

Di telefonia mobile se ne parla molto, sotto tutti gli aspetti e su tutti i media: business fantastico, tecnologia fantascientifica, apparati sempre più miniaturizzati e sempre più efficienti... si parla poco però dei residuati che questa vorticoso corsa in avanti si lascia alle spalle.

Qui non ci interessa tanto indagare sulle ragioni di uno sviluppo tanto accelerato, quanto invece sapere che fine facciano i telefonini della generazione

sto del tutto è di molto inferiore al prezzo del solo ibrido amplificatore RF. Però, sventrarli per recuperare qualche pezzo sembra un vero peccato. Allora, perché non farne un onesto e robusto ricevitore o ricetrasmittente FM in 70 cm?

Quella che segue è la descrizione delle sevizie alle quali è stato sottoposto un apparato Italtel. Immagino comunque che quanto segue possa essere applicato ad altri modelli che utilizzino il medesimo PLL.

Parte prima

Ricetrasmittitore 10 W UHF FM

di Carlo Morelli

precedente, quelli che l'utente, per stare 'al passo coi tempi' smette per rivolgersi all'ultimo e nuovissimo prodotto proposto dalla pubblicità. E il motivo della domanda è ovviamente interessato...

Dai rivenditori di surplus si trovano le parti RF di questi dispositivi funzionanti in UHF banda 400 MHz. Non solo, i soliti fortunati se ne sono accapparrati perfino di nuovi e completi comprandoli, ad un prezzo simbolico, dalla SIP!

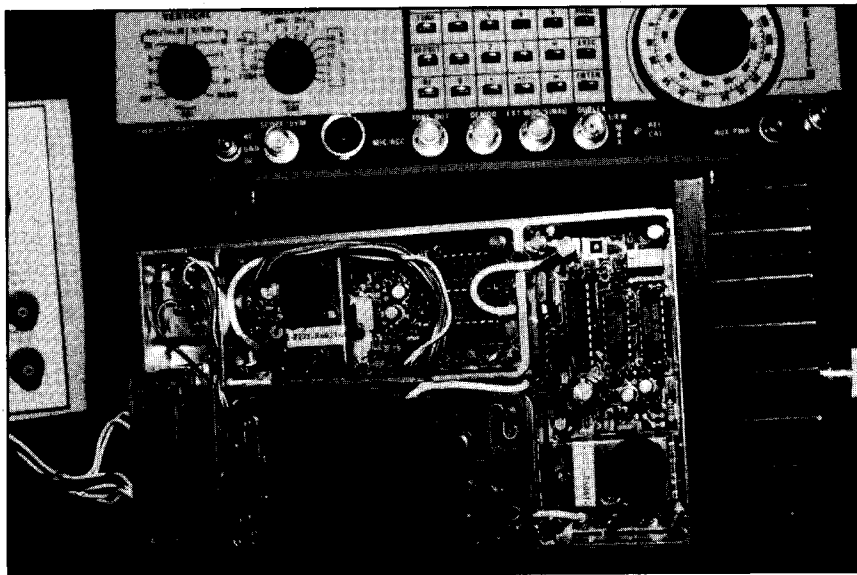
Per il dilettante, vederli e comprarli è quasi automatico: il co-

L'apparato non è proprio straordinario ma si vede che c'è la stoffa per cavarci qualcosa di utile: moduli racchiusi in vani dalla pesante schermatura, connettori coassiali miniatura dorati, filtri in cavità argentate, ecc.

Originariamente l'apparato funzionava in duplex (per ovvie ragioni), nel nostro caso, poiché lavoreremo in simplex o in semi-duplex, dovremo procurarci uno o due relais per commutare l'antenna (è ancora la soluzione più semplice) e per commutare la tensione alle sezioni RX e TX. Il relais d'antenna lo collegheremo al posto del duplexer. Questo è il filtro a linee coassiali (di pregevole fattura, anche qui l'argentatura abbonda) al quale era collegata da un lato l'antenna e dall'altro TX e RX; lo smonteremo salvando i cavetti già intestati da una parte che ci serviranno per collegare il relais.

Tolto il duplexer si può vedere l'amplificatore finale realizzato, come accennato, con un ibrido racchiuso nel suo schermo. Si vede anche il circuito di protezione per il ROS e la commutazione della potenza di uscita, e inoltre uno scatolotto che contiene otto celle con filtri elicoidali:

In primo piano il corpo radio dell'ex telefono cellulare. Si notano i vani aperti dei sintetizzatori TX e RX: sono visibili all'interno gli schermi dei VCO. Il vano chiuso è quello del ricevitore. Si nota, sulla destra, il dissipatore del finale RF e a sinistra il connettore DB25.



da aprire e ammirare: bellissimo! Di questi filtri, cinque sono posti in ricezione e tre stanno sulla linea del TX tra il pilota e l'ibrido finale RF per garantire una bella pulizia in trasmissione.

