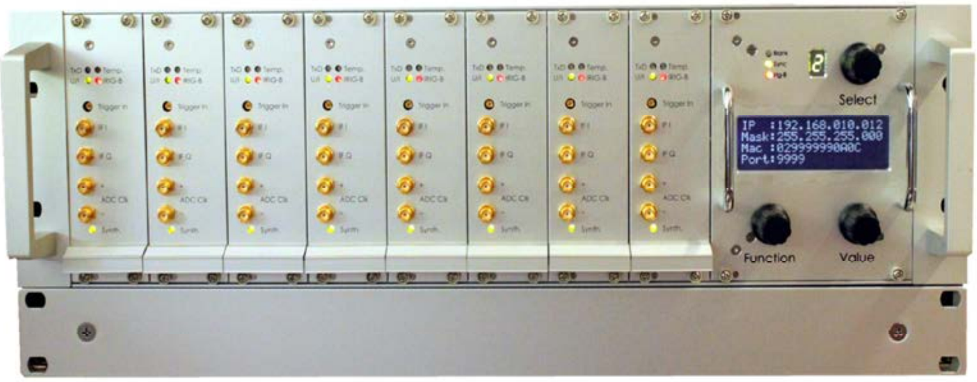
Radiometer Physics
A Rohde & Schwarz Company

Fast Fourier Transform Spectrometer
RPG_XFFTS



19'' FFTS crate equipped with 8 FFTS boards and one FFTS controller

INDEX

Features:

- Maximum instantaneous bandwidth: 2.5 GHz
- Up to 32768 spectral channels at full bandwidth
- Equivalent noise bandwidth (polyphase filter-bank): 88 kHz @ 2.5 GHz bandwidth / 32768 spectral channels
- On-board Ethernet Interface (100Base TX) for spectral readout and FFTS board configuration
- Input for external control signals: blank, sync, frequency reference
- Remote programmable FPGA (Xilinx Virtex6 LX240T)
- Precise time stamping of the processed spectra by on-board IRIG-B time-stamp encoding

1. FFT-Spectrometer FFTS

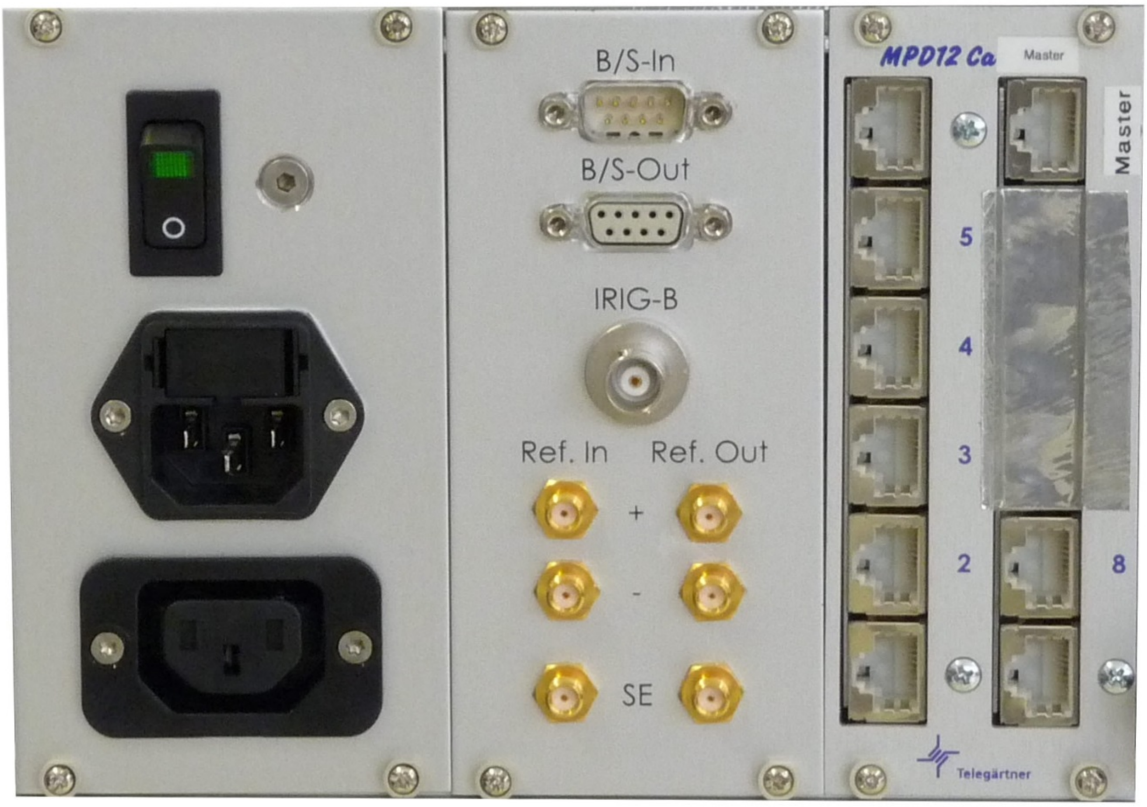
1.1. FFTS-Crate

Dimensions: width/high/depth: 480 mm / 135 mm + (44 mm fan plate) / 485 mm

Weight: 13 kg

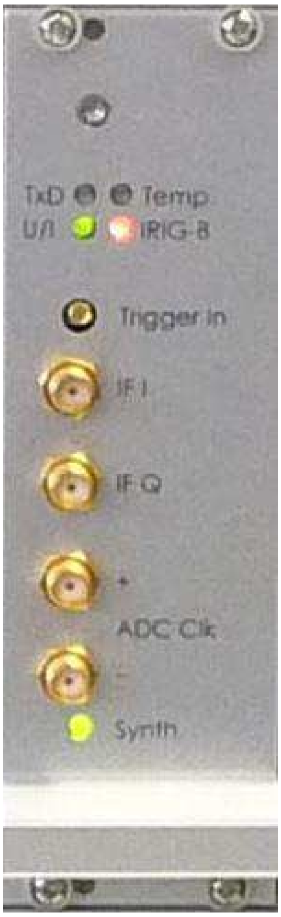
1.2. FFTS-Connectors

1.2.1. FFTS-Crate Connectors



Parameter		Specification
V_{ref} input differential:	connector:	2xSMA
(labeled “Ref-In +/-”)	impedance:	100 Ω / AC coupling
	level:	min. 400 mVpp differential
		max. 1000 mVpp differential
	frequency:	5 – 125 MHz (5 MHz steps)
V_{ref} input single-ended:	connector:	SMA
(labeled “Ref-In SE”)	impedance:	50 Ω / AC coupling
	level:	min. 800 mVpp (+ 2 dBm)
		max. 3000 mVpp (+13 dBm)
	frequency:	5 – 125 MHz (5 MHz steps)
V_{ref} output differential:	connector:	2xSMA
(labeled “Ref-Out +/-”)	impedance:	100 Ω
	level:	600 mVpp differential square wave
	frequency:	100 MHz
V_{ref} output single-ended:	connector:	SMA
(labeled “Ref-Out SE”)	impedance:	50 Ω
	level:	1.5 Vpp square wave
	frequency:	100 MHz
IRIG-B input:	connector:	BNC
	impedance:	25 kΩ / AC Coupling
	level:	Min. 1 Vpp
		max. 12 Vpp
		Max. 12 Vpp
Blank/Sync input:	connector:	9-pin Sub-D male
	impedance:	120 Ω differential
	level:	RS 485
Blank/Sync output:	connector:	9-pin Sub-D female
	impedance:	min. 100 Ω differential
	level:	RS 485 (1.5 V_{diff})
Ethernet (12x):	connector:	RJ-45
	standard:	100 Base-TX
	used ports:	1 – 8 (FFTS-boards)
		12 (FFTS-controller)
	unused ports:	9 – 11
Power supply:	connector:	AC-connector male
	voltage:	108 – 120 V_{AC} or 216 – 240 V_{AC} (automatic selection)
	frequency:	50 – 400 Hz
	current:	max. 9.6 A at 115 V_{AC}
		max. 5.0 A at 230 V_{AC}
	fuse:	5 × 20 mm 10 A slow at 115 V_{AC}
		5 × 20 mm 6.3 A slow at 230 V_{AC}

1.2.2. FFTS-Board Connectors



IF inputs:	connector:	SMA
	impedance:	50 Ω
	level:	absolute maximum +17 dBm! / 0 V DC! nominal -4 dBm
ADC-Clock output:	connector:	2xSMA
	impedance:	100 Ω
	level:	1800 mVpp differential

