

MPLAB[®] X IDE と MPLAB[®]
XC16 コンパイラの使用法、New
Project の作成法、デバッガの使用法

古橋 武

目次

0.1	MPLAB [®] X IDE と MPLAB [®] XC16 コンパイラの使用法, New Project の作成方法	2
0.2	デバッガの使用法	9

0.1 MPLAB[®] X IDE と MPLAB[®] XC16 コンパイラの使用法、 New Project の作成方法

本稿では Microchip 社が無償提供している統合開発環境 [MPLAB[®] X IDE \(Integrated Development Environment : 統合開発環境\)](#)、および、MPLAB[®] XC16 コンパイラのダウンロード、インストール方法と使用方法の概要を紹介します。

MPLAB[®] X IDE は Microchip 社のホームページ → Design → Development Tools → Software Tools for PIC MCUs and dsPIC DSCs → MPLAB[®] X IDE → Downloads → MPLAB[®] X IDE v5.05 (2018 年 10 月時点) とたどることでインストーラ (MPLABX-v5.05-windows-installer.exe) をダウンロードできます。このインストーラを立ち上げ、インストーラの推奨通りに Next ボタンを押していくことで、MPLAB[®] X IDE をインストールできます。無事インストールに成功すれば、C:\Program Files (x86) のフォルダ内に Microchip という名前のフォルダ、ドキュメントフォルダ内に MPLABXProjects という名前のフォルダが作られています。

同様に、[MPLAB[®] XC16 コンパイラ](#)は Microchip 社のホームページ → Design → Development Tools → Software Tools for PIC MCUs and dsPIC DSCs → MPLAB[®] XC Compilers → Downloads → MPLAB[®] XC16 Compiler v1.35 (2018 年 10 月時点) とたどることでインストーラ (xc16-v1.35-full-install-windows-installer.exe) をダウンロードできます。このインストーラを立ち上げ、推奨通りに Next ボタンを押していくことで、XC16 コンパイラをインストールできる。無事インストールに成功すると、Microchip フォルダ内に xc16\v1.35 という名前のフォルダが作られています。

v1.35\support\dsPIC33F.h フォルダ内には dsPIC33FJ128MC802 用のヘッダファイル p33FJ128MC802.h がダウンロードされています。本モータドライブシリーズの[第 5 章「dsPIC マイコンによる DC モータの回転数制御」](#)、[第 6 章「dsPIC マイコンによるブラシレスモータの回転数制御」](#)の全てのプログラムがこのヘッダファイルを使用します。

次に、XC16 コンパイラと同じページにある 16-bit dsPIC33, PIC24E, PIC24H MCUs: Legacy Peripheral Libraries のインストーラをダウンロードし、[Peripheral Libraries](#) を v1.35 フォルダ内にインストールします。これにより、adc.h, timer.h, spi.h などのヘッダファイルを入手できます。ただし、第 5 章、第 6 章ではこれらのヘッダファイルから筆者が必要箇所を抜粋して、コメントを意識したものを使用します。

