OpenSDA on TWR-KW2x

User's Guide

2013
Rev 1.1

Table of Contents

1 Overview

OpenSDA is an open-standard serial and debug adapter. It bridges serial and debug communications between a USB host and an embedded target processor as shown in Figure 1. The hardware circuit is based on a Freescale Kinetis K20 family microcontroller (MCU) with 128 KB of embedded flash and an integrated USB controller.

The host interface supports two application modes:

- Mass Storage Device (MSD): Allows to easily program applications into the flash of the target processor by dragging the new binary onto the drive

- Debug Application: Provides an interface to Integrated Development Environment (IDE) software, allowing the debug of an application directly from the OpenSDA interface.

Both application modes support the USB Communications Device Class (USB CDC Interface) that bridges serial communications between the USB Host and a UART serial interface on the target processor.

In order to switch between these application modes, the OpenSDA interface processor must be started in bootloader mode and the corresponding OpenSDA application firmware has to be dragged and dropped into the drive.

Note: Previous OpenSDA firmware didn't incorporate both MSD and Debug applications into the same firmware. Current version supports both functionalities in the same firmware.

This document provides information required to operate OpenSDA. Information on how to integrate OpenSDA into a custom hardware design and how to create custom OpenSDA applications can be found in the *OpenSDA Developer's Guide* (available October 2012).
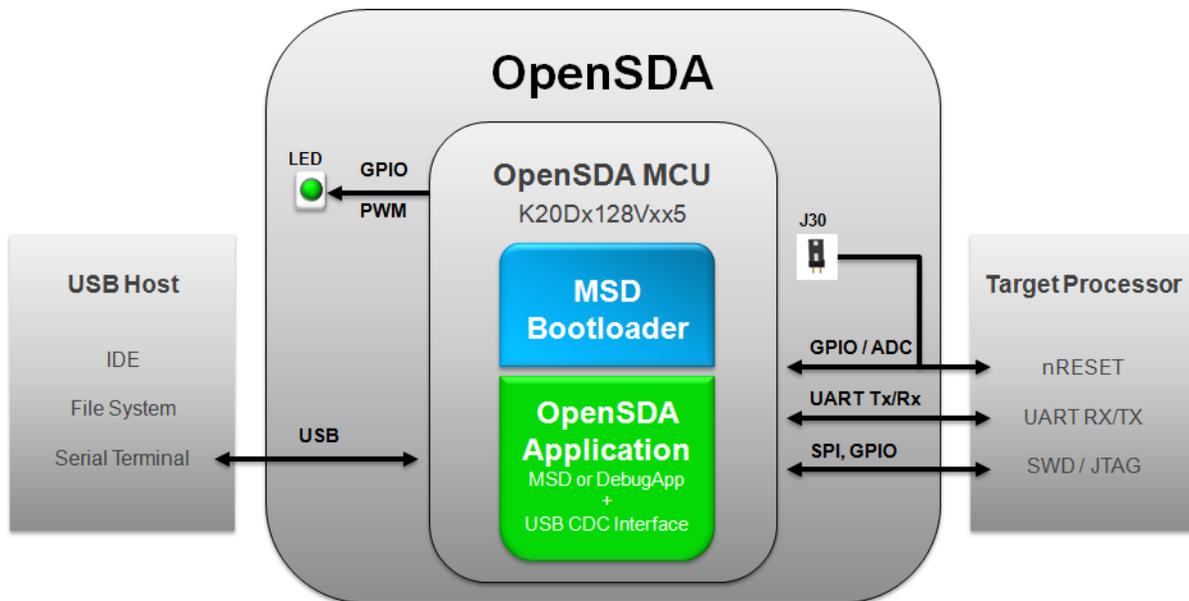

Figure 1. OpenSDA High-Level Block Diagram

1.1 OpenSDA Hardware

OpenSDA is managed by a Kinetis K20 MCU built on the ARM® Cortex™-M4 core. The Kinetis K20 includes an integrated USB controller that can operate at clock rates up to 50MHz. The OpenSDA circuit includes, at a minimum, a status LED and a pushbutton. The pushbutton asserts the Reset signal to the target processor, which could be a microcontroller, digital signal controller, or a microprocessor. SPI and GPIO signals provide an interface to either a JTAG or an SWD debug port on a target processor. Additionally, signal connections are available to implement a UART serial channel. The OpenSDA circuit receives power when plugged into a USB Host. In some designs, the OpenSDA circuit can also control the power supply of the target processor.

