

LTC6946 Synthesizer Development PCB

Andy Talbot G4JNT June 2013 (Updated Feb 2015)

Overview

With the addition of a suitable reference frequency input, this PCB allows a complete Integer-N Synthesizer operating within the frequency range of approximately 370MHz to 6400MHz to be constructed. The PCB has provision for an LTC6946 Synthesizer chip, with voltage regulators, all decoupling components, loop filter and a PIC microcontroller for RS232 ASCII Hex text based control.

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

Before doing anything, obtain a copy of the LTC6946 Data Sheet

<http://cds.linear.com/docs/en/datasheet/6946fa.pdf>

The LTC6946 comes in four variants with the internal VCO covering different sub-bands in the overall range 2.2 to 6.4GHz. The table below lists the four chip variants identified by type number suffix, with the final frequency ranges possible for the output divider settings from 1 to 6

Version	VCO /					
	1	2	3	4	5	6
LTC6946-1	2240	1120	747	560	448	373
	3740	1870	1247	935	748	623
LTC6946-2	3080	1540	1027	770	616	513
	4910	2455	1637	1228	982	818
LTC6946-3	3840	1920	1280	960	768	640
	5790	2895	1930	1448	1158	965
LTC6946-4	4200	2100	1400	1050	840	700
	6390	3195	2130	1597	1278	1065

A test on the-3 variant showed that lock could be maintained over the range 3650 to 6300MHz. (Operation much beyond the manufacturer's quoted limits should not be assumed to be always possible).

Circuit Operation

The circuit diagram is shown in Figure 1. A higher resolution version of the circuit diagram is stored as LTC6946.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 sheet quotes values between 0.5 to 3.3V peak-to-peak, corresponding to -2 to +14dBm in the 51Ω termination resistor R4. This can be increased if loading is too high for a TCXO. Use any value you feel comfortable with, or leave it out altogether

You will also need to design your loop filter, made up of components C2, C3 and R1. The data sheet includes design information for loop filter design but The Nat Semi. / TI documentation and tools for their LMX family of synthesizers can also be used (loop filter design uses the same techniques, whoever makes the chips) 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.

Two RF Outputs are provided. One comes direct from the chip, the other passes to an ERA-2 modamp. Either may be used, or the modamp left out altogether, or jumpered, but any unused output should be terminated with a 51Ω resistor shown as Rx on the diagram. Alternatively, both RF output ports directly from the chip can drive a transformer differentially. See the data sheet for more details,

PCB Construction

Components List

Quantity	Reference	Part	
1	U1	PIC Controller	12F629
1	U2	Synthesizer	LTC6946 (-1, -2 or -3)
1	U3	Modamp	ERA-2
1	U4	3.3V Regulator	LM1117M-3.3
1	U5	5V Reg. (SOT23)	LM3480
1	R2		15
2	R3,R12		0
1	R4	50 – 300	Ref. In Termination.
1	R5		36
3	R6, R7, R8		1k5
3	R9, R10, R11		2k7
1	R13	20k	PGM Pin Pullup (4 freq only)
1	Rx	51	Unused RF Port
1	C1		2u2
4	C4, C5, C7, C14		10n
2	C6, C13		1u
2	C8, C9		470p
7	C10, C12, C17, C19, C20, C21, C24		100n
5	C11, C15,C16,C23,C25		56p - 330p
2	C18, C22		33u (1812 size)
3	L1, L2, L3	100nH – 1uH	(1206 size)
3	R1, C2, C3	Loop Filter	(start with 1k3, 1n8, 10n)

R, C L components 0805 size unless stated otherwise

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 should be approximately the values shown – decoupling for high frequencies is internal to the chip.

The only component that may give any difficulty in installation is the LTC6946 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 six pads each top and bottom, and eight on each side. A marker shows Pin 1 at the top left.

This is how I solder the LTC6946 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 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 through 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.

When 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. Do 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 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 inductors can be 1206 size.

The PCB has been sized to fit inside a standard 37 x 54mm tinplate box, with the RF outputs 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.

Power Supplies

The Synthesizer chip needs both 5V and 3.3 V supplies delivered by the on-board regulators; current consumption is approximately 160mA Input voltage, V_{IN} , can range from 7 to 20V.

Resistors R6 – R11 enable the controller to run with a V_{DD} of 5V which allows in-circuit programming of the PIC to be performed.

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 . The data sheet is essential in understanding all the register contents and the order they have to be set, but the easiest way to start off is to use **LTC6946_Prog.exe** to work out the values for you.

In order to use the serial interface facility in *LTC6946_Prog.exe* you will need to have the appropriate VB6 run time libraries installed on your computer. If you get an associated error message when trying to use the COM port from *LTC6946_Prog.exe*, run the SETUP included in the support package to install the run-time libraries.

This programme included in the archive allows quick calculation of the registers for specified values of F_{osc} , F_{out} and F_{comp} . It can be used in conjunction with a PIC containing the **LTC6946CTL** operating system to send the ASCII Hex values to the chip and save to EEPROM.

PLL Filter Design

A spreadsheet for design of PLL filters for synthesizers **LMXFILTER.XLS** (included in <http://www.g4jnt.com/LTC6946Support.zip>) shown in Figure 3 allows the basic values for the three main components forming the first two poles and the optional RC pair for a third pole to be calculated. Enter values for the figures highlighted in blue.

