TESTALL: Pcmtvxxx

This document decribes the TESTALL program for Wizard products.
Load the program onto a PCMCIA card, then load it into the wizard.
The program starts by showing this on the display:

"Device Detected"
"Device <name>"
"Press OK To Continue"

When VK_ACCEPT is selected the DSP is initialised and the text is displayed:

"TestAll Vx.xx"
"Test: <name>"
"Press OK To Select"

When one of the tests is selected then it is executed. Below is listed all tests that are possible.

Test case:	**Display:**	**Procedure**
TESTALL_EEPROM:	*"EEPROM"*	*checkEEprom*

"EEPROM TEST"
"Device. <name>"
"S/N: <serial>"
"OK For Write Test"
"CANCEL To Skip Test"

The test is started by first reading the contents into the EEPDATA structure. The Eeprom (64 words) is then tested by writing zeros and then 0x5443 + *addr*. The results is then read and tested. Finally a test is performed by writing the EEPDATA back into the EEprom. The result is verified by reading the Eeprom.

"Test Done"
"EEPROM OK"
"Continuos Test", "Check signals"

It is possible to make a continuos test for checking the signals by reading the same address (20) forever.

Possible errors:
"Failed Writing"
"Failed Reading"
"Wrong Data Read"

Test case:	**Display:**	**Procedure**
TESTALL_MIDI:	*"MIDI"*	*checkMidi*

"MIDI Test"
"Empty Input Buffer", "Wait..."
"Sending 200 Bytes", "Wait...."
The Midi test is a loop back test. Where the output of the Midi interface is connected to the input. First the input buffer is emptied. Then 200 bytes are transmitted to the output by calling *putMidi* the data is received by calling *getMidi*.

"Test Done"
"Continous Test", "Check RX/TX"

It is possible to make a continuos test for checking the communication by transmitting forever. (0-255)

Possible errors:
"Error, Unexpected", "MIDI Data Received"
"Error Receiving", "Lost Byte"
"Error Receiving", "Wrong Byte"

Test case:	**Display:**	**Procedure**
TESTALL_485:	*"RS485"*	*check485*

"RS485 Test"
"Empty Input Buffer", "Wait…."
"Sending 200 Bytes", "Wait…."

The RS485test is a loop back test. Where the output of the RS485 interface is connected to the input. First the input buffer is emptied. Then 10 times 200 bytes are transmitted to the output by calling *_putchar* the data is received by calling *_getchar*.

"Test Done", "RS485 OK"
"Continous Test", "Check RX/TX"

It is possible to make a continuos test for checking the communication by transmitting forever. (0-255)

Possible errors:
"Error, Unexpected", "RS485 Data Received"
"Error, Received Char", "With RS485 Disabled"
"Error Receiving", "Lost Byte"
"Error Receiving", "Wrong Byte"

Test case:	**Display:**	**Procedure**
TESTALL_PEDAL:	*"PEDAL"*	*checkPedal*

"Pedal Test"
"OK If No Error", "CANCEL If Error"

"Value = <volt>", "State = <ON/OFF>"

The status of the pedal input is detected and the voltage supply (U_PEDAL) is measured.

Test case:	**Display:**	**Procedure**
TESTALL_VOLTAGES:	*"PSU"*	*checkVoltages*

"Voltages Test"

"U-15: <volt>"	VPP_M15
"U+15: <volt>"	VPP_P15
"PA: <volt>"	VPP_POTA
"PB: <volt>"	VPP_POTB

The supply voltages is measured and displayed.

Test case:	**Display:**	**Procedure**

TESTALL_VIEWINGANGLE: *"V.ANGLE"* *checkViewingAngle*

"Viewing Angle Test"
"OK If No Error", "CANCEL If Error"
"Contrast <angle>%"
"Voltage <volt>"

The contrast voltages is measured and displayed. It is possible to change the contrast by setting the viewing angle up to 100 %. ITU timer 3 is used to control the contrast of the display. The ITU_GRA3 register is used to set the PWM-CONTRAST (TCA3) signal to the front display.

