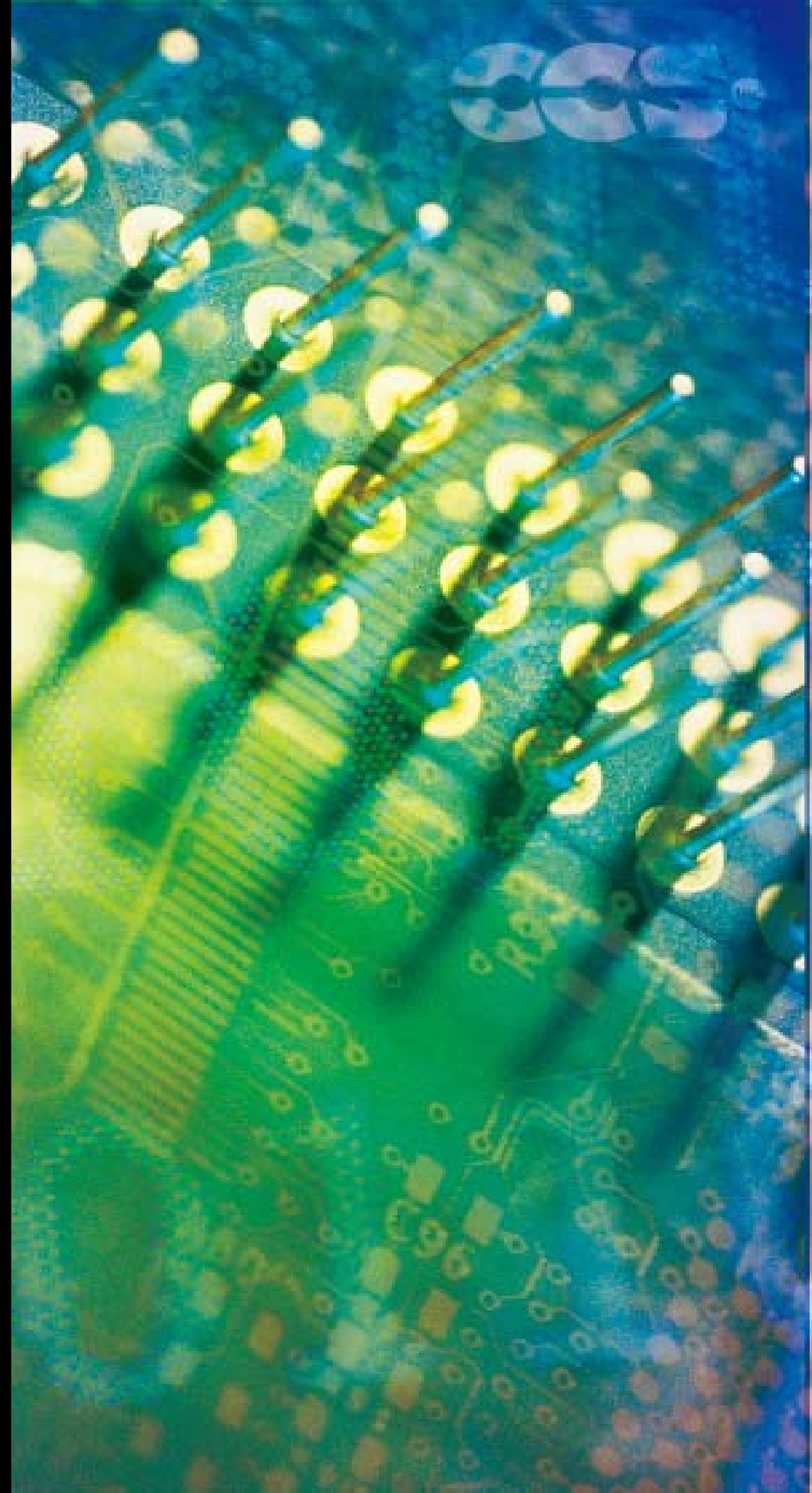
有限会社データダイナミクス
Tel:0729-81-6332 Fax:0729-81-6085
http://www.datadynamics.co.jp/ccs/picc.html

