

# C-Style for C-cubic

## C-Code 操作ガイド


C-Style バージョン : Ver. 1.1.0


C-Code 操作ガイド : Ver. 1.1.0

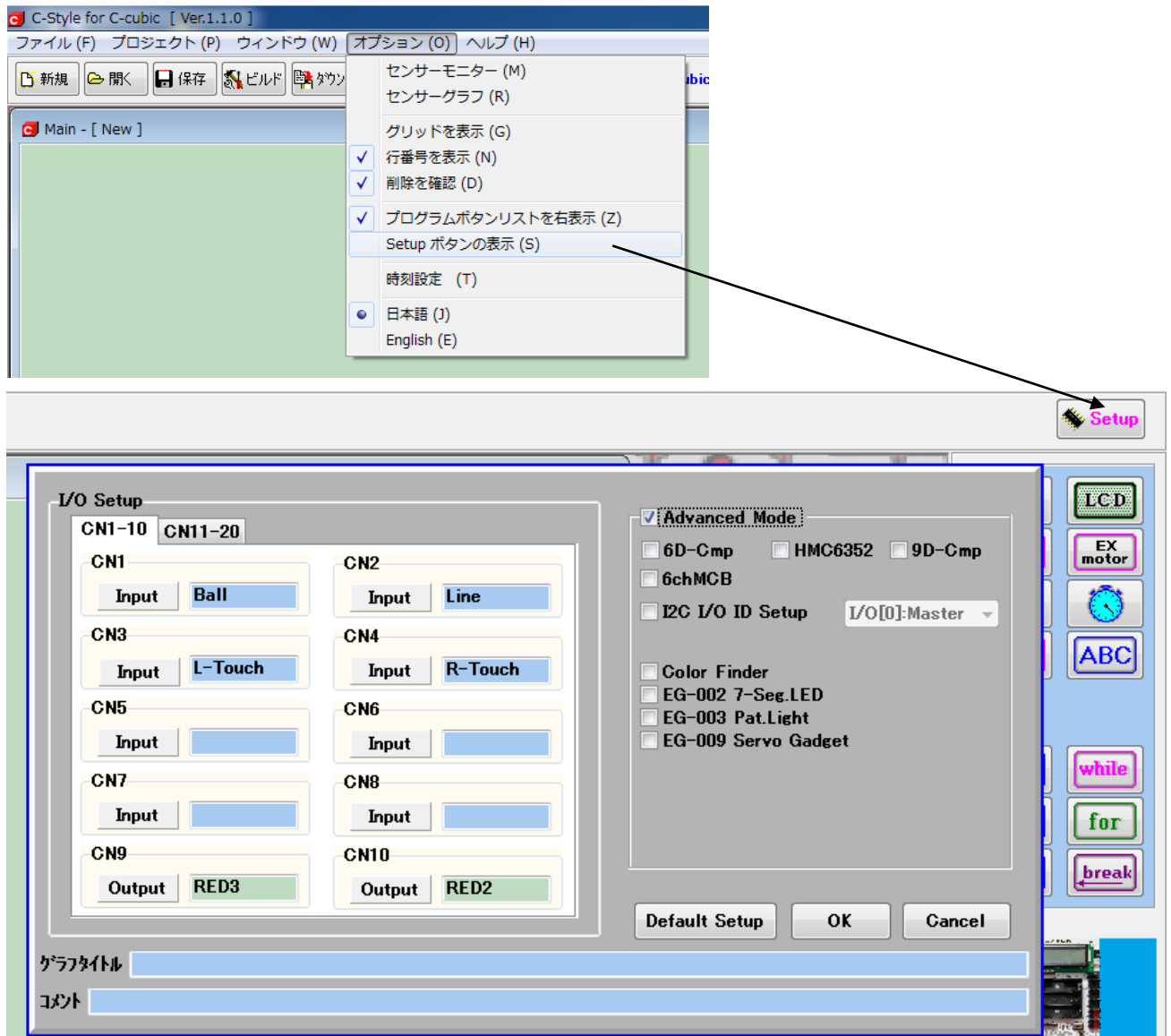
# 目 次

1	C-Code ボタンの説明 .....	2
2	C-Code 編集の説明 .....	7
3	C-Code で超音波距離センサーを使う .....	13
4	C-Code で電子コンパス (HMC6352) を使う .....	14
5	C-Code で多機能電子コンパス (9D-Cmp:DSR1603/6D-Cmp:DSR-1005) を使う .....	14
6	C-Code で SD カードにアクセスする .....	15
6.1	SD 内のファイルを開く .....	15
6.2	SD 内のファイルを読み込む .....	15
6.3	SD 内のファイルにデータを書きこむ .....	16
6.4	SD 内のファイルを閉じる .....	16
6.5	SD 内のファイルポインタを移動する .....	16
6.6	SD 内のファイルを削除する .....	17
6.7	SD 内にフォルダを作成する .....	17
6.8	SD アクセス関数の戻り値 .....	17

## 1 C-Code ボタンの説明

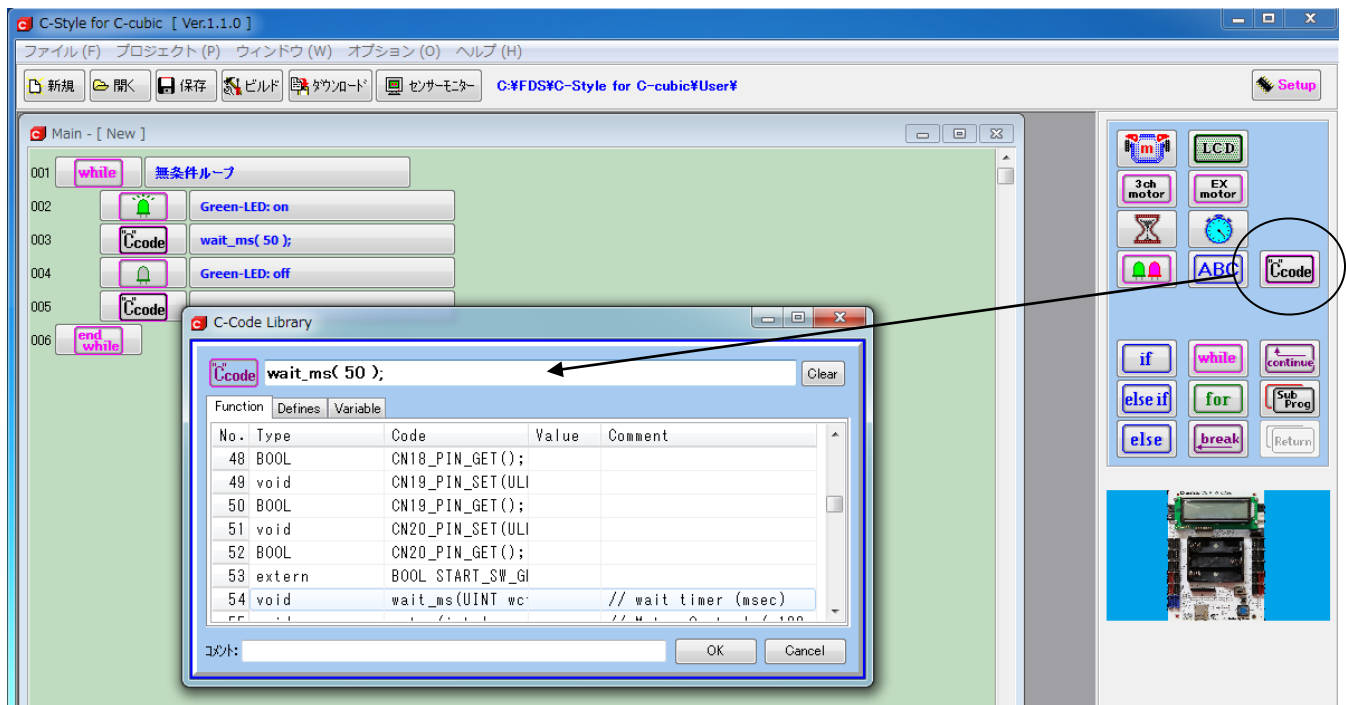
C-Code ボタン  は、C-Style プログラム中に簡単なC言語を直接記述できます。ビルド画面の時に C-Style プログラムボタンからC言語に変換表示されるコードのことです。C-Style に慣れてくると、もう少し高度な記述をしてみたいと思ったことはありませんか？ そんな時に、この「C-Code」ボタンを使って、直接C言語を記述すれば実現できます。