第 5 章、第 6 章で解説するソースコードは、本稿と同じ

モータドライブノート

に圧縮フォルダに入れて掲載してあります。この圧縮フォルダには上述のヘッダファイ

0.1. MPLAB® X IDE と MPLAB® XC16 コンパイラの使用方法, New Project の作成方法3

ルも入れてあります。

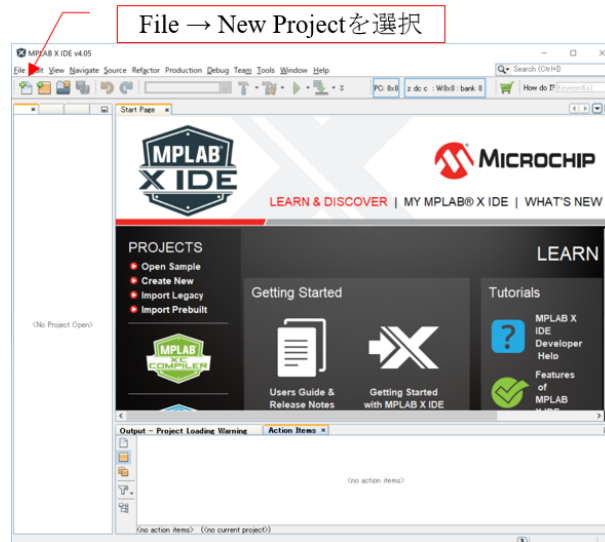


図 1: MPLAB® X IDE の立ち上げと New Project の設定

本稿では、第 5 章で詳述するタイマ 1 による割り込みに関連するファイルを用いて、MPLAB® X IDE による編集、マイコンへの書き込み方法を記します。

MPLAB® X IDE のアイコンを左ダブルクリックすることで、この統合開発環境を立ち上げることができます。図 1 の画面が立ち上がったなら、File → New Project を選択します。次に図 2 のように進み、デバイスとして dsPIC33FJ128MC802 を選択します。

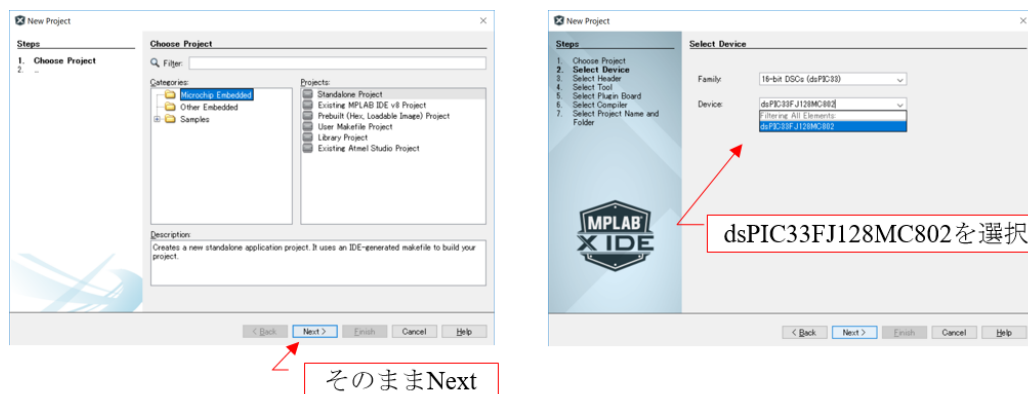


図 2: MPLAB® X IDE: Device 選択

その後は図 3 のように書き込み、デバッグツールとして PICkit3 を選択し、コンパイラに XC16(vx.xx)... を選択します。図 4 は次に表示される画面です。あらかじめ作って

おいたフォルダ（例えば、MPLABXProjects フォルダ内に dsPIC33FJ128MC802 という名前のフォルダを作っておく）をブラウズし、プロジェクト名を自分で決めて（例えば、Timer1_interrupt とします。）入力し、”Set as main project” にチェックを入れて、言語に Shift_JIS を選択します。

