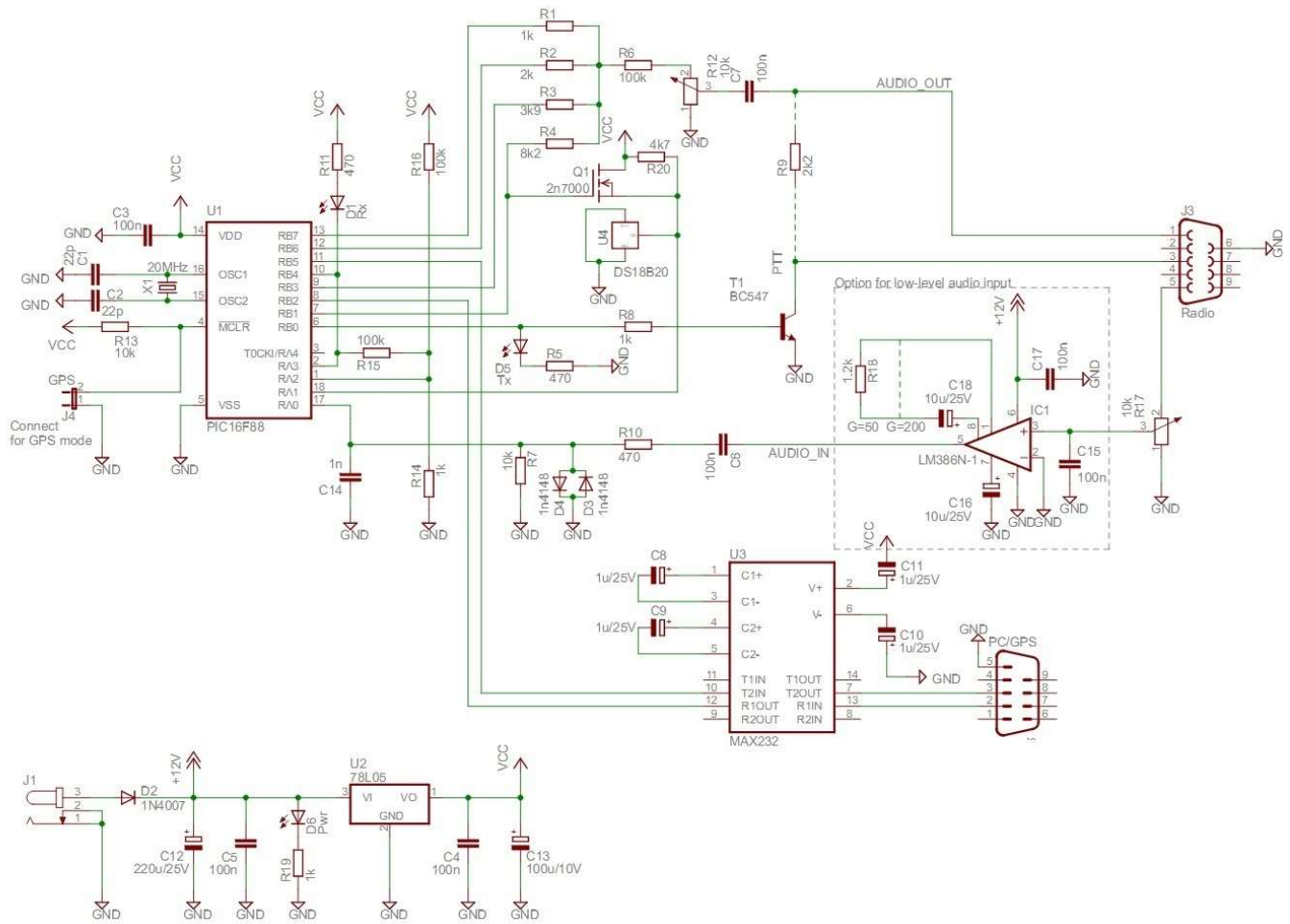


MINI TNC APRS (iw1gdm)

