

COMPILER'S IDE

 **MikroElektronika**

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

TO OUR VALUED CUSTOMERS

I want to express my thanks to you for being interested in our products and having confidence in MikroElektronika.

It is our intention to provide you with the best quality products. Furthermore, we will continue to improve our performance to better suit your needs.

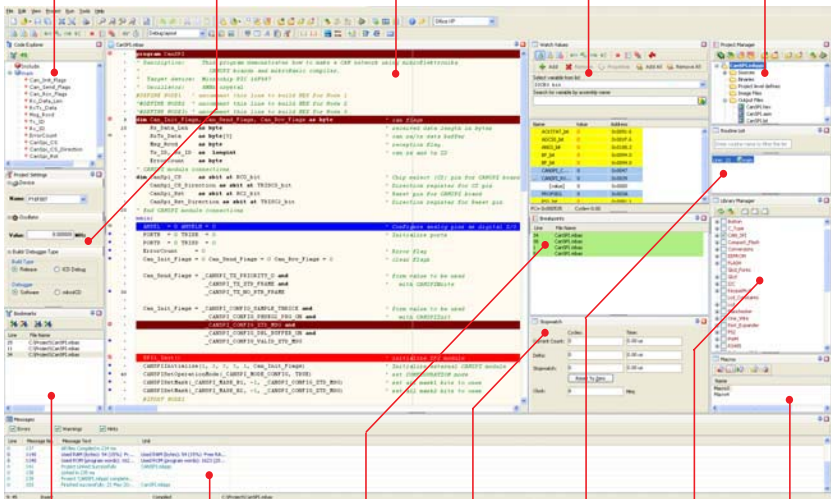


Nebojsa Matic
General Manager

IDE Overview

MikroElektronika's compilers have a user-friendly and intuitive *Integrated Development Environment* (IDE) which provides all the benefits that modern Windows environments offer. Besides, it can be customized to better suit various needs of the user.

Code Explorer **Project Settings** **Code Editor** **Watch Values** **Project Manager**



Bookmarks **Messages** **Breakpoints** **Routine List** **Macros**

Stopwatch **Library Manager**

- Code Explorer** enables easy code writing;
- Project Settings** allows general project settings;
- Code Editor** advanced text editor;
- Watch Values** allows register and variable values monitoring while the program is running;
- Project Manager** allows multiple project management;
- Bookmarks** make navigation through a large code easier;
- Messages** displays all messages generated during compiling and linking;
- Breakpoints** displays list of breakpoints currently set in the project;
- Stopwatch** displays number of cycles/time required for a code portion to be executed;
- Routine List** displays list of routines in alphabetical order;
- Library Manager** enables simple handling of libraries used in the project;
- Macros** allows a series of characters (code portions) to be recorded and reused later in the project;

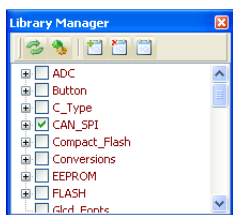
Customizing IDE Layout

Docking windows

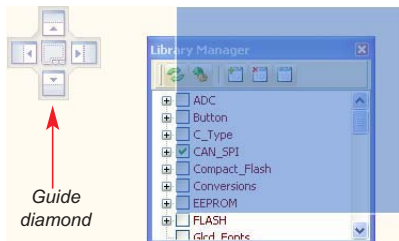
As mentioned above, it is possible to customize the space for code viewing and editing (IDE). There are many options for arranging windows within the IDE such as: tab-dock document windows, dock tool windows to the edge of the IDE, minimize tool windows along the edge of the IDE, tile document windows, reset window placement to the default layout etc.

In order to dock a window within the IDE follow the steps below:

- Step 1:** Click the window you want to dock. In our example, it is the **Library Manager** window which appears by selecting the **Library Manager** option from the **View** drop-down menu. Hold the CTRL key while dragging the window in order to disable docking.



- Step 2:** Drag the window from its current location. A guide diamond will appear. The four arrows of the diamond point towards the four edges of the IDE.



- Step 3:** Move the pointer over the corresponding portion of the guide diamond. An outline of the window appears in the designated area on the screen.

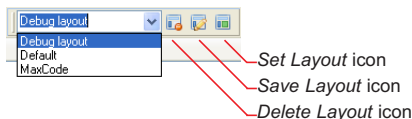
- Step 4:** To dock the window here, release the mouse button.

Saving Layout

Once you have created desired window layout, you can save it by typing its name and pressing the **Save Layout** icon.

To set the desired layout, it is necessary to select it from the layout drop-down list and click the **Set Layout** icon.

To remove a layout from the drop-down list, it is necessary to select it and click the **Delete Layout** icon.

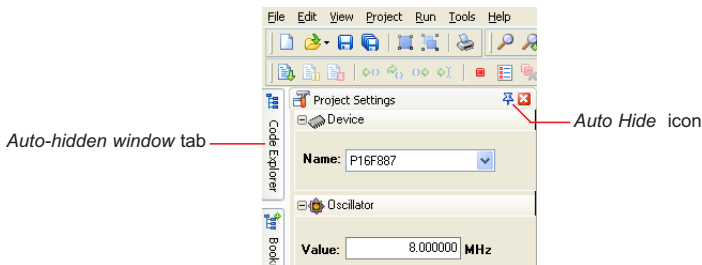


Auto Hide

The **Auto Hide** option enables you to minimize the whole tool window along the edges of the IDE when it is not in use in order to see more of the code.