オプションメニューの「Setup ボタンの表示」を選択すると、画面右側のプログラムボタンリスト上部に  ボタンが表示されます。

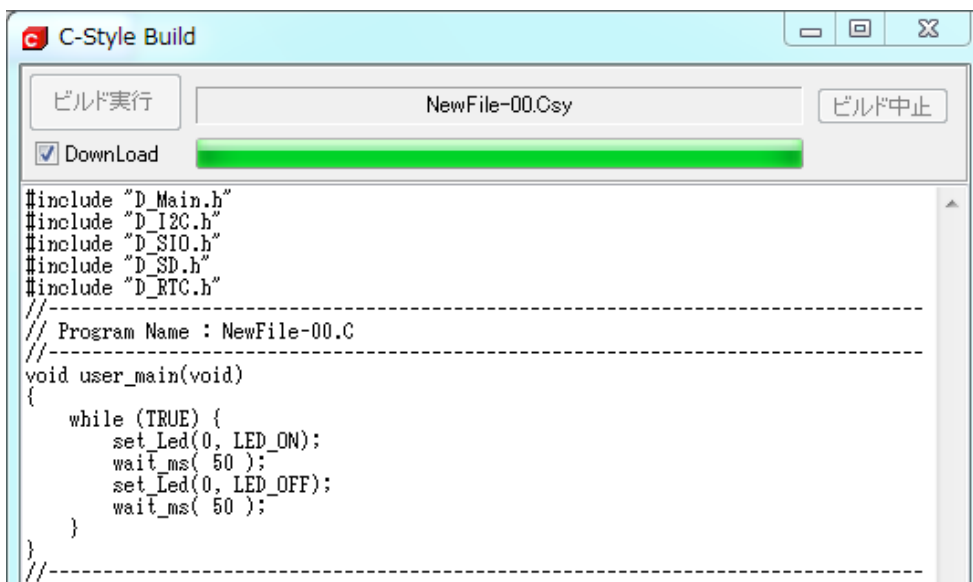


Setup ボタンを表示すると I/O Setup（入出力設定）ダイアログが表示されます。Advanced Mode にチェックを付けて「OK」ボタンでダイアログを閉じると、拡張されたプログラムボタンリストが拡張表示されます。

例えば、C-Style ボタンでは、時間待ちの最小時間は 0.1 秒でしたが、「C-Code」ボタンを使って、直接 C 言語コードを記述することで、1 ミリ秒単位のプログラムが実現できます。

