Synthesized Driver for Beacons with WSJT Data

Andy Talbot G4JNT ***October 2012***

The driver is intended as a versatile source adaptable to all VHF and UHF amateur band beacons transmitting frequency agile data modes such as ISCAT, JT65 and JT4. The hardware is based around an LMX2470 Fractional-N synthesizer, updated in real time from a PIC controller. This, in turn, takes in timing information delivered from an external NMEA data source, and optional 1 PPS signal for enhanced accuracy. The synthesizer chip requires an external reference signal, typically at 10MHz. Both signals can come from a GPSDO reference

As the synth chip can only work over the range 500 – 2600MHz, the output is divided down for lower bands, typically by 8 times although higher or lower values are possible.

The unit as supplied for GB3WGI is made up of three modules:

- An LMX2470 development module, modified to remove and bypass its own PIC controller (See http://www.g4jnt.com/LMX2470_DevModule.pdf)
- A PCB carrying the VCO, divider and driver amplifier
- A PCB carrying a PIC for controlling the synth chip
- A LM7808 8V regulator supplying Vin to all three PCBs.

The circuit diagrams for each module and a photo of the complete assembly are shown below.

VCO/Divider Module (Lower right of Photograph)

The packaged VCO runs at approximately 1156MHz. An attenuated output at this frequency is fed back to the PLL module at a level of -5dBm. The VCO also supplies a PE3513 divide by 8 chip. This will function with up to 1.5GHz maximum input frequency. The divider requires 3.0V supply rail which comes from a dedicated LM317L regulator.

The divider output feeds an SGA6289 broadband amplifier MMIC which in turn drives a BFG591 output transistor for a maximum output power (from a 12.0V supply) in the region of 600mW. For the application here, an 8V regulator (not shown in the diagrams) is installed for the entire driver module. At this Vcc the final stage can deliver around 400mW maximum. A preset resistor in the emitter of the BFG591 allows the output level to be reduced to less than +10dBm

Code Generator (Top centre of the Photograph)

A 16F627A PIC takes in the NMEA timing data and 1PPS signal and determines the correct start point for each part of the message transmitted. Depending on the data type, the synthesizer is programmed in real time over its SPI interface. An external key line is provided for ON-OFF keying a PA stage via a separate modulator. The polarity is ground to Tx, open-circuit for space.

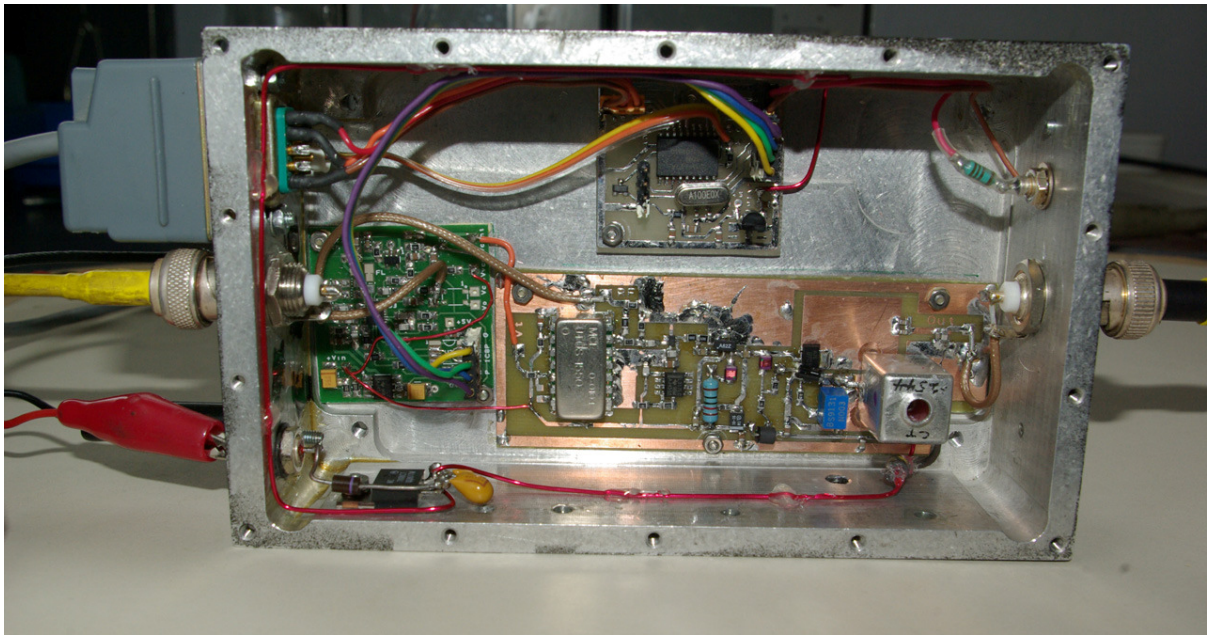
For JT65, the Fractional-N denominator is set for a frequency grid spacing corresponding to the required tone spacing. For JT65B and allowing for the divide by 8 stage, this becomes 11025 / 2048 * 8 = 43.0664Hz. The chip can then be rapidly updated for each symbol of the dat amode by altering the value of just one register, the Fractional Numerator, directly with a value added to an offset corresponding to the JT65 tone number from 0 to 65.

