

# b-CAP サーバ&クライアント

## ユーザーズガイド

Version 1.0.0

January 30, 2015

【備考】



この作品は[クリエイティブ・コモンズ・表示 4.0 国際・ライセンス](https://creativecommons.org/licenses/by/4.0/)で提供されています。

**【改版履歴】**

日付	版数	内容
2015-1-30	1.0.0	初版作成.

## 目次

1. はじめに .....	4
2. b-CAP サーバ .....	5
2.1. 関数仕様.....	5
2.1.1. bCap_Open_Server.....	5
2.1.2. bCap_Close_Server .....	6
2.1.3. bCap_SetCallFunc .....	6
3. b-CAP クライアント .....	7
3.1. 関数仕様.....	7
3.1.1. bCap_Open_Client.....	7
3.1.2. bCap_Close_Client.....	8
3.1.3. bCap_SetTimeout.....	8
3.1.4. bCap_GetTimeout .....	8
3.1.5. bCap_SetRetry .....	9
3.1.6. bCap_GetRetry.....	9
4. b-CAP 通信の制御フロー .....	10
付録 A. b-CAP 関数 ID とコールバック関数.....	11
付録 A.1. ハンドル番号.....	18

## 1. はじめに

本書は, b-CAP サーバおよび b-CAP クライアントを使用した開発を行うユーザ向けの取扱説明書です.

## 2. b-CAP サーバ

b-CAP サーバは、b-CAP 受信スレッドを起動し、受信したパケットに応じて関数をコールバックします。

### 2.1. 関数仕様

#### 2.1.1. bCap\_Open\_Server

##### 書式

```
HRESULT bCap_Open_Server(const char *connect, uint32_t timeout, int *pfd);
```

##### 説明

指定した通信設定で受信スレッドを起動する。

##### 引数

[in]	connect	通信設定
[in]	timeout	受信タイムアウト時間[ms]
[out]	pfd	起動した受信スレッドのファイルディスクリプタ

##### connect

イーサネット(TCP)通信の場合

```
tcp[:DestIP[:DestPort[:SourceIP[:SourcePort]]]]
```

DestIP	クライアントの IP アドレス	(使用しない)
DestPort	クライアントのポート番号	(使用しない)
SourceIP	サーバの IP アドレス	(デフォルト:255.255.255.255)
SourcePort	サーバのポート番号	(デフォルト:5007)

イーサネット(UDP)通信の場合

```
udp[:DestIP[:DestPort[:SourceIP[:SourcePort]]]]
```

※各パラメータの意味はイーサネット(TCP)通信と同様。

シリアル通信の場合

```
com[:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>[:<Flow>]]]]
```

COM Port	COM ポート番号	(デフォルト:1)
BaudRate	ボーレート	(デフォルト:38400)
Parity	パリティビット	(デフォルト:N)
DataBits	データビット	(デフォルト:8)
StopBits	ストップビット	(デフォルト:1)
Flow	フロー制御	(デフォルト:0)

**返回值**

成功した場合は S\_OK(0)を, 失敗した場合は負値を返す.

**備考**

なし

**2.1.2. bCap\_Close\_Server****書式**

```
HRESULT bCap_Close_Server(int *pfd);
```

**説明**

指定したファイルディスクリプタの受信スレッドを停止する.

**引数**

[in,out] pfd            停止する受信スレッドのファイルディスクリプタ

**返回值**

成功した場合は S\_OK(0)を, 失敗した場合は負値を返す.

**備考**

なし

**2.1.3. bCap\_SetCallFunc****書式**

```
HRESULT bCap_SetCallFunc(int32_t id, CALL_FUNC_BCAP func);
```

**説明**

b-CAP パケット受信時にコールバックされる関数を指定する.

**引数**

[in] id            b-CAP 関数 ID  
[in] func          コールバック関数

**コールバック関数**

```
typedef HRESULT (*CALL_FUNC_BCAP)(VARIANT *vntArgs, int16_t Argc, VARIANT *vntRet);
```

[in] vntArgs    引数配列  
[in] Argc       引数配列の要素数  
[out] vntRet    実行結果

**返回值**

成功した場合は S\_OK(0)を, 失敗した場合は負値を返す.

**備考**

コールバック関数の引数の詳細は、「4.付録 A.b-CAP 関数 ID とコールバック関数」を参照.

