

YP-Spur を MATLAB で使うには？

YP-Spur は、Linux ベースで開発されているようであるが、Windows でも使えるようなので、ここでは、Windows 版の MATLAB から使う方法を紹介する。YP-Spur は、Linux ベースなので、MinGW+msys 環境をセットアップする。

http://www.mingw.org/wiki/Getting_Started

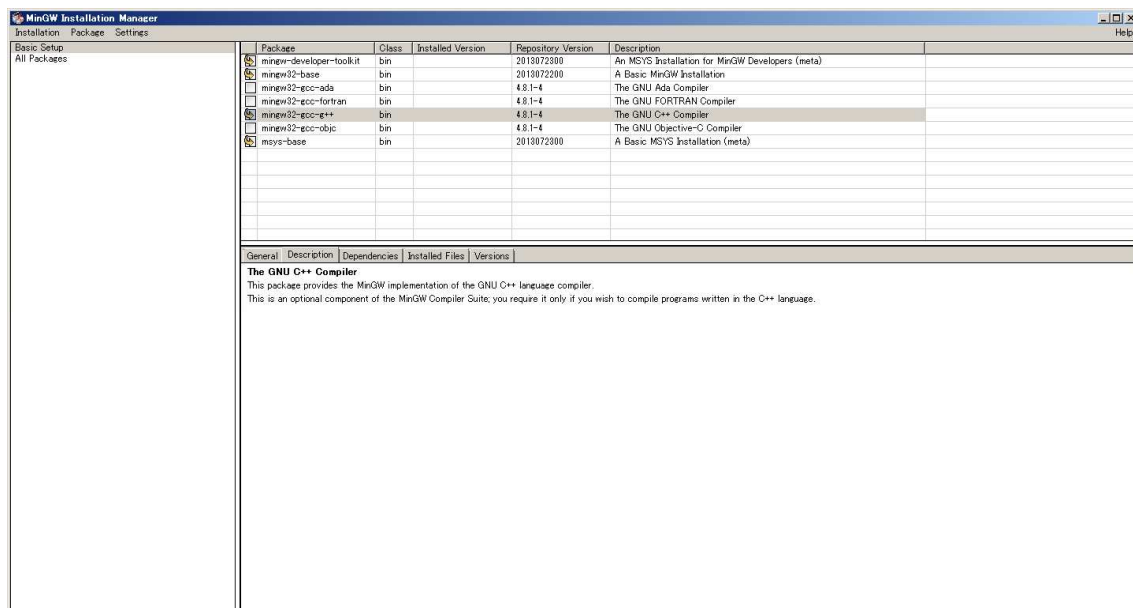
から、

`mingw-get-setup.exe`

をダウンロードし、

`c:¥MinGW`

にインストールする。



ここでは、

`mingw-developer-toolkit`

`mingw32-base`

`mingw32-gcc-g++`

`msys-base`

をインストールする。

MinGW Library の

`mingw32-pthreads-w32`

も必要である。

インストールが終わったら、環境変数の PATH に

`C:¥MinGW¥bin;C:¥MinGW¥msys¥1.0¥bin`

を追加する。

YP-Spur のインストール

yp-spur-1.13.5 をダウンロード。

コマンドウィンドウを起動し、

```
c:¥MinGW¥msys¥1.0¥msys.bat
```

で実行し、コマンド入力画面を出す。

次に、

```
c:¥MinGW¥msys¥1.0¥home¥ikko
```

の中にコピーしておく。

```
cd yp-spur
./configure
```

を実行する。

コンパイルエラーが出る場合には、

オリジナル	修正
<code>#include <pthread.h></code>	<code>#define timespec linux_timespec</code> <code>#include <pthread.h></code> <code>#undef timespec</code>

