

Industrial Stepping Motors Integration in UNICOS

J. Fernadez Cortes, E. Blanco Vinuela, L. Gonzalez Gomez, CERN, Geneva, Switzerland

MOTIVATION

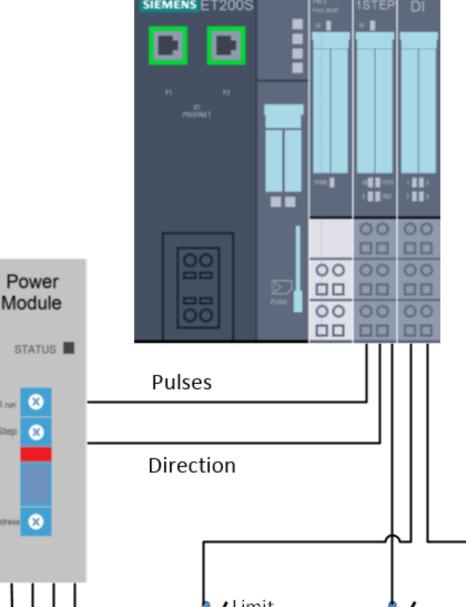
• Legacy installations running non-standardized control applications.

- Support needed high level of expertise.
- Reduced functionalities and flexibility.
- Creation of new projects was a laborious process.

OBJECTIVES

- Projects standardization
- Flexibility to support distinct configurations and architectures

HARDWARE SCHEME

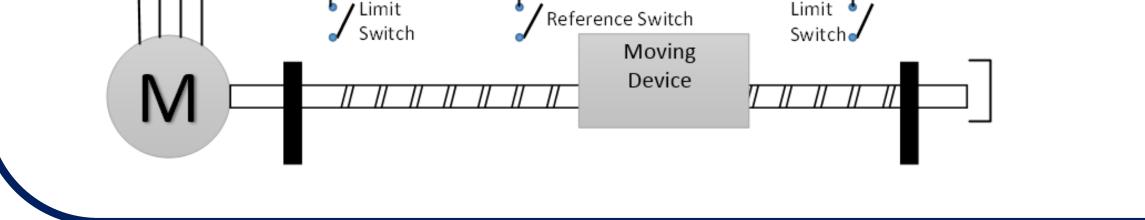


• **SIEMENS** Stepper Motor Module **1STEP** with the Reference Switch input

THPHA144

- Standard decentralized digital Inputs to acquire the switches (2)
- External power module dimensioned according the size of the motor.
- Allows diverse configurations and encoders

- Ease diagnostics and maintenance
- Reduce engineering time and domain expertise