Un progetto di Bob Ball WB8WGA

Schema elettrico



Elenco componenti

R1 = 1K Ω	C1 = 22pF	D4 = 1N4148
R2 = 2K Ω	C2 = 22pF	D5 = diodo led
R3 = 3,9K Ω	C3 = 100nF	D6 = diodo led
R4 = 8,2K Ω	C4 = 100nF	U1 = PIC16F88
R5 = 470 Ω	C5 = 100nF	U2 = 7805
R6 = 100K Ω	C6 = 100nF	U3 = MAX232
R7 = 10K Ω	C7 = 100nF	U4 = DS18B20
R8 = 1K Ω	C8 = 1 μ F	IC1 = LM386
R9 = 2,2K Ω	C9 = 1 μ F	T1 = BC547
R10 = 470 Ω	C10 = 1 μ F	Q1 = 2N7000
R11 = 470 Ω	C11 = 1 μ F	X1 = CRYSTAL 20MHz
R12 = 10K Ω TRIMMER	C12 = 220 μ F	
R13 = 10K Ω	C13 = 100 μ F	
R14 = 1K Ω	C14 = 1nF	
R15 = 100K Ω	C15 = 100nF	
R16 = 100K Ω	C16 = 10 μ F 25V	
R17 = 10K Ω TRIMMER	C17 = 100nF	
R18 = 1,2K Ω	C18 = 10 μ F	
R19 = 1K Ω	D1 = diodo led	
R20 = 4,7K Ω	D2 = 1N4007	
	D3 = 1N4148	

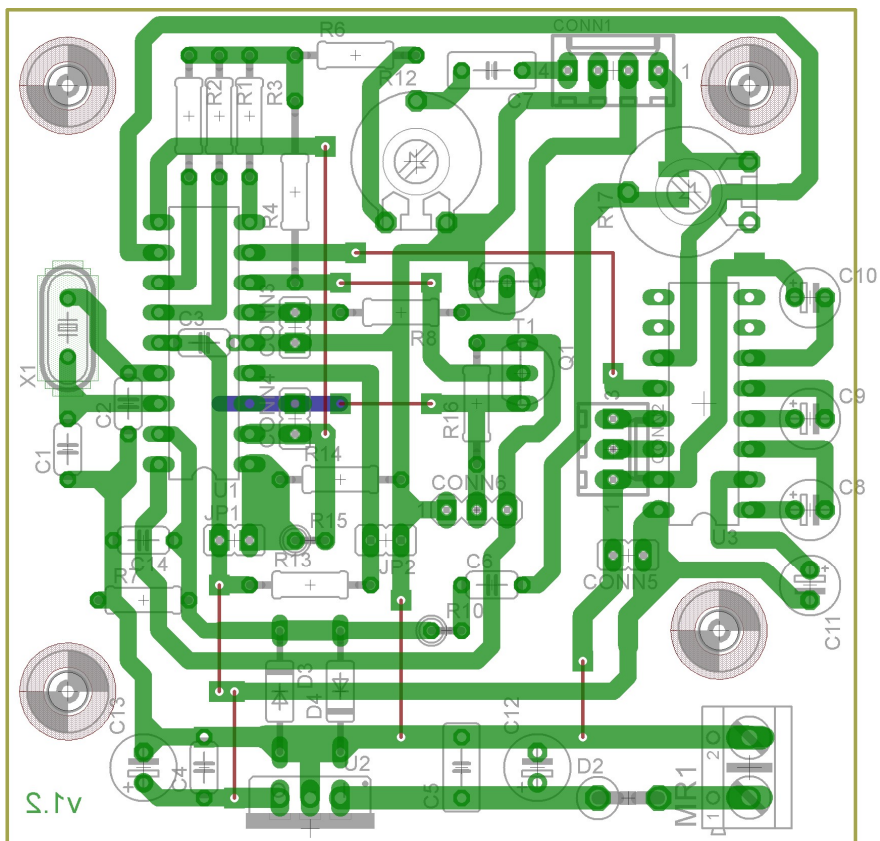
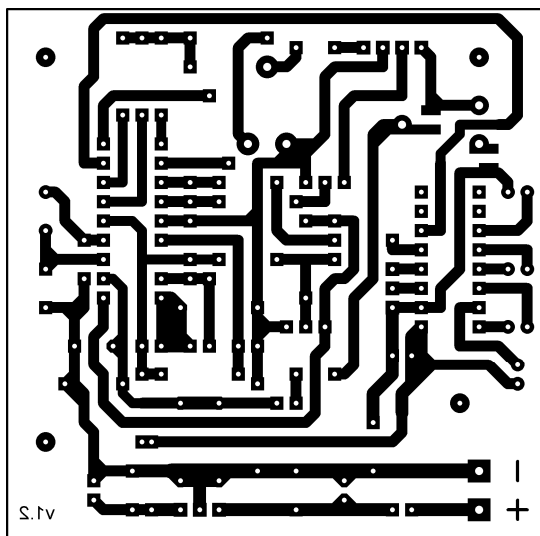


