

In-System Programming - Serial Protocol Stack for C51 Products

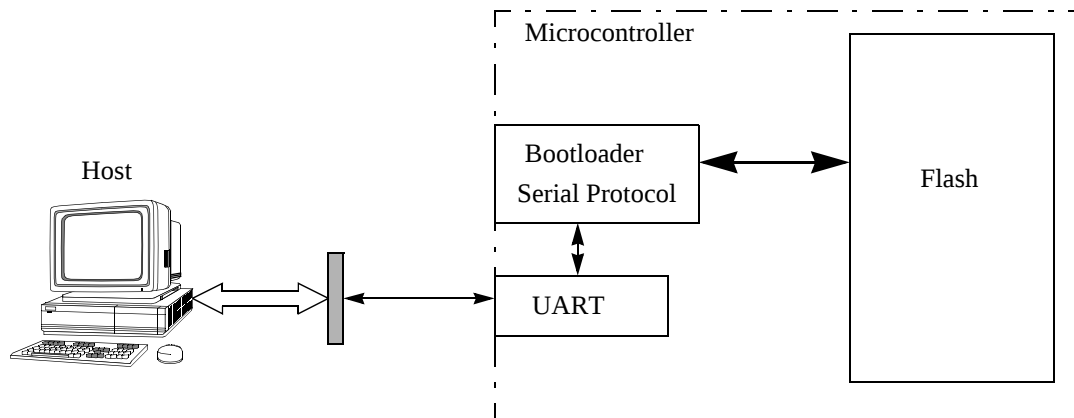
1. Introduction

1.1. Overview

This application note describes the Serial Protocol used to program the Flash code memory of microcontrollers from Atmel Wireless & Microcontrollers. Commands sent over the serial line are interpreted by the on-chip bootloader program.

This applied for T89C51RB2/C2/D2. T89C51CC01 behave same way as RC2 device.

This protocol is compatible to Philips Protocol implemented in P89CR\C+\D+\D2.



1.2. Acronyms

ISP: In-System Programming

SBV: Software Boot Vector

BSB: Boot Status Byte

SSB: Software Security Bit

HW: Hardware Byte

2. Protocol Description

2.1. Physical Layer

The UART used to transmit information has the following configuration:

- Character: 8-bit data
- Parity: none
- Stop: 1 bit
- Flow control: none
- Baudrate: autobaud is performed by the bootloader to compute the baudrate chosen by the host.

2.2. Frame Description

The Serial Protocol is based on the Intel Hex-type records.

Intel Hex records consist of ASCII characters used to represent hexadecimal values and are summarized below:

Record Mark '.'	Reclen	Load Offset	Record Type	Data or Info	Checksum
1-byte	1-byte	2-bytes	1-byte	n-bytes	1-byte

Figure 1. Intel Hex Type Frame

- **Record Mark:**
Record Mark is the start of frame. This field must contain '.'.
- **Reclen:**
Reclen specifies the number of bytes of information or data which follows the Record Type field of the record.
- **Load Offset:**
Load Offset specifies the 16-bit starting load offset of the data bytes, therefore this field is used only for Data Program Record (see Table 1).
- **Record Type:**
Record Type specifies the command type. This field is used to interpret the remaining information within the frame. The encoding for all the current record types is described in Table 1.
- **Data/Info:**
Data/Info is a variable length field. It consists of zero or more bytes encoded as pairs of hexadecimal digits. The meaning of data depends on the **Record Type**.
- **Checksum:**
The two's complement of the 8-bit bytes that result from converting each pair of ASCII hexadecimal digits to one byte of binary, and including the **Reclen** field to and including the last byte of the **Data/Info** field. Therefore, the sum of all the ASCII pairs in a record after converting to binary, from the **Reclen** field to and including the **Checksum** field, is zero.

Note:

1. A data byte is represented by two ASCII characters
2. When the field Load Offset is not used, it should be coded as four ASCII zero characters ('0').

