

Tecnoroll BMB s.r.l.

Via Olona 9, 21040 Vedano Olona - VA

www.tecnoroll.it

FLEXMod 3 modules

Quick start guide

- 1. The use of this product in conformity with the DVB Standard is prohibited in any case, unless the license of patents applicable to the DVB patent group is granted.
- 2. It's the purchaser's duty to check which patents with potentially payable author's rights are used in his/her own final project and act accordingly with the bodies in charge.
- 3. The contents of this manual are subject to change without notice.
- 4. Of the specifications described in this manual, the functions and performance the supply of which is assured with this product are only those that can be confirmed with the separate Basic Firmware attached.
- 5. When the product or information described in this manual is used, our company does not guarantee or grant any license of the intellectual property rights and other rights of our company or of third parties. If any problems concerning the ownership rights of third parties attributable to the above use occur, our company will not be responsible for them.
- 6. Since the product described in this manual was not developed specially for equipment related to human life, those who intend to use it for equipment whose failure or malfunction may directly threaten human life or endanger the human body or for special applications requiring high reliability (such as aerospace, nuclear power control, submarine repeaters, travel control, and medical equipment for life support) must consult our sales responsible personnel in advance. Our company will not be responsible for any damages that may occur due to such use without consulting us.
- 7. Although our company is endeavoring to improve quality and enhance reliability, semiconductor products will fail with a certain probability. Users are requested to pay due attention to safe design techniques such as redundant design, preventive measures against the spreading of fire, over-current, and malfunctions, so that failure of the products described in this manual will not result in accidents leading to injury or death, fire, or social damage.

8. The product described in this manual is not designed to be radiation-resistant.

9. This board is just an 'electronic component' that cannot work in a standalone mode, therefore all the tests and qualifications for electromagnetic compatibility, all the CE certifications and any other regulations concerning the final product implemented by the customer are to be carried out by the customer himself referring to the final product for which this electronic component has been used.

Revisions history:

Rev A, January, 28 2013, dbrini@tecnoroll.it

First release

Index

Revisions history
FLEXMod 35
The bootloader
Memory management
Setup of serial communication
Use of the Bootloader10
Help11
TAB key use12
Selecting the bitstream to be launched
Loading a bitstream
Addendum for IP Providers
Known issue

FLEXMod 3 is a multi-standard modulation platform which can combine high performance with great flexibility and many types of use, either as a finished product or as hardware platform where the customer can implement his own Ips.

Tecnoroll's module is internally organized as follows:



The bootloader

The FLEXMod 3 chipset integrates an advanced bootloader allowing to quickly load and manage several different bitstreams into the system flash memory. This is achieved thanks to an internal command interpreter accessible through the UART port and a standard terminal software such as Windows HyperTerminal.

The bootloader also takes care of the multi-bank flash memories as if they were a single virtual disk, thus allowing the user to bypass the FPGA limitation of 16MB addressing range. Thanks to this feature, the user can get access to the expanded memory in a transparent way, without having to develop any other bootloader bitstream or software. This feature allows to manage up to 16 bitstreams, which can be automatically started at power-up and can be uploaded and downloaded through a standard terminal software for easy upgrading on the field.

Memory management

Whenever switched on, the FPGA loads the Tecnoroll bootloader from the flash memory.



Then, the bootloader makes the other flash memories visible and manages the serial user interface.



The bootloader also links the memories together, so that they are visible as a single virtual hard disk. Example:



The partition occupied by the bootloader (2.5 MB) cannot be changed or overwritten, and is automatically loaded into the FPGA at power up.

The swap area size can be configured by the user and is the partition where the bootloader automatically stores the bitstreams that can't be loaded directly by the FPGA because of the 16MB addressing limitations. When the FPGA loads the bitstream, one of the following 3 scenarioes can occur (shown as an example in the diagram above), and the bootloader automatically recognizes it and takes action accordingly:

- **Bitstream 1**: fully resident in the memory bank directly managed by the FPGA (boot memory); it is loaded directly from its residence partition.
- **Bistream 2**: resident across two different memory banks; in order to be run, it must be loaded into the swap area.
- **Bitstream 3 or 4**: fully resident in a memory bank which is not directly managed by the FPGA; in order to be run, it must be loaded into the swap area.