1.2 OpenSDA Software

OpenSDA software includes a flash-resident USB MSD Bootloader and a collection of OpenSDA Applications. The MSD Bootloader is consistent across all implementations of OpenSDA, but the available OpenSDA Applications may vary from one hardware system to another. The following sections describe the MSD Bootloader and the standard OpenSDA Applications provided by Freescale: the P&E Debug Application and the MSD Flash Programmer.

1.2.1 MSD Bootloader

The OpenSDA MSD Bootloader provides a simple interface for loading an OpenSDA Application into the OpenSDA processor's non-volatile memory. Only one OpenSDA Application may be resident at a time, and this application will be automatically run at startup unless Bootloader mode is selected by putting a jumper on J30 pins while plugging in a USB cable.

The MSD Bootloader appears in the host file system as a removable drive with a volume label of "BOOTLOADER", as Figure 2 shows. An OpenSDA Application is loaded by simply dragging and dropping an OpenSDA Application file onto the BOOTLOADER drive, which implements a pseudo-FAT16 file system. OpenSDA Applications can be in S-record, raw binary file, or an encrypted .SDA file formats. Status information is provided through a text file named LASTSTAT.TXT.
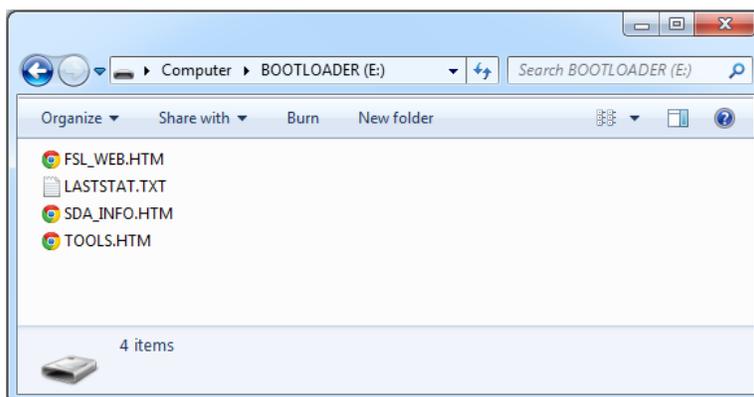

Figure 2. MSD Bootloader file list shown in Windows Explorer on Windows 7

Figure 2 shows the files listed in a typical OpenSDA Bootloader. All files are readable as text files. The *.HTM files execute links to live web pages and, if executed, will open the default browser and load the

specified web page. Table 1 summarizes the files in the MSD Bootloader and MSD Flash Programmer removable drives.

Table 1. OpenSDA Mass Storage Device File List

Filename	Description
FSL_WEB.HTM	Link to the primary hardware development tool's web site on freescale.com
LASTSTAT.TXT	The text in this file reflects the latest status information from the previous operation. If no operation has been performed, the contents of the file will be "Ready.". Following a successful application update operation, the contents will read "Completed.".
SDA_INFO.HTM	Link to the OpenSDA web page that contains the latest **documentation**, **software drivers**, and **OpenSDA Applications** available from P&E Microcomputer Systems Embedded in this file are id-value pairs that provide detailed information about the hardware: | ID Tag	Description
BOARD	Board name (e.g. TWR-MKW24D)
BOOTVER	Bootloader version
APPNAME	Installed application
APPVER	Application version | These values will be available on the web page and they can also be read directly from the SDA_INFO.HTM file contents.
TOOLS.HTM	Link to additional software and hardware tools available for Freescale products.
SERCD89.INF SERCD89.CAT	Windows driver files for the USB CDC serial communications port built into the MSD Flash Programmer OpenSDA Application. These files are available in the MSD Flash Programmer drive, but not in the BOOTLOADER drive.

Note: Some operating systems cache the files on the BOOTLOADER drive. Because of this, files that were copied to the OpenSDA Bootloader drive may be seen in the file system even though they are not stored to the non-volatile flash memory of the OpenSDA controller. These files will no longer appear once power has been cycled to the OpenSDA device.

