Chameleon USB Developer Guide

Draft Version (Beta 9h)

07/08/18

1. Contents

1. Contents....2
2. USB Protocol....3
  2.1. USB ID....3
  2.2. Packet format....3
  2.3. Commands....3
3. Flash ROM Layout....6
  3.1. Core Length....6
  3.2. Core Binary....6
  3.3. Coreinfo block....6
  3.4. Additional ROM Data....6
  3.5. Configuration Data....7
4. Fine print....8

2. USB Protocol

2.1. USB ID

Vendor ID	VID	0x18D8	Individual Computers
Product ID	PID	0x201D	Chameleon

2.2. Packet format

All USB packets are 34bytes:

Offs	len		
0	1	Command	Identifies the command
1	1	Control	Additional parameter
2	32	Data	Command data

2.3. Commands

Often a packet without any specified information in it is returned. This is used to determine when the microcontroller is ready to receive more data and to confirm a command was received.

All Flash Commands (0x01, 0x02, 0x03, 0x13) will only work when the SPI bus is available and not occupied by the FPGA – if in doubt use the FPGA-reset command (0x07) first.

Memory Read/Write (0x90, 0x92, 0x93, 0x9f) commands only work when the FPGA has been started and the respective core supports these commands.

Command	Cmd	Ctrl	Data	Description
Status request	0x00	-	-	Reads Status and returns a packet: Offs: 0 – SPI register; 1 – FPGA status; 2 – Bricked status

<table>
<tr><th>Command</th><th>Cmd</th><th>Ctrl</th><th>Data</th><th>Description</th></tr>
<tr><td>Read Flash</td><td>0x01</td><td>-</td><td>-</td><td>Reads 32 byte from flash ROM using current read pointer and returns a packet:
<table><tr><td>Offs</td><td></td></tr><tr><td>0...31</td><td>Data</td></tr></table></td></tr>
<tr><td>Write Flash</td><td>0x02</td><td>0x00</td><td><table><tr><td>Offs</td><td></td></tr><tr><td>0...31</td><td>Data to write</td></tr></table></td><td>Writes 32 bytes to flash ROM using the current write pointer and returns a packet containing the flash status register:
<table><tr><td>Offs</td><td></td></tr><tr><td>0</td><td>Flash status</td></tr></table></td></tr>
<tr><td>Sector Erase</td><td>0x03</td><td>-</td><td><table><tr><td>Offs</td><td></td></tr><tr><td>0</td><td>Sector</td></tr></table></td><td>Erases the given sector in flash ROM and returns a packet containing the flash status register:
<table><tr><td>Offs</td><td></td></tr><tr><td>0</td><td>Flash status</td></tr></table></td></tr>
<tr><td>Pointerreset</td><td>0x04</td><td>-</td><td>-</td><td>Resets read and write pointers to 0</td></tr>
<tr><td>Read Flash ID</td><td>0x13</td><td>-</td><td>-</td><td>Reads the Flash ID and returns a packet:
<table><tr><td>Offs</td><td></td></tr><tr><td>0</td><td>Manufacturer ID</td></tr><tr><td>1</td><td>Flash ID</td></tr></table></td></tr>
<tr><td>Start FPGA</td><td>0x06</td><td>slot</td><td>-</td><td>Starts the FPGA from the given slot and returns a packet:
<table><tr><td>Offs</td><td></td></tr><tr><td>0</td><td>Core size (3 bytes, MSB first)</td></tr></table></td></tr>
<tr><td>Reset FPGA</td><td>0x07</td><td>-</td><td>-</td><td>Reset/Clear the FPGA and enable SPI bus</td></tr>
<tr><td>Set JTAG slot</td><td>0x08</td><td>-</td><td><table><tr><td>Offs</td><td></td></tr><tr><td>0</td><td>slot</td></tr></table></td><td>Sets FPGA JTAG slot to given slot</td></tr>
<tr><td>Set read pointer</td><td>0xB0</td><td>-</td><td><table><tr><td>Offs</td><td></td></tr><tr><td>0</td><td>Addr LSB</td></tr><tr><td>...</td><td></td></tr><tr><td>3</td><td>Addr MSB</td></tr></table></td><td>Set read pointer to given address</td></tr>
</table>