2.3. Command Description

Table 1. Frame Description

Command	Command Name	data[0]	data[1]	Command Effect
00h	<i>Program Data</i>			Program Nb Data Byte
01h	<i>End Of File</i>	-	-	End of File
02h	<i>Specify Oscillator Frequency</i>	-	-	Not implemented
03h	<i>Write Function</i>	01h	00h	Erase block0 (0000h-1FFFh) ¹
			20h	Erase block1 (2000h-3FFFh) ¹
			40h	Erase block2 (4000h-7FFFh) ¹⁻³
			80h	Erase block3 (8000h-BFFFh) ¹⁻²⁻³
			C0h	Erase block4 (C000h-FFFFh) ¹⁻²⁻³
		04h	00h	Erase BVA & BSB
		05h	00h	Program SSB level 1
			01h	Program SSB level 2
		06h	00h	Program BSB (value to write in data[2])
			01h	Program BVA (value to write in data[2])
		07h	-	Full Chip Erase
		0Ah	02h	Program Osc fuse ¹
			04h	Program BLJB fuse ¹
			08h	Program X2 fuse ¹
04h	<i>Display Function</i>	Data[0:1] = start address Data [2:3] = end address Data[4] = 00h -> Display data Data[4] = 01h -> Blank check		Display Data
				Blank Check
05h	<i>Read Function</i>	00h	00h	Manufacturer Id
			01h	Device Id #1
			02h	Device Id #2
			03h	Device Id #3
		07h	00h	Read SSB
			01h	Read BSB
			02h	Read BVA
			03h	Read Hardware Byte ²⁻³
		08h	00h	Read Bootloader Version ²⁻³
		0Bh	00h	Read Hardware Byte ¹
		0Eh	00h	Read Device Boot ID1 ¹
			01h	Read Device Boot ID2 ¹
		0Fh	00h	Read Bootloader Version ¹
06h	<i>Direct Load of Baud Rate</i>	-	-	Not implemented

Note:

1. Not implemented on RD2
2. Not implemented on RC2
3. Not implemented on RB2

3. Functional Description

3.1. Software Security Bits (SSB)

The SSB protects any Flash access from ISP command.

The command "Program Software Security bit" can only write a higher priority level.

There are three levels of security:

- level 0: **NO_SECURITY** (FFh)
This is the default level.
From level 0, one can write level 1 or level 2.
- level 1: **WRITE_SECURITY** (FEh or 10h for RD2)
For this level it is impossible to write in the Flash memory, BSB and BVA.
The bootloader returns 'P' on write access.
From level 1, one can write only level 2.
- level 2: **RD_WR_SECURITY** (FCh or 00h for RD2)
The level 2 forbids all read and write accesses to/from the Flash memory.
The bootloader returns 'L' on read or write access.
Only a full chip erase in parallel mode (using a programmer) or ISP command can reset the software security bits.
From level 2, one cannot read and write anything.

	Level 0	Level 1	Level 2
Flash	Any access allowed	Read only access allowed	Any access not allowed
Fuse bit	Any access allowed	Read only access allowed	Any access not allowed
BSB & BVA	Any access allowed	Read only access allowed	Any access not allowed
SSB	Any access allowed	Write level 2 allowed	Read only access allowed
Manufacturer info	Read only access allowed	Read only access allowed	Read only access allowed
Bootloader info	Read only access allowed	Read only access allowed	Read only access allowed
Erase block	Allowed	Not allowed	Not allowed
Full chip erase	Allowed	Allowed	Allowed
Blank Check	Allowed	Allowed	Allowed

3.2. Full Chip Erase

The ISP command "Full Chip Erase" erases all User Flash memory (fills with FFh) and sets some bytes used by the bootloader at their default values:

- BSB = FFh
- BVA = FCh
- SSB = FFh and finally erase the Software Security Bits

The Full Chip Erase does not affect the bootloader.