を、以下のファイル全てに適用する。

yp-spur-1.14.0 の場合

```
param.h
```

yp-spur-1.13.5 の場合

```
utility.c, ypspur-coordinator.c, control_vehicle.c, control_motion.c,
param.c, commandad.c, param.h, command.c, command_param.c,
command_set.c, command_get.c, command_run.c, communication.c
odometry.c, ssm_spur_handler.c
```

また、fork,setsid()は、MinGW ではサポートしていないためコメントアウトする。

```
C:¥MinGW¥msys¥1.0¥msys.bat
```

を実行して、コマンドプロンプトを立ち上げる。

```
cd home/ikko
```

ikko の部分は、ログイン名になる。

として実体となるフォルダの

```
C:¥MinGW¥msys¥1.0¥home¥ikko
```

に、yp-spur をコピーする。

```
cd yp-spur
./configure
```

```
make
```

としてバイナリを作成する。

MEX コンパイラの導入

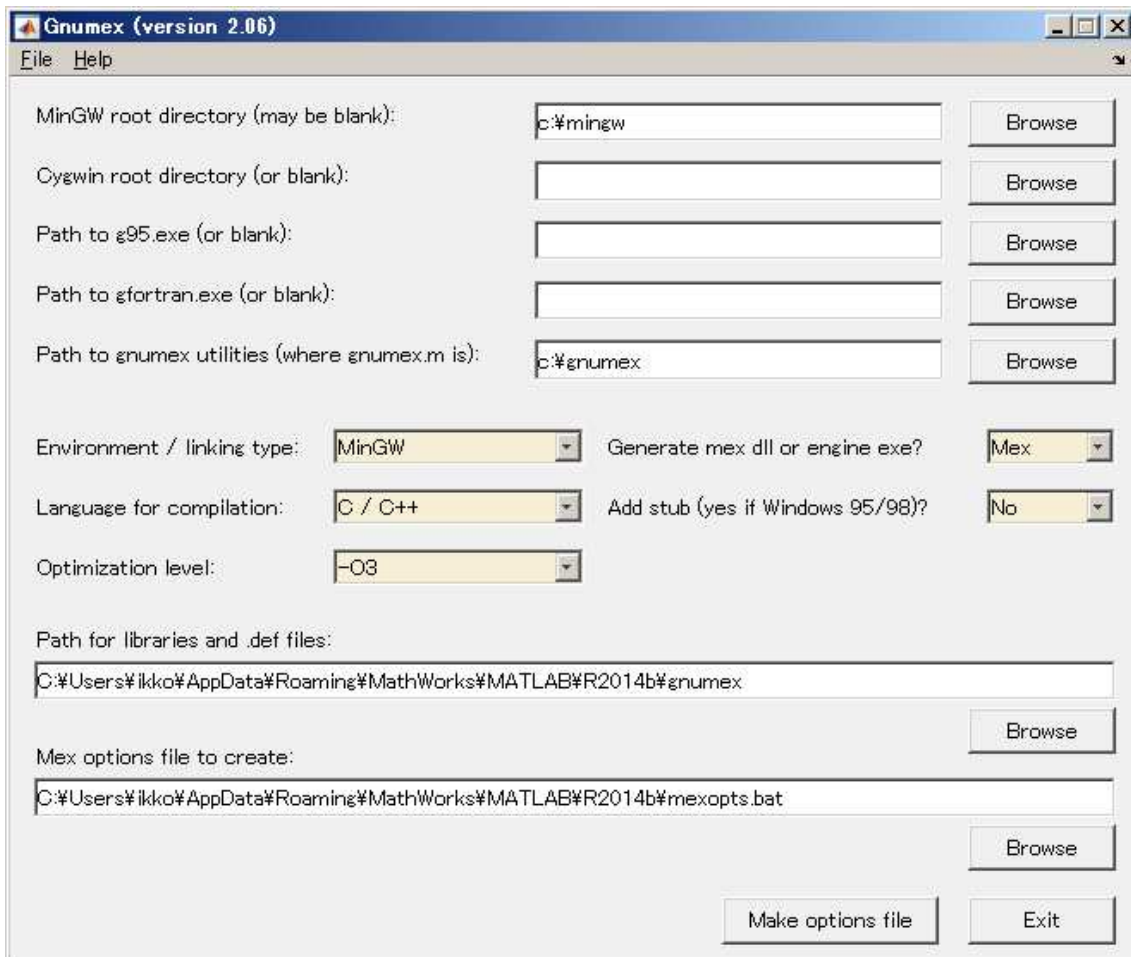
MinGW を使えるようにするため、Gnumex のサイト <http://gnumex.sourceforge.net/> から、

