**Interface-Description**

**STM IGEMS-Interface**

**Attachment** :

**Supplier** : NUM AG

Battenhusstrasse 16

Postfach Lustmühle

CH - 9053 Teufen

Telephone: 071 / 335 04 11

Telefax: 071 / 333 35 87

**Document** : Responsible V. Fleischer

Filename IGEMS-Interface-Description.docx

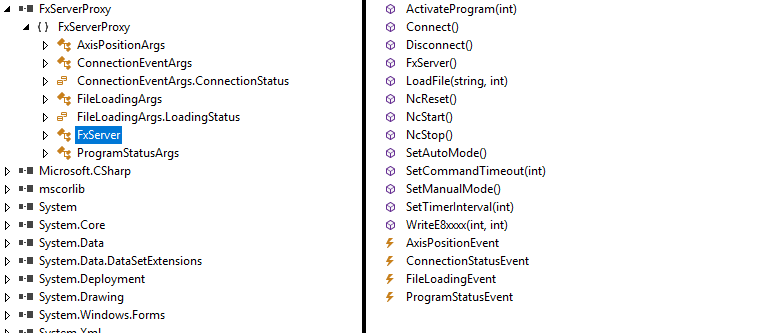
Date 5/15/201814.05.2018

Version V1\_0

# Concept

We supply a FxServer proxy class, to simplify the communication with a NUM controller. This proxy class is available in a C# assembly DLL.

The proxy class provides only the following methods and events:



# Method and Event description

## Connect / Disconnect

After the creation of the proxy class the Connect / Disconnect methods can be used to obviously connect or disconnect to the controller. It is important to correctly disconnect from the controller when exiting the application who uses the proxy class, so that all communication objects can be released.

Normally it is not needed to reconnect to the controller, after a reboot of the system. But some cases it might be needed to disconnect and reconnect again to the controller. If the standard user interface (HMI) is started already, the connection time is very short. Only if no other client is connected to the controller already the initialization of a new connection takes some seconds.

Once the connection is established an ConnectionStatusEvent is raised. Also when the connection is closed such an event is raised. Also after an successful connection to the controller continuous AxisPositionEvents will be raised, as soon as there is a change in axis position values.

After the initial connection to the controller the proxy class also tries automatically to connect to some symbolic variables for handling the commands to the controller and the ProgramStatusEvents. If it fails to connect to the predefined variables an ConnectionStatusEvent with an appropriate argument is raised.

Before connecting to the controller it is possible to change the timeout period for sending commands to the controller by calling the method SetCommandTimeout. 300 milliseconds is the default value for this timeout. Within that timeout period the PLC should recognize a new command and react appropriately.

## Sending commands

To switch operation mode or to activate, start and stop a program the following methods can be used:

* NcStart()
* NcStop()
* NcReset()
* SetManualMode()
* SetAutoMode()

These methods will return 0 if executed without errors. Negative values indicate not meet conditions for execution like

* -1: not connected

Positive values will match the internal FxServer return values, if they occur during the call of the used function.

## Loading and activation of a program

To load a file to the controller the LoadFile method has to be used. It takes 2 arguments. A string for the filename were the file can be found on the loading PC and an integer value for the program number it has to be loaded to on the controller.

After a file has been loaded or any other file that already exists on the controller can be activated.

Therefore the method ActivateProgram(int prg) exists. As the argument it takes the program number which has to be activated.

These methods will return 0 if executed without errors. Negative values indicate not meet conditions for execution like

* -1: not connected
* -2: other file transfer active
* -3: program number currently running, transfer not started

Positive values will match the internal FxServer return values, if they occur during the call of the used function.

After an file has been loaded or activated an FileLoadingEvent will be raised as well as it has failed to load or been activated.

