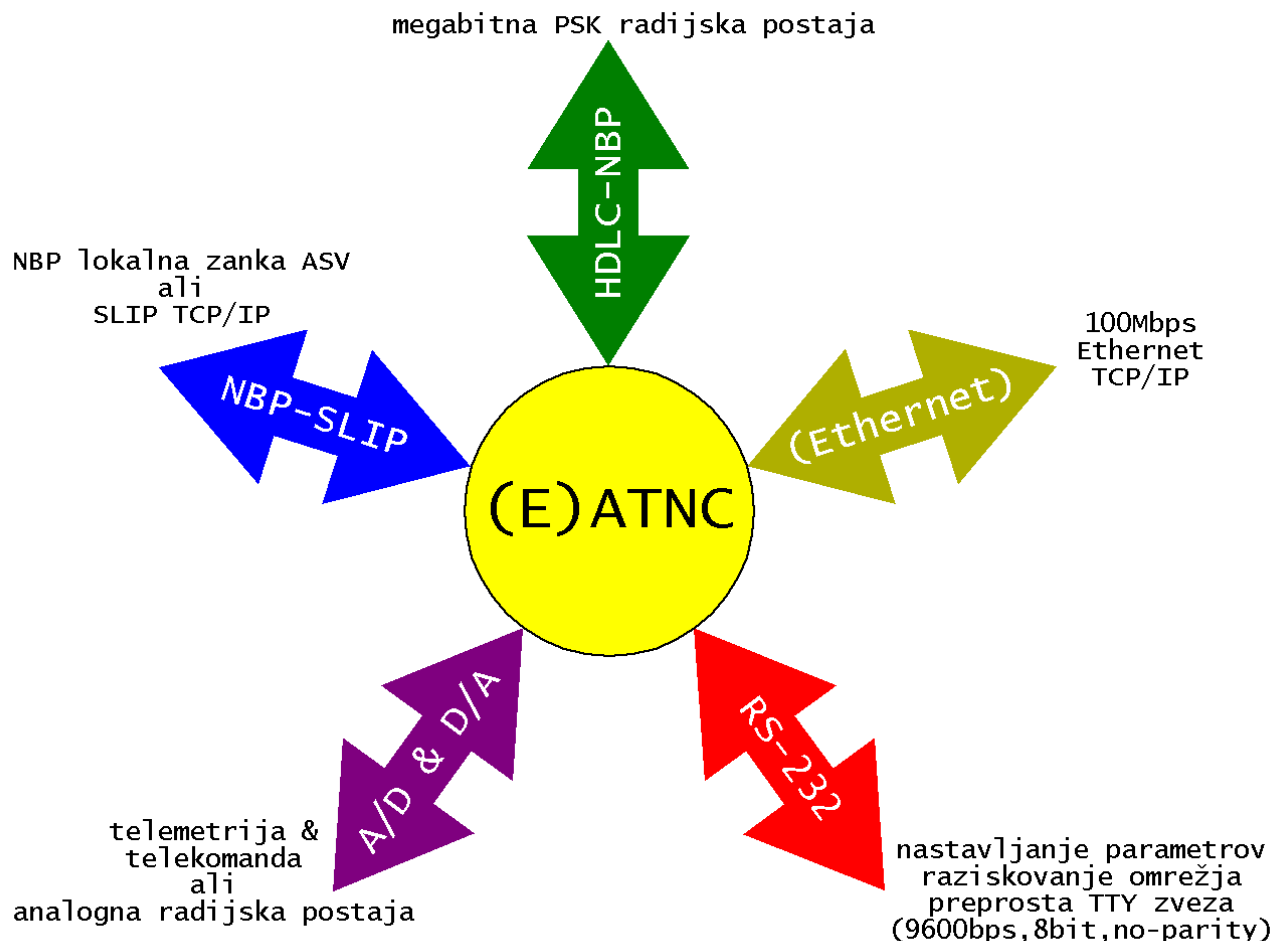


Matjaž Vidmar, S53MV

ArmTNC ali na kratko ATNC z mikrokrmilnikom LPC2138 oziroma LPC2148 (enako ohišje in razpored priključkov, neizkoriščena USB in nekaj malega več pomnilnika) se pogovarja z zunanjim svetom preko štirih vmesnikov: SSP (programski HDLC), UART0, UART1 in analognimi vrati.

Ethernet-ArmTNC ali na kratko EATNC z mikrokrmilnikom LPC2368 oziroma LPC2387 (povečan pomnilnik, enako ohišje in enak razpored priključkov) poleg omenjenih vmesnikov razpolaga še z Ethernet priključkom.

Programska oprema in uporabniški ukazi obeh naprav so podobni z **izjemo ETH**, zato je smiselno en sam skupen opis za (E)ATNC.



Ne-Brezhibni Protokol: pred uporabo (E)ATNCja je nujno preštudirati in dobro doumeti delovanje Ne-Brezhibnega Protokola (NBP). NBP uporablja 32-bitne naslove, ki jih (E)ATNC lahko zapiše oziroma jih lahko vnesemo na dva različna načina: kot HEX števila z osmimi znaki šestnajstiškega številskega sistema od "0" do "F" oziroma kot

skupina šestih velikih črk od "A" do "Z" ali številke od "0" do "9", ki predstavljajo znake 36-iškega (šestintridesetiškega) številskega sistema. Opisana inačica (E)ATNC privzeto uporablja 36-iški številski sistem, ker si je črkovna imena in radioamaterske klicne znake dosti lažje zapomniti kot pa dolga heksadecimalna števila. Način vnosa/izpisa lahko sicer v vsakem trenutku zamenjamo z ukazom "H".

(E)ATNC uporablja programski HDLC (X.25) krmilnik, ki lahko proizvaja radijske okvirje Ne-Brezhibnega Protokola (NBP) oziroma navadne AX.25 okvirje v načinu KISS. SSP je tako sprogramiran, da se obnaša proti bitni sinhronizaciji in skramblerju popolnoma enako kot znani HDLC vmesniki Z8530SCC oziroma SAB82532. Uporabljeni signali so takt, RXD, TXD, /PTT (preklop na oddajo) in /DCD (skvelč sprejemnika).

Zakaj prav HDLC? Združljivost ponazaj z AX.25 še zdaleč ni tako pomembna kot naslednje lastnosti HDLC:

(A) Komuniciramo z okvirji, v HDLC sta začetek in konec okvirja natančno določena z zastavicami. Da se zastavica ne pojavi v podatkih, HDLC vsakih 5 zaporednih enic podatkov vrine ničlo.

(B) HDLC se običajno uporablja skupaj z diferencialnim kodiranjem NRZI, kjer so enice konstantni nivoji, ničle pa preskoki nivojev. Diferencialno kodiranje zagotavlja zanesljivo regeneracijo takta v sprejemniku.

(C) Diferencialno kodiranje je neobčutljivo na polariteto signala, zato ga brez težav uporabljamo z BPSK ali QPSK radijskimi postajami.

(D) Vrivanje ničel naredi iz poljubnih podatkov zaporedje, ki ni več deljivo s preprostimi primitivnimi polinomi. To pomeni, da lahko kot skrambler uporabljamo preprost polinomske delitelj. Skrambliranje podatkov je nujno v zvezah, ki ne prenašajo enosmerne komponente: sklop preko transformatorjev ali radijske postaje z ničelno medfrekvenco (ZIF).

Končno, softverski HDLC se bistveno bolje prilagaja resnični radijski zvezi, kjer toplotni šum tvori številne neveljavne kratke okvirje, ki izredno obremenjujejo hardverski HDLC in še bolj DMA.

Samo za primerjavo, Ethernet RMI ne ustreza nobeni od gornjih štirih zahtev, zato v PHY čipe tlačijo neučinkovita čudesa, da sploh kaj pride skozi transformatorje na UTP. Mikorkrmilnik EATNC ima za Ethernet predviden RMI (Reduced Media-Independent Interface) vmesnik, ki za komunikacijo po kablu potrebuje še PHY vmesnik z izvorom takta, kodiranjem in skrambliranjem podatkov ter regeneracijo takta na sprejemu.

EATNC uporablja PHY čip DP83848C in vtičnico z vgrajenimi ločilnimi transformatorji. Opisana inačica programa razume okvirje kodirane po RFC-894 in sicer IP4 okvirje z oznako tipa 0x0800 ter ARP okvirje z oznako tipa 0x0806. Poleg RFC-894 obstajajo se drugi načini kodiranja Ethernet okvirjev, ki pa jih v resničnem svetu srečamo

zelo poredkoma.

UART0 je namenjen nastavljanju parametrov (E)ATNCja, nadzoru njegovega delovanja in preprosti teleprinterski zvezi s sogovornikom (čvekanju). UART0 vedno deluje s hitrostjo 9600bps, 8bitov, brez paritete in je prirejen za delovanje z večino terminalskih programov (Hyper Terminal v WIN). (E)ATNC preko oddajnika UART0 tudi oddaja odmev tipkanja (echo), razume znak "backspace" (0x08) oziroma "delete" (0x7F) za brisanje in sam vstavlja "line-feed" (0x0A) za vsakim "carriage-return" (0x0D ali **<enter>**), kar je privzeta nastavitve večine terminalov in terminalskih programov.

(E)ATNC lahko deluje v **treh** ali **štirih** različnih načinih:

(1) kot KISS vmesnik za uporabo AX.25 oziroma poljubnega protokola na vmesniku UART1 in HDLC na radijski strani (M=0 in L=0),

(2) kot SLIP vmesnik na UART1 za prenos IP in drugih protokolov vključno z AX.25 preko NBP na radijski strani (M<>0 in L=0),

(3) kot lokalna NBP zanka med (E)ATNCji, ki so povezani v ASV (M<>0 in L<>0) in **(samo EATNC) kot Ethernet-ArMTNC (IP preko NBP).**

(4) (samo EATNC) kot Ethernet<>SLIP pretvornik (M=0 in L<>0).

V načinih (2) in (3), ko je (E)ATNCju dodeljen naslov Ne-Brezhibnega Protokola, se en sam (E)ATNC oziroma skupina med sabo povezanih (E)ATNCjev (preko UART1) obnašajo kot vozlišče omrežja NBP ali ArmSuperVozelj (ASV). Isti (E)ATNCji lahko istočasno delujejo v vozlišču ASV na tri različne načine:

(I) (E)ATNC potrdi sprejem na radijski strani, zavrti naslove, kot to predpisuje NBP in isti okvir spet odda na radijski strani. To je najpreprostejši ASV, ki potrebuje le en (E)ATNC. Prehod preko takšnega preprostega ASVja v istem frekvenčnem pasu in z isto bitno hitrostjo zahteva le en NBP naslov.

(II) (E)ATNC potrdi sprejem na radijski strani, zavrti naslove, kot to predpisuje NBP in okvir pošlje v lokalno zanko (E)ATNCjev, ki sestavljajo ASV. Drug (E)ATNC v isti lokalni zanki sprejme okvir, ponovno zavrti naslove, kot to predpisuje NBP in okvir odda na svoji radijski strani z drugačno bitno hitrostjo in v drugem frekvenčnem pasu. Takšen prehod preko enega ASVja torej zahteva dva NBP naslova za vstopni in izstopni (E)ATNC.

(III) (E)ATNC potrdi sprejem in obdela okvir, ki ni naslovljen neposredno nanj, pač pa na nekega drugega člana lokalne zanke v ASVju. Pri obdelavi zavrti naslove, kot to zahteva NBP, a izvirnega NBP naslovnika zamenja s svojim lastnim NBP naslovom. Hkrati z natančno nastavitvijo življenjske dobe okvirja poskrbi, da pride okvir v lokalni zanki točno do izvirnega naslovnika. Izvirni naslovnik okvirja nič več ne obdeluje, pač pa ga le posreduje na radijsko stran z drugačno bitno hitrostjo in v drugem frekvenčnem pasu. Takšen preskok ASVja zahteva en sam NBP naslov za dva

(E)ATNCja, vstopni in izstopni. Hkrati zamenjava naslovov v prvem (E)ATNCju natančno določa povratno pot.

Posebni primer je NBP naslov VSEM. Če je lokalna NBP zanka vključena (3), (E)ATNC pošlje okvir naslovljen VSEM najprej na radijsko stran in potem isti okvir še v lokalno zanko. V lokalni zanki se okvir VSEM obdela v prvem naslednjem (E)ATNCju glede na ožičenje ASVja. Z naslovom VSEM lahko vsakokrat naslavljamo samo en (E)ATNC v isti lokalni zanki. Ker se v odgovoru VSEM zamenja z dejanskim naslovom, moramo postopek raziskovanja omrežja ponoviti za vsak naslednji (E)ATNC v zanki tako, da dodamo naslov pred njim v pot NBP okvirjev. Mogoče izgleda komplicirano, ampak na ta način lahko enoveljavno raziščemo sestavo in natančno ožičenje katerekoli lokalne NBP zanke.

UART1 vedno deluje s podatkovnimi okvirji, ki so kodirani na enak način kot SLIP (Serial-Line Internet Protocol). Asinhrona oddaja vedno vsebuje ničlo za start, osem (8) podatkovnih bitov in vsaj eno enico za stop. Pariteta se ne uporablja, prav tako se ne uporablja RTS/CTS ali drugačen handshake za nadzor hitrosti prenosa podatkov.

SLIP okvir se vedno začne z znakom FEND in se vedno konča z znakom FEND. Če se v koristni vsebini okvirja pojavi byte, ki ustreza znaku FEND, se v SLIP oddaji nadomesti z zaporedjem znakov FESC+TFEND. Podobno se podatkovni byte FESC nadomesti z zaporedjem znakov FESC+TFESC. Na ta način FEND vedno enoveljavno določa začetek in konec okvirja. V primeru motenj ali napak pri prenosu končni oziroma začetni FEND omogočata zanesljivo ponovno sinhronizacijo za naslednji podatkovni okvir.

Nabor štirih posebnih znakov FEND, FESC, TFEND in TFESC je načeloma poljuben. Posebne znake pa bilo pametno izbrati tako, da se prenos v primeru motenj ali napak čimprej ponovno sinhronizira. Povrhu izbrati takšne posebne znake, ki se manj pojavljajo v podatkih: kodiranje HTTP na primer ne uporablja nestandardnih ASCII znakov nad 127. Vse štiri posebne znake FEND, FESC, TFEND in TFESC zato določa naslednja tabela:

Znak	Pomen	HEX vrednost	Desetiška vrednost
FEND	Frame END	0xC0	192
FESC	Frame ESCape	0xDB	219
TFEND	Transposed Frame END	0xDC	220
TFESC	Transposed Frame ESCape	0xDD	221

Ker je kodiranje SLIP preprosto in zanesljivo, uporabljajo popolnoma enak prenos podatkovnih okvirjev preko UART vmesnikov tudi drugi protokoli. Amaterski AX.25 uporablja kodiranje KISS, ki AX.25 okvirje kodira s postopkom SLIP. KISS bi lahko upravičeno poimenovali Serial-Line AX.25 ali SL-AX.25.

