

---

# How to Emulate Read/Write Code Access To Atmel T89C51xx Using Hitex DProbeHS Emulator

Atmel C51 In-System Programmable (ISP) Flash Microcontrollers embed a bootloader memory sector that enables users to perform self-programming in the Application Flash Memory area.

This application note explains emulation of the (read/write) code access to Atmel T89C51xx microcontrollers using the Hitex DProbeHS Emulator.

This application note applies to the following products: T89C5115, T89C51CC02, T89C51AC2, T89C51CC01, T89C51RB2, T89C51RC2, T89C51IC2, T89C51RD2.



---

**C51 Flash  
Microcontrollers**

---

**Application Note**

Rev. 4124A-8051-05/02

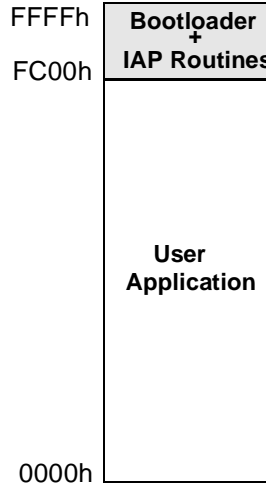


# T89C51xx Memory Mapping

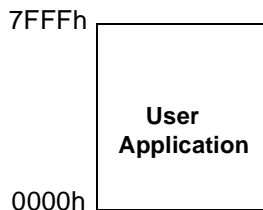
Using the T89C51xx microcontroller, the bootloader program can be either located in a separate area (ROM or Flash), or in the user area.

The following diagrams provide a memory mapping description for the T89C51xx.

**Figure 1.** T89C51RD2 Memory Mapping



**Figure 2.** Other Products Memory Mapping

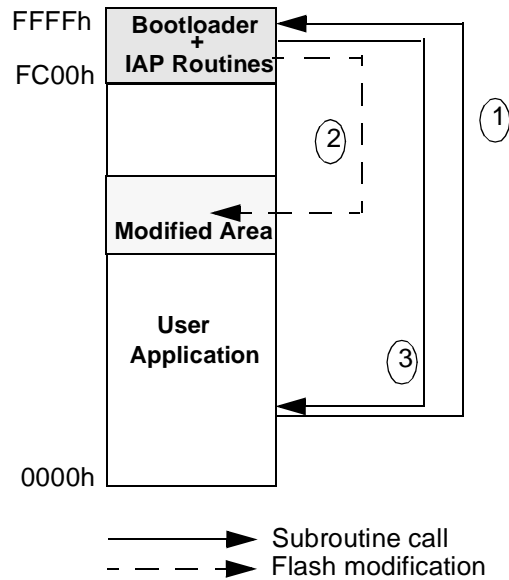


## Flash Access Mechanism Using IAP Routines

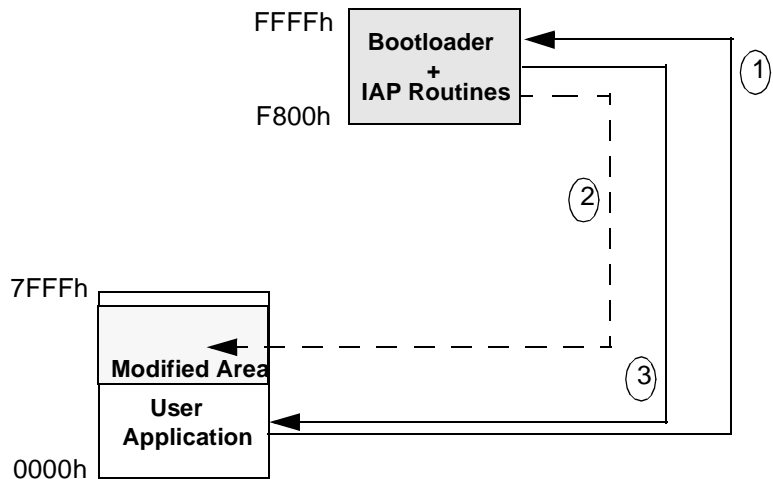
Users can easily call IAP routines in the bootloader programs directly from their application code.

The following diagram describes how to rewrite on the Flash memory.

**Figure 3.** T89C51RD2 Flash Rewrite Description



**Figure 4.** Other Device Flash Rewrite Description



To perform On-chip Flash modification, proceed as follows:

1. Call the IAP routine from the user program.
2. The IAP routine will modify the specified address with a user defined value.
3. The IAP routine ends and program returns to the user program.