4. Protocol Description

4.1. Overview

An initialization step must be performed after each Reset. After microcontroller reset, the bootloader waits for an autobaud sequence.

When the communication is initialized the protocol depends on the record type requested by the host.

4.2. Communication Initialization

The host initializes the communication by sending a 'U' character to help the bootloader to compute the baudrate (autobaud).

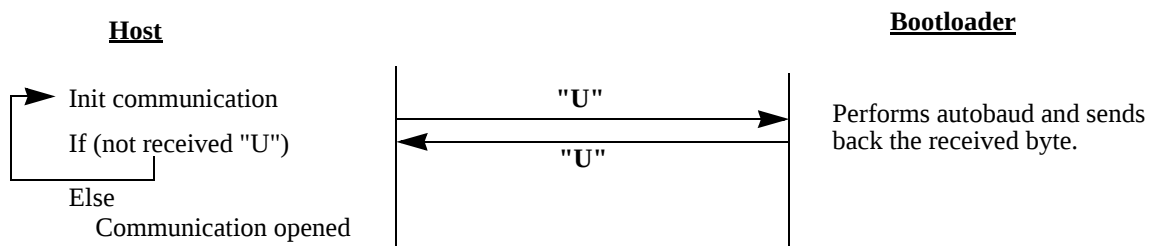


Figure 2. Initialization

4.3. Command Data Stream Protocol

All commands are sent using the same flow. Each frame sent by the host is echoed by the bootloader.

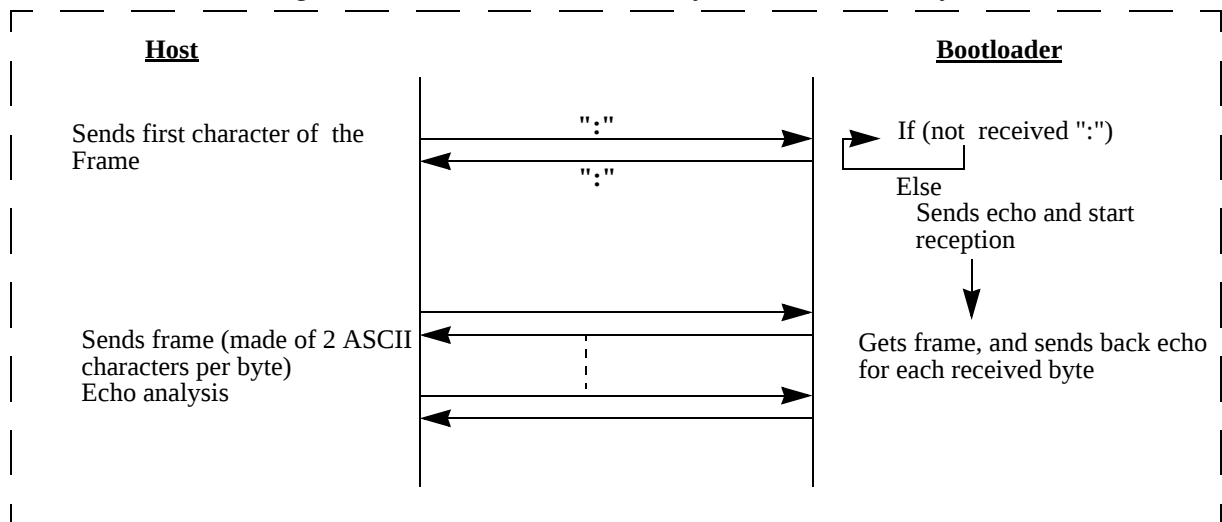


Figure 3. Command Flow

Note:

All commands sent with the echo mechanism will be represented by:

Command Name →

4.3.1. Write / Program Commands

This flow is common to the following frames:

- Flash Programming Data Frame
- EOF or Atmel Frame (only Programming Atmel Frame)
- Oscillator Frame
- Config Byte Programming Data Frame
- Baud Rate Frame

a. Description

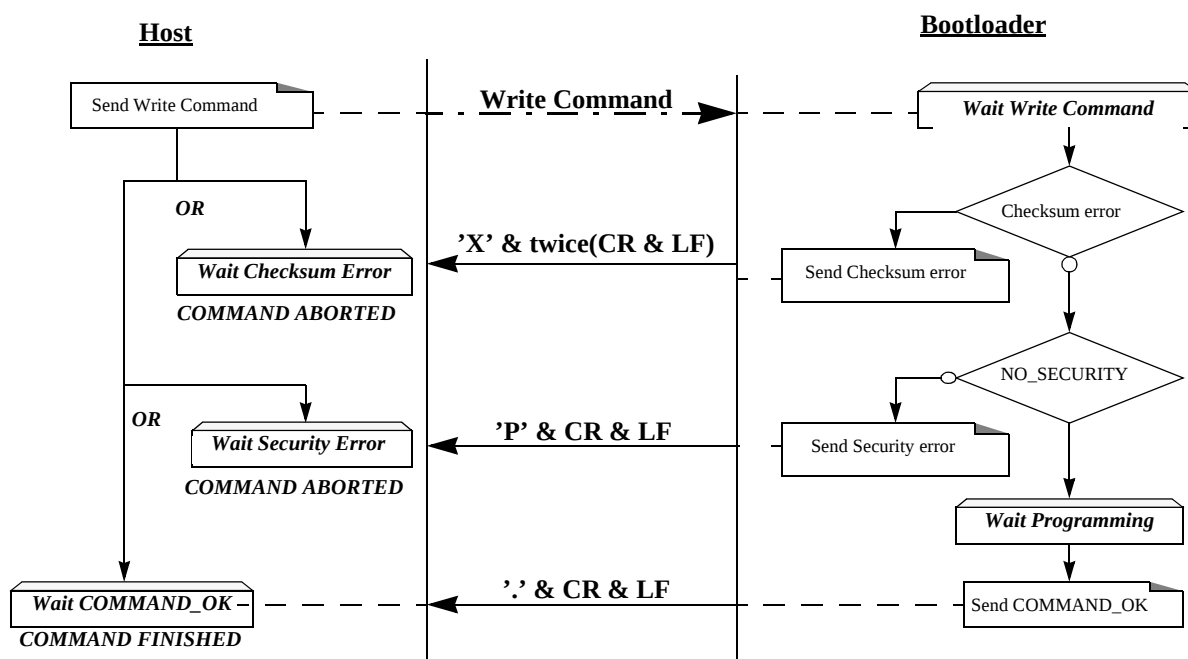


Figure 4. Write/Program Flow

b. Example

Programming Data (write 55h at address 0010h in the Flash)

HOST : 01 0010 00 55 9A

BOOTLOADER : 01 0010 00 55 9A . CR LF

Programming Atmel function (write SSB to level 2)

HOST : 02 0000 03 05 01 F5

BOOTLOADER : 02 0000 03 05 01 F5. CR LF

Oscillator Frame (osc 16 MHz)

HOST : 01 0000 02 10 ED

BOOTLOADER : 01 0000 02 10 ED . CR LF

Writing Frame (write BSB to 55h)

