

MSP430導入用マニュアル

(日本テキサス・インスツルメンツ（株） プロダクト・インフォメーション・センター(PIC) 山内 一男)

rev. 0.1

MSP430F5438Aを題材に、MSP430の初級者向けにソフトウェア開発のポイントと留意点について説明します。

参照しているURLおよびドキュメント内容が改版され、本書内容と相違がでるケースもありますが、ご了承お願いいたします。

また、下記のURLの「お問い合わせ」より、MSP430全般の技術的な内容について、および本書の内容についてお問い合わせいただけます。 <http://focus.tij.co.jp/jp/tihome/docs/homepage.tsp>

目次

1. 準備するドキュメントや機材など	2
1.1. ドキュメント	2
1.2. 評価ボード	2
1.3. Code Examples	2
1.4. JTAGエミュレータ	3
1.5. 開発環境の入手	3
2. ソフトウェア開発のポイント	4
2.1. ブロック図とレジスタ	4
2.2. プログラムヘッダー	4
2.3. Code Examplesとペリフェラル制御	4
2.4. Interrupt Service Routine (ISR)	4
2.5. 留意ポイント	5
3. CCS v 4 操作のポイント	6
3.1. Workspaceの作成	6
3.2. プロジェクトの生成	6
3.3. ソースファイルの移植	6
3.4. Build (コンパイル+リンク)	7
3.5. LED 点滅Exampleの動作確認	7
3.6. Debug例	7
3.7. メモリ配置の確認	8
3.8. その他の留意点	10

この資料は、Texas Instruments Incorporated (TI) が英文で記述した資料を、皆様のご理解の一助として頂くために日本テキサス・インスツルメンツ (日本 TI) が英文から和文へ翻訳して作成したものです。資料によっては正規英語版資料の更新に対応していないものがあります。日本 TI による和文資料は、あくまでも TI 正規英語版をご理解頂くための補助的参考資料としてご使用下さい。製品のご検討およびご採用にあたりましては必ず正規英語版の最新資料をご確認下さい。TI および日本 TI は、正規英語版にて更新の情報を提供しているにもかかわらず更新以前の情報に基づいて発生した問題や障害等につきましては如何なる責任も負いません。

1. 準備するドキュメントや機材など

1.1. ドキュメント

MSP430の下記製品ページを開くと、そこからユーザーズ・ガイドやアプリケーション・ノートにアクセスできます。アプリケーション・ノートには、多くの資料がありますので、ご活用ください。

<http://focus.tij.co.jp/jp/mcu/docs/mcuflashtools.tsp?sectionId=95&tabId=1538&familyId=342>

例として下記のドキュメントをダウンロードします。

*末尾の「SLAU208」は文書番号です。ドキュメントは改版され末尾に a, b, c... と版数記号が付記されます。

- ・ユーザーズ・ガイド ・ MSP430x5xx/MSP430x6xx Family User's Guide (SLAU208)
- ・コンパイラ・マニュアル ・ MSP430 Optimizing C/C++ Compiler v 3.x User's Guide (SLAU132)

ドキュメントは文書番号で検索できます。上段右側の「キーワード検索」に文書番号を入れて探します。

例として「SLAU278」と入力して、下記のドキュメントをダウンロードします。

- ・ハードウェアツール ユーザーズ・ガイド ・ MSP430 Hardware Tools User's Guide (SLAU278)

次に、ターゲットデバイスの製品ページを開きます。

上段右側の「型番検索」にデバイス型番を入れて探すことができます。

例として「F5438A」と入力して探すと、MSP430F5438Aの製品紹介ページが開きます。

<http://focus.tij.co.jp/jp/docs/prod/folders/print/msp430f5438a.html>

この先頭にある「データシート」をダウンロードします。

データシートには、そのデバイスの動作保証スペックなどが記載されていますので、確認をお願いします。

- ・ MSP430F543xA, MSP430F541xA Mixed Signal Microcontroller (SLAS655)

1.2. 評価ボード

各デバイスの製品紹介ページを開いたら、右欄にある「ソフトウェアおよびツール」をクリックします。

この「TI ソフトウェアおよびツール」表の中に、開発評価用のターゲットボードがあります。

MSP430-FET430UIFエミュレータをすでに保有している場合は前者を、保有していない場合は後者を用意します。

- ・ MSP430F5xx 100 ピン・ターゲット・ボード MSP-TS430PZ5X100
- ・ MSP430F5xx 100 ピン・ターゲット・ボードおよび USB プログラマ MSP-FET430U5X100

G2xxxやF20xx のデバイス (N14/20 DIPタイプ) では LaunchPad Development Tool も利用できます。

これら評価ボードについては、下記の文書に説明があります。

- ・ MSP430 Hardware Tools User's Guide (SLAU278)
評価ボードの回路図などがありますので、ボード開発の参考に利用できます。
- ・ MSP-EXP430G2 LaunchPad Experimenter Board User's Guide (SLAU318)
この評価ボードにはJTAGエミュレータが搭載されています。

