

# ***UN ECONOMICO TNC (TERMINAL NODE CONTROLLER) PER IL PACKET RADIO***

*Un economico microprocessore costituisce la base per la  
semplice realizzazione di questo TNC*

**Articolo originale di Bob Ball, WB8WGA**

**Tradotto da Claudio Fantino, IZ1DNJ, I-QRP # 604**

La maggior parte dei radioamatori che si sintonizzano sulla banda dei 2 metri avranno sicuramente ascoltato il *blurrp* dei famosi pacchetti che vengono trasmessi avanti ed indietro. A volte si tratta di un informale QSO, altre volte si tratta di beacon relativi alla posizione di radioamatori A.P.R.S. oppure di trasmissioni di qualcuno che informa della presenza in frequenza di una stazione DX interessante. Il Packet radio ha trovato un largo numero di applicazioni dalla sua introduzione negli anni 80.

Avete mai desiderato, in modo economico, poter decifrare i pacchetti trasmessi in locale? O magari installare il vostro digipeater per realizzare collegamenti con una particolare zona? Od eventualmente consentire che qualcuno si sintonizzi sul vostro Terminal Node Controller (TNC)?

In caso affermativo, questo semplice progetto potrebbe essere ciò che avete sempre desiderato. L'unità può essere facilmente realizzata con un costo sotto i 30 Euro. Dopo la sua realizzazione il software, che vi sarà installato, dovrà assicurare di:

- Controllare l'attività locale dei pacchetti;
- Poter realizzare un completo digipeater;
- Poter trasmettere pacchetti beacon, con il vostro testo desiderato, ad intervalli da voi predefiniti;
- Permettere comunicazioni in modo circolare utilizzando il modo "converse";

- Una volta collegato ad un ricevitore GPS, poter trasmettere la propria posizione in formato APRS NMEA.

Si noti che questo TNC non fornisce tutte le funzionalità come quelli disponibili in commercio. È una versione semplificata che consente le funzionalità essenziali per poter essere presenti in aria e di ottenere un sicuro divertimento con il packet radio. Se desiderate espandere la vostra conoscenza nella programmazione ed esplorare lo standard AX.25, l'unità descritta vi permetterà di modificare il software per supportare quasi tutte le applicazioni che potreste pensare d'utilizzare con il packet radio. Per esempio si potrebbe includere la funzione per il monitoraggio della temperatura, realizzare interfacce per il GPS o realizzare applicazioni di telecomando. La lista delle possibilità è senza limiti. Tutto il codice sorgente per l'unità è disponibile e ciò è un punto di partenza per implementare o modificare la programmazione senza dover ricorrere ad un costoso programmatore. Forniremo, successivamente in questo articolo, le informazioni supplementari sulla programmazione del microprocessore.

Ho pensato di incominciare questo progetto dopo aver esaminato le funzionalità di alcuni nuovi circuiti integrati microprocessori. Continuano a realizzarne sempre più potenti mentre il costo continua a scendere. Dopo la verifica di alcuni differenti tipi, sono stato attratto dalla serie di *MicroChip*, in particolare, da uno siglato PIC16F88. Questo microprocessore da 18 piedini disposti su due linee è venduto al costo di pochi Euro. Esso è provvisto del canale da analogico a digitale (A/D), include un UART per comunicare con il terminale,

dispone di una memoria Flash di programma e di una memoria EEPROM programmabile per memorizzare il nominativo e le opzioni. Può eseguire un'istruzione di programma ogni 200 nanosecondi. Meraviglia! non è certamente l'Intel 8080 che ho utilizzato all'inizio degli anni 70! Dopo avere esaminato le caratteristiche del microprocessore ho cominciato a domandarmi se potesse costituire la base di un semplice TNC.

Una delle cose che ho scoperto è che molti PIC impiegati nei TNC per il packet utilizzano il circuito integrato modem MX614P. I circuiti integrati del modem sono di grossa costruzione ed hanno un elevato consumo di energia e sono costosi. Inoltre, possono essere difficili da reperire. Nella maggior parte dei progetti degli esistenti TNC sono

stati rimpiazzati i componenti utilizzando i microprocessori che sono più potenti e meno costosi.

Questo articolo descriverà come utilizzare l'aumentata capacità interna di questi nuovi microprocessori per codificare e decodificare i segnali digitali. Mentre il progetto qui descritto è studiato per il packet radio, lo schema e la soluzione senza modem descritti dovrebbero permettere di utilizzare ogni tecnica digitale (RTTY, Pactor, ecc). Così, una volta che avete realizzato questo progetto per l'uso del packet, con un po' di lavoro ed ingegnosità, potete riprogrammarlo per accrescere le performance del packet radio o degli altri sistemi digitali radioamatoriali.

Come in molti progetti radioamatoriali, avrei potuto basarmi sul lavoro pubblicato da altri. In particolare, Mike Berg, NOBQH<sup>1</sup>, che ha