## 3. b-CAP クライアント

b-CAP クライアントは、指定した b-CAP サーバとの通信を確立し、任意の b-CAP 関数を送信するインターフェースを提供します。

### 3.1. 関数仕様

#### 3.1.1. bCap\_Open\_Client

##### 書式

```
HRESULT bCap_Open_Client(const char *connect, uint32_t timeout, unsigned int retry, int *pfd);
```

##### 説明

指定した通信設定の b-CAP サーバとの通信を確立する。

##### 引数

[in]	connect	通信設定
[in]	timeout	受信タイムアウト時間[ms]
[in]	retry	送信リトライ回数
[out]	pfd	通信確立したソケットのファイルディスクリプタ

##### connect

イーサネット(TCP)通信の場合

```
tcp[:DestIP[:DestPort[:SourceIP[:SourcePort]]]]
```

DestIP	サーバの IP アドレス	(デフォルト:127.0.0.1)
DestPort	サーバのポート番号	(デフォルト:5007)
SourceIP	クライアントの IP アドレス	(使用しない)
SourcePort	クライアントのポート番号	(使用しない)

イーサネット(UDP)通信の場合

```
udp[:DestIP[:DestPort[:SourceIP[:SourcePort]]]]
```

※各パラメータの意味はイーサネット(TCP)通信と同様。

シリアル通信の場合

```
com[:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>[:<Flow>]]]]
```

COM Port	COM ポート番号	(デフォルト:1)
BaudRate	ボーレート	(デフォルト:38400)
Parity	パリティビット	(デフォルト:N)
DataBits	データビット	(デフォルト:8)
StopBits	ストップビット	(デフォルト:1)
Flow	フロー制御	(デフォルト:0)

**返回值**

成功した場合は S\_OK(0)を, 失敗した場合は負値を返す.

**備考**

なし

**3.1.2. bCap\_Close\_Client****書式**

```
HRESULT bCap_Close_Client(int *pfd);
```

**説明**

指定したファイルディスクリプタの通信を切断する.

**引数**

[in,out] pfd          切断する通信のファイルディスクリプタ

**返回值**

成功した場合は S\_OK(0)を, 失敗した場合は負値を返す.

**備考**

なし

**3.1.3. bCap\_SetTimeout****書式**

```
HRESULT bCap_SetTimeout(int fd, uint32_t timeout);
```

**説明**

指定したファイルディスクリプタの受信タイムアウト時間を設定する.

**引数**

[in] fd              受信タイムアウト時間を設定する通信のファイルディスクリプタ  
[in] timeout        受信タイムアウト時間[ms]

**返回值**

成功した場合は S\_OK(0)を, 失敗した場合は負値を返す.

**備考**

なし

**3.1.4. bCap\_GetTimeout****書式**

```
HRESULT bCap_GetTimeout(int fd, uint32_t *timeout);
```

**説明**

指定したファイルディスクリプタの受信タイムアウト時間を取得する.

**引数**

[in] fd 受信タイムアウト時間を取得する通信のファイルディスクリプタ  
[out] timeout 受信タイムアウト時間[ms]

**返り値**

成功した場合は S\_OK(0)を, 失敗した場合は負値を返す.

**備考**

なし

**3.1.5. bCap\_SetRetry****書式**

```
HRESULT bCap_SetRetry(int fd, unsigned int retry);
```

**説明**

指定したファイルディスクリプタの送信リトライ回数を設定する.

**引数**

[in] fd 送信リトライ回数を設定する通信のファイルディスクリプタ  
[in] retry 送信リトライ回数

**返り値**

成功した場合は S\_OK(0)を, 失敗した場合は負値を返す.

**備考**

- 1 以上 7 以下の値を設定する.
- 1 より小さい場合は 1, 7 より大きい場合は 7 として扱われる.
- イーサネット(TCP)通信の場合は常に 1 として扱われる.

**3.1.6. bCap\_GetRetry****書式**

```
HRESULT bCap_GetRetry(int fd, unsigned int *retry);
```

**説明**

指定したファイルディスクリプタの送信リトライ回数を取得する.

**引数**

[in] fd 送信リトライ回数を取得する通信のファイルディスクリプタ  
[out] retry 送信リトライ回数

**返り値**

成功した場合は S\_OK(0)を, 失敗した場合は負値を返す.