Step 1: Click the window you want to minimize.

Step 2: Click the **Auto Hide** icon on the title bar of the window and it will automatically slide to its tab. While the window is auto-hidden, its name and icon are visible on its tab.



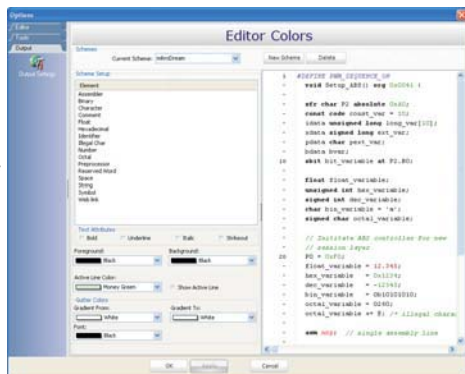
Step 3: To display an auto-hidden window, move your pointer over the tab. The window slides back into view and is ready for use.

Step 4: In order for the window to exit Auto hide mode, click the **Auto Hide** icon  while the window is visible.

Options Window

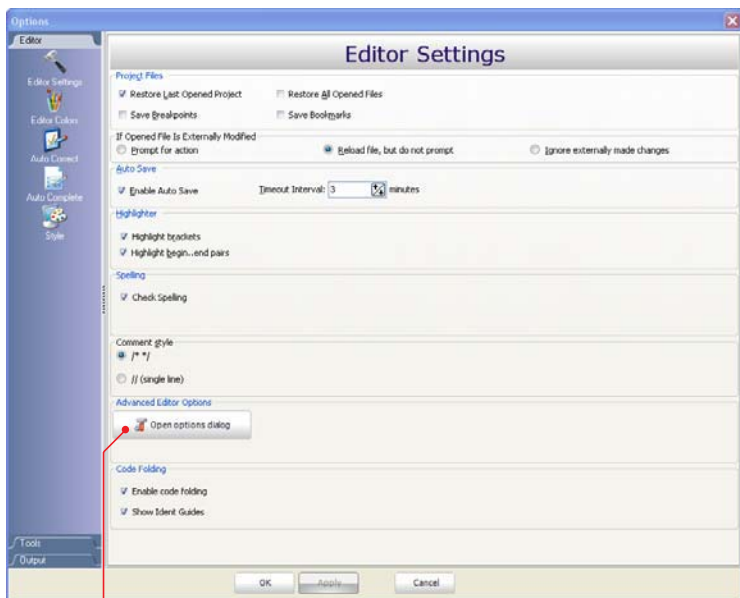
The **Options** window can be accessed by clicking **Options** from the **Tools** drop-down menu or the **Options** icon or by pressing the [F12] key.

The **Options** window consists of three tabs: **Editor**, **Tools** and **Output**. Each of these tabs contains options for modifying the code editor. The code is edited in the same way as in any standard Windows text-editor, which includes familiar **Copy**, **Paste** and **Undo** commands.



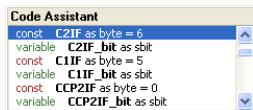
Editor Settings Window

To access the **Editor Settings** window, select the first icon within the **Editor** tab placed on the left portion of the **Options** window. This window enables you to configure the following options: *Project Files*, *Auto Save*, *Highlighter*, *Spelling*, *Advanced Editor Options* and *Code Folding*. To access some of the advanced **Editor** options, click the **Open options dialog** button within the **Advanced Editor Options** option.



Code Assistant [Ctrl+Space]

If you type the first few letters of a word and then press CTRL+SPACE, all the valid identifiers matching these letters will be shown in a floating panel (see figure on the right). Keep on typing to narrow the choice of the identifiers, or select one identifier from the list using the keyboard arrows and press **Enter**.



Parameter Assistant [Shift+Ctrl+Space]

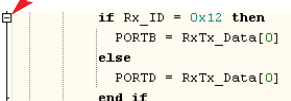
The **Parameter Assistant** option will be automatically invoked by using opening parenthesis "(" or by pressing SHIFT+CTRL+SPACE. If the name of a valid function precedes the parenthesis, then the expected parameters will be displayed in a floating panel. As you type the actual parameter, the next expected parameter becomes bold.



Code Folding

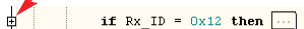
The **Code Folding** option allows the users to selectively hide and display sections of the source code. In this way it is easier to manage large codes by viewing only those sections of the code that are relevant to the current actions.

Expand code symbol

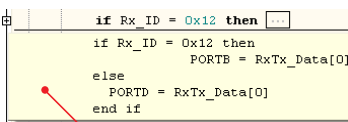


While typing the code, the code folding symbol (AND) appears automatically. Use this symbol to expand/collapse code sections.

Collapse code symbol



If you place the cursor over the tooltip box [...], the collapsed text will be shown in the tooltip style box.



