

BuilditActuator ユーザーマニュアル

Smart Robotics Co.Ltd.

2022/10/17

目 次

1	変更履歴	6
2	はじめに	6
3	使用上の注意とお願い	7
4	システムフォルトについて	8
5	概要	9
5.1	梱包内容	9
5.2	各部の名称と働き	9
5.2.1	状態 LED について	10
5.3	Buildit の状態遷移	10
5.4	ロータの座標系について	12
5.5	電池寿命について	12
6	セットアップ手順	13
6.1	Python パッケージのインストール	13
6.1.1	Windows	13
6.1.2	Ubuntu	13
6.2	PC、電源の接続	13
6.3	Buildit Actuator を回す	13
6.3.1	Windows	13
6.3.2	Ubuntu	14
6.3.3	GUI ツールを用いた操作	14
7	電氣的インターフェイス	17
7.1	電源コネクタ	17
7.2	通信コネクタ	18
7.3	デイジーチェーン接続	18
7.4	RS485 の終端抵抗	18
7.5	保護停止	18
7.6	電源について	19
8	通信プロトコル仕様	20
8.1	概要	20
8.2	シリアル通信の設定	20
8.3	エンディアン	20
8.4	状態の種別	20
8.5	異常	21
8.6	メッセージの種別	21
8.7	メッセージの共通フォーマット	23
8.7.1	応答メッセージの共通フォーマット	24
8.7.2	CRC-8-ATM 計算ルーチンの例	25
8.8	無効なメッセージについて	26
8.9	通信についての注意事項	26
8.10	パラメータについて	27

8.10.1	出荷時のパラメータの値	28
8.10.2	位置に関するパラメータの操作の制限	28
8.11	範囲制限機能について	28
8.11.1	位置範囲制限外の振る舞い	29
8.11.2	速度範囲制限外の振る舞い	29
8.11.3	電流範囲制限外の振る舞い	29
8.11.4	ソフトウェアリミットによる位置範囲制限	29
8.12	保護停止ピン極性の切り替えについて	29
8.13	CLEAR_FAULT_CMD	30
8.13.1	概要	30
8.13.2	事前条件	30
8.13.3	コマンドのペイロードフォーマット	30
8.13.4	例	30
8.13.5	成功応答メッセージのペイロードフォーマット	30
8.14	FAULT	31
8.14.1	概要	31
8.14.2	事前条件	31
8.14.3	コマンドのペイロードフォーマット	31
8.14.4	例	31
8.14.5	成功応答メッセージのペイロードフォーマット	31
8.15	FREE_CMD	32
8.15.1	概要	32
8.15.2	事前条件	32
8.15.3	コマンドのペイロードフォーマット	32
8.15.4	例	32
8.15.5	成功応答メッセージのペイロードフォーマット	32
8.16	GET_LOG_CMD	33
8.16.1	概要	33
8.16.2	事前条件	33
8.16.3	コマンドのペイロードフォーマット	33
8.16.4	例	33
8.16.5	成功応答メッセージのペイロードフォーマット	33
8.16.6	ログレコードについて	34
8.17	GET_LOG_INFO_CMD	35
8.17.1	概要	35
8.17.2	事前条件	35
8.17.3	コマンドのペイロードフォーマット	35
8.17.4	例	35
8.17.5	成功応答メッセージのペイロードフォーマット	35
8.17.6	概要	36
8.17.7	事前条件	36
8.17.8	コマンドのペイロードフォーマット	36
8.17.9	例	36
8.17.10	成功応答メッセージのペイロードフォーマット	37
8.18	GET_REF_CURRENT_CMD	38
8.18.1	概要	38
8.18.2	事前条件	38

8.18.3	コマンドのペイロードフォーマット	38
8.18.4	例	38
8.18.5	成功応答メッセージのペイロードフォーマット	38
8.19	GET_REF_POSITION_CMD	39
8.19.1	概要	39
8.19.2	事前条件	39
8.19.3	コマンドのペイロードフォーマット	39
8.19.4	例	39
8.19.5	成功応答メッセージのペイロードフォーマット	39
8.20	GET_REF_VELOCITY_CMD	40
8.20.1	概要	40
8.20.2	事前条件	40
8.20.3	コマンドのペイロードフォーマット	40
8.20.4	例	40
8.20.5	成功応答メッセージのペイロードフォーマット	40
8.21	HOLD_CMD	41
8.21.1	概要	41
8.21.2	事前条件	41
8.21.3	例	41
8.21.4	コマンドのペイロードフォーマット	41
8.21.5	成功応答メッセージのペイロードフォーマット	41
8.22	NACK	42
8.22.1	概要	42
8.23	失敗の種別	42
8.23.1	失敗応答メッセージのペイロードフォーマット	42
8.24	PROTECTION_STOP	43
8.24.1	概要	43
8.24.2	事前条件	43
8.24.3	コマンドのペイロードフォーマット	43
8.24.4	例	43
8.24.5	成功応答メッセージのペイロードフォーマット	44
8.25	QUERY_SERVO_STATUS_CMD	45
8.25.1	概要	45
8.25.2	事前条件	45
8.25.3	コマンドのペイロードフォーマット	45
8.25.4	例	45
8.25.5	成功応答メッセージのペイロードフォーマット	45
8.26	READY_CMD	47
8.26.1	概要	47
8.26.2	事前条件	47
8.26.3	コマンドのペイロードフォーマット	47
8.26.4	例	47
8.26.5	応答メッセージのペイロードフォーマット	47
8.27	RESET_ROTATION	48
8.27.1	概要	48
8.27.2	事前条件	48
8.27.3	コマンドのペイロードフォーマット	48

8.27.4 例	48
8.27.5 成功応答メッセージのペイロードフォーマット	49
8.28 SET_PARAM_CMD	50
8.28.1 概要	50
8.28.2 事前条件	50
8.28.3 コマンドのペイロードフォーマット	50
8.28.4 例	51
8.28.5 成功応答メッセージのペイロードフォーマット	51
8.29 SET_REF_CURRENT_CMD	52
8.29.1 概要	52
8.29.2 事前条件	52
8.29.3 コマンドのペイロードフォーマット	52
8.29.4 例	52
8.29.5 成功応答メッセージのペイロードフォーマット	52
8.30 SET_REF_POSITION_CMD	54
8.30.1 概要	54
8.30.2 事前条件	54
8.30.3 コマンドのペイロードフォーマット	54
8.30.4 例	54
8.30.5 成功応答メッセージのペイロードフォーマット	55
8.31 SET_REF_POSITION_WITH_VELOCITY_CMD	56
8.31.1 概要	56
8.31.2 事前条件	56
8.31.3 コマンドのペイロードフォーマット	56
8.31.4 例	56
8.31.5 成功応答メッセージのペイロードフォーマット	57
8.32 SET_REF_VELOCITY_CMD	58
8.32.1 概要	58
8.32.2 事前条件	58
8.32.3 コマンドのペイロードフォーマット	58
8.32.4 例	58
8.32.5 成功応答メッセージのペイロードフォーマット	58
9 スペック	60
9.1 仕様	60
9.2 寸法図	60
10 サードパーティ製品のライセンスについて	62

1 変更履歴

- ver.1.0(2020/03/16)
 - 新規作成
- ver.1.1(2020/04/22)
 - 不正なメッセージ ID を指定しない注意の文言を修正
 - SET_PARAM_CMD による PARAM_ID_DEVICE_ID 設定時の振る舞いを修正
 - 読み込み専用パラメータについて追記
- ver.1.2(2022/10/17)
 - GUI の更新に対応
 - 保護停止ピンの極性反転機能について追記
 - 位置指令値設定（速度指定）について追記

2 はじめに

このたびは「Buildit Actuator」をお買い求めいただき、誠にありがとうございます。Buildit Actuator をお使いいただく際は、本書をよくお読みになり、正しくお使いください。また、本書はいつでもご利用いただけるように、大切に保管してください。

- 本製品によるいかなる損害や逸失利益などについても、当社はその一切の責任を負いかねます。
- 本書の一部または全部を無断で転載することはご遠慮ください。
- 本書の内容は予告なく変更されることがあります。
- 本書に記載された写真やイラストは一部が実製品と異なる場合があります。
- 本書に関してご不明な点などお気づきの点がございましたら、当社までご連絡ください。
- 本製品を日本国外に輸出しないでください。

3 使用上の注意とお願い

- ご利用前には必ず十分なリスクアセスメントを行って下さい。
 - － 本製品は直接生命や身体に危害を及ぼす可能性のあるシステムや、多大な損失を発生させる可能性のあるシステムに使用されることを想定していません。
- 強い衝撃を与えたりしないでください。故障や誤動作の原因になります。
- 緊急時以外、回転中の電磁ブレーキによる制動は極力避けて下さい。ブレーキ寿命の低下・故障・誤動作の原因になります。特に回転中の電源 **OFF** やフォルト発生等にはご注意ください。
- 出荷時に設定されている制御パラメータはお使いの制御対象によっては適切でない場合があります。オーバーシュート、振動、追従誤差が大きいといった挙動に問題がある場合は制御パラメータを適切に調整して下さい。
 - － ゲイン変更の際は細心の注意を払って下さい。不適切なゲインは暴走や故障の原因となり、思わぬ怪我や事故を招く恐れがあります。
- 分解・改造は絶対に行わないで下さい。分解・改造した場合は保証対象外となります。
- 使用条件や放熱環境によっては、内蔵モーターが燃損する事があります。特に連続回転させる場合などには放熱に十分ご注意ください。
- 強い電磁ノイズを受けるなどすると、まれに誤動作する可能性があります。使用する際には安全に十分ご注意ください。
- 本製品に防水機能はありません。液体がかかったり、濡れた手で触るなどすると故障の原因となります。
- 異臭・異音・アクチュエータ表面の異常発熱・発火が発生した場合は、ただちに使用を中止し、弊社窓口まで必ずご連絡ください。
- 定格範囲内の使用であっても、長時間使用時に、温度制限が発生する可能性があります。また、ケースが熱くなる為、火傷等にご注意下さい。
- 起動直後数秒間は正しく温度が計測出来ない場合があります。
- まれにロータ (回転部) よりグリスが染み出ることがあります。品質には影響ありませんので、グリスを拭き取ってご使用ください。

4 システムフォルトについて

状態 LED が赤色緑色ともに点灯状態となり、コマンドを全く受け付けない場合、Buildit Actuator 内で深刻なエラー（システムフォルト）が発生しています。システムフォルトが発生した場合は、弊社窓口まで必ずご連絡ください。