1.3. Code Examples

F5438A製品紹介ページの右欄にある「ソフトウェアおよびツール」をクリックして、表示される表の中にある下記 Code Examples をダウンロードします。この中には、ペリフェラル制御プログラムのひな型が入っています。

例 MSP430F543xA, MSP430F541xA Code Examples

このCode Examples は後述の CCS v4 で利用できます。

留意すべき制御タイミングに関するコードも含まれているExampleもありますので、ペリフェラル制御ソフトのベース・ひな形としてご利用をお勧めします。このcodeを試すのが、ペリフェラル制御習得の早道です。

1.4. JTAGエミュレータ

(1) MSP-FET430UIF

このJTAGエミュレータを使い、CCSなどの開発ソフトウェア・ツールと評価ボードを接続します。JTAGコネクタとボード上のMPUとの接続回路は、前記文書の「SLAU278」の下記図に説明があります。4-wireと2-wireの2通りの接続方法があります。

Figure 2-1. Signal Connections for 4-Wire JTAG Communication

Figure 2-2. Signal Connections for 2-Wire JTAG Communication (Spy-Bi-Wire)

MSP-FET430UIFを利用するときは、後述のCCS（あるいはIAR-EW430）をインストールした後、PCのUSBに接続すると、ドライバが自動的にインストールされます。

CCSのバージョンが変わった時、あるいはCCSとIARの利用を切り替えた時などに Firmware Update するように window が表示されるときがあります。その時は update yesとします。これで開発ツール対応と新デバイス対応できるように更新されます。

(2) LaunchPad

LaunchPadは、G2xxxとF20xx シリーズ向けの開発キットです。

User's Guide(SLAU318) Figure 1. LaunchPad Overview の「Emulation」と書かれた部分は、Spy-bi-Wire エミュレータとして使用できます。(2-wire接続のみです。また、Firmware Updateはありません。)

J3クリップをはずして、J4から下記table記載のspy-bi-wire信号を取り出せます。信号線は極力短くします。

Table 2. eZ430 Debugging Interface

また、PCのHyper Terminalとvirtual COMを使い、このエミュレータ経由でデバイスのUARTと接続できます。

1.5. 開発環境の入手

(1) ダウンロードとインストール

下記のURLからCCS v 4 をダウンロードできます。表中にある使用条件・ライセンスの説明を確認してください。

**1 GBほどのサイズがありますので、時間の余裕を見てください。

http://processors.wiki.ti.com/index.php/Download_CCS

下記①は30-day Free版ですが、製品と同じFULL機能が入っています。

製品購入した「Activation-ID」を登録することで、製品版 (Licensed) として使用できます。

下記②は、code-size 16 KB制限があります。製品版と開発機能はほぼ同じですが、モジュール内容は異なります。

製品版に利用変更する場合は、これをアンインストールして、①を再インストールする必要があります。

① Download latest production DVD image

② Download latest production MSP430/C28x code size limited image

ダウンロード (zip) したファイルをワークフォルダに解凍して、生成された setup_ccs4.2.xxxxx.exe を実行してインストールします。

(2) アップデート

CCSv4を起動して、Help->Software Updates ->Find and Install にて、アップデートができます。

アップデートするときは、インターネットに接続が必要です。

**複数PCで共同開発するときは、同じ時期に アップデート して、バージョンを合わせてください。

(3) Emulator ドライバ・インストール

MSP-FET430UIF あるいは LaunchPad をPCに接続すると、ドライバがインストールされます。

2. ソフトウェア開発のポイント

2.1. ブロック図とレジスタ

User's Guide の各ペリフェラルモジュール説明には、ブロック図があります。

例 Figure 4-1. UCS Block Diagram

レジスタ説明には、この図中の制御信号（制御ビット名）と設定条件の説明があります。

例 4.4 UCS Module Registers

Code Examples では、制御ビット名と制御レジスタ名を、symbol 名として使用しています。

例 msp430x54xA_OF_LFXT1_nmi.c の1文

```
UCSCTL7 &= ~(XT1LFOFFG + DCOFFG); // Clear XT1 & DCO fault flags
```

レジスタ名 制御ビット名

このように、ブロック図とレジスタ説明、Code Examplesを見ながら、本文の説明を読むと理解が容易になります。

2.2. プログラムヘッダー

前述の Symbol と物理レジスタや制御ビットとの紐づけは、各デバイスの「ヘッダー」で行っています。

各 Code Examples の先頭にある include 文を、使用するデバイスに合わせて変更します。

```
#include "msp430x54xA.h" → #include "msp430F5438A.h"
```