Test case:	**Display:**	**Procedure**

TESTALL_PCMCIA_NORMAL: *"PCMCIA"* *checkTestAllPCMCIA*

"PCMCIA Test"
"Battery: Good"
"Continue Test?"
"OK To Continue", "CANCEL To Abort"

The test is performed by checking that a PCMCIA card is inserted. Then the battery is checked CARD_BAT and CART_WP. If the test is accepted to continue then the SRAM is checked. The test is performed by first write/read 0 then 0x5a and finally 0xa5. The area that is tested includes all 65 Kbyte.

"Testing SRAM"
"Test Done"
"Continuos Test"

It is possible to make a continuos test for reading and writing to the SRAM.

Possible errors:
"Error PCMCIA", "Not Detected"
"Battery: Warning"
"Battery: Bad"
"Error PCMCIA W/P"
"Error Addr <adr>", "Write 00H Read <val>"
"Error Addr <adr>", "Write 5AH Read <val>"
"Error Addr <adr>", "Write A5H Read <val>"

Test case:	**Display:**	**Procedure**

TESTALL_DSPHOST: *"DSP HOST"*
checkTestAllDSPhost(Dsptst.c)

"DSP host"
"Reset DSP"
"Load DSP", "Host Test"
"Testing Host", "Wait…."

This test is performed by loading DSP test code into the DSP processor. The following sequence is called *resetDSP*, *loadDSP* with DSPTEST0.ASM. These procedures will test the host interface (HI08) to the DSP processor. Finally 256 words (x%58 for 255 < x < 512) are written with *writeDSPmem* to the DSP memory. The words are read back from the DSP memory with *readDSPmem* and compared with the written values.

“Test DONE”, “HOST OK”

Possible errors:
“Error Resetting”
“Error Loading”
“Bank Done Error”
“Error Host3”
“Error Host4”

Test case:	**Display:**	**Procedure**
TESTALL_ASICDPRAM:	*“ASIC DPRAM”*	*checkASICdpram(Dsptst.c)*

“ASIC DPRAM”
“Reset DSP”
“Load DSP”, “DPRAM”
“Testing DPRAM”, “Wait….”

This test is performed by loading the DSP test code DSPTESET0.ASM into the DSP processor. Test data is used for testing (x%58 for 255 < x < 512). The Dual Port ram in the ASIC is tested. The result is read and displayed.

“Errs = <VAR_ERR1> <VAR_ERR2>”
“Mask = <VAR_MASK1> <VAR_MASK2>

Possible errors:
“Error Resetting”
“Error Loading”
“Bank Done Error”
“Error Host3”
“Error Host5”
“Error Testing”

Test case:	**Display:**	**Procedure:**
TESTALL_ASICMEM:	*“ASIC MEM”*	*checkASICmem*

“DSP host”
“Reset DSP”
“Load DSP”, “Host Test”
“Testing host”, “Wait….”

This test is performed by loading the DSP test code DSPTESET0.ASM into the DSP processor. Test data is used for testing (x%58 for 255 < x < 512). The memory is tested and the result is displayed.

“ERRS: <VAR_ERR1> <VAR_ERR2>”
“MASK: <VAR_MASK1> <VAR_MASK2>”
“ADDR: <VAR_ADDR1> <VAR_ADDR2>”
“DATA: <VAR_DATA1> <VAR_DATA2>”

Possible errors:
“Error Resetting”
“Error Loading”
“Bank Done Error”
“Error Host8”

Test case:	**Display:**	**Procedure**
TESTALL_ASICSRAM:	*"ASIC SRAM"*	*checkASICsram*

"DSP SRAM"
"Reset DSP"
"Load DSP", "ASIC SRAM
"Slow SRAM", "Wait…."

This test is performed by loading the DSP test code DSPTESET01.ASM into the DSP processor. A sine table is used for testing. (Sin(2πι/256), 0 < ι < 256). The sram of the ASIC is tested and the result is displayed.

"Test Done", "SRAM OK"

If errors the continuos testing is performed if accepted.

"Errs = <VAR_ERR1>, <VAR_ERR2>"
"Mask = <VAR_MASK1>, <VAR_MASK2>"

Possible errors:
"Error Resetting"
"Error Loading"
"Bank Done Error"
"Error Host6"

Test case:	**Display:**	**Procedure**
TESTALL_ASICTPRAM:	*"ASIC TPRAM"*	*checkASICsram*

"DSP TPRAM"
"Reset DSP"
"Load DSP", "ASIC TPRAM"
"Slow TPRAM", "Wait…."

