Frequency Synthesizer

LNO-HP3xM

Operating Manual

Rev. 1.2

Advantex LLC

November 14, 2013

Russian Federation, 111250, Moscow,
Krasnokazarmennaya st., 13/1
tel. +7 (495) 721-47-74, +7(495) 728-08-03
info@advantex.ru
http://advantex-rf.com, www.advantex.ru

IF/RF & Microwave Design
AdvanteX

Document Revisions

Rev.	Date	Description
1.0	November 14, 2012	LNO Frequency Synthesizer Manual (starting from LNO-HP34M-RF)
1.1	July 11, 2013	Revised sections "SPI Command Set" and "SPI Programming"
1.2	November 14, 2013	Axis on figure 19 are renamed according to their meaning in table 13 and 14. Equations are replaced accordingly.

Contents

List of Figures

List of Tables

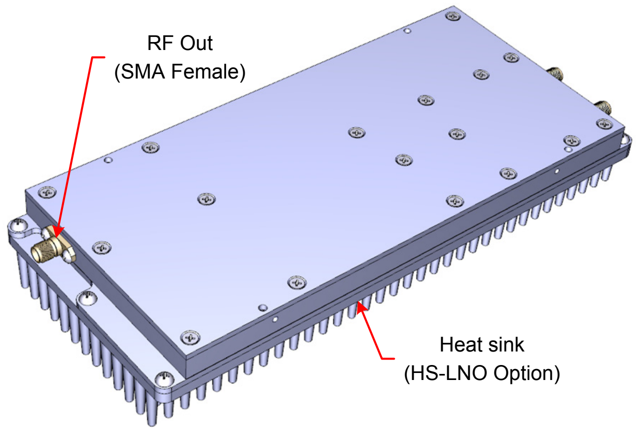

Figure 1: RF Out Connector

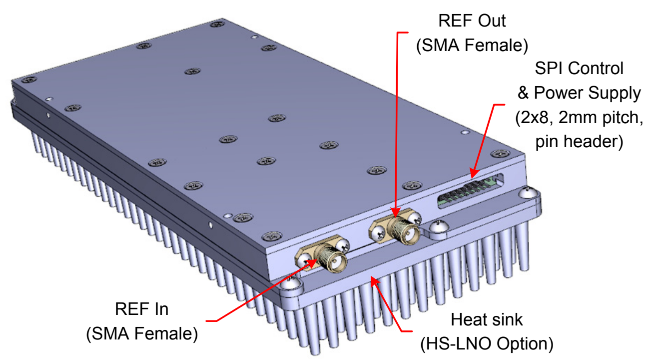

Figure 2: REF In, REF Out and SPI Connectors

1 Getting Started

1.1 Interfaces and Connectors

Figures 1, 2, 3 show external interfaces and connectors of LNO-HP3xM-RF synthesizer with heatsink (HS-LNO option). The synthesizer can be supplied without heatsink but in this case you need to provide good heat sink by your own means. The module has the following connectors:

RF Out – RF signal output, 4 MHz to 8 GHz frequency range with less than 0.001 Hz step, -20 to $+28$ dBm level range with about 0.05 dB step, connector type – SMA, female;

REF In – input of optional external reference frequency signal, rated level 0 dBm. Signal of any frequency can be applied in 20 to 150 MHz range;

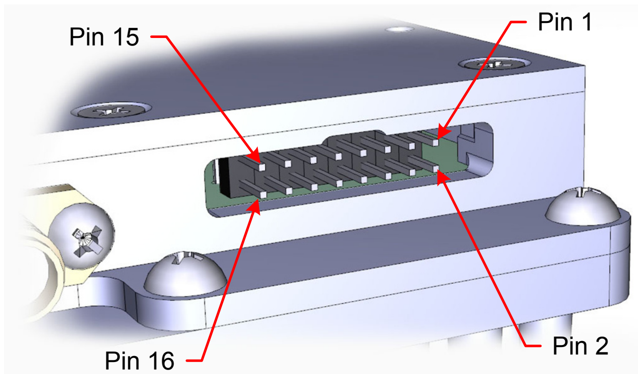

Figure 3: SPI Control and Power Supply interface

REF Out – output of reference frequency signal that is in use at the moment, output level 0 to +6 dBm. If external signal is used as the reference (applied to REF In) then REF Out duplicates this signal, if internal TCXO is used, it duplicates internal reference signal (147 MHz);

SPI Control and Power Supply – SPI interface designed to control LNO module, LVTTL 3.3V levels, max. clock rate is 10 MHz. Connector type: 2 row, 2 mm pitch, 16-pin holder.

Table 1 shows pinout of SPI Control and Power Supply interface.

1.2 LNO-KIT

LNO-KIT includes LNO Frequency Synthesizer and RS2SPI evaluation board (fig. 4). It is designed for quick start and helps to understand SPI protocol command set and algorithms used to control LNO synthesizer. All SPI commands which sent to the module are displayed in log window of the application. So you can easily check yourself while debugging of your own control system of the synthesizer. For remote control of LNO module follow the steps below.

1. Connect LNO module to the RS2SPI board (*Connector 1* located on the top side of the board, see fig. 4) via SPI cable (16-pin, 2-mm pitch, IDC flat cable).

2. Connect RS2SPI board to PC via USB or RS-232 cable (see jumper positions for USB/RS-232 use on fig. 4)

3. Connect RS2SPI board to **+12 VDC** power supply. Max current of the power supply source should be set to **3A** for the DC-DC converters of RS2SPI board to start. Turn on the power. Power supply consumption will be about 0.18A (for non-initialized LNO block).

Table 1: SPI Control and Power Supply pinout

Pin #	Name	Direction (relative to module)	Description
1	LNO_MOSI	In	SPI master output, slave input
3	LNO_SS#	In	SPI select signal, data are latched only if this signal is active (“0” state)
5	LNO_SCK	In	SPI clock signal, data are latched by rising edge of LNO_SCK signal
7	LNO_MISO	Out	SPI master input, slave output
8	INT	Out	Interrupt signal. “0” – normal condition, “1” – output stage overheating. When INT condition occurs, it results in turning off the output stage power supply. After some time when synthesizer is cooled to it’s rated temperature, power supply is turned on automatically, and INT signal returns to "0" state automatically. INT pin is connected to CPLD via 1 kOhm resistor, so if not used you can tie it to the GND
2, 10, 12, 14, 16	GND	–	Ground pins, internally connected to the case body
4, 6	-5V	In	Negative power supply, −5 to −9V allowed. Rated current 60mA
9, 11	+5V	In	Positive power supply, +5.0 to +5.5V allowed. Rated current 1.2A
13, 15	+9V	In	Positive power supply, +9 to +12V allowed, but to reduce overheating and power consumption it’s better not to exceed +10V. Rated current 0.6A