gnumex2.06.zip

をダウンロード、解凍し、trunk の中身を c:%gnumex に入れる。

MATLAB は、管理者権限で起動しないと、書き込みエラーが出るので注意！

```
>>cd c:%gnumex
>>gnumex
```



「Make options file」を作成する。

管理者権限で実行していない場合、アクセスが拒否されるので、以下のメッセージが出るので、手動でコピーする。

```
C:¥Users¥ikko¥AppData¥Roaming¥MathWorks¥MATLAB¥R2014b¥mexopts.bat
to the file
C:¥Program Files¥MATLAB¥R2014b¥bin¥win64¥mexopts¥GNUMEXOPTS.bat
. . .
C:¥Users¥ikko¥AppData¥Roaming¥MathWorks¥MATLAB¥R2014b¥mexopts.stp
to the file
C:¥Program Files¥MATLAB¥R2014b¥bin¥win64¥mexopts¥GNUMEXOPTS.stp
```

確認のためサンプルをコンパイルしてみる.

```
>> cd examples
>> cd gcc
```

```
>> mex -setup
```

警告: 互換性を確保するため古い形式の MEX 構造が提供されていますが、MATLAB の将来のバージョンでは削除される予定です。詳細は、MEX のリリース ノート <http://www.mathworks.com/help/matlab/release-notes.html> を参照してください。

Welcome to mex -setup. This utility will help you set up a default compiler. For a list of supported compilers, see <http://www.mathworks.com/support/compilers/R2014b/win64.html>

Please choose your compiler for building MEX-files:

Would you like mex to locate installed compilers [y]/n? y

Select a compiler:

[1] Microsoft Software Development Kit (SDK) 7.1 in C:¥Program Files (x86)¥Microsoft Visual Studio 10.0

[2] gcc in C:¥mingw¥bin

[0] None

Compiler: 2

Please verify your choices:

Compiler: gcc

Location: C:¥mingw¥bin

```
>>testGCC
```

で実行できるはずである.

Generic DLL の確認 (ここはやらなくてもよい。)

ここでは, MATLAB から mingw で作成した DLL を呼び出せるかのテストを行ってみる. まずは, 以下のプログラムを用意し, MATLAB のワーキングディレクトリに保存する.

test.h

```
#define EXPORT extern "C" __declspec(dllexport)
EXPORT double abcDllTest(int a, int b);
```

test.cc

```
#include "test.h"
EXPORT double abcDllTest(int a, int b)
{
    return (a + b) * 1.23456;
}
```

コンパイル方法は,

```
g++ -I. -O2 -Wno-deprecated -c -o test.o test.cc
g++ -shared -o abc.dll test.o
```

MATLAB を起動し, まずライブラリを呼び出す.

```
>>loadlibrary('abc.dll','test.h');
```

つぎに,

```
>> libfunctions abc -full
クラス lib.abc に対するメソッド:
double abcDllTest(int32, int32)
```

```
>> data=calllib('abc','abcDllTest',1,2)
data =
    3.7037
```

使い終わった場合には,

```
>> unloadlibrary('abc')
```

とすればよい.

yp-spur がコンパイルできている場合には、
 src¥.libs のディレクトリに入って
 include ディレクトリにあるヘッダファイルを全て src¥.libs フォルダ内にコピーしておく。

```
[a,b]=loadlibrary('libyppspur_dll.dll','yppspur.h')
```

