

```
Visual-Cのインストール
Mlibのダウンロード
Visual-C のプロジェクトの作成
def.hの使い方
ウインドウサイズの変え方
プッシュボタン、チェックボックス、ラジオボタン
メモウインドウ設定、Printf
```

## MS Visual C++ 2010 Express のインストール

<u>https://support.microsoft.com/ja-jp/help/2977003/the-latest-supported-visual-c-downloads</u> 上記リンクからインストール



#### ダウンロード、インストールで10分程度

# mlib のダウンロード

## 三輪研HP <u>http://miwalab.ei.st.gunma-u.ac.jp/mlib.html</u>

三輪研究	三輪空可研究室 發馬大学理工学研究院電子传報調						
トップページ 三輪研紹介	研究内容 研究成果 講義関連ページ <b>0ライブラリ</b> MATLAE講座 リンク集						
<u>코ページ</u> > Mlib ver 3.7							
インメニュー	Mlib ver 3.7 Windows環境でのグラフ描画用Cライブラリ						
🕽 トップページ	目次						
3 三輪研紹介							
3 研究内容	→ <u> </u>						
🖸 研究成果							
🖸 講義関連ページ							
0ライブラリ	→ 使用例						
MATLAB講座	▶ 応用例						
3 リンク集	→ <u>あとがき</u>						
ライブラリベージの更新情報	■ 最新ニュース(2013/9/6)						
2013.11.1全面的に更新しました。	いまた キュービーン どの株式の次約さび日本 キャージャー 2002 ビヤマののバワトをガローのです。						
	millを使ったフロソフミングの講報の資料を公開します。詳しい解結がのるのでせび参考にして下るい。 計測制施工学社会pla						
Nesign by Megapx							
emprate by similaritio com	お陰様で3000ダウンロードを突破しました。どうもありがとうございます。 使いものになるかどうかわかりませんが、						
	どうぞ見ていってください。また、質問、ご意見等ございましたら遠慮なくご連絡ください。						
	はじめに						
	電気系や機械系の学生の皆さん等は学部3年までにC言語の基礎的な演習等を受けてきたと思います。また、C						
	言語による数値演算の講義等も受けてきたかも知れません、C言語を使って数値計算をする場合、おそらくUNIXベー						
	スのOSを使用していて、端末ウインドウでプログラムをコンパイル、実行し、その出力結果は端末ウインドウに現れて						
	いたと思います。もしくは,出力結果をファイルに落として,グラフ描画ソフトで結果を確認というのもあったかもしれま						
	せんね.						
	ところが,いざ研究室に配属されると,いきなり難しい数値シミュレーション等をC言語でやることになって困惑した						
	ことはないですか?しかも,UNIXではなくWindowsペースで・・・シミュレーションのアルゴリズムやそのコーディング						
	を考えるだけでも精一杯なのに, 違うOSでしかも統合開発環境のMicrosoft Visual C++とかC++ Builderを使うことに						
	なったりするともう大変です。まずは統合開発環境とは何ぞやというところから、勉強しなければなりません。						

このページの下の方から mlib.h と def.h (ヘッダーファイル) をダウンロードして、分かりやすいところにおいておく

# mlibのインストール

## 6VPJ7-H3CXH-HBTPT-X4T74-3YVY7

## Visual C++ 2010E を起動 → 新しいプロジェクトを作成



Visual C では一つのプログラム につき一つのプロジェクトが必 要

**プロジェクト** 実行ファイル作成 時に必要なリンカ、デバッガ、コ ンパイラ等のファイルやプログラ ムの実行に必要なインクルード ファイル等をひとまとめにしたコ ンテナ

異なるプログラムを実行するに は新たにプロジェクトを作る

# プロジェクトの新規作成

## 以下に指示するどうりに作らないと、 コンパイルできないので注意

新しいプロジェクト	and the second s		ନ <mark>×</mark>
最近使用したテンプレート	並べ替え基準: 既定	• III 🔲	インストールされたテンプレート の検索 🔎
インストールされたテンプ	<sup>2</sup> レート CLR コンソール アプリケ・	ーション Visual C++	種類: Visual C++ コンソール アプリケーションを作成する
CLR Win32	ਊ 🕁 Win32 コンソール アプリ・	ケーション Visual C++	ためのプロジェクトです。
全般	────────────────────────────────────	Visual C++	>
	Windows フォームア		シェクトを選択
	クラス ライブラリ	VVIIISZ Visual C++	
	メイクファイル プロジェク	ント Visual C++	
	空の CLR プロジェクト	Visual C++	
	空のプロジェクト	Visual C++	
①プ	ロジェクトの名前	「を入力	③OKボタン
名前( <u>N</u> ): <	名前を入力してください		
場所( <u>L</u> ): c:	¥users¥miwa¥documents¥visual studio 2010¥F	Projects 🔹	参照(B)
ソリューション名( <u>M</u> ): <	名前を入力してください>		☑ ソリューションのディレクトリを作成( <u>D</u> )
			OK キャンセル