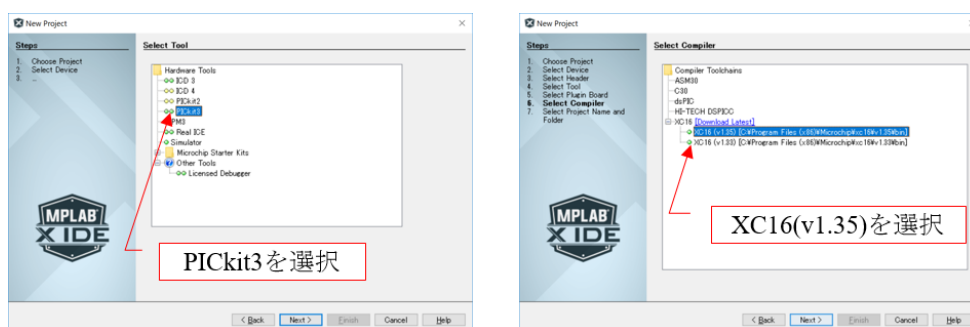


図 3: MPLAB® X IDE: tool, compiler 選択

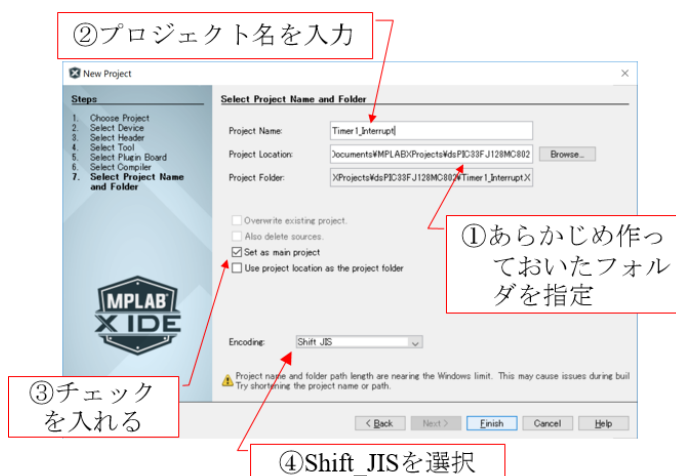


図 4: MPLAB® X IDE: project name, folder, 言語選択

以上が完了した段階で、(上の例では dsPIC33FJ128GP802 のフォルダの中に Timer1_interrupt という名前の) フォルダが作られています。図 5 のように、このフォルダの中へ **モータドライブノート** からダウンロードしておいた「タイマ1による割り込み」フォルダ内のプログラムのソースファイル (Timer1_interrupt.c, timer1.c) およびヘッダファイルの入っているフォルダ (include) をコピーします。

次に、これらのファイルをプロジェクトに追加します。図 6 のように、Source Files の

0.1. MPLAB® X IDE と MPLAB® XC16 コンパイラの使用方法, New Project の作成方法5

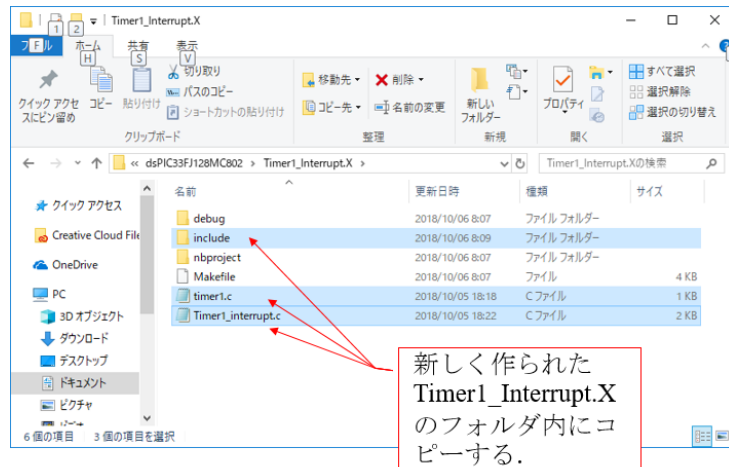


図 5: MPLAB® X IDE: ソースファイル, ヘッダフォルダのコピー

フォルダを右クリックし, ADD Existing Item を選択します. そして, 同図右のように追加したいソースファイルを選択します. 次に, 図 7 の画面のように, Header Files のフォルダを右クリックして, ヘッダファイルを追加します.