**備考**

なし

## 4. b-CAP 通信の制御フロー

b-CAP 通信の制御フローを解説します。図 4-1 にサーバとクライアントの処理の流れを示します。

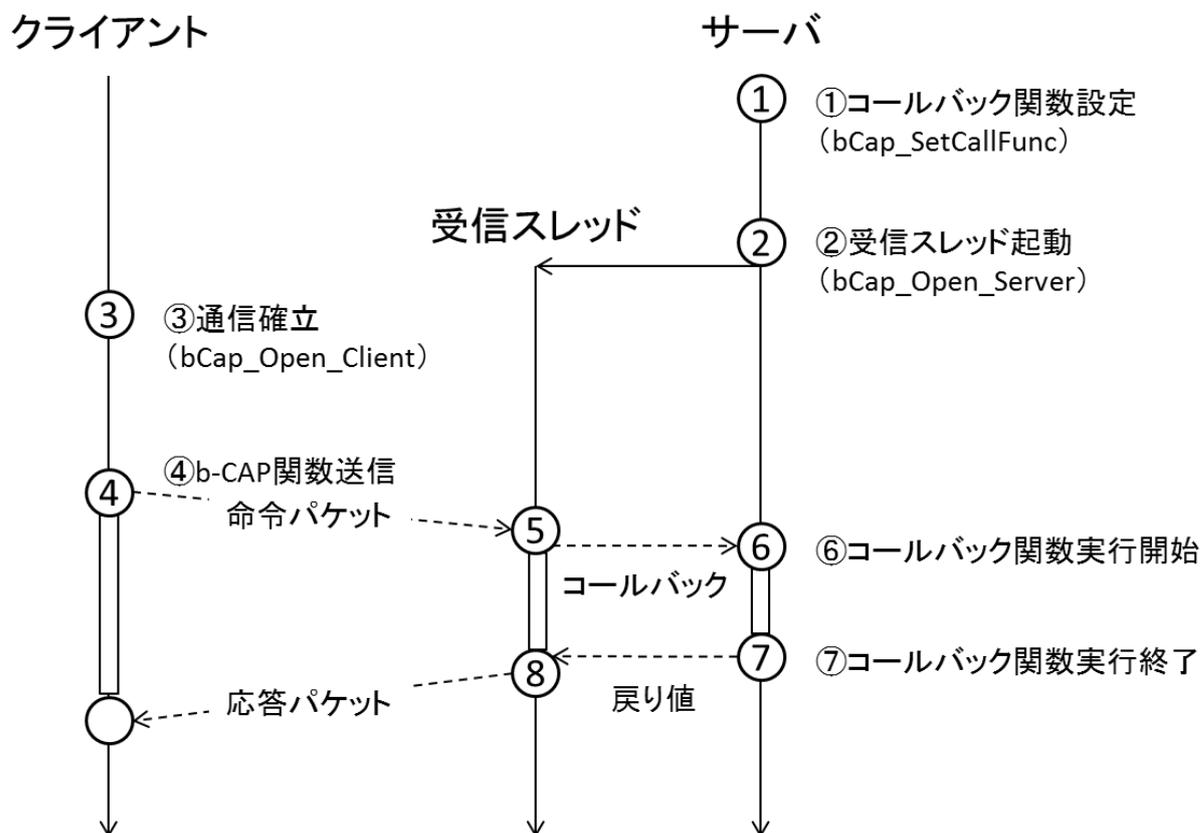


図 4-1 サーバとクライアントの処理の流れ

各処理について説明します。

- ① コールバック関数を設定します。
- ② 受信スレッドを起動します。複数の受信スレッドを起動する場合は、サーバに必要な排他処理を実装してください。イーサネット(TCP)通信の場合、クライアント接続数分の受信スレッドが起動することに注意してください。
- ③ サーバとクライアントの通信を確立します。
- ④ クライアントが提供する b-CAP 関数送信インターフェースを使用します。
- ⑤ 受信スレッドは受信したパケットの b-CAP 関数 ID を確認します。①で設定された b-CAP 関数 ID の場合、⑥を実行します。設定していない ID の場合は、受信スレッドが自動的に E\_NOTIMPL(0x80004001) の応答パケットを送信します。
- ⑥ 受信した b-CAP 関数 ID に対応したコールバック関数が実行されます。
- ⑦ リターンコードに達した時点で終了です。必要であれば vntRet に実行結果を設定します。
- ⑧ 受信スレッドはコールバック関数の戻り値と実行結果を応答パケットとして送信します。

