apci

What it is:

The apci kernel module is a Linux driver for PCI based products manufactured by ACCES. The driver provides register level access to the cards, and it can be used for interrupt driven I/O.

Important Files:

acces licensing.pdf: An informational document that explains a little about the licensing of ACCES software.

apci.c: Implementation file for the apci driver.

apci.h: Header file that provides an interface for user programs to use the driver. This file will need to be included in any project that makes use of the apci driver.

apci_load: A shell script that loads the module into the system and creates the entry for the device in /dev. It must be run with root privileges.

apci_unload: A shell script that unloads the module from the system and removes the entry for the device in /dev. It must be run with root privileges.

build: A shell script that compiles and builds the module. "/usr/src/linux-source-2.6.15" must be replaced with the path to your kernel source tree. This script makes use of sudo.

gpl.text: A copy of the GPL v2. This driver is licensed under the GPL v2, and other licenses are available. For more information please read acces licensing.pdf

Makefile: A makefile for building the module.

(Files below here are located in the apcilib folder.)

apcilib.h: Header file for functions that are implemented in apcilib.c. These functions provide a wrapper for the ioctls used by the driver. They are not required to use the driver, but they are intended to make it easier.

apcilib.c: Implementation of the functions declared in apcilib.h.

Makefile: A makefile for building the tester application.

tester.c: Source file that demonstrates the most basic use of the driver. It prints a listing of all supported cards that are currently attached to the driver, along with some information

about them.

How it is used:

At the time of this writing, there are no card specific samples written for the apci driver. The tester program demonstrates the basics of using the driver. More samples will be added over time.

The apcilib files provide eleven functions for use with the apci driver. These functions are wrappers for the ioctls defined in apci.h.

int apci_get_devices (int fd)

This function will return the number of devices currently attached to the driver or a negative value to indicate an error. The 'fd' parameter for all the apcilib functions is an open file descriptor for the /dev/apci file.

apci_get_device_info (int fd, unsigned long device_index, unsigned int *dev_id, unsigned long base_addresses[6]);

This function retrieves information about the device at the specified device_index. The dev_id or base_addresses parameter may be NULL. For almost all ACCES cards base_addresses[2] will be the only member of the array that is not 0. Returns 0 on success or non-zero on failure.

int apci_write8(int fd, unsigned long device_index, int bar, int offset, __u8 data)
int apci_write16(int fd, unsigned long device_index, int bar, int offset, __u16 data)
int apci_write32(int fd, unsigned long device_index, int bar, int offset, __u32 data)

These functions allow for writing 8, 16, or 32 bits to the card. The 'bar' parameter is the base address register used for the write. On almost all ACCES cards bar will be 2. This information will be available in the product manual. Returns 0 on success or non-zero on failure.

int apci_read8(int fd, unsigned long device_index, int bar, int offset, __u8 *data)
int apci_read16(int fd, unsigned long device_index, int bar, int offset, __u16 *data)
int apci_read32(int fd, unsigned long device_index, int bar, int offset, __u32 *data)

These functions allow for reading 8, 16, or 32 bits from the card. Returns 0 on success or non-zero on failure.

int apci_wait_for_irq(int fd, unsigned long device_index)

This function waits for an IRQ to occur on the card specified by device_index. The function will deadlock the calling thread until either an IRQ occurs or another thread calls apci_cancel_irq. The function returns 0 if an IRQ occurred or -ECANCELED if the wait was canceled.

int apci_cancel_irq(int fd, unsigned long device_index)

This function will cancel a wait for IRQ on the card specified by device_index. This will unlock any thread that is calling apci_wait_for_irq. The function returns 0 if successful or -EALREADY if there is no thread waiting for an IRQ.