EPICS- device for the AVME9325

Jan. 1994 Kay-Uwe Kasemir
Frankensteiner Weg 7
49078 Osnabrück
0541/46142
kasemir@charon.physik.Uni-Osnabrueck.DE

Contents

1 Intro

Refer to the "Acromag Series 9325 High Speed Analog Input Board With RAM, User's Manual" which will be referred as *User's Manual* for more detailed information.
Several shellvariables should be set [1]:

Variable	Value
SHADOW	/home/heras1/kasemir/EpicsShadow
ACRO	/home/Software/ACROMAG
HSE	/home/Software/epics
EPICS_TARGET	lisa12_162

If you install the Acromag support for the first time, a few links in $HSE will be missing. To install them, you type

```
ln -s $ACRO/src/drvAcro.h   $HSE/share/epicsH
ln -s $ACRO/src/drvAcro.c   $HSE/share/src/drv
ln -s $ACRO/src/devAiAcro.c $HSE/share/src/dev
ln -s $ACRO/src/devWfAcro.c $HSE/share/src/dev
```

If you cannot change a file due to access privileges you have to get a private copy of it from sccs:

```
sccs get <the_file>
chmod +w <the_file>
```

2 Board's Factory Setup

The detailed location of all jumpers is given in: *User's Manual*, Page 2.3. Here are the important jumpers including some changes I made:

- base address (UM 2.3.2), factory setup: 0x800000, J9 11-12 closed

  I changed this setting: J9 9-10 closed, so you may read the J9 bits as decimal "16"

  $\longrightarrow$ base 16*0x4000 = 0x400000

- Address Modes 0x39, 0x3d (`VME_AM_STD_USR_DATA`, `VME_AM_STD_SUP_DATA`): J10 closed

- no IRQ: J7 open

  I changed this setting: J7 1-2, 3-4 closed selects IRQ level 3

- 16 differential inputs: J8 2-3, J2 open

- input range -10V...+9.9951: J3 2-3, J4 1-2, J5 open

- giving 0x8000...0x7FF0 (2th complement, signed short): J1 1-2

[1] C-Shell: setenv Variable Value, Bourne, Korn etc.: export Variable=Value

2.1 Base Address

Here is an example of calculating the real address in VxWorks memory from VME address jumpered on board:

```
cu -lttya
-> x=1
-> sysBusToLocalAdrs (0x39,0x800000, &x)
-> x
value = 0xf0800000
-> sysBusToLocalAdrs (0x3d,0x800000, &x)
-> x
value = 0xf0800000
```

The same way you get a physical address of 0xf0400000 in VxWorks memory space if you jumper 0x400000 on the Acromag board.
In `$CONFIG/sysLib.c` you find:

```
/* a24 vme (address set up in sysHwInit()) */
{
(void *) 0xf0000000,
(void *) 0xf0000000,
          0x1000000,                             /* 16 Mbytes */
         0xf0800000 = x + 0xf0401ebc
....
}
```

This means: your converted address must fit in the range 0xf0000000 to 0xf1000000. With my setup the AVME board can be seen at 0xf0400000, valid addresses to be jumpered must not exceed 0xf1000000.

2.2 Connecting Input

The full P2 connector pin description is *User's Manual*section 2.4, page 2.7. As a first test:

Name	Connector	Pin
Common	P2	9A
Ch.0 +	P2	32A
Ch.0 -	P2	31A

Warning: These pins are directly connected with the multiplex-ICs. Avoid input voltages >10 V or electric shocks.

2.3 LED codes

The build-in LEDs on the board's front panel are under software control and thus have no a-priory meaning. This driver, however, implements the following light-show:

red	green	
on	off	initial state
off	off	driver is initialized but idle
	on	conversion is running

3 Driver

Prior to really installing the driver a warning: Every change to EPICS drivers or devices invalidates the database, so may want to save it.

3.1 Saving and restoring the database

In order to save the actual database layout, invoke dct and follow these steps:

```
"Open a Process Database",
"Exit This Menu",
"Report a Process Database"
```

Now give some name, e.g. "save.rep" and answer 'n' to "long form (Y/N):", after a call to

```
"Delete a Process Database"
```

and installing the new drivers and you can restore the database:

```
"Create a Process Database"
```

give some name, e.g. "dbase0",

```
"Modify from a Text File"
```

give your your report's name, e.g. "save.rep",

```
"Exit DCT"
```

3.2 Configuration

At runtime you may change:

verboseAcro
If not 0, you get some diagnostics.
There are a few define-lines to configure in `$SRC/drvAcro.[ch]`. Refer to 2 for the initial or suggested setup. The first entries must correspond to the hardware settings, the rest one has more "degress of freedom".

