

En L-bånds FM-ATV transceiver 3. del

Af OZ7TA Jørgen Kragh, Forelvej 25, 3450 Allerød

23. Afslutning

I denne artikel er vist et eksempel på en simpel, men alligevel ydedygtig transceiver til FM-ATV i L-båndet. Transceiveren er bevidst holdt på et meget enkelt plan, således at det ikke blev en total uoverskuelig opgave at realisere den. Der er formentlig mange punkter, hvor transceiverens performance kan forbedres, men som en start på FM-ATV er den udmærket. Desuden kan de viste kredsløb jo bruges som inspiration for andre.

Modtagerens følsomhed lader selvfølgelig noget tilbage at ønske, men kombineret med en god masteforstærker med en eller to GaAs-FET's (og hvem har ikke en sådan, når man kører 1300 MHz?) kan modtagerens samlede støjtal komme ned omkring 1 dB, og det er tilstrækkeligt.

Masteforstærkere har været beskrevet i litteraturen et utal af gange, så her er der rigeligt at vælge imellem, både af gode og mindre gode (ustabile) konstruktioner.

Med gode antenner og en masteforstærker, i mit tilfælde 2 Tonna 23 elementer og en "hjemmerullet" masteforstærker med 2 Avantek ATF 21186, rækker den viste transceiver ca. 30 km, næsten lige så langt som almindelig "sladrekassetrafik" på UHF, og så er det altså sjovere at kunne se den, man snakker med. Desuden er der jo glæden ved selv at have bygget transceiveren, og det er jo ikke at foragte.

I den modulære form, som jeg har valgt til opbygningen, er det muligt, blot ved at ændre på sender- og modtagermodulerne, at bygge transceiveren til f.eks. 2320 MHz i stedet, baseband processor, AV-forstærker og spændingsregulatorer kan jo genbruges; i softwaren skal der ændres 4 linjer assemblerkode, så fungerer den også på 2320 MHz; med en ny prescaler, fungerer TBB 206 også på de høje frekvenser. I nordsjællandsområdet er vi en del amatører, der er QRV på FM-ATV, mest omkring 1280 MHz med talk-back frekvens på 434.100 MHz. På denne frekvens kan der som regel findes folk, der enten kan sende et prøvebillede eller give gode råd om FM-ATV.

Som nævnt har det taget mig ca. halvandet år at konstruere og bygge transceiveren, men med en målrettet indsats i f.eks. en lokalafdeling, kan det let gøres på en vinter. Næste vinters byggeprojekt?

Litteraturliste.

Den viste litteraturliste er langt fra udtømmende, og skal derfor opfattes som den speciallitteratur, der er absolut nødvendig at have læst, forinden der startes på at bygge projektet.

Udover de nævnte litteraturhenvisninger, henvises der til generel litteratur, både dansk og udenlandsk, vedrørende UHF kredsløb

1: Kragh: Frekvensmoduleret ATV. Manuskript til artikel i OZ, dateret august 1992, accepteret til publikation.

2: Rösch: Ein 200 mW-Sendetreiber für 1296 MHz in SMD-Technik, UKW-Berichte Heft 4 1987.

3: Kragh: Om fase-låste kredsløb, specielt med henblik på frekvenssynthese. Manuskript til artikel i OZ, dateret september 1992, accepteret til publikation

4: Sibighroug: Understanding Small Microcontrollers Motorola.

5: Motorola: MC68HC05K1 Technical Data Book

6: Siemens: ICs for Communication Data Book

Appendix 1. IARU Region 1 FM-ATV standard

Modulationstype:	F5/F3
Videobåndbredde	5 MHz
Forbetoning af video:	CCIR Recommendation 405,1
Farvebærebølge:	4,433618 MHz (PAL)
Max. øjeblikkeligt modulationsindex	0,5
Spids frekvenssving	3,5 MHz
Max. båndbredde	12 MHz ved -40 dB 18 MHz ved -60 dB
Lyd subcarrier:	5,5 MHz eller 6 MHz
Lyd subcarrier niveau i forhold til spids video:	-14 dB
Lyd subcarrier modulationsindex	0,2

Noter:

1. Videofilter med en båndbredde på 5 MHz skal indsættes i video modulator forstærkeren

2. Der skal indsættes en video spids klipper efter videofiltret.

3. Videosignalet til exciteren skal være DC-klampt, for at undgå flytning af centerfrekvensen i afhængighed af billedindhold.