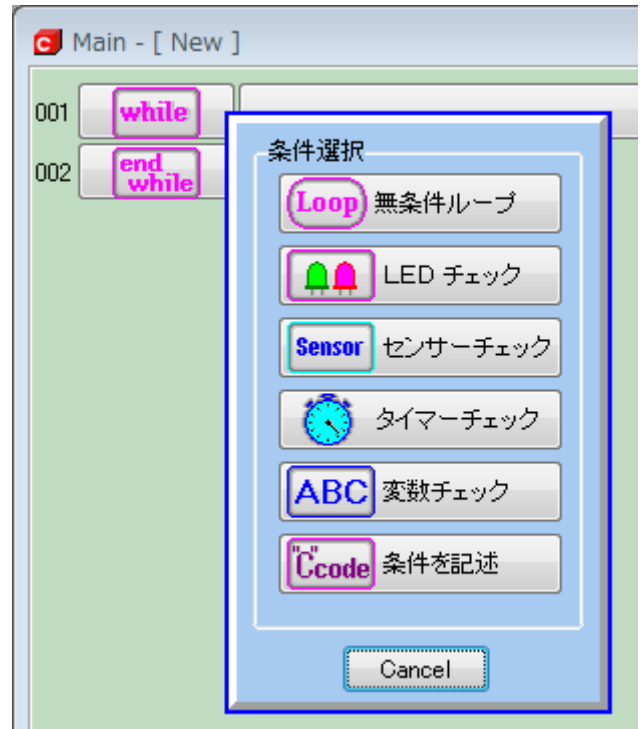


緑色 LED の 50 ミリ秒の高速点滅が出来ます。

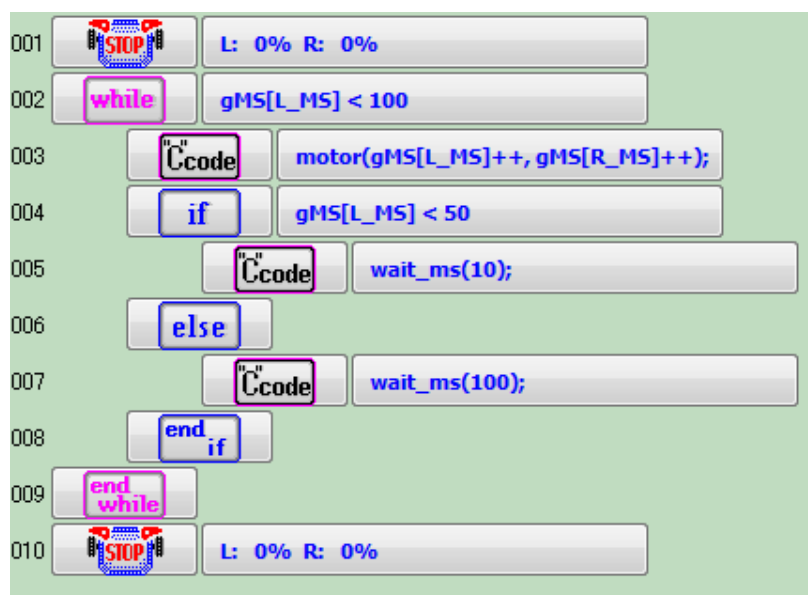


ビルドすると C-Code ボタンで記述されたままのコードが出力されていることがわかりますね！

条件分岐ボタンや条件付き繰り返しボタンにも「C-Code」で条件を直接記述することが出来ます。



while, if 文で C-Code を使った例

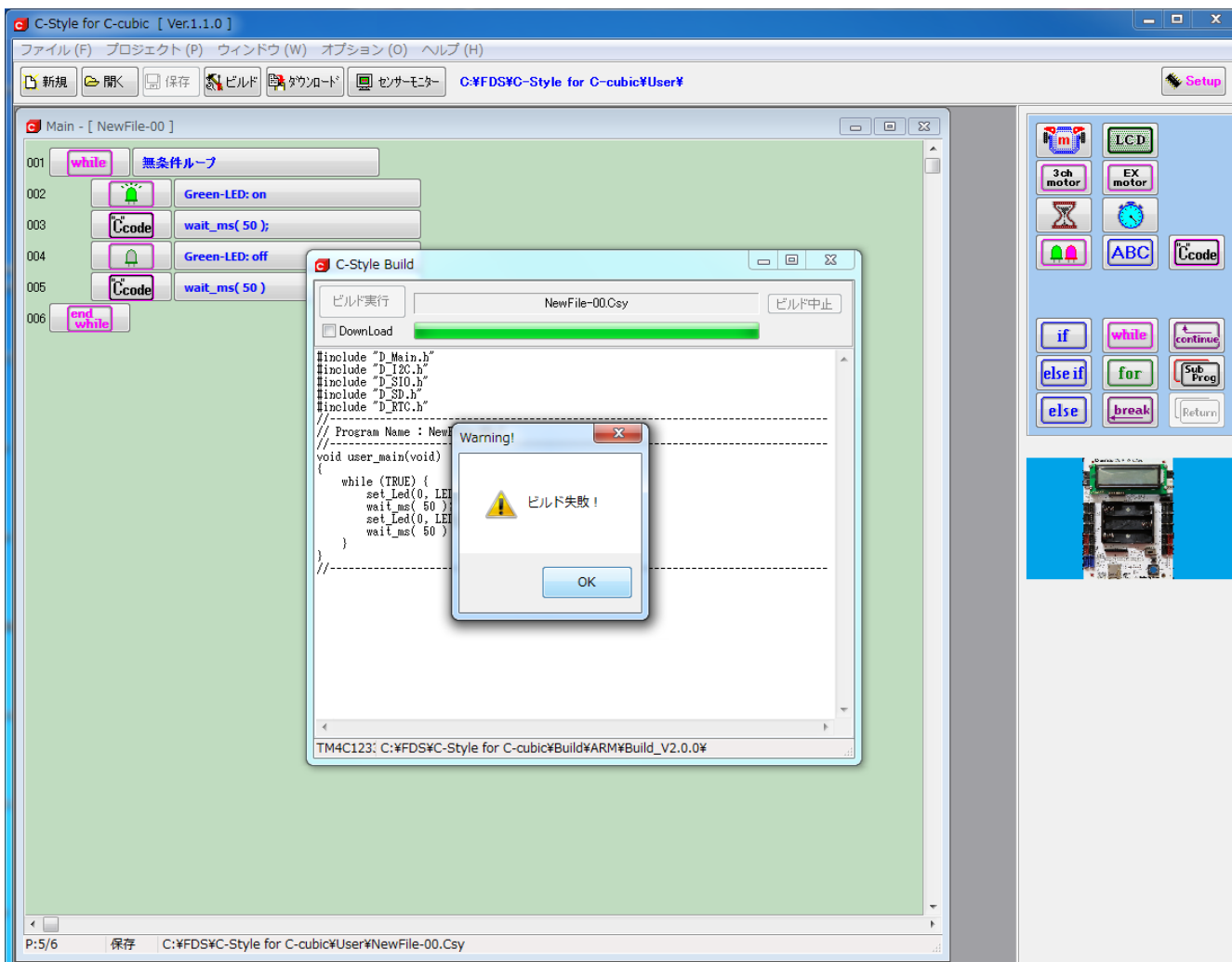


モーター制御関数が保持する現在の速度 `gMS[L_MS]`, `gMS[R_MS]` に+1して100になるまで加速するプログラム例です。

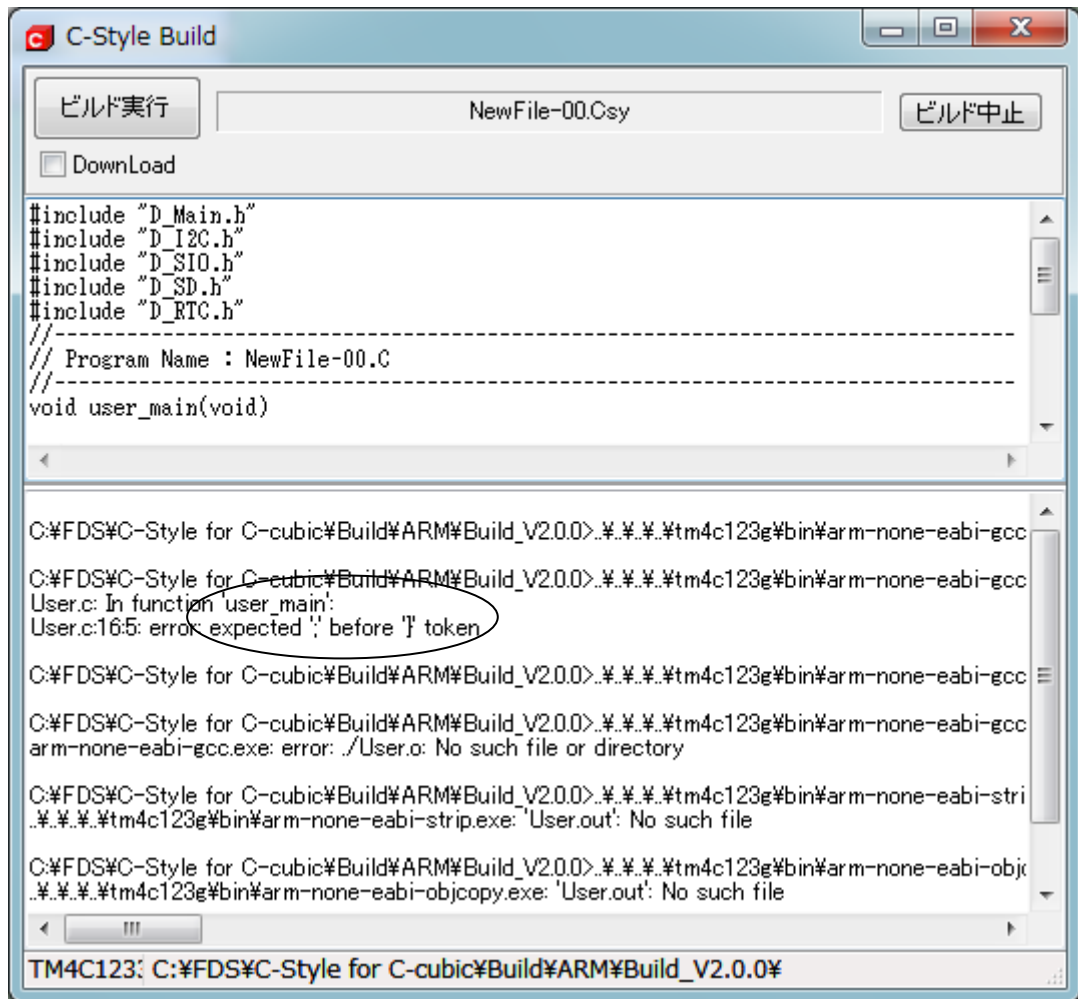
## ■C-Code ボタン使用時の注意

通常の C-Style ボタンだけで作成されたプログラムは、ビルド成功が当たり前でしたが、C-Code ボタンで直接C言語を記述するとタイプミスや、C言語のルール違反でエラーが発生し、ビルド失敗も起こります。

