

# CODESYS

<b>Neues Projekt erstellen</b>	Datei/Neu XV-2xx-V2.3.9 Programm + <Sprache>
<b>Bei Änderung</b>	Projekt/Alles übersetzen/
<b>Simulation</b>	Online/Simulation Online/Einloggen Online/Start Variablen ändern Daten aktualisiert (Forcen, F7)
<b>Hardware definieren</b>	Online/Ausloggen Ressource – Steuerungskonfiguration RMT auf Configuration/Unterelement anhängen/DP-Master DP Parameter/Höchste Stationsadresse 15 RMT auf DP-Master/Unterelement anhängen/XNE-GWBP-PBDP V4.51 Ein-/Ausgänge/Eingansmodule/XNE-8DI24VDC-P /Ausgangsmodule/XNE-8DO-24VDC-0.5A (0.5Ampère pro Ausgang)
<b>Interface</b>	Online/Ausloggen Visualisierung - RMT auf Visualisierung/Objekt einfügen/<Test>
<b>Button</b>	Button = Rechteck einfügen und doppelklicken Text ändern: /Text/<Eingang 1> Farbe ändern: /Farben/Innen (Farbe wenn off) /Farben/Alarmfarbe (Farbe wenn on) Farbwechsel /Variablen/Farbwechsel/PLC_PRG.Eingang1 /Eingabe/PLC_PRG.Eingang1 toggle = Zustandänderung tasten = bei Niederdrücken
<b>LED</b>	Text ändern: /Text/<Ausgang 1> Farbe ändern: /Farben/Innen (Farbe wenn off) /Farben/Alarmfarbe (Farbe wenn on) Farbwechsel /Variablen/Farbwechsel/PLC_PRG.Ausgang1
<b>Textfeld</b>	Text anzeigen /Text/< t = %s> Variabel zuweisen /Variablen/Textausgabe/< Variable PLC_PRG.TON1.ET> (Elapsed Time)
<b>Traceaufzeichnung</b>	Ressourcen – Traceaufzeichnung Extras/Tracekonfiguration /Trigger Position [%]/20 "Unten alle Variablen definieren" Extras/Trace starten Extras/Trace automatisch lesen
<b>Lokale Variable</b>	VAR_INPUT var_enable : BOOL; var_mit_startwert_true : BOOL := TRUE; END VAR
<b>Global Variablen</b>	Ressourcen – Globale_Variablen (Definition identisch mit lokalen Variablen) In Visualisierung .A

**KOP (Kontaktplan)**

<b>Bausteine</b>	
Eingang	=Kontakt
Ausgang	=Spule
Funktionsblock	Einfügen/Funktionsblock/Definierte Funktionsblöcke/<FB_1> Name nicht gleich Blockname
I/O's Zuweisen	Eingang 1: BOOL; --> Eingang1 AT %IX0.0: BOOL;
return	Einfügen - Return (Für Abbruch des jeweiligen Funktionsblock)
Flankenerkennung	Flankenerkennung schaut auf Signal vom ganzen vorderen Pfad. Deshalb zuerst Flankenerkennung, dann alles andere!
Signalausbreitung	Signal breitet sich sehr schnell aus.

**ST (Strukturierter Text)**

<b>Neues Projekt erstellen</b>	identisch
<b>Funktionsblock</b>	im Hauptprogramm: Name des Objekts ("Sequenzler()"); [Enter]; VariablenTyp holen
<b>Kommentar</b>	(* ... *)
<b>Variabler Text in Button</b>	Zustand = %s
<b>Zeitverzögerung</b>	TP (kann nicht gestopt werden) TON, TOF (können gestopt werden)
<b>TON</b>	Deklaration: TONe: TON Aufruf: TONe(IN:=eStart, PT:=T#2s); Ausgang: TONe.Q
<b>Nicht Tasten sondern Toggeln</b>	Buttons als Toggler setzen, und dann Variable im Code zurücksetzen

**Tasks**

<b>Beginn</b>	2 Bausteine erstellen (ST) Globale Variablen definieren, z.B. $x_d: REAL := 1.0$
<b>Task 1 definieren</b>	Funktionsgenerator
<b>Task 2 definieren</b>	Regler
<b>Takt vorgeben</b>	Ressourcen – Taskkonfiguration - RMT Task anhängen Name: Main (egal) Priorität: 1 (je kleiner desto wichtiger) 0 = System Typ: Zyklisch, T#1s
<b>Task zuweisen</b>	RMT auf Main, Programmaufruf anhängen, "..."
<b>Hardwarekonfiguration</b>	siehe oben "Hardware definieren"