



# Overview and Features

## **Overview:**

This integrated C development environment gives developers the capability to quickly produce very efficient code from an easily maintainable high level language.

The compiler includes built-in functions to access the PICmicro® hardware such as `READ_ADC` to read a value from the A/D converter. Discrete I/O is handled by describing the port characteristics in a `PRAGMA`. Functions such as `INPUT` and `OUTPUT_HIGH` will properly maintain the tri-state registers. Variables including structures may be directly mapped to memory such as I/O ports to best represent the hardware structure in C.

The microcontroller clock speed may be specified in a `PRAGMA` to permit built in functions to delay for a given number of microseconds or milliseconds. Serial I/O functions allow standard functions such as `GETC` and `PRINTF` to be used for RS-232 like I/O.

The hardware serial transceiver is used for applicable parts when possible. For all other cases a software serial transceiver is generated by the compiler. The standard C operators and the special built in functions are optimized to produce very efficient code for the bit and I/O functions.

Functions may be implemented inline or separate, allowing to optimize for either ROM concerns or speed concerns. Function parameters are passed in reusable registers. Inline functions with reference parameters are implemented efficiently with no memory overhead.

During the linking process the program structure, including the call tree, is analyzed. Functions that call one another frequently are grouped together in the same page segment. Calls across pages are handled automatically by the tool transparent to the user. Functions may be implemented inline or separate. RAM is allocated efficiently by using the call tree to determine how locations can be re-used. Constant strings and tables are saved in the device ROM.

The output hex and debug files are select-able and compatible with popular emulators and programmers including MPLAB® IDE for source level debugging. PCW includes a powerful Windows IDE. The compiler requires Windows 95, 98, ME, NT4, 2000 or XP.

## **Features**

Built in libraries that work with all chips for RS232 serial I/O, I2C, discrete I/O and precision delays.

Integrates with MPLAB IDE and other simulators and editors for source level debugging. Standard HEX file and debug files ensure compatibility with all programmers.

Formatted printf allows easy formatting and display in HEX or decimal.

Efficient function implementation allows call trees deeper than the hardware stack.

Source code drivers included for LCD modules, keypads, 24xx and 94xx serial EEPROM's, X10, DS1302 and NJU6355 real time clocks, Dallas touch memory devices, DS2223 and PCF8570 serial SRAM, LTC1298 and PCF8591 A/D converters, temperature sensors, digital pots, I/O expander and much more.

Access to hardware features from easy to use C functions, timers, A/D, EEPROM, SSP, PSP, USB, I2C and more.

1, 8, 16 and 32 bit integer types and 32 bit floating point.

Assembly code may be inserted anywhere in the source and may reference C variables.

Automatic linking handles multiple code pages.

Inline functions supported to save stack space; Linker will automatically determine the best architecture or it can be manually specified.

Compiler directives determine if tri-state registers are refreshed on every I/O or if the I/O is as fast as possible.

Constants (including strings and arrays) are saved in program memory.

Standard one bit type (Short Int) permits the compiler to generate very efficient Bit orientated code.

#BIT and #BYTE will allow C variables to be placed at absolute addresses to map registers to C variables.

Reference parameters may be used to improve code readability and inline function efficiency.

PCW has both an integrated editor/compiler and command line compiler.

Special windows show the RAM memory map, C/Assembly listing, and the calling tree.

Interrupt functions supported on PCM/PCH. The compiler generates all startup and clean up code as well as identifying the correct interrupt function to be called.

Plenty of ready to run example programs included.

Updates via the internet for 30 days included.