The TI website also has loop filter design utilities <http://www.ti.com/tool/clockdesigntool>
Details of how to design loop filters are also given in the data sheet for the LTC6946

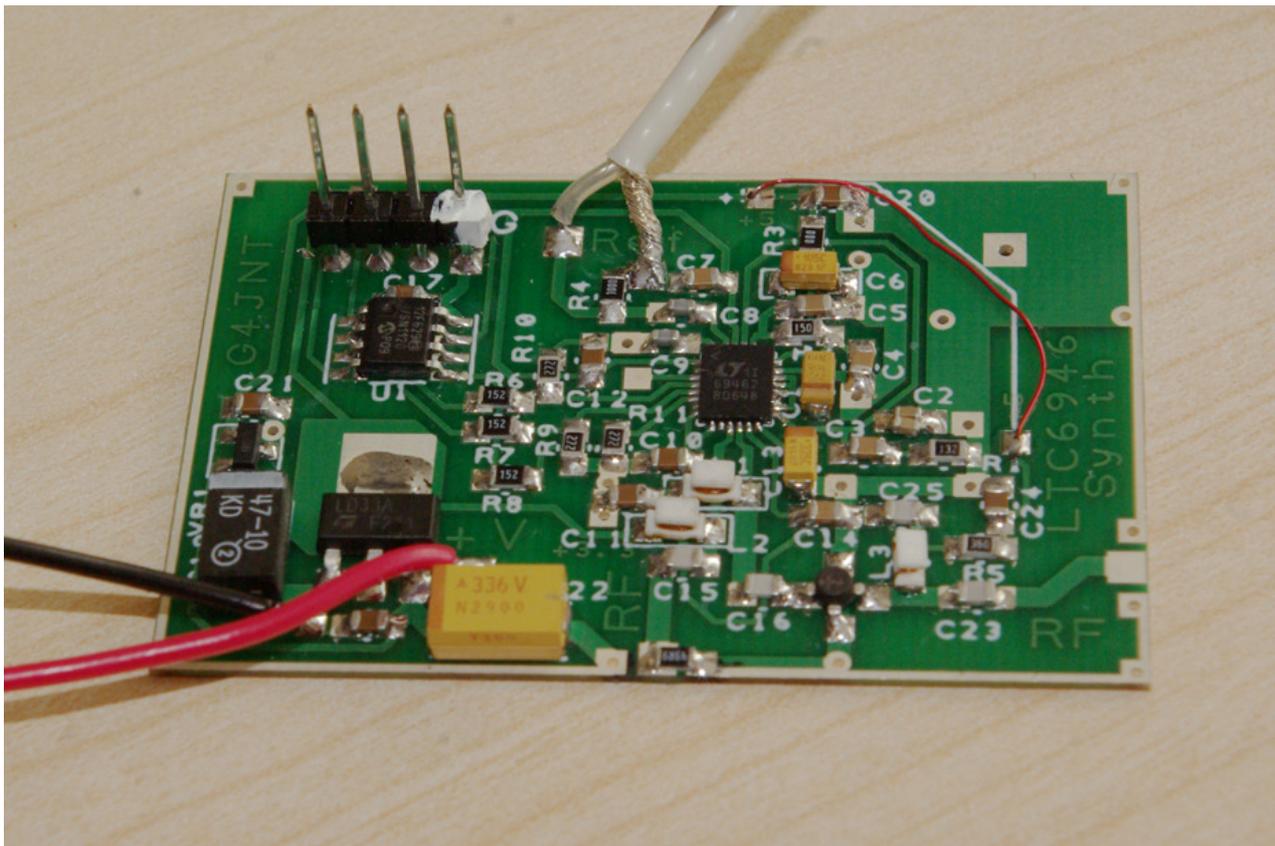
VCO Tuning Sensitivity, K_{VCO} shown in the data sheet gives a tuning sensitivity expressed in %Hz/Volt. Typical values range from 4.7 to 7.2%/V. Assuming a mid range value, of 5.7%/V, for the -2 variant operating at 3.4GHz, this gives an approximate tuning rate of 200MHz/V. A more precise value for any particular frequency can be extracted from then graphs shown there.

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

C2 = 1.8nF
C3 = 10nF
R1 = 1k3

Contents of LTC6946Support.zip www.g4jnt.com/

LTC6946.gif	Circuit Diagram
LTC6946_PCB.jpg	PCB Layout
LTC6946-CTL.asm	PIC RS232 Operating System. Source code
LTC6946-CTL.hex	PIC RS232 Operating System. Object code
LTC6946_Switched.asm	Four Frequency switch selectable. Contol. Source code
LTC6946_Switched.hex	Four Frequency switch selectable. Contol. Object code
LTC6946_Prog.exe	PC / Windows utility, optionally interfacing with LTC6946-CTL
LMXFILTER.XLS	Loop Filter design spreadsheet