画面の例では、“wait\_ms(50)”の最後に‘;’（セミコロン）が抜けているだけでエラーが発生する例です。



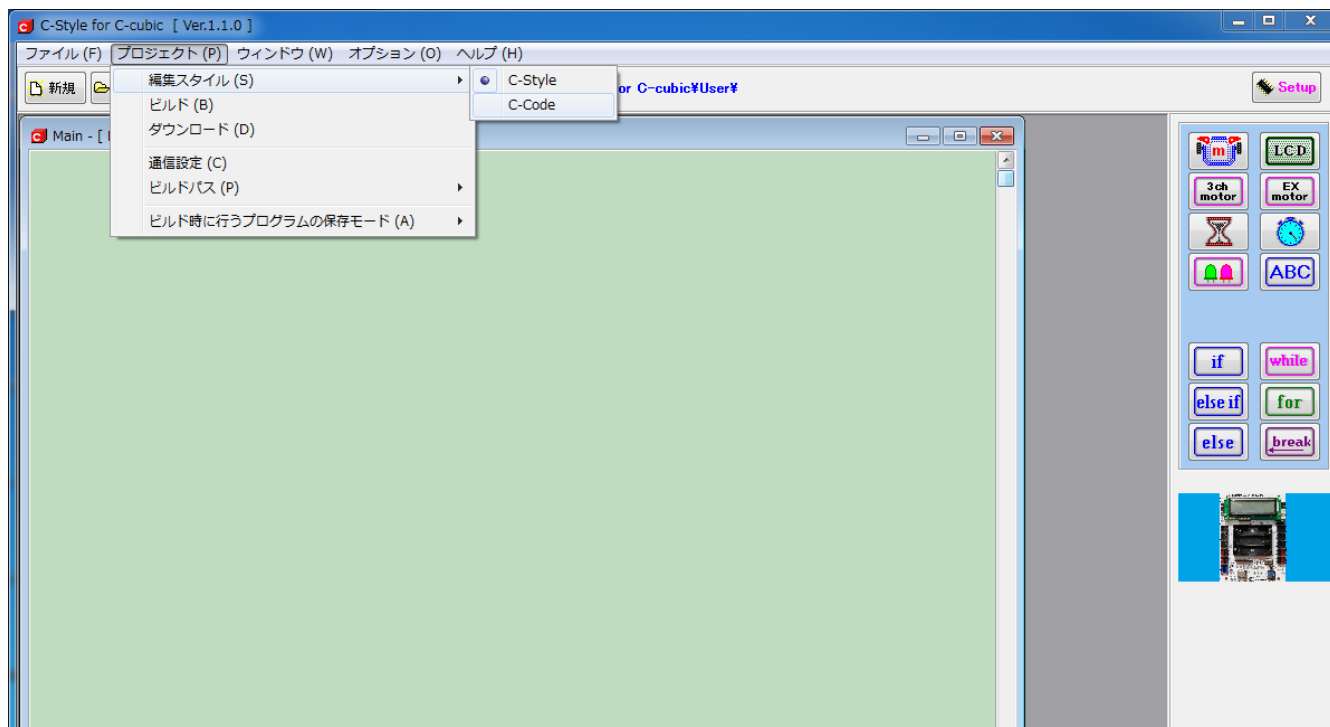
ビルド失敗のダイアログが表示され「OK」ボタンをクリックするとビルド画面は閉じないで、エラー表示をします。



この場合は、「ビルド中止」ボタンをクリックして一旦ビルド画面を閉じてから、問題の箇所を修正します。再度ビルド実行し、成功するまで繰り返します。

## 2 C-Code 編集の説明

Setup ボタン  が表示されている場合、プロジェクトメニューに編集スタイルの変更メニューが追加され、C-Style 編集または C-Code 編集を選択することが出来ます。

