AVR068: STK500 Communication Protocol

Features

- **Interfaces both STK500 and AVRISP**
- **Supports STK500 FW 2.XX**

1 Introduction

This document describes the 2.0 version of the communication protocol between the Atmel STK500 and the PC controlling the STK500. The firmware is distributed with AVR Studio 4.11 build 401 or later.

The definition of all commands, responses, parameters and other defined values can be found in the file “command.h”. This file can be downloaded from the Atmel web site.

All device specific values can be found in the xml files for each part. The xml files are distributed with AvrStudio 4.11 or later. Download the latest AVR Studio 4.x from the Atmel web site, http://www.atmel.com/products/AVR/. The xml file format is described in section 6.

This document mainly refers to STK500, but mostly everything also applies to the AVRISP. In the few cases where the behavior is different between STK500 and AVRISP, it will be noted.

8-bit AVR® Microcontrollers

Application Note

Rev. 2591B-AVR-03/05

2 Communication interface

The communication between the STK500 and the PC is done over RS232 (PC COM port). The STK500 uses: 115.2kbps, 8 data bits, 1stop bits, no parity. The PC should be set up similarly for the communication to work.

3 Message Format

All commands and responses share a common message format:

Figure 3-1. Message format.

MESSAGE_START
SEQUENCE_NUMBER
MESSAGE_SIZE
TOKEN
MESSAGE_BODY
CHECKSUM

Table 3-1. Common message format fields

Parameter Name	Size/Format	Description
MESSAGE_START	1 byte	Always 0x1B
SEQUENCE_NUMBER	1 byte	Incremented by one for each message sent. Wraps to zero after 0xFF is reached.
MESSAGE_SIZE	2 bytes, MSB first	Size of the message body(1)
TOKEN	1 byte	Always 0x0E
MESSAGE_BODY	*MESSAGE_SIZE* bytes	Message body, from 0 to 65535 bytes (1)
CHECKSUM	1 byte	Uses all characters in message including MESSAGE_START and MESSAGE_BODY. XOR of all bytes.

Note: 1. The current STK500 firmware can only handle messages with a message body of maximum of 275 bytes.

The host (PC) sends commands, the STK500 responds with answers.

One command will result in one answer. STK500 cannot send a message to the host that is not an answer to a received command.

The sequence number for an answer is always the same as for the corresponding command.

4 Protocol Layer State Table

This section describes the state machine in the PC software that handles incoming packets from the STK500. When an incoming packet is expected, the state machine is initialized to the *Start* state.

The table describes all states, events and conditions for the handling of incoming data.

Table 4-1. Protocol Layer State Table

Current State	Event	Condition	Action	Next State
Start	Read character from inbuf	Char == ASCII 27		Get Sequence Number
	Read character from inbuf	Char != ASCII 27	Update statistics	Start
	Total timeout		Exit with error	
Get Sequence Number	Read character from inbuf	Received sequence number == Sent sequence number		Get Message Size
	Read character from inbuf	Received sequence number != Sent sequence number	Update statistics	Start
	Total timeout		Exit with error	
Get Message Size 1	Read character from inbuf			Get Message Size 2
	Total timeout		Exit with error	
Get Message Size 2	Read character from inbuf		Calculate message size	Get Token
	Total timeout		Exit with error	
Get Token	Read character from inbuf	Char == ASCII 14		Get Data
	Read character from inbuf	Char != ASCII 14	Update statistics	Start
	Total timeout		Exit with error	
Get Data	Read character from inbuf	Nmb char read == MESSAGE_SIZE		Get Checksum
	Read character from inbuf	Nmb char read < MESSAGE_SIZE		Get Data
	Total timeout		Exit with error	
Get Checksum	Read character from inbuf	Checksum OK	**Exit successfully and handle packet**	
	Read character from inbuf	Checksum NOT OK	Update statistics	Start
	Total timeout		Exit with error	

The state machine on the STK500 side, which receives incoming packets from the PC, is similar apart from that there is no total timeout in the reception.

The total timeout period is from a command is sent to the answer must be completely received. The total timeout period is *200 ms* for the CMD_SIGN_ON command, *5 seconds* for the CMD_READ/PROGRAM_FLASH/EEPROM commands, and *1 second* for all other commands.

5 Commands

STK500 commands are sent in the MESSAGE_BODY part of a message (see chapter 2). This section describes all commands that can be entered to the STK500, and all the possible responses that each command can give back to the host.

In general, all programming modes use the same protocol commands, but various parameters are sent with the different programming modes in order to avoid unnecessary overhead in the protocol layer.

For all commands, the STK500 will return an answer with an answer ID that is equal to the command ID. The first byte in a command is always the command ID, the first byte in an answer is always the answer ID.

5.1 General Commands

These commands are not related to a specific programming mode.

5.1.1 CMD_SIGN_ON

This command returns a unique signature string for the STK500/AVRISP with this implementation of the protocol. The signature is “STK500_2” or “AVRISP_2”.

Table 5-1. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_SIGN_ON	Command id

Table 5-2. Answer format if target is an STK500

Field	Size	Values	Description
Answer ID	1 byte	CMD_SIGN_ON	Answer id
Status	1 byte	STATUS_CMD_OK	
Signature length	1 byte	8	Length of signature string
	8 bytes	“STK500_2”	The signature string (not null-terminated)

Table 5-3. Answer format if target is an AVRISP

Field	Size	Values	Description
Answer ID	1 byte	CMD_SIGN_ON	Answer id
Status	1 byte	STATUS_CMD_OK	
Signature length	1 byte	8	Length of signature string
	8 bytes	“AVRISP_2”	The signature string (not null-terminated)

5.1.2 CMD_SET_PARAMETER

The host can set a multitude of parameters in the STK500. See *5.7 Parameters* for a description of each parameter. All parameters are one-byte values.

Table 5-4. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_SET_PARAMETER	Command id
Parameter ID	1 byte		Which parameter to set
Value	1 byte		The parameter's new value

Table 5-5. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_SET_PARAMETER	Answer id
Status	1 byte	STATUS_CMD_OK or STATUS_CMD_FAILED	A status value indicating the result of the operation

5.1.3 CMD_GET_PARAMETER

The host can also read different parameters from the STK500.

Table 5-6. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_GET_PARAMETER	Command id
Parameter ID	1 byte		Which parameter to get

Table 5-7. Answer format if command succeeded

Field	Size	Values	Description
Answer ID	1 byte	CMD_GET_PARAMETER	Answer id
Status	1 byte	STATUS_CMD_OK	A status value indicating success
Parameter value	1 byte		The parameter value

Table 5-8. Answer format if command fails

Field	Size	Values	Description
Answer ID	1 byte	CMD_GET_PARAMETER	Answer id
Status	1 byte	STATUS_CMD_ FAILED	A status value indicating that the operation failed.

5.1.4 CMD_OSCCAL

This command performs a calibration sequence as described in application note AVR053.

Table 5-9. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_OSCCAL	Command id

Table 5-10. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_OSCCAL	Answer id
Status	1 byte	STATUS_CMD_OK or STATUS_CMD_FAILED	A status value indicating success or failure.

