# Source code

## *FIR filter driver example*

This example is divided into three separate modules:

- A kernel mode device driver to be filtered
- A kernel mode filter device driver
- A user mode application to test the filter driver

This example can be tested on any platform including VPC.

### Filtered device drive sending an array of values

This device driver sources are located under folder "\Sample code\1\DRIVERS\DEMODRVR"

### FIR filter driver filtering the data from filtered driver

This device driver sources are located under folder "\Sample code\1\DRIVERS\FIRFLTDRVR"

### Console application retrieving filtered data and displaying it

This application sources are located under folder "\Sample code\1\Application\FilterTest"

## *Asynchronous I/O request device driver*

This example is divided into three separate modules:

- A kernel mode device driver implementing a lengthy asynchronous write operation
- A user mode application to test the device driver's capabilities

This example can be tested on any platform including VPC.

### Device driver implementing asynchronous I/O request

This device driver sources are located under folder "\Sample code\1\DRIVERS\DEMODRVR"

Note that this is the same device driver used for the filtered device driver.

### Console application receiving I/O completion

This application sources are located under folder "\Sample code\2\Application\ DemoAsync"

## *Real time device driver triggering a control thread in an application*

This device driver example was implemented on a specific ARM based board the Variscite VAR-DVK-OM37, and using the boot select micro switch as a GPIO interrupt trigger. You can use this example on similar boards using the ti_35 based BSP because I used the BSPs GPIO device driver to set the pin direction and setup the interrupt.

This driver should be located in the BSP you will be using to test it. This device driver sources are located under folder "\Sample code\3\DRIVER\RTLLABDRVR\RTLLABDRVR"

The application allows you to create two threads so you can test performance hits using various thread priorities. This application sources are located under folder \Sample code\3\Application\RTLabApp

## *Application using ActivateDeviceEx for loading a device driver*

Basic device driver and application, to practice loading and opening device drivers that you setup not to load at boot time. This practice is helpful when developing a device driver for an efficient debugging procedure since you load and unload the device driver programmatically and therefore you can set it as a release directory module and keep rebuilding and debugging it without resetting the system. The sample code for the device driver and application are under folder "\Sample code\4"

## *How to open and manage more than one instance of device driver*

Even though embedded RTOS device drivers should normally not allow more than one instance of a device driver to be open at the time, sometimes it is required, for example a GPIO device driver that allows access to GPIO hardware from multiple clients. The example at hand allows more than one instance to be opened at the same time. However for simplicity and clarity it restricts it to two instances only. This is a simple device driver, which allows opening two instances by two separate processes at the same time. The application that tests it opens more than one instance and shows how the driver distinguishes between the two instances.

This device driver sources are located under folder "\Sample code\5\DRIVERS\ MIDEVDRVR"

This application sources are located under folder "\Sample code\5\Application\ MultipleOpen"

## *Test demo*

Test demo is built in with DemoDrvr under folder "\Sample code\1\DRIVERS\DEMODRVR\ CTK_Test"

## *DDW Community Edition*

A community edition of the device driver wizard mentioned in the text is available for installation under the folder "\Sample code\Device Driver Wizard Community Edition"