Bookmarks [Ctrl+Shift+NUMBER]

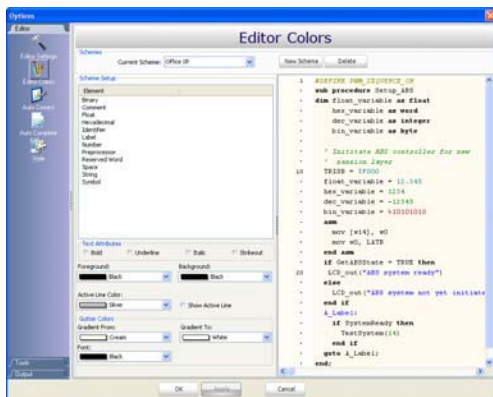
The **Bookmarks** option makes a search through a large code easier. To set a bookmark, use CTRL+SHIFT+NUMBER. The bookmark in the form of the selected number appears on the left side of the program line. To jump to a bookmark, use CTRL+NUMBER.

Goto Line [Ctrl+G]

Similarly, the **Goto Line** option also makes a search through a large code easier. Use the CTRL+G shortcut to activate this option.

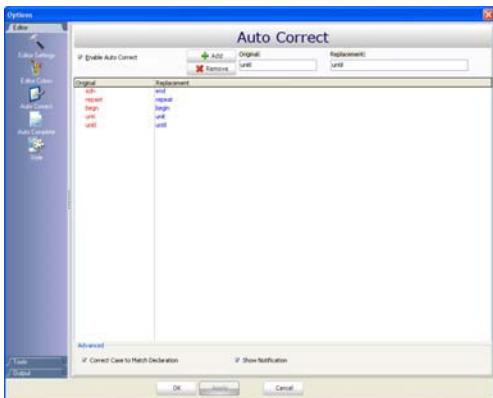
Editor Colors Window

Options contained in the **Editor Colors** window allow you to customize IDE layout, i.e. the colors of all windows and fonts. You can choose any of the predefined colour schemes, edit them or make new ones. Besides, options in this window enable you to change text attributes (**Bold**, *Italic*, Underline or ~~Strikeout~~).



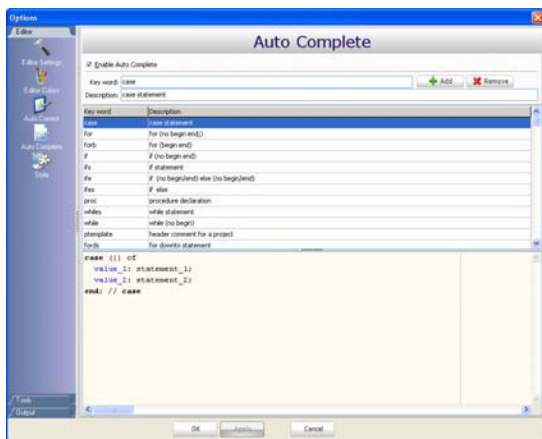
Auto Correct Window

The **Auto Correct** option is used for correcting common typing mistakes. To access the list of recognized typos, click the **Auto Correct** icon in the **Editor** tab. You can also add your own preferences to the list.



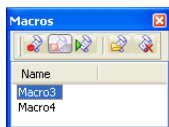
Auto Complete Window

The process of program writing may be sped up by using the **Auto Complete** option in a way that a *keyword* in the program is replaced by the code portion (can be any text) previously assigned to that word. To add a new keyword to the list of valid keywords, click the **Add** button within the **Auto Complete** window, enter its name, description and appropriate code to replace it in the program. To confirm it, click the **Apply** button. If you type a keyword anywhere in the program and press CTRL+J, it will be automatically replaced by the appropriate code.



Macros

A macro represents a series of characters that have been 'recorded' so they can be reused in the code.

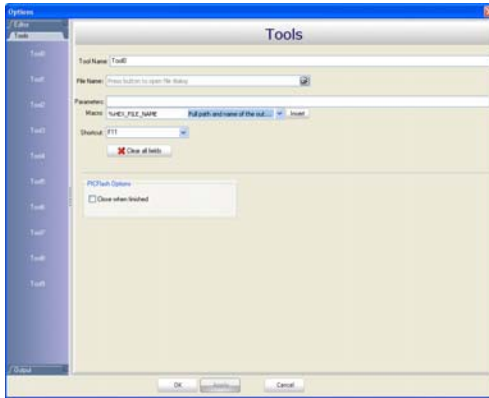


Auto complete macros can retrieve system information and include it in the project:

- `%DATE%` - date of program writing;
- `%TIME%` - current time;
- `%DEVICE%` - device (MCU) name as specified in project settings;
- `%DEVICE_CLOCK%` - clock as specified in project settings; and
- `%COMPILER%` - version of compiler in use.

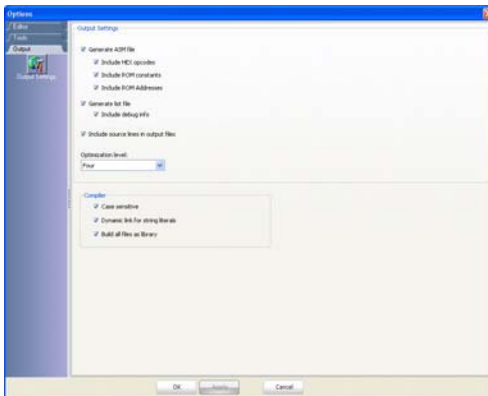
Tools

The **Options** window also contains the **Tools** tab which enables the use of shortcuts to external programs, such as *Calculator* or *Notepad*. Up to 10 different shortcuts may be set by defining potential shortcuts Tool0 - Tool9.