Il ricevitore, nato come detto per il funzionamento in duplex, e dovendo quindi 'convivere' con il segnale del trasmettitore, è realizzato in maniera inconsueta, almeno dal punto di vista dell'OM. Il front-end è tutto passivo e il primo mixer è a diodi, segue poi in modo consueto il filtro a quarzi a 21.4 MHz: si vede che i progettisti hanno preso davvero sul serio il pericolo dell'intermodulazione! La sensibilità di conseguenza non può essere sbalorditiva ma, poiché d'ora in poi funzioneremo in simplex, nessuno vieta di preamplificare (anzi è consigliato).

Dalla parte opposta dell'apparato si possono ammirare le chiusure dei tre vani modulari che contengono il sintetizzatore in ricezione, il sintetizzatore ed il pilota in trasmissione e la sezione ricevitore.

Al ricevitore arrivano, tramite due cavetti coassiali e relativi connettorini dorati, il segnale proveniente dal filtro RX e quello dell'oscillatore locale a PLL.

Come detto il mixer è a diodi e il prodotto del battimento è affidato al filtro a quarzi, segue l'amplificatore di media; il resto non è chiarissimo (specie quando non è disponibile lo schema...) tuttavia si indovina una seconda conversione a circa 62 kHz e un demodulatore a PLL col glorioso, e un po' datato, 4046.

I due sintetizzatori sono praticamente identici. Il quarzo di riferimento è però uno solo e si trova dal lato RX. Ogni PLL è realizzato attorno al noto MOTOROLA MC145156. Il nostro problema sarà quello di informare questo dispositivo circa la frequenza di funzionamento desiderata. Questo infatti è il nostro compito perché il resto è già tutto pronto.

Una volta, quando il nostro

apparato 'faceva il telefono', il microprocessore di gestione pensava a eseguire tutte le operazioni, a cercare il canale d'uso e quindi a trasmettere ai PLL le informazioni necessarie per posizionarli sulle frequenze giuste. Noi invece siamo in possesso solo della parte RF e comunque, quand'anche avessimo il controller a μP , non potremmo utilizzarlo per il nostro scopo. La tentazione di metterci comunque un altro micro, a esser sinceri, c'è stata ma qui si propone una soluzione più semplice.

Vediamo, nella fig. 1, lo schema a blocchi del PLL. I pin 1, 2 e 20 definiscono la canalizzazione, nel nostro caso 12.5 kHz.

Sul pin 12 entrano serialmente i dati relativi alla frequenza. I dati avanzano alla cadenza del clock applicato al pin 11 e, più precisamente, sul fronte in salita di questo. Quando i dati sono entrati, un impulso positivo al pin 13 fa sì che essi vengano 'congelati' nei latch. Da questo momento il PLL è 'informato' sulla frequenza che ci interessa.

Vediamo adesso lo schema elettrico del nostro circuito nella fig. 2. Per il momento supponiamo che tutto ciò che fa capo ai pin A5...A15 della 27512 non esista e che detti pin siano posti a massa. C'è un oscillatore, quarzato per comodità, la sua frequenza non è importante: va bene tutto intorno ai 3-4 MHz. Il segnale dell'oscillatore è applicato ad un 4020 che provvede

a fornire in uscita le successive divisioni. Due di queste servono a generare il clock e l'enable mentre altre cinque producono un conteggio utilizzato per scandire 32 indirizzi della EPROM.

Come è fatta una EPROM non dovrebbe essere più un segreto! Comunque diremo che è possibile immaginarla come una lunga striscia di carta a quadretti con otto quadretti per riga e tante righe quanti sono gli indirizzi. I quadretti incolonnati, come detto a otto a otto, sono visibili alle uscite O0...O7. man mano che facciamo avanzare le righe (gli indirizzi). Eseguire, come facciamo noi, un conteggio per 32 equivale a chiudere ad anello la striscia di carta (all'indirizzo 31 succede l'indirizzo zero and go on...). Ora, se nei quadretti della colonna che fa capo a O1 scriviamo la sequenza di zeri e uno, mostrata nella fig. 2 dello schema in basso a destra, abbiamo realizzato un semplice metodo di trasmissione seriale per il PLL del TX. E' intuitivo che un'altra colonna di quadretti della memoria, con idoneo contenuto, servirà per il PLL dell'RX. Possiamo pensare anche ad una o più colonne per i vari shift del trasmettitore: fate voi. L'importante è che una colonna, nel nostro caso la O7, contenga delle informazioni di 'servizio' per il funzionamento del tutto. I due PLL saranno informati contemporaneamente: riceveranno in comune il clock

