Ready...

- For embedded software tools designed EXCLUSIVELY for PIC® MCUs
- For a complete, low-cost, powerful C solution
- For ready-to-run example programs and tested peripheral drivers to start any project

Set...

- Using a C Compiler with IDE and C Aware Real-time Debugger
- Using In-Circuit Debugger and Programmers
- Using Prototyping Boards or complete Development Kits

GO!...

CCS Inc

www.ccsinfo.com
sales@ccsinfo.com
Phone: 262.522.6500
Sales x35
Tech Support x32

PIC® MCU and dsPIC®DSC are registered trademarks of Microchip Technology, Inc.

Downloaded from Elcodis.com electronic components distributor

CCS C Compiler

Features

The compiler is comprised with Standard C operators and built-in libraries that are specific to PIC ® MCU registers, and access to hardware features from C.

PIC10 / PIC12 / PIC14 / PIC16 / PIC18	Advanced Features in PIC24 & dsPIC® DSCs
1, 8, 16, 32-bit integer types & 32-bit floating point	Also 48 & 64-bit floating point make calculations requiring greater precision or broader range easier
Bit Arrays and Fixed Point Decimals	
#BIT and #BYTE will allow C variables to be placed at absolute addresses to map registers to C variables	#BIT, #BYTE and #WORD will allow C variables to be placed at absolute addresses to map registers
Standard one-bit type (Short Int) permits the compiler to generate very efficient Bit-oriented code	
Constants (including strings and arrays) are saved in program memory	Constants in ROM
Flexible Handling of Constant Data	Enhanced oscillator control to choose from a multitude of clock sources, PLL and power saving options
Variable length Constant Strings	
AddressMod capability to create user defined address spaces in memory device	Function recursion allows for interactive processing algorithms

Processor & Peripheral Controls

The CCS C Compiler for PIC10, PIC12, PIC14, PIC16, PIC18 and PIC24 microcontrollers has over 180 Built-in-Functions to access PIC® MCU hardware is easy and produces efficient and highly optimized code.

Functions such as timers, A/D, EEPROM, SSP, PSP, USB, I2C and more:

- Built-in libraries that work with all chips for RS-232 serial I/O, I²C, discrete I/O and precision delays
- Serial I/O functions allow standard functions such as GETC() and PRINTF() to be used for RS-232 like I/O
- Formatted printf allows for easy formatting and display in HEX or decimal
- Multiple I²C and RS232 ports may be easily defined
- #use rs232() offers options to specify a maximum wait time for getc
- Hardware transceiver used when possible, but for all other occasions the compiler generates a software serial transceiver
- Microcontroller clock speed may be specified in a PRAGMA to permit built-in functions to delay for a given number of microseconds or milliseconds
- Functions such as INPUT() and OUTPUT_HIGH() properly maintain the tri-state registers
- Compiler directives determine if tri-state registers are refreshed on every I/O or if the I/O is as fast as possible
- #USE SPI ()
- Simple functions like READ_ADC() to read a value from A/D converter
- Source code drivers included for LCD modules, keypads, 24xx and 94xx serial EEPROM, X10, DS1302 and NJU6355 real time clocks, Dallas touch memory devices, DS2223 and PCF8570, LTC1298 and PCF8591 A/D converters, temperature sensors, digital pots, I/O expander and much more

Advanced Functions

The compiler can handle inline or separate functions, as well as parameter passing in re-usable registers.

Transparent to the user, the compiler handles calls across pages automatically and analyzes program structure and call tree processes to optimize RAM and ROM Usage.

Additional features include:

- Efficient function implementation allow call trees deeper than the hardware stack
- Automatic linking handles multiple code pages
- Assembly code may be inserted anywhere in the source and may reference C variables
- Function Overloading allows for several functions with the same name, but differences in number and type of parameters
- Default Parameters can be used in a function if arguments are not used in a call
- Interrupt functions supported on PCM/PCH. The compiler generates all startup and clean up code as well as identifying the correct function to be called
- Reference parameters may be used to improve code readability and inline function efficiency
- Generation Of Multiple HEX Files For Chips With External Memory
- Variable Number Of Parameters in a function
- Relocatable Objects / Multiple Compilation Unit (IDE Only)
- Automatic #fuses Configuration

Downloaded from Elcodis.com electronic components distributor

Complete Example Programs

LCD	Frequency counter	Fixed Point	DTMF Tones	Boot Loader
A/D	7 Seg LED	TCP/IP	CRC Calculator	CAN Bus
PWM	Data Logger	Floating Point	CCP	I/O for 8-in Parts
Comparator	Pattern Generator	ICD Debugging	Watchdog Timer	Sleep
PSP	Stepper Motors	Advanced Macros	Analog Comparator	Timers
Serial Interrupts	Tone Generation	Memory Management	Optical Encoder	
Magnetic Card Reader	Temperature Sensor	I2C	USB	

Included C Driver:

Serial EEPROM/Flash
- 2041
- 24xx
- 25xx
- 93xx
- AT2421
- AT25256
- AT29C1024
- AT45DB021
- CE51x
- CE62x
- CE67x
- 9512
- MMC/SD
- DS2?? (1-wire EEPROM)