Win32アプリケーションウイザード



面倒なWindows アプリケーションのひな形となる プロジェクトを簡単に作ってくれるが、、

Win32アプリケーションウイザード

Win32 アプリケーション ウィザ	— К - аа		8 X
<b>ምታባታ</b>	ションの設定		
概要 アプリケーションの設定	<ul> <li>アプリケーションの種類:</li> <li>Windows アプリケーション(W)</li> <li>コンソール アプリケーション(Q)</li> <li>DLL(D)</li> <li>スタティック ライブラリ(S)</li> <li>追加のオプション:</li> <li>空のプロジェクト(E)</li> <li>シンボルのエタスボート(X)</li> <li>プリコンパイル済みヘッダー(P)</li> </ul>	共通ヘッダー ファイルを追加: □ ATL( <u>A</u> ) □ MFC( <u>M</u> )	ゥ
		<b>2</b> 5 〈前へ 次へ〉 完了	<b>も了</b>

## mlibを使用する際は、それ自身がひな形の役割 なので、空のプロジェクトのみ作っておく

# プロジェクトの作成確認

Visual Studio 2010¥Projectsに作成したプロジェクト名のディレクトリ ができていることを確認

## プロジェクトのホームディレクトリは2階層下

ドキュメント¥Visual Studio 2010¥Projects¥プロジェクト名¥プロジェクト名

fしいフォルダー							注	音
ドキュメント	ライブラリ	並べ替え: フォル	ダー 🔻					/Ľ^ ^
名前	更新日時	種類	サイズ		7	下-	-ム	テイ
🛯 Debug	2012/03/12 19:09	ファイル フォルダー						L
aa	2012/03/12 19:08	VC++ Project				クト	<b>、  ]  </b>	Τ
aa.vcxproj	2012/03/12 19:08	VC++ Project Filters File						
🕑 test	2012/03/12 19:09	- C言語ソースファイル						
aa.vcxproj.user	2012/03/12 18:47	USER ファイル						
h) def	2010/11/09 16:56	C/C++ Header	1					
h mlib	2010/08/10 10:48	C/C++ Header	e					

## プロジェクトへのファイルの追加

## 作ったプロジェクトは空のままなので、様々なファイルを追加する必要

#### ソリューションエクスプローラー

ソースファイルやヘッダーファイル 部を<mark>右クリック</mark>でファイルを追加で きる。

◎ソースファイル 主にCソースファイルを新規作成、 もしくは他からコピーしてくれば、 既存のファイルを追加する

#### ◎ヘッダーファイル

mlib.h、def.h は追加必須 その他、Cでよく使うインクルード ファイルがあれば追加

使わない

Oリソースファイル O外部依存関係

aa - Microsoft Visual C++ 2010 Express ファイル(E) 編集(E) 表示(V) プロジェクト(P) デバッグ(D) ツール(I) ウィンドウ(W) ヘルプ(H) - 🖄 🚆 🛅 = 🛅 = 💕 🚽 🥔 👗 🖹 🖺 🖉 - (\* - 🕨 Debug Win32 - 🕒 | 🗗 🌄 ソリューション 'aa' (1 プロジェクト) 🔺 📑 aa 🗀 ソース ファイル 📜 ヘッダー フォ 新しい項目(W)... Ctrl+Shift+A 🔲 リソース ファ ↓ 切り取り(T) Ctrl+X … 既存の項目(G)… Shift+Alt+A 🔚 外部依存関係 コピー(Y) Ctrl+C づ 新しいフィルター(F) 貼り付け(P) Ctrl+V 🦗 クラス(<u>C</u>)... X 削除(D) Del 名前の変更(M) F2 プロパティ(R) 出力 • ₽ × 出力元の表示(S): プロジェクト 'aa' の作成に成功しました。

ヘッダーファイル部に、ホームディレクトリにコ ピーしたmlib.h, def.h を選択し追加する。



ソースファイル部に空のC++ファイル(メインプログラム)を作る ソリューションエクスプローラの[ソースファイル]を<mark>右クリック</mark>で 新たに作成追加できる。