PIC MCU C Compilers



COMPLETE C SOLUTIONS

CCS社はPIC10,PIC12,PIC14,PIC16とPIC18マイコンをサポートしたWindows及びLinux環境におけるC組み込み開発ツールを幅広く提供しております。CCS社はマイクロチップ社 PIC® MCUの開発に必要な最もシンプルで有用なソリューションを提供することを目指しています。

CCS社のコンパイラーにはリアルタイム・クロック、シリアルEEPROM，LCDインターフェイス等の様々な周辺デバイスを使用するための有用なライブラリーと実行可能なサンプル・プログラムが含まれています。

これらのライブラリーとサンプル・プログラムによりアプリケーションの開発工数を低減する　ことが出来ます。CCS社のコンパイラは、他のコンパイラより、アプリケーション開発の生産効率を向上させることが出来るでしょう。

THE CCS C COMPILER

- 内蔵ライブラリーによりすべてのチップでRS232C、I2C、ディスクリートI/O、正確なディレイ機能を提供します
- 1，8，16と32ビット整数タイプと32ビット浮動小数点
- 標準のHEXファイルとデバッグ・ファイルは全てのプログラマーで使用することが出来ます
- フォーマット出力printfを使用することで簡単に16進又は、10進出力が可能です
- ハードウェアスタックより深いコール・ツリーを可能にします
- 複数のタイマー，A/D，EEPROM，SSP，PSP，USB，I2C等の内蔵デバイスを簡単に利用するための使い易いC関数を用意
- アセンブラー・コードをソースに直接，どこにでも入れることができ，Cで定義された変数の参照が可能
- PCWとPCWHにはリアル-タイム・オペレーティング・システム(RTOS)が含まれています
- Windows IDEにはパワフルなCデバッガーが含まれています　(*ICD-Uが必要)
- リンカーは自動的にベストなアーキテクチャーを決定し、効率高いコードを生成します（手動定義も可能）
- コンパイラの指定によりすべてのI/Oの入出力を自動切換え、又は入出力を固定することにより動作速度を早くすることが可能です
- ストリングス，アレイを含む定数をプログラム・メモリーにセーブできます
- 1ビット・タイプ(Short Int)の導入により非常に効率の良いコードを生成します
- #BIT，#BYTEによりレジスタを絶対番地で指定しC変数として使用することが出来ます
- リファレンス・パラメータによりコードの可読性を向上させ、スタック領域の使用を抑えるためのインライン処理を効率良く行えます
- RAMメモリー・マップ，C/アセンブラー・コード，コール・ツリー用のウィンドウを用意

PRE-PROCESSOR

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

FUNCTION QUALIFIERS
#inline
#separate
#int_XXXXX
#int_global
#int_default

DEVICE SPECIFICATIONS
#device
#id
#fuses
#serialize

LIBRARIES
#use delay
#use rs232
#use i2c
#usestandard_io
#use fixed_io
#use fast_io

COMPILER
#case
#opt
#priority
#type
__date__
__device__
__file__
__line__
__time__
__pcb__
__pcm__
__pch__
__pcdsp__
getenv()
#ignore_
warnings
#build
#error