Robotics
- GP2D12
- Line Tracking Sensors

A/D & D/A Converters
- AD7705AD7715
- ADS8320
- LTC1298
- MAX517
- MCP4921
- MCP3204
- MCP3208
- TLC545N

Digital Pots
- AD8400
- DS1868
- MCP41010

USB
- USBN960x
- PIC_USB
- PIC18_USB

Real-Time Clock
- DS1302
- NJU6355
- DS1305
- ISL1209

Sounds
- WTS701
- TONES
- ISD4003

RFID
- EM4095
- EM4402
- EM4150

Accelerometer
- ADXL210

LCD
- GLCD
- KS0108
- LCD
- LCD420
- SED1335
- HDM64GS12

Temperature
- DS1621
- DS162M
- DS1631
- DS1624
- LM75CIM3

CAN Functionality
- MCP251x
- 8xxx8
- 18F4580

Expanded Input/Output
- 74165
- 74595
- MAX7300
- SC28L19x

Serial RAM
- 68HC68R1
- 68HC68R2
- M68AF031
- PCF8570
- D41256
- MT4264

Networking/Internet
- TCP
- PPP
- S7600
- RTL8019
- ENC28J60

Other
- Digital Compass
- Keypad
- Mag Card REader
- PLL Inteface
- Dallas One Wire
- IR Decoder
- Line Tracker
- Servo Control
- X10
- Cyclic Redundancy Code
- RS485
- N9085UD
- PNI11096
- LMX2326

Seconds Counter

```
#include <18F4520.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
#use_delay(clock=20000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)

#define INTS_PER_SECOND 76      // (20000000/(4*256*256))

BYTE seconds;                   // A running seconds counter
BYTE int_count:                 // Number of interrupts left before a
                                // second has elapsed

#int_rtcc                       // This function is called every time
void clock_isr() {              // the RTCC (timer0) overflows (255->0).
                                // For this program this is
     if(--int_count=0) {        // apx 76 times per second.
          ++seconds;
          int_count=INTS_PER_SECOND;
     }
}

void main() {
     BYTE start:

     int_count=INTS_PER_SECOND;
     set_timer0(0);
     setup_coutner(RTCC_INTERNAL, RTCC_DIV_256 | RTCC_*BIT);
     enable_interrupts(INT_RTCC);
     enable_interrupts(GLOBAL);

     {
          printf("Press any key to begin.\n\r");
          getc();
          start=seconds;
          printf("Press any key to stop.\n\r");
          getc();
          printf("%u seconds.\n\r",seconds-start);
     }while (TRUE);
}
```

Simple A/D

```
#include<16F877a.h>
#fuses HS,NOLVP,NOWDT,PUT
#use delay(clock=20000000)
#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7)

void main() {
     int i, value, min, max;

     printf("Sampling:");
     setup_adc_ports(RA0_ANALOG);
     setup_adc(ADC_CLOCK_INTERNAL);
     set_adc_channel(0);
     do {                          //Takes 30 samples from
          min=255;                 //pin A0 and displays the min and
          max=0;                   //max values for that 100ms peroid
          for(i=0; I <= 30; ++i) {
               delay_ms(100);
               value=read_adc();
               if(value < min)
                min = value:
               if(value > max)
                max = value;
          }
          printf("\n\rMin:%x MAX: %x", min, max);
     } while (TRUE);
}
```

Downloaded from Elcodis.com electronic components distributor

Specific Features

*PCD and PCWHD Built-in-Functions

Built-in Functions

RS-232
getc( )
putc( )
gets( )
puts( )
printf( )
kbhit( )
assert( )
fgetc( )
fgets( )
fprintf( )
fputc( )
fputs( )
getch( )
getchar( )
perror( )
putchar( )
setup_uart( )
set_uart_speed( )*

SPI TWO-WIRE I/O
setup_spi( )
setup_spi2( )
spi_read( )
spi_read2( )
spi_write( )
spi_write2( )
spi_data_is_in( )
spi_data_in2( )
spi_xfer( )

VOLTAGE REF
setup_vref( )
setup_low_volt_detect( )
setup_comparator( )*

A/D CONVERSION
setup_adc_port( )
setup_adc( )
set_adc_channel( )
read_adc_channel( )
adc_done( )
setup_adc_port( )*
setup_adc( )*
set_adc_channel( )*
read_adc( )*
adc_done( )*

CYCLILC REDUNDANCY CHECK
crc_calc( )*
crc_init( )*
crc_psuedo_code( )*
help_crc( )*
setup_crc( )*

STANDARD C SPECIAL
bsearch( )
rand( )
srand( )
qsort( )

DIRECT MEMORY ACCESS
dma_start( )*
dma_status( )*
setup_dma( )*

I²C
i2c_start( )
i2c_stop( )
i2c_read( )
i2c_write( )
i2c_poll( )
i2c_isr_state( )
i2c_slaveaddr( )

PARALLEL SLAVE I/O
setup_psp( )
psp_input_full( )
psp_output_full( )
psp_overflow( )

abs( )
acos( )
asin( )
atan( )
atan2( )
ceil( )
cos( )
exp( )
floor( )
labs( )
log( )
log10( )
sin( )
sqrt( )
tan( )
cosh( )
div( )
fabs( )
fmod( )
frexp( )
ldexp( )
ldiv( )
modf( )
pow( )
sinh( )
tanh( )