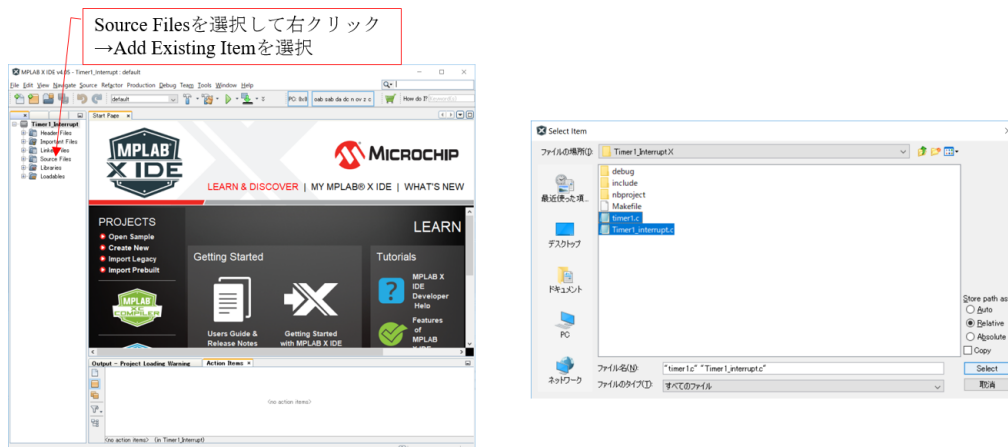
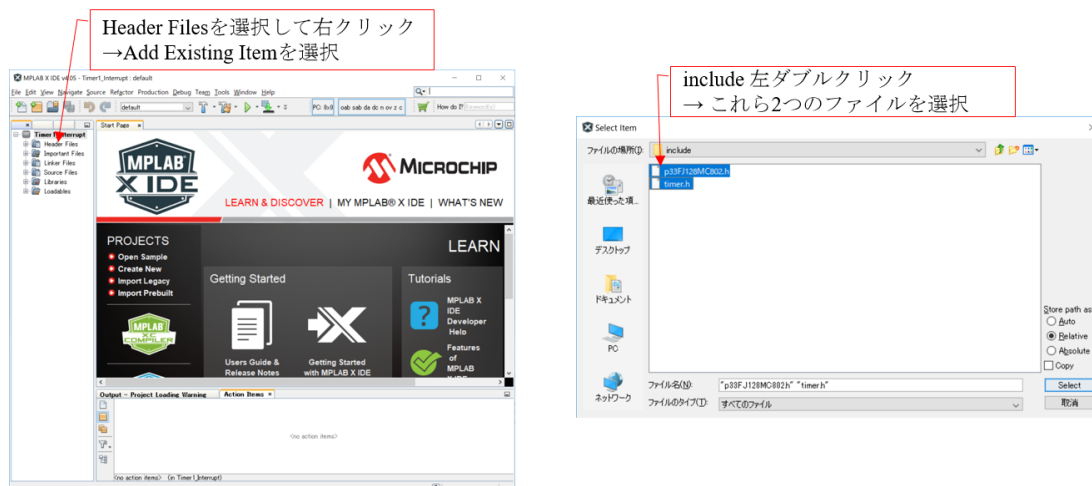


図 6: MPLAB® X IDE: Source の追加

図 7: MPLAB[®] X IDE: Header の追加

0.1. MPLAB® X IDE と MPLAB® XC16 コンパイラの使用方法, New Project の作成方法7

以上で MPLAB® X IDE によるプログラムの編集, PIC マイコンへの書き込み準備が完了しました。ブレッドボードの電源ラインに4本の乾電池 (6[V]) もしくは充電電池 (5[V]) の電源を接続して電圧を印加し, ICSP コネクタに PICkit3 を接続してパソコンと USB ケーブルで接続します。そして, 図8のように, 画面内の Timer1_Interrupt.c を左ダブルクリックすることで, このファイル内のプログラムを開くことができます。そして, Make and Program Device Main Project ボタンをクリックすれば, 同図下の表示が現れて, マイコンの12番ピンには図9に示す波形が出力されます。このプログラムはタイマ1により1[kHz]の繰り返し周波数で割り込みモジュールが起動され, 同モジュールはRA4ポートに1(12番ピンに約3[V])を出力します

