

ブロックプログラミング環境 機能説明書

通信の仕様について

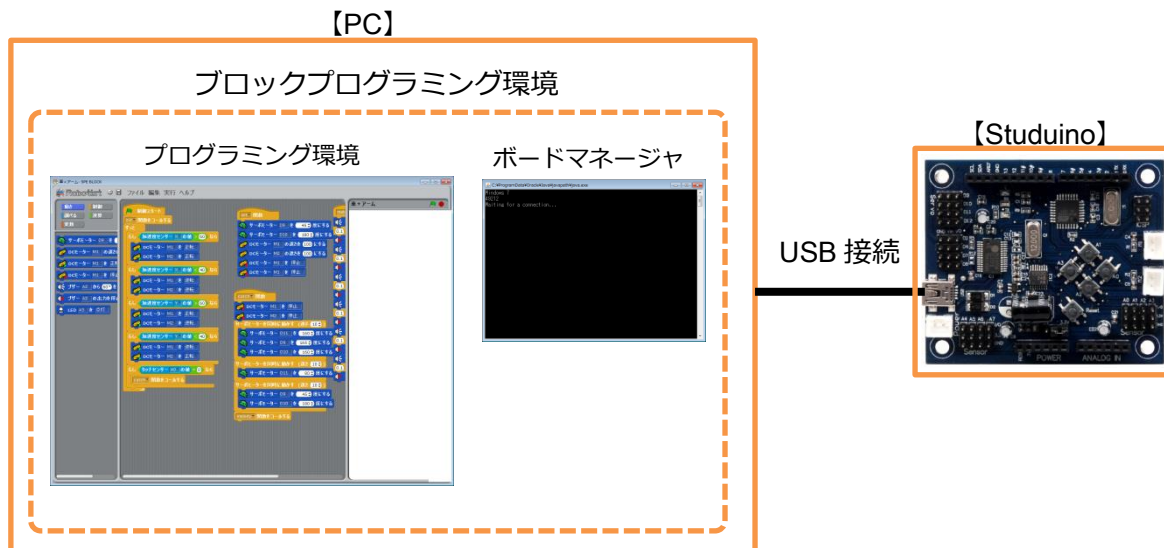
2014/11/11



本ドキュメントは、ブロックプログラミング環境の機能(テストモード等)を熟知した方が対象となっております。ブロックプログラミング環境の機能についての詳細は、別紙「Studuino プログラミング環境 取扱説明書」を参照してください。

1. システムの構成

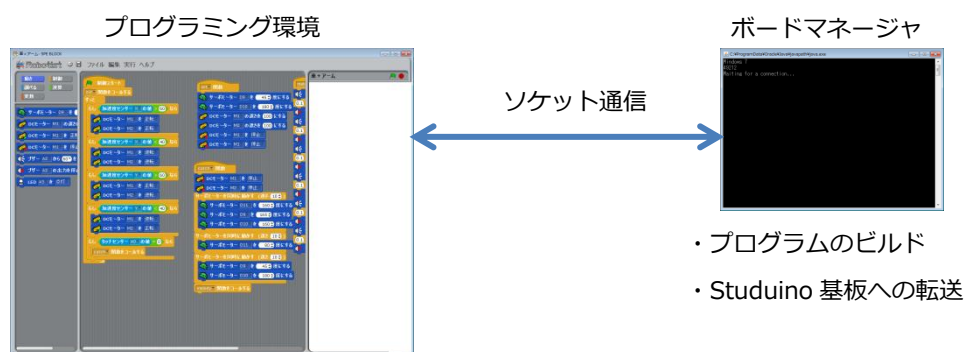
ブロックプログラミング環境は、プログラミング環境とボードマネージャから成ります。



プログラミング環境は、Studuino 基板で動作するロボット制御プログラムを作成する環境です。ボードマネージャはプログラミング環境と通信して、プログラミング環境で作成したプログラムのビルドと Studuino 基板へのスケッチのアップロードを行います。