## On-chip Flash Emulation

### Description

The appropriate Hitex emulator for C51 Flash ISP Microcontrollers, is the DProbeHS-TEH.

The Hitex emulator provides the capability to emulate, write, and read operations from the on-chip Flash.

Users use the IAP routines provided by Atmel in their own code. The Hitex emulator will be able to properly emulate the program behavior when write or read operations in the on-chip Flash occur.

Write access occurs in the IAP using MOVX @DPTR, A and read access using MOVC A, @A+DPTR.

#### Caution:

The T89C51RD2 does not yet provide mechanisms for emulation of internal write Flash access, it is mandatory to use a T89C51RC2/RB2/IC2 to emulate these On-Chip Flash access of T89C51RD2.

### Setup Procedure

The following steps indicate the procedure to emulate read and write access in the on-chip Flash:

1. Identify whether the bootloader software is located in Flash or in ROM for the dedicated Atmel Flash C51 that is used.
2. If the bootloader program is located in Flash (T89C51RD2, T89C51CC01/CC02, T89C51AC2, T89C5115), download the latest software release from the Atmel website [www.atmel.com](http://www.atmel.com).  
If the bootloader program is located in ROM (T89C51RC2/RB2/IC2), the bootloader program is joined to this application note.
3. Run the Hitex emulator as specified in the user manual with or without target board.
4. Define the memory mapping of the Atmel Flash C51 product (16/32/64K Bytes).
5. If the product has less than 64K bytes defined bootloader memory mapping, mapping will depend of the size of the bootloader program:

Start address of the bootloader program will be the upper address of the 64K address space when taking into account its size.

1K Byte size: Start address: FC00h

2K Bytes size: Start address: F800h

4K Bytes size: Start address: F000h

etc....

6. Load the bootloader program in Intel HEX format in the emulation code memory.
7. Load the user application program htx file and its associated symbol file.
8. Emulator is now ready to properly emulate write and read accesses to the On-Chip Flash memory.

## Example

User code is implemented in a T89C51RC2 product featuring 16K Bytes Flash and 2K Bytes bootloader program in ROM.

This example writes and reads AAh value from address 3F00h to 3F0Fh in the user code memory.

1. Run the Hitex emulator as specified in the user manual with or without target board.
2. This example will use the T89C51RC2/RB2/IC2 bootloader program joined to this application note.
3. Bootloader size is 2K Bytes, so it will be located at F800h address. Mapping for the user code and bootloader program will be the following:

Figure 5. User Memory Mapping

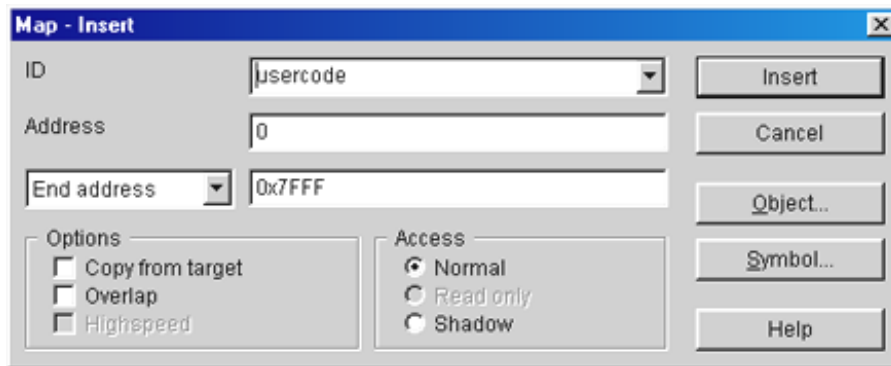
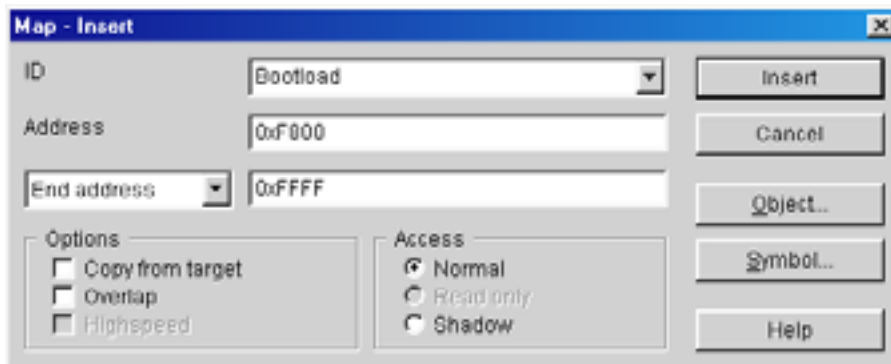
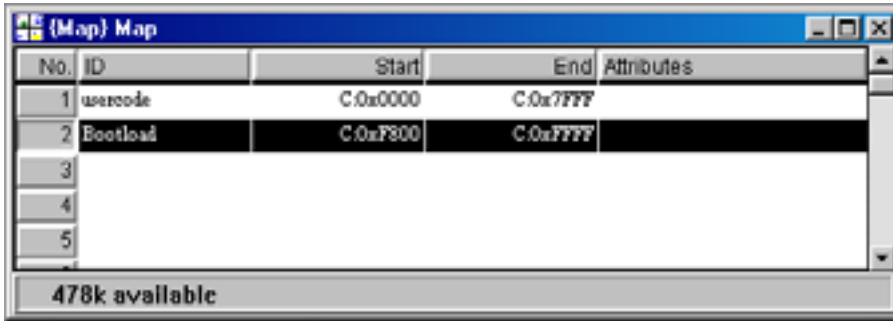


