

# PIC Simulator IDE

## Help Topics

### Table Of Contents:

#### [General info](#)

#### [File menu](#)

[Clear Memory](#), [Load Program](#), [Save Memory](#),

#### [Simulation menu](#)

[Start](#), [Step](#), [Stop](#),

#### [Rate menu](#)

[Step By Step](#), [Slow](#), [Normal](#), [Fast](#), [Extremely Fast](#), [Ultimate](#),

#### [Tools menu](#)

[BASIC Compiler](#), [Microcontroller View](#), [Alternative SFR Viewer](#), [Program Memory Editor](#), [EEPROM Memory Editor](#), [Hardware Stack Viewer](#), [Assembler](#), [Disassembler](#), [Breakpoints Manager](#), [Special Breakpoints](#), [Interactive Assembler Editor](#), [8 x LED Board](#), [Keypad Matrix](#), [LCD Module](#), [Graphical 128x64 LCD Module](#), [Stepper Motor Phase Simulation](#), [I2C EEPROM](#), [Hardware UART Simulation Interface](#), [PC's Serial Port Terminal](#), [Software UART Simulation Interface](#), [Oscilloscope](#), [Signal Generator](#), [7-Segment LED Displays Panel](#), [External Modules](#), [Watch Variables](#),

#### [Options menu](#)

[Select Microcontroller](#), [Change Clock Frequency](#), [Configuration Bits](#), [Save Positions](#), [Save Always On Top](#), [Auto Start Options](#), [Reset Simulation Statistics](#), [Change EEPROM Write Time](#), [Change A/D Conversion Time](#), [Change UART Transmit/Receive Time](#), [Change FLASH/EEPROM Unprogrammed Value](#), [Compact Microcontroller View](#), [Infinite Loop Stops Simulation](#), [Basic Program Tracking](#), [Show Confirmation Boxes](#), [Preserve Input States on Simulation Start](#), [Use Voltage for Analog Inputs](#), [Continuous Analog Input Slider Update](#), [Change Ultimate Rate Refresh Interval](#), [Editor Setup](#), [Change Color Theme](#),

#### [Help menu](#)

[Help Topics](#), [BASIC Compiler Reference Manual](#), [External Modules Manual](#), [Check For Updates](#), [Bug Report Interface](#), [About](#), [View License Information](#),

#### [Special notes](#)

#### [The list of simulated peripherals](#)

##### • [General info](#)

PIC Simulator IDE is powerful application that supplies PIC developers with user-friendly graphical development environment for Windows with integrated simulator (emulator), BASIC compiler, assembler, disassembler and debugger.

The main application window shows all PIC microcontroller internal registers status, mnemonics of the last executed instruction, mnemonics of the next instruction that will be executed, clock cycles and instructions counter and real time duration of the simulation.

##### • [File menu](#)

###### - [Clear Memory](#)

This command will reset simulator to the initial state and clear FLASH program memory and EEPROM data memory working buffers.

###### - [Load Program](#)

This command loads program file into PIC Simulator FLASH program memory buffer. The program

file must be in Intel HEX format. Default extension is HEX. If present in the HEX file, configuration word and EEPROM memory data will also be loaded into working buffers. Once loaded program file can be quickly reloaded by clicking on its location field on the main program interface or by keyboard shortcut SPACE.

- [Save Memory](#)

With this command it is possible to save the contents of the FLASH program memory and EEPROM data memory working buffers to a HEX file.

• [Simulation menu](#)

- [Start](#)

PIC Simulator IDE enters simulation mode and begins the execution of instructions starting from the 0000H program memory location.

- [Step](#)

This command is enabled only when Step By Step simulation rate is selected. Next instruction is executed on every click on keyboard shortcut F2.

- [Stop](#)

PIC Simulator IDE exits simulation mode and presents the information about the total number of executed instructions, duration of the simulation and total real time duration of the simulation in clock cycles and  $\mu$ s on PIC microcontroller level.

• [Rate menu](#)

It enables user to change the simulation rate. It is accessible during the simulation, also.

- [Step By Step](#)

The interval between consecutive instructions is at user will. When the simulator is in Step By Step mode, it is possible to change the values in all special function and general purpose registers, program counter and working register, by clicking on the appropriate name or value field on the program interface. The value in the SFR registers can be changed alternatively by toggling individual bits of the graphical representation and this feature can be used also with other simulation rates. When this simulation rate is selected new main menu item 'STEP' will appear on the program interface. That will enable an easy access to Step command from the Simulation menu. Additional advantage is available for programs generated by integrated Basic compiler. 'Run To Next BASIC Statement' command will start Extremely Fast simulation until the next Basic statement is reached and then automatically switch to Step By Step mode.

