

```
#####
#   setup and test procedure  for NAOMI software release naomi-0.2  #
#   at   WHT La Palma                                     #
#####   X.Gao 10~ 19 /08/20000 #####
```

to set ENV for Naomi on navis

```
source /software/naomi/naomi-0.2/config/setup.csh
```

to startup QI

```
source /software/naomi_atc_src/naomiView/naomiView_login
naomiQI_xg
```

to use iraf

```
?? source /opt/iraf/bin/irafuser.csh
?? mkdir iraf
?? cd iraf
?? mkiraf ( terminal:xterm)
?? ximtool &
?? xgterm &
?? [in the xgterm] cl
```

to start a readout after login navis (refer to WFS_READOUT_CONTROL_V3.doc for change parameter)

1). Start QI first (see above or source naomiset_xg, it is the same)
this will bring you up the SAOtng and QI_sever.

2). open another xterm
rlogin naomidsu
this gets you into the Vxwork shell

3). To start full frame ccd readout as default, using Vxwork, QI_server, SAOtng display

```
#####
```

```
?? !! if you want to change Vxwork from startup without epics to startup with epics, reboot the Vxwork first ( type reboot in VxWork Shell),  
then begin the following procedure. If you quit the SAOtng, before restart QI as in 1). kill QI_server first by use ps -ef | grep xg, which  
will display something like:
```

```
xg 23235      1  0 12:30:30 pts/2      0:11 /software/naomi_atc_src/naomiView/bin/QI_server naomiView 59999
```

```
?? type kill 23235 , then 1).
```

```
?? If there is something wrong, and you want to restart, always follow: reboot Vxwork, press reset button of VME I/F board, switch off/on the  
SDSU controller power supplier. Quit SAOtng, use ps -ef | grep xg, kill Qi_server, the start from 1).
```

```
#####
```

to start a readout without epics

```
<startup_cam_no_epics
```

```
<readout_cam1 or <close1 to stop readout
```

```
<readout_cam2 or <close2 to stop readout
```

```
?? cam1 refer to Master, can2 refer to Slave. During the unsynchronised readout mode, you have to stop cam1 before to start cam2,  
vice versa,
```

```
?? once readout is running, you should be able to see the image displayed on SAOtng. if any thing wrong, try to reboot VME and  
press reset of VME I/F board. then do from 2).
```

to start a readout with epics

```
<startup_cam
```

```
<readout_epics_cam1 or <close1_epics to stop readout
```

```
<readout_epics_cam2 or <close2_epics to stop readout
```

```
?? cam1 refer to Master, can2 refer to Slave. During the unsynchronised readout mode, you have to stop cam1 before to start cam2,  
vice versa,
```

use C40, starting with or without epics is the same as above, then use, (see relevant docs. for how to use C40)

```
<readoutC40_cam1      or <close1C40 ;                <readoutC40_cam2      or <close2C40
```

```
<readoutC40_epics_cam1 or <close1C40_epics;          <readoutC40_epics_cam2    or <close2C40_epics
```

4). To start another applications looks similar just changing the scripts accordingly, ie. If you want to use ap_2 for timing board, sending data back to Vxwork, then, just change

```
sdsuMemory ("cam1", 2, "FDL", "../timing/ap_2/master_appl_download lod")
```

```
sdsuCommand ("cam1",2,"LDP",0,0)
```

```
sdsuCommand ("cam1",2,"SYC",0,0)
```

or

```
sdsuCommand ("cam1",2,"LDA",2,0)
```

```
sdsuCommand ("cam1",2,"SYC",0,0)
```

in the readout_cam1.m4 script or save as new script, then **gmake install** (see following **to make change and recompile CCD readout scripts**).

5). For synchronized read out, see example

```
<synch_both or <synch_apl6
```

6). to start a vmedummy-star without epics

```
<startup_cam_no_epics
```

<VMEdummy_star

7). to start a vmedummy-star with epics

<startup_cam

<VMEdummy_star_epics

8). To change parameters, like exposure time, speed. etc, see Derek's ICD: **WFS_READOUT_CONTROL_V3.doc**

Make sure that you are using the right script, load right code

to make change and recompile Vxwork startup scripts

?? you have to login aocontrol1.ing.iac.es!!!