Output

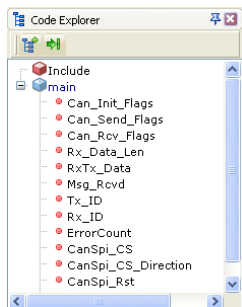
By clicking the **Output Settings** icon within the **Output** tab, it is possible to configure the number and type of output files (ASM files and List file) as well as the optimization level of the compiled code.



Code Explorer Window

The **Code Explorer** window appears by clicking the **Code Explorer** option from the **View** drop-down menu. It gives a clear view of each item declared within the source code. It is possible to jump to the declaration of any item by right clicking it. Apart from the list of defined and declared items, the **Code Explorer** window displays a message about the first error and its location in the code.

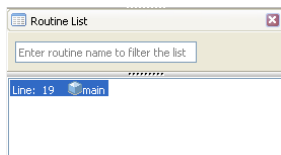
The **Full Expand/Collapse** and **Locate in code** options are available in the **Code Explorer** window.



Routine List Window

The **Routine List** window appears by clicking the **Routine List** option from the **View** drop-down menu. It displays all available routines in alphabetical order. This window can also be accessed by pressing CTRL+L.

Jump to a desired routine may be executed by double clicking it.

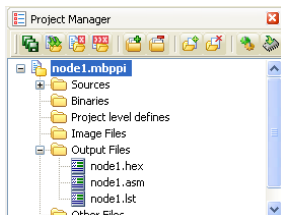


Project Manager Window

The **Project Manager** option from the **View** drop-down menu allows the user to manage multiple projects.

Several projects, which together make a project group, can be simultaneously opened, but only one of them can be active at the time.

Any project within the **Project Manager** window may become active by double clicking it.



Options available in the Project Manager window:

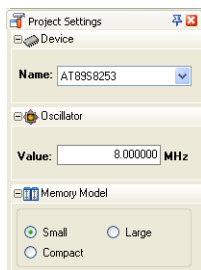
- | | |
|---------------------------------|--------------------------------------|
| - Save Project Group; | - Remove Project From Project Group; |
| - Open Project Group; | - Add File to Project; |
| - Close Project; | - Remove File From Project; |
| - Close Project Group; | - Build Project (Ctrl+F9); and |
| - Add Project To Project Group; | - mE Programmer (F11). |

Project Settings Window

The **Project Settings** window, which appears by clicking the **Project Settings** option from the **View** drop-down menu, contains several options. The **Device** option is used for selecting the appropriate device from the device drop-down list. The **Oscillator** option is used for entering the oscillator frequency value. PIC microcontrollers contain the **Build/Debugger Type** option, whereas 8051 microcontrollers contain the **Memory Model** option used for selecting a desired memory model.



Project settings for PIC



Project settings for 8051



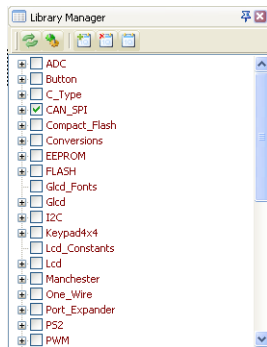
Project settings for AVR

Library Manager Window

The **Library Manager** window, which appears by clicking on the **Library Manager** option from the **View** drop-down list, lists all the libraries (extension **.mcl**) which are currently stored in the **Uses** folder and enables you use them in the project. A desired library is added to the project by putting a tick in the check box next to the library name.

In order to have all library functions accessible, it is sufficient to simply click the **Check All** button and all the libraries will be selected. If none of these libraries are needed in the project, click the **Clear All** button and all the libraries will be cleared from the project.

Only selected libraries will be included in the project and linked if needed.

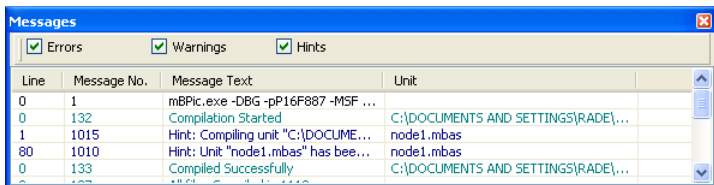


Library Dependencies


Certain libraries use (depend on) functions and/or variables and constants defined in other libraries. For example, the **SPI_Glcd** library uses **Glcd_Fonts** and **Port_Expander** libraries. The **Port_Expander** library uses the **SPI** library. It means that if you check the **SPI_Glcd** library in the **Library Manager** window, all the libraries on which it depends will also be checked.