HOST : 03 0000 03 06 00 55 9F

BOOTLOADER : 03 0000 03 06 00 55 9F . CR LF

4.3.2. Blank Check Command

a. Description

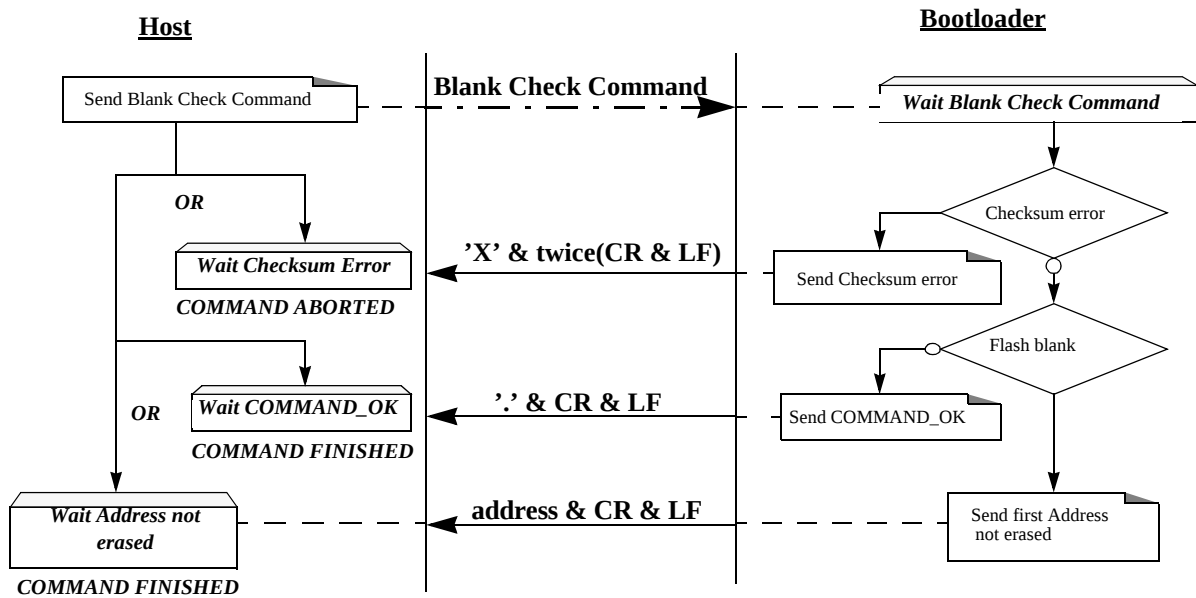


Figure 5. Blank Check Flow

b. Example

Blank Check ok

HOST : 05 0000 04 0000 7FFF 01 78
 BOOTLOADER : 05 0000 04 0000 7FFF 01 78 . CR LF

Blank Check ko at address xxxx

HOST : 05 0000 04 0000 7FFF 01 78
 BOOTLOADER : 05 0000 04 0000 7FFF 01 78 xxxx CR LF

Blank Check with checksum error

HOST : 05 0000 04 0000 7FFF 01 70
 BOOTLOADER : 05 0000 04 0000 7FFF 01 70 X CR LF CR LF

4.3.3. Display Data

a. Description