これにより、制御レジスタや制御ビットのシンボルが、そのデバイス固有の物理レジスタと物理ビットに割り当てられます。つまり「ヘッダー」により、デバイスごとの相違をある程度吸収しソフトウェアの互換性を高めています。ですから、制御レジスタ格納値を16記 0x1234 など記載しないことをお勧めします。

ビット設定がon/offどちらか確認したい時は、このヘッダーをサーチします。

2.3. Code Examplesとペリフェラル制御

Code Examples には、ほとんどのペリフェラル制御のテンプレートがあります。また、マニュアルに書いてある時間制約など見落としやすい項目も、Code Examples には記載がありますので、ぜひ事前に目を通してください。

Code Examples(zip) の中の !README.txt には、Code Examples のリストと簡単な説明があります。

2.4. Interrupt Service Routine (ISR)

Code Example には ISR の例がありますが、もし該当の ISR 関数名が見つからない場合は、下記の方法で登録されている割り込みベクターと ISR 関数名を見つけられます。

① データシート中のベクターアドレス表を検索します。 例 タイトル 「Interrupt Vector Addresses」

② 該当するペリフェラルの Priority 値を探します。

TA0 TA0CCR0 CCIFG0(3) Maskable 54 ← Priority

③ 各デバイスのヘッダー msp430F5438A.h の中を int54 でサーチします。

ヘッダー格納フォルダー ¥CCSV4¥ccsv4¥msp430¥include

C言語の場合は下記が該当します。

```
#define TIMER0_A0_VECTOR (54 * 1u) /* 0xFFEC Timer0_A5 CC0 */
/*#define TIMER0_A0_ISR(func) ISR_VECTOR(func, ".int54") */ /* 0xFFEC Timer0_A5 CC0...
```

④ 上記を参考にして、下記のようにISR関数を記述します。

これで、Time_A0の割り込みベクターにISR(TIMER0_A0_ISR)のアドレスが設定されます。

```
// Timer0 A0 interrupt service routine
#pragma vector=TIMER0_A0_VECTOR
__interrupt void TIMER0_A0_ISR(void) { }
```

2.5. 留意ポイント

(1) 変数初期化

CCSのMSP430コンパイラは、初期値なし変数を初期化しません、不要な初期化時間を省いています。
初期化が必要な変数は、明示的に初期設定をしてください。

(2) Watch Dog Timer

WDTのデフォルト設定は「Power On などの初期起動で動作する」です。必ずアプリケーション先頭でWDTを停止させるか、適切なタイマー値を設定します。さもないと、途中で WDT NMI が入り、再度 main() から走行します。
Example では、先頭で下記のように停止させています。

```
WDTCTL = WDTPW + WDTHOLD;           // Stop WDT
```

(3) InterruptとISR

割り込み（割り込み可）を利用する場合は、対応する ISR を登録する必要があります。
ISR が未登録の場合は暴走につながります。暴走は、さらに violation を発生させるケースもあります。

NMI が起きた場合、C 言語では main() の先頭から走りますので、対応する事象が起きているか該当レジスタをみて判定します。デバッグ中は先頭にブレークポイント設定しておくと、この main() 再走行を検知できます。

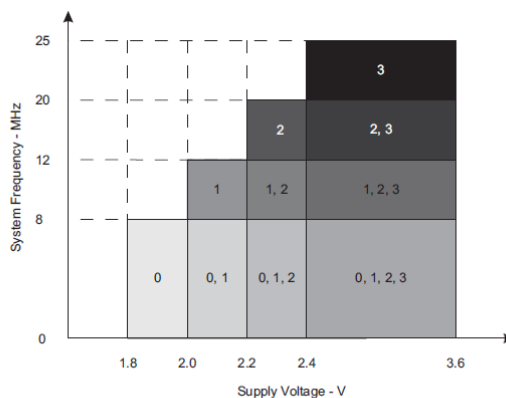
ベクターは16bits 表現ですから、飛び先の ISR は必ず下位の 64KB に配置する必要があります。
64KB 以上（アドレス20bits）に配置すると ISR アドレスが正しく登録できず暴走します。

(4) VccとCPUクロックの注意点

Data-Sheet (SLAS655x) に下記の図があります。Figure 1. Frequency vs. Supply Voltage

他のデバイスでも同じような図があります。

CPUクロックが、この範囲（グレー領域）を絶対にオーバーしないように（時間変化や経年変化を含め）注意してください。この範囲をオーバーするとCPUが誤動作する恐れがあります。



The numbers within the fields denote the supported PMMCOREVx settings.