UNICOS-CPC FRAMEWORK INTEGRATION

STEPPING MOTOR DEVICE TYPE DESIGN UCPC # 3SwENC v1.0 # Wizard v1.8.0 CPC Wizard: 3_Sw_&_ENC - 3SwENC v1.0 Web Documentation · 🎟 🖶 🐟 🐭 🗳 💕 🖆 👯 S7 Instance Generator Resources: 1.8.0-alpha-04 SteppingMotor) SteppingMotorDeviceType - 🔿 Information ManReg01 StsReg01 🗠 🛕 DeviceIdentification Meaning General Data 🕜 ManualRequests ___ MPosR StsReg02 🗠 🛕 DeviceDocumentation Parameter Maximum Value Type Template Folder: C:\Motion-Projects\Labo864\3_Sw_&_ENC\3_Sw_ 🖳 🗛 FEDeviceParameters ___ MSpdR CWRef ---- 🔿 UserExpandable 🛛 true SimRef Output Folder: C:\Motion-Projects\Labo864\3_Sw_&_ENC\3_Sw ___ ParReg 🔖 🔥 ParReg DrvEn Description PRefPos 👜 🛆 PRefPos PosSt Post Process User Template: Browse Clear 🗄 🛆 PMinRan 🛛 _ PMinRan Highest Position that the device car SpdSt - 🛆 PMaxRan ___ PMaxRan Post Process User Process Semantic Rules AuPosRSt - O Meaning Parameter Maxin ___ PS ca Description Highest Position t AuSpdRSt ____ POffset --- <> PrimitiveType FLOAT32 Global Files To Generate 🛛 🚱 MPosRSt _ ____ PMaxSpd ---- O DefaultValue 0.0 MSpdRSt ____ S7 S7 Communication: CompilationInstances: Sch Sch WinCC WinCC ___ PWDt 🗄 🔿 isSpecificationAttribute PrimitiveType ТСТ PosRSt ____ ___ PWDb 🗄 🗛 PSca Logic Flexible Inst Inst Global fil Logic CompilationBaseline SpdRSt 👜 🛆 POffset FLOAT32 ____ PDbT HFPosSt ____ 🚋 🔥 PMaxSpd Generate Symbols File 🛛 IO Commissioning PMinRanSt ____ AuPosR 🔖 🛆 PWDt 🎰 🛆 PWDb DefaultValue AuSpdR Symbols: Genera PMaxRanSt 🗄 🗠 🛆 PDbT ____ AuRefSR PScaSt 0.0 EDeviceInterlocks UNICOS Types To Generate 👩 AuStopR PosModeSt EDeviceAutoRequests AuNewConfigR RefSModeSt _____)utputs 🗠 🗛 FEDeviceManualRequests Device Type Instances _____ AuOffsetR HFCWSt AutoRequests 🗠 🗛 FEDeviceEnvironmentInput AnaDO Analog AnalogAlarm XML AnalogDigital **Device Type Definition** AnalogInput AnalogInputReal AnalogOutput <AttributeFamilyName>FEDeviceParameters</AttributeFamily</pre> **UAB (UNICOS application builder)** AnalogOutputRea <UserExpandable>true</UserExpandable> <Attribute AttributeName>ParReg</AttributeNa eaning>Parameter Register</Meaning (Description>Parametrisation register: This register contains all the boolean parameters <PrimitiveType>SHORTINT16</PrimitiveType: x II <Attribute> <AttributeName>PFsPosOn</AttributeN</pre> **Touch Panel Generation** Meaning>Parameter Fail-Safe position ON/Open</Meaning Description>Fail Safe position of the actuator</Descripti **DeviceDocumentation** PrimitiveType>BIT1</PrimitiveType 3 4 BitPosition>0</BitPosition</pre> ParReg 58 Motor01.visible <No Value> <No Value> Bool SteppingMotor isSpecificationAttribute> Feedback Switches Configuration Manual Restart after Full Stop Remarks Description 59 Motor01.objNam SteppingMotor <No Value> <No Value> WStrin NameRepresentation>Fail-Safe</Nam CP_SLH0107_A Position contorl for CP 0107 collimato No End Switche <TypeRepresentation>STRING</TypeRepresentation> 60 Motor01.unit <No Value> <No Value> WString SteppingMotor <isValueRequired>true</isValueRequired> CP_SLH0107_P Aperture contorl for CP 0107 collimator No End Switches. Encoder FALSE 61 Motor01.setValue <No Value> <No Value> Bool SteppingMotor <Usage>This is the position of the device in case of interlock.</Usage CL_SLH0445_P Position control for CL 0445 Collimator FALSE No End Switches Encoder 2 Motor01.setSpee SteppingMotor <No Value> <No Value> Bool