## 付録A. b-CAP 関数 ID とコールバック関数

b-CAP 関数 ID とコールバック関数の引数について説明します。

表 A-1 b-CAP 関数 ID とコールバック関数の引数

b-CAP 関数 ID	vntArgs	vntRet
ID_SERVICE_START	[0] オプション (VT_EMPTY or VT_BSTR)	なし
ID_SERVICE_STOP	なし	なし
ID_CONTROLLER_CONNECT	[0] 名前(VT_BSTR) [1] プロバイダ(VT_BSTR) [2] マシン(VT_BSTR) [3] オプション(VT_BSTR)	ハンドル番号(VT_I4)
ID_CONTROLLER_DISCONNECT	[0] ハンドル番号(VT_I4)	なし
ID_CONTROLLER_GETEXTENSION	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	ハンドル番号(VT_I4)
ID_CONTROLLER_GETFILE	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	ハンドル番号(VT_I4)
ID_CONTROLLER_GETROBOT	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	ハンドル番号(VT_I4)
ID_CONTROLLER_GETTASK	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	ハンドル番号(VT_I4)
ID_CONTROLLER_GETVARIABLE	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	ハンドル番号(VT_I4)
ID_CONTROLLER_GETCOMMAND	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	ハンドル番号(VT_I4)
ID_CONTROLLER_GETEXTENSIONNAMES	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	使用可能な Extension 名の配列 (VT_BSTR   VT_ARRAY)
ID_CONTROLLER_GETFILENAMEAMES	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	使用可能な File 名の配列 (VT_BSTR   VT_ARRAY)

ID_CONTROLLER_GETROBOTNAMES	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	使用可能な Robot 名の配列 (VT_BSTR   VT_ARRAY)
ID_CONTROLLER_GETTASKNAMES	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	使用可能な Task 名の配列 (VT_BSTR   VT_ARRAY)
ID_CONTROLLER_GETVARIABLENAMES	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	使用可能な Variable 名の配列 (VT_BSTR   VT_ARRAY)
ID_CONTROLLER_GETCOMMANDNAMES	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	使用可能な Command 名の配列 (VT_BSTR   VT_ARRAY)
ID_CONTROLLER_EXECUTE	[0] ハンドル番号(VT_I4) [1] コマンド(VT_BSTR) [2] オプション(VT_VARIANT)	実行結果(VT_VARIANT)
ID_CONTROLLER_GETMESSAGE	[0] ハンドル番号(VT_I4)	ハンドル番号(VT_I4)
ID_CONTROLLER_GETATTRIBUTE	[0] ハンドル番号(VT_I4)	属性(VT_I4)
ID_CONTROLLER_GETHELP	[0] ハンドル番号(VT_I4)	ヘルプ(VT_BSTR)
ID_CONTROLLER_GETNAME	[0] ハンドル番号(VT_I4)	名前(VT_BSTR)
ID_CONTROLLER_GETTAG	[0] ハンドル番号(VT_I4)	タグ(VT_VARIANT)
ID_CONTROLLER_PUTTAG	[0] ハンドル番号(VT_I4) [1] タグ(VT_VARIANT)	なし
ID_CONTROLLER_GETID	[0] ハンドル番号(VT_I4)	ID(VT_VARIANT)
ID_CONTROLLER_PUTID	[0] ハンドル番号(VT_I4) [1] ID(VT_VARIANT)	なし
ID_EXTENSION_GETVARIABLE	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	ハンドル番号(VT_I4)
ID_EXTENSION_GETVARIABLENAMES	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	使用可能な Variable 名の配列 (VT_BSTR   VT_ARRAY)
ID_EXTENSION_EXECUTE	[0] ハンドル番号(VT_I4) [1] コマンド(VT_BSTR) [2] オプション(VT_VARIANT)	実行結果(VT_VARIANT)
ID_EXTENSION_GETATTRIBUTE	[0] ハンドル番号(VT_I4)	属性(VT_I4)
ID_EXTENSION_GETHELP	[0] ハンドル番号(VT_I4)	ヘルプ(VT_BSTR)
ID_EXTENSION_GETNAME	[0] ハンドル番号(VT_I4)	名前(VT_BSTR)
ID_EXTENSION_GETTAG	[0] ハンドル番号(VT_I4)	タグ(VT_VARIANT)
ID_EXTENSION_PUTTAG	[0] ハンドル番号(VT_I4) [1] タグ(VT_VARIANT)	なし
ID_EXTENSION_GETID	[0] ハンドル番号(VT_I4)	ID(VT_VARIANT)

