

StartUSB for AVR™

Manual

All MikroElektronika's development systems represent irreplaceable tools for programming and developing microcontroller-based devices. Carefully chosen components and the use of machines of the last generation for mounting and testing thereof are the best guarantee of high reliability of our devices. Due to simple design, a large number of add-on modules and ready to use examples, all our users, regardless of their experience, have the possibility to develop their projects in a fast and efficient way.

Development System



SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD ...making it simple

StartUSB for AVR

The StartUSB for AVR is a miniature development system that enables you to experiment with the AT90USB162 microcontroller.

Key features:

- Bootloader;
- Proto board;
- USB support;
- 5V power supply voltage.

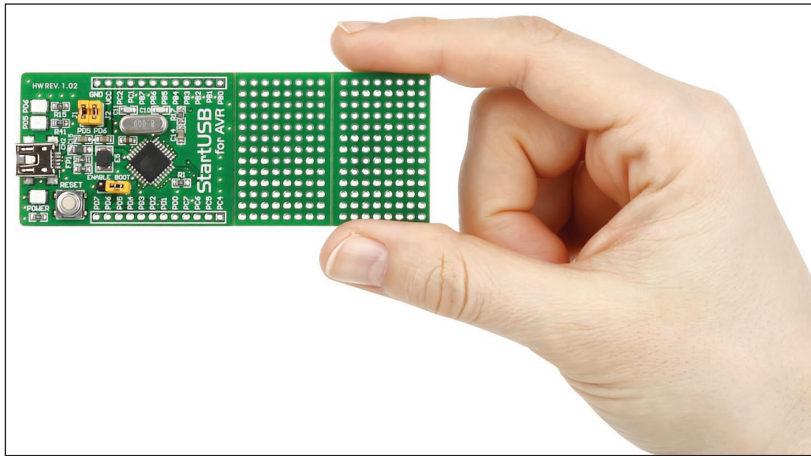


Figure 1: StartUSB for AVR development system

How to connect the development system?

The StartUSB for AVR development system can be easily connected to a PC via a USB connector CN2, Figure 3. The microcontroller pins can be connected to a device via 1x12 connectors CN3 and CN4. An LED labelled POWER is used to indicate whether the board is turned on or off, whereas LEDs PD5 and PD6 can be configured to indicate the logic state on the PD5 and PD6 pins. In order to enable programming via bootloader place jumper J3 (ENABLE BOOT) on the board.

STEP 2: Connect StartUSB for AVR to your PC

Attach StartUSB for AVR to a PC via a USB cable, Figure 3. A PC will automatically start driver installation for a new hardware. If the driver installation fails go one step back and find driver manually in the Program files-Atmel-Flip folder.