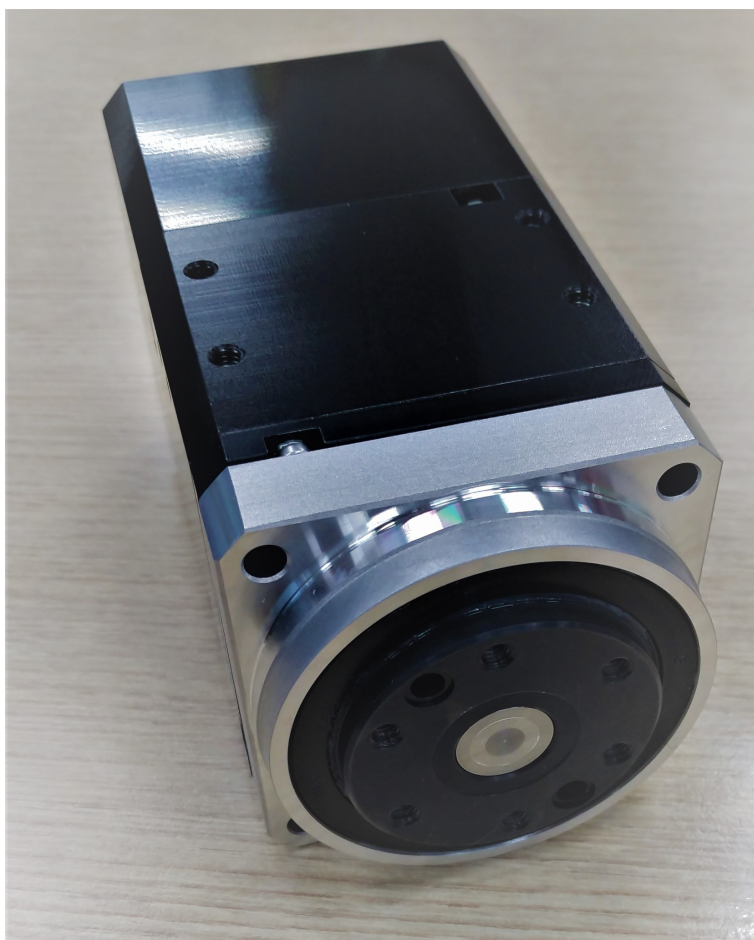


Figure 1: Buildit Actuator

5 概要

Buildit Actuator はモータ、減速機、ブレーキ、クロスローラベアリング、コントローラが一体となったオールインワンアクチュエータです。Buildit Actuator を使用することで、メカや電気に関する様々な設計を省略することができ、アプリケーションの開発に専念することができます。

5.1 梱包内容

Buildit Actuator を使用する前に、必ず梱包されている内容を確認してください。

- Buildit Actuator 本体 (fig. 1) : 1 台
- Buildit Actuator スタートアップガイド : 1 枚

5.2 各部の名称と働き

fig. 2 が各部の名称の一覧です。以下では、各部の働きについて説明します。

名称	説明
電源コネクタ	24V 電源を接続するコネクタです。

名称	説明
電源コネクタ	電源コネクタは2つありますが、機能は等価です。
通信コネクタ	RS485 通信、保護停止の機能を担うコネクタです。
ケース	通信コネクタは2つありますが、機能は等価です。 Buildit Actuator 内で発生した熱はケースを介して放熱されます。 Buildit Actuator が高温状態になりフォルトが発生してしまう場合には、ケースに放熱器を取り付けるなどしてください。
ロータ	Buildit Actuator の出力です。 まれにロータ (回転部) よりグリスが染み出ることがあります。 品質には影響ありませんので、グリスを拭き取ってご使用ください。
取り付け穴 (ロータ)	M4、深さ 5mm の取り付け穴です。
取り付け穴 (フランジ)	M4、深さ 8mm の取り付け穴です。
電源 LED	電源が供給されていると緑色に点灯します。
状態 LED	Buildit Actuator の状態に応じて点灯します。

5.2.1 状態 LED について

Buildit Actuator は複数の状態を持ち、下表のように状態に応じて状態 LED の振る舞いが変わります。

状態	状態 LED	モータ	ブレーキ
ブレーキ保持状態	赤色が点灯、緑色が消灯	非通電	保持
ブレーキ解除状態	赤と緑が交互に点滅	非通電	解除
レディ状態	赤色が消灯、緑色が点滅	制御によって停止	解除
電流制御状態	赤色が消灯、緑色が点灯	電流制御	解除
速度制御状態	赤色が消灯、緑色が点灯	速度制御	解除
位置制御状態	赤色が消灯、緑色が点灯	位置制御	解除
保護停止減速状態	赤色が点滅、緑色が点灯	減速中 ¹	解除
保護停止状態	赤色が点滅、緑色が点灯	制御によって停止	解除
フォルト (ブレーキ保持) 状態	赤色が点滅、緑色が消灯	非通電	保持
フォルト (ブレーキ解除) 状態	赤色が点滅、緑色が消灯	非通電	解除
システムフォルト状態	赤色、緑色共に点灯	非通電	保持

5.3 Buildit の状態遷移

Buildit Actuator は 24V 電源を供給されて起動するとブレーキ保持状態となります。その後の状態遷移は fig. 3 に従います。

¹モータが、制御によって回転を停止しようとしている状態です。

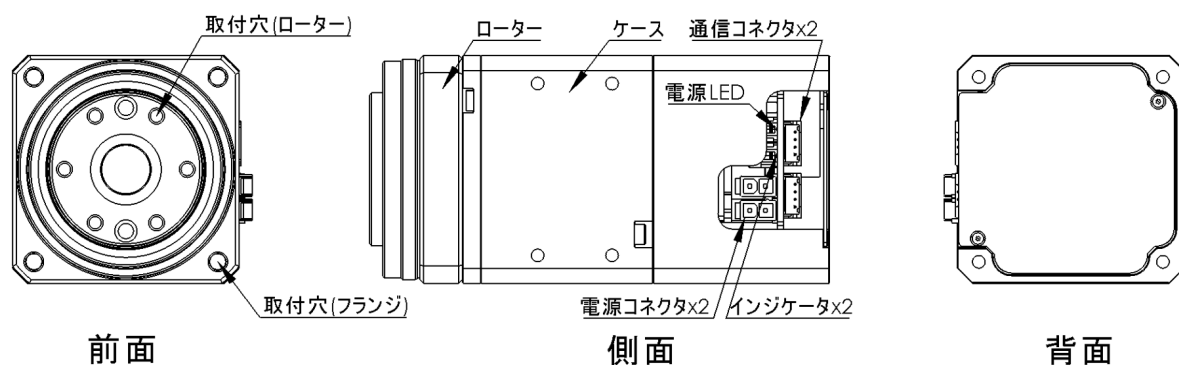


Figure 2: 各部の名称

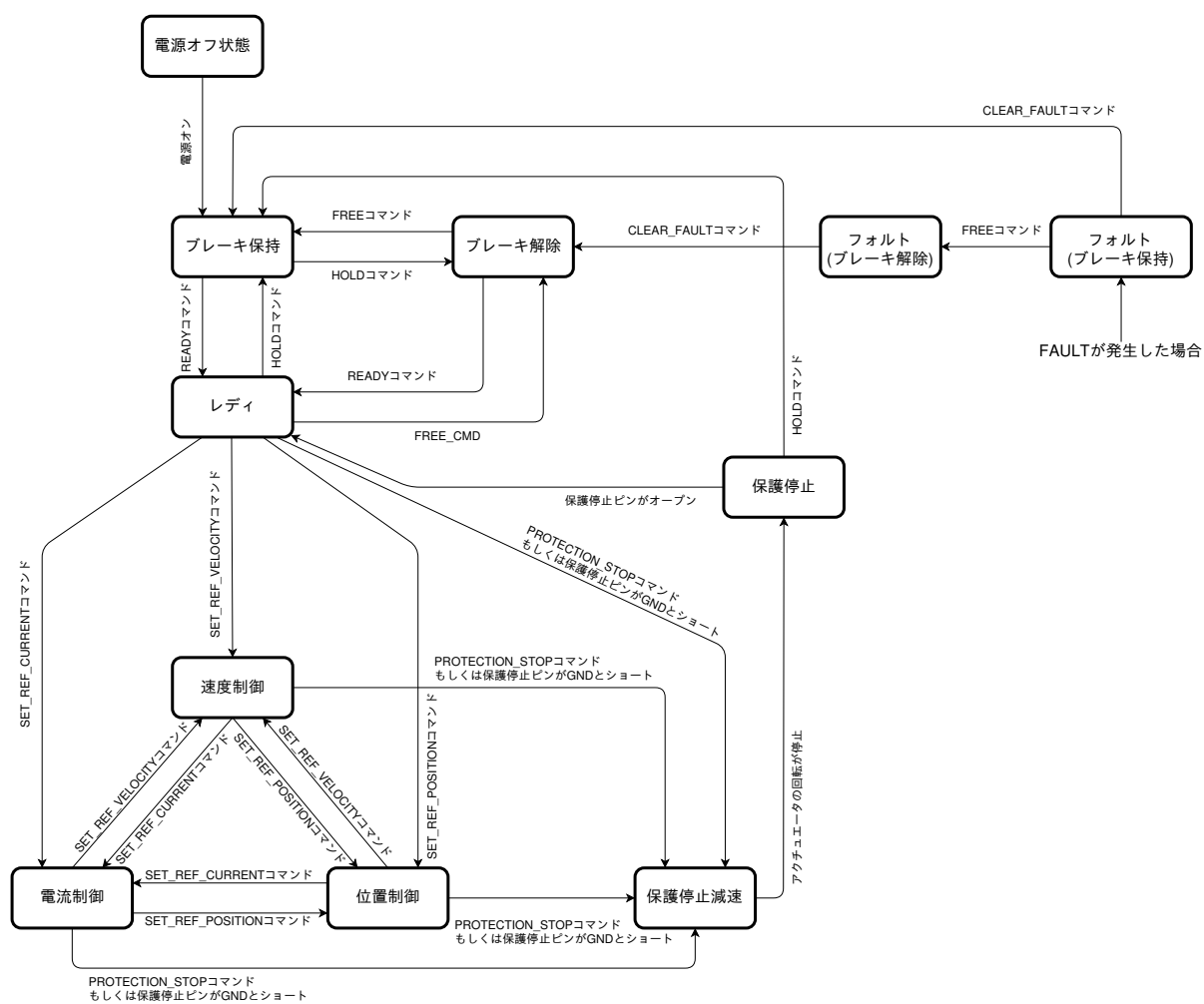


Figure 3: 状態遷移図

5.4 ロータの座標系について

Buildit Actuator のロータの回転方向は向かって反時計回りが正、時計回りが負です。位置制御、速度制御、電流制御指令値及びセンサ値の符号もこれに従います。また、ゼロ点の位置はユーザーが設定することが出来ます (PARAM_ID_POSITION_OFFSET パラメータを参照)。

5.5 電池寿命について

電源オフ時にもロータ回転数を保持するため、Buildit Actuator には電池が内蔵されています。電池の寿命は5年ですが、使用環境によって変化します。

6 セットアップ手順

本章は、Buildit Actuator を PC と接続し、Python パッケージを用いて速度制御や位置制御を行うまでのチュートリアルです。

本書の手順を実施するには、以下の条件を満たす Windows PC もしくは Ubuntu PC が必要です。

- USB ポートを持つこと
- Python3 がインストールされていること (Python3.6 以上を推奨)
- インターネットに接続されていること

6.1 Python パッケージのインストール

6.1.1 Windows

PowerShell やコマンドプロンプト (以下、ターミナル) から以下のコマンドを実行します。

```
$ pip3 install pybuildit2
```

6.1.2 Ubuntu

ターミナルから以下のコマンドを実行します。

```
$ sudo apt-get -y install python3-tk  
$ pip3 install pybuildit2
```

6.2 PC、電源の接続

各機器は、以下の順で接続してください。特に、電源プラグのコンセントへの挿入は最後に行ってください。コンセント差し込み後に電源と Buildit Actuator を接続するとスパークが発生し、接点の劣化や故障の原因となります。

1. 通信ケーブルを PC の USB ポートに接続します
2. 通信ケーブルを Buildit Actuator に接続します
3. Buildit Actuator に終端抵抗を接続します
4. 電源を Buildit Actuator に接続します
5. 電源のプラグをコンセントに差し込みます