Immagine pcb

Vista lato componenti



Descrizione

Questo progetto, come si può leggere dal titolo, appartiene a **Bob Ball WB8WGA**, e lo si può ritrovare su internet in diversi siti. Il mio approccio è stato dal sito di [Claudio Fantino IZ1DNJ](#), e dal momento che anch'io ero alla ricerca di un semplice TNC, ho intrapreso tale realizzazione.

Oltre al sito di Claudio, ho comunque consultato altre fonti correlate, riscontrando che la sostanza era la stessa. Osservando altri schemi, ho modificato il pcb secondo le mie esigenze di spazio e soluzioni pratiche. Diversamente ci si può affidare ai vari progetti che si trovano in rete. Lo schema sopra, propone una parte opzionale per amplificare il segnale in ingresso, ma personalmente ho scartato questa modifica.

Questo dispositivo può operare come Tracker, come TNC e come semplice DIGIPEATER.

Veniamo al sodo. La costruzione è estremamente semplice. Basta armarsi di saldatore, pazienza, e tutti i componenti necessari. La parte che invece richiede un po' di attenzione, riguarda la programmazione del microcontrollore. Non essendo esperto al riguardo, ho provato andando alla cieca sperimentando tutto quanto, e devo dire che, avendo funzionato al primo colpo, la soddisfazione è stata notevole!

Non avendo un programmatore per i PIC, ne ho acquistato uno (visto i loro costi ormai molto contenuti). Ho reperito i programmi necessari, e ho iniziato a fare delle prove tra PIC e programmatore. Appena ho visto che ero in grado di programmare il PIC, ho dato il via alla realizzazione pratica del progetto!

Per capire se ero in grado o meno di gestire questa "impresa" ho iniziato a spulciare qua e là su internet, informazioni al merito. Così mi sono creato le basi per preparare il PIC micro nel seguente modo:

- Programmatore > K150
- Driver adeguati a Win 10 (.....)
- Programma PC > microbrn

Dal sito di Claudio, si evince che bisogna prima predisporre il micro di bootloader, e successivamente caricare il firmware desiderato. Questa configurazione è molto versatile, in quanto consente di caricare le versioni di firmware aggiornate, senza più rimuovere il chip dalla scheda; sarà sufficiente il collegamento al PC ed un programmino dedicato allo scopo. Il bootloader, infatti, ha lo scopo di rendere il PIC programmabile "in-circuit" (ovvero senza la necessità di scalzare il micro dalla scheda) facendoli riconoscere un modo "alternativo" per essere programmato.

Eseguendo tutta la procedura, sono riuscito a caricare il bootloader, e successivamente il firmware necessario. Volendo però programmare il micro con un firmware più aggiornato che ho trovato in questo [sito](#), sono incappato in un problema; la memoria non è sufficiente per mantenere il bootloader ed i firmware più aggiornati. Dal momento che però ho il programmatore esterno, ho raggiunto comunque l'obiettivo, programmandolo direttamente. Questo dettaglio, di non poco conto, mi ha spinto ad una seconda realizzazione del pcb che mi consente di montare uno zoccolo di tipo ZIF (a forza di inserzione zero), cosicché se si dovessero riscontrare dei bug irreversibili, o se si volesse aggiornare il PIC, la procedura di rimozione del chip dalla scheda, risulta semplice e veloce.