1.3. LEDs

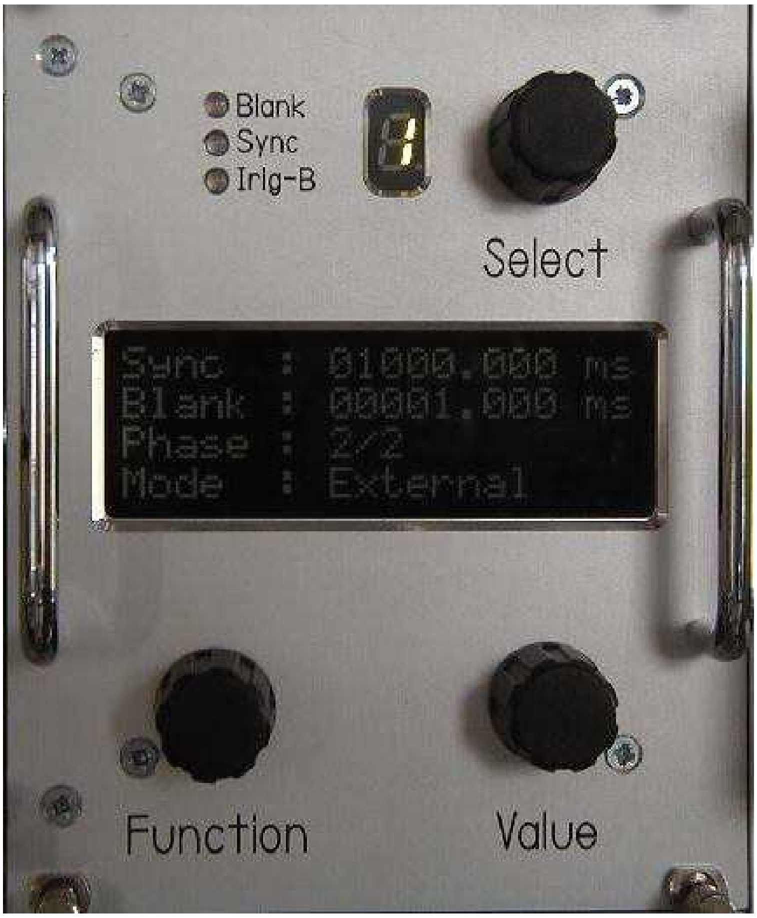
1.3.1.FFTS-Board LEDs

LED	Function	Off	Green	Red	Yellow
TxD	transmit status	not transmitting	transmitting	Phase-time too short to transmit data	-
Temp.	over temperature warning, status of ADC-Temp. regulation	temperature OK, regulation off	ADC-temp. stable, blink: wait for stabilization	75°C over temperature shutdown (reset by host)	-
U/I	voltage/current (U/I) warning	-	all U/I's within 5% tolerance	blink: one or more U/I's out of tolerance	-
IRIG-B	status of IRIG-B time signal	-	100 ms pulse each second: IRIG-B ok	error in IRIG-B signal	-
Synth	status of ADCclock synthesizer	synthesizer not locked	synthesizer locked	synthesizer not locked	-

1.3.2. FFTS-Controller-Board LEDs

LED	Function	Off	Green	Red	Yellow
Blank	Blank-signal and status of controller clock-synthesizer	record IF-inputs	Blank IF-inputs	synthesizer not locked	-
Sync	Sync-signal and status of controller clock-synthesizer	not Phase one	Phase one	synthesizer not locked	-
IRIG-B	status of IRIG-B time signal	-	100 ms pulse each Second: IRIG-B ok	error in IRIG-B signal	-

1.3.3. FFTS-Controller-Board 7-segment LED



```
Sync  : 01000.000 ms
Blank : 00001.000 ms
Phase : 2/2
Mode  : External
```

Controlled by the **Select**-Knob.

o. Overview Page **0.**
Controller Page

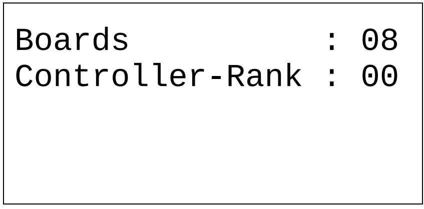
1-A FFTS-Board specific pages

1.4. FFTS-Controller Board-LCD-display

Pages controlled by the **Function**-Knob.

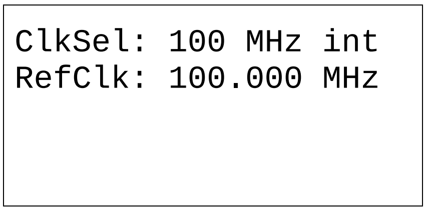
<u>NOTE:</u> The” Function” and “Value” - knobs have turn-and-push functionality!

1.4.1. Overview Pages



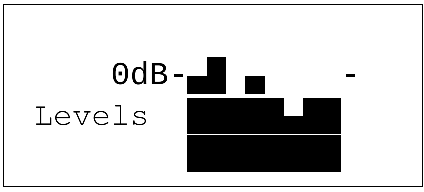
```
Boards          : 08
Controller-Rank : 00
```

<u>1st . Page:</u>
Number of Boards (1 – 8) and Controller-Rank (0 ... n). These values are adjustable with Function- and Value-Knob if the on-board „CHANGE_EN“-Jumper is set.

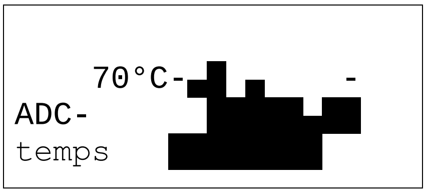


```
ClkSel: 100 MHz int
RefClk: 100.000 MHz
```

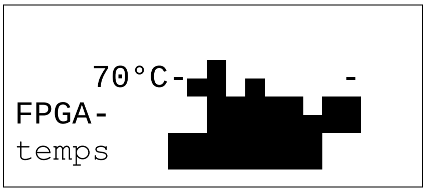
<u>2nd . Page:</u>
Selected Input-Clock (int/ext/ext2) and frequency measured at external Reference-Clock.



<u>3rd . Page:</u>
IF Leveling. Each pixel represents 1 dBm. The display covers the level range -21 to +11 dBm.



<u>4th . Page:</u>
ADC-Temperatures. Each pixel represents 2.5 °C.
The display covers the temperature range 20 to 100 °C.



<u>5th . Page:</u>
FPGA-Temperatures. Each pixel represents 2.5 °C.
The display covers the temperature range 20 to 100 °C.

<u>6th . Page:</u>
Data written by Software.

1.4.2. Pages for FFTS-Controller and FFTS-Boards

```
Sync  : 01000.000 ms
Blank : 00001.000 ms
Phase : 1/2
Mode  : Internal
```

1st . Page: Blank/Sync:
Phase-time (Sync-time)
Blank-time current Phase /
Total Number of Phases
Blank/Sync-generation:
Internal or External

```
IP  :192.168.010.010
Mask:255.255.255.000
Mac :029999990A0A
Port:9999
```

2nd Page: Network:
IP-Address
Subnet-Mask
MAC-Address
Port-Number (default: 9999)

```
Time   : 09:30:00
Day    : 42
Sp-ver: 5.19.5 /2000
Vx-ver: 5.19.4
```

3rd . Page: Time and
Version: IRIG-B time
IRIG-B day of the year
Spartan-Core-Version / Board ID
Virtex-Core-Version [a] [b]

```
5.0V   : 4.98V  02.4A
3.3V   : 3.27V
2.5V   : 2.51V
Clk    : 2500 MHz Int
```

4th . Page: FFTS-Board voltages
1: [a] nominal value : measured value
current at voltage nominal value :
measured value nominal value :
measured value measured
Sampling-Clock-frequency

```
2.5V   : 2.47V analog
1.8V   : 1.82V
1.0V   : 0.98V  10.4A
```

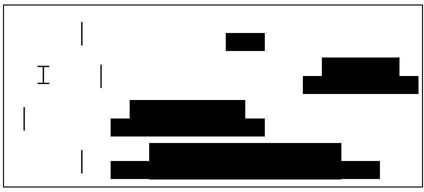
5th . Page: FFTS-Board voltages 2: [a]
nominal value : measured value
nominal value : measured value
nominal value : measured value
current at voltage

```
ADC    : +056.3°C
FPGA   : +065.5°C
Board  : +031.7°C
Level  : -003.4 dB
```

6th . Page: Temperatures and IF-Level: [a]
ADC-temperature
FPGA-temperature
Board-temperature
IF-level [b]

```
Chan.  : 32768
Bandw.: 2500 MHz
IntCnt: 00076218
Dump#  : 00000194
```

7th . Page: FFTS-Parameter: [a]
number of channels [b] Nyquist
bandwidth of FFTS [b] number
of integrated spectra at each
Phase [b] dump number
(increased each Phase) [b]



8th . Page: ADC-Histogram: [a] statistical distribution of ADC-results [b]

[a] Not Available for Controller-Board

[b] Only Available if Virtex-Core is Loaded

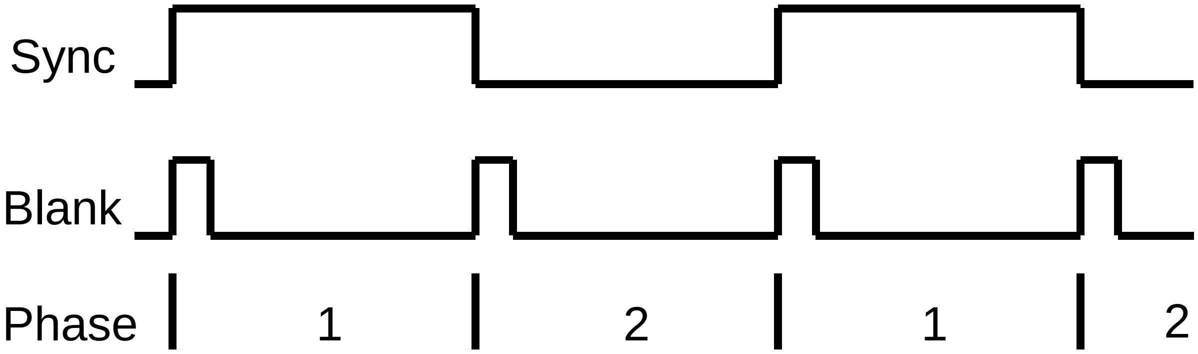
1.5. External Control Signals (Blank/Sync)

