LMX2541 Fractional-N Synthesizer Development PCB

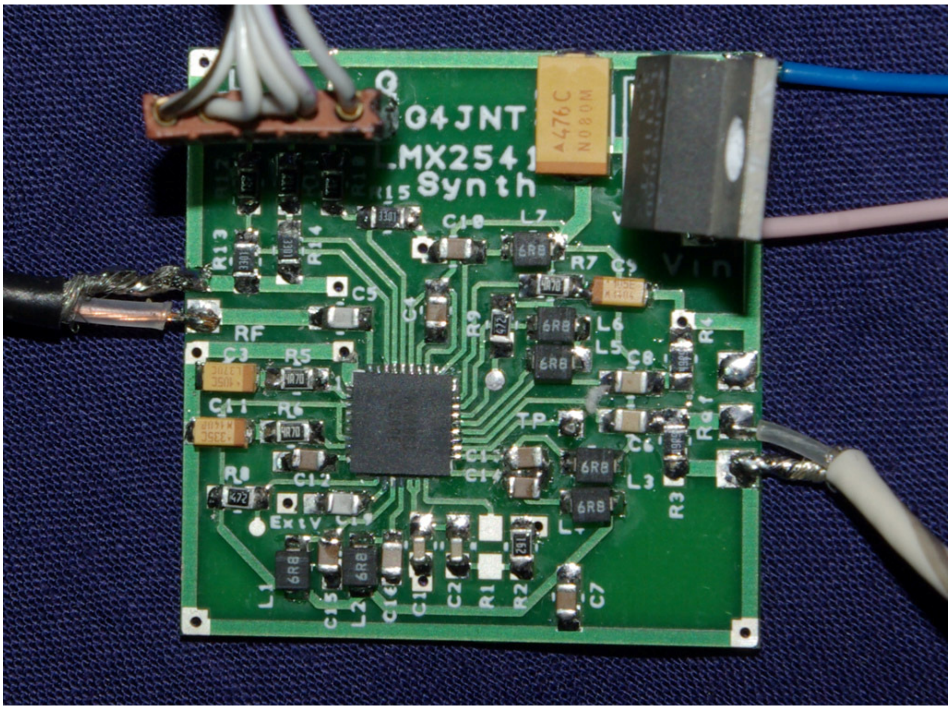
Andy Talbot G4JNT February 2013

UPDATED May 2014

Overview

With the addition of a suitable controller (See Appendix A) and reference frequency input, this PCB allows a complete Fractional-N Synthesizer operating within the frequency range of approximately 32MHz to 4000MHz to be constructed. The PCB has provision for an LMX2541 Synthesizer chip, with voltage regulator, all decoupling components, loop filter and interfacing to 5V control signals for a standard PIC based or other type of controller.

Please note, this is not in any way a kit for a complete microwave synthesizer. It is just the PCB and, optionally, details of a separate PIC interface to allow you to build your own designs using this versatile chip.



Before doing anything, obtain a copy of the LMX2541 Data Sheet from the TI website where all supporting documentation and design software can be found.

*The **Codeloader** Software, an invaluable utility for calculating the register values that need to be set, can also to be found on the website.*

http://www.ti.com/product/lmx2541

You will also need to design your loop filter. The Nat Semi. documentation has comprehensive instructions on how to do this. The spreadsheet included here, LMXFILTER.XLS implements the design equations given in several LMX series data sheets for 2nd and 3rd order filters and provides a starting point.

The LMX2541 comes in six variants with the internal VCO covering different sub-bands in the overall range 1.99 to 4.00GHz When the output divider is used, the frequencies available from each version after division begin to merge, until around 400MHz where all versions of the chip can be used for any arbitrary frequency.

The table below lists the six chip variants identified by type number suffix, with the final frequency ranges possible for output divider settings from 1 to 8, and with the highest possible divider setting of 63. Figures highlighted in red show AM-band coverage for 430MHz and higher.

Version	VCO /								
	1	2	3	4	5	6	7	8	63
Q2060	1990	995.0	663.3	497.5	398.0	331.7	284.3	248.8	31.6
	2240	1120.0	746.7	560.0	448.0	373.3	320.0	280.0	35.6
Q2380	2200	1100.0	733.3	550.0	440.0	366.7	314.3	275.0	34.9
	2530	1265.0	843.3	632.5	506.0	421.7	361.4	316.3	40.2
Q2690	2490	1245.0	830.0	622.5	498.0	415.0	355.7	311.3	39.5
	2865	1432.5	955.0	716.3	573.0	477.5	409.3	358.1	45.5
Q3030	2810	1405.0	936.7	702.5	562.0	468.3	401.4	351.3	44.6
	3230	1615.0	1076.7	807.5	646.0	538.3	461.4	403.8	51.3
Q3320	3130	1565.0	1043.3	782.5	626.0	521.7	447.1	391.3	49.7
	3600	1800.0	1200.0	900.0	720.0	600.0	514.3	450.0	57.1
Q3740	3480	1740.0	1160.0	870.0	696.0	580.0	497.1	435.0	55.2
	4000	2000.0	1333.3	1000.0	800.0	666.7	571.4	500.0	63.5

