Asyril
EYE+ PacScript Library

Version 1.0.0

User’s guide

July 13, 2022

[Revision history]

Version	Date	Contents
1.0.0	13.07.2022	First version.

Table des matières

1. Introduction

This document is a user's guide of EYE+ PacScript library. This library makes it easier to use the EYE+ provider. You can also refer to the online documentation here:

- https://doc.eyeplus.asyril.com/denso

The EYE+ provider connects with the EYE+ controller with ethernet **TCP/IP messaging only**, managing all low-level communication.

Please refer to "EYE+ provider user's guide" for more detail about it.

- https://doc.eyeplus.asyril.com

2. Outline of PacScript Library

The PacScript library is a ".pcs" file that must be added in your denso programs list.
To use it, you must have registered the provider for EYE+. To do so, please refer to "EYE+ provide user's guide".

The PacScript library is an open file containing public and private routines/functions.

3. Public routines/functions

Public routines/functions are those that can be called by the customers to communicate with EYE+.

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

This command must be called at the beginning of each program that use any of the other plugin functions. It is used to define the communication settings.

Inputs:

- ipAddress - is the IP address of your EYE+. The parameter must be a string with the IP address format x.x.x.x.
- portNumber - is the port number of your EYE+. The parameter must be an integer.

3.2. Sub EYE_START_PRODUCTION(ByVal recipe_id As Integer)

This command must be called to start EYE+ in production state using the right recipe.

Inputs:

- recipe_id: the recipe’s unique identifier. The parameter must be an integer between 1 and 65535.

3.3. Sub EYE_STOP(ByVal state As String)

This command is used to stop an EYE+ state and stop the communication. You must always end your program with an call EYE_STOP("production").

Inputs:

- state - is an EYE+ states. The parameter must be a string.

3.4. Sub EYE_GET_PART()

This command is used to request one part from EYE+. This is a blocking command, meaning it will keep going until it gets a response from EYE+.

Returns:

- The returned part is stored in the position variable EYEPos.
  Only the components for X axis, for Y axis and for RZ angle are overwritten in the position EYEPos. You have to define the other components yourself according to your robot setup.

3.5. Sub EYE_PREPARE_PART()

This command is used to request one part from EYE+. This command is not a blocking command. The part coordinates can be retrieved later with the EYE_GET_PART command.

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

This function is used to send raw commands to EYE+.

Inputs:

- command - is the raw command to send to EYE+. This parameter must be a string.

Returns:

- The raw response of the command is returned as direct output. The output is a string.

3.7. Function EYE_CHECK_LAST_ERROR() As Integer

This function is used to check if an error has occurred.

- If no error has occurred, the output is equal to 0.
- If an EYE+ error has occurred, the output is equal to one of the error codes listed in Error codes (error type 4xx or 5xx).

- If a plugin error has occurred, the output is one of the following errors displayed in section Plugin errors (error type 6xx).

Once the error is returned as output from the function, the error is cleared internally (value set to 0).

Returns:

- The last error detected is returned as direct output. The output is an integer.

4. Private routines/functions

Private routines and functions are used internally to make the plugin worked properly (procedures that start with EYE_INTERNAL_). None of those must be called by the customers.

5. Error Codes

601 - Communication error

Signification

An error occurred during the communication operation and was not resolved. You have probably lost one or more communication transfers.

Troubleshoots

Check that your Ethernet cable is properly connected between your EYE+ and your robot controller. If everything looks fine, try to restart your EYE+ and/or your robot’s program.

602 - Client configuration error

Signification

One of the parameters you entered in EYE_CONFIGURE is incorrect.

Troubleshoot

- Check that you have called the function EYE_CONFIGURE at least once at the beginning of your program.
- Check the IP address syntax and port number range you entered in EYE_CONFIGURE and restart your program.

603 - Connection failed

Signification

It is not possible to establish a connection with your EYE+ or you lost the connection while the program was running.

Troubleshoot

- Check that your Ethernet cable is properly connected between your EYE+ and your robot controller.
- Check the IP address and port number you selected in EYE_CONFIGURE.
- Check that you have selected the correct end of line character in EYE+ Studio.
- Check the state of your EYE+. If your EYE+ is in the Error state, get the logs and restart it.

604 - Timeout occurred

Signification

The plugin has reached a timeout when handling a send or receive to your EYE+. This event occurs if no response is received from EYE+ within 33s.

Troubleshoot

- Make sure that the internal EYE+ timeout you set in your program is the same as the one your EYE+ currently uses. If you have set a different timeout without using the plugin, the plugin will not be able to adjust its internal timeout itself. If you want to use a different timeout, use the command:
  - *EYE_RAW_COMMAND("set_parameter timeout xx").*
- Check the state of your EYE+. If your EYE+ is in the Error state, get the System logs and restart it.

699 - Generic error

Signification

An unexpected error occurred.