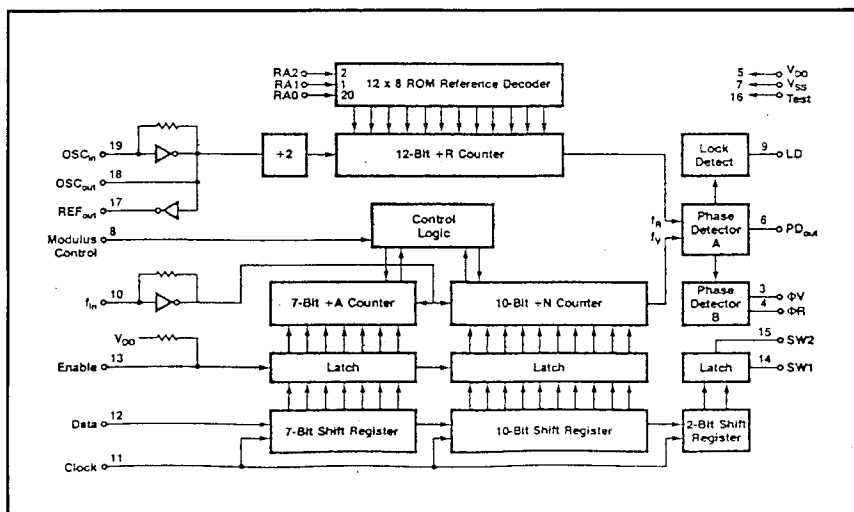
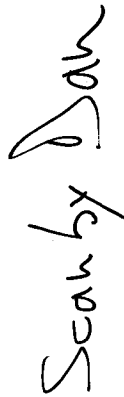


Fig. 1 - Schema a blocchi del PLL MC 145156

Fig. 2



e, al momento opportuno, l'enable. I dati invece li riceveranno, nello stesso tempo, ma ciascuno dalla propria colonna. Per il TX si potrà eventualmente scegliere, a mezzo commutatore, tra diverse colonne contenenti i dati per il funzionamento isofrequenza, shift negativo, shift positivo, ecc.

Come si vede, sempre in fig. 2, la sequenza da passare ai PLL è composta da 19 bit. I primi due, nel nostro caso, non sono usati e comunque sono ininfluenti ai fini della frequenza: possono essere indifferentemente a zero o a uno. Gli altri contengono l'informazione di frequenza, nei modi in cui si dirà più avanti. Volendo quindi compilare la memoria lo faremo colonna per colonna (come la schedina del totocalciol). Dopo 19 quadretti contenenti le informazioni ne seguiranno 13 inutili, questo per arrivare a 32. Il conteggio per 32 realizza una notevole semplificazione circuitale rispetto ad un conteggio per un numero che non sia una potenza di due.

La colonna di 'servizio' deve avere a zero i primi 19 quadretti e a uno gli altri 13. Questo fa sì che venga prodotto in uscita il clock in concomitanza dei dati e invece siano prodotti ben 13 enable quando i dati sono finiti. Certo un enable solo sarebbe stato sufficiente, gli altri 12 non servono a niente ma male non fanno. I pignoli comunque vedano la nota in calce.

Quindi, nell'ipotesi di programmare un solo canale isofrequenza (è l'ipotesi per cui si diceva di pensare a massa i pin A5...A15 della EPROM), si compileranno le due colonne relative alla frequenza del TX e a quella dell'RX più quella di servizio.

Come definire gli zeri e gli uno adeguatamente? La frequenza del PLL in kHz è definita dalla formula:

$$F = (40 * N + A) * 12.5$$

Per il ricevitore, F è quella dell'oscillatore locale (ottenuta dalla frequenza di ricezione sot-

traendole i 21400 kHz di IF). 40 è il fattore di divisione del prescaler: lo possiamo considerare un numero fisso come pure il 12.5 che rappresenta invece la canalizzazione del PLL. A e N sono i due numeri magici che ridotti in binario costituiscono lo scopo di tutto il discorso e rappresentano A: passi di canalizzazione singoli e N: blocchi da 40 passi.

Come trovare N e A lo si capisce meglio con un esempio: 433212.5 kHz è la frequenza che vogliamo assegnare al TX. $433212.5 / 12.5 = 34657$ è il numero di passi di canalizzazione a 12.5 kHz occorrenti per fare F. La parte intera di 34657/40 costituisce il numero di blocchi da 40 step cioè N. Vediamo: $34657/40 = 866.43$ quindi N = 866. Ed infine $34657 - 40 * 866 = 17$. A = 17 (* è il segno della moltiplicazione e / quello della divisione!).