STANDARD C MEMORY
memset( )
memcpy( )
calloc( )
free( )
longjmp( )
malloc( )
memcmp( )
memmove( )
offsetof( )
offsetofbit( )
realloc( )
setjmp( )

MOTOR PWM
get_motor_pwm_event( )*
set_motor_pwm_event( )*
setup_motor_pwm( )*
setup_motor_pwm_duty( )*
setup_motor_unit( )*

DISCRETE I/O
output_low( )
output_high( )
output_float( )
output_bit( )
input( )
output_x( )
input_x( )
port_b_pullups( )
set_trix_x( )
get_tris_s( )
output_drive( )
input_state( )
port_a_pullups( )

DELAYS
delay_cycles( )
delay_us( )
delay_ms( )

CAPTURE/COMPARE PWM
setup_ccp_x( )
set_pwmX_duty( )
set_power_pwm_override( )
set_power_pwmx_duty( )
setup_power_pwm( )
setup_power_pwm_pins( )
setup_capture( )*
get_capture( )*
set_compare_time( )*

PROCESSOR CONTROLS
sleep( )
reset_cpu( )
restart_cause( )
disable_interrupts( )
enable_interrupts( )
ext_int_edge( )
read_bank( )
interrupt_active( )
getenv( )
setup_oscillator( )
clear_interrupt( )
goto_address( )
jump_to_isr( )
lavel_address( )

REAL TIME CLOCK
rtc_alarm_read( )*
rtc_alarm_write( )*
rtc_read( )*
rtc_write( )*
setup_rtc( )*
setup_rtc _alarm( )*

QUADRATURE ENCODER INTERFACE
qui_get_count( )*
qui_set_count( )*
qui_status( )*
setup_qui( )*

ANALOG COMPARE
setup_comparator( )

LCD
lcd_load( )
lcd_symbol( )
setup_lcd( )

TIMERS
setup_timer x( )
set_timer_x( )
get_timer_x( )
setup_counters( )
setup_wdt( )
restart_wdt( )
set_timer_xy( )*
get_timer_xy( )*

BIT MANIPULATION
shift_right( )
shift_left( )
rotate_right( )
rotate_left( )
bit_clear( )
bit_set( )
bit_test( )
swap( )
make8( )
make16( )
make32( )
_mul( )

RTOS
rtos_await( )
rtos_enable( )
rtos_msg_poll( )
rtos_msg_read( )
rtos_msg_send( )
rtos_overrun( )
rtos_run( )
rtos_signal( )
rtos_stats( )
rtos_terminate( )
rtos_wait( )
rtos_yield( )
(RTOS only in PCW
and PCWH packages)

INTERNAL EEPROM
read_eeprom( )
write_eeprom( )
read_program_eeprom( )
write_program_eeprom( )
read_calibration( )
erase_program_eeprom( )
read_external_memory( )
read_program_memory( )
setup_external_memory( )
write_configuration_memory( )
write_external_memory( )
write_program_memory( )
erase_program_memory( )*
read_rom_memory( )*
read_configuration_memory( )*

STANDARD C CHAR
atoi( )
atol( )
atoi32( )
atof( )
tolower( )
toupper( )
isalnum( )
isalpha( )
ismoung( )
isdigit( )
islower( )
isspace( )
isupper( )
isxdigit( )
iscntrl( )
isgraph( )
isprint( )
ispunct( )
itoa( )
strlen( )
strncpy( )
strcmp( )
stricmp( )
strncmp( )
strncmp( )
strcat( )
strstr( )
strchr( )
strrchr( )
strtok( )
strspn( )
strpbrk( )
strlwr( )
sprintf( )
strcoll( )
strncat( )
strtod( )
strtol( )
strtoul( )
strxfrm( )

PARALLEL MASTER PORT
pmp_overflow_input( )
pmp_overflow_output( )
pmp_address( )
pmp_input_full( )
pmp_output_full( )
pmp_overflow( )
pmp_read( )
pmp_write( )
psp_input_full( )
psp_overflow( )
psp_read( )
psp_write( )
setup_pmp( )
setup_psp( )

Downloaded from Elcodis.com electronic components distributor

Preprocessors

STANDARD C
- #define
- #else
- #elif
- #endif
- #error
- #if
- #ifdef
- #include
- #pragma
- #undef
- #ifndef
- #list
- #nolist

FUNCTION QUALIFIERS
- #inline
- #int_default
- #int_global
- #int_xxx
- #separate

DEVICE SPECIFICATION
- #device chip
- #fuses
- #id
- #id checksum
- #id number
- #serialize
- #hexcomment

RTOS
- #use rtos
- #task
- (RTOS only in PCW
- and PCWH packages)

BUILT-IN LIBRARIES
- #use delay
- #use fast_io
- #use fixed_io
- #use i2c
- #use rs232
- #use standard_io
- #use spi

PRE-DEFINED IDENTIFIERS
- _date_
- _device_
- _file_
- _line_
- _pcb_
- _pcm_
- _pch_
- _time_
- _filename_

LINKER
- #import
- #export
- #build

MEMORY CONTROL
- #asm
- #bit
- #byte
- #endasm
- #fill_rom
- #locate
- #reserve
- #rom
- #zero_ram
- #org
- #type
- #word