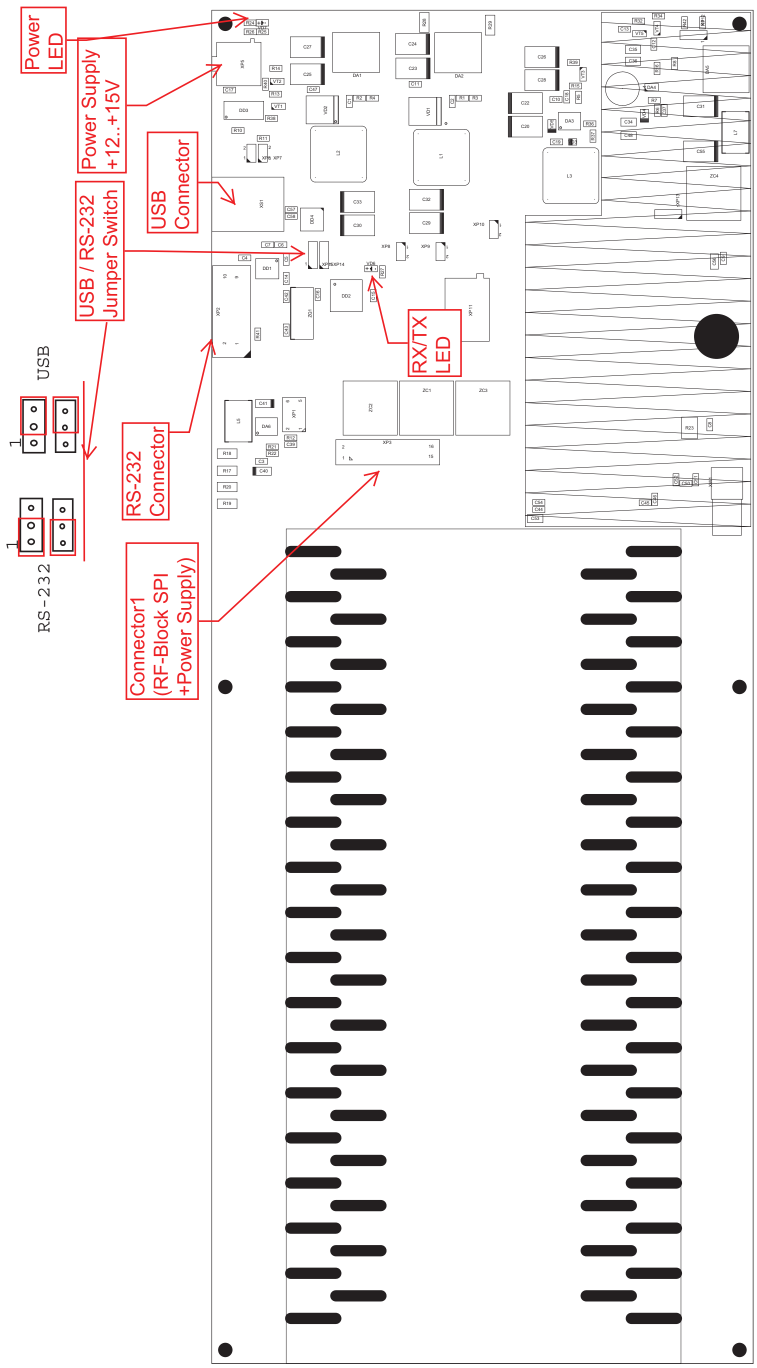

Figure 4: RS2SPI Evaluation Board

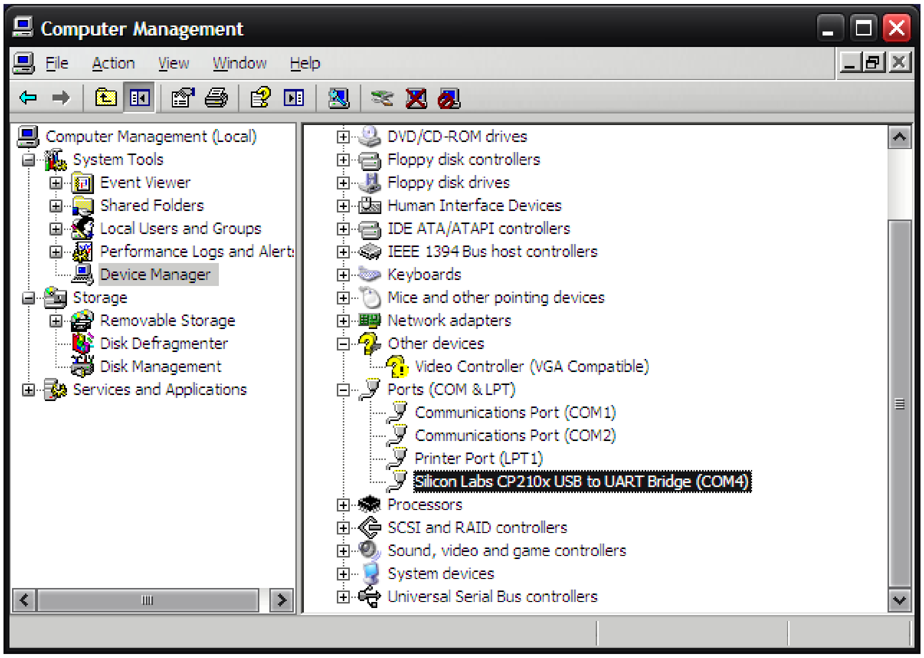

Figure 5: COM-port number

4. When using USB cable find out the COM-port number in Computer Management window (see fig. 5) You can specify another COM-port number (Properties ▷ Port Settings ▷ Advanced ▷ COM Port Number).

5. Launch *RF Debug Application* (double click on `advantex.exe` or `advantex.tcl` if you have Tcl/Tk installed).

6. Select radio-boxes as shown on the fig. 6 (specify the connector of the RS2SPI board to which LNO is connected) and click OK.

7. Select menu Setup ▷ COM_configure (fig. 7, Item 1).

8. Specify COM-port number (fig. 8) as it was determined in fig. 5. Click OK.

9. Select Program tab (fig. 7, Item 2). Press Data Read button (fig. 9, Item 1), wait for the application to download the calibration data from LNO internal memory (the process is indicated in the status bar), then return to the LNO tab (fig. 9, Item 2).

10. To start working with LNO-module press Init button (fig. 10, Item 1), then you can load frequency or output level without Init button. Please note that after each frequency loading output level is automatically reset to minimum value, so you need to load it pressing Load button (Item 5).

Figure 6: Select SPI-connector number

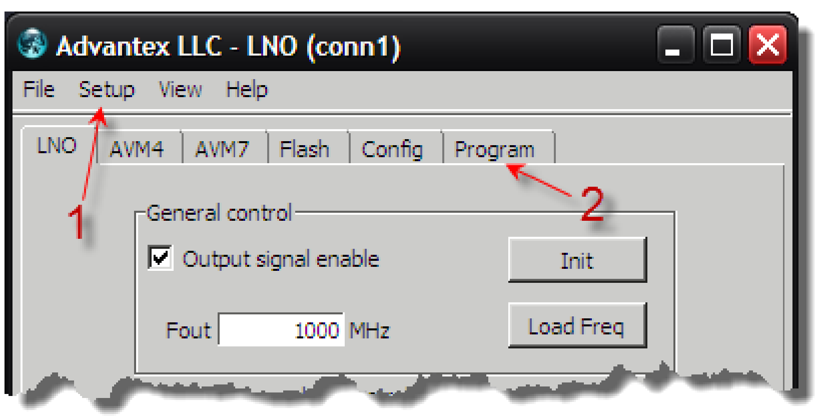

Figure 7: Main Menu

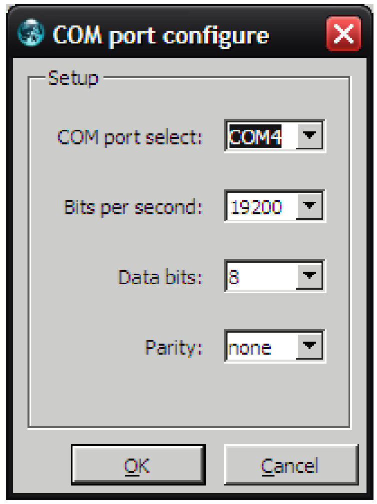

Figure 8: COM-port configuration

It's for safety of devices connected to RF output of the LNO block. You can read internal temperature of the block by pressing Read button (Item 6). After pressing Init button power consumption will increase up to 1A if Output signal enable check-box is checked (i.e. output stage amplifier is on), and 0.6..0.7A if unchecked (output stage is off).

11. If you need to use external reference, please check the box LNO ext reference input enable (fig. 11, Item 1), set the reference frequency value (Item 2) and press Init button (Item 3). To set output level proceed with Load button (Item 4, 5).

2 SPI Command Set

2.1 General Description

LNO synthesizer does not include any MCU that would make all low-level calculations for you, just simple CPLD that works as SPI multiplexer 1-to-4 channels and contains some static registers (fig. 12). This CPLD does not use any clock signal except external SPI LNO_SCK line that changes its state only when loading new data to the LNO module. The reason of this approach is to avoid interference of MCU clock signal to analog lines and to make the response time of LNO module as fast as possible.

Figure 12 shows block diagram of LNO-HP3xM-RF synthesizer with internal control lines (colored in blue). There are two types of control lines: SPI channels and static registers. There is also one special line DDS_IOUPDATE which corresponds to DDS update command which toggles the corresponding DDS pin (IO_UPDATE pin of AD9912) to make new loaded in DDS data valid.