Figure 1. Frequency vs Supply Voltage

3. CCS v 4 操作のポイント

3.1. Workspaceの作成

Workspace は、CCS のプロジェクトを格納するフォルダーです。この配下に、各プロジェクトのディレクトリが生成されます。大きな作業単位、たとえばシステムやサブシステム単位でワーク領域を作ることに相当します。

事前にこのフォルダー（例 workspace1）を作っておきます。CCSv4 を起動して、この workspace 入力を要求されたとき、このフォルダーを指定します。

CCSv4 起動後に切り替えることもできます。

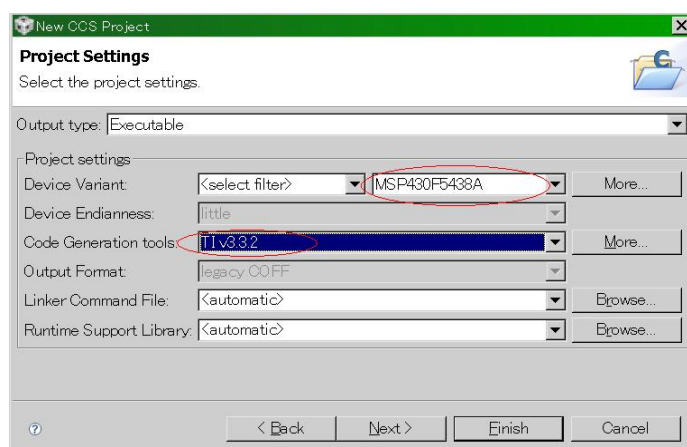
File -> Switch Workspace

3.2. プロジェクトの生成

Workspace の配下に、プロジェクトディレクトリを生成し、使用するデバイスを選択する操作です。C 言語用の project を生成してみます。

[操作法]

- ・ File -> New -> CCS Project : プロジェクト名を指定します。
- ・ Next -> Next -> Next で デバイス種別の選択画面になります。
- ・ デバイス種別 F5438A を選択、他は Default のままとします。



** CGT のバージョンの注意は後で説明します

Finish すると、下記の 2 種類の file が自動生成され、project の入れ物が完成します。

xxxxx.ccxml ・ ・ デバイスとデバッグの設定 file
link_xxxx.cmd ・ ・ リンクコマンド file

3.3. ソースファイルの移植

LED 点滅させる CCS の Code Examples を移植してみます。

[操作法]

- ・ File -> Open File -> ここで CCSv4 格納フォルダーの下記を選択します。
¥ccsv4¥msp430¥examples¥msp430x5xx¥C-sourceSource File¥msp430x5xx_fet_1.c
- ・ ソースファイルが表示されます。この Example は、デバイスヘッダーを下記に修正します。
(本当は、修正しなくとも動きますが、機能追加に備えて訂正しておきます。)
→ #include "msp430F5438A.h"
- ・ File -> Save As ファイル名を main.c として書き込みます。

これで、project にアプリケーション main() ソースが加えられました。

3.4. Build (コンパイル+リンク)

コンパイル、リンクして実行用オブジェクトを生成する操作です。

[操作法]

- ・ Project -> Build Active Project により、コンパイル->リンク 実施されて、xxxx.out が生成されます。
- ・ 必ず、コンパイルエラーやリンクエラーが無いが、console, problem window でチェックします。

3.5. LED 点滅Exampleの動作確認

必要な機器を、下記の様に接続します。

[P C] ---<USB>--- [MSP-FET430UIF] ---<JTAG>--- [評価ボード : MSP-TS430PZ5X100+F5438A]

メニューバー上の虫マーク (Debug Launch) をクリックするか、Target-> Debug Active Project により、評価ボードの Flash-Memory へ Program が書き込まれます。そして CPU が Reset されて Program が start して、main の最初で停止します。

```
int main (void)
{
    ← ここで停止します。
```

Target-> Run あるいは、メニューバーの play マーク |> をクリックすると program が動き、評価ボード上の LED が点滅します。

3.6. Debug例

(1) Debug Example の書き込み

上記の project の main.c を下記のように書き直します。青い●は code ではありません。
ヘッダーは各デバイス用に直します。

```
#include "msp430f5438A.h"
int a,b=0;

int main(void)
{
    int x;
    WDTCTL = WDTPW + WDTHOLD;           // Stop watchdog timer
    a = 5;
    b += 2*a;
    x = a + b;
    ● }
```

最後の } の左欄をダブルクリックすると青い●が表示されます。これがブレークポイントです。
ブレークポイントは、デバッガー内部でも、同様の操作で設定できます。
この状態で、前項のように Build-> Debug で、ターゲットに書き込みます。