- [Slow](#)

The interval is 1500 ms.

- [Normal](#)

The interval is 250 ms.

- [Fast](#)

The interval is around 50 ms.

- [Extremely Fast](#)

The interval is very short and is linearly dependent on the overall computer performance.

- [Ultimate](#)

The main simulator window is not continuously refreshed after every simulated instruction that significantly improve the simulation performance. The refresh interval can be changed using Change Ultimate Rate Refresh Interval command in Options menu.

• [Tools menu](#)

- [BASIC Compiler](#)

Integrated BASIC compiler editor window will be opened. More information available in BASIC Compiler Reference Manual. It can be accessed from the Help menu of the main application window or from the BASIC compiler editor window.

- [Microcontroller View](#)

This command will open the window with the selected PIC microcontroller pinout. Logical states on all I/O pins is graphically displayed, and can be manually changed on the input pins by clicking on the appropriate toggle buttons. It is possible to change analog values on all I/O pins configured as analog inputs. If this window is opened, it will be refreshed during the simulation.

- [Alternative SFR Viewer](#)

This command opens an alternative viewer window for special function registers that can be resized.

- [Program Memory Editor](#)

This is access to PIC Simulator FLASH program memory buffer editor. It is possible to manually

enter instruction opcode by clicking on the display line of the target location.

#### - EEPROM Memory Editor

This opens PIC Simulator data EEPROM memory buffer editor. If this window is opened, it will be refreshed during the simulation. The value in the specific data memory location can be changed by clicking on it.

#### - Hardware Stack Viewer

This command will open the PIC Simulator hardware stack buffer viewer. If this window is opened, it will be refreshed during the simulation.

#### - Assembler

This command starts integrated assembler. Assembler source files can be edited and assembled in the same graphical environment. Default extension is ASM. After the successful assembly process two new files are generated. One with HEX extension that is program file in Intel HEX format and that can be loaded into PIC program memory and the other with LST extension that is assembler output listing. This assembler is excellent solution for assembling source files with size up to 20K. For larger files the assembly process can take some time, but there are no limits in file size. Its limits are that only ORG (.ORG), EQU (.EQU), DB (.DB, DEFB, .DEFB), DW (.DW, DEFW, .DEFW) and END (.END) assembler directives are supported. Configuration word and EEPROM memory buffers are also saved in the generated HEX file. These buffers can be accessed and edited from the Options menu of the Assembler window. DCONF (.DCONF, DEFCONF, .DEFCONF) directive can be used to set the value(s) that will override the defaults.

#### - Disassembler

PIC Simulator IDE has internal disassembler that is started by this command. The disassembling process is automatically initiated by opening this window. The disassembler will always start from the address 0000H. After the operation is completed disassembler will display the output listing file. The generated listing can be saved to disk. User will be prompted to enter the name for the output file. Default extension is LST.

#### - Breakpoints Manager

This command starts integrated debugger that can be used to debug and monitor the program execution. The debugger listing file of the program in memory can be generated using internal assembler listing file (if it exists) or by internal disassembler. It is possible to define up to 10 breakpoints by clicking on individual lines in the loaded program listing. When the simulation starts in faster rate modes it will automatically switch to Step By Step mode when reaching any of these breakpoints. The breakpoints are marked by red circles, and the current value of the PC register is marked by yellow arrow. There is an option to keep the PC pointer in focus during the simulation.

#### - Special Breakpoints

This simulation tool provides the feature to define special breakpoints that will switch the simulation rate to Step By Step mode when the value of the predefined register has changed or reached the predefined value. Up to 5 special breakpoints can be defined to be simultaneously active.

#### - Interactive Assembler Editor

This command starts integrated graphical tool that will enable beginners to write their first assembler routines interactively without having to memorize the mnemonics of individual instructions from PICmicro instruction set. This is a great tool for educational purposes.

#### - 8 x LED Board

This simple simulation module can be used to attach up to eight LEDs to microcontroller pins. The color of each diode can be changed by clicking on the LED color field and the pin assignment can be changed by clicking on the label showing the current pin selection.

#### - Keypad Matrix

This is another simple simulation module for up to 4x4 keypad matrix. The pin assignments can be changed by clicking on the appropriate pin labels.

#### - LCD Module

This command starts integrated LCD Module simulator. Before it can be used for simulation, user should set up the interface parameters from the Setup dialog.

#### - Graphical 128x64 LCD Module

This command starts integrated graphical 128x64 LCD Module simulator. Before it can be used for simulation, user should set up the interface parameters from the Setup dialog.