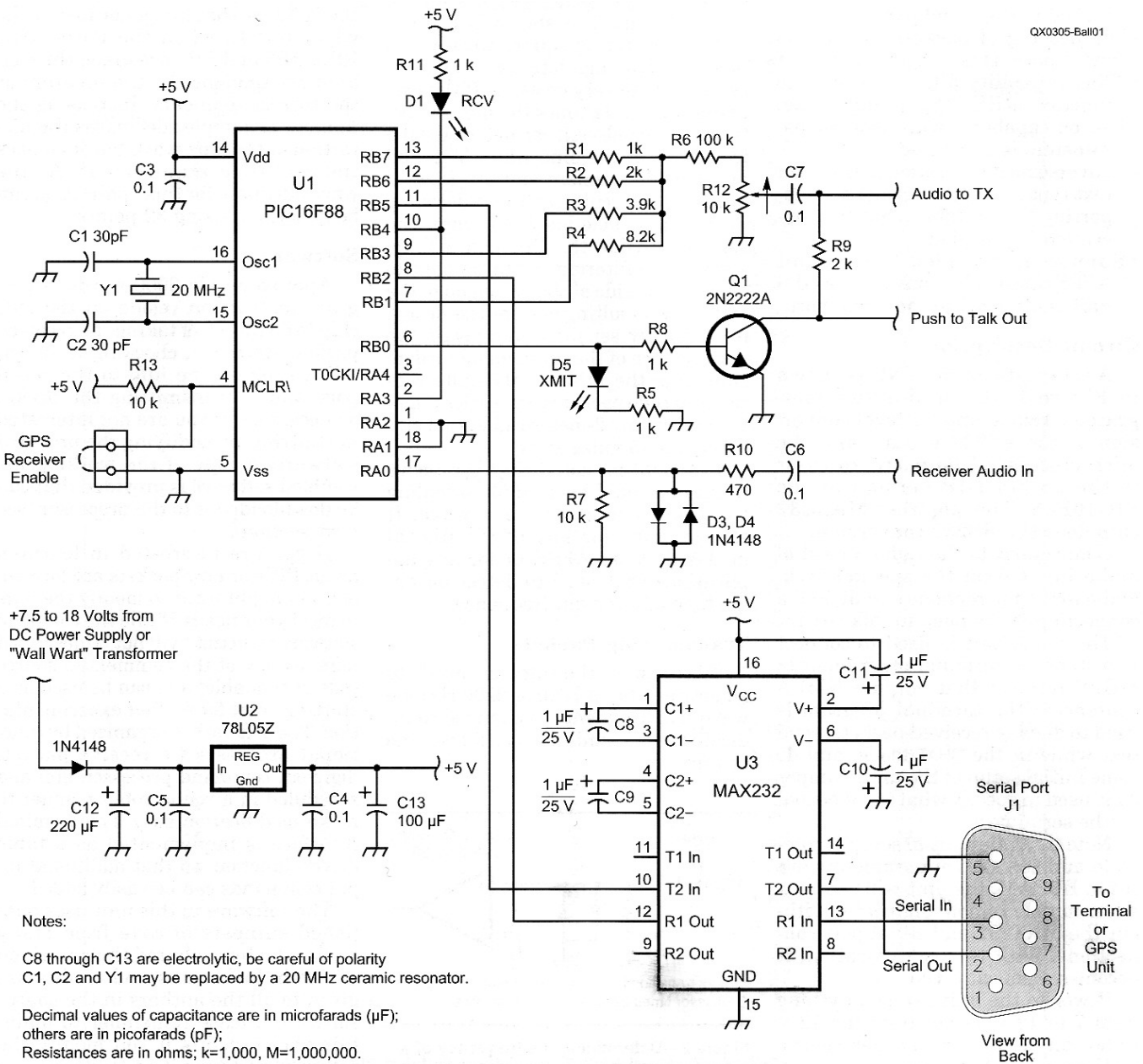


Figure 1—Schematic of the inexpensive TNC.

Figura 1 – Schema del TNC

realizzato il suo lavoro progettando un ricevitore senza modem utilizzando un comparatore esterno. Molti programmatori di PIC<sup>2</sup> hanno pubblicato i loro progetti con una rete di resistenze in cascata, per la vecchia tecnologia di processori, per generare una forma d'onda sinusoidale pulita necessaria per trasmettere in AFSK (Audio Frequency Shift Keying) i toni della trasmissione. Ho unito questo metodo di ricezione trasmissione, aggiungendo un comando d'interfaccia che si colloca dopo il TNC di TAPR<sup>3</sup> (Tucson Amateur Packet Radio) ed utilizzo la tecnologia più avanzata dei PIC per realizzare un TNC semplice con le seguenti caratteristiche:

- Provvisto di un'interfaccia dei comandi che permette il settaggio e la configurazione delle opzioni le quali saranno memorizzate nella ram EE e ripristinate ad ogni accensione del TNC.
- Provvisto di un controllo per verificare se il segnale è packet o no e se è packet se è indirizzato alla mia stazione sarà visualizzato sul terminale connesso alla porta seriale.
- Provvisto della funzione di ripetizione dei pacchetti (fino a 255 byte per pacchetto).
- La possibilità dell'unità di funzionare in modo RELAY. La possibilità di utilizzare la funzione di Beacon con l'impostazione dei parametri.
- La possibilità di operare in modo Converse per trasmettere il testo proveniente dalla porta seriale con caratteristiche di "tavola rotonda", modalità prevista per comunicazioni in modo circolare ed operare in modo chat.
- Provvisto della funzione per trasmettere in APRS il segnale con la propria posizione quando il GPS è connesso alla porta seriale.

#### DESCRIZIONE DEL CIRCUITO

Lo schema del TNC è rappresentato in Fig. 1. esso contiene due circuiti integrati, uno per la conversione del segnale fornito dalla porta RS232 ed il microprocessore. Il microprocessore è il PIC16F88 nella versione DIP a 18 piedini. Il noto circuito integrato MAX232S esegue la conversione della RS232.

La connessione con la radio consiste nel segnale d'ingresso proveniente dal jack dell'altoparlante, nel segnale di uscita collegato al circuito del microfono e la connessione con il PTT (Push To Talk).

La porta seriale è utilizzata per collegarsi un programma standard di terminale od un ricevitore GPS con uscita nello standard NMEA. Il programma di terminale è utilizzato per decodificare i pacchetti ricevuti, inviare il testo quando si utilizza il modo chat e per impostare la configurazione iniziale dell'unità. Il ponticello J4 è utilizzato per specificare cosa è connesso alla porta seriale.

Si noti che dopo il completamento della messa a punto iniziale e la memorizzazione, nella EERAM, dei parametri della stazione, l'unità potrà funzionare come digipeater senza utilizzare la connessione al MAX232 ed alla porta seriale. Con questa configurazione si ottiene un TNC con un solo circuito integrato, il microprocessore PIC.

L'alimentazione dell'unità può essere compresa tra i 7 ed i 18 V. Può essere utilizzata l'alimentazione dalla radio a 12 V, od un semplice alimentatore da presa.

#### RICEZIONE DEI PACKETS

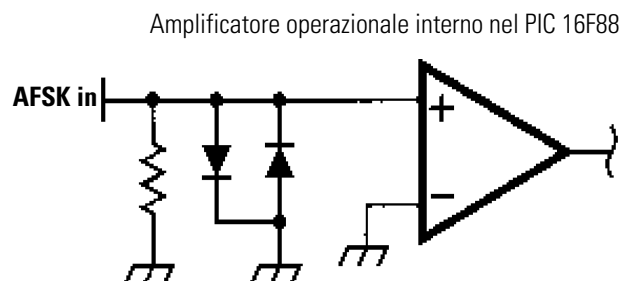
Il Packet radio, come molte altre tecniche di trasmissioni digitali, utilizza il sistema AFSK, *Audio Frequency Shift Keying*, per trasmettere il flusso degli uno e degli zero. Esso utilizza due frequenze, 1200 e 2200 hertz per rappresentare il cambiamento di valore tra uno e zero. Per far ciò, questo TNC usa un comparatore per trovare la coincidenza con il valore zero dell'onda sinusoidale del AFSK. Il TNC calcola l'intervallo fra i passaggi a zero per determinare la presenza della frequenza. Il software allora converte i cambiamenti dei toni in flusso di dati. Per immaginare ciò, si pensi ad un'onda sinusoidale a 1200 o 2200 hertz. Se spuntiamo (clippiamo) il picco della forma d'onda sinusoidale con una coppia di diodi (D1 e D2 nello schema) otteniamo una forma d'onda quadra. Il circuito è visualizzato in Fig. 2.

Se l'onda quadra risultante è rilevata da un detector ogni volta che è a zero, sarà generato un impulso. Se il microprocessore cronometra l'intervallo fra questi impulsi, può determinare la frequenza dell'onda sinusoidale. Il processore F88 ha i temporizzatori ed un circuito di comparazione interno in modo tale che tutto il lavoro di decodifica sia fatto all'interno del circuito integrato. Il software all'interno al PIC può anche svolgere una funzione di filtro digitale come un passa-banda e tagliare le frequenze audio non interessate. La scelta di non utilizzare il modem rende il sistema estremamente sensibile e rivaleggia con le prestazioni dei circuiti integrati commerciali con funzione di modem come il MX614. Le mie unità decodificano normalmente i pacchetti che durano meno di 1 secondo, ricevuti sulla mia radio dei 2 metri ed inviati da stazioni situate oltre i 150 Km..