新しい項目の追加 - aa	and the second s		8 ×
インストールされたテンプレート	並べ替え基準: 既定 ▼ 1111 1111		インストールされたテンプレートの検索 🔎
✓ Visual C++ UI	Windows フォーム	Visual C++	種類: Visual C++ 他の Windows コントロールを含む CLR
コード プロパティ シート	◆ C++ ファイル (.cpp)	Visual C++	フォームを作成します。
	▶ ヘッダー ファイル (.h)	Visual C++	
	プロパティ シート (.props)	Visual C++	
	コンポーネント クラス	Visual C++	
<u>(1) プロガニ</u>	192742		
	ム石を入れる	(	2)追加ボタンな
名前( <u>N</u> ): <名前を入力	してください>		
場所( <u>L</u> ): C:¥Osers¥m	iwa¥documents¥visual studio 2010¥Projects¥aa¥aa¥	•	参照( <u>B</u> )
			追加( <u>A</u> ) キャンセル

Visual Studio 2010¥Projects¥プロジェクト名¥プロジェクト名 の下に作成

プログラムの入力、コンパイル

## ②入力後コンパイル実行ボタンを押す

aa - Microsoft Visual C++ 2010 Express

①作成した新規Cファイル をダブルクリックし、編集 以下のプログラムを入力

ファイル(E) 編集(E) 表示(V) プロジェクト(P) <u>味バ</u>ッグ(D) ツール(I) ウィンドウ(W) ヘルプ(H) - 🖄 🗒 🛅 - 🛅 - 🍃 🔲 🎒 👗 📭 🖳 🔊 - 🔍 🌔 🕨 Debug Win32 🗊 🗞 📐 🔺 🗏 📮 📮 ソリューション エクスプローラー 🔷 👎 🗙 test.cpp\* × #include "def.h" 👒 main(int (グローバル スコープ) 🔜 ソリューション 'aa' (1 プロジェクト) ⊟<mark>‡include</mark> ″def.h″ [**‡include** ″Mlib.h″ 🖌 📑 aa #include "mlib.h" void main (int Number)↓ Printf( "Hello World!"); 🔺 🦢 ソース ファイル test.cpp void main(int Number){ Printf("Hello World!"); 外部依存関係 新規ファイル 追加した既存 100 % - 4 ヘッダーファイル 出力 <del>-</del> 4 х ③コンパイルエラーがあ -出力元の表示(S): ればここにメッセージが 出る。 準備完了 エラーがでなくなるまで プログラムを修正する

# サンプルプログラムの実行結果

### プッシュボタンを押すと、メモウインドウに「Hello World!」と表示される



def.h はウインドウやボタンのサイズ、配置の情報を管理 mlib.h は様々なグラフ化の関数や、ウインドウとOSのメッセージのやり取りが仕事

プログラム内では mlib.h より先に def.h のインクルード文を書くこと

プッシュボタンが押されたとき、main関数を実行する。

main関数はプログラム内に必ず必要

main関数はmlibから、整数型引数Numberを受け取る Numberには押されたプッシュボタンの番号(0,1,2....)が入っている

## def.h のウインドウ初期設定部

#### def.h で定めた定数はすべて大文字



各種ボタン等のサイズ、位置はこの下の方のcomponet()関数で設定

プッシュボタン



- ボタンを押すことにより、main()関数で指定したプログラムが実行 される。
- ・押されたボタン番号が仮引数Numberに入っているため、main()
   関数内でswitch文やif文等より、ボタンによる場合分けが可。
- ・ボタンの位置、サイズ、キャプションはdef.hのcomponent()関数 内にあるグローバル変数sPB[ボタン番号]構造体で設定する。

i番目のボタン 番号の設定	sPB[i].x sPB[i].y sPB[i].w sPB[i].h sPB[i].name	ボタン左上の×座標 ボタン左上のy座標 ボタンの幅 ボタンの高さ ボタンのキャプション(TCHAR型)
------------------	---	---

メモウインドウ



- ・変数やコメント等をプログラム中から表示 させることが可
- メモウインドウ内はスクロールさせること
   ができるため、多数の表示が必要なときに
   使用
- 1行の文字数や、記憶できる行数、表示行数はdef.hの定数
   BUFSIZE\_W、BUFSIZE\_H、DISPSIZE\_Hでそれぞれ指定できる
- 表示には関数Printf("表示文字列",変数等...)を使用、printf
   関数と同じ使い方
- ・メモウインドウは一つだけ配置・位置、サイズ、キャプションは def.hのsME構造体で設定

sME.x	メモウインドウ左上のx座標	
sME.y	メモウインドウ左上のy座標	
sME.w	メモウインドウの幅	
sME.h	メモウインドウの高さ(MEMOWIN_Hで決まる)	



