CMX-USB Device Demo for Freescale MCF5222X

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

These programs are tested on the standard MCF52223 development board in conjunction with a Windows XP system.

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 Device mode in conjunction with the MCF52223.

Basic Directory structure

/hid_led_demo

Contains a Visual Studio 7 Project for accessing an embedded device through the HID class over USB. Using the application the LEDs on the board can be controlled, and status of the switches read. Note: to be able to recompile the project some header files and libraries are needed from the Windows XP Driver Development Kit available from Microsoft.

/projects

Contains developer environment specific files for the two MCF52223 demo projects. One project demonstrates the HID class and the other demonstrates the CDC class.

/src

Contains the USB source code that is project independent.

Source Code

The source code is contained in the following files:

/mcf5222x MCF5222x Directory

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

/mcf5222x/usb-drv USB Basic Device Controller

usb.c usb.h	USB driver source.
usb_config.h	USB driver compile time configuration parameters. This includes another file from the current project based on conditional compiling directives.

/mcf5222x/uart-drv MCF52223 UART Driver

uart.c uart.h	Simple uart driver source code

/mcf5222x/hid-demo HID Class Demo

hid.c hid.h	HID device layer for the USB driver.
hid_generic.c hid_generic.h	Generic HID demo. Allows control of on board leds and read status of on board switches.
hid_kbd.c hid_kbd.h	HID keyboard demo.
hid_mouse.c hid_mouse.h	HID mouse demo.
hid_usb_config.c hid_usb_config.h	USB configuration for the three HID demos.
ints.c	Interrupt and exception handlers.
main.c	Main entry point. Will select which demo to execute.

/mcf5222x/cdc-demo CDC Class Demo

cdc_usb_config.c cdc_usb_config.h	USB configuration for the CDC demo.
ints.c	Interrupt handlers.

main.c	Main loop. Contains CDC to UART bridge functionality.
usb_cdc.c usb_cdc.h	CDC layer for the USB driver. Provides UART like API for an application.

MCF52223 Demonstration Projects

HID Class Project

The HID class project is effectively three demonstrations – depending on which switches are pressed at startup the demonstration is selected – you must restart the device to make a new selection:

HID Keyboard (no switches pressed at startup)

This simple demonstration enables the device to behave as a keyboard – if switch 1 is pressed a page up is sent to the host and if switch 2 is pressed a page down is sent. It is a simple matter to modify this loop to take a different input source and generate different key values. The LEDs on the board work as lock key indicators (Num Lock, Caps Lock, etc…)

Note: you can run this while your standard keyboard is connected to your PC – this will effectively just provide an extra key source to the host.

HID Mouse (switch 1 pressed at startup)

This demonstration turns the device in to a standard mouse. The demo will move the mouse cursor left to right and back while connected to the PC.

Note: you can run this while your standard mouse is connected to your PC – this will effectively just provide an extra mouse movement source to the host.

HID Generic Device (switch 2 pressed at startup)

This demonstration is for use with the VC++ project described below. The device configures itself as a vendor specific HID device and the VC++ application is able to read data from and write data to the device.

The code has two reports – an input report and an output report.

If the device receives an output report from the host application then it sets the development port LEDs to the value sent.

If the device receives an input report then it reads the settings of switches one and two and returns the combined value to the host.

Note: All operations are initiated by the host software – the VC++ application.

CDC Class Project

This demo will turn the device into a USB to UART bridge. On the PC a virtual COM port will be created using the Communication Device Class. All data sent to this port will be gateway to UART0 of the board. All data received on UART0 will be sent to the virtual serial port on the PC.

Installation (Windows XP)

Build the CDC demo application project and download to the mcf52223. When the application is started the PC shall recognize the new device.

When the device is first recognized by the PC, Windows will ask for a driver. Please specify the folder that contains mcf5222x.inf, and windows will install the necessary driver files. After the installation is done, a new serial port can be seen in the device manager. This is the place were COM port properties can be set. The properties set to the virtual COM port on the PC will be applied to UART0 of the board. Note: hardware handshaking is not supported.

Using the CDC demo

After the installation is done, start HyperTerminal or any other terminal client application. Configure it to use the virtual COM port assigned to the mcf52223 (e.g. COM4). You can change the assigned com port in the device manager. Under "Ports (COM & LTP)" you will find the device. Right clicking it and selecting properties you can set which COM port it shall use. Disable handshake mechanisms and local echo in the terminal client. Select line coding properties you like.

Connect the serial port of your PC to UART0 of the board. Start a second HyperTerminal and connect it to the COM port of the PC. Set the same line coding properties as for the virtual COM port was set.

Characters typed in one HyperTerminal will show up in the other and vice versa.

Microsoft VC++ Version 7.0 Generic HID Class Project

To be able to build this project you need to have the Microsoft DDK installed.
The following files are required from the DDK.

hid.lib
hidclass.lib
hidparse.lib
hidpi.h
hidsdi.h
hidusage.h

Once the Generic HID demonstration is running on the target and it is connected to the host PC you can use the GUI to control the device.

The GUI of the application has four buttons to control the four LEDs on the board, and two check boxes to reflect the state of SW1 and SW2. Any status change will take effect immediately.

Note: if the device is not connected to the PC when the PC side demo is started it will give an error sound and exit. Nothing will be seen on the screen.