


# *Plugin Beckhoff Smartsight*

## *Integration Guide*




<b>Document</b>	AsyriL_Plugin_Beckhoff_Smartsight_Integration_Guide_E 000.100.522		
<b>Version Doc</b>	A1	<b>Date</b>	21.03.2018
<b>Version SW</b>	From	2	To

 Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyriil S.A.
000.100.522		Version: A1

# Table of Contents

- 1. INTRODUCTION ..... 4**
  - 1.1. GENERAL INFORMATION..... 4
  - 1.2. OTHER MANUALS..... 5
- 2. GENERAL INTRODUCTION..... 6**
  - 2.1. ASYCUBE SMARTSIGHT AND ASYVIEW..... 6
  - 2.2. TWINGCAT ..... 6
  - 2.3. TWINGCAT-ASYVIEW PLUGIN..... 6
  - 2.4. DOCUMENT CONTENT ..... 7
- 3. PLUGIN INSTALLATION..... 8**
  - 3.1. REQUIREMENTS ..... 8
  - 3.2. WIRING AND CONFIGURATION..... 8
  - 3.3. TWINGCAT 2 ..... 9
  - 3.4. TWINGCAT 3 ..... 9
- 4. IMPLEMENTATION EXAMPLE ..... 10**
  - 4.1. GENERAL USAGE..... 10
  - 4.2. PICK AND PLACE APPLICATION..... 11
    - 4.2.1. *Active WorkingMode*..... 11
    - 4.2.2. *Twincat example*..... 13
  - 4.3. RECIPE MANAGEMENT ..... 16
  - 4.4. ERROR/ALERT HANDLING..... 16
  - 4.5. SAMPLE CODE ..... 16
- 5. SOFTWARE ARCHITECTURE AND FUNCTIONS ..... 17**
  - 5.1. ASYVIEWACCESSMODEL FUNCTION BLOCK (FB) ..... 17
    - 5.1.1. *Inputs and outputs*..... 17
    - 5.1.2. *Actions*..... 18
  - 5.2. PROTOCOLMANAGER FUNCTION BLOCK (FB) ..... 18
    - 5.2.1. *Introduction to the protocol*..... 18
  - 5.3. COMCHANNEL FUNCTION BLOCK (FB)..... 20
  - 5.4. ADMINISTRATIVE FUNCTIONS (FUN) ..... 21
    - 5.4.1. *External use*..... 21
    - 5.4.2. *Internal use*..... 21
  - 5.5. TYPES..... 21
  - 5.6. ALERT LIST ..... 22
- 6. TECHNICAL SUPPORT..... 23**
  - 6.1.1. *For a better service* ..... 23
  - 6.1.2. *Contact*..... 23
- REVISION TABLE ..... 24**

<b>asyril</b> Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyriil S.A.
000.100.522		Version: A1

 Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyрил S.A.
000.100.522		Version: A1

# 1. Introduction

## 1.1. General information

This document is the property of Asyрил SA; it may not be reproduced, modified or communicated, in whole or in part, without our prior written authorization. Asyрил SA reserves the right to modify any information contained in this document for reasons related to product improvements without prior notice. Before using the product, please read this entire document in order to ensure that the product is used correctly. However, if you encounter difficulties when using the product, do not hesitate to contact our customer service department.

In this manual, the safety information that must be respected is split into three types: "Danger", "Important" and "Note". These messages are identified as follows:



### **DANGER!**

Failure to respect this instruction may result in serious physical injury.



### **DANGER!**

This instruction identifies an electrical hazard. Failure to respect this instruction may result in electrocution or serious physical injury due to an electric shock.



### **IMPORTANT!**

Failure to respect this instruction may result in serious damage to equipment.



### **NOTE:**

*The reader's attention is drawn to this point in order to ensure that the product is used correctly. However, failure to respect this instruction does not pose a danger.*



### *Reference ...*

*For more information on a specific topic, the reader is invited to refer to another manual or another page of the current manual.*




### **IMPORTANT!**

Asyрил cannot be held responsible for damage to property or persons caused by the failure to respect the instructions contained in the manual for your machine.



### **NOTE:**

*All dimensions and values in this manual are expressed in millimeters (mm)*


 <b>asyril</b> <small>Experts in Flexible Feeding Systems</small>	<b>Plugin Beckhoff Smartsight Integration Guide</b>	© Copyright Asyriil S.A.
000.100.522		Version: A1

## 1.2. Other manuals

The table below provides a list of documents supplied with the product. Each of these manuals forms an integral part of the set of documentation associated with the product.

Manual title	Reference	Description of the content
<b>Smart Sight Programming manual</b>	Asyriil_SMARTSIGHT_Programming _Guide_EN-revC or superior	Contains a description of how the product works and information on communication and using the product in programming
...		

