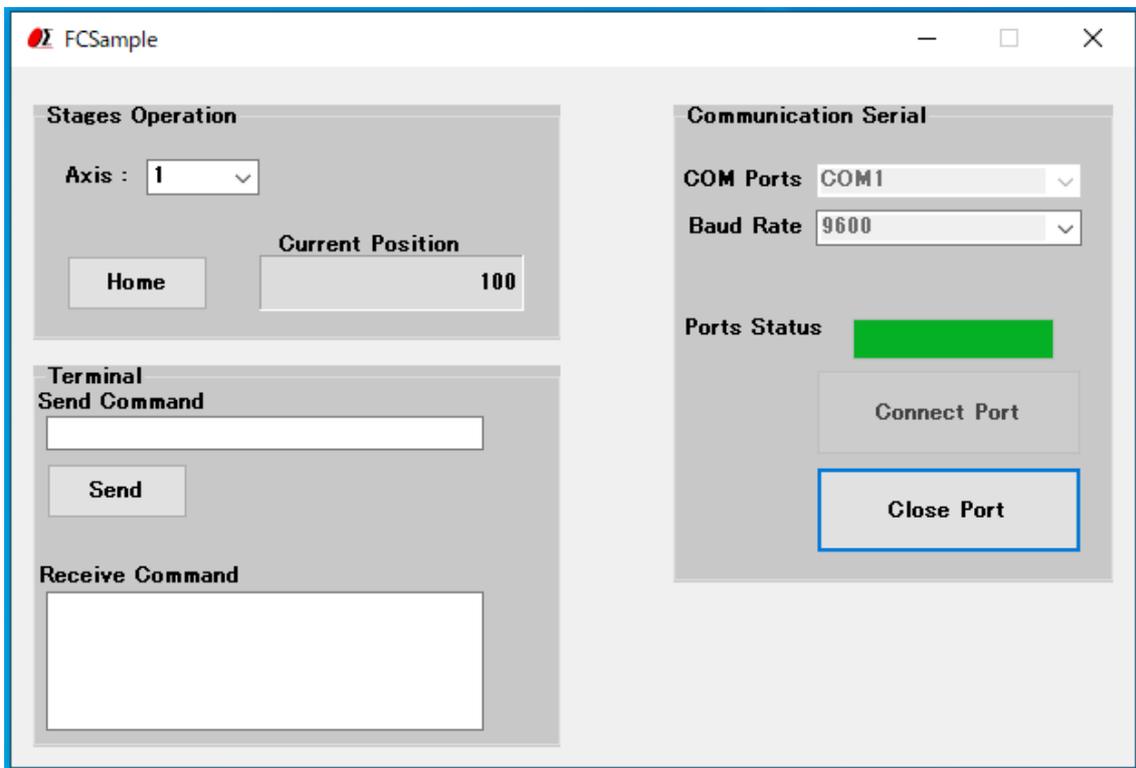


Sigma Koki 30th September 2020

FC Class C++

SKSampleClass Class is C++ (.H and .Cpp) command class of the main common commands of Sigma Koki Controller. It supports 1 to 2 Axis stages. Controllers: FC 911, FC 611, FC511, FC 411, FC 111. By including SKSampleClass Class in any C++ projects Microsoft VS2015, allows you to use simple commands operation in case of Sigma Koki FC Series controllers. Use this SKSampleClass class and customize your program on your own, Sigma Koki are 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 of C++ class. Program contains a simple Terminal commands in order to send Strings through Serial Port.

Main Window:



Class description:

SKSAMPLClass (.h and .cpp) Class contains SIGMA KOKI 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	<code>SKSampleClass::ReturnOrigin(int Axis)</code> Axis: 1~2, Axis = 0 (All axis)

Example:

`ReturnOrigin(1)`

_Home Origin operation for axis 1.

//Move relativity

Content	Function
Move (Relativity)	<code>SKSampleClass::MoveRelativity(int Axis, int Vdata)</code> Axis: 1 or 2 Pulses: Positive or Negative

Example:

`SKSampleClass->MoveRelativity (2, -5000)`

_Relative Move of second axis with pulses 5000

//Move relativity for both axes together

Content	Function
Move (Relativity)	<code>SKSampleClass::MoveRelativityW(int Axis, int Vdata1, int Vdata2)</code> Axis: 0 Vdata1 & Vdata2: positive or negative pulses of axis 1 and axis 2

//Move relativity by unit

Content	Function
Move (Relativity)	<code>SKSampleClass::MoveRelativityByUnit(int Unit, int Axis, double Vdata1)</code> Unit: 1~4 (1:Nanometer, 2: Micrometer, 3: Millimeter, 4: Degree) Axis: 1 or 2 Vdata1: positive or negative pulses

//Move absolutely

Content	Function
Move (Absolute)	<code>SKSampleClass::MoveAbsolutely(int Axis, int Vdata)</code> Axis: 1 or 2 Pulses: could be Positive or Negative Value

Example:

`SKSampleClass->MoveAbsolutely (1, 3000)`

_Relative Move of first axis with pulses 3000

//Move absolutely for both axes together

Content	Function
Move (Absolutely)	<code>SKSampleClass::MoveAbsolutelyW(int Axis, int Vdata1, int Vdata2)</code> Axis: 0 Vdata1 & Vdata2: positive or negative pulses of axis 1 and axis 2

//Move absolutely by unit

Content	Function
Move (Absolutely)	<code>SKSampleClass::MoveAbsolutelyByUnit(int Unit, int Axis, double Vdata1)</code> 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	<code>ResetPosition(Axis)</code>

// ReturnLogicalOrigin

Content	Function
Return Logi	<code>ReturnLogicalOrigin(Axis, cPos)</code>

// Stop emergency

Content	Function
Emergency Stop	StopStageEmergency() All Axis

// decelerate and Stop Stage 1~4

Content	Function
Stop	StopStage(Axis) Axis: 1~2

// Remove emergency

Content	Function
Remove Emergency	SKSampleClass::RemoveEmergency()

// Set Speed

Content	Function
Speed Set	SKSampleClass::Speed(int Axis, int Fast) Axis: 1~2 Fast: Speed (Max and Min depends the controller).

Example:

SKSampleClass->Speed(1, 2000)

_Set speed for 1 Axis with related speed (2000)

// Set Speed both axis together

Content	Function
Speed Set	SKSampleClass::SpeedW(int Axis, double Fast1, double Fast2) Axis: 0 Fast: Speed (Max and Min depends the controller).

// Set Speed by unit

Content	Function
Speed Set	SKSampleClass::SpeedByUnit(int Unit, int Axis, double Fast) Unit: 1~4 (1:Nanometer, 2: Micrometer, 3: Millimeter, 4: Degree) Axis: 1~2 Fast: Speed (Max and Min depends the controller).

//Set the model of controller

Content	Function
Model	SKSampleClass::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 class and use them in your own programs.