

- ↓ Info
- ↓ AXIS ETRAX 100LX MCM4+16
 - ↓ SDK 2.20
- ↓ AXIS ETRAX FS
 - ↓ SDK 2.20
- ↓ ETRAX FS, SDK 2.20, fast configuration step-by-step
- ↓ Flash TRB v1:
- ↓ Flash TRB v2
- ↓ Flash TRB v2 permanently
- ↓ to start DAQ for 1 TRB:
- ↓ to start DAQ for 3 TRB:
- ↓ New Software Development Kit (SDK) for TRB:
 - ↓ ↓ ↓ How to compile the application for the board
 - ↓ How to compile the new kernel
 - ↓ In this page, I'm using linux directory as linux-2.6.
 - ↓ **VERY IMPORTANT:**
 - ↓ **HINTS:**
 - ↓ **SCRIPTS:**
 - ↓ **HISTORY OF KERNEL:**
 - ↓ **WARNINGS**
 - ↓ **Kernel Module**
- ↓ SDK for Etrax FS:
- ↓ Additional tips needed for development board:
- ↓ Second version of dev_system -> only RAM
- ↓ Push new kernel FS from RAM and flash after
- ↓ additional settings for Etrax FS

Info

- Where to find
 - Etrax FS fimage for flashing:
 - hadaq@depc187:/home/hadaq/etrax_soft/etrax_fs/devboard-R2_20/
 - Etrax FS fimage for debugging:
 - hadaq@hadeb05:/home/hadaq/soft/new/etrax_fs/devboard-R2_20/
 - kernel patch advised by Axis was applied but there is still problem with copying big files to a flash.
 - Etrax MCM fimage for flashing:
 - hadaq@hadeb05:/home/hadaq/soft/new/etrax_mcm/gen_image/out_image/etrax2??/
- MAC addresses
 - ETRAX FS boards: etrax[000-099]=00:40:8C:AD:E5:[01-99], for example etrax001=00:40:8C:AD:E5:01
 - ETRAX FS boards: etrax[100-199]=00:40:8C:AD:E6:[01-99], for example etrax100=00:40:8C:AD:E6:00
 - ETRAX MCM boards: etrax[201-250]=00:40:8C:AD:E5:[A1-F0]

AXIS ETRAX 100LX MCM4+16

SDK 2.20

- Path: hadeb05:/home/hadaq/soft/new/etrax_mcm/devboard-R2_20/
- Linux kernel 2.6.26

Installation:

- source init_env

```
export AXIS_TOP_DIR=`command pwd`
echo "Prepending \"${AXIS_TOP_DIR}/tools/build/bin\" to PATH."
export PATH="${AXIS_TOP_DIR}/tools/build/bin:$PATH"
echo "Prepending \"/home/hadaq/soft/cris/bin\" to PATH."
export PATH="/home/hadaq/soft/cris/bin:$PATH"
export AXIS_KERNEL_DIR="${AXIS_TOP_DIR}/os/linux-2.6"
```

- ./configure
- make menuconfig
- Do settings discribed below.
- Do kernel configuration if needed (some drivers might be missing)
 - cd os/linux-2.6/
 - make menuconfig
 - Change settings
- ./configure
- make
- Look at the final target system in target/cris-axis-linux-gnu/

Settings:

- packages/initscripts/init_rc/rc.
 - This script executed rc.local.

```
#!/bin/sh
. /etc/init.d/functions.sh
begin "Init /etc/rc.local"
while [ ! -e "/etc/rc.local" ]
do
    sleep 1
done
/etc/rc.local
end $?
```

- Do not forget Makefile and **.target-makefrag**: AXIS_BUILDTYPE=cris-axis-linux-gnu
- In configure-files/common/common put at the very end the following:

```
# start 'rc' script after all init script
sub packages/initscripts/init_rc
# needed for NFS
sub packages/initscripts/portmap
```

- packages/filesystem/
 - packages/filesystem/rc.local. This script has hard-coded MAC, gets IP and nfs-server name via DHCP (nfs-server IP goes to /etc/resolv.conf), mounts /home/hadaq via nfs.

```
#!/bin/sh
MAC="00:40:8C:CD:00:02"
#/etc/init.d/net.eth0 stop
ip link set eth0 address $MAC
ip link set eth0 up
export DHCP=1
/etc/init.d/net.eth0 restart
sleep 5
nfs_ip=`tail -n1 /etc/resolv.conf | awk '{print $2}'`
ping -c 2 ${nfs_ip}
mount -v -o proto=tcp,rsz=32768,wsz=32768
${nfs_ip}:/var/diskless/etrax /home/hadaq
sleep 1
if [ ! -e "/home/hadaq/etc" ] ; then
    mount -v -o proto=tcp,rsz=32768,wsz=32768
    ${nfs_ip}:/var/diskless/etrax /home/hadaq
fi
sleep 1
. /home/hadaq/etc/rc
```