Notate che la suddetta tecnica potrebbe essere usata per la campionatura della frequenza di una qualsiasi onda sinusoidale. Dovrebbe funzionare per ogni sistema digitale che utilizza l'AFSK o per tutte le applicazioni che devono rilevare un tono audio di una certa frequenza.

#### TRASMISSIONE DEI PACKETS

Per trasmettere, il software deve generare un'onda sinusoidale di 1200 o 2200 hertz. Le due frequenze sono utilizzate per la trasmissione in digitale del valore di *uno* o di *zero*.



Transizioni al passaggio a zero dell'onda sinusoidale.  
Tempi del processore fra queste transizioni

-----

**Figura 2 – Rilevamento della frequenza quando la sinusoide si trova al valore di zero.**

Un'onda sinusoidale è nient'altro che un livello di tensione che varia col tempo. Se il processore può cambiare il livello di tensione



abbastanza spesso (sull'ordine di 32 volte per ciclo), sarà generata un'onda di forma sinusoidale. La figura 3 mostra le tensioni che sono generate sul cursore di R12 quando le porte RB7, RB6, RB5 e RB1 sono impostate, dal programma, con una sequenza binaria, alla giusta frequenza. L'aumento del numero di campionamenti fa diminuire la distorsione dell'onda sinusoidale ma produce un aumento del tempo. Un'onda sinusoidale sufficiente per questa applicazione può essere generata utilizzando 32 punti.

## SOFTWARE

Nel circuito integrato del PIC risiedono circa 3.000 parole di istruzione di programma per assimilare il pacchetto ricevuto, immagazzinarlo, controllarlo per vedere se è valido, trasmetterlo l'informazione contenuta alla porta seriale, e, se occorre, ritrasmetterlo. Se non siete interessati a studiare od a modificare il software fornito, file "Modemless TNC V1.asm", è disponibile una copia del software pronto al funzionamento, file "Modemless TNC V1.HEX", a condizione che sia trasferito nel processore (vedi la sezione seguente).

Se siete interessati a conoscere i PIC o come sono formati i pacchetti, ecc, potreste considerare l'opportunità di modificare il codice del programma per soddisfare le vostre esigenze. Qui non sarà descritta la struttura completa del software, è disponibile<sup>4</sup> una copia del codice sorgente, commentato, e può essere usato come punto di partenza per le vostre ulteriori sperimentazioni. Il software è organizzato secondo i processi funzionali (ad esempio la ricezione dei pacchetti, la modalità digipeater, comando del processore, ecc) ed è programmato in un modo fluido ed intuitivo al fine di semplificare l'apporto di modifiche. L'interfaccia del terminale è effettuata secondo le funzioni guidate della tabella, in modo da potere facilmente inserire altre istruzioni.

Il software in questa unità usa i codici presenti negli articoli di molti radioamatori che hanno pubblicato il loro lavoro sul Web. I riferimenti ed i riconoscimenti sono attribuiti a tutti gli autori nel sorgente del software. Si prega di ricordarsi, se apportate modifiche al software e lo divulgate ad altri, di mantenere questi riferimenti e riconoscimenti.

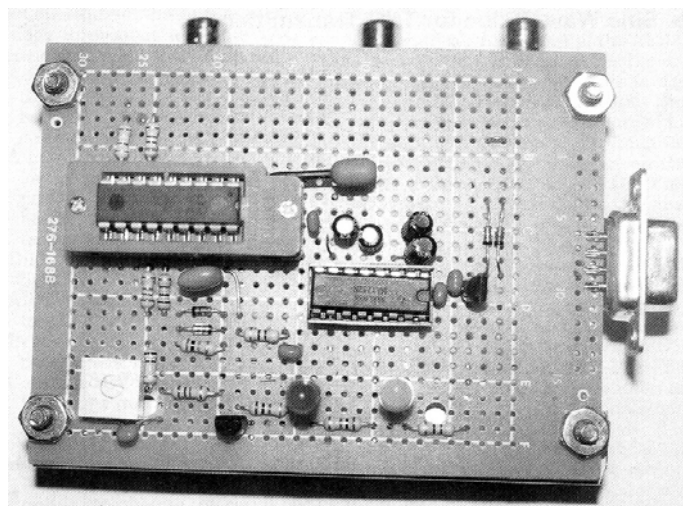
## COSTRUZIONE

La costruzione e la disposizione di questa unità non è critica. Una soluzione è quella di realizzare la costruzione su una basetta standard pre-forata. L'unità come rappresentata in figura 4<sup>5</sup> è stata sviluppata su una basetta standard pre-forata per prototipi ed ha due staffe ad angolo di alluminio come supporto. Come con tutti i circuiti integrati CMOS, prestate attenzione quando maneggiate ed inserite il circuito integrato per evitare danni dovuti all'elettricità statica. È buona norma utilizzare l'apposito cinturino da polso.

Questa unità funziona con una velocità di clock relativamente veloce ed è importante fare in modo da eliminare eventuali interferenze sull'alimentazione dei 5 V. Le capacità di 0,1 µF su ogni circuito integrato dovrebbero essere situate a destra del dispositivo con i collegamenti lunghi più o meno 2,5 cm.. Ho impiegato molto

tempo per mettere a punto il circuito digitale per poi scoprire che il problema era soltanto il rumore sulla linea dei 5 V. I condensatori di filtraggio sono poco costosi e piccoli, usateli liberamente.

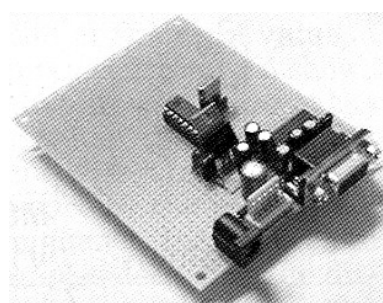
Se utilizzate per questo progetto una basetta pre-forata è buona norma eseguire una verifica prima di installare i circuiti integrati. Per prima cosa applicate i 12 V. e controllate la presenza dei +5V sul piedino 14 del PIC 16F88 e sul piedino 16 del MAX 232. Se tutto è a posto, installate i circuiti integrati utilizzando l'apposito cinturino da polso. Non dimenticatevi di programmare il PIC prima dell'installazione (vedi la sezione successiva).



**Figura 4 – piastra prototipo dell'unità**

Dopo lo sviluppo del prototipo come appare in figura 4, ho scoperto un'altro metodo di costruzione, la popolare basetta per prototipi di Olimex. Questo è il metodo più veloce per costruire il TNC.

Molti progetti necessitano del processore di base, il cristallo, il circuito di alimentazione e la connessione seriale. Olimex ha provveduto a realizzare ed a testare una piastra di montaggio con queste caratteristiche. Inoltre ha disponibile una zona per sviluppare il resto del circuito. In questo caso del TNC, metà dei componenti sono già disponibili sulla piastra.