AX.25 KISS poznamo v treh različicah:

(1) izvorni KISS doda okvirju AX.25 samo byte 0x00 na začetek,
(2) SMACK doda byte 0x80 na začetek in 16-bitni CRC na konec in
(3) FLEXNET doda byte 0x20 na začetek in 16-bitni CRC na konec, ki se računa z drugačnim polinomom kot SMACK.
(E)ATNC pozna samo izvorni KISS in okvirjem dodaja byte 0x00 na začetek ter na drugi strani UART zveze preprosto odreže prvi byte ne glede na njegovo vrednost.

V načinu SLIP (E)ATNC preprosto prenaša vsebino okvirja ne glede na to, če je to okvir Internetnega Protokola inačica 4 (IP4) ali karkoli drugega. Edina omejitev je količina podatkov v okvirju: najmanj 8byte in največ 1500byte, v nekaterih načinih delovanja lahko celo nekoliko več od 1500byte. (E)ATNC prepozna okvirje IP4 po tem, da je vrednost prvega byte podatkov med 0x45 in 0x4F. V tem primeru (E)ATNC lahko poleg privzete poti (ukaz "I") uporabi tudi usmerjanje iz ukaza "A".

V lokalni NBP zanki (E)ATNC uporablja SLIP uokvirjanje za celotne okvirje Ne-Brezhibnega Protokola, kar bi lahko poimenovali SL-NBP. Prvih 32 podatkovnih bitov tedaj predstavlja vrsto in življenjsko dobo okvirja NBP. Življenjska doba se pri vsakem prenosu skozi (E)ATNC zamakne za štiri bite v desno, z leve pa se vrine štiri ničle. Lokalna zanka torej lahko vsebuje največ 8 ali 9 udeležencev, preden življenjska doba pade na nič.

Vrsta NBP okvirja v lokalni zanki je kodirana s skupinami štirih bitov v življenjski dobi. Opisana različica programa za (E)ATNC uporablja dve od petnajstih možnih kodiranj za vrsto NBP okvirja v lokalni zanki:

(1) 1111 ali HEX "F" je običajni NBP okvir med udeleženci zanke (dva NBP naslova za prehod ASVja) in

(2) 1010 ali HEX "A" pomeni preskok NBP okvirja skozi ASV (en sam NBP naslov za preskok ASVja).

Programiranje: točen način delovanja UART1 in ostale parametre izbiramo z ukazi, ki jih posredujemo (E)ATNC s terminalom na UART0. Večino nastavitvev lahko zapečemo v FLASH pomnilnik [LPC2138](#), [LPC2148](#), [LPC2368](#) ali [LPC2387](#) v sektor 25 na začetnem naslovu 0x7B000 s pripadajočim ukazom. Ob RESETu (E)ATNCja se nastavljeni parametri prepisejo iz FLASH sektorja 25 v RAM.

Poleg delovnega programa je smiselno čimprej nastaviti vse parametre (E)ATNCja v FLASH pomnilniku. Vsebina FLASH deviškega [LPC2138](#), [LPC2148](#), [LPC2368](#) ali [LPC2387](#) je namreč nastavljena na same logične enice. To pomeni vsi NBP naslovi se nastavijo VSEM (* ali FFFFFFFF), vsi IP naslovi se nastavijo na 255.255.255.255, vsi časovni parametri in števci se nastavijo na nesmiselno visoke vrednosti 255, 65535 ali 4294967295, pač glede na to, če gre za 8-bitno, 16-bitno ali 32-bitno število. V takšnih razmerah (E)ATNC ne more početi nič pametnega!

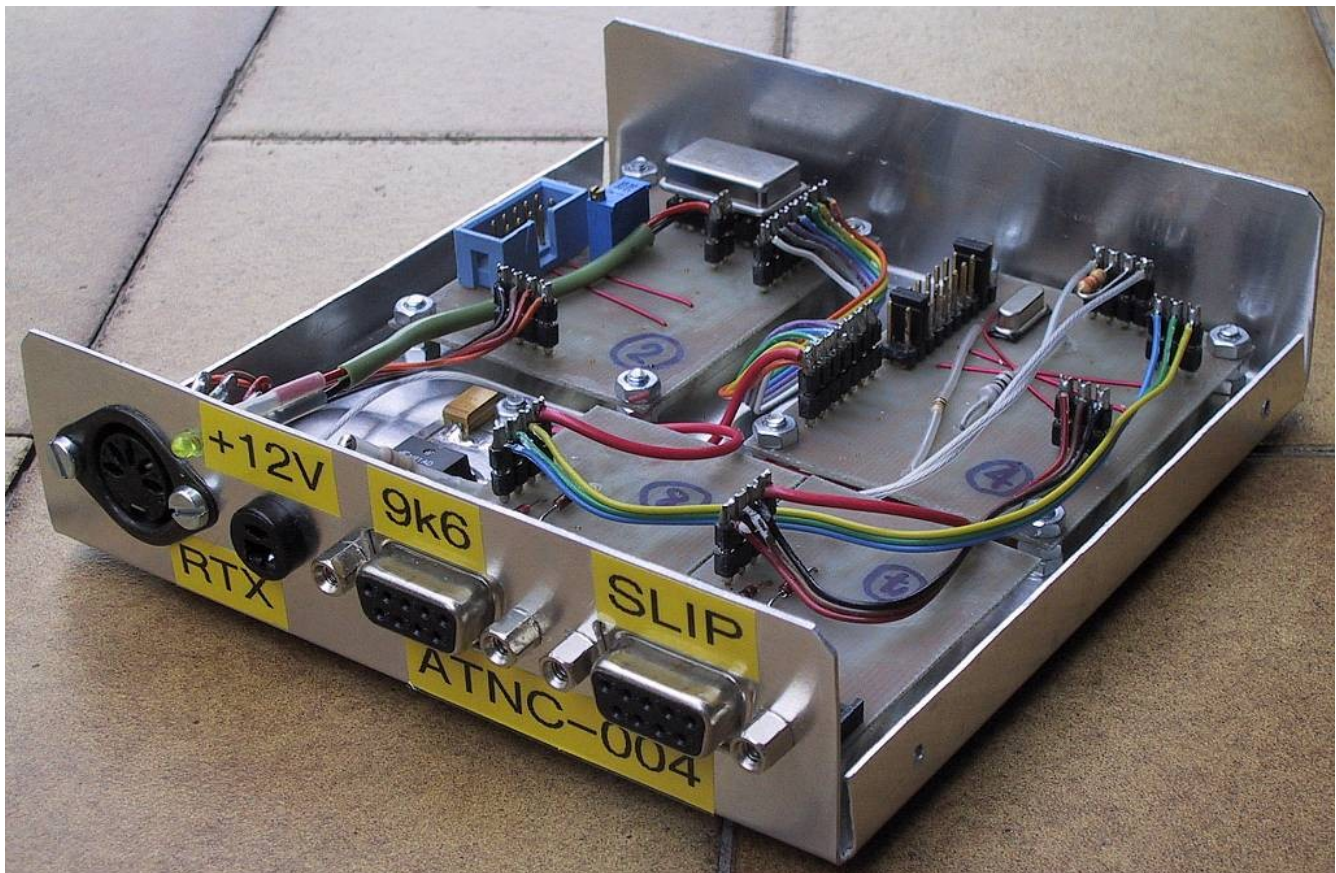
Bližnjica pri programiranju novega (E)ATNC je neposreden vpis v FLASH pomnilnik v sektor 25 na naslovu 0x7B000. Pri tem moramo naš

programator (FlashMagic, H-JTAG itd) obvezno nastaviti tako, da pred programiranjem briše samo tiste sektorje FLASH, ki jih uporablja naša HEX koda. Če programatorju pustimo, da briše cel pomnilnik, bomo pri programiranju parametrov izbrisali program oziroma obratno, pri vpisovanju programa izbrisali parametre...

(E)ATNC omogoča programiranje preko JTAG priključka oziroma preko UART0 in programa FlashMagic. Standardni 20-pinski ARM JTAG ima pri mikrokontrolerih **LPC2138, LPC2148, LPC2368 ali LPC2387** in večini programatorjev dva neizkoriščena signala na priključkih 17 in 19. v (E)ATNCju je na priključek 19 napeljan vhod **P0.14** oziroma **P2.10**, ki ozemljen ob RESETu požene BOOTLOADER za programiranje s FlashMagic preko UART0.

Pozor! Pri programiranju **LPC2138, LPC2148, LPC2368 ali LPC2387** preko JTAG (Cigotag in H-JTAG) morajo biti prisotni **vsi trije takti EATNCja** oziroma **oba takta ATNCja**: takt procesorja, takt bitne sinhronizacije (EPM3064) in **takt PHY Ethernet čipa DP83848C**, sicer programator ne more pognati ISP podprogramov za programiranje FLASH.

JTAG se pri **LPC2138 ali LPC2148** aktivira samo v primeru, ko je ob RESETu ozemljen RTCK. Cigotag ima v ta namen vgrajen PULLDOWN upor. Nekateri drugi programatorji (Keil ULINK2) imajo RTCK nepovezan in potrebujejo dodaten upor, da sploh lahko programirajo **LPC2138** preko JTAG.

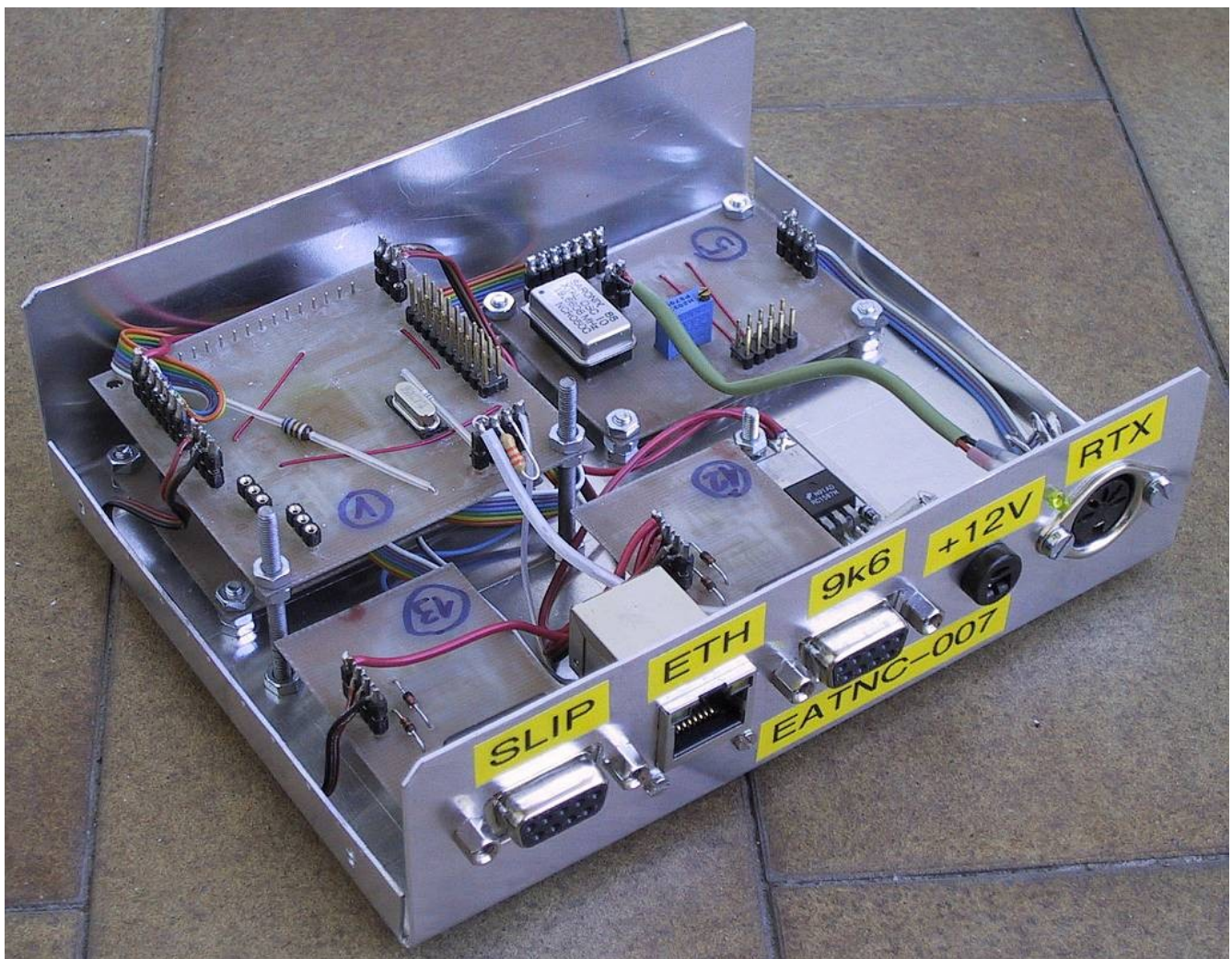


Ko ob RESETu ATNCja JTAG ni aktiviran (RTCK visok ali nepovezan), so TRST, TDI, TMS, TCK povsem navadni vhodi P1.31, P1.28, P1.30 in

P1.29 z vgrajenimi PULLUP upori. Ti štiri vhodi so na razpolago na udobni vtičnici JTAG, kjer z ozemljitvenimi mostički izbiramo opcije v programu. Pri programiranju (RTCK nizek ob RESETu) ti vhodni ne obstajajo in javljajo programu logično ničlo.

Opisana inačica programa za ArmTNC uporablja samo mostiček na TRST oziroma P1.31 na priključku 3. Z ozemljenim P1.31 ali vključenim JTAG deluje ATNC, bolj točno UART0, tako kot v celotnem opisu. ATNC tedaj uporablja 32kbyte pomnilnika in razpolaga s 15 pomnilniškimi bloki za okvirje NBP ne glede na vgrajeni mikrokrmilnik LPC2138 ali LPC2148.

Nepovezan ali visok P1.31 namenoma izključi UART0, da ne pride do neželjenega programiranja ATNCja, recimo v težko dostopnem vozlišču (ASV) na vrhu hriba. UART0 tedaj deluje kot papiga in ponavlja vse, kar sprejme s hitrostjo 9600bit/s, na delovanje in parametre ArmTNCja pa nima vpliva. Povrhu izklop UART0 sprosti en dodaten pomnilniški blok, da NBP razpolaga s 16 bloki pomnilnika.



Ob RESETu LPC2368 ali LPC2387 prečita stanje RTCK, ki je tedaj vhod z vgrajenim PULLUP uporom in lahko vključi TRACE na nogicah P2.0-P2.9, čeprav EATNC teh 10 nogic ne uporablja v noben drug namen. Opisana inačica programa Ethernet-ArmTNCja uporablja isti register

tudi za izbiro med čipoma LPC2368 in LPC2387:
LPC2368: RTCK na masi preko upora 1k8: 58kbyte in 20 blokov,
LPC2387: RTCK nepovezan, aktivni PULLUP: 98kbyte in 45 blokov.