A maximum of 16 bitstreams can be saved inside the flash memory or, at least, as far as there is free space. Please make sure that the swap area is big enough to contain the user bistreams.

If needed, the default 4.5 MBytes swap area size can be set via the FlashFormat command.

Setup of serial communication

The bootloader manages the user interface via serial communication. You can use any terminal to interact with it; for example, you can use Hyper-terminal, which is already present in Windows XP, and configure it according to the following scheme:

Proprietà - COM3	? ×
Impostazioni della porta	
Bit per secondo: 115200	.
Bit di dati: 8	-
Parità: Nessuno	-
Bit di <u>s</u> top: 1	•
Controllo di flusso: Nessuno	-
<u>Bip</u>	ristina
OK Annulla	<u>A</u> pplica

Please, also remember to set the ANSI emulation in the "settings" tab of the terminal connection properties:

Proprietà - FlexMod

? ×

Connect To Settings
Function, arrow, and ctrl keys act as
Backspace key sends
Emulation:
ANSI Terminal <u>S</u> etup <u>Co</u> lors
Telnet terminal ID: ANSI
Backscroll buffer lines: 500
Play sound when connecting or disconnecting
Allow remote host initiated file transfers Exit program upon disconnecting
Input Translation ASCII Setup
UK Annulla

In order to optimize the upload speed of a desired bitstream into the flash memory, set the COM port latency to 1mS (shown below) and set the FLEXMod baudrate to 921600bit/s.



Advanced Settings for COM5		? ×
COM Port Number: USB Transfer Sizes Select lower settings to correct performance problems at low b Select higher settings for faster performance. Receive (Bytes): Transmit (Bytes): 4096 •	▼ Daud rates.	OK Cancel Defaults
BM Options	Miscellaneous Options	
Select lower settings to correct response problems.	Serial Enumerator	
Latency Timer (msec):	Serial Printer	
	Cancel If Power Off	
Timeouts	Event On Surprise Removal	
Minimum Read Timeout (msec):	Set RTS On Close	
Minimum Write Timeout (msec):	Disable Modem Ctrl At Startup	

Use of the bootloader

When you power up the FLEXMod module, when you see the "waiting" dots appearing, you can stop the booting process and get access to the bootloader console by pressing any key within the given time interval (as shown in the screenshots below). The time interval can be set through the SetBootWait command (default time is 1 sec).



A successful access to the bootloader command console shows the following message:



Help

Typing Help followed by the <ENTER> key gives you the following result:

FLEXMod 3 - HyperTerminal								- D X
File Edit View Call Transfer	lelp							
In order to correct Emulation = CodePage = Background color = Foreground color =	ly use the B ANSI 437 Blue Gray	 loot Mana	ager p]	lease	set the	terminal	as foll	 ows:
BACKSPACE = CR+LF = Press any Key to co	Destructive Disabled (CR ntinue	0nly)						
Connected 00:12:07 ANSI	115200 8-N-1	SCROLL	CAPS	NUM	Capture	Print echo		

🇞 FLEXMod 3 - HyperTerminal	_
Eile Edit Yiew Call Iransfer Help □ <th></th>	
Available commands: Baud -> Set serial port Baud Rate FlashFormat -> Erase the FPGA boot Flash memory except for the BootManager LoadBitStream -> Load a bitstream into the Flash memory (Xmodem or Xmodem 1K) SaveBitStream -> Backup to PC a bitstream from the Flash memory (Xmodem 1K) EraseBitStream -> Change the name of the selected bitstream ReBoot -> System Reboot BootSelect -> Bitstream Boot selector SetBootWait -> Time to wait for a key at powerup before loading a Bitstream GetSN -> Return the hardware Serial Number GetTemp -> Return the hardware name and the firmware version -> Please note: after typing a command the TAB key gives you the help for that specific command. >	4
Connected 00:14:37 ANSI 115200 8-N-1 SCROLL CAPS NUM Capture Print echo	11.

