EXAMPLE NAME: FATFS_EXAMPLE_XMC47

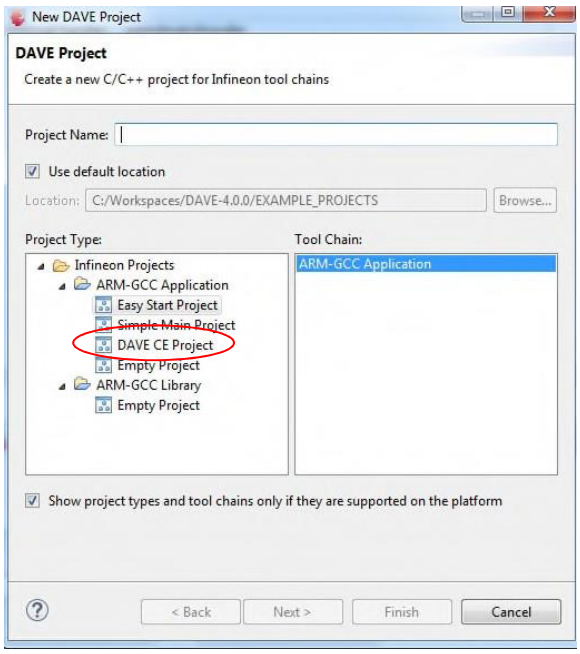
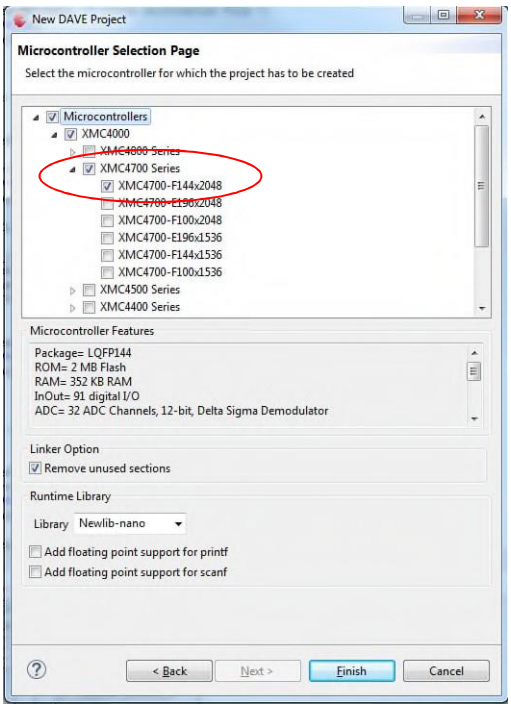
OVERVIEW:
This example demonstrates how to use the FATFS APP for FAT file system support (SDMMC cards in SD Host Controller mode) on XMC microcontroller.
The example shows how to mount/unmount a FAT file system, read/write/copy files, get file size information and how to create directories. Using virtual COM UART interface, output of the read file/get file size operation can be inspected by attaching a serial console (e.g.PuTTY).

DESCRIPTION:
- FATFS APP is used to implement FAT file system support for SDMMC cards using FatFs library.
- UART APP is used to visualize read file/get file size information functions via a serial console.

REQUIRMENTS:
Boards Required: XMC4700 Relax Kit (Order Nr: KIT_XMC47_RELAX_V1)
Micro SD card formatted with FAT file system
Serial console, e.g. PuTTY

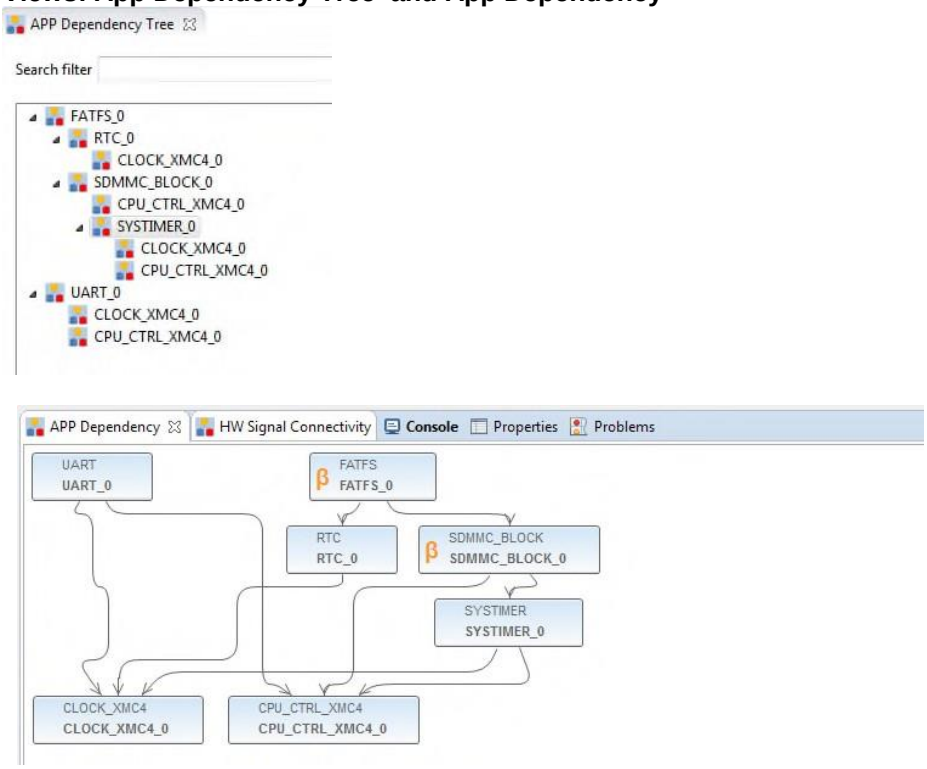
HOW TO CREATE THE PROJECT:

1. Open the DAVE CE and use  "Add IDE New Project Wizard" on the toolbar to add a new DAVE Project.

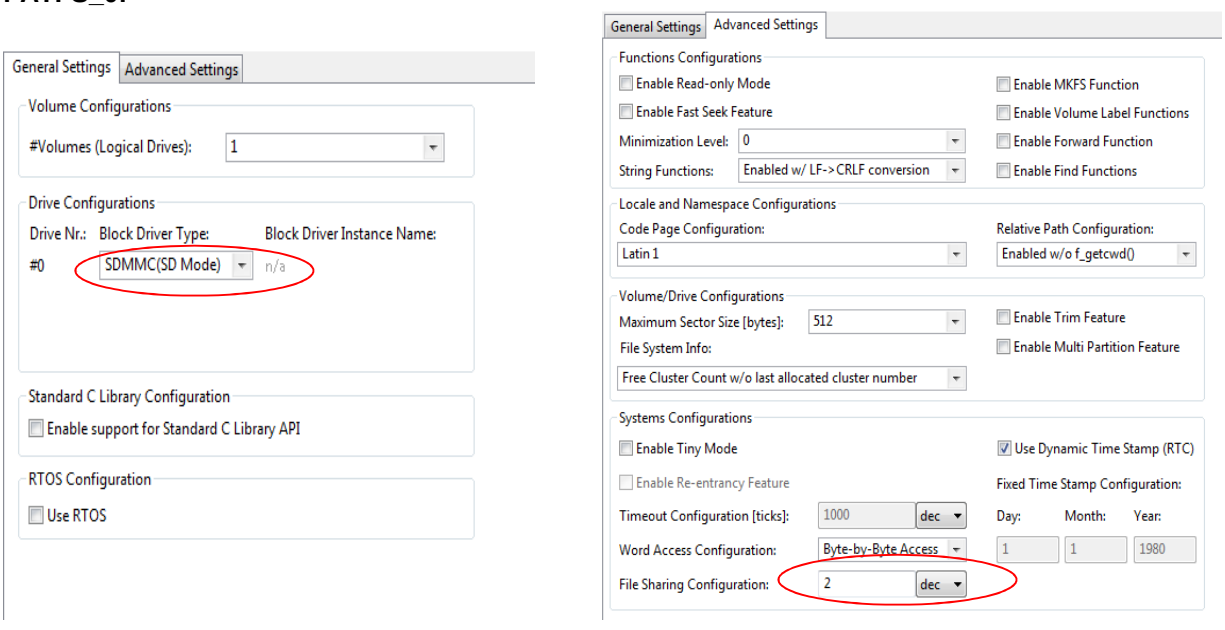
Enter Project Name Select "Project Type ➔ DAVE CE Project"	Select Processors Info ➔XMC4700 Series ➔ XMC4700-F144X2048
	

2. Use the  "Add New APP" in the toolbar to add APPs listed. Configure the App instances with the following configurations.

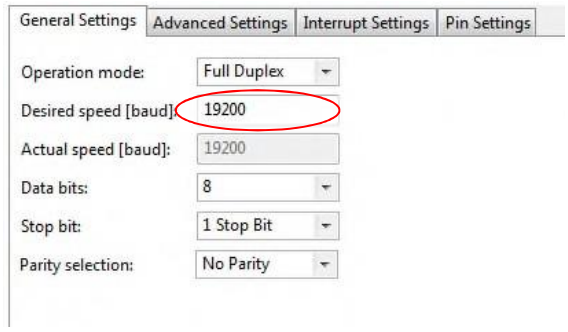
Views: App Dependency Tree and App Dependency

