `temperature` TINYINT UNSIGNED NOT NULL, # temperatura v omarici
  `ltalmid` INT(10) UNSIGNED NOT NULL,  # last temperature alarm ID
  `commlink` TINYINT UNSIGNED NOT NULL, # stanje komunikacij
  `lcalmid` INT(10) UNSIGNED NOT NULL,  # last communication alarm ID
  `awactilt` TINYINT UNSIGNED NOT NULL, # naklon AWAC-a
  `laalmid` INT(10) UNSIGNED NOT NULL,  # last AWAC alarm ID
  `awacmove` TINYINT UNSIGNED NOT NULL, # kakrsnakoli sprememba polozaja AWAC-a
  `lmalmid` INT(10) UNSIGNED NOT NULL  # last AWAC alarm ID
) TYPE = MyISAM COMMENT = 'Trenutno stanje alarmov';

INSERT current_alarms VALUES (0,0,0,0,0,0,0,0,0,0,0,0,0);
```