1.1. プログラミング環境とボードマネージャの通信について

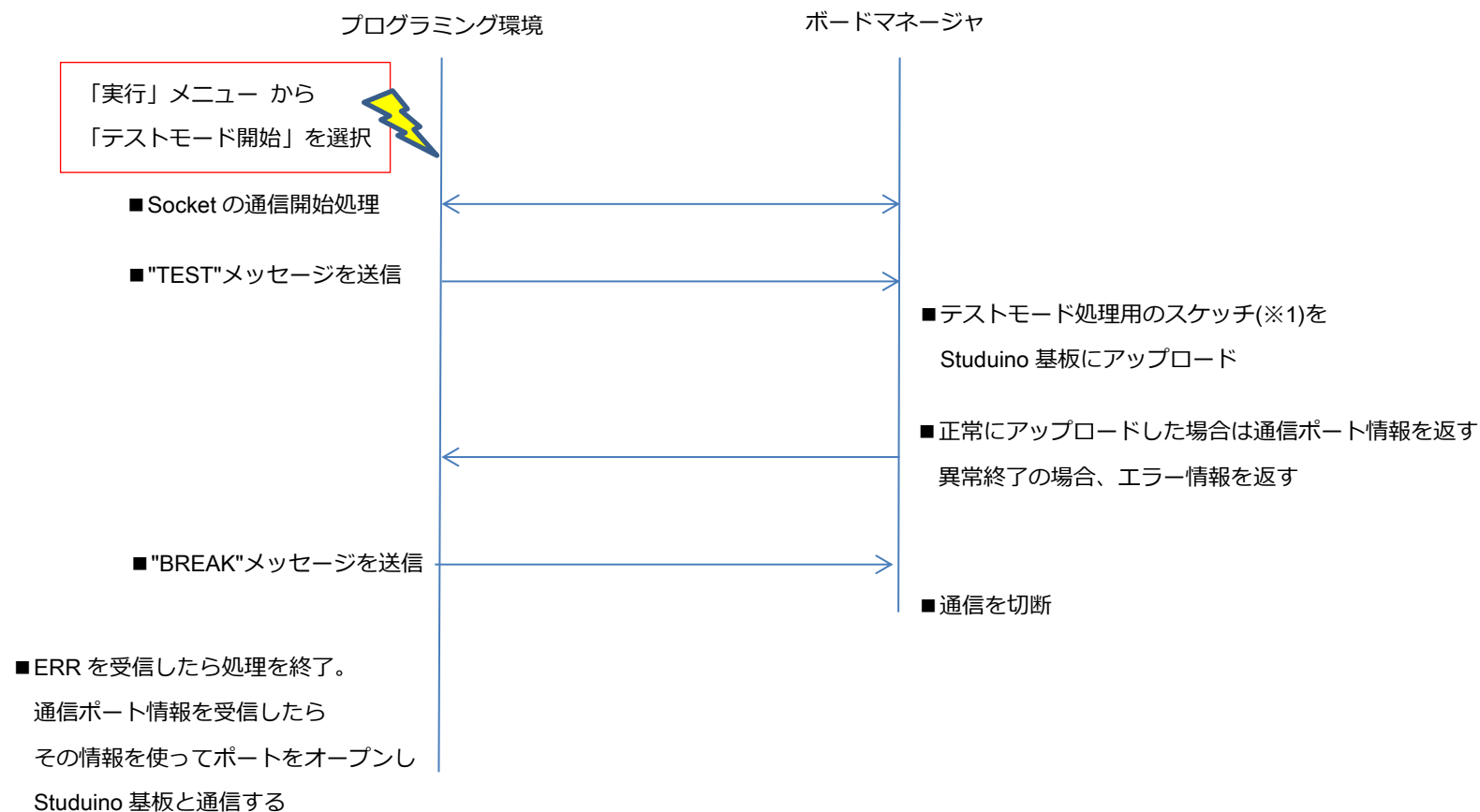
プログラミング環境とボードマネージャはローカルホストでソケット通信をしてメッセージをやりとりします。



プログラミング環境は、テストモード等のイベントが発生すると、ボードマネージャにメッセージを送信し、処理を依頼します。プログラミング環境からボードマネージャへのメッセージは単純な文字列です。ボードマネージャは、プログラミング環境から受信したメッセージに対して処理を行い、必要であれば返信メッセージを送信します。

