

Dynamixel AX-12 プログラム解説

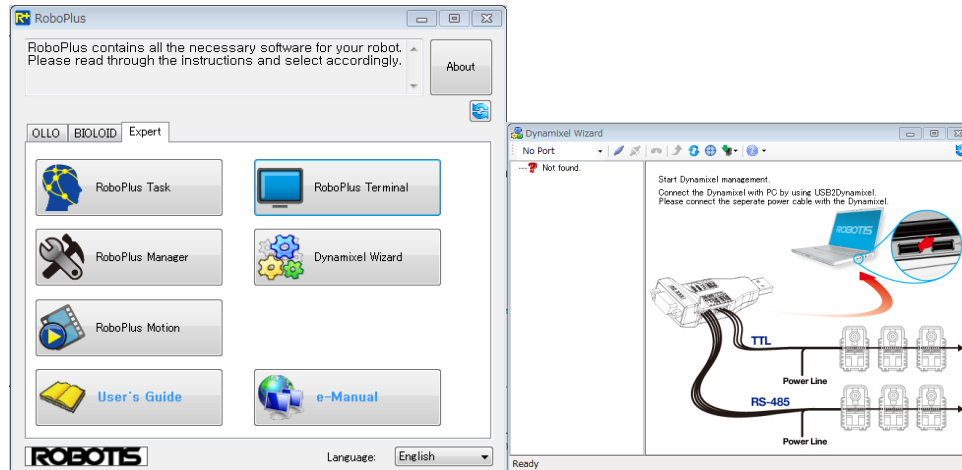
- 初期設定・環境構築

➤ <https://github.com/ais-lab/sotsuken1/wiki/ArduinoIDE>

- RoboPlus

Dynamixel の ID 変更やモードの変更などに用いる。

Dynamixel Wizard を開き



openCM に電源をつなぎ、OpenCM からモータにつなぐ

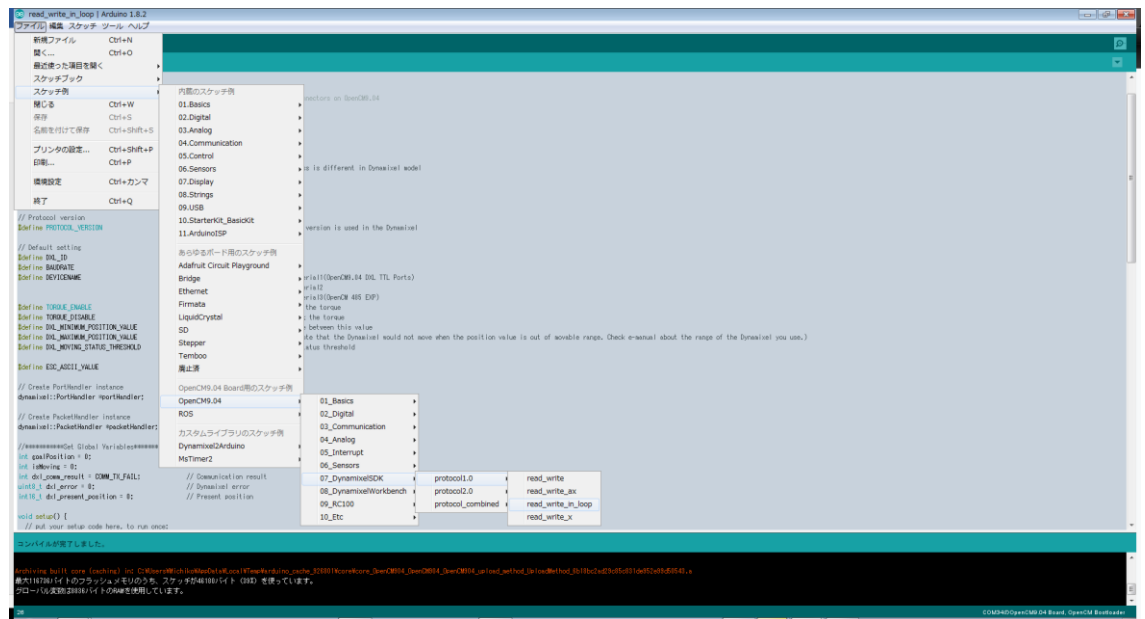
モータの余ってるポートから USB2Dynamixel につなぎ TTL にスイッチを設定する。openCM の電源をつけるとモータの背面のランプが一度点灯したら電源が流れているという合図。

ポートを指定しコネクトすると検索開始ボタンが表示されるため検索を開始する。その後、モータが検出され、id の変更や車輪モード、関節モードなどの変更が可能となる。

- プログラム解説

➤ サンプルプログラム

ファイル→スケッチ例→OpenCM9.04→Dynamixel SDK→Protocol1.0→
read_write_in_loop



- プログラムの大枠の内容

Define：不変な値の定義。

Setup0:コンストラクタと同様。プログラム実行時一度だけ実行されるメソッド

loop0:メインメソッド。プログラム実行中、メソッド内の処理がループする。

- プログラムの詳細内容

トルクの ON/OFF

Setup0内は初期設定。ポートの設定やボードレートの設定など。

以下のプログラム部分が重要

```
// Enable Dynamixel Torque
```

```
dxl_comm_result = packetHandler->write1ByteTxRx(portHandler, DXL_ID,
ADDR_AX_TORQUE_ENABLE, TORQUE_ENABLE, &dxl_error);
```

write1ByteTxRx0メソッド:1byte のデータを書き込むメソッド

引数: (ポートハンドラー, dynamixel の ID, dynamixel のアドレス, 書き込みたい値, エラー値)