Configurazione

Sul sito <http://www.pianetaradio.it/progetti/minitnc.htm> trovate oltre ad alcuni firmware, anche i dettagli su come far funzionare il TNC, in tutte le sue caratteristiche. **Molto importante è scaricare il file "CMD" per UI-VIEW che lo trovate nella sezione "TNC APRS"**. In questo, dopo aver inserito questo file nella relativa cartella del programma UI-VIEW, potete inizializzare i comandi semplicemente andando a definire il tipo di TNC in "**Setup->Comm Setup**". Sul medesimo sito sono riportati anche i comandi che userete dal programma terminale per configurare il dispositivo. Se non siete pratici di programmi terminali, usate il terminal di UI-VIEW non appena siete riusciti a far vedere il tipo di TNC usando il file CMD.

Per il resto, non vi è nulla di complicato. Il mio consiglio, è quello di fare anche una ricerca sul web, editando "WB8WGA TNC". Potrete così analizzare il progetto da più punti di vista ed approfondire meglio i dettagli.

Messa a punto

La messa a punto è estremamente semplice. Dal lato PC se avete eseguito tutto correttamente, il dispositivo lo si potrà verificare subito da programma terminale.

Dal lato dell'RTX, occorre prestare un po' di attenzione ai collegamenti da fare in base al modello di apparato che utilizzate. Occorrerà soltanto una taratura dei livelli e verificare se l'apparato va in trasmissione quando deve farlo (i due LED vi aiuteranno a tale scopo).

Ho riscontrato, che il dispositivo esige il C14 di tipo multistrato! Diversamente risultava infatti essere "sordo", e non decodificava i pacchetti in modo adeguato. Non potendo analizzare il problema con strumentazione dedicata, ho eseguito una miriade di prove sostituendo i valori dei componenti sul ramo di ricezione. Solamente quando ho voluto provare i multistrato al posto di quelli poliestere, la situazione si è ribaltata.

Tuttavia il segnale in ingresso deve risultare un po' più alto del solito, e sul livello giusto tra l'altro. Nel mio caso, utilizzando il KENWOOD F7 il livello l'ho impostato tra il 50 e il 75% della corsa della manopolina. Il trimmer non sarebbe necessario, ma io ho preferito utilizzarlo in modo che carichi leggermente la sezione audio e lo si può escludere con ponticelli a saldatura dal lato rame. Con questo pcb, come potete osservare, si possono implementare diverse configurazioni del trimmer, agendo semplicemente con dei ponticelli di stagno.

Utilizzando la versione v2.31 del firmware, ho faticato a trovare la quadra per quanto riguarda la modalità KISS. Infatti, se abilitate il dispositivo con il comando "KISS ON" e successivamente memorizzate i parametri (comando "PERM"), non si riesce più ad uscire da tale modalità. Inizialmente, con svariate imprecazioni, sono arrivato a riprogrammare il chip. Non temete, non è necessario! Con UI-VIEW, andate nel menu **Setup -> Comm Setup** e nel menu a tendina "Host mode" abilitate la sezione KISS. Successivamente premete il pulsantino "Setup". Nella finestra che si apre, a sinistra nelle prime quattro righe, sono riportate le righe di comando che istruiscono i vari TNC più diffusi. Dal momento che il nostro dispositivo non è un vero e proprio TNC, andremo ad agire in tal modo, lasciando tutti i campi vuoti e cliccando "ok". Insomma è più facile a farsi che a dirsi. Questo succede perché quando abilitiamo la funzione KISS per esempio con il programma terminale, e poi memorizziamo i parametri, il nostro dispositivo non accetta più i comandi stessi dal terminale. Occorrerebbe inviare il comando di uscita in esadecimale, utilizzando uno specifico programma terminale in grado di farlo, ma dal momento che personalmente ne ho provati quattro o cinque prima di trovarne uno, c'è da dire che il tempo sulle lancette va avanti..