ID_EXTENSION_PUTID	[0] ハンドル番号(VT_I4) [1] ID(VT_VARIANT)	なし
ID_EXTENSION_RELEASE	[0] ハンドル番号(VT_I4)	なし
ID_FILE_GETFILE	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	ハンドル番号(VT_I4)
ID_FILE_GETVARIABLE	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	ハンドル番号(VT_I4)
ID_FILE_GETFILENAMEAMES	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	使用可能な File 名の配列 (VT_BSTR   VT_ARRAY)
ID_FILE_GETVARIABLENAMES	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	使用可能な Variable 名の配列 (VT_BSTR   VT_ARRAY)
ID_FILE_EXECUTE	[0] ハンドル番号(VT_I4) [1] コマンド(VT_BSTR) [2] オプション(VT_VARIANT)	実行結果(VT_VARIANT)
ID_FILE_COPY	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	なし
ID_FILE_DELETE	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	なし
ID_FILE_MOVE	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	なし
ID_FILE_RUN	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	コンパイル後ファイル名 (VT_BSTR)
ID_FILE_GETDATECREATED	[0] ハンドル番号(VT_I4)	ファイル作成時刻(VT_VARIANT)
ID_FILE_GETDATELASTACCESSED	[0] ハンドル番号(VT_I4)	ファイル最終アクセス時刻 (VT_VARIANT)
ID_FILE_GETDATELASTMODIFIED	[0] ハンドル番号(VT_I4)	ファイル最終更新時刻 (VT_VARIANT)
ID_FILE_GETPATH	[0] ハンドル番号(VT_I4)	ファイルパス(VT_BSTR)
ID_FILE_GETSIZE	[0] ハンドル番号(VT_I4)	ファイルサイズ(VT_I4)
ID_FILE_GETTYPE	[0] ハンドル番号(VT_I4)	ファイルタイプ(VT_BSTR)
ID_FILE_GETVALUE	[0] ハンドル番号(VT_I4)	ファイル内容(VT_VARIANT)
ID_FILE_PUTVALUE	[0] ハンドル番号(VT_I4)	なし

	[1] ファイル内容(VT_VARIANT)	
ID_FILE_GETATTRIBUTE	[0] ハンドル番号(VT_I4)	属性(VT_I4)
ID_FILE_GETHELP	[0] ハンドル番号(VT_I4)	ヘルプ(VT_BSTR)
ID_FILE_GETNAME	[0] ハンドル番号(VT_I4)	名前(VT_BSTR)
ID_FILE_GETTAG	[0] ハンドル番号(VT_I4)	タグ(VT_VARIANT)
ID_FILE_PUTTAG	[0] ハンドル番号(VT_I4) [1] タグ(VT_VARIANT)	なし
ID_FILE_GETID	[0] ハンドル番号(VT_I4)	ID(VT_VARIANT)
ID_FILE_PUTID	[0] ハンドル番号(VT_I4) [1] ID(VT_VARIANT)	なし
ID_FILE_RELEASE	[0] ハンドル番号(VT_I4)	なし
ID_ROBOT_GETVARIABLE	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	ハンドル番号(VT_I4)
ID_ROBOT_GETVARIABLENAMES	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	使用可能な Variable 名の配列 (VT_BSTR   VT_ARRAY)
ID_ROBOT_EXECUTE	[0] ハンドル番号(VT_I4) [1] コマンド(VT_BSTR) [2] オプション(VT_VARIANT)	実行結果(VT_VARIANT)
ID_ROBOT_ACCELERATE	[0] ハンドル番号(VT_I4) [1] 軸番号(VT_I4) [2] 加速度(VT_R4) [3] 減速度(VT_R4)	なし
ID_ROBOT_CHANGE	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR)	なし
ID_ROBOT_CHUCK	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	なし
ID_ROBOT_DRIVE	[0] ハンドル番号(VT_I4) [1] 軸番号(VT_I4) [2] 移動量(VT_R4) [3] オプション(VT_BSTR)	なし
ID_ROBOT_GOHOME	[0] ハンドル番号(VT_I4)	なし
ID_ROBOT_HALT	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	なし
ID_ROBOT_HOLD	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	なし