**Table 1-1: Other manuals**

 <b>Asyri</b> <small>Experts in Flexible Feeding Systems</small>	<b>Plugin Beckhoff Smartsight Integration Guide</b>	© Copyright Asyri S.A.
000.100.522		Version: A1

## 2. General introduction

Asyri offers different solution for flexible feeding in automation. The Asycube SmartSight combines a flexible feeder and a vision system with its Asyview software controller. In order to reduce the software integration effort for Asycube Smartsight customers using TwinCAT (Beckhoff), a software plugin has been developed.

### 2.1. Asycube Smartsight and Asyview

Smart Sight describes Asyri's intelligent visual part detection system, ensuring straightforward integration of any Asycube flexible feeder with any industrial robot brand. It makes the implementation of high performance flexible feeding system as simple as setting up conventional feeders.

The Smart Sight software is called Asyview. It has an GUI dedicated to the integrator tasks as well as some log functionalities available for the end-user. The configuration of any recipes and calibration tasks is operated through the Asyri HMI.



*More information can be found on Asycube SmartSight documentations.*

### 2.2. TwinCAT

Beckhoff created a global standard for automation with the launch of PC-based control technology in 1986. On the software side, the TwinCAT (The Windows Control and Automation Technology) automation suite forms the core of the control system. The TwinCAT software system turns almost any PC-based system into a real-time control with multiple PLC, NC, CNC and/or robotics runtime systems. TwinCAT 3 is the systematic further development of TwinCAT 2, with which the world of automation technology is being redefined.<sup>1</sup>




*More information can be found on Beckhoff documentations.*

### 2.3. TwinCAT-Asyview plugin

Asyri has developed a TwinCAT library (also called plugin) to simplify the software integration of customers willing to combine an Asycube Smartsight with an automation

---

<sup>1</sup> Copied from <http://www.beckhoff.ch/english.asp?twincat/default.htm>

 <b>asyril</b> <small>Experts in Flexible Feeding Systems</small>	<b>Plugin Beckhoff Smartsight Integration Guide</b>	© Copyright Asyriil S.A.
000.100.522		Version: A1

system running TwinCAT. The plugin is a TwinCAT 2 library. You can use it both in TwinCAT 2 and in TwinCAT 3 environments.

The plugin manages the low-level TCP/IP communication channel and the Asyview specific communication protocol. It also offers an abstraction of an Asyview module, making easy for the user to handle many modules at the same time.

File name	Description
asyviewDIL.lib	The TwinCAT 2 library/plugin
asyviewDIL.library	The TwinCAT 3 library/plugin
MAIN.EXP	Example for TwinCAT 2
TwinCAT Project	Example project for TwinCAT 3

## 2.4. Document content

This document presents the TwinCAT-Asyview plugin and explains how to use it in conjunction with the Asycube Smartsight. Chapter 3 explains how to install the plugin. Chapter 4 shows how to use the plugin and presents some sample code. Chapter 5 presents the plugin architecture and the useful functions. Finally, Chapter 0 gives some information on the technical support.

### 3. Plugin installation

This chapter briefly presents how to install the plugin. The plugin is compatible with the Asycube Smartsight and its Asyview software from version V4.0.

#### 3.1. Requirements

You need a PC running Microsoft Windows with TwinCAT 2 or TwinCAT 3 installed. In order to communicate with the Smartsight, you need an Ethernet port available on your PC and a Beckhoff TCP/IP library (Tc2\_Tcplp).

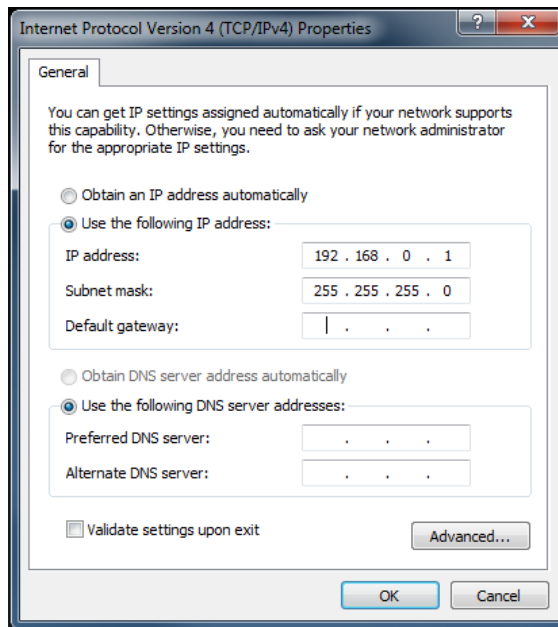
#### 3.2. Wiring and configuration

Connect an Ethernet cable between the PROCESS port from the SmartSight and a port on the PC running TwinCAT. The default configuration for the Asyview server port (PROCESS) is as follows:

IP address	Subnet Mask	Port
192.168.0.70	255.255.255.0	7171

**Table 3-1: Asyview PROCESS TCP/IP parameters**


Configure your PC TCP/IP network configuration (Properties in Change adapter settings) according to the Asyview server port.