- Files **pmap_dump** **pmap_set** **portmap** must be in packages/filesystem/. Needed for NFS RPC.
- packages/filesystem/Makefile. This Makefile contains all file system structure.

```
$(INSTALL) $(OWN) -m 0777 rc.local $(prefix)/etc
$(INSTALL) $(OWN) -d $(prefix)/home
$(INSTALL) $(OWN) -d $(prefix)/home/hadaq
$(INSTALL) $(OWN) -m 0777 portmap $(prefix)/sbin
$(INSTALL) $(OWN) -m 0777 pmap_set $(prefix)/sbin
$(INSTALL) $(OWN) -m 0777 pmap_dump $(prefix)/sbin
```

- Partitions: one must be careful with changing sizes of partitions. This can lead to an error and force read-only file system during boot. Total size is 4 MB.
- Network settings:
 - To enable DHCP check packages/initscripts/net.eth0/conf. It must contain **BOOTPROTO=dhcp**.
 - Default MAC is set in packages/initscripts/mac/conf. But MAC given in packages/filesystem/rc.local is used.
- Time synch via ntpdate (this is done for R2_20)
 - Executable: devboard-R2_20/apps/ntpdate/. To compile automatically copy [compile.pl](#) to apps/ntpdate/ and run the script. Or compile by hand:
 - First compile libntp: you should not compile systime_s.c to avoid multiple definition errors. Then compile ntpdate.
 - Copy ntpdate to /var/diskless/etrax/bin/.
 - packages/filesystem/rc.local => /home/hadaq/etc/rc :

```
#!/bin/sh
/home/hadaq/bin/ntpdate $NFS_IP
export PATH="$PATH:/home/hadaq:/home/hadaq/bin:/home/hadaq/scripts"
```

- Prompt

- packages/shell/ash/profile/prompt/prompt.sh: PS1="[\$HOSTNAME:\$PWD]"
- HOSTNAME
 - packages/initscripts/hostname/conf: HOSTNAME="etrax101"

Howto flash image:

On hadeb05 under root execute: **boot_linux -d eth1 -F -i ./fimage**. Where eth1 - GSI network, eth0 - local network.

Howto produce images for many Etrax MCM:

- cd /home/hadaq/soft/new/etrax_mcm/gen_image/
- To produce images for etrax101,102,103 execute perl script: ./gen_image.pl -e 101 -e 102 -e 103
 - Script will replace MAC address in rc.local and mac/conf, set HOSTNAME and reconfigure and recompile kernel
 - The final images will go to /home/hadaq/soft/new/etrax_mcm/gen_image/out_image/etrax10?

AXIS ETRAX FS

SDK 2.20

- Path: hadeb05:/home/hadaq/soft/devboard-R2_20/
 - Linux kernel 2.6.26
- Path: depc187:/home/hadaq/etrax_soft/etrax_fs/devboard-R2_20/
 - Linux kernel 2.6.26

Scheme of mapping temp sensor numbers to MAC addresses:

- packages/filesystem/id2macTable : map of temp sensor numbers to MAC addresses.
- packages/filesystem/tempid2mac : shell script to extract MAC address from a map.
- packages/filesystem/tempsens->../apps/tempsens/tempsens : binary for extracting temp sensor number.

Settings:

- packages/filesystem/
 - Files **pmap_dump pmap_set portmap** must be in packages/filesystem/. Needed for NFS RPC.
 - Also script tempid2mac to get MAC address from id2macTable must be in packages/filesystem/.

```
#!/bin/sh
TEMPID=`tempsens | grep ID | awk '{print $2}'`      #get ID from app:
tempsens
LINE=`cat /etc/id2macTable | awk '{if($1==tempid) print $2}'`
tempid=$TEMPID`
echo $LINE
```

- Make soft links: tempsens -> ../apps/tempsens/tempsens; id2macTable -> ../id2macTable
- packages/filesystem/rc.local. This script has hard-coded MAC, gets IP and nfs-server name via DHCP (nfs-server IP goes to /etc/resolv.conf), mounts /home/hadaq via nfs. Export NFS_IP for time synch via ntpdate.

```
#!/bin/sh
while [ ! -e "/sbin/tempid2mac" ]
do
    sleep 1
done
MAC=`tempid2mac`
/etc/init.d/net.eth0 stop
ip link set eth0 address $MAC
ip link set eth0 up
export DHCP=1
/etc/init.d/net.eth0 restart
sleep 5
nfs_ip=`tail -n1 /etc/resolv.conf |awk '{print $2}'`
mount -v ${nfs_ip}:/var/diskless/etrax_fs /home/hadaq
mount -o size=4M -t tmpfs tmpfs /dev/shm
sleep 1
if [ ! -e "/home/hadaq/etc" ] ; then
    mount -v ${nfs_ip}:/var/diskless/etrax_fs /home/hadaq
fi
export NFS_IP=${nfs_ip}
sleep 1
. /home/hadaq/etc/rc
```