ID_ROBOT_MOVE	[0] ハンドル番号(VT_I4) [1] 補間指定(VT_I4) [2] ポーズ列(VT_VARIANT) [3] オプション(VT_BSTR)	なし
ID_ROBOT_ROTATE	[0] ハンドル番号(VT_I4) [1] 回転面(VT_VARIANT) [2] 角度(VT_R4) [3] 回転中心(VT_VARIANT) [4] オプション(VT_BSTR)	なし
ID_ROBOT_SPEED	[0] ハンドル番号(VT_I4) [1] 軸番号(VT_I4) [2] 速度(VT_R4)	なし
ID_ROBOT_UNCHUCK	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	なし
ID_ROBOT_UNHOLD	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	なし
ID_ROBOT_GETATTRIBUTE	[0] ハンドル番号(VT_I4)	属性(VT_I4)
ID_ROBOT_GETHELP	[0] ハンドル番号(VT_I4)	ヘルプ(VT_BSTR)
ID_ROBOT_GETNAME	[0] ハンドル番号(VT_I4)	名前(VT_BSTR)
ID_ROBOT_GETTAG	[0] ハンドル番号(VT_I4)	タグ(VT_VARIANT)
ID_ROBOT_PUTTAG	[0] ハンドル番号(VT_I4) [1] タグ(VT_VARIANT)	なし
ID_ROBOT_GETID	[0] ハンドル番号(VT_I4)	ID(VT_VARIANT)
ID_ROBOT_PUTID	[0] ハンドル番号(VT_I4) [1] ID(VT_VARIANT)	なし
ID_ROBOT_RELEASE	[0] ハンドル番号(VT_I4)	なし
ID_TASK_GETVARIABLE	[0] ハンドル番号(VT_I4) [1] 名前(VT_BSTR) [2] オプション(VT_BSTR)	ハンドル番号(VT_I4)
ID_TASK_GETVARIABLENAMES	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	使用可能な Variable 名の配列 (VT_BSTR   VT_ARRAY)
ID_TASK_EXECUTE	[0] ハンドル番号(VT_I4) [1] コマンド(VT_BSTR) [2] オプション(VT_VARIANT)	実行結果(VT_VARIANT)
ID_TASK_START	[0] ハンドル番号(VT_I4) [1] 開始モード(VT_I4)	なし

	[2] オプション(VT_BSTR)	
ID_TASK_STOP	[0] ハンドル番号(VT_I4) [1] 終了モード(VT_I4) [2] オプション(VT_BSTR)	なし
ID_TASK_DELETE	[0] ハンドル番号(VT_I4) [1] オプション(VT_BSTR)	なし
ID_TASK_GETFILENAME	[0] ハンドル番号(VT_I4)	ファイル名(VT_BSTR)
ID_TASK_GETATTRIBUTE	[0] ハンドル番号(VT_I4)	属性(VT_I4)
ID_TASK_GETHELP	[0] ハンドル番号(VT_I4)	ヘルプ(VT_BSTR)
ID_TASK_GETNAME	[0] ハンドル番号(VT_I4)	名前(VT_BSTR)
ID_TASK_GETTAG	[0] ハンドル番号(VT_I4)	タグ(VT_VARIANT)
ID_TASK_PUTTAG	[0] ハンドル番号(VT_I4) [1] タグ(VT_VARIANT)	なし
ID_TASK_GETID	[0] ハンドル番号(VT_I4)	ID(VT_VARIANT)
ID_TASK_PUTID	[0] ハンドル番号(VT_I4) [1] ID(VT_VARIANT)	なし
ID_TASK_RELEASE	[0] ハンドル番号(VT_I4)	なし
ID_VARIABLE_GETDATETIME	[0] ハンドル番号(VT_I4)	現在時刻(VT_VARIANT)
ID_VARIABLE_GETVALUE	[0] ハンドル番号(VT_I4)	変数値(VT_VARIANT)
ID_VARIABLE_PUTVALUE	[0] ハンドル番号(VT_I4) [1] 変数値(VT_VARIANT)	なし
ID_VARIABLE_GETATTRIBUTE	[0] ハンドル番号(VT_I4)	属性(VT_I4)
ID_VARIABLE_GETHELP	[0] ハンドル番号(VT_I4)	ヘルプ(VT_BSTR)
ID_VARIABLE_GETNAME	[0] ハンドル番号(VT_I4)	名前(VT_BSTR)
ID_VARIABLE_GETTAG	[0] ハンドル番号(VT_I4)	タグ(VT_VARIANT)
ID_VARIABLE_PUTTAG	[0] ハンドル番号(VT_I4) [1] タグ(VT_VARIANT)	なし
ID_VARIABLE_GETID	[0] ハンドル番号(VT_I4)	ID(VT_VARIANT)
ID_VARIABLE_PUTID	[0] ハンドル番号(VT_I4) [1] ID(VT_VARIANT)	なし
ID_VARIABLE_GETMICROSECOND	[0] ハンドル番号(VT_I4)	タイムスタンプ(VT_I4)
ID_VARIABLE_RELEASE	[0] ハンドル番号(VT_I4)	なし
ID_COMMAND_EXECUTE	[0] ハンドル番号(VT_I4) [1] モード(VT_I4)	実行結果(VT_VARIANT)
ID_COMMAND_CANCEL	[0] ハンドル番号(VT_I4)	なし
ID_COMMAND_GETTIMEOUT	[0] ハンドル番号(VT_I4)	タイムアウト時間(VT_I4)

