**● Windows10用**

**プログラムのダウンロード**

1. DreamCampus の“レポート”の“プログラミング課題”から，setup.zip とprg.zip をダウンロードする．
2. 同ファイルを解凍する（右クリックし，“プログラムから開く”→“エクスプローラー”を選択）．
3. “robotics”というフォルダを作り，解凍したファイルを入れる．
4. “PC”の“ローカルディスク(C:)”を開いて，そこにroboticsフォルダをコピー（ドラッグ&ドロップ）する．

**Borland C++ Compiler 5.5の設定**

* **Borland C++ Compilerのインストール**

roboticsフォルダ内の setup \ freecommandlinetools2.exe を実行（アイコンをダブルクリック）すると，デフォルトで C: \ borland \ bcc55 にBorland C++ Compiler 5.5 がインストールされる（Cドライブの直下に，borland というフォルダができていることを確認）．

* **PATHの設定**

“PC”を右クリックし，“プロパティ”をクリックする．出てきたウィンド右側の“システムの詳細設定”をクリックすると，“システムのプロパティ”が開く．“詳細設定”のタブの中の“環境変数”をクリックすると，“環境変数”のウィンドが開く．上部にある“ユーザー環境変数”から，“Path”をダブルクリックで選択し，“新規”をクリックして，次の行を追加する（設定はコマンドプロンプトを起動した時点で有効になるので，エラーが出た場合は，再設定後，コマンドプロンプトを立ち上げ直すこと）．

C:\borland\bcc55\bin

* **bcc32.cfg ，ilink32.cfgの転送**

robotics \ setup \ bcc-setup.bat をダブルクリックすることにより，bcc32.cfgとilink32.cfgのファイルがC:\ borland \ bcc55 \ Binへ転送される（ファイルの中身を見るには，ファイルを右クリックし，”編集(E)”を選択）．

* **確認**

robotics \ prg \ test \ helloの中にある，サンプルプログラムhello.cのコンパイルと実行を行う．“スタート”ボタンをクリックし，　　“Windows システムツール”(または“すべてのプログラム”の“アクセサリ”内）にある“コマンドプロンプト”を開いて，以下のコマンドを実行．

**C:\ > cd□c: \ robotics \ prg \ test \ hello　または cd□の後，　アドレスをコピー＆ペースト（補足1～3参照）．**

**C:\ robotics \ prg \ test \ hello > bcc32□hello.c**

**または make （補足2 参照)．**

**C:\ robotics \ prg \ test \ hello > hello**

**または make run（補足2参照）．**

Hello, world !　と表示されれば，設定は完了．なお，コマンドプロンプトのショートカットをデスクトップに置いておくと便利（アイコンをマウスの右クリックして，“送る/デスクトップ(ショートカットを作成)”）.

**< 補足１：　MS-DOSコマンド >**

1. cd□フォルダ名 でフォルダ間を移動し，cd□.. で一つ上のフォルダへ移動する．したがって，仮にc:\ robotics \ prgフォルダにいる場合に，cd□.. と入力し，roboticsフォルダへ移動後，cd□prg と入力すると，再びprgフォルダに戻ってくる．
2. dドライブ内へ移動する場合は，まず，cd□d: を入力．
3. フォルダウインドのアドレスバーに表示されているアドレスを，マウスの右クリックでコマンドプロンプトへコピー&ペーストすることにより，フォルダ名を入力する手間が省ける（ウィンドのフォルダをコマンドプロンプトへドロップも可）.

アドレスバーが表示されていない場合は，フォルダの設定で，“表示/ツールバー/アドレスバー”にチェックを入れることにより，アドレスバーが現れる．

アドレスバー動かない場合は，“表示/ツールバー/ツールバーを固定する” のチェックを外して，アドレスバーを下に移動する．

アドレスバーにパスが表示されない場合は，“ツール/フォルダオプション/表示/アドレスバーにファイルのパス名を表示する”にチェックを入れる．

1. dir コマンドにより，フォルダ内のファイル名を参照できる．

**< 補足 ２：　makefile の使い方 >**

同じフォルダ中のmakefileを使うとプログラムのコンパイル，実行等，何かと便利になる．使い方は以下に示す通り．

1. make でコンパイルされる．
2. make□runで，プログラムが実行される．
3. make□clean で，不要なファイルが消去される．
4. エラー，ワーニングを見たいときは，make > filename.txt を実行することで，内容を確認できる．
5. makefileは，“メモ帳” で確認・編集可能（メモ帳のアイコンをデスクトップのアイコンにしておき，ドラッグ&ドロップすると便利）．
6. 異なるファイルをコンパイルするときは，makefile中の，TARGET = \*\*\* のところを変更．