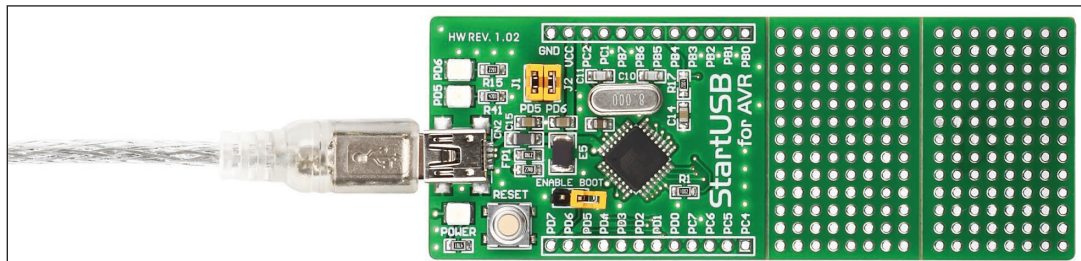


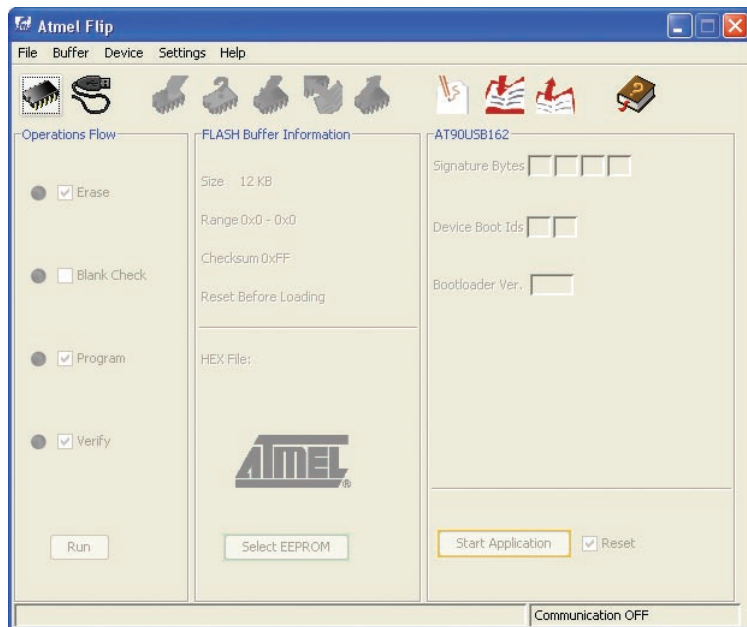
Figure 3: Connecting the system to a PC

STEP 3: Start Atmel FLIP program

To start this program double click on the Atmel FLIP icon

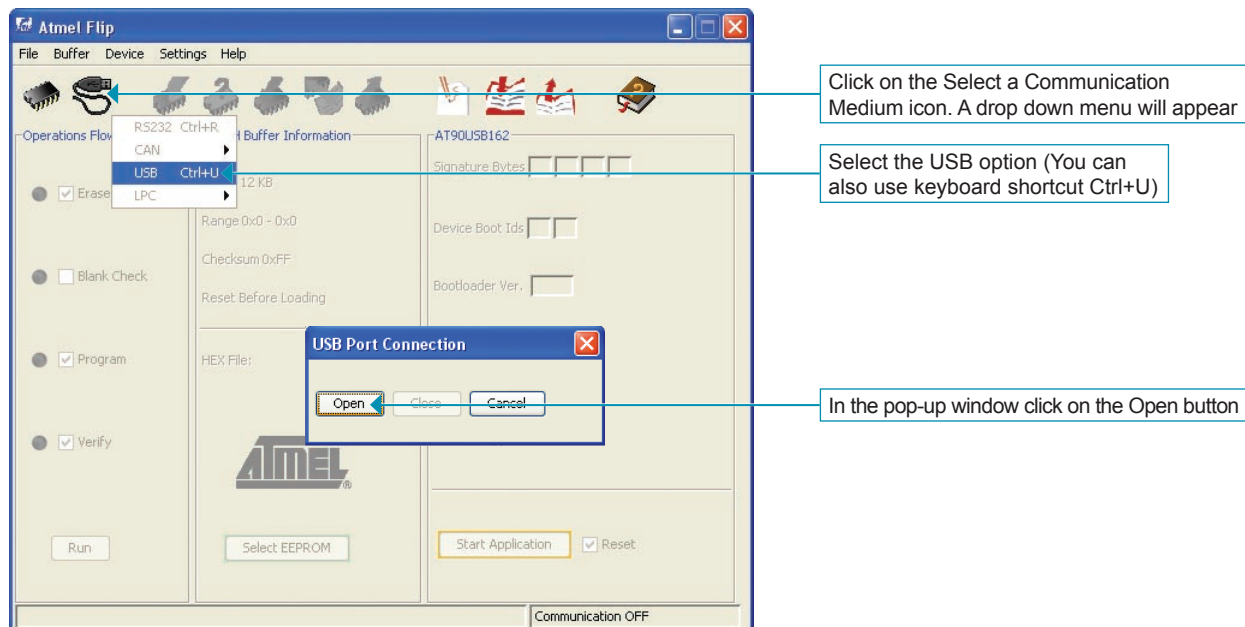
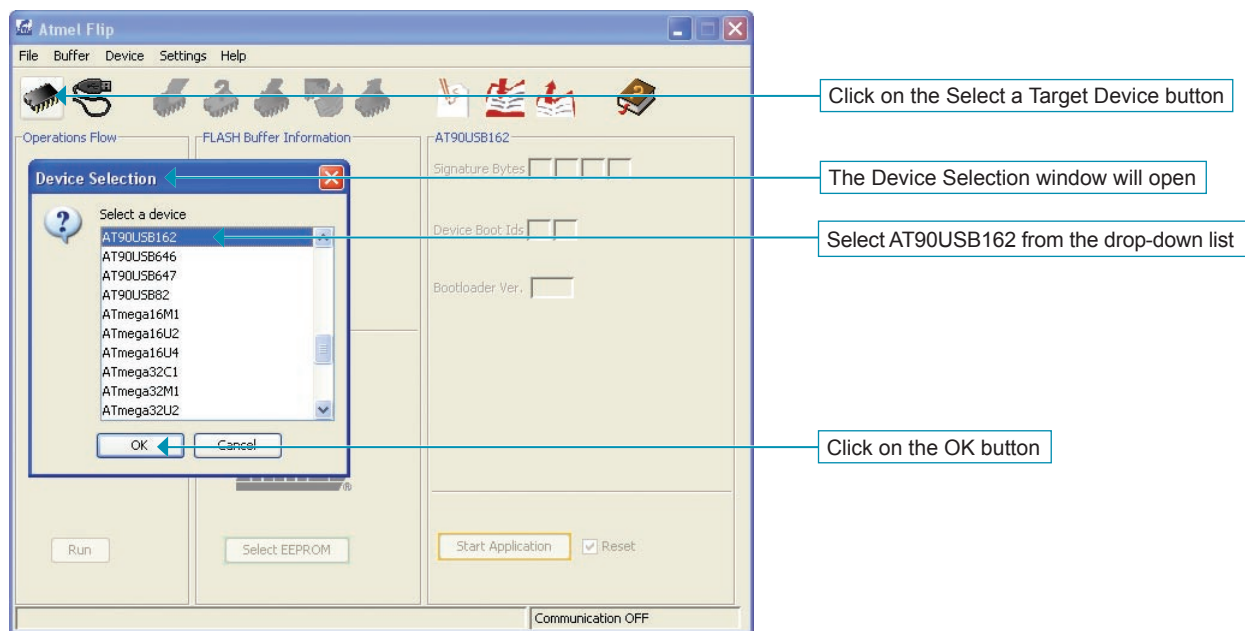


