

DT-HIQ

AVR BOOTLOADER V1.0

USER MANUAL

Trademarks & Copyrights

Windows is a registered trademark of Microsoft Corporation. Pentium is a registered trademark of Intel Corporation. AVR is a registered trademark of Atmel Corporation. CodeVisionAVR is copyright by Pavel Haiduc, HP InfoTech s.r.l. BASCOM-AVR is copyright by MCS Electronics. AVR Bootloader is copyright by Innovative Electronics.

Table of Contents

1.	Introd	luction	3
	1.1.	AVR Bootloader Specifications	3
	1.2.	Supported Modules	3
	1.3.	System Requirements	3
2.	AVR	Bootloader	3
	2.1.	Running AVR Bootloader	3
	2.2.	Program Interface	4
	2.3.	Menu and Shortcuts	4
		2.3.1. File	4
		2.3.2. Program	4
		2.3.3. Read	5
		2.3.4. Compare	5
		2.3.5. Help	6
	2.4.	Operation Flow Panel	6
3.	Confi	guration Examples for CodeVisionAVR	7
4.	Confi	guration Examples for BASCOM-AVR	9

1. Introduction

AVR Bootloader v1.0 is a software from Innovative Electronics which supports microcontroller programming through bootloader. AVR Bootloader v1.0 is used to program microcontrollers by utilizing the self-programming features on DT-AVR Bootloader Micro System modules. This Windows[®] based software provides a simple and easy to use interface and can be combined with the CodeVisionAVR[®], BASCOM-AVR[®], or other AVR IDE microcontrollers.

1.1. AVR BOOTLOADER SPECIFICATIONS

The AVR Bootloader v1.0 specifications are as follows:

- UART interface with a speed of 115200 bps.
- Supports Flash, EEPROM, Lock Bit, and Fuse Bit Programming.
- Supports Intel HEX or EEP format (for EEPROM).
- Compatible with Windows[®] XP or Vista.

1.2. SUPPORTED MODULES

At this moment AVR Bootloader v1.0 supports programming of the following DT-AVR Micro System Bootloaders:

- DT-AVR ATMEGA128L Micro System Bootloader.
- DT-AVR ATMEGA168 Micro System Bootloader.

1.3. SISTEM REQUIREMENTS

The minimum requirements are:

- Pentium[®] processor or above.
- 32 MB RAM.
- 1 MB of hard disk free space.
- CD-ROM/DVD-ROM drive.
- COM Port with UART RS-232 interface with a speed of 115200 bps or USB (Virtual COM Port).
- Windows[®] XP or Vista.

2. AVR BOOTLOADER

This section will explain about the included softwares along with programming feature options.

2.1. RUNNING AVR BOOTLOADER

To run the AVR Bootloader v1.0, perform one of these following steps:

- Double click the **AVR Bootloader v1.0**.exe icon on the Windows Explorer.
- Right click on the AVR Bootloader v1.0 .exe file and select "Open".

2.2. PROGRAM INTERFACE

File Program Read Co	copyright IE 2008 mpare Help			
Chip CO ATmega128 💌 CC	M Port IM1 Auto Detect	Device Reset Chip		
Operations Flow	Programmed Section	Hex File		
Check Signature	C FLASH and EEPROM	FLASH		
🔽 Erase	FLASH	EEPROM		
🔽 Blank Check	C EEPROM	testapp128.eep		
✓ Program	Software Lock Bits			
Verify	SLB = 11 No protection	1		
Program Lock Bits	SLB = 10 Further progra	amming disabled		
Run	C SLB = 00 Programming	and Verification disabled		
Close after Run				

Figure 1 AVR Bootloader GUI

2.3. MENU AND SHORTCUT

2.3.1. FILE

2.3.2.