Two seconds after the JT65B sequence is finished. The synthesizer is reprogrammed to deliver an exact carrier frequency. The JT65 frequency grid cannot be utilised as this will not give an exact value, so instead three registers of the chip are updated to set a new grid of 40Hz, corresponding to 5Hz steps at the final output. The JT65 frequency grid is restored immediately before the sequence is due to be sent at the next even minute.

Once the carrier frequency is established, the CW information is sent on the external key line. The CW message is sent twice, the first time immediately after the JT65 sequence, once the frequency grid is re-established, then again at the odd minute:30 seconds point determined from the NMEA timing data. The key line is used to inhibit RF during the slight glitch that occurs while the grid is being updated.

The three wire SPI interface is on the 4-way ribbon cable (colours yellow to purple) visible in the photograph. Timing information comes in via the 9 pin connector (pin compatible with the G3RUH GPSDO) and links to the controller board via the three way ribbon cable (red-orange-yellow).

The remaining four way header shown unconnected is for in-circuit programming of the PIC. The connections for this are the 'JNT non-optimum standard' for PIC-ICP, and are NOT the same as those on the PICKIT programmer – an interface lead will have to be made up if reprogramming of the PIC contents is attempted.



Reprogramming the PIC

Table 1 shows the relevant part of the PIC code containing frequency and message data. This is all stored in EE memory, and with care, can be updated by reprogramming this part of the chip only. Note, the CW message can be a variable length field and MUST have the NULL Terminator. The variable length option means it must occupy the last position in the EE data. Be very careful when editing EE data not to accidently add extra bytes in the preceding sections

To change message data only, the two sections highlighted in red are applicable. The CW message is self explanatory and must be in upper case; note the null terminator. The include file *jt65symb.inc* (or any other file name you may choose) has to be generated using the utility *GENJT65.EXE* contained in http://www.g4jnt.com/JT65CODE.ZIP Note this calls up *JT65CODE.EXE* written by K1JT, also included in the .ZIP archive, so make sure both are saved in the same subdirectory)

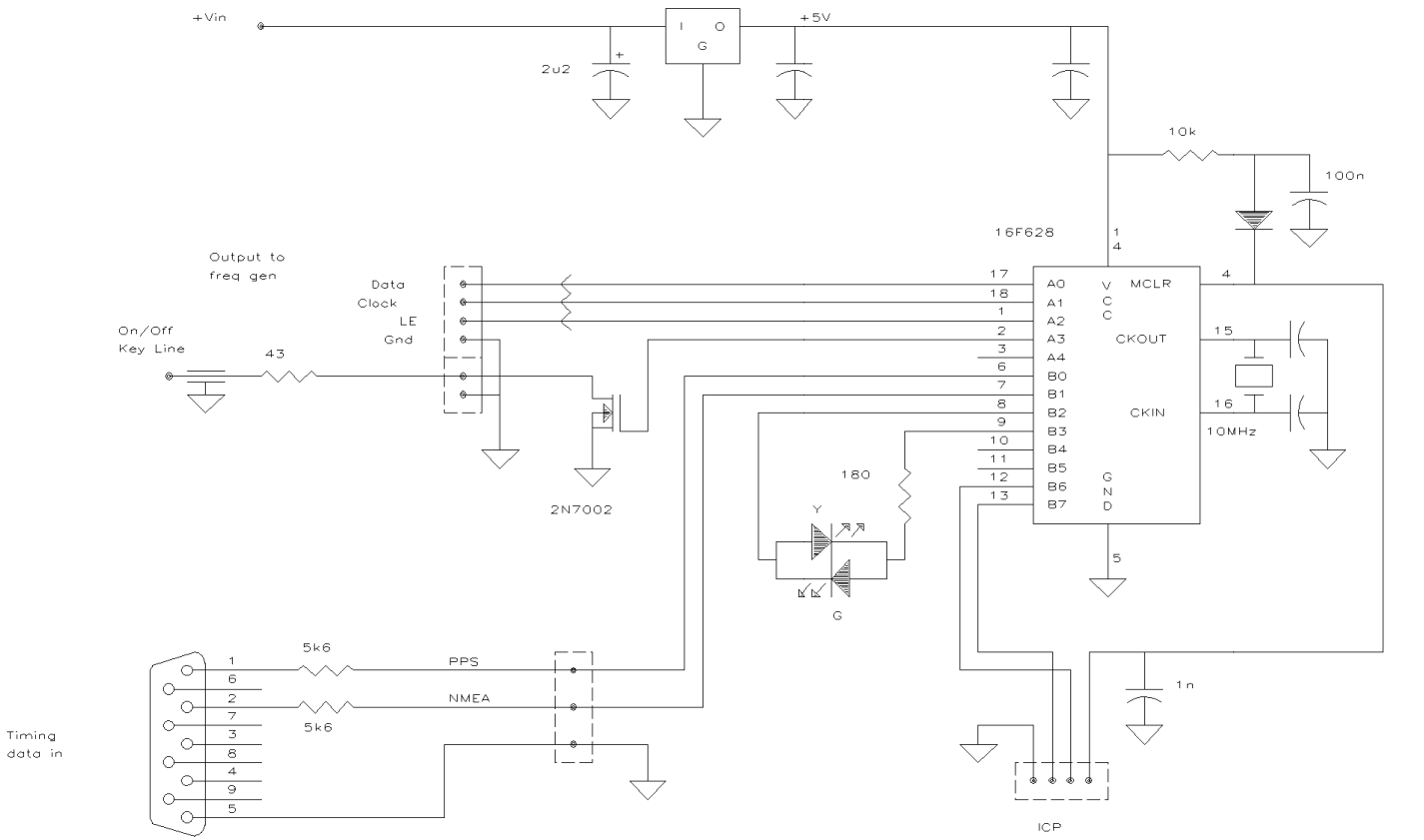
To change the frequency is a more complex procedure. Consult the data sheet for the LMX2470 to work out the values of N, F and D needed. Two sets of values will be required. One for a frequency grid for JT65 and one for a grid with nice round numbers of Hz to set the carrier frequency accurately.
Only Reg0, Reg1 and Reg7 carry frequency and grid information, highlighted in green in the listing below. The other registers must not be touched as they control PLL configuration information and are only programmed into the chip once at startup.