COMPILER CONTROL
- #case
- #opt
- #priority
- #ignore_warnings
- #export
- #import
- #module

Example C/ASM Listing

```
...................done=FALSE;
09C: BCF 3B, 1
...................while (!done&input(PIN_B2)) {
09D: BTFSC 3B, 1
09E: GOTO 0BC
09F: BTFSS 06, 2
0A0: GOTO 0BC
...................       level=limit*16;
0A1: MOVF 3D, W
0A2: MOVWF 3C
0A3: SWAPF 3C, F
0A4: MOVLW F0
0A5: ANDWF 3C, F
...................       if(get_rtcc()>71)
0A6: MOVF 01, W
0A7: MOVWF 20
0A8: MOVLW 48
0A9: SUBWF 20, W
0AA: BTFSS 03, 0
0AB: GOTO 0AE
...................                 output_high(PIN_B1);
0AC: BSF 06, 1
...................       else
0AD: GOTO 0AF
...................                 output_low(PIN_B1);
0AE: BCF 06, 1
...................       if(++limit==0x24)
0AF: INCF 3D, F
0B0: MOVLW 24
0B1: SUBWF 3D, W
0B2: BTFSC 03,2
...................                 limit=0;
0B3: CLRF 3D
...................       output_bit(PIN_B3,
...................       shift_left(&data,1,0));
0B4: BCF 03, 0
0B5: RLF 2D, F
0B6: BTFSC 03, 0
0B7: GOTO 0BA
0B8: BCF 06, 3
0B9: GOTO 0BB
0BA: BSF 06, 3
```

Standard C Syntax

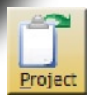
- if, else, while, do, switch, case, for, return, goto, break, continue
- ! ~ ++ -- * = = , & |
- */% << >> ^ && || ? :
- <= < > >= == !=
- = += -= *= /= %= >>= <<= &= ^=m |=
- typedef, static, suto, const, enum, struct, union
- Arrays up to 5 subscripts
- Structures and Unions may be nested
- Custom bit fields (1-8 bits) within structures
- ENUMerated types
- CONstant variables, arrays, structures, and strings
- Full function parameter support (any number and kind)
- C++ reference parameters and comments allowed

Standard C Syntax

- Supports user defined data storage locations
- C data types may reside in any type of storage
- User-defined access routines
- Implements a virtual memory scheme
- Located C data in program memory
- Targets with external memory can use the external bus for data

Downloaded from Elcodis.com electronic components distributor

CCS C Windows IDE & C Aware Real-time Debugger



Project Navigation

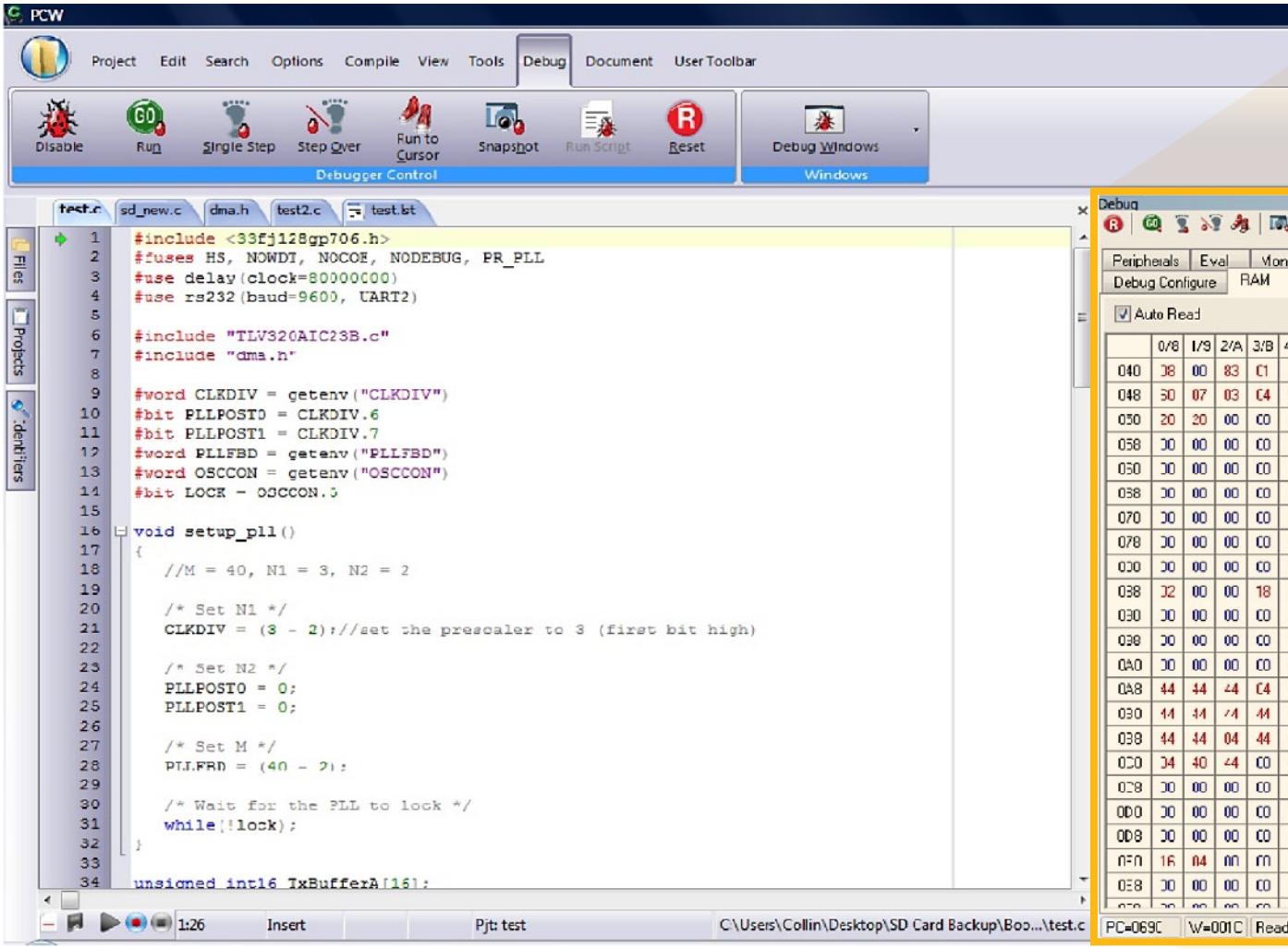
File bar shows all project related files and can quickly open or compile a file.