**Figura 5 – piastra prototipo Olimex**

La piastra acquistata da Olimex contiene il circuito integrato del processore, tutti i componenti per il canale della seriale e tutti i componenti della parte dell'alimentazione, il tutto su una piastra dalle dimensioni di 100x80 millimetri. Con l'aggiunta di pochi componenti l'unità è completa. La connessione con l'interfaccia radio sarà realizzata secondo le proprie preferenze ed esigenze. Io utilizzo i connettori RCA per l'ingresso ed uscita audio e lo spinotto standard del microfono per la connessione del PTT (push-to talk). Questa soluzione per l'entrata ed uscita audio ed il PTT permette l'uso di cavi audio/video già pronti in commercio. Potete scegliere i connettori già pronti in commercio secondo le vostre esigenze.

Except as indicated, decimal values of capacitance are in microfarads ( $\mu\text{F}$ ); others are in picofarads (pF); resistances are in ohms; k = 1,000. n.c. = No connection

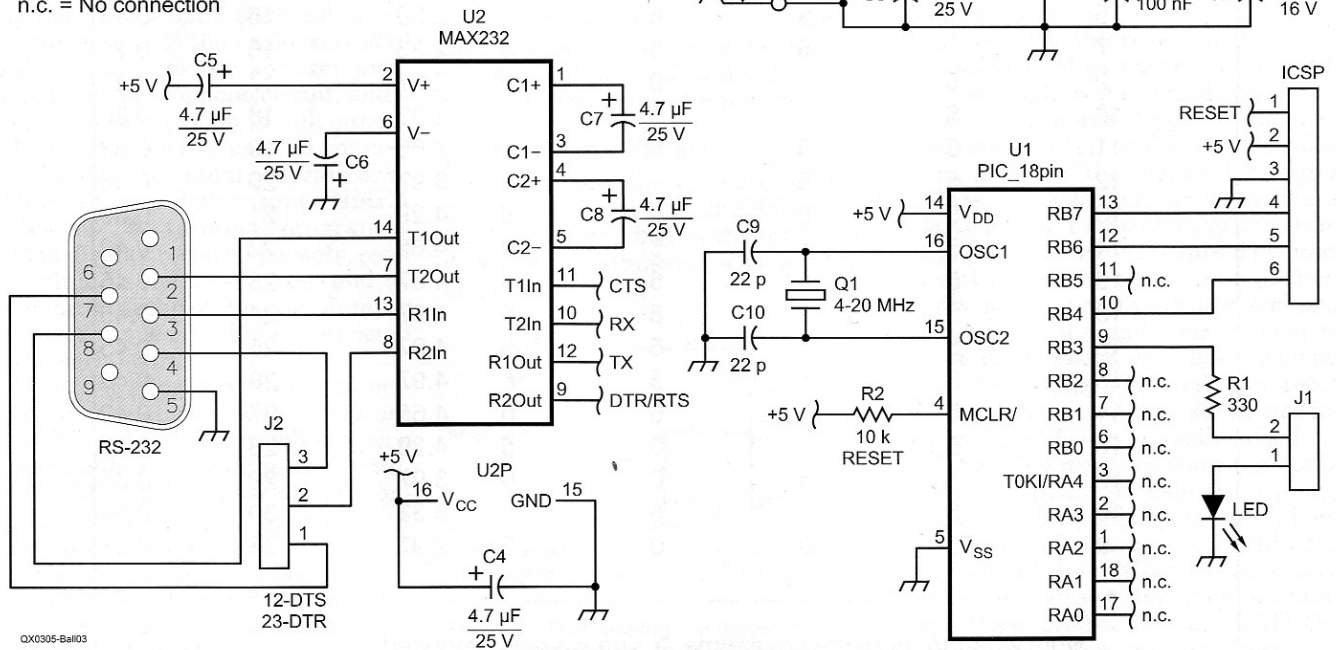


Figura 6 – Schema della piastra fornita da Olimex

La piastra pronta di Olimex non aumenta di molto il costo del progetto. La piastra prototipo realizzata per il PIC F88 è disponibile direttamente da Olimex a \$12.95 più spese di spedizione. Altre ditte<sup>6</sup> forniscono piastre dotate di PIC 16F88 per \$18.95 più spedizione. Un'immagine della piastra di Olimex, come esce dalla scatola d'imballaggio per la spedizione, è visibile in figura 5. Lo schema della piastra è rappresentato nella figura 6.

Un'immagine del TNC completo utilizzando la piastra Olimex è visibile nella figura 7. Notate il particolare del montaggio dentro un contenitore di plastica per la scuola comprato nel locale magazzino di WalMart per 71 centesimi. Rende la realizzazione piacevole e mantiene l'unità pulita ed asciutta.

### PROGRAMMAZIONE DELL'UNITÀ

#### CODICE ASSEMBLY

Il file esadecimale "Modemless TNC V1.HEX" per il TNC è fornito con il software e può essere utilizzato direttamente. Se non desideri riassemblare il software, salta questa sezione.

Se, tuttavia, avete deciso di cimentarvi nella sperimentazione con il codice assembly per questo il programma, è disponibile il sorgente "Modemless TNC V1.asm" che dovrà poi essere ri-assemblato. Ciò può essere fatto con l'assemblatore ed il simulatore resi disponibili gratuitamente da MicroChip sul suo sito Web. Prima di tutto scaricate ed installate sul vostro PC lo sviluppatore gratuito MP LAB disponibile sul sito Web di MicroChip. Utilizzando le previste istruzioni, assemblate il file fornito per il TNC. **Nella finestra delle opzioni del progetto di MPLAB, impostate a INHX8M il formato dell'uscita per il file esadecimale.**

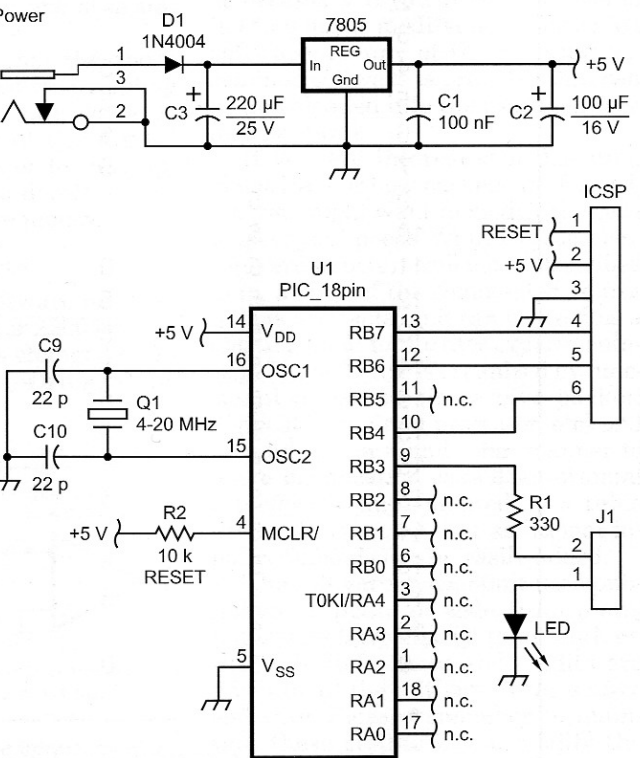


Figura 7 – Costruzione completa del TNC sulla piastra Olimex

#### PROGRAMMAZIONE DELLA MEMORIA FLASH

Ora che il vostro codice è assemblato, si possono scegliere un paio di opzioni per la programmazione del circuito integrato PIC 16F88. Se possedete un programmatore di PIC, od avete un amico che ne ha uno, un metodo è quello di trasferire nel microprocessore il disponibile file esadecimale "Modemless TNC V1.HEX" e siete a posto. Se avete deciso di eseguire questa procedura, la sezione seguente sulla programmazione con Boot Loader può essere saltata.