Cigotag drži RTCK na masi preko vgrajenega PULLDOWN upora, da program dela s samo 58kbyte in 20 bloki ne glede na vgrajeni čip. Isti program za LPC2368 ali LPC2387 v ohišju s 100 priključki je seveda možno prenesti tudi na druge mikrokrmilnike iz družin LPC23xx oziroma LPC24xx. Drugi mikrokrmilniki so vgrajeni v ohišja s 144 oziroma 208 nogicami, kar pomeni nekoliko drugačno razmestitev priključkov. Količina pomnilnika žal nikjer ni večja kot pri LPC2387...

Mikrokrmilnike programiramo s FlashMagic preko UART0 tako, da na JTAG priključek postavimo na zadnje mesto mostiček za zagon BOOTLOADERja. V program FlashMagic moramo vstaviti vsaj približno vrednost takta procesorja. V ATNCju je to frekvenca kristala 14.7MHz (običajno). V EATNCju je to notranji RC oscilator mikrokrmilnika 4MHz, saj ob RESETu zunanji kristalni oscilator še ne deluje. BOOTLOADER v vsakem primeru uporablja AUTOBAUD na UART0, ki pri taktu 4MHz deluje vsaj do 19200bps, pri višjih taktih mikrokrmilnika pa lahko nastavimo FlashMagic še hitreje.

RESET: po RESETU se (E)ATNC (obvezno mostiček na TRST=P1.31 v ATNC!) javi z inačicama programa in mikrokontrolerja:

*** ArmTNC - S53MV 01.11.2010 ID=0002FF25 ***

oziroma:

*** Ethernet ArmTNC - S53MV 01.11.2010 ID=1800F935 ***

kjer je:

0002FF25 oznaka za čip LPC2138,

0402FF25 oznaka za čip LPC2148,

1600F925 oznaka za čip LPC2368 oziroma

1800F935 (starejši) ali 1700FF35 (novejši) čip LPC2387.

Vse nastavitve številnih parametrov lahko seveda opravimo preko terminala na UART0. Če odtipkamo nepoznan ukaz oziroma prazno vrstico (samo <enter>), (E)ATNC javi:

??? Nepoznan ukaz (A,B,C,F,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,Y,Z,?)

oziroma:

??? Nepoznan ukaz

(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,?)

Ukaz "?": ?<enter> nam izpiše kratka navodila kot pomoč pri programiranju:

*** Seznam veljavnih ukazov ArmTNC ***

A <IP> <naslov#1> <naslov#2> <naslov#3> - (A)RP usmerjanje

B <baudrate> - nastavi (B)audrate za UART1 (SLIP)

C <perioda> - (C)vekanje / kokodakanje (izhod: prazna vrstica)

F 123456789 - vpis trenutnih parametrov v (F)LASH in RESET ArmTNC

H <preklop> - oblika vnosa/izpisa naslovov N36/(H)EX

I <naslov#1> <naslov#2> <naslov#3> - naslov za SL(I)P okvirje

J <besedilo> - besedilo (J)avljanja (izklop: prazna vrstica)

K <ukaz> <geslo> - (K)omanda in geslo z izpisom ADC

L <naslov#1> <naslov#2> <naslov#3> - naslovi (L)okalne zanke

M <naslov> - (M)oj naslov (0=KISS RX/TX)
N <naslov#1> <naslov#2> <naslov#3> - (N)aslov za pogovor
O <stevilo> - meritev (O)dziva stevila byte, 0=izklop
P <stevilo> <zakasnitev> <minblok> - (P)onavljjanja
Q <quartz> <takt> - (Q)uartz in takt ARM procesorja
R - (R)ESET ArmTNCja
S <slot> <glava> <rep> - zakasnitve (S)lot, glava in rep
T <z DCD> <brez DCD> - (T)ecnoba preklopa na oddajo /65536
U - vkljuci izpis (U)porabnih okvirjev
V - vkljuci izpis (V)seh okvirjev
W <zakasnitev> - HayesAT (W)AIT
Y <m1> <o1> <e1> <m2> <o2> <e2> - (Y)skala/enota TLM 1/2
Z <vklop/izklop> - izpis %DCD, %PTT, prosto, (Z)ank/s, cas
? - izpisi ta seznam

oziroma:

*** Seznam veljavnih ukazov Ethernet ArmTNC ***

A <IP> <naslov#1> <naslov#2> <naslov#3> - (A)RP usmerjanje
B <baudrate> - nastavi (B)audrate za UART1 (SLIP)
C <perioda> - (C)vekanje / kokodakanje (izhod: prazna vrstica)
D <IP> <maska> <nacin> - IP (D)domena
E <hitrost-X> <naslov> - (E)thernet hitrost, MDIX in MAC naslov
F 123456789 - vpis trenutnih parametrov v (F)LASH in RESET EATNC
G <trajanje> - (G)lej EthernetARP cache
H <preklop> - oblika vnosa/izpisa naslovov N36/(H)EX
I <naslov#1> <naslov#2> <naslov#3> - naslov za SL(I)P okvirje
J <besedilo> - besedilo (J)avljanja (izklop: prazna vrstica)
K <ukaz> <geslo> - (K)omanda in geslo z izpisom ADC
L <naslov#1> <naslov#2> <naslov#3> - naslovi (L)okalne zanke
M <naslov> - (M)oj naslov (0=KISS ali SLIP<>Ethernet)
N <naslov#1> <naslov#2> <naslov#3> - (N)aslov za pogovor
O <stevilo> - meritev (O)dziva stevila byte, 0=izklop
P <stevilo> <zakasnitev> <minblok> - (P)onavljjanja
Q <quartz> <takt> - (Q)uartz in takt ARM procesorja
R - (R)ESET EATNCja
S <slot> <glava> <rep> - zakasnitve (S)lot, glava in rep
T <z DCD> <brez DCD> - (T)ecnoba preklopa na oddajo /65536
U - vkljuci izpis (U)porabnih okvirjev
V - vkljuci izpis (V)seh okvirjev
W <zakasnitev> - HayesAT/EthernetARP (W)AIT
X <vklop/izklop> - izpis MDI(X), POL, ANC, DUP, 10M, LINK, MAC
Y <m1> <o1> <e1> <m2> <o2> <e2> - (Y)skala/enota TLM 1/2
Z <vklop/izklop> - izpis %DCD, %PTT, prosto, (Z)ank/s, cas
? - izpisi ta seznam

Opis posameznih ukazov nadaljujem v smiselnem vrstnem redu, ki ga zahteva razumevanje delovanja (E)ATNCja:

Ukaz "F": trenutne nastavitve vpišemo v FLASH pomnilnik z ukazom:

F 123456789<enter>

Pri tem pride do popolnega RESETa (E)ATNCja. Število 123456789 je mišljeno kot ključ, da ne bi prišlo do nenamerne vpisa v FLASH. V primeru napačnega števila (E)ATNC javi napako:

*** Napacen kljuc ***

Ukaz "R": če nastavitve ne želimo shraniti v FLASH pomnilnik, pač pa bi (E)ATNC radi ponovno pognali s privzetimi nastavitvami iz FLASH, damo (E)ATNCju samo popoln RESET. To lahko storimo na dva načina: hardversko, s prekinitvijo napajanja oziroma z ukazom: **R<enter>**

Ob vpisu v FLASH pomnilnik oziroma kakršnemkoli drugem RESETu se (E)ATNC javi:

***** ArmTNC - S53MV 01.11.2010 ID=0002FF25 *****

oziroma:

***** Ethernet ArmTNC - S53MV 01.11.2010 ID=1800F935 *****

Ukaz "Q": nastavi taktno frekvenco ARM procesorja. Ker je to zelo nevaren ukaz, ki brez programatorja (H-JTAG ali FlashMagic) nima več poti nazaj, je temu primerno zavarovan. Frekvenco moramo v ukazu navesti dvakrat in to dve različni števili: najprej frekvenco kristala in potem frekvenco ARM procesorja, kot jo pomnoži PLL.

Obe navedeni frekvenci morata biti napisani v Hz (namenoma zelo dolga števila) v predpisanih mejah med 40MHz in 110MHz za takt ARM procesorja. Druga frekvenca mora biti natančen štirikratnik prve. Če tem pogojem ni zadoščeno, ukaz ne spreminja nastavitve, pač pa le prikaže trenutno nastavitve.

Ob zagonu (E)ATNC najprej preveri nastavljen frekvenco takta v FLASH pomnilniku. Če ta ni v predpisanih mejah, jo takoj zamenja s privzeto vrednostjo 58.9824MHz, kar ustreza kristalu za 14.7456MHz. Ob prvem zagonu (E)ATNCja vsebuje FLASH sektor 25 (nastavitve programa) same enice (vrednosti 0xFF), kar je zagotovo nedovoljena vrednost takta in se takoj zamenja s privzeto vrednostjo. Če se torej zmotimo pri nastavitvi takta z ukazom "Q", zagotovo pomaga brisanje FLASH sektorja 25 s programatorjem.

Nastavitve frekvence takta se takoj ob zagonu uporabi za: (1) nastavitve periode glavne ure TIMER0 na 1 mikrosekundo in (2) nastavitve bitne hitrosti UART0 na 9600bit/s. Nastavitve frekvence takta ne menja modula deljenja za UART1, pač pa vpliva na učinek in prikaz ukaza "B" (3). Končno se nastavitve frekvence takta uporablja tudi brisanje in programiranje FLASH pomnilnika z ukazom "F" (4).

Na primer, kristal 16MHz in pripadajočo frekvenco takta ARM procesorja nastavimo z ukazom: **Q 16000000 64000000<enter>**
(E)ATNC odgovori:

***** Quartz: 16000000Hz*4=takt: 64000000Hz (FLASH+RESET)**

Nova nastavitve taktne frekvence ARM procesorja se uporabi takoj v ukazih "B" (3) in "F" (4). Ura (1) in UART0 (2) se ponastavijo šele po shranjevanju nastavitve v FLASH z ukazom:
F 123456789<enter>

Ukaz "Z": preprost izpis stanja (E)ATNCja z ukazom: **Z<enter>**
(E)ATNC odgovori:

DCD: 0.0% PTT: 0.0% 44blok 75646zank/s 0d/0h/28min/29s

Odstotki DCD in PTT se računajo vsako sekundo s korakom ene

milisekunde (tisoč zaporednih vzorcev). sledi število prostih blokov pomnilnika, število glavnih zank programa v eni sekundi ter celotni čas delovanja (E)ATNCja od zadnjega RESETa v dnevih, urah, minutah in sekundah.

Izpis stanja vsako sekundo vključimo z ukazom: **Z 1<enter>**

(E)ATNC odgovori:

*** Izpis stanja: 1=VKLOP

```
DCD: 0.7% PTT: 0.9% 44blok 75556zank/s 0d/0h/28min/33s
DCD: 0.8% PTT: 0.5% 44blok 75671zank/s 0d/0h/28min/34s
DCD: 0.0% PTT: 0.0% 44blok 75611zank/s 0d/0h/28min/35s
DCD: 0.5% PTT: 0.9% 44blok 75552zank/s 0d/0h/28min/36s
DCD: 0.8% PTT: 0.5% 44blok 75679zank/s 0d/0h/28min/37s
DCD: 0.0% PTT: 0.0% 44blok 75611zank/s 0d/0h/28min/38s
```

Izpis stanja vsako sekundo izključimo z ukazom: **Z 0<enter>**

(E)ATNC odgovori:

*** Izpis stanja: 0=IZKLOP

Ob RESETu (E)ATNCja je sekundni izpis stanja vedno izključen, ne glede na vsebino FLASH pomnilnika.

Ukaz "S": z njim nastavimo časovne parametre preklapljanja radijske postaje na oddajo. Ti časovni parametri so nujno potrebni v kateremkoli načinu delovanja (E)ATNC (vključno s pretvornikom Ethernet<>SLIP, da se HDLC vmesnik ne zatakne).

Primer: **s 3000 1000 200<enter>**

Vse tri veličine so izražene v mikrosekundah, odgovor (E)ATNCja se glasi:

*** slot: 3000us Glava: 1000us Rep: 200us

Prvo število (3000) je slot-time, to je časovna rezina, v kateri (E)ATNC preverja zasedenost radijskega kanala (/DCD), preden preklopi na oddajo. Čas 3000us oziroma 3ms je povsem smiseln za 1.2288Mbit/s BPSK postajo, kjer en IP okvir traja do 10ms.

Drugo število je trajanje sinhronizacijske glave paketa, ki lahko vsebuje več okvirjev. Trajanje sinhronizacijske glave je odvisno predvsem od hitrosti preklopa radijske postaje na oddajo ter časovne konstante AGC v sprejemniku na drugem koncu zveze. Pri dolžini glave moramo upoštevati, da bo zakasnitev AGC sprejemnika (ZIF) dosti večja za močne signale! Navedeni primer 1000us je srednja vrednost za 1.2288Mbit/s BPSK postaje. 500us je spodnja meja za hitre ZIF-PSK postaje. Stare 13cm PSK postaje z dvojnimi mešanjem potrebujejo tudi 2000us, hitre UWBFM radijske postaje za 5.7GHZ pa manj kot 200us.

Tretje število je trajanje repa, ki se dodaja vsakemu okvirju znotraj paketa. Med oddajo repa mora HDLC vmesnik izprazniti vse notranje FIFO pomnilnike in dodati zaključno zastavico okvirja. Softverski HDLC v (E)ATNCju deluje z 32-bitnimi podatki, zato je trajanje repa odvisno od dolžine okvirja v bajtih. Okvirji dolžin 100, 101, 102 in 103 bajtov imajo različno dolge repe, okvirja dolžin 100 in 104 pa enako dolga repa! Nastavljena dolžina repa mora dovolj velika za vse okvirje, sicer se okvirji določenih dolžin

Tahko izgubijo.

Dolžina repa pri 1.2288Mbit/s se giblje okoli 100us. Priporočena vrednost v tem primeru je zato 200us. Stari AX.25 sprejemniki počasnih sogovornikov, ki imajo hardverski HDLC ali celo neučinkovit DMA kot izvorni AX.25 SuperVozelj, potrebujejo dodaten mrtvi čas repa okvirja do 500us za lastno delovanje, ki se prišteva potrebni dolžini repa okvirja!

Vsi trije časovni parametri se shranjujejo v FLASH z ukazom:
F 123456789<enter>

Ukaz "T": z njim nastavimo verjetnost oziroma tečnobo, da (E)ATNC ob izteku slot-time preklopi na oddajo. Primer: T 30000 300<enter>
Tečnoba se meri v enotah 1/65536, (E)ATNC odgovori:
*** Tecnoba: 30000/65536 brez DCD: 300/65536

Vrednost 65535 pomeni 99.9%. Prvo število (v primeru 30000) pomeni verjetnost 45.8%, da bo (E)ATNC preklopil na oddajo na nezasedenem kanalu (/DCD neaktiven). Drugo število (v primeru 300) pomeni verjetnost 0.5%, da bo (E)ATNC preklopil na oddajo v vsakem primeru, ne glede na stanje /DCD. Ta zadnja možnost je v resnici izhod v sili, ko se je nek drugi udeleženec zataknil na oddaji in blokira radijski kanal. Za drugo število izbiramo majhno vrednost, da preprečimo nepotrebne motnje v brezhibnem omrežju.