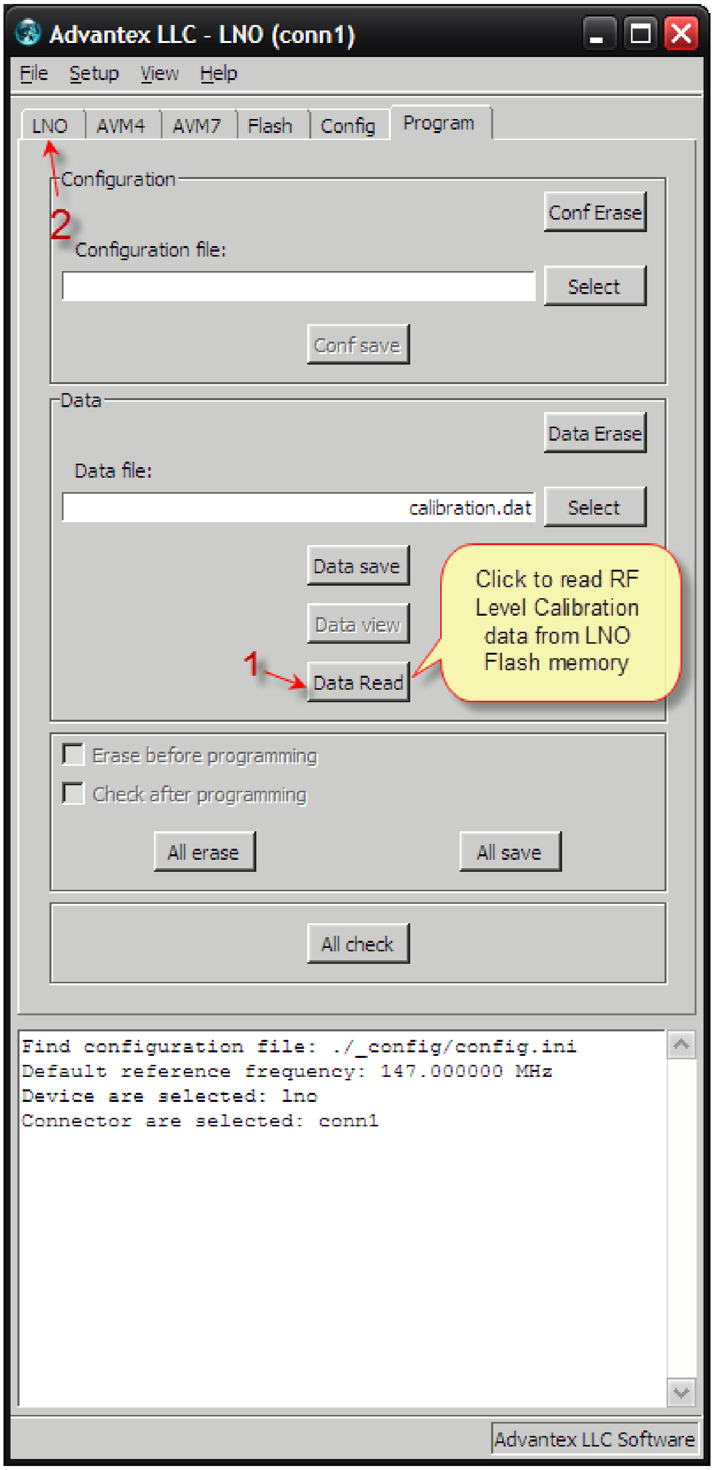

Figure 9: Read calibration data from LNO Flash-memory

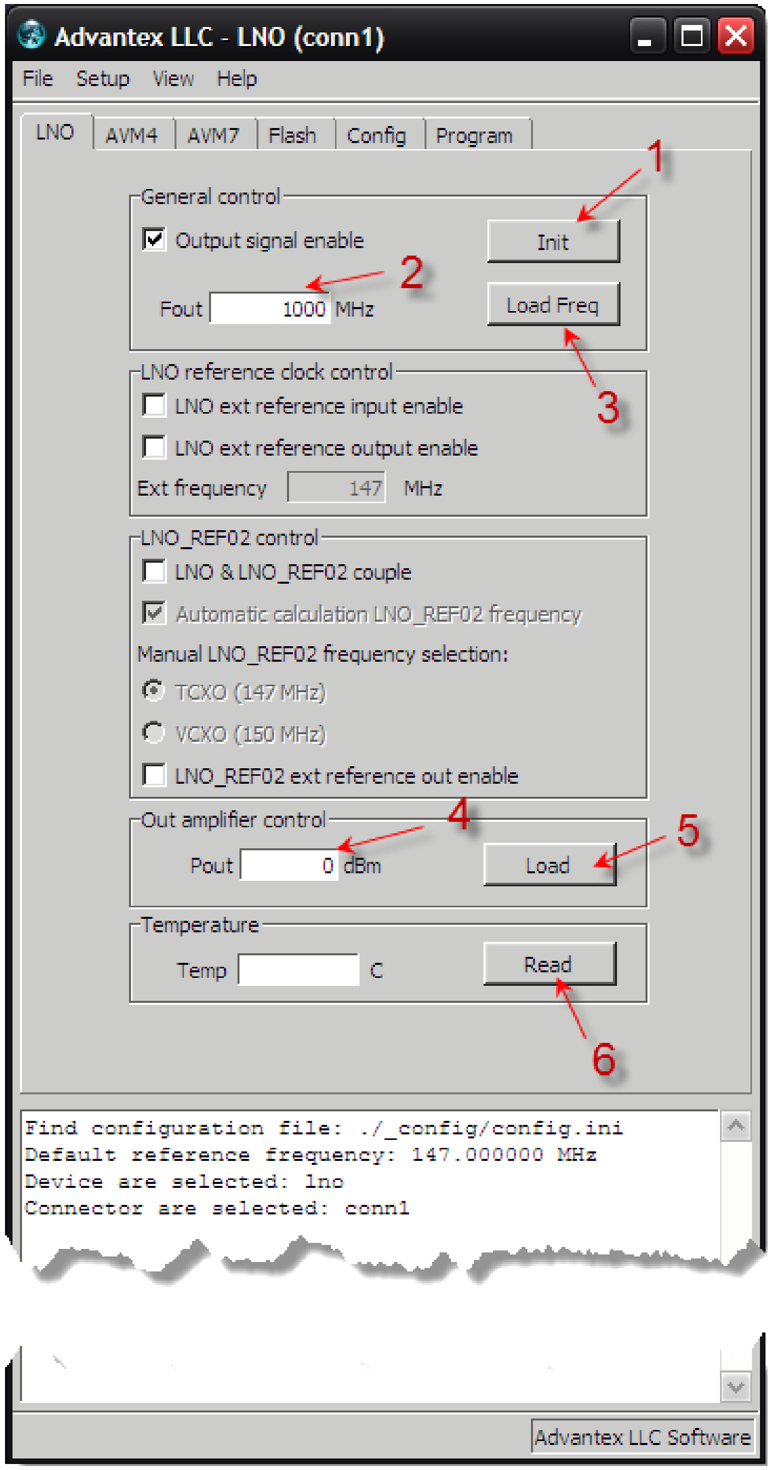

Figure 10: LNO tab

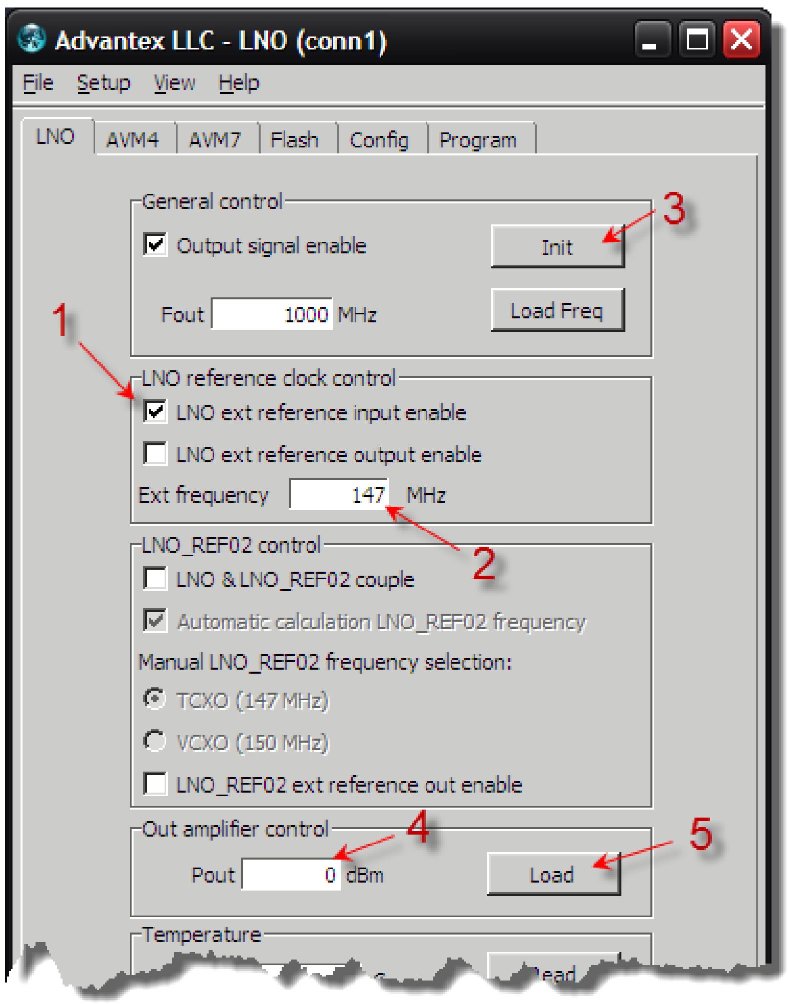

Figure 11: Ext. Reference

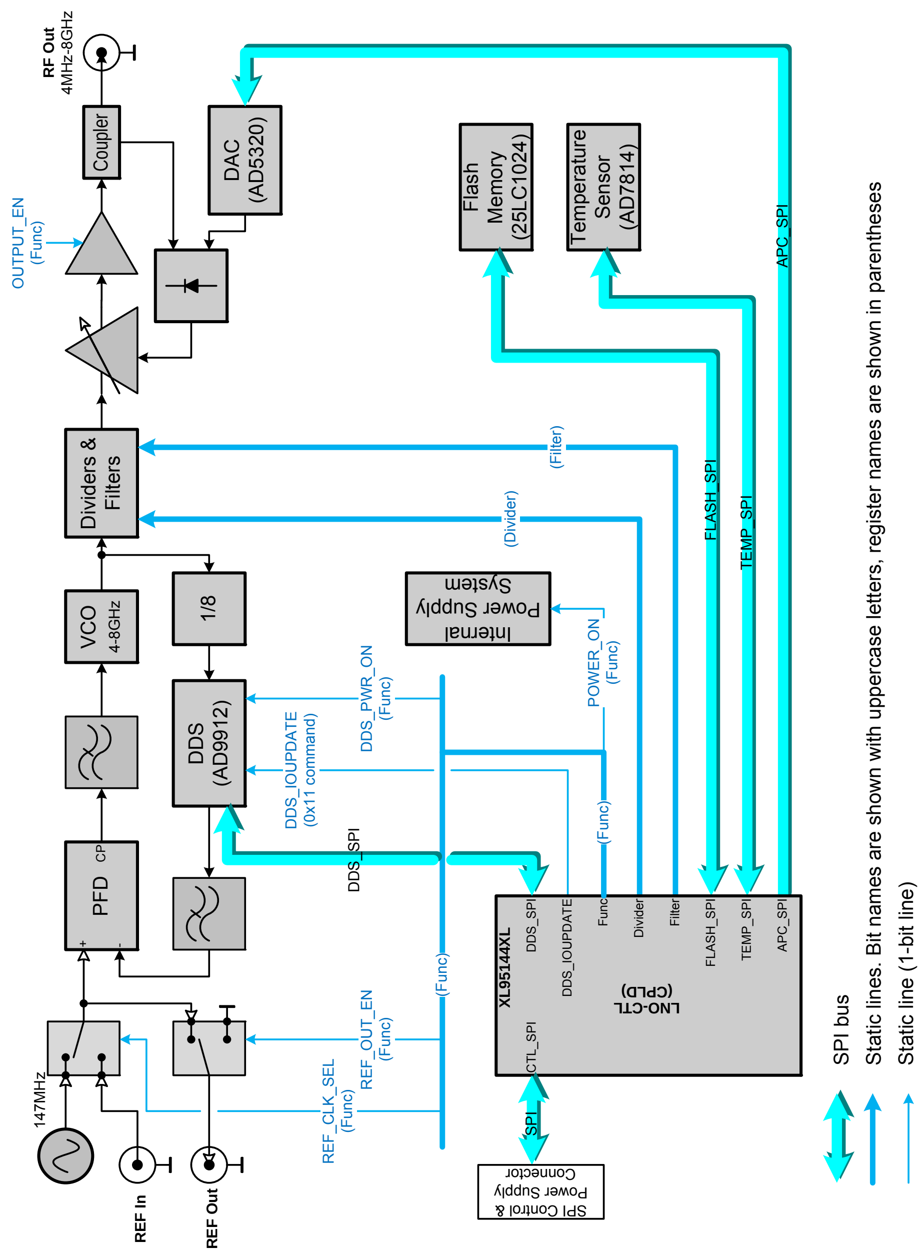

Figure 12: LNO synthesizer block diagram

There are four internal SPI channels:

DDS_SPI is connected to DDS (AD9912) that is placed in PLL loop. It is used to control frequency and phase of VCO output signal.

APC_SPI is connected to DAC (AD5320) that is responsible for APC (Automatic Power Control) system. Output signal level in dBm is proportional to the value loaded to the DAC.

FLASH_SPI is connected to Flash memory (25LC1024) which stores device ID data (signature, part number, serial number, date), calibration data for APC system and DDS spur killer, etc.

TEMP_SPI is connected to internal temperature sensor (AD7814) which can be used to retrieve temperature of the module.

There are three static registers:

Func register is used to control internal power supply system and external reference input and output operation.

Divider register is used to control additional dividers which extend the frequency range of the signal at VCO output.

Filter register is used to control filter paths used to suppress harmonic components of output signal.