Figure 6. Bootloader Memory Mapping



**Figure 7.** Final Complete Memory Mapping

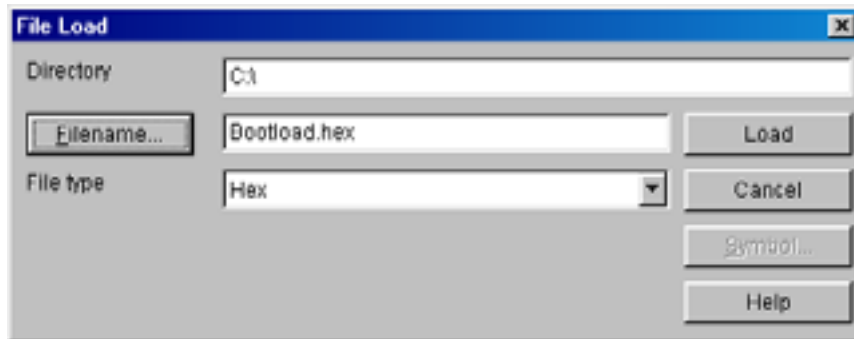


| No. | ID       | Start    | End      | Attributes |
|-----|----------|----------|----------|------------|
| 1   | usercode | C:0x0000 | C:0x7FFF |            |
| 2   | Bootload | C:0xF800 | C:0x7FFF |            |
| 3   |          |          |          |            |
| 4   |          |          |          |            |
| 5   |          |          |          |            |

478k available

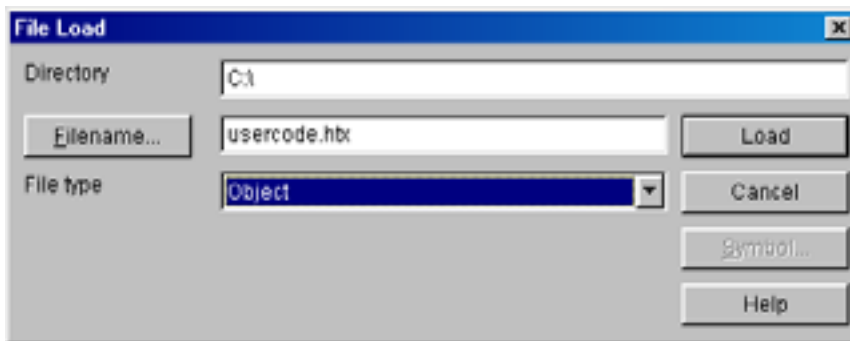
4. Load the bootloader program in Intel HEX format.

**Figure 8.** Load Bootloader Program



5. Load the user application program htx file and its associated symbol file.

**Figure 9.** Load Bootloader Program



6. Emulator is now ready to properly emulate write and read access to the On-chip Flash memory.

## Trace and Trigger Conditions

### Trace Without Trigger Condition

Since write access to On-chip Flash is based on MOVX instruction, trace can be performed on the On-chip Flash access.