This test is performed by loading the DSP test code DSPTESET04.ASM into the DSP processor. A sine table is used for testing. (Sin(2πι/256), 0 < ι < 256). The tpram of the ASIC is tested and the result is displayed.

"Test Done", "TPRAM OK"

If errors the continuos testing is performed if accepted.

"Errs = <VAR_ERR1>, <VAR_ERR2>"
"Mask = <VAR_MASK1>, <VAR_MASK2>"

Possible errors:
"Error Resetting"
"Error Loading"
"Bank Done Error"
"Error Host6"

Test case:	**Display:**	**Procedure**
TESTALL_ASICFIFO:	*"ASIC FIFO"*	*checkASICfifo*

"ASIC FIFO"

"Reset DSP"
"Load DSP", "ASIC FIFO"
"ASIC FIFO", "Wait...."

This test is performed by loading the DSP test code DSPTESET02.ASM into the DSP processor. Then a sine function is written to the dsp memory on the addresses 0x100 – 0x200. The speed and io is set and the test is started. The IO_INFO is read from the dsp and if there is an AESERR1 . Then the VAL_PHASE_VAL is set (0-23) errors are cleared and waits 50 ms for the phase to settle. Then it wait again 50 ms and the VAR_ERR1 and VAR_ERR2 is tested. The test is repeated for all 24 phase of the sine function.

Possible errors:
"Error Resetting"
"Error Loading"
"Bank Done Error"

Test case:	**Display:**	**Procedure**
TESTALL_LED:	*"LEDS"*	*testLEDs*

"LED Test"
"All Leds On"

All leds are turned on and off when the test is completed.

Test case:	**Display:**	**Procedure**
TESTALL_FLASH:	*"FLASH"*	*testFlash*

"FLASH Test"
"AMD_AM29F499AB"

The test is performed by calling *flashExecute* and reading the flash type.

Test case:	**Display:**	**Procedure**
TESTALL_PCMCIA_FAST:	*"PCMCIA2"*	*checkTestAllPCMCIA*

"PCMCIA Test"
"Battery: Good"
"Continue Test?"
"OK To Continue", "CANCEL To Abort"

The test is performed by checking that a PCMCIA card is inserted. Then the battery is checked CARD_BAT and CART_WP. If the test is accepted to continue then the SRAM is checked. The test is performed by first write/read 0 then 0x5a and finally 0xa5. The area that is tested includes only 1 Kbyte.

"Testing SRAM"
"Test Done"
"Continuos Test"

It is possible to make a continuos test for reading and writing to the SRAM.

Possible errors:
"Error PCMCIA", "Not Detected"
"Battery: Warning"
"Battery: Bad"
"Error PCMCIA W/P"
"Error Addr <adr>", "Write 00H Read <val>"
"Error Addr <adr>", "Write 5AH Read <val>"
"Error Addr <adr>", "Write A5H Read <val>"

Test case:	**Display:**	**Procedure**
TESTALL_PCMCIA_BOX:	*"PCM.BOX"*	*checkOurPCMCIA*

"PCMCIA Command Test"

The PCMCIA card is inserted and the commands from the dual port is displayed.

"CMD String: <cmd>"

Test case:	**Display:**	**Procedure**
TESTALL_CLOCKSPEED:	*"CLOCK"*	*selectClockSpeed*

"Clock Speed"
"OK If No Error"
"Clock: <dspclock>"

It is possible to manual increase the dsp clock speed in the interval between 40 –195.

Test case:	**Display:**	**Procedure**
TESTALL_UPDATESERIAL:	*"S/N UPDATE"*	*updateSerial*

"Device: <name>"
"Press OK To select"

First the *selectDevice* procedure is called that makes it possible to setting the device type.

"Use ADJUST And CURSOR"
"Keys To Enter New S/N"
"S/N: <serial>"
"Press OK When Done"

Then the programSN and getSerial is called that makes it possible to enter the serial number. When accepted the device and serial number is written into the Eeprom.