1.2.2 P&E Debug Application

The P&E Debug Application is an OpenSDA Application that provides debugging and a virtual serial port all in one application. It provides a run-control debug interface that controls the JTAG or SWD debug interface to the target processor. It also provides a USB communications device class (CDC) interface that bridges serial communications between the USB Host and a UART serial interface on the target processor. Several integrated development environments support the P&E Debug Application including CodeWarrior for Microcontrollers (v10.3 and later), IAR Embedded Workbench (v6.40.3 and later), Keil MDK (v4.54 w/ P&E patch), as well as software from P&E Microcomputer Systems.

USB Drivers for all P&E Microcomputer Systems debug tools can be found online at www.pemicro.com/opensda.

The P&E Debug Application is designed to debug the resident target processor in the OpenSDA system with limited support for off-board devices within the same processor family as the resident target processor.

1.2.3 MSD Flash Programmer

The MSD Flash Programmer offers a very easy and convenient way to program applications into the flash of the target processor. Like the MSD Bootloader, the MSD Flash Programming Application emulates a FAT16 file system and enumerates as a mass-storage device. It appears as a removable drive in the host operating system with a volume label that matches the board name (for example, TWRKW2xDxxx). Raw binary and Motorola S-Record files that are copied to the drive are programmed directly into the target memory device. This is most often the flash of the target microcontroller, but it could also be another memory device, such as a serial flash, depending on the specific implementation.
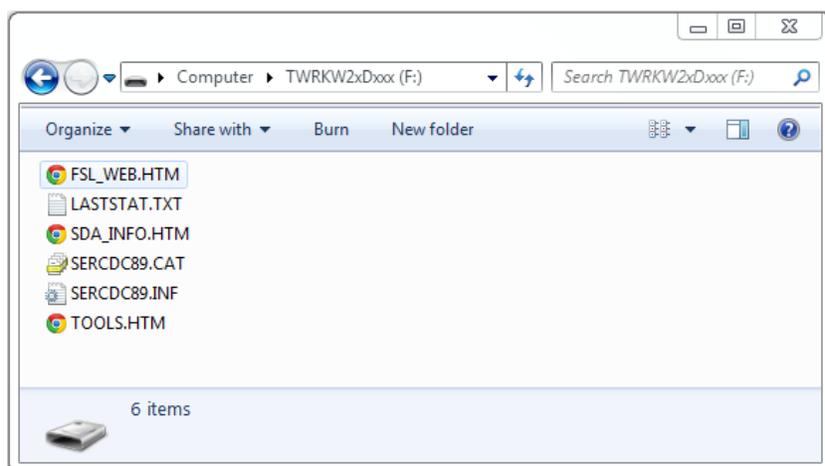

Figure 3. MSD Flash Programmer file list shown in Windows Explorer on Windows 7

The MSD Flash Programmer is designed to program a specific target configuration. It does not support verification or configuration and is not recommended as a production programmer. The Cyclone products from P&E Microcomputer Systems are recommended for production programming. They provide production programming solutions with many different features and verification options that are highly configurable: http://www.pemicro.com/opensda/tools.

The MSD Flash Programmer also provides a USB communications device class (CDC) interface that bridges serial communications between the USB Host and a UART serial interface on the target processor. The drivers required to install this USB CDC device on a Windows operating system are embedded within the pseudo file system of the MSD Flash Programmer itself. USB drivers are available at www.pemicro.com/opensda that can be preinstalled; however, it is not necessary to do so to run the MSD Flash Programmer. If Windows fails to find the CDC drivers on its own, the OS can be pointed to the drivers on the MSD Flash Programmer's removable drive.

Board and flash programming operation status are also available in text files on the removable drive. The same files that are accessible in the MSD Bootloader are also available in the MSD Flash Programmer's removable drive. Figure 3 shows the files available on TWR-KW2x hardware. Refer to Table 1 for a description of each file.

Note: The MSD Flash Programmer currently only operates correctly on Windows operating systems. However, the USB CDC virtual serial port functions properly in Windows, Linux and Mac operating systems.

1.2.4 USB Communications Device Class (CDC)

Several of the default OpenSDA Applications provided by Freescale, including the MSD Flash Programmer and the P&E Debug Application, provide a USB Communications Device Class (CDC) interface that bridges serial communications between the USB Host and a UART serial interface on the target processor.

With a Windows host operating system, the USB CDC interface will automatically be assigned a COM number, such as COM10 or COM11. To understand which COM number has been assigned to the unit, refer to the Windows Device Manager (Start->Run->"mmc devmgmt.msc") and look in the "Ports" section. Many free terminal applications, such as HyperTerminal, Tera Term and PuTTY, are available to send and receive serial traffic in Windows.