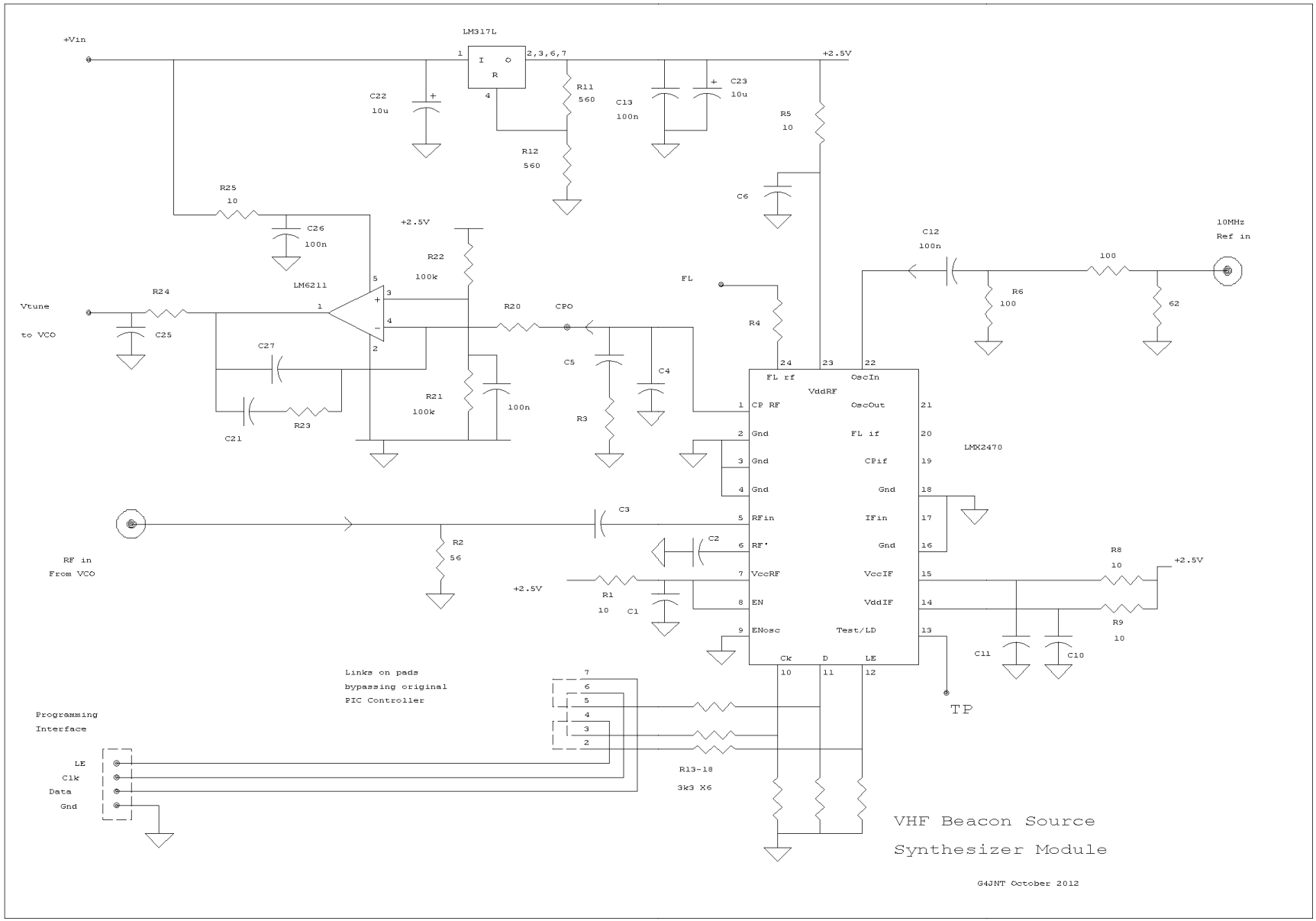
Note that in the listing shown, there are two possible values for Reg 0 (containing the 12 LSBs of the frequency setting) – one of which is remmed out. ONE of these lines MUST ALWAYS be remmed out – only a single Reg0 value must be used! The two possible values correspond to the desired JT65 reference or sync tone when the receiver is tuned to place the nominated carrier frequency (defined by the other grid) to either 800Hz or 1500Hz