Circuit Operation

The circuit diagram is shown in Figure 1. A higher resolution version of the circuit diagram is stored as lmx2541y.gif. The reference signal is ideally supplied from an external master oscillator or low phase noise clock generator.   Reference input level is not critical, and the data sheets suggests -40dBm as a lower limit; the maximum should not exceed 3.3V peak. The 51Ω termination resistor, R3 can be increased if loading is too high for a TCXO. Use any value you feel comfortable with, or leave it out altogether

The charge pump output is filtered in C1, C2 and R2 . Programmable on- chip components allow for up to a fourth order filter – see the data sheet for more details'

PCB Construction

The PCB layout is shown in Figure 2, component references should tie-in with those in Figure 1. Component values not shown will depend on the design frequency and the PLL filter. Coupling and decoupling capacitors around the chip are 100nF – decoupling for high frequencies is internal to the chip   Other decoupling , including those for internal regulators should use the values shown.

The only component that may give any difficulty in installation is the LM2541 itself. This should be fitted first, before any other components are added.   The chip has 0.5mm pad spacing and is of the type where the connections are on the edge of the ceramic package, with no gull-wing tabs.   The package is rectangular, with three pads each top and bottom, and nine on each side.   A marker shows Pin 1 at the top left.

This is how I solder the LMX2541 device.   Use a decent high magnification viewer; at least a headband type or for even greater magnification an old 50mm SLR camera lens is ideal.   The pads on the PCB for the chip should first be tinned, aiming to keep an equal thickness of solder on each pad as shown in Photo 1. Then the pads should be liberally coated with flux.   Double, triple and quadruple check k the IC alignment, place it central over the footprint so an equal amount of pad is showing on each side. Make sure there is flux still there on each pad. Press the IC down firmly, checking again for alignment. Peer though a magnifier at the IC pads sitting over the correct pads on the PCB while still pressing down on the IC. Look away for a few seconds to relax your eyes, return and check alignment again.

Once absolutely certain each IC connection is above its pad, and unmoved, and still keeping it pressed down, using a medium-sized (typically 1mm) chisel-end soldering-iron bit place it in turn on each pad close to the IC, while watching the solder melt under the IC connections. The flux will cause a sort-of wicking action, drawing molten solder up from the pre-tinning to fill the gap to fix at least some of IC connections. Gently prod the chip to make sure it is now fixed by at least some pads and won't move of its own accord. Now apply more flux to the IC / PCB and load the soldering iron tip with solder. Wipe the iron against each side of the chip in turn, gently pushing solder into the gaps. Providing there is enough flux present, it should nicely flow into the minute gaps between each IC connection and its PCB pad. The flux – somehow, if there's enough of it – always seems to prevent unwanted bridges forming. These are more likely if there is not enough flux, or the solder on the bit has been allowed to get dirty and oxidised.

Once the chip appears to be fitted, breath a sigh of relief and get out the high gain magnifier. Look carefully at each joint making sure there is a continuous solder flow from PCB pad to IC connection. And also that there are no bridges. Bridges can be removed using solder wick / braid, again making use of extra flux at all times to aid solder flow and cleaning.

When all the pads are soldered, the next stage is soldering the grounding pad on the underside. It is important that the IC has no chance of moving should the whole chip end up at solder melting point (which won't harm it, provided this temperature is not sustained for more than a second or two). The best way to ensure it doesn't move is to gently clamp the PCB upside down onto a non heat conducting surface. Paper clips holding the PCB onto a piece of cardboard or thin plywood works well. This ensures the IC is kept pressed onto its pads while the underside is soldered. Dd not apply too much force so the PCB bends! Apply more flux to the underside, running it into the plated through holes. Use a larger soldering iron bit, as quickly as possible apply solder until it flows though and is sucked down the 9 holes on the underside. When molten solder appears in a majority of holes stop heating and applying solder, and that should be it.

If you have the right sort of a diode tester facility on a DVM, the connections to each pad of the IC can now be verified. The tester must have a voltage on the probes of no more than 2 – 3V open circuit, and no more than a couple of 100µA test current. If it meets these criteria connect the negative lead to ground and probe each IC connection in turn. All will show a voltage of between 0.7 to 1.8V. Any that Indicate the full Open circuit voltage are almost certainly not connected. I can't vouch this test will work for other chips or different manufacturer's products, but it does for this one. The tester must be a diode / continuity test with a constant current / voltage reading. Using the ohms range is not nearly so helpful.