が使えるようになる。enum 関連はサポートしていないので、ワーニングが出る。もし、それに関連する命令を使いたい場合には、直接数字を使う。

もし、エラーが出る場合には、`mex -setup`

を実行し、コンパイラを切り替えて、再実行してみるとよい。

```
>> libfunctions libyppspur_dll
```

クラス lib.libyppspur_dll に対するメソッド:

```
YPSpur_adjust_pos      YPSpur_near_ang      YPSpur_tilt      YP_openfree
YPSpur_circle YPSpur_near_pos YPSpur_unfreeze YP_set_control_state
YPSpur_free      YPSpur_orient      YPSpur_vel      YP_set_io_data
YPSpur_freeze      YPSpur_over_line      YP_get_ad_value      YP_set_io_dir
YPSpur_get_force      YPSpur_set_accel      YP_get_error_state      YP_set_parameter
YPSpur_get_pos          YPSpur_set_angaccel          YP_get_parameter
YP_set_parameter_array
YPSpur_get_vel          YPSpur_set_angvel          YP_get_parameter_array
YP_set_wheel_accel
YPSpur_init      YPSpur_set_pos      YP_get_vref      YP_set_wheel_vel
YPSpur_init_socket      YPSpur_set_vel      YP_get_wheel_ang      YP_wheel_ang
YPSpur_initex      YPSpur_spin      YP_get_wheel_torque      YP_wheel_torque
YPSpur_isfreeze      YPSpur_stop      YP_get_wheel_vel      YP_wheel_vel
YPSpur_line      YPSpur_stop_line      YP_get_wheel_vref
```

以下のコマンドを使うと、

```
>> libfunctionsview libyppspur_dll
```

戻り値のデータ型	名前	引数
int32	YPSpur_adjust_pos	(int32, double, double, double)
int32	YPSpur_circle	(int32, double, double, double)
int32	YPSpur_free	
int32	YPSpur_freeze	
[double, doublePtr, doublePtr]	YPSpur_get_force	(doublePtr, doublePtr)
[double, doublePtr, doublePtr, doublePtr]	YPSpur_get_pos	(int32, doublePtr, doublePtr, doublePtr)
[double, doublePtr, doublePtr]	YPSpur_get_vel	(doublePtr, doublePtr)
int32	YPSpur_init	
[int32, cstring]	YPSpur_init_socket	(cstring, int32)
int32	YPSpur_initex	(int32)
int32	YPSpur_isthreeze	
int32	YPSpur_line	(int32, double, double, double)
int32	YPSpur_near_ang	(int32, double, double)
int32	YPSpur_near_pos	(int32, double, double, double)
int32	YPSpur_orient	(int32, double)
int32	YPSpur_over_line	(int32, double, double, double)
int32	YPSpur_set_accel	(double)
int32	YPSpur_set_angaccel	(double)
int32	YPSpur_set_angvel	(double)
int32	YPSpur_set_pos	(int32, double, double, double)
int32	YPSpur_set_vel	(double)
int32	YPSpur_spin	(int32, double)
int32	YPSpur_stop	
int32	YPSpur_stop_line	(int32, double, double, double)
int32	YPSpur_tilt	(int32, double, double)
int32	YPSpur_unfreeze	
int32	YPSpur_vel	(double, double)
int32	YP_get_ad_value	(int32)
int32	YP_get_error_state	
[int32, doublePtr]	YP_get_parameter	(int32, doublePtr)
[int32, doublePtr]	YP_get_parameter_array	(int32, doublePtr)
[double, doublePtr, doublePtr]	YP_get_vref	(doublePtr, doublePtr)
[double, doublePtr, doublePtr]	YP_get_wheel_ang	(doublePtr, doublePtr)
[double, doublePtr, doublePtr]	YP_get_wheel_torque	(doublePtr, doublePtr)
[double, doublePtr, doublePtr]	YP_get_wheel_vel	(doublePtr, doublePtr)
[double, doublePtr, doublePtr]	YP_get_wheel_vref	(doublePtr, doublePtr)
int32	YP_openfree	
int32	YP_set_control_state	(int32, int32)
int32	YP_set_io_data	(uint8)
int32	YP_set_io_dir	(uint8)
int32	YP_set_parameter	(int32, double)
[int32, doublePtr]	YP_set_parameter_array	(int32, doublePtr)
int32	YP_set_wheel_accel	(double, double)
int32	YP_set_wheel_vel	(double, double)
int32	YP_wheel_ang	(double, double)
int32	YP_wheel_torque	(double, double)
int32	YP_wheel_vel	(double, double)

