

UNIVERSAL ROBOTS

クイック・スタート・マニュアル

機種名 : ARH305A

Rev. 1.01



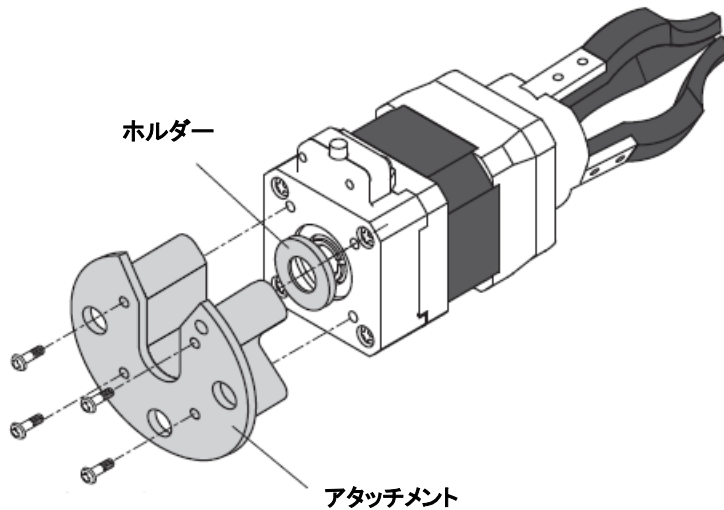
 Plexmotion

1. 準備と設置

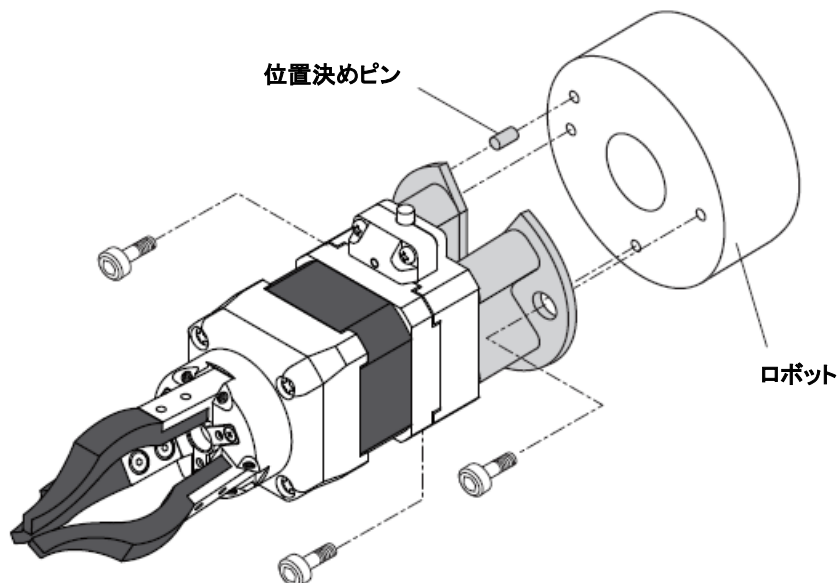


ロボットとの設置作業は、ロボットの電源がオフの状態で行ってください。

1.1. ハンドの取り付け



1. ロボットハンド本体に専用アタッチメントを取り付けます(専用アタッチメントにネジは同梱します)。

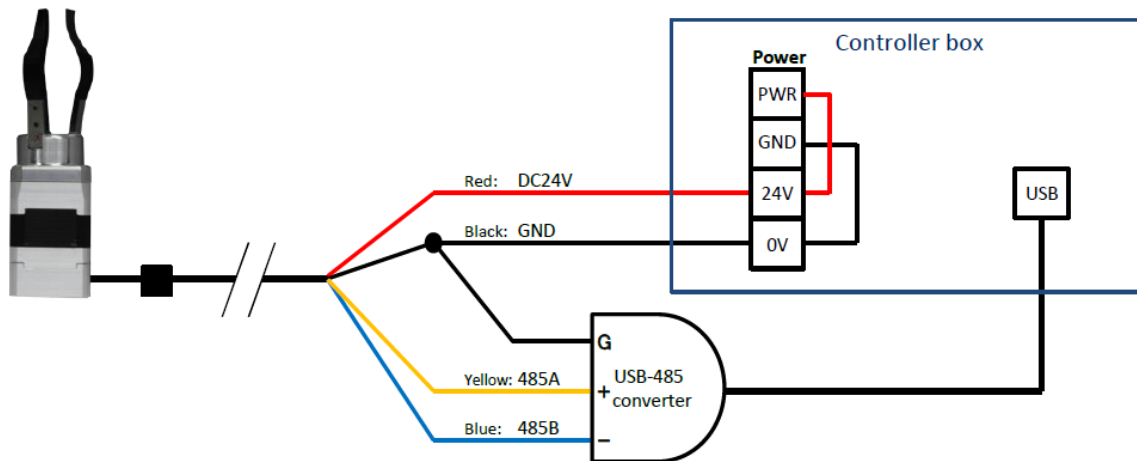


2. アタッチメントをロボットに取り付けます。
3. ハンドの本体ケーブルとインターフェースケーブルとを連結します。
4. インターフェースケーブルとロボットとを接続します。

1.2. 通信接続

1.2.1. CB シリーズ RS485 通信

CB シリーズで通信によるハンド操作を行う際には、コントローラボックスとハンドとを接続してください。



1. 通信用延長ケーブルと、USB-RS485 コンバータを用意します。
2. ハンドの本体ケーブルと通信用延長ケーブルを連結します。
3. 通信用ケーブルの赤線と黒線を、それぞれコントローラボックスの Power ブロック 24V, 0V に結線します。
4. 通信用ケーブルの黒線を分岐させ、USB-RS485 コンバータの G 端子と結線します。
5. 通信用ケーブルの黄線と青線を、それぞれ USB-RS485 コンバータの+端子と-端子とに結線します。
6. USB-RS485 コンバータとコントローラボックスとを USB で接続します。

1.2.2. e シリーズ RS485 通信

e シリーズではロボットのツール先端コネクタに RS485 通信機能を備えています。ハンドとインターフェースケーブルとを接続するだけで、そのまま御使用できます。

その際には、3.2.2 項を確認の上、ツール I/O のシリアル通信設定を行ってください。

2. URCaps のインストール

1. ロボットの設定画面から“URCaps”をタップします。
2. URCaps のファイルをコピーした USB メモリを Polyscope に挿入し、“+”をタップします。
3. ファイル画面から ShinanoARH-1.X.urcap を選択し、アクティブな URCaps の欄に“ShinanoARH”を追加します。
4. “再起動”をタップしてロボットを再起動します。