CONT	ROL SYS ⁻	TEM DESIG	
UCPC # 3SwENC v1	.0 # Wizard v1.8.0		DATA_BLOCK Motor01 CPC_FB_STPMOT
CPC With S7 Inst Resourt General Data (2) Type Templat	zard: 3_Sw_&_ENC - 3SwEN tance Generator <i>rces: 1.8.0-alpha-04</i> te Folder: C:\Motion-Projects\Labo84	C v1.0 54\3_Sw_&_ENC\3_Sw_&_ENC_UAB\Resou 54\3_Sw_&_ENC\3_Sw_&_ENC_UAB\Outpu	<pre>PSTPMOT.ParReg := 2#0000010001000000; PSTPMOT.PRefPos := 0.0; PSTPMOT.PMinRan := 0.0; PSTPMOT.PMaxRan := 100.0; PSTPMOT.PSca := 500.0; PSTPMOT.POffset := 0.0; PSTPMOT.POffset := 0.0; PSTPMOT.PMaxSpd := 27; PSTPMOT.PWDt := T#5.0s; PSTPMOT.PWDt := 0.1;</pre>
Post Process User T Process Semant		Post Process User Template:	PSTPMOT.PDbT := 0.5;
Global Files To Gener Communic Compilation Generate Symbols Fil	cation: 🔽 CompilationInstar Baseline: 🔽 CompilationOE		<pre>IF SM_SL_HORIZONTAL.OnSt = TRUE THEN IF SM_SL_HORIZONTAL.OpMoSt = 1.0 THEN IF SM_SL_LEFT.PosModeSt THEN SM_SL_LEFT.AuPosR := DB_AS_ALL.AS_SET.SM_SL_LEFT_Pos.PosSt; ELSE SM_SL_LEFT.AuPosR := SM_SL_LEFT.PosSt;</pre>
UNICOS Types To Ge			END_IF;
Device Type AnaDO Analog AnalogAlarm AnalogDigital AnalogInput AnalogInputReal AnalogOutput AnalogOutput		Instances 0 0 0 0 1 0 0 0 0 0 0 0 0 0	<pre>IF SM_SL_RIGHT.PosModeSt THEN SM_SL_RIGHT.AuPosR := DB_AS_ALL.AS_SET.SM_SL_RIGHT_Pos.PosSt; ELSE SM_SL_RIGHT.AuPosR := SM_SL_RIGHT.PosSt; END_IF; SM_SL_LEFT.AuRefSR := FALSE; SM_SL_RIGHT.AuRefSR := TRUE; SM_SL_LEFT.AuRefSR := TRUE; SM_SL_RIGHT.AuRefSR := TRUE; SM_SL_LEFT.AuRefSR := SM_SL_LEFT.PosSt; SM_SL_RIGHT.AuPosR := SM_SL_RIGHT.PosSt; </pre>
ch Panel Generation	n 2 3	4 5 6 V13	SCADA Importation
or01.visible or01.objName or01.unit	SteppingMotor <no td="" value<=""> SteppingMotor <no td="" value<=""> SteppingMotor <no td="" value<=""> SteppingMotor <no td="" value<=""></no></no></no></no>	> <no value=""> Bool 1 <no value=""> WString 40 <no value=""> <no value=""> WString 10 <no value=""></no></no></no></no></no>	<pre># Object: SteppingMotor #Config Line : CPC_SteppingMotor;deviceNumber;Alias[,DeviceLinkList];Description;</pre>
tor01.setValue		> <no value=""> Bool 1 <no value=""></no></no>	CPC_SteppingMotor;1;SM_SL_UP,SM_SL_VERTICAL;Stepping motor vertical slit upper;;;

CDependentAtt	ribules/>
<constraints <="" td=""><td>></td></constraints>	>
<td>Attribute></td>	Attribute>

	17	CL_SLH0445_A	Aperture contorl for CL 0445 collimator	No End Switches.	Encoder	FALSE
_ \	18	CK_MKS14_ATT	Attenuation Control for MKS14	No End Switches.	Encoder	FALSE

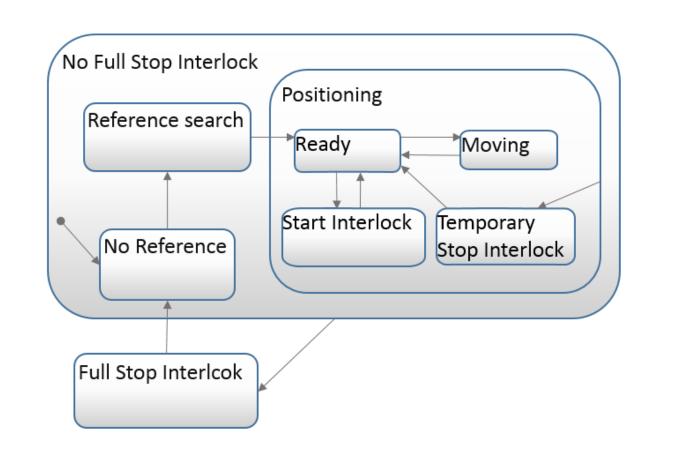
	•	11 0					
63	Motor01.StsReg01	SteppingMotor	PLCCOIS01	<no value=""></no>	Word	2	%DB282.DBW0
64	Motor01.StsReg02	SteppingMotor	PLCCOIS01	<no value=""></no>	Word	2	%DB282.DBW2
65	Motor01.ManReg01	SteppingMotor	PLCCOIS01	<no value=""></no>	Word	2	%DB289.DBW0
66	Motor01.PosSt	SteppingMotor	PLCCOIS01	<no value=""></no>	Real	4	%DB287.DBD0

