devIcv150.c File Reference

ADAS ICV150 device support for EPICS R3.14. More...

Functions

void	**icv150CfgAdc** (int card, int resolution)
void	**icv150CfgScan** (int card, int signal)
void	**icv150OverSampling** (int card, int on)
void	**icv150CfgGain** (int card, int signal, int gain)
void	**icv150StoreGains** (int card)
void	**icv150CfgExtTrig** (int card, int event)
void	**icv150CfgAutoScan** (int card)
void	**icv150SoftTrig** (int card)

Variables

int	**devIcv150Verbose** = 0

Detailed Description

The ADAS ICV150 is a 12/14/16-bit multiplexed ADC board with 32 input signals. Only boards configured with differential inputs are supported. The number of signals be extended up to 128 using 48 signals extension boards ICV110. The number of signals can be set by calling the function **icv150CfgScan() before iocInit**. The default number of signals is 32.

ICV150 Device Support accepts up to 4 boards in a VME crate, starting from address **0x500000** with an increment of 0x1000. Each ICV150 uses an interrupt vector, starting from 0xC0 for the board 0.

The device supports 12, 14 or 16 bit ADC resolution. The resolution must be configured by calling the function **icv150CfgAdc() before iocInit**. The default resolution is 16 bit.

Since hardware revision J, 16 bit boards have an oversampling mode. This mode can be configured by calling the function **icv150OverSampling()**.

Signals may be scanned automatically (default) or on an external trigger on J3. On external trigger, signals are scanned once and an interrupt is generated at the end of conversion. A database event is generated by the interrupt service routine to allow records processing. The automatic scanning can be set by calling the function **icv150CfgAutoScan()**. The scanning on external trigger can be set by calling the function **icv150CfgExtTrig()**. A soft trigger can be generated by calling the function **icv150SoftTrig()**.

There are two different ways to control the gain of an input signal:

- using an ICV150 AO record
- by calling the configuration function **icv150CfgGain()**.

The gains can be saved in NOVRAM by calling the function **icv150StoreGains()**.

Record Support

The device supports **AI** and **WAVEFORM** record types for signal input and **AO** record type for signal gain control. The device type DTYP is **ICV150** for all record types.

A WAVEFORM record may contain the following type of data (FTVL): USHORT, LONG, ULONG, FLOAT, DOUBLE. For types USHORT, LONG and ULONG the waveform contains raw ADC data. For types FLOAT and DOUBLE the waveform contains raw ADC data if LOPR is equal to HOPR; else the ADC data are converted according to the following formula:

VAL = RAW * (HOPR - LOPR) / RAWF + LOPR

where RAW is the ADC value and RAWF is the highest ADC value.

Function Documentation

void icv150CfgAdc (int card,
int resolution
)

This IOC shell function selects the ADC resolution.

Note:
This function must be called **before iocInit**.

Parameters:
[in] *card* ICV150 card number (valid range 0 to 3)
[in] *resolution* ADC resolution (valid range 12, 14 or 16 bits)

void icv150CfgScan (int card,
int signal
)

This IOC shell function changes the number of signals scanned on the board, starting from the first signal. It supersedes the default number of signals given by the straps on ST3. The current acquisition mode is restarted.

Note:
This function must be called **before iocInit**.

Parameters:
[in] *card* ICV150 card number (valid range 0 to 3)
[in] *signal* number of signals to scan (valid range 1 to 128)

void icv150OverSampling (int card,
int on
)

This IOC shell function allows to enable/disable the oversampling mode.

Note:
Oversampling is available since hardware revision J on 16 bit ADC boards only.

Parameters:
[in] *card* ICV150 card number (valid range 0 to 3)
[in] *on* oversampling mode (valid range 0 = OFF, not 0 = ON)

void icv150CfgGain (int card,
int signal,
int gain
)

This IOC shell function changes the gain code value of an input signal. The gain code is stored in the on board RAM. To make this change permanent, it is necessary to store the gain codes in the on board NOVRAM by calling the function **icv150StoreGains()**. The current acquisition mode is restarted.

Parameters:

[in] *card* ICV150 card number (valid range 0 to 3)
[in] *signal* signal number (valid range 0 to 31)
[in] *gain* gain code (valid range 0 to 15)

void icv150StoreGains (int card)

This IOC shell function stores the gain codes in NOVRAM. The current acquisition mode is restarted.

Note:
The number of changes in NOVRAM is limited to 10000. So you should avoid calling this function after each reboot.

Parameters:

[in] *card* ICV150 card number (valid range 0 to 3)

void icv150CfgExtTrig (int card,
int event
)

This IOC shell function stops the current acquisition mode and enables an external trigger on J3. In this mode, data are scanned once on trigger and an interrupt is generated at the end of conversion. A database event is generated by the interrupt service routine to allow records processing.

Parameters:

[in] *card* ICV150 card number (valid range 0 to 3)
[in] *event* database event number (valid range 0 to 255)

void icv150CfgAutoScan (int card)

This IOC shell function stops the current acquisition mode and starts automatic scanning. In this mode, signals are permanently scanned and data are always available.

Parameters:

[in] *card* ICV150 card number (valid range 0 to 3)

void icv150SoftTrig (int card)

This IOC shell function generates a software trigger only if external trigger mode was selected by **icv150CfgExtTrig()**.

Parameters:

[in] *card* ICV150 card number (valid range 0 to 3)

References **devIcv150Verbose**.

Variable Documentation

int devIcv150Verbose = 0

This IOC shell variable allows to print debug messages. Valid range is:

- 0 no message is printed
- 1 messages at initialization are printed
- 2 initialization and I/O messages are printed

Referenced by **icv150SoftTrig()**.