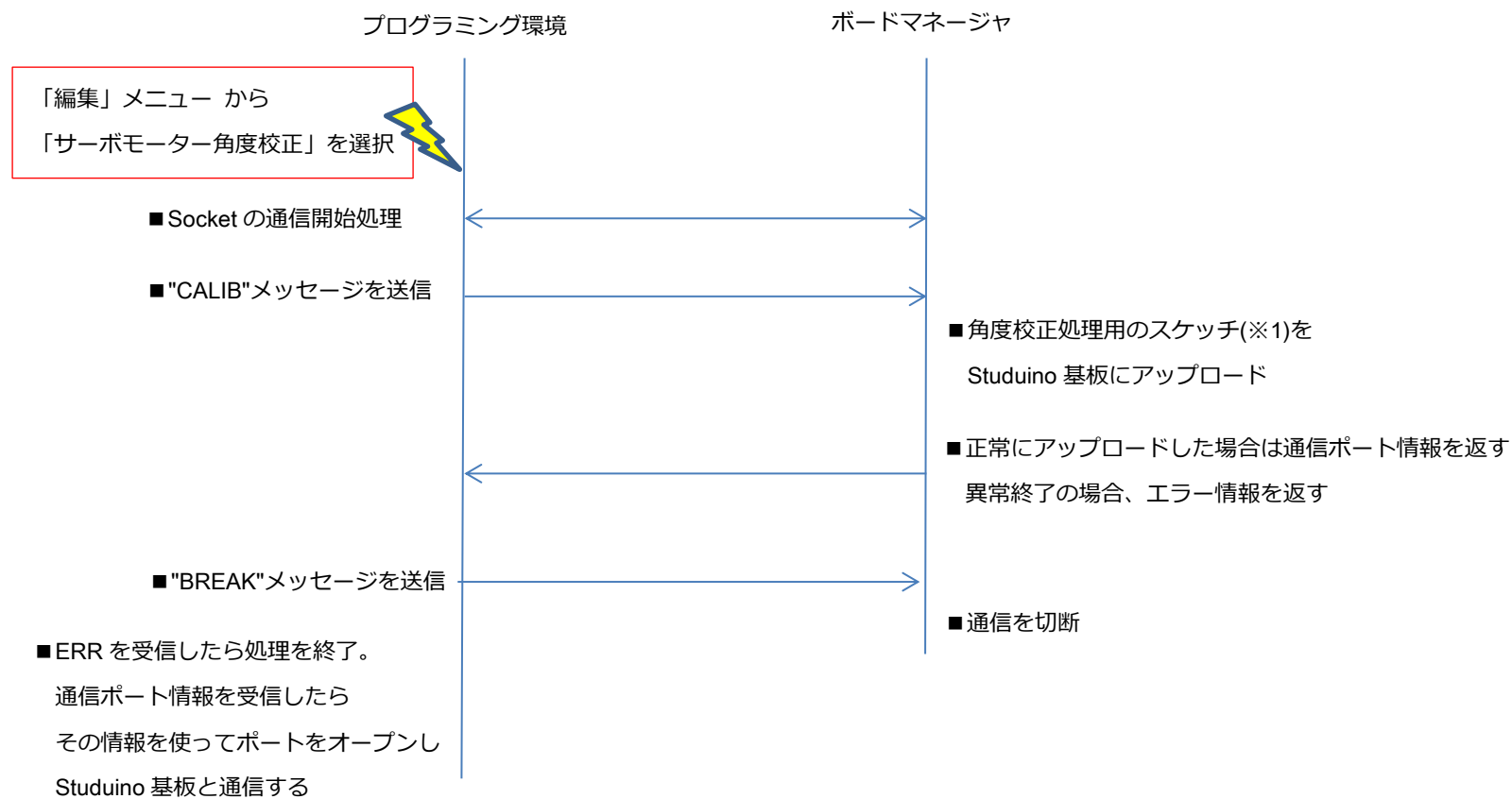
以下にプログラミング環境-ボードマネージャ間の通信のシーケンスを記します。

● テストモード開始時の通信



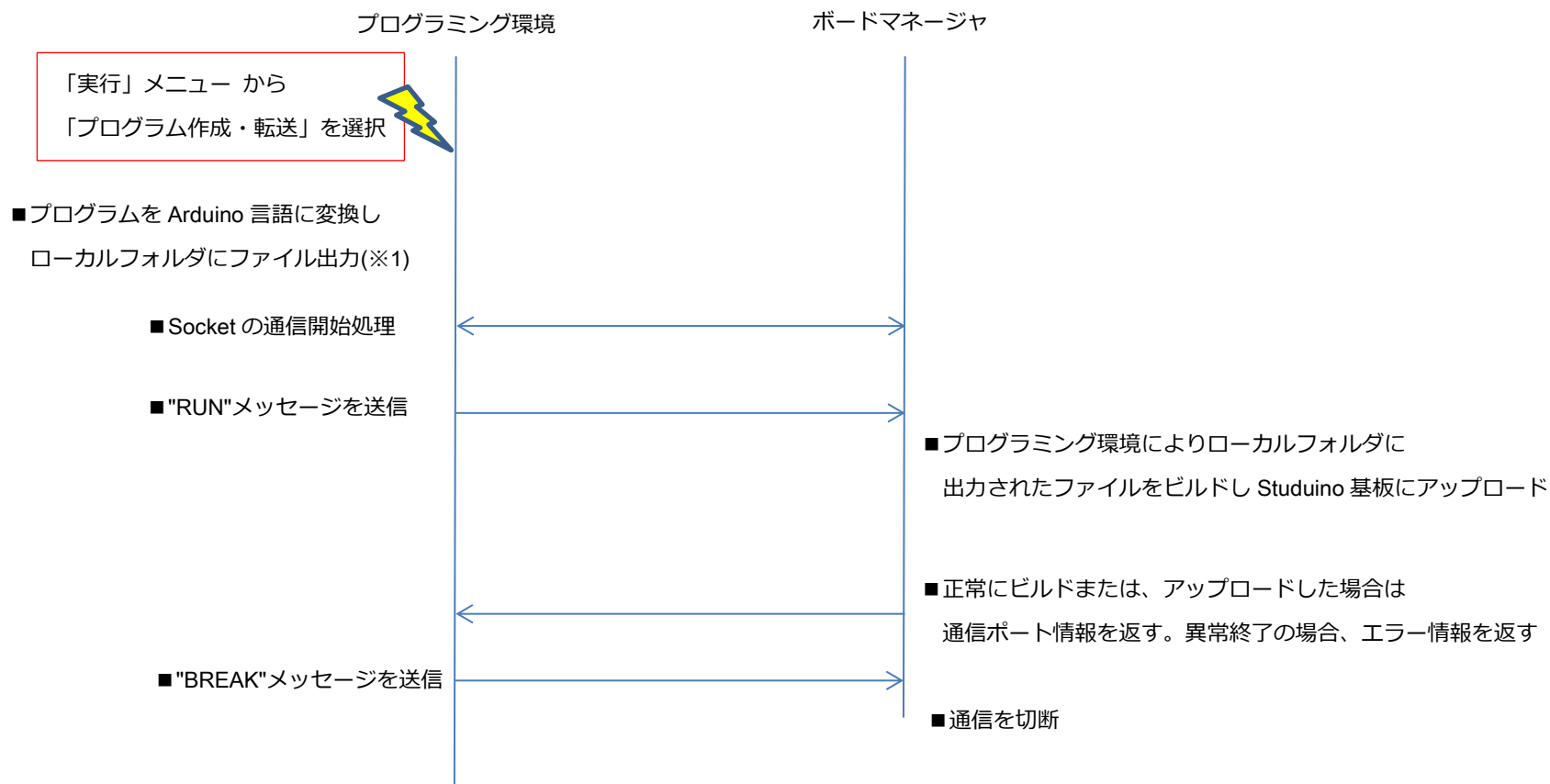
(※1)スケッチは、bpe/ar.cpp.hex になります。

- サーボモーター角度校正時の通信



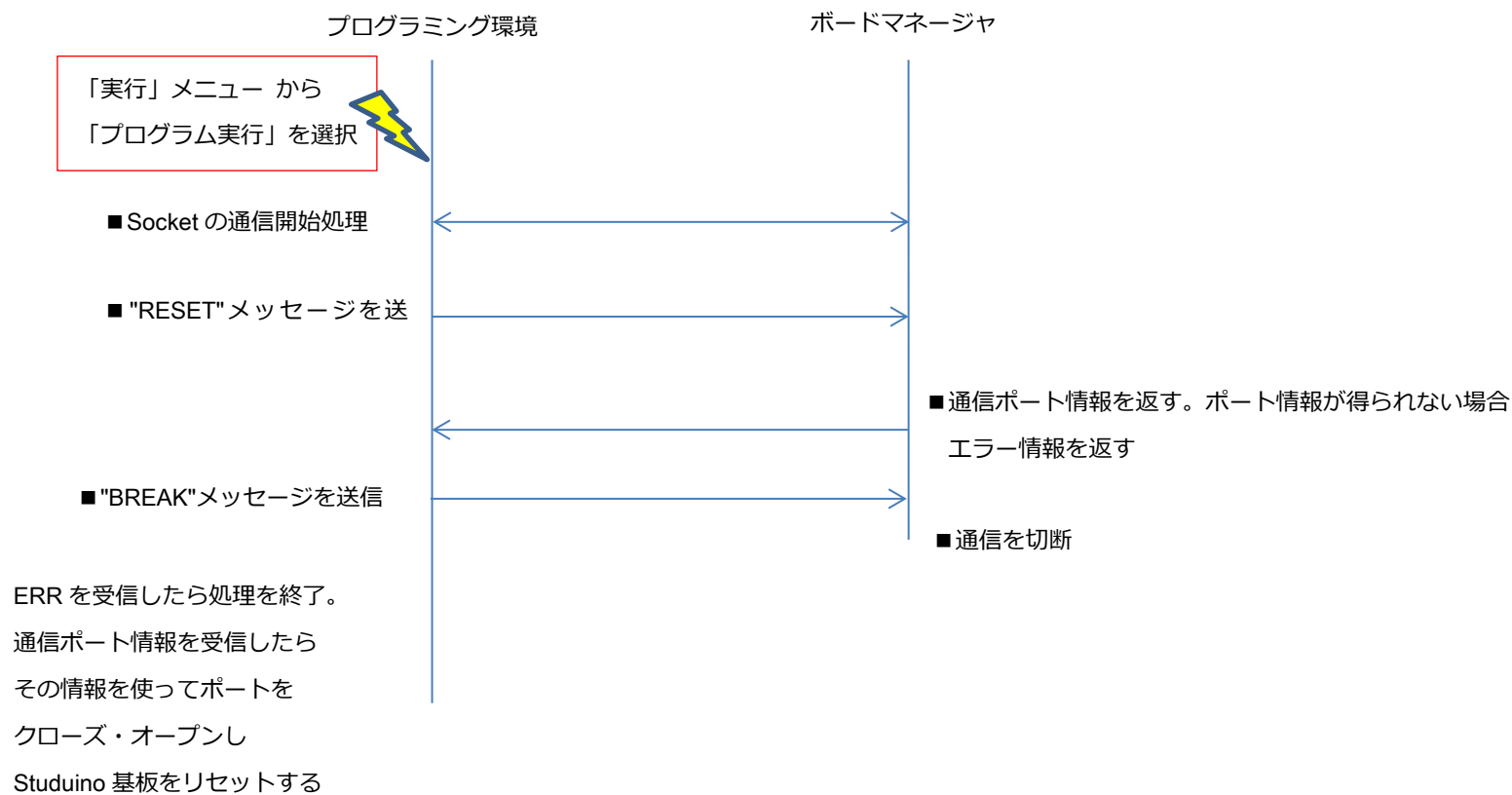
- (※1)スケッチは、bpe/ar.cpp.hex になります。

- プログラム作成・転送時の通信



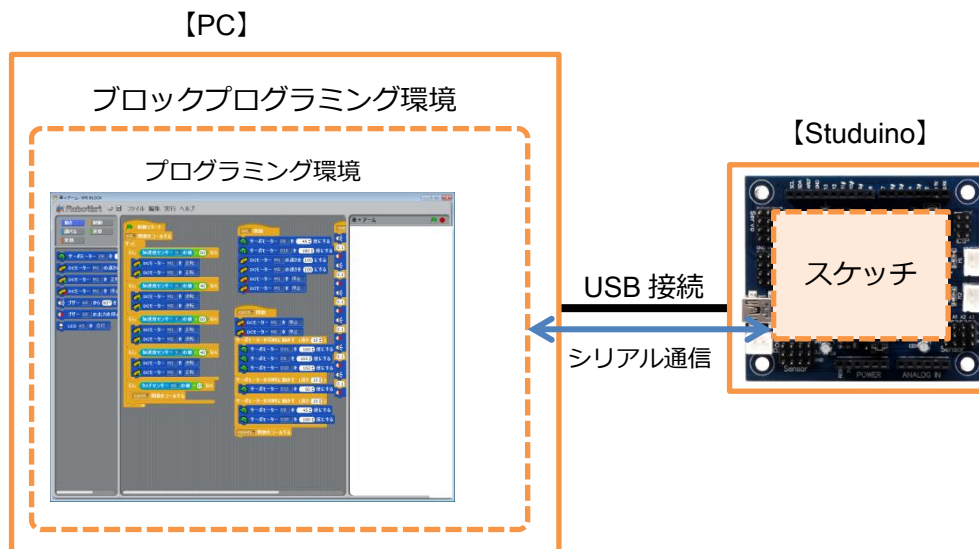
(※1)ローカルフォルダは、bpe/user フォルダになります。

- プログラム実行時の通信



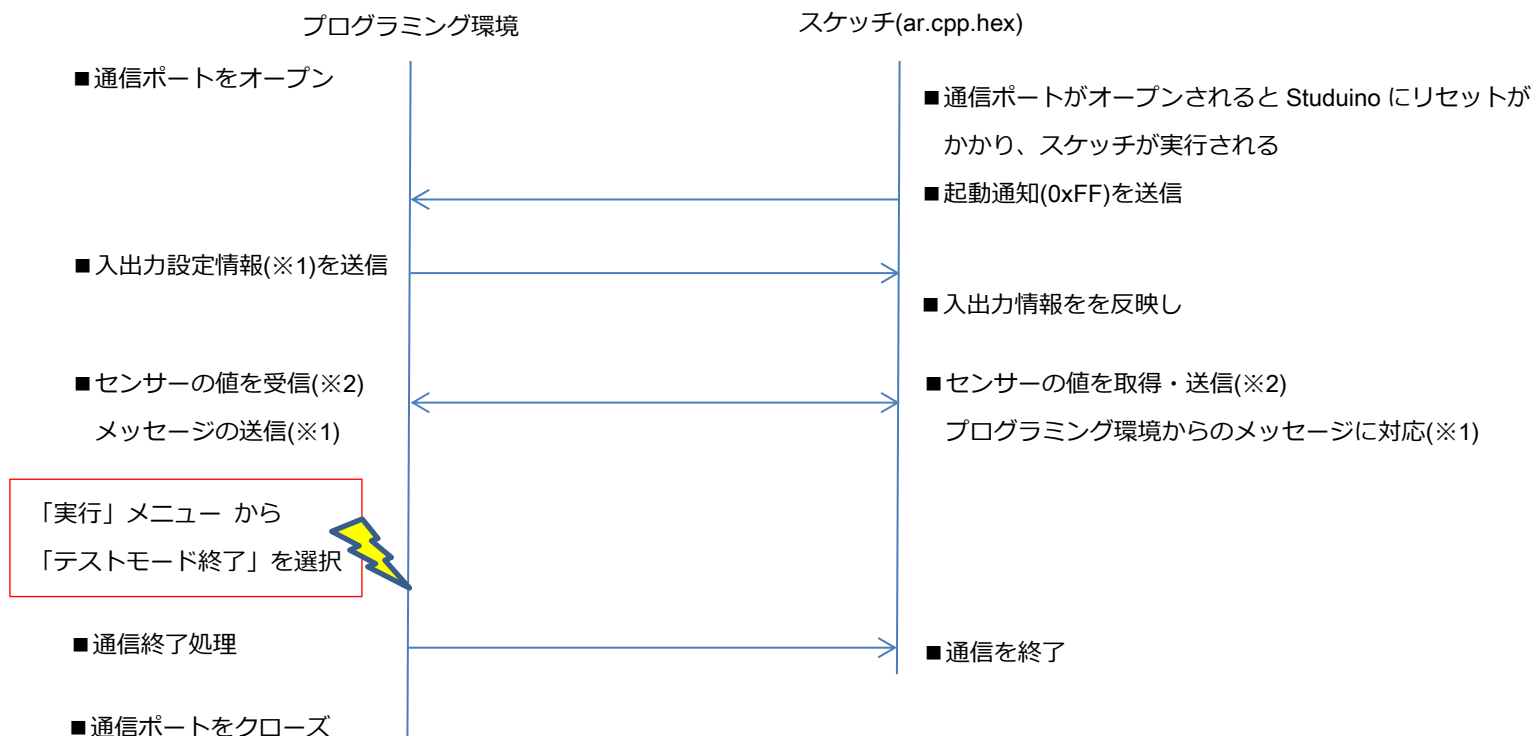
1.2. プログラミング環境と Studuino 基板の通信について