**< 補足 ３：　C言語プログラミング用エディタ >**

・　フリーソフト TeraPad では，カラー表示で，TAB幅（なるべく“4”に設定）やフォントの大きさを設定できる (ソースファイルは，c:\robotics\Toolsの中)．

・　ソフト内で，コンパイルや実行を行えるようになっているが，それを使うよりも，makefileを使う方が便利．

**GLUT(OpenGL)の設定**

* **glut32.libの生成**

コマンドプロンプトから以下のコマンドを実行することにより，glut32.dllを使ってglut32.libを生成する．

**C:\> cd□c: \ robotics \ setup \ glut-3.7.6-bin**

**C:\ robotics \ setup \ glut-3.7.6-bin > implib□glut32.lib□glut32.dll**

* **glut.h , glut32.lib , glut32.dll の転送**

glut-setup.bat ( c:\robotics / setup内) をダブルクリックする（ファイルを見るには，ファイルを右クリックし，”編集(E)”を選択）．次に，glut-3.7.6-binフォルダ内のglut32.dllを C:\Windows \System にコピーする（ワーニングは無視）．

* **確認**

Ｃ: \ robotics \ prg \ test \ winの中にある，サンプルプログラムwin.cのコンパイルと実行を行う．コマンドプロンプトを開いて，以下のコマンドを実行

**C:\ > cd□c: \ robotics \ prg \ test \ win**

**C:\ robotics \ prg \ test \ win> make**

**C:\ robotics \ prg \ test \ win> make□run**

ウィンドが生成されることを確認する（ピンクのウィンドが現れるはず．その右上の×を左クリックすることで，プログラムは終了）．

注）「glut32.lib が見つかりません」というようなエラーが出る場合は，glut-3.7.6-binフォルダ内のglut32.lib (またはglu32)を作業フォルダ(\winなど)にコピーしてコンパイルする．

**サンプルプログラムの実行**

* **ファイルの内容**

裏面あり

フォルダ内には，以下のような多数のファイルがある．

invK3d.c（全体の処理）

invK3d\_1.c（ライブラリ）

glshapes.c（描画関数）

glwindow.c（OpenGLライブラリ）

詳細は，c:\robotics \ docフォルダ内にある，“プログラムの説明.ppt”を参照（同じフォルダ内に配布プリントがある）．

* **コンパイル方法**

複数のソースファイルからコンパイルを行って実行ファイルを作成するには，makefile を使うと便利．

**C:\ > cd□c: \ robotics \ prg \ sample**

**C:\ robotics \ prg \ sample> make**

**C:\ robotics \ prg \ sample> make□run**

なお，makeコマンドを使用しない場合は，以下のように入力．

　　> bcc32□invK3d.c□invK3d\_l.c□glshapes.c□glwindow.c

実行ファイルは先頭のファイル名で作成される．つまり，

　　> invK3d

プログラミング課題がどうしてもできない人は．．．

と入力すればプログラムが実行される（コンパイル後は，実行ファイル“invk3d.exe”を左ダブルクリックしても実行される）．

* **操作方法**

HOME　 :　軌道追従開始・一時停止

ESC :　プログラムの終了

矢印 :　視点の回転

e, s, d, D, f, c :　視点の平行移動

* **描画速度**

invK3d.cで定義されているMY\_GL\_CUTの値を変えれば描画速度を調節することができる．値が大きい程高速．（値は0以上）．

**< 補足 ４：　プログラミング作成のコツ >**

・　テストフォルダを新たに作って，追加，削除，変更等を加え，試す．

・　とりあえずの成功版は，日付をつけて保存しておく．

・ 成功版の説明をHereis.txt という名前のファイルに書く．

・　プログラムの段落は，スペースで開けるよりも，TABキーを使ってあけるほうが，間違いが少なく，美しく，故にバグが少なくなる．

**プログラム課題の提出**

● 提出日と提出方法

課題は，15週目の授業日（例年，1月末 or 2月はじめ） までに，DreamCampusの“レポート”の“プログラミング課題”に提出する．

● 実行ファイル（\*.exe）を含むソースプログラムの入ったフォルダをzipで圧縮（フォルダを右クリックし，“送る”→“圧縮(zip形式)フォルダー”を選択）して提出．フォルダ名は，“学籍番号（氏名）”にすること（学籍番号は半角英数字）．なお，提出は作業フォルダのみ（インストール用フォルダ等は不要）