(2) Debug 画面

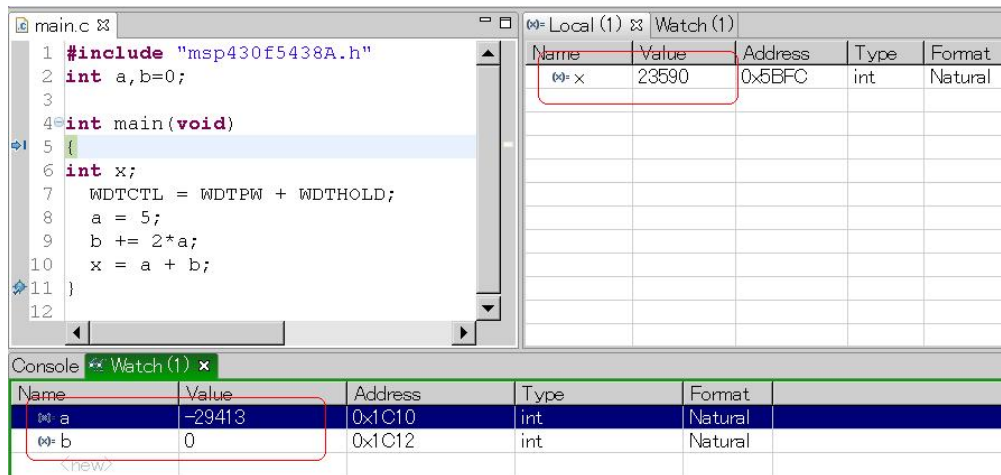
Debug 画面例を下記に示します。

最初は、main の入り口で、停止します。右側にある Local (local 変数) /Watch (global 変数) window で値とアドレスを調べられます。

(例では、変数を同時に表示するため、もう一つ別に watch-window を開いています)

Local 変数は自動で表示されます。

Global 変数は <new>をクリックして a, b 変数名を入力すると表示されます。

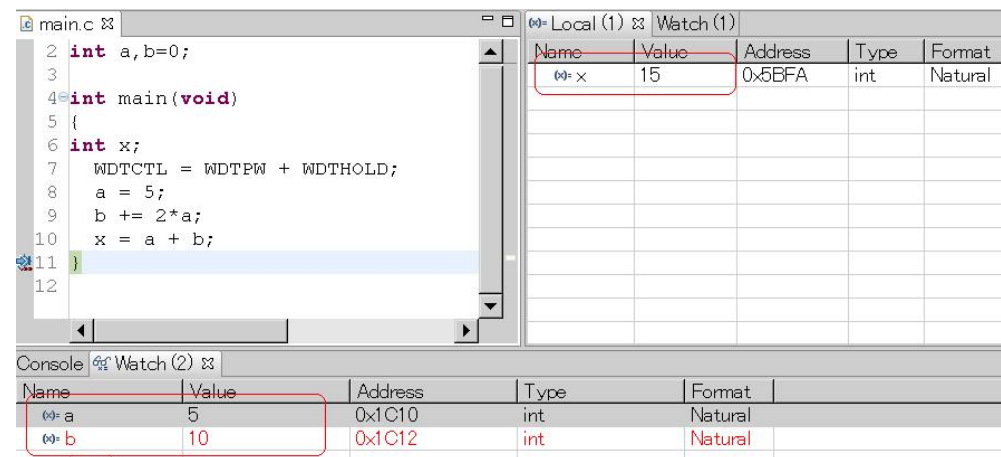


この入口では、変数 b のみ初期値がセットされています。

変数表示 window の FORMAT 欄の Natural を右クリック→Format で表示形式（16 進、10 進など）を変えられます。

さらに Run させると、ブレークポイント設定した main 出口の●で止まります。

画像の例は下記となります。



各変数に値が入りました。変数 b は書き換えが起きたので赤表示されます。

各変数の Value 欄をクリックして、数値を書きかえることができます。

ここで、さらに Run すると、Local 変数 x は消滅し、表示から消えます。

3.7. メモリ配置の確認

プログラムが、メモリ上にどう配置されたかの確認方法について、前項で build した Debug Example の例で、簡単にポイントを説明します。

(1) セクション

最初に入手した下記のマニュアルに、プログラムの配置される「section」の説明があります。

特に関係するのは、code (.text) と variable (.bss) です。

コンパイラマニュアル ・ MSP430 Optimizing C/C++ Compiler v 3.x User's Guide (SLAU132)

Table 4-1. Initialized Sections Created by the Compiler

Table 4-2. Uninitialized Sections Created by the Compiler

(2) リンクコマンド

メモリ名とそのアドレス範囲、セクション名と配置するメモリ名を定義して、object の配置を決めます。リンクコマンドのうち、アプリケーションに関係する部分だけ抜粋したものを下記に示します。

```
<リンクコマンド: xxxxf5438a.cmd>
/* SPECIFY THE SYSTEM MEMORY MAP */
MEMORY
{
    RAM                : origin = 0x1C00, length = 0x4000      配置アドレス、サイズ
    FLASH              : origin = 0x5C00, length = 0xA380
    FLASH2             : origin = 0x10000, length = 0x35C00

    /* SPECIFY THE SECTIONS ALLOCATION INTO MEMORY */
    SECTIONS
    {
        .bss           : {} > RAM                /* GLOBAL & STATIC VARS */
        .stack         : {} > RAM (HIGH)          /* SOFTWARE SYSTEM STACK */
        .text          : {} >> FLASH | FLASH2     /* CODE */
        .text:_isr     : {} > FLASH              /* ISR CODE SPACE */
    }
}
```