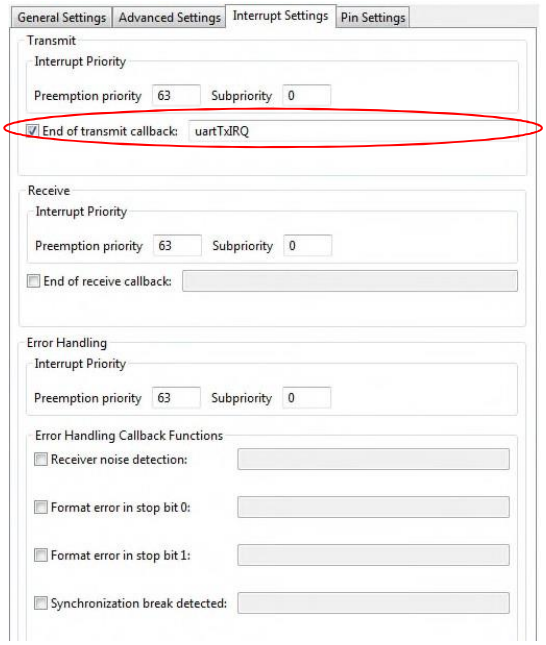


FATFS_0:



UART_0:

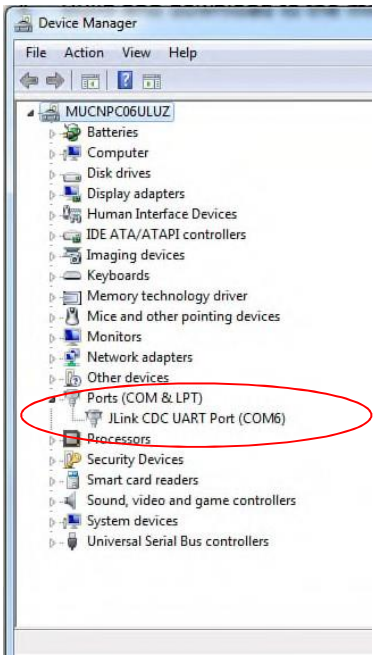




3. Generate the code for configurations made and change main.c accordingly.
4. Build and download to the microcontroller.

HOW TO TEST:

- Connect Micro USB cable to Relax Kit USB plug X101 (DEBUG port) and to a PC/laptop.
- Check inside the "Computer Management/Device manager" on your PC/laptop on which COM port the "JLink CDC UART Port" appears.

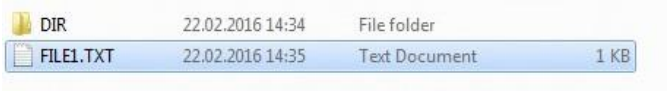


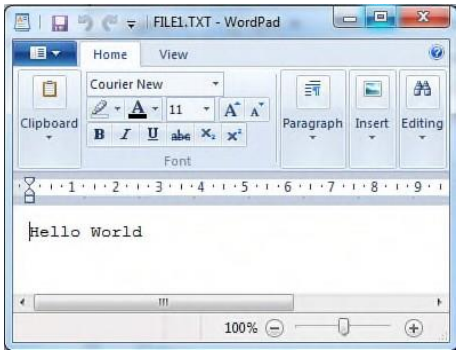
- Open a terminal (e.g. PuTTY) select the COM port observed inside previous step and configure the serial/UART connection with these parameters:
    - Speed [baud]: 19200
    - Data bits: 8
    - Stop bit: 1 Stop Bit
    - Parity selection: No Parity
- Insert SD card into Micro SD card slot of XMC4700 Relax Kit.
- Run the demo in the uC.
- Insert SD card into PC/laptop to validate file and folder structure on SD card (Please open text files with Word Pad to see content in plain text as most editors will only show content in binary/hex format).

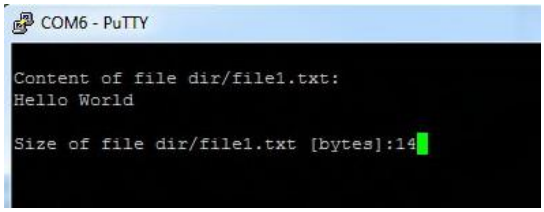
OBSERVATIONS:

1st run

- File “FILE1.TXT” is created. Content is the string “Hello World\r\n”.
- Directory “DIR” is created.
- File “FILE1.TXT” is copied into directory “DIR”.
- Content of “DIR/FILE1.TXT” is same as content of file “FILE1.TXT”.
- File size of file “DIR/FILE1.TXT” is 14 Byte.
- Content and file size information of file “DIR/FILE1.TXT” are output to console.







Subsequent runs

- File “FILE1.TXT” gets appended as it already has been created. Each time the example runs, the file gets appended by the string “Hello World\r\n”. Thus, content of “FILE1.TXT” is n-times the string “Hello World\r\n”.
- Directory “DIR” is present from previous run.
- File “FILE1.TXT” is copied into directory “DIR”. Previously present file gets overwritten.
- Content of “DIR/FILE1.TXT” is same as content of file “FILE1.TXT”.
- File size increases with each run. Overall size of file “DIR/FILE1.TXT” is n-times 14 Byte.
- Content and file size information of “DIR/FILE1.TXT” are output to console.