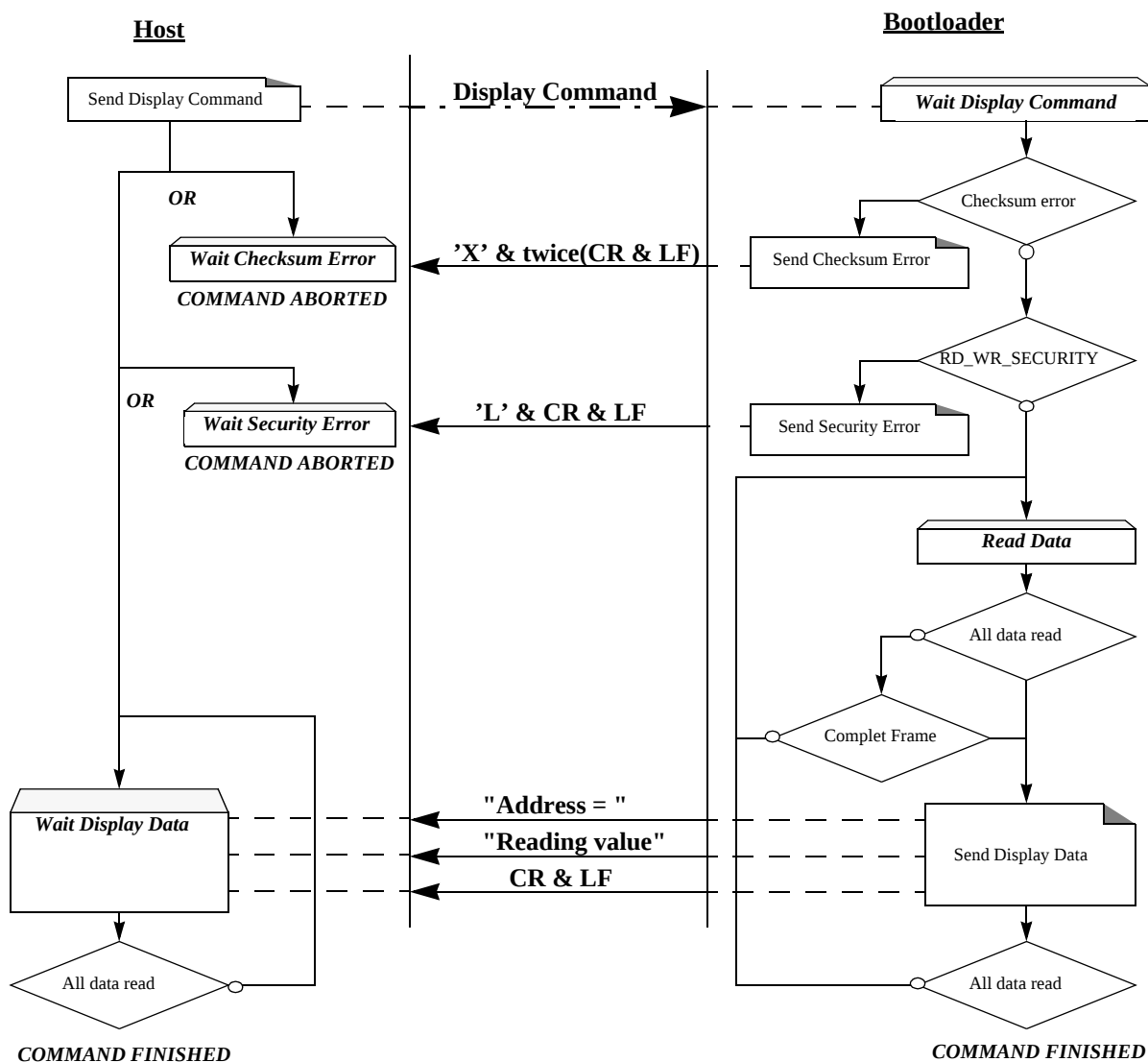


Figure 6. Display Flow

Note:

The maximum size of block is 400h. To read more than 400h bytes, the Host must send a new command.

b. Example

Display data from address 0000h to 0020h

HOST	:	05 0000 04 0000 0020 00 D7
BOOTLOADER	:	05 0000 04 0000 0020 00 D7
BOOTLOADER		0000=-----data----- CR LF (16 data)
BOOTLOADER		0010=-----data----- CR LF (16 data)
BOOTLOADER		0020=data CR LF (1 data)

4.3.4. Read Function

This flow is similar for the following frames:

- Reading Frame
- EOF Frame/ Atmel Frame (only reading Atmel Frame)