6.3 Buildit Actuator を回す

6.3.1 Windows

ターミナルから以下のコマンドを実行します。COM ポートの名称はご使用の環境に合わせて修正してください。

```
$ python3
>>> from pybuildit2 import *
>>> buildit = Buildit(port="COM8", timeout_ms=3000) #for Win
>>> deviceId = 1
>>> qss = buildit.query_servo_status(deviceId)
>>> print("position[deg]: ", qss.position(unit="deg"))
>>> print("velocity[rpm]: ", qss.velocity(unit="rpm"))
>>> print("temperature[°C]: ", qss.temperature())
>>> print("state: ", buildit.last_mcp_status().str_state())
>>> buildit.force_ready(deviceId)
>>> buildit.set_ref_velocity(deviceId, 40, unit="rpm")
>>> buildit.set_ref_position(deviceId, 180, unit="deg")

buildit.set_ref_velocity(deviceId, 40, unit="rpm") を実行すると、40 回転/分 (RPM) で定速回転します。

buildit.set_ref_position(deviceId, 180, unit="deg") を実行すると、180 度の位置に回転し、その角度を維持します。
```

6.3.2 Ubuntu

ターミナルから以下のコマンドを実行します。シリアルポートの名称はご使用の環境に合わせて修正してください。

```
$ python3
>>> from pybuildit2 import *
>>> buildit = Buildit(port="/dev/ttyUSB0", timeout_ms=3000) #for Linux
>>> deviceId = 1
>>> qss = buildit.query_servo_status(deviceId)
>>> print("position[deg]: ", qss.position(unit="deg"))
>>> print("velocity[rpm]: ", qss.velocity(unit="rpm"))
>>> print("temperature[°C]: ", qss.temperature())
>>> print("state: ", buildit.last_mcp_status().str_state())
>>> buildit.force_ready(deviceId)
>>> buildit.set_ref_velocity(deviceId, 40, unit="rpm")
>>> buildit.set_ref_position(deviceId, 180, unit="deg")

buildit.set_ref_velocity(deviceId, 40, unit="rpm") を実行すると、40 回転/分 (RPM) で定速回転します。

buildit.set_ref_position(deviceId, 180, unit="deg") を実行すると、180 度の位置に回転し、その角度を維持します。
```

6.3.3 GUI ツールを用いた操作

Python パッケージ (pybuildit2) をインストールすると、同時に GUI ツール (builditctl-gui) もインストールされます。ターミナルから以下のコマンドを実行すると、builditctl-gui(fig. 4) が起動します。

```
$ builditctl-gui
```

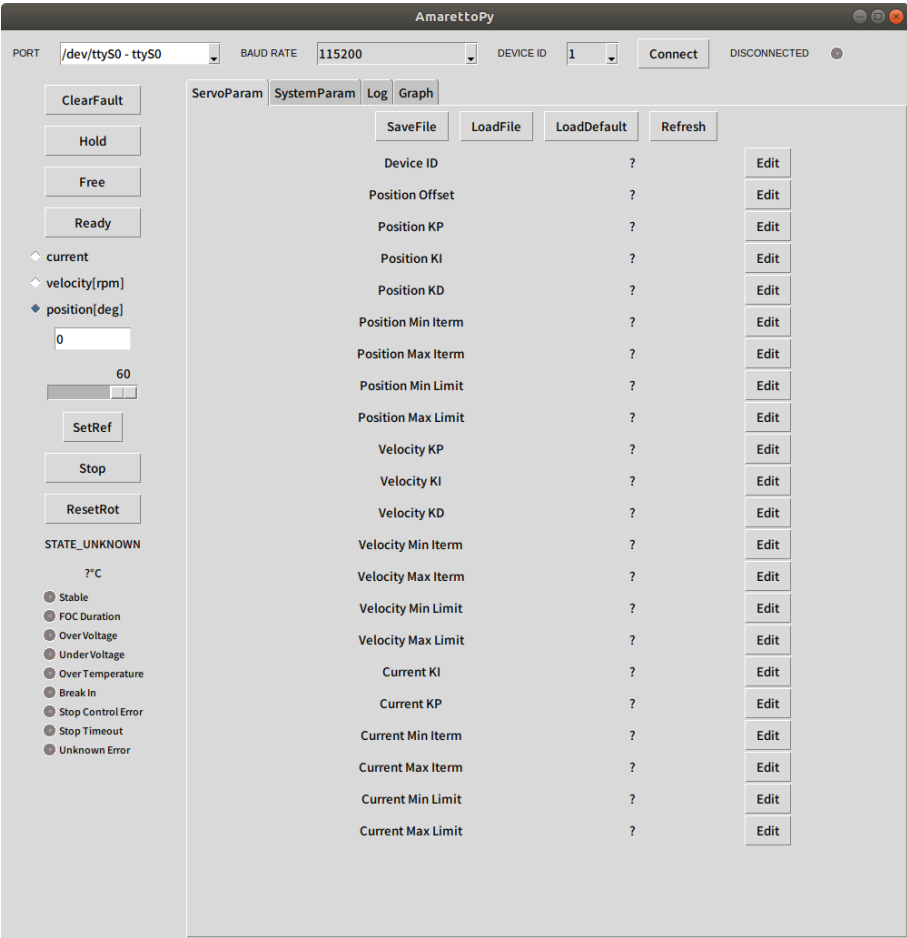


Figure 4: builditctl-gui

Buildit Actuator との通信を開始するには、ウィンドウ上部で、ポート、デバイス ID を設定し、ボーレートを 115200 に設定して Connect ボタンを押下します。

通信が開始されると、電流、速度、角度のグラフが表示され、ウィンドウの左下に Buildit Actuator の状態やフォルトの有無、内部の温度が表示されます。

フォルトが発生していないことを確認し²、Ready ボタンでレディ状態に遷移します。次に、ラジオボタンから velocity[rpm] を選択し、その下のテキストボックスに 10 と入力します。SetRef ボタンを押すと、指令値 10 回転/分 (RPM) で回転します。

停止する場合は Stop ボタンを押します。正常に停止した場合はレディ状態に遷移します。電源をオフする場合は、Hold ボタンを押しモータへの通電を止めてから、Buildit Actuator への電源供給を止めます。

Buildit Actuator の制御パラメータは中央の「ServoParam」タブから変更することができます。

²フォルトが発生している場合、ClearFault ボタンを押下してから続きの操作を行ってください。

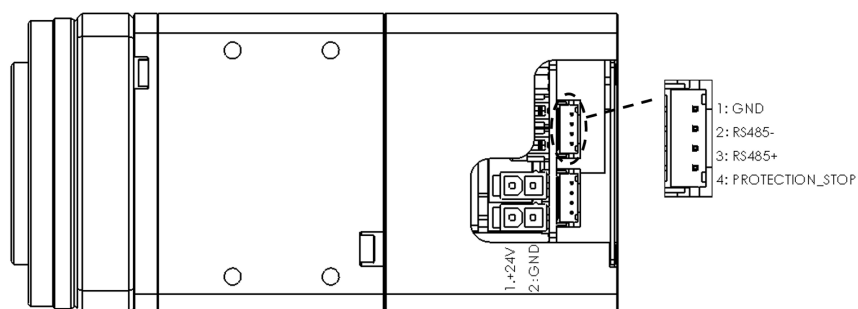


Figure 5: コネクタのピン配置

7 電氣的インターフェイス

Buildit Actuator には 2 種類 4 つのコネクタがあります。以下では、それぞれのコネクタの機能を説明します。

7.1 電源コネクタ

Buildit Actuator には 2 つの電源コネクタがあります。2 つの電源コネクタの機能は等価で、どちらを使用してもかまいません。

電源ケーブルを作製する際は、以下の部品、またはその相当品をご使用ください。

部品	型番	メーカー
ハウジング	39012020(5557-02R)	Molex
コンタクトピン	39000039(5556TL)	Molex

電源コネクタのピン配置 (fig. 5) は以下のとおりです。

ピン番号	機能	名称
1	+24V	+24V 電源ピン
2	GND	GND ピン

7.2 通信コネクタ

通信コネクタは、以下の2つの機能を有します。

- RS485 通信
- 保護停止

Buildit Actuator には2つの通信コネクタがあります。2つの通信コネクタの機能は等価で、どちらを使用してもかまいません。

通信ケーブルを作製する際は、以下の部品、またはその相当品をご使用ください。

部品	型番	メーカー
ハウジング	PHR-4	日本圧着端子製造
コンタクトピン	SPH-002T-P0.5L	日本圧着端子製造

通信コネクタのピン配置 (fig. 5 は以下のとおりです。

ピン番号	機能	名称
1	GND	GND ピン
2	RS485-	RS485-ピン
3	RS485+	RS485+ピン
4	PROTECTION_STOP	保護停止ピン

7.3 デイジーチェーン接続

電源ケーブルと通信ケーブルは、複数の Buildit Actuator に対してデイジーチェーン接続することができます。Buildit Actuator をデイジーチェーン接続するときは、各コネクタの1ピン同士を接続、2ピン同士を接続... というようにストレートケーブルで接続してください。

7.4 RS485 の終端抵抗

fig. 6 のように、RS485 通信線の両端には終端抵抗を取り付けてください。

7.5 保護停止

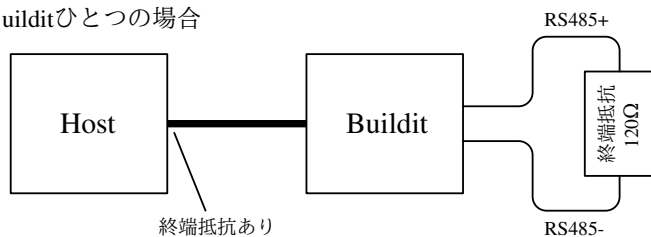
通信コネクタの 4pin(PROTECTION_STOP) を 1pin(GND) とショートさせることで³Buildit Actuator を保護停止状態に遷移することができます。また、保護停止状態のとき 4pin をオープンにすることで、READY 状態に遷移することができます (4pin がオープンの状態で保護停止状態に遷移すると、即座に READY 状態に遷移します)。

保護停止状態では HOLD_CMD 以外のコマンドは受け付けません。

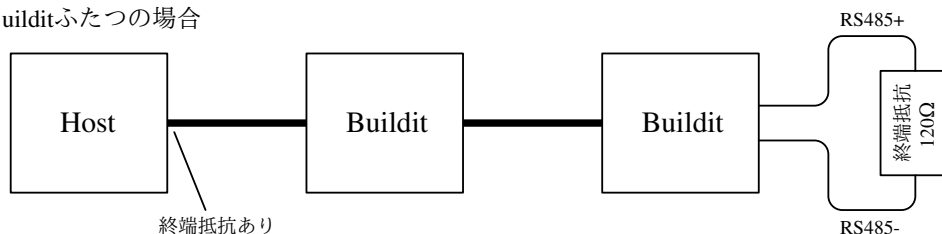
保護停止状態では、Buildit Actuator の回転は制御によって停止します。ブレーキによる保持は行いません。

³パラメーター「PARAM_ID_PROTECTION_STOP_PIN_MODE」により、4pin をオープンにすることで保護停止 ON にするよう切り替えることも可能です。

Builditひとつの場合



Builditふたつの場合



Buildit複数の場合

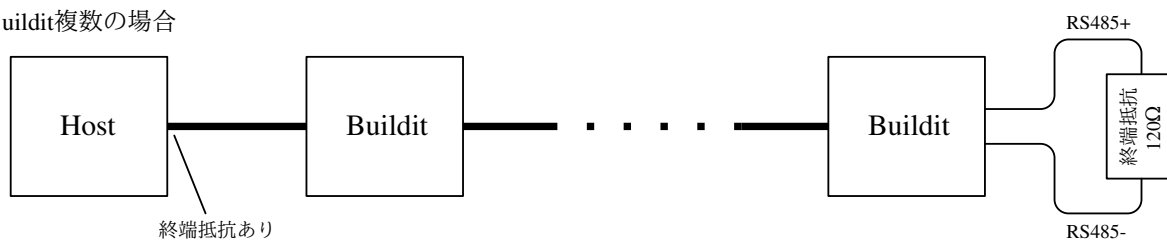


Figure 6: RS485 の終端抵抗

7.6 電源について

Buildit Actuator の定格動作範囲内での消費電流は約 3A(約 70W) です。ですが、起動時や急激な減速時にはより多くの電力を消費しますので、余力を持った電源を用意してください。また、回転が止まったり逆転する際には逆起電力が発生し、電源電圧が上昇することがあります。