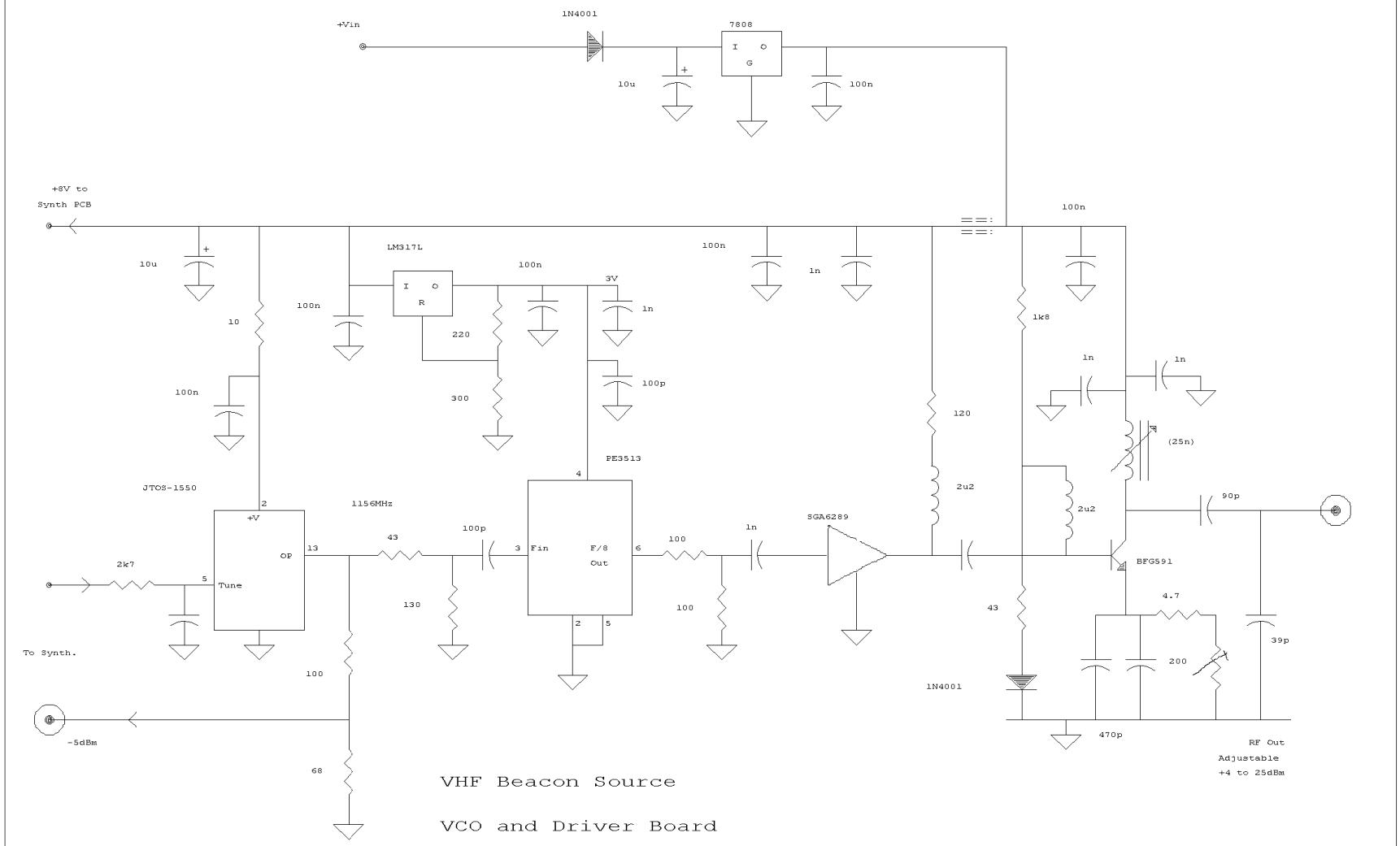
The sync tone must always be delivered to the receiver at 1270.46Hz (subject to the 5.38Hz grid setting accuracy).

For a 1500Hz tuning point, the sync tone must fall at $F_{carrier}$ - 1500 + 1270.46Hz.
For 800Hz tuning tone, the sync has to be placed at $F_{carrier}$ - 800 + 1270.46Hz.