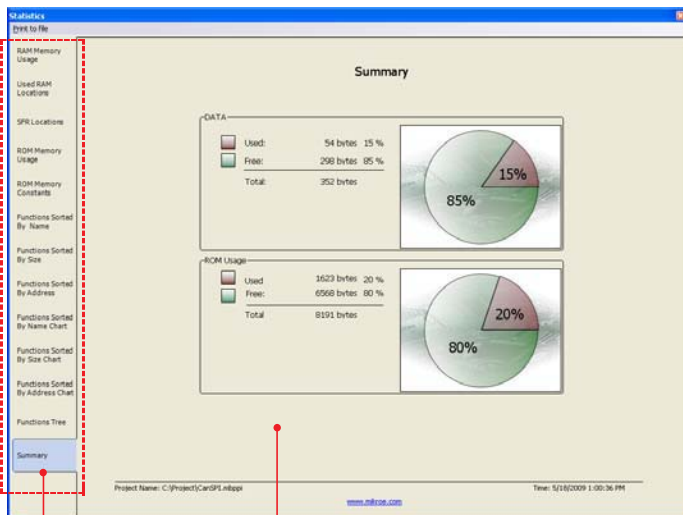
Messages Window

The **Messages** window appears by clicking on the **Messages** option in the **View** drop-down menu. It can also be accessed by clicking the **Windows** option from the same menu. The **Messages** window is located at the bottom of the main window by default and displays locations and types of errors encountered by the compiler. If errors are encountered during the process of compiling, the compiler will report them and will not generate a hex file. The compiler also reports warnings, but these do not affect the output file. In other words, only errors can interfere with the generation of hex file.



Statistics Window

After successful compilation, you can review statistics of your code. Click the **Statistic** icon  and choose appropriate statistic overview from the window for it to appear.



Statistics tabs

Main Statistics window with the Summary tab activated

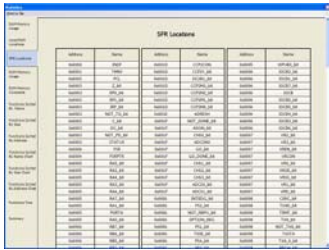
Here are examples of some of the statistic overviews:



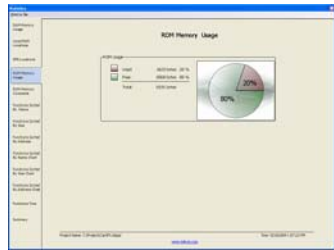
RAM Memory Usage



Used RAM Locations



SFR Locations



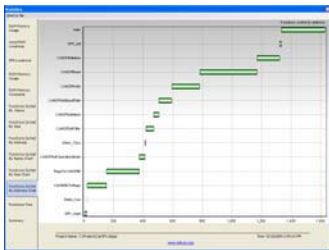
ROM Memory Usage



Functions Sorted By Name



Functions Sorted By Name Chart



Functions Sorted By Address Chart

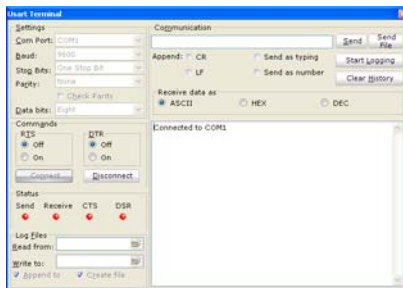


Functions Tree

Integrated Tools

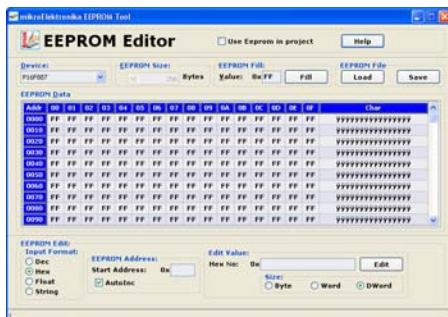
USART Terminal Window

Compilers include the *USART* terminal for RS232 communication. As such, it represents a good substitute for the standard *Windows Hyper* terminal. To show its window, select the **USART Terminal** option from the **Tools** drop-down menu or click the **USART Terminal** icon from the **Tools** toolbar. The **USART Terminal** window contains options for serial communication configuration and displays sent/received data.



EEPROM Editor Window

The **EEPROM Editor** window appears by selecting the **EEPROM Editor** option from the **Tools** drop-down menu. It is used for manipulating MCU's EEPROM memory. For the purpose of modifying individual locations, it is necessary to enter a new value in the **.hex** format for appropriate address. If the new value is to be of Byte, Word or DWord type, select appropriate option at the bottom of the window, enter the value in the **Edit Value** field and click the **Edit** button. At last, click the **Save** button to save data as a document with the **.hex** extension. If the **Use EEPROM in Project** option is checked, this data will be automatically loaded during the process of programming.

