**The Peripheral Writer**

**Introduction**

* The peripheral writer device driver consists of four data channels.
* The user can write to a channel and can read from a channel.
* The channel index (0-3) is used to indicate the current channel.
* Any read will read from the current channel.
* Any write will write to the current channel and increment the channel index.
* You can programmatically set and get the channel index.
* You can get peripheral information and store in a structure.

**Device File**

The device file for our driver is **/dev/perwr**.

**Functions**

int **open**(const char\* deviceFile, O\_RDWR);

* The device file for our driver is **/dev/perwr**.
* Returns a file descriptor.

int **write**(int fd, char\* buffer, size\_t len);

* Uses the file descriptor returned by **open**.
* Writes **len** bytes from **buffer**.
* **write** returns the actual number of bytes written.

int **read**(int fd, char\* buffer, size\_t len);

* Uses the file descriptor returned by **open**.
* Reads up to **len** bytes into the **buffer**.
* **read** returns the actual number of bytes read.

int **close**(int fd);

* Closes the file referred to by the file descriptor.
* Returns 0 on success, -1 on error.

int **ioctl**(int fd, int command, …);

* Issues a command to the driver referred to by the file descriptor.
* Return -1 on error.

**Ioctls**

**PERIPHERAL\_WRITER\_GET\_CHANNEL\_INDEX**

Gets the channel index and stores the value into an int.

ioctl(fd, PERIPHERAL\_WRITER\_GET\_CHANNEL\_INDEX, &perIndex);//perIndex is an int

**PERIPHERAL\_WRITER\_SET\_CHANNEL\_INDEX**

Sets the channel index passing the index as an int

ioctl(fd, PERIPHERAL\_WRITER\_GET\_CHANNEL\_INDEX, &perIndex); //perIndex is an int

**PERIPHERAL\_WRITER\_GET\_INFO**

Gets peripheral information and stores into a structure of the form:

typedef struct peripheral\_info {

int num\_channels;

int size\_channel;

} PERIPHERAL\_INFO;

ioctl(fd, PERIPHERAL\_WRITER\_GET\_INFO, &perInfo);//perInfo is of type PERIPHERAL\_INFO