#### PROGRAMMAZIONE CON BOOT LOADER

Se intendete eseguire esperimenti con il software od installare aggiornamenti del software, il programma Boot Loader è una soluzione semplice per programmare l'unità. Una volta che è

installato, è molto più veloce e conveniente che utilizzare una unità esterna di programmazione. Con il Boot Loader, il circuito integrato può essere riprogrammato senza rimuoverlo dalla piastra del circuito usando lo stesso collegamento seriale che si utilizza per il normale funzionamento.

Per fare ciò il programma boot loader di *MicroChip* utilizza la caratteristica del PIC 16F88 che permette di scrivere i propri comandi di istruzione. Con questa caratteristica, un utente può fare, una volta per tutte, la programmazione nella memoria flash, con un semplice programma (chiamato "*Bloder-F88-20MHz.hex*"), quindi potete utilizzare il programma "*Screamer-v14.exe*" per trasferire il vostro software dal PC al PIC 16F88 con il collegamento seriale. Ciò significa che l'utente, per mezzo di un normale PC con un collegamento seriale al TNC, può caricare futuri aggiornamenti del software senza rimuovere il microprocessore dal TNC. Sul Web sono disponibili parecchi programmi, gratuiti, tipo boot loader. Segnalo il sito<sup>7</sup> Spark Fun Electronics, che fornisce una versione gratuita e che funziona bene con il TNC. Inoltre ho incluso una copia di boot loader per il PIC 16F88 (con l'autorizzazione) prelevata dal sito di Spark Fun Electronics con il codice sorgente.

1. Scarica ed installa sul PC la parte del programma chiamata "*Screamer-v14.exe*". Questa operazione è veloce.
2. Scarica dal sito Web il file esadecimale bootloader relativo alla versione del F88 (denominato "*Bloder-F88-20Mhz.hex*").
3. In un programmatore di PIC, vostro o di un vostro amico, posizionatevi il vostro PIC 16F88 e poi con l'apposito software, tipo "*ICPROG*" installatevi il programma Bloder in formato esadecimale del circuito integrato F88, "*Bloder-F88-20Mhz.hex*".  
**Quando mettete in funzione il programmatore per eseguire quanto descritto in questo paragrafo, assicuratevi di disabilitare, nella schermata di configurazione, il Master Clear, piedino A5 (piedino del MCLR). Ciò consente che la PORTA A5 sia usata come connessione per l'ingresso della funzione di GPS.**
4. Inserite il vostro PIC 16F88 nel TNC ed applicate l'alimentazione al TNC, I LED dovrebbero lampeggiare alternativamente per indicare che il programma boot loader è stato installato con successo.
5. Collegare il cavo seriale tra il PC ed il TNC. Avviate nel PC il programma "*Screamer-v14.exe*" per la programmazione del PIC. Selezionate il file esadecimale del TNC, "*Modemless TNC V1*" ed avviate l'esecuzione.
6. Quando l'unità indica *Waiting for Pic Broadcast*, dovete eseguire una serie di accensioni del TNC finché il programma si avvia (concedete 3 o 4 secondi affinché le capacità si scarichino). *Screamer* esegue la programmazione del circuito integrato in circa 10 secondi.
7. Chiudete il programma *Screamer* e riaprite sul PC il programma di terminal. L'interprete dei comandi dovrebbe essere disponibile ai vostri comandi. Utilizzando *Bloder* sono disponibili molte informazioni nella prevista sezione di files di aiuto relativa al codice bootloader,.

Ora la vostra unità è programmata. Il boot loader dovrebbe rimanere così finché dura il circuito integrato. Per riprogrammare l'unità in il futuro, ripetete i punti da cinque a sette.

#### PROGRAMMAZIONE DELLA MEMORIA EE

Una volta che il programma è caricato ed operativo, l'unità deve essere personalizzata per la stazione che l'utilizza. Informazioni sul nominativo della stazione, informazioni sul beacon, ecc. devono essere impostate e conservate. Come la maggior parte dei TNC, ciò è fatto da un terminale. La RAM EE del PIC è usata per questo e le informazioni sono ottenute per mezzo della porta seriale. Quindi viene eseguito un comando (PERM) per scrivere l'informazione nella RAM EE. Una volta che il comando *perm* è stato eseguito, i dati dovranno durare molti anni e dovranno resistere alla sostituzione del programma ed alle transizioni dell'alimentazione.

#### IMPOSTAZIONE DELLE OPZIONI

Le opzioni della stazione sono impostate per mezzo del terminal asci sulla porta seriale. Io utilizzo *Windows Hyperterminal*. Questo programma viene fornito nel software di *Windows*. Sicuramente altri emulatori di terminali funzioneranno altrettanto bene. Impostate il terminale a 9600 baud, su 8 bit, nessuna parità (8N1), Back Space (BS) per la cancellazione a ritroso del carattere ed eco. Se l'opzione dell'*eco* non è disponibile sul vostro programma terminal, il TNC dispone di un comando opzione per permettere l'eco locale.

Dopo la regolazione di queste opzioni, collegate un cavo seriale standard fra il vostro PC ed il TNC. Dopo l'accensione del TNC dovrebbe apparire il seguente messaggio.

#### **WB8WGA Modemless TNC V1.08 Type "Help" for Information cmd:**

Se il messaggio non compare, iniziate un accurato esame per la ricerca di errori sulle connessioni del circuito integrato del microprocessore e sui connettori della porta seriale.

Una volta che tutto funziona si può provare a ricevere alcuni pacchetti. Collegate un cavo audio tra il vostro altoparlante della radio dei 2 metri ed il TNC. A questo punto aprite lo squelch e regolate il livello audio al punto in cui il LED si illumina. Ora chiudete lo squelch ed il LED dovrebbe spegnersi. Se quanto descritto in questo paragrafo non funziona, verificate i collegamenti sul cavo audio del ricevitore ed i collegamenti (polarità) del LED

Una volta fatta questa operazione, dovrete vedere i pacchetti comparire sul vostro schermo.

Prima di incominciare a trasmettere con il nuovo TNC, è necessario impostare i vostri dati della stazione e salvarli nella EERAM. Il FCC interviene decisamente sui pacchetti che non sono identificati correttamente e li scarta. Mentre in modalità comando, il TNC accetterà sia i caratteri minuscoli che maiuscoli.

Prima di tutto occorrerà impostare il vostro nominativo utilizzando il comando *mycall* come nel seguente esempio:

```
cmd: mycall WB8WGA
OK
cmd:
```

Si noti che tutti i nominativi hanno un'identificazione secondaria da 0 a 15 e che di default è 0 (ad esempio WB8WGA-9, nel caso si utilizzi l'APRS).