1.5.1. Blank/Sync input and output pin assignment

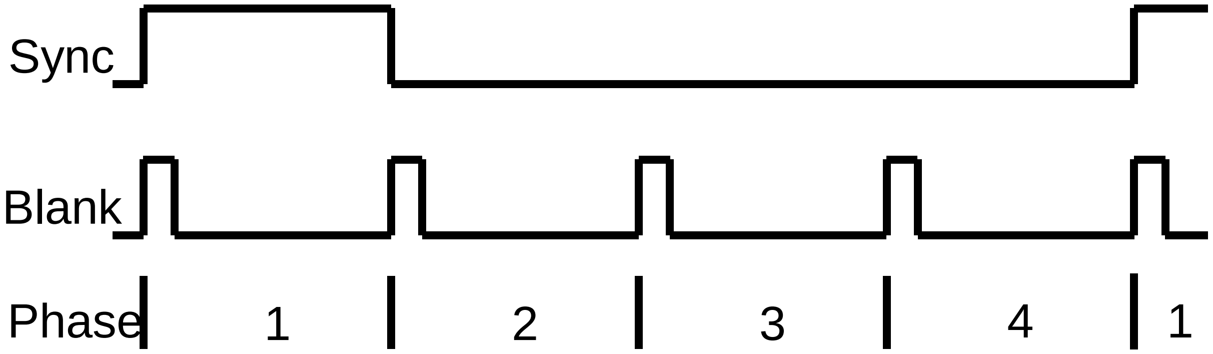
1	SYNC-		
2	BLANK-		
3	IRIG-CLK- [c]		
4	IRIG-AM- [c]		
5	GND		
6	SYNC+		
7	BLANK+		
8	IRIG-CLK+ [c]	9	IRIG-AM+ [c]

[c] only for stacking multiple crates

1.5.2. Signal example for a two Phase measurement



1.5.3. Signal example for a four Phase measurement



1.5.3. Blank/Sync signal description

The first measured phase is indicated by a high-level of the Sync-signal over the whole phase.

A new phase is indicated by a rising edge of the Blank-signal. While the Blank-signal remains high, the measurement is stopped (no spectral integration). Whenever the Blanksignal is low, the FFTS integrate spectra.

The FFTS-Controller is able to get these synchronisation signals from external (Blank/Sync input) and to chain them to the Blank/Sync output or generate the signals internally and forward them to the Blank/Sync output. The control-signals are differential at RS 485 level.

1.6. FFTS-Board-Description

1.6.1. Power-Supply

The only external supply-voltage to the boards is 5.0 V_{DC} at up to 10 A. All other voltages are on-board generated by dedicated power supplies. The external 5.0V supply is generated by the power supplies built into the backside of the crate.

1.6.2. ADC-Section

The analog-to-digital converter (ADC) has a resolution of 10 bits. It supports sample rates from 200 Mega samples per second (Msps) up to 5 Giga samples per second (Gsps). An on-board synthesizer generates the sampling-clock from a crate-internal 100 MHz reference clock signal.ina a The internal 100 MHz reference clock is either generated by an internal crystal-oscillator (50 ppm) located on the FFTS-Controller board or derived from the external reference frequency input (5 – 125 MHz).

1.6.3. Configuration

The basic configuration is stored in a Flash-PROM on the FFTS-Board. During power-up phase, this data is loaded by a Spartan-FPGA. After that, the Spartan receives IP- and MAC-Address from the FFTS-Controller board and sets up an on-board TCP-server for communication with the Backend-PC. Now the Backend-PC is able to load the desired FFT-Core to the Virtex-FPGA by Ethernet and start further configurations.

1.6.4. Control-Signals

The signals **BLANK**, **SYNC**, **IRIG_AM** and **IRIG_CLK** are distributed by the FFTSController over the backplane to each FFTS-Board. The signals **BLANK** and **SYNC** are used to synchronize the measurements of all FFTS-Boards. **IRIG_AM** and **IRIG_CLK** are needed to distribute the time information to the FFTS-Boards.

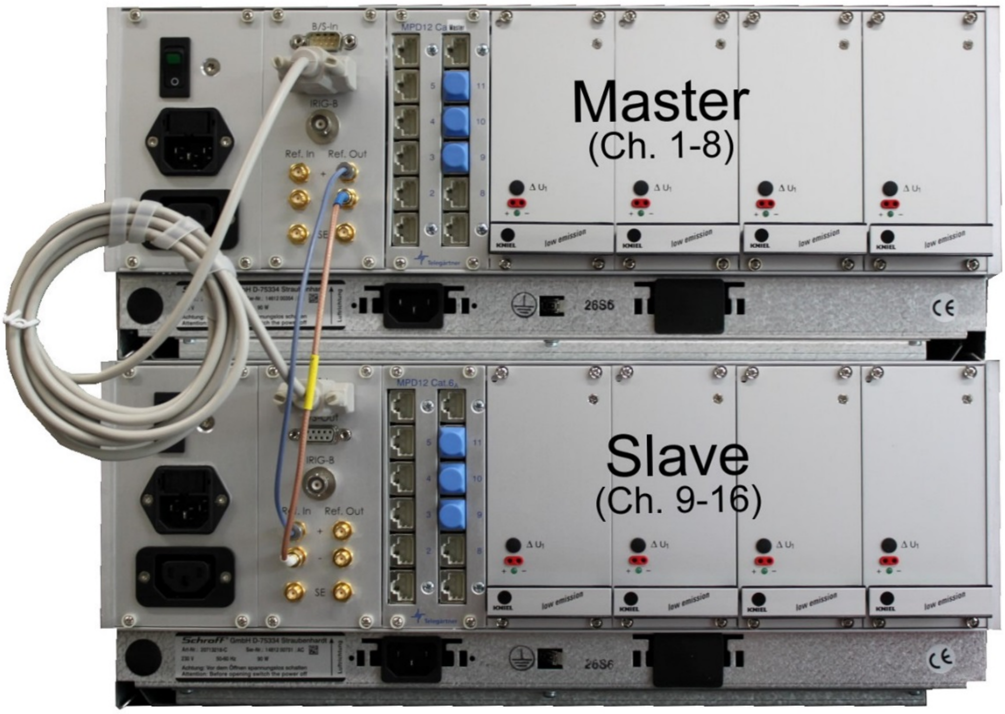


Fig. 1 Rear Side cabling in Master/Slave configuration

2. FPGA-signal processing cores

The default core for the Virtex-6 FPGA has 32768 spectral channels and is able to process spectra of 2.5 GHz bandwidth. The channel spacing (Δv) is 2.5 GHz divided by 32768 channels = 76 kHz. The FPGA core implements a 4-tap poly-phase filter bank algorithm. The filter coefficients are optimized for astronomical line observations. The equivalent noise bandwidth (ENBW) is 1.16 × Δv, thus 88 kHz at 2.5 GHz bandwidth and 32768 channels. Compared to a standard window-FFT processing, the poly-phase filter bank algorithm has a significant reduced frequency scallop loss, less bandwidth expansion, and faster sidelobe fall-off.

The FFTS is delivered with a standard FPGA core: 2.5 GHz bandwidth / 32768 channels. Additional cores with higher frequency resolution (smaller bandwidth and more spectral channels) are available upon request.

3. Software

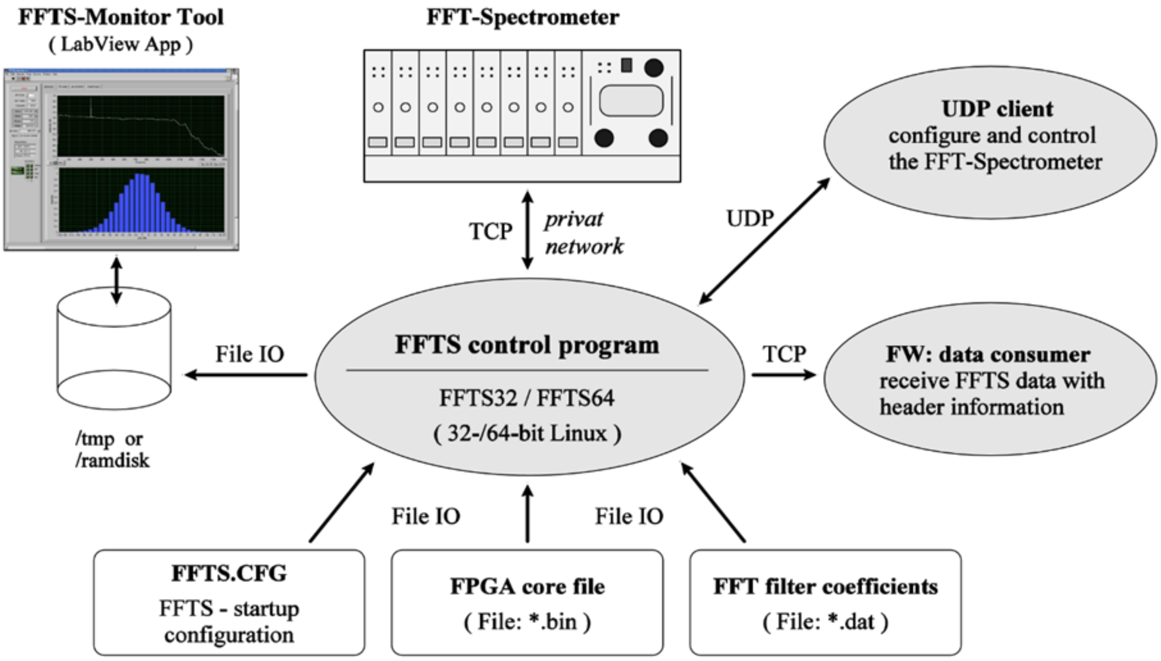
3.1. FFTS – The FFTS control program

The FFTS is controlled by a multi-threaded program: FFTS (FFTS software). FFTS can serve FFTS systems (arrays) with up to 32 FFTS-Boards and 4 FFTS-Crates. For each FFTS-Board one program thread is started, which controls the board configuration, collects data and handles house-keeping information. Currently, the FFTS software is available for 32- and 64-bit Linux-PCs (Kernel 2.6.18 or later).