Ridurre un numero decimale in binario è un'operazione eseguibile in vari modi, alcuni dei quali già descritti su queste pagine (alle quali eventualmente si rimanda...). Con una calcolatrice e conoscendo i numeri binari da 0 a 15 (esadecimali), il modo più rapido è illustrato nell'esempio seguente:

$866 / 256 = 3.3828125$
 $3.3828125 - 3 = 0.3828125 \rightarrow 3$ in binario = 0011
 $0.3828125 * 16 = 6.125$
 $6.125 - 6 = 0.125 \rightarrow 6$ 0110
 $0.125 * 16 = 2$
 $2 - 2 = 0 \rightarrow 2$ 0010

Il 256 della divisione è la massima potenza di 2 per la quale il numero è divisibile. Gli interi vengono sottratti e «messi da parte»: costituiscono l'oggetto della nostra ricerca. Il registro N, a 10 bit, sarà dunque caricato con 11,0110,0010, cadono di conseguenza i primi due zeri.

Per il registro A, si procede analogamente:

$17 / 16 = 1.0625 \rightarrow 1$ in binario = 0001
 $1.0625 - 1 = 0.0625$
 $0.0625 * 16 = 1$
 $1 - 1 = 0 \rightarrow 1$ 0001

Qui è 16 la massima potenza di due per la quale 17 è divisibile. La combinazione di bit è 001,0001; cade il primo zero poiché il registro è a sette bit.

Il messaggio completo conterrà i bit di N seguiti dai bit di A e preceduti da due bit inutilizzati che per esempio porremo a 0. Il risultato (finalmente!) è rappresentato dai 19 bit:

00,1101100010,0010001.

Le virgole hanno il solo scopo di migliorare la leggibilità.

Ora che il trasmettitore è programmato supponiamo di voler avere sull'RX la stessa frequenza. Come detto sopra, il valore di media frequenza del ricevitore è 21.4 MHz, quindi bisognerà programmare $433.2125 - 21.4 = 441.8125$ MHz.

Provate a fare il conto da soli; il risultato binario è 0011001101110011001.

Ancora uno sforzo e avremo la EPROM pronta. Intanto accendete il saldatore mentre facciamo la schedina...

Guardiamo la tabella: ads è la colonna degli indirizzi della

ads	7	6	5	4	3	2	1	0	hex
0	0	0	0	0	0	0	0	0	00
1	0	0	0	0	0	0	0	0	00
2	0	0	0	0	0	0	0	0	03
3	0	0	0	0	0	0	1	1	03
4	0	0	0	0	0	0	1	1	00
5	0	0	0	0	0	0	1	0	20
6	0	0	0	0	0	0	1	1	03
7	0	0	0	0	0	0	0	1	01
8	0	0	0	0	0	0	0	0	00
9	0	0	0	0	0	0	0	1	01
10	0	0	0	0	0	0	1	1	03
11	0	0	0	0	0	0	0	1	01
12	0	0	0	0	0	0	0	0	00
13	0	0	0	0	0	0	0	0	00
14	0	0	0	0	0	0	1	1	03
15	0	0	0	0	0	0	0	1	01
16	0	0	0	0	0	0	0	0	00
17	0	0	0	0	0	0	0	0	00
18	0	0	0	0	0	0	1	1	03
19	1	0	0	0	0	0	0	0	80
20	1	0	0	0	0	0	0	0	80
.	1	0	0	0	0	0	0	0	80
.	1	0	0	0	0	0	0	0	80
31	1	0	0	0	0	0	0	0	80

Vediamo prima l'hardware: lo schema di figura 1 mostra una possibile soluzione per le interconnessioni, e credo che non sia il caso di dilungarsi oltre il dire che il "circuito di controllo sintetizzatori" è quello proposto nella prima parte. Il "circuito di squelch", la cui funzione è ovvia, è proposto invece di seguito e la figura 2 mostra come è stato realizzato.

Il segnale di bassa frequenza in uscita al pin 11 del DB25 viene applicato ad un filtro passa

evitare fastidiosi "mitragliamenti" con segnali deboli.