**Table 3-2: Example of you PC network configuration.**

Test the network connection with a PING in the Windows Command Prompt.



 Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyriil S.A.
000.100.522		Version: A1


### 3.3. TwinCAT 2

- Open TwinCAT PLC Control
- Create a new project (File->New->PC)
- Add the library (Insert -> Additional library -> Brows for the \*.lib file)

### 3.4. TwinCAT 3

- Open Visual Studio
- Create a new TwinCAT Project (File->New->Project)
- Create a new PLC program (Right-click on PLC -> Add new item -> Standard PLC project)
- Install the library (Right-click on References -> Add library -> Library Repository -> Install -> Browse for the \*.library file)
- Add the library (Right-click on References -> Add library -> (Controller) -> AsyviewDIL 2.0 Asyriil S.A.)

You can also convert the TwinCAT 2 library for TwinCAT 3. Open the library in TwinCAT 3 and follow the converter wizard. You will find the missing communication functions in the Beckhoff Communication->Tcplp->Tc2\_Tcplp library.

 Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyriil S.A.
000.100.522		Version: A1

## 4. Implementation example

This chapter presents some examples on how to use the library in the TwinCAT PLC Structured Text language. We recommend to open and to run the examples (Section 0) in order to understand how it behaves.

The IEC 61131 code run cyclically based on a task cycle time. The plugin performance depends on the configured cycle time. As an example, a full pick and place cycle (see Section 4.2) will take around 30 ms with a 50 us task cycle time or around 350 ms with a 5 ms task cycle time. A typical task cycle time of 1 ms seems appropriate to achieve decent performances.

### 4.1. General usage

This section presents the general usage of the plugin in the Twincat environment. The code samples are presented here to clarify the concept but are not meant to be used as is in your application. Working code examples are provided in the library package.

In order to communicate with the Asyview, you will have first to declare a module (type `AsyviewAccessModel`) and a message (type `MESSAGE`):

```
VAR
    asyview:AsyviewAccessModel;
    msg:MESSAGE;
    ...
END_VAR
```

You will then have to configure the module:


```
asyview.remoteIpAddress:='127.0.0.1';
asyview.remoteIpPort:=7171;
```

Then start the module and check whether an error occurred or the connection gets successful:

```
asyview.Start();
IF asyview.isError THEN
    ...
ELSIF asyview.isConnected THEN
    ...
END_IF
```

And don't forget to run the module function block at each cycle call:

```
asyview();
```

 Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyriil S.A.
000.100.522		Version: A1

When the module is connected, you can construct, send a message and check whether the message has been sent successfully. Please note that in order to be ready to send the next message you have to trig the SendMessage action with the execute field to FALSE:

```

msg.msgType:=request;
msg.address:='a[0]/c[0]/m[0]';
msg.keyword:='start';
msg.callType:=async;
msg.parameter:='';
msg.standardParameter:='';

asyview.SendMessage(execute:=TRUE,msgRequest:=msg);

IF asyview.isError THEN
  asyview.SendMessage(execute:=FALSE);
  ...
ELSIF asyview.isDone THEN
  asyview.SendMessage(execute:=FALSE);
  ...
END_IF

```

## 4.2. Pick and place application

The SmartSight has first been designed for pick and place applications. The machine controller just has to request a new position to the SmartSight. The SmartSight handles autonomously if it has to feed with the hopper, vibrate the asycube plate or take a picture. This is the basic workflow, known as Active WorkingMode in the Asyview.

This section first presents the basic workflow on the Asyview side and then shows an example code for TwinCAT, based on the plugin.

### 4.2.1. Active WorkingMode

Figure 4-1 shows the standard interaction between a Smart Sight module and the machine to synchronize the two cycles and transfer the position information.