During the initialization phase, the FFTS program loads startup parameter from the FFTS configuration file “FFTS.cfg” and establishes the TCP connection to the FFTS hardware. Following this, FFTS uploads the FPGA core file (e.g. “virtex6_32K_2500_4wola.bin”) and the poly-phase filter coefficients (e.g. “FLATTOP4WOLA32K.dat”) to the FFTS-Boards via Ethernet. From that moment the FFTS is ready for measurements.

The FFTS software is controlled by the UDP/SCPI command interface, which is structured along the baselines documented in RD-001. A summary table of all currently implemented SCPI commands is listed in section 3.4. The FFTS software uses a TCP-network connection to transmit the spectral data together with header information (integration time, time stamp,...) to an external data consumer (e.g. the observing system) for data recording. The implemented protocol is compatible to the SCPI backend data stream interface, which is documented in RD-01.

Independent of the FFTS control program, there is an on-line monitor display available (FFTS-Monitor Tool) for the FFTS, which is implemented as a LabView application. The
FFTS-Monitor Tool provides additional information about the status of the FFTS
(bandpass, level, Blank/Sync time, temperatures, etc.) to the user. A description of the FFTS-Monitor Tool is given in section 3.3. The communication between the FFTS software and the FFTS-Monitor Tool is implemented by FileIO. The connection between the individual programs is illustrated in the following diagram.



3.1.1 FFTS Initialization Software

Before you start the FFTS main program, make sure that the FFTS-Crate is correctly connected to the Ethernet switch and the FFTS-PC along the recommendations made in section 4.

Switch on the FFTS (plus ventilation!), the Ethernet switch and the FFTS-PC.

After successful login, the FFTS program is called from the command line, just by typing FFTS.

To make use of the different bandwidths, the program is started using:

./FFTS –configfile xxx.cfg (where xxx stands for FFTS_2000 for the 2000MHz core and for
FFTS.cfg for the 2500MHz core or FFTS_multicore.cfg for mixed operation)
Depending on the crate configuration, one has to edit the file FFTS_multicore.cfg thus assigning the individual bandwidth to the individual board numbers.

Example: (excerpt from “FFTS_MULTICORE.cfg” for 2500MHz and 2000MHz bandwidths)

```
# SETUP: 2500 MHz bandwidth, 32k spectral channels
[FPGA1]
coreFile  = ../XILINX/virtex6_32K_2500_4wola_19-6_2500.bin
bandwidthMHz  = 2500
numberOfChannels = 32768
specFilter              = 2
windowFile              = ../windows/FLATTOP4WOLA32K.dat
WOLA                    = 4
ADCdelayFile                    = ./ADCdelay.data


# SETUP: 2000 MHz bandwidth, 32k spectral channels
[FPGA2]
coreFile  = ../XILINX/virtex6_32K_2000_4wola_19-8_2490.bin
bandwidthMHz  = 2000
numberOfChannels = 32768
specFilter              = 2
windowFile              = ../windows/FLATTOP4WOLA32K.dat
WOLA                    = 4
ADCdelayFile                    = ./ADCdelay.data

# configuration settings  BOARD->FPGA<num>
[BOARD_FPGA]
```

```
FFTSboard01          = FPGA1
FFTSboard02          = FPGA1
FFTSboard03          = FPGA1
FFTSboard04          = FPGA1
FFTSboard05          = FPGA1
FFTSboard06          = FPGA1
FFTSboard07          = FPGA1
FFTSboard08          = FPGA2
```

Board 08 will run at 2000MHz whereas all other boards run at 2500MHz

The FFTS software is stopped by sending the Ctrl+C command.

Example: Successful startup sequence for a multi-core FFTS:

```
  +-------------------------------------------------
-----------+   |
|
  |   *** XFFTS: The eXtended Bandwidth FFT-Spectrometer ***
|
  |
|
  |
|
  |        VERSION: 20110331   RELEASE/BUILD: Mar 31 2011
|
  |
|
  |            --- RPG Radiometer physics GmbH ---
|
  |        Birkenmaarstr. 10, 53340 Meckenheim, Germany
|
  |              http://www.radiometer-physics.de
|
  |
|
  +------------------------------------------------------------
+


                Configuration file: FFTS_MULTICORE.CFG
                Telescope name:     RPG
                Backend name:       XFFTS
                Port SCPI  (UDP):   16210
                Port FitsWr(TCP):   25144
                Used FFTS boards:   2

                >>> Multi-core setup (4 FPGA cores). <<<
                FPGA-Core  1:        300 MHz / 32768 channels
                FPGA-Core  2:       2500 MHz /
32768 channels
                FFTS-Board #01:     2500 MHz / 32768 channels
                FFTS-Board #02:      300 MHz / 32768 channels

                Simulation-Mode:    NO


REMARK: Create FitsWriter
server .. REMARK: FitsWriter
server started.
REMARK: FitsWriterServer() started!
REMARK: FitsWriterServer(): Wait for Client at port 25144!

REMARK: Create FitsWriter thread ..
REMARK: FitsWriter thread started.
REMARK: Install signal handler ... okay.

REMARK: Wait and hold on until all TCP ports are closed ...
```

```
REMARK: Connect FFTS master (192.168.10.10) ... okay.
REMARK: Setup Blank, Sync, Phases to FFTS Master ... Okay.

REMARK: Startup 2 worker thread(s). One for every FFTS board.
REMARK: Connect FFTS board 01 (192.168.10.11) ... okay.
REMARK: Connect FFTS board 02 (192.168.10.12) ... okay.
REMARK: FFTS board 01 :: Load FPGA:
         -> ../XILINX/spain/virtex4_32K_2500_4wola_19-
4m_2500.bin.
REMARK: FFTS board 02 :: Load FPGA:
         -> ../XILINX/spain/virtex4_32K_300_4wola_19-
4p_312.bin.
REMARK: FFTS board 01 :: Download finished.
REMARK: FFTS board 01 :: Setup common defaults ..
REMARK: FFTS board 01 :: Setup okay.
REMARK: FFTS board 01 :: Bandwidth: 2500 MHz
REMARK: FFTS board 01 :: Load window:
../windows/FLATTOP4WOLA32K.dat.
REMARK: FFTS board 01 :: 128K point Window loaded.
REMARK: FFTS board 02 :: Download finished.
REMARK: FFTS board 02 :: Setup common
defaults .. REMARK: FFTS board 02 :: Setup
okay.
REMARK: FFTS board 02 :: Bandwidth: 750 MHz
REMARK: FFTS board 02 :: Load window:
../windows/FLATTOP4WOLA16K.dat. REMARK: FFTS board 02 :: 128K
point Window loaded.

REMARK: All FFTS boards are successfully initialized.

REMARK: FFTS board 01 :: Select ADC input I
REMARK: FFTS board 01 :: Startup data transfer ...
Okay. REMARK: FFTS board 02 :: Select ADC input I
REMARK: FFTS board 02 :: Startup data transfer ... Okay.
REMARK: Enable global Blank/Sync ... Okay.
REMARK: Create SCPI thread ..
REMARK: SCPI control thread started.. Okay.

REMARK: Optimize ADC/FPGA timing: 32/32
REMARK: FFTS board 01, SN 5.000 :: ADC/FPGA timing:  1 -  4 (
3)
REMARK: FFTS board 02, SN 5.001 :: ADC/FPGA timing:  9 – 13 (
5)
REMARK: FFTS board 01 :: Reset ADC ..
REMARK: FFTS board 02 :: Reset ADC ..
REMARK: ADC calibration successful.
REMARK: ADC-interleaving .................. okay.

REMARK: Now the XFFTS is ready for
observing. Good luck!
```

3.1.2. UDP/SCPI Command Interface

The following Python program shows a very simple way to send and receive UDP/SCPI commands to/from the FFTS software. If the program is called on an external PC, the line UDPHOST must declare the IP number of the FFTS-PC. The UDPPORT number must agree with the port given in the FFTS.cfg file (see section 3.2).

```
#!/usr/bin/env python

# udptelnet -- "telnet" like application using UDP instead of
TCP
 from
string
import
atoi from
socket
import *
from
thread
import *
 def
recv(ud
pCliSoc
k):
while
1:
        data, addr =
udpCliSock.recvfrom(BUFSIZE)
if not data: break         print
'RECV> ',data

UDPHOST = 'localhost' # or the valid IP number xxx.xxx.xxx.xxx
UDPPORT = 16210

BUFSIZE = 16*1024
ADDR = (UDPHOST, UDPPORT)
 udpCliSock =
socket(AF_INET,
SOCK_DGRAM)

start_new_thread(recv,(
udpCliSock,))

w
h
i
l
e
1
:
    data =
raw_input('')
print 'SEND> ',
data     data =
data + '\n'     if
```

```
not data: break
udpCliSock.sendto(d
ata, ADDR)

udpCliSock.close()
```

Examples:

```
/home/FFTS/sh> ./udptelnet
RPG:XFFTS:version?
SEND> RPG:XFFTS:version?
RECV> RPG:XFFTS:VERSION 20110331 2011-03-31T23:44:35.001+0000

RPG:XFFTS:BAND1:numSpecChan?
SEND> RPG:XFFTS:BAND1:numSpecChan?
RECV> RPG:XFFTS:BAND1:NUMSPECCHAN 32768 2011-03-
31T23:47:15.866+0000
SEND> RPG:AFFTS:cmdSyncTimeRPG:XFFTS:cmdSynctime 500000
SEND> RPG:XFFTS:cmdSyncTime 500000
RECV> RPG:XFFTS:CMDSYNCTIME 500000 2011-03-31T23:49:15.192+0000

RPG:XFFTS:configure
SEND> RPG:XFFTS:configure
RECV> RPG:XFFTS:CONFIGURE 2011-03-31T23:52:05.873+0000
```

3.1.3. TCP Data Interface

The FFTS software transmits spectra data together with header information by a TCPnetwork connection to a data consumer program. The protocol is documented in RD-01.