プログラミング環境は、テストモード時とサーボモーター角度校正時に、ボードマネージャによって Studuino 基板にアップロードされたスケッチとシリアル通信して、リアルタイムに Studuino 基板に接続されたアクチュエータを動かしたり、センサーの値を表示したりします。



1.3. テストモード時の通信について

テストモード時、プログラミング環境はスケッチと下記ように通信します。



(※1)メッセージの詳細は、1.3.1 を参照。

(※2)メッセージの詳細は、1.3.2 を参照。

1.3.1.プログラミング環境から Studuino 基板への通信について

プログラミング環境からスケッチへの送信データは、1 メッセージ 4Byte で、各 1Byte の先頭 2 ビット(bit7,6)でデータの送信順を表しています。

7	6	5	4	3	2	1	0
1	1	ID					

7	6	5	4	3	2	1	0
1	0	Data(high)					

7	6	5	4	3	2	1	0
0	1	Data(middle)					

7	6	5	4	3	2	1	0
0	0	Data(low)					

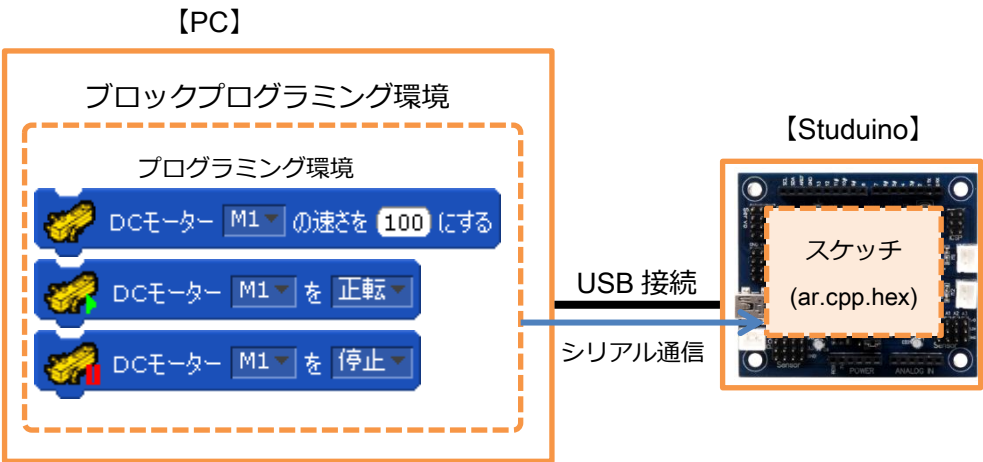
● DC モーター

DC モーター情報は、下記のフォーマットで送信します。

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	1	ID						1	0		pin				action
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	1							0	0	Setting					

名前	設定内容	設定値	意味
ID	パーツ ID	0	DC モーターへの設定
pin	接続コネクタ	b'0	M1 コネクタ
		b'1	M2 コネクタ
action	処理の指定	b'00	ON
		b'01	OFF
		b'10	パワー
setting	処理の設定	action で ON を指定した場合、b'00 で正転、b'01 で反転	
		action で OFF を指定した場合、b'00 で停止、b'01 で惰性	
		action でパワーを指定した場合、設定値(0~100)	