- Add to the end of packages/filesystem/Makefile:

```
$(INSTALL) $(OWN) -m 0777 portmap      $(prefix)/sbin
$(INSTALL) $(OWN) -m 0777 pmap_set      $(prefix)/sbin
$(INSTALL) $(OWN) -m 0777 pmap_dump     $(prefix)/sbin
$(INSTALL) $(OWN) -d                    $(prefix)/home/hadaq
$(INSTALL) $(OWN) -m 0777 id2macTable   $(prefix)/etc
$(INSTALL) $(OWN) -m 0777 rc.local      $(prefix)/etc
$(INSTALL) $(OWN) -m 0777 tempsens      $(prefix)/sbin
$(INSTALL) $(OWN) -m 0777 tempid2mac    $(prefix)/sbin
```

- Add init_rc and portmap to configure-files/common/common like it is done for ETRAX MCM
- Login (bash shell sequence):
 - /etc/profile (packages/shell/ash/profile/common/profile)

```
ENV=/etc/shrc
HOSTNAME=`/home/hadaq/bin/hostname`
export ENV HOSTNAME
. /home/hadaq/etc/profile
```

- /etc/shrc

```

for FILE in /etc/profile.d/*.sh ; do
    if [ -x $FILE ]; then
        . $FILE
    fi
done
unset FILE

```

- /etc/profile.d/path.sh (packages/shell/ash/profile/path/path.sh)

```

PATH=/usr/local/bin:/usr/bin:/bin:/home/hadaq:/home/hadaq/bin:/home/hadaq/scripts
[ $(id -u) -eq 0 ] && PATH="/usr/local/sbin:/usr/sbin:/sbin:$PATH"
export PATH

```

- /etc/profile.d/prompt.sh (packages/shell/ash/profile/prompt/prompt.sh)
- /home/hadaq/etc/profile (lxhadesdaq:/var/diskless/etrax_fs/etc/profile)

```

export HOSTNAME=`/home/hadaq/bin/hostname`
export HOSTIP=`/home/hadaq/bin/hostip`

```

- Time synch via ntpdate
 - Executable: devboard-R2_20/apps/ntpdate/. To compile automatically copy [compile.pl](#) to apps/ntpdate/ and run the script. Or compile by hand:
 - First compile libntp: you should not compile systime_s.c to avoid multiple definition errors. Then compile ntpdate.
 - Copy ntpdate to /var/diskless/etrax/bin/.
 - packages/filesystem/rc.local => /home/hadaq/etc/rc :

```

#!/bin/sh
/home/hadaq/bin/ntpdate $NFS_IP
export PATH="$PATH:/home/hadaq:/home/hadaq/bin:/home/hadaq/scripts"

```

- Readout with DMA needs the following change:
 - packages/udev/iop_fw_load-R1_1_0/iop_fw_load

```

#!/bin/sh
HOTPLUG_FW_DIR=/home/hadaq/firmware
#HOTPLUG_FW_DIR=/usr/lib/hotplug/firmware
echo 1 > /sys/$DEVPATH/loading
cat $HOTPLUG_FW_DIR/$FIRMWARE > /sys/$DEVPATH/data
echo 0 > /sys/$DEVPATH/loading
exit 0

```

Howto flash image:

- On hadeb05
 - Login as root; **cd /home/hadaq/soft/devboard-R2_20/; source init_env**
 - Switch BOOTSEL to 'A' (netboot) on the TRB and reset the board.
 - Execute: **boot_etraxfs -d eth0 -F -i ./fimage**. Where eth1 - GSI network, eth0 - local network.
 - Switch BOOTSEL to '0' and reset the TRB.
 - Check on the server that the board's MAC address is in the DHCP config file and IP is in /etc/hosts.
- On hadesdaq02 for TRBs in cave in HADES VLAN
 - Login as root; **cd /home/hadaq/etrax_soft/etrax_fs/devboard-R2_20/; source init_env**
 - fimage should be copied to hadesdaq02 beforehand. **On hadesdaq02 you cannot build fimage, you can only flash!**
 - Switch BOOTSEL to 'A' and use **boot_etraxfs d eth1 -F -i ./fimage** to flash fimage. After flashing switch BOOTSEL to '0'.
 - Once you have fimage with proper block sizes flashed, you can use FTP to upgrade kernel and file system (you do not need to switch BOOTSEL):
 - ftp etrap054
 - login as root
 - Run command: **put fimage flash_all**