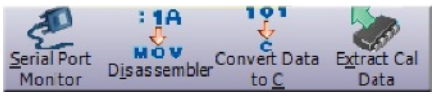
Identifier bar shows all project functions and identifiers.

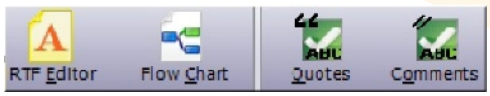
Wizards

Simplifies configuration of drivers and peripherals to start projects quickly. Forms based on interactive questions to aid in set-up of options, such as calculating and showing the timer options based on your clocks.

Included in PCW: CAN Bus, USB, RS-485, and many more







Tools

- Device Selector
- Edit/Add to Device Database
- Generate C Constant Declarations from hex/binary
- Special Function Register Reference
- Serial Port Monitor
- File compare for list or source files

Editor Features

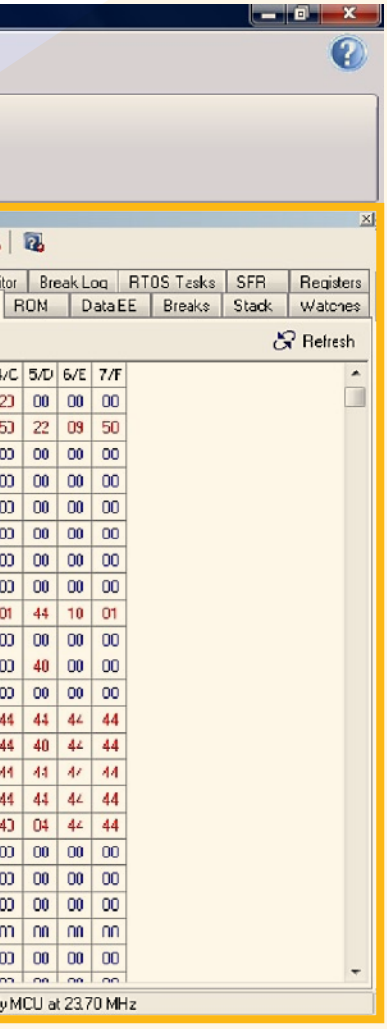
- RTOS--integrated for maximum efficiency and multi-tasking allowed with deterministic scheduling
- Automated C indenting
- Context Sensitive Help
- Color Syntax Highlighting

Downloaded from Elcodis.com electronic components distributor

IDE compiler capabilities include many utilities to aid in program design and editing. The C Aware Real-time Debugger allows for high-level debugging in a C. It is included in all IDE compilers and can be used with the CCS ICD and Mach X, and Microchip ICD2 and RealICE.

Special Viewers

Include quick and easy access to data sheets, valid fuses, interrupts for devices, hex file disassembler, .COD file interpreter, and an advanced source/list file compare.

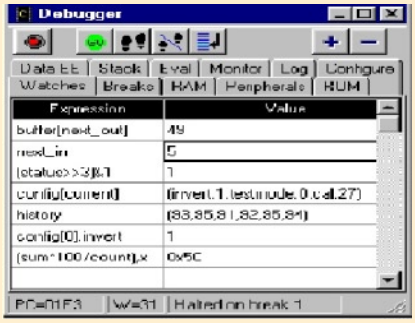


RAM

The RAM window allows the user to view all the memory locations in the device RAM.

- C brace matching
- Multiple open windows
- Technical Support Wizard
- Multiple Compilation Unit preprocessor directives
- Documentation Generator
- Flow Chart Editor
- RTF Documentation Generator
- Spellchecker
- Download Manager

C Aware Real-time Debugger

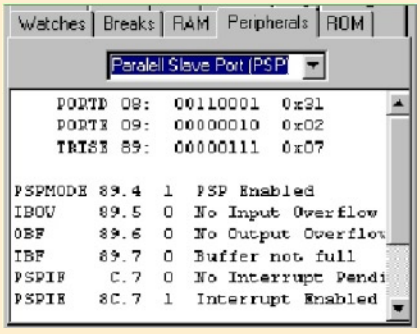


Watches

Full C expressions can be specified. Arrays and structures are understood and shown in natural form. Variables can be modified and break points can be set in file line.