*The simulator (Asyview interface) details the same method and describes the instructions used and the responses obtained.*

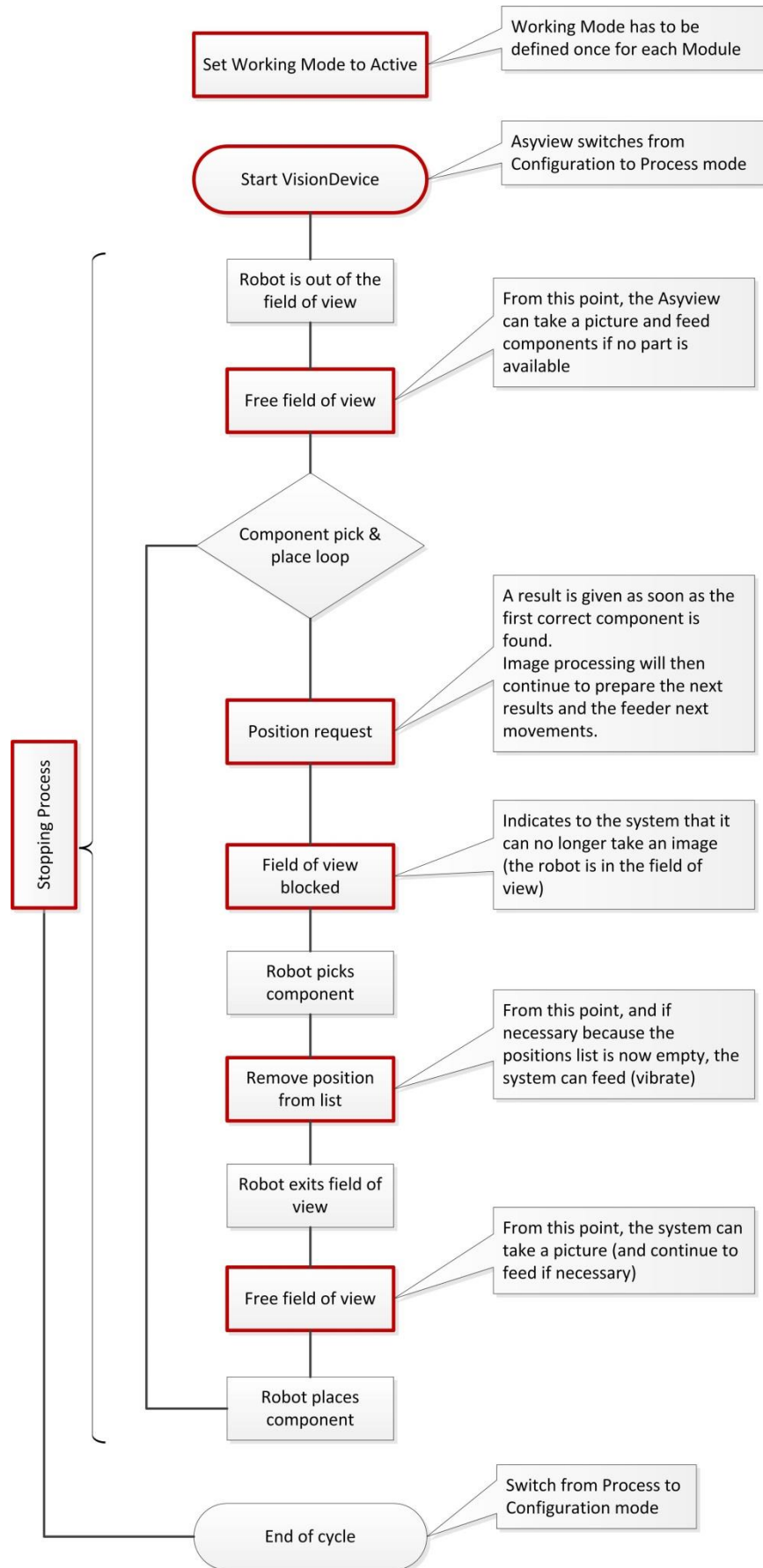



Figure 4-1: Schematic of the Active type (in red: Asyview/machine interactions)

 Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyriil S.A.
000.100.522		Version: A1

## 4.2.2. TwinCAT example

The code here below presents an implementation example for the basic workflow. The code samples are presented here to clarify the concept but are not meant to be used as is in your application. Working code examples are provided in the library package.

```

PROGRAM MAIN
VAR
  asyview:AsyviewAccessModel;
  msg:MESSAGE;
  result:MESSAGE_RESULT;
  pickPosition: ARRAY [0..3] OF LREAL;
  errorText:STRING(255);
  state:INT:=0;
END_VAR

CASE state OF
1:  (*start module*)
  asyview.remoteIpAddress:='192.168.0.1';
  asyview.remoteIpPort:=7171;
  asyview.Start();

  IF asyview.isError THEN
    errorText:=errorToString(asyview.error);
    state:=-1;
  ELSIF asyview.isConnected THEN
    state:=state+1;
  END_IF

2:  (*set active*)
  msg.msgType:=request;
  msg.address:='a[0]/c[0]/m[0]';
  msg.keyword:='setparameter';
  msg.callType:=async;
  msg.parameter:='name=workingmode:workingmode=active';
  msg.standardParameter:='';


  asyview.SendMessage(execute:=TRUE,msgRequest:=msg);

  IF asyview.isError THEN
    asyview.SendMessage(execute:=FALSE);
    errorText:=errorToString(asyview.error);
    state:=-1;
  ELSIF asyview.isDone THEN
    asyview.SendMessage(execute:=FALSE);
    state:=state+1;
  END_IF

3:  (*start*)
  msg.msgType:=request;
  msg.address:='a[0]/c[0]/m[0]';
  msg.keyword:='start';
  msg.callType:=async;
  msg.parameter:='';
  msg.standardParameter:='';

  asyview.SendMessage(execute:=TRUE,msgRequest:=msg);