Oba parametra tečnobe se shranujeta v FLASH z ukazom:
F 123456789<enter>

Ukaz "M": nastavi lastni naslov (E)ATNCja za NBP: M S53MV<enter>
Ta ukaz nastavi lastni naslov S53MV (36-iško) in (E) ATNC odgovori:
*** Moj naslov: S53MV

Če za lastni naslov vstavimo 0, se NBP izključi in (E)ATNC dela v čisto navadnem "KISS" načinu (1) v poljubnem protokolu, ki prenaša UART1 oziroma kot pretvornik "Ethernet<>SLIP" (4) brez HDLC:
*** Moj naslov: 0 (KISS RX/TX)
oziroma:
*** Moj naslov: 0 (KISS ali SLIP<>Ethernet)

Lastni naslov je 32-bitno število, zapisano v 36-iški obliki (do 6 ali 7 znakov) oziroma v heksadecimalni obliki (z 8 znaki). NBP prepoveduje naslov iz samih ničel, naslov iz samih enic pa je namenjen VSEM (* ali FFFFFFFF) in ta ne sme biti dodeljen kot lastni naslov udeleženca. Lastni naslov različen od 0 preklopi (E)ATNC v način (2) "SLIP" ali (3) "NBP zanka" na UART1 in vse funkcije NBP delujejo.

Lastni naslov se shranjuje v FLASH z ukazom: F 123456789<enter>

Ukaz "H": omogoča izbiro med 36-iško obliko vnosa/izpisa vseh naslovov ter heksadecimalno obliko vnosa/izpisa naslovov. Primer:
M<enter>

```
*** Moj naslov: S53MV
H 1<enter>
*** Oblika: 1=HEX(8znakov)
M<enter>
*** Moj naslov: 032A3880
H 0<enter>
*** Oblika: 0=N36(crke+stevilke)
M<enter>
*** Moj naslov: S53MV
```

36-iška oblika omogoča nabor 36 različnih znakov: 26 velikih črk angleške abecede od "A" do "Z" in 10 arabskih številke od "0" do "9". (E)ATNC zapisuje 36-iška števila v obratnem vrstnem redu: prvi je najnižji znak in zadnji najvišji znak. Pri izpisu se vodilne ničle na koncu brišejo, torej "S53MV0" postane "S53MV".

Naslovi v 36-iškem sistemu so načeloma dolgi do 6 znakov, kar predstavlja:

$$6 * \log_2(36) = 31.02 \text{ bit}$$

Do 32 bitov manjka malo manj kot en bit, kar pomeni, da za določene naslove Ne-Brezhibnega Protokola potrebujemo še sedmi 36-iški znak, ki je lahko le "0" (se ne izpiše) ali "1". Najvišje možno 32-bitno število, 32 enic ali posebni NBP naslov VSEM se v opisanem 36-iškem sistemu zapiše kot "3Z141Z1". (E)ATNC omogoča poenostavljen vnos naslova VSEM z zvezdico "*" in ga tudi tako izpiše.

Heksadecimalni izpis naslova je namenoma vedno na vseh 8 znakov, da se vidno razlikuje od 36-iškega izpisa. Heksadecimalna števila lahko seveda vnesemo tudi z manj znaki. Heksadecimalni način uporablja običajni vrstni red vnosa oziroma izpisa znakov: prvi znak z leve je vedno najvišji znak, zadnji znak na desni pa najnižji znak.

Ob RESETu (E)ATNC privzame 36-iški vnos/izpis vseh naslovov, ne glede na vsebino FLASH pomnilnika.

Ukaz "B": nastavi bitno hitrost UART1, na primer: **B 921600<enter>**
(E)ATNC hitrosti ne nastavi takoj, čeprav takoj odgovori:
*** Baudrate UART1: 921600bit/s (FLASH+RESET)

Izbrano nastavitvev za UART1 moramo najprej shraniti v FLASH z ukazom: **F 123456789<enter>**
Vrednost se bo nastavila šele ob RESETu (E)ATNCja, ki samodejno sledi vpisu v FLASH. Če (E)ATNC RESETiramo brez vpisa v FLASH, bitna hitrost ostane takšna, kot je bila tam prej vpisana.

(E)ATNC nastavi bitno hitrost v skladu s tistim, kar zmorejo delilniki takta UART1 brez ulomkov. Bitna hitrost se vedno zaokrožuje na najbližjo možno vrednost. Ista bitna hitrost velja v vseh načinih delovanja (E)ATNCja: (1) KISS, (2) SLIP, (3) Ethernet/NBP zanka in (4) Ethernet<>SLIP pretvornik.

Ukaz "P": nastavi število in zakasnitev ponavljanj ter minimum

prostih blokov v NBP, na primer z ukazom: **P 33 10000 5<enter>**
(E)ATNC odgovori:

***** Ponovi: 33-krat Zakasnitev: 10000us Min: 5blok**

Prvo število 33 pomeni, da se oddaja nepotrjenega okvirja lahko ponavlja do 33-krat.

Drugo število 10000us (ali 10ms) je časovna konstanta, iz katere se izračuna čas ponavljanja nepotrjenega okvirja. Čas ponavljanja je sorazmeren številu ponavljanj, da se prezasedeno omrežje samodejno upočasni. Končni rezultat je pomnožen še z ključnim številom, da se ponavljanja različnih udeležencev ne prekrivajo.

Tretje število 5 pomeni minimalno število prostih blokov pomnilnika (E)ATNCja. Če število prostih blokov upade pod to vrednost zaradi velike obremenitve NBP, (E)ATNC odda okvirje samo še enkrat in jih ne ponavlja več, čeprav se števci ponavljanja posameznih okvirjev še niso iztekli. Po končani oddaji paketa (E)ATNC zavrže natančno toliko okvirjev, da se število prostih blokov povzpne na predpisano mejno vrednost. (E)ATNC vedno najprej zavrže najstarejše okvirje.

Če tretje število postavimo na 0, (E)ATNC ne zavrže nobenega starega okvirja, pač pa se začnejo izgubljati okvirji, ki kakorkoli prispejo v (E)ATNC, kar ni dobro. Če tretje število postavimo **15, 16, 20, 45** oziroma več kot je vseh blokov na razpolago, (E)ATNC ne ponavlja nobenega okvirja, čeprav ni bil potrjen. V tem zadnjem primeru postaneta prva dva parametra ponavljanja brezpredmetna, radijska zveza pa izgublja veliko okvirjev, saj mehanizem ponavljanja NBP ne dela več!

Vsi trije parametri P se shranjujejo v FLASH z ukazom:
F 123456789<enter>.

Ukaz "N": nastavi pot do sogovornika za neposredno teleprintersko zvezo (čvekanje), za daljinske ukaze ter za meritev odzivov. Čeprav NBP ne omejuje števila naslovov v poti, omejitev pomnilnika (E)ATNC omejuje število naslovov na 16. Torej največ 15 vozlišč do končnega naslovnika.

Ukaz: **N S53MV4 S53MV2 S55YFE<enter>**
nastavi pot do naslovnika S55YFE preko vozlišč S53MV4 in S53MV2.
(E)ATNC odgovori:
***** Naslov: S53MV4,S53MV2,S55YFE**

Pot se shranjuje v FLASH z ukazom: **F 123456789<enter>**.

Ukaz "C": preklopi (E)ATNC v pogovorni način (teleprintersko čvekanje) z ukazom: **C<enter>**
(E)ATNC odgovori:
***** Pogovorni način *****

Vse besedilo, ki sledi ukazu C, se oddaja v okvirjih kot koristni tovor. Začetek tovara z besedilom je označen z 32-bitno ničlo. Sprejemni (E)ATNC preko te ničle razločuje med pogovorom (lahko je

daljinski ukaz) in podatkovnimi okvirji drugih protokolov, ki nikoli ne začenjajo z ničlami!

V primeru, da je NBP izključen (M=0) oziroma ni prostega pomnilnika, (E)ATNC v čakalnem načinu na vsako odtipkano vrstico odgovori:

*** Sporocilo izgubljeno ***

Izhod iz pogovornega načina je prazna vrstica (samo <enter>), ki (E)ATNC preklopi nazaj v ukazni način in to tudi javi:

*** Ukazni nacin ***

Ukaz "C <perioda>" omogoča samodejno ponavljanje (kokodakanje) istega sporočila, ki ga vtipkamo le enkrat. Periodo kokodakanja se da nastaviti med 1s in 255s. Med kokodakanjem lahko vtipkamo novo, drugačno sporočilo oziroma ustavimo kokodakanje s prazno vrstico (samo <enter>).

Ukaz "C <perioda>" je primeren za opazovanje delovanja oddaljenega (E)ATNCja oziroma njegove telemetrije, na primer:

C 3<enter>

*** Kokodakanje s periodo 3s ***

/////<enter>

S53MV5>S53MV4,S53MV>**** DCD: 1.1% PTT: 1.3% 43blok 75567zank/s
0d/1h/17min/52s

/////

S53MV5>S53MV4,S53MV>**** DCD: 0.6% PTT: 1.1% 43blok 75553zank/s
0d/1h/17min/55s

/////

S53MV5>S53MV4,S53MV>**** DCD: 0.7% PTT: 1.1% 43blok 75548zank/s
0d/1h/17min/58s

/////

S53MV5>S53MV4,S53MV>**** DCD: 1.1% PTT: 1.0% 43blok 75552zank/s
0d/1h/18min/1s

<enter>

*** Ukazni nacin ***

Po vsakršnem RESETu se (E)ATNC vedno zažene v ukaznem načinu, ne glede na vsebino FLASH pomnilnika.

Ukaz "U": preklopi izpis samo na (u)porabne okvirje, to se pravi okvirje, katerih podatkovno polje začenja z 32-bitno ničlo in se nadaljuje z besedilom vsebine. Primer: U<enter>

*** Uporabni okvirji na zaslon ***

S53MV5>S53MV4,S53MV>hojhojhoj

Izpis vsebuje vse naslove in besedilo uporabnega okvirja. Kontrolni znaki v besedilu se ne izpišejo. Po RESETu (E)ATNC privzame izpis samo uporabnih okvirjev, ne glede na stanje FLASH.

Ukaz "V": preklopi izpis na (v)se okvirje, ne glede na to, za kakšno vrsto okvirjev gre, sprejetih ali oddanih, na HDLC radijskem vmesniku ali v lokalni NBP zanki. Primer: V<enter>

*** vsi okvirji na zaslon ***

R(36FA5905)S53MV5>S53MV4,S53MV>hojhojhoj

T(36FA5905)*

Izpis vsebuje najprej izvor okvirja: R=HDLCradioRX, T=HDLCradioTX, L=NBPzankaRX, S=NBPzankaTX. Sledi v oklepajih psevdonaključno število okvirja, za tem pa naslovi in besedilo povsem enako kot v primeru uporabnih okvirjev. (E)ATNC zna izpisati samo vsebino NBP okvirjev. Vsebine KISS ali SLIP okvirjev ne zna pravilno razvozljati.

Gornji primer prikazuje, kako sprejetemu podatkovnemu okvirju sledi potrditev. Ukaz V je torej mišljen predvsem nadzoru in učenju delovanja Ne-Brezhibnega Protokola. Če želimo opazovati celoten promet na radijskem kanalu, tudi tisti, ki naslovljen na naš (E)ATNC, niti VSEM, niti na našo NBP lokalno zanko, potem (E)ATNC preklopimo v način KISS!

Ukaz "L": z njim nastavimo do največ 8 naslovov ATNCjev oziroma EATNCjev, ki so v lokalni NBP zanki povezani v ArmsV. **Hkrati s tem ukazom vklopimo (seznam "L" različen od nič) Ethernet vmesnik EATNCja (v načinih 3 in 4) ali izklopimo (prazen seznam "L") Ethernet vmesnik EATNCja (v načinih 1 in 2).**

Z ukazom: **L S53MV2 S53MV4 S53MV<enter>**
vstavimo v lokalno zanko (E)ATNCje z naslovi S53MV2, S53MV4 in S53MV. (E)ATNC potrди ukaz z odgovorom:
***** Lokalni naslovi: S53MV2+S53MV4+S53MV**

Lastnega (E)ATNCja ni treba navesti. Naslovi morajo biti navedeni v pravilnem vrstnem redu, kot so (E)ATNCji povezani v lokalno zanko, sicer naslavljanje v načinu preskok ne bo delovalo pravilno! Razen lastnega naslova in naslova VSEM (E)ATNC posluša tudi na vseh naslovih, ki so navedeni v lokalni zanki.

Ko na radijski HDLC strani (E)ATNC sliši okvir, ki je naslovljen na enega od naslovov v lokalni zanki, potrди sprejem, zavrti naslovno polje, kot običajno in naslovnika zamenja z lastnim naslovom. Obdelani okvir potem pošlje v lokalno zanko in mu dodeli toliko življenj "A", kolikor je bil oddaljen pravi naslovník v lokalni zanki.

Ko se takšnemu okvirju izteče življenjska doma, naj bi prispel ravno do pravega (E)ATNCja prejemnika, ki že obdelani okvir samo posreduje na radijsko HDLC stran. Zamenjava naslova pri tem zagotavlja, da je v naslovnem polju zapisana pravilna pot za odgovor. Takšen preskok prihrani polovico naslovov v Ne-Brezhibnem Protokolu!

Naslov VSEM (* ali FFFFFFFF) ne sme v ukaz "L", ker (E)ATNC samodejno pošilja okvirje naslovljene VSEM na oba izhoda HDLC radio in lokalna NBP zanka, če je vključena. Naslov VSEM kjerkoli v seznamu "L" pomeni, da se vsi NBP okvirji usmerjajo v lokalno zanko, torej (E)ATNC sploh ne preklopi radijske postaje na oddajo.

Naslov VSEM (* ali FFFFFFFF) na prvem mestu v seznamu "L" še dodatno ob vsakem RESETU izključi prekinitev FIQ. Takšna nastavitvev je mogoče smiselna za (E)ATNC, ki (še) ni povezan na bitno

sinhronizacijo in radijsko postajo oziroma ga uporabljamo samo kot pretvornik Ethernet<>SLIP ali Ethernet<>NBP oziroma za dodatno telemetrijo in telekomando v lokalni zanki ArmsVja.

Izklop FIQ z naslovom VSEM na prvem mestu seznama "L" je smiseln predvsem v načinu (4) Ethernet<>SLIP pretvornik. ARM procesor se z izklopom hitrih prekinitev FIQ radijskega kanala bistveno razbremeni, kar omogoča Ping precej daljši od 8 blokov (11832byte) tudi pri 100Mbit/s Ethernet povezavi.

UART1 deluje v načinu (3) Ethernet-ATNC in lokalna NBP zanka, ko seznam "L" vsebuje vsaj en veljaven naslov. Istočasno z vklopom lokalne zanke se IP promet preusmeri na Ethernet vmesnik.

