OpenMikroBootloader
Working Note

Boot-loader Protocol

Nov 23, 2015 Preliminary

1 Introduction

A Working Note is an intermediate step in the documentation process. It gathers together the content from various informal development documents, discussions, etc into a single place. One or more Working Notes form the basic for the next step, which is one or more Standard/TechNote pairs.

1.1 Served Use Cases

1.2 Unserved Use Cases

2 Specified Sections

This is the usual section organization for a Technical Note, to accumulate the Standard and Technical Note content in its eventual order.

2.1 Introduction

The specification documents the communication protocol between a boot-loader source and destination. In many cases this is between a PC and a micro-controller but is not required. An example that does not use a PC would be boot-loading a micro-controller directly from a SD card or similar.

2.2 Conventions

2.2.1 Data type conventions

Since the project is aimed at different compilers and architectures, the following type conventions will be used across this document:

Word type consists of 2 bytes

Double word (DWord) consists of 4 bytes.

2.3 Intended Use

Note that this section of the Standard is informative, not normative.

2.4 Reference and Context

2.5 Message Formats

2.5.1 Link

This message is sent to the PC by the micro-controller when it first boots to tell the PC it may request the micro-controller enter boot-loader mode.

Instruction

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment
Command	0xEF				Instructs the PC the micro-controller may be instructed to enter bootloader mode.

Reply

None

2.5.2 UnLink

This message is sent to the PC by the micro-controller when it leave boot-loader mode and starts the application

Instruction

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment
Command	0xEE				Instructs the PC the micro-controller is leaving boot-loader mode and the application is starting

Reply

None

2.5.3 Sync

This message is sent by the PC to instruct the micro-controller to enter boot-loader mode

Instruction

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment
Command	0xED				Instructs the micro-controller to enter bootloader mode.

Reply

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment

Command	0xED				The micro-controller has entered boot-loader mode

2.5.4 Request Flash Information

This message requests flash information from the micro-controller such that the source (PC, SD Card, etc) can, at a minimum, calculate addresses for the boot-loaded code image, etc.

Instruction

Field	Byte 0				
Command	0x80				

Reply

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment
Command	0x80				The Command this reply is responding to
Structure Size	Word				Size of the structure being sent, including this field
EraseBlock Size	DWord				Size of a block of flash that can be erased
WriteBlock Size	DWord				Size of a block of flash that can be written. This may not be the actual size of flash actually written but the size of a buffer that the micro-controller can hold in one write cycle
ProgramFlash Size	DWord				The amount of Flash the micro-controller contains
Bootloader Address	DWord				Memory offset where the boot-loader code begins
Bootloader Size	DWord				Size of the boot-loader code currently in flash
MCU Configuration Address	DWord				Address where the Configuration Bits start in the MCU
Microcontroller family	Byte				It can be PIC16F, PIC18F, PIC18FJ, PIC24, dsPIC30, dsPIC33FJ, dsPIC33EP, PIC32MX, PIC32MZ. The host application needs this for HEX file processing.
Revision number	Byte	Byte			Various features will depend on this field.
Application name	32 bytes long				String containing application name, which identifies the microcontroller to host application.

The Micro-controller Family field is defined as :

Family	Code
PIC16F	0x00
PIC18F	0x10
PIC18FJ	0x11
PIC24	0x20
dsPIC30	0x30
dsPIC33FJ	0x40
dsPIC33EP	0x41
PIC32MX	0x50
PIC32MZ	0x51

2.5.5 Set Address Information

This message is sent to the micro-controller to update its internal address offsets to be used in subsequent commands

Instruction

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment
Command	0x02				
Address Type	Byte				The type of Address that is being passed
Address	DWord				Address Value

Address Type	Name	Comment
0x00	WriteAddress Offset	Sets the initial offset for the next write command. Subsequent write commands must increment this value internally after each write instruction.

0x01	EraseAddress Offset	Sets the initial offset for the next erase command. Subsequent erase commands must increment this value internally after each erase instruction.

Reply

None

2.5.6 Reset

This message instructs the micro-controller to reboot itself.

Instruction

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment
Command	0x81				Instructs the micro-controller to break the link with the PC and reset itself

Reply

None

2.5.7 Erase Blocks

This message instructs the micro-controller to erase the passed parameter number of blocks. A block size is assumed to be the “EraseBlock Size” retrieved in the Request Flash Information message and starting from the EraseAddress Offset sent in the Set Address Information message or from the current location of the offset stored within the micro-controller, whichever occurred last.

Instruction

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment
Command	0x10				Instructs the micro-controller to erase a number of blocks starting at the current erase address.
Number of Blocks	DWord				The number of EraseBlock Size blocks to erase

Reply

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment
Command	0x10				The erase instruction has completed

2.5.8 Write Block

This message instructs the micro-controller to write the passed full block of data to the flash. A block size is assumed to be the “WriteBlock Size” retrieved in the Request Flash Information message and starting from the WriteAddress Offset sent in the Set Address Information message or from the current location of the offset stored within the micro-controller, which ever occurred last.

Instruction

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment
Command	0x30				Instructs the micro-controller to write the passed data to the flash.
Data	array[0.. WriteBlockSize-1] of Byte				A full block of data to write to flash

Reply

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment
Command	0x30				The write instruction has completed

2.5.9 Write Partial Block

This message instructs the micro-controller to write the passed data to the flash. A block size is assumed to be the “WriteBlock Size” retrieved in the Request Flash Information message and starting from the WriteAddress Offset sent in the Set Address Information message or from the current location of the offset stored within the micro-controller, which ever occurred last.

Instruction

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment
Command	0x31				Instructs the micro-controller to write the passed data to the flash.
Count	DWord				Number of valid Bytes in the Data Array. It is a multiple of the size of a Flash write bytes (PIC=1, dsPIC/PIC24=2, PIC32=4).
Data	array[0.. WriteBlock Size-1] of Byte				A partial block of data to write

Reply

Field	Byte 0	Byte 1	Byte 2	Byte 3	Comment
Command	0x31				The write instruction has completed

2.6 States

2.7 Interactions

2.8 Background Information

Table of Contents