4. Senderen skal være forsynet med filtre, der hindrer udsendelse af spurii.

5. Hvis video spids frekvenssvinget sænkes under 3,5 MHz, skal lyd subcarrier frekvenssvinget sænkes tilsvarende.

Appendix 2. Programligning til MC68HC05K1

Se næste sider

Appendix 3. Uddrag af datablad med programmeringsvejledning for TBB 206

Programming

Programming of the dividers and status control are performed on a 3-line bus. The assignment of

Appendix 2. Programlistning til MC68HC05K1.

```
*****
*   Program for Motorola MC68HC705K1/MC68HC805K1   *
*   controlling 2 TBB206 frequency synthesizers operating as   *
*   transmitter oscillator and receiver 1. local oscillator in   *
*   a FM ATV transceiver covering the frequency range   *
*   1240 - 1299 MHz in 1 MHz steps.   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*   Author:  OZ7TA   *
*   Version: 2.2   *
*   Date:    March 1993   *
*   s. e. & o.   *
*****
```

org rom

```
*****
*   Initialization   *
*****
```

```
start
porta equ    $00
portb equ    $01
lda    #$f0
sta    ddra      ;set port a i/o direction
lda    #$03
sta    ddrb      ;set port b i/o direction
lda    #$0f
sta    $0010     ;set pull-down on port a inputs
clr    $01       ;set port b to 00
jsr    delay     ;wait for the TBB 206s to settle
clr    $00       ;set port a to 0000xxxx
clr    $00e3
clr    $00e4
clr    $00e5
jsr    send      ;send content of 00e5 on Data
bset   6,$00     ;set IX Enable to high
bset   7,$00     ;set RX Enable to high
```

```
*****
*   Setting up status word 2 and reference   *
*   division ratio for the TBB 206s   *
*   Status word 2:  00 28 32   *
*   Reference word: 00 06 45   *
*****
```

```
bclr   6,$00     ;set TX Enable low
bclr   7,$00     ;set RX Enable low
lda    #$00
sta    $00e0
lda    #$28
sta    $00e1
lda    #$32
sta    $00e2
jsr    datsend   ;send staus word 2
jsr    entx      ;toggle TX Enable
jsr    enr       ;toggle RX Enable
lda    #$00
sta    $00e0
lda    #$06
sta    $00e1
lda    #$45
sta    $00e2
bclr   6,$00
bclr   7,$00
jsr    datsend   ;send reference division ratio
jsr    entx
jsr    enr
```

```
*****
*   Scanning the BCD switches and updating the present channels   *
*   Adding offsets to the channel numbers   *
*   Scrambling data and sending data to the TBB 206s   *
*   RX channel offset: 384a   *
*   TX channel offset: 5dc0   *
*****
```

```
rxch   jsr    delay     ;wait for BCD switches to settle
        bclr   1,$01     ;set to read RX switches
```

```

        jsr    chano    ;get RX channel number
        lda    $00e3
        eor    $00e9    ;check if RX channel is changed
        bne    newrx   ;if so, update RX synthesizer
        jmp    txch    ;continue with TX channel
newrx   lda    $00e9
        sta    $00e3    ;update present RX register
        ldx    #$14
        mul                    ;multiply channel number by 20
        stx    $00e6    ;store high byte
        sta    $00e7    ;store low byte
        lda    #$4a
        add    $00e7    ;add RX offset to low byte
        sta    $00e7    ;and store
        lda    #$38
        adc    $00e6    ;add RX offset to high byte
        sta    $00e6    ;and store
        jsr    scram    ;scrambling data
        bclr   7,$00    ;set RX Enable low
        jsr    datsend  ;send data to RX TBB 206
        jsr    enrxd   ;toggle RX Enable
txch    jsr    delay    ;wait for switches to settle
        bset   1,$01    ;set to read TX switches
        jsr    chano    ;get TX channel number
        lda    $00e4
        eor    $00e9    ;check if TX channel is changed
        bne    newtx   ;if so, update TX synthesizer
        jmp    rxch    ;return to scan the RX switches
newtx   lda    $00e9
        sta    $00e4    ;update TX frequency register
        ldx    #$14
        mul                    ;multiply channel number by 20
        stx    $00e6    ;store high byte
        sta    $00e7    ;store low byte
        lda    #$c0
        add    $00e7    ;add TX offset to low byte
        sta    $00e7    ;and store
        lda    #$5d
        adc    $00e6    ;add TX offset to high byte
        sta    $00e6    ;and store
        jsr    scram    ;scrambling data
        bclr   6,$00    ;set TX Enable low
        jsr    datsend  ;send data to TX TBB 206
        jsr    entxd   ;toggle TX Enable
        jmp    rxch    ;return to scan the RX switches