To illustrate the data transfer, a simple consumer program (FW.c) – implemented in C – is available in source code. The program compiles with:

```
gcc -Wall -O2 -lm -o FW FW.c swap.c
```

FW is called from the command line with the IP number of the FFTS-PC, e.g. FW 134.104.70.12.
If started on the FFTS-PC, the IP number can be replaced by the keyword 'localhost'.

Example: Output of FW program for a two board FFTS.

```
FFTS> FW localhost

FitsWriter is connected to localhost and waiting for data...

-----[ DUMP: 1 --- received:
262224 Bytes ]----- IEEE:
[EEEI] dataFormat:     [F   ]
package Length: 262224
BackendName:    [XFFTS]
Timestamp:      [2011-03-31T23:51:52.6660GPS ]
Integration:    998993 us
Phase:          1
```

```
BE-Sections:    2
Blocking:       1
- - - - - - - - - - - - - -
BE#  1, channels: 32768
BE#  2, channels: 32768
-----------------------------------------------

-----[ DUMP: 2 --- received:
262224 Bytes ]----- IEEE:
[EEEI] dataFormat:     [F   ]
package Length: 262224
BackendName:    [XFFTS]
Timestamp:      [2011-03-31T23:51:53.6660GPS ]
Integration:    998993 us
Phase:          2
BE-Sections:    2
Blocking:       1
- - - - - - - -
- - - - - BE#
1, channels:
32768
BE#  2, channels: 32768
-----------------------------------------------
```

3.1.4. Example: SCPI command sequence

The following UDP/SCPI command sequence demonstrates the successful configuration of the FFTS for a measurement with 4 FFTS-Boards. The first two boards are setup to 2048 spectral channels, the second two boards transmit spectra with 8192 channels. Blank and Sync time is set to 1 second and 5 ms, and internal generation.

```
RPG:XFFTS:cmdMode INTERNAL             # use internal Blank/Sync generator
RPG:XFFTS:cmdSyncTime 1000000            # Sync time:  1 sec
RPG:XFFTS:cmdBlankTime 5000              # Blank time: 5 ms
RPG:XFFTS:cmdUsedSections 1 1 1 1 0 0 0 0 # select the first
four FFTS-Boards
RPG:XFFTS:Band1:cmdNumSpecChan 2048      # set FFTS-Board 1 to
2048 channels RPG:XFFTS:Band2:cmdNumSpecChan 2048    #
RPG:XFFTS:Band3:cmdNumSpecChan 8192    # set FFTS-Board 3 to 8192 channels
RPG:XFFTS:Band4:cmdNumSpecChan 8192    #
RPG:XFFTS:configure                      # activate commanded parameters
RPF:XFFTS:calADC                         # optimize ADC interleaving
RPG:XFFTS:start                          # start measurement and data transfer
RPG:XFFTS:stop                           # stop measurement and data transfer
```

3.2. FFTS-Configuration-File

```
#
# Fast Fourier Transform Spectrometer
#
# >>> Multi-Core setup <<<
#
# setup and configuration file / multi-core version
#
# 2011-03-31,  bklein@mpifr-bonn.mpg.de
#

[common]
telescopeName     = RPG       # maximum 10 chars
backendName       = XFFTS           # maximum  8 chars
UDPportNumber     = 16210
TCPportNumber     = 25144
tempFiles         = /ramdisk/ # directory for temporary files

# SETUP:  300 MHz bandwidth, 32k spectral channels
[FPGA1]
coreFile    = ../XILINX/virtex6_32K_300_4wola_19-4p_312.bin
bandwidthMHz      = 300
numberOfChannels = 32768
specFilter        = 2
windowFile        = ../windows/FLATTOP4WOLA32K.dat
WOLA              = 4
ADCdelayFile      = ./ADCdelay.data

# SETUP: 2500 MHz bandwidth, 16k spectral channels
[FPGA2]
coreFile    = ../XILINX/virtex6_32K_2500_4wola_19-
4m_2500.bin bandwidthMHz      = 2500 numberOfChannels
= 32768
specFilter        = 2
windowFile        = ../windows/FLATTOP4WOLA16K.dat
WOLA              = 4
ADCdelayFile      = ./ADCdelay.data

# configuration settings  BOARD->FPGA<num>
[BOARD_FPGA]
FFTSboard01       = FPGA1
FFTSboard02       = FPGA2
FFTSboard03       = FPGA1
FFTSboard04       = FPGA2
FFTSboard05       = FPGA1
FFTSboard06       = FPGA2
FFTSboard07       = FPGA1
FFTSboard08       = FPGA2

[FFTScrate]
usedFFTScontrollers     = 1
usedFFTSboards    = 8
```

```
[FFTScontroller]
refFreqMHz01            = 100      # INT(ernal) frequency is 100 MHz
refFreqMHz02            = 100      # reference frequency range: 5 - 125 MHz,
refFreqMHz03         =    100      # step size is: 5 MHz
refFreqMHz04         =    100
refFreqSource01       =   INT
refFreqSource02  = INT

[FFTScontrollerIPs]
IPcontrollerNum01 = 192.168.10.10 # master FFTScontroller /
crate 1
IPcontrollerNum02 = 192.168.10.20 # FFTScontroller / crate 2

[FFTSboardIPs]
IPboardNum01     = 192.168.10.11
IPboardNum02     = 192.168.10.12
IPboardNum03     = 192.168.10.13
IPboardNum04     = 192.168.10.14
IPboardNum05     = 192.168.10.15
IPboardNum06     = 192.168.10.16
IPboardNum07     = 192.168.10.17
IPboardNum08     =
192.168.10.18
IPboardNum09     =
192.168.10.21 ...
###############################################################
###
```

3.3. UDP/SCPI commands

The UDP/SCPI-commands are subdivided in three parts: **Methods**, **Properties** and **Band dependent commands**. The following list compiles all UDP/SCPI commands, which are currently implemented in the FFTS software.

3.3.1. Methods

SCPI command:	Description:
XFFTS:START	Start measurement in first phase
XFFTS:STOP	Stop measurement (after last phase)
XFFTS:ABORT	Abort measurement after current phase
XFFTS:CONFIGURE	Configure FFTS; activate all commanded settings
XFFTS:INITSYNTHESIZER	Initialize all FFTS on-board synthesizer

3.3.2. Properties

SCPI command:	Description:
XFFTS:STATE	FFTS state: ENABLED, DISABLED
XFFTS:BLANKTIME	Get Blank time in µs
XFFTS:CMDBLANKTIME	Set new Blank time in µs, min. 1 ms
XFFTS:SYNCTIME	Get Sync time in µs
XFFTS:CMDSYNCTIME	Set new Sync time in µs, range 100 ms – 5 s
XFFTS:NUMPHASES	Get number of Blank/Sync Phases
XFFTS:CMDNUMPHASES	Get new number of Phases, range 1 – 4 Phases
XFFTS:MODE	Blank/Sync mode: INTERNAL or EXTERNAL
XFFTS:CMDMODE	Set Blank/Sync mode to INTERNAL or EXTERNAL
XFFTS:USEDSECTIONS	Display which FFTS-Boards (sections) are selected to transmit spectra via TCP protocol
XFFTS:CMDUSEDSECTIONS	Select FFTS-Boards which are allowed to transmit spectra via TCP protocol
XFFTS:VERSION	Get FFTS software version number
XFFTS:RELEASE	Get FFTS release date
XFFTS:CALADC	(re-)calibrate the ADC interleaving of all ADCs
XFFTS:INFO	Display FFTS-Board information for board *n* (*n*: 1..x)
XFFTS:DUMP	Dump out *m* spectra from selected boards via TCP
XFFTS:SAVEADCDELAYS	Save all ADC delays in file “ADCdelay.data”
XFFTS:LOADADCDELAYS	(re-)load ADC delays from file “ADCdelay.data”

3.3.3. Band dependent commands

The UDP/SCPI protocol denotes FFTS-Boards as BANDs, beginning with BAND1. A FFTS system with many FFTS-Boards in several FFTS-Crates, the FFTS-Board on the left side in the first crate is identified as BAND 1. Consequently, the last BAND marked the FFTSBoard in the last crate on the right side (see section 4.1 for an illustration of the FFTSBoard numbering).

Currently, the FFTS software supports up to 32 FFTS-Boards in 4 FFTS-Crates.

The following SCPI commands only refer to the FFTS-Board *n* (BAND*n*).