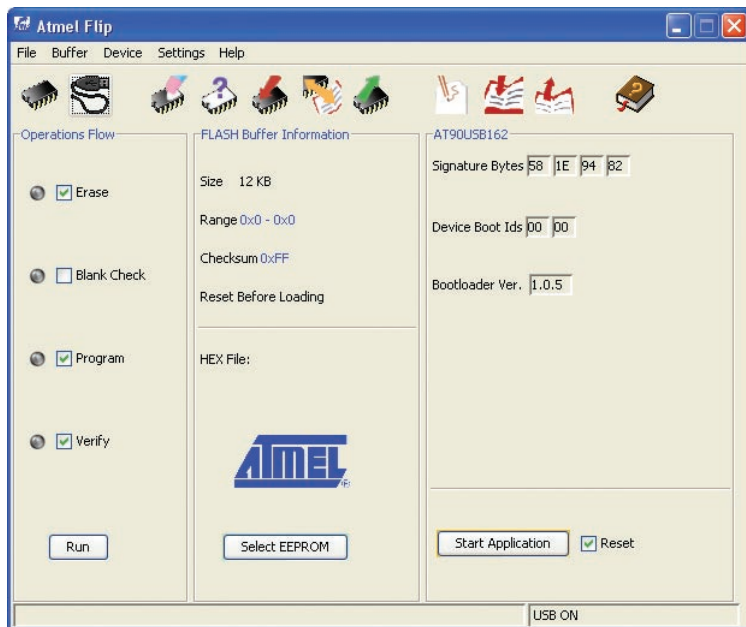
. The Atmel FLIP window will appear on the screen.



STEP 4: Link the program with development system

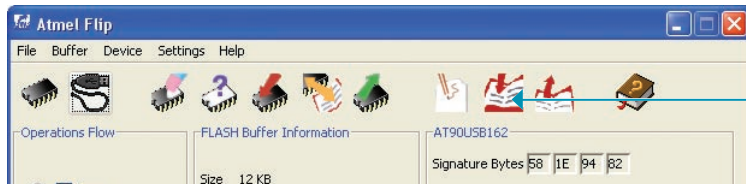
After Atmel FLIP is started press **RESET** button on the StartUSB for AVR development board.