8 通信プロトコル仕様

本章では Buildit Actuator のモーター制御用通信プロトコル (以下 MCP) の仕様について述べます。

8.1 概要

- Buildit Actuator は PC やマイコンといったホストから RS485 シリアル通信を行って制御されます。
- これらの通信はサーバクライアント方式で行われます。つまり、Buildit Actuator は外部からのコマンドを受け取り、そのコマンドに対して応答を行います。
- コマンドと応答に用いるデータを合わせてメッセージと呼びます。

8.2 シリアル通信の設定

RS485 によるシリアル通信の設定は以下の通りです。

名前	値
データ長	8bit
ストップビット	1bit
パリティ	なし
ボーレート	115200bps

8.3 エンディアン

バイトオーダーはリトルエンディアンです。例えば、16 ビットのデータにおいて、十進数の 1000(0x03E8) は 0xE8 0x03 というバイト列になります。32 ビットのデータにおいて十進数の 99999999(十六進数の 0x05F5E0FF) は 0xFF 0xE0 0xF5 0x05 というバイト列になります。

8.4 状態の種別

起動中の Buildit Actuator は複数の状態を持ち、状態の種別を表す為に状態 ID を用います。これらの状態は各コマンドの応答メッセージで確認することが出来ます。

名前	値	説明
STATE_HOLD	0	ブレーキ保持状態
STATE_FREE	1	ブレーキ解除状態
STATE_READY	2	レディ状態
STATE_CURRENT_SERVO	3	電流制御状態
STATE_VELOCITY_SERVO	4	速度制御状態
STATE_POSITION_SERVO	5	位置制御状態
STATE_PROTECTION_STOPPING	12	保護停止減速状態
STATE_PROTECTION_STOP	13	保護停止状態
STATE_FAULT_FREE	14	フォルト（ブレーキ解除）状態
STATE_FAULT_HOLD	15	フォルト（ブレーキ保持）状態

Buildit Actuator は起動後、コマンドを受理出来るようになるとブレーキ保持状態になります。

8.5 異常

起動中の Buildit Actuator は異常を検知すると、対応するフォルトフラグをセットし、フォルト (ブレーキ保持) 状態に遷移します。これらのフラグはQUERY_SERVO_STATUS コマンドの応答メッセージで確認することが出来ます。

フォルト種別	フォルト ID	説明
SERVO_FAULT_FOC_DURATION	0x0001	制御計算タイムアウト
SERVO_FAULT_OVER_VOLT	0x0002	高電圧状態
SERVO_FAULT_UNDER_VOLT	0x0004	低電圧状態
SERVO_FAULT_OVER_TEMP	0x0008	高温状態 (※)
SERVO_FAULT_OVER_POSITION_LIMIT	0x0010	位置範囲制限外
SERVO_FAULT_BREAK_IN	0x0040	過負荷・高電流
SERVO_FAULT_STOP_CONTROL_ERROR	0x0100	停止制御位置偏差過大
SERVO_FAULT_STOP_TIMEOUT	0x0200	保護停止タイムアウト
SERVO_FAULT_EXTERNAL	0x0800	コマンドトリガフォルト (フォルトコマンド受理時に発

※温度異常フォルトは温度センサの値が 85℃ 以上になったら発生し、温度センサの値が 75℃ 未満になるまでこのフォルトをリセットすることはできません。

8.6 メッセージの種別

MCP ではメッセージの種別を表す為にメッセージタイプ ID を用います。成功応答の場合、対応するコマンド ID に 128 を加えたものが設定されます。そうでない場合は失敗応答 (NACK) が用いられます。プロトコルで以下に挙げるメッセージタイプ ID 以外は予約されています。指定しないようにして下さい。

メッセージ種別	メッセージタイプ ID	説明
QUERY_SERVO_STATUS_CMD	0x01	サーボ関連のセンサ値を取得
READY_CMD	0x10	STATE_READY 状態へ遷移
FREE_CMD	0x11	STATE_FREE 状態へ遷移
HOLD_CMD	0x12	STATE_HOLD 状態へ遷移
CLEAR_FAULT_CMD	0x13	フォルト情報クリア
PROTECTION_STOP_CMD	0x14	保護停止
SET_REF_CURRENT_CMD	0x20	電流指令値設定
GET_REF_CURRENT_CMD	0x21	電流指令値取得
SET_REF_VELOCITY_CMD	0x22	速度指令値設定
GET_REF_VELOCITY_CMD	0x23	速度指令値取得
SET_REF_POSITION_CMD	0x24	位置指令値設定
GET_REF_POSITION_CMD	0x25	位置指令値取得
SET_REF_POSITION_WITH_VELOCITY_CMD	0x26	位置指令値設定 (速度指定)
SET_PARAM_CMD	0x30	パラメータ設定
GET_PARAM_CMD	0x31	パラメータ取得

メッセージ種別	メッセージタイプ ID	説明
RESET_ROTATION_CMD	0x32	回転オフセットの初期化
NACK	0xff	失敗応答

8.7 メッセージの共通フォーマット

MCP で用いるメッセージのフォーマットを以下に示します。

width(bits):	24	8	8	8	16	0...
	+-----+-----+-----+-----+-----+-----+					
			device	Message	Payload	
	Header	CRC	ID	Type	Size	Payload
			ID			
	+-----+-----+-----+-----+-----+-----+					

- Header
 - メッセージの先頭を表すマジックナンバーであり、1byte 目は 0xab 2byte 目は 0xcc 3byte 目は 0xba となります。
- CRC
 - メッセージの 5byte 目以降の CRC(初期値 0 で CRC-8-ATM 方式で符号化したもの)。
- device ID
 - コマンドの場合は送り先となるデバイスの ID。
 - 応答の場合は送り元となるデバイスの ID。
 - Buildit Actuator は自分以外へのコマンドについては無視します。
 - Buildit Actuator は応答については無視します。
 - 指定できる ID は、1~127。 0 および 128~255 までは使用不可 (Reserved) となり、Buildit Actuator はこれらの値が指定されても無視します。
- Message Type ID
 - メッセージタイプ ID。未定義の値だった場合、失敗応答 (MCP_INVALID_MSG_TYPE) となります。
 - 応答メッセージの場合は応答のトリガとなったコマンドのメッセージタイプ ID の最上位ビットが 1 に変化したものになります。
- Payload Size
 - Payloadのバイト数。メッセージタイプに対して不正な値だった場合、失敗応答 (MCP_INVALID_COMMAND_PAYLOAD_SIZE) となります。
- Payload
 - コマンドに紐づくパラメータ等を格納する領域。

8.7.1 応答メッセージの共通フォーマット

応答メッセージのペイロードの先頭 2byte は以下のフォーマットとなります。これを以後 MCP Status と呼びます。

```
width(bits):   4   1   3       8
               +-----+-----+
               | S | U | R |   R   |
               | T | N | V |   V   |
               |   | | 1 |   2   |
               +-----+-----+
               <-- MCP Status  -->
```

- ST
 - 状態 ID。
- UN
 - 未通知エラーフラグ。未通知エラーがあれば 1、そうでなければ 0。(無効なメッセージを受信するといったような NACK を返せないエラーが発生したことを表します)
- RV1
 - 予約領域。
- RV2
 - 予約領域。

8.7.2 CRC-8-ATM 計算ルーチンの例

MCP で用いる CRC-8-ATM の計算は、C 言語であれば以下のようなコードで実行できます。

```
static uint8_t tbl[256] = {
0x00, 0x07, 0x0e, 0x09, 0x1c, 0x1b, 0x12, 0x15,
0x38, 0x3f, 0x36, 0x31, 0x24, 0x23, 0x2a, 0x2d,
0x70, 0x77, 0x7e, 0x79, 0x6c, 0x6b, 0x62, 0x65,
0x48, 0x4f, 0x46, 0x41, 0x54, 0x53, 0x5a, 0x5d,
0xe0, 0xe7, 0xee, 0xe9, 0xfc, 0xfb, 0xf2, 0xf5,
0xd8, 0xdf, 0xd6, 0xd1, 0xc4, 0xc3, 0xca, 0xcd,
0x90, 0x97, 0x9e, 0x99, 0x8c, 0x8b, 0x82, 0x85,
0xa8, 0xaf, 0xa6, 0xa1, 0xb4, 0xb3, 0xba, 0xbd,
0xc7, 0xc0, 0xc9, 0xce, 0xdb, 0xdc, 0xd5, 0xd2,
0xff, 0xf8, 0xf1, 0xf6, 0xe3, 0xe4, 0xed, 0xea,
0xb7, 0xb0, 0xb9, 0xbe, 0xab, 0xac, 0xa5, 0xa2,
0x8f, 0x88, 0x81, 0x86, 0x93, 0x94, 0x9d, 0x9a,
0x27, 0x20, 0x29, 0x2e, 0x3b, 0x3c, 0x35, 0x32,
0x1f, 0x18, 0x11, 0x16, 0x03, 0x04, 0x0d, 0x0a,
0x57, 0x50, 0x59, 0x5e, 0x4b, 0x4c, 0x45, 0x42,
0x6f, 0x68, 0x61, 0x66, 0x73, 0x74, 0x7d, 0x7a,
0x89, 0x8e, 0x87, 0x80, 0x95, 0x92, 0x9b, 0x9c,
0xb1, 0xb6, 0xbf, 0xb8, 0xad, 0xaa, 0xa3, 0xa4,
0xf9, 0xfe, 0xf7, 0xf0, 0xe5, 0xe2, 0xeb, 0xec,
0xc1, 0xc6, 0xcf, 0xc8, 0xdd, 0xda, 0xd3, 0xd4,
0x69, 0x6e, 0x67, 0x60, 0x75, 0x72, 0x7b, 0x7c,
0x51, 0x56, 0x5f, 0x58, 0x4d, 0x4a, 0x43, 0x44,
0x19, 0x1e, 0x17, 0x10, 0x05, 0x02, 0x0b, 0x0c,
0x21, 0x26, 0x2f, 0x28, 0x3d, 0x3a, 0x33, 0x34,
0x4e, 0x49, 0x40, 0x47, 0x52, 0x55, 0x5c, 0x5b,
0x76, 0x71, 0x78, 0x7f, 0x6a, 0x6d, 0x64, 0x63,
0x3e, 0x39, 0x30, 0x37, 0x22, 0x25, 0x2c, 0x2b,
0x06, 0x01, 0x08, 0x0f, 0x1a, 0x1d, 0x14, 0x13,
0xae, 0xa9, 0xa0, 0xa7, 0xb2, 0xb5, 0xbc, 0xbb,
0x96, 0x91, 0x98, 0x9f, 0x8a, 0x8d, 0x84, 0x83,
0xde, 0xd9, 0xd0, 0xd7, 0xc2, 0xc5, 0xcc, 0xcb,
0xe6, 0xe1, 0xe8, 0xef, 0xfa, 0xfd, 0xf4, 0xf3
};

uint8_t CRCCalculate(const uint8_t* buf, uint32_t size)
{
    uint8_t c = 0x00;
    for (int i=0; i< size; i++) {
        c = tbl[(c ^ buf[i])];
    }
    return c;
}
```