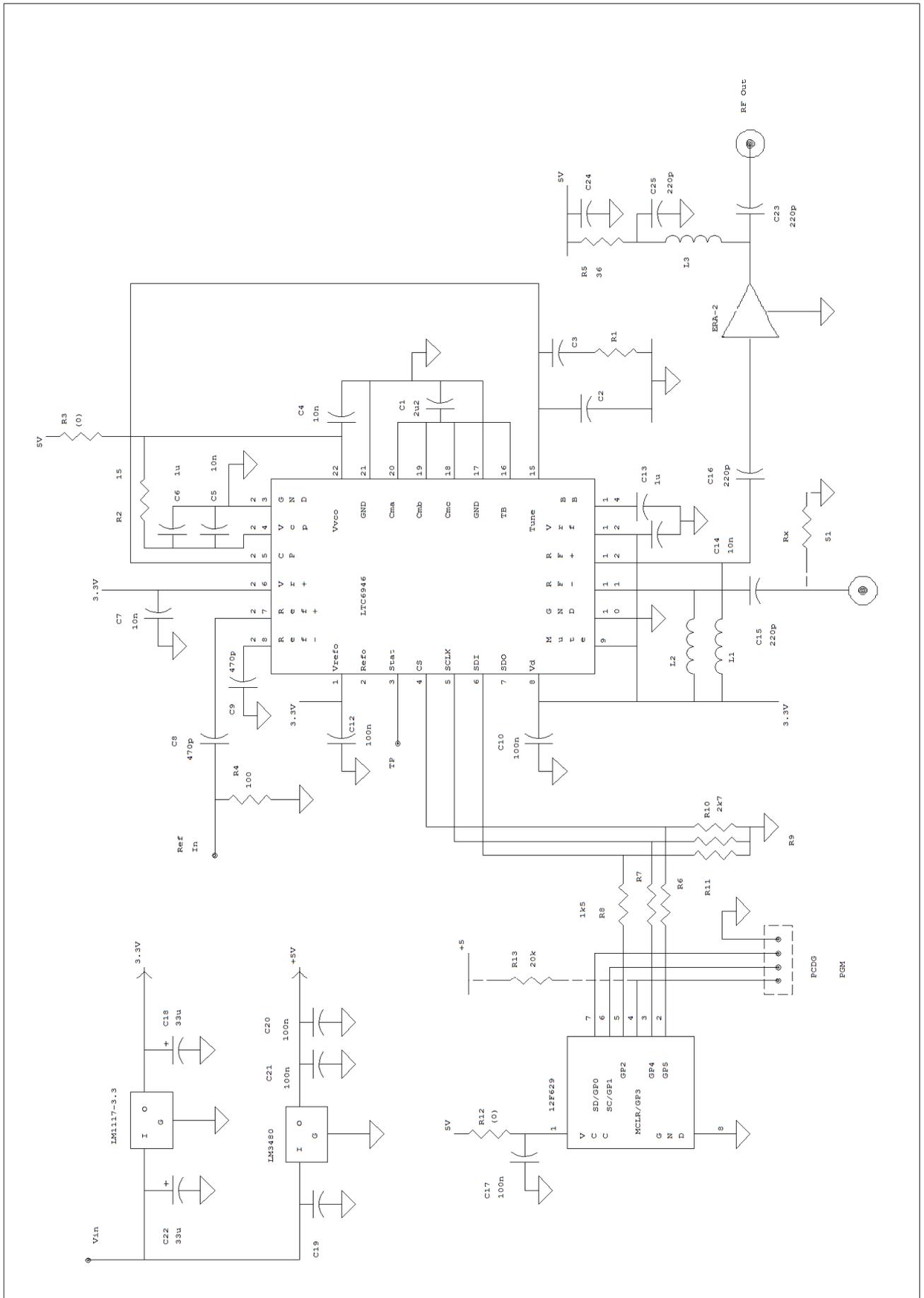
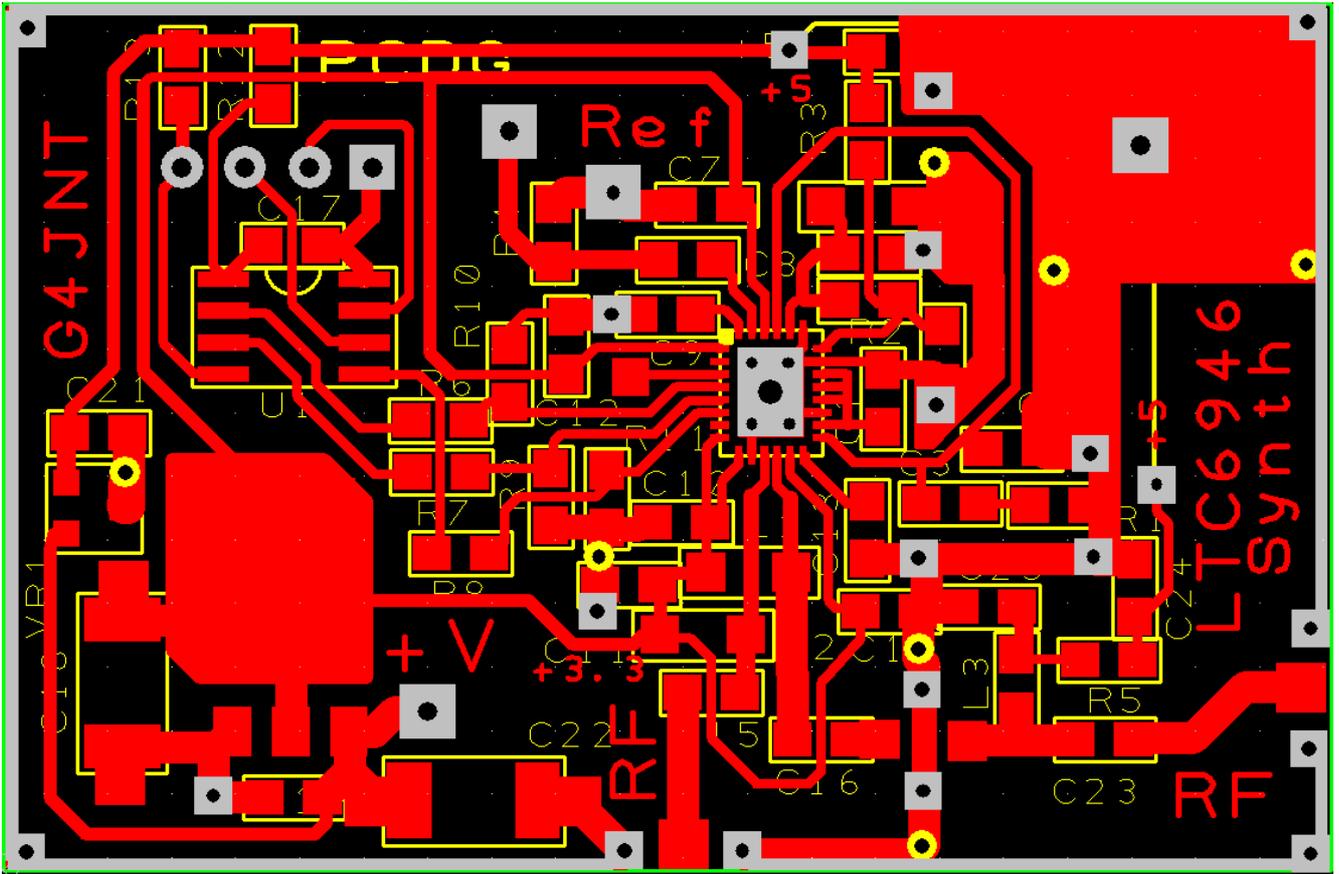