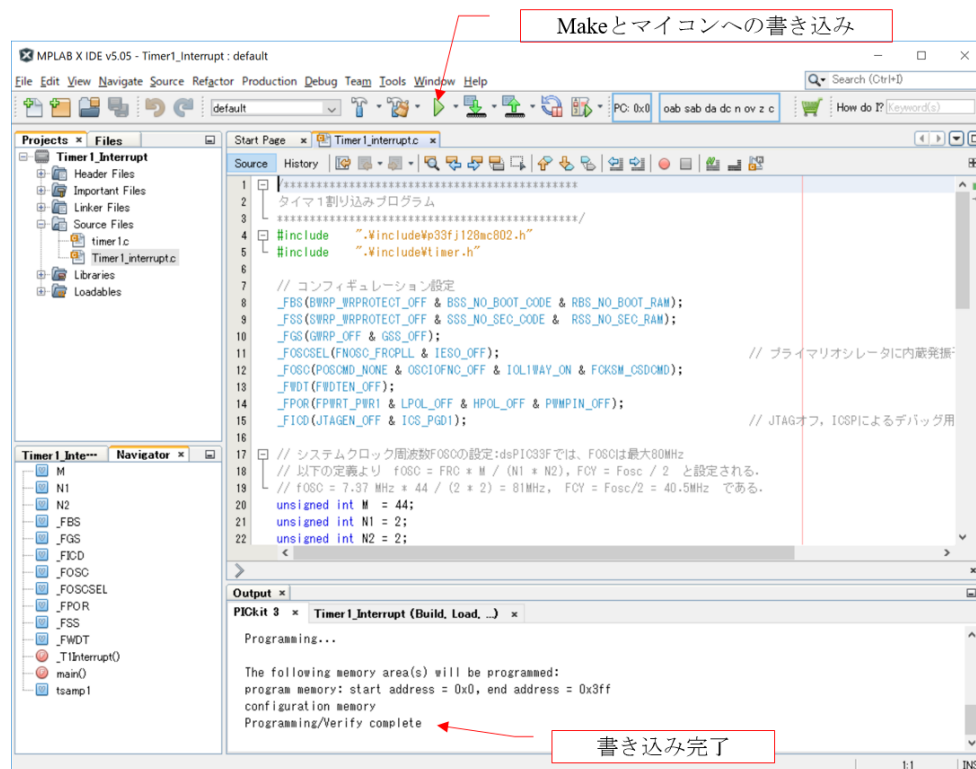


図 8: MPLAB® X IDE: Make とマイコンへの書き込み

図9はオシロスコープ画面のスナップショットです。画面の横軸は25[ns/div], 縦軸は1[V/div]である。画面の右下には観測波形の繰り返し周波数が1.00401[kHz]と表示されています。これはタイマ1による割り込み周波数の計測値です。また, RA4ポートに出力されるパルスの幅(約3[V]の電圧が出力されている期間)は約25[ns]です。このプログラムでは, RA4ポートに1を出力する命令を実行した直後に同ポートに0を出力する命令を実行しています。12番ピンに約3[V]の電圧が出力され, 0を出力する命令の実行

直後に 12 番ピンに 0[V] の電圧が出力されます。パルス幅より 1 命令の実行所要時間が約 25[ns] であることが分かります。データシートによると Device Instruction クロック周波数 (Device Operating 周波数とも呼ばれています。) $FCY = FO_{SC}/2$ である。1 命令の実行時間は $1/FCY = 1/40.5$ [MHz] ≈ 25 [ns] であり、観測結果はこの値と一致します。