a. Description

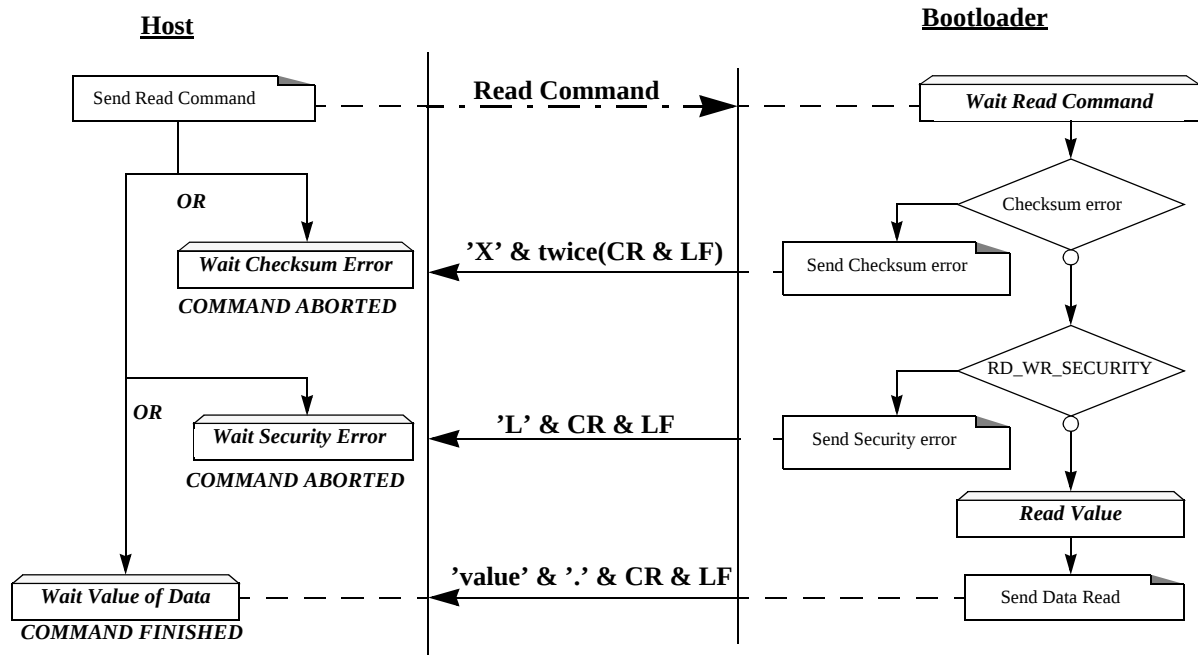


Figure 7. Read Flow

b. Example

Read function (read BVA)

HOST : 02 0000 05 07 02 F0
 BOOTLOADER : 02 0000 05 07 02 F0 Value . CR LF

Atmel Read function (read Bootloader version)

HOST : 02 0000 01 02 00 FB
 BOOTLOADER : 02 0000 01 02 00 FB Value . CR LF

5. References

5.1. Bibliography

- [1] Hexadecimal Object File Format Specification
Intel Revision A 6 January 1988
- [2] T89C51RD2 Datasheet
Atmel Wireless & Microcontrollers

6. Appendix: Protocol Validation & Diagnostic

6.1. XAF

State_test	Action	Expected Result	Comment
0	Full Chip Erase	Command ok	Erase all Flash, BSB, BVA and SSB
1	Write BSB = 0x55	Command ok	
2	Read BSB	BSB = 0x55	
3	Write BVA = 0x55	Command ok	
4	Read BVA	BVA = 0x55	
5	Write SSB = level1	Command ok	Level1 Read only for Flash, Xaf and Fuse.
6	Read SSB	SSB = level1	
7	Write BSB = 0xAA	Error Security bit - Command ko	Can not write BSB
8	Read BSB	BSB = 0x55	
9	Write BVA = 0xAA	Error Security bit - Command ko	Can not write BVA
10	Read BVA	BVA = 0x55	
11	Write SSB = level2	Command ok	Level2 Read and Write access forbidden
12	Read SSB	SSB = level2	
13	Write BSB = 0x00	Error Security bit - Command ko	Can not write BSB
14	Read BSB	Error Security bit - Command ko	Can not read BSB
15	Write BVA = 0x00	Error Security bit - Command ko	Can not write BVA
16	Read BVA	Error Security bit - Command ko	Can not read BVA
17	Write SSB = level1	Error Security bit - Command ko	Can not write SSB
18	Read SSB	SSB = level2	
19	Full Chip Erase	Command ok	Erase all Flash, BSB, BVA and SSB
20	Read BSB	BSB = 0xFF	
21	Read BVA	BVA = 0xFC	
22	Read SSB	SSB = 0xFF	