Dividers and filters paths are shown in detail on figure 13.

2.2 Format of SPI Commands

LNO SPI commands consist of one command byte C[7:0] following by N data bytes. Number of data bytes (N) depends on the particular command. Data on LNO_MOSI signal line are latched on rising edge of LNO_SCK signal as shown on figure 14. MSB (Most Significant Bit) is loaded first, LSB (Least Significant Bit) – last. The first is the command byte C[7:0] all other bytes are data (fig. 15). The command byte works as the address for multiplexer implemented in CPLD and defines the destination where to transfer following data – to one of the SPI channels or to one of the static registers, and the type of operation – writing or reading.

At reading cycle data on LNO_MISO line are switched on the falling edge of LNO_SCK signal and should be latched by master on rising edge of LNO_SCK signal accordingly as shown in figure 16.

2.3 SPI Commands

Table 2 shows commands C[7:0] used to control LNO module.

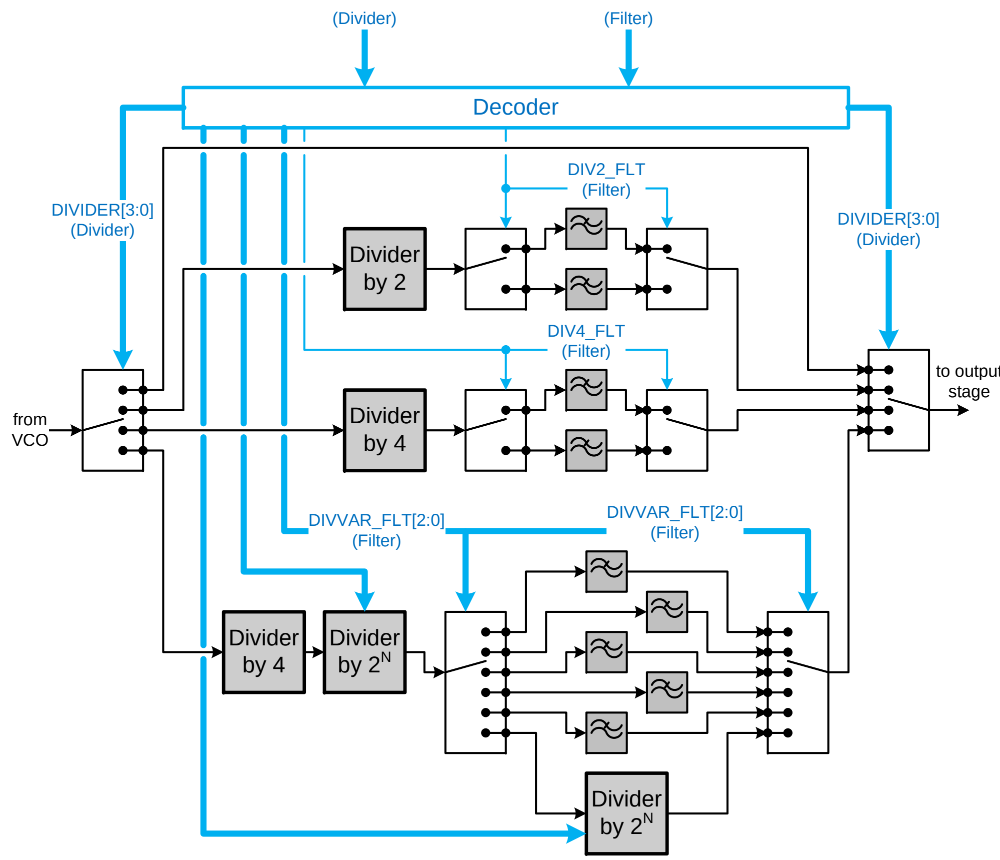

Figure 13: Output dividers and filters

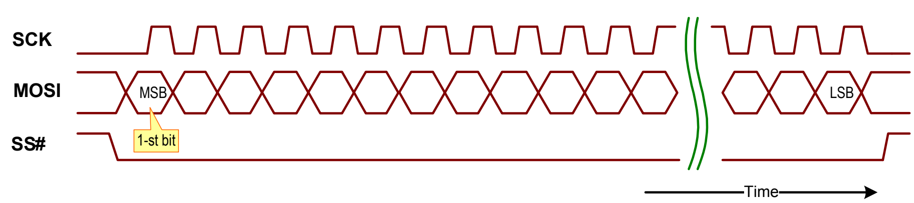

Figure 14: LNO SPI writing cycle diagram

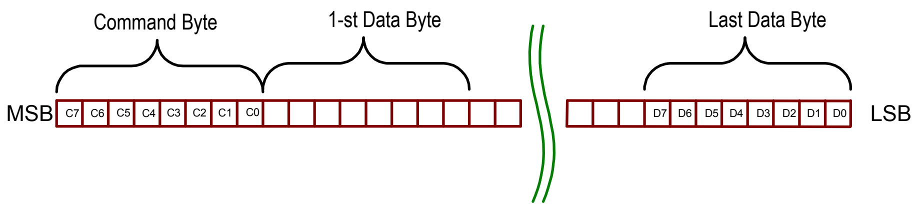

Figure 15: Command and data bytes

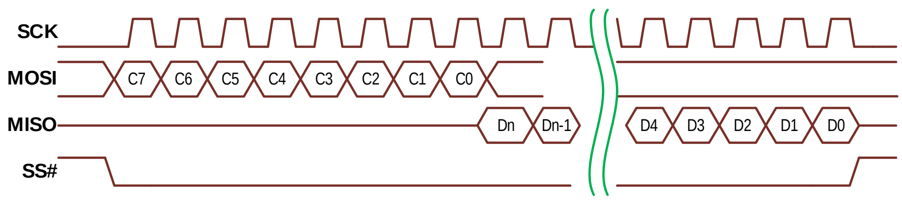

Figure 16: LNO SPI reading cycle diagram

Table 2: LNO SPI Commands C[7:0]

