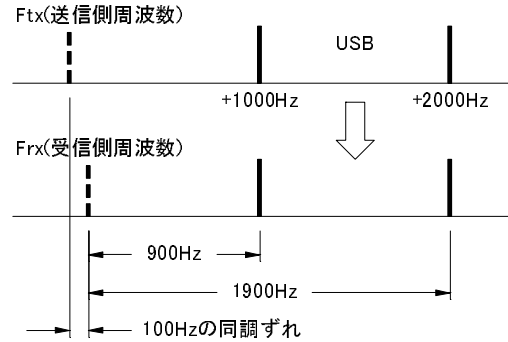


本機の使用目的 JusTune-SSB は、キャリアが抑圧されている J3E 電波の周波数を送受間で超精密に合わせるための、送信側及び受信側の補助具です。

正確同調の必要性 J3E 電波を受信する場合、例えば右図のように、受信側のキャリア周波数が送信側のそれより高いとすると、元の信号よりも低い音で聞こえるということになり、音声の忠実度が下がります。

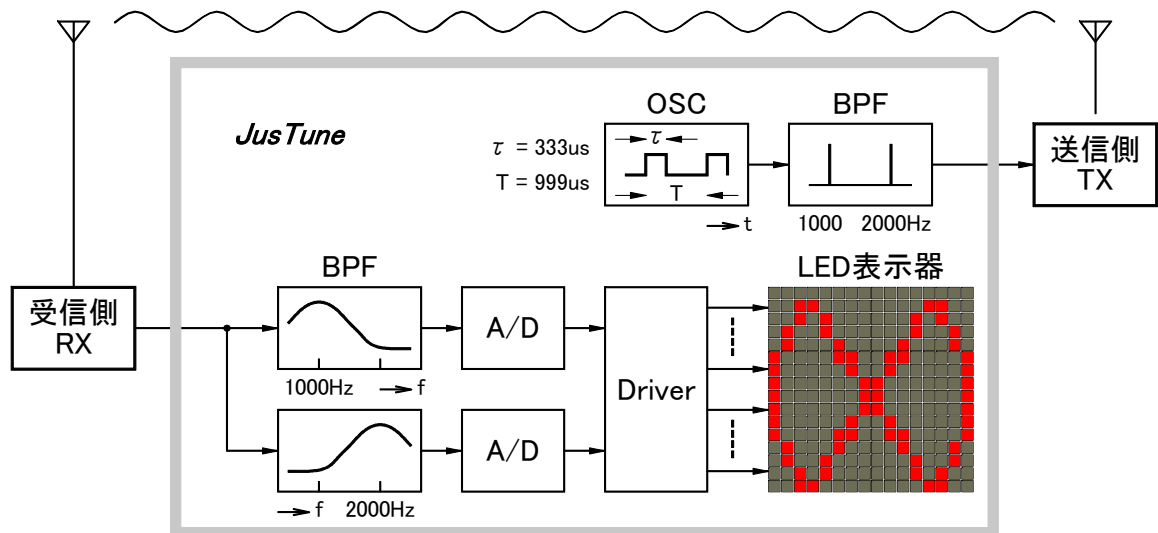
受信側では送信側のキャリア周波数を知る術がありませんから、受信側のキャリア周波数は送信側のそれと ±数 10Hz 外れていることもあります。



動作の説明 下図に JusTune の構成を示します。参照周波数発生部から、周波数比が 1 : 2 となる 2 トーンが送信側送信機に送られます。周波数は約 $1,000\text{Hz}$ と約 $2,000\text{Hz}$ ですが、それが変動しても周波数比が正確であれば支障ありません。

受信信号を BPF で各トーンに区別した後に、A/D 変換します。それぞれの値でマトリックス LED (16 列 16 桁を使用) を駆動します。

受信側のキャリア周波数が送信側より 100Hz 高いとすると、復調信号の周波数は 900Hz と $1,900\text{Hz}$ となり、両者は整数比が維持されていませんからリサージュ図形は静止せず、離調状態であることが分かります。