L'uscita dell'integrato controlla il transistor NPN (va bene qualunque cosa) in funzione di interruttore: quando in conduzione, "cortocircuita" verso massa il segnale di BF. Ciò avviene in presenza di rumore (assenza di portante). Il condensatore tra collettore ed emitter, assieme alle resistenze alle quali fa capo, produce, volutamente, un effetto passa basso sul segnale utile; ciò avviene quando il transistor

Parte seconda

Ricetrasmittitore 10 W UHF FM

di Carlo Morelli

La scheda "millefori" sulla quale è realizzato il circuito di controllo dei sintetizzatori (le logiche in alto e la EPROM), il circuito di squelch (a sinistra della EPROM), i regolatori di tensione a 5 volt e il relais di commutazione della tensione di alimentazione RX/TX. A fianco della scheda, il selettore delle frequenze.

alto realizzato attorno all'induttanza da 47 mH. Le componenti alte in frequenza, relative al rumore (assenza di portante) vengono raddrizzate dal diodo e la tensione continua, ad esse proporzionale, così ottenuta è avviata ad una metà dell'integrato LM358 che provvede al confronto con il livello di tensione realizzato con il trimmer (manopola di comando dello squelch).

La resistenza, da 1 Mohm, tra uscita e ingresso non invertente, produce una certa isteresi, ad