	C[7:0]	# of bytes	Description
1	0x01	1	Writing to Func register
2	0x81	1	Reading from Func register
3	0x02	1	Writing to Divider register
4	0x82	1	Reading from Divider register
5	0x03	1	Writing to Filter register
6	0x83	1	Reading from Filter register
7	0x10	–	Access to DDS_SPI (DDS AD9912)
8	0x11	1	Toggling DDS IO_UPDATE signal (DDS AD9912)
9	0x20	2	Access to APC_SPI (DAC AD5320)
10	0x30	2	Access to TEMP_SPI (Temperature Sensor AD7814)
11	0x70	–	Access to FLASH_SPI (Flash Memory 25LC1024)

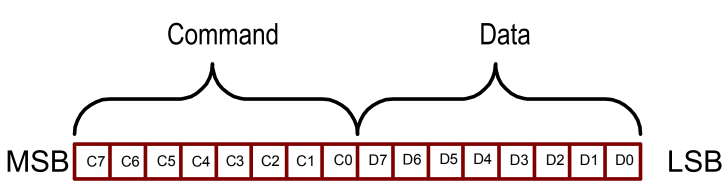

Figure 17: Writing 1 data byte

Table 3: Func register

D7	D6	D5	D4	D3	D2	D1	D0	Description
			DDS_PWR_ON	OUTPUT_EN	REF_OUT_EN	REF_CLK_SEL	POWER_ON	Bit Name
0	0	0	0	0	0	0	0	Default Values
x	x	x	x	x	x	x	0	Internal Power Supply System OFF
x	x	x	x	x	x	x	1	Internal Power Supply System ON
x	x	x	x	x	x	0	x	External signal applied to REF In input is used as reference
x	x	x	x	x	x	1	x	Embedded TCXO (147 MHz) is used as reference frequency
x	x	x	x	x	0	x	x	REF Out output is OFF
x	x	x	x	x	1	x	x	REF Out output is ON
x	x	x	x	0	x	x	x	RF Out output is OFF
x	x	x	x	1	x	x	x	RF Out output is ON
x	x	x	0	x	x	x	x	DDS power is OFF. This also resets the DDS settings, so you need to init it again after power on
x	x	x	1	x	x	x	x	DDS power is ON

2.3.1 Func register writing C[7:0]=0x01

This command writes 1 data byte D[7:0] to Func register (fig. 17). Table 3 shows the meaning of these data bits.

2.3.2 Func register reading C[7:0]=0x81

This command reads the content of Func register. See table 3 and figure 16 for more details.

2.3.3 Divider register writing C[7:0]=0x02

This command writes 1 data byte D[7:0] to Divider register (fig. 17). It defines the division factor of VCO output signal. VCO frequency range is from 4 to 8 GHz, so to obtain any other frequency which is lower than 4 GHz you need to divide VCO frequency. To cover all range from 4 MHz to 8 GHz output dividers

Table 4: Divider register

D7	D6	D5	D4	D3	D2	D1	D0	Divider factor
				DIVIDER3	DIVIDER2	DIVIDER1	DIVIDER0	Bit Name
0	0	0	0	0	0	0	0	Default Values
x	x	x	x	0	0	0	0	1
x	x	x	x	0	0	0	1	2
x	x	x	x	0	0	1	0	4
x	x	x	x	0	0	1	1	8
x	x	x	x	0	1	0	0	16
x	x	x	x	0	1	0	1	32
x	x	x	x	0	1	1	0	64
x	x	x	x	0	1	1	1	128
x	x	x	x	1	0	0	0	256
x	x	x	x	1	0	0	1	512
x	x	x	x	1	0	1	0	1024

with 2^N division factors are used (fig. 13). Table 4 shows the meaning of data bits controlling these output dividers.

2.3.4 Divider register reading C[7:0]=0x82

This command reads the content of Divider register. See table 4 and figure 16 for more details.

2.3.5 Filter register writing C[7:0]=0x03

This command writes 1 data byte D[7:0] to Divider register (fig. 17). To suppress 2-nd, 3-rd and other harmonics of output signal low-pass filters are used. LNO synthesizer has 9 filters with different pass bands (fig. 13). Table 5 shows how to turn on the appropriate filter for given output frequency.

2.3.6 Filter register reading C[7:0]=0x83

This command reads the content of Filter register. See table 5 and figure 16 for more details.

2.3.7 DDS_SPI Access C[7:0]=0x10

This command is used to access the DDS (AD9912) controlling phase and frequency of VCO output signal. Number of data bytes can be different but not

Table 5: Divider register

D7	D6	D5	D4	D3	D2	D1	D0	Output frequency f_{OUT}, MHz
			DIV2_FLT	DIV4_FLT	DIVVAR_FLT2	DIVVAR_FLT1	DIVVAR_FLT0	Bit Name
0	0	0	0	0	0	0	0	Default Values
x	x	x	0	0	0	0	0	$f_{OUT} < 62.5$
x	x	x	0	0	0	0	1	$62.5 \leq f_{OUT} < 135$
x	x	x	0	0	0	1	0	$135 \leq f_{OUT} < 210$
x	x	x	0	0	0	1	1	$210 \leq f_{OUT} < 340$
x	x	x	0	0	1	0	0	$340 \leq f_{OUT} < 560$
x	x	x	0	0	1	0	1	$560 \leq f_{OUT} \leq 1000$
x	x	x	0	0	1	1	1	$1000 < f_{OUT} < 1500$
x	x	x	0	1	1	1	1	$1500 \leq f_{OUT} \leq 2000$
x	x	x	0	1	1	1	1	$2000 < f_{OUT} < 2850$
x	x	x	1	1	1	1	1	$2850 \leq f_{OUT} \leq 4000$
x	x	x	x	x	x	x	x	$4000 < f_{OUT} \leq 8000$, the RF-path for this frequency range is defined by Divider register content

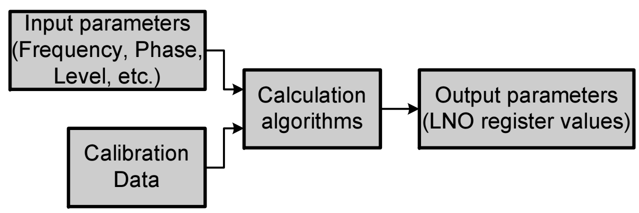

Figure 18: Input and output data for LNO programming

less than 3 bytes. Actual length of data is defined in first 2 data bytes (Long Instruction). For more information refer to section 3 and AD9912 data sheet.

2.3.8 APC_SPI Access C[7:0]=0x20

This command is used to access the DAC (AD5320) controlling APC (Automatic Power Control) system of LNO module, i.e. it defines the level of output signal. This command uses 2 data bytes D[15:0]. For more information refer to section 3 and AD5320 data sheet.

3 SPI Programming

3.1 Parameter Calculations

Working with synthesizer implies some calculation procedure that should be implemented on the user side. It includes algorithms of APC DAC value calculations based on the calibration data, DDS FTW register value calculations based on the input frequency, etc.

Thus we have some input parameters in user friendly format (e.g. frequency, phase, level as float numbers), some calibration data stored in flash memory embedded to the module, and as a result of calculation algorithms – output data in low-level format (i.e. register values), see figure 18. This section describes how to find these output parameters.

