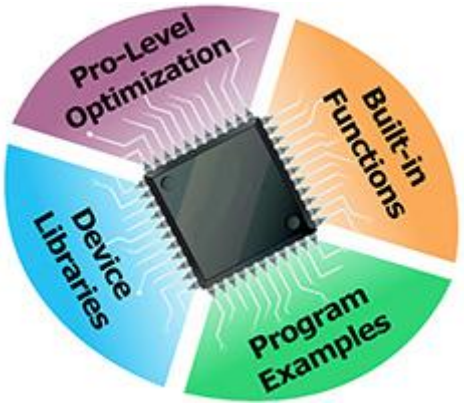
Custom Computer Services, Inc.

# Compiler Features


CCS C Compilers are designed specifically for the PIC® MCU architecture, unlike competitive compilers based on a GNU or common engine with a generic code generator. Every aspect of the CCS C Compiler is specially optimized for the PIC® MCU.

The **Pro-Level Optimization** of our PIC® compilers include Standard C Constructs, numerous Pre-Processor Functions, and the largest library of **Built-In Functions**. This provides developers with unique access to device hardware features at the embedded C language level. C syntax and special functions have a uniform syntax across all chip families, allowing for migration to a new chip trivial.**Ready-to-Run Program Examples**, and **Device Libraries**, empower rapid development of applications incorporating leading edge technologies such as capacitive touch, wired and wireless communications, motion and motor controls, and energy management.




All of our compilers are **ANSI C Compliant** with ***Pro-Level Optimization***. We offer flexible software configurations to fit any project requirements. Compilers are compatible with many third-party tools such as **Microchip MPLAB® and MPLAB® X**.

# Pro-Level Optimization

**Include Files**

Device specific include files contain all the information the compiler needs to optimize code generation for the specific PIC® MCU.

- Instruction memory size
- Data memory size
- Pin functionality
- Memory banking
- Peripheral resources
- Hardware stack size

**String Optimization**

- **7-bit ASCII String Compression** - Decrease system usage through improved string compression
- **Switch Statements** - Easily perform string comparisons which result in tighter and more maintainable source code and a smaller ROM footprint
- **Variable Length Constant Strings**
- ***printf*** - Reduce usage of multiple string output through use of this function

**Efficient Data Structures Mapped into Program Memory**

Flexible constant data structure handling allows the compiler to handle lookup tables that are virtually unlimited in size. This is of particular interest to developers using large lookup tables for trigonometric functions or storing FPGA configuration memory images in on-chip MCU memory. Constants (including strings and arrays) are saved in program memory. DSP performance can be enhanced by manually assigning variables to data spaces for faster access with pre-processor directives.

**Additional Features Include:**

- **#opt compress** - Optimize for space instead of speed with up to 60% reduction of program memory
- **Efficient Function Management** - Allows for call trees to be deeper than the hardware stack
- **Automatic Linking** - Practical handling of multiple code files
- **In-line Assembly** - Insert assembly code anywhere in the source and reference C variables
- **Function Overloading** - Allows for several functions with the same name with differences in number and types of parameters
- **Default Parameters** - These values are used when arguments are not passed in
- **Variable Number of Parameters** - Define functions which allow for any number of variables
- ***Short Int*** - Permits the compiler to generate very efficient bit oriented code
- **Intelligent Interrupt Handling**
- **Automatic Fuse Configuration**
- **Function Recursion for PIC24 and dsPIC® DSC Devices**
- **Relocatable Objects / Multiple Compilation Unit** (IDE Only)

# Largest Library of Built-In Functions

Our compilers contain hundreds of built-in functions that simplify access to hardware while producing efficient and highly optimized code.

**Device Peripheral Access:**

- $I^2C$ and SPI
- A/D converters
- On-chip data EEPROM
- LCD controllers
- Processor controls
- Capacitive touch
- Memory Access
- Real-time Clock

**External Memory Access:**

- Create User Defined Address Spaces
- Generate Multiple HEX Files for Applications

**Advantages:**

- Connect to a number of busses quickly
- Easily setup Capture/Compare/PWM Peripherals
- Specify microcontroller clock speed
- Generate precision delays in microseconds & milliseconds
- Setup and read values from on-chip A/D converters
- PLL and power saving options
- Swiftly integrate Capacitive Touch into projects
- Drive internal PIC® LCD controllers
- Maintain tri-state registers
- Map PIC® registers to C variables
- Interface codec chips to dsPIC® DSCs

## Powerful PIC® MCU Specific Device Libraries

Why "recreate the wheel" coding your own functions in assembly or C? Many libraries are supplied with our CCS C Compiler products to aid in your embedded development.

**Included Drivers:**

- LCD Modules
- Keypads
- Serial EEPROM
- Real-time Clocks
- Touch Sensors
- Memory Devices
- A/D Converters
- Temperature Sensors
- Digital Potentiometers
- I/O Expanders
- & more...

**Included Standard C Math Libraries:**

- 1/8/16/32-bit Integers and 32-bit Floating Point Support - All Devices
- 48/64-bit Integers and 64-bit Floating Point Support - PIC24 and dsPIC® DSC Devices
- Fixed Point Decimal - Provides decimal representation at integer speed
- & more...

# Ready-to-Run Program Examples

The CCS C compiler includes a library of example programs for many common applications. Each example program contains a header with instructions on how to run the example, and if necessary, the wiring instructions for interfacing external devices.

Some of the examples included:

- **FAT** - Access/read/write files on a SD/MMC card that has a FAT file system. Run a long term log on your PIC® MCU, saving each entry by appending to a file. Then read the results on your PC by opening the file.
- **SIM/SMART Card** - Access the contact and phone number information on a SIM/SMART card, commonly found on GSM/GPRS cell phones.
- **Frequency Generator** - Generate tones, frequencies and DTMF using only one user define general purpose I/O pin. This is done by implementing a software PWM on one pin.
- **XTEA Cipher Library** - Transmit encrypted data between two PIC® MCUs using XTEA Encryption.
- **XML Parser** - Parse XML documents using the minimal resources of a PIC® MCU.