```

```

*****
*                               SUBROUTINES                               *
*****

```

```

*****
*                               Subroutine toggling the TX Enable       *
*****

```

```

entxd   bset    6,$00
        nop
        bclr   6,$00
        nop
        bset   6,$00
        rts

```

```

*****
*                               Subroutine toggling the RX Enable       *
*****

```

```

enrxd   bset    7,$00
        nop
        bclr   7,$00
        nop
        bset   7,$00
        rts

```

```

*****
*                               Subroutine serially transmitting the content of
*                               RAM 00e5 to TBB 206 in question          *
*****

```

```

send     lda    #$08
        bclr   4,$00    ;ensure Clock is low
sendst   asl    $00e5    ;shift a bit into c
        bcs    setdata  ;check if c is high
        bclr   5,$00    ;set Data to low

```

```

nop
bset      4,$00    ;set Clock to high
nop
bclr      4,$00    ;set Clock to low
jmp       end      ;jump to end
setdata   bset      5,$00    ;set Data to high
nop
bset      4,$00    ;set Clock to high
nop
bclr      4,$00    ;reset Clock
end        deca
bne       sendst   ;check if all 8 bits are sent
bclr      5,$00    ;set data to low
rts

```

```

*****
*          Subroutine scanning two BCD switches and returning          *
*          channel settings in hex in register 00e9                    *
*****

```

```

chano     bset      0,$01    ;set to read tens digit
          lda       $00      ;read tens digit
          and       #$0f     ;use only input from BCD switch
          ldx      #$0a
          mul       ;multiply acca by 10
          sta      $00e9     ;store result
          bclr     0,$01     ;set to read ones digit
          lda       $00      ;read ones digit
          and       #$0f     ;use only input from BCD switch
          add      $00e9     ;add ones to tens
          sta      $00e9     ;and store
          clc
          lda      $00e9
          sub      #$28
          bcs      minifr    ;check if channel < 40
          jmp      out       ;exit if channel >= 40
minifr    lda      #$50
          sta      $00e9     ;force channel to 80
out       rts

```

```

*****
*          Subroutine sending the contents of RAM 00e0,                *
*          00e1 and 00e2 to the TBB 206 in question                    *
*****

```

```

datSEND   lda      $00e0    ;get 1. byte
          sta      $00e5
          jsr      send     ;send 1 byte
          lda      $00e1    ;get 2. byte
          sta      $00e5
          jsr      send     ;send 2. byte
          lda      $00e2    ;get 3. byte
          sta      $00e5
          jsr      send     ;send 3. byte
          rts

```

```

*****
*          Subroutine scrambling the contents                          *
*          of RAM 00e6 and 00e7 and storing the result in            *
*          RAM 00e0 to 00e2 in accordance with the TBB 206         *
*          scrambling procedure                                       *
*          Content of RAM 00e0 to 00e2 after scrambling:            *
*          00e0:  00 + 6 MSBs of A                                    *
*          00e1:  LSB of A + 7 MSBs of N                             *
*          00e2:  5 LSBs of N + 111                                  *
*****

```

```

scram     lda      $00e7    ;get low byte
          and      #$7f     ;set MSB to 0
          lsra     ;shift 1 place to right
          sta      $00e0    ;store in RAM 00e0
          lda      $00e6    ;get high byte
          lsra
          lsra
          lsra     ;shift 3 places to right
          sta      $00e1    ;store temporarily
          lda      $00e7    ;get low byte
          rora     ;roll LSB into carry
          lda      $00e1
          rora     ;roll LSB from 00e7 into MSB of 00e1
          sta      $00e1    ;and store
          clc