#### - Stepper Motor Phase Simulation

This simulation tool will show simplified (2-pole) graphical presentation of a unipolar stepper motor phases in both full-step and half-step modes. During the simulation Step Counter field will display the current absolute rotor position measured in steps from the first well-defined rotor position.

#### - I2C EEPROM

This command starts integrated simulation module for external I2C EEPROMs from 24C family.

#### - [Hardware UART Simulation Interface](#)

This command opens interface for hardware UART simulator.

#### - [PC's Serial Port Terminal](#)

This is an integrated tool that is independent from the simulator. This terminal is connected to PC's serial port and can be used for communication with a real PIC device for the purpose of serial communication routines testing. Comm port number and baud rate can be set using appropriate commands from the menu.

#### - [Software UART Simulation Interface](#)

This command opens interface for software UART simulator. It is a serial device that can communicate with PIC microcontroller running software implemented UART routines (SERIN, SERININV, SEROUT and SEROUTINV statements) using real-time serial communication simulated on RX and TX pins.

#### - [Oscilloscope](#)

This is a very useful integrated tool for tracking logic levels on microcontroller pins during the simulation. It is a four channel digital oscilloscope. User can assign target pins to the oscilloscope channels and change the length of the display interval using commands from the Settings menu. Input and output pins are painted with different colors that are user selected. If Pull-up option is selected for an oscilloscope channel, upon transition from output to input pin state, the pin will be set to high default logic level.

#### - [Signal Generator](#)

With this simulation tool user can define up to four independent continuous pulses generators with selectable target input pins, pulse periods and duty cycles. Suitable animation will show the current phase of each pulse generation.

#### - [7-Segment LED Displays Panel](#)

This integrated simulation tool enables user to define connection with up to four 7-segment LED displays. The connection definition includes pin assignments for all display segments and one enable line to control display selection when more than one display is used with parallel connection of segments - multiplexing. Active levels for all connection lines can be inverted to suit hardware requirements. There is also an option to change the color used to paint the active LEDs on the displays and Keep Last Display option to simulate slow eye response effect for programs that use multiplexing displays.

#### - [External Modules](#)

This tool should be used to establish automation interface with up to five external client/server modules. It is required to enter the class name supplied by external device in the form ApplicationName.ObjectName in order to establish connection with it. External client/server applications will be started and terminated automatically with PIC Simulator IDE. More information available in External Modules Manual. It can be accessed from the Help menu of the main application window.

#### - [Watch Variables](#)

During the simulation of programs written using integrated basic compiler, this tool can be used to watch the current values of all variables declared in the simulated basic program. It is also possible to add user defined variables to the list to monitor other memory locations of interest during the simulation. This feature is useful for memory monitoring for simulated program files, that are not compiled with the integrated basic compiler. User added variables will be remembered between sessions as long as the same program file is loaded in the simulator. Variables from the watch list can be easily removed with the Delete Variable command, so the list can contain variables of special interest only. Other commands and options include: Change Variable Value (can be also started by a single-click on the variable from the list), Display HEX Values, Confirm Delete.

### • [Options menu](#)

#### - [Select Microcontroller](#)

This command is used to change the PIC microcontroller model selection for the simulation. It will reset the application to the initial state.

#### - [Change Clock Frequency](#)

This command allows user to change the frequency parameter that is used for the calculation of the real time duration of the simulation. The entered value in MHz is remembered for the future sessions. This parameter is also used by a number of statements in BASIC compiler that use some form of timing routines (WaitMs, WaitUs, Serin, Serout, ...). The default value is 4 MHz.

#### - [Configuration Bits](#)

This option should be used to change the default settings for the configuration bits. That information will be saved in all HEX files generated by integrated assembler. This option is also available in the

menu of the assembler window.

- [Save Positions](#)

With this option selected, the positions of the windows on the screen will be remembered.

- [Save Always On Top](#)

With this option selected, the Always On Top setting for all of the windows with this feature will be remembered.

- [Auto Start Options](#)

With this utility users can define actions that will be performed on the application startup. These actions include automatic opening of various tools and simulation interfaces from the Tools menu and automatic loading of the last used files in the simulator, assembler and basic compiler.

- [Reset Simulation Statistics](#)

This command will set to zero clock cycles counter, instructions counter and real time duration of the simulation. This command can be used to determine the duration of the specific portion of simulated program.

- [Change EEPROM Write Time](#)

This command is used to change the number of clock cycles that will be used for the EEPROM write interval. The realistic value is around 5ms. The default value is 1000 clock cycles.