(E)ATNC preklopimo v način (2) SLIP z ukazom: L 0<enter>
in dobimo potrditev:
*** Lokalni naslovi: IZKLOP=SLIP
oziroma:
*** Lokalni naslovi:0 (KISS/SLIP in izklopljen Ethernet)

Lokalna NBP zanka uporablja povsem enako SLIP uokvirjanje kot KISS ali pravi SLIP, le da predstavljajo vsebino celi NBP okvirji. (E)ATNC sicer vsebuje dva različna sprejemnika za (2) SLIP in (3) lokalno NBP zanko zaradi različnega dodeljevanja pomnilnika.

ATNC v načinu (1) KISS lahko uporablja en ali drug sprejemnik, kar izbiramo z ukazom "L". Lokalna NBP zanka je v načinu KISS sicer vedno izključena, je pa njen sprejemnik hitrejši in boljši od SLIP, ki mora odgovarjati na neumne ukaze za WIN modem.

EATNC v načinu (1) KISS uporablja isti sprejemnik kot (2) SLIP z vgrajenim odzivom na HayesAT ukaze. Način (4) Ethernet<>SLIP pretvornik uporablja sprejemnik NBP lokalne zanke (3), ki NE podpira HayesAT ukazov in ni združljiv z WIN modemi.

Nastavitev načina delovanja in naslove lokalne zanke se shranjuje v FLASH z ukazom: F 123456789<enter>.

Ukaz "I": nastavi privzeto pot za podatkovne (IP in druge) okvirje v UART1 načinu (2) SLIP oziroma (3) Ethernet. Čeprav NBP ne omejuje števila naslovov v poti, omejitev pomnilnika (E)ATNC omejuje število naslovov na 16. Torej največ 15 vozlišč do končnega naslovnika.

Ukaz: I S53MV4 S53MV2 S55YFE<enter>
nastavi pot do naslovnika S55YFE preko vozlišč S53MV4 in S53MV2, torej deluje podobno kot ukaz "N". (E)ATNC odgovori:
*** IP naslov: S53MV4,S53MV2,S55YFE

Privzeta pot IP se shranjuje v FLASH z ukazom: F 123456789<enter>.

Ukaz "A": omogoča nastavitev dodatne NBP-ARP tabele usmerjanja IP okvirjev poleg privzete poti iz ukaza I. Ukaz A, ki mu sledi ena sama številka, določa način delovanja dodatnega NBP-ARP usmerjanja: A 0<enter>

```
*** ARP usmerjanje: 0=IZKLOP
44.150.1.91>S53MV4,S53MV5,S53MV
A 1<enter>
*** ARP usmerjanje: 1=BRANJE
44.150.1.91>S53MV4,S53MV5,S53MV
A 3<enter>
*** ARP usmerjanje: 3=VPIS
44.150.1.91>S53MV4,S53MV5,S53MV
A 2<enter>
*** ARP usmerjanje: 2=VPIS-PRIVZETO
44.150.1.91>S53MV4,S53MV5,S53MV
```

Če je vključen vpis v tabelo, je vnašanje poti v tabelo samodejno. Ročno dodamo pot tako, da ukazu A sledi IP številka in pot, sestavljena iz NBP naslovov:

```
A 44.150.1.113 S55YFE S50YFE<enter>
*** ARP usmerjanje: 2=VPIS-PRIVZETO
44.150.1.91>S53MV4,S53MV5,S53MV
44.150.1.113>S55YFE,S50YFE
```

Neželjeni vpis v NBP-ARP tabeli izbrišemo tako, da za A navedemo le IP številko:

```
A 44.150.1.113<enter>
*** ARP usmerjanje: 2=VPIS-PRIVZETO
44.150.1.91>S53MV4,S53MV5,S53MV
```

Velikost NBP-ARP tabele je omejena z razpoložljivim pomnilnikom na 10 vpisov za 10 različnih IP številok. Tabela se shrani v FLASH z ukazom: F 123456789<enter>.

Ukaz "w": omogoča nastavitve zakasnitve (timeout) za Hayes modemske protokole (načina 1 in 2) oziroma čas čakanja (timeout) na ARP odgovor pri Ethernetu (načina 3 in 4). Ker ta dva različna timeouta nista nikoli aktivna istočasno, si delita isto mesto v pomnilniku in isti ukaz za nastavljanje.

Hayes zakasnitev je potrebna za vzpostavitev SLIP povezave v WINXP, ko (E)ATNC odgovarja na "AT" modemske ukaze in na zaporedje "+++". Priporočena vrednost za HayesAT timeout je 0.6 sekunde ali 600000us. Pozor! HayesAT timeout je vključen tudi v načinu KISS. Premajhna vrednost lahko prepreči delovanje programov za AX.25.

Ko EATNC ne pozna ciljne MAC številke za določen IP4 podatkovni okvir, odda ARP vprašanje na Ethernetu. Pripadajoči podatkovni okvir čaka na ARP odgovor s ciljno MAC številko. Da ne bo čakal za vedno, je čas odgovora omejen z EthernetARP timeout. Windows običajno odgovorijo v 100us do 1ms. Priporočena vrednost za EthernetARP timeout je 10ms ali 10000us:

```
Dva primera:
W 600000<enter>.
*** HayesAT WAIT: 600000us
oziroma:
W 10000<enter>.
*** HayesAT/EthernetARP WAIT: 10000us
```

HayesAT/EthernetARP timeout (zakasnitev) se shranjuje v FLASH z ukazom: F 123456789<enter>.

Ukaz "E" (samo EATNC): omogoča nastavitve hitrosti in načina delovanja Ethernet PHY vmesnika DP83848C ter lastne MAC številke (48-bitnega naslova) Ethernet vmesnika. DP83848C lahko nastavimo za 10Mbit/s, za 100Mbit/s oziroma dopustimo, da samodejno izbere hitrost in druge podrobnosti Etherneta po postopku "Auto Negotiation". Ker se dve avtomatiki lahko do onemoglosti lovita za rep, je smiselno v vsaki Ethernet zvezi vsaj na eni strani avtomatiko izključiti.

Ethernet je sicer izredno obširen standard, ki opisuje računalniške zveze s hitrostmi od 1Mbit/s do 10Gbit/s po najrazličnejših prenosnih poteh: koaksialnem kablu, UTP paricah, svetlobnem vlaknu, radijski zvezi (WLAN) itd. Danes je najbolj razširjen UTP kabel s štirimi paricami, od katerih se pri hitrostih 10Mbit/s in 100Mbit/s uporabljata le dve parici, ena za sprejem in druga za oddajo podatkov v načinu FULL-DUPLEX, kar obvladata DP83848C in LPC2368/2387 v EATNCju.

Vsak Ethernet okvir vsebuje sinhronizacijsko zaporedje in podatkovno polje. Dolžina Ethernet okvirja je vedno celo število byte. Najmanjša dolžina Ethernet okvirja je 68byte, največja dolžina okvirja pa je običajno omejena na 1518byte. Podatkovno polje vsebuje naslov prejemnika, naslov pošiljatelja, vrsto tovora (lahko je tudi dolžina okvirja, kar se ne uporablja), sam koristni tovor in na koncu še 32-bitni CRC (4byte).

Ethernet MAC naslovi so dolgi 48bitov ali 6byte. Opis vrste tovora je dolg 2byte: 0x0800 pomeni tovor IP4 okvir, 0x0806 pa tovor ARP okvir. Celotna glava podatkovnega polja je torej dolga 14byte, CRC porabi 4byte in za koristni tovor ostane največ 1500byte. Pri tem je treba zelo kratke okvirje (ARP ali res kratki IP) dopolniti z ničlami do najmanjše dovoljene dolžine Ethernet okvirjev! Gornja meja 1500byte je danes privzeta vrednost za Internetni Protokol.

Vsak udeleženelec v istem Ethernet omrežju mora imeti drugačen MAC naslov. Poleg tega pri izbiri naslova pazimo, da je prvi byte 0x00 (ničla), ker so vrednosti različne od nič namenjene skupinskemu naslavljanju ("multicast address", na primer IP6 "host discovery" in "router discovery"). Ostalih 5 byte je načeloma poljubnih, a morajo biti različni od naslovov drugih udeležencev (domači računalniki, usmerjevalnik, strežnik itd).

Na sprejemu EATNC hardversko preverja naslov. Sprejema le Ethernet okvirje, naslovljene na njegovo lastno "unicast" MAC številko in na "broadcast" (VSEM) številko FF-FF-FF-FF-FF-FF. Vse ostale okvirje sprejemnik že hardversko zavrne, da ne obremenjujejo mikroprocesorja EATNCja. Na oddaji se EATNC vedno javlja z lastno MAC številko v vseh Ethernet okvirjih.

Hitrost in način PHY vnesemo kot eno desetiško številko: 10 za 10Mbit/s, 100 za 100Mbit/s. PHY EATNCja vedno deluje v duplex

načinu, kar je za UTP kabel in za sodobna MAC stikala oziroma usmerjevalnike tudi edino smiselno. Zamenjavo sprejemne in oddajne parice (MDIX) vsilimo tako, da vnesemo 11 oziroma 101. Vsa ostala števila pomenijo vklop avtomatike "Auto Negotiation".

Pri vnosu MAC naslova vstavimo med posamezne byte "-" (minus) ali drugačno ločilo, na primer:

```
E 11 00-AB-CD-12-34-56<enter>.
```

EATNC hitrosti in načina PHY ter MAC naslova ne nastavi takoj, čeprav takoj odgovori:

```
*** Ethernet: 10Mbit/s-X MAC: 00-AB-CD-12-34-56 (FLASH+RESET)
```

Izbrano hitrost in način PHY ter MAC naslov moramo najprej shraniti v FLASH z ukazom:

```
F 123456789<enter>.
```

Vrednosti se bodo nastavile šele ob RESETu EATNCja, ki samodejno sledi vpisu v FLASH. Če EATNC RESETiramo brez vpisa v FLASH, hitrost in način PHY ter MAC naslov ostanejo takšni, kot so bili tam prej vpisani.

Ukaz "X" (samo EATNC): izpis stanja Ethernet PHY DP83848C z ukazom:

```
X <enter>
```

EATNC odgovori:

```
MDI POL ANC DUP 10M LINK MAC: 00-10-DC-5B-96-CF
```

MDI pomeni, da sta sprejemna in oddajna parica na pričakovanih priključkih. Obratno MDIX pomeni, da sta sprejemna parica in oddajna parica UTP kabla zamenjani med sabo. Če je avtomatika vključena tudi na drugi strani zveze, lahko tudi ta menja parice in iz MDI oziroma MDIX sploh ne moremo ugotoviti, ali imamo križno vezan kabel!

POL se lahko pojavi le pri hitrosti 10Mbit/s in pomeni, da je polariteta signala napačna: med sabo zamenjani žici (+) in (-) parice Ethernet sprejemnika. PHY EATNCja jasno ne more ugotoviti, kako sta povezani žici parice oddajnika. Polaritete tudi ni možno ugotoviti pri hitrosti 100Mbit/s, ker se tam uporablja diferencialno kodiranje, ki je na polariteto signala neobčutljivo. PHY DP83848C zna samodejno popraviti napačno polariteto signala pri sprejemu 10Mbit/s. Žal mrežne kartice nekaterih PC računalnikov niti pripadajoči gonilniki za winXP tega ne znajo: javijo sicer vzpostavljeno zvezo, oddaja PC računalnik proti EATNCju gre, a sprejem EATNCja nazaj na PC računalnik ne!

ANC pomeni "Auto Negotiation Complete": PHY DP83848C se je s sogovornikom dokončno zmenil za hitrost, duplex in (če je potrebno) menjavo paric.

DUP pomeni sposobnost sogovornika za FULL-DUPLEX. EATNC to zahteva!

10M pomeni, da je PHY DP83848C zaznal 10Mbit/s signal oziroma smo ga v ta način dela prisilili z ukazom "E". EATNC bo sledil ugotovitvam PHY DP83848C in se bo preklopil na izmerjeno hitrost.

LINK pomeni vzpostavljeno zvezo: priključen UTP kabel, vključen sogovornikov PHY in zaznan veljaven signal v obeh smereh zveze.

MAC: 00-10-DC-5B-96-CF je Ethernet naslov sogovornika. Če ima EATNC v omrežju več sogovornikov, se tu izpiše zadnji slišani MAC.

Izpis stanja PHY vsako sekundo vključimo z ukazom: `X 1<enter>`
EATNC odgovori:

```
*** Izpis PHY: 1=VKLOP
MDIX MAC: FF-FF-FF-FF-FF-FF
MDI MAC: FF-FF-FF-FF-FF-FF
MDIX MAC: FF-FF-FF-FF-FF-FF
MDI MAC: FF-FF-FF-FF-FF-FF
MDIX ANC DUP 10M LINK MAC: FF-FF-FF-FF-FF-FF
MDIX ANC DUP 10M LINK MAC: FF-FF-FF-FF-FF-FF
MDIX ANC DUP 10M LINK MAC: 00-22-64-5C-4B-9C
MDIX ANC DUP 10M LINK MAC: 00-22-64-5C-4B-9C
MDIX ANC DUP 10M LINK MAC: 00-22-64-5C-4B-9C
```

Gornji primer prikazuje priklop Ethernet kabla, ko je EATNC nastavljen na "Auto Negotiation". Prva štiri stanja so brez priključenega UTP kabla in tedaj PHY DP83848C preko postopka "Auto Negotiation" poizkuša z menjavo paric. Po priklopu kabla oziroma vklopu Ethernet sogovornika potrebuje "Auto Negotiation" nekaj sekund, preden javi vzpostavljeno zvezo in zaključek dogovarjanja. Končno steče promet in se vpiše MAC naslov sogovornika.

Izpis stanja PHY vsako sekundo izključimo z ukazom: `X 0<enter>`
EATNC odgovori:

```
*** Izpis PHY: 0=IZKLOP
```

Ob RESETu EATNCja je sekundni izpis PHY vedno izključen, ne glede na vsebino FLASH pomnilnika.

Ukaz "D" (samo EATNC): omogoča nastavitve IP domene z dvema maskama: katera ARP vprašanja EATNC sprejema in katera ARP vprašanja EATNC zavrača. Pravilna nastavitve IP4 domene zahteva podrobno razumevanje delovanja Ethernet-ARP (Address Resolution Protocol). ARP poskrbi za pravilno posredovanje IP4 okvirjev, ko zveza uporablja še dodaten protokol na nižji ravni MAC s svojimi lastnimi MAC naslovi.

Najpreprostejši način prenosa IP4 okvirjev je SLIP. SLIP vsebuje goli IP4 okvir, ki ima v svoji glavi vpisana IP4 naslova prejemnika in pošiljatelja. Ta naslovna informacija je povsem zadostna, da gre okvir po pravi poti v širni svet izbranemu naslovniku naproti. Pri nastavitvi SLIP v windows moramo zato nastaviti le lastno IP4 številko. Poleg nje še IP4 številko (številke) enega ali več imenskih strežnikov (Domain Name Server ali DNS), kar izkoriščajo protokoli na višji ravni (recimo HTTP).