<table>
<tr><th>Command</th><th>Cmd</th><th>Ctrl</th><th>Data</th><th>Description</th></tr>
<tr><td>Set Write pointer</td><td>0xB1</td><td>-</td><td><table><tr><td>Offs</td><td></td></tr><tr><td>0</td><td>Addr LSB</td></tr><tr><td>...</td><td></td></tr><tr><td>3</td><td>Addr MSB</td></tr></table></td><td>Set write pointer to given address</td></tr>
<tr><td>Read Chameleon memory</td><td>0x90</td><td>-</td><td><table><tr><td>Offs</td><td></td></tr><tr><td>0</td><td>Addr LSB</td></tr><tr><td>...</td><td></td></tr><tr><td>3</td><td>Addr MSB</td></tr></table></td><td>Initiates reading from the FPGA (RAM) at the given address.

This command is specific to the FPGA core being used and must be supported by it.</td></tr>
<tr><td>Write Chameleon memory</td><td>0x92</td><td>-</td><td><table><tr><td>Offs</td><td></td></tr><tr><td>0</td><td>Addr LSB</td></tr><tr><td>...</td><td></td></tr><tr><td>3</td><td>Addr MSB</td></tr></table></td><td>Introduces writing to the FPGA (RAM) at the given address. Microcontroller returns a packet.

This command is specific to the FPGA core being used and must be supported by it.</td></tr>
<tr><td>Memory write data</td><td>0x93</td><td>num</td><td><table><tr><td>Offs</td><td></td></tr><tr><td>0</td><td>Num bytes data</td></tr></table></td><td>Must follow a 0x92 command. Contains num bytes that will be sent to the FPGA. Microcontroller returns a packet.

This command is specific to the FPGA core being used and must be supported by it.</td></tr>
<tr><td>Write Stop</td><td>0x9F</td><td>-</td><td>-</td><td>Abort writing. Microcontroller returns a packet.</td></tr>
<tr><td>Chameleon version</td><td>0xF0</td><td>-</td><td>-</td><td>Microcontroller returns a packet:

<table><tr><td>Offs</td><td></td></tr><tr><td>0</td><td>SD-Card present</td></tr><tr><td>1</td><td>Firmware version</td></tr></table></td></tr>
<tr><td>Start Bootloader</td><td>0xF1</td><td>-</td><td>-</td><td>Start the bootloader of the Microcontroller

Note: This command is broken in firmware version 0x11, which was shipped with the first chameleon version (with mini-din breakout cable)</td></tr>
</table>

3. Flash ROM Layout

The Flash ROM of the Chameleon is organized into 16 blocks of 1MB, which each may contain its own FPGA Core (which can be started from the Turbo Chameleon 64 Main Menu). A core binary may be followed by additional ROM data.

3.1. Core Length

Offset	Length	
+0	3	Offset to first byte behind Core (and Coreinfo) → ROM Offset

3.2. Core Binary

Offset	Length	
+3	N	Core Binary Data (.rbf) with their bits reversed

3.3. Coreinfo block

Offset	Length	
+3+N	4	Magic (“ch64”)
	4	Version (0x00000001)
	4	Core length
	4	Core offset
	4	ROM length
	4	ROM offset
	0x40	Core name
	4	Info length
	4	Info offset

offset and length of this info block come always last so they can be found be seeking backwards from the rom offset which is given as the first 3 bytes before the core binary.

3.4. Additional ROM Data

Offset	Length	
+3+N+0x60	M	ROM Binary Data

Currently the size of the chameleon ROM is max. 0x090000 bytes (9 64kb blocks)

3.5. Configuration Data

The last 64k Block of the 1MB slot is reserved for configuration data

Offset	Length	
`0xf0000`	`0x10000`	Turbo Chameleon 64 Configuration Data

4. Fine print

The Chameleon is not designed, authorized or warranted to be suitable for use in life-support devices or systems or other critical operations. Inclusion of the product in such applications is understood to be fully at the customer's risk.

Names and other trademarks referred to within this manual are the property of their respective trademark holders. None of these trademark holders are affiliated with Individual Computers.

Windows® is a registered trademark of Microsoft Corporation in the United States and/or other countries.

Mac OSX is a registered trademark of Apple Computers in the United States and/or other countries.

For updates and further information visit http://wiki.icomp.de/wiki/Chameleon

Individual Computers Jens Schönfeld GmbH
Im Zemmer 6
52152 Woffelsbach
Germany

good hardware for good computers.