(use gmake or /usr/local/bin/gmake)

```
source /software/naomi/naomi-0.2/config/setup.csh
cd /software/naomi/naomi-0.2/naomiCam/startup/
```

make change to any *.vws file, if not re-writable, make your own file, then only use

gmake

which will display message like:

```
mv167: install
gmake[2]: Entering directory `/usr/software/naomi/naomi-0.2/naomiCam/startup/O.mv167'
Generating VxWorks startup_cam_epics_xg script
Installing /usr/software/naomi/naomi-0.2/naomiCam/.bin/mv167/startup_cam_epics_xg
gmake[2]: Leaving directory `/usr/software/naomi/naomi-0.2/naomiCam/startup/O.mv167'
gmake[1]: Leaving directory `/usr/software/naomi/naomi-0.2/naomiCam/startup'
```

to make change and recompile CCD readout scripts

```
source /software/naomi/naomi-0.2/config/setup.csh
cd /software/naomi/naomi-0.2/naomiCam/dspsrc/scripts
```

make change to any *.m4 file, if not re-writable, make your own file, then only use

gmake

which will display message like:

```
m4 /software/naomi/naomi-0.2/config/naomi.m4scr VMEC40test.m4 > /usr/software/naomi/naomi-0.2/naomiCam/bin/asm56000/scripts/VMEC40test
m4 /software/naomi/naomi-0.2/config/naomi.m4scr VMEcam1Rd.m4 > /usr/software/naomi/naomi-0.2/naomiCam/bin/asm56000/scripts/VMEcam1Rd
```

to recompile DSP code

```
source /software/naomi/naomi-0.2/config/setup.csh
cd /software/naomi/naomi-0.2/naomiCam/vme/ap_*,
```

where you want to change, after modification, type

make install

which will print message like:

```
.applTop/./asm56000/asm56000 -b -d PRG NORM -l vmeboot.lst -i ../ap_boot ../ap_boot/vmeboot.asm
.applTop/./asm56000/asm56000 -b -d PRG NORM -d TEST HEAD -d READ MANUAL -l vme_appl.lst -i ../ap_boot vme_appl.asm
.applTop/./asm56000/dsplnk -b vme_appl.cld -v vme_appl.cln vmeboot.cln
dsplnk: Beginning pass 1
dsplnk: Opening link file vme_appl.cln
dsplnk: Closing link file vme_appl.cln
dsplnk: Opening link file vmeboot.cln
dsplnk: Closing link file vmeboot.cln
dsplnk: Beginning section and symbol relocation
dsplnk: Beginning pass 2
dsplnk: Opening link file vme_appl.cln
dsplnk: Closing link file vme_appl.cln
dsplnk: Opening link file vmeboot.cln
dsplnk: Closing link file vmeboot.cln
.applTop/./asm56000/srec -bs vme_appl.cld
.applTop/./asm56000/asm56000 -b -d PRG DOWNLOAD -d TEST HEAD -d READ MANUAL -l vme_appl.lst -i ../ap_boot vme_appl.asm
.applTop/./asm56000/dsplnk -b vme_appl.cld -v vme_appl.cln vmeboot.cln
dsplnk: Beginning pass 1
dsplnk: Opening link file vme_appl.cln
dsplnk: Closing link file vme_appl.cln
dsplnk: Opening link file vmeboot.cln
dsplnk: Closing link file vmeboot.cln
dsplnk: Beginning section and symbol relocation
dsplnk: Beginning pass 2
dsplnk: Opening link file vme_appl.cln
dsplnk: Closing link file vme_appl.cln
dsplnk: Opening link file vmeboot.cln
dsplnk: Closing link file vmeboot.cln
.applTop/./asm56000/cldlod vme_appl.cld > /usr/software/naomi/naomi-0.2/naomiCam/bin/asm56000/vme/ap_4/vme_appl.lod
```

DSP Software (there are README file in ../dspsrc/timing and ../dspsrc/vme subdirectory)

Timing board DSP software is listed in WFS_READOUT_CONTROL_V3.doc
can be found in aocontrol1 under
/software/naomi/naomi-0.2/naomiCam/dspsrc/timing

VME I/F board DSP software
can be found in aocontrol1 under
/software/naomi/naomi-0.2/naomiCam/dspsrc/vme

- ap_1: VME DSP code for receiving data from Timing board and sending to Vxwork vi VMEBus
script file, see example (in /software/naomi/naomi-0.2/naomiCam/dspsrcscripts) VMEcam1Rd, VMEcam2Rd
VMEcam1_dummy, VMEcam1_mode7, etc.
- ap_2: VME DSP code for receiving data from Timing board and sending to C40
script file, see example (in /software/naomi/naomi-0.2/naomiCam/dspsrcscripts) readoutC40_cam1,
readoutC40_cam2, readoutC40_epics_cam1, etc
- ap_3: VME DSP code for self test, no Timing board required. it sends header and data to Vxwork vi
VMEBus and at the same time to C40 (remember to disconnect jump on SDSU_RS422 RMT_RST/FIFORD.
User specifies all data and header information.
script file, see example (in /software/naomi/naomi-0.2/naomiCam/dspsrcscripts) VMEC40test.
- ap_4: VME DSP code for sending simulated spot image to Vxwork and C40 (do not connect RMT_RST to FIFORD,
so the data can be sent to C40? has not yet been tested for C40)
script file, see example (in /software/naomi/naomi-0.2/naomiCam/dspsrcscripts) VMEdummy_star
- ap_boot: VME DSP boot code for all applications
- ap_56k: VME DSP code for using EVM56k to program the EEPROM chip 29c256