ポートハンドラーとエラー値は固定。

引数 DXL_ID に動かしたい dynamixel の id を指定。

ADDR_AX_TORQUE_ENABLE は define にて 24 と定義されている。

この 24 は dynamixel へ命令を送るアドレス番号である。

(<http://emanual.robotis.com/docs/en/dxl/ax/ax-12a/>)

上記のページへアクセスすると以下のようなテーブルがある。

2. 3. Control Table of RAM Area

Address	Size (Byte)	Data Name	Description	Access	Initial Value
24	1	Torque Enable	Motor Torque On/Off	RW	0
25	1	LED	Status LED On/Off	RW	0
26	1	CW Compliance Margin	CW Compliance Margin	RW	1
27	1	CCW Compliance Margin	CCW Compliance Margin	RW	1
28	1	CW Compliance Slope	CW Compliance Slope	RW	32
29	1	CCW Compliance Slope	CCW Compliance Slope	RW	32
30	2	Goal Position	Target Position	RW	-
32	2	Moving Speed	Moving Speed	RW	-
34	2	Torque Limit	Torque Limit(Goal Torque)	RW	ADD 14&15
36	2	Present Position	Present Position	R	-
38	2	Present Speed	Present Speed	R	-
40	2	Present Load	Present Load	R	-
42	1	Present Voltage	Present Voltage	R	-
43	1	Present Temperature	Present Temperature	R	-
44	1	Registered	If Instruction is registered	R	0
46	1	Moving	Movement Status	R	0
47	1	Lock	Locking EEPROM	RW	0
48	2	Punch	Minimum Current Threshold	RW	32

テーブルの左の列が **dynamixel** へ命令を送るアドレス番号である。

例えば、アドレス 24 番をみると、size が 1byte, **dynamixel** のトルクを ON/OFF 切り替える命令が実行可能なアドレスであることがわかる。

つまり、`write1ByteTxRx()`にて 1byte のデータを 24 番のアドレスに書き換えるという命令を行っている。自分が動かしたいモータが複数の場合はこの一行を複数個分書き、それぞれモータの id を設定してあげる必要がある。

モータの回転

`Loop()`メソッド内でモータを動かしている。

トルクの場合と同様に以下のプログラムで動かしている。

```
dxl_comm_result = packetHandler->write2ByteTxRx(portHandler, DXL_ID, ADDR_AX_GOAL_POSITION, goalPosition, &dxl_error);
```

先ほどは `write1ByteTxRx()`を用いたが、今回は `write2ByteTxRx` となっている

点に注意する。

2. 3. Control Table of RAM Area

Address	Size (Byte)	Data Name	Description	Access	Initial Value
24	1	Torque Enable	Motor Torque On/Off	RW	0
25	1	LED	Status LED On/Off	RW	0
26	1	CW Compliance Margin	CW Compliance Margin	RW	1
27	1	CCW Compliance Margin	CCW Compliance Margin	RW	1
28	1	CW Compliance Slope	CW Compliance Slope	RW	32
29	1	CCW Compliance Slope	CCW Compliance Slope	RW	32
30	2	Goal Position	Target Position	RW	-
32	2	Moving Speed	Moving Speed	RW	-
34	2	Torque Limit	Torque Limit(Goal Torque)	RW	ADD 14&15
36	2	Present Position	Present Position	R	-
38	2	Present Speed	Present Speed	R	-
40	2	Present Load	Present Load	R	-
42	1	Present Voltage	Present Voltage	R	-
43	1	Present Temperature	Present Temperature	R	-
44	1	Registered	If Instruction is registered	R	0
46	1	Moving	Movement Status	R	0
47	1	Lock	Locking EEPROM	RW	0
48	2	Punch	Minimum Current Threshold	RW	32

その理由として引数である ADDR_AX_GOAL_POSITION に注目する。Define にて 30 と定義されている。アドレス表の 30 を見ると size が 2byte となっている。よって、先ほどの write1byteTxRx() では書き込むデータサイズが 1byte 足りないため、write2byteTxRx() を用いる必要がある。また、30 番のアドレスに書き込む値として変数 goalPosition を引数に用いている。この値の詳細は 30 番のアドレス表の青字で記載されている Goal Position を参考に値を検討する。

その他の Moving, Present Position などは write ではなく read を用いてアドレスから値を読み取り、dynamixel が動いているかどうか、今の角度情報などを取得している。

つまり、dynamixel を動かすだけであれば、以下の一文を動かしたい dynamixel 分記載してやればモータは goalPosition で指定した角度まで動く。

```
dxl_comm_result = packetHandler->write2ByteTxRx(portHandler, DXL_ID, ADDR_AX_GOAL_POSITION, goalPosition, &dxl_error);
```

また、車輪モードとして動かしたい場合は dynamixel のアドレス 32 番、Moving speed に適切な値を引数として書き込んであげればよい。

- 上記を理解した上でモータが動かない場合

上記で紹介した **sample** プログラムは関節モードの **dynamixel** のみ動く。

よって、**roboplus** より車輪モードから関節モードに変更すると動く。