L'altro parametro quello deve essere regolato è il nominativo di destinazione. Ciò è usato quando l'unità trasmette il beacon in modalità APRS o entra nel modo chat. Il formato per questo comando è *unproto callsign1 v callsign2*, dove *callsign1* è la destinazione finale ed il *callsign2* (opzionale) è il digipeater che dovrebbe essere usato per inviare il pacchetto alla destinazione. Possono essere utilizzati un massimo di 3 digipeater. Un esempio è questo:

```
cmd: unproto n0qbh-15 v wb8wga-10 v relay v wide
OK
cmd:
```

### TRASMISSIONE DEL PACKET

Ora è il momento di inviare qualche *blurrps*. Effettuate i collegamenti audio e PTT del trasmettitore. Una volta eseguiti i collegamenti alla vostra radio, è necessario regolare la deviazione del trasmettitore. Il sistema migliore è quello di eseguire un test e la calibrazione della deviazione.

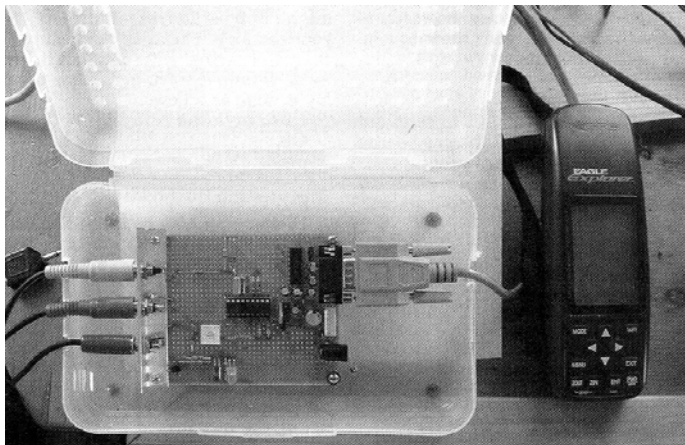


Figura 8 – Unità GPS collegata al TNC

Il comando *calibrate* è provvisto di un sistema d'aiuto per mezzo di due toni stabili da 1.200 o 2.200 Hz utilizzati per la taratura. Passate in trasmissione ed alternate i due toni ogni volta con la pressione della barra spaziatrice. L'operazione è la seguente:

```
cmd: CALIBRATE
Calibrate Mode. Space Bar to Toggle, Control C to Exit
<ctl-c>
OK
cmd:
```

Potete anche avere bisogno di impostare il numero di ritardi prima dell'inizio della trasmissione del pacchetto per permettere al vostro trasmettitore di stabilizzarsi quando passa in trasmissione. Ciò si ottiene utilizzando il comando *txdelay n*. Ogni incremento del numero *n* viene aggiunto un ritardo (approssimativamente di 6 ms). La sintassi del comando è:

```
cmd: TXDELAY 60
OK
cmd:
```

Il vostro TNC ora dovrebbe essere pronto per operare. Sintonizzate la radio sulla locale frequenza del packet.

### OPERAZIONE BEACON

Se volete che la vostra unità invii un beacon, occorre inserire il testo di beacon. Ciò si fa inserendo:

```
cmd: btext This is Bob in Jefferson, NH
OK
cmd:
```

Poi, può essere impostato l'intervallo del beacon con un valore compreso tra 1 e 59 minuti. Per esempio, una frequenza di beacon di mezz'ora risulterebbe così:

```
cmd: beacon every 30
OK
cmd:
```

Per disattivare il beacon occorre inserire:

```
cmd: beacon every 0
OK
cmd:
```

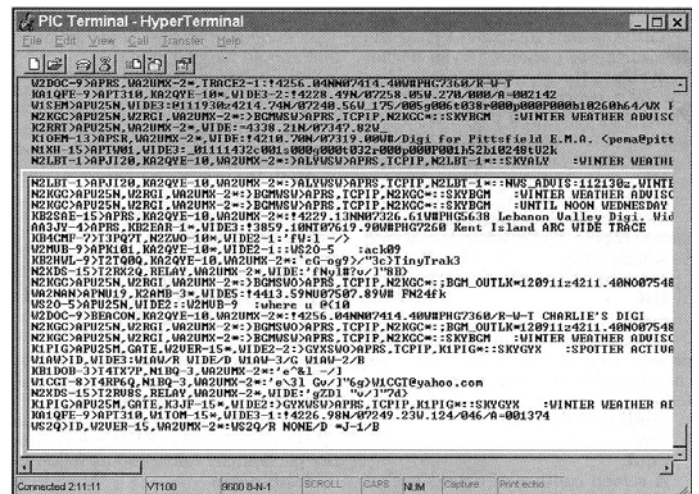


Figura 9 – videata dell'uscita del packet utilizzando Hyperterminal, con il comando monitor attivo

### OPERAZIONE MONITOR

Per il monitoraggio dei pacchetti sono disponibili tre opzioni. Se non volete utilizzare l'interfaccia seriale dell'unità (ad esempio nel caso di utilizzo in modo digipeater dedicato), il monitor può essere spento. Ciò può essere fatto con il seguente comando:

```
cmd: monitor off (vedi errata corregge di seguito)
OK
cmd:
```

Monitorare tutti i pacchetti, indipendentemente dal loro nominativo impostando:

```
cmd: monitor all (vedi errata corregge di seguito)
OK
cmd:
```

Il monitoraggio selettivo di pacchetti indirizzati a *mycall* o *alias* può essere compiuto impostando il monitor come segue:

```
cmd: monitor me (vedi errata corrige di seguito)
OK
cmd:
```

Nella figura 9 potete vedere la ricezione di un traffico reale, visualizzato dal programma *Hyperterminal* e prelevato dal PC con un'immagine a schermo. In questo caso l'opzione *monitor* è impostata *all*.

Se non lasciate collegato il PC al vostro TNC per il monitoraggio (ad esempio, nell'applicazione remota ed indipendente come il digipeater), è una buona prassi spegnere il monitor quando la configurazione della RAM EE è stata completata. Questo ridurrà il carico di lavoro ed il consumo del processore. La caratteristica di monitor è spenta automaticamente se l'unità è in modo di ricevente GPS (ponte J4 installato).

#### OPERAZIONE ALIAS

L'operazione *alias* è opzionale ed è utilizzata nel processo di ricezione. Permette che la vostra unità risponda ai pacchetti che potrebbero essere non indirizzati direttamente al vostro nominativo. L'*alias* può essere una stringa di sette caratteri, con SSID facoltativo. Ad esempio, impostare il mio *alias* al *relay* è fatto come segue:

```
cmd: myalias relay
OK
cmd:
```

#### DIGIPEATING

Il metodo *digipeating* può essere attivato o disattivato dal seguente comando:

```
cmd: digi on
OK
cmd:
```

oppure

```
cmd: digi off
OK
```

Se la funzione *digipeating* è attivata, tutti i pacchetti corrispondenti a *mycall* o *myalias* saranno *digipeater* (ripetuti) alla stazione seguente specificata nell'intestazione.