Pri prenosu IP4 okvirjev preko omrežij AX.25 oziroma NBP moramo koristni tovor pravilno usmeriti po naslovih AX.25 oziroma NBP. V ta namen imajo Megabitni AX.25 TNCji oziroma (E)ATNCji vgrajeno privzeto pot in AX.25-ARP oziroma NBP-ARP avtomatiko. Posamezne poti do znanih ciljev lahko seveda vpišemo tudi ročno.

Pri prenosu IP4 okvirjev preko Ethernet omrežja moramo koristni tovor pravilno usmeriti na pravi Ethernet MAC naslov. Čeprav so

Ethernet zveze izključno točka-točka in natančno naslavljanje znotraj IP4 tovora to omogoča, programska oprema naših računalnikov IZRECNO PREPOVEDUJE, da bi IP4 okvirje preprosto pošiljali VSEM na "broadcast" Ethernet MAC naslov FF-FF-FF-FF-FF-FF. Katerikoli Ethernet vmesnik mora zato vsebovati usmerjevalno tabelo, ki IP4 naslovom dodeljuje pripadajoče Ethernet MAC naslove.

Usmerjevalno tabelo IP4<>Ethernet-MAC lahko vpišemo ročno. Običajno tabelo samodejno gradi protokol Ethernet-ARP, ki je vgrajen v programsko opremo naših računalnikov. Poleg prenosa podatkov opravlja Ethernet zveza še dodatno nalogo usmerjanja prometa, da nesodelujočih udeležencev niti prenosnih poti do njih ne obremenjujemo s prometom, ki njim ni namenjen. Pozor, Ethernet-ARP je čisto svoj protokol, ki je neodvisen in deluje drugače od AX.25-ARP ali NBP-ARP avtomatike, čeprav je končni namen podoben!

Ko IP4 tovor želi preko Etherneta, Ethernet vmesnik pogleda v lastno tabelo usmerjanja. Če v njej ni IP4 naslova prejemnika, Ethernet vmesnik pošlje najprej Ethernet-ARP vprašanje na "broadcast" naslov VSEM FF-FF-FF-FF-FF-FF. Prejemnikov Ethernet vmesnik se na takšno vprašanje javi z lastno Ethernet-MAC številko v Ethernet-ARP odgovoru. Ethernet pošiljatelj lahko zdaj usmeri čakajoči IP4 okvir na pravi MAC naslov in si hkrati zapomni pot v lastni tabeli usmerjanja. Vnosi IP4<>MAC v tabelo usmerjanja se samodejno brišejo čez določen čas (običajno 2min do 10min), ko se postopek ARP ponovi, da se vsebina tabele samodejno osvežuje.

Delovanje Ethernet-ARP v naših domačih računalnikih, usmerjevalnikih itd je še dodatno zakomplicirano z IP4 domeno z namenom, da se zmanjša velikost usmerjevalnih tabel in pogostnost ARP vprašanj. IP4 domena računalnika ali usmerjevalnika potrebuje tri nastavitve: naslov domene, masko in naslov privzetega prehoda (Default Gateway). Naslov domene je običajno kar enak naslovu udeleženca. Maska pove, kateri del naslova udeleženca opisuje domeno.

Na primer, če sem na mojem računalniku nastavil IP4 naslov 44.150.1.146 in masko 255.255.255.0, to pomeni, da je moja IP4 domena katerakoli IP4 številka, ki začneja s 44.150.1... Katerikoli prejemnik iz skupine 44.150.1... se smatra za bližnjega sogovornika v Ethernet omrežju in ARP vprašanja se naslavljaajo neposredno nanj.

Vsi ostali IP4 naslovi, ki so izven izbrane domene 44.150.1..., to je širno svetovno internetno omrežje, naj bi bili dostopni le preko privzetega prehoda. Zato nima smisla pošiljati številnih ARP vprašanj za vsak tak zunanji IP4 naslov posebej, pač pa le eno samo ARP vprašanje za IP4 naslov privzetega prehoda (Default Gateway). Windows dodatno zahteva, da se privzeti prehod nahaja znotraj IP4 domene!

Opisani način razmišljanja s privzetim prehodom se poruši, ko začnemo graditi obsežno radijsko omrežje s številnimi neodvisnimi povezavami v širni svet interneta. Obširno radijsko omrežje torej ni izvedljivo s programsko opremo cenениh WLAN škatlic. Privzeti prehod vedno zahteva omrežje z drevesno strukturo, strogo hierarhijo in enim samim izhodom v širni svet interneta, torej enakovredno poštarski telefonski centrali. Zanke v omrežju, rezervne poti in

dodatne povezave v širni internet niso dovoljene!

Za gradnjo obširnega radijskega omrežja torej potrebujemo nekaj več! Zelo preprosta in učinkovita rešitev je Ne-Brezhibni Protokol in Ethernet-ArmTNC. Povsem jasno mora predlagana rešitev nadomestiti vse pomanjkljivosti in biti hkrati povsem združljiva z neumnimi WLAN škatlicami in še bolj neumnimi windows računalniki. Na strani uporabnika NBP-(E)ATNC lahko izkoristi naslov VSEM in samodejno gradnjo NBP-ARP tabele za iskanje poti v širni svet. Na strani strežnika (WLAN omrežja) EATNC uporablja dvojno IP4 domeno: sprejema ARP poizvedovanja za določeno domeno in zavrača ARP poizvedovanja za določeno poddomeno.

Primer: internetni prehod S50YFE z naslovom 44.150.8.1 ima v svojem lokalnem Ethernet omrežju še IP-webcam z naslovom 44.150.8.2, windows računalnik z naslovom 44.150.8.3 in dva EATNCja. Domene vseh treh IP škatlic: usmerjevalnik S50YFE, IP-webcam in windows računalnik so nastavljene na 44..., torej maska 255.0.0.0.

Prvi EATNC je namenjen dostopu iz omrežja NBP: sprejema ARP poizvedovanja za domeno 44... in zavrača ARP poizvedovanja za poddomeno 44.150.8... Drugi EATNC je uporabljen kot pretvornik Ethernet<>SLIP za Megabitni TNC za dostop iz omrežja AX.25: sprejema ARP poizvedovanja za domeno 44.150.8... in zavrača ARP poizvedovanja za tri lokalne naslove Ethernet omrežja.

Prvi EATNC nastavimo: **D 44.150.8.0 255.0.0.0 255.255.255.0<enter>**

Prvi EATNC (omrežje NBP) odgovori:

***** IP domena: 44.150.8.0 Sprejmi: 255.0.0.0 Zavrni: 255.255.255.0**

Drugi EATNC (prehod Ethernet<>SLIP) nastavimo:

D 44.150.8.0 255.255.255.0 255.255.255.252<enter>

Drugi EATNC odgovori:

***** IP domena: 44.150.8.0 Sprejmi: 255.255.255.0 Zavrni: 255.255.255.252**

IP4 naslov in obe maski zapišemo kot štiri desetiške številke, ločene s pikami ali drugimi ločili. Za maske velja običajni dogovor iz IP sveta: desetiška števila pretvorimo v dvojiška, logična enica pomeni, da se bit naslova na tem položaju primerja, logična ničla pomeni, da se bit naslova na tem položaju ne primerja. Sicer nerodno, ampak pravila igre IP4 in Ethernet so vkovana v zvezde in ni ga korenjaka, ki bi se opogumil popraviti nesmisle.

Maska "Sprejmi: 0.0.0.0" pomeni, da se noben bit naslova ne primerja in EATNC odgovarja na vsa ARP poizvedovanja. Maska "Zavrni: 255.255.255.255" pomeni, da EATNC zavrača natančno en sam naslov v ARP poizvedovanjih. To navsezadnje ni omejitev, saj se mora vsaj en udeleženec nahajati na drugi strani Ethernet omrežja... Povrhu EATNC že sam po sebi nikoli ne odgovarja na ARP poizvedovanja, ki imajo enaka izvorni in ciljni IP naslov (da se winXP ne zakantajo...)

Če je na EATNC priključen en sam računalnik, načeloma kakšne posebne nastavitve IP domene sploh ne potrebujemo. Zaradi neumnosti v nekaterih inačicah windows je smiselno nastaviti IP4 naslov in domeno "sprejmi" povsem enako kot v windows. Poddomeno "zavrni" nastavimo na 255.255.255.255, da zavračamo lastni IP naslov.

Nastavitev IP domene, mask sprejmi in zavrni se shranjuje v FLASH z ukazom: `F 123456789<enter>`.

Ukaz "G" (samo EATNC): omogoča vpogled v Ethernet-ARP cache (tabelo usmerjanja) ter nastavitev življenjske dobe (trajanja) vseh vpisov v cache. Tabela usmerjanja se gradi kot rezultat lastnega ARP poizvedovanja EATNCja oziroma ARP odgovorov drugih udeležencev v Ethernet omrežju. Razlika med ukazoma "D" in "G" je torej v smeri ARP poizvedovanja: "D" vpliva na rezultat poizvedovanja drugih, "G" pa na uporabo lastnega poizvedovanja.

Čeprav je uporaba pomnilnikov in časovnih konstant v živem nasprotju s smernicami Ne-Brezhibnega Protokola, se mora EATNC tu podrediti izredno neumnim, ampak standardiziranim računalniškim protokolom. Ethernet-ARP cache v EATNC dovoljuje do največ 25 vpisov. Pri prekoračitvi pomnilnika se najprej brišejo najstarejši vpisi v tabeli usmerjanja.

EATNC za svoja lastna ARP poizvedovanja ne uporablja niti IP4 domene, niti maske niti privzetega prehoda. EATNC preprosto pošlje ARP vprašanje za vsak novi IP4 naslov cilja. Takšna rešitev se je izkazala preprosta in učinkovita. Tabela s 25 vpisi je v vsakem primeru zadosti velika, da ARP poizvedovanja niso preveč pogosta.

Trajanje vpisov nastavimo z ukazom: `G 100<enter>`
EATNC odgovori:

```
*** EthernetARP cache trajanje: 100s  
44.150.1.146>00-01-02-DD-06-21 187us/81s  
44.150.1.91>00-10-DC-5B-96-CF 92us/63s  
44.150.1.223>00-22-64-5C-4B-9C 404us/31s
```

Priporočena življenjska doba vpisov je 100 sekund ali manj. Če nastavimo trajanje 0, se ARP cache ne uporablja in EATNC za vsak podatkovni okvir sprašuje z ARP.

Seznam vpisov vsebuje IP4 naslov, pripadajoči Ethernet MAC naslov, čas odziva na ARP vprašanje v mikrosekundah in preostalo življenjsko dobo vpisa v sekundah. Gornji primer prikazuje vsebino ARP cacheja, ko so na EATNC priključeni preko Ethernet-MAC stikala trije računalniki in vsi trije istočasno brskajo po internetu preko EATNCja.

Čas odziva na ARP vprašanje nam pove, kako moramo nastaviti EthernetARP timeout z ukazom "w" s primernim varnostnim faktorjem. Če se v seznamu pojavi vpis, ki ima čas odziva enak nič mikrosekund, (0us), gre za ARP vprašanje sogovornika. Lahko je tudi "gratuitous" ARP vprašanje, ki ga pošiljajo windows ob vzpostavitvi Ethernet zveze in nanj ne smemo odgovarjati. Če se v seznamu pojavi vpis, ki ima čas odziva enak timeoutu, gre za zakasneni ali nepoklicani ARP odgovor. Verjetno bo treba povečati timeout z ukazom "w".

Seznam ARP vpisov lahko vsebuje več vpisov za isti par IP in MAC številka. EATNC bo v vsakem primeru uporabil samo prvi veljaven vpis v seznamu za posredovanje podatkovnega okvirja.

RESET vedno briše celoten ARP seznam. Nastavitev trajanja vpisov v EthernetARP cache se shranjuje v FLASH z ukazom: **F 123456789<enter>**

Daljinski ukazi se lahko izvajajo, ko (E)ATNC prejme besedilo, ki takoj za 32-bitno ničlo vsebuje še zaporedje štirih ulomkovih črt "////". Štirim ulomkovim črtam sledi en ali več znakov, ki določajo daljinski ukaz. Primeri daljinskih ukazov, kjer udeleženec S53MV5 pošilja ukaze preko posrednika S53MV4 in se udeleženec S53MV nanje samodejno javlja (daljinski ukazi vključeni z "J"):

```
////<enter>
S53MV5>S53MV4,S53MV>**** Nepoznan daljinski ukaz (#,/,>,?,@) ****
////#<enter>
S53MV5>S53MV4,S53MV>**** MDI ANC DUP LINK MAC: 00-10-DC-5B-96-CF
/////<enter>
S53MV5>S53MV4,S53MV>**** DCD: 1.2% PTT: 1.5% 43blok 75632zank/s
0d/3h/48min/43s
////><enter>
S53MV5>S53MV4,S53MV>****>
////?<enter>
S53MV5>S53MV4,S53MV>**** S53MV JN65TW 1276.87MHz 2.0000Mbit/s
////@<enter>
S53MV5>S53MV4,S53MV>**** 1:80=7.7% 2:356=11.1V K:0
```

Da ne pride do neskončnega zankanja, se zaporedje znakov "////" v odgovoru vedno zamenja z zaporedjem znakov "****". Odgovor na nepoznan ukaz vsebuje kratek seznam veljavnih ukazov.

Daljinski ukaz "#" izpiše PHY stanje EATNCja drugega udeleženca na povsem enak način kot ukaz "X" na lastnem EATNCju.

Daljinski ukaz "/" izpiše stanje (E)ATNCja drugega udeleženca na povsem enak način kot ukaz "Z" na lastnem (E)ATNCju.

Daljinski ukaz ">" vrne nespremenjen ostanek vsebine ukaza za meritev odziva in druge preizkuse zveze.

Na daljinski ukaz "?" se (E)ATNC javi z vnaprej pripravljenim besedilom javljanja (običajno zanimive informacije o sogovorniku, ki jih je vpisal z ukazom "J").

Daljinski ukaz "@" je namenjen nadzoru (dve analogni vrednosti) in upravljanju (ena komanda vklop-izklop ali RESET) vozlišča ASV.

Daljinski ukaz: **////@<enter>**
izpiše izmerjeni analogni vrednosti (telemetrijo) in stanje telekomande na povsem enak način kot ukaz "K" na lastnem (E)ATNCju.

Daljinski ukaz: **////@ <ukaz> <geslo><enter>**
se izvede samo takrat, ko je geslo pravilno oziroma izključeno: vključi (ukaz=1) oziroma izključi (ukaz=0) izhod za telekomando.

Daljinski ukaz: **////@ <geslo> <geslo><enter>**
daljinsko RESETira (E)ATNC samo takrat, ko je geslo pravilno oziroma izključeno (nastavljeno na FFFFFFFF).