Fig 2 PCB Component Placing



LMX PLL SYNTHESIZER FILTER DESIGN			
Fout	2320.905 MHz	116.04525 kHz	
Fref	20000 kHz	20000000 Hz	20000000 Hz
Loop BW	100 kHz	100000 Hz	100000 Hz
Charge Pump Current	1.6 mA	0.0016 A	0.0016 A
VCO	60 MHz/V	60000000 Hz/V	60000000 Hz/V
Phase Margin	48 deg	0.837758061 rad	0.837758061 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	

Figure 3 Screen capture from the PLL Filter design spreadsheet File - LMXFILTER.XLS

Appendix A

LTC6946CTL PIC Operating System

The 12F629 PIC containing the **LTC6946CTL** 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 a four digit hexadecimal representation of their new contents, terminated with a [cr]. Only responses are shown on the screen, key presses are not echoed as they are typed.

Example:

R0208 [cr] (Note that the 0x or the hyphen shown on the response must not be typed).

Followed by **W [cr]**

```
G4JNT
LTC6946 CTRL
Rxxxx [CR]
[W]rite reg EE
[INIT] from EE

0000 0001 0011 1000    0x01-38
0000 0010 0000 1000    0x02-08
0000 0011 0011 0000    0x03-30
0000 0100 0000 0010    0x04-02
0000 0101 0000 0100    0x05-04
0000 0110 1000 0000    0x06-80
0000 0111 0110 0011    0x07-63
0000 1000 1111 1001    0x08-F9
0000 1001 1101 0011    0x09-D3
0000 1010 1100 0000    0x0A-C0

0000 0010 0000 1000    0x02-08

2-8 Written
```

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.

