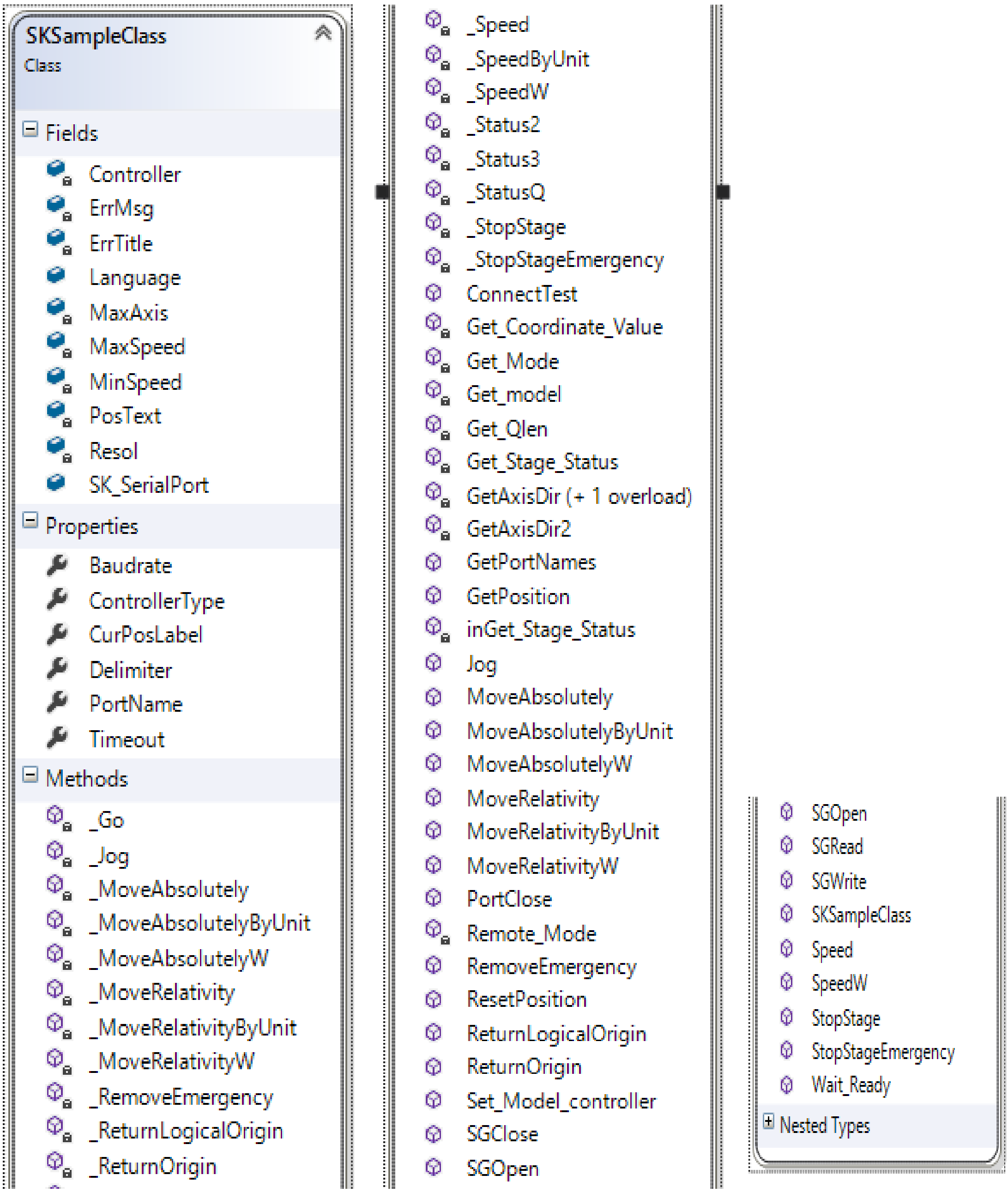
Sigma Koki 30th September 2020

FC DotNet Class

FC DotNet Class is VB and C# class for Controller, Stages and functions of the main common commands of Sigma Koki FC Series controller. It supports 1 to 2 Axis stages. Controllers: FC 911, FC 611, FC511, FC 411, FC 111. By including SKSampleClass (DotNet program Class) in any projects (.net) Microsoft VS2015, allows you to use simple commands operation in case of Sigma Koki FC Series controllers. Use this class to customize your program on your own, Sigma Koki is not responsible in case of edit, add and delete. A source program (Windows Form type) is provided with that class to simplify the usability of functions and commands for both VB and C# classes. Program contains a simple Terminal commands in order to send Strings through Serial Port.