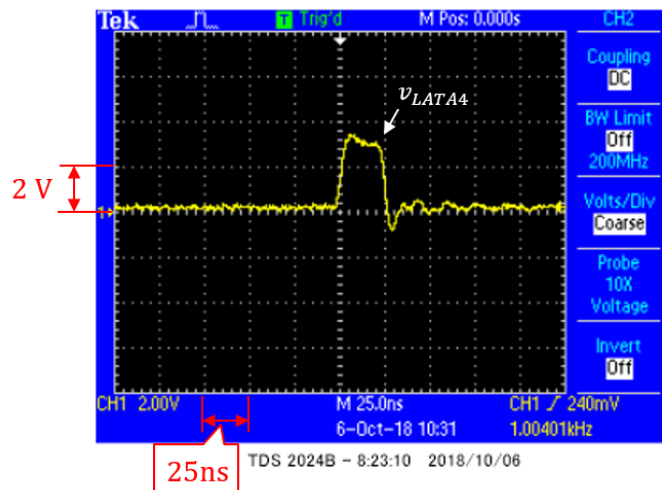


図 9: タイマ 1 割り込みプログラムによる 12 番ピンの出力波形

0.2 デバッガの使用法

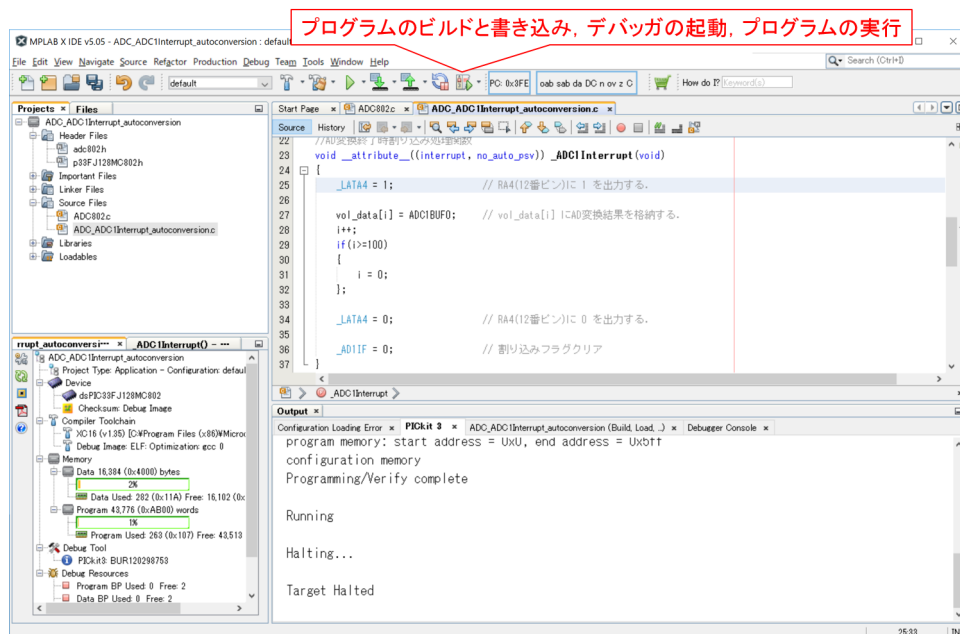


図 10: デバッガの起動

第5章7.4節の Vol_data の値は、MPLAB X IDE のデバッガ機能を利用しました。図10に示すボタンを左クリックすることで、プログラムのビルドとマイコンへの書き込み、デバッガの起動、プログラムの実行を行うことができます。図11の一時停止ボタンにより、プログラム実行を一時停止にすると、vol_data の値を読み出すことができます。図中の Variables のタブを左クリックすると、図12の表示データの入力画面に切り替わります。Vol_data と入力し、図13のように、+ ボタンを左クリックすることで Vol_data の一覧を見ることができます。ただし、データは16進数形式で表示されています。Vol_data を右クリックすると、メニューウィンドウが現れまう。Display Value Column As → Decimal と進んで左クリックすると、データの表示形式が図14のように10進数に切り替わります。もう一度 Vol_data を右クリックし、Export Data → CSV File → Displayed Format と選択して、左クリックすることで、CSV形式でデータをパソコン等にセーブできます。図7.22のグラフは、セーブされたデータをエクセルでグラフ表示したものです。