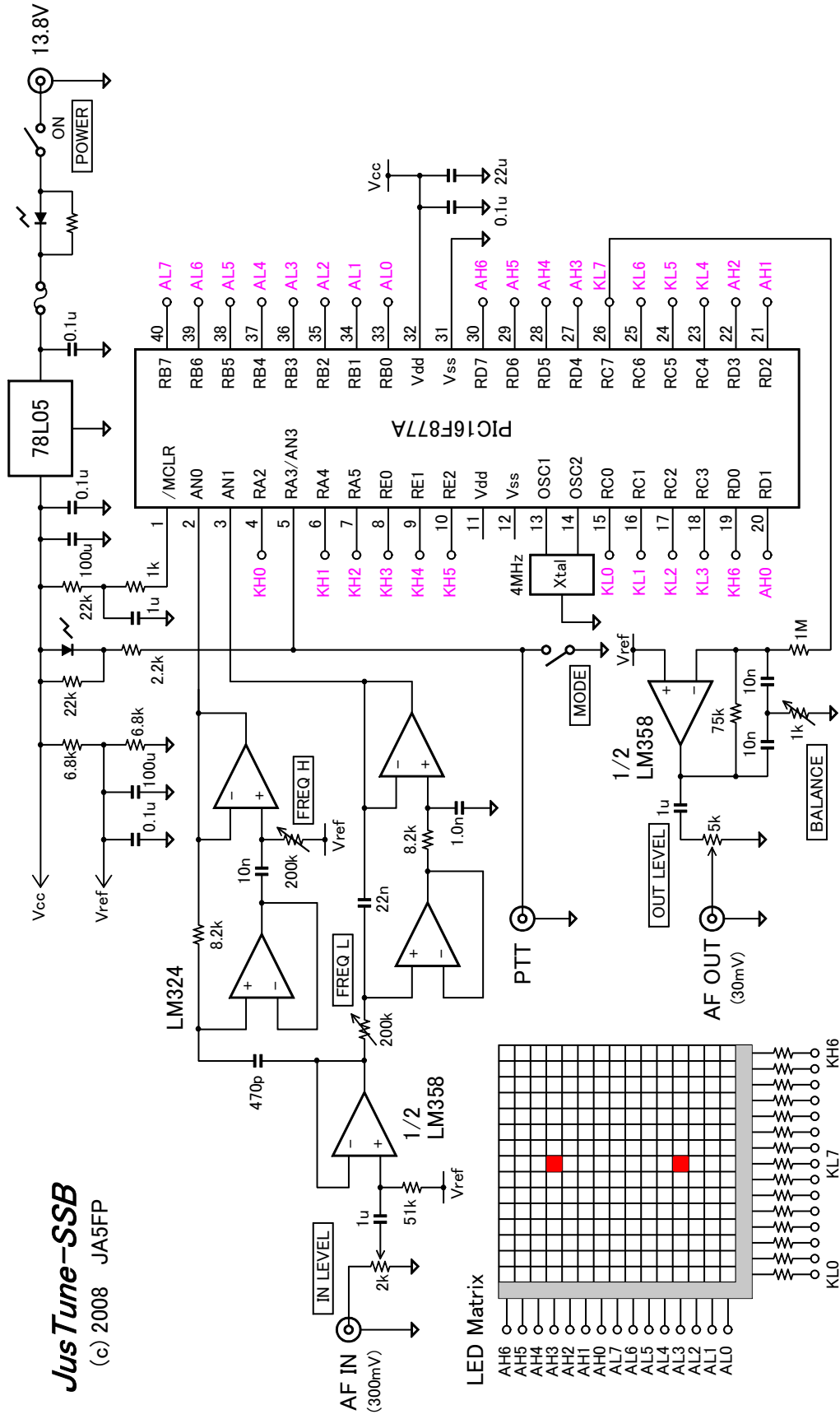


主な使用部品 小型化するために PIC で実現しています。

IC : PIC16F877A LM324 LM358 など

Jus Tune-SSB

(c) 2008 JA5FP



アセンブラ コード

```
; TITLE "justune-ssb.asm (c) 2008 JA5FP"
list p=16f877a
#include <p16f877a.inc>
errorlevel -302
__CONFIG _LVP_OFF & _CPD_OFF & _CP_OFF & _PWRTE_ON & _WDT_OFF & _WRT_OFF & _XT_OSC

hrz EQU 024h ; 水平振幅データ
vrt EQU 025h ; 垂直振幅データ
tau1 EQU 026h ; カウンタ変数 1
tau2 EQU 027h ; カウンタ変数 2
;*****
ORG 000h ;
goto main
ORG 004h ;

main CLRF STATUS
BSF STATUS,RPO ; Bank1
CLRF INTCON ; 割込み禁止
MOVLW b'00001011'; RA2 RA4 RA5 を出力
MOVWF TRISA ;
MOVLW b'00001110'; RA3 をデジタル入力
MOVWF ADCON1
CLRF TRISB ; PORTB を出力に設定
CLRF TRISC ; PORTC を出力に設定
CLRF TRISD ; PORTD を出力に設定
CLRF TRISE ; PORTE を出力に設定
BCF STATUS,RPO ; Bank0
MOVLW b'00111111'; PortA 出力初期化
MOVWF PORTA ;
CLRF PORTB ; PortB 出力初期化
MOVLW b'11111111'; PortC 出力初期化
MOVWF PORTC
MOVLW b'00000001'; PortD 出力初期化
MOVWF PORTD