6.2. Flash

State_test	Action	Expected Result	Comment
0	Full Chip Erase	Command ok	Erase all Flash, BSB, BVA and SSB
1	Write 55h from start_add = 0000h to end_add = 7FFFh	Command ok	
2	Read from start_add = 0000h to end_add = 7FFFh	All flash = 55h	
3	Full Chip Erase	Command ok	Erase all Flash, BSB, BVA and SSB
4	Write AAh at 5555h	Command ok	
5	Read at 0x5555h	Value = AAh	
6	Blank Check	not ok at address 5555h	Return the address of the first byte not erased
7	Full Chip Erase	Command ok	Erase all Flash, BSB, BVA and SSB
8	Blank Check	Command ok	
9	Write SSB level1	Command ok	Level1 Read only for Flash, Xaf and Fuse.
10	Read SSB	SSB = level1	
11	Write AAh at 5555h	Error Security bit - Command ko	
12	Read at 5555h	Value = FFh	
13	Write SSB level2	Command ok	Level2 Read and Write access forbidden
14	Read SSB	SSB = level2	
15	Write AAh at 5555h	Error Security bit - Command ko	
16	Read at 5555h	Error Security bit - Command ko	
17	Full Chip Erase	Command ok	Erase all Flash, BSB, BVA and SSB
18	Write 55h from start_add = 0000h to end_add = 7FFFh	Command ok	
19	Read from start_add = 0000h to end_add = 7FFFh	All flash = 0x55h	
20	Erase Block0	Command ok	Erase Flash between 0000h - 1FFFh
21	Blank Check	not ok at address 2000h	Return the address of the first byte not erased
22	Erase Block1	Command ok	Erase Flash between 2000h - 3FFFh
23	Blank Check	not ok at address 4000h	Return the address of the first byte not erased
24	Erase Block2	Command ok	Erase Flash between 4000h - 7FFFh
25	Blank Check	Command ok	

6.3. Hardware Byte

State_test	Action	Expected Result	Comment
0	Full Chip Erase	Command ok	Erase all Flash, BSB, BVA and SSB
1	Write SSB level2	Command ok	Level2 Read and Write access forbidden
2	Read SSB	SSB = level2	
3	Read Hardware Byte	Error Security bit - Command ko	
4	Full Chip Erase	Command ok	Erase all Flash, BSB, BVA and SSB
5	Read Hardware Byte	HW = xxh	HW readable

6.4. Specific Informations

State_test	Action	Expected Result	Comment
0	Full Chip Erase	Command ok	Erase all Flash, BSB, BVA and SSB
1	Read Bootloader Version	Command ok	
2	Read Device boot ID1	Command ok	
3	Read Device boot ID2	Command ok	
4	Read Manufacturer Code	Command ok	
5	Read Device id 1	Command ok	
6	Read Device id 2	Command ok	
7	Read Device id 3	Command ok	
8	Write SSB level2	Command ok	Level2 Read and Write access forbidden
9	Read SSB	SSB = level2	
10	Read Bootloader Version	Command ok	
11	Read Device boot ID1	Command ok	
12	Read Device boot ID2	Command ok	
13	Read Manufacturer Code	Command ok	
14	Read Device id 1	Command ok	
15	Read Device id 2	Command ok	
16	Read Device id 3	Command ok	
17	Full Chip Erase	Command ok	Erase all Flash, BSB, BVA and SSB

6.5. Checksum control

All commands defined in Table 1 must be sent with a wrong checksum value to validate the 'X' returned by bootloader.