```

 Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyriil S.A.
000.100.522		Version: A1

```

IF asyview.isError THEN
  asyview.SendMessage(execute:=FALSE);
  errorText:=errorToString(asyview.error);
  state:=-1;
ELSIF asyview.isDone THEN
  asyview.SendMessage(execute:=FALSE);
  state:=state+1;
END_IF

4:  (*move robot out of field of view*)
    ...
    state:=state+1;

5:  (*unlock fov*)
    msg.msgType:=request;
    msg.address:='a[0]/c[0]/m[0]/i[0]';
    msg.keyword:='setparameter';
    msg.callType:=async;
    msg.parameter:='name=fieldofview:imageconfigurationname=default:locke
d=false';
    msg.standardParameter:='';

    asyview.SendMessage(execute:=TRUE,msgRequest:=msg);

IF asyview.isError THEN
  asyview.SendMessage(execute:=FALSE);
  errorText:=errorToString(asyview.error);
  state:=-1;
ELSIF asyview.isDone THEN
  asyview.SendMessage(execute:=FALSE);
  state:=state+1;
END_IF

6:  (*getresult*)
    msg.msgType:=request;
    msg.address:='a[0]/c[0]/m[0]';
    msg.keyword:='getresult';
    msg.callType:=async;
    msg.parameter:='';
    msg.standardParameter:='';


    asyview.SendMessage(execute:=TRUE,msgRequest:=msg,msgResponse=>msg);

    result:=stringToMessageResult(msg.parameter);

IF asyview.isError THEN
  asyview.SendMessage(execute:=FALSE);
  errorText:=errorToString(asyview.error);
  state:=-1;
ELSIF asyview.isDone THEN
  asyview.SendMessage(execute:=FALSE);
  state:=state+1;
END_IF

7:  (*pick and go*)
    pickPosition:=result.pos;
    ...
    state:=state+1;

```

 Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyriil S.A.
000.100.522		Version: A1

```

8:   (*lock fov*)
    msg.msgType:=request;
    msg.address:='a[0]/c[0]/m[0]/i[0]';
    msg.keyword:='setparameter';
    msg.callType:=async;
    msg.parameter:='name=fieldofview:imageconfigurationname=default:locke
d=true';
    msg.standardParameter:='';

    asyview.SendMessage(execute:=TRUE,msgRequest:=msg);

    IF asyview.isError THEN
      asyview.SendMessage(execute:=FALSE);
      errorText:=errorToString(asyview.error);
      state:=-1;
    ELSIF asyview.isDone THEN
      asyview.SendMessage(execute:=FALSE);
      state:=state+1;
    END_IF

9:   (*removeresult*)
    msg.msgType:=request;
    msg.address:='a[0]/c[0]/m[0]';
    msg.keyword:='removeresult';
    msg.callType:=async;
    msg.parameter:=concat('id=',UDINT_TO_STRING(result.id));
    msg.standardParameter:='';

    asyview.SendMessage(execute:=TRUE,msgRequest:=msg);

    IF asyview.isError THEN
      asyview.SendMessage(execute:=FALSE);
      errorText:=errorToString(asyview.error);
      state:=-1;
    ELSIF asyview.isDone THEN
      asyview.SendMessage(execute:=FALSE);
      state:=state+1;
    END_IF


10:  (*place and go*)
    state:=state+1;

11:  (*unlock fov*)
    msg.msgType:=request;
    msg.address:='a[0]/c[0]/m[0]/i[0]';
    msg.keyword:='setparameter';
    msg.callType:=async;
    msg.parameter:='name=fieldofview:imageconfigurationname=default:locke
d=false';
    msg.standardParameter:='';

    asyview.SendMessage(execute:=TRUE,msgRequest:=msg);

    IF asyview.isError THEN
      asyview.SendMessage(execute:=FALSE);
      errorText:=errorToString(asyview.error);
      state:=-1;
    ELSIF asyview.isDone THEN

```

 Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyriil S.A.
000.100.522		Version: A1

```

        asyview.SendMessage(execute:=FALSE);
        state:=6;
      END_IF
    END_CASE

    asyview();

```

### 4.3. Recipe management

If you need to load or save a recipe, you will have to build your message the same way it has been presented above. Please refer to the SmartSight documentation to get the proper wording. Bear in mind that you need to escape some special characters such as “.” in the file path “C:/...” will become “C&#58/...”.