Trace buffer with no specific trigger condition for Code Write access will be the following for this dedicated code:

Figure 10. Source Code With Flash Code Write Access

```

[List1] List [%pc]
BP PC CC Line # Source
// Read code address 0x3F00
153 printf("read code at 0x3F00 : ");
154 value = __api_rd_code_byte(0x3F00);
155 send_byte_ascii(value);
156 send_cr_lf();

// Write code page between address 0x3F00-3F7F to 0xFF
159 for (cpt=0; cpt<128; cpt++)
160 {
161     tab[cpt] = 0xFF;
162     if (cpt==255) break;
163 }
164 printf("Write code page between 0x3F00-3F7F to 0xFF \n");

166 send_cr_lf();

Source file: USERCODE.C Address: C:0x03BE Total Lines: 303
    
```

Figure 11. Trace Buffer

| Frame # | Address  | State | Data   | Instruction    | External | ss mmm upp nn  |
|---------|----------|-------|--------|----------------|----------|----------------|
| -124    | C:0xFC5D |       | 22     | RET            | ff       | 00 000 004, 35 |
| -120    | C:0xFC81 |       | 53D1F7 | ANL OD1, #0F7  | ff       | 00 000 002, 16 |
| -116    | C:0xFC84 |       | A3     | INC DPTR       | ff       | 00 000 002, 16 |
| -112    | C:0xFC85 |       | DEF2   | DJNZ R6, 0FC79 | ff       | 00 000 002, 16 |
| -108    | C:0xFC79 |       | 05A2   | INC AUXR1      | ff       | 00 000 002, 18 |
| -106    | C:0xFC7B |       | ED     | MOVX A, @DPTR  | ff       | 00 000 001, 08 |
| -104    | C:0xFC7C |       | A3     | INC DPTR       | ff       | 00 000 002, 17 |
| -100    | C:0xFC7D |       | 05A2   | INC AUXR1      | ff       | 00 000 002, 16 |
| -98     | C:0xFC7F |       | 9159   | ACALL 0FC59    | ff       | 00 000 002, 16 |
| -94     | C:0xFC59 |       | 43D108 | ORL OD1, #8    | ff       | 00 000 002, 16 |
| -90     | C:0xFC5C |       | FD     | MOVX @DPTR, A  | ff       | 00 000 002, 16 |
| -88     | C:0x3F0F | CU    | AA     |                | ff       |                |
| -87     | C:0xFC5D |       | 22     | RET            | ff       | 00 000 004, 35 |
| -83     | C:0xFC81 |       | 53D1F7 | ANL OD1, #0F7  | ff       | 00 000 002, 16 |
| -79     | C:0xFC84 |       | A3     | INC DPTR       | ff       | 00 000 002, 16 |

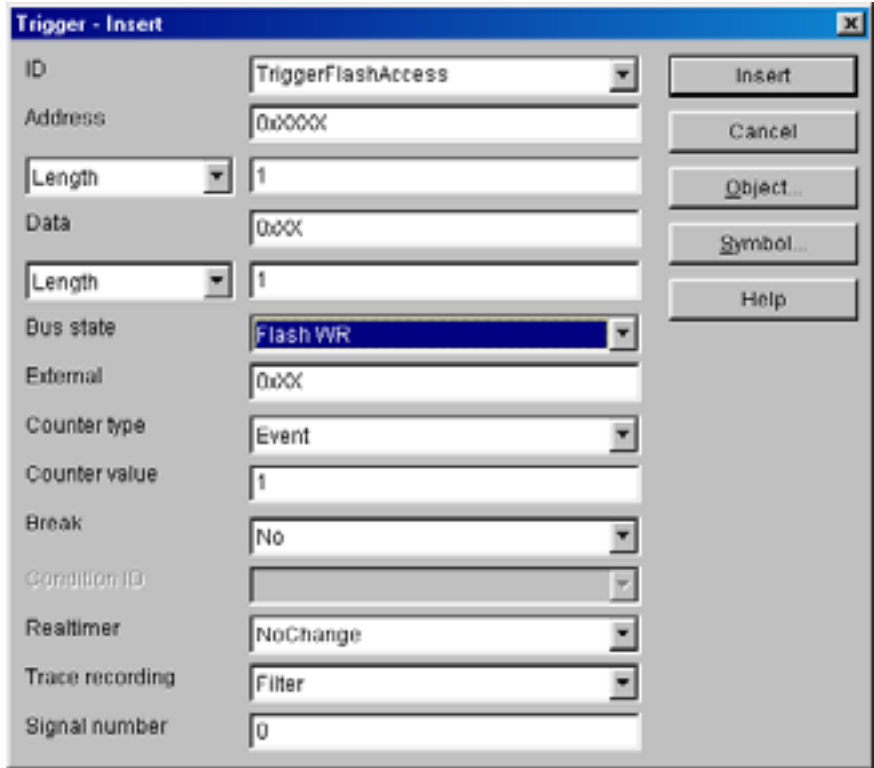
Scope: ""

Code Write access to the Flash is mentioned in the trace buffer by the keyword “CW” (for Code Write) in the state column.