<No Value> CPC SteppingMotor:2:SM_SL_DOWN.SM_SL_VERTICAL:Stepping motor vertical slit lower: :SM SUP FRONT: Stepping motor positioning front: :: : EuroCirCol ::

MODEL-BASED DESIGN

• Program code designed based on state machines

• Models help to have a clear program structure



Specification of the instance

lain State		
StateName	CASE StateIndex OF	IF StateIndex = State2Name THEN
en:	StateName:	transition actions;
entry_actions;	IF first time in state THEN	END IF;
du: during_actions;	entry actions;	IF StateIndex = State3Name THEN
ex:	first time in state := FALSE;	transition actions;
exit_actions;	ELSE	END_IF;
ondition1] 12	during_actions;	
ransition_actions}	END_IF;	IF StateIndex <> StateName THEN
State2Name	IF condition1 THEN	exit_actions;
Statezhame	StateIndex := State2Name;	first_time_in_state := TRUE;
[condition2]	ELSIF condition 2 THEN	END_IF;
/{transition_actions}	StateIndex := State3Name;	
State3Name	END_IF;	

USE CASES

CTF3

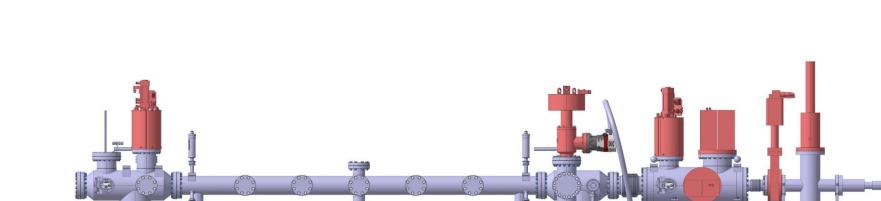
- Hardware: eight motors running in the CTF3 Complex.
- **Operated** with "Knobs" with the experiment control system (through FESA based application)
- Different installations with diverse wiring, controllers and encoders.



KnobsOpen application - SCT.USER.SETUP - (INCA) 25 Sep 2017 09:42:18 SCT - 00 SETUP UNDEFINED 63/128 		FMB_Ph	VGP1 VGR1
CA.MOV0870-H Command Mode POSITIONING Status1 Status2 Requested Position Ref		Back Motor	
Init 4300.00 Requested Position 4300.00 VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV		PCOs Vertical PCO POSITIONING M	Motors
No Exception to display 09:41:41 - Opening knob: CA.MOV0870-Hdone	Model of the ANKA SET UP by Miguel Gil Costa. CERN. Geneva. Switzerland.	Horizontal PCO POSITIONING M	Right 24.3 24.3 Left 31.4 31.4 31.4

ANKA

- Hardware: six stepping motors running in the ANKA experiment.
- **Operated** directly using the WinCC OA SCADA provided by UNICOS
- Integrated with the vacuum control system using the UNICOS -CPC Vacuum package.





Speeds

Gap Size

Offsets

Not Ready Out of Range

OUTCOME

STANDARDIZATION

 Extended the functionality of the **UNICOS-CPC** framework: **motion** with stepping motors.

DEVELOPMENT

- Reduced the time of developing and deploying motion projects. - Flexibility with different architectures and configurations.

SUPPORT

- Optimized support to installations using standard solutions with already existing diagnostic tools.

Beams department Industrial Controls & Safety Systems Group