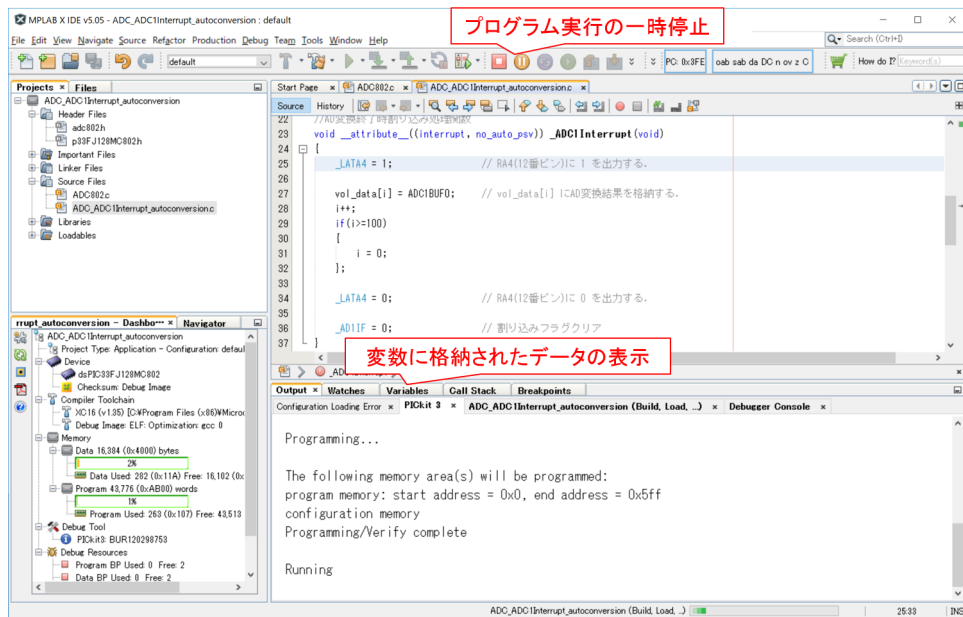


図 11: デバッガの一時停止

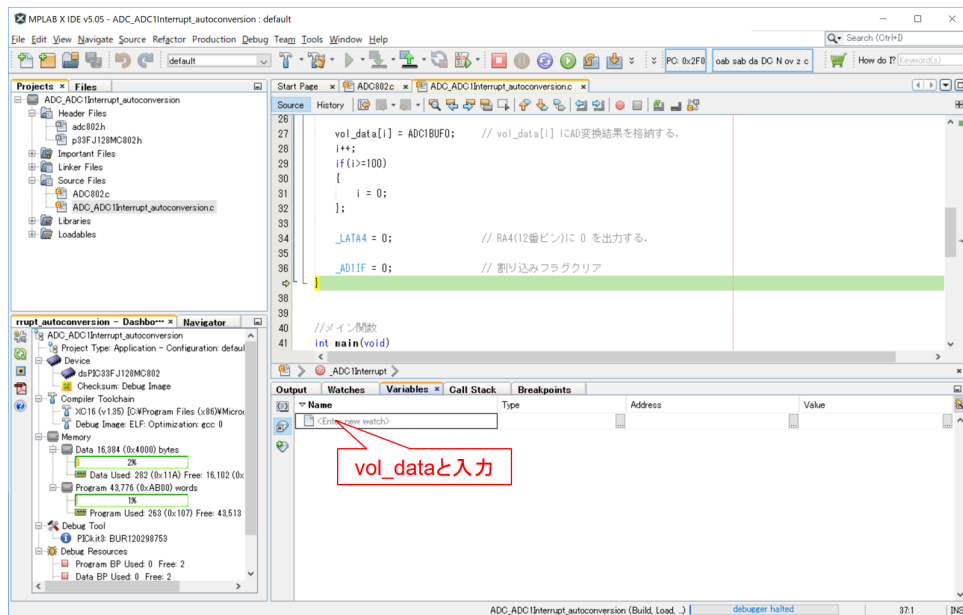


図 12: 表示データの指定

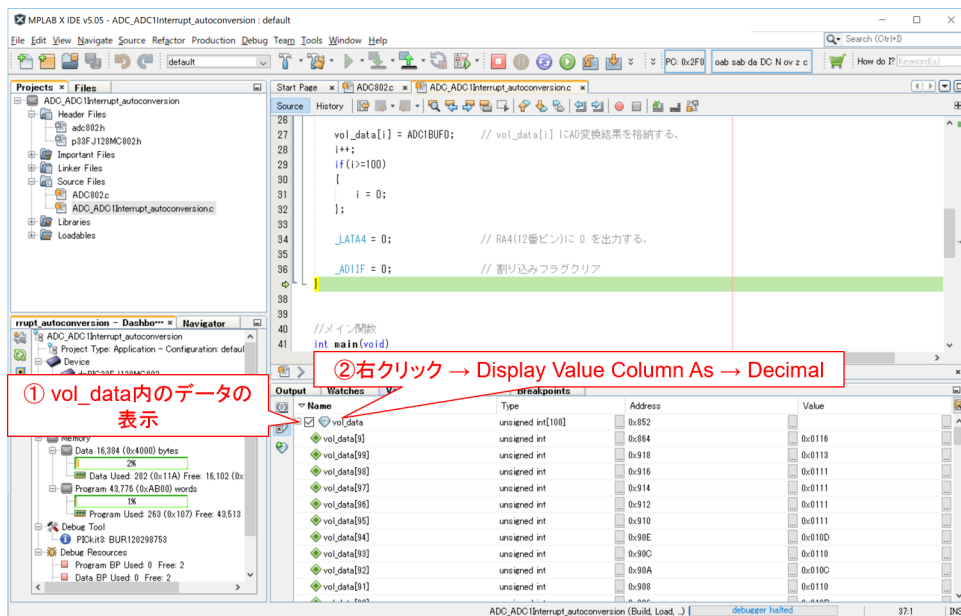


