On-The-Go driver for MCF5222x and MAX3353e

This project includes a complete host, a complete device and the OTG functionality.

The way the device part works is documented in “USB Device Demo_mcf5253.doc”
The way the host part works is documented in “USB Device Demo_mcf52223.doc”

This document describes how to use the complete dual role demonstration – with reference to the above two documents.

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-otg/projects
Contains developer environment and MCU specific files.

/usb-otg/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-otg/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-otg/src
Contains the USB source code that is developer environment independent.

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

Source Code

The source code is contained in the following files:

/ usb-common/mcf5222x

hcc_types.h	Common type definitions.
target.c	Hardware (board) specific routines. Mainly related to
target.h	initialization.

/usb-common/mcf5222x/uart-drv

uart.c	Simple SCI driver.
uart.h	

/usb-common/terminal

terminal.c	Simple terminal implementation. It is able to execute
terminal.h	commands specified during initialization.

/usb-common/terminal/utils

utils.c	Basic utilities mainly related to string conversion.
utilsl.h	

/usb-otg/src/mcf52222x

hid_demo_app.c	Hid demo app for on-the-go.
otg_main.c	On-the-go demo application.

/usb-otg/src/mcf52222x/i2c-drv

i2c.c	Basic i2c driver for MCF5222x
i2c.h	

/usb-otg/src/mcf52222x/otg-drv

max3353.c	MAX3353 driver
mcf52223.c	OTG driver for MCF52223
otg-low.h	
otg.c	On The Go driver.
otg.h	

How to Use This Demonstration Project

This demonstration is designed to work with two M52223EVB boards connected to each and setup for OTG usage. To run a particular demonstration application, refer also to the Host and Device demonstration documents.

The demo will turn the MCF5222x into a dual role USB device. The device can work as an USB host or as an USB peripheral.

When working as a host, the usb-host-hid-demo application will be executed.

When working as a peripheral, the usb-device-hid-demo will be executed. After one of these demo applications has been started, the device will work as described in the documentation of the specific demo.
The OTG demo will use UART0 of the board as a control communication channel. You can issue some commands and can read status information over this channel.

Starting the demonstration:

Load the project file otg.mcp into Code Warrior and the compile the demo. Use your flash programmer to download the application to both of your boards. If you use Code Warrior and USB ColdFile Multilink, please do not start the application on the first device after downloading. This way the application can be started out-of reset after the debugger is disconnected.

Connect UART0 of both boards to COM port of your PC. Please configure both COM ports to use 38400 bps, 8 data bits, 1 stop bit and no parity. Start hyper terminal on both COM ports.

After both boards are programmed, connect using an on-the-go USB cable, and start the demo applications. Both boards will start up in a mode selected by the USB cable. One of the boards will execute the host demo and the other the device demo automatically.

Both boards will print some information on its serial line, and execute a simple terminal program.

The terminal will accept three commands:

help
: Will print all supported commands with a brief description.

host
: This will try to switch to host mode.

peripheral
: This will try to switch to peripheral mode.

To get the boards communicate properly, you use the above commands to switch to the right mode on each of them (one should run in host and the other one in peripheral mode).