SCPI command:	Description:
XFFTS:BAND*n*:NUMSPECCHAN	Get number of spectral channels
XFFTS:BAND*n*:CMDNUMSPECCHAN	Set number of spectral channels (only power of two values are allowed: 8192, 4096, 2048, ...)
XFFTS:BAND*n*:BANDWIDTH	Get bandwidth [MHz] of FFTS-Board *n*
XFFTS:BAND*n*:CMDBANDWIDTH	Set bandwidth [MHz]. Possible bandwidth depends on the FPGA core. Default: 2500 MHz
XFFTS:BAND*n*:MIRRORSPECTRA	Get “mirror spectra” info: 0:mirror off (default)/ 1:mirror on (e.g. for even Nyquist zone sampling)
XFFTS:BAND*n*:CMDMIRRORSPECTRA	Set “mirror spectra”: 0:mirror off / 1:mirror on (e.g. for even Nyquist zone sampling)
XFFTS:BAND*n*:CALADC	Calibrate/optimize ADC interleaving
XFFTS:BAND*n*:ADCDELAY	Depending on the FPGA core and the total bandwidth, the timing between the ADC and FPGA has to be adjusted. For the default core and 2.5 GHz bandwidth, this command is more or less obsolete.
XFFTS:BAND*n*:TIME	Get GPS/IRIG-B time and date
XFFTS:BAND*n*:TEMPERATURE	Get FFTS-Board temperatures in °C for the ADC, FPGA and power supplies
XFFTS:BAND*n*:SPECFILTER	Due to the ADC interleaving, a narrow interference line (*birdie*) can appear in one frequency bin exactly in the middle of the band. Mostly, this birdie can be minimized by calibrating the ADCs after they reached the final stable temperature, e.g. 5 minutes. If not, the command SPECFILTER 2 divides the total bandwidth in 2 parts and remove the birdie by interpolating between the direct neighbor bins of both parts. The SPECFILTER can also be defined in the FFTS configure file: FFTS.cfg.

3.4. FFTS-Monitor GUI

The Monitor-GUI can only update its indicators when the FFTS-Software is running.

With the **FFTS#**-control element, the FFTS-Board to monitor is selected.

BW / MHz: Full usable bandwidth for the current FFTS configuration

Channels: Number of spectral channels

Sync: Phase time in ms

Blank: Blank time in ms

Phase: current Phase / total number of Phases

Mode: Internal: Blank/Sync is generated by the FFTS-Controller
External: Blank/Sync is accepted from an external source

Int-Time: Integration time of the current spectrum

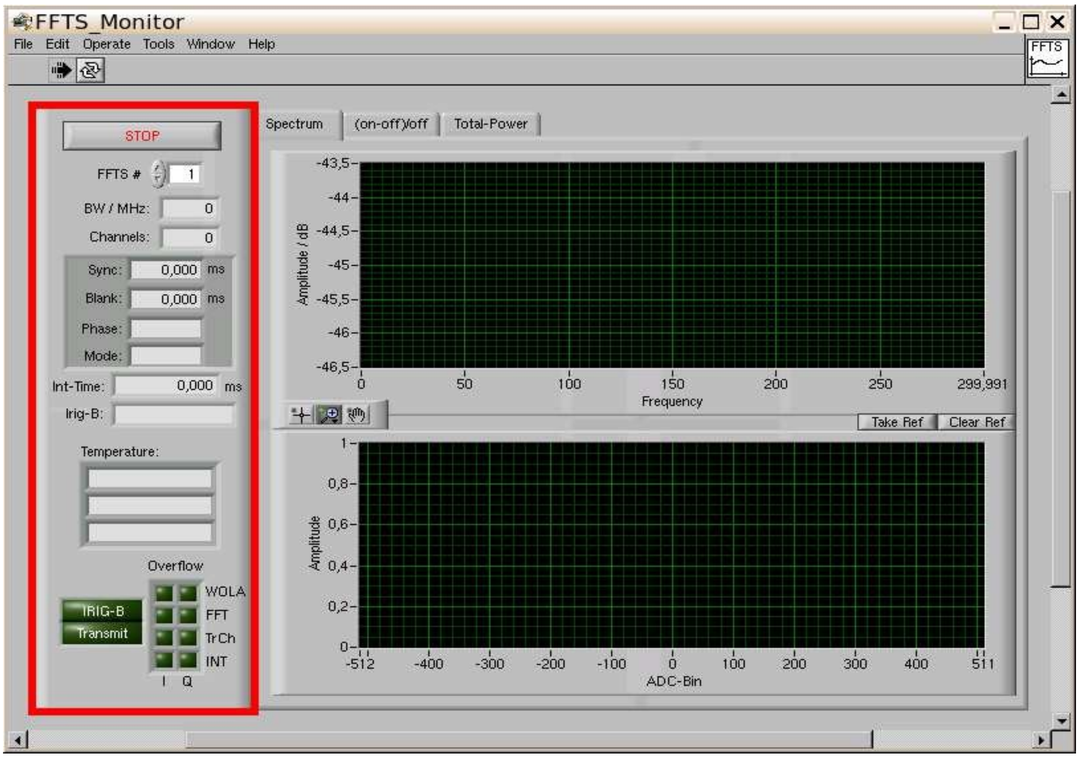
Irig-B: GPS time in format HH:MM:SS.microseconds

Temperature: Temperature of ADC-chip, main FPGA-chip and FFTS-Board, in °C

IRIG-B: green=OK / red=IRIG-B-Signal is corrupted or not available

Transmit: green=OK / red=Data transfer to the FFTS-PC was not successful

Overflow: Overflow flags for several points of the signal process pipeline

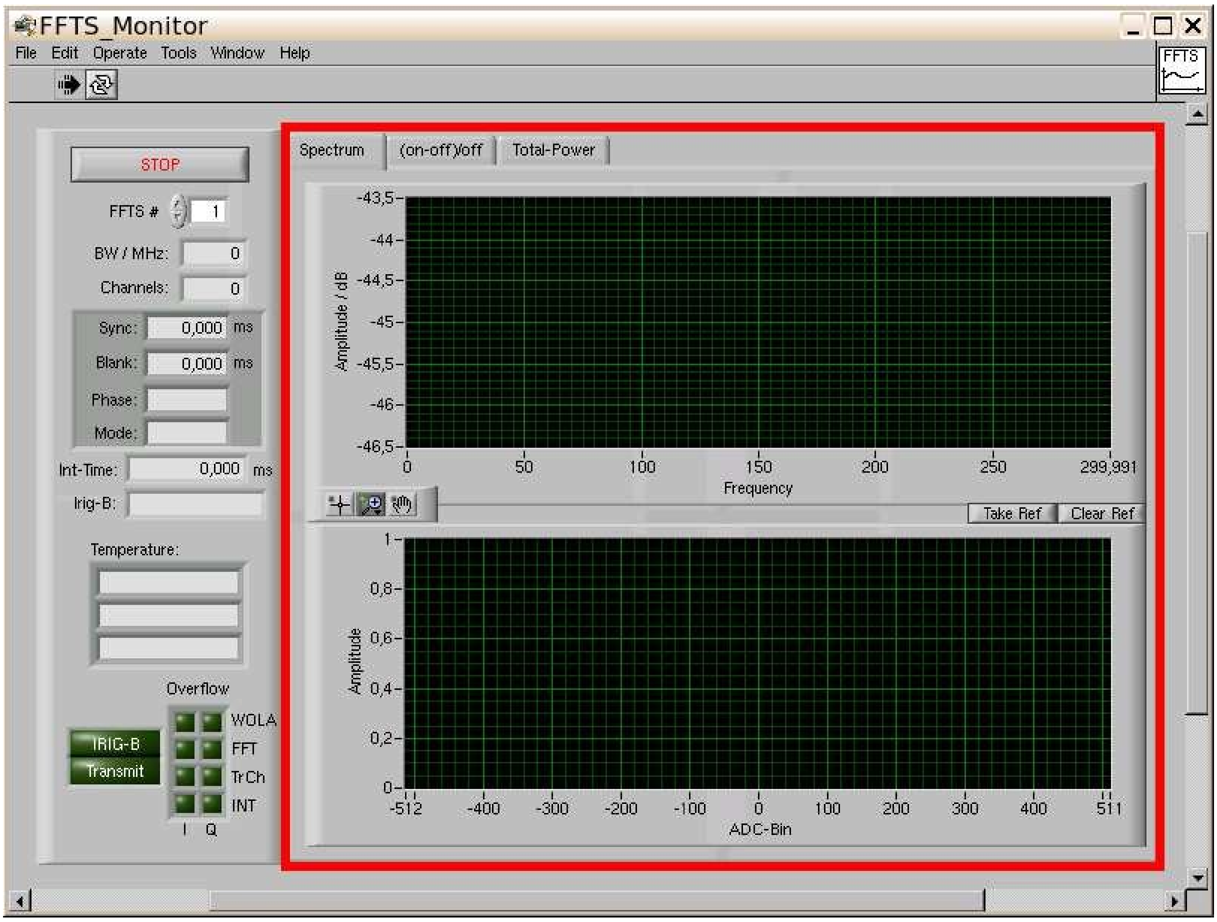


FFTS # 1
BW / MHz: 0
Channels: 0
Sync: 0,000 ms
Blank: 0,000 ms
Int-Time: 0,000 ms

3.4.1. Spectrum display

The upper diagram displays measured spectra (bandpasses). X-axis is frequency in MHz, Y-axis is power in dB. To zoom in and out, use the **zoom-tools** in the lower left corner of the diagram or change the first or the last scale-label available on both axis. To compare spectra, you can freeze the current spectrum by pushing the **Take Ref** button and clear it by pushing the **Clear Ref** button.

The lower diagram shows the statistical distribution of the ADC-output. The X-axis represent the digital value of the ADC; the Y-axis is the normalized number of occurrences for each ADC value.