### 4.4. Error/alert handling

When an error occurs, you can get the information in the ALERT output of the Function Blocks. You will then need to reset the corresponding Function Block and solve the problem. Here below is an example on how to reset the error in your main program:

```

-2:  (*resetting error*)
      IF asyview.isError THEN
        asyview.Reset();
      END_IF
      state:=state-1;

```

### 4.5. Sample code

You can find working sample code in the library package, for both TwinCAT 2 and TwinCAT 3. Feel free to adapt this code for your needs.



## 5. Software architecture and functions

This chapter briefly presents the main Functions Blocks (FB) and Functions (FUN) that need to be implemented in your application, as well as some other programming elements.

The current version of the library offers only a first level of abstraction handling the low-level communication and the protocol specific to the Asyview.

### 5.1. AsyviewAccessModel Function Block (FB)

The `AsyviewAccessModel` FB is the entry point of the library. It is important to understand how to use it. We recommend to instantiate one `AsyviewAccessModel` FB per Smartsight module so you can communicate in parallel with different modules. Calling `AsyviewAccessModel` instance automatically calls the internal `protocolManager` and `comChannel` Function Block instances.

This section presents mostly the FB inputs/outputs and actions.

#### 5.1.1. Inputs and outputs

You will have to deal with the following inputs and outputs:

	Name	Type	Description
<b>INPUTS</b>	<code>execute</code>	BOOL	This input is used as an argument of the actions. The actions are executed on a rising edge of <code>execute</code> .
	<code>msgRequest</code>	MESSAGE	This input is used as an argument of the <code>SendMessage</code> action.
	<code>remoteIpAddress</code>	T_IPv4Addr	Configuration parameter to be set before the <code>Start</code> action.
	<code>remoteIpPort</code>	UDINT	Configuration parameter to be set before the <code>Start</code> action.
<b>OUTPUTS</b>	<code>msgResponse</code>	MESSAGE	This output gives the response of the <code>SendMessage</code> action.
	<code>isDone</code>	BOOL	<code>SendMessage</code> action has finished.
	<code>isBusy</code>	BOOL	<code>SendMessage</code> action is busy.
	<code>isError</code>	BOOL	Is true if either <code>comChannel</code> or <code>protocolManager</code> is in error.
	<code>error</code>	ALERT	Reports the error cause.

	isReady	BOOL	Is true is ready to send or receive a message.
	isConnected	BOOL	comChannel is connected.
	isReadyToConnect	BOOL	comChannel is ready to connect and protocolManager is ready.

### 5.1.2. Actions

You can use the following actions:

**Abort:** it aborts the comChannel and the protocolManager to be ready for a fresh start.

**Reset:** it clears the error and resets or aborts the comChannel and the protocolManager to be ready for a fresh start.

**SendMessage:** it sends a message on a rising edge of the execute input.

**Start:** it sets the configuration inputs values and start the comChannel.

**Stop:** it stops the comChannel and resets or aborts the protocolManager to be ready for a fresh start.

## 5.2. protocolManager Function Block (FB)

The protocolManager is a Function Block hidden for the end-user. However, it may be interested to understand its working principle for debug purpose. The protocolManager has the following structure:

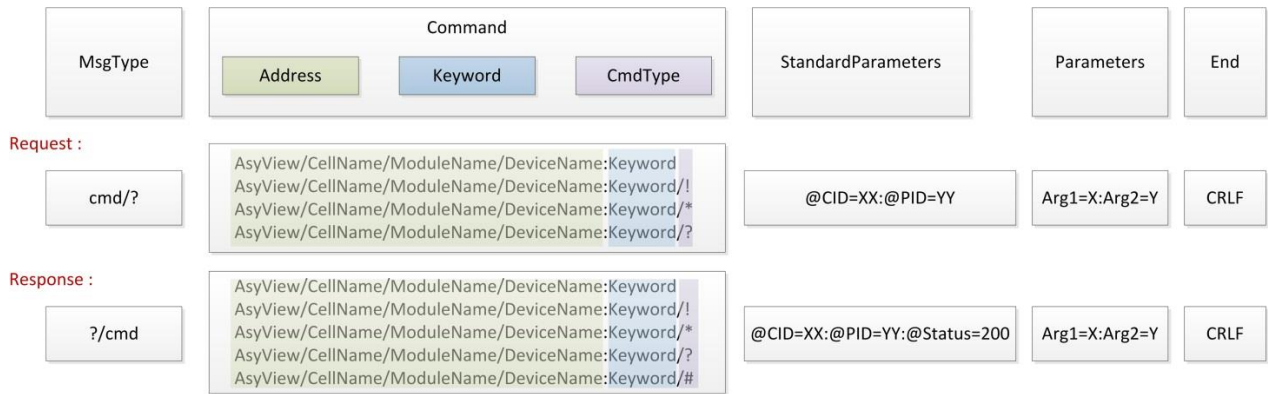
	Name	Type
INPUTS	comChannel	POINTER TO ComChannel
	msg	MESSAGE
	msgType	MSG_TYPE
OUTPUTS	msgResult	MESSAGE
	isReady	BOOL
	isError	BOOL
	error	ALERT