#### SALVATAGGIO DEI VOSTRI PARAMETRI

Dopo aver inserito i vostri parametri, potete visualizzarli per controllare che siano corretti. Ciò può essere fatto con il comando *disp* nel seguente modo:

```
cmd: disp
TXDALAY 60
ECHO on
GPS $GPRMC
TRace off
MONitor on
DIGIpeater on
```

```
BEACON on EVERY 20
UNPROTO aprs v relay v wide v wide
MYCALL wb8wga-0
MYALLIAS Bob in Jefferson, NH
cmd:
```

Non preoccupatevi, per ora, dei comandi GPS o TRACE, saranno descritti in una sezione successiva.

Se tutto è corretto, i parametri possono essere salvati nella memoria EE con il comando *perm* (permanente) come segue:

```
cmd: perm
OK
cmd:
```

#### FUNZIONE TRACE

Per la prova o l'esperimento del software, è previsto il comando *trace*. Fornisce un'esposizione esadecimale di un pacchetto che è formattato per essere trasmesso e non è ancora stato inviato. Inoltre fornisce un'esposizione esadecimale dei pacchetti ricevuti. La relativa sintassi è:

```
cmd: TR xmit
OK
cmd:
```

oppure

```
cmd: TR rcv
OK
cmd:
```

#### MODO CONVERSE

Il TNC permetterà la trasmissione di innumerevoli informazioni, stringhe e qualunque cosa che sia generato dalla porta seriale. Ciò permette di operare in modo *chat* in cui gli utenti possono conversare come se fossero ad una tavola rotonda. Questo modo è disponibile con il seguente comando:

```
cmd: converse
Entering Converse Mode, Hit Control C to Exit
Hello World
<ctl-C>
cmd:
```

Nel modo *converse* altre stazioni in frequenza che trasmettono delle stringhe avranno visualizzate anche le proprie stringhe, come in un dibattito in una tavola rotonda.

#### OPERAZIONI APRS

Automatic Position Reporting System (APRS) è un aspetto divertente ed utile del packet e dell'attività di radioamatore. Il sistema APRS fornisce dei validi mezzi per trasmettere le informazioni relative alla posizione, alle condizioni meteorologiche e messaggi. Questo software per il TNC supporta alcune delle funzioni comuni dell'APRS utilizzate oggi da molti radioamatori nella banda dei 2 metri (in Europa a 144,800 MHz) disponendo di molte funzionalità ad un prezzo

veramente economico. Sono disponibili diversi buoni libri<sup>8</sup> per invogliarvi ad utilizzare l'APRS.

Se state usando il vostro TNC configurato in APRS come stazione fissa presso il QTH, potrete ottenere, con il vostro nuovo TNC, l'inserimento nel sistema APRS, vedere i vostri rapporti in Internet e nello stesso tempo fornire un digipeater per i radioamatori locali. Per ottenere ciò, dovete semplicemente impostare il testo per *unproto* e per il *beacon*. Inoltre il *digi* ed il *myalias* devono essere abilitati (on) se volete svolgere la funzione di digipeater nei confronti degli altri radioamatori. Ad esempio, riporto la sequenza dei seguenti comandi che utilizzo sulla mia unità come digipeater dal mio QTH domestico:

**cmd: myalias RELAY**

**OK**

**Cmd: btext !4424.17N/07126.40W# /R Bob in Jafferson, NH**

**OK**

**Cmd: unproto APRS v RELAY v WIDE v WIDE**

**OK**

**Cmd: digi on**

**OK**

**cmd:**

Si ricorda che nella stringa su riportata dovete sostituire i valori con la longitudine e latitudine della vostra postazione fissa (Mine è situata a 44.24.17 N e 71.126.40 W). Il simbolo # alla fine rappresenta che voi svolgete la funzione di digipeater.

### **SENDING NATIONAL MARINE ELECTRONICS ASSOCIATION (NMEA) PROTOCOLLO PER IL GPS**

Un altro metodo per inviare i pacchetti APRS è utilizzando la vostra unità GPS per fornire informazioni relative alla posizione. La caratteristica di un TNC impostato come beacon è quella di poter trasmettere i rapporti di posizione da un'unità GPS utilizzando la connessione seriale. L'unità del GPS è collegata alla porta seriale del TNC anziché ad un PC e le stringhe nel formato NMEA-0183 sono trasmesse ogni intervallo del beacon anziché il normale testo di beacon (btext). La foto del TNC collegato al GPS è visibile nella Figura 8. Nota per attivare questa caratteristica. Per attivare la funzione la ricevente del GPS deve essere deve essere inizializzata impostando il software ad interpretare i segnali presenti sulla porta seriale come codice NMEA anziché le istruzioni del TNC. Spegnendo e riaccendendo il TNC si ritornerà al modo di funzionamento normale.

#### *IMPOSTAZIONE NMEA PER IL VOSTRO GPS*

Ci sono molti tipi di strutture di codice ASCII utilizzati dalle unità GPS. Il software del nostro TNC supporta i seguenti codici: \$GPGGA, \$GPGLL e \$GPRMC. Questi sono i più comuni ma controllate il vostro manuale del GPS per accertare se i vostri dati sono compatibili con uno dei protocolli del TNC. Quando il beacon va in fuori tempo massimo (timeout) nel TNC, attenderà e trasmetterà l'ultima sequenza valida in formato NMEA fornita dal GPS. Quindi sarà trasmessa la vostra posizione più recente. Questa opzione deve essere impostata dalla stringa di comando come segue:

**cmd: GPS \$GPGGA**

**OK**

**cmd:**

oppure

**cmd: GPS \$GPRMC**

**OK**

**cmd:**

oppure

**cmd: GPS \$GPGLL**

**OK**

**cmd:**

Inoltre ricordatevi che i parametri di *mycall*, *unproto*, e *beacon* devono essere impostati prima di trasmettere il beacon. Un esempio di *unproto* per l'APRS potrebbe essere così:

**cmd: unproto APRS v RELAY v WIDE v WIDE**

**OK**

**cmd:**

Confrontatevi con altri utilizzatori locali dell'APRS per trovare la vostra migliore impostazione.

Il valore del *beacon every n* deve essere regolato in modo adeguato al fine di determinare i vostri cambiamenti di posizione e nello stesso tempo non sovraccaricare la rete APRS con troppi rapporti.

Utilizzando il comando *perm* queste opzioni possono essere memorizzate in modo permanente. Dopo le impostazioni inserite il ponticello (jumper) del GPS e spegnete e poi riaccendete il l'unità TNC. Ora l'unità, una volta riaccesa, comincerà a trasmettere i vostri rapporti di posizione.

Notate che l'unità continuerà a ripetere i pacchetti quando nel modo GPS l'opzione *digi* è attivata.

### **POSSIBILITÀ FUTURE**

Spero che questo progetto stimoli il vostro interesse nelle operazioni di packet radio e nell'utilizzo del microprocessore. Una volta che l'unità hardware è stata messa a punto è abbastanza flessibile per essere programmata per fare molte cose utili nella vostra stazione radio. L'utilizzo della piastra per prototipi della Olimex permette di poter velocemente iniziare un nuovo progetto. La tecnica di ricezione e trasmissione senza modem può essere un buon punto di partenza per altri sistemi digitali. L'uso del concetto di boot loader fornisce molta flessibilità per i cambiamenti dei vostri progetti radioamatoriali. Dopo che avrete compreso ciò che il protocollo AX25 inserisce nei pacchetti del vostro TNC le possibilità sono limitate solo dalla vostra immaginazione. Spero di poter sentire presto il vostro *blurrrp* in aria.