1.	Load <u>F</u> LASH Function Shortcut Button	: Opens .hex file for Flash memory. : Ctrl + F
2.	Load EEPROM Function Shortcut Button	: Opens .hex or .eep file for EEPROM memory. : Ctrl + M
3.	<u>R</u> ecent Files Function	: Displays 10 of the last files opened.
4.	E<u>x</u>it Function Shortcut Button	: Exits the software. : Ctrl + X
PR	OGRAM	
1.	Erase Function Shortcut Button	: Erases target IC. : Ctrl + E

2.	<u>B</u> lank Check	
	Function	: Checks whether the target IC is blank.
	Shortcut Button	: Ctrl + B

	3.	<u>F</u> LASH	
		Function	: Writes codes stored in the Hex File - Flash to the Flash target memory.
		Shortcut Button	: Ctrl + P
	4.	<u>E</u> EPROM	
		Function	: Writes codes stored in the Hex File - EEPROM to the EEPROM target memory.
		Shortcut Button	: Ctrl + O
	5.	<u>Lock Bits</u> Function Shortcut Button	: Opens a dialog box for Lock Bits configuration. : Ctrl + L
	4	A 11	
	0.	Function	: Runs a series of instruction according to the option in Operation Flow.
		Shortcut Button	: F9
2.3.3.	RE	AD	
	1.	<u>F</u>LASH Function	: Reads the target Flash IC memory and saves it into a file.
		Shortcut Button	: Ctrl + R
	2	EEDDOM	

2.	<u>E</u> EPROM	
	Function	: Reads the target EEPROM IC memory and saves it into a file.
	Shortcut Button	: Ctrl + Q
3.	<u>C</u> hip Signature Function Shortcut Button	: Reads the target IC signature chip code. : Ctrl + C
4.	Lock Bits Function Shortcut Button	: Reads the target IC Lock Bits configuration. : Ctrl + T
5	Fuco Rite	

5.	<u>F</u> use Bits	
	Function Shortcut Button	: Reads the target IC Fuse Bits configuration. : Ctrl + U
6	Bootloader Version	

6. <u>B</u>ootloader Version

Function	: Reads the bootloader version programmed inside the target
	IC.
Shortcut Button	: Ctrl + K

2.3.4. COMPARE

1.	<u>F</u> LASH	
	E	

•••		
	Function	: Compares the contents of the Hex File - Flash and target IC Flash memory.
	Shortcut Button	: Ctrl + V
2.	<u>E</u> EPROM	
	Function	: Compares the contents of the Hex File - EEPROM and target IC EEPROM memory.
	Shortcut Button	: Ctrl + G

2.3.5. HELP

• <u>A</u>bout

Function

: Displays the AVR Bootloader software version and link to the Innovative Electronic website.

2.4. OPERATION FLOW PANEL

Operation Flow Panel is used to determine what commands that will be automatically performed during the programming when using the **Program - All** menu or the "**Run**" button. The available commands are Check Signature, Erase, Blank Check, Program, Verify, and Program Lock Bits.

For Erase command, if **Preserve EEPROM on Chip Erase** is checked/selected, then only the Flash memory will be erased when the command is performed. Otherwise, if **Preserve EEPROM on Chip Erase** is not checked/selected, then both the Flash memory and the EEPROM memory will be erased when the command is performed.

For Program and Verify command, if "FLASH and EEPROM" is selected on the **Programmed Section** panel then Program and Verify command will be performed on both memories. Otherwise if only one of them is selected (FLASH or EEPROM), then the Program and Verify command will only be performed in the selected memory.

Before selecting Program and Verify, make sure that the file that will be processed have been decided beforehand. The file, that will be written to Flash or EEPROM, will be displayed on the **Hex File** panel.

For Program Lock Bits command, the safety level for flash memory is determined in the options in **Software Lock Bits** panel. If the chosen safety level is "No Protection" then adding or reading the Flash memory code is still possible without performing Erase command.

On the the next security level, adding code to the Flash memory can no longer be performed, unless you perform Erase command beforehand.

On the highest level, adding and reading Flash memory reading can no longer be performed, unless you perform Erase command beforehand.

After performing all commands in the Operation Flow, by selecting **Program - All** menu or clicking the "**Run**" button, the AVR Bootloader v1.0 will automatically perform **Reset Chip** so that the DT-AVR Bootloader Micro System exits bootloader and run the programmed application.

However, if the programming options are done manually (by pressing **Auto Detect Device** button, selecting **Read - Chip Signature**, and so on), then DT-AVR Bootloader Micro System will still be in bootloader mode. And the **Reset Chip** button must be pressed so that DT-AVR Bootloader Micro System module exits bootloader and performs the programmed application commands.