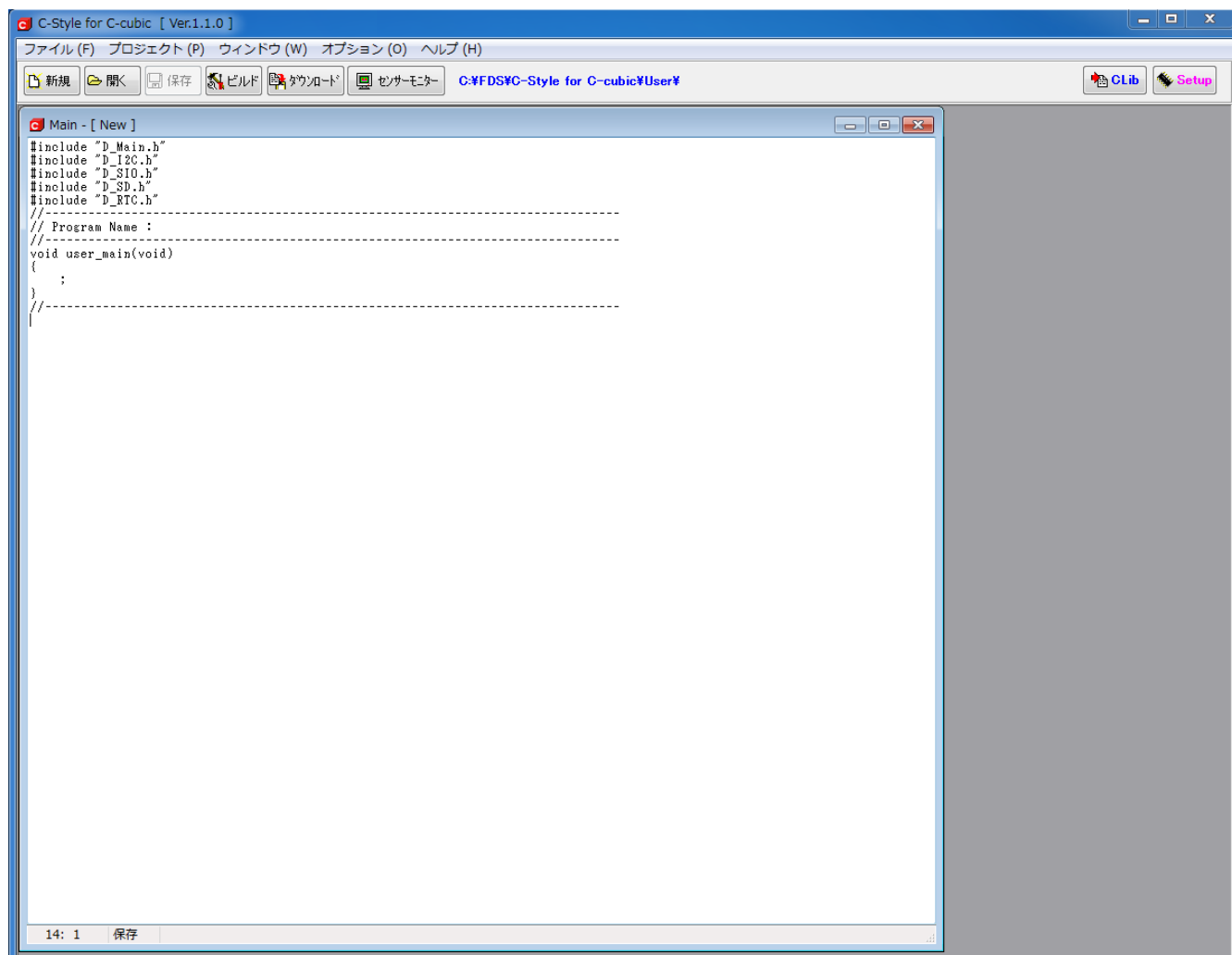





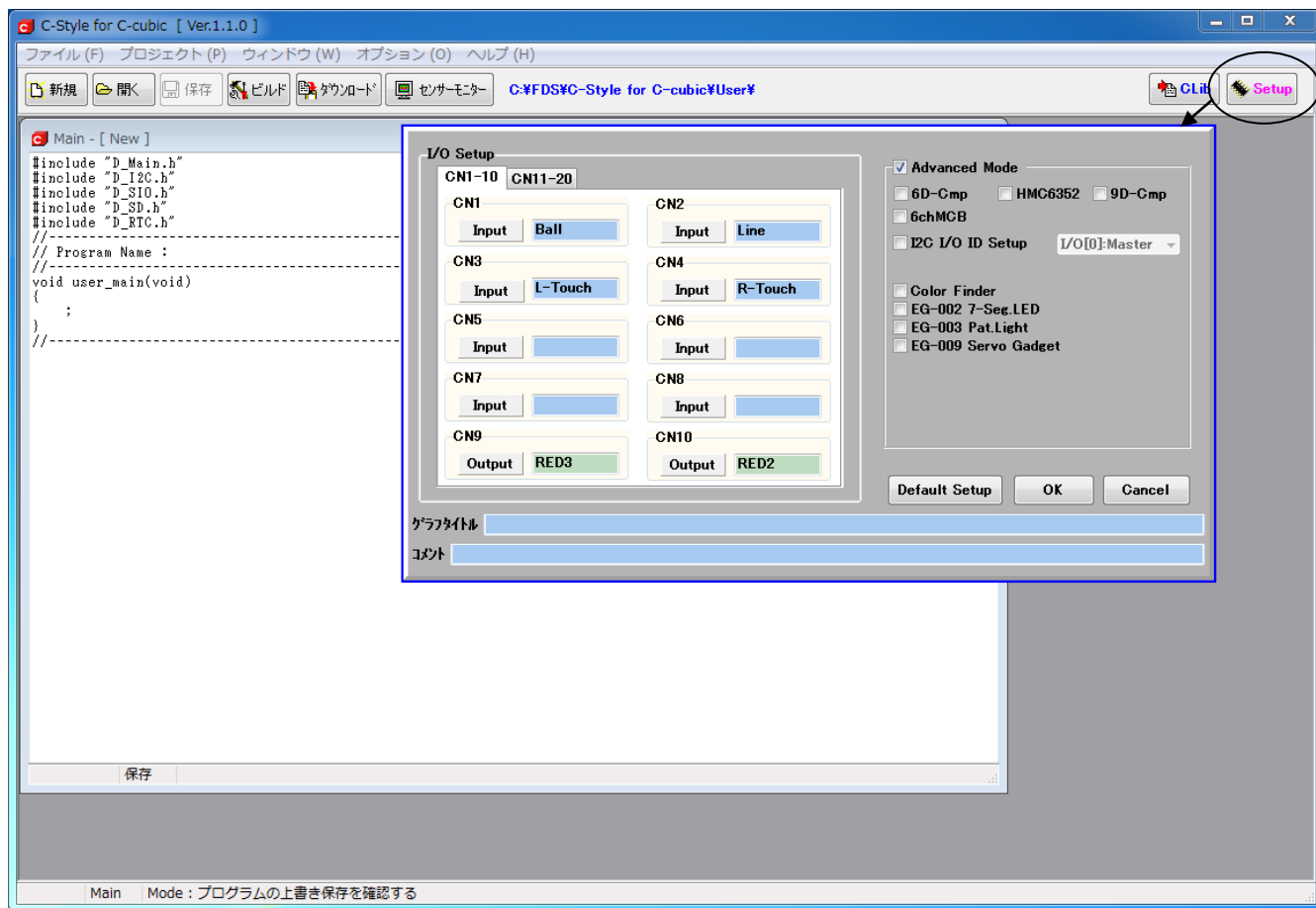
編集スタイルメニューの C-Code 編集を選択すると、C-Style のプログラムボタンリストの表示が無くなり、全て C 言語の編集となります。