## Job activation

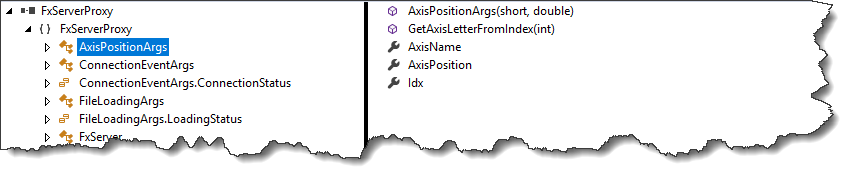
Specific for the application at STM is that on the NCK runs a “main program” in an endless loop. This main program does nothing then calling subprograms which are the end user programs or other small programs to do particular jobs. Once the current job is finished, a new job can be loaded and activated. The main program runs under program number 200 and the jobs will be loaded unter program number 201 or maybe subsequent numbers.

As said before, jobs can be loaded to a program number only if this program is not been executed. After it is loaded to the program memory the writing of an E-Parameter activates this job. Therefore the proxy class gives the ability to write an E-Parameter by calling the WriteE8xxxx method. Here it is needed to give the number for that E80000 parameter. 1 means E80001 and so on. The transferred value will define the subprogram number which shall be executed.

The activation of such job is not confirmed directly but by raising an ProramStatusEvent with that subprogram number as CurrentProgram.

## AxisPositionEvent

The transfer of the axis positions happens in an event called AxisPositionEvent. The object of the type AxisPositionArgs has 3 properties. AxisName, AxisPosition, Idx.

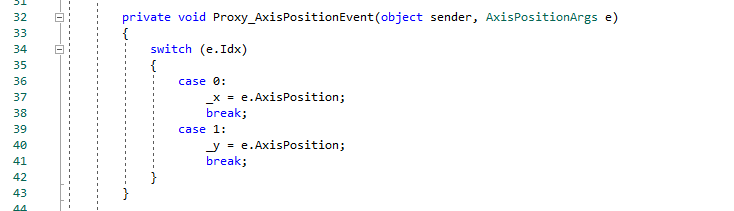


Idx: index of axis, 0 is always used for X, 1 for Y, 2 for Z, 3 for U,… 6 for A and so on

AxisName holds the letter for a given Idx

AxisPosition gives the position in mm or degree

As the event is called from a different thread as the main GUI thread, it is important that the handling is thread safe. Either copy the data on to thread safe variables



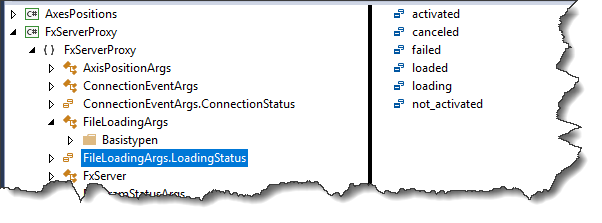
Or use thread safe access to windows forms:



For easy handling the only 1 axis is transferred in such an event. If more axis change their position at same time, multiple events are raised.

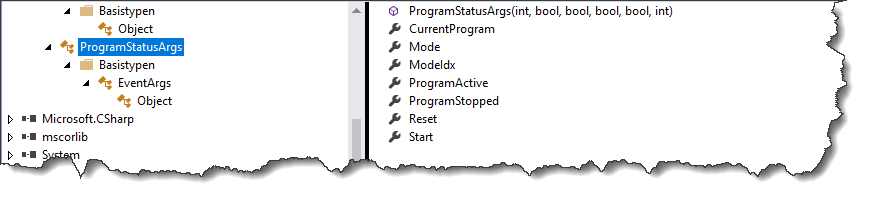
## FileLoadingEvent

Loading, activation or abortion of such an activity raise an FileLoadingEvent. The enumeration of that loading status can hold one of the following values.



## ProgramStatusEvent

In an ProgramStatusEvent the current status of the controller is published.



* CurrentProgram shows which program is currently activated or running
* Mode shows the current user mode (AUTO or HAND or MDI etc)
* ModeIdx shows the index number of that Mode
* Start shows that NC Start is active
* ProgramActive / ProgramStopped shows that a program is active or stopped (M0).
* Reset shows that a Reset is happening on the NCK

# Sample code to work with the proxy class





# Modifikationen

|  |  |  |  |
| --- | --- | --- | --- |
| **SW-Version:** | **Datum:** | **Autor** | **Kommentar:** |
| **V1\_0** | 15.05.201 | V. Fleischer |  |

**Änderungen:**