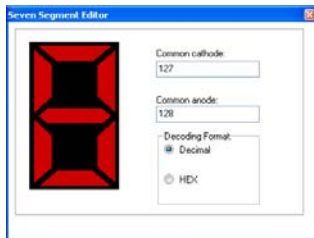


ASCII Chart Window

Hex Chart	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NA	000	001	010	011	020	021	030	031	040	041	050	051	060	061	070
1	080	081	082	083	084	085	086	087	088	089	08A	08B	08C	08D	08E	08F
2	090	091	092	093	094	095	096	097	098	099	09A	09B	09C	09D	09E	09F
3	0A0	0A1	0A2	0A3	0A4	0A5	0A6	0A7	0A8	0A9	0AA	0AB	0AC	0AD	0AE	0AF
4	0B0	0B1	0B2	0B3	0B4	0B5	0B6	0B7	0B8	0B9	0BA	0BB	0BC	0BD	0BE	0BF
5	0C0	0C1	0C2	0C3	0C4	0C5	0C6	0C7	0C8	0C9	0CA	0CB	0CC	0CD	0CE	0CF
6	0D0	0D1	0D2	0D3	0D4	0D5	0D6	0D7	0D8	0D9	0DA	0DB	0DC	0DD	0DE	0DF
7	0E0	0E1	0E2	0E3	0E4	0E5	0E6	0E7	0E8	0E9	0EA	0EB	0EC	0ED	0EE	0EF
8	0F0	0F1	0F2	0F3	0F4	0F5	0F6	0F7	0F8	0F9	0FA	0FB	0FC	0FD	0FE	0FF
9	100	101	102	103	104	105	106	107	108	109	10A	10B	10C	10D	10E	10F
A	110	111	112	113	114	115	116	117	118	119	11A	11B	11C	11D	11E	11F
B	120	121	122	123	124	125	126	127	128	129	12A	12B	12C	12D	12E	12F
C	130	131	132	133	134	135	136	137	138	139	13A	13B	13C	13D	13E	13F
D	140	141	142	143	144	145	146	147	148	149	14A	14B	14C	14D	14E	14F
E	150	151	152	153	154	155	156	157	158	159	15A	15B	15C	15D	15E	15F
F	160	161	162	163	164	165	166	167	168	169	16A	16B	16C	16D	16E	16F

The **ASCII Chart** window is a handy tool used when working with LCD display. Select the **ASCII Chart** option from the **Tools** drop-down menu or click the **View ASCII Chart** icon from the **Tools** toolbar. The **ASCII Chart** window shows standard ASCII character codes.

Seven Segment Editor



The **Seven Segment Editor** window is a convenient visual panel which displays decimal/hex values for any pattern the user would like to display on a 7-segment display. These values will be shown in the boxes next to the image. A desired pattern is created by clicking on the parts of the 7-segment image. Select the **Seven Segment Editor** option from the **Tools** drop-down menu or click the **Seven segment editor** icon from the **Tools** toolbar.

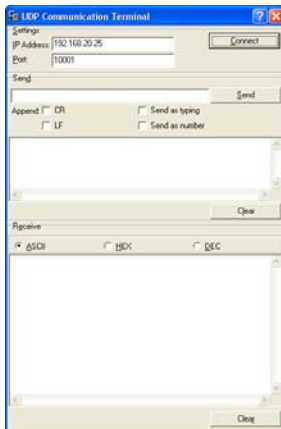
HID Terminal



The **mikroBasic PRO for PIC** compiler includes the **HID terminal** for USB communication. To access it, select the **HID Terminal** option from the **Tools** menu.

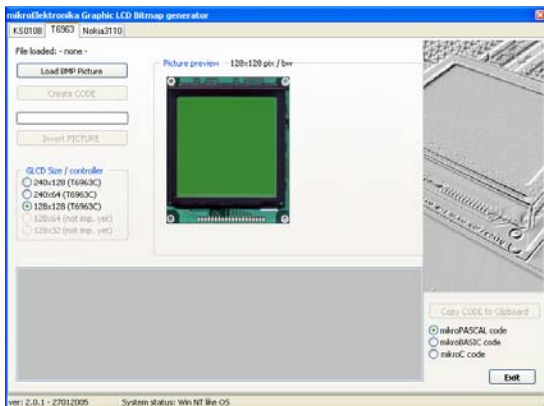
UDP Terminal

The compiler includes the UDP Terminal. To access it, select the **UDP Terminal** option from the **Tools** menu.



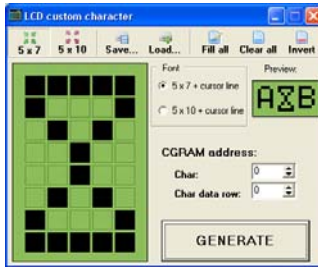
Graphic LCD Bitmap Generator

The compiler includes the Graphic LCD Bitmap Generator. To access it, select the **GLCD Bitmap Editor** option from the **Tools** drop-down menu. For it to appear in the window, select the type of display to use and load desired picture in the **bmp** format. The picture must be monochromatic and must be in the same resolution as the display selected. The generator output is a program compatible code.



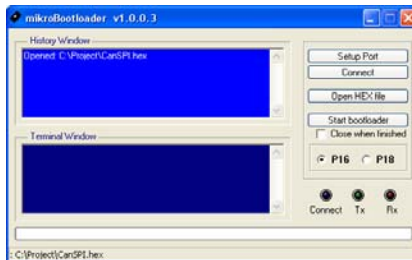
LCD Custom Character Generator

The compiler includes the LCD Custom Character generator. To access it, select the **LCD Custom Character** option from the **Tools** menu. It is used similarly to the **Seven Segment Editor** tool. Create a symbol by clicking the small square fields, specify its position and row and click the **GENERATE** button. The generator output is a program compatible code.



mikroBootloader

The PIC16F87X family of microcontrollers has the ability to modify its own program memory while working. This feature allows a small bootloader program to be loaded into the microcontroller memory. Thanks to it, a new executable code (firmware) can be loaded into the microcontroller, if needed, using serial communication. No extra programmer is needed. The simplest bootloaders start the user code execution, unless a new firmware needs to be downloaded. If it is necessary to download a new firmware, the bootloader receives data and writes it into the program memory. There are many variations and additional features that can be added to improve reliability and simplify the use of the bootloader. mikroBootloader can be used only with PIC microcontrollers which are able to modify their own *Flash* memory.