(E)ATNC na drugem koncu zveze S53MV pove na svojem UART0, kaj je javljal (primer za izpis samo uporabnih okvirjev):

```
S53MV>S53MV4,S53MV5>////  
!!! Javljam nepoznan ukaz !!!  
S53MV>S53MV4,S53MV5>////#  
!!! Javljam PHY stanje !!!  
S53MV>S53MV4,S53MV5>/////  
!!! Javljam stanje !!!  
S53MV>S53MV4,S53MV5>////>  
!!! Merilni odziv !!!  
S53MV>S53MV4,S53MV5>////?  
!!! Javljam besedilo !!!  
S53MV>S53MV4,S53MV5>////@  
!!! Telekomanda in javljanje ADC !!!
```

Ukaz "J": omogoča nastavitev vnaprej pripravljenega besedila javljanja. Običajno so to zanimive informacije o vozlišču oziroma radijski postaji, na primer radioamaterski klicni znak, lokator, nadmorska višina, frekvenca, antena ipd.

Ukaz: **J<enter>**

prikaže nastavljeno besedilo javljanja:

```
*** Javi: S53MV JN65TW 1276.87MHz 2.0000Mbit/s
```

Ukaz: **J<presledek><enter>**

briše nastavljeno besedilo javljanja in izključi vse daljinske ukaze vključno z javljanjem:

```
*** Javljanje izklopljeno ***
```

Ukaz: **J<presledek>besedilo s podatki o postaji S53MV<enter>**

vstavi besedilo in vkluči vse daljinske ukaze:

```
*** Javi: besedilo s podatki o postaji S53MV
```

Besedilo samodejnega javljanja je omejeno s pomnilnikom na 64 znakov. Besedilo in posledično vklop/izklop daljinskih ukazov se shranjuje v FLASH z ukazom: **F 123456789<enter>**.

Ukaz "K": omogoča preverjanje delovanja telemetrije in telekomande (E)ATNC (ASV) ter nastavitev gesla za telekomando. Telemetrija omogoča meritev dveh analognih napetosti na A/D vhodih **P0.28** in **P0.25** v ATNC oziroma **P0.23** in **P0.26** v EATNC v razponu od 0V do +3.3V, kjer 10-bitni A/D pretvornik daje rezultat med 0 in 1023. Rezultat se lahko dodatno preračuna in izpiše z mersko enoto, kar nastavimo z ukazom "Y".

Telekomanda omogoča daljinski vklop oziroma izklop digitalnega izhoda **P1.16** v ATNC oziroma **P1.31** v EATNC in se izvede vedno ne glede na prisotnost gesla. Geslo je 32-bitno število zapisano v heksadecimalni obliki, kjer vrednost FFFFFFFF pomeni izklop preverjanja gesla pri daljinskem ukazu "@".

Ukaz: **K 1<presledek>ABC123<enter>**

vkluči **P1.16** oziroma **P1.31** ter nastavi geslo "0000A123":

*** 1:80=7.7% 2:0=0.0V K:1 Geslo: 00ABC123

Telemetrijska vhoda P0.28 in P0.25 ter telekomandni izhod P1.16 so napeljeni na priključek ATNC za analogno radijsko postajo. Pri tem je P0.28 izmenično sklopljen in torej vhodno vezje potrebuje predelavo za merjenje enosmerne napetosti. Izhod P1.16 krmili NPN tranzistor, ki lahko preklaplja višje napetosti in tokove.

Telemetrijska vhoda P0.23 in P0.26 ter telekomandni izhod P1.31 so napeljeni na priključek EATNC za analogno radijsko postajo. Pri tem P0.23 omogoča predelavo za uporovni delilnik oziroma izmenični sklop. Izhod P1.31 krmili NPN tranzistor, ki lahko preklaplja višje napetosti in tokove.

Telekomandni izhod P1.16 oziroma P1.31 je ob RESETu (E)ATNCja vedno izklopljen. Nastavljeno geslo se shrani v FLASH z ukazom:
F 123456789<enter>

Ukaz "Y": omogoča nastavitvev preračunavanja rezultata A/D pretvorbe in izpis merske enote. Z ukazom "Y" vstavimo šest 32-bitnih števil v heksadecimalni obliki. Rezultat A/D pretvorbe se najprej pomnoži s prvim številom, nato se deli s 1024 in končno se mu prišteje drugo število. Tretje število predstavlja štiri byte za štiri ASCII znake merske enote.

Preostala tri števila se uporabljajo na enak način za drugi telemetrijski kanal. Prvo in drugo število sta lahko predznačena (negativna). Končni rezultat se izpiše na eno decimalno mesto, torej dodatno deljen z 10. Če ima merska enota manj kot štiri ASCII znake, neuporabljene znake nadomestimo z ničlami, ki se ne izpišejo.

Primer: Y 3DC 0 25 140 0 56<enter>

(E)ATNC odgovori:

*** Yskala/enota: 00003DC 0000000 0000025 0000140 0000000
0000056

Prva konstanta oziroma množitelj 0x3DC je v desetiški obliki 1000. Druga konstanta oziroma odmik je 0. Tretja konstanta 0x25 je ASCII znak "%" odstotek. Prvi telemetrijski kanal se bo torej izpisoval v odstotkih od "0.0%" do "99.9%".

Četrta konstanta oziroma množitelj 0x140 je v desetiški obliki 320. Peta konstanta oziroma odmik je 0. Šesta konstanta 0x56 je ASCII velika črka "V". Drugi telemetrijski kanal se bo torej izpisoval v voltih od "0.0V" do "31.9V".

Ukaz "O": omogoča natančno meritev odzivnih časov in posledično kakovosti celotnega omrežja, ki uporablja Ne-Brezhibni Protokol. Ukaz "O" deluje na podoben način kot Ping v Internetnem Protokolu: vsako sekundo odda vprašanje v obliki daljinskega ukaza ter izmeri skupni čas potovanja vprašanja in odgovora.

Tovor preizkusnega okvirja vsebuje 32-bitno ničlo za oznako besedila, zaporedje "////" za daljinski ukaz, zaporedje ">>>>" za

oznako meritve odziva, 32-bitni čas oddaje okvirja in izbrano število byte kot preizkusni del tovora. Preizkusni del tovora se izračuna kot ostanek polinomskega deljenja 32-bitne ure s primitivnim polinomom $1+X^{28}+X^{31}$. Preizkusna vsebina je torej psevdonaključno dvojiško zaporedje, ki ga je enostavno računati in preverjati na sprejemu ter vsebuje res dobro premešane enice in ničle.

Ker se kot seme psevdonaključnega izvora uporablja ura (E)ATNCja, je vsak preizkusni okvir drugačen. Takšen preizkus zveze je torej dosti bolj verodostojen kot Ping iz WIN, ki vedno uporablja isto zaporedje "abcdef..." ali kar je še slabše, ničle v Linux-u. V ukazu "0" navedemo dolžino preizkusnega tovora podobno kot v Ping. Dolžina preizkusnega tovora je navzgor omejena na 1488byte, kar daje skupaj z glavo koristen tovor dolžine 1504byte. Preizkus zveze s 1000byte poženemo z: `0 1000<enter>`

(E)ATNC potrdi ukaz ter za vsako oddano vprašanje izpiše klicaj "!". Če (E)ATNC v kateremkoli okvirju z besedilom zazna zaporedje ">>>>" na pravem mestu, takšen okvir razume kot merilni okvir, izračuna čas potovanja okvirja ter preveri vsebino preizkusnega tovora okvirja, kjer vsak napačno sprejeti byte šteje kot ena napaka:

```
*** odziv: 1000byte
!S53MV5>S53MV4,S53MV> 43.5ms 1000byte 0napak
!S53MV5>S53MV4,S53MV> 82.4ms 1000byte 0napak
!S53MV5>S53MV4,S53MV> 26.6ms 1000byte 0napak
!S53MV5>S53MV4,S53MV> 28.0ms 1000byte 0napak
!S53MV5>S53MV4,S53MV> 53.4ms 1000byte 0napak
!S53MV5>S53MV4,S53MV> 45.6ms 1000byte 0napak
!S53MV5>S53MV4,S53MV> 52.9ms 1000byte 0napak
!S53MV5>S53MV4,S53MV> 53.0ms 1000byte 0napak
```

Meritev odzivnih časov preprosto izključimo tako, da postavimo dolžino preizkusnega tovora na nič: `0 0<enter>`

(E)ATNC odgovori:

```
*** odziv: 0=IZKLOP
```

Po kakršnemkoli RESETu (E)ATNCja je meritev odzivnih časov vedno izključena, ne glede na vsebino FLASH.

Pes čuvaj: (E)ATNC ima vgrajenega psa čuvaja, ki lahko v primeru očitne napake povzroči popolni RESET (E)ATNCja. Pes čuvaj ovohava preklon sprejem/oddaja iz milisekundne zanke programa. Če v času 7 sekund pes čuvaj niti enkrat ne zazna PTT manjši od 98%, sproži popolni RESET. Tak dogodek ima lahko več vzrokov znotraj in zunaj (E)ATNCja.

Zunanji vzrok RESETa je lahko hardverska napaka, na primer prekinitev takta za HDLC vmesnik. (E)ATNC se pri tem lahko zatakne na oddaji in čez 7 sekund pes čuvaj ugrizne z RESETom. Zunanji razlog je lahko tudi JTAG programator, ki zaustavlja procesor.

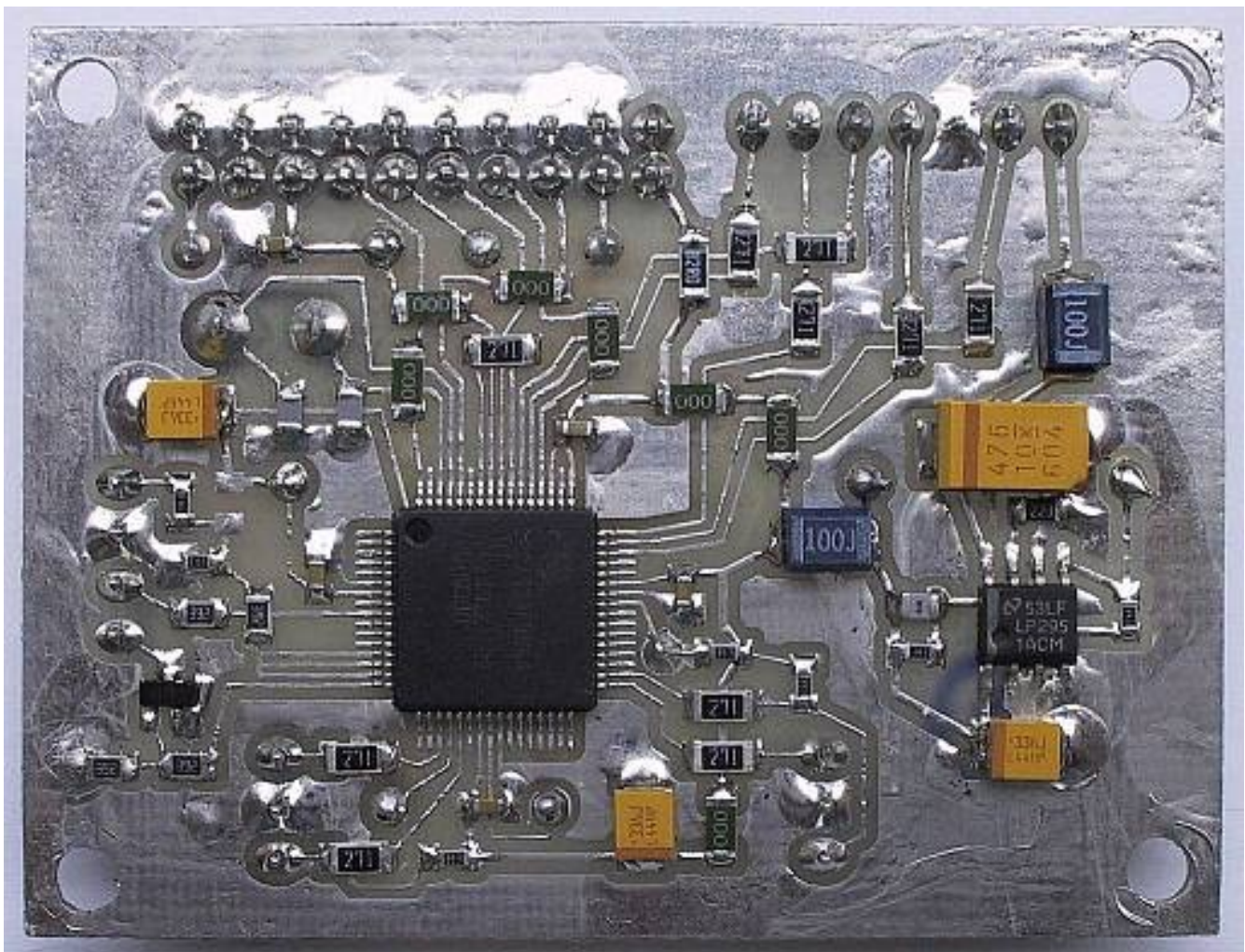
Notranji razlog RESETa je lahko preobremenitev procesorja, če so hitrosti HDLC in UART1 nastavljene previsoko in sta oba vmesnika obremenjena z gostim prometom. Če se procesor zatakne in se glavna

zanka ne izvaja več, čez 7 sekund pes čuvaj ugrizne z RESETom.

Pes čuvaj ATNCja uporablja isti takt procesorja in je odvisen od kristalnega oscilatorja ter nastavitvev PLLja za množenje taktne frekvence. Pes čuvaj torej deluje v vsakem primeru, da se le takt procesorja popolnoma ne ustavi.

Pes čuvaj EATNCja uporablja notranji RC oscilator 4MHz mikrokontrolerov LPC2368 oziroma LPC2387. Torej je pes čuvaj popolnoma neodvisen od kristalnega oscilatorja in nastavitvev PLLja za množenje taktne frekvence.

Pomnilnik LPC2138 (ATNC) vključuje 32kbyte hitrega RAM neposredno na vodilu ARM procesorja. Od teh jih je 24kbyte namenjeno 15 (16 brez UART0) delovnim blokom dolžine 1632byte (0x660), ki načeloma shranjujejo en okvir. Prvih 8byte je namenjeno opisu vsebine bloka, naslednjih 80byte je namenjeno NBP glavi okvirja (številka, naslovi, ločila) in ostanek koristnemu tovoru z rezervo na koncu za računanje CRCja. En blok lahko torej prenaša en cel IP okvir dolžine celo nekaj več od predpisanih 1500byte.



Hitrost 2Mbit/s na HDLC, 921600bit/s na SLIP in komaj 15 (16 brez

UART0) razpoložljivih delovnih blokov LPC2138 dopuščajo Ping dolžine preko 65500byte, kar je omejitev WinXP. Tako velik Ping sicer potrebuje kar 45 blokov, a počasnejša SLIP povezava ravno pravilno upočasnjuje pretok podatkov, da se skromni pomnilnik ATNC ne zabaše. Ozko grlo na pravem mestu torej v praksi pospešuje pretok podatkov, da se ne izgublja kje drugje!