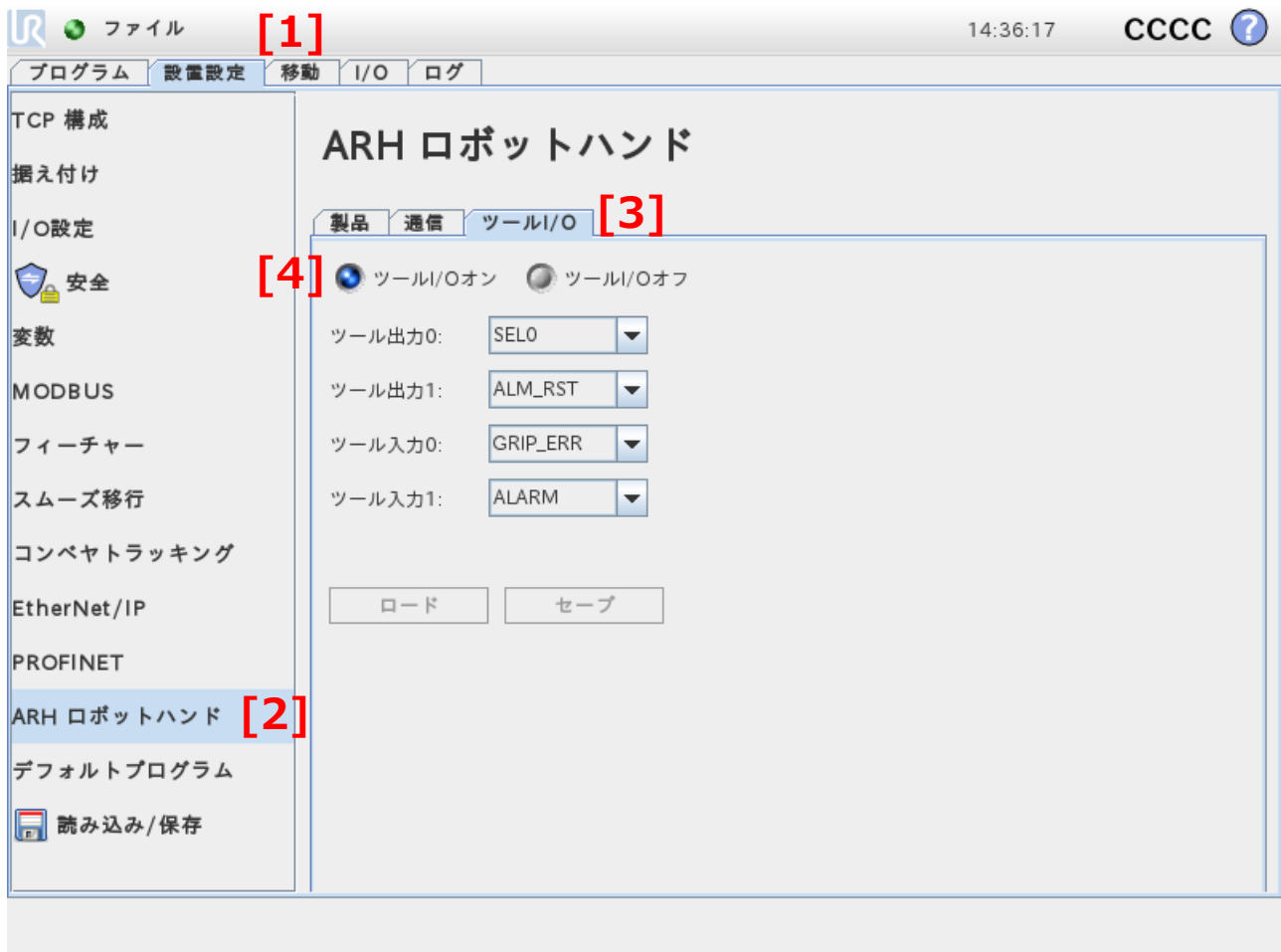


他社製品の URCaps がインストールされていると、ハンドが誤動作する場合がございます。
不要な URCaps はアンインストールしてください。

3. コンフィグレーション

3.1. 設置設定

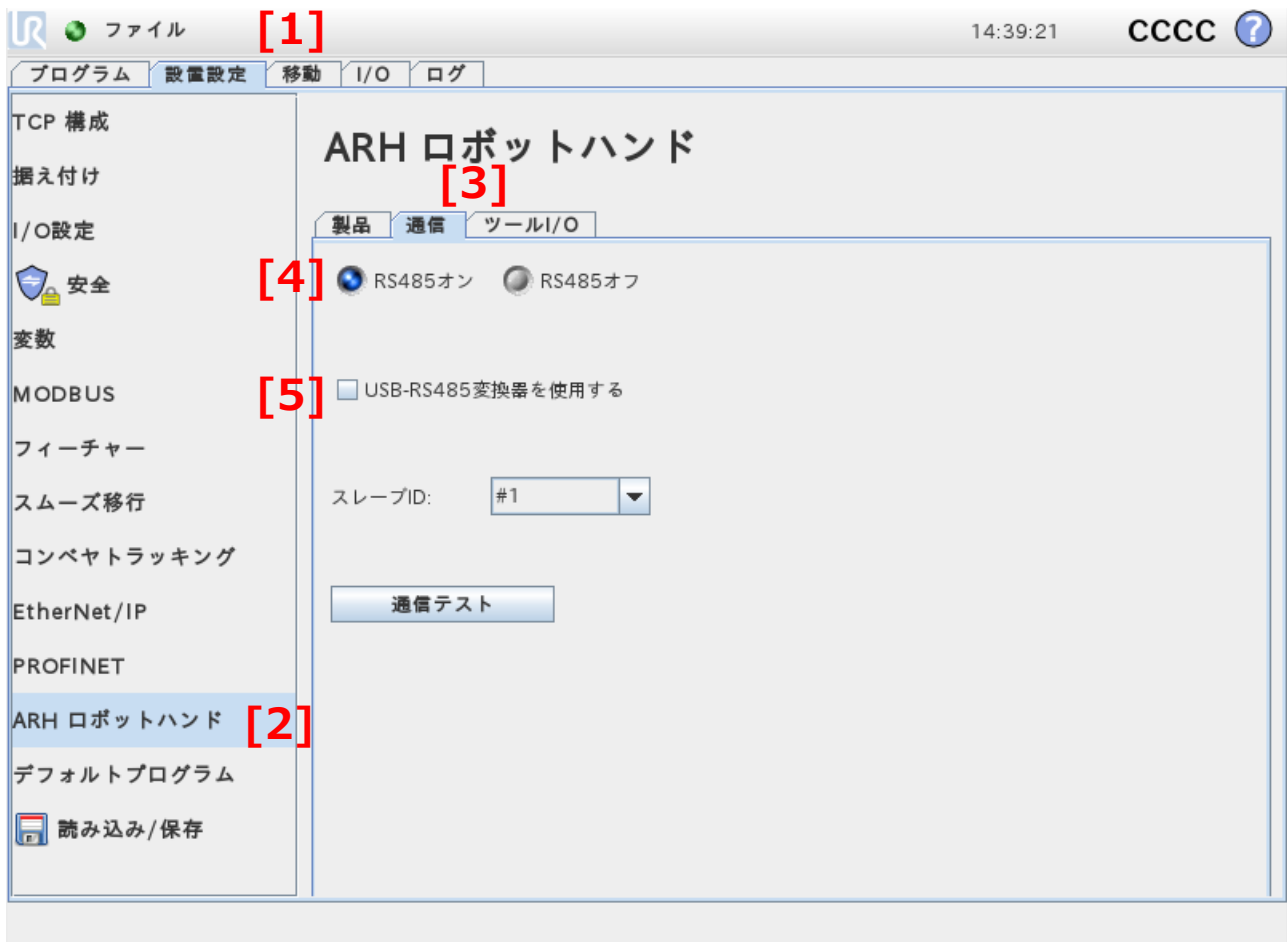
3.1.1. ツール I/O 設定



ツール I/O でのハンド操作:

1. “設置設定”をタップします。
2. “ARH ロボットハンド”をタップします。
3. “ツール I/O”のタブをタップします。
4. “ツール I/O オン”をタップします。

3.1.2. RS485 通信設定



RS485 通信でのハンド操作:

1. “設定設定”をタップします。
2. “ARH ロボットハンド”をタップします。
3. “通信”のタブをタップします。
4. “RS485 オン”をタップします。

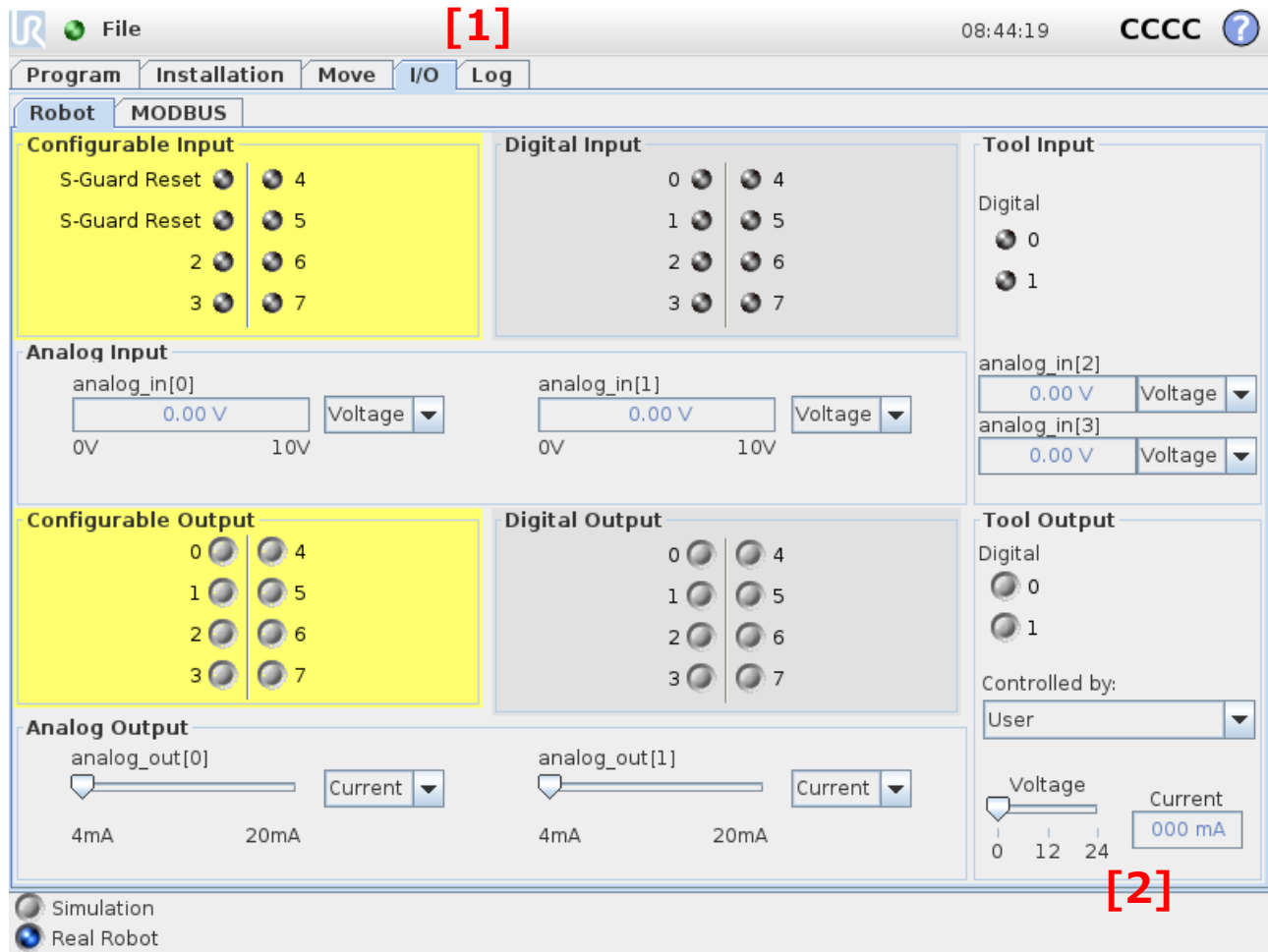
USB での通信操作:

5. USB-RS485 変換器を使用する場合は、“USB-RS485 変換器を使用する”にチェックを入れます。

3.2. I/O 設定

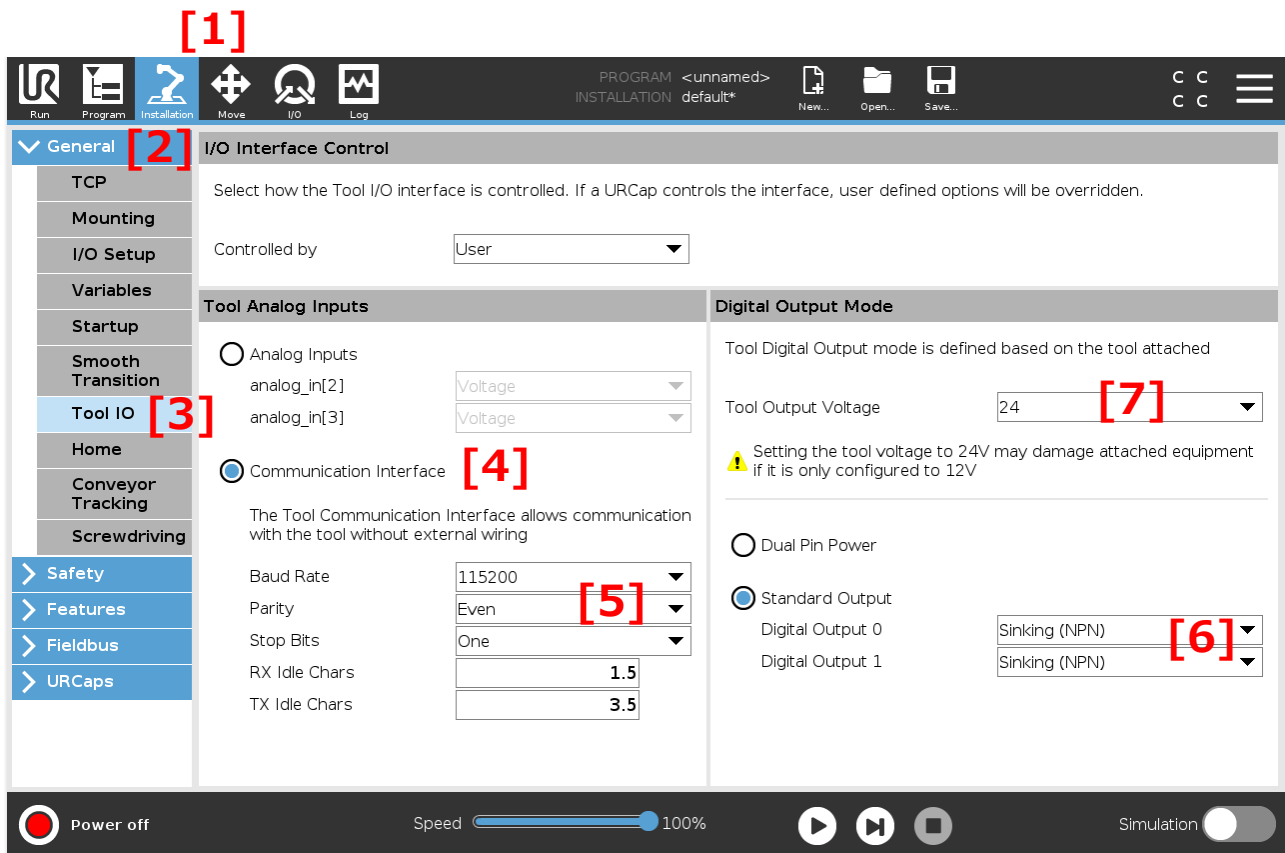
ツール I/O で使用する際には、ツール I/O から電源を供給するための I/O 設定をしてください。

3.2.1. CB シリーズ (Polyscope バージョン 3.X)



1. メニューより“I/O”をタップします。
2. ツール出力の電圧を 24 に設定します。

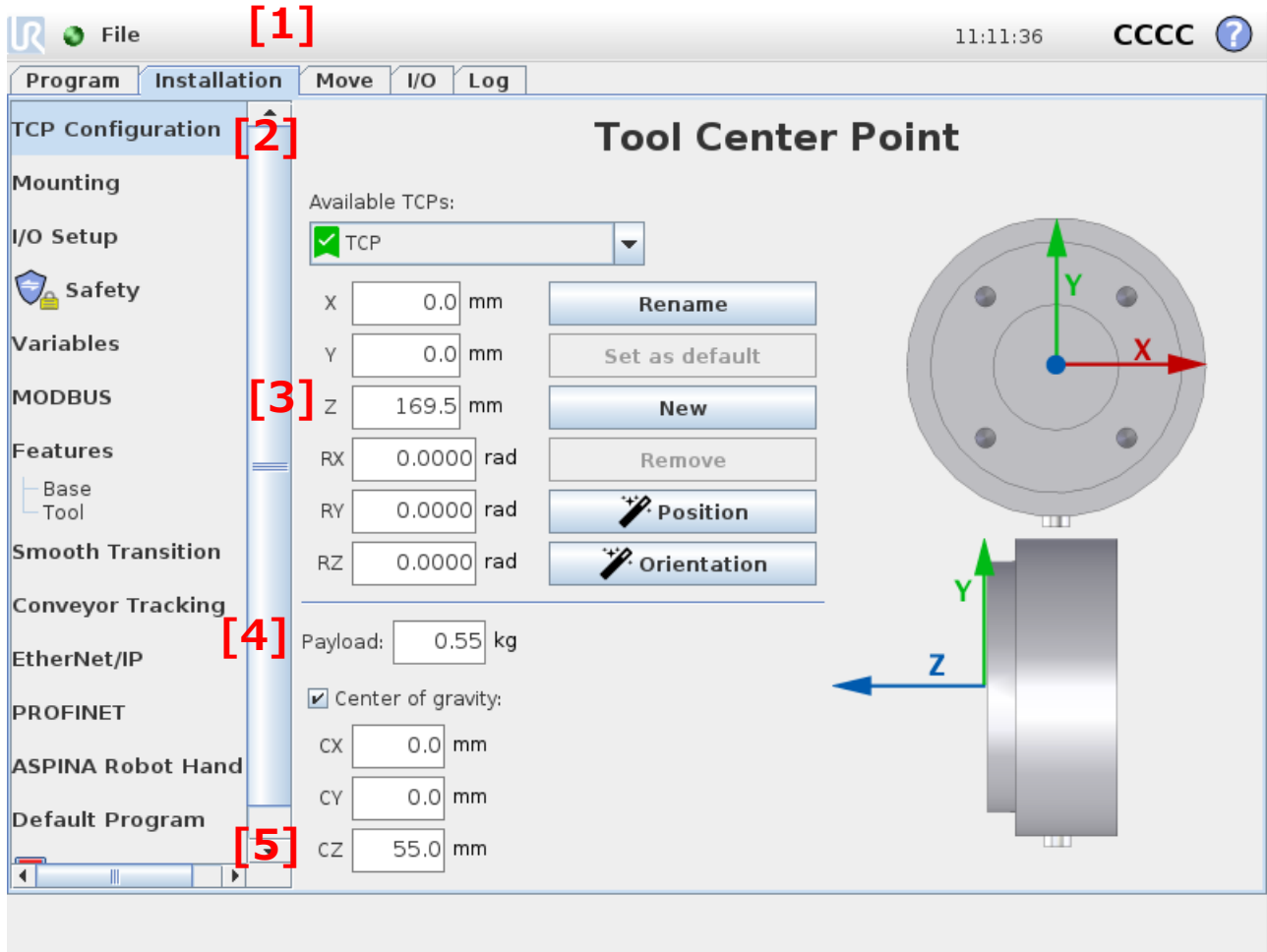
3.2.2. e シリーズ (Polyscope バージョン 5.X)



1. “設置設定”をタップします。
2. “全般”をタップします。
3. “ツール I/O”をタップします。
4. “通信インターフェース”を選択します。
5. ボーレートを「115200」、パリティを「偶数」、ストップビットを「1」に設定します。
6. “標準出力”を選択し、デジタル出力をどちらも「シンク(NPN)」に設定します。
7. ツールの出力電圧を「24」に設定します。

3.3. ツール中心点 (TCP)

3.3.1. CB シリーズ (Polyscope バージョン 3.X)



1. “インストレーション”をタップします。
2. “TCP 構成”をタップします。
3. Z の入力欄をタップし、ハンドの高さを入力します。
4. 有効荷重をタップし、ハンドの重さを入力します。
5. 重力中心をタップし、cz の入力欄に中心までの距離を入力します。



上の画像は標準爪および標準アタッチメントを装着した場合の数値です。
 ハンドの高さは、爪の長さや取り付けアタッチメントのサイズを考慮した数字を入力してください。

3.3.2. e シリーズ (Polyscope バージョン 5.X)

[1]

