Antti-Brain

Issue 13

August 2009

Editorial

xxx

Antti Lukats
Antti.Lukats@googlemail.com

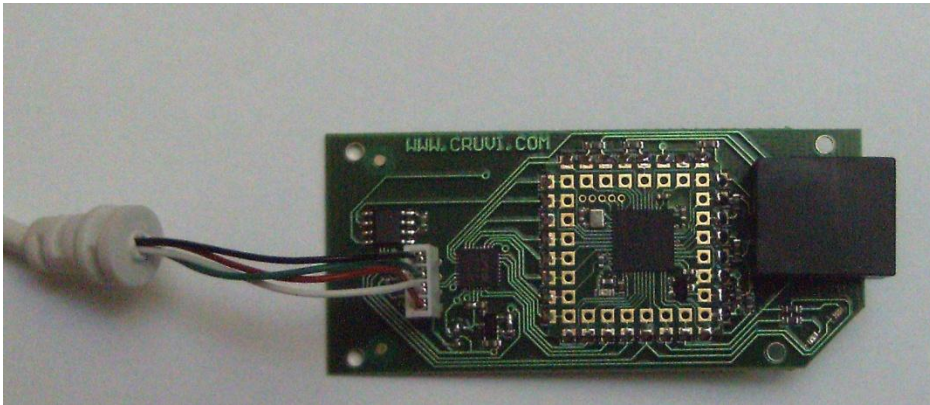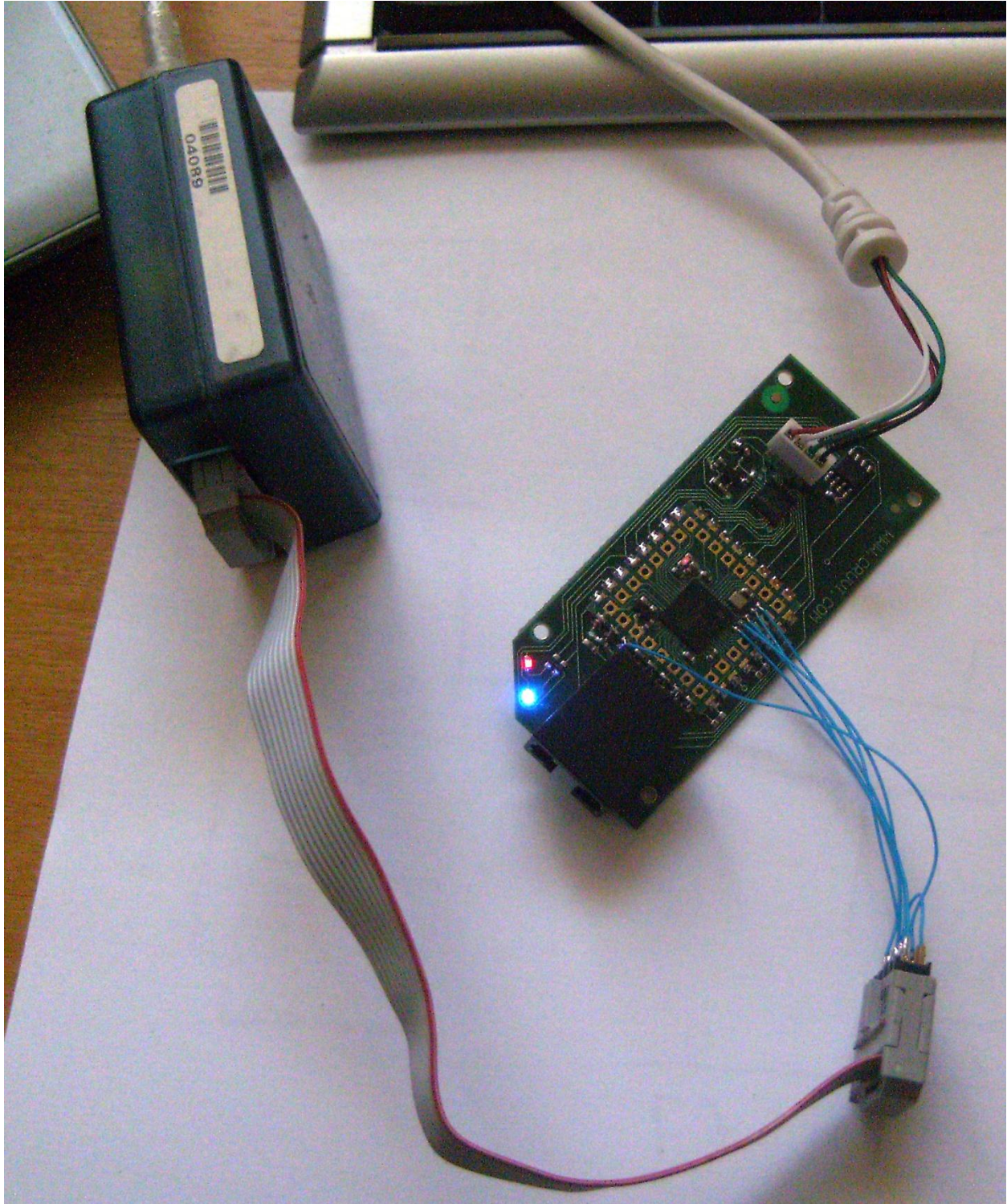http://groups.google.com/group/antti-brain

Cover Story

xxx

U2TOOL

Something I wanted to-do long time ago. Something I should have done long time ago.

Here it is! Not as beautiful, not as featured as I wanted it to be, but it's here on my desk now.