図 13: 表示データ形式を 10 進数に指定

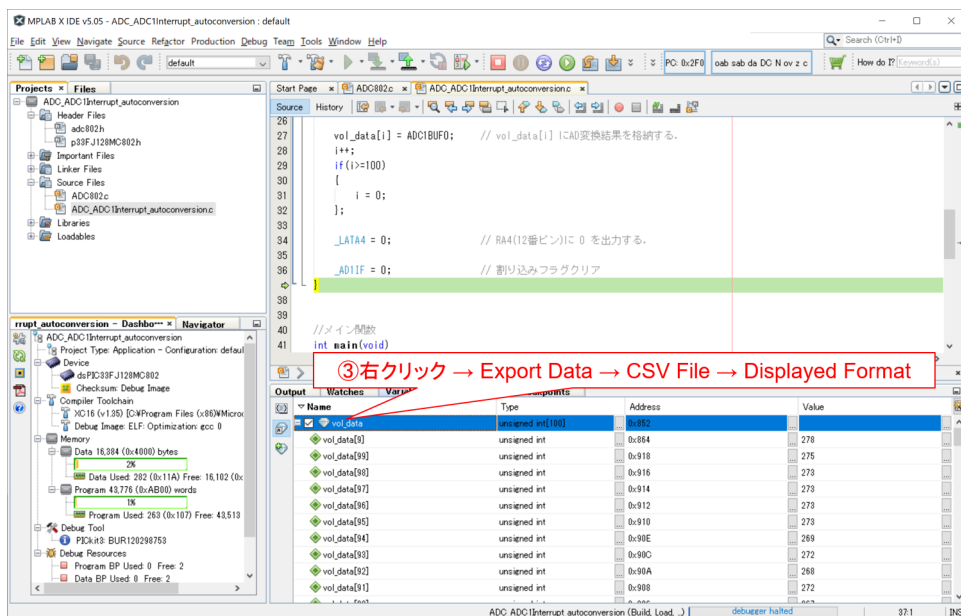


図 14: CSV ファイルへ出力

著者

古橋 武

名古屋大学工学研究科計算理工学専攻

本稿の内容は、著作権法上で認められている例外を除き、著者の許可なく複製することはできません。