When the USB CDC interface enumerates on the Linux host operating system, it is generally assigned a tty* hardware identifier. Many free terminal applications, such as minicom, are available to send and receive serial traffic in Linux. Make sure you have the appropriate permissions in the operating system to access this port. Drivers, which can be downloaded from http://www.pemicro.com/opensda, help configure these permissions.

The OpenSDA Applications do not drive the serial transmit pin to the target microcontroller until the CDC serial port is opened by an application on the host operating system. After that, these pins will be driven and will remain driven even after the USB CDC serial port is closed.

Note: There is a known Windows issue that affects most terminal applications. The virtual COM port will be removed from the device manager when the USB cable is unplugged even if a terminal application has the virtual COM port opened. The virtual COM port will not be accessible even after the virtual serial device is reconnected unless the virtual COM port is released by the application prior to the reconnection, which may require closing the application first. Close the terminal application before unplugging the OpenSDA USB cable.

2 Getting Started with OpenSDA

Follow these simple steps to interact with embedded OpenSDA.

2.1 Getting into Bootloader Mode

The Bootloader runs automatically when a powered USB cable is attached to the OpenSDA-compliant device and immediately executes the installed OpenSDA Application unless forced into Bootloader mode by the user. To force the device into Bootloader mode, use the following procedure:

1. Unplug the USB cable (if attached).
2. Insert a jumper in J30 (1-2 pins)
3. Plug in a USB cable from a USB Host to the OpenSDA USB port.
4. A removable drive will be visible in the host file system with a volume label of BOOTLOADER.

For more information, double-click the FSL_WEB.HTM link to open the tools pages for the development system in use and explore the other files embedded in the hardware. See Table 1 for a description of these files.

2.2 Loading an OpenSDA Application

1. Locate the desired OpenSDA Application. Freescale standard OpenSDA Applications may be downloaded from the web page links provided in the MSD Bootloader or MSD Flash Programmer removable drives (SDA_INFO.HTM).
2. Enter Bootloader mode by following the instructions in Section 2.1.
3. Drag/drop or copy/paste the desired application to the BOOTLOADER drive.
4. Unplug the USB cable from the host
5. Remove jumper from J30
6. Plug in the USB cable to the host
5. The OpenSDA application should now be running

2.3 Using the MSD Flash Programmer

1. Load the appropriate version of the MSD Flash Programmer application into the OpenSDA board as described in Section 2.2. There are many versions of the MSD Flash Programming application and it is important to choose the version which is specific to the hardware being programmed.
2. Copy an S-record (commonly a file with .s19 or .srec extension) or a raw/flat binary file (commonly a file with a .bin extension) to the MSD removable drive with a volume label of the target hardware (for example, TWRKW2xDxxx).
3. If programming is successful, the embedded application executes automatically. Regardless, the contents of the LASTSTAT.TXT file are updated with the latest status information.

3 LED Status Indicator

The OpenSDA LED indicator is used by the MSD Bootloader and the standard OpenSDA Applications to provide mode and status information. A description of the LED indicator states and patterns is provided in Table 2.

Table 2. LED Status and Mode information

OpenSDA Mode	State Description	LED Pattern
Bootloader	Prior to USB enumeration	Off
Bootloader	Idle – running normally with no error conditions	Blinking: 500ms on, 500ms off
Bootloader	Error	2 sec off followed by rapid on/off blinks
Application	Prior to USB enumeration	Off
Application	Running normally with no error conditions and no USB activity	On
Application	USB activity (i.e. MSD or CDC activity)	Blinking
Application	Error	2 seconds off followed by 8 rapid on/off blinks

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: http://www.reg.net/v2/webservices/Freescale/Docs/TermsandConditions.htm

Freescale, the Freescale logo, Altivec, C-5, CodeTest, CodeWarrior, ColdFire, C_Ware, Energy Efficient Solutions logo, Kinetis, mobileGT, PowerQUICC, Processor Expert, QorIQ, Qorriva, StarCore, Symphony, and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Airfast, BeeKit, BeeStack, ColdFire+, CoreNet, Flexis, MadniV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SafeAssure, SMARTMOS, TurboLink, Vybrid, and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.