Universal USB Tool, hardware revision A. The U2TOOL Base board is fitted with A3P060 based ST32 Stamp module.

U2TOOL-FT245 with ST32ACT (A3P060 FPGA) first test successful, the LED's are connected to FTDI data port and the data pins are set to high using bit-bang mode. It just worked, first time, made HDL code, assigned I/O's programmed FPGA, and started my FTDI test application, click and LED's are on or off as set by the software.

Connecting some CRUVI™ LED board to the cut-half piece of network cable, plugging into the RJ45 jack. Running FTDI's Mprog, selecting "altera_usb_blaster.ept", erase, program ok! Now adding USB Blaster HDL code, connecting the JTAG interface pins to some pins in RJ45 jack. Running Libero, programming ok. Launching Quartus, tools -> programmer, selecting Altera USB Blaster, trying auto-scan, fail! Well sure there is no JTAG device connected, saying yes I want JTAG Debugger, clicking run, ha some LED's blink showing that Quartus JTAG debugger is trying to scan the chain, and that real outputs are toggling.

How nice, I now have my own USB Blaster – finally!

Oh, sure the thing I wanted is not USB Blaster of course, but some nice piece of programmable hardware, that can be USB Blaster as well. The 99 other uses are to be defined.

But does it really work? I have an Stratix board, let's try it out, hum 9V supply? Ok, I have one, but hey what's that, Altera plug has reverse polarity? No, I don't fry the Stratix just with the attempt to check the new programming gadget. Ok, I make RJ45 to Actel JTAG adapter cable, just wires. Connecting to A3P060-Stamp on Cruvi™ baseboard. Quartus programmer, scan, yes JTAG ID is read well.

But, could we also program the Actel using Quartus? Let's add the SVF programming file. Not possible, what's that? After some RTFM it sure is so, Quartus can generate SVF but not play them back at all. Ok, let's use the Actel STAPL file then. Renaming stp to jam, adding file, error!

[some troubleshooting here]

1) Actel SVF file, convert to JAM using Altera s2j_23.exe
2) Remove FREQUENCY and CRC lines using text editor
3) **quartus_jli.exe actel.jam –a RUN_FILE**
4) Success! A3P060 was really burned ok, within 1 min 15 seconds!

Wau, how nice. And it seems U2TOOL works pretty fast, Actel FlashPro what is high speed USB needs about 1 minute to flash the A3P060, U2TOOL with Altera jam player executed in 1.15 (but U2TOOL is full speed only).

Next steps:

There is small EEPROM on board, how to connect it the best way? Ah, I use the OE (output enable) bit from USB Blaster protocol, and connect the internal EEPROM when external bus is tri-stated.

Done, FPGA updated, PC software, writing test for 93LC46, 12 lines of code, press F9, executing, click "Test EEPROM", works! Cool.

Now, we want to program SPI memories as well, right? Soldering the another half of the cut-half RJ45 cable to 100mil header that matches my 6-pin micro-SD socket adapter.

Another 15 lines of Code, running PC test to read AT45DB161D ID Code. This cannot be, it works! Well, at least the return value is 0x26 what is correct device ID for 161D, but how come that 26 is also read when there is no IC? My mistake, the SPI xfer function did not return a value, so that 0x26 was just

random number from PC stack, funny it happened to match the ID of the SPI flash chip I used for testing. Fixing the software, starting again, and yes now full ID is read ok, 0x1F, 0x26. And for smaller chip it reads 0x1F, 0x25 so it is not a mistake any more.

It's almost boring when things work that easy. Ok, maybe I cheated just a little, there was some occasional troubleshooting in between, but it was really not much to fix. Just some typographic mistakes done by me.

FPGA FM Transmitter III

Audio-DSP's II

C

AL3101 vs. TAS3xxx

c

Branching and conditions

AL3101 has no JMP instructions at all, so all 1024 instructions are executed (fetched!) once each audio frame. There is a special command that checks flags, and sets SKIP count, forcing a number of instructions to be skipped (that is to be executed as NOP's).

TAS3xxx has JMP instruction with optional condition check. JMP's consume 4 words and 4 clocks, as each JMP instruction has to be padded with 3 NOP's (cannot be filled with other code!).

TAS3xxx also has memory loads with condition that could be considered as 0 overhead skips. So conditional load on TAS3xxx is one instruction (or actually less than 1 !), while on AL3101 it is two instructions.

AL3101 has circular sample RAM access mode for filters, where a user inaccessible frame counter is added to the RAM address, those allowing to access "past samples" as needed for filter algorithm. But hey, AL3101 can process 8 channels, but that frame counter is always incremented by one? How is that possible channels would overwrite the values? How silly, the answer is so simple ☺ Let's assume we have 2 channels (stereo), so each frame we read left sample, and write to RAM address 0, then we read right sample, and write to RAM address 512. Now we have portioned the sample RAM for 2 channels, and there is no overwrite, both channels have 512 sample buffers. It's so simple, I feel stupid that I had to think about it.

My Own Processor V

xxx

Organize my Life™

xxx

Shorts

Stories.

Lattice support or FTDI chips

Starting from ispVM 17.5 there is support for FTDI FT2232 based USB Programming added. The MachXO Control kit does not anymore have Cypress USB for cable but uses FTDI solution.

This is good news, as it may make it possible to use cheap FT2232 cables as official Lattice USB Cable. FT2232 is directly connected to JTAG there is no extra CPLD used.

References

- http://www.trioflex.com
- http://www.cruvi.com

Instead of adding the URL links at the end of each issue, I will be adding them to the Trioflex online link collection, so they can be updated more frequently.