

電子情報理工学特論I

電気電子コース 弓仲、三輪

— 実習形式の講義による実践的ソフト&ハード講座 —

講義形態:実習

場 所 :高度人材育成センター(HRCC) 2F PC室

講義目的:WindowsベースのCプログラミングによる数値計算のグラフ化、アニメーション技法、ハードウェア記述言語HDLの習得

講義内容:(前半)Microsoft Visual C++ 2010 Express、フリーのグラフ作成用C言語ライブラリmlibの使い方、波形描画、信号処理、アニメーション等の演習を行う。7回目にプログラム課題を課す。

(後半)デジタル回路を設計するためのコンピュータ言語HDLについて学び、回路の設計、構成を記述してハードウェアを作成できる知識を習得する。

評価法:出席重視、中間課題(三輪分)、レポート(弓仲分)で成績を評価

講義実施場所

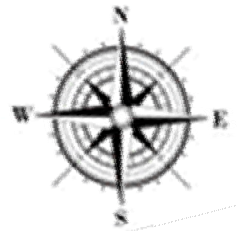


Kinki Campus
桐生キャンパス

高度人材育成センター
2F PC室

電気電子棟

6号館



Windows で計算するには？

電卓ソフトを使う

Windows で数値データをグラフ化するには？

EXCELを使う

Windows で数万行の数値データファイルの数値計算をするには？

Cプログラミングならできるかも。。でもどうやって？

Windows で絵をかけますか？

標準のペイントソフト、パワーポイントを使う

Windows で放物線運動のアニメーションするにはどうするか？

専用のフリーソフトでも探す。

Windows でグラフをリアルタイムでアニメーションするには？

。。。。。。

**C言語プログラミングは知っていても、今までの知識では
Windows でその結果をリアルタイム表示するのは困難**

この講義(三輪分)を履修することにより

- Windowsの**フリーソフト**のみで数値計算結果のグラフ化ができる
- C言語による数値シミュレーション結果のアニメーション処理
- WindowsベースのC言語GUIアプリケーション開発のとっかかり

この講義を履修する条件

- **C言語プログラミング経験者**
(C言語の文法は教えません。自分でいくらでも調べられます)
- C言語でシミュレーション、数値計算等を行い、結果を気軽に見たい
- PCを用いてリアルタイムなデータ取得、グラフ化をしたい

三輪研ホームページ→講義関連ページ→電子情報理工学特論I→専用HPへのリンク

The screenshot shows the homepage of the Miwalab research group. The navigation menu at the top includes '講義関連ページ' (Lecture-related pages). A red arrow points to the '電子情報理工学特論I' link within this menu. The main content area features a large article about wave propagation, and a sidebar on the left contains a 'メインメニュー' (Main menu) with various navigation options.

The screenshot shows the lecture page for '2019 電子情報理工学特論I'. The page includes the following information:

- 担当教員:** 弓仲原史・三輪空司
- 更新日:** 2019年
- 講義内容:** 座学&実習。場所は高度人材育成センター 2F コンピュータ室 (PC 24台)。講義内容は電子情報通信技術の理工学的基礎となるプログラミング技術について講述する。前半はWindowsのVisual Cをベースとした数値計算、グラフ化のプログラミングを習得する。後半でハードウェア記述言語であるHDLを用いたハードウェア設計法を習得する。評価法は出席重視、中間課題(三輪分)、レポート(弓仲分)で成績を評価。
- 前半の講義のお知らせ:**
 - 2回目以降はHRC棟(リンク中の3番の建物)の2Fシミュレーション室で実施します。
 - 1回目は総研棟304で実施します。
- 講義内容表:**

実施日	内容 (pdf資料)	資料
第1回 (10/1)	ガイダンス、Windowsプログラミングの基礎	サンプル A B C
第2回 (10/8)	mlibの使い方、ウインドウ初期設定関連	演習2-2,3,4回答例
第3回 (10/15)	mlibでの文字列の扱い	演習3-1,2回答例
第4回 (10/29)	mlibの使い方、Plot関連	例題4ソース 演習4回答例
第5回 (11/5)	物体の運動シミュレーション	例5-1、例5-2、演習5-3、演習5-3回答例
第6回 (11/12)	物体の衝突シミュレーション	例6-1、例6-2、演習6-3 演習6-3回答例
第7回 (11/19)	中間課題実施	
第8回 (11/26)	中間課題実施	
第9回以降 (12/3~)	弓仲先生担当分	