One protocolManager FB is automatically instantiated in the AsyviewAccessMode1 FB.

### 5.2.1. Introduction to the protocol

This section briefly introduces the protocol and the communication asynchronous mode.

The protocol manager takes care of coding and decoding the string and messages exchanged between the machine controller and the Asyview. You do not necessary need to understand this in detail but here below is the general description of the protocol.



**Figure 5-1: Protocol description**



*More detail can be found in the SmartSight documentation.*

The native communication mode is the asynchronous mode. Here is a quick introduction. The request will launch the corresponding action. The first response is a request acknowledge. The result of the action is sent after the end of the execution by a callback.

If necessary, it is possible to ask about the status of the executed action by query with the command type “/?” and the corresponding PID.

The associated command types for the asynchronous mode are:

- Asynchronous Call: “/!”
- Asynchronous Query (interrogation): “/?”
- Asynchronous CallBack: “/#”

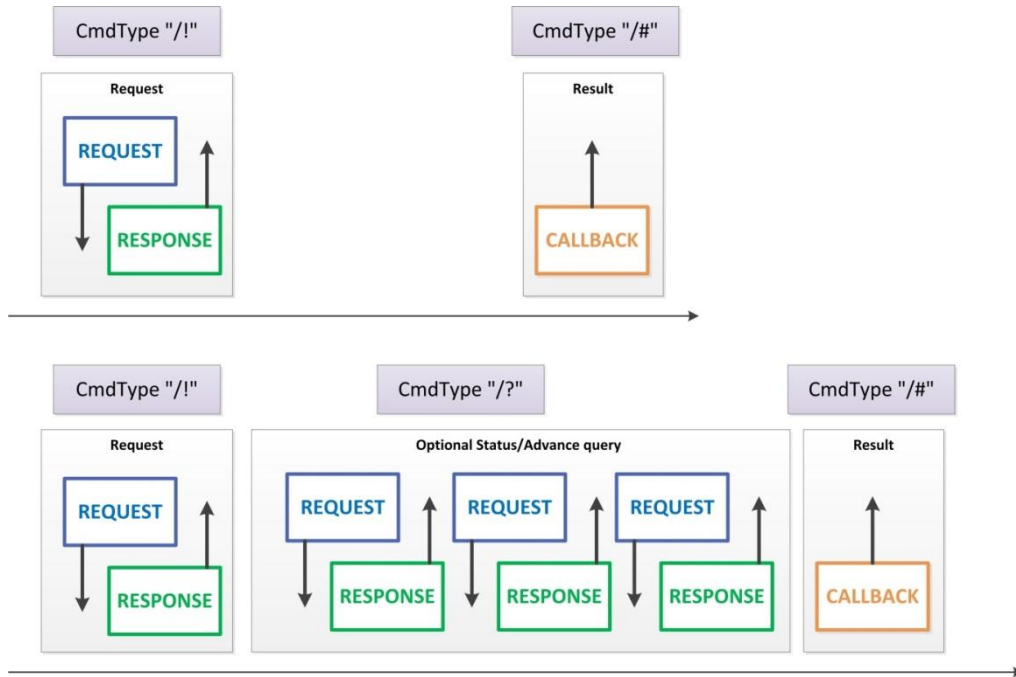



Figure 5-2: Asynchronous mode with and without interrogation of the status.

### 5.3. comChannel Function Block (FB)

The comChannel is a Function Block hidden for the end-user. However, it may be interested to understand its working principle for debug purpose. The comChannel has the following structure:

	Name	Type	
INPUTS	sSrvNetID	T_AmsNetID	
	sRemoteHost	T_IPv4Addr	
	nRemotePort	UDINT	
	tConnectionReconnect	TIME	
	tSendReceiveReconnect	TIME	
	enable	BOOL	
	dataToSend	STRING(255)	
	sendLength	UDINT	
	OUTPUTS	dataReceived	STRING(255)
		isConnected	BOOL
isError		BOOL	
isReadyToConnect		BOOL	
	isReadyToConnect	BOOL	

 Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyriil S.A.
000.100.522		Version: A1

	isReceiveReady	BOOL
	error	ALERT

One comChannel FB is automatically instantiated in the AsyviewAccessModel FB.