5.1.5 CMD_LOAD_ADDRESS

This command will load an address into the STK500. The next Program Flash, Read Flash, Program EEPROM or Read EEPROM command will operate from the address set with this command. The command is used in all programming modes. All the abovementioned commands will increment an internal address counter, so this command needs only to be sent once.

Table 5-11. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_LOAD_ADDRESS	Command id
Address	4 bytes		The address, four bytes, MSB first

For word-addressed memories (program flash), the Address parameter is the word address.

If bit 31 is set, this indicates that the following read/write operation will be performed on a memory that is larger than 64KBytes. This is an indication to STK500 that a *load extended address* must be executed. See datasheet for devices with memories larger than 64KBytes.

Table 5-12. Answer format if command succeeded

Field	Size	Values	Description
Answer ID	1 byte	CMD_LOAD_ADDRESS	Answer id
Status	1 byte	STATUS_CMD_OK	A status value indicating success

5.1.6 CMD_FIRMWARE_UPGRADE

When the host is trying to connect to the programmer, it checks the firmware version. A firmware upgrade is initiated if a newer version is available on the PC.

The STK500 can “reboot” into upgrade mode by using this command. (If the PROGRAM button on STK500 is pressed while turning on power, upgrade mode is also entered.)

Table 5-13. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_FIRMWARE_UPGRADE	Command id
Parameter ID	10 bytes	”fwupgrade”	String to enable upgrade mode

Table 5-14. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_FIRMWARE_UPGRADE	Answer id
Status	1 byte	STATUS_CMD_OK or STATUS_CMD_FAILED	A status value indicating the result of the operation

If the status returned is STATUS_CMD_OK, the STK500 will disconnect and enter upgrade mode.

5.2 ISP Programming Commands

These commands handles FLASH, EEPROM, fuse bytes, lock bits, signature and oscillator calibration programming in ISP mode.

5.2.1 CMD_ENTER_PROGMODE_ISP

This command will make the target device enter programming mode.

XML path: /AVRPART/ICE_SETTINGS/STK500_2/IspEnterProgMode/

Table 5-15. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_ENTER_ PROGMODE_ISP	Command id
timeout	1 byte	XML: timeout	Command time-out (in ms)
stabDelay	1 byte	XML: stabDelay	Delay (in ms) used for pin stabilization
cmdexeDelay	1 byte	XML: cmdexeDelay	Delay (in ms) in connection with the EnterProgMode command execution
synchLoops	1 byte	XML: synchLoops	Number of synchronization loops
byteDelay	1 byte	XML: byteDelay	Delay (in ms) between each byte in the EnterProgMode command.
pollValue	1 byte	XML: pollValue	Poll value: 0x53 for AVR, 0x69 for AT89xx
pollIndex	1 byte	XML: pollIndex	Start address, received byte: 0 = no polling, 3 = AVR, 4 = AT89xx
cmd1	1 byte		Command Byte # 1 to be transmitted
cmd2	1 byte		Command Byte # 2 to be transmitted
cmd3	1 byte		Command Byte # 3 to be transmitted
cmd4	1 byte		Command Byte # 4 to be transmitted

Note: The pollValue parameter indicates after which of the transmitted bytes on the SPI interface to store the return byte, as the SPI interface is implemented as a ring buffer (one byte out, one byte in).

Table 5-16. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_ENTER_ PROGMODE_ISP	Answer id
Status	1 byte	STATUS_CMD_TOUT, STATUS_CMD_OK or STATUS_CMD_FAILED	A status value indicating the result of the operation

5.2.2 CMD_LEAVE_PROGMODE_ISP

This command will make STK500 leave programming mode. The device will be put into normal operating mode.

XML path: /AVRPART/ICE_SETTINGS/STK500_2/IspLeaveProgMode/

Table 5-17. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_LEAVE_PROGMODE_ISP	Command id
preDelay	1 byte	XML: preDelay	Pre-delay (in ms)
postDelay	1 byte	XML: postDelay	Post-delay (in ms)

Table 5-18. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_LEAVE_PROGMODE_ISP	Answer id
Status	1 byte	STATUS_CMD_OK	This command will always return STATUS_CMD_OK

5.2.3 CMD_CHIP_ERASE_ISP

This command will perform a chip erase on the target device.

XML path: /AVRPART/ICE_SETTINGS/STK500_2/IspChipErase/

Table 5-19. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_CHIP_ERASE_ISP	Command id
eraseDelay	1 byte	XML: eraseDelay	Delay (in ms) to ensure that the erase of the device is finished
pollMethod	1 byte	XML: pollMethod	Poll method, 0 = use delay 1= use RDY/BSY command
cmd1	1 byte		Command Byte # 1 to be transmitted
cmd2	1 byte		Command Byte # 2 to be transmitted
cmd3	1 byte		Command Byte # 3 to be transmitted
cmd4	1 byte		Command Byte # 4 to be transmitted

Table 5-20. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_CHIP_ERASE_ISP	Answer id
Status	1 byte	STATUS_CMD_OK or STATUS_CMD_TOUT	A status value indicating the result of the operation

5.2.4 CMD_PROGRAM_FLASH_ISP

This command will program data into the FLASH memory of the target device if it succeeds.

XML path: /AVRPART/ICE_SETTINGS/STK500_2/IspProgramFlash/

Table 5-21. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_PROGRAM_FLASH_ISP	Command id
NumBytes	2 byte		Total number of bytes to program, MSB first
mode	1 byte	XML: mode	Mode byte (explained below)
delay	1 byte	XML: delay	Delay, used for different types of programming termination, according to mode byte
cmd1	1 byte		Command 1 (Load Page, Write Program Memory)
cmd2	1 byte		Command 2 (Write Program Memory Page)
cmd3	1 byte		Command 3 (Read Program Memory)
poll1	1 byte	XML: pollVal1	Poll Value #1
poll2	1 byte	XML: pollVal2	Poll Value #2 (not used for flash programming)
Data	N bytes		N data

Table 5-22. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_PROGRAM_FLASH_ISP	Answer id
Status	1 byte	STATUS_CMD_OK, STATUS_CMD_TOUT, STATUS_RDY_BSY_TOUT	A status value indicating the result of the operation

5.2.4.1 Mode byte description

The *mode* parameter is essential for how this command works. The bits in the mode byte have the following meanings:

Table 5-23. The bits in the mode byte

Bit #	Description	Mode
0	Word/Page Mode (0 = word, 1 = page)	
1	Timed delay	Word Mode
2	Value polling	
3	RDY/BSY polling	
4	Timed delay	Page Mode
5	Value polling	
6	RDY/BSY polling	
7	Write page	

The *Word/Page Mode* bit selects if the device supports page programming or not.

The command bytes are different for word and page mode. In word mode, the ISP commands *Write Program Memory* and *Read Program Memory* are used. In page mode, *Load Page*, *Write Program Memory Page* and *Read Program Memory* are used. The read instruction is used if *Value Polling* is specified in the mode bit. The Low/High byte selection bit (3rd bit in the Load Page, Write Program Memory commands) is handled by STK500, so leave this bit cleared.