この画面で直接 C 言語のコードを記述するか、または、別のテキスト編集ソフトで編集した C 言語ソースファイルを開くことも出来ます。

ビルド及びダウンロードは C-Style 同様に行うことが出来ます。

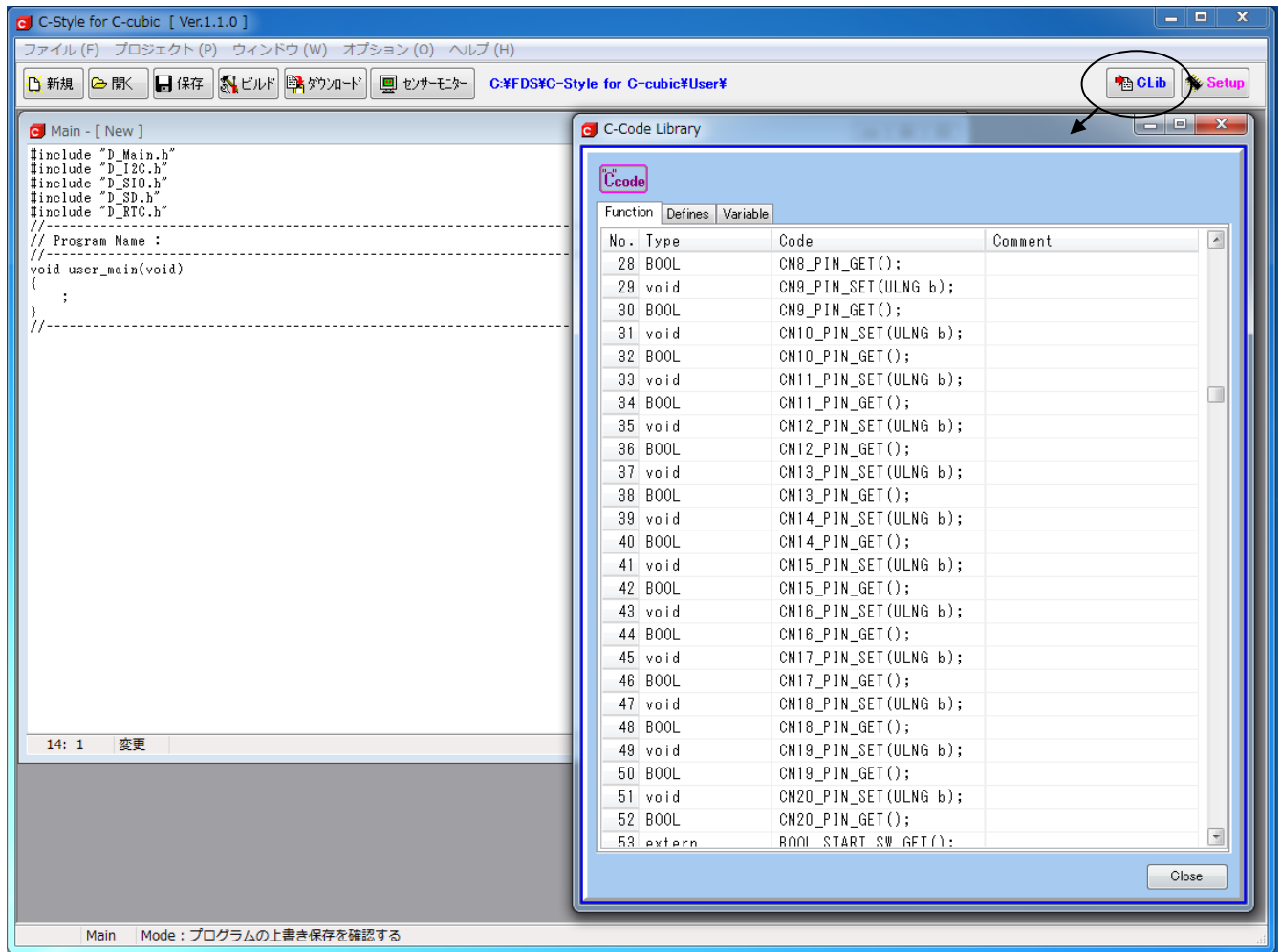


C-Code 編集モードでの入出力設定は  ボタンで行います。



入出力設定は該当するファイルを開いた時、または新規作成した場合は「Default 設定」に初期化されますので、標準の入出力で無い場合は目的のプログラムに合う設定を行ってください。