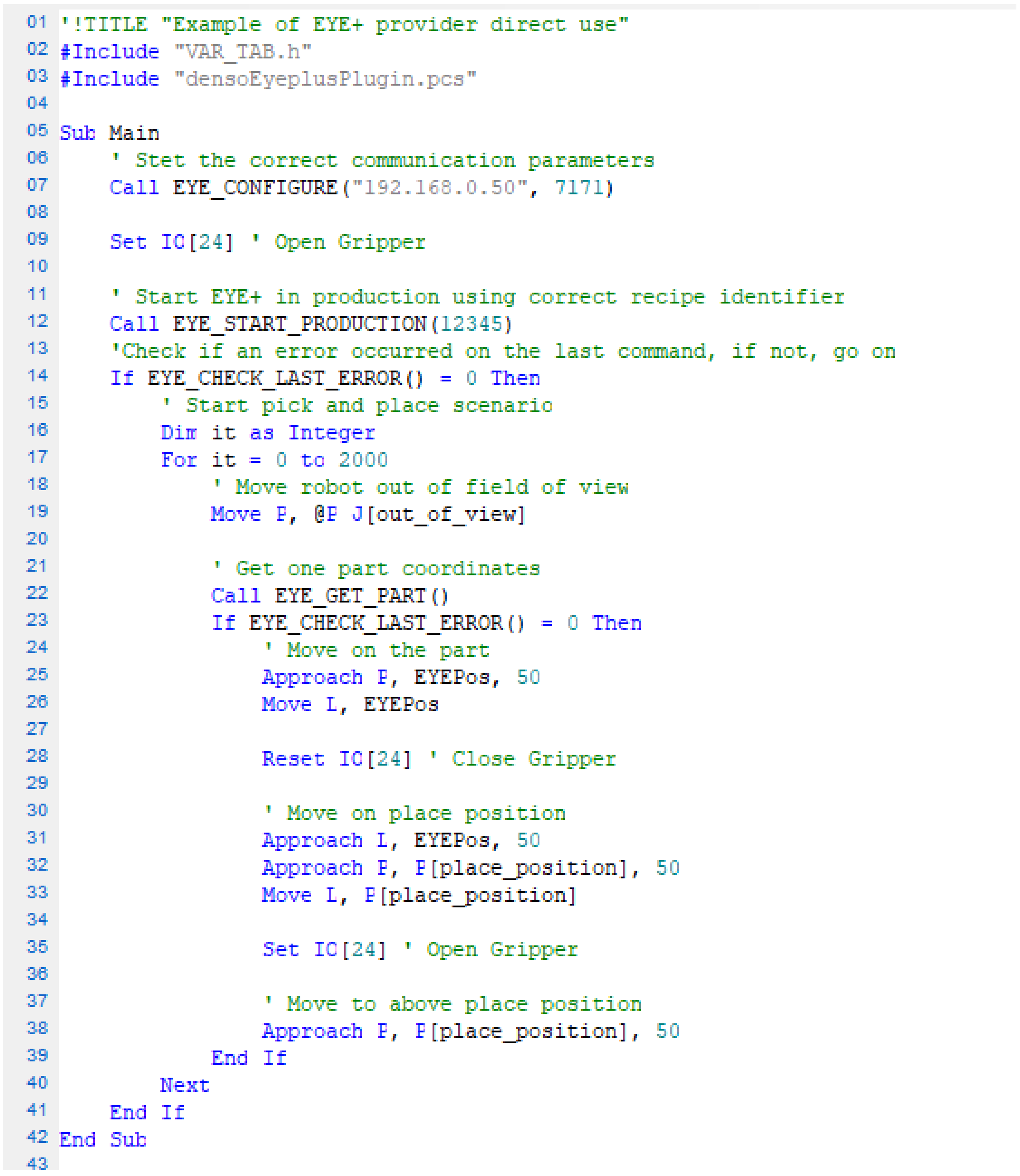
Troubleshoot

Download the Communication logs of your EYE+ and contact our support team.

6. Sample program

Here is a sample program to show how to use the library.

Main



```
'!TITLE "Example of EYE+ provider direct use"
#Include "VAR_TAB.h"
#Include "densoEyeplusPlugin.pcs"

Sub Main
    ' Stet the correct communication parameters
    Call EYE_CONFIGURE("192.168.0.50", 7171)

    Set IO[24] ' Open Gripper

    ' Start EYE+ in production using correct recipe identifier
    Call EYE_START_PRODUCTION(12345)
    'Check if an error occurred on the last command, if not, go on
    If EYE_CHECK_LAST_ERROR() = 0 Then
        ' Start pick and place scenario
        Dim it as Integer
        For it = 0 to 2000
            ' Move robot out of field of view
            Move P, @P J[out_of_view]

            ' Get one part coordinates
            Call EYE_GET_PART()
            If EYE_CHECK_LAST_ERROR() = 0 Then
                ' Move on the part
                Approach P, EYEPos, 50
                Move L, EYEPos

                Reset IO[24] ' Close Gripper

                ' Move on place position
                Approach L, EYEPos, 50
                Approach P, P[place_position], 50
                Move L, P[place_position]

                Set IO[24] ' Open Gripper

                ' Move to above place position
                Approach P, P[place_position], 50
            End If
        Next
    End If
End Sub
```