

Asyrii
EYE+ PacScript ライブラリ

Version 1.0.0

ユーザーズ
ガイド

2022 年 7 月 13 日

〔改訂履歴〕

バージョン	日付	内容
1.0.0	13.07.2022	初版。

目次

1. はじめに.....	4
2. PacScript ライブラリの概要	5
3. パブリックルーチン/関数.....	6
3.1. Sub EYE_CONFIGURE(ByVal ipAddress As String, ByVal portNumber As Integer).....	6
3.2. Sub EYE_START_PRODUCTION(ByVal recipe_id As Integer).....	6
3.3. Sub EYE_STOP(ByVal state As String)	6
3.4. Sub EYE_GET_PART().....	7
3.5. Sub EYE_PREPARE_PART()	7
3.6. Function EYE_RAW_COMMAND(ByVal command as String) As String	7
3.7. Function EYE_CHECK_LAST_ERROR() As Integer.....	7
4. プライベートルーチン/関数	9
5. エラーコード	10
601 - 通信エラー.....	10
意味	10
対処方法.....	10
602 - Client 設定エラー	10
意味	10
対処方法.....	10
603 - 接続失敗.....	10
意味	10
対処方法.....	10
604 - タイムアウト発生.....	11
意味	11
対処方法.....	11
699 - 一般エラー.....	11
意味	11
対処方法.....	11
6. サンプルプログラム	12

1. はじめに

本書は EYE+ PacScript ライブラリのユーザーズガイドです。このライブラリを使用することで、EYE+ プロバイダをより簡単に使用することができます。以下のオンライン資料も参照ください。

- <https://doc.eyepplus.asyrl.com/denso>

EYE+ プロバイダは、イーサネット TCP/IP メッセージのみ(すなわち、全ての下層コミュニケーション)を用いて EYE+ コントローラに接続します。

詳細は、「EYE+プロバイダ ユーザーズガイド」をご参照ください。

- <https://doc.eyepplus.asyrl.com>

2. PacScript ライブラリの概要

PacScript ライブラリは、お客様のデンソープログラムリストに追加する必要がある「.pcs」ファイルです。

ライブラリを使用するには、すでに EYE+プロバイダが登録済みである必要があります。登録方法は「EYE+プロバイダ ユーザーズガイド」を参照ください。

PacScript ライブラリは、パブリックおよびプライベートルーチン/関数を含んだオープンファイルです。

3. パブリックルーチン/関数

パブリックルーチン/関数は、EYE+と通信するためにお客様が呼び出すことができるものです。

3.1. Sub EYE_CONFIGURE(ByVal ipAddress As String, ByVal portNumber As Integer)

このコマンドは、他のいかなるプラグイン機能を使用する各プログラムの最初で呼び出す必要があります。通信設定を定義するために使用します。

入力:

- **ipAddress** – お客様の EYE+ の IP アドレスです。このパラメータは、IP アドレスフォーマットが xxxx の文字列でなければなりません。
- **portNumber** – お客様の EYE+ のポート番号です。このパラメータは、整数でなければなりません。

3.2. Sub EYE_START_PRODUCTION(ByVal recipe_id As Integer)

このコマンドは、正しいレシピを使用して Production 状態で EYE+ を起動するために呼び出す必要があります。

入力:

- **recipe_id**: レシピ固有の識別子です。このパラメータは 1 ～ 65535 の整数でなければなりません。

3.3. Sub EYE_STOP(ByVal state As String)

このコマンドは、EYE+ の状態を停止させ、通信を停止させるために使用します。
EYE_STOP("production") の呼び出しにより、常にプログラムを終了させる必要があります。

入力:

- **state** – EYE+ の状態です。このパラメータは文字列でなければなりません。

3.4. Sub EYE_GET_PART()

このコマンドは、EYE+から1個の部品をリクエストするために使用します。これはブロッキングコマンドで、EYE+からレスポンスを受け取るまで動作し続けます。

戻り値:

- 返された部品は、位置変数 EYEPos に保存されます。X 軸、Y 軸、RZ 角度の要素のみが位置変数 EYEPos に上書きされます。お客様のロボット設定に合わせて、他の要素をお客様自身で定義する必要があります。

3.5. Sub EYE_PREPARE_PART()

このコマンドは EYE+から1個の部品をリクエストするために使用します。このコマンドは、ブロッキングコマンドではありません。この部品の座標は、後から EYE_GET_PART コマンドを使用して取り込むことができます。

3.6. Function EYE_RAW_COMMAND(ByVal command as String) As String

この関数は、Raw コマンドを EYE+に送信するために使用します。

入力:

- command – EYE+に送信する Raw コマンドです。このパラメータは文字列でなければなりません。

戻り値:

- このコマンドの Raw レスポンスは、直接出力として返されます。この出力は文字列です。

3.7. Function EYE_CHECK_LAST_ERROR() As Integer

この関数は、エラー発生の有無を確認するために使用します。

- エラーが発生していない場合、出力は0となります。

- EYE+のエラーが発生した場合、「[Error codes](#)」に記載されているエラーコードの一つ(error type 4xx または 5xx)が出力されます。
- プラグインエラーが発生した場合、「[Plugin errors](#)」に表示されるエラーの一つ(error type 6xx)が出力されます。

エラーが関数からの出力として返されると、このエラーは内部でクリアされます(値は0に設定されます)。

戻り値:

- 最後に検出されたエラーが直接出力として返されます。この出力は整数です。

4. プライベートルーチン/関数

プライベートルーチン/関数は、プラグインを正常に機能させるために内部的に使用されます (EYE_INTERNAL_で始まるプロシージャです)。いずれもお客様が呼び出す必要はありません。

5. エラーコード

601 – Communication error

意味

通信動作中にエラーが発生し、解決されていません。1つ以上の送信を失った可能性があります。

対処方法

お客様の EYE+ とロボットコントローラ間の Ethernet ケーブルが適切に接続されているか確認してください。もし全て問題無いようでしたら、EYE+およびロボットプログラムを再起動してください。

602 – Client configuration error

意味

お客様が EYE_CONFIGURE に入力したパラメータのひとつが誤っています。

対処方法

- EYE_CONFIGURE 関数をプログラムの最初に少なくとも 1 回は呼び出しているか確認してください。
- お客様が EYE_CONFIGURE に入力した IP アドレスの書式およびポート番号範囲を確認し、プログラムを再起動してください。

603 – Connection failed

意味

お客様の EYE+ との接続を確立することができない、またはプログラム実行中に接続が切れました。

対処方法

- お客様の EYE+ とロボットコントローラ間の Ethernet ケーブルが適切に接続されているか確認してください。
- お客様が EYE_CONFIGURE で選択した IP アドレスとポート番号を確認してください。
- EYE+ Studio で正しい行末文字が選択されているか確認してください。
- お客様の EYE+ の状態を確認してください。もし Error 状態だった場合、ログを取得し再起動してください。

604 – Timeout occurred

意味

お客様の EYE+ への送受信を実行中に、プラグインがタイムアウトに達しました。これは 33 秒以内に EYE+ からのレスポンスが受信されなかった場合に発生します。

対処方法

- お客様がプログラム内で設定した EYE+ 内部タイムアウトと EYE+ が現在使用しているタイムアウトが同じになるようにしてください。プラグインを使用せずに異なるタイムアウトを設定していた場合、プラグインは自身の内部タイムアウトを自身で調整できません。もし異なるタイムアウトを使用したい場合は、このコマンドを使用してください。
 - `EYE_RAW_COMMAND("set_parameter timeout xx")`.
- お客様の EYE+ の状態を確認してください。もし Error 状態だった場合、System ログを取得し再起動してください。

699 – Generic error

意味

予期しないエラーが発生しました。

対処方法

お客様の EYE+ の通信ログをダウンロードし、弊社サポートチームに連絡ください。

6. サンプルプログラム

以下はライブラリの使用方法を示したサンプルプログラムです。

Main

```
01 '!TITLE "Example of EYE+ provider direct use"
02 #Include "VAR_TAB.h"
03 #Include "densoEyeplusPlugin.pcs"
04
05 Sub Main
06   ' Stet the correct communication parameters
07   Call EYE_CONFIGURE("192.168.0.50", 7171)
08
09   Set IC[24] ' Open Gripper
10
11   ' Start EYE+ in production using correct recipe identifier
12   Call EYE_START_PRODUCTION(12345)
13   'Check if an error occurred on the last command, if not, go on
14   If EYE_CHECK_LAST_ERROR() = 0 Then
15     ' Start pick and place scenario
16     Dim it as Integer
17     For it = 0 to 2000
18       ' Move robot out of field of view
19       Move F, @F J[out_of_view]
20
21       ' Get one part coordinates
22       Call EYE_GET_PART()
23       If EYE_CHECK_LAST_ERROR() = 0 Then
24         ' Move on the part
25         Approach F, EYEPos, 50
26         Move L, EYEPos
27
28         Reset IC[24] ' Close Gripper
29
30         ' Move on place position
31         Approach L, EYEPos, 50
32         Approach F, F[place_position], 50
33         Move L, F[place_position]
34
35         Set IC[24] ' Open Gripper
36
37         ' Move to above place position
38         Approach F, F[place_position], 50
39       End If
40     Next
41   End If
42 End Sub
43
```