Table 6 shows input parameters. Some of them are in ready-to-use format, like power_on, outamp_en, other can not be loaded to LNO directly. Table 7 shows intermediate variables used in calculation algorithms, table shows output parameters which are downloaded to LNO registers.

n_pow

```
if(fr_out>4000) then
{n_pow=0}
else
{n_pow = floor(log(4000.0/fr_out)/log(2))+1}
```

Table 6: LNO input parameters

Parameter	Type	Corresponding register bits	Description
power_on	1 bit	POWER_ON (Func), DDS_PWR_ON (Func)	Turns ON/OFF LNO internal power supply system. “0” – power system is OFF (standby mode), “1” – power is ON (normal operation)
outamp_en	1 bit	OUTPUT_EN (Func)	Turns ON/OFF RF Out output. “0” – output stage is OFF, “1” – ON
ref_clk_sel	1 bit	REF_CLK_SEL (Func)	Selects reference frequency source (internal TCXO or external signal applied to REF In input). “0” – external, “1” – internal
ref_out_en	1 bit	REF_OUT_EN (Func)	Turns ON/OFF REF Out output. “0” – OFF, “1” – ON
fr_out	float	–	Frequency of output signal (at RF Out output) in MHz
p_out	float	–	Level of output signal (at RF Out output) in dBm
fr_ref	float	–	Frequency of reference signal (internal TCXO frequency or at REF In input if external source is used) in MHz. By default TCXO frequency is 147.0 MHz, exact value is stored in Flash memory, see section 3.4 for more details. So it is a good practice to read this value from flash before using it, since it can be slightly different from nominal value 147 MHz
phase	float	–	Relative phase shift of output signal in radians. For example, if you synchronize two LNO synthesizers form the same reference source, and set the same output frequency for both of them, you can shift phase of one signal relative to another in 360 degree range

Table 7: Intermediate variables and output parameters

Parameter/variable	Type	Description
Intermediate		
fr_vco	float	VCO output frequency ($4000 \leq$ fr_vco ≤ 8000)
divider	integer	Output division factor, multiple to 2, $1 \leq$ divider ≤ 1024
Output		
ftw	48 bit	DDS Frequency Tuning Word
ptw	16 bit	DDS Phase Tuning Word
n_pow	4 bit	Division exponent (divider=2^{n_pow})
divvar_flt	3 bit	Filter selection when divider > 4
div2_flt	1 bit	Filter selection when divider = 4
div4_flt	1 bit	Filter selection when divider = 2
poutbits	12 bit	Data value for APC DAC (AD5320), corresponds to output signal level

divider

```
divider = 2^n_pow (2 to the n_power)
```

fr_vco

```
fr_vco = fr_out*divider
```

ftw

```
ftw = round((2^51)*fr_ref/fr_vco)
```

ptw

```
ptw = round((2^16)*phase*fr_ref/(2π*fr_out))
```

divvar_flt

```
if { fr_out < 62.5} then { divvar_flt = 0}
if { fr_out >= 62.5 and fr_out < 135} then { divvar_flt = 1}
if { fr_out >= 135 and fr_out < 210} then { divvar_flt = 2}
if { fr_out >= 210 and fr_out < 340} then { divvar_flt = 3}
if { fr_out >= 340 and fr_out < 560} then { divvar_flt = 4}
if { fr_out >= 560 and fr_out <= 1000} then { divvar_flt = 5}
else divvar_flt = 7
```

div2_flt

```
if { fr_out > 2000 and fr_out < 2850} then { div2_flt = 0}
```

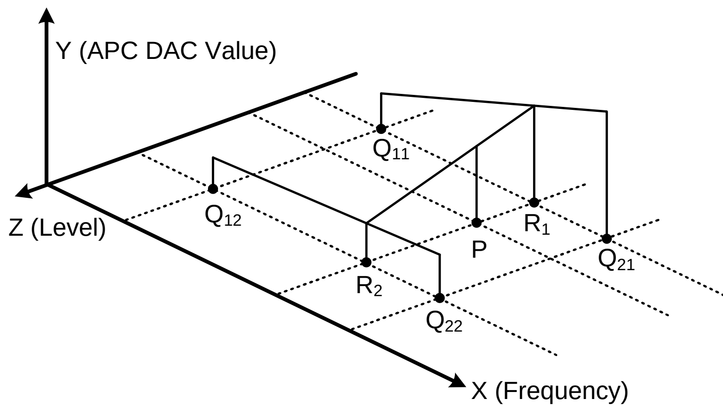

Figure 19: Bilinear interpolation

```
if { fr_out >= 2850 and fr_out <= 4000} then { div2_flt = 1}
else {div2_flt = x (i.e. doesn't matter)}
```

div4_flt

```
if { fr_out > 1000 and fr_out < 1500} then { div4_flt = 0}
if { fr_out >= 1500 and fr_out <= 2000} then { div4_flt = 1}
else {div2_flt = x (i.e. doesn't matter)}
```

poutbits

The calculation of poutbits is based on the calibration values (DAC values) residing on 2-dimensional rectilinear grid (frequency and level). To find DAC value (i.e. poutbits value) for the arbitrary frequency-level point, bilinear interpolation can be used. Calibration data is retrieved from the flash memory – table CTYPE=8, for more details see section 3.4. Suppose that we have four calibration points (Q_{11}, Q_{12}, Q_{21}, Q_{22}) around the point which DAC value we need to find (P), see fig 19. X axis means frequency in MHz, Z axis – output level in dBm, Y – DAC value. So we have $Q_{11} = (x_1, z_1)$, $Q_{12} = (x_1, z_2)$, $Q_{21} = (x_2, z_1)$, $Q_{22} = (x_2, z_2)$, $P = (x, z)$, $Y(Q_{11})$, $Y(Q_{12})$, $Y(Q_{21})$, $Y(Q_{22})$ – are calibration values, and we need to find $Y(P)$.

First we need to find Y values of auxiliary points R_1 and R_2, where $R_1 = (x, z_1)$ and $R_2 = (x, z_2)$:

$$Y(R_1) \approx \frac{x_2 - x}{x_2 - x_1} Y(Q_{11}) + \frac{x - x_1}{x_2 - x_1} Y(Q_{21})$$

$$Y(R_2) \approx \frac{x_2 - x}{x_2 - x_1} Y(Q_{12}) + \frac{x - x_1}{x_2 - x_1} Y(Q_{22})$$

Table 8: Turning on the internal power supply system at initialization process

C[7:0]	D7	D6	D5	D4	D3	D2	D1	D0
	–	–	–	dds_pwr_on	output_en	ref_out_en	ref_clk_sel	power_on
0x01	0	0	0	0	1	x	x	1

Table 9: Turning on the DDS power supply at initialization process

C[7:0]	D7	D6	D5	D4	D3	D2	D1	D0
	–	–	–	dds_pwr_on	output_en	ref_out_en	ref_clk_sel	power_on
0x01	0	0	0	1	1	x	x	1

Then we need to find interpolated value between auxiliary points R_1 and R_2:

$$Y(P) = \frac{z_2 - z}{z_2 - z_1} Y(R_1) + \frac{z - z_1}{z_2 - z_1} Y(R_2).$$

DAC maximum value (0x0FFF) corresponds to minimum output level, value 0x0000 – corresponds to maximum signal level at RF Out output.

3.2 Initialization

After power on the LNO synthesizer is in standby mode, i.e. internal power supply system is off, and DDS, APC DAC and other registers are in their default state. So just after power on you need to initialize it by the following procedure:

1. Set minimum output signal level by loading to SPI 0x200FFF.
2. Turn on the internal power supply by loading to SPI value shown in table 8. ref_out_en bit and ref_clk_sel should be set according to your needs, see table 6.
3. Turn on the DDS power supply by loading to SPI value shown in table 9. Turning on the DDS power supply as separate operation is required for all power supply voltages were settled before DDS power-on.
4. Reset the DDS – 0x10001201
5. Update DDS – 0x1100 (DDS IOUPDATE)

6. Initialize DDS:
   (a) 0x10000080 (write to DDS at address 0x0000)
   (b) 0x10001090 (write to DDS at address 0x0010)
   (c) 0x10040BFF (write to DDS at address 0x040B)
   (d) 0x10040C03 (write to DDS at address 0x040C)
7. Update DDS – 0x1100 (DDS IOUPDATE)

3.3 Setting Frequency, Phase and Level

It should be noted that the lower value of poutbits variable, the higher output signal power. It should be also taken into account that setting the same poutbits value but at different frequencies will result in totally different output power. For not to exceed the output level while changing frequency it's very important to apply the following procedure. First you should save old poutbits value into some temporary variable, e.g. poutbits_prev, then calculate new ftw and poutbits values for new frequency. If poutbits_prev is greater or equal to poutbits then set frequency first and then output level:

1. 0x1061AB[ftw] (setting frequency)
2. 0x1100 (DDS IOUPDATE)
3. 0x020[n_pow] (Divider register)
4. 0x03[“000”] [div2_flt] [div4_flt] [divvar_flt] (Filter register)
5. 0x200[poutbits] (setting output level)

If poutbits_prev is less than poutbits then set output level first and then frequency:

1. 0x200[poutbits] (setting output level)
2. 0x1061AB[ftw] (setting frequency)
3. 0x1100 (DDS IOUPDATE)
4. 0x020[n_pow] (Divider register)
5. 0x03[“000”] [div2_flt] [div4_flt] [divvar_flt] (Filter register)

Changing the phase doesn't require loading poutbits. So to set new phase of output signal you need to load the following:

1. 0x1061AD[ptw]
2. 0x1100 (DDS IOUPDATE)

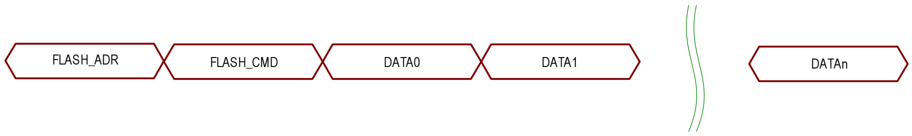

Figure 20: Structure of flash-memory data transfer

Table 10: Flash-memory commands

Command	Byte	Description
FLASH_CMD_READ	0x03	Read data
FLASH_CMD_WRITE	0x02	Write data
FLASH_CMD_WREN	0x06	Write data enable
FLASH_CMD_WRDI	0x04	Write data disable
FLASH_CMD_RDSR	0x05	Read status register
FLASH_CMD_WRSR	0x01	Write status register
FLASH_CMD_PE	0x42	Page erase
FLASH_CMD_SE	0xD8	Byte erase
FLASH_CMD_CE	0xC7	Erase all
FLASH_CMD_RDID	0xAB	Read ID and Turn on flash-memory Power Supply
FLASH_CMD_PDP	0xB9	Turn off flash-memory Power Supply