At the bottom of the page, there is a link: [三輪研HP講義ページに移動](#)



<http://miwalab.ei.st.gunma-u.ac.jp/master/mlib.html>

三輪担当分の講義の流れ (予定)

1. ガイダンス
2. mlibの使い方
3. mlibでの文字列の扱い
4. mlibの使い方、グラフ描画
5. 物体の運動シミュレーション
6. 物体の衝突シミュレーション
7. 中間課題実施
8. 中間課題実施

<https://www.vector.co.jp/soft/winnt/prog/se408868.html>

トップ > ダウンロード > Windows > プログラミング > C言語

ソフト詳細 レビュー コメント・評価 スクリーンショット ダウンロード

mlib

数値計算の確認やリアルタイムグラフ描画のための、C++の知識のいらないGUIベースCライブラリ

ユーザーの評価(0人): ★★★★★
コメント: 0件
コメントを見る

■関連キーワード
ライブラリ, リアルタイム, 描画, 数値, C++, 知識, グラフ, GUI

ソフト詳細説明

工学系の大学生の皆さん等13年までにC言語の基礎的な演習や、数値演算の講義等も受けてきたはずですが、おそらくUNIX系のOSでテキストベースの実行環境で、だったと思います。ところがいざ研究室に配属されるといきなり高度な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でまもなくWindowsベースで...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統合開発環境のMicrosoft Visual C++とかC++ Builderを使うことにならなければならぬ...
▼ 続きを見る ▼

「C言語」カテゴリーの人気ランキング

- 1位 [Visual Windows for BC++](#) - BC++5.5のウィンドウズ統合環境 ★★★★★
- 2位 [学習用C言語開発環境 ベータ版](#) - これまでで最もシンプルなC言語開発環境 ★★★★★
- 3位 [C言語を始めよう! Ver2](#) - C言語ミニ開発環境 ★★★★★
- 4位 [EWCLIB](#) - VC++で簡単にウェブカメラ画像入力プログラムを作成できる(ヘッダファイル形式) ★★★★★
- 5位 [mlib](#) - 数値計算の確認やリアルタイムグラフ描画のための、C++の知識のいらないGUIベースCライブラリ ★★★★★

PCでのプログラム開発

LABVIEW

グラフィカルプログラミング環境、独自のプログラミング概念
フローチャート方式、解析、データの可視化、かなり高価、
計測が得意

MATLAB

グラフィカルプログラミング環境、計測用のGUIを作成可能
独自のプログラミング言語、非常に多くの関数を持つ
解析、データ可視化が得意、高価、インタープリタ、ドライバの
対応は中

C++ builder
XE、Delphi

GUIプログラミング開発環境、様々なボタンを配置し、そのボタン
に対するアクションを記述する、高価、ドライバの対応は良
信号処理等のプログラムは自作する

Visual Basic

GUIプログラミング開発環境、様々なボタンを配置し、そのボタン
に対するアクションを記述する(イベントドリブン)、フリー、
計測器のドライバが対応していない場合がある。

Visual-C

C++プログラミング統合開発環境、GUIは簡単には作れない。
フリー、ドライバの対応は良、信号処理等のプログラムは自作する

Microsoft Visual C++ によるプログラミング

C++言語

オブジェクト指向を行う為に、C言語に追加された仕様。C++言語はプログラミング基礎、応用でやったC言語とはベツモノ。

作成可能なアプリケーション

コンソールアプリケーション

プログラミング応用(Linux)でやったような端末画面でのテキストベースのプログラミング

DLL(Dynamic Link library)