"Update completed"
"<device>: <serial>"

The showResult procedure is called that displayes the Eeprom data: device name (M2000, G-Force, Viking5, Finalizer ect.) and serial number (1200000).

Name	Serial numbers	Device Type
M2000:	11xxxxx	M2000
G-Force:	14xxxxx	GUITAR2

Viking5:	15xxxxx	VIKING5
Viking6:	16xxxxx	VIKING6
Viking7:	17xxxxx	VIKING7
M3000:	18xxxxx	VIKING8
Viking9:	19xxxxx	VIKING9
D22:	20xxxxx	VIKING10
Finalizer:	12xxxxx	MD2
DB-MAX:	13xxxxx	DBMAX

Test case:	**Display:**	**Procedure**
TESTALL_SELFTEST:	*"SELFTEST"*	*selfTest; showTestAll*

"TestAll V8.10"
"BUILT-IN TEST Vx.x"
"Select with wheel"
"and then press OK"

"Test: <name>"

This is the standard test that is included in all wizard applications. A new test menu is shown and it allows the user to perform the following tests. The *selfTest* is located in Api/Selftest.c. Depending on the device type it is some of the following tests that can be performed.

TEST	Display	Procedure
DISPLAY	"Display"	*checkDisplay (Testlib.c)*
KEY	"Keys"	*checkKeys (Frntchk.c)*
GAIN_CONTROL	"DELAY+CTRL"	*checkGainControllers (Frntchk.c)*
ADJUST_WHEEL	"ADJUST wheel"	*checkEncoder (Frntchk.c)*
LED	"LEDs"	*checkLeds (Frntchk.c)*

ANALOG ***performConnectorTest*(CONN_ANALOG) (Printchk.c)**
This procedure calls *checkAnalog* that performs a *startTest*(TEST_FIFO). Depending on the device type the test procedure is performed. An analog output is connected to an analog input. The VCA levels is set and the mute is turned off. The procedure *changeDSPio*(ANA_ROUTING, AESIN_XLR, SRCCLK_480) is called. The db levels are shown on the front leds and the levels are checked to be –12db.

DIGITAL ***preformConnectorTest*(CONN_DIGITAL) (Printchk.c)**
This procedure calls *checkDigital* that performs the *startTest*(TEST_FIFO). Depending on the device type some of the following test procedures are performed. AES digital out to input, SPDIF digital out to input or ADAT digital out to input. The *changeDSPio*(DIG_ROUTING, *dIOType*[dtest], SRCCLK_480) is called. The db levels are shown on the front leds and the levels are checked to be 0 db.

MIDI ***performConnectorTest*(CONN_MIDI) (Printchk.c)**
The procedure calls *checkMIDI* or checkRS485 depending on the device type. The input buffer is emptied. Then 0xc0 and *mmiData* (0-127) is transmitted, received and verified.

PEDAL ***performConnectorTest*(CONN_PEDAL) (Printchk.c)**
The procedure call *checkPedal*, A jack plug must be connected and trip to ground. The pedal voltage is displayed.

PCMCIA ***checkPCMCIA (Printchk.c)***
The procedures are called *performPCMCIAcheck* and *checkPCMCIAdata*. It is checked if the card is inserted, that the battery is ok and it is not write protected. Then the memory is cleared. Finally 0x5a and 0xa5 are written and verified. The test is done for the address area 0 – 65536.

BATTERY ***checkBattery (Printchk.c)***
The voltage of the battery is measured by calling getBatteryVoltage. If the voltage is less than 2,5 V then an error is reported.

SYSTEM ***checkSystem (Printchk.c)***
This procedure performs the test of the Eeprom (checkEEdata), DSP (fastDSPTest), and extended Ram (quickERamTest) depending on device type.

The files Frntchk.c and Printchk.c contains all the procedures to test the front panel and system board.

Test case:	Display:	Procedure
TESTALL_VIK4TEST:	*"WIZ2TEST"*	*vik4Test; showTestAll*

Performs the *vik4Test* in module vik4test.c. Performed by starting DSPTEST5. Details about the vik4Test see next chapter.

Test case:	Display:	Procedure
TESTALL_EXTRAM:	*"EXT.RAM"*	*checkExternRam*

Performed by starting DSPTEST6 with downloaded test data (256 < x < 256+58).