ID_COMMAND_PUTTIMEOUT	[0] ハンドル番号(VT_I4) [1] タイムアウト時間(VT_I4)	なし
ID_COMMAND_GETSTATE	[0] ハンドル番号(VT_I4)	状態(VT_I4)
ID_COMMAND_GETPARAMETERS	[0] ハンドル番号(VT_I4)	コマンドパラメータ(VT_VARIANT)
ID_COMMAND_PUTPARAMETERS	[0] ハンドル番号(VT_I4) [1] コマンドパラメータ (VT_VARIANT)	なし
ID_COMMAND_GETRESULT	[0] ハンドル番号(VT_I4)	実行結果(VT_VARIANT)
ID_COMMAND_GETATTRIBUTE	[0] ハンドル番号(VT_I4)	属性(VT_I4)
ID_COMMAND_GETHELP	[0] ハンドル番号(VT_I4)	ヘルプ(VT_BSTR)
ID_COMMAND_GETNAME	[0] ハンドル番号(VT_I4)	名前(VT_BSTR)
ID_COMMAND_GETTAG	[0] ハンドル番号(VT_I4)	タグ(VT_VARIANT)
ID_COMMAND_PUTTAG	[0] ハンドル番号(VT_I4) [1] タグ(VT_VARIANT)	なし
ID_COMMAND_GETID	[0] ハンドル番号(VT_I4)	ID(VT_VARIANT)
ID_COMMAND_PUTID	[0] ハンドル番号(VT_I4) [1] ID(VT_VARIANT)	なし
ID_COMMAND_RELEASE	[0] ハンドル番号(VT_I4)	なし
ID_MESSAGE_REPLY	[0] ハンドル番号(VT_I4) [1] 返信メッセージ (VT_VARIANT)	なし
ID_MESSAGE_CLEAR	[0] ハンドル番号(VT_I4)	なし
ID_MESSAGE_GETDATETIME	[0] ハンドル番号(VT_I4)	メッセージ作成時刻 (VT_VARIANT)
ID_MESSAGE_GETDESCRIPTION	[0] ハンドル番号(VT_I4)	メッセージ説明文(VT_BSTR)
ID_MESSAGE_GETDESTINATION	[0] ハンドル番号(VT_I4)	メッセージ送り先(VT_BSTR)
ID_MESSAGE_GETNUMBER	[0] ハンドル番号(VT_I4)	メッセージ番号(VT_I4)
ID_MESSAGE_GETSERIALNUMBER	[0] ハンドル番号(VT_I4)	メッセージ連番(VT_I4)
ID_MESSAGE_GETSOURCE	[0] ハンドル番号(VT_I4)	メッセージ送り元(VT_BSTR)
ID_MESSAGE_GETVALUE	[0] ハンドル番号(VT_I4)	メッセージ本文(VT_VARIANT)
ID_MESSAGE_RELEASE	[0] ハンドル番号(VT_I4)	なし

## 付録A.1. ハンドル番号

コールバック関数の引数や実行結果に必要なハンドル番号について説明します。図 A.1-1にハンドル番号生成の流れを示します。

実行結果にハンドル番号を設定する関数（ID\_CONTROLLER\_CONNECT や ID\_CONTROLLER\_GETVARIABLE など）は、コールバック関数実行時にメモリの割り当てを行います。サーバは確保したメモリを解放関数（ID\_CONTROLLER\_DISCONNECT や ID\_VARIABLE\_RELEASE など）が呼ばれるまで保持し続ける必要があります。

