

STARTEN SIE EIN NEUES PROJEKT WIE EIN PROFI:

Wie Schweitzer Engineering Laboratories vom ergonomischen Engineering mit zenon profitierte

Nachdem der Startschuss für ein neues Automatisierungsprojekt gefallen ist, müssen viele wichtige Entscheidungen getroffen werden. Oft machen sich Ingenieure sofort an die SCADA-Entwicklung ohne wirkliches Verständnis für die wichtigsten Design- und Entwicklungsaufgaben, die in den kommenden Wochen oder Monaten auf sie zukommen. Das ist eine sehr kritische Phase, in der eine scheinbar harmlose Entscheidung ausschlaggebend dafür sein kann, ob ein Projekt pünktlich abgeschlossen wird und die Budgetvorgaben und Spezifikationen erfüllt sind, oder ob es zu spät fertig wird, schlampig umgesetzt und voller Fehler ist.

Werfen wir einen Blick auf ein kürzliches Beispiel, bei dem COPA-DATA mit Schweitzer Engineering Laboratories (SEL) an einer großen Schaltanlagen-Automatisierungslösung zusammenarbeitete.

DIE HERAUSFORDERUNG

Unsere Hauptaufgabe war die Zusammenarbeit mit den Technikern von SEL in der Designphase ihres SCADA-Projekts. Nach einigen Diskussionen über Kommunikationsprotokolle und Treiber nahmen wir uns die Vorgehensweise bei der Erstellung von Variablen oder TAGs vor. In zenon gibt es verschiedene Wege, um Variablen oder TAGs zu erstellen.

Einerseits können Variablen einzeln manuell angelegt werden. Diese Methode ist jedoch meistens fehleranfällig und kann außerdem sehr zeitaufwändig sein. Darum schlossen wir diese Option gleich zu Beginn aus. Für manche spezifische Treiber bietet zenon die Möglichkeit, die Variablen über Online- oder Offlinemodus zu importieren. Für kleinere Projekte ergibt sich dadurch eine massive Zeitersparnis. Der Online-/Offline-Import von Treibern kann theoretisch alle TAGs für ein bestimmtes Gerät in weniger

als zehn Minuten importieren. Diese Option mussten wir jedoch auch ausschließen, da die Geräte bereits im Feld im Einsatz und außerdem ein paar Kontinente weit entfernt von uns waren. Ein zusätzlicher Nachteil dabei ist, dass dieser Import zu einer flachen, unstrukturierten Variablenliste geführt hätte. Eine weitere Möglichkeit, die wir in Betracht zogen, war ein CSV-Import. zenon bietet nämlich auch eine CSV-Importfunktion, mit der Variablen anhand einer CSV-Definitionsdatei angelegt und modifiziert werden können. Dies ist eine häufige Vorgehensweise und ein praktischer Weg, um umfangreiche Änderungen sehr schnell durchzuführen. Da das Projekt jedoch sehr groß war und mehrere Techniker gleichzeitig daran arbeiten sollten, war absehbar, dass jeder Projektant eine etwas andere Methode bei Export, Modifikation und Import verwenden würde, die möglicherweise zu inkonsistenten Ergebnissen führen würde.



State	Name	Description	IEC data type
	2411.AI	2411 Analog Inputs	
	IA_Mag	Structure element	DINT/<embedded> 1
	IB_Mag	Structure element	DINT/<embedded> 2
	IC_Mag	Structure element	DINT/<embedded> 3
	IN_Mag	Structure element	DINT/<embedded> 4
	Freq	Structure element	DINT/<embedded> 5
	VAB	Structure element	DINT/<embedded> 6
	VAVG	Structure element	DINT/<embedded> 7
	VBC	Structure element	DINT/<embedded> 8
	VCA	Structure element	DINT/<embedded> 9
	VN	Structure element	DINT/<embedded> 10
	Q	Structure element	DINT/<embedded> 11
	P	Structure element	DINT/<embedded> 12
	PF	Structure element	DINT/<embedded> 13

Abbildung 1: Konfiguration strukturierter Datentypen für den SEL-2411 Programmable Automation Controller.