For the PLL to remain in locked condition, it's good idea to set new phase by several small phase shifts. Phase difference between old and new consecutive values should meet the following condition:

$$\frac{\Delta ptw}{2^{16}} \ll 1.$$

3.4 Flash Memory

LNO synthesizer has 1 Mbit (131072 bytes) Flash-memory. Its address space is within 0x00000 to 0x1FFFF range. Memory space is divided into pages, 256 bytes each. It is possible to work with separate bytes and with pages. Pages start at addresses 0xXXX00, where XXX is in range 000 to 1FF. The number of data bytes in packet can be different and depends on the actual memory command. General structure of the packet is shown in figure 20. First command byte, FLASH_ADR (0x70 – Flash-memory access, see table 2), is for CPLD multiplexer, second byte is command for the 25LC1024 memory, followed by data bytes if required. Flash-memory commands are listed in table 10.

3.4.1 FLASH_CMD_READ (0x03)

Data read command, see figure 21. First byte – FLASH_SPI access command for CPLD multiplexer (0x70), second – flash data read command FLASH_CMD_READ (0x03), then three bytes which contains data address ADR[2:0]. Actually it's

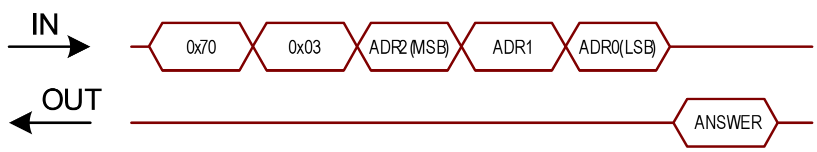

Figure 21: Flash-memory data read command

Figure 22: Flash-memory data write command

start address, so if you hold SS# signal in active position ("0") you will get data at next address for each CLK cycle. For example if we have the following data (3 bytes) 0xAA, 0xBB, 0xCC which are located at addresses 0x000006, 0x000007 and 0x000008 respectively. To retrieve these three bytes at once we need to send the following command: 0x0703000006000000. Last three zero bytes in this command are used to generate CLK cycles to retrieve three consecutive data bytes. As a response on MISO line we will obtain the following: 0xAABBCC.

3.4.2 FLASH_CMD_WRITE (0x02)

Data write command, see figure 22. First byte – FLASH_SPI access command for CPLD multiplexer (0x70), second – flash data write command FLASH_CMD_WRITE (0x02), then three address bytes ADR[2:0] and data bytes. First data byte will be written at ADR[2:0] address, second at ADR[2:0]+1 and so on, but all written data should reside inside the same page. This way maximum number of data bytes is 256 if ADR[2:0] points to the start of page. Otherwise if address ADR[2:0] plus number of bytes exceeds the address of end of page, some data of this page will be overwritten. This will result in lost of some data.

3.4.3 FLASH_CMD_WREN (0x06)

Write enable command, see figure 23. This command sets WEL bit to "1" in flash-memory status register. To enable any write or erase operation, this bit should be set to "1", otherwise these operations won't be performed. WEL bit is reset automatically to "0" after performing the following commands:

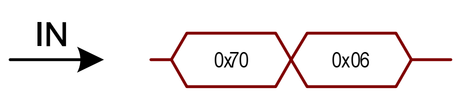

Figure 23: Flash-memory write enable command

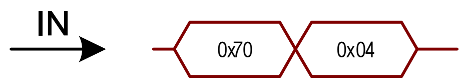

Figure 24: Flash-memory write/erase disable command

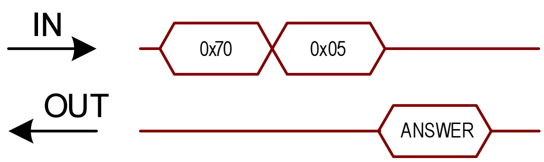

Figure 25: Flash-memory status register reading

- FLASH_CMD_WRDI
- FLASH_CMD_WRSR
- FLASH_CMD_WRITE
- FLASH_CMD_PE
- FLASH_CMD_SE
- FLASH_CMD_CE

After power on WEL bit is also in “0” state.

3.4.4 FLASH_CMD_WRDI (0x04)

This command disables write and erase operations by setting WEL bit to “0” in status register of flash-memory. Command is shown in figure 24.

3.4.5 FLASH_CMD_RDSR (0x05)

This command reads flash-memory status register, see figure 25. Table 11 shows the content of status register.

WIP – flash-memory busy. “1” – write operation in progress (is not finished). “0” – all operations are finished.

WEL – write enable. “0” – write/erase operation disabled, “1” – enabled.

Table 11: Flash-memory status register

	R7	R6	R5	R4	R3	R2	R1	R0
Read/Write	–	–	–	–	R/W	R/W	R	R
	X	X	X	X	BP1	BP0	WEL	WIP

Table 12: Flash-memory block protection

BP1	BP0	Protected area	Unprotected area
0	0	–	All address area (00000h-1FFFFh)
0	1	Upper 1/4 address space (18000h–1FFFFh)	Lower 3/4 address space (00000h-17FFFh)
1	0	Upper 1/2 address space (10000h-1FFFFh)	Lower 1/2 address space (00000h-0FFFFh)
1	1	All address space (00000h-1FFFFh)	–

Figure 26: Flash-memory write to status register

BP1, BP0 – memory block write protection. All address space is divided into 4 blocks and these bits can be used to control protection for write/erase operations for these blocks, see table 12.

To perform reading operation you need to send 0x700500.

3.4.6 FLASH_CMD_WRSR (0x01)

Write to status register, figure 26. Actually this command is used to set BP0 and BP1 bits, since WEL bit can be controlled via FLASH_CMD_WREN and FLASH_CMD_WRDI commands.

3.4.7 FLASH_CMD_PE (0x42)

Command erases content of a page which address is inserted in command, see figure 27.

3.4.8 FLASH_CMD_SE (0xD8)

Clears one byte, figure 28.

Figure 27: Flash-memory page erase command

Figure 28: Flash-memory byte erase command

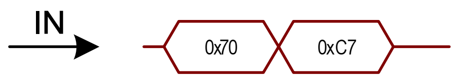

Figure 29: Flash-memory clear all command

3.4.9 FLASH_CMD_CE (0xC7)

Clears all memory, figure 29.

3.4.10 FLASH_CMD_RDID (0xAB)

This command turns on the memory power supply and reads ID number of the chip (ID=0x29), figure 30. To perform reading you should send 0x70AB00, and you get 0x29 in response.

3.4.11 FLASH_CMD_PDP (0xB9)

Turns off the memory chip power supply, see figure 31.

3.5 Memory address and data mapping

Address and data mapping is shown in table 13.

Table 13: Memory address and data mapping

Address	Variable/Value	Description
0x00	0xAA	CONFIGBLKSIG0
0x01	0xBB	CONFIGBLKSIG1
0x02	0xCC	CONFIGBLKSIG2
0x03	0xDD	CONFIGBLKSIG3
0x04	PID0	Product ID (LSB)

(continued on next page)

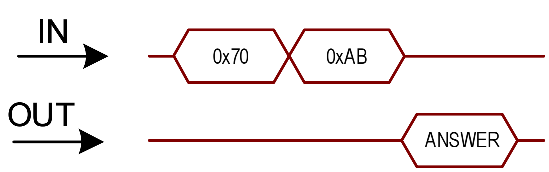

Figure 30: Flash-memory power on and read ID command

(continued Table 13, beginning on page 33)

Address	Variable/Value	Description
0x05	PID1	Product ID (MSB)
0x06	SID0	Software ID (LSB)
0x07	SID1	Software ID (MSB)
0x08	SN0	Serial Number (LSB)
0x09	SN1	Serial Number (MSB)
0x0A	LOT	Lot
0x0B	DY	Year of production
0x0C	DM	Month of of production
0x0D	DD	Day of production
0x10	REF_FR0	Reference frequency in Hz (LSB)
0x11	REF_FR1	Reference frequency in Hz
0x12	REF_FR2	Reference frequency in Hz
0x13	REF_FR3	Reference frequency in Hz (MSB)
0x14	DATA_SIZE0	Size of Data Block (LSB)
0x15	DATA_SIZE1	Size of Data Block
0x16	DATA_SIZE2	Size of Data Block
0x17	DATA_SIZE3	Size of Data Block (MSB)
0x18	FLASH_SIZE0	Flash-memory size (LSB) = 0x00
0x19	FLASH_SIZE1	Flash-memory size = 0x00
0x1A	FLASH_SIZE2	Flash-memory size = 0x00
0x1B	FLASH_SIZE3	Flash-memory size (MSB) = 0x02
0x00FE	CRC0	CRC 16-bit (LSB)
0x00FF	CRC1	CRC 16-bit (MSB)
0x0100 (0xXXX00)	0x99	TABLESIG0
0x0101	0x88	TABLESIG1
0x0102	0x77	TABLESIG2
0x0103	0x66	TABLESIG3
0x0104	CTYPE	Table type (characteristic type)
0x0105	XVALUE	Type of X axis values
0x0106	YVALUE	Type of Y axis values
0x0107	ZVALUE	Type of Z axis values
0x0108	ZCOUNT0	Number of points on Z axis (LSB)
0x0109	ZCOUNT1	Number of points on Z axis
0x010A	ZCOUNT2	Number of points on Z axis
0x010B	ZCOUNT3	Number of points on Z axis (MSB)
0x010C	XYCOUNT0	Number of points on X axis (LSB)
0x010D	XYCOUNT1	Number of points on X axis