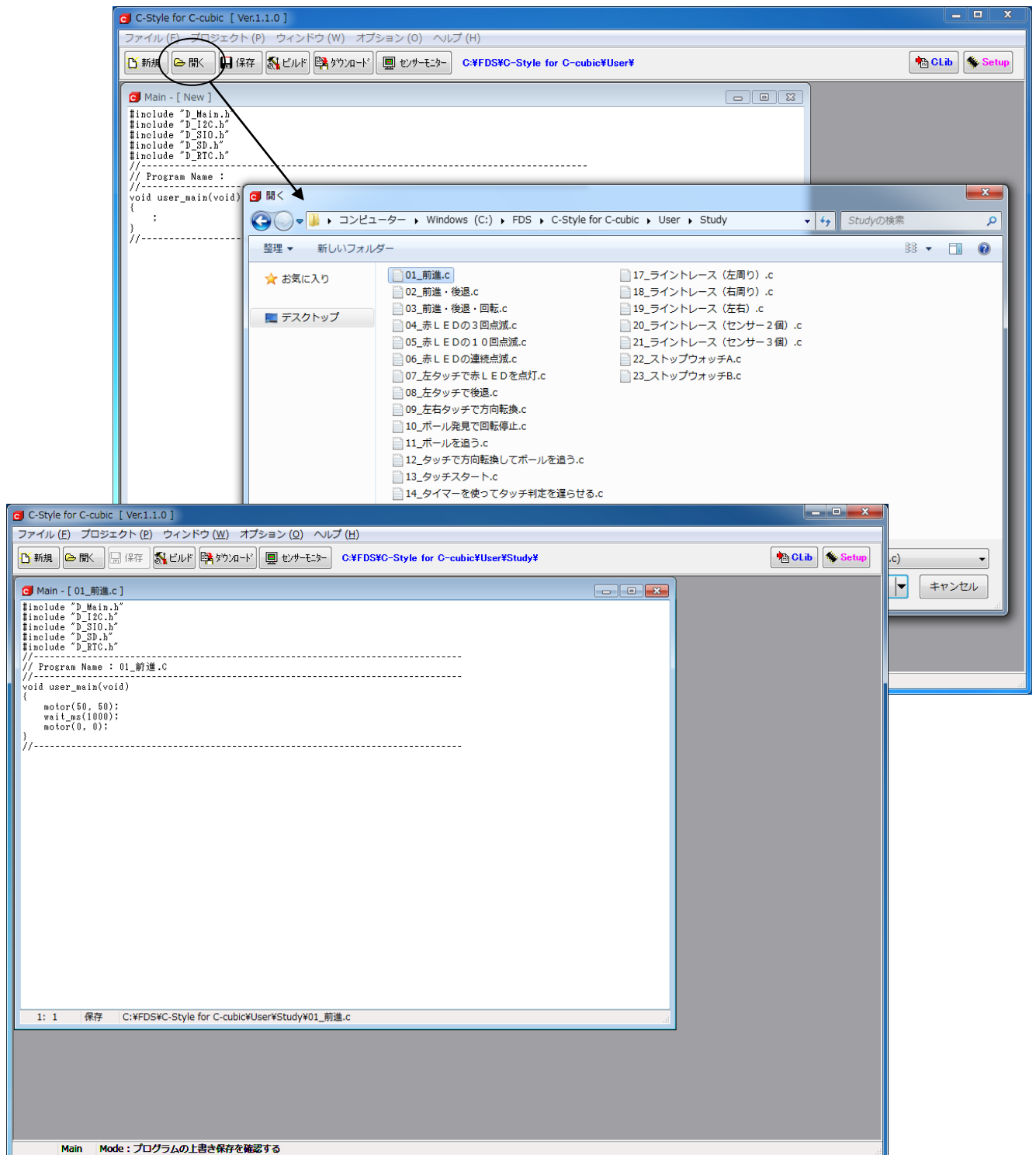
## C-Code Library の表示



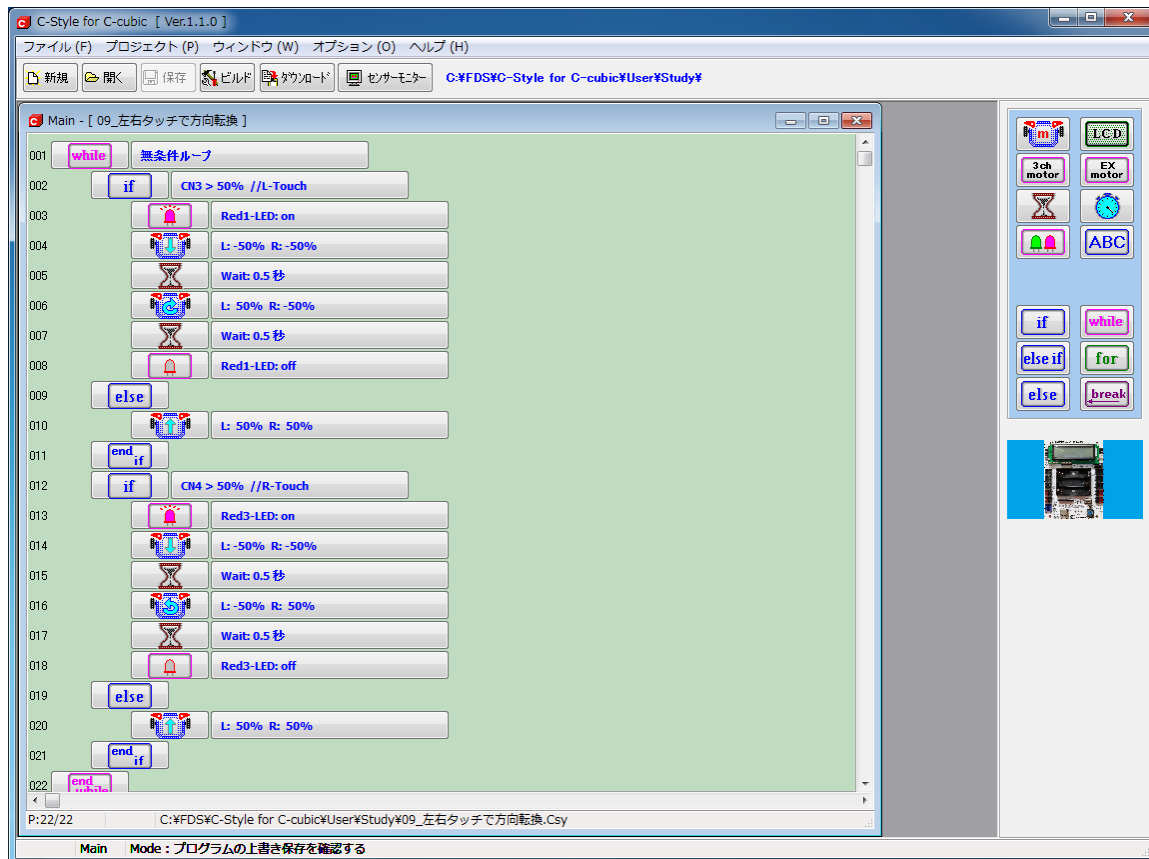
C-cubic で使用できる関数および定義、外部変数などのライブラリを常に表示しながら C-Code の編集をサポートできます。該当する関数などにカーソルを移動してダブルクリックすると、C-Code 編集領域に転記されます。

C-Style でビルドしたファイルは常にC言語ソースファイルとして残されていますので、1 からC言語を記述することなく、C-Code 編集が行えます。

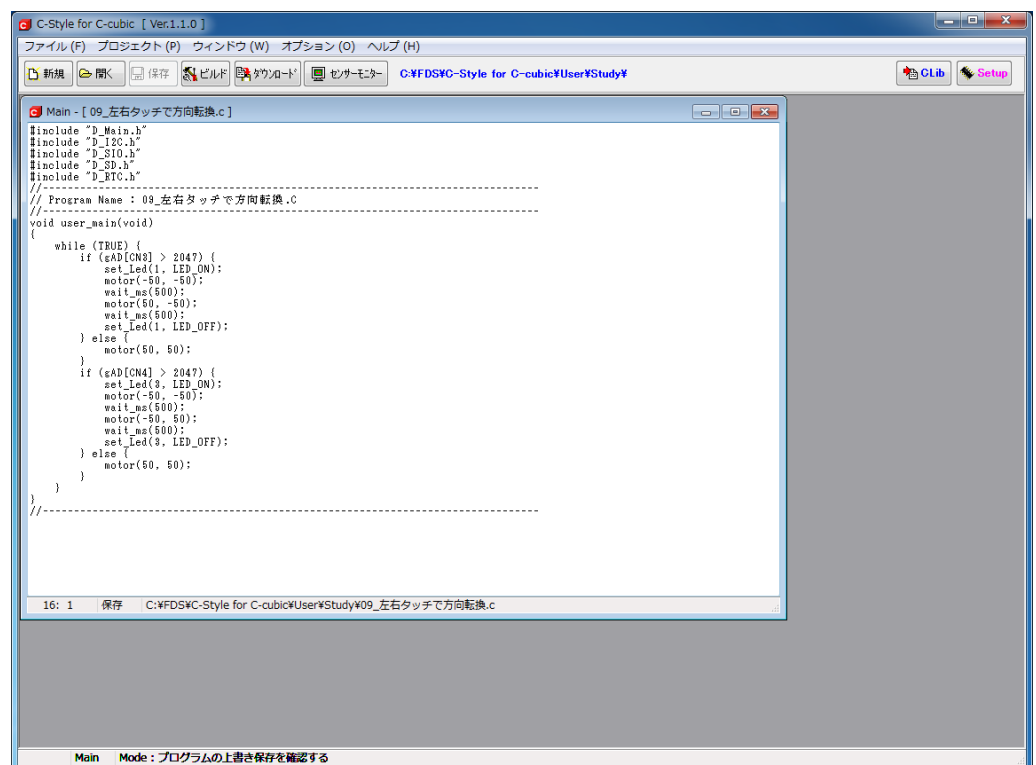
### C-Style の1秒前進プログラムのC言語ソースファイルを開く



## 付属の C-Style プログラムをビルドする



## C-Code 編集に切替えてから C 言語ソースコードを開く



### 3 C-Code で超音波距離センサーを使う

関数名は `UINT get_ping(BYTE pno)`

I/O Setup 画面の CN10～CN7 までを PING 設定に変更します。

計測には 20mSec プログラムは占有されます。(CN10～CN7 の 4 個の場合  $20\text{mSec} \times 4 = 80\text{mSec}$ )

pno パラメーターは、計測した CN 7 から CN 10 の内、戻り値としてほしい CN 番号を指定  
定数 : CN7～CN10 (CN1～CN10 は D\_Main.h で 0～9 の値が定数定義されています)

例 : PING 設定が CN10～CN7 までの場合

```
if (get_ping(CN10) < 100) motor(0, 0); // CN10 の値が 10cm 以下で停止
if (get_ping(CN9 ) < 100) motor(50, 0); // CN9 の値が 10cm 以下で右旋回
if (get_ping(CN8 ) < 100) motor(0, 50); // CN8 の値が 10cm 以下で左旋回
if (get_ping(CN7 ) < 100) motor(-50, -50); // CN7 の値が 10cm 以下で後退
:
:
```

入出力設定で PING 設定された CN の `gAD[n]` の値は `get_ping(n)` を呼出した時に計測され  
A/D の値として格納されています。

PING 設定が CN10 のみの場合は、`gAD[CN10]` だけが超音波距離センサーの計測値となります。

通常の A/D の値は 0～4095 ですが、PING の場合は、0～3000 の値が格納されます。単位 : mm  
但し 30mm 以下は計測不能または未接続、3000mm 以上は 3000mm の測定値となります。