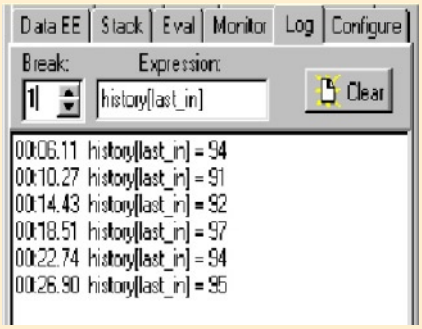
Data EE

A monitor allows character I/O to and from the target platform. The target platform can printf to this debugger window and getc from it.



Peripherals

Special function registers are grouped by a function and each bit is fully interpreted in the debugger window.



Logging

Configurable to save data each time a specified source line is executed. Set-up profiles can be saved and used with any project.

Debugger data can be printed or saved to disk file.

Downloaded from Elcodis.com electronic components distributor

NEW in 2008

Products for Production & Development

PRIME8
Production Programmer

FOR PRODUCTION

In-circuit gang programmer for Microchip Flash devices:

- 8 Selectable Targets
- 2.5V, 3.3V, 5V target vdds available
- 2MB flash that can store up to 4 programs
- SD card reader to load targets in the field
- CCS Programmer Control Software
- LCD screen with user interface for functions

LOAD-n-GO
Handheld Programmer

FOR PRODUCTION

Low-cost in-circuit programmer for Microchip Flash devices:

- AA battery operated or 9V DC Adapter
- 2MB flash that can store up to 4 programs
- Automatic shut-off to conserve batteries
- 2.5V, 3.3V, 5V target vdds available
- CCS Programmer Control Software
- Free software updates to add devices

ICD-U64
In-Circuit Debugger/ Programmer

FOR DEVELOPMENT & PRODUCTION

- Programs all Microchip Flash devices
- In-circuit programmer uses modular jack
- Powered from USB bus
- Target voltages from 2V to 5V
- Debug capability with CCS PCW, PCWH and PCWHD
- Standalone utility for easy downloading
- Command-line interface for integration with other software
- Supports automatic serial numbering
- Windows & linux host supported

Over 30% FASTER than an ICD-U40!

USB Master
Development Board

FOR DEVELOPMENT

Create embedded devices that function as the master of a USB bus and control slave devices.

This development board combines the PIC18F67J10 device with the Vinculum VNC1L to create:

- USB Hub
- Read and Write to FAT formatted Flash drives
- Communicate with printers and CDC devices
- Utilize HID class devices in embedded applications

DSP Analog
Development Board

FOR DEVELOPMENT

Board designed for audio signal processing development. Uses a TI audio codec chip that can acquire data from microphone and drive headphones.

- Equipped to record, process and playback audio signals up to 24 buts at 44.1 Khz
- SD card reader to store recorded audio or signal data
- LCD display and large number of digital inputs and 2 ADC inputs
- Learn to use Direct Memory Access for real-time digital signal applications

Downloaded from Elcodis.com electronic components distributor

Development Kits

Basic Kits

CCS offers a wide variety of all-inclusive Development Kits that bring hardware and software together to provide an innovative package of development tools. The CCS Development Kit allows engineers to design, develop, implement, and test applications directly on PIC® MCUs and dsPIC® DSCs. Novices and experienced developers alike save both time and money with the CCS Development Kits without a need to buy any additional equipment.

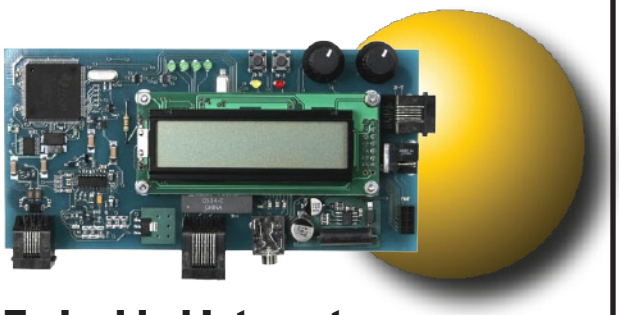
Kit name	Push Button	3 LEDs	POTS	RS-232	I/O Pins		IDE Compiler	Special Feature
					Total	Analog		
PIC12F675	✓	✓	2	1	6	4	PCW	Board includes the 14-pin part for ICSP and debugging at the C level
PIC12F683	✓	✓	2	1	6	4	PCW	Similar to PIC12F675 with twice the RMA and EE-PROM, 3 timers and Capture/Compare/PWM module
PIC16F877A	✓	✓	1	1	30	7	PCWH	Basic features for quick and easy learning
PIC16F887	✓	✓	1	1	30	12	PCWH	Enhanced features of 877A family with additional I/O
PIC18F4520	✓	✓	1	1	30	11	PCWH	Basic features that require more RMA and Data EEPROM space
PIC18F6722	✓	✓	1	2	48	11	PCW	RS-232 Level Converter connected to the C6/C7 UART and G1/G2 UART
PIC18F8722	✓	✓	1	2	29	13	PCW	External Flash and RAM
PIC18F67J10	✓	✓	1	2	48	10	PCWH	Basic features for 3.3V applications
PIC24F	✓	✓	1	2	48	16	PCDIDE	Runs at 16 MIPS
PIC24H	✓	✓	1	2	48	18	PCDIDE	Runs at 40 MIPS –Low drive current
DSP Starter	✓	✓	1	1	10	1	PCDIDE	Real ICE™ connector and a Header to access the available GPIO, Runs at 30 MIPS