### **NOTE**

<sup>1</sup> Mike Berg's, N0QBH, lavora sui decoder packet senza modem per l'AX25 e le sue realizzazioni possono essere viste sul suo sito Web: [www.ringolake.com](http://www.ringolake.com)

<sup>2</sup> Sul Web si trovano molti riferimenti che utilizzano una rete in cascata per generare un'onda sinusoidale. MicroChip ([www.microchip.com](http://www.microchip.com)) ha una nota applicativa e schemi, vedi AN655, che forniscono molte informazioni su questa tecnica. Byron Garrabrandt, N6BG, utilizza questa tecnica nella sua molto piccola, diffusa ed originale unità dal nome Tiny Trak, realizzata con il PIC

16F84. Lo schema può essere visto sul sito Web di Byro: [www.byron.com/tinytrak/tinytrak.zip](http://www.byron.com/tinytrak/tinytrak.zip)

<sup>3</sup> Molte eccellenti pubblicazioni sono disponibili sul sito Web del TAPR ([www.tapr.org](http://www.tapr.org)). In questo progetto è stato utilizzato il contributo della pubblicazione del codice sorgente così come le informazioni del manuale in linea del TNC2. Su questo sito si trova una buona documentazione per comprendere il protocollo AX25. Ho utilizzato le specifiche del protocollo AX25 ed una buona documentazione su come utilizzarlo è prodotta da John Hansen, W2FS. L'ordine le sequenze sono simili per le unità TNC1 e TNC2 di TAPR.

<sup>4</sup> I files esadecimale e del codice sorgente per questo progetto sono disponibili sul sito Web dell'associazione ([www.arrl.org/qexfiles](http://www.arrl.org/qexfiles)).

<sup>5</sup> Le foto di questo articolo sono state autorizzate da Imre Szauter.

<sup>6</sup> La piastra Olimex con installato il PIC 16F88 è disponibile presso Spark Fun Electronics al costo di \$ 18.95 ([www.sparkfun.com](http://www.sparkfun.com)). Le foto

della piastra, il software e lo schema sono state ricavate (con l'autorizzazione) dal sito Web.

<sup>7</sup> Boot loader per il PIC 16F88 è disponibile gratuitamente da: [www.sparkfun.com](http://www.sparkfun.com). Il software è costituito da 2 parti, la parte più piccola, chiamata Bloader che sarà programmata sul microprocessore e la parte chiamata Screamer che sarà installata sul PC. Le copie di Bloader e Screamer sono anche disponibili sul sito Web dell'associazione.

<sup>8</sup> APRS – "Moving Hams on Radio and the Internet" di Stan Horzepa, W1LOU, fornisce una buona descrizione della storia e funzionamento dell'APRS.

## ERRATA CORRIGE POBBLICATO SU QEX Luglio/Agosto 2005

Ho ricevuto molte considerazioni sull'articolo del TNC e le ripropongo soprattutto per accelerare i vostri sforzi nella costruzione.

1. Il testo alla pagina 8 dell'articolo descrive il comando monitor. Nell'esecuzione del software il comando reale è: **mon** e non **monitor**, quindi può essere utilizzato il comando **mon off** oppure **mon all** oppure **mon me**. Un grazie a K1IRK.
2. Se pensate di riassemblare il software ed utilizzate MPLAB 7.01 o successivi, vedrete 3 simboli sconosciuti dovuti ad un inaspettato cambiamento di nome da parte di *MicroChip*. Come precisato da Leo Coleman, ciò può essere facilmente rimediato eseguendo i seguenti cambiamenti nel file "Modemless TNC V1.asm":  
sostituite TOIE in TMROIE  
sostituite TOIF in TMROIF  
sostituite \_WRT\_ENABLE\_OFF in \_WRT\_PROTECT\_OFF
3. Phil, K1IRK e Rod Kreuter, WA3ENK, hanno avvertito una certa oscillazione ad alta frequenza nel circuito di ricezione che inducono il LED della ricezione a lampeggiare. Phil e Rod suggeriscono di rendere lo schema più affidabile al fine di non ricevere più pareri negativi su questo argomento. Essi suggeriscono quanto segue:
  - A. Scollegate da massa il piedino 1 del PIC 16F88 (lasciate a massa il piedino 18). Inserite una resistenza da 470 Ω tra il piedino 1 e massa.
  - B. Inserite una resistenza da 100 KΩ tra il piedino 1 e 2 del PIC 16F88. Ciò fornirà una leggera isteresi e stabilizzerà il circuito contro le oscillazioni.
  - C. Inoltre suggerisco di mettere dei condensatori da 1 nf (0,001 µf) tra il piedino 17 e massa per rimuovere un eventuale segnale di clock a 20 MHz proveniente dal comparatore.  
Non ho avvertito queste oscillazioni su ognuna delle cinque unità che ho costruito. Questa potrebbe essere una botta di fortuna! Penso che Phil e Rod abbiano avuto una buona idea ed abbiano in programma di modificare una delle mie unità per prova. Un grazie a Phil ed a Rod.
4. Inoltre Phil propone il seguente suggerimento sul circuito di trasmissione:  
Ho guardato l'uscita con un oscilloscopio utilizzando la funzione calibrazione ed è sembrata giusta. Ho pensato che potrebbe essere una buona idea mettere un condensatore attraverso il potenziometro per livellare i punti che con approssimazione compongono la forma d'onda sinusoidale, piuttosto che lasciarlo fare alla radio.
5. Rod Kreuter, WA3ENK, che ha esperienza con i TNC, ha inviato un certo numero di e-mail in merito ad un problema di configurazione del terminale.

Il TNC prevede un carattere di inizio nuova linea alla fine della sequenza di un comando da parte del terminale. Se vede solamente il ritorno carrello per il fine linea, ripete i singoli caratteri finché non interviene il fuori tempo di 7 secondi, poiché esso pensa che l'operatore sia andato a prendersi un caffè nel bel mezzo di un comando! Io utilizzo *Hyperterminal* con Win-XP, che ha l'opzione per l'impostazione dell'invio degli inizi delle nuove linee. Rod ha avvertito il problema con *Procomm* sotto DOS. Se avvertite questo problema e quando lavorate non potete regolare il vostro programma di terminale, per favore contattatemi e fatemelo sapere.

**Qui termina la mia traduzione del testo originale pubblicato su QEX Marzo/Aprile 2005 da WB8WGA e dell'errata corrige pubblicata su QEX Luglio/Agosto 2005**

**L'inglese lo conosco molto poco, ho utilizzato il traduttore automatico sul Web e poi conoscendo l'argomento ho apportato i necessari aggiustamenti.**

**Spero di non aver fatto troppi errori ed essere stato il più fedele possibile al testo originale.**

**Buona costruzione e buon divertimento de IZ1DNJ, Claudio - I-QRP # 604.**