MEMORY
#byte
#bit
#locate
#reserve
#rom
#zero_ram
#asm
#endasm
#org
#fill_rom

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: M0VF 01, W
0A7: MOVWF 20
0A8: MOVLW 48
0A9: SUBWF 20, W
0AA: BTFSS 03, 0
0AB: GOTO OAE
.....                 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
```

TYPEMOD SUPPORT:

- ユーザー定義されたデータの保存場所をサポート
- Cデータ・タイプを、たとえばシリアルEEPROMの任意の場所に置く事ができます
- ユーザー定義アクセス・ルーチン
- バーチャル・メモリー方式をインプリメント
- プログラム・メモリー上にCのデータを配置
- 外部メモリーを持ったターゲット(例18F8720)では複数データのための外部バスの使用

RTOS SUPPORT:

- 最高効率のための言語への統合
- 確定的スケジューリング方式でマルチ-タスクが可能

STANDARD C:

- if, else, while, do, switch, case, for, return, goto, break, continue
- ! ~ ++ -- * + - , & |
- * / % << >> ^ && || ?:
- <= < > >= == !=
- = += -= *= /= %= >>= <<= &= ^=m |=
- typedef,static,auto,const,enum,struct,union
- 配列は5つまでの予約語
- ストラクチャーとユニオンがネストされます
- ストラクチャー内でのカスタム・ビット・フィールド(1-8ビット)
- エニュメレート タイプ
- ROMスペース内へのコンスタント バリアブル,アレイ及びストリ
- 関数パラメーターとしてあらゆる数字と文字をサポート
- C++リファレンス・パラメーターとコメント

CAPABILITES

BUILT-IN FUNCTIONS

RS232 I/O
getc()
putc()
gets()
puts()
printf()
kbhit()
set_uart_speed()
fgetc()
fgets()
fprintf()
fputc()
assert()
perror()
fputs()
setup_uart()

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

DISCRETE I/O
output_low()
output_high()
output_float()
output_bit()
input()
output_X()
input_X()
port_b_pullups()
set_tris_X()
port_a_pull-ups()

INTERNAL LCD
setup_lcd()
lcd_load()
lcd_symbol()

SPI TWO WIRE I/O
setup_spi()
spi_read()
spi_write()
spi_data_is_in()

PARLLEL SLAVE I/O
setup_psp()
psp_input_full()
psp_output
full()
psp_overflow()

DELAYS
delay_us()
delay_ms()
delay_cycles()

TIMERS
setup_timer_X()
set_timerX()
get_timerX()
setup_counters()
setup_wdt()
restart_wdt()

CAPTURE/ COMPARE/PWM
setup_ccpX()
set_pwmX_duty()

PROCESSOR CONTROLS
sleep()
reset_cpu()
restart_cause()
disable_ interrupts()
enable_ interrupts()
ext_int_edge()
read_bank()
write_bank()
label_address()
goto_address()
clear_interrupts()
setup_oscillator()
setjmp()
longjmp()

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

A/D CONVERSION
setup_adc_ports()
setup_adc()
set_adc_channel()
read_adc()

ANALOG COMPARE
setup_comparator()

VOLTAGE REF
setup_vref()

STANDARDC MEMORY
memset()
memcpy()
memmove()
memcmp()
memchr()
offsetof()
offsetofbit()
setjmp()
longjmp()
calloc()
malloc()
realloc()
free()
bsearch()
qsort()

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

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

SPECIAL MEMORY
read_eeprom()
write_eeprom()
read_program_eeprom()
write_program_eeprom()
read_calibration()
erase_program_eeprom()
write_program_memory()
read_program_memory()
read_external_memory()