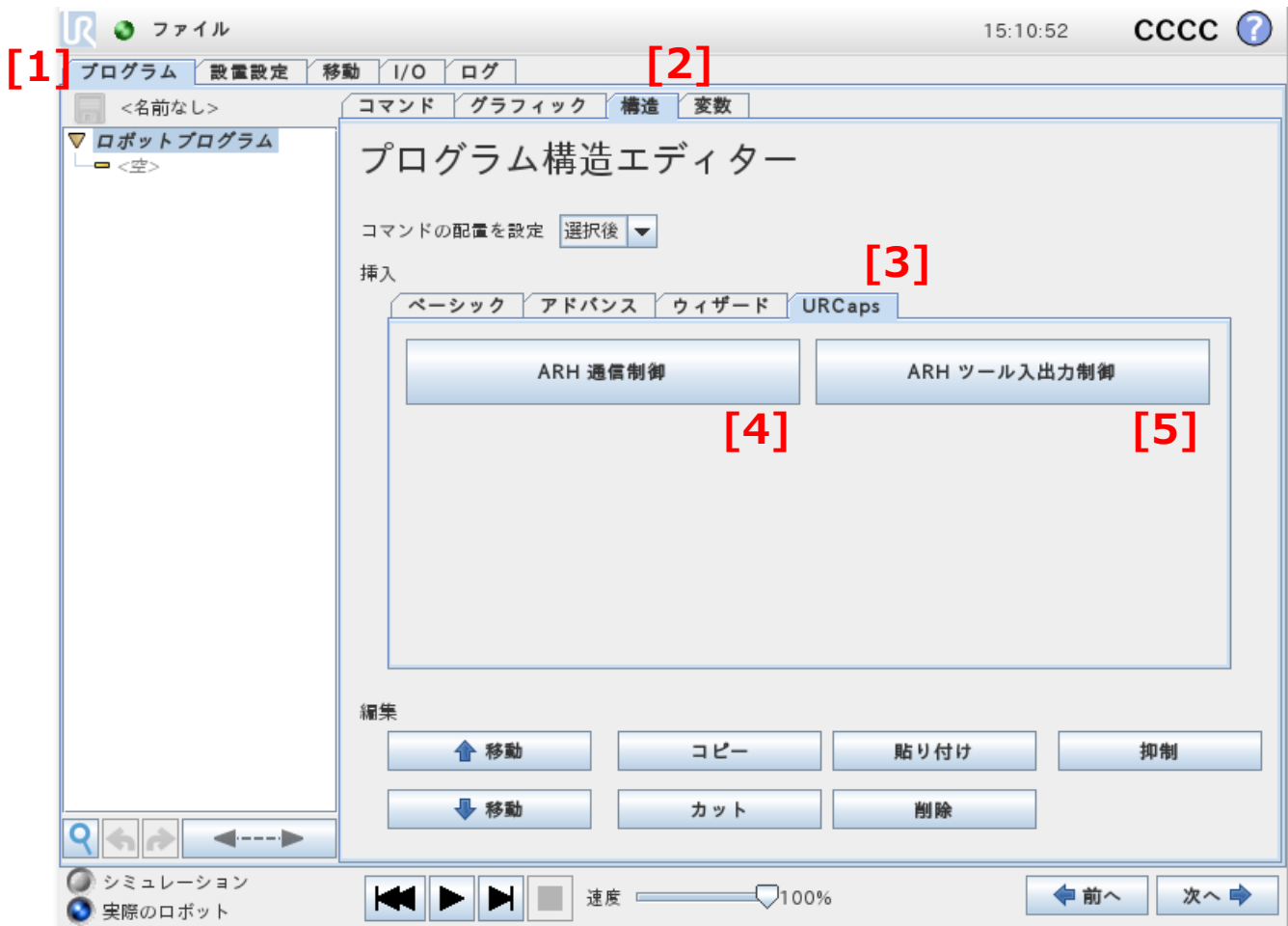
The screenshot displays the Polyscope software interface for configuring the Tool Center Point (TCP). The interface includes a top toolbar with icons for Run, Program, Installation, Move, I/O, and Log. The main window is divided into a left sidebar with a 'General' section and a main configuration area. The 'General' section contains a list of settings: TCP, Mounting, I/O Setup, Variables, Startup, Smooth Transition, Tool IO, Home, Conveyor Tracking, and Screwwdriving. The 'TCP' setting is selected, and its configuration is shown in the main area. The 'Tool Center Point' section includes a 'Position' field with X (0.0 mm), Y (0.0 mm), and Z (169.5 mm) inputs, and an 'Orientation' field with RX, RY, and RZ (all 0.0000 rad) inputs. The 'Payload and Center of Gravity' section includes a 'Payload' field (0.55 kg) and a checked 'Center of gravity' checkbox with CX, CY, and CZ (0.0 mm, 0.0 mm, 55.0 mm) inputs. To the right, a 3D diagram shows a gripper with coordinate axes (X, Y, Z) and a red dot indicating the TCP. Red callout boxes [1] through [5] point to the 'Settings' icon, the 'TCP' button, the Z input field, the Payload field, and the 'Center of gravity' checkbox respectively.

1. “設置設定”をタップします。
2. “TCP”をタップします。
3. Z の入力欄をタップし、ハンドの高さを入力します。
4. 荷重設定をタップし、ハンドの重さを入力します。
5. 重心をタップし、cz の入力欄に中心までの距離を入力します。



上の画像は標準爪および標準アタッチメントを装着した場合の数値です。
 ハンドの高さは、爪の長さや取り付けアタッチメントのサイズを考慮した数字を入力してください。

4. プログラムへの追加



1. プログラム画面を開きます。
2. “構造”をタップします。
3. プログラムに挿入する項目から“URCaps”をタップします。

RS485 通信でのハンド操作:

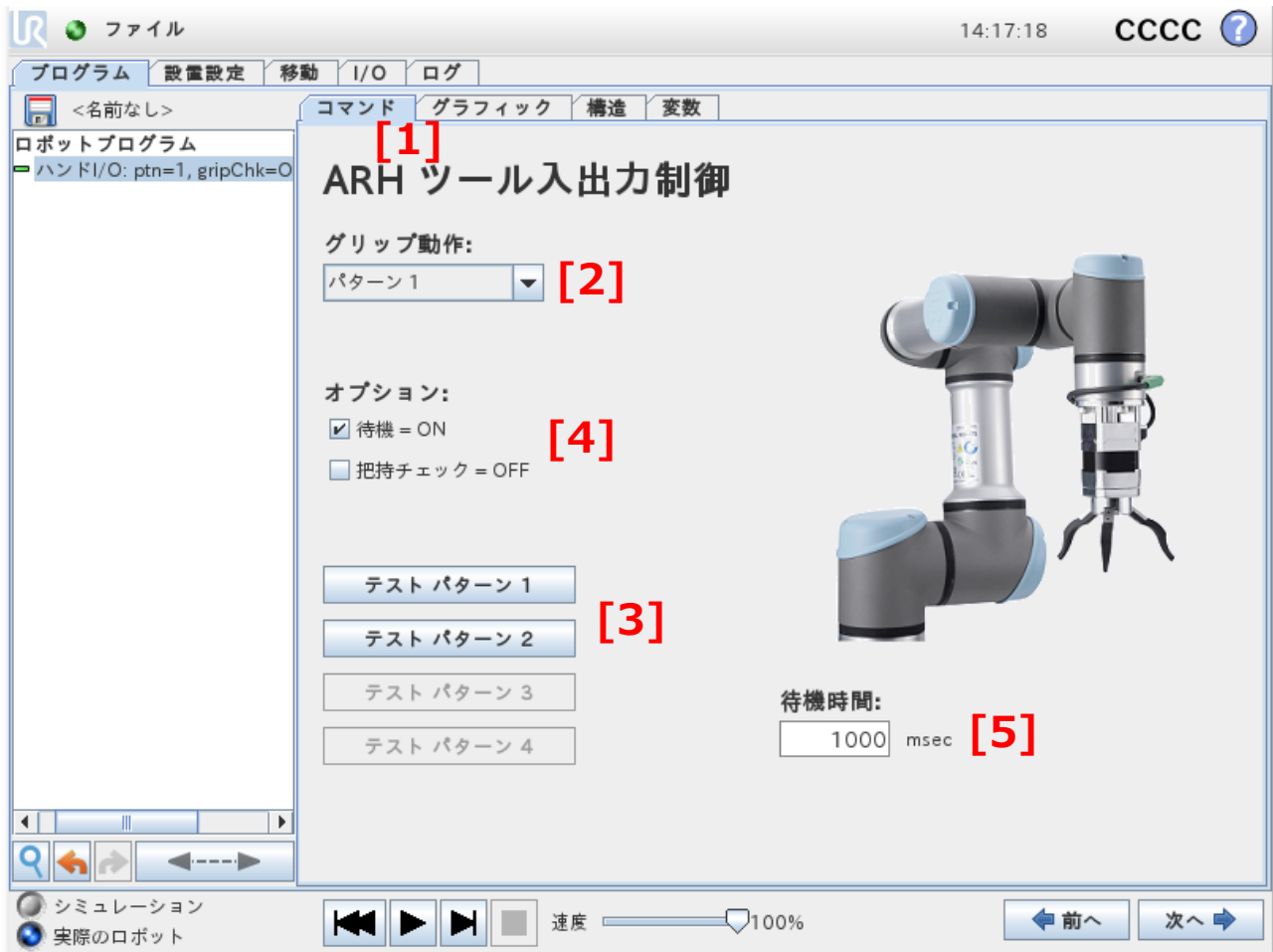
4. “ARH 通信制御”をタップします。

ツール I/O でのハンド操作:

5. “ARH ツール入出力制御”をタップします。

5. プログラムの設定

5.1. ARH ツール入出力制御 コマンド画面



1. “コマンド”をタップします。
2. グリップパターン番号を選択します。
(※デフォルトはパターン 1 が全開動作、パターン 2 が全閉動作です。)
3. パターン番号のボタンをタップすることで開閉動作を確認できます。

把持チェック:

4. 閉動作時の把持チェックを許可するには、“待機 = ON”と“把持チェック = ON”にチェックを入れます。

待機時間の調整:

5. ハンド動作中の待ち時間を調整するには、待機時間の入力欄をタップし、時間を入力します。



ツール I/O による把持チェックは、
ツール I/O の入力 0 もしくは入力 1 に、ハンドの GRIP_ERR 信号を割り当てます。



グリップパターン番号 1~4 は、
ハンド専用の PC アプリケーションで事前ティーチング可能です。



グリップパターンのパターン 3 と 4 は、
インストレーション画面の”ツール出力 1”が SEL*の場合のみ有効です。

5.2. ARH 通信制御 コマンド画面

5.2.1. ベーシックメニュー

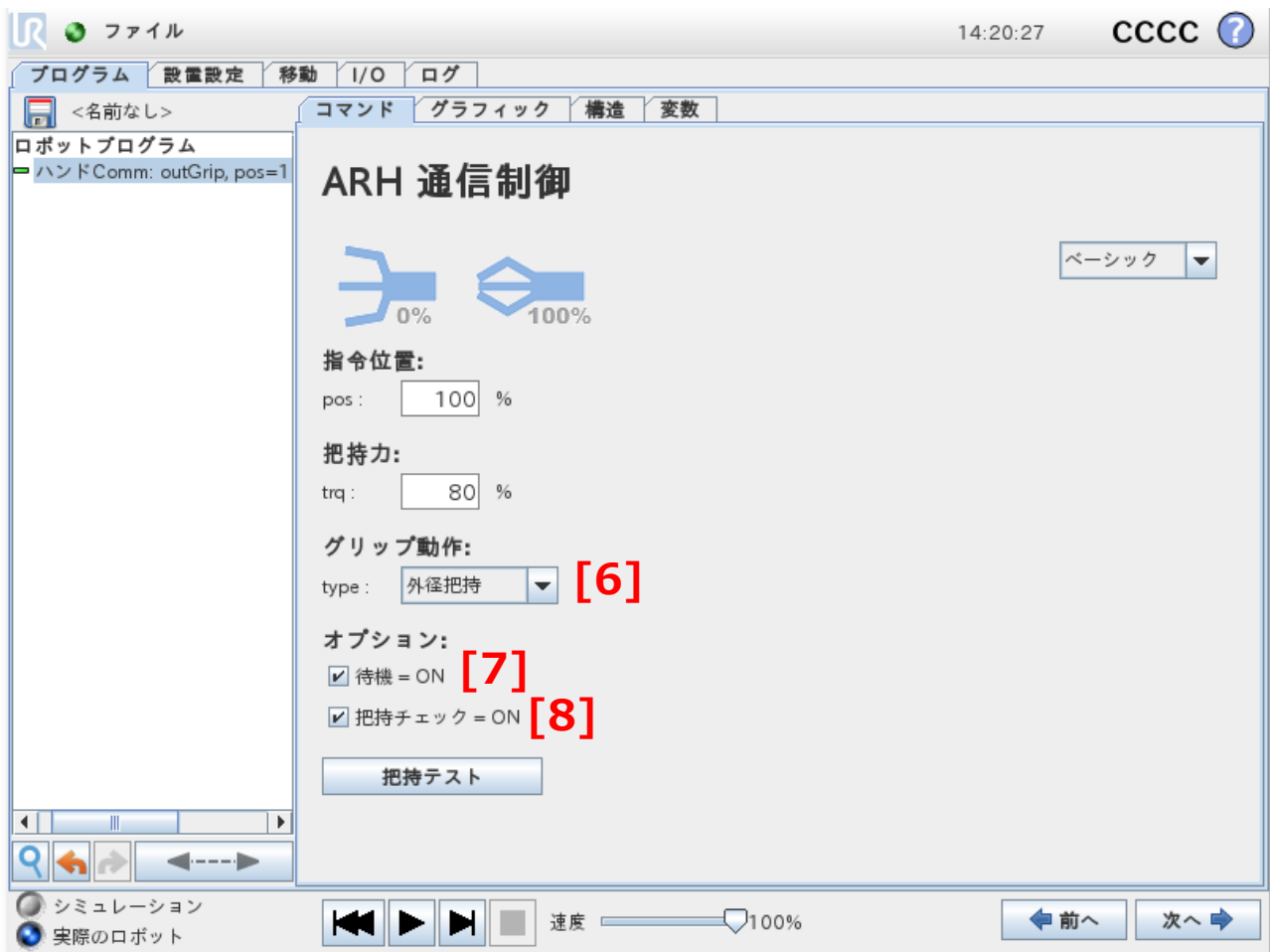


1. “コマンド”をタップします。
2. “ベーシック”を選択します。
3. 指令位置の入力欄をタップし、目標位置を入力します。
4. 把持力の入力欄をタップし、把持力を入力します。
5. “把持テスト”をタップすることで、実動作によるパラメータ確認ができます。



テストをタップする前に、ハンドの目標位置を更新してください。
目標位置が固定のままでは動作しません。

< 把持チェックの設定 >



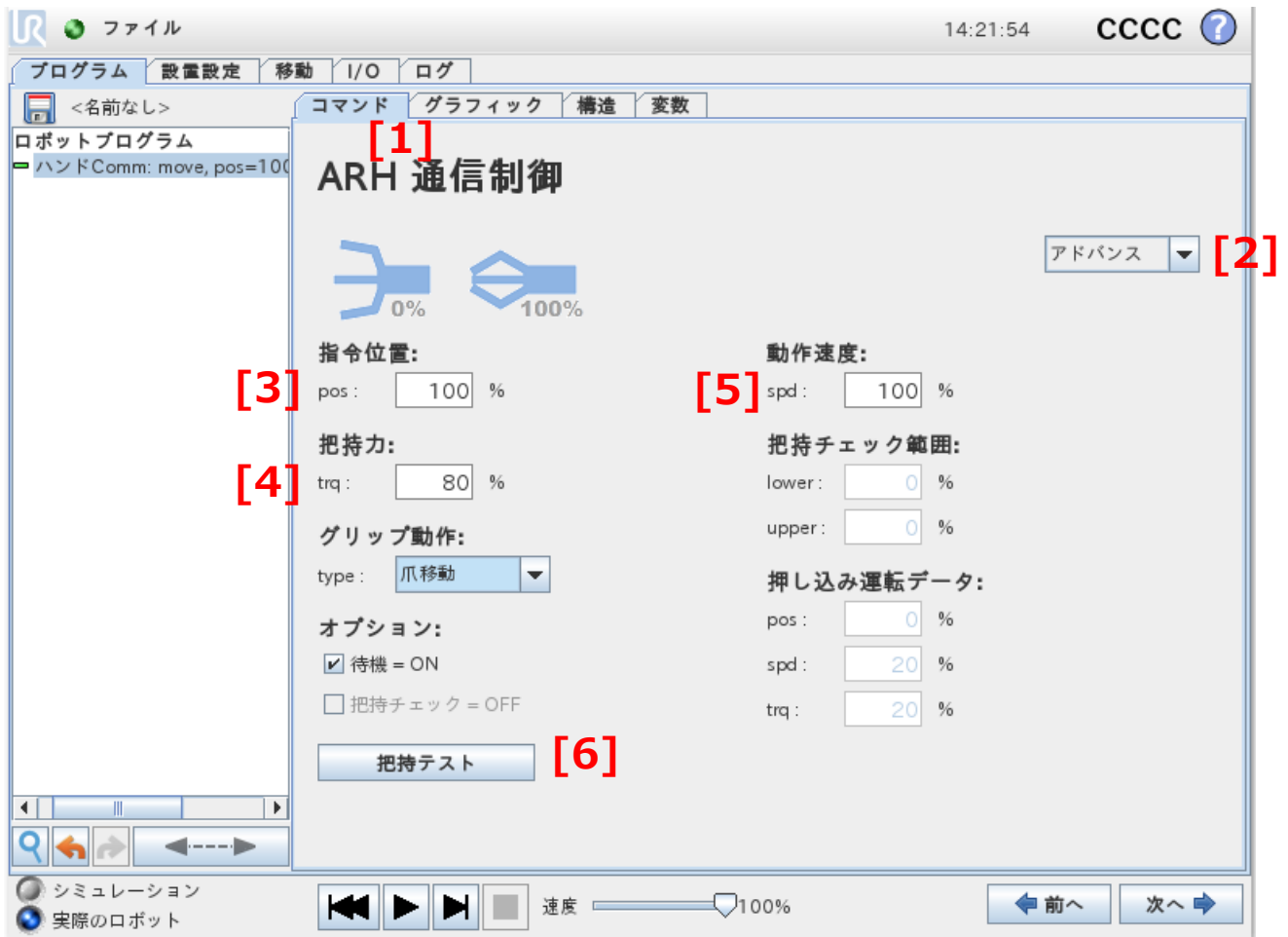
把持チェック:

6. ワークを外側から掴む場合は“外径把持”、内側から掴む場合は“内径把持”を選択します。
7. “待機=ON”にチェックを入れます。
8. “把持チェック=ON”にチェックを入れます。



把持チェックはハンドが動作完了するのを待って実行するため、
待機 = OFF ならば把持チェック = ON が選択できません。

5.2.2. アドバンスメニュー



1. “コマンド”をタップします。
2. “アドバンス”を選択します。
3. 指令位置の入力欄をタップし、目標位置を入力します。
4. 把持力の入力欄をタップし、把持力を入力します。
5. 動作速度の入力欄をタップし、速度を入力します。
6. “把持テスト”をタップすることで、実動作によるパラメータ確認ができます。



テストをタップする前に、ハンドの目標位置を更新してください。
目標位置が固定のままでは動作しません。

<把持チェック、押し込み運転の設定>



把持チェック:

7. ワークを外側から掴む場合は“外径把持”、内側から掴む場合は“内径把持”を選択します。
8. “待機=ON”にチェックを入れます。
9. “把持チェック=ON”にチェックを入れます。
10. 把持チェック範囲の入力欄をタップし、把持チェック範囲の下限と上限を調整します。

押し込み運転:

11. 押し込み運転を行うには、押し込み量(pos.)のパラメータを 0 以外に設定し、押し込み速度(spd.)と押し込みトルク(trq.)を調整します。
12. “把持テスト”をタップすることで、実動作によるパラメータ確認ができます。



把持チェックはハンドが動作完了するのを待って実行するため、
待機 = OFF ならば把持チェック = ON が選択できません。



把持チェックの上下限値は「下限<上限」になるように設定してください。
下限=上限は把持チェックが無効となります。



動作タイプを”爪移動”にすると押し込み運転無効となります。
押し込み量、押し込み速度、押し込みトルクの設定はできません。



押し込み運転のパラメータは、
pos,spd,trq のいずれかが 0 だと押し込み運転無効となります。

6. 仕様

6.1. URCaps 仕様

対応製品

- UR3e, UR5e, UR10e
- UR3, UR5, UR10 (CB3.1)

ソフトウェアバージョン条件

- PolyScope 3.7 以上(CB3.1) および 5.0 以上(e-Series)

6.2. パラメータ設定範囲

各パラメータの設定範囲は下記の通りです。通信操作は全てパーセントで設定します。

表 1 I/O 操作パラメータ

項目名	パラメータ名	単位	有効範囲
待機時間	ハンド動作待機時間	msec	0 ~ 10000

表 2 通信操作パラメータ

項目名	パラメータ名	単位	有効範囲
指令位置 pos.	ハンド目標位置	%	0 ~ 100
把持力 trq.	ハンド位置決めトルクリミット	%	15 ~ 100
動作速度 spd.	ハンド運転速度	%	10 ~ 400
把持チェック範囲 lower	グリップエラー範囲 下限値	%	0 ~ 100
把持チェック範囲 upper	グリップエラー範囲 上限値	%	0 ~ 100
押し込み運転 pos.	押し込み量	%	0 ~ 100
押し込み運転 spd.	押し込み速度	%	10 ~ 100
押し込み運転 trq.	押し込みトルクリミット	%	15 ~ 100

把持チェック:

- ベーシックメニューでは、選択した動作タイプ(外径把持, 内径把持)に合わせ、グリップエラー範囲が自動設定されます。
- アドバンスメニューでは、目標位置の更新時にグリップエラー範囲が自動設定されます。

押し込み運転:

- ベーシックメニューでは押し込み運転ができません。
- 押し込み量のパラメータは、目標到達位置に対し、どのくらい手前から押し込み開始するかを、開閉幅の%で設定します。
- 動作タイプを「内径把持」にすると、ハンドが開く側へ押し込み運転します。

6.3. UR スクリプト

UR スクリプト命令により、いくつかの拡張機能を使用できます。

以下の命令は、Polyscope の代入コマンドやスクリプトコマンドから呼び出せます。



ShinanoARH の UR スクリプトはハンドとの通信を伴います。
設置設定で RS485 通信を許可してください。(※3.1.2 項参照)

表 3 追加スクリプト一覧

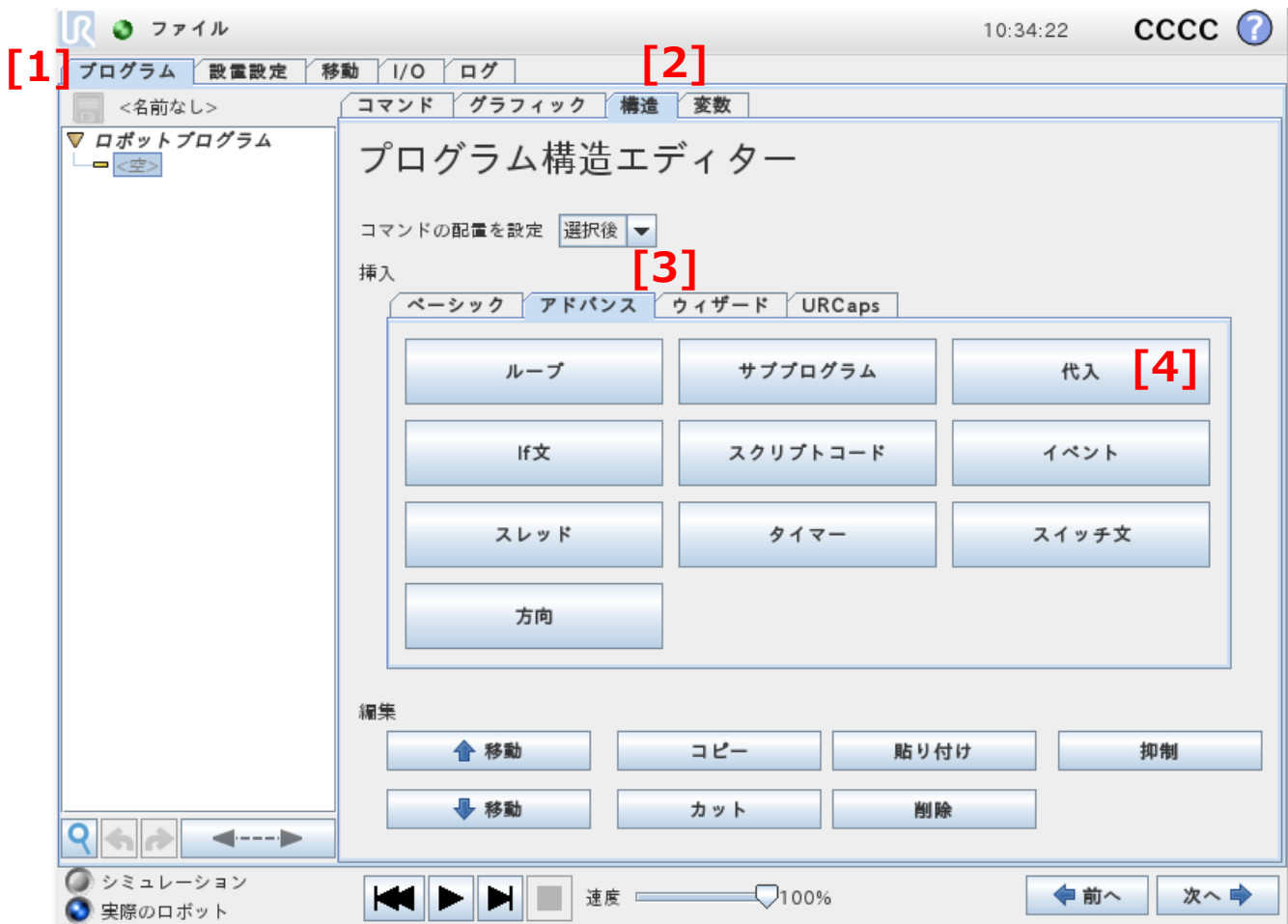
スクリプト名	機能
<code>arh_get_position()</code>	爪の現在位置を 0%~100%で返します。
<code>arh_get_torque()</code>	ハンドが把持している時の出力を%単位で返します。
<code>arh_get_temperature_c()</code>	ハンドの内部温度を摂氏[°C]で返します。
<code>arh_get_encoder()</code>	内蔵エンコーダのパルスカウント値を返します。 全閉全開のパルスカウントと比較してハンドが何度閉じているかを計算するのに使用します。
<code>arh_is_alarm()</code>	ハンドが異常を検知すると True を返します。 正常ならば False を返します。
<code>arh_is_grip_err()</code>	ハンドの爪先位置が把持チェックの範囲に到達したら True を返します。 ワークを掴んで把持チェック範囲外になれば False を返します。
<code>arh_is_ready()</code>	ハンドの動作が完了、または停止中は True を返します。位置決め運転、押し込み運転中は False を返します。
<code>arh_move(<pos>, <trq>)</code>	0%~100%の範囲でハンドの爪先位置と把持力を指定して動かします。 動作開始 OK ならば True, NG ならば False を返します。
<code>arh_move_and_wait(<pos>, <trq>)</code>	0%~100%の範囲でハンドの爪先位置と把持力を指定して動かし、ハンドの位置決めが完了するまで待機します。 動作開始 OK ならば True, NG ならば False を返します。
<code>arh_move_pos(<pos>)</code>	0%~100%の範囲でハンドの爪先位置を指定して動かします。 動作開始 OK ならば True, NG ならば False を返します。