例えばC言語の関数のように、入力を指定して、なんらかの処理後の出力を返すような関数のライブラリで、Windowsで動くアプリケーションから利用できる

GUI(Graphical User Interface)プログラミング

ウインドウ内に様々なボタンが配置された、ボタンに対応した処理を行うプログラミング

Visual C++でGUIを使ったプログラムを作るには

MFCアプリ
ケーション

WindowsのGUIを司る便利な関数をC++のクラス概念を使って、オブジェクト指向プログラミングを容易に開発できるようにしたもの。**概念の理解に難**

Win32アプリ
ケーション

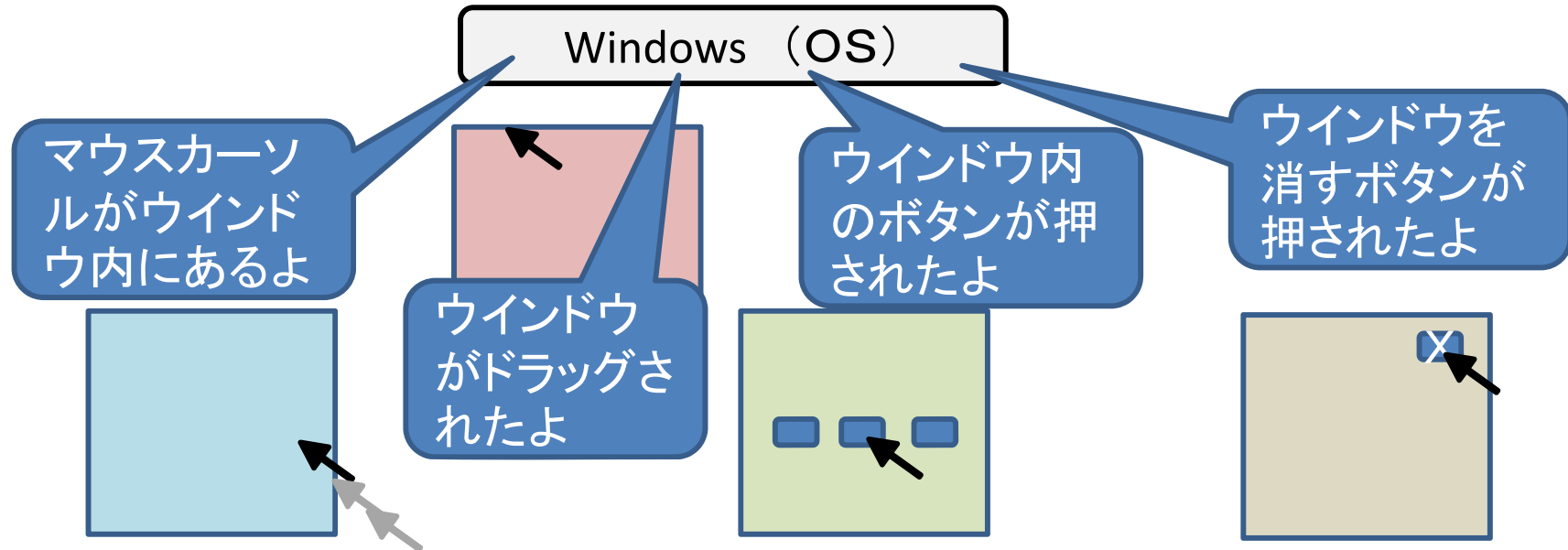
WindowsのGUIを司る便利な関数群Win32APIを使ったC言語プログラミング古くから使われている。

ボタンをクリックしたときの動作等、Windowsが制御しているあらゆるウインドウ処理にうまく割り込めるようにプログラムしないといけない。非常に面倒

Windowsプログラミング

Windowsでは、各ウインドウはOSとは独立に動作

OSはマウスの様々な応答(イベント)発生に対するメッセージをウインドウに教えてやる作業を、繰り返しているだけ



各ウインドウではメッセージを受け取って、様々なイベントに対応した処理を呼び出して実行する

我々がwindowsのGUIを使って新たに何かの仕事をさせるときは、

1. ウインドウを新たに作らないといけない
2. そのウインドウで起こったイベントを受け取らないといけない。
3. そのイベントに対応する処理内容を自らプログラミングしないといけない