Downloaded from Elcodis.com electronic components distributor

Internet Connectivity Kits

Embedded Ethernet

On-board ENC28J60 chip and MMC/SD card reader.
SPI controlled, 10Mbit/sec, full duplex Ethernet transceiver IC.
Drivers and example TCP/IP code included.

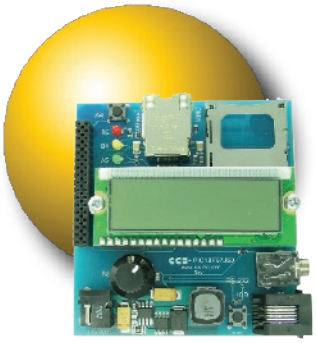


Embedded Internet

Demonstrates TCP/IP and Internet connectivity. Both a 56k modem and 10MB Ethernet connection provide access to the Internet.

CCS provides a port of Microchip's TCP/IP stack and an API for developing applications.

Examples include a simple web server to allow web clients to view readings from around the world and a SMTP/E-mail client to send E-mails.



3.3V Ethernet Controller

The PIC18F67J60 chip offers more memory at operating voltages of 2.0 to 3.6V and is 10-BaseT (10Mbps) compliant.

The PIC18F67J60 can run at speeds up to 40 Mhz and Ethernet transfers up to 10 Mbit/sec at 3V.

Example programs include a complete web server, e-mail generator and SD Card Read/Write.

Wireless Kits

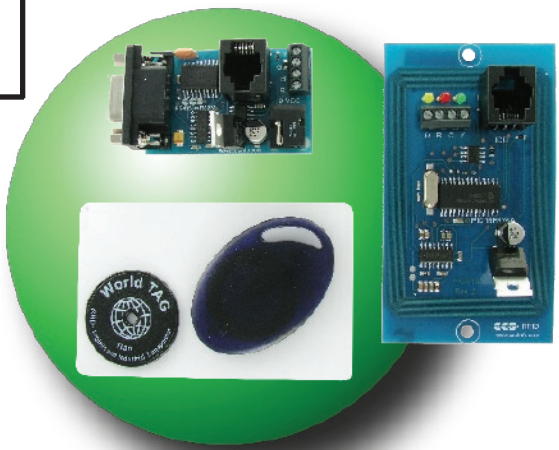
CCS Wireless-Ember Zigbee™ Edition

Introduction to developing ZigBee™ applications for the Ember EM260 processor with PIC16 and PIC18 devices. Includes full communication protocols with the Ember-ZigBee™stack.

The minuscule EM260 module is a 2.4 Ghz IEEE 802.15.4 compliant transceiver (3.3V operation) with a SPI interface and Insight Port for advanced network debugging with Insight Desktop.

Base Station board combines the EM260 module with PIC18LF4620 device that has 10 I/O Pins (3 may be Analog). The two battery-operated Sensor boards combine the EM260 module with PIC16LF886 device that has 5 I/O Pins (1 may be Analog).

NEED NEW



RFID

Simple read-only and read/write transponder to demonstrate multiple contactless communication possibilities. A manual with source code examples explains how to use the drivers, enabling you to quickly develop your own RFID applications.

RFID Prototype Board has a short range RFID antenna connected to an external RFID transceiver IC. The RFID prototype board connects to external components using a four-wire RS485 bus. The RS485 connection on the RFID Prototype board is to accomodate multi-drop/ multinode network of RFID units and other RFID related components

Kit also includes: RFID Prototyping Board, RS485-to-RS232 Prototyping Board, Two Read-Only RFID Transponders, and One Read/Write RFID Transponder (Password Protection and can be made Read-Only or Write-Only)

Downloaded from Elcodis.com electronic components distributor

Human Interface Kits

ACE Kit

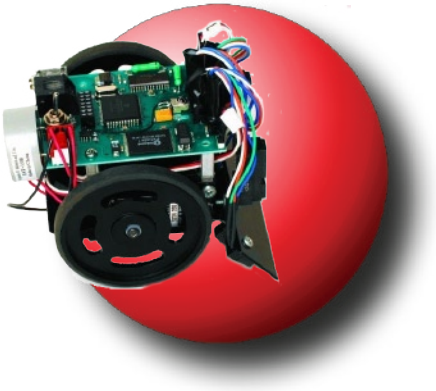
The ACE Kit is the perfect solution to to an advanced engineer's development needs. This kit provides multiple accessories packaged together to provide diverse programming situations.

This board contains connectors and expansions to debug a variety of situations before the final target platform is designed.

The ACE Kit utilizes the CCS Software Prototyping Board - A PIC® MCU designer's best friend.