変数は .bss:RAM に配置し、コードは .text:FLASH に配置します。

F5438Aは 64KB 以上の部分が FLASH2 と定義されています。

ISRのコードは 64KB 以下に配置されます。

(3) リンクマップ

リンクマップのうち、アプリケーションの code と variable 関係だけ抜粋したものを下記に示します。C言語の初期化ルーチンやライブラリもリンクされるので、itemが多く見にくいですが、ポイントは下記です。特に、Symbol map には、ヘッダー定義されたシンボルも表示されるので、editor search などで見つけます。

```
<リンクマップ: xxxx.map>
MEMORY CONFIGURATION
-----
name          origin      length      used      unused
-----
RAM           00001c00      00004000      000000b4      00003f4c      配置アドレス、サイズ、使用サイズ
FLASH        00005c00      0000a380      000000be      0000a2c2
FLASH2       00010000      00035c00      00000000      00035c00

SECTION ALLOCATION MAP
output
section  page   origin      length      attributes/
-----
.bss     0       00001c00      00000014      UNINITIALIZED
              00001c10      00000004      main.obj (.bss)
.text    0       00005c32      0000008c
              00005c78      0000002a      main.obj (.text:main)

address    name
-----
00001c10   a
00001c12   b
00005c78   main
```

.bssにmainの2変数 a, b (2words) が配置され、.textには main-code が 0x5c78 番地から 2a バイト配置されています。同じことが、symbol-map から分かりますが、変数は debugger でアドレス確認したほうが早いです。

3.8. その他の留意点

(1) プロジェクトの移植

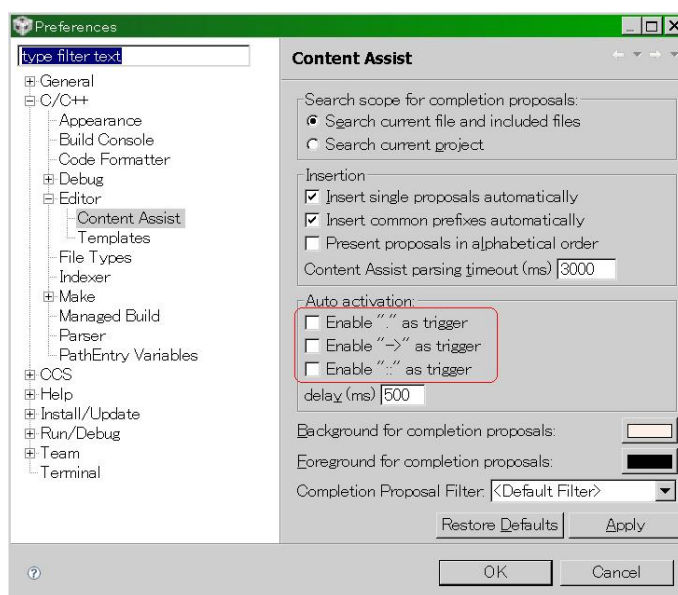
他のマシンで開発した Project を別のマシンに移植するときは、Project フォルダ配下をコピーして、Project-> Import Existing CCS... を使い、Project を取り込みます。

留意すべきは、3.2. 項で説明した CGT-version が異なる場合は、警告メッセージが出ますので、移植先の CGT-version に再設定して、build し直す必要が出てきます。2 台の CCS v 4 の事前バージョン管理が必要となります。

Workspace ごとコピーして、Switch Workspace で移植するのは、避けたほうが無難です。Workspace には、その PC の開発状況・設定状況などの情報も書かれていますので、間違いを引き起こす恐れがあります。

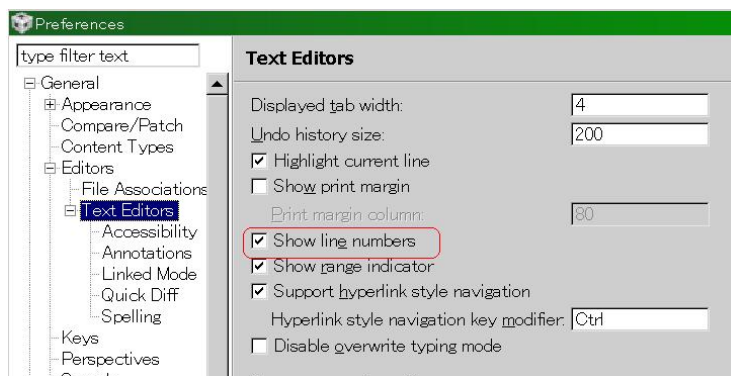
(2) CCS v 4 の速度アップ設定

Windows-> Preferences、C/C++ -> Editor -> Content Assist を選択、Auto Activation 欄の 3 つのチェックを外します。これでサーチが停止しますので、Editor でソース修正時の速度が上がります。