setup_external_memory()
write_external_memory()

WINDOWS IDE COMPILER

IDE機能

プロジェクト管理
　ニュー・プロジェクト・ウィザード
　プロジェクト内のすべてのファイルの検索
デバッグ機能 *要ICD-U(S)
　モニター出力
　変数ウォッチ
　スタック・トレース
　RAM ビュー
　式評価
ツール/ユティリティー
　ソース、又は、リスト・ファイルの比較
　デバイス・エディター
　シリアル・ポート・モニター
　逆アセンブラー
ウィンドウズ・エディター
　Ｃ言語用のインデント処理
　プロジェクト中のテキスト検索
　コンテキスト・センシティブ・ヘルプ
　対応する括弧　｝又は　）　のサーチ

データシートへの直接リンク

アセンブラー，シンボル，コール・ツリーと
スタティクス・ファイル・ビューワー

デバイスごとのコンフィグレーションヒューズ及び割り込み設定

CODと.COFFファイル・インタープリター

”List File”
リスト・ファイルはCステートメントと対応するアセンブラー命令を表示。デバッガはリスト・ファイル内でブレークポイントの設定、ステップ実行が出来ます。

”Statistics Window”
スタティクス・ウィンドウは各ファイルの行とステートメントと各機能で使用される。ROM，RAMをパーセンテージ[%]で表示します。

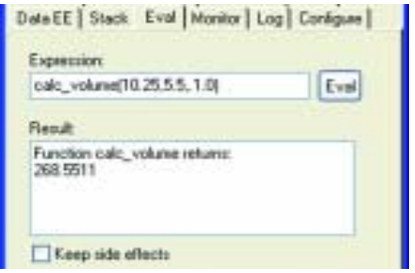
シリアル・ポート・モニター・ユティリィーでPCのシリアル・ポートとターゲットが相互に通信

”Windows Editor”
PCWとPCWHにはCを意識した構文のカラーハイライト、｝又は ）の対応する括弧のサーチ等のC言語ソースファイルの編集に便利なウィンドウズ・エディターが含まれています。

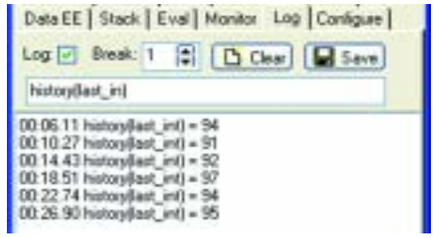
”New Project Wizard”
新規プロジェクト・ウィザードはデバイスごとの各種機能の使用の有無を分かり易いフォームで設定し、初期設定用の.Cと.Hファイルを生成します。フォームを利用することにより、ユーザーの負担を軽減します。例えば、ユーザーが使用するクロックに合わせてディレイタイマーの設定、組み込みデバイスのレジスターの設定が行えます。

C AWARE DEBUGGER

CCS社はマイクロチップ社のフラッシュPIC Rシリーズ・マイクロコントローラのためのハイレベルなデバッギング・ソリューションを提供しております。デバッガーはすべてのWindows IDEコンパイラーに含まれており、デバッグはフラッシュPIC RとICDユニットを接続して行ないます。将来エミュレータの機能とdsPIC Rのサポートを予定しています。

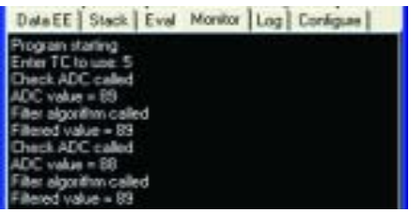


拡張した評価の仕方としては、デバッガーから目的の関数をコールする事によって行なえます。関数は与えられたパラメータで呼び出され、結果が表示されます。このパワフルなツールはボトムアップによるテストを簡単にします。

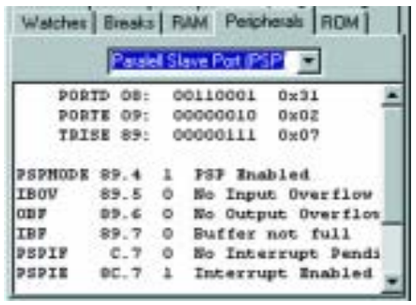


デバッガーはプログラムで与えられたブレークポイントに達して、実行される度に式を評価しログを取ることが出来ます。

上のスクリーンショットは複合された式の評価のウォッチ・ウィンドウを表示しています。デバッガーはユーザー定義(# defines)とマクロを含む、有効ないかなるCの式も評価することが出来ます。各項目毎に表示形式を指定することが可能です。式は指定されたソースの行の範囲に関連した評価をすることが出来ます。



Cコンパイラーはデバッガー・モニター・ウィンドウでgetc(),print()からのI/Oコマンドを実行するように設定出来ます。これはデバッグのためにRS232ポートをチェックするためのエクストラ・ピンとハードウエアを省くことが出来ます。



すべてのMCUの周辺機器のそれぞれの状態をテキストで見ることが出来ます。

スタックはインタラプトとインライン呼び出しを含んだ全てのCの関数呼び出しを表示します。関数のパラメータはそれぞれの呼び出しの状態にそって表示されます。

INCLUDED C DRIVERS

SERIAL EEPROM/FLASH
2041
24xx
25xx
93xx
AT2421
AT25256
AT29C1024
AT45DB021
CE51x
CE62x
CE67x

SERIAL RAM
68HC68R1
68HC68R2
M68AF031
PCF8570

A/D & D/A CONVERTERS
AD7705
AD7715
ADS8320
LTC1298
MAX517

DIGITAL POTS
AD8400
DS1868

ACCELEROMETER
ADXL210

OTHER
Digital Compass
Keypad
Mag Card Reader
PLL Interface
Dallas One Wire
IR Decoder
Line_Tracker
Servo Control
X10

REAL-TIME CLOCK
DS1302
NJU6355

SOUNDS
WTX701
TONES
ISD4003

RFID
EM4095
EM4102
EM4150

CAN FUNCTIONALITY
MCP2510
8XXX8

EXPANDED INPUT/OUTPUT
74165
74595
MAX7300

USB
USBN960X
PIC_USB
PIC18-USB

LCD
GLCD
KS0108
LCD
LCD420
SED1335
HDM64GS12

TEMPRERATURE
DS1621
DS1621M
DS1631
DS1624

NETWORKING/INTERNET
TCP
PPP
SC28L19X
S7600
RTL8019
RS485

SECONDS TIMER