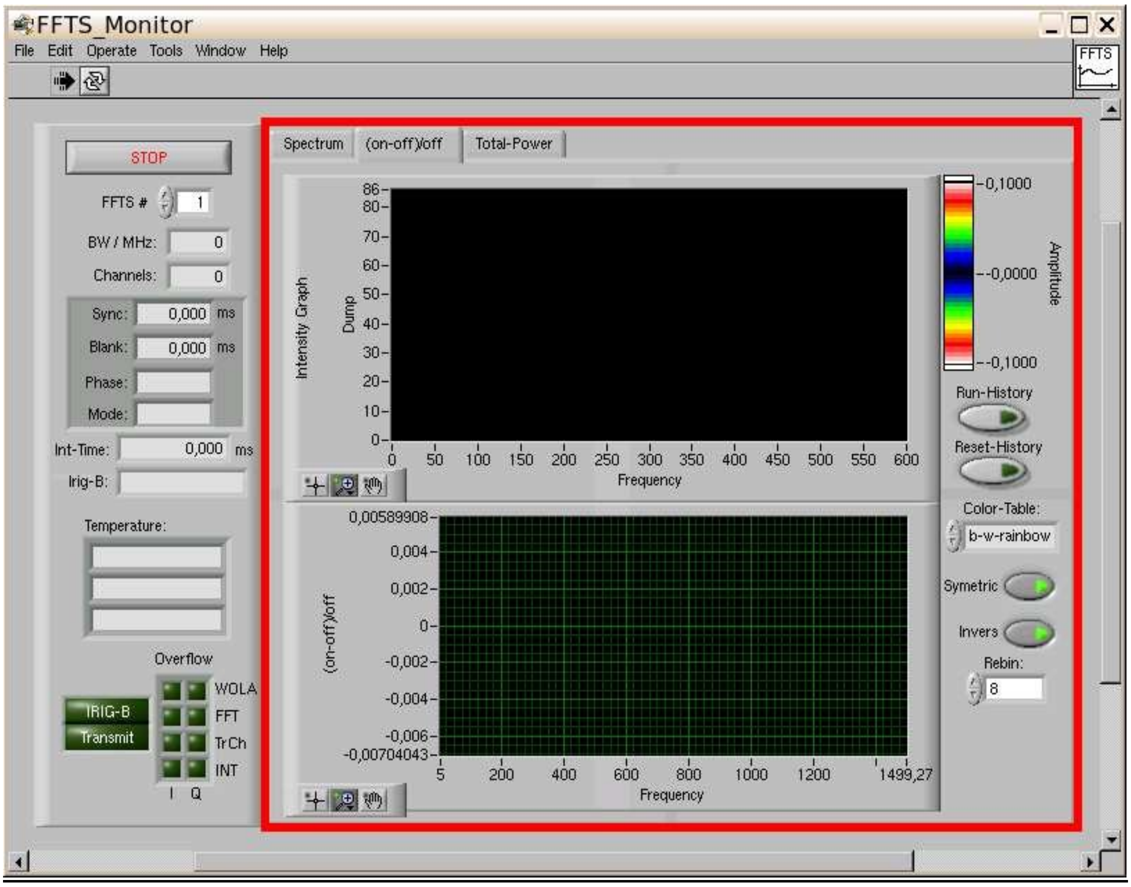
FFTS # 1
BW / MHz: 0
Channels: 0
Sync: 0,000 ms
Blank: 0,000 ms
Phase:
Mode:
Int-Time: 0,000 ms
Irig-B:

3.4.2. (on-off)/off display

The upper diagram shows a waterfall-plot of normalized spectra. The normalization is calculated by (on-off)/off, where off is the first spectrum which is recorded after pressing the **Run-History** button. Consequently, all following dumps are denoted as on spectra. After starting the GUI and pressing the **Run-History** button, it is necessary to push the **Reset-History** button once to initialize the waterfall plot. Every mouse-click on the **ResetHistory** button will clear the waterfall-plot. Zooming to the plots is possible in the same way as already described in section 3.3.3. In addition, the GUI allows changing the colors and orientation of the intensity-scale by the control elements **Color-Table**, **Symmetric** and **Invers**. The intensity scale can be adjusted by changing the first and last scale-label.

The lower panel displays single (on-off)/off calculations. As long as the **Run-History** button is not activated, the off spectra are derived from Phase 1 and the on spectra are recorded in Phase 2. It is possible to average adjacent frequency channels to reduce the resolution and the noise-floor of the spectra. The number of channels which are smoothed together is specified by the **Rebin** field.

Note: Avoid changing the Rebin factor while running a waterfall-plot because it affects the number of frequency channels in this plot too.



FFTS # 1

BW / MHz: 0

Channels: 0

Sync: 0,000 ms

Blank: 0,000 ms

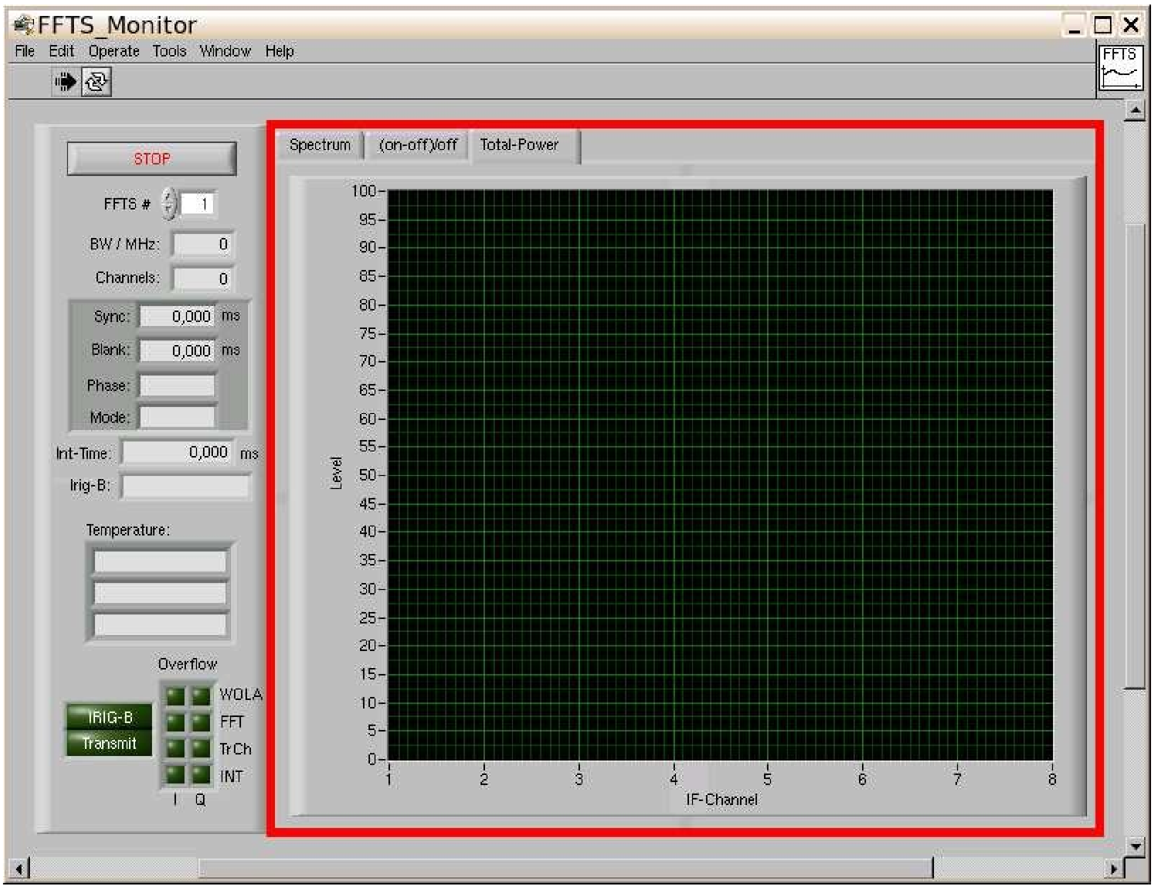
Int-Time: 0,000 ms

Color-Table: b-w-rainbow

Rebin: 8

3.4.3. Total-Power-Display

This diagram displays the IF-input level of up to 32 FFTS-Boards.



FFTS # 1
BW / MHz: 0
Channels: 0
Sync: 0,000 ms
Blank: 0,000 ms
Phase:
Mode:
Int-Time: 0,000 ms
Irig-B:

3.5. FPGA core file

The default FPGA core file (2.5 GHz bandwidth / 32768 spectral channels) is named: `virtex6_32K_2500_4wola.bin`
Name and location of this core file is specified in the FFTS.cfg startup settings. See section 3.2.