呼び出し方は,

```
>> calllib('libyemspur_dll', 'YPSpur_init')
ans =    -1
```

とする.

```
[a,b]=loadlibrary('libyemspur_md_dll', 'yemspur-md.h')
```

も可能である.

```
>> libfunctions libyemspur_md_dll
```

クラス lib.libyemspur_md_dll に対するメソッド:

YPSpur_md_adjust_pos	YPSpur_md_init_socket	YPSpur_md_set_accel
YPSpur_md_tilt	YP_md_get_wheel_ang	YP_md_set_parameter
YPSpur_md_circle	YPSpur_md_initex	YPSpur_md_set_angaccel
YPSpur_md_unfreeze	YP_md_get_wheel_torque	YP_md_set_parameter_array
YPSpur_md_free	YPSpur_md_isfreeze	YPSpur_md_set_angvel
YPSpur_md_vel	YP_md_get_wheel_vel	YP_md_set_wheel_accel
YPSpur_md_freeze	YPSpur_md_line	YPSpur_md_set_pos
YP_md_get_ad_value	YP_md_get_wheel_vref	YP_md_set_wheel_vel
YPSpur_md_get_force	YPSpur_md_near_ang	YPSpur_md_set_vel
YP_md_get_error_state	YP_md_openfree	YP_md_wheel_ang
YPSpur_md_get_pos	YPSpur_md_near_pos	YPSpur_md_spin
YP_md_get_parameter	YP_md_set_control_state	YP_md_wheel_torque
YPSpur_md_get_vel	YPSpur_md_orient	YPSpur_md_stop
YP_md_get_parameter_array	YP_md_set_io_data	YP_md_wheel_vel
YPSpur_md_init	YPSpur_md_over_line	YPSpur_md_stop_line
YP_md_get_vref	YP_md_set_io_dir	