After fitting the main chip, installing the rest of the components should be straightforward!
Apart from the large electrolytics around the LM1117, the majority of components can be of 0805 or 0603 size. The pads for C12 to C16 and C19 are a bit tight for 0805 components, but these can be squeezed on with care. The inductors can be 1206 size

The PCB has been sized to fit inside a standard 37mm tinplate box, with the RF output provided by a socket soldered to the side in the conventional way for hard-cabled RF solutions.

All power supply and signal pins are provided with through board connections, so these can be made on either side.

Components List

Quantity	Reference	Part
1	IC1	LM1117 - 3.3
1	IC2	LMX2541 (choose frequency variant)
2	R3,R4	51 or higher to suit reference
2	R9,R8	4k7
3	R12,R10,R11	1k6
3	R13,R14,R15	3k3
1	C18	22u Rated for V_{IN} maximum
1	C17	47u > 6V wkg.
3	R5,R6,R7	4.7
2	C3,C9	1u
1	C11	4u7
8	C4,C7,C10,C12	
	C13,C14,C15,C16	100nF
4	C5,C6,C8,C19	Typ. 470pF– 10nF to suit Freq.
7	L1 to L7	Typ. 1u to 10uH
3	C1, C2, R2	Loop Filter, custom values
[1]	R1	FastLock (if used)

Power Supplies

The Synthesizer chip needs 3.3 V_{DD} supplied by the on-board regulator; current consumption is approximately 100mA. Input voltage, V_{IN}, can range from 4.5 to 20V. A heatsink is not necessary on the regulator for input voltages in the lower range.

Resistors R10 – R15 allow the chip to be controlled with 5V logic levels. This allows a PIC to be used with a V_{DD} suitable for allowing in-circuit programming. Vin to the regulator can be conveniently made equal to this 5V, which is then supplied as one from a second external regulator.

Programming and Getting it Going

The chip's internal registers have to be programmed to the correct values for your desired frequency and fractional grid, and the loop gain set, and etc etc etc . *The data sheet and an understanding of Fractional-N synthesizers is essential* here, but the easiest way to start off is to use National's ***CodeLoader*** software to work out the values for you. This can be downloaded from the website shown above.

CodeLoader shows a graphical representation of the RF PLL where you can interactively set frequencies and division ratios, as well as all the other PLL settings.

.

Also select whatever you want from the TP test point on pin 20. Setting this to deliver the divided down N or R dividers is instructive - it actually supplies a square wave at half the divided down or reference value and can make a quick confidence check of correct register programming before a complete closed loop system is established.

Look at the waveform from here on a scope and experiment with programming the R3 register to toggle between divided down R and N values.

Otherwise, the default registers programmed into the PIC will work as a starting point, but read the data sheet before modifying registers at will. For most operations, only R0, R1 and R2 need to be set to give a particular frequency, grid and reference. The other registers can be left alone.

The ***LMX2541_PROG.EXE*** programme include in the archive allows quick calculation of the registers for specified values of Fosc, Fout and frequency tuning grid. It can be used in conjunction with a PIC containing the ***LMX2541CTL*** operating system as described in the appendix.

PLL Filter Design

The design of PLL filters for Fractional-N synthesizers is no different in concept from that for Integer-N designs, apart from the fact that the reference frequency is usually (a lot) higher with corresponding reduction in the integer part of N, and usually a higher loop bandwidth. The spreadsheet ***LMXFILTER.XLS*** (included in **http://www.g4jnt.com/lmx2541Support.zip**) as shown in Figure 3 lets the basic values for the three main components forming the first two poles and the optional RC pair for a third pole be calculated. Enter values the figures highlighted in blue.

The TI website also has loop filter design utilities **http://www.ti.com/tool/clockdesigntool**

R1 provides for Fastlock operation if desired. See the data sheet for how this is implemented. For most applications it can be left out.

As an example, and to get going quickly, try these values for a starting point:

K_{VCO}	30MHz/V
BW	30kHz
F_{REF}	10MHz (= F_{COMP})
R	1
I_{CP}	1mA (Mid-Range)
C1	680pF
C2	15nF
R2	1k6
R1	Fastlock, Not installed