<TAB> key use

The <TAB> key is used to get specific help on the various commands: if pressed after writing only a part of a command, it lists all the commands which start with that written part, such as in the following screenshot.

FLEXMod 3 - HyperTerminal						
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>C</u> all <u>T</u> ransfer <u>H</u> elp						
🗅 😂 📨 🐉 🗈 🎦 😭						
*** FLEXMod 3.0 FPG ** Copyright 2011-201 Type "HELP" for comma >re Available commands: RenameBitstream Re >Re_	A Boot Load 3 By Tecnor ************ nds. boot	der V 2.0 roll BMB	****** sr] ** *****			
Connected 00:19:38 ANSI	115200 8-N-1	SCROLL	CAPS NUM	Capture	Print echo	//

If used after writing the whole command, it provides specific help on the correct syntax and on any additional options of the command itself.



Selecting the bitstream to be launched

One of the most important operations you can perform thanks to the bootloader is selecting the bitstream to be loaded when the FLEXMod is powered up. The relevant command is BootSelect followed by <ENTER>. The following table appears:

FLEXMo	d 3 - Hyperl	Ferminal				
<u>F</u> ile <u>E</u> dit	<u>V</u> iew <u>C</u> a	ll <u>T</u> ransfer <u>H</u> elp)			
) 🏅 🗈					_
Nn	Туре	Start	Stop	Size	Bitstream Name	^
Ø SWP	Boot Boot	00000000 00280000	0027FFFF 006FFFFF	2560 Kb 4608 Kb	Boot Manager Bitstream Swap Area (9)	
1* 2 3 4	Boot Boot 	00700000 00B00000	00AFFFFF 00E7FFFF	4096 Kb 3584 Kb Kb Kb	DVB-T IP-ASI ASI-IP	
5 6 7 8 9	Swap Swap	00E80000 012C0000	012BFFFF 016BFFFF	KD Kb 4352 Kb 4096 Kb	DVB-T 4CH DEMO DVB-T2 DEMO	
10 11 12 13 14 15				Kb Kb Kb Kb Kb		
16 Free Pleas	Flash se sele	Memory: 25 ct the Bit	.856 KByte stream to	Kb es boot from	::[00]	• 11
Connected	00:28:07	ANSI	115200 8-N-1	SCROLL	APS NUM Capture Print echo	1

The *"Type"* column indicates whether the bitstream can be uploaded directly by the FPGA or whether it must pass though the swap area.

The "Start" and "Stop" columns indicate the start and stop addresses of the bitstream.

The "Size" column displays the total size occupied by the bitstream inside the flash memory.

The last column is for the bitstream name.

Note: the number of the bitstream currently present in the swap area is indicated between parentheses in the swap area row.

An asterisk beside the bitstream number (first column) indicates that this bitstream is the one selected for automatic startup when the module is powered up.

In order to select the bitstream to be launched, simply enter the corresponding number followed by the <ENTER> key. The user will be then asked whether the bitstream should be started automatically or not when powering up the device.

Note: If the FLEXMod module is managed through a string of commands from a microcontroller without passing through the ANSI terminal, it is highly recommended not to set any automatic bitstream startup options. In this case the bootloader gives you direct access to the command interpreter without needing to manually stop the booting procedure. When the booting procedure is complete and you have accessed the command interpreter, simply send the BootSelect xx string (where xx is the number of bitstream that you want to start) which in this case is only a temporary choice.

Loading a bitstream

It is possible to load a bitstream into the FLASH memory by using the LoadBitStream command.