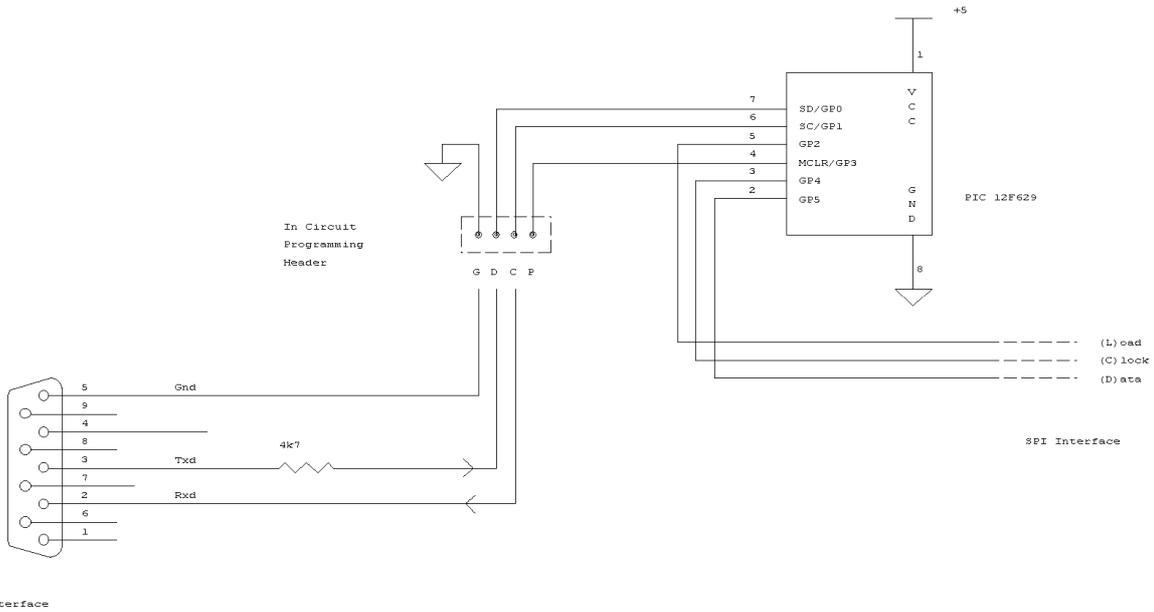


Figure A1 PIC Interface for LTC6946-CTL Operating System

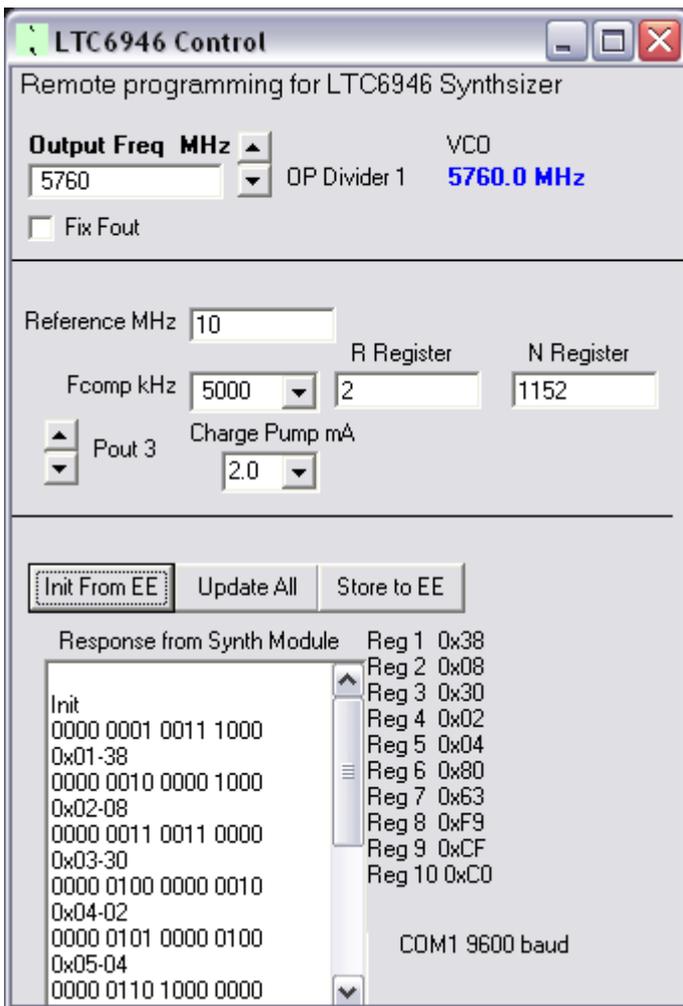


Figure A2 LTC6946_Prog User Screen

The *LTC6946_Prog* Software can send the ASCII text automatically to the controller, or can be used to simply calculate register values

NOTE. A later version of **LTC6946_Prog.EXE** is now included in the support archive.

The latest version will drive this PIC firmware as well as the four-frequency switched version described in the section entitled *LTC6946 Frequency Setting using Links* located at the end of this document

Appendix B

Phase Noise Plots

The first plot was obtained by mixing the outputs from two identical LTC6946–3 modules set to 5000 and 5020MHz respectively, viewing the resulting 20MHz mixer product using an SDR-IQ. Both modules were fed from the same 10MHz reference oscillator. The comparison frequency F_{COMP} was set at 5MHz for each, with a charge pump current of 2mA. All other settings were identical between the two modules.

If we assume the phase noise from each of the two similar modules is the same, the spectrum plot will show the cumulative power-added value and therefore will be 3dB higher than the phase noise for one synthesizer alone.

The resolution bandwidth is 9.7Hz, so the individual phase noise in dBc/Hz of each individual module is therefore 13dB lower than the value read off from the plot. At 20kHz spacing, the value is in the region of -75dBc/Hz.

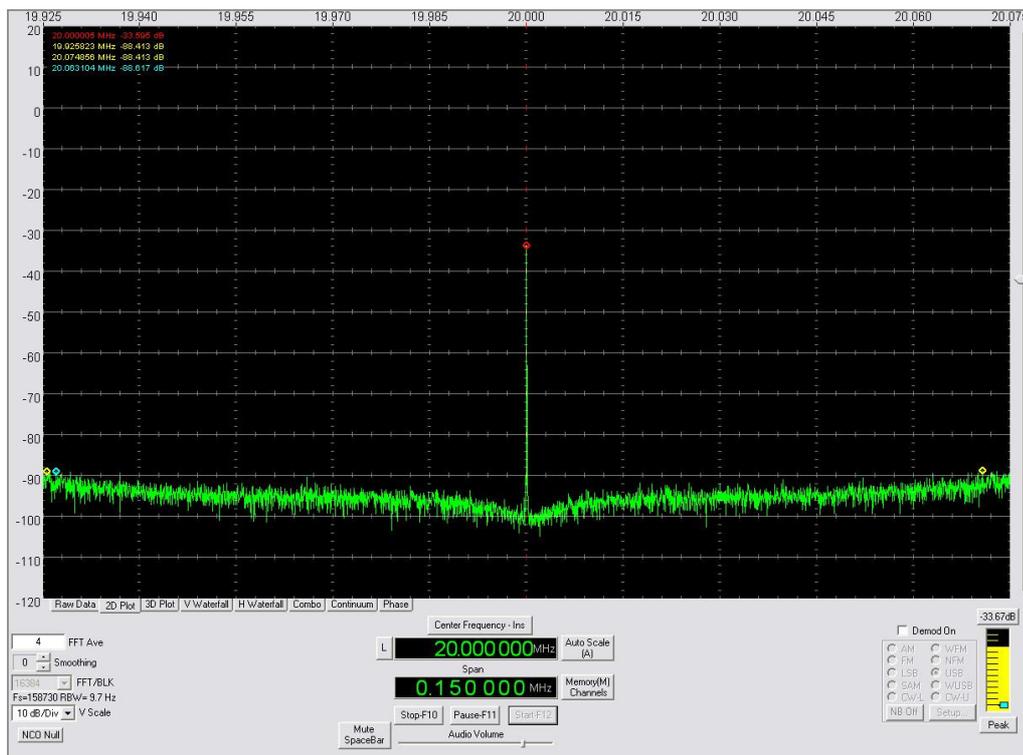


Figure B1 Phase noise plot generated by mixing the outputs from two LTC6946-3 Synthesizer modules running at 5000 and 5020MHz respectively. $F_{COMP} = 5\text{MHz}$

The second plot shows the result from two –2 variants set at 3390 and 3410MHz respectively. F_{COMP} is the maximum possible for the reference used, 10MHz, with charge pump current set to 8mA on each.

The phase noise is in the region of –88dBc/Hz, the lower value being achieved by using the highest possible comparison frequency and optimising the charge pump current. The discrete spuri at around +/-5kHz spacing are an artefact of the SDR-IQ measuring receiver.

