MCF52223 USB Host Demonstration Suite

This package contains a set of demonstration programs for the MCF52223.

These demonstrations are provided as complete Metrowerks Code Warrior projects with full source code.

The code is designed to show how to use various aspects of USB Host mode in conjunction with the MCF52223.

Basic Directory structure

The directory tree groups files by functionality and by dependency. Subfolders named after a developers environment contain files specific to the particular environment. Subfolders named after an MCU hold files specific to that MCU. Folders named after a project hold files specific to that project. For example the folder "/usb-peripheral/project/CodeWarrior/mcf52223/cdc" contains files specific to usb peripheral functionality, CodeWarrior developers environment, MCF52223 MCU and the cdc demo project.

/usb-common
Contains files shared by multiple projects independent of host or device functionality. Still these files may be specific to a micro controller. Such files are placed to a subfolder named after the specific micro controller.

/usb-host
Contains usb-host specific demo projects and source code.

/usb-host/projects
Contains developer environment specific files.

/usb-host/projects/CodeWarrior/mcf52223
CodeWarrior specific files for the mcf52223 MCU. Each demo has its project file in a subdirectory where the subdirectory name equals to the name of the demo.

/usb-host/projects/gcc-gmake/mcf52223
GNUC and GNU make specific files. This folder contains GNU toolset specific files. These projects have been prepared under Linux. Each demo has its project file in a subdirectory where the subdirectory name equals to the name of the demo.

/usb-host/src
Contains the USB source code that is environment and target independent.

/usb-host/src/mcf5222x
Contains source code that is target dependent. Each demo has its source files in a subdirectory where the subdirectory name equals to the name of the demo.

Source Code

The package contains the following files.

/usb-common/mcf5222x

hcc_types.h	Common type definitions.
mcf5222x_reg.h	Register definitions for the MCF5222x MCUs.
target.c	Hardware (board) specific routines. Mainly related to
target.h	initialization.

/usb-common/mcf5222x/timer-drv

timer.c	Simple mS resolution timer implementation.
timer.h	

/usb-common/mcf5222x/uart-drv

uart.c	Polled UART driver.
uart.h	

/usb-common/terminal

hcc_terminal.c	Terminal server implementation. Communication channel
hcc_terminal.h	and implemented command set can be configured runtime.

/usb-common/utils

utils.c	Small utilities. Used only is standard C library is not
utils.h	available.

/usb-host/src/mcf5222x/

host_ints.c	Interrupt handlers for host demos.

/usb-host/src/mcf5222x/usb-drv

usb_host.c	USB Host driver source.
usb_host.h	
usb_utils.c	USB Host driver utility functions.
usb_utils.h	

/usb-host/src/mcf5222x/hid

hid_demo.c	Demo implementation. Prints information about connected devices to serial line. (Gateways information between the HID drivers and the serial line.)
hid_parser.c	Read and write access to HID report items.
hid_parser.h	
hid_usage.h	HID usage codes.
host_hid.c	Basic HID functionality (class specific requests, etc.) used
host_hid.h	by the HID drivers.

host_hid_joy.c	HID Joystick driver.
host_hid_joy.h	
host_hid_kbd.c	HID Keyboard driver.
host_hid_kbd.h	
host_hid_mouse.c	HID Mouse driver.
host_hid_mouse.h	

/usb-host/src/mcf5222x/mass-storage

| mst_main.c | Main control code of mass storage demo. |

/usb-host/src/mcf5222x/mass-storage/mst-drv

scsi.c	SCSI layer for USB Mass Storage Class host driver
scsi.h	
usb-mst.c	USB layer of USB Mass Storage Class host driver
usb_mst.h	

/usb-host/src/mcf5222x/mass-storage/thin-lib

mst_glue.h	Links sector driver of FAT file system to SCSI layer of mass storage driver.
thin_usr.h	Header file to be included with FAT file system
thin_lib.a	Limited function FAT file system library
fat_sthin.h	
thin-lib.a	File system library for CodeWarriro and GCC.
thin-lib-gcc.a	

MCF52223 USB Host Projects

HID Project

If the HID project is downloaded to the MCF52223 Development board and a terminal is attached to the first serial port (115200,N,8,1) then the following can be demonstrated :

1. Plug a mouse to the USB port and the all mouse operations will cause the mouse co-ordinates and the status of the three buttons to be displayed on the terminal.
2. Plug a keyboard to the USB port and all key presses will be displayed including the modifier key status and the scan code of the pressed key.
3. Plug a joystick to the USB co-ordinates x,y, rz, the status of the six buttons, the value representing the slider position and the current angle reflecting the hat switch position.

Mass Storage Project

Attach a terminal to the first serial port (115200,N,8,1) and download this project. When a USB drive is connected to the board then the following commands can be entered at the terminal to view the drive:

dir   Prints a directory listing
dump   Dumps the contents of a file in ASCII and Hex form
help   Prints help information
type   Prints the contents of a file.

The user may develop their application by using the CMX-FFS-THIN library to access the FAT file system. They can also extend the above command list.

The following file API commands are available:

- f_open()
- f_close()
- f_read()
- f_write()
- f_tell()
- f_getc()
- f_putc()
- f_rewind()
- f_eof()
- f_seek()
- f_filelength()
- f_getfreespace()
- f_findfirst()
- f_findnext()

Please consult the CMX-FFS-THIN user manual for full documentation of these functions.

The FAT file system library supplied is a limited functionality, evaluation version. Developers requiring a full version should order with CMX-FFS-FAT or CMX-FFS-THIN to get the full functionality and performance for the system.