8.8 無効なメッセージについて

Buildit Actuator は受信データを一度受信バッファに格納します。バッファの中のバイト列はフォーマットに従ってパースされ、使い終わったデータは受信バッファから破棄されます。

この際、以下の条件を満たす場合、Buildit Actuator は無効なメッセージと判断します。

- CRC が一致しない
 - メッセージ全体を受信バッファから破棄します。
- 受信に 1 秒以上かかる
 - 受信開始から受信を完了せず 1 秒を超えた場合、無効なメッセージと判断し、その時点での受信済データは受信バッファから破棄されます。
- Header がマジックナンバーと一致しない もしくは Payload Size が 248 より大きい
 - 先頭 8byte 分のデータをスキャンし次のマジックナンバー開始位置を探します。見つかった場合、その位置までのデータを破棄し、見つからなければ 8byte 全て破棄します。

Buildit Actuator は無効なメッセージと判断した場合、失敗応答を返しません。後続のデータを新しいメッセージの開始として処理します。起動もしくは最後の応答メッセージ送信からその直後の応答メッセージを返すまでの間に無効なメッセージを受信した場合、直後の応答メッセージの UN ビットは 1 になり、受信しなかった場合は 0 になります。

Buildit Actuator の受信処理速度をこえてデータを送り続けると Buildit Actuator の受信バッファが溢れてシステムフォルトが発生する場合があります。

8.9 通信についての注意事項

- Buildit Actuator はデイジーチェーン接続されることを想定しています。この通信路では 2 台以上の Buildit Actuator またはホストが同時に送信しないように注意する必要があります。
- Buildit Actuator に対してコマンドを送信してから応答が返るまでの時間 (レイテンシ) についての保証はありません。

8.10 パラメータについて

プロトコルで以下に挙げるパラメータ ID 以外は予約されています。指定しないようにして下さい。

パラメータ	ID	符号	ビット幅	説明
PARAM_ID_CURRENT_MAX_LIMIT	0x14	あり	16bit	電流制限範囲の最大値
PARAM_ID_CURRENT_MIN_LIMIT	0x15	あり	16bit	電流制限範囲の最小値
PARAM_ID_VELOCITY_KP	0x20	あり	16bit	速度制御の P ゲイン
PARAM_ID_VELOCITY_KI	0x21	あり	16bit	速度制御の I ゲイン
PARAM_ID_VELOCITY_KD	0x22	あり	16bit	速度制御の D ゲイン
PARAM_ID_VELOCITY_MAX_ITERM	0x23	あり	32bit	速度制御の積分項の最大値
PARAM_ID_VELOCITY_MIN_ITERM	0x24	あり	32bit	速度制御の積分項の最小値
PARAM_ID_VELOCITY_MAX_LIMIT	0x25	あり	16bit	速度制限範囲の最大値
PARAM_ID_VELOCITY_MIN_LIMIT	0x26	あり	16bit	速度制限範囲の最小値
PARAM_ID_POSITION_KP	0x30	あり	16bit	位置制御の P ゲイン
PARAM_ID_POSITION_KI	0x31	あり	16bit	位置制御の I ゲイン
PARAM_ID_POSITION_KD	0x32	あり	16bit	位置制御の D ゲイン
PARAM_ID_POSITION_MAX_ITERM	0x33	あり	32bit	位置制御の積分項の最大値
PARAM_ID_POSITION_MIN_ITERM	0x34	あり	32bit	位置制御の積分項の最小値
PARAM_ID_POSITION_MAX_LIMIT	0x35	あり	32bit	位置制限範囲の最大値
PARAM_ID_POSITION_MIN_LIMIT	0x36	あり	32bit	位置制限範囲の最小値
PARAM_ID_POSITION_MAX_SOFT_LIMIT	0x37	あり	32bit	ソフトウェアリミット範囲の最大値
PARAM_ID_POSITION_MIN_SOFT_LIMIT	0x38	あり	32bit	ソフトウェアリミット範囲の最小値
PARAM_ID_POSITION_OFFSET	0x3a	あり	16bit	位置センサの 0 点オフセット
PARAM_ID_DEVICE_ID	0x80	なし	8bit	デバイス ID (1~127)
PARAM_ID_FIRMWARE_VERSION (※)	0x81	なし	128bit	バージョン情報
PARAM_ID_POWER_ON_TIME (※)	0x82	なし	32bit	累計電源 ON 時間 [秒]
PARAM_ID_CALIBRATION (※)	0xa0	なし	16bit	磁石アライメント情報
PARAM_ID_POSITION_SYS_OFFSET (※)	0xa1	なし	16bit	組付補正オフセット
PARAM_ID_PROTECTION_STOP_PIN_MODE	0xa2	なし	8bit	保護停止ピン極性

- 各積分項は対応する最大値を上回った場合は最大値に制限され、最大値以下でかつ最小値を下回った場合は最小値に制限されます。
- 各範囲制限の最大値はその時点での対応する範囲制限の最小値以上である場合のみ有効です。
- 位置センサの 0 点オフセットの値を変更することで、Buildit Actuator の 0 点を変更することができます。
- 制御パラメータのうちゲインや積分項の最大・最小値はデフォルト値が推奨です。不適切な変更は意図せぬ動作を引き起こす恐れがあります。
- (※) これらのパラメータは読み込み専用で変更出来ません。

8.10.1 出荷時のパラメータの値

パラメータ	出荷時の値
PARAM_ID_CURRENT_MAX_LIMIT	5000
PARAM_ID_CURRENT_MIN_LIMIT	-5000
PARAM_ID_VELOCITY_KP	8000
PARAM_ID_VELOCITY_KI	16000
PARAM_ID_VELOCITY_KD	0
PARAM_ID_VELOCITY_MAX_ITERM	65536000
PARAM_ID_VELOCITY_MIN_ITERM	-65536000
PARAM_ID_VELOCITY_MAX_LIMIT	5000
PARAM_ID_VELOCITY_MIN_LIMIT	-5000
PARAM_ID_POSITION_KP	160
PARAM_ID_POSITION_KI	0
PARAM_ID_POSITION_KD	800
PARAM_ID_POSITION_MAX_ITERM	98304000
PARAM_ID_POSITION_MIN_ITERM	-98304000
PARAM_ID_POSITION_MAX_LIMIT	2147483647
PARAM_ID_POSITION_MIN_LIMIT	-2147483648
PARAM_ID_POSITION_MAX_SOFT_LIMIT	2147483647
PARAM_ID_POSITION_MIN_SOFT_LIMIT	-2147483648
PARAM_ID_POSITION_OFFSET	0
PARAM_ID_DEVICE_ID	1
PARAM_ID_PROTECTION_STOP_PIN_MODE	0

8.10.2 位置に関するパラメータの操作の制限

以下のパラメータについては STATE_READY 状態、STATE_PROTECTION_STOP 状態、STATE_PROTECTION_STOPPING 状態では変更することが出来ません。

- PARAM_ID_POSITION_MAX_LIMIT
- PARAM_ID_POSITION_MIN_LIMIT
- PARAM_ID_POSITION_OFFSET

8.11 範囲制限機能について

Buildit Actuator の制御系は、外側から位置制御系、速度制御系、電流制御系の三重ループ構造になっています。つまり、位置制御の出力は速度指令値となり、速度制御の出力は電流指令値となります。各制御系では、対応するセンサ値がその系の指令値に追従するようフィードバック制御が行われます。Buildit Actuator は各制御モードでユーザーから与えられた指令値をその制御系の入力として使用します。例えば、位置制御を行う場合、ユーザーから与えられた位置制御指令値が位置制御系の指令値として利用されます。ですが、それぞれの制御モードでリミットオーバー時の振る舞いが異なります。

8.11.1 位置範囲制限外の振る舞い

位置範囲制限外になった場合、全ての制御モードでフォルトになります。

8.11.2 速度範囲制限外の振る舞い

位置制御中・速度制御・保護停止中において、制御に使われる速度指令値は `PARAM_ID_VELOCITY_MIN_LIMIT` と `PARAM_ID_VELOCITY_MAX_LIMIT` の間の値になるようにリミットされます。この時、`QueryServoStatus` コマンドで得られる指令値の値はリミットされたものではなく、ユーザーが設定した値を返すことに注意して下さい。

実際の速度が速度範囲制限外になってとしても、何も起こりません。

8.11.3 電流範囲制限外の振る舞い

全ての制御モードにおいて、制御に使われる電流指令値は `PARAM_ID_CURRENT_MIN_LIMIT` と `PARAM_ID_CURRENT_MAX_LIMIT` の間の値になるようにリミットされます。この時、`QueryServoStatus` コマンドで得られる指令値の値はリミットされたものではなく、ユーザーが設定した値を返すことに注意して下さい。

8.11.4 ソフトウェアリミットによる位置範囲制限

位置をある範囲で制限したいが範囲外になった際にエラーではなく停止としたい場合にはソフトウェアリミット機能を使用してください。位置範囲制限の範囲より内側にソフトウェアリミット範囲を設定することでソフトウェアリミット機能を使用することができます。

移動中にソフトウェアリミット範囲をオーバーすると保護停止状態となり、停止後は自動的に Ready 状態となります。

ソフトウェアリミット範囲外にいる状態では、移動開始時に制御モード毎の制限がかかります。

- ・ 電流制御: 移動指令を受け付けません。
- ・ 速度制御: リミット範囲内方向の指令のみを受け付けます。上限を超えていけばーを、下限を越えていけば+の指令を受け付けます。
- ・ 位置制御: リミット範囲内への位置指令のみ受け付けます。

8.12 保護停止ピン極性の切り替えについて

通信コネクタの 4pin を使用して外部から保護停止する場合の論理を選択することができます。`PARAM_ID_PROTECTION_STOP_PIN_MODE` を変更することで以下のように設定できます。

- ・ 0x00: 4pin と GND のショートで保護停止 ON (デフォルト)
- ・ 0x01: 4pin をオープンにすると保護停止 ON
- ・ 0x10: 保護停止ピン無効化

8.13 CLEAR_FAULT_CMD

8.13.1 概要

フォルトフラグをリセットします。STATE_FAULT_FREE 状態であれば、STATE_FREE 状態へ遷移し、STATE_FAULT_HOLD 状態であれば、STATE_HOLD 状態へ遷移します。

8.13.2 事前条件

- Buildit Actuator の状態が以下のいずれかであること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。
 - STATE_FAULT_HOLD
 - STATE_FAULT_FREE
- 低電圧状態でないこと。
- SERVO_FAULT_OVER_TEMP が発生している場合、温度センサの値が 75 °C 以下であること。

8.13.3 コマンドのペイロードフォーマット

ペイロードはありません (ペイロードサイズは 0)。

8.13.4 例

※バイト列は 16 進数表記

Header	CRC	ID	Type ID	Size	Payload
ab cc ba	09	01	13	00 00	なし

8.13.5 成功応答メッセージのペイロードフォーマット

width (bits)	16
	+
	MCP
	+

- MCP
 - 「共通フォーマット」を参照。

8.14 FAULT

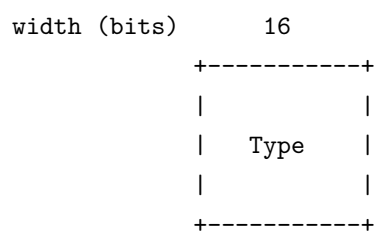
8.14.1 概要

フォルトを発生させます。

8.14.2 事前条件

- Buildit Actuator が通信可能な状態であること。

8.14.3 コマンドのペイロードフォーマット



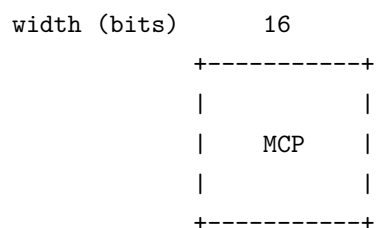
- Type
 - 0 であればコマンドトリガフォルトを発生させます。Buildit Actuator は STATE_FAULT_HOLD 状態に遷移します
 - 1 であればシステムフォルトを発生させます（再起動しない限り通信が出来なくなります）