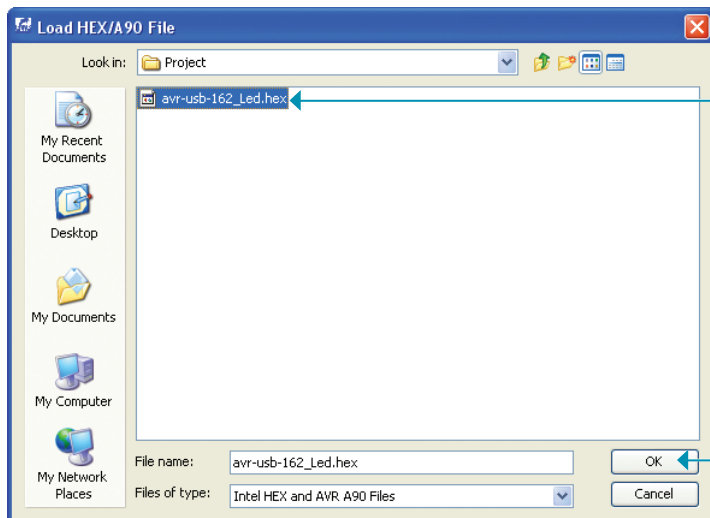


After the link is established all options and buttons will become active

STEP 5: Browse for .hex code

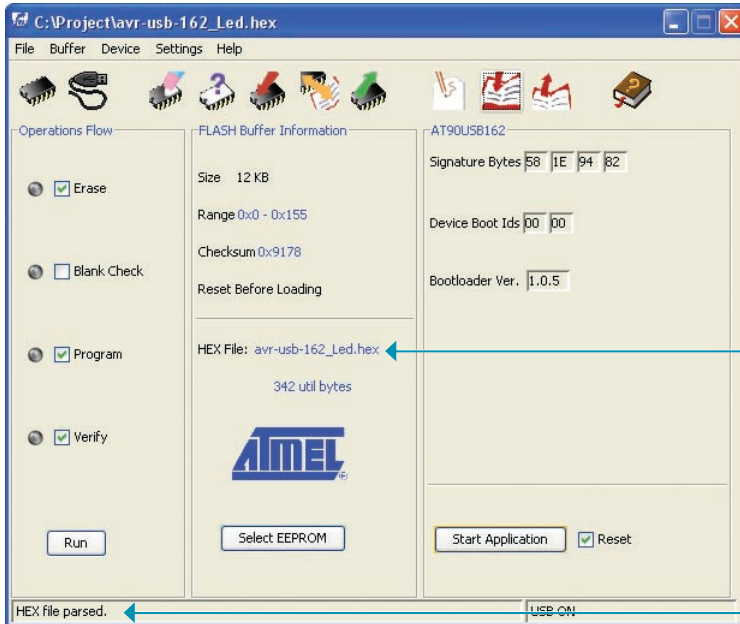


Click on the Load HEX File icon



In the pop-up window browse for a .hex file

Click on the OK button



After the .hex file is loaded, an information about it will appear in the Atmel FLIP window

Information that .hex file is parsed

STEP 6: MCU programming



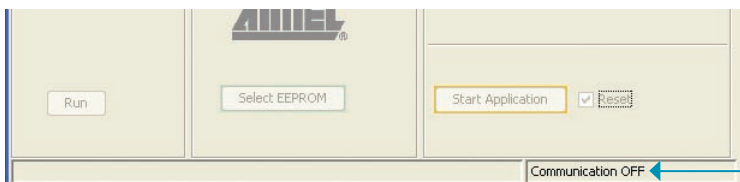
Click on the Run button



Click on the Start Application button

Information that .hex file verification is passed

After the MCU programming is completed, the Atmel FLIP window will become inactive.



Information that link with StartUSB for AVR is disabled (Communication OFF)

NOTE: When the MCU programming is finished, remove jumper J3 (ENABLE BOOT) from the StarUSB for AVR development system. Otherwise, the system will reenter bootloader mode .