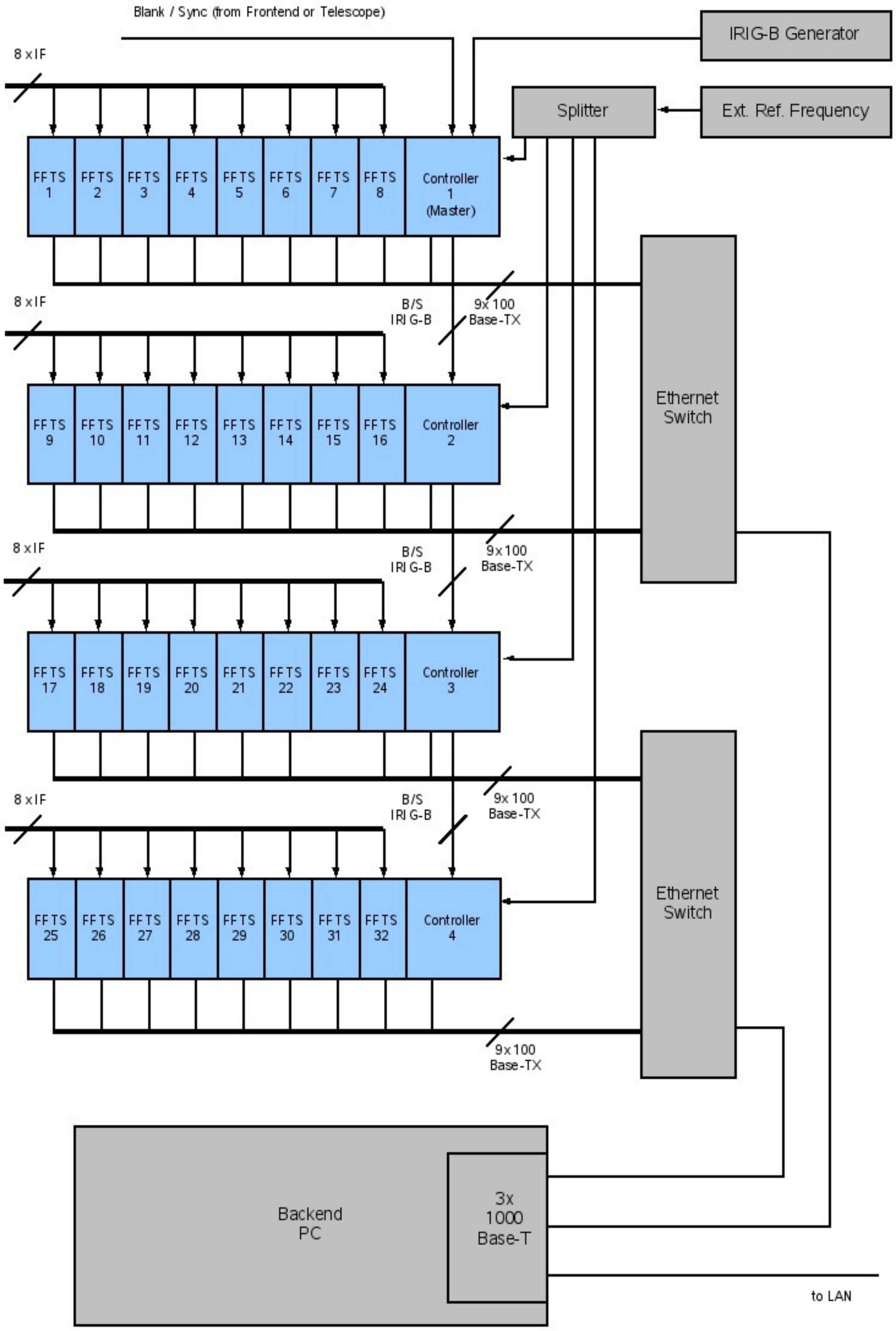
Additional core files are available on request!

3.6. Polyphase filter bank coefficients / Window file

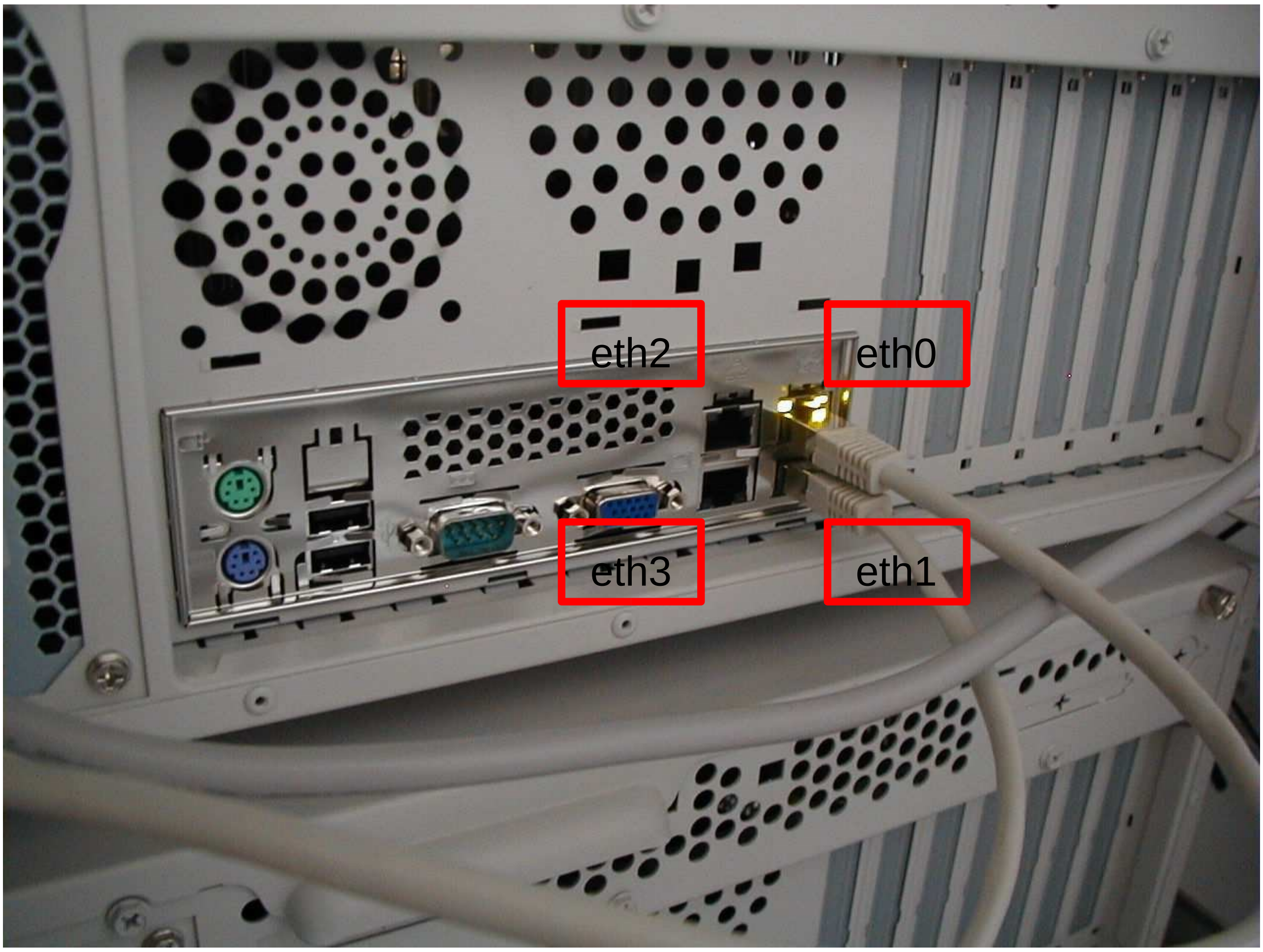
The filter coefficients are uploaded during the initialization phase of the FFTS. The default filter coefficients (File: `FLATTOP4WOLA32K.dat`) are optimized for astronomical line observations.
The ENBW is 1.16 × Δv. (see sections 2.)
Name and location of the filter coefficient file is specified in the FFTS.cfg startup settings. See section 3.2.

4. Environment

4.1. Recommended Wiring-Scheme



4.2. Recommended IP-Configuration



eth0 : LAN

eth1 : FFTS-Crate 1 + 2 (recommended IP:

192.168.10.1) eth2 : FFTS-Crate 3 + 4

(recommended IP: 192.168.20.1)

4.3. Ethernet-Switch

A buffered Ethernet-Switch that is capable of storing in minimum of one standard Ethernetframe in each port is necessary to connect the FFTS-Boards to the FFTS-PC. If more than 1 FFTS-Board is connected to one PC we recommend a 1000BaseT-connection. One PC can handle up to 32 FFTS-Boards. In this case two of the recommended Switches are needed.

Our recommendation is a **HP ProCurve2810-24G Switch** (24-Port Gigabit-Switch)

4.4. Backend-PC

4.4.1. Hardware Requirements

- Server-motherboard with:

  1 x 10/100/1000 LAN-Port
  1 x independent Gigabit-Ethernet-Port (2 Gigabit-Ports to operate with 2 Switches) e.g. Asus P5BV-C/4L (4 x Gigabit-LAN) Intel Core2Quad-CPU, 2 GByte Ram

4.4.2. Software Requirements

- Operating System:

  Debian GNU/Linux 4.0r3 or later, Kernel 2.6.18 or later
  (vnc4server, vnc4viewer, gcc, make, python, labview-runtime-engine)

- Network setup:

  10/100/1000-Lan: Use a network-setup, that covers your local LAN
  requirements e.g. 'dhcp'

  Gigabit-LAN 1: static IP-address: 192.168.10.1
  (recommended) Netmask 255.255.255.0

  Gigabit-LAN 2: static IP-address: 192.168.20.1
  (recommended) Netmask 255.255.255.0

- RAMDISK:

  Used for communication with the LabView FFTS-Monitor Tool.

– create a RAM-Disk-Volume with a maximum size of 1 Gbyte at /media/ramdisk:

```
root> mount -t ramfs -o maxsize=1048576 none /media/ramdisk/
```

– allow normal users to acces this RAM-Drive:

```
root> chmod 0777 /media/ramdisk/
```

– create a symbolic link /ramdisk pointing to /media/ramdisk: **root> ln -s /media/ramdisk/ /ramdisk** • VNC-Session:

  Useful to display the FFTS-Monitor Tool on different PCs.

– login as normal user and startup an VNC server session e.g.:

```
vncserver -geometry 1024x768 -alwaysshared :2
```

– now the VNC can be started on several PCs in the local network, e.g.:

```
vncviewer -shared <FFTS-IP number>:2
```

5. Reference documents

RD-01

APEX SCPI socket command syntax and backend data stream format,
2006-03-29

(available at: http://www.radiometer-physics.de/rpg/docs/FFTS/APEX-MPI-ICD-0005-R1_0.pdf)