(continued on next page)

(continued Table 13, beginning on page 33)

Address	Variable/Value	Description
0x010E	XYCOUNT2	Number of points on X axis
0x010F	XYCOUNT3	Number of points on X axis (MSB)
0x0110	0x33	XROWSIG0
0x0111	0x22	XROWSIG1
0x0112	X_MULT	X multiplier
0x0113	–	Not used
0x0114	X_L0	X grid value 0 (LSB)
0x0115	X_H0	X grid value 0 (MSB)
0x0116	X_L1	X grid value 1 (LSB)
0x0117	X_H1	X grid value 1 (MSB)
0x0118	X_L2	X grid value 2 (LSB)
0x0119	X_H2	X grid value 2 (MSB)
0x011A	X_L3	X grid value 3 (LSB)
0x011B	X_H3	X grid value 3 (MSB)
0x011C	X_L4	X grid value 4 (LSB)
0x011D	X_H4	X grid value 4 (MSB)
	0x55	ZROWSIG0
	0x44	ZROWSIG1
	Z_L0	Z grid value 0 (LSB)
	Z_H0	Z grid value 0 (MSB)
	Y_L0	Y value for X grid value 0 and Z grid value 0 (LSB)
	Y_H0	Y value for X grid value 0 and Z grid value 0 (MSB)
	Y_L1	Y value for X grid value 1 and Z grid value 0 (LSB)
	Y_H1	Y value for X grid value 1 and Z grid value 0 (MSB)
	Y_L2	Y value for X grid value 2 and Z grid value 0 (LSB)
	Y_H2	Y value for X grid value 2 and Z grid value 0 (MSB)
	0x55	ZROWSIG0
	0x44	ZROWSIG1
	Z_L1	Z grid value 1 (LSB)
	Z_H1	Z grid value 1 (MSB)
	Y_L0	Y value for X grid value 0 and Z grid value 1 (LSB)

(continued on next page)

(continued Table 13, beginning on page 33)

Address	Variable/Value	Description
	Y_H0	Y value for X grid value 0 and Z grid value 1 (MSB)
	Y_L1	Y value for X grid value 1 and Z grid value 1 (LSB)
	Y_H1	Y value for X grid value 1 and Z grid value 1 (MSB)
	Y_L2	Y value for X grid value 2 and Z grid value 1 (LSB)
	Y_H2	Y value for X grid value 2 and Z grid value 1 (MSB)
0xXXXFE	CRC0	CRC 16-bit (LSB)
0xXXXFF	CRC1	CRC 16-bit (MSB)

.

Memory space is divided into two blocks: configuration block (00000h-000FFh) and data block. Data of each block is supplied with check sum (CRC). CRC for configuration block is placed at 000FEh-000FFh, CRC for data block is placed at the end of last page of data block. The address of the CRC word for data block can be found as DATASIZE[3:0]+0x100. CRC calculation algorithm is standard CCITT, 16-bit, polynomial A001h, starting with FFFFh. CRC is calculated for all data (even for unused space) that is multiple to memory page, i.e. for configuration block – from 0h to FD, and for data block – from 100h to DATASIZE[3:0]+0xFF inclusively.

3.5.1 Configuration Block

Configuration block occupies 256 bytes, from 00000h to 000FFh. First four bytes are signature of the configuration block 0xDDCCBBAA.

PID[1:0] – product ID (unsigned integer 0 to 65535), first (5-digit in DEC) number in full serial number of particular device. This ID corresponds to the part number of the device. For example for the following partnumber LNO-HP35M-RF the ID will be 04608, while full serial number can be 04608-3021-014.

SID[1:0] – software ID, it can be treated as version of set of calibration data tables in data block.

SN[1:0] – serial number (unsigned integer 0 to 999), last number (3-digit in DEC) in full serial number of particular device. It's 014 for the example above.

LOT – lot number (unsigned integer 0 to 9).

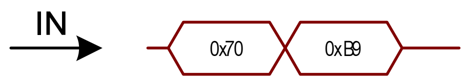

Figure 31: Flash-memory power off command

Table 14: Type of Data Table (CTYPE)

CTYPE	Description
0	Type is undefined
0x08	Calibration data for APC system
0x0A	Calibration data for DDS SpurKiller for 147 MHz reference. Used to suppress some low-order spurs
0x0B	Calibration data for DDS SpurKiller for 150 MHz reference. Used to suppress some low-order spurs
0x0C	Calibration data for switching between 147 and 150 MHz reference to avoid spurs. Used in conjunction with LNO-REF-20M-RF 147/150 MHz dual frequency synthesizer

DY – year of production starting from 1970 (unsigned integer). Thus real year is DY+1970. Last digit of the year is coded as first digit in second digit group of full serial number, i.e. 3 in the example above.

DD – day of production (unsigned integer 1 to 31).

FR_REF[3:0] – reference frequency value expressed in Hz (unsigned integer, normally 147 000 000). Please note, that before using it in calculations you need to express it in MHz, i.e. you should to change its type to float and then divide this value by 1 000 000.

DATA_SIZE[3:0] – size of data block excluding its CRC (2 bytes).

FLASH_SIZE[3:0] – size of embedded flash-memory in bytes.

Unused bytes in configuration block are filled with zeroes.

3.5.2 Data block

Data block may contain one or more calibration data tables. Each table starts with new page. The structure of the table is described below.

First 4 bytes (TABLESIG[3:0]) of the table are signature 0x66778899.

Type of data table (CTYPE) The signature is followed by type of data table (CTYPE), the description is shown in table 14.

Tables with different types can be placed in memory in any order. Table with CTYPE=0x08 is required, others are optional.

For CTYPE=0x08 table X-data are frequency grid, normally it's not regular: 10 to 100 MHz with 1 MHz step, 100 MHz to 1 GHz with 10 MHz step, and 1

Table 15: X,Y and Z-data format (XVALUE, YVALUE, ZVALUE)

{X|Y|Z}VALUE	Description
0	Type is not defined
1	2-byte integer
2	2D.2 - fixed point (2 digits after point) value, 2-byte. To convert it to float you should divide it by 100.0

to 8 GHz with 25 MHz step. X-data values are expressed in MHz. Z-data are power level grid values, normally it's regular: from -10 to +26 dBm with 2 dB step. Values are in dBm. Y-data are calibrated DAC values (2-byte unsigned integer) which correspond to the appropriate frequency (X-data value in MHz) and level (Z-data value in dBm). Y-data values should be decoded as follows:

- 0xFFFF – calibration point is not valid, it can't be used in interpolation algorithm.
- 0xFFFF > Y-data > 0x7FFF – calibration point can be used in interpolation algorithm, but its precision is not guaranteed.
- Y-data ≤ 0x7FFF – calibration point is valid and can be used in interpolation algorithm.

X,Y,Z-data type (XVALUE, YVALUE, ZVALUE) Table 15 shows types of X, Y and Z-data values.

Number of Z-data points (ZCOUNT[3:0]) 4-byte integer. It is the number of Z-axis grid points.

Number of X and Y-data points per row (XYCOUNT[3:0]) 4-byte integer. It is the number of X-axis grid points, and it is equal to number of Y-data points per row (i.e. per one Z-data value).

X-data value multiplier (X_MULT) Actually it is commonly used for frequency data. If this value equals 6 then X-data (frequency) is expressed in MHz, if 3 – in kHz, 0 – in Hz.

X_H[XYCOUNT-1:0] and X_L[XYCOUNT-1:0] MSB and LSB bytes of X-axis grid values. Usually they are frequency values.

ZROWSIG[1:0] Each data row starts with signature ZROWSIG[1:0]=0x4455.

Z_H[ZCOUNT-1:0] and Z_L[ZCOUNT-1:0] Z_H[N] and Z_L[N] are MSB and LSB bytes of Z-value for the following by Y-data Y_H[XYCOUNT-1:0,N] and Y_L[XYCOUNT-1:0,N].

Y_H[XYCOUNT-1:0] and Y_L[XYCOUNT-1:0] Z-value is followed by Y-data row.