The package Class (SKSampleClass.CS or SKSampleClass.VB)

1. SKSampleClass：Stages Class Diagram



SKSampleClass
Class

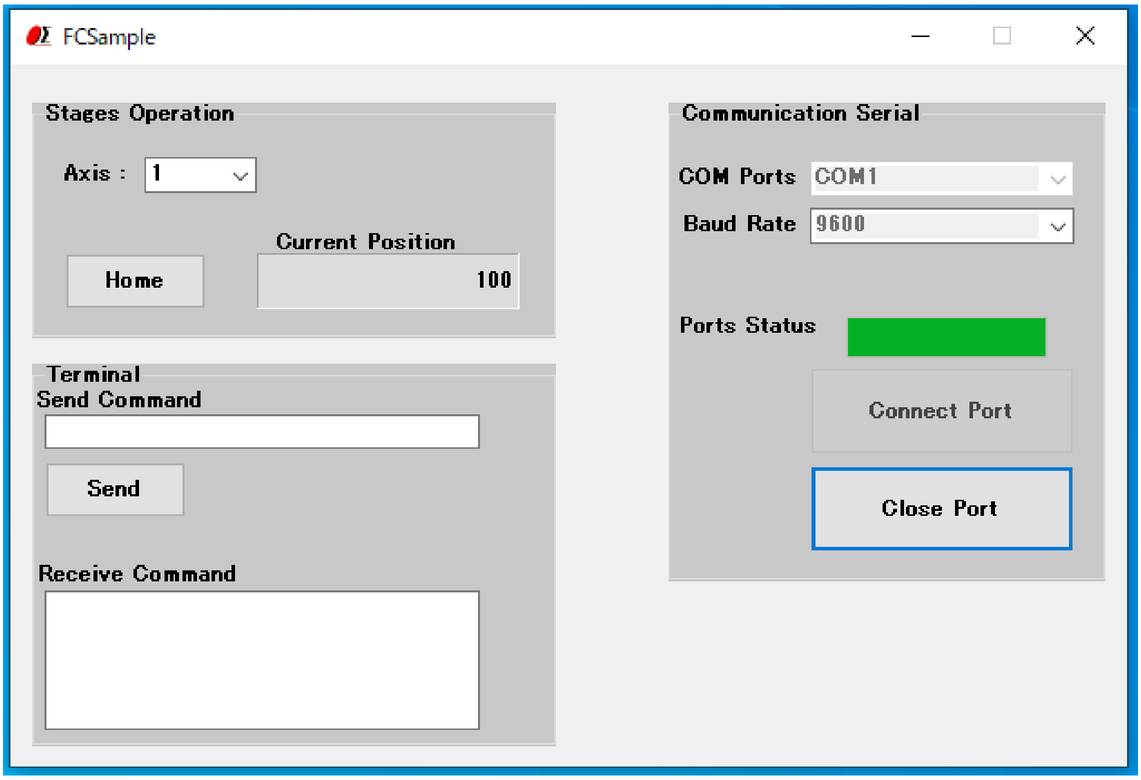
Fields
- Controller
- ErrMsg
- ErrTitle
- Language
- MaxAxis
- MaxSpeed
- MinSpeed
- PosText
- Resol
- SK_SerialPort

Properties
- Baudrate
- ControllerType
- CurPosLabel
- Delimiter
- PortName
- Timeout

Methods
- _Go
- _Jog
- _MoveAbsolutely
- _MoveAbsolutelyByUnit
- _MoveAbsolutelyW
- _MoveRelativity
- _MoveRelativityByUnit
- _MoveRelativityW
- _RemoveEmergency
- _ReturnLogicalOrigin
- _ReturnOrigin
- _Speed
- _SpeedByUnit
- _SpeedW
- _Status2
- _Status3
- _StatusQ
- _StopStage
- _StopStageEmergency
- ConnectTest
- Get_Coordinate_Value
- Get_Mode
- Get_model
- Get_Qlen
- Get_Stage_Status
- GetAxisDir (+ 1 overload)
- GetAxisDir2
- GetPortNames
- GetPosition
- inGet_Stage_Status
- Jog
- MoveAbsolutely
- MoveAbsolutelyByUnit
- MoveAbsolutelyW
- MoveRelativity
- MoveRelativityByUnit
- MoveRelativityW
- PortClose
- Remote_Mode
- RemoveEmergency
- ResetPosition
- ReturnLogicalOrigin
- ReturnOrigin
- Set_Model_controller
- SGClose
- SGOpen
- SGOpen
- SGRead
- SGWrite
- SKSampleClass
- Speed
- SpeedW
- StopStage
- StopStageEmergency
- Wait_Ready

Nested Types

Main Window:



Axis : 1

Current Position 100

COM Ports COM1

Baud Rate 9600

SKSamplClass Class contains SK Commands used FC Series controller, the class containing serial port resource in order to use it in the other programs.

Note that the class has the most relevant commands as SGWrite, SGread, SGOpen, SGClose related to serial port resource.

The class contains WaitReady() function in order to work with real time response of Stage status.