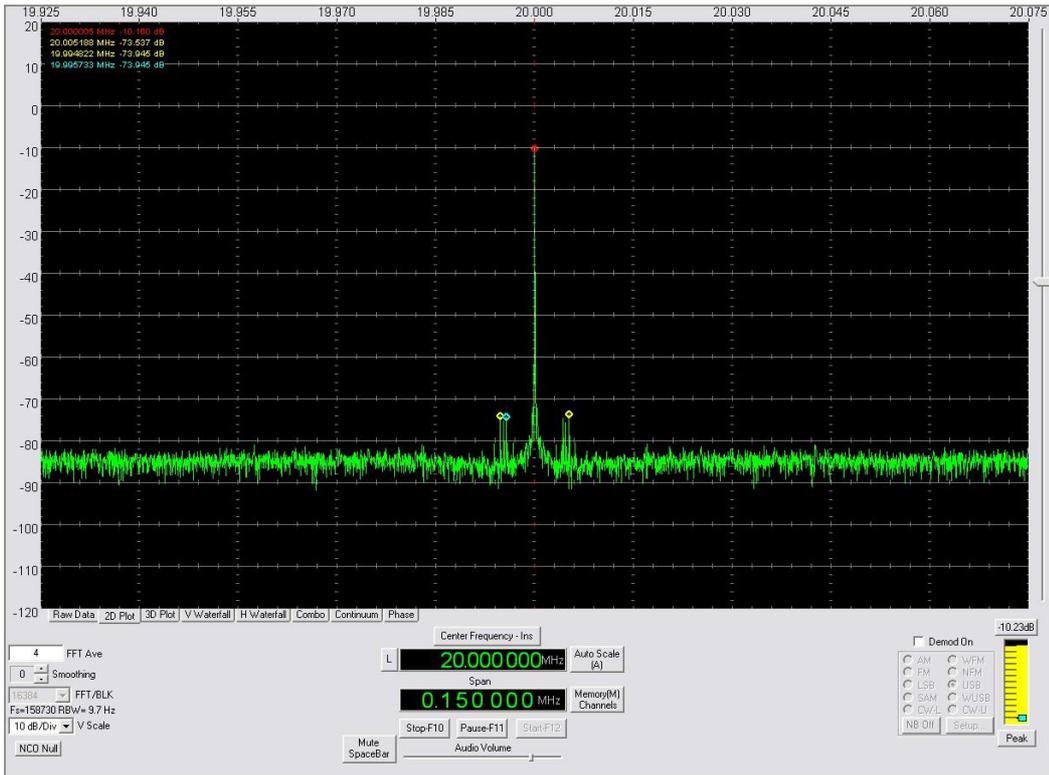


Figure B2 LTC6946-2 at 3390 and 3410MHz, $F_{COMP} = 10\text{MHz}$

However, keeping the same configuration but lowering F_{COMP} to 250kHz results in a considerably worse phase noise at -55dBc/Hz at 20kHz spacing. The loop filter is not optimised for this value of F_{COMP} , and an improved phase noise performance should be possible by choosing appropriate component values. This was the best possible, setting the charge pump to its maximum value of 11mA

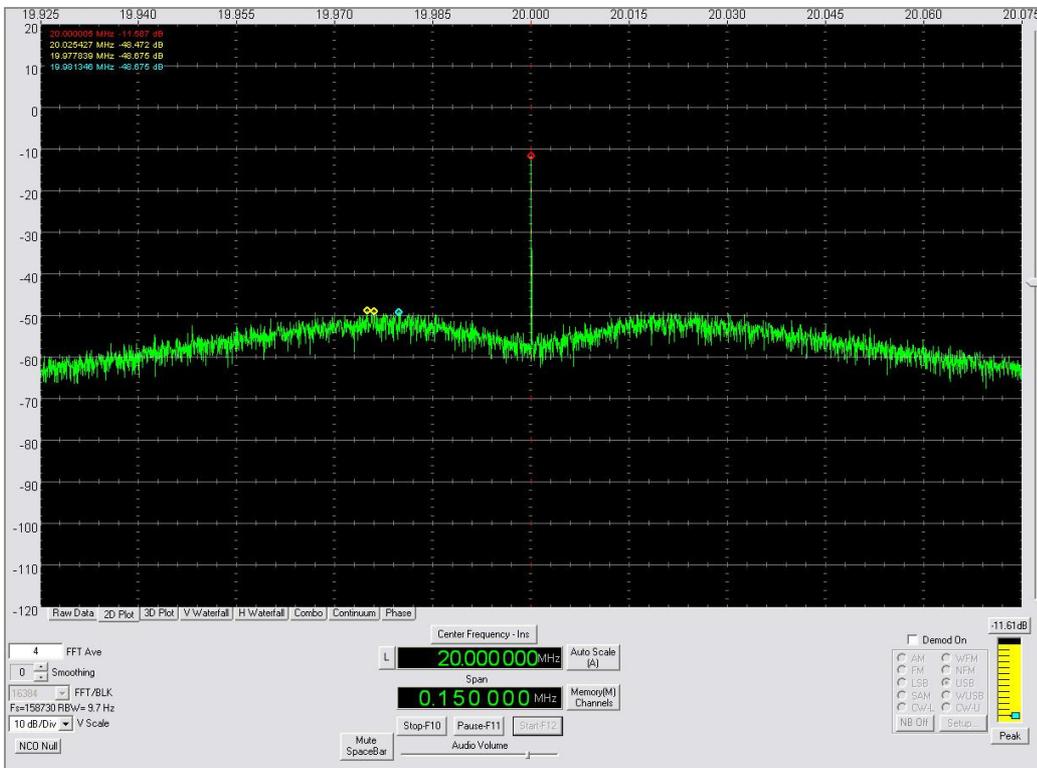


Figure B3 As above, but reducing F_{COMP} to 250kHz (no loop filter component change)