🌯 FLEXMod 3 - HyperTerminal	
<u>File Edit View Call Iransfer H</u> elp	

** FLEXMod 3.0 FPGA Boot Loader V 2.04 **	
** Copyright 2011-2013 By Tecnoroll BMB srl **	
Tune "HELP" for commands	
>LoadBitStream	
LoadBitstream : Upgrade the FPGA boot_Flash_memory.	
Parameters : Password, BitStr.Nr, Image Name, optional parameters	
Upt.Param. : U=LKSUM mode I=LKL mode, U=Normal mode I=Pre-HLK mode Note :: Use Y-modem or Ymodem-1K protocol to upled the EPCO Ritstroa	
Note : The password is 741852.	"
>LoadBitstream 741852, 3, MyBitStream	
	-
Connected 01:20:34 ANSI 115200 8-N-1 SCROLL CAPS NUM Capture Print echo	11.

In the above screenshot you can see the command help with all the options and the typical syntax. Usually, only three parameters are enough to complete the process: the password, the location of the bitstream within the table and the bitstream name. The default transfer protocol is Xmodem-1K with CRC; if you want to change the upload mode, just add the optional corresponding parameters. Once the <ENTER> key is pressed, you will see a sequence of characters. This means that the module is ready to receive the bitstream to be loaded.



Select the "Transfer" and "Send File" option from the HyperTerminal menu.



Select the file you want to upload and set the "protocol" to 1K Xmodem, then press the "Send" button.

S FLEXMod 3 - HyperTerminal	- O X
<pre>FLEXMod 3 - HyperTerminal File Edit View Call Iransfer Help File Edit View Call Iransfer Iransfer Help File Edit View Call Iransfer Irans</pre>	
Connected 01:27:37 ANSI 11:2200 8-N-1 SCROLL CAPS NUM Capture Print echo	//

If everything is working fine, a window showing the upload progress should open.

FLEXMod 3 - HyperTerminal						•	
Eile Edit View Call Iransfer He	lp						
**************************************	GA Boot Loader 13 By Tecnorol	V 2.04 1 BMB srl	*** ** **				
Tupo "HELP" fo 1K Xmode	m file send for FLEXMod	3					
>LoadBitstream Start = 016C00 Sending:	C:\Users\Diego\Docume	nts\Documentazion	i TRoll\Bootl	oader\MyBitSt	rez		
Packet:	159 Error chee	king: CRC					
Retries:	0 Total re	tries: 0					
Last error	:						
File:	111		128K of	1583K			
Elapsed:	00:00:04 Remai	ning: 00:00:45	Throughp	ut: 32768 cps	5		
		C	Cancel	cps/bps			
Connected 01:27:37 ANSI	115200 8-N-1 SCI	ROLL CAPS	NUM	Capture	Print echo		

When the file has been uploaded, the module calculates the CRC.

FLEXMod 3 - HyperTerminal								x
<u>File Edit V</u> iew <u>C</u> all <u>T</u> ransfer <u>H</u> elp								
D 🗃 📨 🍒 🗈 🎦 😭								
>LoadBitstream 741852 Start = 016C0000 Sto CCCCCCCCC File type : Xilinx Bi Blocks : 1584 Flash ERR : 0 File CRC : 28F1CC77 Flash CRC :	, 3, MyBit p = 02B7FF tstream	Stream FF						*
Connected 00:01:21 ANSI	921600 8-N-1	SCROLL	CAPS	NUM	Capture	Print echo		1.

After a few seconds, a report on the use of the flash memory is also displayed. The bitstream upload is now complete.