According to the mode, different termination methods are selected – *Timed delay*, *Value polling* or *RDY/BSY polling*.

For paged operation, the *Write page* bit decides if a *Write Program Memory Page* command should be issued after the data has been loaded into the page buffer. For devices with page size bigger than what can be transferred to STK500 in one command, several CMD_PROGRAM_FLASH_ISP commands must be issued. In such a case, only the last command should have the *Write Page* mode bit set.

Note: Only bit 0-6 are set in the XML file, because bit 7 is not constant and must be controlled by the PC software.

When *value polling* is used to determine when a programming operation is complete, *poll1* must be supplied. This value indicates which value will be read from the device until the programmed value is read. This indicates end of programming. *poll2* is used only for EEPROM programming.

5.2.5 CMD_READ_FLASH_ISP

This command will read data from the FLASH memory of the target device if it succeeds.

XML path: /AVRPART/ICE_SETTINGS/STK500_2/IspReadFlash/

Table 5-24. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_READ_FLASH_ISP	Command id
NumBytes	2 byte	XML: blockSize	Total number of bytes to read, MSB first
cmd1	1 byte		Read Program Memory command byte #1. Low/High byte selection bit (3rd bit) is handled in the FIRMWARE.

Table 5-25. Answer format if the command is executed

Field	Size	Values	Description
Answer ID	1 byte	CMD_READ_FLASH_ISP	Answer id
Status1	1 byte	STATUS_CMD_OK	Indicates success. Will always read OK
Data	N bytes		The data read from the device
Status2	1 byte	STATUS_CMD_OK	A status value indicating the result of the operation. Will always read OK

Table 5-26. Answer format if the command was not executed

Field	Size	Values	Description
Answer ID	1 byte	CMD_READ_FLASH_ISP	Answer id
Status	1 byte	STATUS_CMD_FAILED	Indicates failure

5.2.6 CMD_PROGRAM_EEPROM_ISP

See the CMD_PROGRAM_FLASH_ISP command.

5.2.7 CMD_READ_EEPROM_ISP

See the CMD_READ_FLASH_ISP command.

5.2.8 CMD_PROGRAM_FUSE_ISP

This command programs the fuses of the target device.

Table 5-27. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_PROGRAM_FUSE_ISP	Command id
cmd1	1 byte		Command Byte #1
cmd2	1 byte		Command Byte #2
cmd3	1 byte		Command Byte #3
cmd4	1 byte		Command Byte #4

Note: cmd1, cmd2, cmd3 and cmd4 are the four bytes of the low-level program fuse ISP command

Table 5-28. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_PROGRAM_FUSE_ISP	Answer id
Status1	1 byte	STATUS_CMD_OK	Will always read OK
Status2	1 byte	STATUS_CMD_OK	Will always read OK

5.2.9 CMD_READ_FUSE_ISP

This command reads the fuses of the target device.

Table 5-29. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_READ_FUSE_ISP	Command id
RetAddr	1 byte	XML: pollIndex	Return address
cmd1	1 byte		Command Byte #1
cmd2	1 byte		Command Byte #2
cmd3	1 byte		Command Byte #3
cmd4	1 byte		Command Byte #4

Note: RetAddr indicates after which of the transmitted bytes on the SPI interface to store the return byte, as the SPI interface is implemented as a ring buffer (one byte out, one byte in)

Table 5-30. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_READ_FUSE_ISP	Answer id
Status1	1 byte	STATUS_CMD_OK	A status value indicating the result of the operation, always OK
data	1 byte		The fuse byte read from the device
Status2	1 byte	STATUS_CMD_OK	A status value indicating the result of the operation, always OK

5.2.10 CMD_PROGRAM_LOCK_ISP

See CMD_PROGRAM_FUSE. This command is basically the same as the program fuse command, only that ISP commands for programming the lock byte must be supplied.

5.2.11 CMD_READ_LOCK_ISP

See CMD_READ_FUSE. This command is basically the same as the read fuse command, only that ISP commands for reading the lock byte must be supplied.

5.2.12 CMD_READ_SIGNATURE_ISP

See CMD_READ_FUSE. This command is basically the same as the read fuse command, only that ISP commands for reading one of the signature bytes must be supplied.

5.2.13 CMD_READ_OSCCAL_ISP

See CMD_READ_FUSE. This command is basically the same as the read fuse command, only that ISP commands for reading the OSCCAL byte must be supplied.

5.2.14 CMD_SPI_MULTI

This is a generic command that can be used to execute any of the ISP commands. The command writes a number of bytes to the SPI bus, and returns a number of bytes.

Table 5-31. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_SPI_MULTI	Command id
NumTx	1 byte	0-255	Number of bytes to transmit
NumRx	1 byte	0-255	Number of bytes to receive
RxStartAddr	1 byte		Start address of returned data. Specifies on what transmitted byte the response is to be stored and returned.
TxData	0-255 bytes		The data be transmitted. The size is specified by NumTx

If the number of bytes to receive is greater than number of bytes to transmit, then the firmware will pad with the necessary 0x00 bytes. This is in order to save time-consuming transfer from PC to the programmer.

Table 5-32. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_SPI_MULTI	Answer id
Status1	1 byte	STATUS_CMD_OK	Will always read OK
data	0-255 bytes		The data read from the ISP bus as indicated in the command
Status2	1 byte	STATUS_CMD_OK	Will always read OK

5.3 Parallel Programming Mode Commands

5.3.1 CMD_ENTER_PROGMODE_PP

This command will make the target device enter programming mode if it succeeds.

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpEnterProgMode/

Table 5-33. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_ENTER_ PROGMODE_PP	Command id
stabDelay	1 byte	XML: stabDelay	Delay (in ms) used for pin stabilization
progModeDelay	1 byte	XML: progModeDelay	Delay (in ms) in connection with the EnterProgMode command execution
latchCycles	1 byte	XML: latchCycles	Number of xtal cycles used to latch OSCCAL
toggleVtg	1 byte	XML: toggleVtg	Toggle Vtg when entering prog.mode (0=no, 1=yes). For parts with RSTDSBL functionality
powerOffDelay	1 byte	XML: powerOffDelay	Power-off delay. Additional delay (in ms) after Vtg is turned off in order to make sure the Vtg is low enough
resetDelayMs	1 byte	XML: resetDelayMs	RSTDELAY #1 (in ms) Additional delay between Vtg is turned on and reset goes high.
resetDelayUs	1 byte	XML: resetDelayUs	RSTDELAY #2(in us x 10) Additional delay between Vtg is turned on and reset goes high. Total delay is RSTDELAY #1 (ms) + RSTDELAY #2 (us x 10)

Table 5-34. Answer format (same for all results)

Field	Size	Values	Description
Answer ID	1 byte	CMD_ENTER_ PROGMODE_PP	Answer id
Status	1 byte	See table below	A *Result Value* indicating the result of the operation

Table 5-35. Valid *Result Values* for the answer to this command

Values	Description
STATUS_CMD_OK	Operation succeeded
STATUS_CMD_ FAILED	Operation failed

5.3.2 CMD_LEAVE_PROGMODE_PP

This command will make the target device leave programming mode if it succeeds.

