

Kinetis SDPHost User's Guide



Contents

Chapter 1 Introduction.....3

Chapter 2 Overview.....4

Chapter 3 Serial Download Protocol..... 5

Chapter 4 Typical setup.....7

Chapter 5 SDPHost Command-Line usage8

Chapter 6 Usage example and SDP command description.....10

 6.1 read-register command..... 10

 6.2 Write-register command..... 11

 6.3 Write-file command..... 11

 6.4 Error-status command.....12

 6.5 Jump-address command..... 12

Chapter 7 Copyright..... 13

Chapter 1

Introduction

This document describes the usage of SDPHost, a PC host application. SDPHost provides a command line interface to send serial download protocol (SDP) commands to NXP's i.MX device enumerated on the host PC as either a USB-HID or UART device running ROM code in serial download mode. SDPHost can be a very useful tool in the factory programming/manufacturing process. It can be invoked from other applications, and as such, can be very useful tool in testing automation software, development and test setups, or manufacturing environments.

Chapter 2

Overview

This document provides brief introduction to the serial download protocol, typical factory programming setup, and usage of the tool and description of its command line interface. The document concludes by providing a set of examples in using SDPHost to carry out SDP commands with the device.

Chapter 3

Serial Download Protocol

Serial Download Protocol is a set of commands supported by NXP's i.MX devices in the Boot ROM application's serial download mode. This chapter describes briefly the commands without providing the protocol format details that are already covered in the device's reference manual document. SDPHost provides the user with a simple user-friendly command-line interface. A typical user of SDPHost application never has to learn about the internal SDP protocol format and structures in detail.

The main purpose of serial download protocol is to provide means to download bootable images from a PC to the device's internal or external RAM memory. There are a set of commands to read and write a memory/register unit, read status of the last command, download images to a given address in internal/external memory, and provide the address to jump and execute the downloaded image. Below table describes the serial download protocol format:

Table 1. SDP Command Format

BYTE Offset	Size	Name	Description
0	2	COMMAND TYPE	<p>Following commands are supported:</p> <ul style="list-style-type: none"> • 0x0101 READ_REGISTER • 0x0202 WRITE_REGISTER • 0x0404 WRITE_FILE • 0x0505 ERROR_STATUS • 0x0B0B JUMP_ADDRESS
2	4	ADDRESS	<p>Only relevant for following commands:</p> <p>READ_REGISTER, WRITE_REGISTER, WRITE_FILE and JUMP_ADDRESS.</p> <p>For READ_REGISTER and WRITE_REGISTER commands, this field is the address to a register. For WRITE_FILE and JUMP_ADDRESS commands, this field is an internal or external memory address.</p>
Table continues on the next page...			

Table 1. SDP Command Format (continued)

6	1	FORMAT	Format of access, 0x8 for 8-bit access, 0x10 for 16-bit and 0x20 for 32-bit access. Only relevant for READ_REGISTER and WRITE_REGISTER commands.
7	4	DATA COUNT	Size of data to read or write. Only relevant for WRITE_FILE, READ_REGISTER and WRITE_REGISTER commands. For WRITE_FILE command, the DATA COUNT field is in bytes.
11	4	DATA	Value to write. Only relevant for WRITE_REGISTER command.
15	1	RESERVED	Not used

Chapter 4

Typical setup

Typically, SDPHost is used in the manufacturing or factory programming process, or in the development phase of the device firmware application. SDPHost is supported on Windows® OS, Linux® OS, and Mac® OS host environments. The release package consists of all three binaries in the respective OS folders.

Test setup would typically be the device connected to PC Host via USB or UART. The SDPHost would run on the PC host, and the device would run in Boot ROM serial recovery mode.

Typically, the device or engineering development board contains set of pins or boot switches that can be used to boot the device in serial downloader mode. The device's reference manual provides the documentation on booting the device in serial downloader mode.