```

```

        lda    $00e7    ;get low byte
        rora   ;roll MSB into carry
        lda    $00e6    ;get high byte
        rora   ;roll MSB from 00e7 into LSB of acca
        lsll
        lsll
        lsll          ;shift 5 LSBs into 5 MSBs
        ora    #$07     ;set 3 LSBs to 111
        sta    $00e2    ;and store in 00e2
        rts

*****
*          Subroutine giving a delay of 99 msecs when using          *
*          a 3.58 MHz clock crystal                                  *
*****
delay   sta    $00e8    ;save acca content
        lda    #$74     ;set number of outer loops
oulp    clr
inlp    decx           ;set number of inner loops to 256
        bne   inlp     ;do inner loop
        deca
        bne   oulp     ;do outer loop
        lda    $00e8    ;restore acca
        rts

*****
        org    vectors
        dw    rom
        dw    rom
        dw    rom
        dw    start

```

the content of the telegrams to the functional units is made with a target address that is transmitted in the last three bits of the telegram before the EN signal.

In addition to the assignment data, the target address also contains status information: single/dual-modulus and synchronous/asynchronous data transfer.

The following addresses are valid:

- ...001 EN: status 1
- ...010 EN: status 2
- ...011 EN: status 3
- ...110 EN: N divider = single-modulus
- ...111 EN: N/A divider = dual-modulus
- ...101 EN: R divider + synchronous transfer
- ...100 EN: R divider + asynchronous transfer

The decision single-/dual-modulus is made simultaneously with the programming of the N or N/A divider.

Target address 000 is reserved for testing purposes.

In TBB 206 there is provision for newly programming the dividers synchronously with the reference frequency. This ensures that the control starts from the momentary status of the phase detector when a frequency/channel is altered. In the lock-in situation is particular, the control operation begins with a phase difference of zero. Thus there is no possibility of brief error signals on the phase detector, as can occur in asynchronous data transfer.

In asynchronous programming the "new data" are transferred asynchronously to the status of the R, N

(N/A) divider into intermediate registers L2. There is no guarantee that all data, i.e. R and N (N/A) dividers, will be present in time, so this will result in error signals on the phase detector output - because the division factors in the dividers do not correspond - that are weighted by the system as a phase difference and thus additionally have to be compensated.

Programming

Synchronous programming offers the possibility of data transfer while maintaining the phase difference that exists at the time of (internal) programming. This is done as follows.

1. Setting of the mode "synchronous transfer" by appropriate programming of the R divider. This setting is maintained until programmed differently. With the EN of the R-divider programming the "new data" are only read into capture register L1.

2. Programming of the N or N/A divider. With the EN signal of the programming the "new" data are likewise only read into capture register L1. At the same time synchronous transfer into intermediate register L2 is primed for all dividers. This transfer is then made with the next zero crossing of the particular divider. A phase difference existing at the instant of transfer is maintained, this being the start condition for the further control operation, based on the "newly" read data.

Because of the synchronizing circuit therefore, the delay in data transfer is maximally $2 \times 1/f_{ref}$.

The synchronous programming operation is

always initiated by the EN signal of the N or N/A divider and also applied to the R divider, even if its data have not been altered. If the data content of the R divider (eg for a change of reference frequency) is to be altered, this consequently has to be done before programming of the N og N/A divider.

Synchronous transfer is of particular advantage if extreme channel jumps are to be made with short settling time. For fast "coarse" settling a switch is made to a greater reference frequency, thus increasing the bandwidth of the loop. The original reference frequency is returned to when the "quasi-set" status is reached. Control to the "actual" value is then made with the known settling response as in small channel jumps. Synchronous transfer ensures that no extra phase errors occur as a result of changing the reference frequency.