スクリプト名	機能
arh_move_and_wait_pos(<pos>)	0%~100%の範囲でハンドの爪先位置を指定して動かし、ハンドの位置決めが完了するまで待機します。 動作開始 OK ならば True, NG ならば False を返します。
arh_stop()	動作中のハンドを減速停止させます。 減速開始 OK ならば True, 既に停止していれば False を返します。
arh_set_speed(<spd>)	ハンド動作前に、運転速度を 10%~400%の範囲で設定します。 引数が正常ならば True, 異常ならば False を返します。
arh_set_torque(<trq>)	ハンド動作前に、把持力を 15%~100%の範囲で設定します。 引数が正常ならば True, 異常ならば False を返します。
arh_set_grip_check_range(<pos_lower>, <pos_upper>)	ハンド動作前に、把持チェック位置範囲の下限と上限を%で設定します。 引数が正常ならば True, 異常ならば False を返します。 引数 lower>upper の場合も False を返します。
arh_set_push_in_data(<pos_width>, <spd>, <trq>)	ハンド動作前に、押し込み運転を有効とするパラメータを設定します。 押し込み量 pos_width は、目標位置に対して数%手前の位置で減速して押し込み開始するように調整します。 押し込み速度、押し込み力も%単位で設定します。 引数が正常ならば True, 異常ならば False を返します。 全ての引数が0の場合は押し込み運転を無効として True を返します。

6.3.1. スクリプト実行方法

スクリプトは、「代入」コマンドもしくは「スクリプトコード」コマンドで実行します。

<代入コマンドの例>



1. プログラム画面を開きます。
2. “構造”をタップします。
3. プログラムに挿入する項目から“アドバンス”をタップします。
4. “代入”をタップします。

<代入コマンドの例(続き)>

ファイル 10:38:40 CCCC ?

プログラム 設定設定 移動 I/O ログ

<名前なし> コマンド グラフィック 構造 変数

ロボットプログラム
var_1:=...

代入 **[5]** 入力元 式

選択した変数に式の値を代入する。

数 式

var_1 := **[6]**

名称の変更

Input
<入力> True (HI) False (LO) Esc ← バックスペース

Output
<出力> and or xor not 7 8 9 -

Variable
<変数> = ≠ () < > / * 4 5 6

Pose
<ポーズ> - " [] ≤ ≥ , + 1 2 3 送信

Function
<機能> **[7]** ABC ◀ ▶ 0 .

5. “コマンド”をタップします。
6. 式の入力欄をタップします。
7. Function をタップします。

<代入コマンドの例(続き)>



8. 機能リストから使用したいスクリプトコードを選択します。
9. “送信”ボタンをタップします。

以上の操作で、ハンドのスクリプトコードをプログラムに追加し、結果が変数 var_1 に記憶されます。

6.3.2. スクリプト使用例

スクリプトを用いたプログラム例は以下の通りです。

```

1  ▼ BeforeStart
2  └─ arh_set_speed(100)
3  └─ arh_set_torque(80)
4  └─ arh_set_grip_check_range(90, 100)
5  └─ arh_set_push_in(0, 0, 0)
6  ▼ Robot Program
7  └─ MoveJ
8     └─ Waypoint_1
9     └─ arh_move_and_wait_pos(100)
10    └─ pos:=arh_get_position()
11    └─ gerr:=arh_is_grip_error()
12    └─ If gerr ≠ True
13       └─ Popup: gripping failed
14    └─ If pos < 60
15       └─ Popup: work is different
16    └─ MoveJ
17       └─ Waypoint_2
18    └─ arh_move_and_wait_pos(0)

```

1. ロボットプログラム実行前に、運転速度や把持力、把持チェックなどのパラメータを設定しておく（2～5 行目）
2. MoveJ でウェイポイント 1 へ移動（7～8 行目）
3. ハンドを 100%の位置へ動かし、位置決めが完了するまで待機（9 行目）
4. ハンド動作後の現在位置を変数 pos に、把持チェックの結果を変数 gerr に代入（10～11 行目）
5. 変数 gerr が True ならば「把持失敗」のポップアップを表示し、変数 pos が 60%未満ならば「ワークが違う」のポップアップを表示（12～15 行目）
6. MoveJ でウェイポイント 2 へ移動（16～17 行目）
7. ハンドを 0%の位置へ動かし、位置決めが完了するまで待機（18 行目）

7. 補足事項

7.1. ARH Tool I/O 入出力指定

インストレーション画面の”ツール I/O”タブは、ハンドの入出力信号を選択します。

表 4 入力信号一覧

信号名	機能	URCap 対応
SRV_ON	ハンドの通電と無通電を切り替える信号です。	—
HOME	開閉位置検出動作を開始する信号です。	—
DRIVE	グリップを開始する信号です。	—
STOP	グリップを中断する信号です。	—
ALM_RST	異常停止状態を復帰する信号です。	—
SEL0		○
SEL1	運転番号を選択してグリップを開始する信号です。	○
SEL2		○

表 5 出力信号一覧

信号名	機能	URCap 対応
SRV	ハンド通電中に信号を出力します。	—
READY	ハンドが通電かつ待機中に信号を出力します。	—
ALARM	異常が発生したら信号の出力を止めます。	○
GRIP_ERR	把持失敗したら信号を出力します。	○
AREA	ハンドの爪先がエリア範囲内であれば信号を出力します。	—

インストレーション画面の”ロード”ボタンをタップすると、ハンドの内部設定値を取得できます。

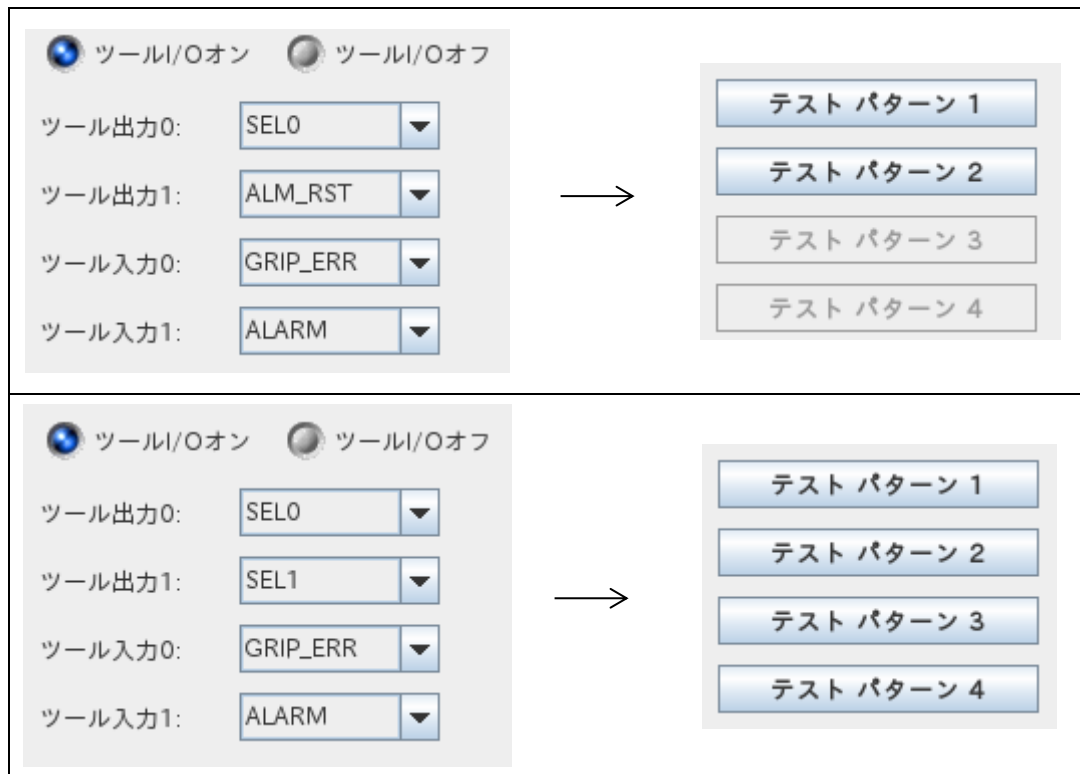
但し、通信不可であれば取得できませんので、事前にハンドの入出力信号設定を確認してから機能を選択してください。