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpLeaveProgMode/

Table 5-36. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_LEAVE_ PROGMODE_PP	Command id
stabDelay	1 byte	XML: stabDelay	Delay (in ms) used for pin stabilization
resetDelay	1 byte	XML: resetDelay	Delay (ms) for holding RESET low

Table 5-37. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_LEAVE_ PROGMODE_PP	Answer id
Status	1 byte	STATUS_CMD_OK	A *Result Value* indicating the result of the operation

5.3.3 CMD_CHIP_ERASE_PP

This command will perform a chip erase on the target device if it succeeds.

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpChipErase/

Table 5-38. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_CHIP_ ERASE_PP	Command id
pulseWidth	1 byte	XML: pulseWidth	Width (in ms) of the /WR pulse. 0 => 1 cycle
pollTimeout	1 byte	XML: pollTimeout	Timeout period (in ms) to wait for RDY/BSY flag to rise. If 0, RDY/BSY flag is NOT used.

Table 5-39. Answer format (same for all results)

Field	Size	Values	Description
Answer ID	1 byte	CMD_CHIP_ ERASE_PP	Answer id
Status	1 byte	See table below	A *Result Value* indicating the result of the operation

Table 5-40. Valid *Result Values* for the answer to this command

Values	Description
STATUS_CMD_OK	Operation succeeded
STATUS_RDY_BSY_ TOUT	No response from target device within specified timeframe

5.3.4 CMD_PROGRAM_FLASH_PP

This command will program data into the FLASH memory of the target device if it succeeds. For devices with the FLASH organized in pages, the data address and size used with this command must confirm to that of the device. I.e. one ProgramFlash command is used to program one page in the target device.

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpProgramFlash/

Table 5-41. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_PROGRAM_ FLASH_PP	Command id
Nmb bytes (MSB)	1 byte		Total number of bytes to program (MSB)
Nmb bytes (LSB)	1 byte		Total number of bytes to program (LSB)
mode	1 byte	XML: mode	Mode byte (explained below)
pollTimeout	1 byte	XML: pollTimeout	`pollTimeout` (in ms)
Data	1 byte	Data 1	
Data	1 byte	...	
Data	1 byte	Data N	

Table 5-42. Answer format (same for all results)

Field	Size	Values	Description
Answer ID	1 byte	CMD_PROGRAM_ FLASH_PP	Answer id
Status	1 byte	See table below	A *Result Value* indicating the result of the operation

Table 5-43. Valid *Result Values* for the answer to this command

Values	Description
STATUS_CMD_OK	Operation succeeded
STATUS_RDY_BSY_ TOUT	No response from target device within specified timeframe

5.3.4.1 Mode byte description

- Bit 0: This bit indicates whether to use byte '0' or page '1' programming.
- Bit 1-3 are the pagesize bits, pagesize are given in bytes not words, see table below:

Table 5-44. Pagesize bit configuration

Pagesize	Bit 3, 2, 1
256	0 0 0
2	0 0 1
4	0 1 0
8	0 1 1
16	1 0 0
32	1 0 1
64	1 1 0
128	1 1 1

- Bit 4-5 are not in use.
- Bit 6 must be set to '1' when it is the very last page to be programmed, otherwise '0'
- Bit 7 indicates if a page write should be issued (*Transfer data to flash*). Normally it should always be set '1'. However, if the page size of the target device is too large to be covered by one Program Flash command (because the amount of available SRAM in STK500 is limited) this can be used to let 2 or more commands fill the page buffer of the target device. The *transfer data to flash* flag should then only be set on the last command.

Note: Only Bit 0-3 are set in the XML file, cause Bit 6-7 are not static and must be controlled by the PC Frontend.

5.3.5 CMD_READ_FLASH_PP

This command will read data from the FLASH memory of the target device if it succeeds. For devices with the FLASH organized in pages, the data address and size used with this command must confirm to that of the device. I.e. one ReadFlash command is used to read one page in the target device.

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpReadFlash/

Table 5-45. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_READ_FLASH_PP	Command id
Nmb bytes (MSB)	1 byte		Total number of bytes to program (MSB)
Nmb bytes (LSB)	1 byte		Total number of bytes to program (LSB)

Table 5-46. Answer format if the command is executed

Field	Size	Values	Description
Answer ID	1 byte	CMD_READ_FLASH_PP	Answer id
Status	1 byte	STATUS_CMD_OK	
Data	Nmb bytes		Data read from device. Will be padded with 0's if errors occurred during read-out.
Status	1 byte	STATUS_CMD_OK	A *Result Value* indicating the result of the operation

5.3.6 CMD_PROGRAM_EEPROM_PP

This command programs one page the EEPROM memory of the target device if it succeeds.

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpReadEeprom/

Command format: See CMD_PROGRAM_FLASH_PP.

5.3.7 CMD_READ_EEPROM_PP

This command will read data from the EEPROM memory of the target device if it succeeds.

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpReadEeprom/

Command format: See CMD_READ_FLASH_PP

5.3.8 CMD_PROGRAM_FUSE_PP

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpProgramFuse/

Table 5-47. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_PROGRAM_ FUSE_PP	Command id
Address	1 byte		Address of fuse byte to program (low, high, ext, ext2)
Data	1 byte		Fuse byte value to be programmed
pulseWidth	1 byte	XML: pulseWidth	Width (ms) of /WR pulse (0 => 1 cycle)
pollTimeout	1 byte	XML: pollTimeout	Time-out (ms) for polling RDY/BSY (0 = don't poll)

Table 5-48. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_PROGRAM_ FUSE_PP	Answer id
Status	1 byte	See table below	A *Result Value* indicating the result of the operation

Table 5-49. Valid *Result Values* for the answer to this command

Values	Description
STATUS_CMD_OK	Operation succeeded
STATUS_RDY_BSY_ TOUT	No response from target device within specified timeframe

5.3.9 CMD_READ_FUSE_PP

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpReadFuse/

Table 5-50. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_READ_FUSE_PP	Command id
Address	1 byte		Address of fuse byte to read

Table 5-51. Answer format if command is executed

Field	Size	Values	Description
Answer ID	1 byte	CMD_ READ_FUSE	Answer id
Status	1 byte	STATUS_CMD_OK	
Data	1 byte		Data read from device. Contains the read low, high or ext fuse byte. Will be padded with 0's if errors occurred during read-out.

5.3.10 CMD_PROGRAM_LOCK_PP

See CMD_PROGRAM_FUSE.

Note: Address must be sent but is ignored by firmware.

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpProgramLock/

5.3.11 CMD_READ_LOCK_PP

See CMD_READ_FUSE_PP.

Note: Address must be sent but is ignored by firmware.

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpReadLock/

5.3.12 CMD_READ_SIGNATURE_PP

See CMD_READ_FUSE.

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpReadSign/

5.3.13 CMD_READ_OSCCAL_PP

See CMD_READ_FUSE.

