Document Revision 1.0 / 31-12-2010

Hi there

I have had a STM32VL-Discoveryboard for some time , but hasn't used it as you needed Windows in order to use the onboard Jtag/Programmer.

As i mainly use Linux (Ubuntu 10.04 LTS for now) , i was disapointed with ST not supporting their debugger on Linux.

I found an opensource project called Versaloon http://www.versaloon.com/ , where Simon was using the exact same chip , as the one used for Jtag on the STM32VL.

I finally have succeded in adapting Versaloon to run SWD programming , on the embedded ARM Jtag on the STM8S-Discovery or STM32VL-Discovery boards.

For now i can only get vsprog to work , but Simon is working on adapting the OpenOCD driver to the new USB_TO_XXX scheme used in the latest versaloon versions.
http://www.versaloon.com/bbs/viewtopic.php?p=43#p43

I have reported my problems with OOCD , and given some hints & output here.
http://www.versaloon.com/bbs/viewtopic.php?p=50#p50

Remember this WILL erase the onboard ST-Link software , and afaik there is no way to restore it. But if you are using Linux/*nix , the ST-Link software isn't supported anyways

On the STM8S board i have used the SWIM_IN and SWIM_RST_IN pins , as they are protected by 220 ohm resistors. I was a bit worried about the 680ohm pullup on SWIM , but it doesnt seem to give any problems. The SWIM & SWIM_RST pins are set to INPUT-Floating.

SWD pins on the STM8S CN7 plug are like this :

1: VDD (Unused)
2: SWDIO/T_JTMS
3: GND
4: SWCLK/T_JTCK
Remember to connect GND to the target , and you can even "steal" 3v3 for the target , from pin-1 on the JTAG.

On the STM32VL board **i have reversed the "Original" ST-Link pins used for SWD.**
This is due to the way Versaloon operates SWDIO/T_JTMS , where i suppose Simon used on pin as permanently input for speed reasons.
And as the build-in JTAG used 2 pins for T_JTCK , it made sense to use those for T_JTMS.

So the pinout is like this (Reversed from whats shown in the schematic) , and **DONT connect the jumpers on CN3 according to the manual.**

STM32VL STANDALONE Programmer
When jumpers/wires are removed from CN3 , in order to use the JTAG as a standalone unit.

The pinout on CN2 is like this :
1: GND via 10K ??? (Unused)
2: SWDIO/T_JTMS
3: GND
4: SWCLK/T_JTCK

STM32VL ONBOARD Programmer
When using the onboard JTAG to program the onboard target chip (the value-line chip) , you can't place the two CN3 jumpers as the manual says.

The CN3 Pins are like this :
1: T_JTMS (From onboard JTAG , differs from schematic)
2: SWCLK (To Onboard value-line chip)
3: T_JTCK (From onboard JTAG , differs from schematic)
4: SWDIO (To Onboard value-line chip)

What i did was connect a jumper on Pin 2-3 , and a short Test-wire on pin 1-4

Modifying the STM8S

HW:
Just solder a standard 0.1" pinheader in the 7 empty holes below the usb connector.
I used a 2x4 block where i pulled out one pin (pin2) , easy to see why.

Connect an ARM Jtag to the soldered plug , and "Unlock/Program" it.
I used a signalyzer-lite and OpenOCD (from SVN ... Well actually Simons script) from here
http://www.versaloon.com/bbs/viewtopic.php?p=43#p43

I just modified openocd_update , and added --enable-ft2232_libftdi to the "./configure" around line 80 , so i had support for both hw dongles.
[code]
./configure --enable-vsllink --enable-ft2232_libftdi --enable-maintainer-mode
[/code]

You will need "libftdi devel" installed on your distro (>= 1.17 is needed for FT2232H)
http://www.intra2net.com/en/developer/libftdi/
On Ubuntu just use Synaptic , and you'll get >= 1.17

SW:
1: Start OOCD
sudo openocd -f interface/signalyzer-lite.cfg -f target/stm32.cfg

2:
telnet localhost 4444

3 (Once) : stm32x unlock 0

4:
reset halt
flash probe 0
stm32x mass_erase 0

flash write_bank 0 Versaloon_GCC.bin 0

disconnect & connect the USB cable on the STM8S
You are done :-)

Modifying the STM32VL

This was a bit more difficult (programming) , as the JTAG pins weren't connected to any headers. But there was access to SWD on the backside of the PCB. Via some Solderbridges (SB)
SB6 is STM_JTCK (SWCLK)
SB10 is STM_JTMS (SWDIO)
So i soldered some thin wires on the part of the solderbridge that was closest to the pinheaders.

At first i had hoped to use my newly build STM8S SWD progger to program the JTAG chip on the STM32 , but no matter what i did i couldb't get
vsprog to do it. vsprog reckognized the chip , but couldn't read/write it. I found out the chip was locked/protected , and as the "OOCD" i build above , using Simons script. Still have some problems with the vsllink hardware , i gave up on that.

I booted into windows , and used a "Genuine ST-Link" to unprotect & flash the JTAG chip.
Now ... I actually wonder if another STM32VL would have worked instead of a "Genuine ST-Link"

When i booted back into Linux the onboard JTAG worked fine , and i now could program the value-line chip on the board from the onboard JTAG.

Note !!!
For both the above solutions , there is another option :
Use the buildin serial bootloader (if you can solder wires on a chip w. 0.5mm spacing)
Serial-Loader program : http://code.google.com/p/stm32flash/
http://gostm32.blogspot.com/2010/09/better-program-for-bootloader.html

Try it out if you like
Bingo from AvrFreaks