Programming: Tables

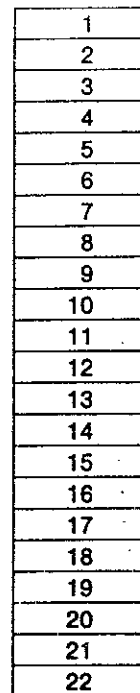
Status bit		Multifunction outputs	
MFO 1	MFO2	MFO 1 (push pull)	MFO 2 (open-drain N)
0	0	FRN	FVN
0	1	ØVN	ØR
1	0	PRT 1	PRT 2
1	1	PRT 1	CURRENT*

* Input

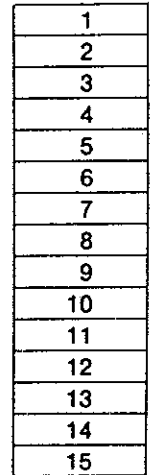
Status bit		PD output current
PC current 1	PD-current 2	typ. (mA)
0	0	0.25
1	0	0.5
0	1	1.0
1	1	2.0

Status Bit		Voltage-doubler frequency
Voltage doubler		f_{INT}
1	2	Off
0	0	$R_i: 1$
1	0	$R_i: 2$
0	1	$R_i: 4$
1	1	

N/A Divider Dual-modulus

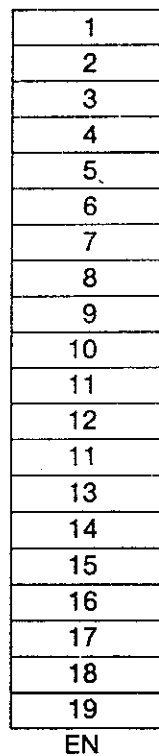


N Divider Single-modulus



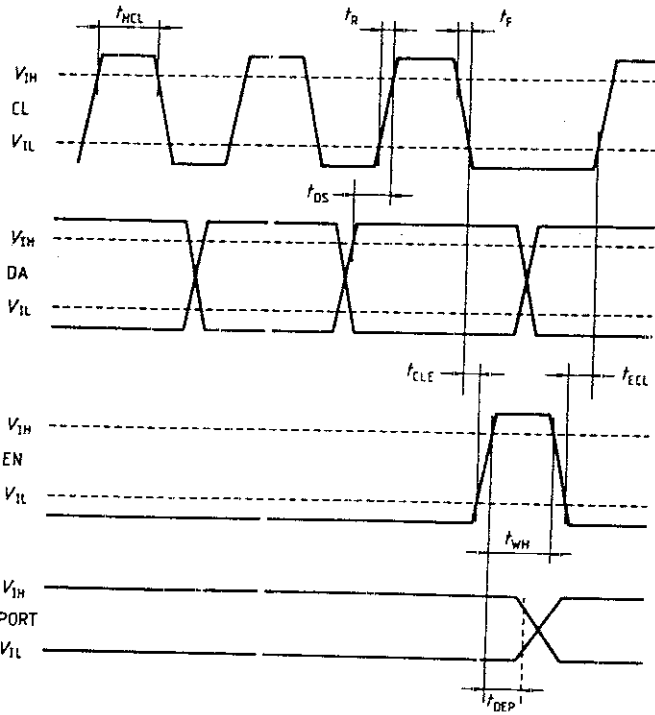
Address 111: dual-modulus
Address 110: single-modulus

R Divider

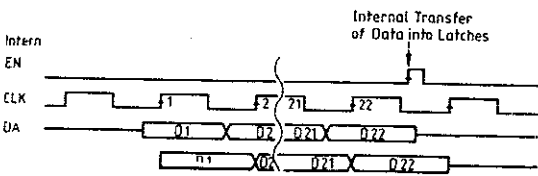
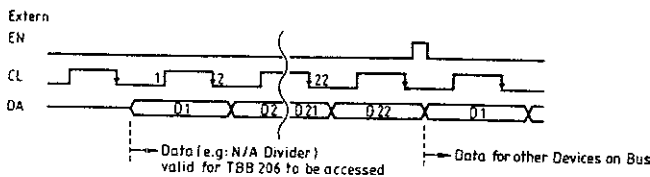