**Trace Trigger Condition for Code Write Access**

To be able to trace only Code write to check a memory array, a specific trigger can be defined:

**Figure 12.** Trigger Definition for Code Write Access



Trace buffer with this trigger condition for Code Write access will be the following:



Figure 13. Trace With Only Flash Code Write Access

| Frame # | Address  | State | Data | Instruction | External | Data          |
|---------|----------|-------|------|-------------|----------|---------------|
| -112    | C:0x3F00 | CW    | AA   |             | ff       | ss mmm ppp,nn |
| -111    | C:0x3F00 | CW    | AA   |             | ff       |               |
| -110    | C:0x3F01 | CW    | AA   |             | ff       |               |
| -109    | C:0x3F02 | CW    | AA   |             | ff       |               |
| -108    | C:0x3F03 | CW    | AA   |             | ff       |               |
| -107    | C:0x3F04 | CW    | AA   |             | ff       |               |
| -106    | C:0x3F05 | CW    | AA   |             | ff       |               |
| -105    | C:0x3F06 | CW    | AA   |             | ff       |               |
| -104    | C:0x3F07 | CW    | AA   |             | ff       |               |
| -103    | C:0x3F08 | CW    | AA   |             | ff       |               |
| -102    | C:0x3F09 | CW    | AA   |             | ff       |               |
| -101    | C:0x3F0A | CW    | AA   |             | ff       |               |
| -100    | C:0x3F0B | CW    | AA   |             | ff       |               |
| -99     | C:0x3F0C | CW    | AA   |             | ff       |               |
| -98     | C:0x3F0D | CW    | AA   |             | ff       |               |

Scope: "???"

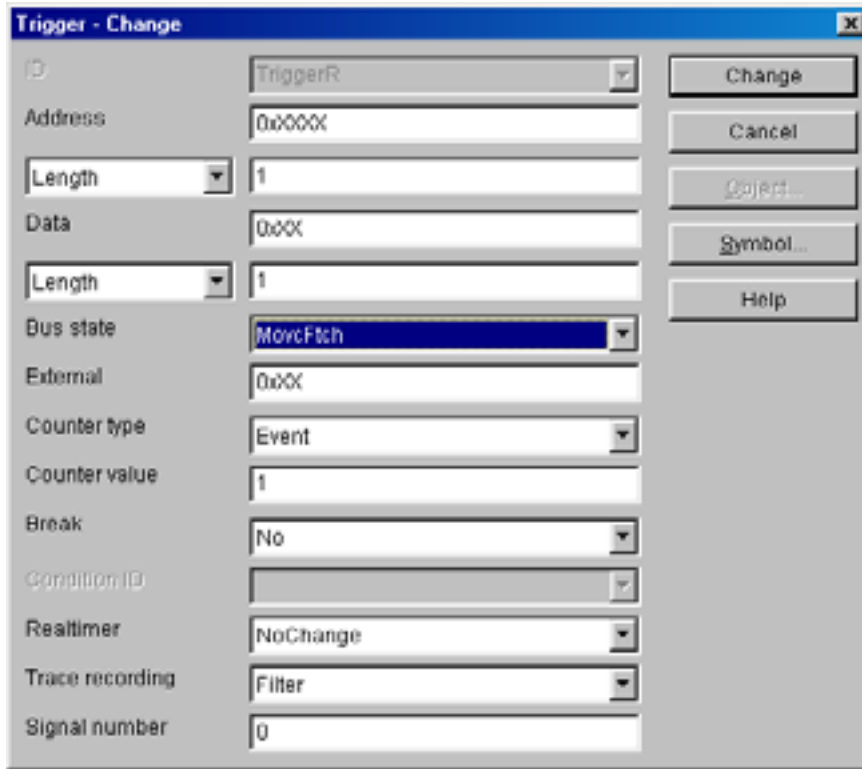
### Trace Trigger Condition For Code Read Access

Flash read access is done using MOV C A, @A+DPTR. Thus trigger and trace are done using standard trigger definition for MOV C instruction.