è interdetto.

L'altra metà dell'integrato realizza il buffer di uscita il cui guadagno è variabile con la resistenza posta tra l'uscita e l'ingresso invertente. Un altoparlantino è pilotabile direttamente. Tuttavia un finale di bassa frequenza controllato da un potenziometro del volume è quasi d'obbligo. Nella figura 1 è rappresentato dal triangolo innominato, a significare che spetta a chi legge provvedere...

Come a carico del medesimo lettore resta il compito di racchiudere in un contenitore il ricetrasmittitore, ed i circuiti sin qui descritti, è importante che l'altoparlante sia posto all'esterno, oppure che la massima potenza sonora sia limitata, in quanto i sintetizzatori sono microfonicici e il tutto rischia di innescare aliegramente.

Il frontale del nostro contenitore poi supporterà: interruttore di accensione e spia eventuale, spie RX/TX, manopole per il volume e per lo squelch, presa per microfono, commutatore potenza min/max, commutatore iso/duplex/ecc, selettori decadici rotanti (thumbwheel) per unità dei MHz, centinaia di kHz e decine kHz.

Questi selettori, come si desu-

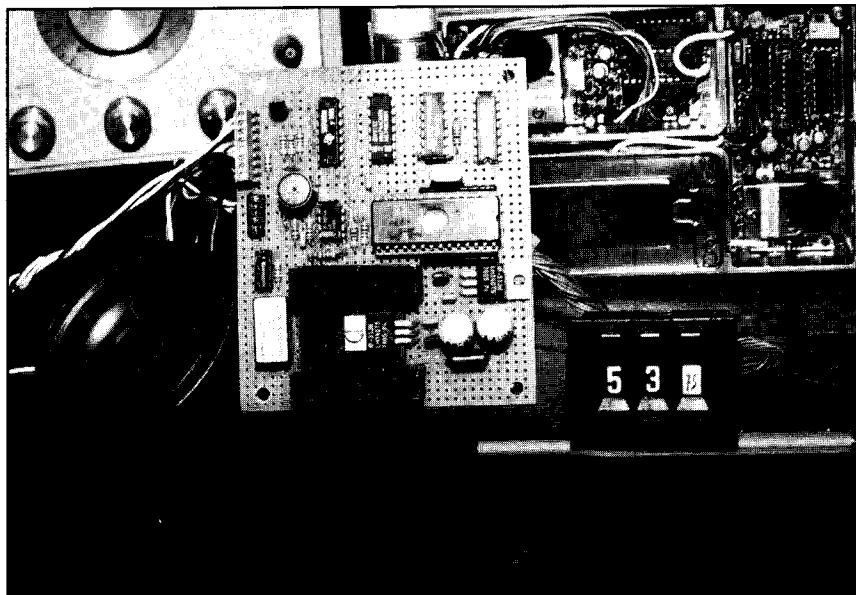
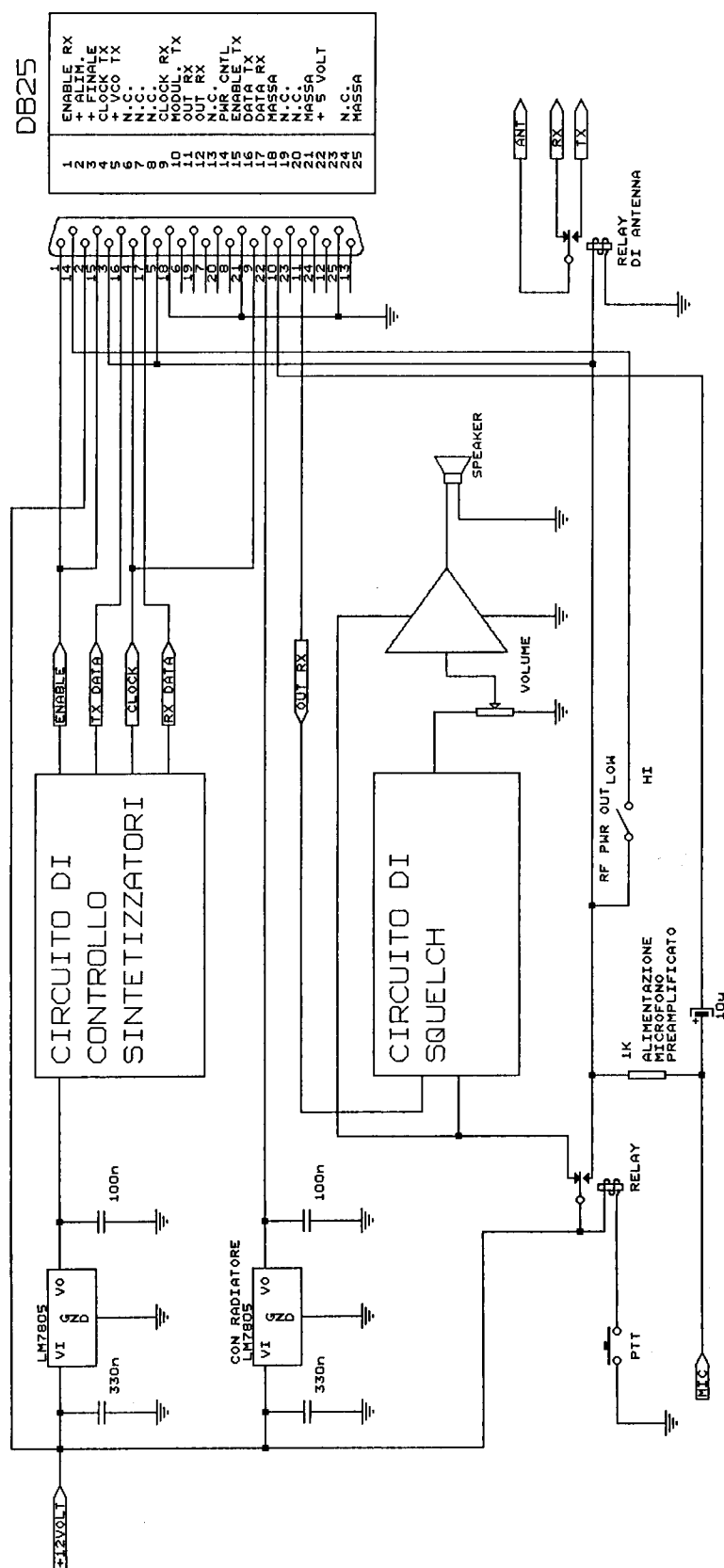