8.14.4 例

※バイト列は 16 進数表記

					device		Message		Payload		
	Header		CRC		ID		Type ID		Size		Payload
	-----		-----		-----		-----		-----		-----
	ab cc ba		a7		01		3d		02 00		00 00

8.14.5 成功応答メッセージのペイロードフォーマット



- MCP
 - 「共通フォーマット」を参照。

8.15 FREE_CMD

8.15.1 概要

ブレーキを解除し、STATE_FREE 状態へ遷移します。

8.15.2 事前条件

- Buildit Actuator の状態が以下のいずれかであること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。
 - STATE_FREE
 - STATE_READY
 - STATE_HOLD
 - STATE_FAULT_HOLD
 - STATE_FAULT_FREE
- 低電圧状態でないこと。

8.15.3 コマンドのペイロードフォーマット

ペイロードはありません (ペイロードサイズは 0)。

8.15.4 例

※バイト列は 16 進数表記

		device	Message	Payload	
Header	CRC	ID	Type ID	Size	Payload
-----	-----	-----	-----	-----	-----
ab cc ba	df	01	11	00 00	なし

8.15.5 成功応答メッセージのペイロードフォーマット

width (bits)	16
	+-----+
	MCP
	+-----+

- MCP
 - 「共通フォーマット」を参照。

8.16 GET__LOG__CMD

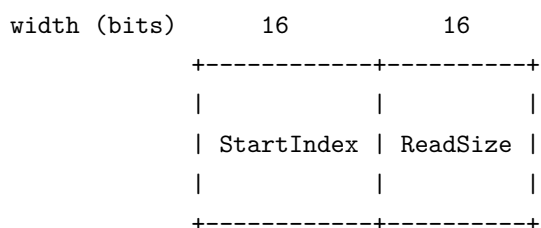
8.16.1 概要

ログを取得します。過去に起こった異常なイベントやその直前のイベントについての情報を得ることが出来ます。ただし、ログレコード件数は最大 1023 件で、古いレコードは上書きされます。

8.16.2 事前条件

- Buildit Actuator が通信可能な状態であること。

8.16.3 コマンドのペイロードフォーマット



- StartIndex
 - 取得するログレコードの開始番号。
- ReadSize
 - 取得するログレコードの個数。指定できる値は最大は 10 までです。それを超える場合は失敗応答 NACK(MCP_INVALID_COMMAND_PAYLOAD) を返します。

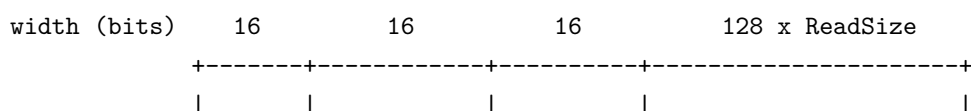
ログレコードの番号は、0 が最新のレコードを表します。(StartIndex + ReadSize) の値が GET_LOG_INFO_CMD コマンドで取得した ReadableSize を超える場合は、失敗応答 NACK(MCP_INVALID_COMMAND_PAYLOAD) を返します。

8.16.4 例

※バイト列は 16 進数表記

		device	Message	Payload	
Header	CRC	ID	Type ID	Size	Payload
-----	-----	-----	-----	-----	-----
ab cc ba	f9	01	06	04 00	00 00 0a 00

8.16.5 成功応答メッセージのペイロードフォーマット



	MCP		StartIndex		ReadSize		LogRecord x ReadSize	
+-----+-----+-----+-----+-----+								

- MCP
 - 「共通フォーマット」を参照。
- StartIndex
 - 取得したログレコードの開始番号。
- ReadSize
 - 取得したログレコードの個数。
- LogRecord x ReadSize
 - 取得した ReadSize 個分のログレコード。

8.16.6 ログレコードについて

ログレコードのフォーマットを以下に示します。

width(bits):	32	8	8	8	8	64
+-----+-----+-----+-----+-----+-----+						
	Index		Level		Group	
					Sub Group	
					Code	
						Payload
+-----+-----+-----+-----+-----+-----+						

- Index
 - ログ番号。
- Level
 - ログレベル。
- Group
 - ログの大分類。
- Sub Group
 - ログの小分類。
- Code
 - ログコード。
- Payload
 - ログに紐づくパラメータ等を格納する領域。

8.16.6.1 ログレベルについて

ログレベルは値が低い程重要です。Buildit のログ機能では起こった全てのログを保存する訳ではありません。ログレベルが 1 || 3 のイベントが起こった場合のみ、その直前の数個のイベントと合わせて記録します。

値	説明
0x01	深刻なエラー
0x02	エラー
0x03	警告
0x04	情報

8.16.6.2 ログの見方

Group	SubGroup	説明
0x01	0x01	電源 ON イベント
0x01	0x02	起動イベント
0x01	0x10	SystemFault 発生イベント
0x01	0x11	SystemFault 発生イベント
0x04	*	コマンド受信イベント。 SubGroup はメッセージタイプ ID、Code はエラー ID を表す。
0x05	*	NACK 送信イベント。 SubGroup はメッセージタイプ ID を表す。
0x10	0x01	Fault 発生イベント
0x10	0x02	保護停止発生イベント
0x14	0x02	不正なパケット受理イベント

8.17 GET LOG INFO CMD

8.17.1 概要

ログ情報を取得します。

8.17.2 事前条件

- Buildit Actuator が通信可能な状態であること。

8.17.3 コマンドのペイロードフォーマット

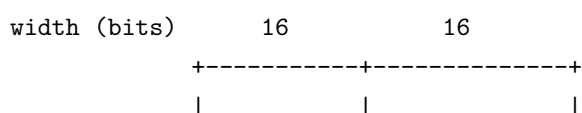
ペイロードはありません (ペイロードサイズは0)。

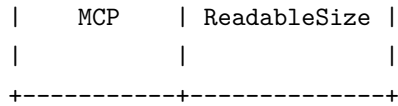
8.17.4 例

※バイト列は 16 進数表記

Header	CRC	device ID	Message Type ID	Payload Size	Payload
ab cc ba	d6	01	05	00 00	なし

8.17.5 成功応答メッセージのペイロードフォーマット





- MCP
 - 「共通フォーマット」を参照。
- ReadableSize
 - 取得可能なログレコード数。 ## GET_PARAM_CMD

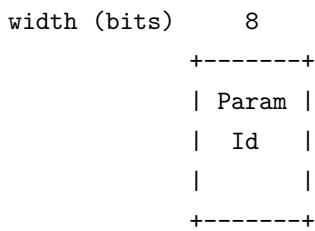
8.17.6 概要

パラメータの値を取得します。

8.17.7 事前条件

- Buildit Actuator が通信可能な状態であること。

8.17.8 コマンドのペイロードフォーマット



- Param Id
 - 取得するパラメータの ID。有効でない ID を指定した場合は失敗応答 NACK(MCP_INVALID_COMMAND_PAYLOAD) となります。
 - 本書で記載されていない有効な ID も存在します。そのような ID を使うと不安定な動作を引き起こす場合があるので使用しないようご注意ください。
 - Param Id 及び Param の詳細は「パラメータについて」を参照してください。

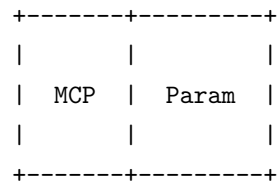
8.17.9 例

※バイト列は 16 進数表記

		device	Message	Payload	
Header	CRC	ID	Type ID	Size	Payload
ab cc ba	3f	01	31	01 00	80

8.17.10 成功応答メッセージのペイロードフォーマット

width (bits) 16 8 or 16 or 32 or 128



- MCP
 - 「共通フォーマット」を参照。
- Param
 - 取得したパラメータの値。符号の有無、ビット幅は設定するパラメータによって異なります。

Param のビット幅の詳細は「パラメータについて」を参照。

8.18 GET_REF_CURRENT_CMD

8.18.1 概要

電流制御の指令値 [mA] を取得します。最後に SET_REF_CURRENT_CMD で設定された値を返します。仮にその値が PARAM_ID_CURRENT_MAX_LIMIT 及び PARAM_ID_CURRENT_MIN_LIMIT で設定された範囲外であっても同様です。

8.18.2 事前条件

- Buildit Actuator の状態が STATE_CURRENT_SERVO であること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。

8.18.3 コマンドのペイロードフォーマット

ペイロードはありません (ペイロードサイズは 0)。

8.18.4 例

※バイト列は 16 進数表記

		device	Message	Payload	
Header	CRC	ID	Type ID	Size	Payload
-----	-----	-----	-----	-----	-----
ab cc ba	3e	01	21	00 00	なし

8.18.5 成功応答メッセージのペイロードフォーマット

width (bits)	16	16
	-----+	-----+
		Ref
	MCP	Cur
	-----+	-----+

- MCP
 - 「共通フォーマット」を参照。
- RefCur
 - 電流制御の指令値 [mA]。符号あり 16bit 整数値です。

8.19 GET_REF_POSITION_CMD

8.19.1 概要

位置制御の指令値を取得します。最後に SET_REF_POSITION_CMD で設定された値を返します。仮にその値が PARAM_ID_POSITION_MAX_LIMIT 及び PARAM_ID_POSITION_MIN_LIMIT で設定された範囲外であっても同様です。

8.19.2 事前条件

- Buildit Actuator の状態が STATE_POSITION_SERVO であること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。

8.19.3 コマンドのペイロードフォーマット

ペイロードはありません (ペイロードサイズは 0)。

8.19.4 例

※バイト列は 16 進数表記

		device	Message	Payload	
Header	CRC	ID	Type ID	Size	Payload
-----	-----	-----	-----	-----	-----
ab cc ba	95	01	25	00 00	なし

8.19.5 成功応答メッセージのペイロードフォーマット

width (bits)	16	32
	-----	-----
		Ref
	MCP	Pos
	-----	-----

- MCP
 - 「共通フォーマット」を参照。
- RefPos
 - 位置制御の指令値。符号あり 32bit 整数値です。0x10000 でエンコーダー 1 周分の移動量に相当します。

8.20 GET_REF_VELOCITY_CMD

8.20.1 概要

速度制御の指令値 [rpm/100] を取得します。最後に SET_REF_VELOCITY_CMD で設定された値を返します。仮にその値が PARAM_ID_VELOCITY_MAX_LIMIT 及び PARAM_ID_VELOCITY_MIN_LIMIT で設定された範囲外であっても同様です。

8.20.2 事前条件

- Buildit Actuator の状態が STATE_VELOCITY_SERVO であること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。

8.20.3 コマンドのペイロードフォーマット

ペイロードはありません (ペイロードサイズは 0)。

8.20.4 例

※バイト列は 16 進数表記

		device	Message	Payload	
Header	CRC	ID	Type ID	Size	Payload
-----	-----	-----	-----	-----	-----
ab cc ba	e8	01	23	00 00	なし

