

CODESYS

| | |
|--------------------------------|--|
| Neues Projekt erstellen | Datei/Neu XV-2xx-V2.3.9 Programm + <Sprache> |
| Bei Änderung | Projekt/Alles übersetzen/ |
| Simulation | Online/Simulation Online/Einloggen Online/Start Variablen ändern Daten aktualisiert (Forcen, F7) |
| Hardware definieren | 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) |
| Interface | Online/Ausloggen Visualisierung - RMT auf Visualisierung/Objekt einfügen/<Test> |
| Button | 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 |
| LED | Text ändern: /Text/<Ausgang 1> Farbe ändern: /Farben/Innen (Farbe wenn off) /Farben/Alarmfarbe (Farbe wenn on) Farbwechsel /Variablen/Farbwechsel/PLC_PRG.Ausgang1 |
| Textfeld | Text anzeigen /Text/< t = %s> Variabel zuweisen /Variablen/Textausgabe/< Variable PLC_PRG.TON1.ET> (Elapsed Time) |
| Traceaufzeichnung | Ressourcen – Traceaufzeichnung Extras/Tracekonfiguration /Trigger Position [%]/20 "Unten alle Variablen definieren" Extras/Trace starten Extras/Trace automatisch lesen |
| Lokale Variable | VAR_INPUT var_enable : BOOL; var_mit_startwert_true : BOOL := TRUE; END VAR |
| Global Variablen | Ressourcen – Globale_Variablen (Definition identisch mit lokalen Variablen) In Visualisierung .A |

KOP (Kontaktplan)

| | |
|-------------------|---|
| Bausteine | |
| 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)

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

Tasks

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