AcroBaseAddr
Array containing all base addresses.

AcroIntNumber
Interrupt numbers. The driver programmes the boards to use this value, it is not hardware coded. Valid values start at `(IVEC_TO_INUM(IV_USER_0)+0)`.

NUM_ACROBOARDS
The number of boards you want to define.

ACRO_ILEVEL
Interrupt level the board uses. Must correspond to the on-board jumpered value.

ACRO_BOARD_TIMEOUT
Time in system ticks the driver waits for the board to become available.

ACRO_ACQ_TIMEOUT
Time in system ticks the driver waits for an end-of-conversion interrupt.

ACRO_NUM_CHANNELS
For single ended input this must be set to 32, for differential input use 16.

UNIPOLAR_CONVERSION
If you jumper unipolar conversion (not 2th complement), you must define this.

RAW_OFFSET
This value is added to the board's value (0x8000 ... 0 ... 0x7FF0) to get positive values (0 ... 0x8000 ... 0xFFF0).

WF_COPY_...
Parameters for the task that finishes the sampling of a waveform: copying, freeing the board, processing the record.

3.3 Installation

To announce the driver to EPICS you have to add it in the related ascii-file:

```
cd $SHADOW/share/ascii
```

Now add to the end of "drvSup.ascii":

```
"drvAcro"
```

As after every change to any ascii file we have to update the EPICS-data:

```
cd $SHADOW/share/sdr
make
```

Now link the driver's sources into the EPICS source dir:

```
ln -s $HSE/share/src/drv/drvAcro.c    $SHADOW/share/src/drv
ln -s $HSE/share/epicsH/drvAcro.h $SHADOW/share/epicsH
```

To tell epics to compile the new driver we have to edit the Imakefile:

```
cd $SHADOW/Vx/share/drv
add under SRCS:
        $(SRC)/drvAcro.c\
add under OBJS:
        drvAcro.o\
add near end:
        MakeObjectFromSrc(drvAcro)
```

Now build a new Makefile from the Imakefile:

```
cd $SHADOW/Vx/$EPICS_TARGET/drv
sh build.it
```

Now you may really build the driver.

```
cd $SHADOW/Vx/$EPICS_TARGET/bin
make
```

3.4 Tests

There is a simple routine to scan one value from one channel of the AVME9325 without using interrupts. You may use this for a first test:

```
board = 0
        (short) value=0
channel = 0
        AcroScanChannelPolled (board, channel, &value)
```

While `AcroScanChannelPolled` is running the green LED on the Acromag board should be on for a very short time.

3.5 Using Interrupts

Setting jumpers J7 you may select the interrupt *level* from 1 to 7. No interrupt will occur if you leave J7 completely empty. The chosen interrupt *level* must appear in the configuration file, refer to section 3.2 ! In addition you have to give the interrupt *number* (not vector!) in the same file. The drivers sets this interrupt number on the board, it is not accessible by jumpers. To test the interrupt-controlled scanning you may use the following routine:

```
board = 0
        (short) value=0
channel = 0
        AcroScanChannelIRQControlled (board, channel &value)
```

4 Devices

This package contains two EPICS-devices: One for analog input records and one for waveform records.

4.1 Waveform record

The waveform device needs a waveform record with a RATE field. In previous EPICS-releases this field was included by default. It is necessary for specifying the time intervall between successive samples (which is the inverse of the sample frequency, of course). To include this you have to change the file
`$SHADOW/share/ascii/waveformRecord.ascii`.
Add to the end of this file:

```
"Sampl. Rate[mic.sec]" RATE NULL DBF_FLOAT NO 1 100 12  CON 5 CON 2.147e9
```

The value "12" is the prompt number and has to be the next free numbner in the ascii file. This change to the ascii-file will be compiled by the make-command in the sdr-directory, see below.

4.2 Installation

Link the sources:

```
        cd $SHADOW/share/src/dev
        ln -s $HSE/share/src/dev/devAiAcro.c .
        ln -s $HSE/share/src/dev/devWfAcro.c .
```

Change the related ascii-file:

```
        cd $SHADOW/share/ascii
```

Append the following line to devSup.ascii:

```
"ai"            VME_IO          "devAiAcro"             "AVME 9325"
"waveform"      VME_IO          "devWfAcro"             "AVME 9325"
```

As after every change to any ascii file we have to update the EPICS-data:

```
        cd $SHADOW/share/sdr
        make
```

To tell epics to compile the new device we have to edit the Imakefile:

```
cd $SHADOW/Vx/share/dev

add to Imakefile under SRCS:
        $(SRC)/devAiAcro.c\
        $(SRC)/devWfAcro.c\
add under OBJS:
        devAiAcro.o\
        devWfAcro.o\
add near end:
        MakeObjectFromSrc(devAiAcro)
        MakeObjectFromSrc(devWfAcro)
```