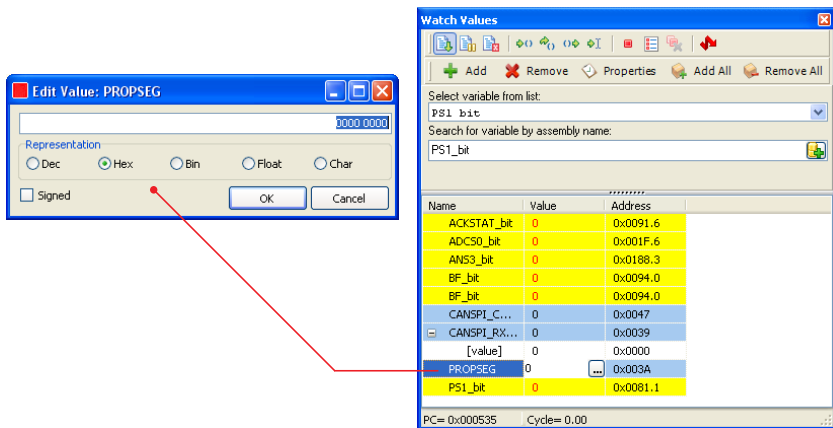


Software Simulator Overview

A software simulator is an integral part of the compiler. It is designed to simulate the operation of microcontrollers and assist the user in debugging code written for these devices. After you have successfully compiled your project, you can run the software simulator by selecting the **Start Debugger** option from the **Run** drop-down menu, or by clicking the **Start Debugger** icon from the **Debugger** toolbar. Starting the software simulator makes more options available such as *Step Into*, *Step Over*, *Step Out*, *Run to Cursor*, etc. The next line to execute is highlighted in blue by default.

Watch Values Window

The **Watch Values** window is the main software simulator window which allows you to monitor addresses and values of the microcontroller's variables and registers while simulating the program. To access it, select the **Debug Windows > Watch Window** option from the **View** drop-down menu.



There are two ways of adding variable/register to the watch list. One is by adding their real names (variable name in the code). It is sufficient to select the desired variable/register in the **Select variable from list** field and click the **Add** button. The other is by adding their assigned names (assembly variable name). Just type the assigned name of the variable/register you want to display in the **Search the variable by assembly name** field and click the **Add** button.

Note

The *Software Simulator* simulates the program flow and execution of instruction lines, but it cannot fully emulate device behavior, i.e. it doesn't update timers, interrupt flags, etc.

Variables can also be removed from the **Watch Values** window by selecting the desired variable and clicking the **Remove** button.

The **Add All** button adds all variables to the list.

The **Remove All** button removes all variables from the list.

Values change during the process of simulation. Recently changed items are colored red.

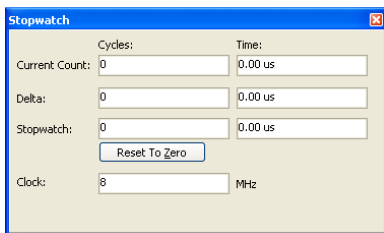
Double click on a variable or the **Properties** button opens the **Edit Value** window in which you can assign a new value to the selected variable/register. You can choose between decimal, hexadecimal, binary, float or character format of the variable/register representation. All representations except float are unsigned by default. For signed representation, check the box next to the **Signed** label.

A variable/register value can also be changed by double clicking on it's field and typing a new value directly.

Stopwatch Window

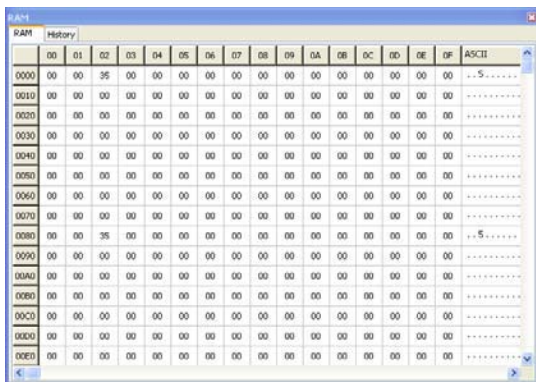
The software simulator **Stopwatch** window appears by selecting the **Debug Windows** › **View Stopwatch** option from the **View** drop-down menu.

The **Stopwatch** window displays the number of cycles or time required for a code portion to be executed. The stopwatch measures the execution time (number of cycles) since the moment the software simulator started and can be reset at any point. The **Delta** field shows the number of cycles between the lines where the software simulator action started and ended. The user can change the clock value in the **Stopwatch** window, which will cause the required time to be recalculated. Changing clock value does not affect current project settings, but provides accurate simulation.



RAM Window

The software simulator **RAM** window appears by selecting **Debug Windows › RAM Window** from the **View** drop-down menu.



The **RAM** window displays a map of the microcontroller RAM memory, with recently changed items colored red. The value of any field may be changed by double-clicking it.