XML PATH: /AVRPART/ICE_SETTINGS/STK500_2/PpReadOsccal/

5.3.14 CMD_SET_CONTROL_STACK

Uploads the control stack to the STK. This is used for both PP and HVSP.

Note: The Control stack must always be uploaded before performing any programming commands in high voltage mode if the STK500 has been powered down.

To check if the controller has a valid control stack: Read PARAM_CONTROLLER_INIT

Se chapter 5.7.14: PARAM_CONTROLLER_INIT

Table 5-52. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_SET_CONTROL_STACK	Command id
Data	32 bytes		Control stack Data

Table 5-53. Answer format (same for all results)

Field	Size	Values	Description
Answer ID	1 byte	CMD_SET_CONTROL_STACK	Answer id
Status	1 byte	STATUS_CMD_OK	A *Result Value* indicating the result of the operation

5.4 High Voltage Serial Programming Commands

This chapter describes the High Voltage Serial Programming (HVSP) commands. Note that the **SetControlStack** command is required for HVSP as for PP. Description of the **SetControlStack** is found in section 5.3.14 under Parallel Programming.

5.4.1 CMD_ENTER_PROGMODE_HVSP

This command will make the target device enter programming mode if it succeeds.

XML path: /AVRPART/ICE_SETTINGS/STK500_2/HvspEnterProgMode/

Table 5-54. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_ENTER_ PROGMODE_HVSP	Command id
StabDelay	1 byte	XML: stabDelay	Delay (in ms) used for pin stabilization
CmdexeDela y	1 byte	XML: cmdexeDelay	Delay (in ms) in connection with the EnterProgMode command execution
SynchCycles	1 byte	XML: synchCycles	Number of synchronization clock cycles
LatchCycles	1 byte	XML: latchCycles	Number of PulseXtal1_HVSP cycles
ToggleVtg	1 byte	XML: toggleVtg	Toggle Vtg when entering prog.mode (0=no, 1=yes). For parts with RSTDSBL functionality
PowoffDelay	1 byte	XML: powoffDelay	Power-off delay. Additional delay (in ms) after Vtg is turned off in order to make sure the Vtg is low enough
resetDelay1	1 byte	XML: resetDelay1	RSTDELAY #1 (in ms) Additional delay between Vtg is turned on and reset goes high.
resetDelay2	1 byte	XML: resetDelay2	RSTDELAY #2(in us x 10) Additional delay between Vtg is turned on and reset goes high. Total delay is RSTDELAY #1 (ms) + RSTDELAY #2 (us x 10)

Table 5-55. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_ENTER_ PROGMODE_HVSP	Answer id
Status	1 byte	STATUS_CMD_OK or STATUS_CMD_FAILED	A *Result Value* indicating the result of the operation. If failed the target voltage is >5.5V or <4.5V

5.4.2 CMD_LEAVE_PROGMODE_HVSP

This command will make the target device leave programming mode if it succeeds.

XML path: /AVRPART/ICE_SETTINGS/STK500_2/HvspLeaveProgMode/

Table 5-56. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_LEAVE_ PROGMODE_HVSP	Command id
stabDelay	1 byte	XML: stabDelay	Delay (in ms) used for pin stabilization
resetDelay	1 byte	XML: resetDelay	Delay (in ms) in connection with the LeaveProgMode command execution and Reset_low_duration

Table 5-57. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_LEAVE_ PROGMODE_HVSP	Answer id
Status	1 byte	STATUS_CMD_OK	A *Result Value* indicating the result of the operation. Will always read OK

5.4.3 CMD_CHIP_ERASE_HVSP

This command will perform a chip erase on the target device if it succeeds.

XML path: /AVRPART/ICE_SETTINGS/STK500_2/HvspChipErase/

Table 5-58. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_CHIP_ERASE_ HVSP	Command id
pollTimeout	1 byte	XML: pollTimeout	Timeout period (in ms) to wait for RDY/BSY flag to rise. If 0, RDY/BSY flag is NOT used.
eraseTime	1 byte	XML: eraseTime	Delay (in ms) to ensure that the erase of the device is finished. If 0, polling will be used.

Table 5-59. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_CHIP_ERASE_ HVSP	Answer id
Status	1 byte	STATUS_CMD_OK or STATUS_RDY_BSY_TOUT	A *Result Value* indicating the result of the operation.

5.4.4 CMD_PROGRAM_FLASH_HVSP

This command will program data into the FLASH memory of the target device if it succeeds. For devices with the FLASH organized in pages, the data address and size used with this command must confirm to that of the device. I.e. one ProgramFlash command is used to program one page in the target device.

XML path: /AVRPART/ICE_SETTINGS/STK500_2/HvspProgramFlash/

Table 5-60. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_PROGRAM_FLASH_HVSP	Command id
NumBytes	2 byte	XML: blockSize	Total number of bytes to program, MSB first
Mode	1 byte	XML: mode	Mode byte (explained below)
pollTimeout	1 byte	XML: pollTimeout	pollTimeout (in ms)
Data 1	1 byte		Data 1
…	.....		.....
Data N	1 byte		Data N

Table 5-61. Answer format

Field	Size	Values	Description
Answer ID	1 byte	CMD_PROGRAM_FLASH_HVSP	Answer id
Status	1 byte	STATUS_CMD_OK or STATUS_RDY_BSY_TOUT	A *Result Value* indicating the result of the operation.

5.4.4.1 Mode byte description

- Bit 0: This bit indicates whether to use byte ‘0’ or page ‘1’ programming.
- Bit 1-3 are the pagesize bits, pagesize are given in bytes not words, see table below:

Table 5-62. Pagesize bit configuration

Pagesize	Bit 3, 2, 1
256	0 0 0
2	0 0 1
4	0 1 0
8	0 1 1
16	1 0 0
32	1 0 1
64	1 1 0
128	1 1 1

- Bit 4-5 are not in use.
- Bit 6 must be set to ‘1’ when it is the very last page to be programmed, otherwise ‘0’
- Bit 7 indicates if a page write should be issued (*Transfer data to flash*). Normally it should always be set ‘1’. However, if the page size of the target device is too large to be covered by one Program Flash command (because the amount of available SRAM in STK500 is limited) this can be used to let 2 or more commands fill the page buffer of the target device. The *transfer data to flash* flag should then only be set on the last command.

Note: Only Bit 0-3 are set in the XML file, cause Bit 6-7 are not static and must be controlled by the PC Frontend.

5.4.5 CMD_READ_FLASH_HVSP

This command will read data from the FLASH memory of the target device if it succeeds. For devices with the FLASH organized in pages, the data address and size used with this command must confirm to that of the device. I.e. one ReadFlash command is used to read one page in the target device.