8.20.5 成功応答メッセージのペイロードフォーマット

width (bits)	16	16
	-----+	-----+
		Ref
	MCP	Vel
	-----+	-----+

- MCP
 - 「共通フォーマット」を参照。
- RefVel
 - 速度制御の指令値 [rpm/100]。符号あり 16bit 整数値です。

8.21 HOLD_CMD

8.21.1 概要

ブレーキを保持し、STATE_HOLD 状態へ遷移します。

8.21.2 事前条件

- Buildit Actuator の状態が以下のいずれかであること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。
 - STATE_FREE
 - STATE_READY
 - STATE_HOLD
 - STATE_FAULT_HOLD
 - STATE_FAULT_FREE

8.21.3 例

※バイト列は 16 進数表記

		device		Message	Payload	
Header	CRC	ID	Type ID	Size	Payload	
-----	-----	-----	-----	-----	-----	
ab cc ba	62	01	12	00 00	なし	

8.21.4 コマンドのペイロードフォーマット

ペイロードはありません (ペイロードサイズは 0)。

8.21.5 成功応答メッセージのペイロードフォーマット

width (bits)	16
+-----+	
MCP	
+-----+	

- MCP
 - 「共通フォーマット」を参照。

8.22 NACK

8.22.1 概要

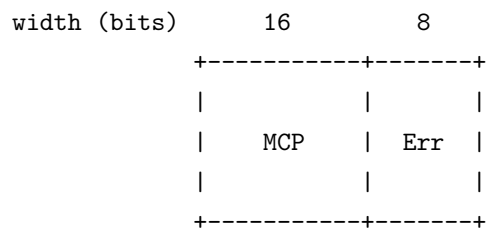
コマンドの失敗情報をホストに通知します。

8.23 失敗の種別

Buildit Actuator では複数のエラーが発生する可能性があります。エラーの種別を表す為に MCP エラー ID を用います。

名前	エラー ID	説明
MCP_INVALID_COMMAND_PAYLOAD_SIZE	0x03	不正なペイロードサイズ
MCP_INVALID_MSG_TYPE	0x04	不正なメッセージタイプ ID
MCP_INVALID_COMMAND_PAYLOAD	0x05	不正なペイロード
MCP_INVALID_OPERATION	0x06	不正な操作
MCP_OUT_OF_POSITION_LIMIT	0x09	ロータの角度が位置制限範囲外

8.23.1 失敗応答メッセージのペイロードフォーマット



- MCP
 - 「共通フォーマット」を参照。
- Err
 - MCP エラー ID。

8.24 PROTECTION_STOP

8.24.1 概要

保護停止します。STATE_PROTECTION_STOPPING に遷移し、ブレーキを使わずに停止させます。異常が発生せず回転速度が十分小さくなった時点で、保護停止ピンがグラウンドと接続されていれば、STATE_PROTECTION_STOP へ遷移し、そうでなければ STATE_READY に遷移します。

8.24.2 事前条件

- Buildit Actuator の状態が以下のうちいずれかであること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。
 - STATE_READY
 - STATE_CURRENT_SERVO
 - STATE_VELOCITY_SERVO
 - STATE_POSITION_SERVO
 - STATE_PROTECTION_STOP

8.24.3 コマンドのペイロードフォーマット

```
width (bits)      16
                  +-----+
                  |         |
                  | Timeout |
                  |         |
                  +-----+
```

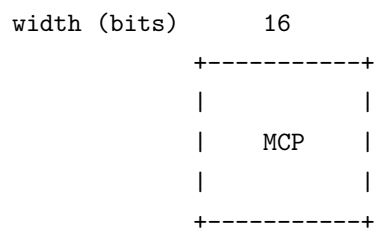
- Timeout
 - モータ制動による停止完了までのタイムアウト時間 [msec]。符号なし 16bit 整数値です。最低値は 500ms とし、それ以下を指定した場合は 500ms として扱われます。
 - タイムアウト時は、STATE_FAULT 状態に遷移します。

8.24.4 例

※バイト列は 16 進数表記

Header	CRC	device ID	Message Type ID	Payload Size	Payload
ab cc ba	ff	01	14	02 00	F4 01 (500)

8.24.5 成功応答メッセージのペイロードフォーマット



- MCP
 - 「共通フォーマット」を参照。

8.25 QUERY_SERVO_STATUS_CMD

8.25.1 概要

サーボ関連のセンサ値を取得します。

8.25.2 事前条件

- Buildit Actuator がシステムフォルト状態を除くいずれの状態であっても本コマンドを受け付けます。

8.25.3 コマンドのペイロードフォーマット

ペイロードはありません (ペイロードサイズは 0)。

8.25.4 例

※バイト列は 16 進数表記

		device	Message	Payload	
Header	CRC	ID	Type ID	Size	Payload
-----	-----	-----	-----	-----	-----
ab cc ba	7d	01	01	00 00	なし

8.25.5 成功応答メッセージのペイロードフォーマット

width (bits)	16	32	16	16	32	8	16
	+	+	+	+	+	+	+
	MCP	Pos	Vel	Cur	Ref	Temp	Faults
	+	+	+	+	+	+	+

- MCP * 「共通フォーマット」を参照。
- Pos
 - 位置センサ値。符号あり 32bit 整数値。0x10000 でエンコーダー 1 周分の移動量に相当します。ファームウェア起動直後は 0 以上 0xffff 以下の値となります。
- Vel
 - ギア変換後の回転速度 [rpm/100]。符号あり 16bit 整数値です。
- Cur
 - 電流センサ値 [mA]。符号あり 16bit 整数値です。
- Ref
 - 現在の状態に応じた制御指令値。符号あり 32bit 整数値です。符号あり 16bit 整数値の指令値は符号あり 32bit 整数値に拡張されます。

現在の状態	Ref の値
STATE_CURRENT_SERVO	電流制御指令値
STATE_VELOCITY_SERVO	速度制御指令値
STATE_POSITION_SERVO	位置制御指令値
上記以外	0

- Temp
 - 温度センサ値 [°C]。符号なし 8bit 整数値です。
- Faults
 - フォルト ID の論理和。

8.26 READY_CMD

8.26.1 概要

Ready 状態へと遷移させます。

8.26.2 事前条件

- Buildit Actuator の状態が以下のうちいずれかであること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。
 - STATE_READY
 - STATE_FREE
 - STATE_HOLD
- ロータの角度が PARAM_ID_POSITION_MIN_LIMIT と PARAM_ID_POSITION_MAX_LIMIT で指定された位置版制限の中にあること。ロータの角度が範囲外にあると失敗応答 NACK(MCP_OUT_OF_POSITION_LIMIT) となります。

8.26.3 コマンドのペイロードフォーマット

ペイロードはありません (ペイロードサイズは 0)。

8.26.4 例

※バイト列は 16 進数表記

Header	CRC	device ID	Message Type ID	Payload Size	Payload
ab cc ba	b4	01	10	00 00	なし

8.26.5 応答メッセージのペイロードフォーマット

width (bits)	16
+-----+	
MCP	
+-----+	

- MCP
 - 「共通フォーマット」を参照。

8.27 RESET_ROTATION

8.27.1 概要

Buildit が何回転しているかを変更します。

8.27.2 事前条件

- エンコーダー異常が発生していないこと。発生している場合は失敗応答 (MCP_INVALID_ENCODER_VALUE) となります。
- Buildit Actuator の状態が以下のうちいずれでもないこと。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。
 - STATE_READY
 - STATE_POSITION_SERVO
 - STATE_PROTECTION_STOPPING
 - STATE_PROTECTION_STOP

8.27.3 コマンドのペイロードフォーマット

```
width (bits)      16
                  +-----+
                  |       |
                  |  Rot  |
                  |       |
                  +-----+
```

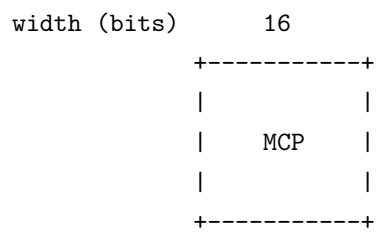
- Rot
 - リセット後の回転数。符号あり 16bit 整数値です。
 - 2147483647 や-2147483648 に近い値で位置制御を行わないよう注意して下さい。位置情報のオーバーフロー/アンダーフローが起こって意図せぬ挙動を起こす危険があります。

8.27.4 例

※バイト列は 16 進数表記

				device	Message	Payload		
	Header		CRC		ID		Type ID	
					Size		Payload	
	-----				-----			
	ab cc ba		97		01		32	
							02 00	
							00 00	

8.27.5 成功応答メッセージのペイロードフォーマット



- MCP
 - 「共通フォーマット」を参照。

8.28 SET_PARAM_CMD

8.28.1 概要

パラメータを設定します。

設定されたパラメータは、Buildit Actuator の電源をオフにしても Buildit Actuator 内に保存されます。

パラメータに設定された値は即座に Buildit Actuator の動作に反映されます。ただし、PARAM_ID_DEVICE_ID を設定した場合は Buildit Actuator は通信を再開するまで 1 秒程度かかります。PARAM_ID_DEVICE_ID を設定した場合は 2 秒以上時間を開けてから次のコマンドを送信するようにして下さい。

- 以下のパラメータについては読み込み専用である為、設定出来ません。
 - PARAM_ID_FIRMWARE_VERSION
 - PARAM_ID_POWER_ON_TIME
 - PARAM_ID_CALIBRATION
 - PARAM_ID_POSITION_SYS_OFFSET

8.28.2 事前条件

- パラメータPARAM_ID_POSITION_MAX_LIMIT、PARAM_ID_POSITION_MIN_LIMIT、PARAM_ID_POSITION_OFFSET については以下の状態以外であること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。
 - STATE_READY
 - STATE_PROTECTION_STOP
 - STATE_PROTECTION_STOPPING
- 上記以外のパラメータについては、Buildit Actuator が通信可能な状態であること。

8.28.3 コマンドのペイロードフォーマット

width (bits) 8 8 or 16 or 32

```
+-----+-----+
| Param |       |
|  Id   | Param |
|       |       |
+-----+-----+
```

- Param Id
 - 設定するパラメータの ID。不正な ID は失敗応答 NACK(MCP_INVALID_COMMAND_PAYLOAD) となります。
 - Param Id の詳細は「パラメータについて」を参照してください。
- Param
 - 設定するパラメータの値。符号の有無、ビット幅は設定するパラメータによって異なります。
 - 無効なパラメータ値は失敗応答 NACK(MCP_INVALID_COMMAND_PAYLOAD) となります。
 - Param の詳細は「パラメータについて」を参照してください。

8.28.4 例

※バイト列は 16 進数表記

					device		Message		Payload			
	Header		CRC		ID		Type ID		Size		Payload	
	-----		-----		-----		-----		-----		-----	
	ab cc ba		e2		01		30		00 02		80 01	
											(PARAM_ID_DEVICE_ID, 1)	

8.28.5 成功応答メッセージのペイロードフォーマット

width (bits)	16
	+-----+
	MCP
	+-----+

- MCP
 - 「共通フォーマット」を参照。

8.29 SET_REF_CURRENT_CMD

8.29.1 概要

電流制御の指令値 [mA] を設定します。モーターに流れる電流が大きい程、発生するトルクは大きくなります。PARAM_ID_CURRENT_MAX_LIMIT 及び PARAM_ID_CURRENT_MIN_LIMIT で設定された範囲外であればその範囲に制限された値が反映されます。

8.29.2 事前条件

- Buildit Actuator の状態が以下のうちいずれかであること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。
 - STATE_READY
 - STATE_CURRENT_SERVO
 - STATE_VELOCITY_SERVO
 - STATE_POSITION_SERVO

8.29.3 コマンドのペイロードフォーマット

```
width (bits)      16
                  +-----+
                  |         |
                  |  Cur   |
                  |         |
                  +-----+
```