Fatta la regola, trovato l'inganno. Noi useremo UI-VIEW con la procedura sopra descritta, solamente per andare a disabilitare il KISS MODE. A questo punto possiamo nuovamente dare i nostri comandi mediante il programma terminale, o dal Terminal di UI-VIEW **dove diviene necessario far memorizzare la nuova configurazione (comando "PERM") per far sì che il KISS OFF risulti spento**. Se non date il comando "PERM", alla

successiva riaccensione del dispositivo, oppure con il comando "RESTART", lui si setterà con i parametri memorizzati precedentemente. Farete sicuramente un po' a cazzotti con queste procedure, poiché UI-VIEW va in conflitto quando non comunica con il TNC. Non temete l'utilizzo della combinazione "Ctrl Alt Canc -> Termina programma". Quando capirete la logica comportamentale, riuscirete a gestirvi in completa libertà. Un altro piccolo aspetto di non poco conto, è il sistema operativo che utilizzerete. Si sa ormai che Win10 è l'onnipotente. Continuo a non capire che utilità abbia... Ritornando a noi, potrebbe essere che fatterete un po' a trovare i driver per far funzionare la porta seriale ricavata da un convertitore USB->SERIALE. Tale porta, ormai scomparsa su quasi tutti i PC, è ancora molto utilizzata secondo me in giro per il mondo.. A tale proposito c'è anche da dire che osservando lo schema elettrico, noi faremo una doppia conversione dei livelli logici, vi pare corretto? Infatti il microcontrollore vuole i segnali TTL, che arrivano dalla COM RS232, che arrivano da una porta USB... Se siete bravi, potete azzardare la modifica con un convertitore da USB -> TTL 5V, eliminando la parte del MAX 232. Io ho provato il tutto con un cavo FTDI e funziona. Naturalmente bisognerà collegarsi ai pin del microcontrollore, utilizzati come TX ed RX togliendo l'integrato MAX232.

Come si presenta il mio TNC



Foto 1



Foto 2

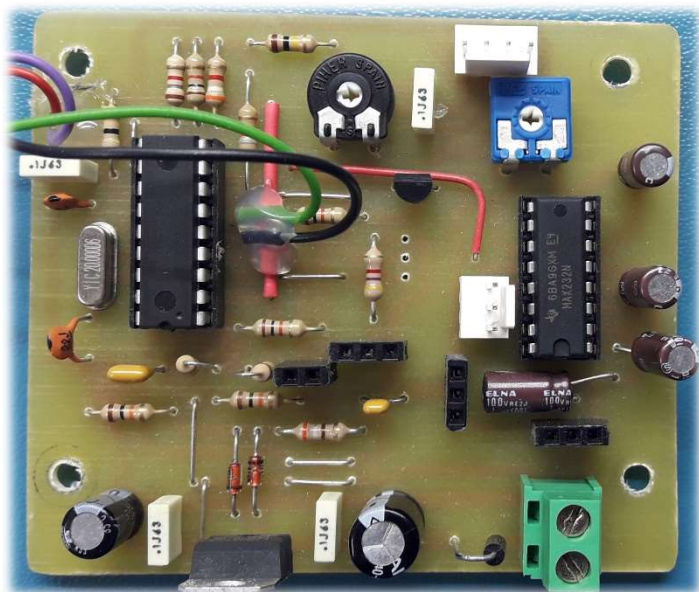


Foto 3

Nella **foto 1**, l'aspetto finale. Per la semplicità del progetto non vi è nessuna indicazione serigrafica.

Nella **foto 2**, vista dei connettori per il collegamento a PC e radio. Il connettore che va alla radio, è compatibile con quello dei Tiny Track. Da tale connettore viene fornita anche la tensione di alimentazione alla scheda proprio come sui dispositivi byonics.

Nella **foto 3** la scheda da me realizzata. Dal momento che era solo una prova, il montaggio risulta poco curato.