XML path: /AVRPART/ICE_SETTINGS/STK500_2/HvspReadFlash/

Table 5-63. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_READ_FLASH_HVSP	Command id
NumBytes	2 byte	XML: blockSize	Total number of bytes to read, MSB first

Table 5-64. Answer format if the command is executed

Field	Size	Values	Description
Command ID	1 byte	CMD_READ_FLASH_HVSP	Answer id
Status1	1 byte	STATUS_CMD_OK	Temporary status, will always read OK
Data	Byte*N		Data read from device. Will be padded with 0's if errors occurred during read-out.
Status2	1 byte	STATUS_CMD_OK	A *Result Value* indicating the result of the operation. Will always read OK

5.4.6 CMD_PROGRAM_EEPROM_HVSP

See the CMD_WRITE_FLASH_HVSP command.

5.4.7 CMD_READ_EEPROM_HVSP

See the CMD_READ_FLASH_HVSP command.

5.4.8 CMD_PROGRAM_FUSE_HVSP

This command programs one fuse byte, addressed by the Fuse Address byte.

XML path: /AVRPART/ICE_SETTINGS/STK500_2/HvspProgramFuse/

Table 5-65. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_PROGRAM_FUSE_HVSP	Command id
Fuse Address	1 byte	0, 1 or 2	Address of byte to write (0=low, 1=high, 2=ext)
Fuse Byte	1 byte		Fuse byte value (low, high or ext) to be programmed
pollTimeout	1 byte	XML: pollTimeout	Poll timeout

Table 5-66. Answer format

Field	Size	Values	Description
Command ID	1 byte	CMD_PROGRAM_FUSE_HVSP	Answer id
Status	1 byte	STATUS_CMD_OK or STATUS_RDY_BSY_TOUT	A *Result Value* indicating the result of the operation.

5.4.9 CMD_READ_FUSE_HVSP

This command programs one fuse byte, addressed by the Fuse Address byte.

XML path: /AVRPART/ICE_SETTINGS/STK500_2/HvspReadFuse/

Table 5-67. Command format

Field	Size	Values	Description
Command ID	1 byte	CMD_ READ_FUSE_HVSP	Command id
Fuse Address	1 byte	0, 1 or 2	Address of byte to read (0=low, 1=high, 2=ext)

Table 5-68. Answer format if command was executed

Field	Size	Values	Description
Command ID	1 byte	CMD_ READ_FUSE_HVSP	Answer id
Status	1 byte	STATUS_CMD_OK	A *Result Value* indicating the result of the operation. Will always read OK
Fuse Byte	1 byte		Data read from device. Contains the read low, high or ext fuse byte. Will be padded with 0's if errors occurred during read-out.

5.4.10 CMD_PROGRAM_LOCK_HVSP

See CMD_PROGRAM_FUSE_HVSP.

Note: Address is required but ignored.

5.4.11 CMD_READ_LOCK_HVSP

See CMD_READ_FUSE_HVSP.

Note: Address is required but ignored.

5.4.12 CMD_READ_SIGNATURE_HVSP

See CMD_READ_FUSE_HVSP.

5.4.13 CMD_READ_OSCCAL_HVSP

See CMD_READ_FUSE_HVSP.

5.5 Special answers

Answers that don't correspond to a specific command.

5.5.1 ANSWER_CKSUM_ERROR

This is an answer generated by the transport layer, indicating that a message with incorrect checksum has been received.

Table 5-69. Answer format

Field	Size	Values	Description
Command ID	1 byte	ANSWER_CKSUM_ERROR	Answer id
Status	1 byte	ANSWER_CKSUM_ERROR	A status value indicating checksum error

5.6 Return values

This section describes all possible return values and their meaning in detail.

5.6.1 Success

Table 5-70. Success return value

Value	Meaning
STATUS_CMD_OK	Command executed OK

5.6.2 Warnings

All warnings have MSB set to 1 and MSB-1 set to 0.

Table 5-71. Warning return values

Value	Meaning
STATUS_CMD_TOUT	Command timed out
STATUS_RDY_BSY_TOUT	Sampling of the RDY/nBSY pin timed out
STATUS_SET_PARAM_MISSING	The 'Set Device Parameters' have not been executed in advance of this command

5.6.3 Errors

All errors have MSB and MSB-1 set to 1.

Table 5-72. Error return values

Value	Meaning
STATUS_CMD_FAILED	Command failed
STATUS_CKSUM_ERROR	Checksum Error
STATUS_CMD_UNKNOWN	Unknown command

5.7 Parameters

The following parameters can be read and/or written by the CMD_GET_PARAM and CMD_SET_PARAM commands:

Table 5-73. Available parameters

Value	Meaning	R/W
PARAM_BUILD_NUMBER_LOW	Firmware build number, high byte	R
PARAM_BUILD_NUMBER_HIGH	Firmware build number, low byte	R
PARAM_HW_VER	Hardware version	R
PARAM_SW_MAJOR	Firmware version number, major byte	R
PARAM_SW_MINOR	Firmware version number, minor byte	R
PARAM_VTARGET	Target Voltage	RW
PARAM_VADJUST	Adjustable (AREF) voltage	RW
PARAM_OSC_PSCALE	Oscillator timer prescaler value	RW
PARAM_OSC_CMATCH	Oscillator timer compare match value	RW
PARAM_SCK_DURATION	ISP SCK duration	RW
PARAM_TOPCARD_DETECT	Top card detect	R
PARAM_STATUS	Returns status register	R
PARAM_DATA	DATA pins values used in HVPP mode	R
PARAM_RESET_POLARITY	Active low or active high RESET handling	W
PARAM_CONTROLLER_INIT	Controller initialization	RW

5.7.1 PARAM_BUILD_NUMBER_LOW

The PARAM_BUILD_NUMBER_LOW and PARAM_BUILD_NUMBER_HIGH together return a number that is incremented for each build of the firmware. Mainly for ATMEL internal use.

5.7.2 PARAM_BUILD_NUMBER_HIGH

See PARAM_BUILD_NUMBER_LOW.

5.7.3 PARAM_HW_VER

Returns a hardware revision number.

5.7.4 PARAM_SW_MAJOR

The PARAM_SW_MAJOR and PARAM_SW_MINOR returns the firmware version.

5.7.5 PARAM_SW_MINOR

See PARAM_SW_MAJOR.

5.7.6 PARAM_VTARGET

This parameter only applies to STK500, not the AVRISP.

STK500 has a controllable target voltage supply, which can be set and monitored with this parameter. The parameter value is voltage in volts x10, i.e. a parameter value of 42 (decimal) corresponds to 4.2V. The VTARGET voltage is adjustable between 0 and 6V.

5.7.7 PARAM_VADJUST

This parameter only applies to STK500, not the AVRISP.

STK500 has an adjustable analog reference in the same way as the supply voltage. This adjustable reference voltage (AREF) can be set and monitored with the PARAM_ADJUST parameter. The parameter value is voltage in volts x10, i.e. a parameter value of 42 (decimal) corresponds to 4.2V. The VADJUST voltage is adjustable between 0 and 6V.

5.7.8 PARAM_OSC_PSCALE

This parameter only applies to STK500, not the AVRISP.