-- [SergeyYurevich](#) - 16 Dec 2009

ETRAX FS, SDK 2.20, fast configuration step-by-step

1. Download tarballs (devboard-R2_20.tar.gz, cris-dist-1.64-1.i386.rpm) from http://www.axis.com/products/dev_sdk/download_dist.php
2. Unpack them and install
3. cd devboard-R2_20/; source init_env; ./configure
4. Select which product configuration to use: 6. fs_nor
5. For flash memory used in TRBv2:
 - Use [flash.c](#) instead of ./devboard-R2_20/tools/build/fsboot/cbl/flash.c
 - Use [cfi_probe.c](#) instead of ./devboard-R2_20/os/linux-2.6/drivers/mtd/chips/cfi_probe.c
 - Use [axisflashmap.c](#) instead of ./devboard-R2_20/os/linux-2.6/arch/cris/arch-v32/drivers/axisflashmap.c
 - Use [readinode.c](#) instead of ./devboard-R2_20/os/linux-2.6/fs/jffs2/readinode.c
 - Use [config](#) instead of ./devboard-R2_20/os/linux-2.6/.config
 - Use [axis_config.h](#) instead of ./devboard-R2_20/axis_config.h
6. All NOR-flashes Axis have used have 64 kByte erase blocks while our flash chip ([PC28F256P30B85](#) Intel StrataFlash Embedded Memory) has 128 kByte erase blocks. This means that we have to patch with [configure_erase_block.patch](#) the AXIS SDK to be able to set 128 kByte erase blocks.
 - cd devboard-R2_20/; patch -p 0 < configure_erase_block.patch

After patching we have to set 'Flash memory block size' to 131072 Bytes or 0x20000 in hex (when running 'make menuconfig'). This will generate new Makefile with --eraseblock=131072 option:

```

$SORTDIR mkfs.jffs2 --little-endian \
    --pagesize=$(PAGE_SIZE) \
    --cleanmarkers --eraseblock=0x20000 \
    --pad=$(PSIZE_jffs2_0) \
    --root=. > $(INAME_jffs2_0) ; \

```

1. cd devboard-R2_20/; make menuconfig
 - Hardware Configuration
 - Processor ETRAX FS
 - (0x20000) Flash memory block size (bytes)
 - [*] Partition table
 - Partition table configuration:
 - (0x020000) Rescue partition size

- (0x340000) Flash1 size
 - (0x4A0000) Flash2 size
 - (1) MTD partition number for rescue partition
 - (2) MTD partition number for kernel partition
 - (4) MTD partition number for rootfs partition
 - (3) MTD partition number for rwfs partition
 - [*] USB Device Filesystem
 - (A1) Control button address
 - General Configuration
 - Linux kernel (Linux 2.6.x)
 - Kernel serial console port (Serial port 0)
 - [*] Enable udev
 - Shell (dash)
 - [*] Start a console shell
 - Network Hardware Configuration
 - [*] Enable Ethernet support
 - (1) Network interfaces
 - (eth0) Device name for interface 1
 - Network Protocol Configuration
 - [*] Enable DHCP support
 - TELNETD support (Enable utelnetd support)
 - Network Application Configuration
 - [*] Enable ipsetd support
 - SSH support (No ssh support)
 - [*] Enable FTP client support
 - [*] Enable IPtables support
 - Debug Configuration
 - [*] Enable strace support
 - Libraries Configuration
 - [*] Enable OpenSSL support
 - Standard C library (glibc)
2. cd apps/; tar -xvf tempsens.tar; cd apps/tempsens; make clean; make
 3. cd packages/initscripts; tar -xvf init_rc.tar
 4. cd packages/initscripts; tar -xvf portmap.tar
 5. Add init_rc, portmap to configure-files/common/common like described above for ETRAX MCM
 6. cp id2macTable devboard-R2_20/.
 7. cd packages/filesystem/
 - ln -s ../../apps/tempsens/tempsens tempsens
 - ln -s ../../id2macTable id2macTable
 8. Install stuff needed for NFS:
 - cd apps;
 - cvs -d :ext:hadaq@lxi003.gsi.de:/misc/hadesprojects/daq/cvsroot checkout etrax_apps_r20/tcp_wrappers_7.6
 - cd tcp_wrappers_7.6/; Adapt paths in Makefile
 - make linux
 - cd apps/etrax_apps_r20/; ln -s tcp_wrappers_7.6 tcp_wrappers
 - cvs -d :ext:hadaq@lxi003.gsi.de:/misc/hadesprojects/daq/cvsroot checkout etrax_apps_r20/portmap_5beta
 - cd portmap_5beta/; Adapt paths in Makefile
 - make
 - cp pmap_dump pmap_set portmap packages/filesystem/.
 9. Prepare packages/filesystem/
 - cd packages/filesystem/
 - cp rc.local packages/filesystem/
 - cp tempid2mac packages/filesystem/
 - ln -s ../../apps/tempsens/tempsens tempsens

- In -s ../id2macTable id2macTable
- Add these changes to packages/filesystem/Makefile

```
$(INSTALL) $(OWN) -m 0777 rc.local      $(prefix)/etc
$(INSTALL) $(OWN) -d                    $(prefix)/home
$(INSTALL) $(OWN) -d                    $(prefix)/home/hadaq
$(INSTALL) $(OWN) -m 0777 portmap       $(prefix)/sbin
$(INSTALL) $(OWN) -m 0777 pmap_set      $(prefix)/sbin
$(INSTALL) $(OWN) -m 0777 pmap_dump    $(prefix)/sbin
$(INSTALL) $(OWN) -m 0777 id2macTable   $(prefix)/etc
$(INSTALL) $(OWN) -m 0777 rc.local      $(prefix)/etc
$(INSTALL) $(OWN) -m 0777 tempsens      $(prefix)/sbin
$(INSTALL) $(OWN) -m 0777 tempid2mac    $(prefix)/sbin
```