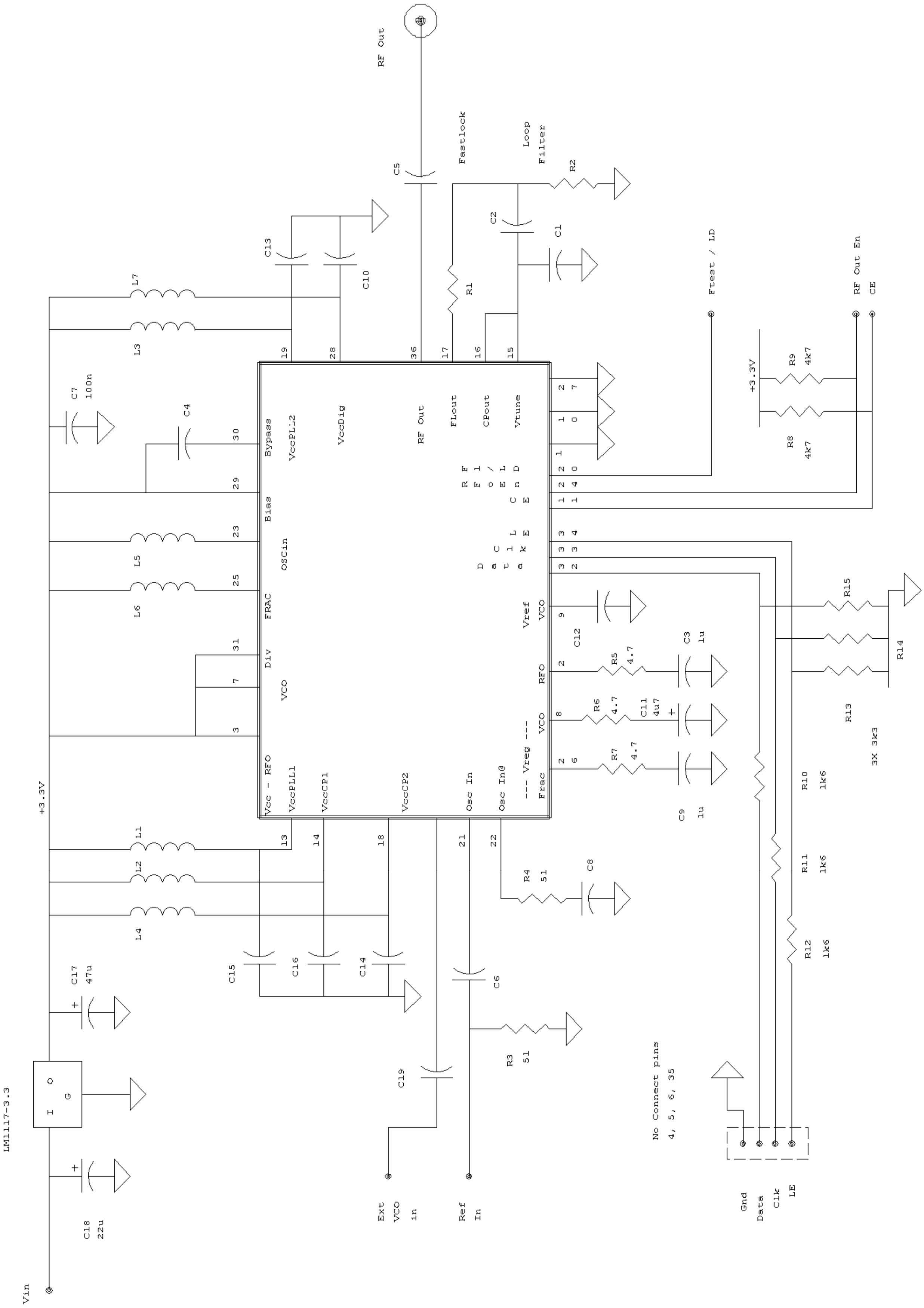
VCO Tuning Sensitivity, K_{VCO}, extracted from Data Sheet

2060E	13 – 23 MHz / V
2380E	16 - 30
2690E	17 - 32
3030E	20 - 37
3320E	21 - 37
3740E	24 – 42

The lower number in the range applies when the VCO is at its lowest frequency and the higher number applies when the VCO is at its highest frequency. A linear approximation can be used for frequencies between these two cases.

Contents of LMX2541Support.zip

LMX2541_PROG.EXE	(Register values, and control via RS232 / PIC O/S)
LMX2541CTL.ASM	(PIC firmware for RS232 control)
LMX2541CTL .HEX	“ “ “
LMXFILTER.XLS	
LMX2541Y.gif	(Higher resolution circuit diagram)
LMX2541_Built.jpg	


RF Out
C5
Fastlock
Loop
Filter
R2
C2
C1
R1
C13
C10
L7
L3
Ftest / LD
RF Out En
CE
+3.3V
R9
4k7
R8
4k7
C7
100n
C4
19
28
36
17
16
15
Bypass
VccPLL2
VccDig
RF Out
FLout
CPout
Vtune
30
29
Bias
23
OSCin
25
FRAC
31
Div
7
VCO
3
2
7
1
0
1
2 0
2 4
1 1
RF
F1
o/
EL
nD
C
E
3 4
3 3
3 2
D
a C
t l L
a k E
Vref
VCO
9
C12
RFO
2
R5
4.7
C3
1u
VCO
8
R6
4.7
C11
4u7
--- Vreg ---
Frac
2
6
R7
4.7
C9
1u
R15
R14
R13
3X 3k3
R10
1k6
R11
1k6
R12
1k6
Vcc - RFO
VccPLL1
VccCP1
VccCP2
Osc In
Osc In@
13
14
18
21
22
R4
51
C8
L5
L6
+3.3V
L1
L2
L4
C15
C16
C14
C17
47u
C6
R3
51
C19
LM1117-3.3
I
O
G
C18
22u
Vin
Ext
VCO
in
Ref
In
No Connect pins
4, 5, 6, 35
Gnd
Data
Clk
LE