Preostalih 8kbyte je razdeljeno med 4kbyte tabele dvojnikov NBP, 176byte vmesnika tipkovnice terminala na UART0, 1.5kbyte vmesnika zaslona terminala na UART0, 1kbyte za parametre, ki se prepišejo iz FLASH sektorja 25, razne notranje spremenljivke in vse tri sklade: glavni, IRQ in FIQ. ASCII terminal na UART0 ima torej povsem neodvisen pomnilnik, kar omogoča nadzor tudi v primeru, ko je NBP zasedel vse razpoložljive bloke. Zaslonski vmesnik je krožni dolžine 1.5kbyte, kar pomeni, da bo izpisovanje daljših sporočil na UART0 porezано, na primer ob gostem prometu in vklopljenem izpisu vseh okvirjev z ukazom "v".

Prevedeni program h91 je dolg 18.8kbyte in se izvaja izključno iz FLASH (500kbyte na razpolago v enem kosu). Isti program dela tudi na LPC2148. LPC2148 ima dodatnih 8kbyte pomnilnika RAM za USB, ki je žal nekoliko počasnejši in ga program, pisan za LPC2138, (še) ne zna izkoriščati.

Blok 1kbyte parametrov, ki se prepišejo iz sektorja 25 v RAM ob RESETu, je v sedanji inačici dodeljen na popolnoma enak način kot v EATNCju tako, da zadnjih 32byte (nastavitve Ethernet vmesnika) ni uporabljeno.

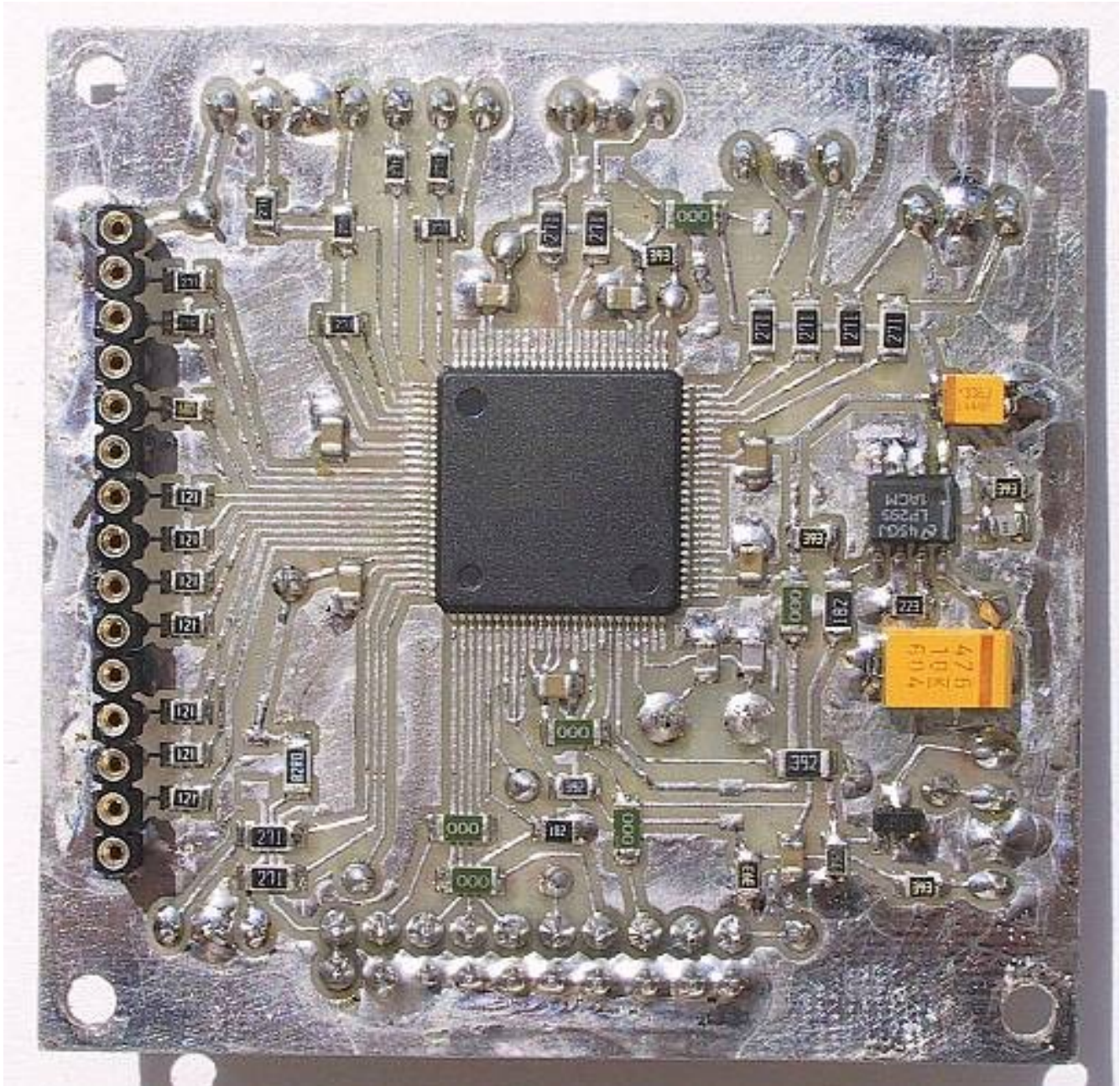
Časovno kritični deli programa so pisani tako, da upoštevajo pipeline dekodiranja ukazov v procesorskih jedrih ARM7. Časovno kritični deli so torej pisani linearno z uporabo makrojev ARM zbirnika, brez zank in s čimmanj pogojnimi skoki. Na LPC2138 to omogoča delovanje softverskega HDLC do nekje 3.5Mbit/s v simpleksu ali delovanje UART1 do 3.7Mbit/s v dupleksu. Navedene hitrosti veljajo za takt procesorja 59MHz z baud-rate kristalom 14.7456MHz.

Višje hitrosti lahko dosežemo edino z višjim taktom procesorja, kar bo zahtevalo prehod na jedro ARM9. Starejši LPC2138 sicer delujejo zanesljivo nekje do 80MHz, novejši LPC2138/01 in LPC2148 pridejo tudi do 100MHz. Pri taktu 59MHz in UART1 na 3.7Mbit/s je gornja meja za softverski HDLC okoli 2.6Mbit/s (FIQ), da procesorju ostane še dovolj zmogljivosti za obdelavo UART1 (IRQ).

Pomnilnik LPC2368 (EATNC) vključuje 32kbyte hitrega RAM neposredno na vodilu ARM procesorja, 16kbyte RAM na ločenem vodilu Ethernet vmesnika, 8kbyte RAM na ločenem vodilu USB vmesnika in 2kbyte battery-powered RAM s počasnim dostopom. Skupno torej 58kbyte.

Prevedeni program e45 je dolg 24.1kbyte in se izvaja izključno iz FLASH (500kbyte na razpolago v enem kosu). RAM se uporablja izključno za podatke. Pomnilnik čipa LPC2368 izberemo tako, da ob RESETu sklenemo RTCK na maso preko zaporednega upora 1k8 (je že vgrajen na tiskanem vezju).

Prvih 8kbyte neposrednega RAM na vodilu se uporablja za protokolske spremenljivke NBP (podobno kot ATNC na LPC2138): 4kbyte tabele dvojnikov NBP, 176byte vmesnika tipkovnice terminala na UART0, 1kbyte za parametre, ki se prepišejo iz FLASH sektorja 25, sedem krožnih vmesnikov za dodelitve blokov (7-krat 192byte), Ethernet-ARP cache s 25 vpisi po 16byte (skupno 400byte) in vse tri sklade: glavni, IRQ in FIQ.



Preostalih 24kbyte neposrednega RAM je namenjeno 15 delovnim blokom dolžine 1632byte (0x660), ki načeloma shranjujejo en okvir. Prvih 8byte je namenjeno opisu vsebine bloka, naslednjih 80byte je namenjeno NBP glavi okvirja (številka, naslovi, ločila) in ostanek koristnemu tovoru z rezervo na koncu za računanje CRCja. En blok lahko torej prenaša en cel IP okvir dolžine celo nekaj več od predpisanih 1500byte.

Vseh 8kbyte RAM na ločenem vodilu USB vmesnika je namenjeno dodatnim 5 delovnim blokom dolžine 1632byte (0x660). Skupno torej 20 delovnih blokov.

16kbyte RAM na Ethernet vodilu se uporablja za 10 Ethernet okvirjev dolžine 1.5kbyte (9 za sprejem in en sam za oddajo) in seznam kazalcev na Ethernet okvirje. Začetek Ethernet okvirjev je namenoma zamaknjen tako, da IP vsebina začenja na okroglem naslovu, kar omogoča učinkovito 32-bitno prepisovanje vsebine iz Ethernet sprejemnika v delovne bloke glavnega pomnilnika in obratno, iz delovnih blokov glavnega pomnilnika v Ethernet oddajnik.

Pri hitrosti 100Mbit/s je procesor prepočasen, da bi lahko sproti čital podatke iz Ethernet vmesnika. Takrat je število zaporedno sprejetih okvirjev omejeno na 8 s pomnilnikom na Ethernet vodilu, kar dopušča Ping največje dolžine 11832byte. Ta številka je vsekakor večja od običajnega TCP okna.

Pri hitrosti 10Mbit/s procesor zmore sproti čitati podatke in število zaporednih okvirjev omejuje glavni pomnilnik EATNCja, bolj točno število razpoložljivih blokov. 20 razpoložljivih delovnih blokov LPC2368 dopušča Ping največje dolžine 29592byte.

2kbyte battery-powered RAM s počasnim dostopom je v celoti namenjen krožnemu zaslonскому vmesniku ASCII terminala na UART0. ASCII terminal ima torej povsem neodvisen pomnilnik, kar omogoča nadzor tudi v primeru, ko je NBP zasedel vse razpoložljive bloke. Zaslonki vmesnik je krožni dolžine 2kbyte, kar pomeni, da bo izpisovanje daljših sporočil na UART0 porezano, na primer ob gostem prometu in vklopljenem izpisu vseh okvirjev z ukazom "V" oziroma ob zelo dolgem seznamu "G".

Pomnilnik LPC2387 (EATNC) vključuje 64kbyte hitrega RAM neposredno na vodilu ARM procesorja, 16kbyte RAM na ločenem vodilu Ethernet vmesnika, 16kbyte RAM na ločenem vodilu USB vmesnika in 2kbyte battery-powered RAM s počasnim dostopom. Skupno torej 98kbyte. Ker so vse ostale enote enake LPC2368 in je tudi razporeditev priključkov enaka, se program, pisan za LPC2368 lahko izvaja tudi na LPC2387 brez kakršnihkoli predelav.

Dodatnih 32kbyte RAM z neposrednim dostopom in dodatnih 8kbyte USB RAM omogočata dodatnih 25 delovnih blokov. Skupno število razpoložljivih delovnih blokov s čipom LPC2387 je torej 45. Dodatne bloke pomnilnika vključimo tako, da držimo RTCK ob RESETu nepovezan (visok). Pri Ethernet povezavi 10Mbit/s dopušča 45 razpoložljivih delovnih blokov LPC2387 Ping dolžine preko 65500byte, kar je omejitev WinXP.

Isti program za EATNC bi moral delati tudi na drugih predstavnikih družin LPC23xx in LPC24xx, saj imajo dovolj RAMa in iste vhodno/izhodne enote, večinoma celo na istih naslovih. Le število priključkov ohišja je večje in zato je njihova razporeditev drugačna.

Časovno kritični deli programa so enaki tistim iz ATNCja z LPC2138: napisani so tako, da upoštevajo pipeline dekodiranje ukazov v procesorskih jedrih ARM7. Časovno kritični deli so torej pisani linearno z uporabo makrojev ARM zbirnika, brez zank in s čimmanj pogojnimi skoki. Na LPC2368 oziroma LPC2387 se isti program žal izvaja približno 5% počasneje kot na LPC2138.

Izvajanje programov na LPC2368 oziroma LPC2387 dodatno upočasnjuje Ethernet vmesnik. Najverjetnejši vzrok je razmeroma počasna sinhronizacija med taktom procesorja in taktom 50MHz RMI Ethernet vmesnika. Sedanji EATNC omogoča delovanje softverskega HDLC do nekje 3.2Mbit/s v simpleksu ali delovanje UART1 do 3.7Mbit/s v dupleksu. Navedene hitrosti veljajo za takt procesorja 59MHz z baud-rate kristalom 14.7456MHz.

Višje hitrosti lahko dosežemo edino z višjim taktom procesorja, kar bo zahtevalo prehod na jedro ARM9. Razpoložljivi primerki LPC2387 sicer delujejo nekje do 100MHz. Pri taktu 59MHz in UART1 na 3.7Mbit/s je gornja meja za softverski HDLC okoli 2.2Mbit/s (FIQ), da procesorju ostane še dovolj zmogljivosti za obdelavo UART1 (IRQ). Prekinitvi FIQ in IRQ sicer upočasnjujeta primopredajo okvirjev z Ethernet vmesnikom.

Po drugi strani bi veljalo poskusiti radijsko zvezo preko Ethernet MAC RMI (Reduced Media-Independent Interface) oziroma MII (Media-Independent Interface) vmesnikov v družinah LPC23xx oziroma LPC24xx. Jasno moramo razviti lasten PHY v programirljivi logiki, da se prilagodimo radijskemu kanalu. Omejitev navzgor je tu 100Mbit/s. Verjetno bi se dalo tudi z nižjimi hitrostmi, saj MAC vmesnik v LPC23xx/24xx vsebuje le statično logiko in naj bi RMI takt smeli tudi zaustavljati.

Načrt programa: je osnovni namen tega članka, kaj mora početi naprava: ATNC oziroma EATNC, ki bo teorijo Ne-Brezhibnega Protokola uresničila v delujočem radijskem omrežju s številnimi uporabniki in vozlišči. Isti načrt lahko seveda s pridom uporabimo tudi z novejšimi, zmogljivejšimi gradniki v bodočem, še zmogljivejšem omrežju.

Najpomembnejši del načrta je oprema vozlišč na hribih, saj omrežja ne moremo preprosto zamenjati iz dneva v dan. NBP je sicer tako načrtovan, da dopušča veliko svobode končnemu uporabniku brez zahteve po predelavi ASVjev. Na primer, sedanja NBP-ARP avtomatika je le izpopolnjena inačica preverjene AX.25-ARP avtomatike, ki omogoča še veliko širitev v različne smeri.

Če bo kdaj prišlo do uporabe IP6, bo NBP-ARP avtomatiko smiselno prestaviti iz IP4 naslovov na Ethernet-MAC naslove, kar prinaša znatno poenostavitev delovanja EATNCja. NBP-ARP avtomatiko na IP4 naslovih so namreč narekovali stari protokoli SLIP (modem za windows) in AX.25, ki se z IP6 verjetno ne bojo nikoli uporabljali.

Končno, NBP ni v nobenem pogledu vezan na Internetni Protokol, pač

pa (E)ATNC omogoča preprosto komunikacijo s poljubnim protokolom, da se le prilagodimo razpoložljivim vmesnikom.

* * * * *