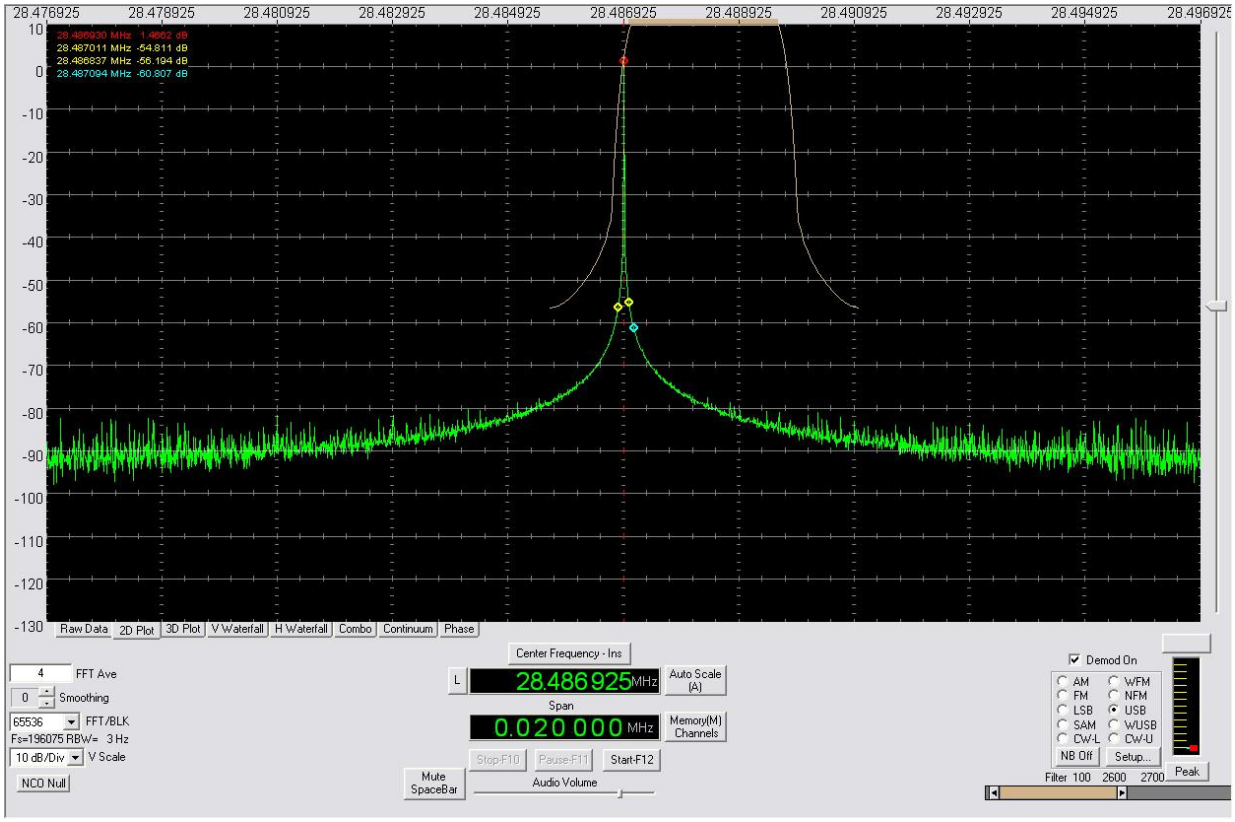
```
ConfigData     ;24 bit words sent in sequence for registers that remain unchanged
;  Fout  1155.894164MHz   Resolution  43.066406Hz
;  R =  1      D =  232199      N =  115      F =  136862
   de  0x41, 0xB0, 0x71    ;  Reg1   R + D (LSBs)
   de  0xC0, 0x07, 0x43    ;  Reg2   IF Disabled
   de  0x20, 0x00, 0x15    ;  Reg3   Charge Pump Current
   de  0xBC, 0x2B, 0xC7    ;  Reg4   Timeout / FastLock
   de  0x10, 0x1F, 0x49    ;  Reg5   IF not used
   de  0x00, 0xC4, 0x4B    ;  Reg6   Config and setup bits
   de  0x0E, 0x02, 0x1D    ;  Reg7   D(MSBs) + F (MSBs)
   de  0x00, 0x08, 0x6F    ;  Reg8   Dither settings
EndConfigData

;  Fout  1155.894164MHz   Resolution 43.066406Hz (11025/2048 * 8)
;  R =  1      D =  232199      N =  115      F =  136862
JTFreqData     ;Registers affecting freq and grid for JT65 sequence
;       de  0x0E, 0x6D, 0x3C    ;  Reg0   N + F (LSBs  JT65 sync for 1500Hz reference tone)
        de  0x0E, 0x6E, 0x40    ;  Reg0   N + F (LSBs   ditto for 800Hz reference tone)
        de  0x41, 0xB0, 0x71    ;  Reg1   R + D (LSBs)
        de  0x0E, 0x02, 0x1D    ;  Reg7   D(MSBs) + F (MSBs)
EndJTFreqData

;  Fout  1155.896MHz   Resolution 40Hz   /8 = 144.487
;  R =  1      D =  250000      N =  115      F =  147400
CarrierFreqData        ;Registers affecting freq and grid for exact freq setting
   de  0x0E, 0x7F, 0x90    ;  Reg0   N + F (LSBs)
   de  0x41, 0x09, 0x01    ;  Reg1   R + D (LSBs)
   de  0x0F, 0x42, 0x3D    ;  Reg7   D(MSBs) + F (MSBs)
EndCarrierFreqData

JT65MsgData
  include      "jt65symb.inc"     ;JT65 tones

CWMsg
        de      "GB3WGI IO64BL",0
```