1.1.1 The *vik4Test* procedure is located in Api/vik4Test.c

This test performs the ASIC test of analog and digital IO. It is used for testing the hardware concerning the functions of the ASIC DARC 3. The dsp is loaded with the test software DSPTEST5. This software performs a test of the bank A and B located in Y memory the test is also updating the values of inputs VAR_RPPM, VAR_RPPM_OUT, VAR_LPPM and VAR_LPPM_OUT used to update the peak meters for signal inputs and outputs (Before and after gain). The variable VAR_GAIN is multiplied with the input samples. That is used by the H8 application to test the processing of the input signal. The general purpose memory can be tested by setting VAR_GPMTEST external X memory (0x4500). The vik4Test starts to initialise a number of parameters used by the dsp test program.

InitVolatile : Initialisation of memory variables for fifo buffers, digital and analog input/outputs. Levels etc.
InitLevels: Setting the R/L I/Olevels for inputs controlled by the H8.
InitIoSetup: Setting the dsp variables: IO_SETUP, DIO_SETUP, VAR_FIFO_VAL, VAR_NEW_FIFO_SEL
InitTestResult: Setting the dsp variable: VAR_TEST_MODE
InitGPMTest: Writing test data into the X memory from address 0x100 to 0x134.
WriteSineTable: Writing a sine table into the Y memory from address 0x100 to 0x1FD.

The test is then started and errors are cleared in the dsp testprogram. The thread *updDisp* is started and running concurrently. This thread is updating the display, leds and meter levels by reading the updated VAR_LPPM and VAR_RPPM values from the DSP. The content of the ASIC IO_INFO is also displayed.

The following test can be selected and performed from the wizard display:

Display	Procedure
"LEVELS"	*editLevels*

Resets all the level values to zero then allows the user to modify the level parameters. The cursor keys are used to select parameter and the potentiometer is used to change the values. The H8 port B is used to sending the data to the serial register (4094 and AD8403) that controls the gain output pins (DA_DEEMP0, DA_DEEMP1, OUTPUT_BYPASS, IN_ATT)

"IN L"	Attenuation (0-255) of analog left input
"IN R"	Attenuation (0-255) of analog right input
"OUT L"	Attenuation (0-255) of digital left input
"OUT R"	Attenuation (0-255) of digital right input
"ATT/94"	Setting attenuation level on the inputs (Depending on device)
"RLYBYP"	Turning the bypass relay on/off. (Analog input -> output)
"DEEMP0"	Setting the control signal to the DB level on digital output (DEM0)
"DEEMP1"	Setting the control signal to the DB level on digital output (DEM1)
"DIG IN"	Change of the dsp parameter VAL_GAIN_VAL. (Conversion from DB)
"METERS"	Setting the display of the led meter to be input or output

Display	**Procedure**
"IO/DIO"	*editIo, editDio, editInput*

This test contains the possibility to read and modify the ASIC registers: I/O Setup, Digital I/O Setup and Input-information located at 0x5160 – 0x5162.

The *editIo* contains the following configuration parameters I/O setup 0x5160:

"SRC CLK"	Change the sample rate: 44100, 4800, DIGIN (From AES), EXT (External)
"FIFO1"	Input to the fifo1 buffer from analog or digital inputs: AN, DIG
"FIFO2"	Input to the fifo2 buffer from analog or digital inputs: AN, DIG
"FIFO3"	Input to the fifo3 buffer from analog or digital inputs: AN, DIG
"DA1 OUT"	Analog output 1 taken from buffer: MUTED, FIFO1, FIFO2 or FIFO3
"DA2 OUT"	Analog output 2 taken from buffer: MUTED, FIFO1, FIFO2 or FIFO3
"DA3 OUT"	Analog output 3 taken from buffer: MUTED, FIFO1, FIFO2 or FIFO3
"AES OUT"	AES outout taken from buffer: MUTED, FIFO1, FIFO2 or FIFO3
"ADAT OUT"	ADAT output taken from buffer: MUTED, FIFO1, FIFO2 or FIFO3
"Y2 OUT"	Yahama 2 output taken from buffer: MUTED, FIFO1, FIFO2 or FIFO3
"SOUNDRAM"	Is soundram static or dynamic ram: SRAM or DRAM
"AES IN"	From where is AES input taken: XLR (Prof.), SPDIF (Con.) or OPTO
"OPTO OUT"	How is the opto interface routed: LOOP (In -> Out), ADAT or AES input
"PLL CLK"	-75 (40 – 75 Mhz) or +75 (95 – 150 Mhz)
"VCO 1 / 2"	VCO1 chose (40 – 130 Mhz) or VCO2 chosen (130 – 230 Mhz)
"48K/96K"	Sample rate relay (Depending on device). Same as ATT/94 output