プログラミング環境上の DC モーターブロックに対する情報を送信します。



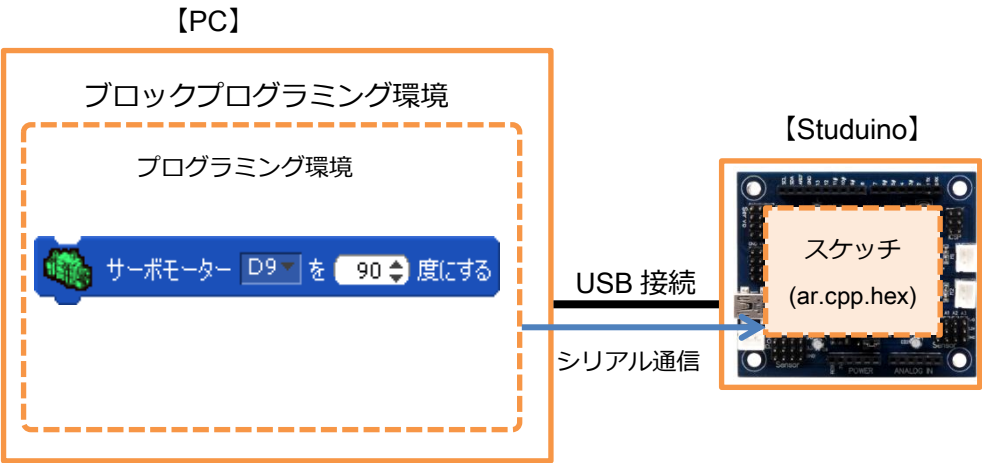
● サーボモーター

サーボモーター情報は、下記のフォーマットで送信します。

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	1	ID						1	0				pin		
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	1							0	0	degree					

名前	設定内容	設定値	意味
ID	パーツ ID	1	サーボモーターへの設定
pin	接続コネクタ	b'000	D2 コネクタ
		b'001	D4 コネクタ
		b'010	D7 コネクタ
		b'011	D8 コネクタ
		b'100	D9 コネクタ
		b'101	D10 コネクタ
		b'110	D11 コネクタ
		b'111	D12 コネクタ
degree	角度	0～180	pin で指定したサーボの角度

プログラミング環境上のサーボモーターブロックに対する情報を送信します。



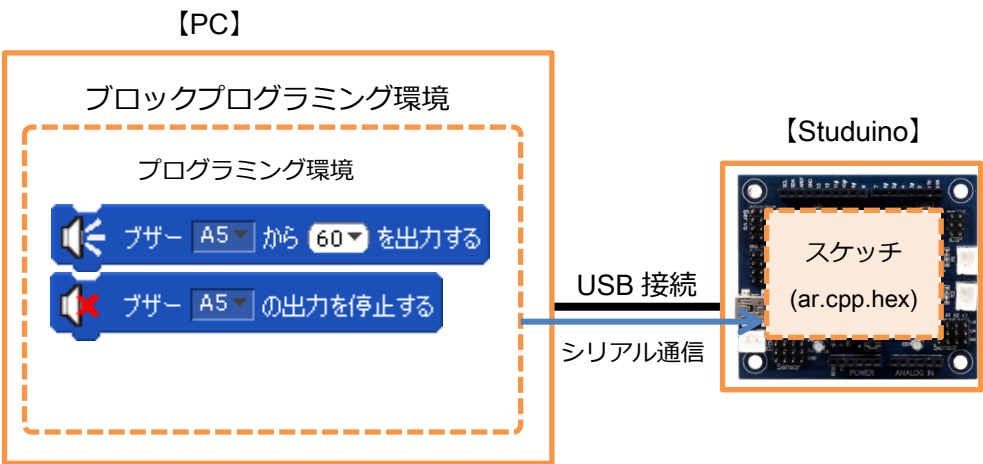
● ブザー

ブザー情報は、下記のフォーマットで送信します。

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	1	2						1	0				pin		
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	1		action					0	0	freq					

名前	設定内容	設定値	意味
ID	パーツ ID	2	ブザーへの設定
pin	接続コネクタ	b'000	A0 コネクタ
		b'001	A1 コネクタ
		b'010	A2 コネクタ
		b'011	A3 コネクタ
		b'100	A4 コネクタ
		b'101	A5 コネクタ
action	処理	b'0	OFF
		b'1	ON
freq	処理の設定	action で ON を指定した場合、音の高さ(0~100)	

プログラミング環境上のブザーブロックに対する情報を送信します。



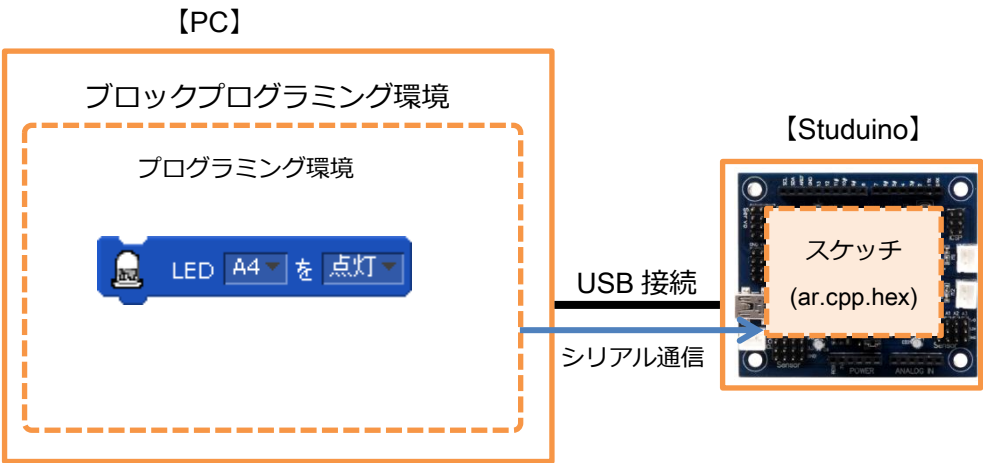
● LED

LED 情報は、下記のフォーマットで送信します。

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	1	3						1	0				pin		
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	1		action					0	0						