そこで、メモリを管理するテーブルを作成し、テーブルからメモリを一意に呼び出すためのキーを生成します。このキーをハンドル番号としてクライアントに通知します。

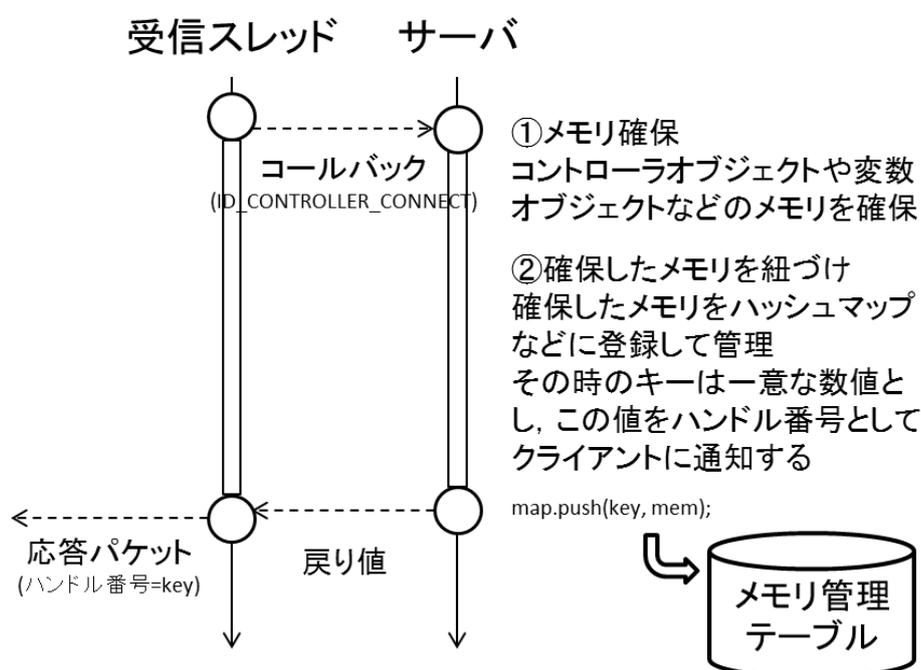


図 A.1-1 ハンドル番号の生成

引数にハンドル番号を含む関数(ID\_CONTROLLER\_EXECUTE など)が実行された場合は、ハンドル番号をキーにメモリ管理テーブルから確保したメモリを取得します。

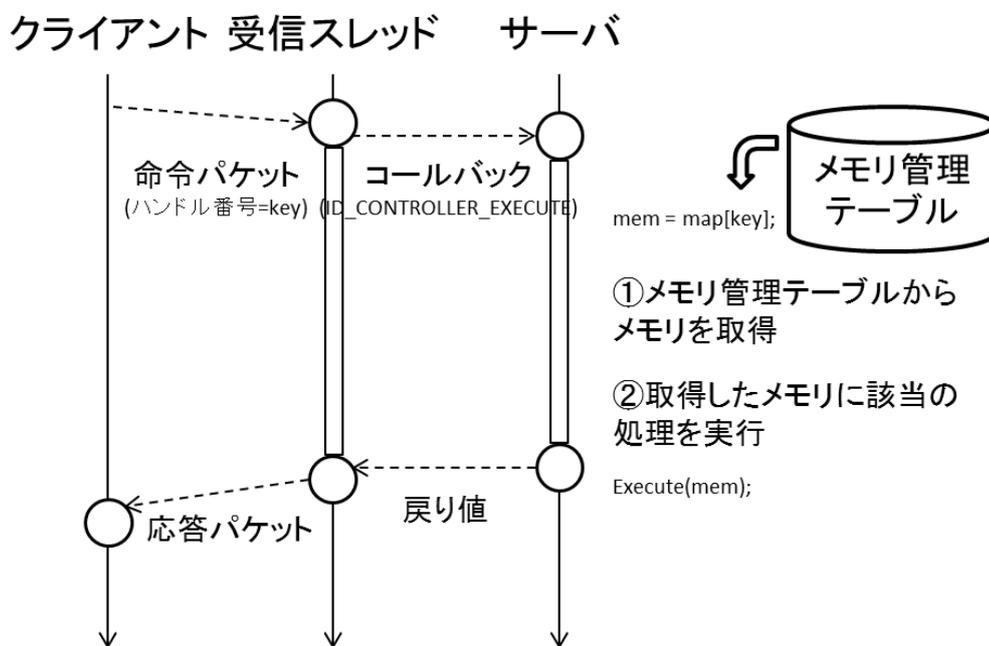


図 A.1-2 ハンドル番号の使用