### 2-1.メモウインドウに押したプッシュボタンの番号を表示するプロ グラムを作れ。尚、Printf関数はprintf関数と使い方は同じ

Mlib for C++ Langua	ge (ver 3.7)		
Push Button0	Push Button1	Push Button2	押したブッシュボタンは0番です。 押したブッシュボタンは1番です。
			押したブッシュボタンは2番です。  押したブッシュボタンは0番です
0.5	Check BoxU	O Radio Buttonu	押したブッシュボタンは1番です。
0.6			押したブッシュボタンは2番です。 

#### 2-2.メモウインドウに押したプッシュボタンの番号を英語で表示する プログラムを作れ.if文、switch文を使う

Mlib for C++ Langua	ge (ver 3.7)	20.0	La-A 🔾 Affert 🗋 🕬 📑 Longer
Push ButtonO	Push Button1	Push Button2	The button number was zero The first button was pushed You pushed the Push Button?
0.5	□ Check Box0	⊙ Radio Button0	The button number was zero The first button was pushed You pushed the Push Button2

ヒント:main 関数内では、プッシュボタンの番号は引数Numberを参照

チェックボックス



- ・状態のON、OFFを変更できるボックスであり、
   クリックごとにON,OFFを切り替える。
- def.h内の定数CK\_NUMで指定した個数の
   チェックボックスを配置可
- ・複数個指定した場合、0から昇順に番号を自動割り振り
- ・位置、サイズ、キャプションはdef.hのcomponent()関数内にある るグローバル変数sCK[チェックボックス番号]構造体を設定

i番目のチェック ボックスの設定sCK [i].xチ sCK [i].yチ sCK [i].wsCK [i].wチ sCK [i].hチ sCK [i].name	Fェックボックス左上の×座標 Fェックボックス左上のy座標 Fェックボックスの幅 Fェックボックスの高さ Fェックボックスのキャプション(TCHAR型)
--	--

•ON,OFFの状態は変数sCK[チェックボックス番号].chkに反映され、
 ONのとき1、OFFのとき0

•main 関数内でいつでも参照可

演習2-3

#### 2-3. チェックボックスを二つ作成し、ボタンを押すと二つのチェッ クボックスのON, OFF状態をメモウインドウに表示するプ ログラムを作れ

Push Button0	Push Button1	Push Button2	チェックボックスDはオン チェックボックス1はオフ
0.5	🔽 Check Box0	⊂ Radio Button0	
0.6	Check Box1		

Push Button0	Push Button1	Push Button2	チェックボックスDはオン チェックボックスDはオフ	チェックボックス1はオフ チェックボックス1はオン
0.5	🗆 Check Box0	○ Radio Button0		
0.6	☑ Check Box1			







- グループ化された複数の候補の中からグループ
   内で一つを選択するボタンであり、クリックごとに
   ON,OFFを切り替え
- def.h内の定数RD\_NUMで指定した数のラジオ ボタンを配置可

・位置、サイズ、キャプションはsRD[ラジオボタン番号]構造体を設定

i番目のラジオ ボタンの設定	sRD[i].x	ラジオボタン左上の×座標
	sRD[i].y	ラジオボタン左上のy座標
	sRD[i].w	ラジオボタンの幅
	sRD[i].h	ラジオボタンの高さ
	sRD[i].name	ラジオボタンのキャプション(TCHAR型)

ラジオボタンのグループ化はdef.h内のsRD[].chkの初期値で設定
 例) ラジオボタン数:6個、グループ:2種類(0,1番と2,3,4,5番)
 sRD[0].chk=1; sRD[2].chk=1; それ以外は0

・選択の状態は変数sRD[ラジオボタン番号].chkに反映され、
 選択→1、それ以外→0、メイン関数内で参照可

演習2-4

### 2-4.6つのラジオボタンのうち、0~2、3~5の2グループを作成 し、ボタンを押すと、各ラジオボタンの状態をメモウインドウに 表示するプログラムを作れ。



#### 最初は何も選択されない





5をクリックしたとき、 3, 4, 5が同一グループなので、4が OFFになり、5がオンになる 0, 1, 2、は別グループなので0が0 Nのまま

## 演習 2-2

```
void main (int Number){
   switch (Number){
   case 0:
      Printf("The button number was zero¥n");
      break;
   case 1:
      Printf("The first button was pushed ¥n");
      break;
   case 2:
      Printf("You pushed the Push Button2 ¥n");
      break;
   }
```