LTC6946 Control
Remote programming for LTC6946 Synthesizer

Output Freq MHz **VCO**
OP Divider 1 **3410.0 MHz**

Fix Fout

Reference MHz **R Register** **N Register**

Fcomp kHz **Charge Pump mA**
Pout 1

Init From EE Store to EE

Response from Synth Module

0000 0001 0011 1000	Reg 1 0x38
0x01-38	Reg 2 0x08
	Reg 3 0x50
	Reg 4 0x01
	Reg 5 0x01
0000 0010 0000 1000	Reg 6 0x55
0x02-08	Reg 7 0x63
	Reg 8 0xE9
0000 0011 0101 0000	Reg 9 0xDA
0x03-50	Reg 10 0xC0
0000 0100 0000 0001	
0x04-01	
0000 0101 0000 0001	

COM1 9600 baud

LTC6946 Control
Remote programming for LTC6946 Synthesizer

Output Freq MHz **VCO**
OP Divider 1 **3390.0 MHz**

Fix Fout

Reference MHz **R Register** **N Register**

Fcomp kHz **Charge Pump mA**
Pout 1

Init From EE Store to EE

Response from Synth Module

0000 0001 0011 1000	Reg 1 0x38
0x01-38	Reg 2 0x08
	Reg 3 0x50
	Reg 4 0x01
	Reg 5 0x01
0000 0010 0000 1000	Reg 6 0x53
0x02-08	Reg 7 0x63
	Reg 8 0xE9
0000 0011 0101 0000	Reg 9 0xDA
0x03-50	Reg 10 0xC0
0000 0100 0000 0001	
0x04-01	
0000 0101 0000 0001	

COM3 9600 baud

LTC6946 Frequency Setting using Links

Alternative firmware '*LTC6946_Switched*' for the 12F629 PIC controller allows any one of four pre-stored frequencies and settings to be loaded based on two links set on the programming header.

A third link allows normal link-select operation, or a programming mode where RS232 ASCII text commands are used to programme the memory locations associated with each of the four frequencies.

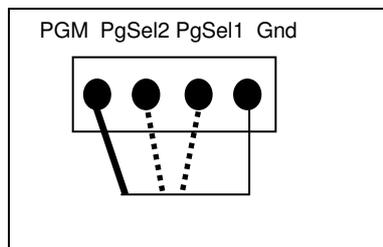
Hardware Modification

A pull up resistor first has to be installed on the PGM pin. The latest PCB has provision for this, shown as R13 in the top left of the PCB and the diagram shown above includes this resistor. It is not particularly critical and a value of 20k is about right. As this connection functions as the PGM pin when blowing new firmware into the PIC, ensure then value is not too low to impede the PIC programming process. For normal operation the PGM pin is pulled high. To enter programming mode (using the RS232 interface to modify register settings in EEPROM), connect this pin to ground. The frequency selection links are installed on the two remaining pins. There is no need for additional resistors on these as weak pull-ups are included within the PIC.

Normal One-of-Four Frequency Selection

Four completely separate sets of registers are stored in EE memory. These are referred to as 'Pages' and are selected by two links PgSel1 and PgSel2, Whenever a link setting is changed, the new data set is recalled in its entirety and loaded into the synthesizer chip. This allows the complete setup to be changed for each of the four selections. For normal frequency changing within a single band, most of the registers will usually remain unchanged with only the contents of the N divider altered to generate the new frequency. However, the option of complete reconfiguration is available.

Normal one-of-four frequency selection is started when the programming link is not installed. A PgSel link fitted, connecting that pin to ground, is defined as logic 1. No link is a logic zero. Therefore, with no links fitted Page 0 data is loaded into the chip; with both links installed Page 3 data is loaded.



Programming Mode

Programming the four sets of EEPROM register contents is performed using a serial RS232 interface and terminal software such as *Hyperterm*, using commands similar to those for the original '*LTC6946-ctl*' firmware. However, accessing the programming menu is done in a different way from that firmware.

It is only possible to get into Programming Mode by installing the programming link on PGM, then resetting, or turning on, the controller. The serial interface is connected in the same manner as the original programming interface, as shown in Figure A1 (after removing all PgSel links). The LTC6946 chip is not loaded with any data during programming {*}. Ensure the RS232 interface is connected, and a serial communications programme such as *Hyperterm* is running. RS232 Parameters are 9600 Baud, 8 bit, No parity, 2 stop bits.

When the controller starts up, an introductory menu similar to that shown below appears {**} listing all four pages of register values. Pages 0 to 3 contain the register contents loaded for each possible setting of the links.

```

G4JNT
LTC6946 4 Freq Control
Rxxxx [CR]
[W]rite reg EE
Pn Page
EXIT

Page 0
0000 0001 0011 1000 0x01-38
0000 0010 0000 1000 0x02-08
0000 0011 0000 0000 0x03-00
0000 0100 0000 0101 0x04-05
0000 0101 0000 0110 0x05-06
0000 0110 0111 0010 0x06-72
0000 0111 0110 0011 0x07-63
0000 1000 1111 1110 0x08-FE
0000 1001 1100 1111 0x09-CF
0000 1010 1100 0000 0x0A-C0
Page 1
0000 0001 0011 1000 0x01-38
0000 0010 0000 1000 0x02-08
0000 0011 0000 0000 0x03-00
0000 0100 0000 0101 0x04-05
0000 0101 0000 0110 0x05-06
0000 0110 0101 0100 0x06-54
0000 0111 0110 0011 0x07-63
0000 1000 1111 1110 0x08-FE
0000 1000 1111 1110 0x08-FE
0000 1010 1100 0000 0x0A-C0

```

```

Page 2
0000 0001 0011 1000 0x01-38
0000 0010 0000 1000 0x02-08
0000 0011 0000 0000 0x03-00
0000 0100 0000 0101 0x04-05
0000 0101 0000 0110 0x05-06
0000 0110 0111 0110 0x06-36
0000 0111 0110 0011 0x07-63
0000 1000 1111 1110 0x08-FE
0000 1001 1100 1111 0x09-CF
0000 1010 1100 0000 0x0A-C0
Page 3
0000 0001 0011 1000 0x01-38
0000 0010 0000 1000 0x02-08
0000 0011 0000 0000 0x03-00
0000 0100 0000 0101 0x04-05
0000 0101 0000 0110 0x05-06
0000 0110 0001 1000 0x06-18
0000 0111 0110 0011 0x07-63
0000 1000 1111 1110 0x08-FE
0000 1001 1100 1111 0x09-CF
0000 1010 1100 0000 0x0A-C0

```

The **[R]**egister setting and **[W]**rite commands are used in the same way as for the original RS232 real-time controller. Two new commands are introduced: the Page command **Pn**, and **EXIT** (which has to be typed in full).