>> libfunctionsview libyppspur_md_dll

戻り値のデータ型	名前	引数
[int32, YPSpurPtr]	YPSpur_md_adjust_pos	(YPSpurPtr, int32, double, double, double)
[int32, YPSpurPtr]	YPSpur_md_circle	(YPSpurPtr, int32, double, double, double)
[int32, YPSpurPtr]	YPSpur_md_free	(YPSpurPtr)
[int32, YPSpurPtr]	YPSpur_md_freeze	(YPSpurPtr)
[double, YPSpurPtr, doublePtr, doublePtr]	YPSpur_md_get_force	(YPSpurPtr, doublePtr, doublePtr)
[double, YPSpurPtr, doublePtr, doublePtr, doublePtr]	YPSpur_md_get_pos	(YPSpurPtr, int32, doublePtr, doublePtr, doublePtr)
[double, YPSpurPtr, doublePtr, doublePtr]	YPSpur_md_get_vel	(YPSpurPtr, doublePtr, doublePtr)
[int32, YPSpurPtr]	YPSpur_md_init	(YPSpurPtr)
[int32, YPSpurPtr, cstring]	YPSpur_md_init_socket	(YPSpurPtr, cstring, int32)
[int32, YPSpurPtr]	YPSpur_md_initex	(YPSpurPtr, int32)
[int32, YPSpurPtr]	YPSpur_md_isfreeze	(YPSpurPtr)
[int32, YPSpurPtr]	YPSpur_md_line	(YPSpurPtr, int32, double, double, double)
[int32, YPSpurPtr]	YPSpur_md_near_ang	(YPSpurPtr, int32, double, double)
[int32, YPSpurPtr]	YPSpur_md_near_pos	(YPSpurPtr, int32, double, double, double)
[int32, YPSpurPtr]	YPSpur_md_orient	(YPSpurPtr, int32, double)
[int32, YPSpurPtr]	YPSpur_md_over_line	(YPSpurPtr, int32, double, double, double)
[int32, YPSpurPtr]	YPSpur_md_set_accel	(YPSpurPtr, double)
[int32, YPSpurPtr]	YPSpur_md_set_angaccel	(YPSpurPtr, double)
[int32, YPSpurPtr]	YPSpur_md_set_angvel	(YPSpurPtr, double)
[int32, YPSpurPtr]	YPSpur_md_set_pos	(YPSpurPtr, int32, double, double, double)
[int32, YPSpurPtr]	YPSpur_md_set_vel	(YPSpurPtr, double)
[int32, YPSpurPtr]	YPSpur_md_spin	(YPSpurPtr, int32, double)
[int32, YPSpurPtr]	YPSpur_md_stop	(YPSpurPtr)
[int32, YPSpurPtr]	YPSpur_md_stop_line	(YPSpurPtr, int32, double, double, double)
[int32, YPSpurPtr]	YPSpur_md_tilt	(YPSpurPtr, int32, double, double)
[int32, YPSpurPtr]	YPSpur_md_unfreeze	(YPSpurPtr)
[int32, YPSpurPtr]	YPSpur_md_vel	(YPSpurPtr, double, double)
[int32, YPSpurPtr]	YP_md_get_ad_value	(YPSpurPtr, int32)
[int32, YPSpurPtr]	YP_md_get_error_state	(YPSpurPtr)
[int32, YPSpurPtr, doublePtr]	YP_md_get_parameter	(YPSpurPtr, int32, doublePtr)
[int32, YPSpurPtr, doublePtr]	YP_md_get_parameter_array	(YPSpurPtr, int32, doublePtr)
[double, YPSpurPtr, doublePtr, doublePtr]	YP_md_get_vref	(YPSpurPtr, doublePtr, doublePtr)
[double, YPSpurPtr, doublePtr, doublePtr]	YP_md_get_wheel_ang	(YPSpurPtr, doublePtr, doublePtr)
[double, YPSpurPtr, doublePtr, doublePtr]	YP_md_get_wheel_torque	(YPSpurPtr, doublePtr, doublePtr)
[double, YPSpurPtr, doublePtr, doublePtr]	YP_md_get_wheel_vel	(YPSpurPtr, doublePtr, doublePtr)
[double, YPSpurPtr, doublePtr, doublePtr]	YP_md_get_wheel_vref	(YPSpurPtr, doublePtr, doublePtr)
[int32, YPSpurPtr]	YP_md_openfree	(YPSpurPtr)
[int32, YPSpurPtr]	YP_md_set_control_state	(YPSpurPtr, int32, int32)
[int32, YPSpurPtr]	YP_md_set_io_data	(YPSpurPtr, uint8)
[int32, YPSpurPtr]	YP_md_set_io_dir	(YPSpurPtr, uint8)
[int32, YPSpurPtr]	YP_md_set_parameter	(YPSpurPtr, int32, double)
[int32, YPSpurPtr, doublePtr]	YP_md_set_parameter_array	(YPSpurPtr, int32, doublePtr)
[int32, YPSpurPtr]	YP_md_set_wheel_accel	(YPSpurPtr, double, double)
[int32, YPSpurPtr]	YP_md_set_wheel_vel	(YPSpurPtr, double, double)
[int32, YPSpurPtr]	YP_md_wheel_ang	(YPSpurPtr, double, double)
[int32, YPSpurPtr]	YP_md_wheel_torque	(YPSpurPtr, double, double)
[int32, YPSpurPtr]	YP_md_wheel_vel	(YPSpurPtr, double, double)