The loaded bitstream is now available in the table:

Nr	Туре	Start	Stop	Size	Bitstream Name	
0 SWP	Boot Boot	00000000 00280000	0027FFFF 006FFFFF	2560 Kb 4608 Kb	Boot Manager Bitstream Swap Area (9)	
1* 2 3 4 5 6	Boot Boot Swap	00700000 00B00000 016C0000	00AFFFFF 00E7FFFF 0187FFFF	4096 Kb 3584 Kb 1792 Kb Kb Kb Kb	DVB-T IP-ASI ASI-IP MyBitStream	
8 9 10 11 12 13 14 15 16	Swap Swap	00E80000 012C0000	0128FFFF 0168FFFF	4352 Kb 4096 Kb Kb Kb Kb Kb Kb Kb	DVB-T 4CH DEMO DVB-T2 DEMO	

Addendum for IP Providers - Generating the bitstream file to be loaded into the FLEXMod module

The following screenshots show the correct way to generate the bitstream file to be loaded into the flash memory of the FLEXMod module.

In the Xilinx ISE Project Navigator, inside the "Processes" panel, select "Generate Programming File" and then "Process Properties".

ISE Project Navigator (P.49d) - C:\Xilinx_S6A_DSP\Xilinx_ISE\Xilinx_ISE.xif	e - [Design Summary (Program	iming File Generated)]					_ 0	×	
∑ File Edit View Project Source Process Tools Window I	ayout Help							- 8 ×	
	18 / 🖻 🔀 🕞 🗉	l 🖻 🥬 🗞 🕨 🗵	* 🖓						
Design ↔ □ 日 × 📊 🖨 Design Overv	iew 🔺	FLEXMod30 Project Status (01/29/2013 - 12:04:17)							
View:	roperties	Project File:	Xilinx_ISE.xise	Parser Errors:		No Errors			
🚛 Hierarchy 🦳 📄 Mode	ule Level Utilization	Module Name:	FLEXMod30	Implementation	State:	Programming	File Generated	E	
Imm	ig Constraints at Report E	Target Device:	xc6slx150-3csg484	• Errors:		No Errors			
E Clock	(Report	Product Version:	ISE 14.4	• Warnings		916 Warnings (2 new)			
*	Timing	Design Goal:	Balanced	 Routing Results: 		All Signals Completely Routed			
No Processes Purping Sec. 1 - Parse	rnings r Messages	Design Strategy:	Xilinx Default (unlocked)	• Timing Co	Timing Constraints: <u>All Constraints Met</u>				
Synth	resis Messages	Environment:	System Settings	• Final Timi	• Final Timing Score: 0 (Timing Report)				
Processes: FLEXMod30 - Behavioural	lation Messages Messages								
Design Summary/Reports	and Route Messages		Device Utiliz	ation Summary	Summary				
🕅 🖶 🎽 User Constraints	ig Messages	Slice Logic Utilization		Used	Available	Utilization	Note(s)		
CAA Synthesize - XST All Implement Decign All Implement Decign	plementation Messages	Number of Slice Registers		2,312	184,304	1%	,		
Generate Programming File	urts	Number used as Flip Flops	2,311						
Configure Target Device 11 Run Analyze Design Ning Chi P. P.	isage Filtering	Number used as Latches	0						
Refun	immary Contents	Number used as Latch-thr	0						
Bit Stop	c Report	Number used as AND/OR	1						
View Text Report	ings	Number of Slice LUTs Number used as logic			92,152	3%			
Force Process Up-to-Date	5				92,152	2%	1		
	-	Number using O6 outpu	it only	1,641				-	
Start Cosign Cos	Design Summary (Programming F	File Generated)	×						
Console								+□♂×	
am_RAM75_RAMD_O> is 🙀 况 Process Properties	not drive any load	d pins in							
the design.									
A WARNING: PhysDesignRules: 567 - The signal <inst avr="" controller="" inst="" p="" port="" rs232="" rx<=""></inst>	fifo/BU2/U0/grf.rf/mf	.em/qdm.dm/Mr							
am_RAM80_RAMD_O> is incomplete. The signal do	les not drive any load	d pins in							
the design.									
Process "Generate Programming File" completed su	accessfully							_	
< III								•	
Console 🔇 Errors 🔬 Warnings 🕷 Find in Files Results									
Edit the properties for the highlighted process									
🐵 🤛 🛷					IT 🔺	No 🖧 🛱 🗉	12:07 11 💊 29/01/2	013	

The "Process Properties - General Options" window opens, and we suggest you tick the "Enable Bitstream Compression" option to reduce the total dimensions of the file.