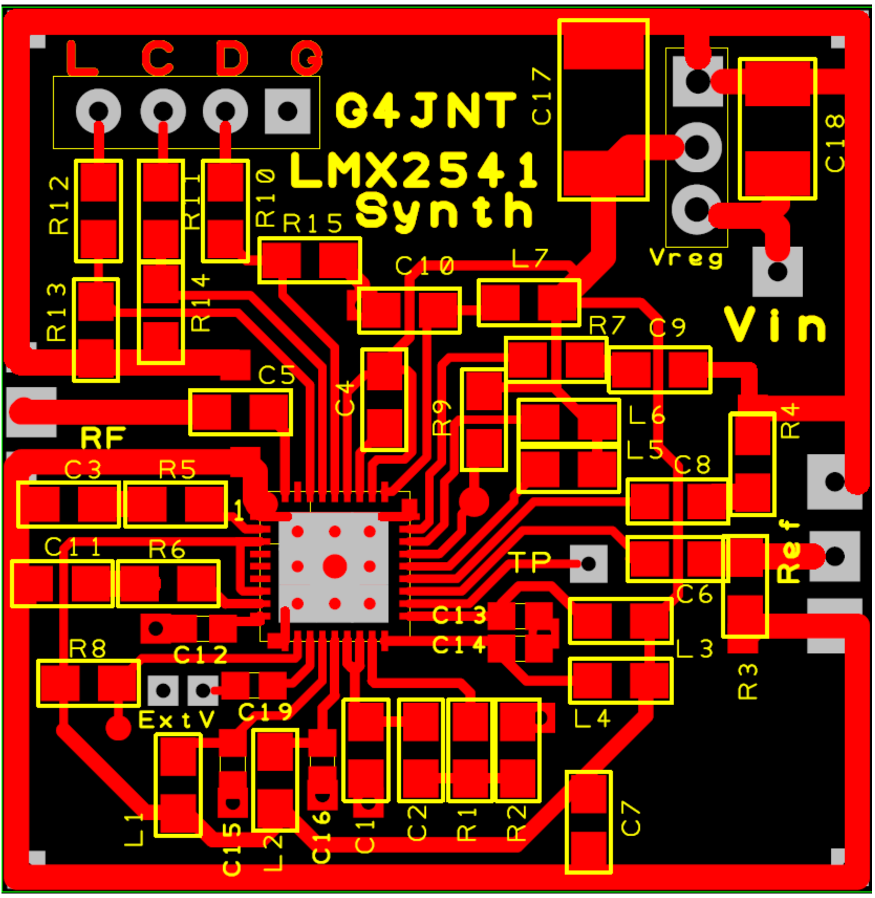
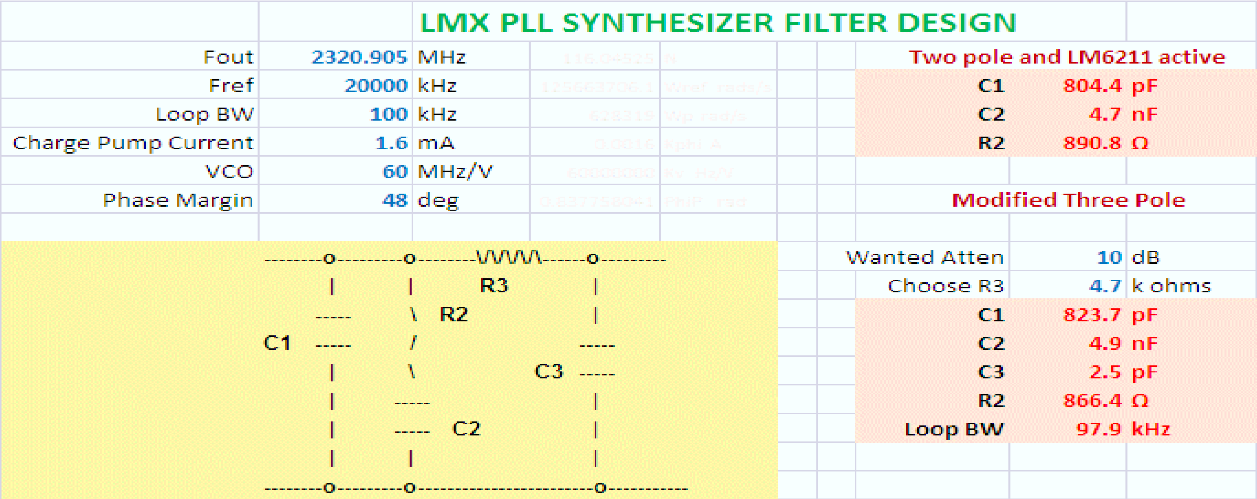