yp-spur の起動

```
cd C:\MinGW\msys\1.0\home\ikko\yp-spur\src\libs
```

とすると、DLL、EXE ファイルができていますのでこれを使う。

まず、yp_coordinator.exe を起動する。

```
ypspur-coordinator.exe -p M1.param -d COM2
```

Param は、サイトからダウンロードして予めサイトからダウンロードして同じフォルダに入れておく。 COM 番号は、10 以上の場合には、[¥¥¥¥.¥¥COM10](#) などと書く必要があるの
で、もし、10 以上の番号になってしまった場合には、変更する必要があります。

サンプルスクリプトの対応

CS_GL=2

C	MATLAB
Spur_init()	YPSpur_init()
Spur_set_vel(0.2);	YPSpur_set_vel(v)
Spur_set_accel(1.0);	YPSpur_set_accel(v)
Spur_set_angvel(M_PI / 2.0);	YPSpur_set_angvel(w)
Spur_set_angaccel(M_PI / 2.0);	YPSpur_set_angaccel(w)
Spur_set_pos_GL(0, 0, 0);	YPSpur_get_pos(2, x,y,th)
Spur_stop_line_GL(1.0, 0, 0);	YPSpur_stop_line(2, x,y,th)
Spur_over_line_GL(1.0 - 0.005, 0.0, 0.0)	YPSpur_over_line(2,x,y,th)
Spur_spin_GL(M_PI / 2);	YPSpur_spin(2, th)
Spur_near_ang_GL(M_PI / 2, M_PI / 18.0)	YPSpur_near_ang(2,th,d)
Spur_stop_line_GL(1.0, 0.1, M_PI / 2.0);	YPSpur_stop_line(2, x,y,th)
Spur_stop();	YPSpur_stop()
Spur_free();	YPSpur_free()
Spur_get_pos_GL(&x, &y, &theta);	YPSpur_get_pos(2, x,y,th)

MATLAB からの呼び出しは、test_circle で行う。

YPSpur_init.m

```

if ~libisloaded('libyppspur_dll')
    [yp_a,yp_b]=loadlibrary('libyppspur_dll.dll','yppspur.h');
end

TF_x=0;TF_y=0;TF_th=0;
TF_x=libpointer('doublePtr',TF_x);
TF_y=libpointer('doublePtr',TF_y);
TF_th=libpointer('doublePtr',TF_th);

GL=2;LC=3;FS=4;
% typedef enum
%     {
%         CS_BS = 0,
%         CS_SP,
%         CS_GL,
%         CS_LC,
%         CS_FS,
%         CS_BL,
%         CS_MAX
%     } YPSpur_cs;

calllib('libyppspur_dll','YPSpur_init');
%最大速度設定
calllib('libyppspur_dll','YPSpur_set_vel',2.5);

```

```

%最大加速度設定
calllib('libyemspur_dll','YPSpur_set_accel',5.0);
%角速度設定
calllib('libyemspur_dll','YPSpur_set_angvel',pi/2);
%角加速度設定
calllib('libyemspur_dll','YPSpur_set_angaccel',pi);

%初期位置設定
calllib('libyemspur_dll','YPSpur_set_pos',GL, 0, 0, 0 );
calllib('libyemspur_dll','YPSpur_set_pos',LC, 0, 0, 0 );

```

test_circle.m

```

clear all;close all;

YPSpur_init;
time1=0;history=[0,0,0];j=1;

f=calllib('libyemspur_dll','YPSpur_circle',2,0,1,-0.5);
while(1)
    time1=time1+0.01;
    if(time1>0.05)

[aa,TF_x,TF_y,TF_th]=calllib('libyemspur_dll','YPSpur_get_pos',GL,TF_x,TF_
y,TF_th)
        pos=[TF_x,TF_y,TF_th];
        history(j,:)=pos;
        j=j+1;
        time1=0;
    end
    plot(history(:,1),history(:,2))
    axis equal;
    drawnow

    if(time1>1)
        ff=calllib('libyemspur_dll','YPSpur_stop');
        break
    end
end
end