名前	設定内容	設定値	意味
ID	パーツ ID	3	LED への設定
pin	接続コネクタ	b'000	A0 コネクタ
		b'001	A1 コネクタ
		b'010	A2 コネクタ
		b'011	A3 コネクタ
		b'100	A4 コネクタ
		b'101	A5 コネクタ
action	処理	b'0	OFF
		b'1	ON

プログラミング環境上の LED ブロックに対する情報を送信します。



● 入出力設定

Studuino 基板に接続しているパーツ情報を送信します。

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	1	4						1	0		pin				

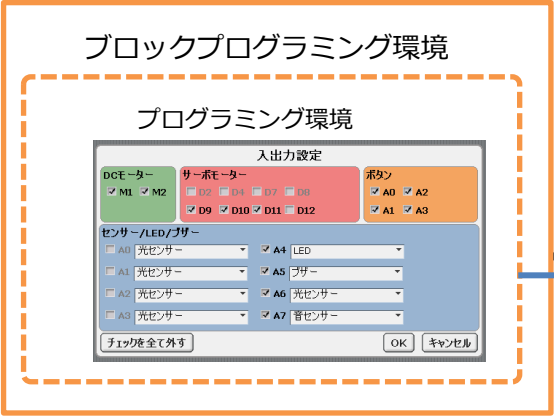
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	1							0	0		parts				

名前	設定内容	設定値	意味
ID	処理 ID	4	入出力設定処理
pin	接続コネクタ	0	M1 コネクタ
		1	M2 コネクタ
		2	D2 コネクタ
		3	D4 コネクタ
		4	D7 コネクタ
		5	D8 コネクタ
		6	D9 コネクタ
		7	D10 コネクタ
		8	D11 コネクタ
		9	D12 コネクタ
		10	A0 コネクタ
		11	A1 コネクタ
		12	A2 コネクタ
		13	A3 コネクタ
		14	A4 コネクタ
		15	A5 コネクタ
		16	A6 コネクタ
		17	A7 コネクタ
parts	接続パーツ	0x01	DC モーター
		0x02	サーボモーター
		0x03	LED
		0x04	ブザー

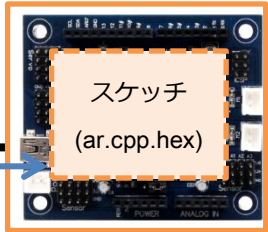
		0x10	光センサー
		0x11	タッチセンサー
		0x12	音センサー
		0x13	赤外線フォトリフレクタ
		0x14	加速度センサー
		0x15	ボタン

プログラミング環境で設定された入出力情報を送信します。

【PC】



【Studuino】



USB 接続

シリアル通信

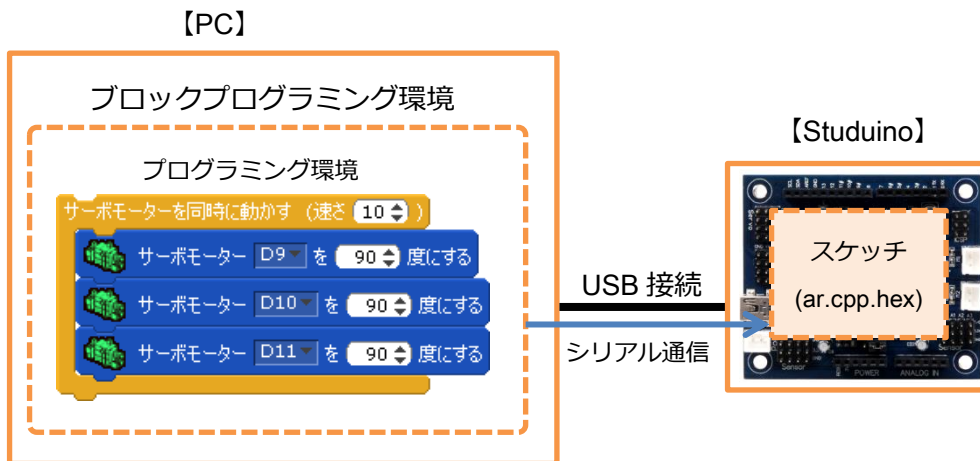
● サーボモーター同期処理

Studuino 基板に接続しているパーツ情報を送信します。

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	1	11						1	0						

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	1							0	0		speed				

名前	設定内容	設定値	意味
ID	処理	11	サーボモーター同期処理
bundle	同期するサーボモーター情報	b'0	先頭
		b'1	終端
speed	速度	0~20	サーボモーターを動かす速さ



上図の場合、最初に「サーボモーターを同時に動かす」ブロック情報を開始(bundle:b'0)で送信します。次に「サーボモーター」ブロック情報を送信します。最後に「サーボモーターを同時に動かす」ブロック情報を終端(bundle:b'1)で送信します。