Fig 2 PCB Component Placing



LMX PLL SYNTHESIZER FILTER DESIGN

Fout	2320.905	MHz	116.04525	N	
Fref	20000	kHz	125663706.1	Wref rads/s	
Loop BW	100	kHz	628319	Wp rad/s	
Charge Pump Current	1.6	mA	0.0016	Kphi A	
VCO	60	MHz/V	60000000	Kv Hz/V	
Phase Margin	48	deg	0.837758041	PhiP rad	

Two pole and LM6211 active		
C1	804.4	pF
C2	4.7	nF
R2	890.8	Ω

Modified Three Pole		
Wanted Atten	10	dB
Choose R3	4.7	k ohms
C1	823.7	pF
C2	4.9	nF
C3	2.5	pF
R2	866.4	Ω
Loop BW	97.9	kHz

```
--------o---------o--------VVVV\------o---------
        |         |        R3         |
      -----       \   R2              |
C1    -----       /                 -----
        |         \           C3    -----
        |       -----                 |
        |       -----    C2           |
        |         |                   |
--------o---------o-------------------o-----------
```

Figure 4 Screen capture from the PLL Filter design spreadsheet File - *LMXFILTER.XLS*

Appendix A

LMX2541CTL PIC Operating System

A 12F629 PIC containing the ***LMX2541CTL*** code generates SPI data to allow the registers in the synthesizer chip to be set directly to the wanted values using a serial RS232, or COM Port interface with simple ASCII text commands from any terminal emulator programme such as Hyperterm. The register values can be programmed into non-volatile EEPROM memory in the PIC. At turn on, these stored values are recalled and sent to the synthesizer for immediate start up.

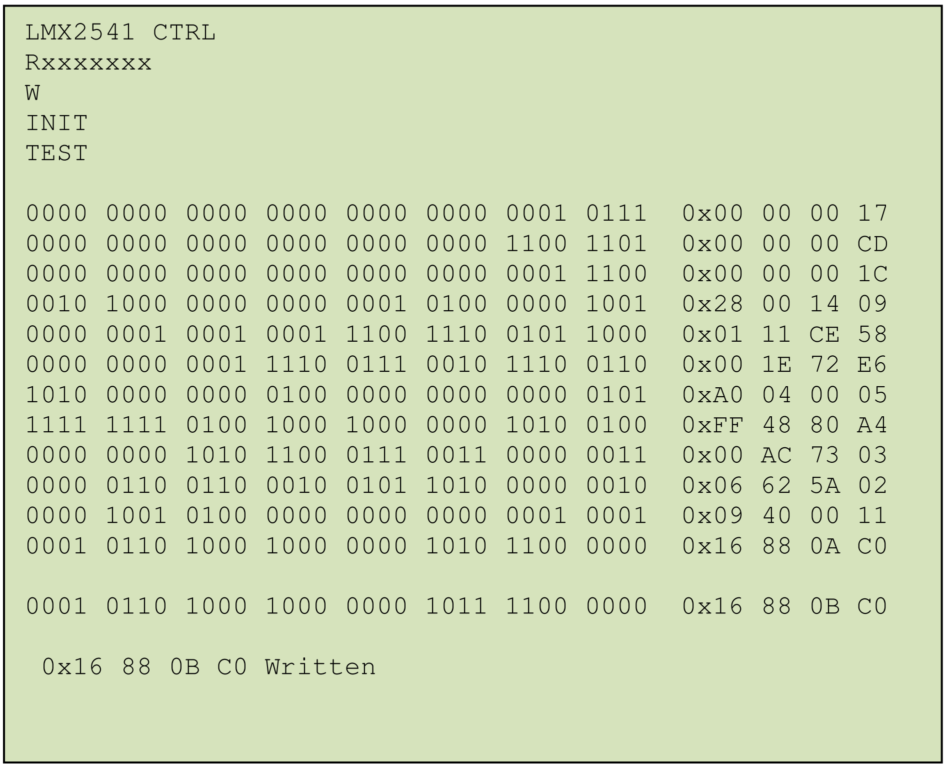
Connect the RS232 lead and SPI connections as shown in Figure A1. The 4.7k resistor on the RS232 interface serves to limit the current driven into the PIC input pin from the typical +/-9V of most RS232 interfaces. Set the RS232 settings to 9600 baud, 8 bits, 2 stop bits, no parity, 9600-N-8-2. Flow control should be set OFF. Turn on or reset the processor module, whereupon a display should appear, similar to that shown below. This shows a summary of allowed commands, followed by the stored register contents in both binary and hex that are immediately loaded into the chip.