Chapter 5

SDPHost Command-Line usage

SDPHost provides an easy-to-use command line interface. The syntax of the usage text is:

usage:

```
sdphost [-?|--help] [-v|--version] [-p|--port <name>[,<speed>]]
[-u|--usb [[<vid>,<pid>]]] [-V|--verbose] [-d|--debug]
[-j|--json] [-t|--timeout <ms>] -- command <args...>
```

The following table describes each command-line argument:

Table 2. SDPHost command arguments

Argument	Description
-?/--help	Displays the usage syntax and description of much what is in this table
-v/--version	Displays tool version
-p/--port <name>[,<speed>]	Connect to target over UART. Specify COM port and optionally baud rate (default=COM1,115200)
-u/--usb [[<vid>,<pid>]	Connect to target over USB HID device denoted by vid/pid (default=0x15a2,0x0083). The device's reference manual provides the USB-HID device's VID and PID
-V/--verbose	Print extra detailed log information
-d/--debug	Print really detailed log information
-j/--json	Print output in JSON format to aid automation. The last -V/-d/-j takes precedence.
-t/--timeout <ms>	Set packet timeout in milliseconds (default=5000)
SDP Commands:	Following entries show the usage of Sdphost to send the SDP commands to the device
read-register <addr> [<format> [<count> [<file>]]]	Read one or more registers at the given address. Data Format must be 8(byte), 16(half-word), or 32(word); default format is 32 bits. Count is number of bytes to read; default count is size of format (i.e. one register). Output file is binary; default is hex display on stdout.

Table continues on the next page...

Table 2. SDPHost command arguments (continued)

write-register <addr> <format> <data>	Write one register at address. Data Format must be 8 (byte), 16(half-word), or 32(word). Data is data value to write.
write-file <addr> <file> [<count>]	Write file data at address. Count is the size of data to write in bytes; size of file will be used by default.
error-status	Read error status of last command.
jump-address <addr>	Jump to entry point of image with IVT at specified address.

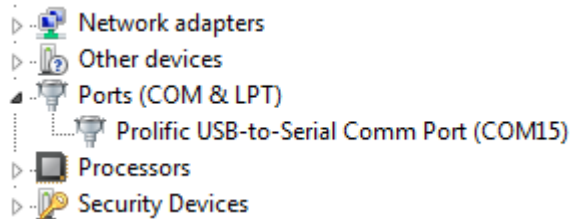
Chapter 6

Usage example and SDP command description

This section of the user guide provides example usages of sdphost application and command line arguments.

Here is a list of common arguments that can be used with the Sdp command on the command line.

- **The -p argument:** required when the device communicates with the host in SDP mode via UART. The mandatory argument that goes with -p is the COM port number for the connected device. On Windows OS, it can be discovered in the Window's Device Manager application under *Ports*.



- **The -u argument:** required when the device communicates with the host in SDP mode via USB-HID. The device's vid and pid should be passed in as the arguments for SDPHost to communicate with the correct device among the enumerated devices.
- **The -j option:** a very useful option that outputs the results in JSON and can be used in several different ways to benefit the manufacturing process.
- **The -d option:** prints extra detail that could benefit in debugging the issue. For USB-HID transfers, the option causes SDPHost to print out the Report Id of the command, along with the data being sent and received.

The SDPHost prints out the response from device with data and the response code. If there is an error or the device failed to connect, SDPHost prints out a suitable error message, including the error code returned by the device, if any.

6.1 read-register command

Read-register command is used to read the contents of device memory location or register value. The address of the register or memory location should be passed in as the first argument. Optional arguments include the data format of the register value in number of bits and number of bytes to read.

Examples of Sdphost read-register command.

- Example with device connected to COM port 15

```
C:\Users\B18551>sdphost.exe -p COM15 -- read-register 0x2efc0000 32 10
33 2a 4a 5d cf 59 ac 0a 78 ef
Status (HAB mode) = 305411090 (0x12343412) HAB enabled.
```

- Example with device connected in USB-HID mode
 - Example with verbose or standard response

```
C:\work\Tools>sdphost.exe -u 0x15a2,0x0083 -- read-register 0x2efc0000 32 10
33 8a 4e 5d 4f 19 ac 0a 78 ee
Status (HAB mode) = 305411090 (0x12343412) HAB enabled.
```

