

CREATING FIRST PROJECT in **mikroC** for 8051

 **MikroElektronika**

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

Project

The mikroC for 8051 organizes applications into projects consisting of a single project file (extension .mcpj) and one or more source files (extension .c). MikroC for 8051 IDE allows you to manage multiple projects (see *Project Manager*). Source files can be compiled only if they are part of a project.

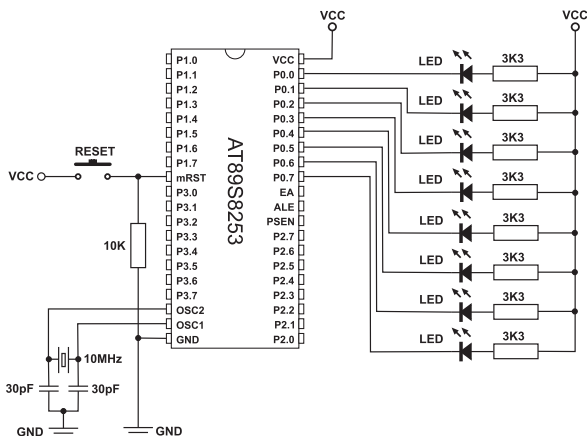
The project file contains the following information:

- ▶ Project name and optional description;
- ▶ Target device;
- ▶ Memory model;
- ▶ Device flags (configuration word);
- ▶ Device clock;
- ▶ List of the project source files;
- ▶ Header files (*.h);
- ▶ Binary files (*.mcl);
- ▶ Image files; and
- ▶ Other files.

Note that the project doesn't include files in the same way as preprocessor does. For more information, see *Add/Remove Files from Project*.

In this simple project demonstration, we will create a new project, write some code, compile it with *mikroC for 8051* and test the results. Our example will make LED diodes blink so it can be easily tested on the 8051 microcontrollers.

Hardware Connection



You will need the the connection scheme shown above to test the code for the 8051 microcontroller. LED diodes are connected to Port P0. In this example you can use any other port because this simple program will change the state of all ports in the same way. Prior to start, you have to go through the three basic steps:

Step 1: Install the compiler

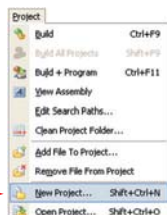
Install the *mikroC for 8051* compiler. Desktop shortcut and start menu shortcuts will be created.

Step 2: Run the compiler

Run the *mikroC for 8051* compiler. The *mikroC for 8051 IDE* (Integrated development Environment) will appear.

Step 3: Create a New Project

The easiest way to create a project is by means of the *New Project Wizard*, drop-down menu **Project** > **New Project** or by clicking the icon *New Project* from *Project Toolbar*.



Click here to start a new project →

New Project Wizard Steps

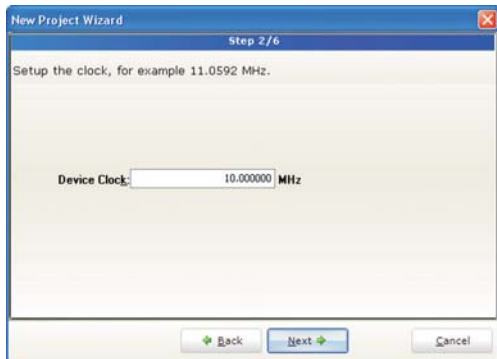
The first window provides basic information on project settings which will be applied in the following steps.



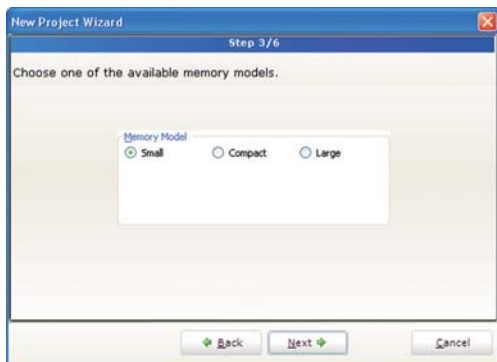
Step 1: From the *Device* drop-down list, select a microcontroller you want to write a code for.



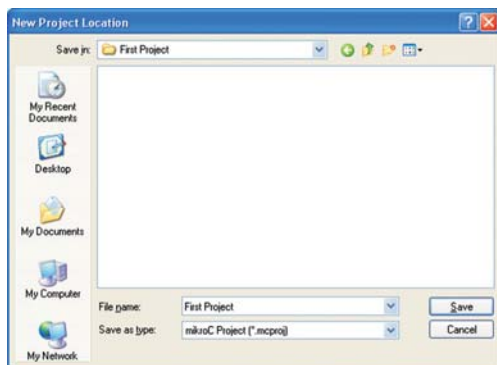
Step 2: Enter the oscillator frequency value.