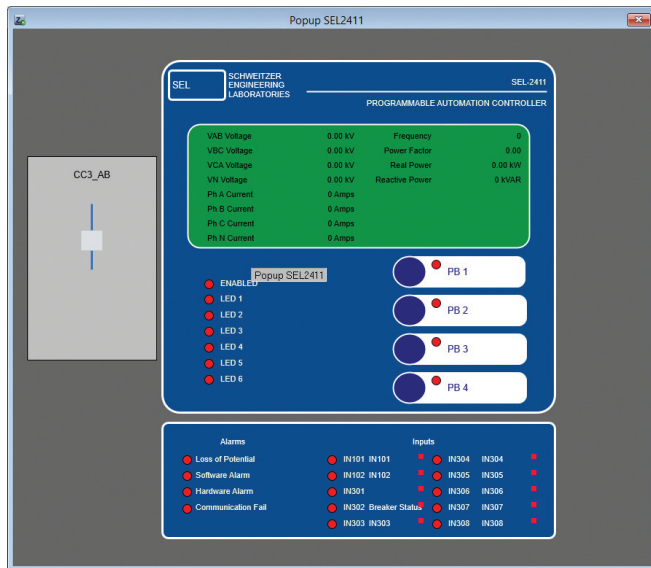


Abbildung 2: Ein SEL-2411 Popup-Bild mit Echtzeitdaten aus dem Gerät.

DIE LÖSUNG

Schließlich fanden wir eine Lösung – zenon Datentypen: eine objektorientierte Methode zum Anlegen von Variablen, die für Konsistenz sorgt und Vererbung unterstützt. Im elektrischen System dieser speziellen Schaltanlagen-Automatisierungsanwendung gab es hunderte Instanzen von etwa einem Dutzend verschiedener SEL-IED-Typen. Durch den Einsatz strukturierter Variablen in zenon konnten wir einen Datentyp für jeden Gerätetyp erstellen. Der Zeitaufwand, den wir am Anfang in die Konfiguration der Datentypen mit passenden Objekten und Einstellungen investierten, ist, in Relation zu der Zeitersparnis und den organisatorischen Vorteilen, die daraus resultieren, gering.

Wir haben z. B. für den SEL-2411 Programmable Automation Controller strukturierte Datentypen konfiguriert, wie aus *Abbildung 1* ersichtlich. Da jede SEL-2411 mit identischen DNP Maps konfiguriert ist, konnten wir sogar so weit gehen, auf der Datentyp-Ebene alle Alarm- und Eventbedingungen, ID-Labels, Steuerungseigenschaften sowie die DNP-Adressen einzustellen.

Wenn die Struktur einmal angelegt ist, kann sie später immer noch erweitert oder verändert werden, auch wenn schon Variablen (Instanzen) darauf basierend erstellt worden sind. Ein Beispiel: wenn ein Alarm zu dem existierenden Struktur-Element 24.11AI.PF auf Datentyp-Ebene hinzugefügt wird, dann erben alle Instanzen dieses Typs diese Modifikation automatisch.

ABER DAS WAR ERST DER ANFANG ...

Nachdem wir die Vorteile der zenon Datentypen genutzt hatten, gingen wir mit dem objektorientierten Konzept noch einen Schritt weiter und beschäftigten uns mit dem Thema Popup-Bilder. Bei diesem System war ein Faceplate

oder ein Popup-Bild für jede Geräteinstanz geplant. Wenn ein Benutzer im Einlinienschaltbild auf die SEL-2411 klickt, wird ein SEL-2411 Popup-Bild angezeigt (siehe *Abbildung 2*), mit Echtzeitdaten von diesem spezifischen Gerät. Das ist an sich nichts Besonderes in einem HMI/SCADA-System, aber unser Ziel war es, diese Aufgabe so einfach, schnell und zuverlässig wie möglich zu erledigen.

Wir begannen mit der Erstellung eines einzelnen Bildes für die SEL-2411 in zenon. Wir stellten einen Rahmen für die Schablone ein, damit sie in der Runtime verschoben werden konnte und konfigurierten die Schablone so, dass sie mehrmals geöffnet werden konnte (z. B. um zwei unterschiedliche SEL-2411-Geräte nebeneinander zu vergleichen). Dann erzeugten wir mithilfe von Symbolen und nativen zenon Bildelementen eine 1:1-Kopie des Geräts. Als Voraussetzung hatten wir auch einen Satz interner „Dummy“-Variablen in zenon angelegt, die auf dem Bild als Platzhalter beim Projektieren agieren sollten.

Wir wussten, dass wir, um die 100 verschiedenen Instanzen der SEL-2411 zu visualisieren und zu repräsentieren, ein einzelnes Bild in Kombination mit der Bildsubstituierung in zenon verwenden könnten, mit der im Prinzip die Variablenverknüpfung hinter den Bildobjekten ersetzt wird. Diese Funktion bietet zenon seit vielen Jahren. In Vorgängerversionen von zenon 7.11 hätte man jedoch für die Repräsentation der 100 verschiedenen Instanzen der SEL-2411 mit Standard-Funktionalität 100 verschiedene Bildumschaltungsfunktionen anlegen müssen, eine für jede Instanz.