To be able to trace only Code read access to check a memory array a specific trigger can be defined:

- Choose MovcFtch in the Bus State selection to be able to trace code access
- Choose Filter in the Trace recording selection to trace only Code read access

**Figure 14.** Trigger Definition for Code Read Access



Trace buffer with this trigger condition for Code Read access will be the following:

Figure 15. Trace With Only Flash Code Read Access

| Frame # | Address  | State | Data | Instruction | External | ss mmm μμμ,nn |
|---------|----------|-------|------|-------------|----------|---------------|
| -128    | C:0x0538 | F     | 00   |             | ff       |               |
| -127    | C:0x3F00 | F     | AA   |             | ff       |               |
| -126    | C:0x3F01 | F     | AA   |             | ff       |               |
| -125    | C:0x3F02 | F     | AA   |             | ff       |               |
| -124    | C:0x3F03 | F     | AA   |             | ff       |               |
| -123    | C:0x3F04 | F     | AA   |             | ff       |               |
| -122    | C:0x3F05 | F     | AA   |             | ff       |               |
| -121    | C:0x3F06 | F     | AA   |             | ff       |               |
| -120    | C:0x3F07 | F     | AA   |             | ff       |               |
| -119    | C:0x3F08 | F     | AA   |             | ff       |               |
| -118    | C:0x3F09 | F     | AA   |             | ff       |               |
| -117    | C:0x3F0A | F     | AA   |             | ff       |               |
| -116    | C:0x3F0B | F     | AA   |             | ff       |               |
| -115    | C:0x3F0C | F     | AA   |             | ff       |               |
| -114    | C:0x3F0D | F     | AA   |             | ff       |               |
| -113    | C:0x3F0E | F     | AA   |             | ff       |               |
| -112    | C:0x3F0F | F     | AA   |             | ff       |               |
| -111    | C:0x3F10 | F     | 2F   |             | ff       |               |

Scope: ""

Code Read access to the Flash is mentioned in the trace buffer by the keyword “F” (for Fetch) in the state column. Data read

## Code Memory View

Once the program is stopped or when a breakpoint occurs, code memory is visualized using the View Memory menu:

| Address  | Hex   | ASCII            |
|----------|---|------------------|
| C:0x3F00 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |                  |
| C:0x3F10 | 04 88 E1 79 2F 4C 08 64 E7 FA F8 A0 03 A2 38    | /...pL.d.....8   |
| C:0x3F20 | 5 80 56 77 50 51 F9 86 08 49 80 08 AF 5C 5B 10  | 5..u10...L...N.  |
| C:0x3F30 | 00 01 01 00 00 00 00 00 00 00 00 00 00 00 00    | .....            |
| C:0x3F40 | 77 86 C2 3E 7C 7E 10 6C EA C9 52 0C 50 80 70 08 | u...i".L.R.P.).  |
| C:0x3F50 | 04 2E 16 0C F7 1E 40 E5 EE 07 40 37 60 24 03 A9 | .....H...H7g..   |
| C:0x3F60 | 30 00 03 0E 00 51 C0 90 F0 50 75 4C 66 20 F9 1E | 0...0...buL+..   |
| C:0x3F70 | 10 41 C0 A5 5E 50 50 00 50 4C 40 F5 90 62 7C 22 | .A..XP.ILK..b1"  |
| C:0x3F80 | 85 40 EF 08 27 8F E5 95 87 C2 07 57 09 0E FD 85 | .8.."......H.... |



## References

Product datasheets

In-System Programming: Flash API Rev 1.0.0 for T89C51RB2/RC2

## Abbreviations

ISP: In-System Programming

Bootloader: Atmel dedicated program located inside the T89C51xx to perform In-System Programming

HSB: Hardware configuration byte

IAP: In-Application Programming



## Atmel Headquarters

### Corporate Headquarters

2325 Orchard Parkway  
San Jose, CA 95131  
TEL 1(408) 441-0311  
FAX 1(408) 487-2600

### 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 Tsimhatsui  
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  
TEL 1(408) 441-0311  
FAX 1(408) 436-4314

### Microcontrollers

2325 Orchard Parkway  
San Jose, CA 95131  
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  
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  
TEL 1(719) 576-3300  
FAX 1(719) 540-1759

### Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Data- com

Avenue de Rochepleine  
BP 123  
38521 Saint-Egreve Cedex, France  
TEL (33) 4-76-58-30-00  
FAX (33) 4-76-58-34-80

---

### e-mail

[literature@atmel.com](mailto:literature@atmel.com)

### Web Site

<http://www.atmel.com>

### © Atmel Corporation 2002.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

ATMEL® is a registered trademark of Atmel.

Other terms and product names may be the trademarks of others.



Printed on recycled paper.