Win32API(Application Programming Interfaces)を利用

Win32アプリケーションによるウィンドウの表示

```
#include <windows.h>
#include <tchar.h>

// プロトタイプ宣言
LRESULT CALLBACK WndProc(HWND hWnd, UINT msg, WPARAM wp, LPARAM lp);
```

```
// 開始位置
int WINAPI WinMain(HINSTANCE hInst, HINSTANCE hPrevInst,
                  LPSTR pCmdLine, int showCmd)
```

```
{
    WNDCLASSEX wc;
    HWND hWnd;
    MSG msg;

    // ウィンドウクラスの情報を設定
    wc.cbSize = sizeof(wc);           // 構造体サイズ
    wc.style = CS_HREDRAW | CS_VREDRAW; // スタイル
    wc.lpfnWndProc = WndProc;         // ウィンドウプロシージャ
    wc.cbClsExtra = 0;                // 拡張情報1
    wc.cbWndExtra = 0;                // 拡張情報2
    wc.hInstance = hInst;             // インスタンスハンドル
    wc.hIcon = (HICON)LoadImage(      // アイコン
        NULL, MAKEINTRESOURCE(IDI_APPLICATION), IMAGE_ICON,
        0, 0, LR_DEFAULTSIZE | LR_SHARED);
    wc.hIconSm = wc.hIcon;            // 子アイコン
    wc.hCursor = (HCURSOR)LoadImage( // マウスカーソル
        NULL, MAKEINTRESOURCE(IDC_ARROW), IMAGE_CURSOR,
        0, 0, LR_DEFAULTSIZE | LR_SHARED);
    wc.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH); // ウィンドウ背景
    wc.lpszMenuName = NULL;           // メニュー名
    wc.lpszClassName = _T("Default Class Name"); // ウィンドウクラス名

    // ウィンドウクラスを登録する
    if (RegisterClassEx( &wc ) == 0) { return 1; }
```

ウィンドウクラスの設定

```
// ウィンドウを作成し、そのハンドル番号を取得
```

```
hWnd = CreateWindow(
    wc.lpszClassName, // ウィンドウクラス名
    _T("Sample Program"), // タイトルバーに表示する文字列
    WS_OVERLAPPEDWINDOW, // ウィンドウの種類
    CW_USEDEFAULT, // ウィンドウを表示する位置(X座標)
    CW_USEDEFAULT, // ウィンドウを表示する位置(Y座標)
    CW_USEDEFAULT, // ウィンドウの幅
```

```
    CW_USEDEFAULT, // ウィンドウの高さ
    NULL, // 親ウィンドウのウィンドウハンドル
    NULL, // メニューハンドル
    hInst, // インスタンスハンドル
    NULL // その他の作成デー
```

```
);
```

```
if ( hWnd == NULL ) { return 1; }
```

```
// ウィンドウを表示する
ShowWindow( hWnd, SW_SHOW );
UpdateWindow( hWnd );
```

```
// メッセージループ
```

```
while( 1 )
```

```
{
```

```
    BOOL ret = GetMessage( &msg, NULL, 0, 0 ); // メッセージを取得する
    if( ret == 0 || ret == -1 )
```

```
    { // アプリケーションを終了させるメッセージが来ていたら、
      // あるいは GetMessage() が失敗したら(-1 が返されたら)、ループを抜ける
      break;
    }
```

```
    else
```

```
    { // メッセージを処理する
```

```
      TranslateMessage( &msg );
```

```
      DispatchMessage( &msg );
```

```
    }
```

```
return 0;
```

メッセージ取得 ウィンドウプロシージャにメッセージを送る

```
// ウィンドウプロシージャ
```

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT msg, WPARAM wp, LPARAM lp)
```