MOVLW b'00000111'; PortE 出力初期化
MOVWF PORTE

tr BTFSS PORTA,3 ; モード切替
goto osc
BSF STATUS,RPO ; Bank1
MOVLW b'00000100'; AN0 AN1 AN3 を A/D 入力 ADRESH へ左詰
```

```

MOVWF ADCON1 ;
BCF STATUS,RPO ; Bank0
MOVLW b'01001001'; クロック=4MHz/8=2us   AN1 対象
MOVWF ADCON0 ;
call w20 ; チャージまで 20us 待機
BSF ADCON0,GO ; 変換開始
wad0 BTFSC ADCON0,GO
goto wad0
MOVF ADRESH,W ; 水平データ保存
MOVWF hrz
NOP
NOP
MOVLW b'01000001'; クロック=4MHz/8=2us   AN0 対象
MOVWF ADCON0 ;
call w20 ; チャージまで 20us 待機
BSF ADCON0,GO ; 変換開始
wad1 BTFSC ADCON0,GO
goto wad1
MOVF ADRESH,W ; 垂直データ保存
MOVWF vrt
SWAPF vrt ; 垂直データ精度低減
MOVLW 0x0f
ANDWF vrt,F
SWAPF hrz ; 水平データ精度低減
MOVLW 0x0f
ANDWF hrz,F
fork MOVF hrz,W ; hrz=0x0f,0x0e
SUBLW 0x0d
BTFSC STATUS,C
goto kh2
BCF PORTD,0 ; KH6 を出力
goto fora
kh2 MOVF hrz,W ; hrz=0x0d,0x0c,0x0b
SUBLW 0x0a
BTFSC STATUS,C
goto kh1
MOVF hrz,W
call cvhr
MOVWF PORTE ; KH2 を出力
goto fora
kh1 MOVF hrz,W ; hrz=0x0a,0x09,0x08
SUBLW 0x07
BTFSC STATUS,C