Studuino 側のスケッチは、最後の情報を受け取ると、指定されたサーボモーターを同期させて動かします。

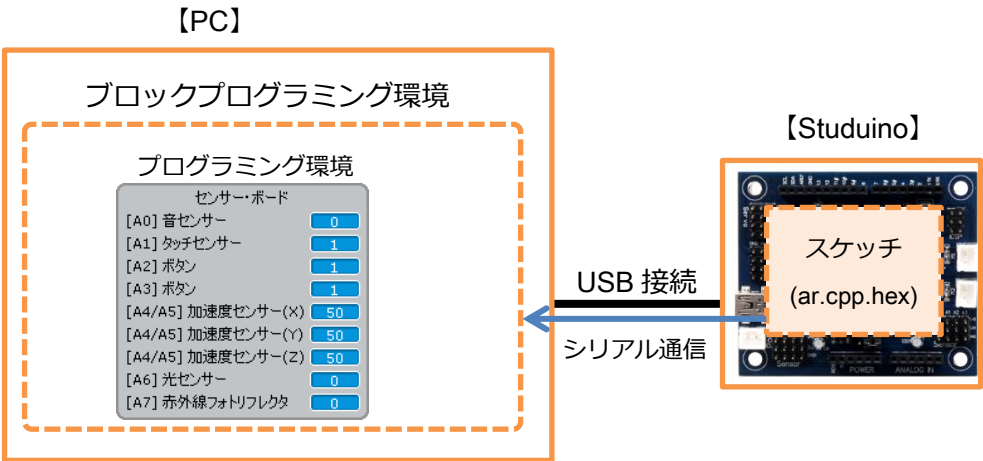
1.3.2.Studuino 基板からプログラミング環境への通信について

スケッチからプログラミング環境への送信データは、1 メッセージ 2Byte で、各 1Byte の先頭 1 ビット(bit7)でデータの送信順を表しています。

7	6	5	4	3	2	1	0
1	ID						

7	6	5	4	3	2	1	0
0	data						

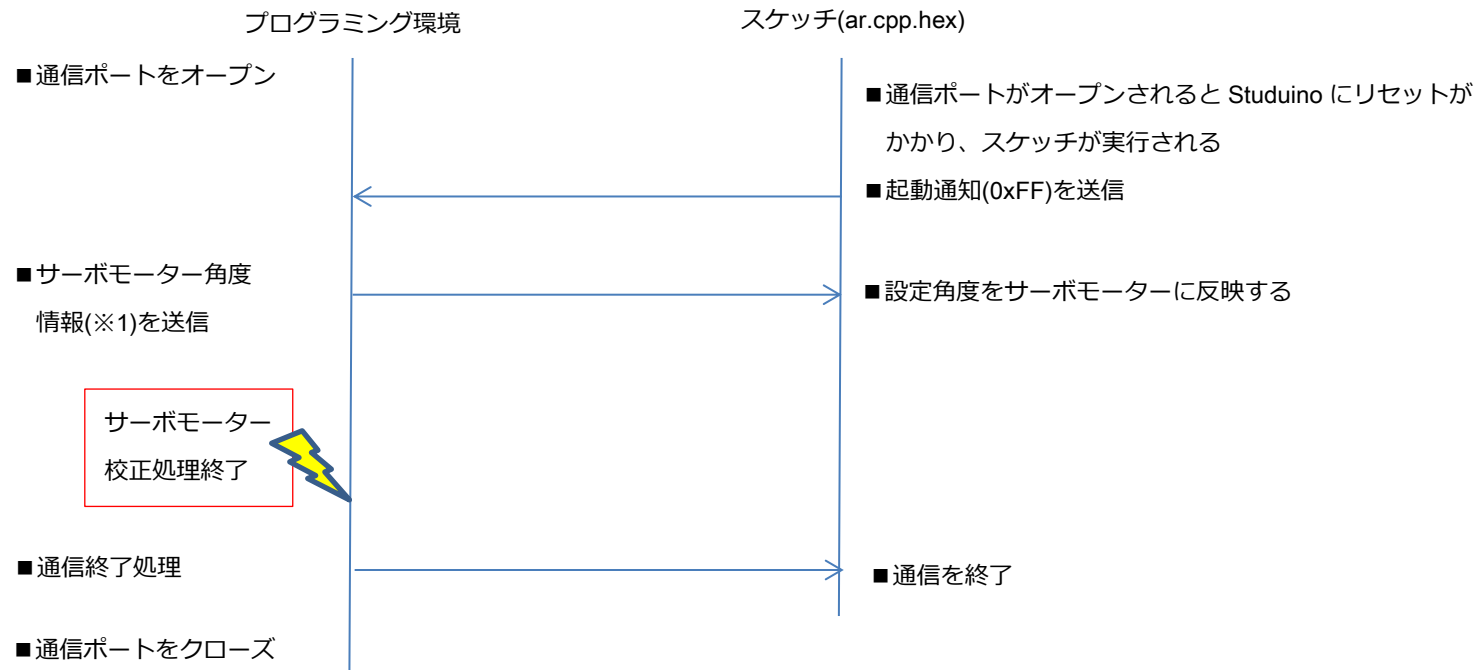
名前	設定内容	設定値	意味
ID	番号	0～13	番号
data	センサー値	data	0～100



スケッチからのセンサー値の送信前に、プログラミング環境から入出力設定情報を送信します。プログラミング環境とスケッチ間のポート情報(Studuino に接続されたパーツ情報)の整合性が取れた状態で、本メッセージを送る仕様になっています。

1.4. サーボモーター校正時の通信について

サーボモーター校正時、プログラミング環境はスケッチと下記ように通信します。



(※1)メッセージの詳細は、1.4.1 を参照。

1.4.1.プログラミング環境から Studuino 基板への通信について

サーボモーター構成情報は、下記のフォーマットで送信します。

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	1	ID						1	0					pin	
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	1							0	0	degree					

名前	設定内容	設定値	意味
ID	パーツ ID	1	サーボモーターへの設定
pin	接続コネクタ	b'000	D2 コネクタ
		b'001	D4 コネクタ
		b'010	D7 コネクタ
		b'011	D8 コネクタ
		b'100	D9 コネクタ
		b'101	D10 コネクタ
		b'110	D11 コネクタ
		b'111	D12 コネクタ
degree	角度	90±α	サーボモーター角度構成ボードで設定された値(α)を 90 度に付加した値

