

マニピュレータの動作確認

Updated on: 2019-11-12

[トップページ](#)

- [サンプルプログラムのコンパイル](#)
- [サンプルプログラムの実行](#)
- [課題](#)

簡単なサンプルプログラムを利用してマニピュレータの関節（サーボモータ）動作を確認します。

サンプルプログラムのコンパイル

1. サーボモータドライバをダウンロードしてコンパイルします。
ターミナルを起動し、以下のコマンドを実行します。

```
1  $ cd ~/Downloads/
2  $ git clone https://github.com/ROBOTIS-GIT/DynamixelSDK.git
3  $ cd DynamixelSDK/c/build/linux64/
4  $ make
5  mkdir -p ./objects/
6  gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../include -m64 -fPIC -g -c
7  ../src/dynamixel_sdk/group_bulk_read.c -o ./objects/group_bulk_read.o
8  gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../include -m64 -fPIC -g -c
9  ../src/dynamixel_sdk/group_bulk_write.c -o ./objects/group_bulk_write.o
10 gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../include -m64 -fPIC -g -c
11 ../src/dynamixel_sdk/group_sync_read.c -o ./objects/group_sync_read.o
12 gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../include -m64 -fPIC -g -c
13 ../src/dynamixel_sdk/group_sync_write.c -o ./objects/group_sync_write.o
14 gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../include -m64 -fPIC -g -c
15 ../src/dynamixel_sdk/packet_handler.c -o ./objects/packet_handler.o
16 gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../include -m64 -fPIC -g -c
17 ../src/dynamixel_sdk/port_handler.c -o ./objects/port_handler.o
18 gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../include -m64 -fPIC -g -c
19 ../src/dynamixel_sdk/protocol1_packet_handler.c -o ./objects/protocol1
20 _packet_handler.o
21 gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../include -m64 -fPIC -g -c
22 ../src/dynamixel_sdk/protocol2_packet_handler.c -o ./objects/protocol2
23 _packet_handler.o
24 gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../include -m64 -fPIC -g -c
25 ../src/dynamixel_sdk_linux/port_handler_linux.c -o ./objects/port_han
26 dler_linux.o
27 g++ -shared -fPIC -m64 -o ./libdxl_x64_c.so ./objects/group_bulk_read.o
28 ./objects/group_bulk_write.o ./objects/group_sync_read.o ./objects/gr
29 oup_sync_write.o
30 ./objects/packet_handler.o ./objects/port_handler.o
31 ./objects/protocol1_packet_handler.o ./objects/protocol2_packet_handler.o
32 ./objects/port_handler_linux.o -lrt
```

2. サーボモータ確認用プログラムをダウンロードします。

```
1  $ cd ~/Downloads/
2  $ git clone https://github.com/takahasi/dynamixel_servo_check.git
3  $ cd dynamixel_servo_check
```

3. サーボモータ確認用プログラムがドライバのAPIを利用できるように、サーボモータドライバのライブラリとヘッダーをコピーします。

```
1  $ cp -r ~/Downloads/DynamixelSDK/c/include/dynamixel_sdk/ .
2  $ cp ~/Downloads/DynamixelSDK/c/build/linux64/libdxl_x64_c.so .
3  $ ls
4  CMakeLists.txt  dynamixel_sdk  libdxl_x64_c.so  LICENSE  README.md
5  servo_check.c
```

4. サーボモータ確認プログラムをコンパイルします。

```
1  $ mkdir build
2  $ cd build
3  $ cmake ../
4  -- The C compiler identification is GNU 5.4.0
5  -- The CXX compiler identification is GNU 5.4.0
6  -- Check for working C compiler: /usr/bin/cc
7  -- Check for working C compiler: /usr/bin/cc -- works
8  -- Detecting C compiler ABI info
9  -- Detecting C compiler ABI info - done
10 -- Detecting C compile features
11 -- Detecting C compile features - done
12 -- Check for working CXX compiler: /usr/bin/c++
13 -- Check for working CXX compiler: /usr/bin/c++ -- works
14 -- Detecting CXX compiler ABI info
15 -- Detecting CXX compiler ABI info - done
16 -- Detecting CXX compile features
17 -- Detecting CXX compile features - done
18 -- Configuring done
19 -- Generating done
20 -- Build files have been written to: /home/geoff/Downloads/dynamixel
21 _servo_check/build
22 $ make
23 [ 50%] Building C object CMakeFiles/servo_check.dir/servo_check.c.o
24 [100%] Linking C executable servo_check
25 [100%] Built target servo_check
26 $ ls
27 CMakeCache.txt  CMakeFiles  cmake_install.cmake  Makefile  servo_check
```

サンプルプログラムの実行

1. シリアルポートへアクセスするために、シリアルデバイスへのアクセス件を付与します。
下記のコマンドでパーミッショングループにユーザを追加します。

```
1  $ sudo gpasswd -a ユーザ名 dialout
```

上記のユーザ名の部分を現在お使いのユーザ名に変更します。

上記のようにパーミッショングループにユーザを追加しないとサーボモータ制御ソフトウェアはハードウェアへアクセスできないため、エラーになります。

また、 **アクセス権をシステムに反映されるために、上記コマンド実行後は一旦ログアウトして再ログインしてください。**

2. サーボモータ確認プログラムを実行し、サーボモータの動作を確認します。

プログラムにサーボIDを指定します。CRANE+のサーボIDは1～5です。

また、サーボモータの目標位置を設定します。

CRANE+で利用しているサーボモータは1024段階（0～1023）で目標位置を指定できますので、下記の例では中央値である512を指定します。

すべてのモータに512を指定すると、CRANE+は直立した姿勢となります。

注意：プログラムを実行すると指定したサーボモータは高速で指定位置に移動します。電源を入れる前にマニピュレータをまっすぐ上向きに近い姿勢にしてください。

```
1  $ cd ~/Downloads/dynamixel_servo_check/build/
2  $ ./servo_check 1 512
3  opened port
4  changed buadrate
5  dynamixel has been successfully connected
6  [ID:003] GoalPos:512 PresPos:760
7  [ID:003] GoalPos:512 PresPos:757
8  [ID:003] GoalPos:512 PresPos:744
9  [ID:003] GoalPos:512 PresPos:730
10 [ID:003] GoalPos:512 PresPos:718
11 [ID:003] GoalPos:512 PresPos:705
12 [ID:003] GoalPos:512 PresPos:687
13 [ID:003] GoalPos:512 PresPos:672
14 [ID:003] GoalPos:512 PresPos:656
15 [ID:003] GoalPos:512 PresPos:640
16 [ID:003] GoalPos:512 PresPos:623
17 [ID:003] GoalPos:512 PresPos:604
18 [ID:003] GoalPos:512 PresPos:586
```

19	[ID:003]	GoalPos:512	PresPos:564
20	[ID:003]	GoalPos:512	PresPos:543
21	[ID:003]	GoalPos:512	PresPos:522

課題

- 1. サーボID 1 から 5 まで、全サーボモータの動作を確認してみましょう。
- 2. 上記プログラムの指令値を変更することでロボットアームをの手先を任意の姿勢に到達させてみましょう（例えば、机の上にある物体を掴む姿勢など）。

問い合わせ先：

高橋 三郎（パナソニック アドバンステクノロジー）(takahashi dot saburo at jp dot panasonic dot com)

長谷川 孔明（豊橋技術科学大学）

This project is maintained by [takahasi](#)

Generated on 2019-11-13

Hosted on GitHub Pages — Theme by [orderedlist](#)



This work is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).