Software simulator options:

Name	Description	Function key
Start Debugger	Start up software simulator.	[F9]
Run/Pause Debugger	Run or pause software simulator.	[F6]
Stop Debugger	Stop software simulator.	[Ctrl+F2]
Toggle Breakpoint	During the process of simulating, the program executes until a breakpoint has been reached. The <i>Toogle Breakpoints</i> option sets new breakpoints or removes those already set at the current cursor position. The list of all breakpoints may be viewed by selecting the <i>Breakpoints</i> option from the <i>Run</i> menu. To locate a breakpoint, double click on it in the <i>Breakpoints</i> window.	[F5]
Run To Cursor	Execute the program until the cursor position has been reached.	[F4]
Step Into	Execute the current program line, then halts. If the program line executed calls another routine, the software simulator steps into the routine and halts after executing the first instruction within it.	[F7]
Step Over	Execute the current program line, then halts. If the program line executed calls another routine, the software simulator will not step into it. The whole routine will be executed and the software simulator halts at the first instruction following the call.	[F8]
Step Out	Execute all remaining program lines within the subroutine. The software simulator halts immediately upon exiting the subroutine.	[Ctrl+F8]

Keyboard Shortcuts

A complete list of keyboard shortcuts available in the compiler's IDE:

IDE Shortcuts

F1	Help
Ctrl+N	New Unit
Ctrl+O	Open
Ctrl+Shift+O	Open Project
Ctrl+Shift+N	New Project
Ctrl+K	Close Project
Ctrl+F4	Close unit
Ctrl+Shift+E	Edit Project
Ctrl+F9	Build
Shift+F9	Build All
Ctrl+F11	Build and Program
Shift+F4	View breakpoints
Ctrl+Shift+F5	Clear breakpoints
F11	Start mE Programmer
Ctrl+Shift+F11	Project Manager
F12	Options
Alt+X	Close mikroC PRO

Ctrl+E	Incremental Search
Ctrl+L	Routine List
Ctrl+G	Goto line
Ctrl+J	Insert Code Template
Ctrl+Shift+.	Comment Code
Ctrl+Shift+,	Uncomment Code
Ctrl+number	Goto bookmark
Ctrl+Shift+number	Set bookmark
Ctrl+Shift+I	Indent selection
Ctrl+Shift+U	Unindent selection
TAB	Indent selection
Shift+TAB	Unindent selection
Alt+Select	Select columns
Ctrl+Alt+Select	Select columns
Ctrl+Alt+L	Convert Selection to lowercase
Ctrl+Alt+U	Convert Selection to uppercase
Ctrl+Alt+T	Convert to Titlecase
Ctrl+T	USART Terminal
Ctrl+Q	Quick Converter

Basic Editor Shortcuts

F3	Find, Find Next
Shift+F3	Find Previous
Alt+F3	Find in Files
Ctrl+A	Select All
Ctrl+C	Copy
Ctrl+F	Find
Ctrl+R	Replace
Ctrl+P	Print
Ctrl+S	Save unit
Ctrl+Shift+S	Save All
Ctrl+V	Paste
Ctrl+X	Cut
Ctrl+Y	Delete entire line
Ctrl+Z	Undo
Ctrl+Shift+Z	Redo

mikroICD Debugger and Software Simulator Shortcuts

F2	Jump To Interrupt
F4	Run to Cursor
F5	Toggle Breakpoint
F6	Run/Pause Debugger
F7	Step into
F8	Step over
F9	Debug
Ctrl+F2	Stop Debugger
Ctrl+F5	Add to Watch List
Ctrl+F8	Step out
Alt+D	Disassembly view
Shift+F5	Open Watch window
Ctrl+Shift+A	Show Advanced Breakpoints

Advanced Editor Shortcuts

Ctrl+Space	Code Assistant
Ctrl+Shift+Space..	Parameters Assistant
Ctrl+D	Find declaration

DISCLAIMER

All the products owned by MikroElektronika are protected by copyright law and international copyright treaty. Therefore, this manual is to be treated as any other copyright material. No part of this manual, including product and software described herein, may be reproduced, stored in a retrieval system, translated or transmitted in any form or by any means, without the prior written permission of MikroElektronika. The manual PDF edition can be printed for private or local use, but not for distribution. Any modification of this manual is prohibited.

MikroElektronika provides this manual 'as is' without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose.

MikroElektronika shall assume no responsibility or liability for any errors, omissions and inaccuracies that may appear in this manual. In no event shall MikroElektronika, its directors, officers, employees or distributors be liable for any indirect, specific, incidental or consequential damages (including damages for loss of business profits and business information, business interruption or any other pecuniary loss) arising out of the use of this manual or product, even if MikroElektronika has been advised of the possibility of such damages. MikroElektronika reserves the right to change information contained in this manual at any time without prior notice, if necessary.

All the product and corporate names appearing in this manual may or may not be registered trademarks or copyrights of their respective companies, and are only used for identification or explanation and to the owners' benefit, with no intent to infringe.

HIGH RISK ACTIVITIES

The products of MikroElektronika are not fault – tolerant nor designed, manufactured or intended for use or resale as on – line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of Software could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). MikroElektronika and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

Copyright 2003 – 2009 by MikroElektronika. All rights reserved



If you want to learn more about our products, please visit our website: 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 question, comment or business proposal, do not hesitate to contact us: office@mikroe.com