The communication channel manages the TCP/IP communication with the Asyview. It uses the FB\_ClientServerConnection connection from the Tc2\_Tcplp library for the lower level. It takes care of sending and receiving data on this channel.

## 5.4. Administrative functions (FUN)

The following functions can be useful.

### 5.4.1. External use

`errorToString`: takes an ALERT as input and returns a 255 characters string

`stringToMessageResult`: takes a message string as input and returns a MESSAGE\_RESULT containing the position and identification of the localized part

### 5.4.2. Internal use

`callTypeToString`

`messageStandardParameterToString`

`messageToString`

`msgTypeToString`

`stringToMessage`

`stringToMessageStandardParameter`


`newError`

`clearError`

## 5.5. Types

The following types have been created for this library:

- ALERT (STRUCT): error/alert identification and description
- ALERT\_TYPE (ENUM): error/alert severity
- CALL\_TYPE (ENUM): asyview message call type
- MESSAGE (STRUCT): message content including type and address
- MESSAGE\_RESULT (STRUCT): position and identification of a part localized by the Smartsight system
- MESSAGE\_STANDARD\_PARAMETER (STRUCT): advanced communication parameters
- MSG\_TYPE (ENUM): request or response

 Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyriil S.A.
000.100.522		Version: A1

- SYS\_ID (ENUM): show the system concerned by the error/alert

You will find their content in the library code.

## 5.6. Alert list

Here below it the error/alert list that can happen in the plugin. Advanced information is available on the ALERT output of the related Function Block.

code	cmpt	sub address	description
1	unknown	unknown	unknown error
2	com channel	connection	unable to connect
3	com channel	receive	error while receiving
4	com channel	send	error while sending
5	com channel	send	data size too big
6	com channel	receive	data size too big
7	com channel	connection	connection lost
8	com channel	receive	no connection, cannot start
9	com channel	send	no connection, cannot start
10	com channel	receive	cannot start, transition not allowed
11	com channel	send	cannot start, transition not allowed
12	com channel	abort	cannot abort, transition not allowed
13	protocol manager	send	type not implemented
14	general		not implemented
15	protocol manager	send	unknown message call_type
16	protocol manager	receive	decoding error: call_type
17	protocol manager	receive	too many messages in a packet
18	protocol manager	receive	incoherent data size
19	protocol manager	receive	decoding error: msg_type
20	protocol manager	receive	decoding error: address
21	protocol manager	receive	decoding error: keyword
22	protocol manager	abort	cannot abort, transition not allowed
23	asyview access model	send	cannot send, transition not allowed
24	asyview access model	receive	error message received
25	com channel	reset	cannot reset, transition not allowed
26	protocol manager	execute	cannot execute, transition not allowed
27	protocol manager	reset	cannot reset, transition not allowed
28	asyview access model	start	cannot start, transition not allowed
29	protocol manager	send	incoherent message msg_type
30	protocol manager	send	message too long, will be truncated

<b>asyril</b> Experts in Flexible Feeding Systems	Plugin Beckhoff Smartsight Integration Guide	© Copyright Asyriil S.A.
000.100.522		Version: A1

## 6. Technical Support

### 6.1.1. For a better service ...

Have you read the FAQ and the checklist and still not found an answer your questions?

Before contacting us, please note down the following information concerning your product:


- Serial number and product key for your equipment
- Software version(s) used
- Error message, alarm, or visual signals displayed by the interface.

### 6.1.2. Contact

You can find extensive information on our website: **[www.asyril.com](http://www.asyril.com)**

You can also contact our Customer Service department:

**[support@asyril.com](mailto:support@asyril.com)**

 <b>asyril</b> <small>Experts in Flexible Feeding Systems</small>	<b>Plugin Beckhoff Smartsight Integration Guide</b>	© Copyright Asyriil S.A.
000.100.522		Version: A1

## Revision table

Rev.	Date	Author	Comments
A	14.12.2015	PeD	Initial version
A1	21.03.2018	GuB	V2 compatible with Asyview V4, adding error 30



This document is the property of Asyrl SA; it may not be reproduced, modified or communicated, in whole or in part, without our prior written authorization. Asyrl SA reserves the right to modify any information contained in this document for reasons related to product improvements without prior notice



Asyrl SA  
Z.I. le Vivier 22  
Ch-1690 Villaz-St-Pierre  
Switzerland  
Tel. +41 26 653 71 90  
Fax +41 26 653 71 91  
info@asyrl.com  
www.asyrl.com