- Example with debug response

```
C:\work\Tools>sdphost.exe -d -u 0x15a2,0x0083 -- read-register 0x2efc0000 32 10
[01 01 01 2e fc 00 00 20 00 00 00 0a 00 00 00 00]
<03 12 34 34 12 cf 19 ac 0a 78 ef ed a5 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00>
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00>
<04 33 a8 4a 59 cf 19 ac 0a 78 ef ed a5 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00>
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00>
33 a8 4a 59 cf 19 ac 0a 78 ef
Status (HAB mode) = 305411090 (0x12343412) HAB enabled.
```

- Example with JSON response

```
C:\work\Tools>sdphost.exe -j -u 0x15a2,0x0083 -- read-register 0x2efc0000 32 10
{
  "command" : "read-register",
  "response" : [],
  "status" : {
    "description" : "305411090 (0x12343412) HAB enabled.",
    "value" : 305411090
  }
}
```

6.2 Write-register command

The write-register command is used to write a value to the device memory location or register address. The address of the register or memory location should be passed in as the first argument followed by data format of the register value in number of bits and the data to write.

```
C:\work\Tools>sdphost.exe -u 0x15a2,0x0083 -- write-register 0x2efc0000 32 0xAABBCCDD
Status (HAB mode) = 305411090 (0x12343412) HAB enabled.
Reponse Status = 311069202 (0x128a8a12) Write complete.

C:\work\Tools>sdphost.exe -u 0x15a2,0x0083 -- read-register 0x2efc0000 32 10
dd cc bb aa 4f 19 ac 0a 78 ee
Status (HAB mode) = 305411090 (0x12343412) HAB enabled.
```

6.3 Write-file command

The write-file command is used to write the content of a binary file to the device's memory (internal RAM or SDRAM). The command requires the address of the location where the file contents will be written. The second parameter should specify the name of the binary file to be downloaded on the device. By default, SDPHost writes the entire file length of bytes to the target memory. If fewer bytes are required, the optional third argument can be used to specify the size of data.

Typically, write-file is used to program the device with boot image and jump-address is used to start execution of boot image on the device.

```
C:\work\Tools>sdphost.exe -u 0x15a2,0x0083 -- read-register 0x2efc0000 32 10
33 08 4a 5d cf 59 ac 0a 78 ef
Status (HAB mode) = 305411090 (0x12343412) HAB enabled.

C:\work\Tools>sdphost.exe -u 0x15a2,0x0083 -- write-file 0x2efc0000 sample
Preparing to send 4 (0x4) bytes to the target.
(1/1)100% Completed!
Status (HAB mode) = 305411090 (0x12343412) HAB enabled.
Reponse Status = 2290649224 (0x88888888) Write File complete.

C:\work\Tools>sdphost.exe -u 0x15a2,0x0083 -- read-register 0x2efc0000 32 10
11 22 33 44 00 00 00 00 00 00
Status (HAB mode) = 305411090 (0x12343412) HAB enabled.
```

6.4 Error-status command

Error-status command is used to read the error code from device for the last command.

Example of error-status command in JSON format

```
C:\work\Tools>sdphost.exe -j -u 0x15a2,0x0083 -- error-status
{
  "command" : "error-status",
  "response" : [ 858993459 ],
  "status" : {
    "description" : "305411090 (0x12343412) HAB enabled.",
    "value" : 305411090
  }
}
```

6.5 Jump-address command

Jump-address is typically used after successful write-file command where the boot image or any executable image is successfully downloaded on the device's memory for execution. Jump-address command require the address of the image-vector-table (IVT). IVT can be part of the image or can be downloaded separately. It is a data structure used by rom that provides information of the boot image entry point and other parameters used for authenticating the image for secure boot. IVT is described in more detail in device's reference manual. Jump-address command will result in execution of the image once rom process the IVT and on a successful authentication of the image.

```
C:\work\Tools>sdphost.exe -u 0x15a2,0x0083 -- jump-address 0x2efc0000
Status (HAB mode) = 305411090 (0x12343412) HAB enabled.
```

Chapter 7

Copyright

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP 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 NXP 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. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

arm