Address 101: synchronous transfer*
Address 100: asynchronous transfer*

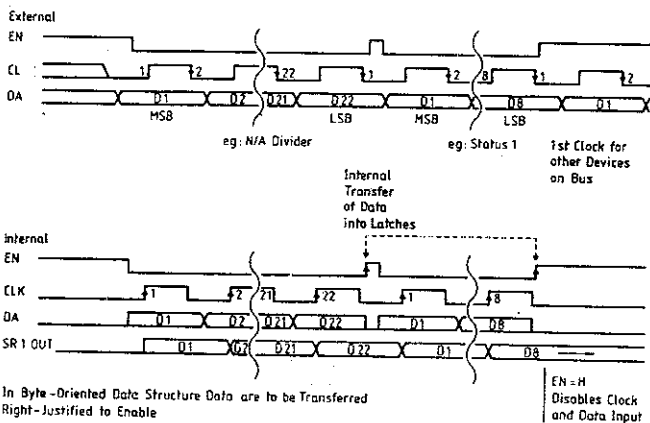
Pulse Diagram for 3-Line Bus



Pulse Diagram for 3-Line Bus: "Dynamic" EN Control



Pulse Diagram for 3-Line Bus



In Byte-Oriented Data Structure Data are to be Transferred Right-Justified to Enable

EN = H Disables Clock and Data Input

Programming: Times

Initialization, sequence: R divider before N divider
 Single-modulus: Σ 50 bits
 Status 2: 16 bits

R divider 19 bits
 N divider 15 bits

Dual-modulus: Σ 57 bits
 Status 2: 16 bits
 R divider 19 bits
 N/A divider 22 bits
 Change of channel, at same reference frequency
 Single-modulus 15 bits
 Dual-modulus 22 bits

If the reference frequency is altered, the R divider also has to be loaded (+19 bits).
 Maximum repetition rate for channel change $f_{RI}:N$.

Note:

After the supply voltage is applied, there must be at least three clock pulses on the clock input, with EN = Low, before the first telegram.

- * Status 2 must always be used to initialize. (Bit 1-8 of status 2 have no default values.)
- ** Only valid if multifunction outputs are in PORT status, otherwise don't care.
- *** DA of TBB 206 can only produce Low (open-drain output).
- Address 111: dual-modulus
- Address 110: single-modulus
- Address 101: synchronous transfer*
- Address 100: asynchronous transfer*
- * Also determines transfer mode for N, N/A divider.

	min.	max.	Unit
f_{CL} Clock frequency		1,3	MHz
t_{HCL} H pulse width (CL)	400		ns
t_{DS} Data setup	100		ns
t_{CLE} Setup time clock-enable	5		ns
t_{ECL} Setup time enable-clock	100		ns
t_{WH} H pulse width (enable)	80		ns
t_R, t_F Transition times		10	μ s
t_{DEP} Delay time enable port	1		μ s

Programming: Transmission Protocol Status 1

MFO1
MFO2
PD polarity
Modulus output
Preamplifier (standby)
Anti-backlash pulse
Voltage doubler 1
Voltage doubler 2
PORT 1
PORT 2
Divider (standby)
PD current 1
PD current 2
0 0
0 Target address 1
1 0

1
2
3
4
5
6
7
8
EN

Status 2*

	0	1
1		See table
2		pos
3	neg	open-drain
4	push-pull	active
5	standby	40 ns
6	20 ns	See table
7		1**
8		1**
9	0**	active
10	0**	See table
11	standby	
12		
13		
14		
15		
16		

EN

Status 3: Lock Detector read

	0	1
1		Target address
2		
3		

EN

1

EN

Status lock detect: lock-in = Low***

OZ

Fra andre blade

En enkel headset-mikrofon.

På hovedtelefonens bøjle lader en "boom" til en mikrofon sig let clipse, og med en lille electret mikrofon kan man på denne simple måde lave sig et headset og nyde dets fordele - f. eks. begge hænder fri under arbejdet.

Dette viser Doug Brede, W3AS/7 klart og tydeligt i en lille artikel.

A 5 \$ Headset Mike, QST JAN 1993 pp 48-49

Lysfølsom ladecontroller til solceller.

I OZ har dette emne været omtalt bl.a. i januar (side 41) og marts (side 153) 1993, og nu beskriver WB8VGE sin nye controller, der er på et print med meget beskedne dimensioner - ca. 9 x 10 cm. Den klarer 6A, men kan let dimensioneres for 3 A. Der er også anvisning på de ændringer, der skal til, hvis der arbejdes med 24 V i stedet for de 12 V, som er omtalt i artiklen.