10. ./configure

11. make

If you want to change only partition sizes then it is enough to run in the main directory:

- Change partition sizes: **make menuconfig**
- Change a byte-offset in kernel configuration (0x020000 gives a byte-offset of 131072):
 - **make -C packages/os/linux-2.6 menuconfig**
 - Drivers for built-in interfaces ---> (131072) Byte-offset of partition table sector
- Build a new kernel: **make -C packages/os/linux-2.6 install**
- **./configure**
- **make images**

-- [SergeyYurevich](#) - 15 Dec 2009

Flash TRB v1:

1. to take 'root' permission, type `su`,
2. change directory for `hadeb05:/home/hadaq/soft/devboard-R2_01/`
3. type `. init_env`
4. set SW1 on the board to 3 [from 7]
5. `./flashit -d eth1 -i ./de-etrax0XX/fimage`, where XX is number of TRB
6. set SW1 on the board to 7 [from 3]

Flash TRB v2

1. to take 'root' permission, type `su`,
2. type `. init_env`
3. change directory for `hadeb05:/home/hadaq/soft/devboard-fsv2/`
4. `boot_etrafs -d eth1 -i ./de-etrax0XX/kimage`, where XX is number of TRB

Flash TRB v2 permanently

1. to take 'root' permission, type `su`,
2. type `. init_env`
3. change directory for `hadeb05:/home/hadaq/soft/devboard-R2_10/`
4. `boot_etrafs -d eth0 -F -i ./fimage`
5. set SW1 on the board to 0 [from A]

to start DAQ for 1 TRB:

1. `trb_ip` - contains the IP of trb for other scripts,
2. `INIT_FPGA` - to program FPGA,
3. `RESET_TDC` - to reset TDCs on the TRB [before that one has to reset CTU],
4. `START_READOUT` - to start readout on TRB [after that one has to start CTU],
5. `EVTBUILDER` - open eventbuilder, netmem and sniffer.

to start DAQ for 3 TRB:

1. `trb_ip`, `trb_ip2`, `trb_ip3`,
2. `INIT_FPGA_FOR_ALL`,
3. `RESET_TDC_FOR_ALL`,
4. `START_READOUT_FOR_ALL`,
5. `EVTBUILDER_FOR_ALL`.

another things:

1. `LOG_IN` - to log into trb.

New Software Development Kit (SDK) for TRB:

- Working directory: `radek@depc187:~/devboard-R2_1`
- Set environment: `. init_env`
- Voltage switch in the lab in GSI: <http://haepc02.gsi.de>, user: admin, pass: oscypeczek 😊

How to compile the application for the board

1. You have to create new directory in the `~/devboard-work_2.4/apps/`
2. Type: `make cris-axis-linux-gnu` in the `~/devboard-work_2.4/` directory
3. Copy some `Makefile` for example from `~/devboard-work_2.4/apps/hello/` to your directory and modify it
4. Write the application
5. Copy binary file to the `/home/hadaq` on the board, by typing: `scp binary_file hadaq@lxxhadesdaq:/var/diskless/etrax/`

How to compile the new kernel

- A. Go to the `radek@depc187:~/devboard-work_2.4/os/linux` directory
- B. Change the kernel `make menuconfig`
- C. Use the script `~/devboard-work_2.4/os/linux/mrproper.sh COMMENT_WORD`. This script creates the copy of your `.config` file, configures the target and makes images. You have to only flash it to the board.

In this page, I'm using linux directory as linux-2.6.

- to change something one should use:
 - configuration scripts: `./configure [-v]`; `./make menuconfig`; `./os/linux/make menuconfig`; `./make`
 - configuration files `./packages`; `./configure-files/`
 - remember that `./Makefile` is generated by `./configure` script

VERY IMPORTANT:

❗ In the `c./onfigure-files/common/common` one has to remove all lines with the 'status_led' or 'set_led', because we don't have any default diodes on the portB, you can do this by changing `./configure-files/common/common` file.

❗ Turn off DHCP in `./packages/initscripts/net.eth0/conf` -> comment the line:

`DHCP_CLIENT="/sbin/udhcpc -i eth0 -H $(hostname)"` and set `BOOTPROTO="none"`

❗ To flash any image, log in as 'radek' on the 'depc187.gsi.de', go to the `./devboard-work_2.4` directory, set the environment, by typing `. init_env`, and now you can use `./flashitall`

HINTS:

💡 If something is strange(new settings in kernel doesn't work(especial the waitstate during the `./flashit`)), use:
make mrproper !!! %BR The best way to be ensure that we put good version of kernel **FOR NOW** :

```
./os/linux/make menuconfig
cp .config config
make mrproper
cp config .config
make menuconfig= =exit with any changes
cd ../../
cp os/linux/.config ./kernelconfig
./configure
make
```

see `./os/linux/mrproper.sh`

💡 If you want to change IP number, you have to 'flashit'

💡 if `Verification...Error at 0x8000002` if you put the kernel to the `exis_board`: you are usind wrong