The STK500 has a programmable clock generator used to supply a clock signal to the target device. The generator is actually the TIMER2 of the AT90S8535 MCU. The timer in running in “Toggle OC2 line” mode, where the timer value is cleared on a compare match. The following code example shows how the firmware handles changes in PARAM_OSC_PSCALE or PARAM_OSC_CMATCH:

```
// Stop Timer2
TCCR2 = 0x18;

// Initialize counter value
TCNT2 = 0xFF;

// Set compare match value
OCR2  = osc_cmatch;

// Set timer operation mode and prescaler
TCCR2 = (0x18 | (0x07 & osc_pscale));
```

The STK500 system clock is f_{sys} = 7.37MHz., which gives the following relation between prescaler and compare values and the resulting frequency:

f = prescaled clock / (compare value + 1) / 2

The prescale clock is the system clock divided by some factor according to this table:

Table 5-74. Prescaler values

Prescaler Value	Prescaled clock frequency
0	0
1	f_{sys}
2	f_{sys} / 8
3	f_{sys} / 32
4	f_{sys} / 64
5	f_{sys} / 128
6	f_{sys} / 256
7	f_{sys} / 1024

The host software can compute the prescale and match values by a using a brute force approach, scanning through all possible prescale and match values, computing the resulting frequency, and using the parameters that gives the best result.

5.7.9 PARAM_OSC_CMATCH

This parameter only applies to STK500, not the AVRISP.

See PARAM_OSC_PSCALE.

5.7.10 PARAM_SCK_DURATION

When using the ISP programming interface, the ISP clock frequency must not exceed what the target device supports. (The maximum ISP clock frequency depends on the device system clock, internal clock division etc.)

The STK500 and AVRISP supports ISP frequencies from 4 kHz up to 1.8 MHz. The value for PARAM_SCK_DURATION can be found using the following algorithm:

```
#define T_STK500        135.63e-9
#define T_AVRISP        271.27e-9
#define B               12.0

unsigned char CalcSckDur(int freq)
{
    if (STK500)
    {
        if (freq >= 1843200)
               sck_dur = 0;
        else if (freq >= 460800)
               sck_dur = 1;
        else if (freq >= 115200)
               sck_dur = 2;
        else if (freq >= 57600)
               sck_dur = 3;
        else
               sck_dur = ceil(1/(2 * B * freq * T_STK500) - 10/B);
    }
    else // if (AVRISP)
    {
        if (freq >= 921600)
               sck_dur = 0;
        else if (freq >= 230400)
               sck_dur = 1;
        else if (freq >= 57600)
               sck_dur = 2;
        else if (freq >= 28800)
               sck_dur = 3;
        else
               sck_dur = ceil(1/(2 * B * freq * T_AVRISP) - 10/B);
    }

    return __min(254, sck_dur);// 255 is an illegal value
}
```

5.7.11 PARAM_TOPCARD_DETECT

This parameter only applies to STK500, not the AVRISP.

Expansion cards can be connected to STK500. Each type of card has an id circuitry, so that the STK500 can detect which type of card that is mounted.

The following IDs are in use:

Table 5-75. Topcard IDs

Card	ID
STK501	0xAA
STK502	0x55
STK503	0xFA
STK504	0xEE
STK505	0xE4
STK520	0xDD

5.7.12 PARAM_DATA

This parameter only applies to STK500, not the AVRISP.

This parameter returns the value on the PROG DATA connector used for High-Volt Parallel Programming.

5.7.13 PARAM_RESET_POLARITY

The STK500 and AVRISP can program both AT90 (AVR) family and AT89 (8051) family of microcontrollers. They have different RESET pin polarity. The AVR has active low reset, while the AT89 has active high.

This parameter sets the polarity of the reset signal. Set the parameter to 1 when programming AVRs, and 0 when programming AT89 controllers.

Note: STK500 and AVRISP store this parameter in EEPROM, so they are available the next time power is applied to the programmers. For STK500, also the RESET button change polarity according to this parameter.

5.7.14 PARAM_CONTROLLER_INIT

This parameter is internally set to 0 when the programmer MCU resets. The host software can write any value it wants to this parameter, and it can be read back later. This parameter is intended to be used as a way of telling if the power on STK500/AVRISP has been lost or turned off and then back again.

This way, the host software can tell if the programmer needs to be initialized again before continuing with its operation.

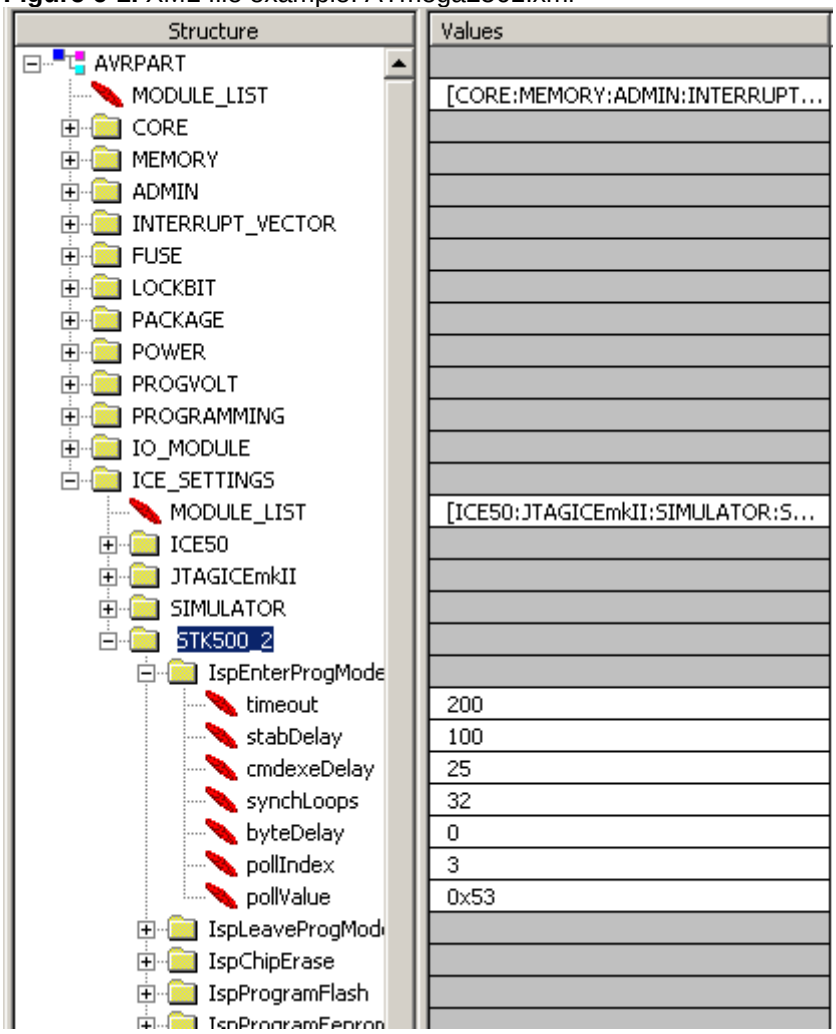
6 XML Parameter Values

One of the major enhancements in the 2.0 version of the STK500 protocol is the ability to describe parameters for all details of the programming algorithms. A parameter set is specific for a certain AVR device, thus the parameter settings belong in the XML part description file. Below is a description where to find the STK500 parameter values in the device specific XML file.