(3) リスト行番号の表示

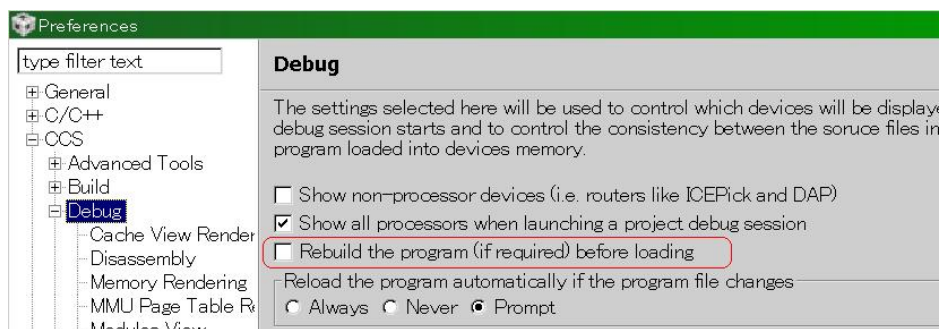
同じ Preferences 画面で、General-> Editors-> Text Editors : Show Line Numbers にチェックします。
(ただし、行番号は印刷されません)



(4) 自動 Rebuild の停止

Debug 時の Program Load のとき、デフォルト設定では自動で Rebuild されます。

自動 Rebuild を止める方法は、下記の Rebuild the program のチェックをはずして停止できます。



以上

ご注意

日本テキサス・インスツルメンツ株式会社（以下TIJといいます）及びTexas Instruments Incorporated（TIJの親会社、以下TIJないしTexas Instruments Incorporatedを総称してTIといいます）は、その製品及びサービスを任意に修正し、改善、改良、その他の変更をし、もしくは製品の製造中止またはサービスの提供を中止する権利を留保します。従いまして、お客様は、発注される前に、関連する最新の情報を取得して頂き、その情報が現在有効かつ完全なものであるかどうかご確認下さい。全ての製品は、お客様とTIJとの間に取り引契約が締結されている場合は、当該契約条件に基づき、また当該取引契約が締結されていない場合は、ご注文の受諾の際に提示されるTIJの標準販売契約約款に従って販売されます。

TIは、そのハードウェア製品が、TIの標準保証条件に従い販売時の仕様に対応した性能を有していること、またはお客様とTIJとの間で合意された保証条件に従い合意された仕様に対応した性能を有していることを保証します。検査およびその他の品質管理技法は、TIが当該保証を支援するのに必要とみなす範囲で行なわれております。各デバイスの全てのパラメーターに関する固有の検査は、政府がそれ等の実行を義務づけている場合を除き、必ずしも行なわれておりません。

TIは、製品のアプリケーションに関する支援もしくはお客様の製品の設計について責任を負うことはありません。TI製部品を使用しているお客様の製品及びそのアプリケーションについての責任はお客様にあります。TI製部品を使用したお客様の製品及びアプリケーションについて想定されうる危険を最小のものとするため、適切な設計上および操作上の安全対策は、必ずお客様にてお取り下さい。

TIは、TIの製品もしくはサービスが使用されている組み合わせ、機械装置、もしくは方法に関連しているTIの特許権、著作権、回路配置利用権、その他のTIの知的財産権に基づいて何らかのライセンスを許諾するということは明示的にも黙示的にも保証も表明もしておりません。TIが第三者の製品もしくはサービスについて情報を提供することは、TIが当該製品もしくはサービスを使用することについてライセンスを与えるとか、保証もしくは是認するということを意味しません。そのような情報を使用するには第三者の特許その他の知的財産権に基づき当該第三者からライセンスを得なければならない場合もあり、またTIの特許その他の知的財産権に基づきTIからライセンスを得て頂かなければならない場合もあります。

TIのデータ・ブックもしくはデータ・シートの中にある情報を複製することは、その情報に一切の変更を加えること無く、かつその情報と結び付けられた全ての保証、条件、制限及び通知と共に複製がなされる限りにおいて許されるものとします。当該情報に変更を加えて複製することは不正で誤認を生じさせる行為です。TIは、そのような変更された情報や複製については何の義務も責任も負いません。