Display	**Procedure**

The *editDio* contains the following configuration parameters digitial I/O setup 0x5161:

"ADAT LI"	ADAT left input channel selector (1-8)
"ADAT RI"	ADAT right input channel selector (1-8)
"ADAT LO"	ADAT left output channel selector (1-8)
"ADAT RO"	ADAT right output channel selector (1-8)
"ADATPLL"	Changes form freq. PLL to phase PLL on first, second or third sync.
"DIG IN"	Setting the digital input: AES, SRC (Clock), ADAT or Y2
"CBLSYNC"	Disable or enable Rx/Tx channel status block syncronisation (ADAT)
"AUTOCRC"	Automatic CRC-generation on AES-transmission
"DPACCESS"	Access to Dual Port RAM's from ASIC disable/enable
"Y2 SEL"	Y2 transmitter data shift selector

"DSP GPM"	Setting the VAR_GPMTEST parameter to start the dsp test
"ERRORS"	Number of errors seen during test
"MASK"	Last error seen by xor of write an read value
"I/O BUFFER"	Setting the dsp parameter VAR_FIFO_VAL to FIFO1/2/3_L. Specifies the input fifo buffer that the dsp program are reading the samples from
"DSPCLOCK"	Setting the dsp speed (40 – 110 Mhz)

Display	**Procedure**

The *editInput* contains the following input information form address 0x5162:

"AES IN"	When the clock is synchronised from the AES input. LOCK / UNLOCK
"AES ERR"	Error in the AES-signal (No lock, parity error or biphase coding error)
"VALID."	Validity bit LOW or HIGH
"CSC"	CSC-error occurred in received AES-signal since last read
"PARITY"	Parity-error occurred in received AES-signal.
"BIPHASE"	Biphase coding error occurred in AES-receiver since last read
"RES LCK"	Out of lock situation occurred in AES-receiver since last read
"DIGRATE"	Digital sample rate in received signal
"ADAT IN"	When the clock is synchronised from the ADAT input LOCK/UNLOCK
"ADATLOCK"	Out of lock situation occurred in ADAT-receive since last read
"Y2 IN"	Yahama input LOCK/UNLOCK
"HIFS CLK"	HIFS clock LOCK/UNLOCK

Display	**Procedure**
"CHSTAT"	*showChStatIn*
"CH AND"	*editChStatMask(0)*
"CH OR"	*editChStatMask(1)*
"TEST"	*showTestResult*

Displays the result of the dsp testing of the bank memories. It is possible to select between SLOW and FAST test the VAR_TEST_MODE is set to 0 (slow) and 1 (fast). The result of VAR_ERR1 and VAR_MASK1 is displayed during the test. The index, mode, time and mask are displayed.

Display	**Procedure**
"MIC. RLY"	*editMicRelays*

This test performs the control of the microphone relays. (Only for Gold channel) The settings are shifted from the H8 port B into the AD8403 relay.

"R MIC/LN"	Select right MIC/LINE input (bit 2)
"L MIC/LN"	Select left MIC/LINE input (bit 3)
"L PHANTM"	Select left phantom ON/OFF (bit 4)
"R PHANTM"	Select right phantom ON/OFF (bit 5)
"L GAIN"	Select left gain PAD/x1 (bit 6)
"L PAD+2"	Select left pad +20/+40 (bit 7)
"R GAIN"	Select right gain PAD/x1 (bit 8)
"R PAD+20"	Select right pad +20/+40 (bit 9)
"LED PATN"	Setting the led scan rate