MANCHMAL IST ES SO EINFACH

Da wir in diesem Projekt zenon 7.11 verwendeten, konnten wir auch die Vorteile der Bildsubstituierung mit

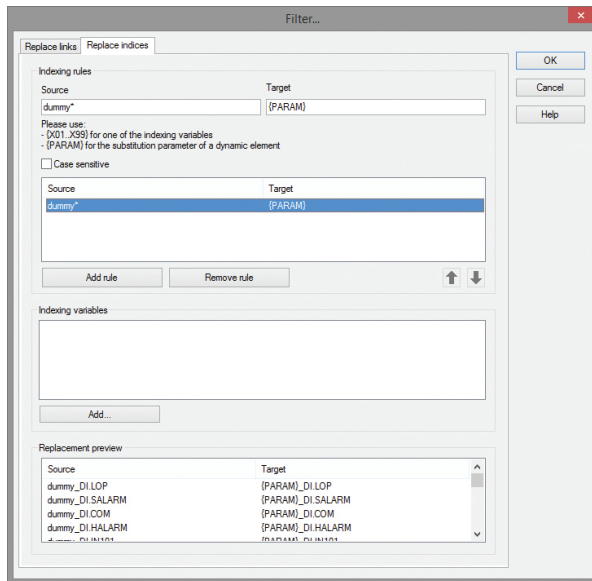


Abbildung 3:
Basierend auf der Ersetzungsregel – Quelle:
dummy* / Ziel: {PARAM} – in der
Bildumschaltungsfunktion und abhängig von der
vom Benutzer angeklickten Schaltfläche werden die
korrekten Variablen in der Runtime angezeigt.

Parametrierung nutzen. Das bedeutete im Endeffekt, dass wir die 100 verschiedenen Instanzen des SEL-2411-Gerätetyps mit nativer Funktionalität visualisieren konnten, indem wir lediglich ein Bild und eine Bildumschaltungsfunktion anlegten!

Aber wie funktioniert das genau? Basierend auf unserer Struktur erstellten wir gemäß einer Namenskonvention eine Instanzvariable namens A1_CC1_AB_2411 und für die Abbildung des zweiten Geräts ein Set namens A2_CC1_AB_2411.

In der Bildumschaltungsfunktion, die das SEL-2411-Bild öffnet, wechselten wir zur Registerkarte „Indizes ersetzen“, wo wir die folgende Ersetzungsregel eingaben (siehe *Abbildung 3*):

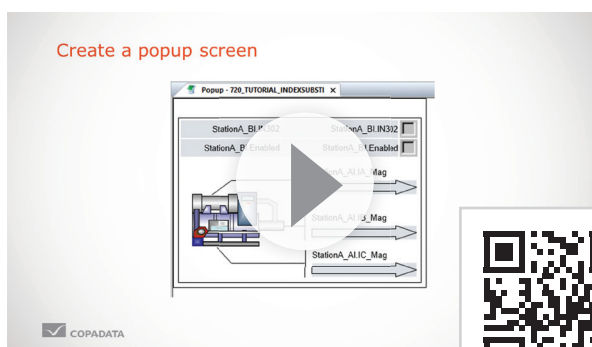
Quelle: dummy*
Ziel: {PARAM}

Der Inhalt von {PARAM} wird in der Runtime eingefügt und dieser Parameter wird aus dem aufrufenden Element ausgelesen. In unserem Fall war das die Schaltfläche, die sich auf dem Einlinienschaltbild befindet. In zenon 7.11 und neueren Versionen ist diese Schaltfläche mit unserer einzelnen SEL-2411-Bildumschaltungsfunktion verknüpft, sie beinhaltet aber auch eine Texteigenschaft für den Parameter, der ersetzt werden soll. Dort gaben wir den eindeutigen Instanznamen ein, in unserem Beispiel entweder A1_CC1_AB_2411 oder A2_CC1_AB_2411. In der Runtime werden dann die korrekten Variablen angezeigt, abhängig von der Schaltfläche, die der Benutzer anklickt.

ERGONOMIE MIT ZENON

Das ist nur ein kleines, aber effektives Beispiel dafür, wie sich smarte Entscheidungen in der frühen Designphase eines HMI/SCADA-Projekts sowie der Einsatz unterstützender Software nachhaltig rentieren.

Video:
Sparen auch Sie Zeit und Geld mit den ergonomischen Engineering-Funktionen von zenon, wie in diesem Fallbeispiel beschrieben.
Scan & Play!



<http://kaywa.me/mZRO1>

LOUIS PAGLAICETTI,
TECHNICAL CONSULTANT