```
#include<16f877a.h>
#fuses HS,NOLVP,NOWDT,PUT
#use delay(clock=20000000)
#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7)

#define high_start 76
byte seconds, high_count;

#INT_RTCC //Interrupt procedure
clock_isr() { //called every time RTCC
   high_count -= 1; //flips from 255 to 0

   if(high_count==0) {
     ++seconds;
     high_count=high start;
         //Inc SECONDS counter every
    }     //76 times to keep time
}

void main() { //A simple stopwatch program
   byte start, time;

   high_count = high_start;
   setup_timer_0( RTCC_INTERNAL | RTCC_DIV_256 );
   set_timer0(0);
   enable_interrupts(INT_RTCC);
   enable_interrupts(GLOBAL);

   do {
       printf("Press any key to begin.¥n¥r");
       getc();
       start = seconds;
       printf("Press any key to stop.¥r¥n");
       getc();
       time = seconds - start;
       printf("%U seconds.¥n¥r", time);
   } 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 pin A0
      min = 255;  //and displays the min and max
      max = 0;    //values for that 100ms period
      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);
}
```

ICD IN-CIRCUIT DEBUGGER

CCS社のICD-UはMicrochip社の PIC16FxxとPIC18Fxx MCUのための完全なインサーキット・デバッギングとプログラミング用の専用機器です。ICDはインサーキット・シリアル・プログラミング(ICSP) 機能を持ったすべてのフラッシュチップをサポートしています。
ICDはPCWとPCWHデバッガーとスタンド-アローンICDコントロール・ソフトウエアで動作します。スタンド-アローン・ソフトウエアでターゲット・チップにプログラムし、ICDのファームウエアをアップデートすることが出来ます。
CCSのICD-U40はUSBバスからの電源で40MHzクロック周波数で操作することが出来ます。

COMPLETE EXAMPLE PROGRAMS

LCD
A/D
CCP
PWM
COMP
PSP
Serial Interrupts
I2C
WDT
Sleep
Boot Loader
DTMF
Optical Encoder
Frequency Counter
7 Seg LED
CRC Calculator
Data Logger
Pattern Generator
Stepper Motors
Tone Generation