- [Change A/D Conversion Time](#)

This command is used to change the number of clock cycles that will be used for the A/D conversion time. The setting in the ADCON0 register is ignored by the simulator. The default value is 100 clock cycles.

- [Change UART Transmit/Receive Time](#)

This command is used to change the number of clock cycles that will be used for the UART transmit/receive time by internal hardware UART simulator. The realistic value depends on the defined baud rate. The default value is 1000 clock cycles.

- [Change FLASH/EEPROM Unprogrammed Value](#)

This command is used to change the unprogrammed bit values for FLASH program memory and data EEPROM memory from 1 to 0, and vice versa from 0 to 1. If unprogrammed bit value is 1 then unprogrammed FLASH location will contain 3FFFh and unprogrammed EEPROM location will contain FFh. If unprogrammed bit value is 0 then all unprogrammed memory will be filled with zeros.

- [Compact Microcontroller View](#)

If this option is turned on Microcontroller View windows will be displayed in more compact form.

- [Infinite Loop Stops Simulation](#)

Checking this option will force the simulator to automatically stop the simulation when an infinite loop is encountered.

- [Basic Program Tracking](#)

Currently simulated basic statement will appear highlighted in Basic compiler window when this option is enabled.

- [Show Confirmation Boxes](#)

When this option is enabled the confirmation boxes showing results of operations will be displayed and will require user response to be closed.

- [Preserve Input States on Simulation Start](#)

If this option is enabled then the states of digital and analog inputs on the Microcontroller View window will not be reset to OFF digital state and zero analog value when the simulation is started again.

- [Use Voltage for Analog Inputs](#)

Use this option if you prefer to see voltage value (0.00V-5.00V) instead of raw analog value (0-1023) for analog input states in Microcontroller View windows.

- [Continuous Analog Input Slider Update](#)

If this option is enabled then the analog input values will be continuously updated with scrolling of the analog input slider in the Microcontroller View window. Otherwise, the update will occur on the closing of analog slider pop-up only.

- [Change Ultimate Rate Refresh Interval](#)

This command allows user to change the refresh interval (in milliseconds) for the main simulation interface when the simulation is running at Ultimate rate. Its value however does not affect the simulation performance considerably. The default value is 500ms.

- [Editor Setup](#)

With this setup tool it is possible to change various properties of basic compiler and assembler code editors.

- [Change Color Theme](#)

This command will open a dialog with the rich list of available color themes, so that user can change application appearance.

- [Help menu](#)

- [Help Topics](#)

This command will display Help Topics. This help file contains general information about the application with description of all menu items.

Help viewer window features navigation panel showing topics and subtopics of the displayed help file. Right-click on the navigation panel will show popup menu with Show All Subtopics and Hide All Subtopics commands. Single-click on the item from the navigation panel will move focus on the display panel to the appropriate position. Double-click on the topic item will show/hide its subtopics. The display panel shows the content of the loaded help file. Right-click will display popup menu containing various options and commands including: Copy, Copy RTF, Copy HTML, Print, Font Increase, Font Decrease, Font Reset, Always On Top. Help viewer window is resizable and will remember both its position and size. The vertical separator between navigation and display panels is moveable and its position will also be saved after the viewer is closed.

- [BASIC Compiler Reference Manual](#)

BASIC Compiler Reference Manual will be displayed in the help viewer.

- [External Modules Manual](#)

External Modules Manual will be displayed in the help viewer.

- [Check For Updates](#)

This tool will enable user to establish connection with OshonSoft.com web site to check out if there is a new software release available for download. Version log file will be displayed after the response from the web site has been received.

- [Bug Report Interface](#)

This interface should be used to send the reports about possible bugs in software to OshonSoft.com. In addition to the user written part the full report will contain a part that is generated by software (system report).

- [About](#)

This command will display the basic information about the software package.

- [View License Information](#)

This command will display the information about the installed license for the software. Info button will display information about the add-on modules enabled by the license.

- [Special notes](#)

- Watchdog Timer is not simulated (CLRWDT instruction is executed as NOP)
  - Power-down Mode is not simulated (SLEEP instruction will stop the simulation)

- [The list of simulated peripherals](#)

- Digital I/O
  - Data EEPROM Memory
  - Interrupts
  - Timer0 Module
  - A/D Converter Module
  - Comparator Module
  - Voltage Reference Module
  - Hardware UART Module
  - Timer1 Module (Bits T1OSCEN and \T1SYNC in T1CON are ignored.)
  - Timer2 Module
  - Capture, Compare and PWM modes of CCP modules
  - Reading FLASH program memory