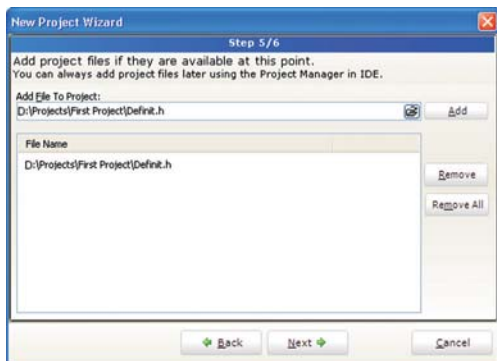
Step 3: Select the desired memory model.



Step 4: Specify the location where your project will be saved.



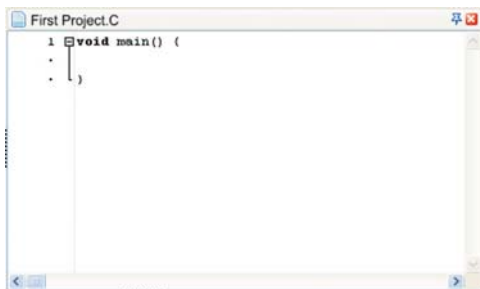
Step 5: Add project files to the project if they are available at this point. You can always add project files later using *Project Manager*.



Step 6: Click *Finish* and IDE will create the project file and an accompanying source files named the same as your project.



A new empty source file will be automatically opened in the *Code Editor* so you can start writing the source code.



This is the source code that will make all microcontroller's I/O pins to change their logic state every second. Since LED are connected to the Port P0, they will blink at the same rate.

```
LedBlinking.c
*
* PORT0, PORT1, PORT2 and PORT3.
*
* * Test configuration:
*
* MCU: AT89S8253
* Dev.Board: Easy8051B
* Oscillator: External Clock 10.0000 MHz
* Ext. Modules: -
* SW: mikroC for 8051 v1.0
*
* * NOTES:
*
* None.
20 */
*
* void main() {
* do {
*     P0 = 0x00; // Turn ON diodes on PORT0
*     P1 = 0x00; // Turn ON diodes on PORT1
*     P2 = 0x00; // Turn ON diodes on PORT2
*     P3 = 0x00; // Turn ON diodes on PORT3
*     Delay_ms(1000); // 1 second delay
*     P0 = 0xFF; // Turn OFF diodes on PORT0
30     P1 = 0xFF; // Turn OFF diodes on PORT1
*     P2 = 0xFF; // Turn OFF diodes on PORT2
*     P3 = 0xFF; // Turn OFF diodes on PORT3
*     Delay_ms(1000); // 1 second delay
*
* } while (1); // Endless loop
* }
```

Compilation

When you have created the project and have written the source code, it's time to compile it. Select **Project > Build** from the drop-down menu or click the *Build Icon* from the *Project Toolbar*. If more than one project is opened you can compile all open projects by selecting **Project > Build All** from the drop-down menu, or click the *Build All Icon* from the *Project Toolbar*.

If there are some errors, you will be notified in the *Message Window*. If no errors are encountered, the *mikroC for 8051* will generate output files.

Output Files

Upon successful compilation, the *mikroC for 8051* will generate output files in the project folder (folder which contains the project file .mcpj). Output files are summarized in the table below:

Format	Description	File Type
Intel HEX	Intel style hex records. Use this file to program 8051 MCU.	.hex
Binary	mikro Compiled Library. Binary distribution of application that can be included in other projects.	.mcl
List File	Overview of 8051 memory allotment: instruction addresses, registers, routines and labels.	.lst
Assembler File	Human readable assembly with symbolic names, extracted from the List File.	.asm

Assembly View

After compiling the program in the *mikroC for 8051*, you can click the *View Assembly* icon or select **Project > View Assembly** from the drop-down menu to review the generated assembly code (.asm file) in a new tab window. Assembly language uses symbolic names and is much easier to understand, therefore.

Customizing Projects

Edit Project

You can change basic project settings such as type of chip, oscillator frequency and memory model in the *Project Settings* window. Any change in the *Project Setting* window affects only currently active project which means that if more than one project is opened, you have to ensure that the right project is set as active one in the *Project Manager*.