All commands are terminated with a carriage return. The original 'INIT' command has been removed. Commands are not echoed back as they are typed.

To change the EE contents for any page, first use the '**Pn**' command to set the page to be worked-on. Replace **n** by 0, 1, 2 or 3 as required. For example **P2 [cr]** sets page 2. When the P command is accepted, the existing register settings in that page are shown. To alter any particular register, use the '**Rxxxx**' command. For example, '**R0635 [cr]**' changes register 6 to hold the value hex 0x35 (register 6 is the lowest byte of the N divider).

A response like `0000 0110 0011 0101 0x06-35` will be generated, showing the new contents.

Type **W [cr]** to store this new value into EE memory, generating a response similar to : `06-35 Written`.

The process is repeated for each register to be changed.

Finally, type **EXIT**. The entire memory contents are listed in the same format as when programming mode was started, followed by the word *EXIT*. all further RS232 communication is inhibited and the controller is now in normal running mode, monitoring the link settings and changing the generated frequency according to the new EEPROM contents.

```

Page 2
0000 0001 0011 1000 0x01-38
0000 0010 0000 1000 0x02-08
0000 0011 0000 0000 0x03-00
0000 0100 0000 0101 0x04-05
0000 0101 0000 0110 0x05-06
0000 0110 0111 0110 0x06-36
0000 0111 0110 0011 0x07-63
0000 1000 1111 1110 0x08-FE
0000 1001 1100 1111 0x09-CF
0000 1010 1100 0000 0x0A-C0

0000 0110 0011 0101 0x06-35

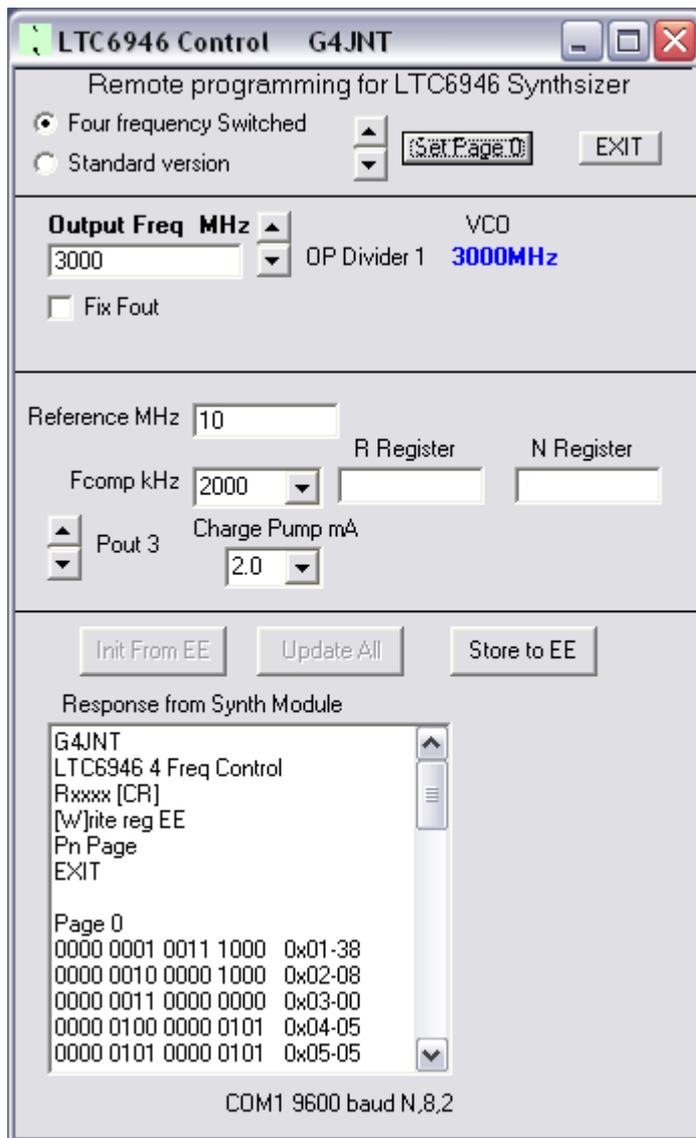
06-35 Written

```

----- Latest Update -----

The latest version of **LTC6946_Prog.exe** can now be used both with the original single frequency PIC code and with the four frequency version. The programming link has to be installed, and the check box labelled 'Four Frequency Switched' selected. Use the up / down arrows to select Page 0 – 3, then set all parameters for that page of memory. When done, click 'Store to EE' which will also load the Synthesizer chip with the settings, allowing verification of correct operation. (There is no option for updating the chip without storing to EE).

Repeat for each page, selecting each page from the up / down arrows at the top. Click the Exit button, at the top to end programming mode, then remove the programming link to allow normal operation.



Latest version of **LTC6946_Prog.exe** with Four-frequency option selected

- {*} In programming mode the non-loaded chip could be generating some arbitrary and unlocked signal output. Ensure this cannot cause any unwanted effects or damage in external equipment.
- {**} The values shown in the listing are the default settings in the software as downloaded. These generate four frequencies of 550, 540, 530, 520MHz for pages 0 – 3 respectively.