Check/select **Close after Run** if you want AVR Bootloader v1.0 to automatically close after performing all commands in Operation Flow. This function is useful when AVR Bootloader v1.0 is combined with other IDE AVR such as CodeVisionAVR or BASCOM-AVR.

3. CONFIGURATION EXAMPLES FOR CODEVISIONAVR

Combining the AVR Bootloader v1.0 with CodeVisionAVR is done by setting AVR Bootloader v 1.0 to open automatically after compiling. This setting can be done via the menu **Project** \rightarrow **Configure**.



Figure 2 Project Configuration Menu

Select the "After Make" tab and then check/select "Execute User's Program". After that press the Program Settings button to bring up the "User Program Settings" dialog window as shown in Figure 3.

Set the "**Program Directory and Filename**" option to indicate the location of theAVR Bootloader v1.0 Software. After that press the "**OK**" button.

Configure Project testing128.prj	
Files C Compiler Before Make After Make	
Program the Chip Execute User's Program Program Settings	nfoT
💦 User Program Settings	×
Program Directory and FileName: D:\Project\AVR BOOT\AVR Bootloader V1.0.exe Command Line Parameters:	C
Working Directory:	
<u>✓</u> <u>D</u> K <u>X</u> <u>C</u> ancel	
	* * * *

Figure 3 User Program Settings Dialog window

i Information		
Compiler Assembler Programmer		
Chip: ATmega128 Program type: Application Memory model: Small Optimize for: Size (s)printf features: int, width (s)scanf features: int, width Promote char to int: No char is unsigned: Yes 8 bit enums: Yes Enhanced core instructions: On Automatic register allocation: On		
2297 line(s) compiled No errors No warnings		
Bit variables size: 0 byte(s)		
Data Stack area: 100h to 4FFh Data Stack size: 1024 byte(s) Estimated Data Stack usage: 28 byte(s)		
Global variables size: 0 byte(s)		
Hardware Stack area: 500h to 10FFh Hardware Stack size: 3072 byte(s)		
Heap size: 0 byte(s)		
EEPROM usage: 0 byte(s) (0.0% of EEPROM) Program size: 721 words (1.1% of FLASH)		
Execute User's Progra	🗙 <u>C</u> ancel	

Figure 4 Information Dialog Window After A Successful Make Process

After performing these steps, if **Compile and Make** process (e.g. by pressing the **shift** + **F9** key) is succeeded, an information dialog window will appear as shown in Figure 4.

If the **Make** process is succeeded and the dialog window shown on Figure 4 appears, simply hit the Enter key on the keyboard or press the "**Execute User's Program**" button in the information dialog window to run the AVR Bootloader v1.0. After AVR Bootloader v1.0 is open, you just need to hit Enter key on the keyboard or press the "**Run**" button to start programming.

Don't forget to check the option **"Close after Run"** in order to close AVR Bootloader v1.0 automatically after the rest of programming process is complete.

4. CONFIGURATION EXAMPLES FOR BASCOM-AVR

Combining the AVR Bootloader v1.0 with BASCOM-AVR is done by setting AVR Bootloader v 1.0 to open automatically after compiling. This can be done by setting the programmer type via menu **Options** \rightarrow **Programmer**.

BASCOM-AVR Options
Compiler Communication Environment Simulator Programmer Monitor Printer
Programmer External programmer
Erase warning Auto Flash AutoVerify Upload Code and Data rogram after compile Set focus to terminal emulator after programming Other
Program "D:\Project\AVR BOOT\AVR Bootloader V1.0.exe"
Default Dk Cancel

Figure 5 Programmer Configurations Menu

Set the programmer type to "**External programmer**" and check the "**Program after compile**" option in order to open AVR Bootloader v1.0 automatically after successful compilation.

Also set the file location to show the location of AVR Bootloader v 1.0 software. In the example in figure 5, the location is in "D:\Project\AVR BOOT\AVR BOOT\AVR Bootloader V1.0.exe"

After performing these steps, if **Compile and Make** process (e.g. by pressing **F7**) is succeeded, AVR Bootloader v1.0 will run automatically. After the AVR Bootloader v1.0 opens, simply hit Enter key on the keyboard or press the **"Run"** button to start programming.