`CO:wqNFIG_ETRAX_DEF_R_WAITSTATES`. It should be `9aa6`

💡 somethimes you have to 'make menuconfig' in `os/linux` (if you see some errors in `os/linux` directory)

💡 one dont has to do 'make' in `os/linux`. Instead this one has to 'make' in `./devboard` (but somethimes it helps)

💡 if you want to use shared memory, add the proper line:

```
partition fstab type=tmpfs dev=tmpfs mpoint=/dev/shm mopts=defaults dump=0 pass=0
```

to the `./configure-files/common/common` file

and add `$(INSTALL) -d $(DEV)/shm` to `./packages/devices/standard/Makefile`

and add the proper line to the `./packages/filesystem/Makefile` 💡 if you want to use device `mem`,put the line: `$(MKMOD) -m 0644 $(DEV)/mem c 1 1` to `./packages/devices/axis-2.4/Makefile`

💡 to remove some initialize scripts, one has to comment the proper line in the `./Makefile` in the section `SUBDIRS`. This line will be start from `packages/initscripts`.

💡 if you want to add some directory to the structure, see `./packages/filesystem/Makefile`

💡 to change size of partiions, use `./make menuconfig` and `Hardware Configuration` section

💡 to use NFS:

1. put in the file `./configure-files/common/common` to the section `Images`, `partition table` and `fstab` the following line:

```
partition fstab type=nfs dev=140.181.75.158:/var/diskless/etrax mpoint=/home/hadaq
mopts=noexec,noauto dump=0 pass=0
```

1. In the `./configure-files/common/common` file add the line `sub packages/initscripts/mount_nfs`
2. Copy `mountall-R1_0_1` into `mount_nfs` in the `./packages/initscripts/` directory. Enter to this and change the `install:` section in the `Makefile` to this one:

```
$(INSTALL) $(OWN) -m 0755 rc $(prefix)/etc/init.d/mount_nfs
$(LN) -sf ../init.d/mount_nfs "$(prefix)"/etc/rc3.d/S99mount_nfs
```

and in the `rc` file the proper line:

```
begin "Mounting network filesystem"
mount -v /home/hadaq
```

💡 If you want change the **PATH**, or add the **MANPATH** variable in the `init_env` script, go to the file and see `./configure-files/source/functions` to the section `function path`.

see manual by typing: `man -l ./configure-files/source/documentation/man/man3/functions.3`

💡 If you want to change MAC address, see `packages/initscripts/mac/rc` and `packages/initscripts/mac/conf`, see [scripts](#)

💡 If you are using 'flashtail', you erase `SERNO` (serial number) variable,. You can put it by `bootblocktool -a SERNO 00408CCD00XX` on the board.(put the real numbers for XX

💡 If you want to change the `PATH` variable on the board, see `./packages/initscripts/common/linuxrc`

💡 If you want to change `ptablespec` go to the file `configure-files/common/common` to the section `Images`, `partition table` and `fstab`. For example, if you want to change permission, change `csum` to 'no'

💡 If you want to change something in the environment, add e.g. `PATH` , add the line `PATH=/home/hadaq/:$PATH` to the `packages/shell/ash/profile/common/profile`

💡 If you want to add some file to the filesystem on the board, see `packages/filesystem/Makefile`

💡 If you get `segmentation fault`, when you start `daq_evtbuilder` probably, shared memory was not set. Check taht `/dev/shm` exist and `tmpfs` is mounted there

💡 The size Flash Memory is 4MB. It is divided into Rescue, Flsh1 and Flash2 partitions. You can change the size of these partitions in he followong steps:

1. type: `make menuconfig` in the main directory,
2. enter to the `Hardware configuration`

Remember the sum of all partitions couldn't be more than size of the Flash Memory

💡 `./reconfigure` script uses the `DEV_BOARD_PRODUCT` to set the type of the `AXIS` product. I put by hand this variable to this script as `devboard_82`

💡 every changes in `./devboard-work/make menuconfig` are stored in `axis_config.h` and in `.config` files

SCRIPTS:

🔗 `./changeSettingsTo.sh` HOST Prepare the particular settings for the HOST, which will be used in `./make`

🔗 `./os/linux/mrproper.sh` `COMMENT_WORD` Use this scripts after `./os/linux/make menuconfig` to be sure that you are using proper version of kernel

HISTORY OF KERNEL:

1. change from old 2.4 kernel
2. disable the `SCSI` and `USB` options

WARNINGS

- A. There is no portmap on the board(therefore we are using `nfs` with `-nolock` option)

Kernel Module

💡 There are exported kernel symbols in the `/proc/ksyms` file

💡 If in your library, `make` was created symbols with `_`, instead of symbols without `_`, you have to add `-mlinux` option to your `Makefile`

SDK for Etrax FS:

Compiling `cris-dist-1.63-v32`:

1. Problem with `format_reg`, `format_sup_reg`, `print_with_operands` you have to change the lines in the file

`binutils/opcodes/cris-dis.c`

```
static char *format_reg
    PARAMS ((struct cris_disasm_data *, int, char *, enum cris_disass_family));
static char *format_sup_reg
    PARAMS ((unsigned int, char *, enum cris_disass_family));
static void print_with_operands
    PARAMS ((const struct cris_opcode *, unsigned int, unsigned char *,
             bfd_vma, disassemble_info *, const struct cris_opcode *,
             unsigned int, unsigned char *, enum cris_disass_family));
```

to

```
static char *format_reg
    PARAMS ((struct cris_disasm_data *, int, char *, boolean));
static char *format_sup_reg
    PARAMS ((unsigned int, char *, boolean));
static void print_with_operands
    PARAMS ((const struct cris_opcode *, unsigned int, unsigned char *,
             bfd_vma, disassemble_info *, const struct cris_opcode *,
             unsigned int, unsigned char *, boolean));
```

1. Add to the `$PATH` the path to the new `cris-v32` crosscompiler.
2. Install a fresh SDK use the `./install` script. When asked which product to configure for press enter to use the default.
3. Run the `make menuconfig` in section *Hardware Configuration* change the *Processor* for `=ETRAX FS`.
4. Run the `configure` script to download all packages. If *configure* says the *compiler is too new* -> edit `./configure-files/source/functions` and comment the lines 2339-2342.
5. Copy a FS enabled [kernelconfig-2.6](#) to the SDK root directory.
6. `make init_env && . init_env`
7. `make -C packages/os/linux-2.6 crisv32-axis-linux-gnu`
8. `make -C packages/os/linux-2.6 menuconfig`. Remember in section *Hardware setup* the *Processor type* should be `ETRAX-FS-V32` and *DRAM size* = 128.
9. `make`
10. To flash the board use the `boot_etraxfs` script.
 1. To write the whole flash using the host NIC eth1: `boot_etraxfs -F -i /path/to/fimage -d eth1;`
 2. To write the whole flash except the rescue partition use: `boot_etraxfs -f -i /path/to/fimage;`
 3. To upload the kernel image into ram and execute it from there use: `boot_etraxfs -i /path/to/kimage;`

It describes on the web: http://developer.axis.com/wiki/doku.php?id=sd2.01_for_fs


💡 I have to force 10Mb connection, because my server was too busy > `=ethtool -s eth0 autoneg off speed 10 duplex full`

Additional tips needed for development board:

1. Modify `CONFIG_ETRAX_FLASH_BUSWIDTH` to 4 in `kernelconfig`
 - 💡 If you couldn't compile `os` directory, unchanged it and try again
 - When I tried this, the kernel is not compiling anymore! And I even found in the sourcecode, that it is not implemented, so keep it at 2
2. Modify `CONFIG_ETRAX_SDRAM_GRP0_CONFIG` to 0x156c in `kernelconfig`
3. In `packages/romfs_meta/common/Makefile` remove the line than contains `mnt.romfs_meta.txt`

4. Modify the file `packages/romfs_meta/common/mnt.romfs_meta.txt` by removing the whole contents of the file and rebuild the package with `make -C packages/romfs_meta/common install`. (This causes the `/etc` directory to be part of the cramfs file system instead of being writable).
5. Modify the file `packages/initscripts/common/linuxrc` by commenting out the whole section starting with `MOUNT=`, ending with the first `'fi'` after that and rebuild the package with `make -C packages/initscripts/common install`. (This stops the init script from trying to mount the jffs2 filesystem).
6. Overwrite `packages/os/linux-2.6/Makefile` with file in the Attachment

- **Makefile:** `packages/os/linux-2.6/Makefile`

1. Modify `os/linux-2.6/arch/cris/drivers/axisflashmap.c` in the function `init_axis_flash`:
`mymtb->read(mymtd, CONFIG_ETRAX_PTABE_SECTOR, 512, &len, page); --> mymtb=NULL`
`mymtd = flash_probe(); --> mymtb = NULL; this is not necessary ?`
2. Modify the kernel as follows: do `make -C packages/os/linux-2.6 menuconfig`. In the second General Setup menu, modify the Kernel command line so that the root device is `mtdblock0` instead of `mtdblock3`. (This causes linux to search for the init script in the first mtd device, rather than the fourth, which is consistent with a RAM-only image). I.e. the new kernel command line is `'root=/dev/mtdblock0 init=/linuxrc'`. Rebuild the kernel with `make -C packages/os/linux-2.6 install`.
3. Recreate all images with `'make images'`
 To enable port A (program FPGA via jam) modify `CONFIG_ETRAX_PA_BUTTON_BITMASK` for `0x0`, `CONFIG_ETRAX_PA_CHANGEABLE_DIR` for `0x0F` and `CONFIG_ETRAX_PA_CHANGEABLE_BITS` for `0xFF`

http://developer.axis.com/wiki/doku.php?id=32_bits_wide_flash

Second version od dev_system -> only RAM

According page: http://developer.axis.com/wiki/doku.php?id=sd2.01_for_fs:

1. `install`
2. `make menuconfig`, change Hardware Configuration > =Processor for ETRAX_FS,
3. `./configure`,
4. `make init_env && . init_env`
5. overwrite [kernelconfig-2.6](#) in the main directory,
6. `make -C packages/os/linux-2.6 crisv32-axis-linux-gnu`,
7. `make -C packages/os/linux-2.6 menuconfig`,
8. `make`
 According page:http://developer.axis.com/wiki/doku.php?id=32_bits_wide_flash
9. remove the line: `$(INSTALL) -m 0644 mnt.romfs_meta.txt $(prefix)/mnt/romfs_meta.txt` from `packages/romfs_meta/common/Makefile`
10. modify the file `packages/romfs_meta/common/mnt.romfs_meta.txt` by removing the whole contents of the file (Ignore contents: flash).
11. `make -C packages/romfs_meta/common install`
12. comment out the section `MOUNT` in file `packages/initscripts/common/linuxrc`
13. `make -C packages/initscripts/common install`
14. `make -C packages/os/linux-2.6 menuconfig`:
 - A. => second 'General Setup' => 'Kernel command line' -> `"root=/dev/mtdblock1 init=/linuxrc"`
 - B. => 'Hardware setup' -> `CONFIG_ETRAX_FLASH_BUSWIDTH = 4 -> SDRAM_GRP0_CONFIG = 0x156c`
 - C. => 18. bit in `CONFIG_ETRAX_MEM_GRP1_CONFIG = 0`
 - D. => "Drivers for built-in interfaces" -> 'Axis flash-map support' = on, `CONFIG_ETRAX_PTABE_SECTOR = 131072`
 - E. => 'Memory Technology Devices' => 'RAM/ROM/Flash chip drivers' -> `CONFIG_MTD_CFI_ADV_OPTIONS = on`, -> `CONFIG_MTD_CFI_GEOMETRY = on`, -> only `CONFIG_MTD_MAP_BANK_WIDTH_4` and `CONFIG_MTD_CFI_I2` is on. -> 'Support for Intel/Sharp flash chips' = on,
15. overwrite the [Makefile](#) in the `./packages/os/linux-2.6/Makefile`,
16. overwrite [axisflashmap.c](#) in the `./os/linux-2.6/arch/cris/arch/drivers/axisflashmap.c`
17. modify `os/linux-2.6/arch/cris/drivers/axisflashmap.c` in the function `init_axis_flash`:
 comment out the line:

```
mymttd->read(mymtd, CONFIG_ETRAX_PTABLE_SECTOR, 512, &len, page);
```

The line: `mymttd = flash_probe();` change to `mymttd = NULL;`

1. `make -C packages/os/linux-2.6 install`
2. `make`

Modify `os/linux-2.6/arch/cris/drivers/axisflashmap.c` in the function `init_axis_flash`:

`mymttd = flash_probe(); --> mymttd = NULL;` **this is not necessary ?**

Push new kernel FS from RAM and flash after

- [kimage_onlyram](#): kimage for booting to RAM

1. `cd hadaq@hadeb05:~/soft/devboard-fs_onlyram`
2. `su` to have root permission
3. `. init_env`
4. switch the `BOOT SELECT1` to Network RT/TX (it's 5),
5. `boot_etraxfs -d eth0 -i ./kimage`
6. `cd ~/soft/devboard-fs_32v2`
7. `ftp 192.168.0.90`, type the login and password,
8. `ftp> cd tmp`
9. `ftp> put fimage`
10. `ftp> exit`
11. `telnet 192.168.0.90`, type the login and password,
12. `cd /tmp`
13. `eraseflash -v /dev/cflash0`
14. `cat /tmp/fimage > /dev/cflash0`
15. switch the `BOOT SELECT1` to NOR Flash (it's 0),
16. reset the board

additional settings for Etrax FS

1. to force changing MAC address I changed `packages/initscripts/mac/rc` and `packages/initscripts/mac/Makefile`,

-- [RadekTrebacz](#) - 26 Sep 2005

| I | Anhang | Aktion | Größe | Datum | Wer | Kommentar |
|---|---|---------------------------|--------|---------------------|--------------------------------|--|
|  | Makefile | verwalten | 4.9 K | 19 Sep 2006 - 12:29 | RadekTrebacz | 2. version, put it to: <code>./packages/os/linux-2.6/Makefile</code> |
|  | compile.pl.txt | verwalten | 2.5 K | 09 Jul 2010 - 16:17 | SergeyYurevich | script to compile ntpdate for Etrax FS/MCM |
|  | configure_erase_block.patch | verwalten | 5.6 K | 27 May 2010 - 12:59 | SergeyYurevich | patch for 128 kByte erase blocks in our Flash chip |
|  | kernelconfig-2.6 | verwalten | 13.0 K | 19 Sep 2006 - 12:36 | RadekTrebacz | version 2: put in the main directory |

Dieses Topic: [DaqSlowControl](#) > [TDCReadoutBoard](#) > [EmbeddedLinuxDev](#)

Topic-Revision: 09 Jul 2010, [SergeyYurevich](#)