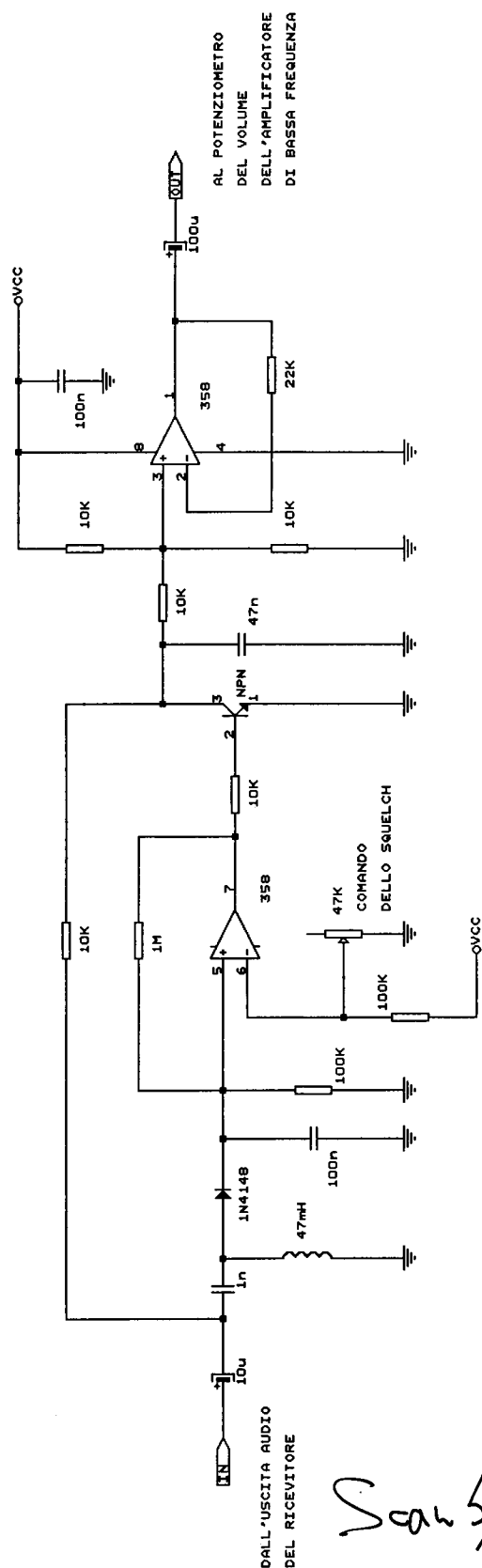
Fig. 1



me dallo schema presentato nella precedente puntata, forniscono all'uscita la codifica BCD del numero mostrato nella finestrella. L'ultimo, quello delle decine di kHz, andrà modificato con un po' di pazienza. Infatti di questo servono solo otto delle dieci posizioni che può assumere. Intanto le posizioni inutili possono essere escluse introducendo dei piolini di fermo negli appositi fori e poi, e qui viene il difficile, occorre sostituire, sul tamburo rotante, i numeri da 0 a 7 compresi, con la sequenza 00; 12.5; 25; 37.5; 50; 62.5; 75; 87.5. E non è propriamente uno scherzo, viste le ridotte dimensioni. Nel mio caso ho preparato una strisciolina di carta adeguatamente compilata ed incollata sul tamburo. I decimali però non sono riuscito a farceli stare...

Se siete arrivati sin qui, allora è il momento di programmare una EPROM con tutti gli ottocento canali che a passi di 12,5 kHz stanno tra 430 e 440 MHz.

Fig. 2



Il programmino in qbasic che segue serve appunto a realizzare un file binario su disco contenente le informazioni da caricare poi su EPROM utilizzando un normale EPROM writer.

I nomi delle variabili sono autoesplicanti così ho risparmiato in commenti, d'altra parte trattandosi di un programma "usa e getta" non è il caso di sprecarsi troppo...

Le variabili dichiarate all'inizio sono quelle dove più facilmente avrete da intervenire, per modifiche e personalizzazioni, e sono di immediata comprensione.

Se oltre agli shift proposti se ne volessero aggiungere altri vanno di conseguenza aggiunte le righe relative con l'indicazione della frequenza e della colonna associata. Come nell'esempio.

```
freq = zerofreq + (espressione)
GOSUB calcoli
col = (numero della colonna,
max 6)
GOSUB colonna
```

Il programma prima realizza un file vuoto (tutti i byte a FFhex) poi produce quello vero. L'operazione serve a far sì che le zone di memoria EPROM non utilizzate non contengano caratteri casuali prelevati da file presenti sul disco e che il DOS ha dichiarato cancellati!

Tutta l'operazione prende un discreto tempo e il contatore, che si vede scandire prima le locazioni e poi le frequenze, serve appunto a tranquillizzare circa il corretto e costante funzionamento del PC.

Digitato (senza errori!) e lanciato, il programma produrrà appunto un file di nome "eprom.bin" dalle dimensioni di 64 kbyte, che come detto andrà trasferito sulla 27512 che equipaggia il circuito presentato nella prima parte.

Ora i vostri thumbwheel indicano la frequenza di funzionamento dell'apparato.

Possiamo considerare finito il lavoro. Sarebbe, come già detto, a prevedere un preamplificatore da sistemarsi tra filtro RX

A lavoro ultimato, pur senza contenitore, siamo in trasmissione sulla frequenza di 435.375 MHz. La potenza erogata è di 15 watt.

e mixer. Tuttavia, anche lasciato così, se non pretendete proprio di farci i DX, penso possa dare delle belle soddisfazioni se non altro dal punto di vista economico...

Considerazioni finali.

La realizzazione di quanto esposto, pur implicando una certa pratica in costruzioni in V-UHF, è alla portata anche di dilettanti non particolarmente dotati di strumentazione. Se non si pretende la risposta piatta su tutta la banda ma ci si accontenta di una "fetta" più ristretta, il tutto può essere messo a punto con il watmetro e il collega disponibile in frequenza.

D'altra parte non va dimenticato che dei 10 MHz di cui si è parlato, solo una parte è utilizzabile dai titolari di licenza. E comunque, titolari o no, l'utilizzo di apparati radioelettrici al di fuori dei modi previsti dal Ministero comporta ben note sanzioni. (OldMan avvisato...)