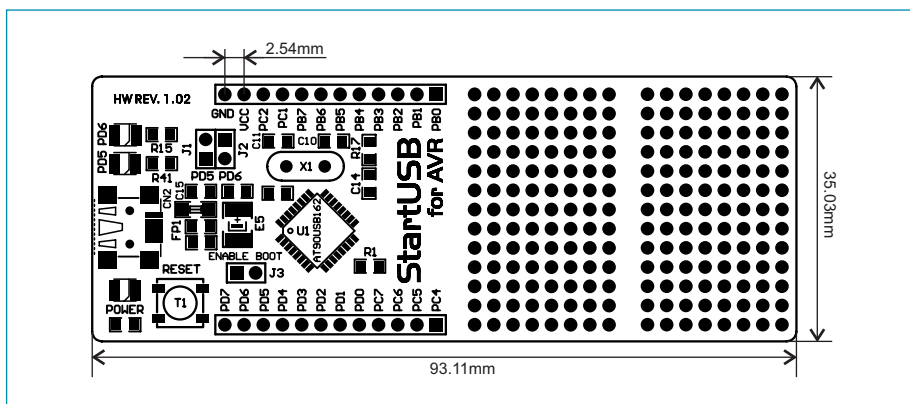


Figure 4: Dimensions of the StartUSB for AVR development system

Adding FLIP to mikroC PRO for AVR compiler

Follow the next few steps to add FLIP to mikroC PRO for AVR tools list. By doing so you can start programming of MCU on StartUSB for AVR directly from the compiler.

The process of adding FLIP tool to mikroBasic PRO for AVR and mikroPascal PRO for AVR is the same as for mikroC PRO for AVR.

STEP 1: Open tools menu

In the compiler window select **Options** from the **Tools** menu or press F12 on keyboard.

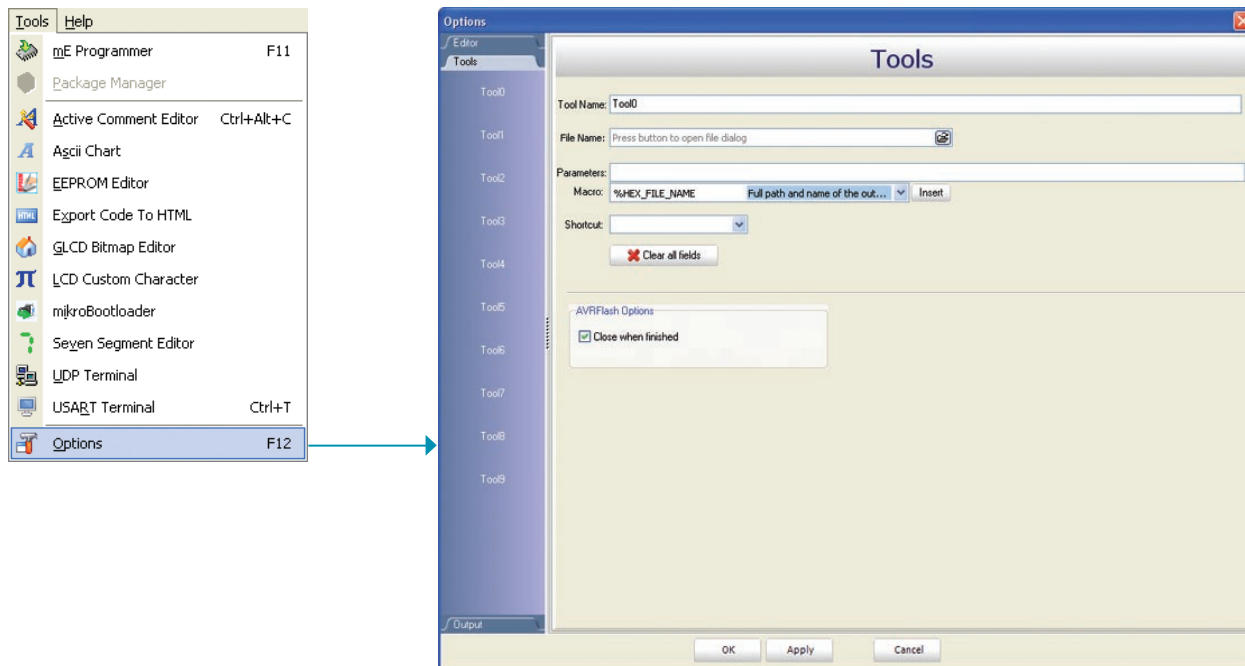
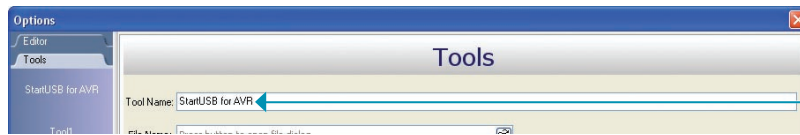
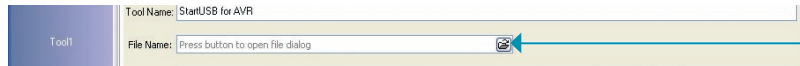