## 4 C-Code で電子コンパス (HMC6352) を使う

I/O Setup 画面の HMC6352 のチェックボックスにチェックマークを付けます。

関数名は `UINT get_hmc(void)`

戻り値は地磁気の方角 (0~359) の値が返されます。(0 度が北の方向です)

例 1 : 地磁気の方角が北 : 0 度付近 (±5 度) になったらモータ停止

```
UINT d;
d = get_hmc();
if (355 < d || d < 5) motor(0,0);
```

例 2 : 地磁気の方角が北 : 180 度付近 (±5 度) になったらモータ停止

```
UINT d;
d = get_hmc();
if (175 < d && d < 185) motor(0,0);
```

ダイセン電子工業以外の HMC6352 を使用する場合は、連続読出しを設定する `BOOL set_hmc(void)` を実行する必要があります (一度実行すればそのセンサーは以後連続読出しの設定を記憶します)

※ダイセン製 (DSR1302) の場合は、出荷時に実行されています。

## 5 C-Code で多機能電子コンパス (9D-Cmp:DSR1603/6D-Cmp:DSR-1005) を使う

I/O Setup 画面の 9D-Cmp または 6D-Cmp のチェックボックスにチェックマークを付けます。

関数名は `UINT get_bno(BYTE dno)` // 6D-Cmp の場合は `UINT get_dir(BYTE dno)`

dno パラメータは、戻り値としてほしいデータ番号 0~2 を指定します。

0: 地磁気の方角 (0~359)	0 度が北
1: 前後の角度 (0~359)	180 度が水平 (6D-Cmp: 0~179, 90 度が水平)
2: 左右の角度 (0~179)	90 度が水平

例 : 地磁気の方角が北 : 0 度付近 (±5 度) になったらモータ停止

```
UINT d;
d = get_bno(0); // 6D-Cmp の場合は d = get_dir(0);
if (355 < d || d < 5) motor(0,0);
```

## 6 C-Code で SD カードにアクセスする

C-Code Library には、SD カードにアクセスする関数が用意されています。

### 6.1 SD 内のファイルを開く

関数名は ULNG fsd\_open (FSD\_FILE \*fp, char\* path, BYTE mode)

戻り値は6.8 SD アクセス関数の戻り値を参照してください。

fp パラメータは、FSD\_FILE 構造体のアドレスを指定します。

path パラメータは、SD 内のファイル名を指定します。ファイル名は8文字+3文字（拡張子）を指定してください。

mode パラメータは、ファイルを開くモード（下記参照）を指定します。

FSD\_READ : ファイル読み込み指定

FSD\_WRITE : ファイル書き込み指定

FSD\_CREATE\_NEW: ファイル新規作成指定

例 1 : SD 内にファイル「test.dat」を新規作成指定で開く

```
ULNG ret;
FSD_FILE fp;
ret = fsd_open (&fp, "test.dat", FSD_WRITE | FSD_CREATE_NEW);
```

例 2 : SD 内のファイル「test.dat」を読み込み指定で開く

```
ULNG ret;
FSD_FILE fp;
ret = fsd_open (&fp, "test.dat", FSD_READ);
```

### 6.2 SD 内のファイルを読み込む

関数名は ULNG fsd\_read (FSD\_FILE \*fp, void\* buff, UINT btr, UINT\* br)

戻り値は6.8 SD アクセス関数の戻り値を参照してください。

fp パラメータは、FSD\_FILE 構造体のアドレスを指定します。

buff パラメータは、読み込むバッファのアドレスを指定します。

btr パラメータは、読み込むデータのサイズを指定します。

br パラメータは、読み込んだデータのサイズを格納するバッファのアドレスを指定します。

例 1 : SD 内にあるオープンしたファイルから、32 バイトのデータを読み込む

```
UINT res;
char buf[64];
ret = fsd_read (&fp, buf, 32, &res);
```



### 6.3 SD 内のファイルにデータを書きこむ

関数名は ULNG fsd\_write (FSD\_FILE \*fp, void\* buff, UINT btw, UINT\* br)

戻り値は6.8 SD アクセス関数の戻り値を参照してください。

fp パラメータは、FSD\_FILE 構造体のアドレスを指定します。

buff パラメータは、書き込むバッファのアドレスを指定します。

btw パラメータは、書き込むデータのサイズを指定します。

br パラメータは、書き込んだデータのサイズを格納するバッファのアドレスを指定します。

例 1 : SD 内にあるオープンしたファイルに対し、32 バイトのデータを書きこむ

```
UINT res;
char buf[64];
ret = fsd_write (&fp, buf, 32, &res);
```

### 6.4 SD 内のファイルを閉じる

関数名は ULNG fsd\_close (FSD\_FILE \*fp)

戻り値は6.8 SD アクセス関数の戻り値を参照してください。

fp パラメータは、FSD\_FILE 構造体のアドレスを指定します。

例 1 : SD 内にあるオープンしたファイルを閉じる。

```
ret = fsd_close (&fp);
```

### 6.5 SD 内のファイルポインタを移動する

関数名は ULNG fsd\_lseek (FSD\_FILE \*fp, ULNG ofs)

戻り値は6.8 SD アクセス関数の戻り値を参照してください。

fp パラメータは、FSD\_FILE 構造体のアドレスを指定します。

ofs パラメータは、ファイルポインタ移動バイト数を指定します。

※ファイルポインタとは、ファイルを入出力する現在位置を示します。

例 1 : ファイルポインタを 10 バイト移動する。

```
UINT res;
ret = fsd_lseek (&fp, 10);
```

## 6.6 SD 内のファイルを削除する

関数名は ULNG fsd\_delete (char \*path)

戻り値は6.8 SD アクセス関数の戻り値を参照してください。

path パラメータは、SD 内のファイル名を指定します。ファイル名は8文字+3文字（拡張子）を指定してください。

例 1 : SD 内のファイル「test.dat」を削除する

```
ULNG ret;
ret = fsd_delete ( "test.dat" );
```

## 6.7 SD 内にフォルダを作成する

関数名は ULNG fsd\_mkdir (char \*path)

戻り値は6.8 SD アクセス関数の戻り値を参照してください。

path パラメータは、SD 内のファイル名を指定します。ファイル名は8文字+3文字（拡張子）を指定してください。

例 1 : SD 内にフォルダ「home」を作成する

```
ULNG ret;
ret = fsd_mkdir ( "home" );
```

## 6.8 SD アクセス関数の戻り値

戻り値	説明
0	成功
4	ファイルが見つかりません
8	ファイルは既に存在しています
9	ファイル/ディレクトリが無効です
12	ドライブが書き込み禁止です

メモ

お問合せ



〒660-0892 兵庫県尼崎市東難波町 5 丁目 21 番 14 号

TEL 06-6401-1211 (FAX 06-6401-1244)

URL <http://www.fdsnet.co.jp/>

Email [Ccubic@fdsnet.co.jp](mailto:Ccubic@fdsnet.co.jp)

※C-Style は株式会社ダイセン電子工業が作成しています。