● フォルダ内に“工夫点.txt” を作成し，そこに，この作品の工夫点，およびオリジナル点等を書く．オリジナル点の例としては，“関節を増やす”，“関節の長さ，太さ，色などを変化させる”，“ロボットアームとは別のものを創る”などなど．

● 提出後，プログラムに関する簡単な質問を行うこともある．

● 採点時には，オリジナル点の難易度と新規性を高く評価する．極力，人と違ったチャームポイントを作品に持たすようにすること．なお，同じソースコードの場合は，双方減点．

1 モデル及び座標系設定

本演習では，**図１**に示すような３自由度ロボットマニピュレータモデルを使用し，座標系を設定する．

**3次元3自由度ロボットマニピュレータの軌道追従制御**

2 運動学解析

• 先端位置



• 各関節の位置



• 先端の速度



ただし，，， 

注1: , , ,

注2: ヤコビ行列が正方行列でない場合... invK3d\_l.c で，逆行列：MATinvgauss() の代わりに，擬似逆行列：MATpse\_invgauss() を使う．

3 軌道制御

時間でロボットマニピュレータの先端を0から(=1にセット)に，下記の運動を用いて移動させることにする（などの ’’はgoal，’’はstartを意味するものとする）．



ただし，



直線軌道の場合，先端の運動指令値：



**図１**３自由度ロボットマニピュレータモデル







次に，**図２**に示すような分解速度制御(Resolved Velocity Control)で軌道追従制御を行う．

**図２**分解速度制御系のブロック線図



*•* 先端指令速度(位置誤差のフィードバックを含む)



ただしは実際のロボットの手先位置（実際のロボットの関節角から順運動学の式を使って求めたもの）．

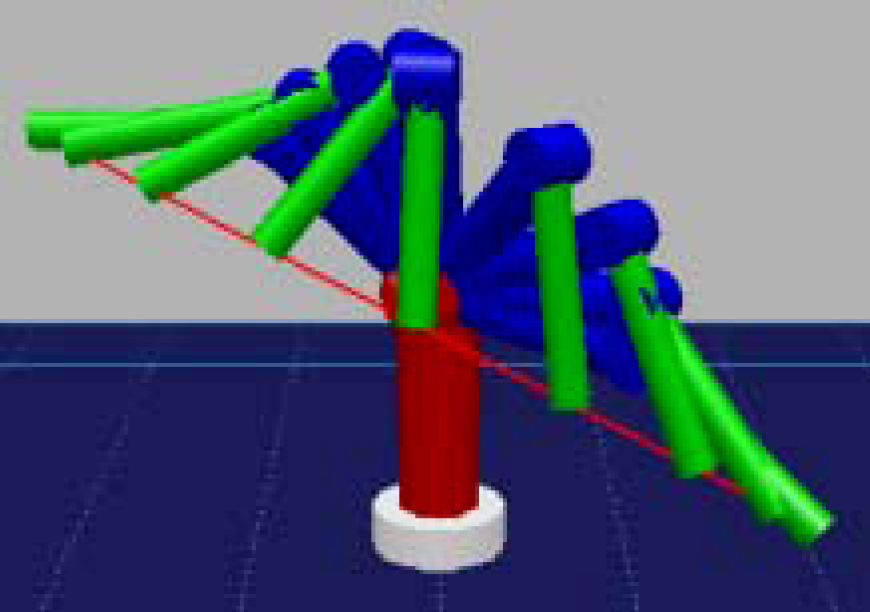
*•* 関節角速度と角度



ここで，は1サンプル前のの値であり，はサンプリング時間とする．

4 計算機シミュレータ

**図３**計算機シミュレータの表示(Borland C++とGLUT)



裏面あり

* プログラミング課題 がどうしてもできない人のための課題

課題：

今後，どのようなロボットを開発したいか？ 目的，機構，センサ，制御方法等を示しながら，図を用いて説明せよ．

提出日：ロボット工学の試験日

サイズ：A4（左上ホッチキス，表紙不要）

形式：手書き

その他：プログラミング課題の方が最高点は高いです．また，両方提出した場合，どちらも評価しますが，点数は単純な足し算よりも低くなります．