Now build a new Makefile from the Imakefile:

```
cd $SHADOW/Vx/$EPICS_TARGET/dev
sh build.it
```

Now you may really build the device.

```
cd $SHADOW/Vx/$EPICS_TARGET/bin
make
```

5 Usage

There are different operating modes for the AVME9325 driver:

5.1 Analog Input: one channel, one value

There is an example report file for dct you may use: `$ACRO/ioc/acro.rep`. In detail you have to consider for analog input records:

Device Type, DTYP
Set this to `AVME 9325`.

Input Specification, INP
This must contain a string like

```
#C<card number> S<signal>
```

The card number must be in the range from 0 to NUM_ACROBOARDS-1. Signal is the number of the board's channel you want to see, valid signal numbers are in 0...ACRO_NUM_CHANNELS-1. Refer to 3.2 to see the limits. In dct you are prompted for these values:

```
|----------------------|
|   Enter VME address  |
|                      |
| Card   : 0           |
| Signal : 0           |
| Exit   :             |
| Quit   :             |
|                      |
|                      |
|----------------------|
```

Linearization, LINR
Should be set to LINEAR.

Engineer Units Full/Low, EGUF/EGUL
Set this to -10 and +10 if you configured the card for -10V, 9.9976V input range.

Engineering Units, EGU
"Volts" is a good idea.
For displaying the digitized voltages there is an prepared dm-file: `$ACRO/ioc/acro.dl`.

5.2 Waveform: scanned

This type of waveform record is scanned periodically by EPICS. You should consider:

Device Type, DTYP
Similar to analog input records.

Input Specification, INP
dito.

Scan Mechanism, SCAN
Set to `1 second` or what you like, *but not Event* !

Rearm the waveform, RARM
Set to 1.

Number of Elements, NELM
Determines the size of the value-array. Use 512 for example.

Field Type of Value, FTVL
Must be `SHORT` or `USHORT`.

Sampl. Rate[mic.sec], RATE
This field specifies the time intervall in μ-seconds between samples. If you set the RATE field to 1953 μs and NELM is 512, one scan of the waveform record will take

$$512 \cdot 1953\mu\text{sec} \approx 1\text{sec}$$

If you view a 1 Hz sine wave this way, you will see one full period. *The smallest possible value is 5,* this gives a max. scan frequency of $1/(5\mu s) = 200$ kHz.
Example files: `$ACRO/ioc/wavescan.rep` for dct, `$ACRO/ioc/wave.dl` for the display manager.

5.3 Waveform: external start trigger

In this mode the record occupies the Acromag board the whole time. You cannot connect more than one record to each board, because the records programmes the board to start sampling on a falling edge on the trigger input connector (BNC on the front pannel, use TTL input). After finishing scanning the board genertes an event and the record gets scanned by EPICS, fetches data and restarts the mechanism.

Device Type, DTYP

Input Specification, INP

Number of Elements, NELM

Field Type of Value, FTVL

Sampl. Rate[mic.sec], RATE
Similar to analog input records.

Scan Mechanism, SCAN
Set to Event.

Event Number, EVNT
Choose a unique event number, for example 100.

Rearm the waveform, RARM
Set to 1.
Example files: `$ACRO/ioc/wavestart.rep` for dct, `$ACRO/ioc/wave.dl` for the display manager.
In the fastes mode the board uses an intervall of $5\mu s$. With 512 elements in the record each full conversion takes approx. 2.5 millisecs. So the maximal trigger frequency should be 390 Hz. You may, however, use a higher trigger frequency, because triggers are ignored on conversion. The display managers updates approx. twice per second.

5.4 Waveform: external stop trigger

This record is also driven by the external trigger, but the board is *always* sampling data. On receiving an external trigger the board continues to take NELM/2 samples. So the record contains NELM/2 pre- and NELM/2 post-trigger samples.

Device Type, DTYP

Input Specification, INP

Number of Elements, NELM

Field Type of Value, FTVL

Sampl. Rate[mic.sec], RATE

Scan Mechanism, SCAN

Event Number, EVNT
Similar to last section.

Rearm the waveform, RARM
Set to 0.
Example files: `$ACRO/ioc/wavestop.rep` for dct, `$ACRO/ioc/wave.dl` for the display manager.
Caveat: If your trigger is too fast, i.e. the board was not able to sample NELM/2 points before receiving the trigger, you get invalid data at the start of the record's value array. Only the the second half of the array, which was sampled after receiving the trigger, is always valid.