```

Run_test.c をベースに開発する.

```

if ~libisloaded('libyemspur_dll')
    [yp_a,yp_b]=loadlibrary('libyemspur_dll.dll','yemspur.h');
end

TF_x=0;TF_y=0;TF_th=0;
TF_x=libpointer('doublePtr',TF_x);
TF_y=libpointer('doublePtr',TF_y);
TF_th=libpointer('doublePtr',TF_th);

```

```
GL=2;LC=3;FS=4;
calllib('libyppur_dll','YPSpur_init');
calllib('libyppur_dll','YPSpur_set_vel',0.2);
calllib('libyppur_dll','YPSpur_set_accel',1.0);
calllib('libyppur_dll','YPSpur_set_angvel',pi/2);
calllib('libyppur_dll','YPSpur_set_angaccel',pi/2);

calllib('libyppur_dll','YPSpur_set_pos',GL, 0, 0, 0 );
calllib('libyppur_dll','YPSpur_set_pos',LC, 0, 0, 0 );
fprintf('line¥n');
f=calllib('libyppur_dll','YPSpur_stop_line',GL,1.0,0,0);
f=0;
while(~f)
    f = calllib('libyppur_dll','YPSpur_over_line',GL,1.0-0.005,0,0);
end
fprintf('spin¥n');
calllib('libyppur_dll','YPSpur_spin',GL,pi/2);
f=0;
while(~f)
    f = calllib('libyppur_dll','YPSpur_near_ang',GL,pi/2,pi/18);
end
calllib('libyppur_dll','YPSpur_set_vel',0.3);
calllib('libyppur_dll','YPSpur_set_accel',1.0);
calllib('libyppur_dll','YPSpur_set_angvel',pi);
calllib('libyppur_dll','YPSpur_set_angaccel',pi);
f=calllib('libyppur_dll','YPSpur_stop_line',GL,1.0,0.1,pi/2);
f=0;
while(~f)
    f = calllib('libyppur_dll','YPSpur_over_line',GL,1,0.1-0.005,pi/2);
end
fprintf('spin¥n');
calllib('libyppur_dll','YPSpur_spin',GL,pi);
f=0;
while(~f)
    f = calllib('libyppur_dll','YPSpur_near_ang',GL,pi,pi/18);
end
```

```

fprintf('line№n');
f=calllib('libyemspur_dll','YPSpur_stop_line',GL,0,0.1,pi);
f=0;
while(~f)
    f = calllib('libyemspur_dll','YPSpur_over_line',GL,0+0.005,0.1,pi);
end
fprintf('spin№n');
calllib('libyemspur_dll','YPSpur_spin',GL,-pi/2);
f=0;
while(~f)
    f = calllib('libyemspur_dll','YPSpur_near_ang',GL,-pi/2,pi/18);
end
fprintf('line№n');
f=calllib('libyemspur_dll','YPSpur_stop_line',GL,0,0,-pi/2);
f=0;
while(~f)
    f = calllib('libyemspur_dll','YPSpur_over_line',GL,0,0+0.005,-pi/2);
end
fprintf('spin№n');
calllib('libyemspur_dll','YPSpur_spin',GL,0);
f=0;
while(~f)
    f = calllib('libyemspur_dll','YPSpur_near_ang',GL,0,pi/18);
end
calllib('libyemspur_dll','YPSpur_stop');
calllib('libyemspur_dll','YPSpur_free');
[aa,TF_x,TF_y,TF_th] =
calllib('libyemspur_dll','YPSpur_get_pos',GL,TF_x,TF_y,TF_th);
fprintf('%f %f %f№n',TF_x,TF_y, TF_th * 180.0/pi);

```