In the same window you should also select the "Configuration Options" and set the "Configuration rate" to

22MHz in order to speed up the booting process.

gory	Switch Name	Property Name	Value		
General Options	-g ConfigRate:	Configuration Rate	22		
Startup Options	-g ProgPin:	Configuration Pin Program	Pull Up		
Readback Options	-g DonePin:	Configuration Pin Done	Pull Up		
Encryption Options	-g TckPin:	JTAG Pin TCK	Pull Up		
suspenu/ wake Options	-g TdiPin:	JTAG Pin TDI	Pull Up		
	-g TdoPin:	JTAG Pin TDO	Pull Up		
	-g TmsPin:	JTAG Pin TMS	Pull Up		
	-g UnusedPin:	Unused IOB Pins	Pull Down		
	-g UserID:	UserID Code (8 Digit Hexadecimal)	0xFFFFFFFF		
	-g ExtMasterCclk_en:	Enable External Master Clock			
	-g ExtMasterCclk_divide:	Setup External Master Clock Division	1		
	-g SPI_buswidth:	Set SPI Configuration Bus Width	1		
	-g TIMER_CFG:	Watchdog Timer Value	0xFFFF		
		Place MultiBoot Settings into Bitstream			
	-g next_config_reboot:	-g next_config_reboot: MultiBoot: Insert IPROG CMD in the Bitfile			
	-g next_config_addr:	MultiBoot: Starting Address for Next Configuration	0x0000000		
	-g next_config_new_mode	MultiBoot: Use New Mode for Next Configuration	V		
	-g next_config_boot_mode:	MultiBoot: Next Configuration Mode	001		
	-g golden_config_addr:	MultiBoot: Starting Address for Golden Configuration	0x0000000		
	-g failsafe_user:	MultiBoot: User-Defined Register for Failsafe Scheme	0x0000		
		Property display level: Advanced 💌 🗹 Display	switch names Defaul		

In *"Process Configurations"*, please select the *"Startup Options"* and make sure that the settings match those shown in the <u>following screenshot:</u>

ategory	Switch Name	Property Name	Value
General Options	-g StartUpClk:	FPGA Start-Up Clock	CCLK
- Startup Options	-g DonePipe:	Enable Internal Done Pipe	
Readback Options	-g DONE_cycle:	Done (Output Events)	Default (4)
Encryption Options Suspend/Wake Options	-g GTS_cycle:	Enable Outputs (Output Events)	Default (5)
Suspena, make options	-g GWE_cycle:	Release Write Enable (Output Events)	Default (6)
	-g LCK_cycle:	Wait for DCM and PLL Lock (Output Events)	Default (NoWait)
	-g DriveDone:	Drive Done Pin High	

Once you have generated the **.bit** file in ISE, use the ISE iMPACT software to generate the final bitstream. In the ISE iMPACT window, please select "*Create PROM File*" from the "*iMPACT Flows panel*".

BE IMPACT (P.49d)		
<u>File Edit View Operations Output Debug</u>	<u>W</u> indow <u>H</u> elp	
🗋 🖻 🎉 🛛 🕾 🗉 🖉 🖉		
iMPACT Flows	↔□₽×	
Boundary Scan		
Create PROM File (PROM File Formatter)		
🗄 📄 WebTalk Data		
iMPACT Processes	↔□₽×	
Console		↔□₽×
		*
		+
Console 🚯 Errors 🔥 Warnings		4

In the "*PROM File Formatter*" window, select "*Generic Parallel PROM*" (Step 1), then set "*16M* [16777216]" in the multiple-choice window (Step 2), and enter the name you want to give to the file in the "*Output File Name*" box. Don't forget to set the "*File Format*" to "*BIN (Swap Bits OFF)*" (Step 3).