Michael Bryce, WB8VGE, The SunSwitch, QST OCT 1993 pp 24-27

Teoretiske betragtninger over modtagerdata og -størrelser.

DJ6HP fortsætter nu sin gennemgang af dette emne, og det er krydsmodulation og intermodulation, der er under luppen i anden del, og det bliver gjort grundigt! Fortsættelse er lovet.

Hans-Joachim Pietsch, DJ6HP, Theoretische Betrachtung von Empfängerdaten und -größen (2), CQ DL 11/93 pp. 747-751.

OZ8T

Fra andre blade

Transvertere for 13 cm båndet.

Dette emne har vi hidtil ikke beskæftiget os ret meget med i OZ, men da der for nylig er kommet en interessant konstruktionsartikel fra den kendte SHF-mand DB6NT's hånd [1], og da der også har været en lignende i QST for et lille års tid siden [2] må det være på sin plads at omtale disse. De kan måske anspore flere til at blive QRV på SHF - et område hvor så mange i dag finder de nye udfordringer!

Begge konstruktioner bruger 144 MHz som mellemfrekvens altså f.eks. i form af en 2 m transceiver, men ellers adskiller de sig ret væsentligt fra hinanden - DB6NT's er med indbygget lokaloscillator og yder 1 W output, WA8NLC's er med separat lokaloscillator - beskrevet i [3] - og yder 10-20 mW, idet han også ønsker, at den kan anvendes som portabel station til f.eks. anvendelse på Field Day. Til hjemmebrug benyttes så et udvendigt PA-trin. Er det ønskeligt at arbejde med et endnu lavere støjtal i modtagerdelen, er en passende forstærkerbeskrevet i [4].

Begge gør brug af de fine muligheder printed wiring rummer for klar, logisk og elegant opbygning af kredsløbene især af f.eks. båndfiltre. SMD-komponenter finder også udbredt anvendelse. Begge artikler er særdeles grundige - DB6NT's med en omfattende stykliste og printudlæg, der måler 90 x 250 mm. Begge anfører endvidere leverandører af print, komponenter og komplette byggesæt.

Litteraturlisten til WA8NLC's artikel er omfattende, idet den foruden at henvise til artikler om de tilsluttede apparater også går noget videre og viser, hvor der er mere viden at hente om dette spændende hjørne af mikrobølgeteknikken.

En enkelt reference er til "ham radio" men ellers er resten til QST og andre ARRL-publikationer.

Men det må i denne forbindelse også omtales, at for 1,3 GHz har DB6NT også konstrueret en transverter, der først blev beskrevet i [5] og så for et halvt år siden i [6]. Modtageren har NF < 2 og senderen yder 1,5 W out. Og det er lige som i 2,3 GHz-transverteren med nyeste teknologi omhyggeligt - trin for trin - beskrevet og illustreret (Printudlægget måler 96 x 120 mm).

Fra litteraturlisten til [6] bør et par henvisninger [7] og [8] medtages her.

- 1 Michael Kühne, DB6NT, 2,3-GHz-Lineartransverter, CQ DL 11/93 pp 752-756,
- 2 Jim Davey, WA8NLC, A No-Tune Transverter for the 2304-MHz Band, QST DEC 1992 pp 33-39
- 3 Rick Campbell, KK7B, A Clean, Low-Cost Microwave Local Oscillator, QST JUL 1989 pp 15-21,
- 4 Al Ward, WB5LUA, Simple, Low-Noise Microwave Preamplifiers, QST MAY 1986 pp 31-36 og 75,
- 5 Michael Kühne, DB6NT, 1,3-GHz-Lineartransverter, Dubus 4/92,
- 6 Michael Kühne, DB6NT, 1,3-GHz-Lineartransverter, cq-DL 5/93 pp 313-316,
- 7 Al Ward, WB5LUA, Low-Noise VHF and L-Band GaAs FET Amplifiers, RF Design Feb 1989 pp. 38-...
- 8 Günter Schwarzbeck, DL1BU, Rauscharmer Vorverstärker für das 23-cm-Band, cq-DL 11/85 pp 624-...

OZ8T



Generalagent for
YAESU MUSEN

BETAFON

ISTEDGADE 79 · 1650 KØBENHAVN V · TELEFON 31 31 02 73