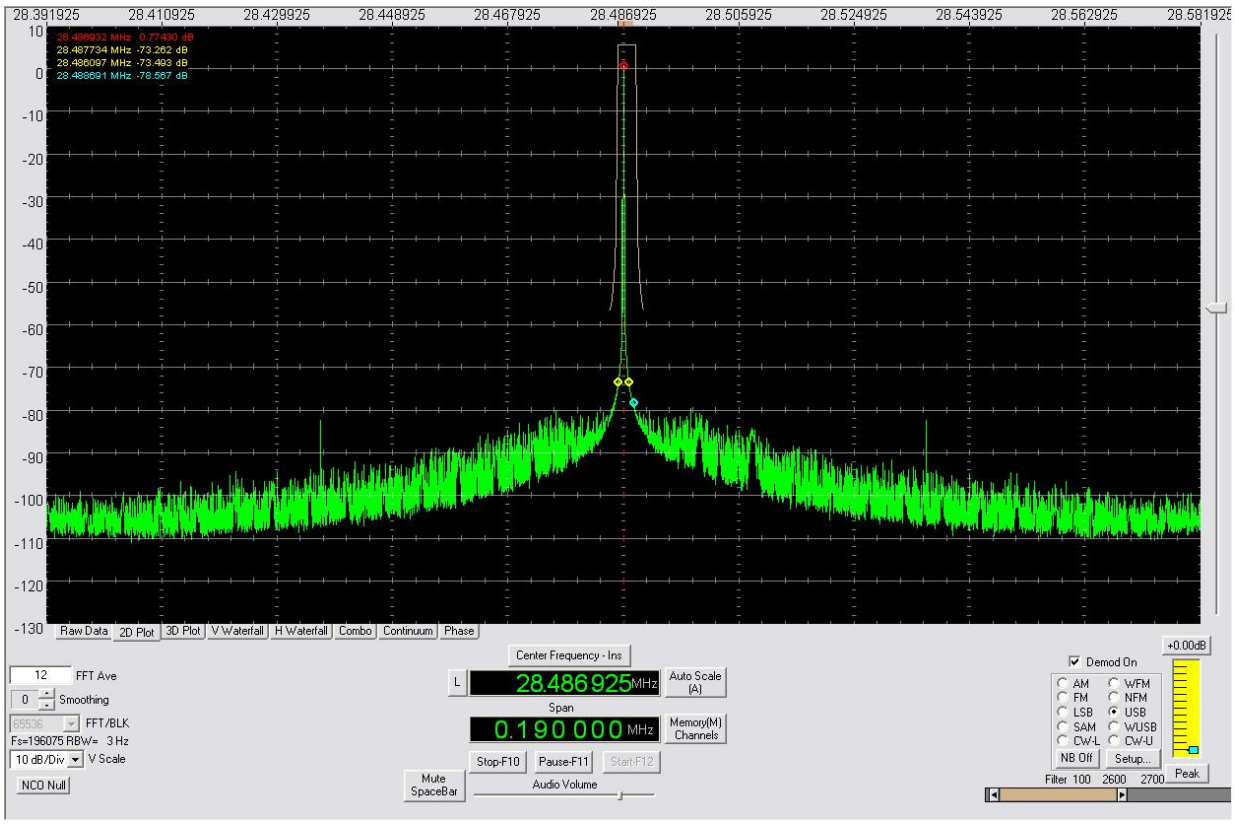

+Vin
I
O
G
+5V
2u2
10k
100n
16F628
Output to
freq gen
Data
Clock
LE
Gnd
On/Off
Key Line
43
2N7002
180
Y
G
A0
A1
A2
A3
A4
B0
B1
B2
B3
B4
B5
B6
B7
V
C
C
MCLR
CKOUT
CKIN
G
N
D
10MHz
1n
5k6
PPS
NMEA
5k6
Timing
data in
ICP
WSJT Code
Generator
G4JNT October 2012


+Vin
LM317L
I
O
R
+2.5V
C22
10u
R11
560
C13
100n
C23
10u
R5
10
R12
560
C6
R25
10
C26
100n
+2.5V
R22
100k
LM6211
R24
Vtune
to VCO
C25
C27
R21
100k
100n
R20
CPO
C5
C4
R3
C21
R23
FL
R4
C12
100n
10MHz
Ref in
100
R6
100
62
FL rf
OscIn
VddRF
CP RF
OscOut
Gnd
FL if
LMX2470
Gnd
CPif
Gnd
Gnd
RFin
IFin
RF'
Gnd
VccRF
VccIF
EN
VddIF
ENosc
Test/LD
Ck
D
LE
C3
C2
R2
56
RF in
From VCO
+2.5V
R1
10
C1
R8
10
+2.5V
R9
10
C11
C10
TP
Links on pads
bypassing original
PIC Controller
Programming
Interface
LE
Clk
Data
Gnd
R13-18
3k3 X6
VHF Beacon Source
Synthesizer Module
G4JNT October 2012