また、”セーブ”ボタンをタップすると、画面上の設定をハンド側に上書きできます。

入出力信号の切り替えは電源起動後から有効になりますので、”セーブ”ボタンをタップしたら、ロボットハンドおよび PolyScope を再起動してください。

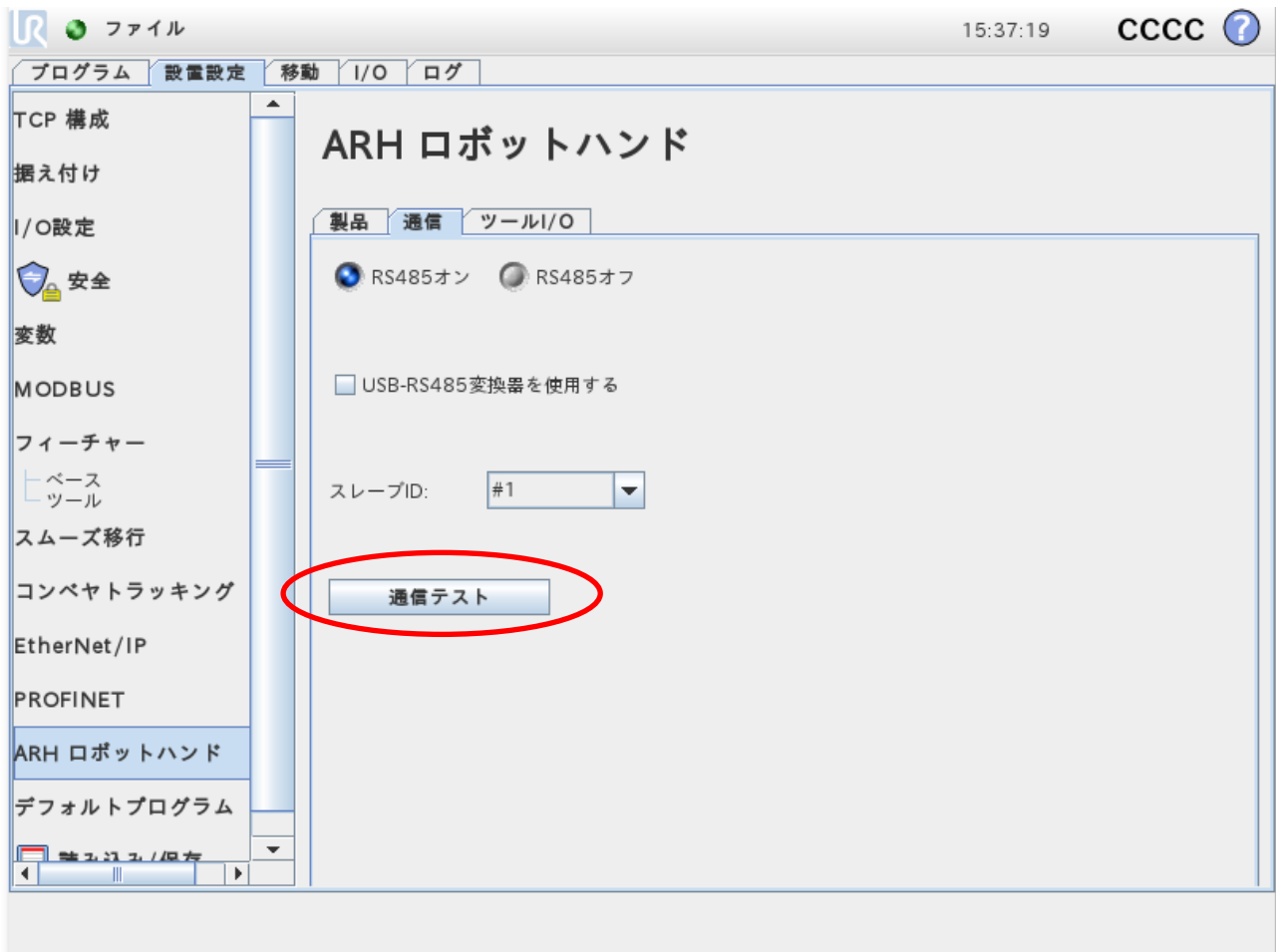
7.2. ARH Tool I/O グリップパターン指定

グリップパターンのパターン 3 と 4 は、インストレーション画面の”ツール出力 1”が SEL* の場合のみ有効です。



7.3. 設置設定における通信チェック

設置設定画面の通信タブにある”通信テスト”ボタンをタップすることで、通信テストが行えます。



“通信テスト”をタップすると、ハンドに状態照会の通信コマンドを送信して結果を表示します。

通信テストの結果、表示されるメッセージは以下の通りです。

表 6 通信テスト出力メッセージ

エラー: 通信失敗	通信が失敗しています。 ロボットとの配線や SlaveID の設定を確認してください。
ハンドステータス = handoff	ハンドが無通電状態です。 無通電時はハンドへの開閉動作を受け付けません。
ハンドステータス = alarm	ハンドが異常停止しています。 異常停止時はハンドへの開閉動作を受け付けません。
ハンドステータス = notHome	原点復帰運転が未実施です。 原点復帰運転により全閉位置と全開位置を検出する必要があります。
ハンドステータス = handBusy	ハンドがビジー状態です。 開閉動作実行中はビジー状態になります。
ハンドステータス = gripErr	ハンドが把持失敗状態です。 ハンドの爪先が把持チェック範囲内に位置しています。
ハンドステータス = handReady	ハンドがレディ状態です。 開閉動作指令を受け付ける状態になっています。

8. トラブルシューティング

8.1. 把持失敗によるプログラム停止

プログラム実行中に把持失敗を検知すると、以下のポップアップ画面が表示されます。



把持失敗を検知したら速やかにプログラムを停止させ、ワークの位置を見直すなどの修正を行ってください。
把持失敗としないタイミングで停止した場合は、不要にもかかわらず把持チェックのチェックボックスがオンになっていないか御確認ください。

8.2. アラーム検知によるプログラム停止

プログラム実行中にハンドの異常停止を検知すると、以下のポップアップ画面が表示されます。



アラームを検知したら速やかにプログラムを停止させ、アラームの要因を取り除いてください。
また、ハンドへの電源供給がされていない場合はハンドの ALARM 信号出力がオフ固定となるため、ロボットがアラームを検知します。ハンドに電源が供給されているか御確認ください。

8.3. 通信失敗によるプログラム停止

プログラム実行中にハンドとの通信が途絶えると、以下のポップアップ画面が表示されます。



通信が失敗したら速やかにプログラムを停止させ、設置設定で以下の点を御確認ください。

- USB-RS485 変換器のチェックの有無
- スレーブ ID がハンドの内部設定値と一致しているか
- ツール IO の通信設定が正しくされているか(3.2.2 項参照)

設定に問題が無ければロボットの電源を落とし、ロボットとハンドが正しく接続されているか確認してください。

9. URCaps リリースノート

9.1. ShinanoARH version 1.1.0

仕様変更

- 日本語表記への対応(※Polyscope の言語設定を日本語にすること)
- UR スクリプトコードの追加
- “Enter Position:”→”Position :”などの英語表記を修正
- ツール入出力コマンド画面のハンド画像を変更
- 設置設定のモデル選択で”ARH350A”, ”ARH250A”を追加し、選択時は速度の上限を 100%に制限

バグ修正

- 通信コマンド画面でメニューをアドバンスからベーシックに戻した際に、動作速度のパラメータが 100%に戻らない事象を修正

10. 改版履歴

日付	版	内容
2019/10/08	0.00	初版作成
2019/10/18	0.01	1.2 節 CB シリーズコントローラボックスの結線図を修正 3.1 節 通信設定とツール I/O 設定に分割 3.3 節 ツール中心点(TCP)を追加 5.1 節 見出しを RobotHand I/O に変更 5.1 節 コマンド画面を更新 5.1 節 把持チェックの説明文を修正 5.2 節 見出しを RobotHand Communication に変更 5.2 節 通信操作にベーシックメニューとアドバンスメニューを追加 5.2.1 項 TestGrip 操作の注意書きを追加 5.2.1 項 把持チェックの説明文を更新 5.2.2 項 把持チェックの説明文を更新 5.2.2 項 押し込み運転の説明文を更新 5.2.3 項 パラメータについての補足説明を追加
2019/11/01	0.02	1.1 節 取り付け方法のイラストを追加 2 章 インストール画面を更新 3 章 “ASPINA RobotHand”を”ARH RobotHand”に変更 3.1.1 項 ツール I/O 設定画面を更新 3.1.2 項 RS485 設定画面を更新 4 章 構造画面を更新 5.1 節 “RobotHand I/O”を”ARH Tool I/O”に変更 5.1 節 ドライブパターン番号をグリッパパターン番号に修正 5.2 節 “RobotHand Communication”を”ARH Communication”に変更 6 章 補足説明の追加
2019/11/07	0.03	3.2 節 I/O 設定を CB シリーズと e シリーズに分けて記載
2019/11/07	1.00	リビジョンを 1.00 に更新
2019/11/11	1.00.01	7 章 トラブルシューティングを追加
2019/11/29	1.00.02	2 章 URCaps インストール注意点を追記 3.3.2 項 e シリーズの TCP 設定を追記 6 章に仕様を追加し、補足説明を 7 章に変更 6.1 節 URCaps 仕様を追加 7.3 節 通信チェックの説明を追加
2020/02/07	1.00.03	スクリーンショットを日本語表記に変更 6.3 節 UR スクリプトを追加 9 章 URCaps リリースノートの追加 10 章 改版履歴を移動

日付	版	内容
2020/03/03	1.00.04	スクリーンショットの更新 6.3 節 UR スクリプト追加一覧を作成 6.3.1 項 スクリプト実行方法を追加 6.3.2 項 スクリプト使用例を追加
2020/03/12	1.00.05	5.1 節 「pattern」を「パターン」に変更 7.2 節 テスト選択の画像を更新
2020/04/13	1.00.06	6.3.1 項 スクリプト実行方法に説明文を記載
2020/04/24	1.00.07	1.2.2 項 e シリーズの RS485 通信について追記
2020/05/27	1.01	リビジョンを 1.01 に更新