Commands Description:

//Home Return Origin

Content		Function
Home	VB	ReturnOrigin(int Axis) Axis: 1～2, Axis = 0 (All axis)
	C#	ReturnOrigin(int Axis) Axis: 1～2, Axis = 0 (All axis)

Example:

VB

ReturnOrigin(1)

_Home Origin operation for axis 1.

//Move relativity for Axis 1 & 2

Content		Function
Move (Relativity)	VB	MoveRelativity(int Axis, int Vdata) Axis: 1 or 2 Vdata: Positive or Negative Pulses
	C#	MoveRelativity(int Axis, int Vdata) Axis: 1 or 2 Vdata: Positive or Negative Pulses

Example:

C#:

MoveRelativity (2, -5000)

_Relative Move of second axis with pulses 5000

//Move relativity for both axis together

Content		Function
Move (Relativity)	VB	MoveRelativityW(int Axis, int Vdata1, int Vdata2) Axis: 0 Vdata1 & Vdata2: Positive or Negative Pulses
	C#	MoveRelativityW(int Axis, int Vdata1, int Vdata2) Axis: 0 Vdata1 & Vdata2: Positive or Negative Pulses

//Move relativity by Unit

Content		Function
Move (Relativity)	VB	MoveRelativityByUnit(int Unit, int Axis, double Vdata1) Unit: 1～4　　(1: Nanometer, 2: Micrometer, 3: Millimeter, 4: Degree) Axis: 1 or 2 Vdata1: Positive or Negative Pulses
	C#	MoveRelativityByUnit(int Unit, int Axis, double Vdata1) Unit: 1～4　　(1: Nanometer, 2: Micrometer, 3: Millimeter, 4: Degree) Axis: 1 or 2 Vdata1: Positive or Negative Pulses

Example:

C#

MoveRelativityByUnit(1, 2, 50)

_Relative move of 2nd axis with 50 nanometer.

//Move absolutely for axis 1 & 2

Content		Function
Move (Absolute)	VB	MoveAbsolutely(int Axis, int Vdata) Axis: 1 or 2 Vdata: could be Positive or Negative Pulses
	C#	MoveAbsolutely(int Axis, int Vdata) Axis: 1 or 2 Vdata: could be Positive or Negative Pulses

Example:

C#:

MoveAbsolutly (1, 3000)

_Absolute move of 1st axis with 3000 pulses

// Move absolutely for both axis together

Content		Function
Move (Absolute)	VB	MoveAbsolutelyW(int Axis, int Vdata1, int Vdata2) Axis: 0 Vdata1 & Vdata2 : Positive or Negative Pulses
	C#	MoveAbsolutelyW(int Axis, int Vdata1, int Vdata2) Axis: 0 Vdata1 & Vdata2 : Positive or Negative Pulses

// Move absolutely by unit

Content		Function
Move (Absolute)	VB	MoveAbsolutelyByUnit(int Unit, int Axis, int Vdata1) Unit: 1～4 (1: Nanometer, 2: Micrometer, 3: Millimeter, 4: Degree) Axis: 1 or 2 Vdata1: Positive or Negative Pulses
	C#	MoveAbsolutelyByUnit(int Unit, int Axis, int Vdata1) Unit: 1～4 (1: Nanometer, 2: Micrometer, 3: Millimeter, 4: Degree) Axis: 1 or 2 Vdata1: Positive or Negative Pulses

// Logical origin setting

Content	Function
Set Position 0	ResetPosition(int Axis)

// ReturnLogicalOrigin

Content		Function
Return Logi	VB	ReturnLogicalOrigin(int Axis, int cPos)
	C#	ReturnLogicalOrigin(int Axis, int cPos)

// Stop emergency

Content	Function
Emergency Stop	StopStageEmergency() All Axis

// decelerate and Stop Stage 1～4

Content	Function
Stop	StopStage(Axis) Axis: 1～4,

// Set Speed

Content		Function
Speed Set	VB & C#	Speed(int Axis, double Fast) Axis: 1～2 Fast: Speed (Max and Min depends the controller).

Example:

Speed(1, 2000)

_Set speed for 1 Axis with related speed (2000)

// set speed for both axis together

Content		Function
Speed Set	VB & C#	SpeedW(int Axis, double Fast1, double Fast2) Axis: 0 Fast1 & Fast2 : Speed　Max and Min depends the controller.

Content		Function
Speed Set	VB & C#	SpeedByUnit(int Unit , int Axis, double Fast) Unit: 1～4 (1: Nanometer, 2: Micrometer, 3: Millimeter, 4: Degree) Axis: 0 Fast1 & Fast2 : Speed　Max and Min depends the controller.

//Set the model of controller

Content	Function
Model	Set_Model_controller()

See SKSampleClass to use the other commands

For ease of use, the provided project includes a C # Windows Forms application that allows users to retrieve commands from the classe and use them in your own programs.