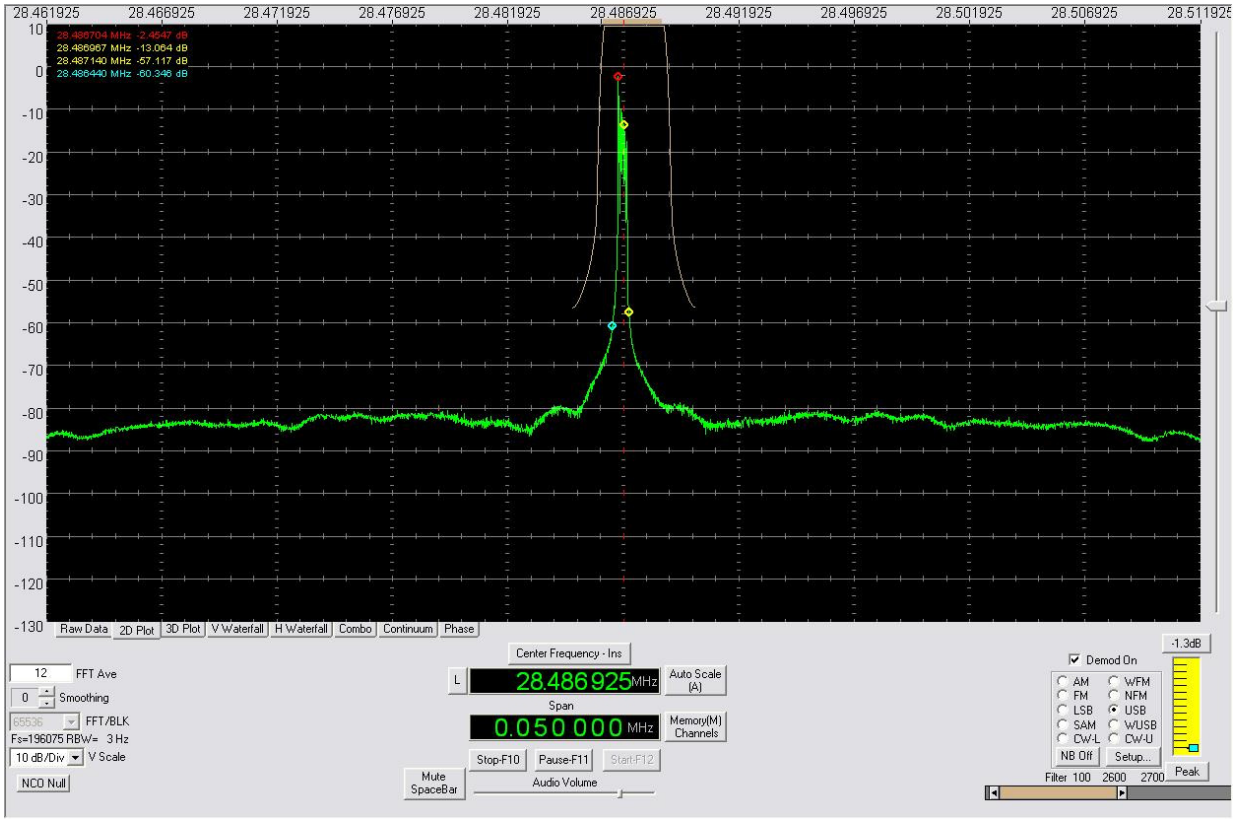

+Vin
1N4001
7808
I O G
10u
100n
+8V to Synth PCB
LM317L
I O R
3V
1n
220
300
100p
10
100n
JTOS-1550
+V
OP
Tune
2k7
To Synth.
1156MHz
43
130
100p
PE3513
Fin
F/8 Out
100
SGA6289
120
2u2
1k8
(25n)
90p
BFG591
4.7
200
39p
470p
RF Out
Adjustable
+4 to 25dBm
-5dBm
68
VHF Beacon Source
VCO and Driver Board
G4JNT October 2012