Registers are updated one at a time by typing R followed by eight digit hexadecimal representation of their new contents, terminated with a [cr].

Example:

R16880AC0[cr] *(Note that the 0x shown on the response must not be typed)*.

Followed by W [cr]



```
LMX2541 CTRL
Rxxxxxxx
W
INIT
TEST

0000 0000 0000 0000 0000 0000 0001 0111  0x00 00 00 17
0000 0000 0000 0000 0000 0000 1100 1101  0x00 00 00 CD
0000 0000 0000 0000 0000 0000 0001 1100  0x00 00 00 1C
0010 1000 0000 0000 0001 0100 0000 1001  0x28 00 14 09
0000 0001 0001 0001 1100 1110 0101 1000  0x01 11 CE 58
0000 0000 0001 1110 0111 0010 1110 0110  0x00 1E 72 E6
1010 0000 0000 0100 0000 0000 0000 0101  0xA0 04 00 05
1111 1111 0100 1000 1000 0000 1010 0100  0xFF 48 80 A4
0000 0000 1010 1100 0111 0011 0000 0011  0x00 AC 73 03
0000 0110 0110 0010 0101 1010 0000 0010  0x06 62 5A 02
0000 1001 0100 0000 0000 0000 0001 0001  0x09 40 00 11
0001 0110 1000 1000 0000 1010 1100 0000  0x16 88 0A C0

0001 0110 1000 1000 0000 1011 1100 0000  0x16 88 0B C0

 0x16 88 0B C0 Written
```

The final LS digit of the eight defines the register address – see the data sheet for full details of register addressing. The new value is immediately transmitted to the synthesizer chip, which should straightaway respond accordingly.

To save this new register value to non-volatile memory in the PIC, type W[cr] after the register contents have been accepted. A response will show that individual register has been written to EEPROM.

To store multiple registers , enter these one at a time, issuing a W command for each. This is not as tedious as it sounds as subsequent updates will quite likely only require that one or two registers be altered at a time.

If you get in a mess, type INIT[cr] which has the same effect as switching off and back on; reloading all the registers with the prestored values.   When updating the EEPROM with several new values, it is worth issuing INIT commands periodically to check these have all been stored correctly.

The *LMX2541_Prog* Software (Figure A2) can send the ASCII text automatically to the controller, To enable this part, simply set the COM Port window to the one desired.

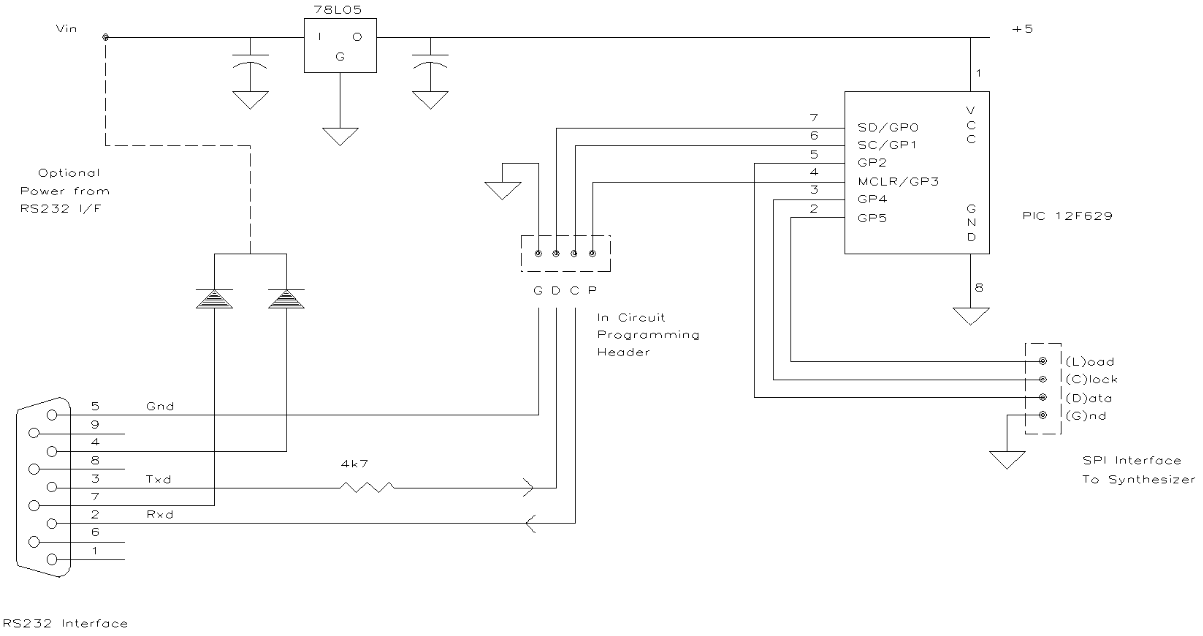
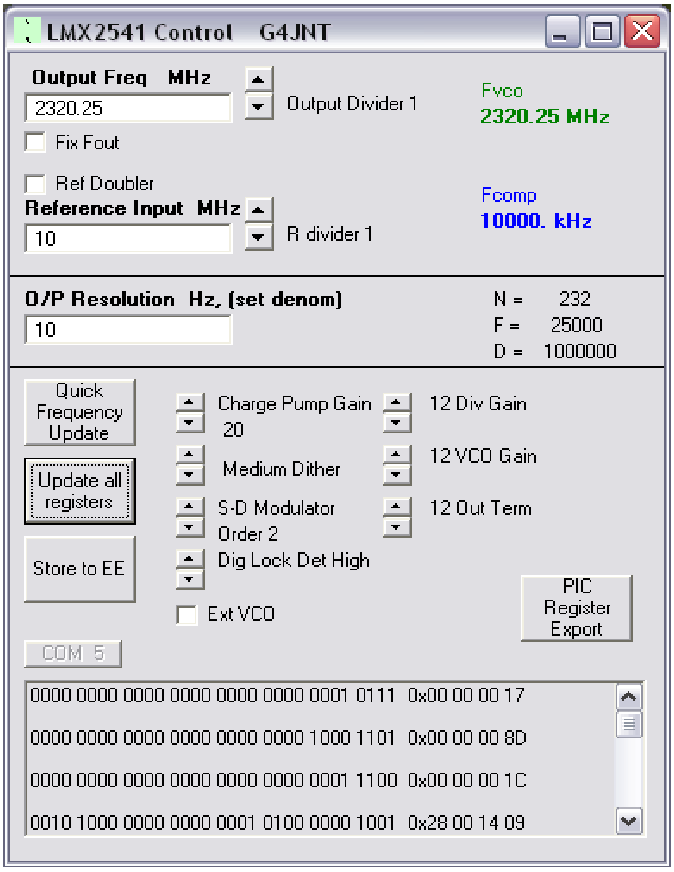