```
{
```

```
    switch( msg )
```

```
    {
```

```
        case WM_DESTROY: // ウィンドウを破棄するとき
```

```
            PostQuitMessage( 0 );
```

```
            return 0;
```

```
    }
```

```
// 他のメッセージは、デフォルトの処理を行う
```

```
return DefWindowProc( hWnd, msg, wp, lp );
```

```
}
```

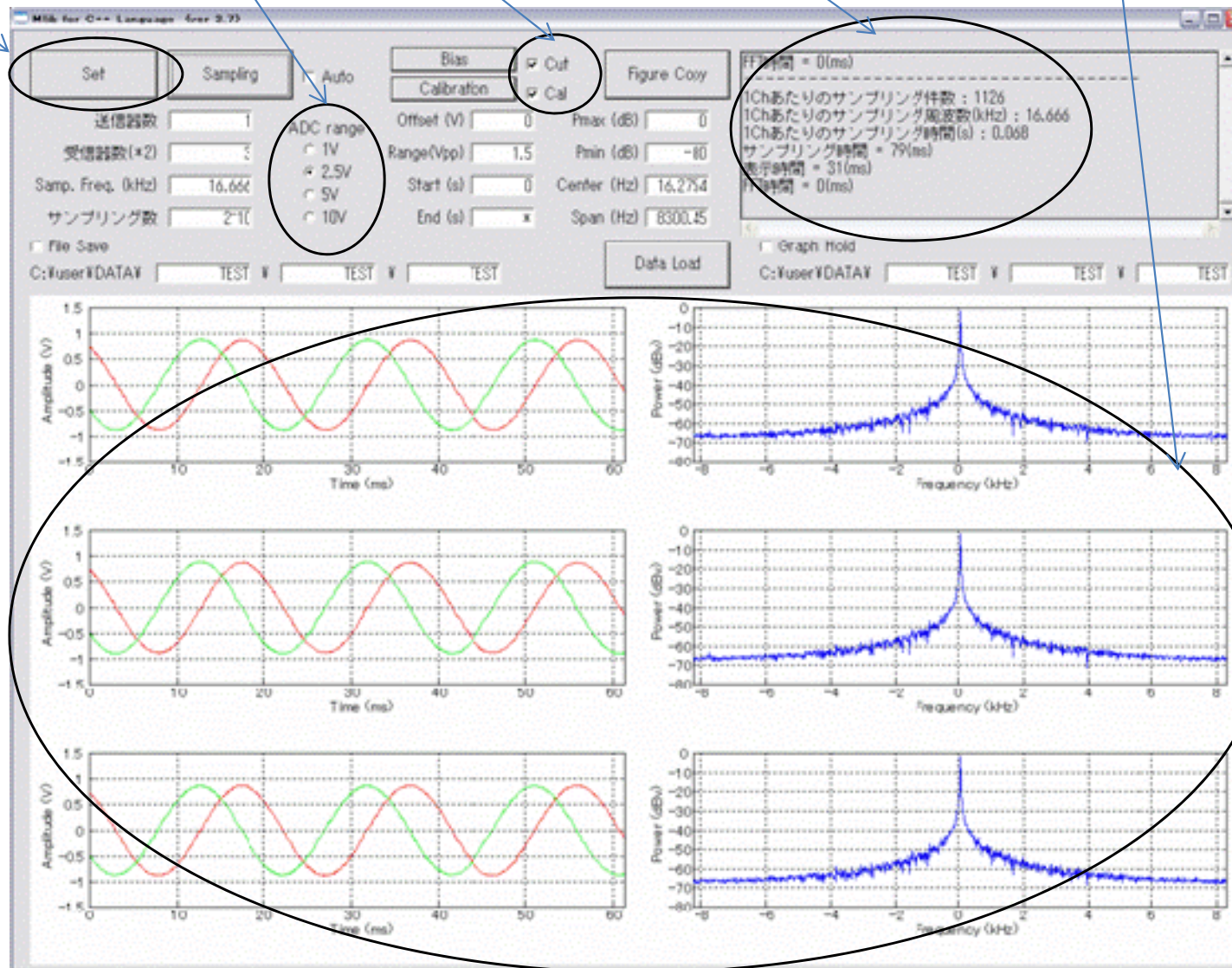
http://www.geocities.jp/ky_webid/win32c/002.html

mllib 3.7 の利用が便利

mLibによるプログラム作成

mLibは、1. **ウインドウ作成**, 2. **イベントのお知らせ処理**をある程度やってくれる。
加えて**グラフ化の便利な関数**を備えている。我々は処理内容のみを書けばよい

ボタン、ラジオボタン、チェックボックス、テキスト入出力、1次元グラフ、2次元グラフ



Win32アプリケーションの作成

この講義で使用する Visual C のバージョン

Microsoft 提供 無償版

Visual C++ 2010 Express 以下のリンクから取得可能

<https://support.microsoft.com/ja-jp/help/2977003/the-latest-supported-visual-c-downloads>

Visual Studio 2015、2017 も上記リンクから取得可能

学生が利用可能な全学ソフトウェアライセンスでもvisual studioを取得可能 <http://sw.media.gunma-u.ac.jp/ms.html>

• Visual Studio 2019 Community エディション (最新版)

<https://visualstudio.microsoft.com/ja/thank-you-downloading-visual-studio/?sku=Community&rel=16>