Il programma

```
zerofreq = 430000 'frequenza
di partenza
shift1 = -1600 'shift negativo
shift2 = 1600 'shift positivo
nrch = 799 'numero canali
filename$ = "eprom.bin" 'nome
del file contenente i dati
```



```
DIM chmatrix(8, 32)
CLS
blank% = &HFF
OPEN filename$ FOR BINARY
AS #1 LEN = 1
FOR x = 1 TO 65536
PUT #1, x, blank%
LOCATE 20, 50: PRINT "blank
="; x
NEXT
CLOSE #1
```

```
OPEN filename$ FOR BINARY
AS #1 LEN = 1
'colonna di servizio
chmatrix(7, 0) = 0
chmatrix(7, 1) = 0
chmatrix(7, 2) = 0
chmatrix(7, 3) = 0
chmatrix(7, 4) = 0
chmatrix(7, 5) = 0
chmatrix(7, 6) = 0
chmatrix(7, 7) = 0
```

```
chmatrix(7, 8) = 0
chmatrix(7, 9) = 0
chmatrix(7, 10) = 0
chmatrix(7, 11) = 0
chmatrix(7, 12) = 0
chmatrix(7, 13) = 0
chmatrix(7, 14) = 0
chmatrix(7, 15) = 0
chmatrix(7, 16) = 0
chmatrix(7, 17) = 0
chmatrix(7, 18) = 0
chmatrix(7, 19) = 1
chmatrix(7, 20) = 1 'anche 0
chmatrix(7, 21) = 1 '0
chmatrix(7, 24) = 1 '0
chmatrix(7, 25) = 1 '0
chmatrix(7, 26) = 1 '0
chmatrix(7, 27) = 1 '0
chmatrix(7, 28) = 1 '0
chmatrix(7, 29) = 1 '0
chmatrix(7, 30) = 1 '0
chmatrix(7, 31) = 1 '0
```

```

FOR channel% = 0 TO nrch
' ricezione -----
freq = zerofreq - 21400 + chan-
nel% * 12.5
GOSUB calcoli
col = 0
GOSUB colonna
' trasmissione iso -----
freq = zerofreq + channel% *
12.5
GOSUB calcoli
col = 1
GOSUB colonna
' trasmission shift 1 -----
freq = zerofreq + shift1 + chan-
nel% * 12.5
GOSUB calcoli
col = 2
GOSUB colonna
' trasmission shift 2 -----
freq = zerofreq + shift2 + chan-
nel% * 12.5
GOSUB calcoli
col = 3
GOSUB colonna

```

```

FOR rowmatrix% = 0 TO 31
byteval% = 0
byteval% = byteval% + chma-
trix(7, rowmatrix%) * 128
byteval% = byteval% + chma-
trix(6, rowmatrix%) * 64
byteval% = byteval% + chma-
trix(5, rowmatrix%) * 32
byteval% = byteval% + chma-
trix(4, rowmatrix%) * 16
byteval% = byteval% + chma-
trix(3, rowmatrix%) * 8
byteval% = byteval% + chma-
trix(2, rowmatrix%) * 4
byteval% = byteval% + chma-
trix(1, rowmatrix%) * 2
byteval% = byteval% + chma-
trix(0, rowmatrix%) * 1

```

```

PUT #1, rowmatrix% + 1 + off-
set, byteval%
NEXT
canale = 430000 + mega *
1000 + chilo% * 100 + choffset%
* 12.5
LOCATE 20, 50: PRINT "fre-
quenza =", canale, " "

choffset% = choffset% + 1
IF choffset% = 8 THEN
choffset% = 0
chilo% = chilo% + 1
IF chilo% = 10 THEN
chilo% = 0
mega = mega + 1
END IF
END IF

offset = choffset% * 32 + chilo%
* 256 + mega * 4096
NEXT
CLOSE #1
END

```

***** routines *****

```

calcoli:
passi = (freq / 12.5)
c = passi / 40
N = INT(c)
A = passi - N * 40
c = N / 256
byte1 = INT(c)
c = (c - byte1) * 16
byte2 = INT(c)
c = (c - byte2) * 16
byte3 = INT(c)
c = A / 16
byte4 = INT(c)
c = (c - byte4) * 16
byte5 = INT(c)
RETURN

```

colonna:

```

riga = 0
byte% = byte1
GOSUB bcdtobin
byte% = byte2
GOSUB bcdtobin
byte% = byte3
GOSUB bcdtobin
byte% = byte4
GOSUB bcdtobin2
byte% = byte5
GOSUB bcdtobin
RETURN

```

```

bcdtobin:
IF (byte% AND 8) = 8 THEN
chmatrix(col, riga) = 1
ELSE
chmatrix(col, riga) = 0
END IF
riga = riga + 1
bcdtobin2:
IF (byte% AND 4) = 4 THEN
chmatrix(col, riga) = 1
ELSE
chmatrix(col, riga) = 0
END IF
riga = riga + 1
IF (byte% AND 1) = 1 THEN
chmatrix(col, riga) = 1
ELSE
chmatrix(col, riga) = 0
END IF
riga = riga + 1
RETURN

```

D