Figure 5: Options window

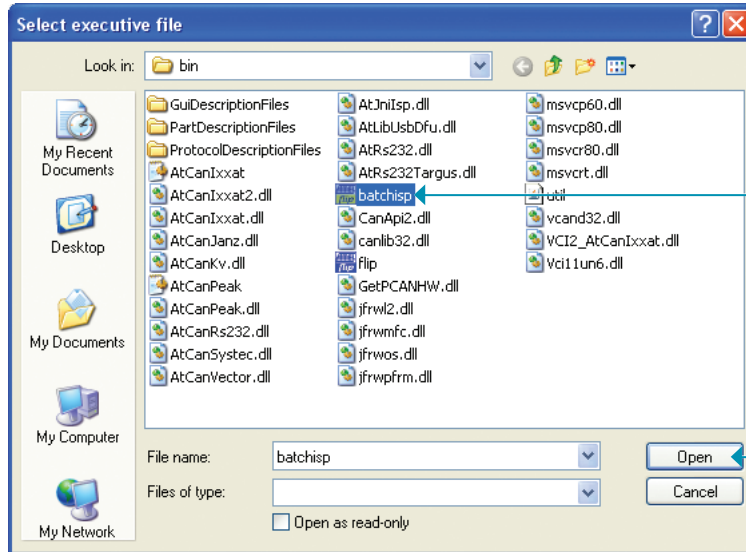
STEP 2: Add tool parameters



Rename Tool0 to StartUSB for AVR



Click on the open button

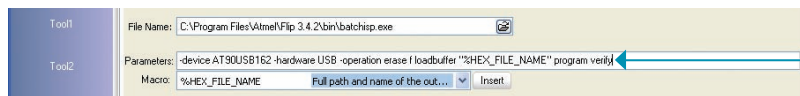


Find batchisp in the C:\Program Files\Atmel\Flip 3.4.2\bin folder

Click on the Open button

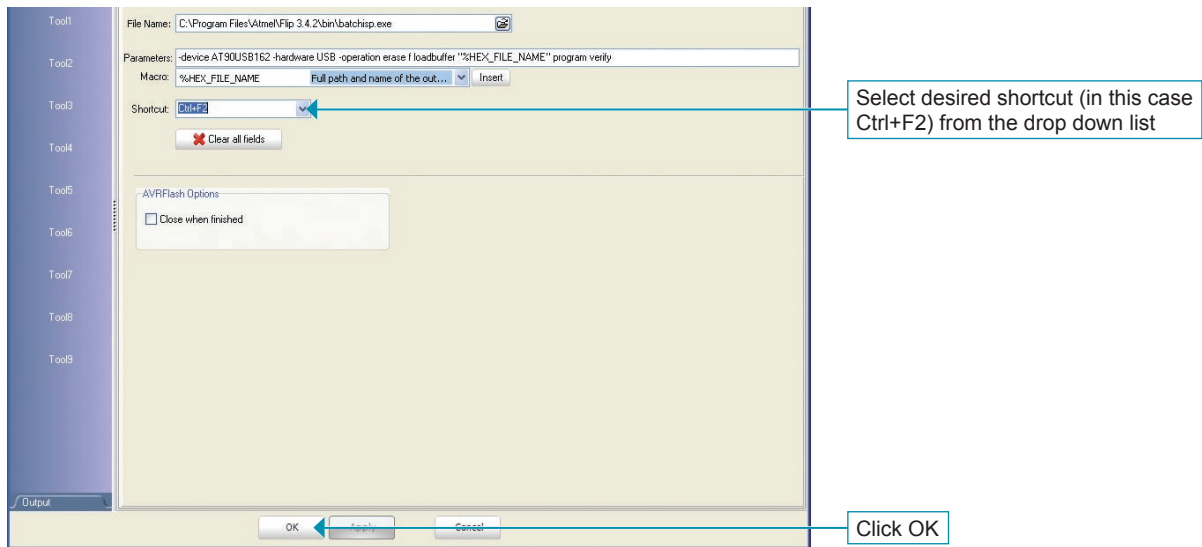


A path to batchisp.exe file will appear in the File Name box



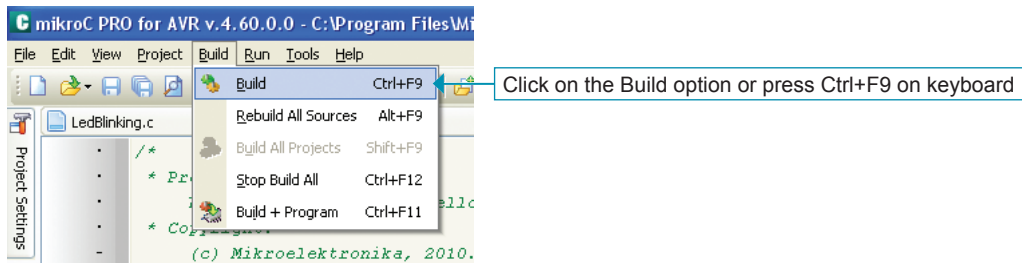
In parameters text box type in command line path: **-device AT90USB162 -hardware USB -operation erase f loadbuffer \"%HEX_FILE_NAME\" program verify**

NOTE: If you copy command line path retype every quotation mark



STEP 3: Compile source code

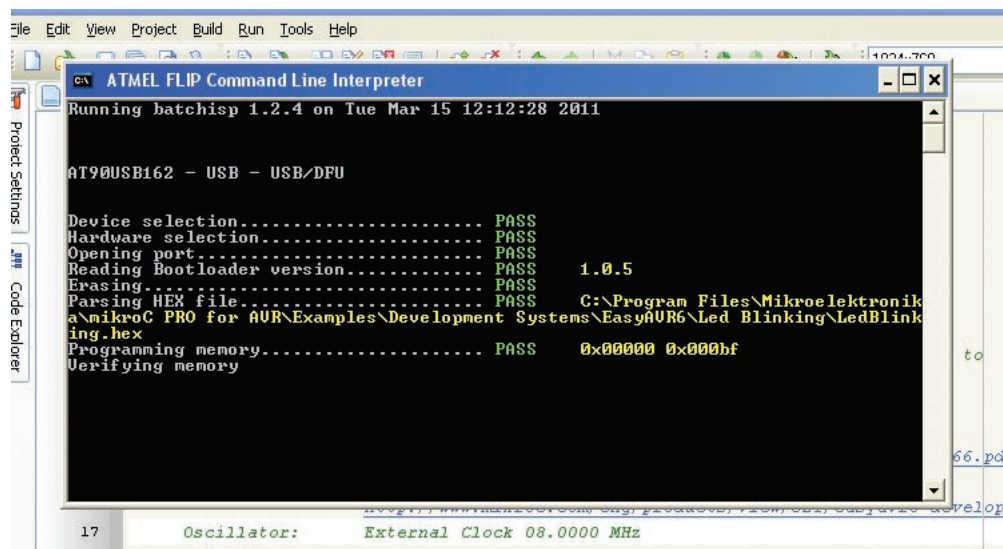
After the source code is written, select the Build option from the Build menu to compile it.



When the source code is compiled, the .hex file is generated and ready to be uploaded into the MCU.

STEP 4: Upload .hex file into MCU

To upload .hex file we will use StartUSB for AVR that was created in step 2. Just press Ctrl+F2 (or chosen shortcut) and .hex file will be automatically transferred into the MCU, Figure 6.



```
File Edit View Project Build Run Tools Help
ATMEL FLIP Command Line Interpreter
Running batchisp 1.2.4 on Tue Mar 15 12:12:28 2011

AT90USB162 - USB - USB/DFU

Device selection..... PASS
Hardware selection..... PASS
Opening port..... PASS
Reading Bootloader version..... PASS      1.0.5
Erasing..... PASS
Parsing HEX file..... PASS      C:\Program Files\Mikroelektronika\mikroC PRO for AVR\Examples\Development Systems\EasyAVR6\Led Blinking\LedBlinking.hex
Programming memory..... PASS      0x000000 0x0000bf
Verifying memory
```

17 Oscillator: External Clock 08.0000 MHz

Figure 6: Uploading .hex file

After uploading is over, a command line window (figure 6) will disappear which means that .hex file is uploaded.



SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD ...making it simple

If you want to learn more about our products, please visit our website at www.mikroe.com

If you are experiencing some problems with any of our products or just need additional information, please place your ticket at www.mikroe.com/en/support

If you have any questions, comments or business proposals, do not hesitate to contact us at office@mikroe.com

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Mikroe:

[MIKROE-682](#)