Figure A1 PIC Interface for *LMX2541CTL* Operating System



Output Freq MHz: 2320.25

Output Divider 1

Fvco 2320.25 MHz

□ Fix Fout

□ Ref Doubler

Reference Input MHz: 10

R divider 1

Fcomp 10000. kHz

O/P Resolution Hz, (set denom): 10

N = 232
F = 25000
D = 1000000

Charge Pump Gain 20

12 Div Gain

Medium Dither

12 VCO Gain

S-D Modulator Order 2

12 Out Term

Dig Lock Det High

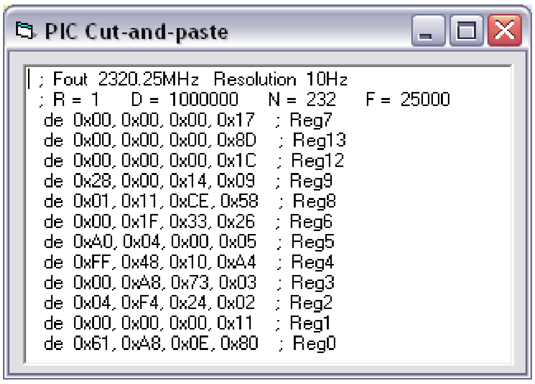
□ Ext VCO

```
0000 0000 0000 0000 0000 0000 0001 0111   0x00 00 00 17

0000 0000 0000 0000 0000 0000 1000 1101   0x00 00 00 8D

0000 0000 0000 0000 0000 0000 0001 1100   0x00 00 00 1C

0010 1000 0000 0000 0001 0100 0000 1001   0x28 00 14 09
```



```
; Fout  2320.25MHz  Resolution  10Hz
; R = 1      D = 1000000     N = 232     F = 25000
 de  0x00, 0x00, 0x00, 0x17   ; Reg7
 de  0x00, 0x00, 0x00, 0x8D   ; Reg13
 de  0x00, 0x00, 0x00, 0x1C   ; Reg12
 de  0x28, 0x00, 0x14, 0x09   ; Reg9
 de  0x01, 0x11, 0xCE, 0x58   ; Reg8
 de  0x00, 0x1F, 0x33, 0x26   ; Reg6
 de  0xA0, 0x04, 0x00, 0x05   ; Reg5
 de  0xFF, 0x48, 0x10, 0xA4   ; Reg4
 de  0x00, 0xA8, 0x73, 0x03   ; Reg3
 de  0x04, 0xF4, 0x24, 0x02   ; Reg2
 de  0x00, 0x00, 0x00, 0x11   ; Reg1
 de  0x61, 0xA8, 0x0E, 0x80   ; Reg0
```

Fig A2 ***LMX2541_Prog*** **User Screen**