After installing AvrStudio 4, all xml files can be found at:

"\Program Files\Atmel\AVR Tools\PartDescriptionFiles\"

Figure 6-1. XML file example: ATmega2561.xml



Open the XML file in xml editor/viewer (e.g XML Notepad or Internet Explorer). All device specific values for STK500 and AVRISP are located under STK500_2 node. For parameters for the CMD_ENTER_PROGMODE_ISP command, look in /AVRPART/ICE_SETTINGS_STK500_2/IspEnterProgMode.

7 Command Sequence Example

This chapter contains examples of how to connect to the STK500 from the PC Frontend and how to read signature from a device.

See chapter 5 Commands for the description of the commands and parameters.

7.1 Connect

The sequence of commands and parameters sent from AvrStudio to the STK500 in order to connect is listed below.

- CMD_SIGN_ON
- CMD_GET_PARAMETER, PARAM_TOPCARD_DETECT
- CMD_GET_PARAMETER, PARAM_HW_VER
- CMD_GET_PARAMETER, PARAM_SW_MAJOR
- CMD_GET_PARAMETER, PARAM_SW_MINOR

7.2 Read Signature

7.2.1 In System Programming

The sequence of commands and parameters sent from AvrStudio to the STK500 in order to read the device signature through ISP is listed below. Note that one already have to be connected to do this.

- CMD_GET_PARAMETER, PARAM_TOPCARD_DETECT
- CMD_GET_PARAMETER, PARAM_CONTROLLER_INIT
- CMD_SET_PARAMETER, PARAM_CONTROLLER_INIT
- CMD_SET_PARAMETER, PARAM_RESET_POLARITY
- CMD_ENTER_PROGMODE_ISP
- CMD_READ_SIGNATURE_ISP
- CMD_READ_SIGNATURE_ISP
- CMD_READ_SIGNATURE_ISP
- CMD_LEAVE_PROGMODE_ISP

7.2.2 High Voltage Programming

The sequence of commands and parameters sent from AvrStudio to the STK500 in order to read the device signature through High Voltage Programming is listed below. Note that one already have to be connected to do this. This sequence goes for both High Voltage Parallel Programming and High Voltage Serial Programming.

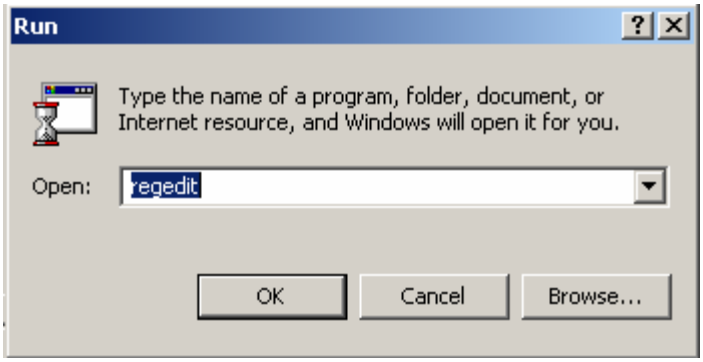
- CMD_GET_PARAMETER, PARAM_TOPCARD_DETECT
- CMD_GET_PARAMETER, PARAM_CONTROLLER_INIT
- CMD_SET_CONTROL_STACK
- CMD_GET_PARAMETER, PARAM_CONTROLLER_INIT
- CMD_ENTER_PROGMODE_PP (_HVSP)
- CMD_READ_SIGNATURE_PP (_HVSP)
- CMD_READ_SIGNATURE_PP (_HVSP)
- CMD_READ_SIGNATURE_PP (_HVSP)
- CMD_LEAVE_PROGMODE_ PP (_HVSP)

7.3 STK500 Communication Logging

For further details and examples of the communication between AvrStudio and STK500 one can set up logging of all communication to a text file. This can be done by adding a register key in the Registry as described below.

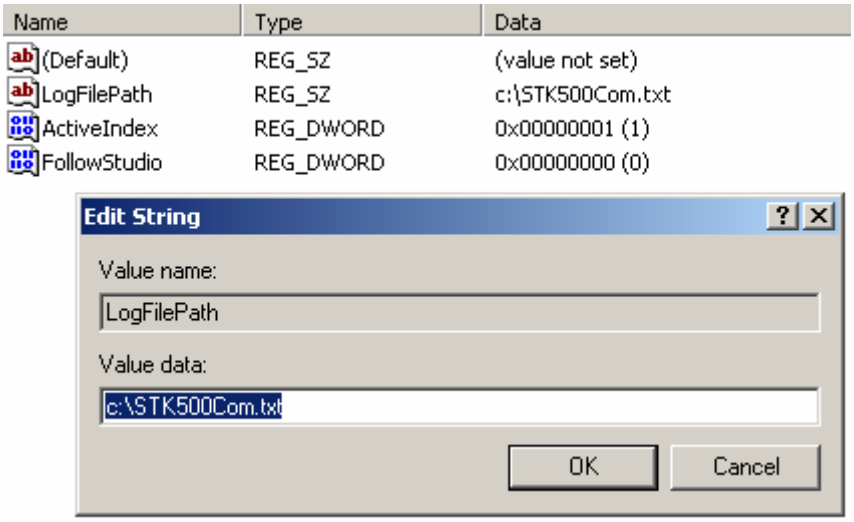
1. Open the Registry by running “regedit”:

Figure 7-1. The “Run” dialog box



2. Browse to the path: HKEY_CURRENT_USER\Software\Atmel\AVRTools\STK500\
3. Make a new String Value (right-click > New > String Value) named “LogFilePath”
4. Enter Data e.g. ”c:\STK500Com.txt” (right-click ”LogFilePath” > Modify > enter Value Data)

Figure 7-2. The “Edit String” dialog box



After editing the Registry, open AvrStudio and start the STK500 Programming Dialog. All commands will now be written to the text file specified in the Registry. This file will look like this:

Figure 7-3. Example log file contents

```
Port opened successfully
Returned status: Command succeeded

Sending packet 01/21/2005 17:17:51.046
( 200ms) > 1B 01 00 01 0E 01 14
Sequence number 1, message size 1, checksum 20
CMD_SIGN_ON

Receiving packet 01/21/2005 17:17:51.046
( 200ms) < 1B 01 00 0B 0E
( 190ms) < 01 00 08 53 54 4B 35 30 30 5F 32 02
Sequence number 1, message size 11, checksum 2
CMD_SIGN_ON
Returned status: Command succeeded

Sending packet 01/21/2005 17:17:51.056
(1000ms) > 1B 02 00 02 0E 03 9A 8C
Sequence number 2, message size 2, checksum 140
CMD_GET_PARAMETER
```

Note #1

Note #2

Notes: 1. This is the Command ID, in this case 0x03 which is CMD_GET_PARAMETER
2. This is the Parameter ID, in this case 0x9A which is PARAM_TOPCARD_DETECT

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

2591B-AVR-03/05