- Cur
 - 電流制御の指令値。単位は無単位です。符号あり 16bit 整数値です。

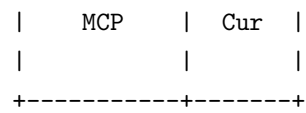
8.29.4 例

※バイト列は 16 進数表記

Header	CRC	device ID	Message Type	Payload Size	Payload
ab cc ba	83	01	20	02 00	E8 03
					(1000)

8.29.5 成功応答メッセージのペイロードフォーマット

```
width (bits)      16      16
                  +-----+-----+
                  |         |         |
```



- MCP
 - 「共通フォーマット」を参照。
- Cur
 - 電流センサ値。符号あり 16bit 整数値です。

8.30 SET_REF_POSITION_CMD

8.30.1 概要

位置制御の指令値を設定します。PARAM_ID_POSITION_MAX_LIMIT 及び PARAM_ID_POSITION_MIN_LIMIT で設定された範囲外であればその範囲に制限された値が反映されます。ただし、2147483647 や -2147483648 のような符号あり 32bit 整数で表現可能な値の上限・下限近辺の値を指令値に使うことは避けて下さい。オーバーフロー/アンダーフローにより意図せぬ挙動になる恐れがあります。

8.30.2 事前条件

- Buildit Actuator の状態が以下のうちいずれかであること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。
 - STATE_READY
 - STATE_CURRENT_SERVO
 - STATE_VELOCITY_SERVO
 - STATE_POSITION_SERVO

8.30.3 コマンドのペイロードフォーマット

```
width (bits)      32
                  +-----+
                  |  Ref  |
                  |  Pos  |
                  |      |
                  +-----+
```

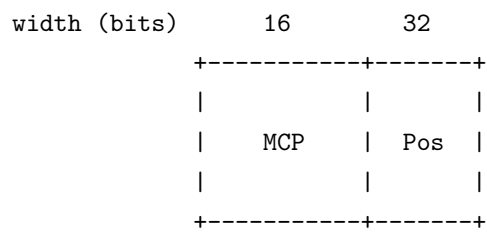
- RefPos
 - 位置制御の指令値。符号あり 32bit 整数値です。0x10000 でエンコーダー 1 周分の移動量に相当します。

8.30.4 例

※バイト列は 16 進数表記

				device	Message	Payload		
	Header		CRC		ID		Type ID	
							Size	
	ab cc ba		76		01		24	
							04 00	
							00 00 01 00	
							(1 周分)	

8.30.5 成功応答メッセージのペイロードフォーマット



- MCP
 - 「共通フォーマット」を参照。
- Pos
 - 位置センサ値。符号あり 32bit 整数値です。0x10000 でエンコーダー 1 周分の移動量に相当します。ファームウェア起動直後は 0 以上 0xffff 以下の値となります。

8.31 SET_REF_POSITION_WITH_VELOCITY_CMD

8.31.1 概要

位置制御の指令値を設定します。このコマンドによる回転時の速度は同時に指定した速度指令値以下に制限されます。PARAM_ID_POSITION_MAX_LIMIT 及び PARAM_ID_POSITION_MIN_LIMIT で設定された範囲外であればその範囲に制限された位置指令値が反映されます。ただし、2147483647 や -2147483648 のような符号あり 32bit 整数で表現可能な値の上限・下限近辺の値を指令値に使うことは避けて下さい。オーバーフロー/アンダーフローにより意図せぬ挙動になる恐れがあります。

8.31.2 事前条件

- Buildit Actuator の状態が以下のうちいずれかであること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。
 - STATE_READY
 - STATE_CURRENT_SERVO
 - STATE_VELOCITY_SERVO
 - STATE_POSITION_SERVO

8.31.3 コマンドのペイロードフォーマット

width (bits)	32	16
	+-----+-----+	
	Ref	
	Pos	vel
	+-----+-----+	

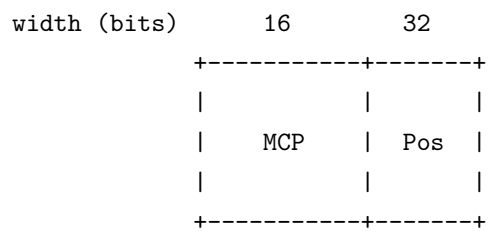
- RefPos
 - 位置制御の指令値。符号あり 32bit 整数値です。0x10000 でエンコーダー 1 周分の移動量に相当します。
- vel
 - 速度指令値 [rpm/100]。符号あり 16bit 整数値です。どちらの方向に回転する場合でも絶対値で指定してください。

8.31.4 例

※バイト列は 16 進数表記

		device	Message	Payload	
Header	CRC	ID	Type ID	Size	Payload
-----	-----	-----	-----	-----	-----
ab cc ba	76	01	26	06 00	00 00 01 00 E8 03
					(1 周分・1000:10rpm)

8.31.5 成功応答メッセージのペイロードフォーマット



- MCP
 - 「共通フォーマット」を参照。
- Pos
 - 位置センサ値。符号あり 32bit 整数値です。0x10000 でエンコーダー 1 周分の移動量に相当します。ファームウェア起動直後は 0 以上 0xffff 以下の値となります。

8.32 SET_REF_VELOCITY_CMD

8.32.1 概要

速度制御の指令値を設定します。PARAM_ID_VELOCITY_MAX_LIMIT 及び PARAM_ID_VELOCITY_MIN_LIMIT で設定された範囲外であればその範囲に制限された値が反映されます。

8.32.2 事前条件

- Buildit Actuator の状態が以下のうちいずれかであること。これを満たさない場合は失敗応答 NACK(MCP_INVALID_OPERATION) となります。
 - STATE_READY
 - STATE_CURRENT_SERVO
 - STATE_VELOCITY_SERVO
 - STATE_POSITION_SERVO

8.32.3 コマンドのペイロードフォーマット

```
width (bits)      16
                  +-----+
                  |  Ref  |
                  |  Vel  |
                  |      |
                  +-----+
```

- RefVel
 - 速度制御の指令値 [rpm/100]。符号あり 16bit 整数値です。

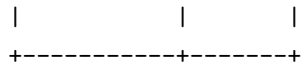
8.32.4 例

※バイト列は 16 進数表記

	device	Message	Payload	
Header	CRC	ID	Type ID	Size
ab cc ba	47	01	22	02 00
				E8 03
				(1000: 10rpm)

8.32.5 成功応答メッセージのペイロードフォーマット

```
width (bits)      16      16
                  +-----+-----+
                  |      |      |
                  |  MCP  |  Vel  |
```



- MCP
 - 「共通フォーマット」を参照。
- Vel
 - 速度センサ値。符号あり 16bit 整数値です。

9 スペック

9.1 仕様

項目	値
寸法	61mm x 61mm x 124mm
重量	1100g
制御モード	電流制御、速度制御、位置制御
ブレーキ	保持用ブレーキ (制動不可)
温度	0℃～40℃
湿度	80%RH 以下、ただし結露なきこと
通信形式	RS485(コマンド方式)
通信 ch	1ch(ディジーチェーン接続)
通信電圧	3.3V
通信速度	115200bps
通信方向	半二重通信
電源電圧	24V ± 5%
定格消費電流	3A
減速比	100
定格回転数	30rpm
定格トルク	6Nm
出力軸の許容モーメント荷重	40Nm

9.2 寸法図

寸法図を fig. 7 に示します。

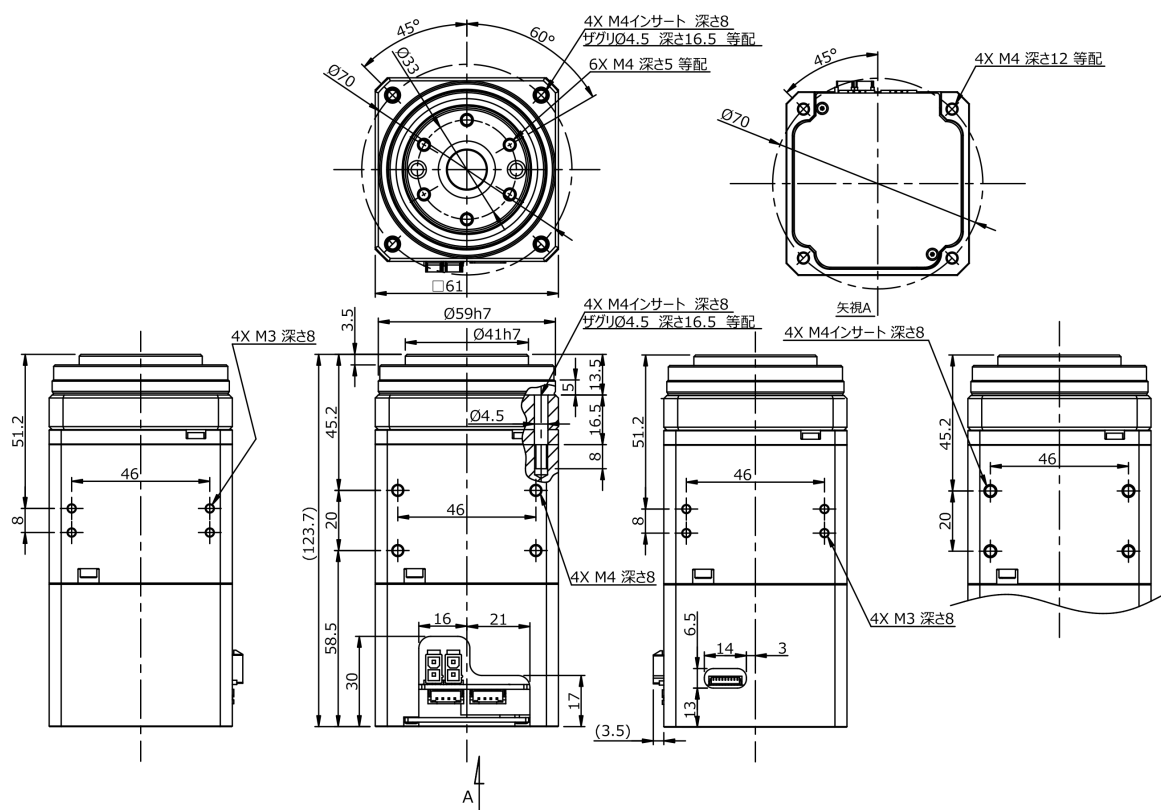


Figure 7: 寸法図

10 サードパーティ製品のライセンスについて

本製品には以下のソフトウェアが含まれます。

- MCSDK_v5.1.3
- STM32F4xx HAL Driver

これらのソフトウェアの使用は、次のライセンス条件によって規定されています。

```
*****
*
* ©Copyright (c) 2018 STMicroelectronics International N.V.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted, provided that the following conditions are met:
*
* 1. Redistribution of source code must retain the above copyright notice,
*   this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
*   this list of conditions and the following disclaimer in the documentation
*   and/or other materials provided with the distribution.
* 3. Neither the name of STMicroelectronics nor the names of other
*   contributors to this software may be used to endorse or promote products
*   derived from this software without specific written permission.
* 4. This software, including modifications and/or derivative works of this
*   software, must execute solely and exclusively on microcontroller or
*   microprocessor devices manufactured by or for STMicroelectronics.
* 5. Redistribution and use of this software other than as permitted under
*   this license is void and will automatically terminate your rights under
*   this license.
*
* THIS SOFTWARE IS PROVIDED BY STMICROELECTRONICS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS, IMPLIED OR STATUTORY WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
* PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY
* RIGHTS ARE DISCLAIMED TO THE FULLEST EXTENT PERMITTED BY LAW. IN NO EVENT
* SHALL STMICROELECTRONICS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
* OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
* EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*****
```