Managing Project Group

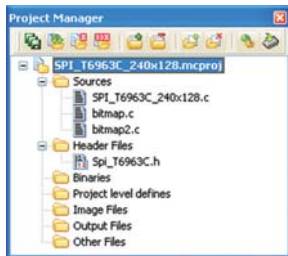
The mikroC for 8051 IDE provides a convenient option which enables several projects to be opened simultaneously. If you have several projects being related in some way, you can create a project group.

The project group may be saved by clicking the *Save Project Group* icon in the *Project Manager* window. It may also be reopened by clicking the *Open Project Group* icon. All relevant data about the project group is stored in the project group file (extension .mpgroup).

Add/Remove Files from Project

The project can contain the following file types:

- ▶ .c source files;
- ▶ .h header files;
- ▶ .mcl binary files;
- ▶ .pld project level defines files (future upgrade);
- ▶ image files;
- ▶ .hex, .asm and .lst files, see output files. These files cannot be added or removed from project; and
- ▶ other files.



The list of relevant source files is stored in the project file (extension .mcproj).

To add source file to the project, click the *Add File to Project* icon. Each added source file must be self-contained, i.e. must have all necessary definitions after preprocessing.

To remove file(s) from the project, click the *Remove File from Project* icon.

Note

To include the header files (extension .h), use the preprocessor directive `#include`. See *File Inclusion* for more information.

Source Files

Source files containing C code should have extension `.c`. The list of source files relevant to the application is stored in the project file with extension `.mcproj`, along with other project information. You can compile source files only if they are part of the project.

Creating new source file

To create a new source file, do the following:

Select **File** › **New Unit** from the drop-down menu, or press `CTRL+N`, or click the *New File* icon from *File Toolbar*.

A new tab will be opened. It is a new source file. Select **File** › **Save** from the drop-down menu, or press `CTRL+S`, or click the *Save File* icon from *File Toolbar* and name it as you want.

If you use *New Project Wizard*, an empty source file, named after the project with extension `.c`, will be created automatically. The *mikroC 8051* doesn't require you to have a source file named the same as the project, it's just a matter of convenience.

Opening an existing file

Select **File** › **Open** from the drop-down menu, or press `CTRL+O`, or click the *Open File* icon from the *File Toolbar*. In *Open* dialog browse to the location of the file that you want to open, select it and click the *Open* button.

The selected file is displayed in its own tab. If the selected file is already opened, its current *Editor* tab will become active.

Printing an open file

Make sure that the window containing the file you want to print is an active window.

Select **File** › **Print** from the drop-down menu or press `CTRL+P`.

In the *Print Preview Window*, set a desired layout of the document and click the OK button. The file will be printed on the selected printer.

Note

Use the preprocessor directive `#include` to include header files with extension `.h`. Do not rely on the preprocessor to include source files other than headers. See *Add/Remove Files from Project* for more information.

Saving file

Make sure that the window containing the file that you want to save is active.

Select **File** > **Save** from the drop-down menu, or press Ctrl+S, or click the *Save File* icon from *File Toolbar*.

Saving file under a different name

Make sure that the window containing the file that you want to save is active.

Select **File** > **Save As** from the drop-down menu. The *New File Name* dialog will be displayed.

In the dialog, browse to the folder in which you want to save the file.

In the *File Name* field, modify the name of the file you want to save.

Click the *Save* button.

Closing file

Make sure that the tab containing the file that you want to close is the active tab.

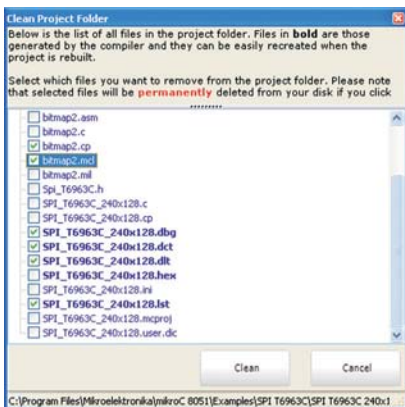
Select **File** > **Close** from the drop-down menu, or right click the tab of the file that you want to close and select the *Close* option from menu.

If the file has been changed since it was last saved, you will be prompted to save your changes.

Clean Project Folder

This menu gives you possibility to choose which files from your current project you want to delete.

Select the files you want to remove from the project and click *Clean*. Please note that only files marked in bold are generated by compiler and can be recreated after compiling.





MikroElektronika

DEVELOPMENT TOOLS | COMPILERS | BOOKS

If you are experiencing problems with any of our product or you just want additional information, please let us know.

TECHNICAL SUPPORT: www.mikroe.com/en/support

If you have any question, comment or business proposal, please contact us.

web: www.mikroe.com

e-mail: office@mikroe.com