Kit also includes: Software Prototyping Board, LCD/ Keypad Set, Experimenter's Set, USB Add-On (does NOT include PIC18F4550 chip), PIC16F877A Reprogrammable Chip, 30 I/O Pins (7 Can Be Analog), and 2-digit 7-segment LED

Robotics



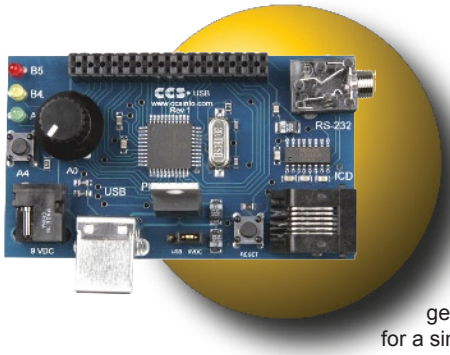
An introduction into the world of robots for both beginners and advanced robot enthusiasts. Included devices allow the robot to see, sense magnetic fields, speak, accept external commands, and move.. fit for robot sumo competition!

Each device has its own chapter in the included exercise book, describing how to operate and use its drivers. Bonus chapters on Real Time Operating Systems (RTOS) and advanced project ideas are included.

The electronic compass and text-to-speech converter are unique to the CCS Robotics Kit. The compass allows the robot to move freely and still know its heading and location. The text-to-speech converter provides a more personal way to interact with people.

Kit also includes: Controller Board and TV Remote

Bus Introductory Kits

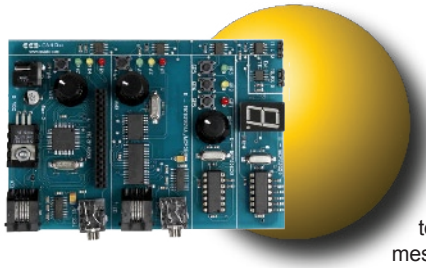


USB

Has a PIC18F4550, an external USB controller, and a function generator. Applications for a simple human interface applications and a high speed example emulating an oscilloscope. PC software (with source) is included to communicate with the USB board. Board uses the PIC18F4550, Microchip's PIC® MCU with full speed USB peripheral.

Example programs with the CCS C compiler and C source code are provided: How to configure the board to act as a HID device. How to configure the board to act as a USB device that accepts and receives bulk mode transfers.

USB examples are also compatible with the Microchip PIC16C7x5 USB peripheral, which CCS also provides C source code with the CCS C compiler.



CAN Bus

Controller Area Network (CAN) is a serial bus system for a network of controllers. There are four nodes that are able to transmit and receive messages from the network. CAN Bus offers a secure communication channel to exchange up to 8 bytes between several network nodes.

Node 1: The PIC18F4580 includes an integrated CAN peripheral.
Node 2: A PIC16F876A is connected to a MCP2515 (external CAN peripheral with SPI interface)
Node 3 & 4: MCP25050s (stand-alone CAN expanders) pre-programmed by CCS to respond to specific CAN IDs.

Nodes 1-3 have a potentiometer, three LEDs and three pushbuttons connections. Node 4 is connected to a 7-segment LED.

Embedded Serial Busses

An introduction to SPI and I²C serial busses, with no wiring needed. The board has two nodes and shares common components between different devices, but maintain their own potentiometer, LEDs, pushbutton and RS-232 port .

Node 1: A PIC16F877A chip is connected to a 74HC165 chip expanding to bank of 8 DIP switch inputs, 74HC595 chips expands output to display information on three 7-Segment LEDs, and a serial real-time clock.
U A PIC16F876A chip, which shares an I2C temperature sensor and a serial EEPROM with the first node. This allows for the investigation into data collision, while accessing shared components. Both nodes have.

Downloaded from Elcodis.com electronic components distributor

Programmers/ Debuggers

CCS has a complete line of Programmers and Debuggers for all Microchip PIC10, PIC12, PIC14, PIC16, PIC18, PIC24 and dsPIC® devices. The entire line supports in-circuit debugging at the C level with any IDE compiler, and in-circuit programming with the IDE or the stand-alone CCS Programmer Control Software for all Flash-supported devices.

The CCS ICD units work with the CCS C Aware Real-Time Debugger for detailed debugging information at the C level. ICD-U40 is powered by the USB bus and can be modified to power the target board at 5V. ICD-S40 communicates via a RS232 serial bus and is powered by the target board.

The Mach X Programmer is a full-featured device programmer with a standard ICD connector and a 40-pin ZIF socket with advanced ICSP signal routing logic to accommodate various pin-outs (8-40 pins). All Flash memory, One-Time Programmable (OPT) and MCPxxxx(CAN Bus chips) can be programmed at the user selectable voltage range of 2V to 5V.

	ICD-U40	ICD-S40	MACH X	LOAD-n-GO	Prime 8
In-Circuit Programming	✓	✓	✓	✓	✓
In-Circuit Debugging	✓	✓	✓	✓	✓
ZIF Socket	—	—	✓	—	—
PC Interface	USB	RS-232	USB	USB	USB
Power	USB	Target	USB	USB, 4 AA Batteries and AC Adapter	USB and AC Adapter
Target Boards	1	1	1	1	8

Downloaded from Elcodis.com electronic components distributor