```

```

goto kl
MOVF hrz,W
call cvhr
MOVWF PORTA ; KH1 を出力
goto fora
kl MOVF hrz,W ; hrz=0x07,0x06,0x05,0x04,0x03,0x02,0x01,0x00
call cvhr
MOVWF PORTC ; KL を出力
fora MOVF vrt, W ; vrt=0x0e,0x0d,0x0c,0x0b,0x0a,0x09,0x08
SUBLW 0x07
BTFSC STATUS,C
goto al
MOVF vrt,W
call cvvt
MOVWF PORTD ; AH を出力
goto enda
al MOVF vrt,W ; vrt=0x07,0x06,0x05,0x04,0x03,0x02,0x01,0x00
call cvvt
MOVWF PORTB ; AL を出力
goto enda
enda MOVLW d'26'
MOVWF tau1
decy DECFSZ tau1
goto decy
MOVLW b'00111111'; ブランキング
MOVWF PORTA ;
CLRF PORTB ; PortB 出力初期化
MOVLW b'11111111'; PortC 出力初期化
MOVWF PORTC
MOVLW b'00000001'; PortD 出力初期化
MOVWF PORTD
MOVLW b'00000111'; PortE 出力初期化
MOVWF PORTE
BSF STATUS,RP0 ; Bank1
MOVLW b'00001110'; リフレッシュ RA3 をデジタル入力
MOVWF ADCON1 ;
BCF STATUS,RP0 ; Babk0
goto tr
osc MOVLW b'00001000'; リファレンスモード AL3xKL7 点灯
MOVWF PORTB ;
MOVLW b'00010001'; AH3xKL7 点灯
MOVWF PORTD ;
osc1 MOVLW b'11111111'; RC7 セット

```

```

MOVWF PORTC ;
call wt0 ; 333us 待ち
NOP ; 調整用
NOP ; 調整用
NOP ; 調整用
NOP ; 調整用
MOVLW b'01111111'; RC7 クリア
MOVWF PORTC
call wt0 ; 666us 待ち
call wt0 ;
NOP ; 調整用
goto tr ; 発振継続

```

```

; -----

```

```

wt0 MOVLW d'32'; 322us サブルーチン
MOVWF tau2 ; ループカウンタ
wt1 MOVLW d'2'
MOVWF tau1 ; 内ループカウンタ
wt2 DECFSZ tau1
goto wt2
DECFSZ tau2
goto wt1
NOP ; 調整用
NOP ; 調整用
return

```

```

w20 MOVLW d'5'; 20us サブルーチン
MOVWF tau1 ; ループカウンタ
w2 DECFSZ tau1
goto w2
return

```

```

cvhr ADDWF PCL,F ; マトリックス変換サブブルーチン
RETLW b'11111110'; 00h KL0(RC0)
RETLW b'11111101'; 01h KL1(RC1)
RETLW b'11111011'; 02h KL2(RC2)
RETLW b'11110111'; 03h KL3(RC3)
RETLW b'11101111'; 04h KL4(RC4)
RETLW b'11011111'; 05h KL5(RC5)
RETLW b'10111111'; 06h KL6(RC6)
RETLW b'01111111'; 07h KL7(RC7)
RETLW b'11111011'; 08h KH0(RA2)
RETLW b'11101111'; 09h KH1(RA4)

```

```
RETLW b'11011111'; 0ah KH2(RA5)
RETLW b'11111110'; 0bh KH3(RE0)
RETLW b'11111101'; 0ch KH4(RE1)
RETLW b'11111011'; 0dh KH5(RE2)
cvvt ADDWF PCL,F
RETLW b'00000001'; 00h AL0(RB0)
RETLW b'00000010'; 01h AL1(RB1)
RETLW b'00000100'; 02h AL2(RB2)
RETLW b'00001000'; 03h AL3(RB3)
RETLW b'00010000'; 04h AL4(RB4)
RETLW b'00100000'; 05h AL5(RB5)
RETLW b'01000000'; 06h AL6(RB6)
RETLW b'10000000'; 07h AL7(RB7)
RETLW b'00000011'; 08h AH0(RD1)
RETLW b'00000101'; 09h AH1(RD2)
RETLW b'00001001'; 0ah AH2(RD3)
RETLW b'00010001'; 0bh AH3(RD4)
RETLW b'00100001'; 0ch AH4(RD5)
RETLW b'01000001'; 0dh AH5(RD6)
RETLW b'10000001'; 0eh AH6(RD7)
```

END