TIの製品もしくはサービスについてTIにより示された数値、特性、条件その他のパラメーターと異なる、あるいは、それを超えてなされた説明で当該TI製品もしくはサービスを再販売することは、当該TI製品もしくはサービスに対する全ての明示的保証、及び何らかの黙示的保証を無効にし、かつ不正で誤認を生じさせる行為です。TIは、そのような説明については何の義務も責任もありません。

TIは、TIの製品が、安全でないことが致命的となる用途ないしアプリケーション（例えば、生命維持装置のように、TI製品に不良があった場合に、その不良により相当な確率で死傷等の重篤な事故が発生するようなもの）に使用されることを認めておりません。但し、お客様とTIの双方の権限有る役員が書面でそのような使用について明確に合意した場合は除きます。たとえTIがアプリケーションに関連した情報やサポートを提供したとしても、お客様は、そのようなアプリケーションの安全面及び規制面から見た諸問題を解決するために必要とされる専門的知識及び技術を持ち、かつ、お客様の製品について、またTI製品をそのような安全でないことが致命的となる用途に使用することについて、お客様が全ての法的責任、規制を遵守する責任、及び安全に関する要求事項を満足させる責任を負っていることを認め、かつそのことに同意します。さらに、もし万一、TIの製品がそのような安全でないことが致命的となる用途に使用されたことによって損害が発生し、TIないしその代表者がその損害を賠償した場合は、お客様がTIないしその代表者にその全額の補償をするものとします。

TI製品は、軍事的用途もしくは宇宙航空アプリケーションないし軍事的環境、航空宇宙環境にて使用されるようには設計もされていませんし、使用されることを意図されていません。但し、当該TI製品が、軍需対応グレード品、若しくは「強化プラスチック」製品としてTIが特別に指定した製品である場合は除きます。TIが軍需対応グレード品として指定した製品のみが軍需品の仕様書に合致いたします。お客様は、TIが軍需対応グレード品として指定していない製品を、軍事的用途もしくは軍事的環境下で使用することは、もっぱらお客様の危険負担においてなされるということ、及び、お客様がもっぱら責任をもって、そのような使用に関して必要とされる全ての法的要求事項及び規制上の要求事項を満足させなければならないことを認め、かつ同意します。

TI製品は、自動車用アプリケーションないし自動車の環境において使用されるようには設計されていませんし、また使用されることを意図されていません。但し、TIがISO/TS 16949の要求事項を満たしていると特別に指定したTI製品は除きます。お客様は、お客様が当該TI指定品以外のTI製品を自動車用アプリケーションに使用しても、TIは当該要求事項を満たしていなかったことについて、いかなる責任も負わないことを認め、かつ同意します。

Copyright © 2011, Texas Instruments Incorporated
日本語版 日本テキサス・インスツルメンツ株式会社

弊社半導体製品の取り扱い・保管について

半導体製品は、取り扱い、保管・輸送環境、基板実装条件によっては、お客様での実装前後に破壊/劣化、または故障を起こすことがあります。

弊社半導体製品のお取り扱い、ご使用にあたっては下記の点を遵守して下さい。

1. 静電気

- 素手で半導体製品単体を触らないこと。どうしても触る必要がある場合は、リストストラップ等で人体からアースをとり、導電性手袋等をして取り扱うこと。
- 弊社出荷梱包単位（外装から取り出された内装及び個装）又は製品単品で取り扱いを行う場合は、接地された導電性のテーブル上で（導電性マットにアースをとったもの等）、アースをした作業者が行うこと。また、コンテナ等も、導電性のものを使うこと。
- マウンタやはんだ付け設備等、半導体の実装に関わる全ての装置類は、静電気の帯電を防止する措置を施すこと。
- 前記のリストストラップ・導電性手袋・テーブル表面及び実装装置類の接地等の静電気帯電防止措置は、常に管理されその機能が確認されていること。

2. 温・湿度環境

- 温度：0～40℃、相対湿度：40～85%で保管・輸送及び取り扱いを行うこと。（但し、結露しないこと。）

- 直射日光が当たる状態で保管・輸送しないこと。

3. 防湿梱包

- 防湿梱包品は、開封後は個別推奨保管環境及び期間に従い基板実装すること。

4. 機械的衝撃

- 梱包品（外装、内装、個装）及び製品単品を落下させたり、衝撃を与えないこと。

5. 熱衝撃

- はんだ付け時は、最低限260℃以上の高温状態に、10秒以上さらさないこと。（個別推奨条件がある時はそれに従うこと。）

6. 汚染

- はんだ付け性を損なう、又はアルミ配線腐食の原因となるような汚染物質（硫黄、塩素等ハロゲン）のある環境で保管・輸送しないこと。
- はんだ付け後は十分にフラックスの洗浄を行うこと。（不純物含有率が一定以下に保証された無洗浄タイプのフラックスは除く。）

以上