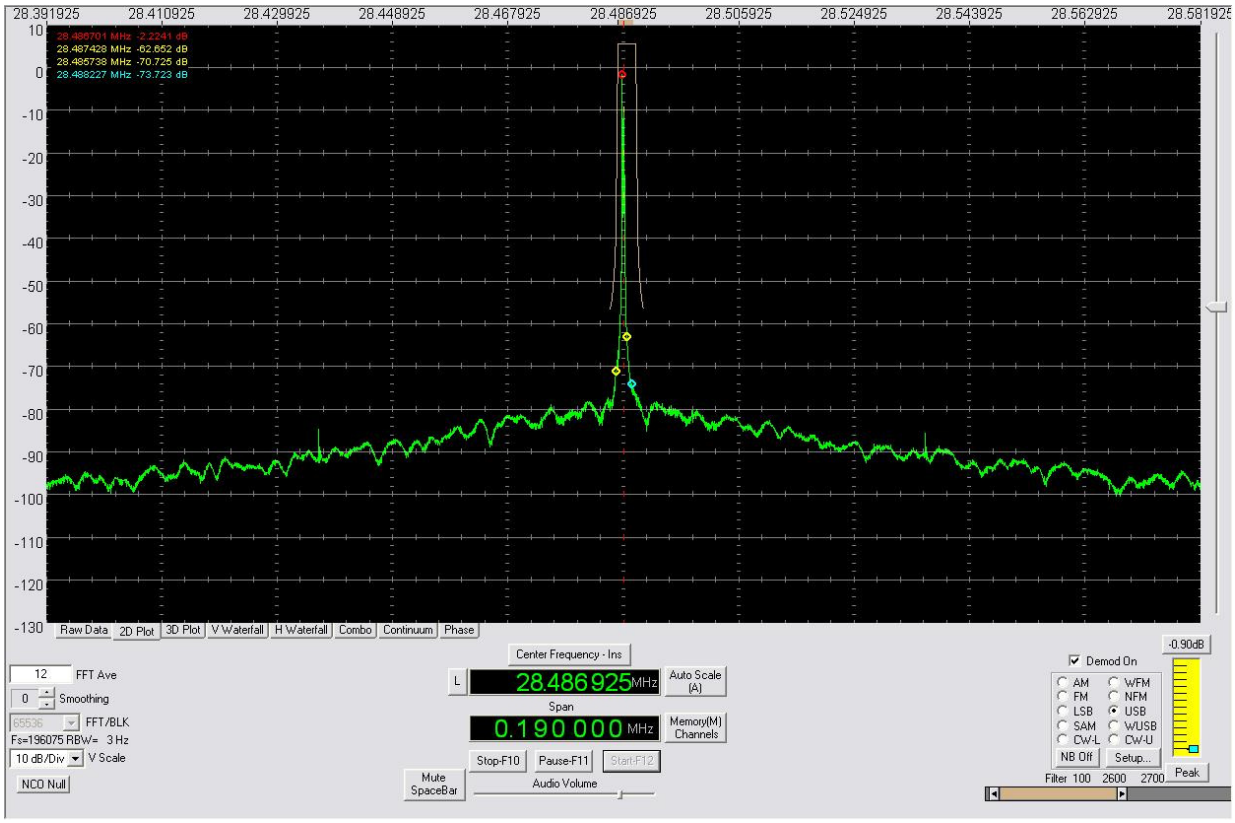
Output Spectrum Plain Carrier





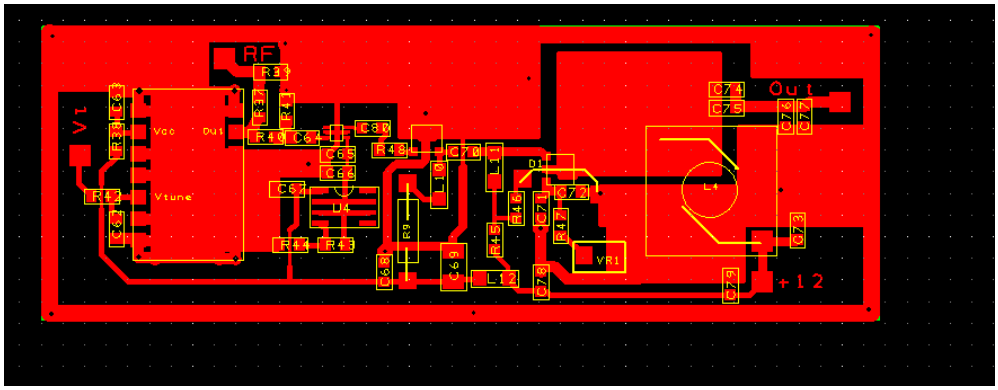
Output Spectrum During JT65 sequence





Appendix A

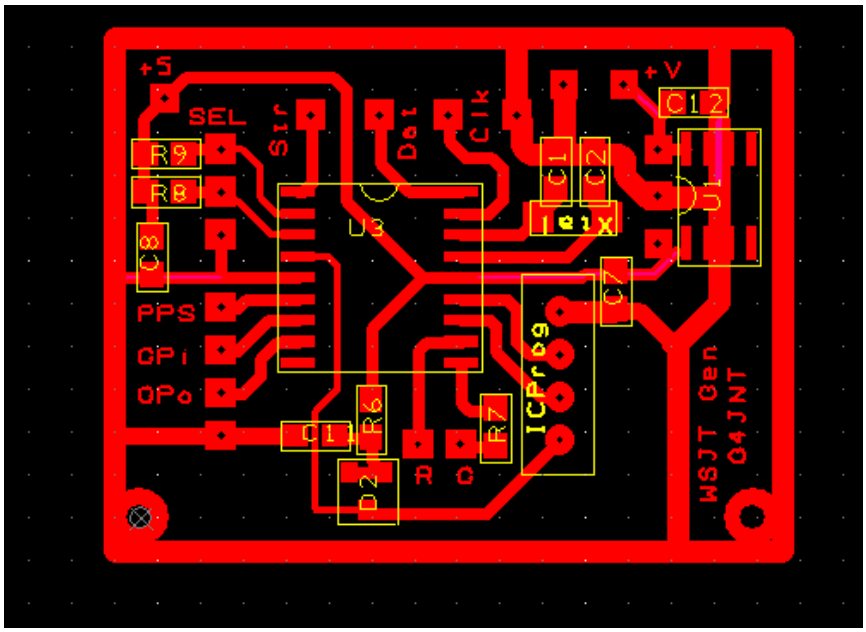
RF Source PCB Layout



Copper layouts at 1:1 Scale for direct printing to acetate or Press-N-Peel

144BcnSrc_CopperMirror1.pdf and 144BcnSrc_Copper1.pdf

PIC Code Generator PCB Layout



Copper layouts at 1:1 Scale for direct printing to acetate or Press-N-Peel

WSJTGEN_PCBMirror1.pdf and WSJTGEN_PCB1.pdf

(nb. The FET controlling the external On-Off keying is not inclued on this PCB layout)

PCB Layouts in .PDF Format http://www.g4jnt.com/144_Driver_PCBs.zip

LMX2470 Synthesizer Module