Step 1.	Select Storage Target	Step 2.	Add Storage Device(s)		Step 3.	Enter Data
Storage Device 	Type : PROM FPGA JAN re Single FPGA re MultiBoot FPGA re MultiBoot FPGA re MultiBoot FPGA re from Paralleled PROMs allel PROM	Parallel PROM Add Storage 10M [167772	(Bytes) 16M [16777216]	•	General File Detail Checksum Fill Value Output File Name MyBitStre Outpot File Location Filesh/PROM File Property File Format Loading Direction Number Of Revisions Revision 0 Start Address Add Non-Configuration Data	Value eam UP UP 1 0 Files No
Description:	,					
In this step, you • Checks • Output • Output • File For	u will enter information to assist in setting up sum Fill Value: When data is insufficient to t File Name: This allows you to specify the t File Location: This allows you to specify is mate DROM files can be generated in any u	and generating a PR fill the entire memor base name of the file the directory in which aumber of industry st	ROM file for the targeted storage device and y of a PROM, the value specified here is use to which your PROM data will be written the file named above will be created tandard formate. Depending on the PROM fil	d mode. ed to ca	alculate the checksum of the un	used portions.

Now you can proceed to generate the BIN file, which can now be uploaded into the module through the procedure described above (see "Loading a Bitstream").

Known issue

The presence of low power spurious signals in the spectrum output of the FLEXMod 3 starter kit is already known and is not due to any defects or malfunctions in the FLEXMod module.

The FLEXMod 3 chipset can work either with an internal clock source or with an external one. In order to obtain the best performance also with the DVB-T2 modulator, we have decided to mount an external oscillator (not envisaged in the original design) on the starter kit through an additional patch.

Although this patch, which is integrated in the current version of the starter kit, causes some spurious signals, these are in any case very low and have no effect whatsoever on the use of the starter kit itself.

In future versions of the starter kit, the additional oscillator will be integrated into the circuit, thus finally solving this little issue.

If the user designs his own PCB and embeds this oscillator correctly (if needed), he will obtain a much better output spectrum than that of the current version of our starter kit.



In the picture above you can see the output spectrum of the FLEXMod module with the external oscillator. The visible spurious signals are the harmonics of the reference 20 MHz oscillator mounted on the patch board.

Here below you can see the patch board with the oscillator:



Agilent Spe	ectrum A	.nalyzer - Sw	rept SA									
L <mark>XI</mark>	F	RF 50 Ω	AC AC		SE	NSE:INT		ALIGN AUTO	12:43:24 A	M Mar 26, 2012	A4	topustion
Input N	lech	Atten 6	dB	PNO: Fast 😱	Trig: Free	Run	Avg Type Avg Hold:	: Pwr(RMS) >10/10	TRAC TYP	^E 123456 E A WAWAA	A	tenuation
				IFGain:Low	#Atten: 6	αB						Mech Atten
											Auto	Man
10 dB/div	/ R(ef -10.00	dBm								riato	man
												Enable
												Elec Atten
-20.0											On	Off
	ſ											
-30.0												
												Elec Atten
-40.0												0 dB
-50.0												
-60.0												
-70.0												
-80.0												
-90.0	يا لي			<u>k </u>							Mech	n Atten Step
		" handruck	سالا كياسا		- I I was	healach		harder	-L		<u>2dB</u>	10dB
100												
-100												
											M	ax Mixer Lvl
Center	500.0	MHz	^						Span 9	00.0 MHz		-10.00 dBm
#Res B	W 100) kHz		#VBW	300 Hz*			Sweep	30.2 s (1001 pts)		
	ianmor	t Complete	od					STATUS				
Al Al	gimer	ir complete	eu					314105				

In the above picture, you can see the output spectrum of the FLEXMod module without the external oscillator, or with an external oscillator properly implemented on the PCB.