

Einstieg A3s Programmierung

Einstieg A3s Programmierung

Ergänzend zur Hilfe OPUS Projektor Manual wird mit diesem Einstieg auf nützliche Besonderheiten der Topcon Display Programmierung hingewiesen.



Copyright

Alle Rechte dieser Dokumente, einschließlich Übersetzung, Nachdruck, Vervielfältigung etc., vorbehalten. Jede Weitergabe, Bearbeitung u.ä. – egal in welcher Form – ist ohne schriftliche Genehmigung untersagt. Verarbeitung, Vervielfältigung oder Verbreitung nur mit schriftlicher Genehmigung. Gebrauchs-, Handels- und Warennamen (Warenzeichen) werden ohne besondere Kennzeichnung verwendet. I. d. R. handelt es sich bei den meisten um eingetragene und geschützte Namen bzw. Zeichen, deren Nutzung den gesetzlichen Bestimmungen unterliegt.

Alle angegebenen Daten, Darstellungen u. ä. dienen allein der Produktbeschreibung und sind nicht als zugesicherte Eigenschaften im Rechtssinne aufzufassen. Etwaige Schadensersatzansprüche - gleich aus welchem Rechtsgrund - sind ausgeschlossen, soweit weder Vorsatz noch grobe Fahrlässigkeit vorliegt.

Es kann keine Gewähr übernommen werden, dass die angegebenen Schaltungen oder Verfahren (auch teilweise) frei von Schutzrechten Dritter sind.

Änderungen und Irrtümer vorbehalten.

Druckdatum: 17. 08. 2020

Inhaltsverzeichnis

1 Einleitung 4

2 Hardware 5

3 Software 7

3.1 Prozessdatenspeicherung 7

3.2 Schriftarten definieren 9

3.3 CAN ID 9

3.4 Analogen Wert einlesen 10

3.5 User Parameter schreiben und lesen 10

3.5.1 Schreiben 10

3.5.2 Lesen 10

3.6 Zyklisches Ausführen eines Scripts 11

3.7 Mehrsprachigkeit 12

1 Einleitung

Zur Visualisierung von Prozessdaten einer Mobilsteuerung und Vorgaben von Werten empfiehlt HAWE Hydraulik SE die Anbindung durch CAN-Bus. Dazu haben sich in der Vergangenheit die Displays der Firma Topcon Electronics GmbH & Co. KG, Opus Serie bewährt. In der folgenden Beschreibung und im Vorlagenprojekt wird dazu ein Opus A3s beispielhaft hergenommen. Diese Anleitung liefert auf kompakte Weise nützliche Zusatzinformationen, welche in fast jedem Topcon Projekt verwendet werden können. Eine intensive Software Schulung zur Programmiersoftware Opus Projektor ist zwingend notwendig und teilweise Voraussetzung, um diese Anleitung zu verstehen.

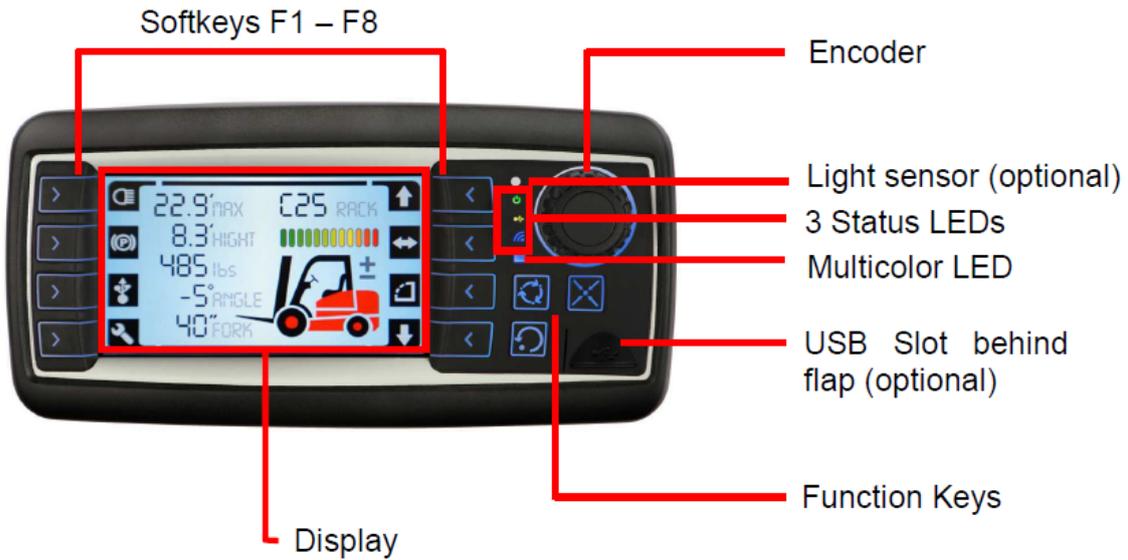
Die Programmiersoftware Opus Projektor von Topcon ist lizenzpflichtig.

In der Demoversion können maximal zwei Seiten und 5 Variablen angelegt werden. Bestehende Projekte können angeschaut, jedoch nicht neu kompiliert werden.

2 Hardware

Hardware

OPUS A3 STANDARD



Schematic diagram of the OPUS A3 STANDARD Landscape

Abbildung 2.1: A3s Display

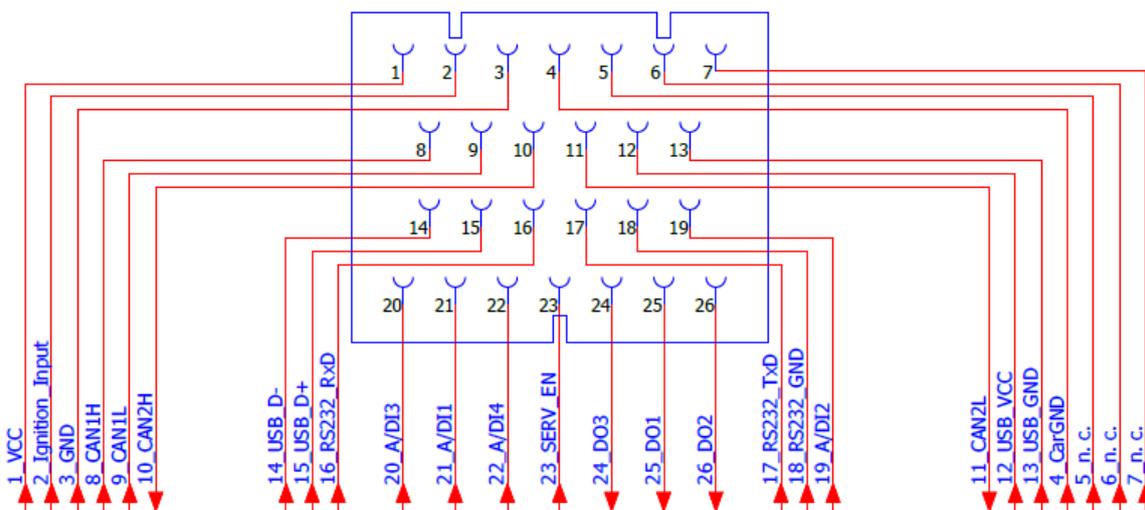


Abbildung 2.2: A3s Display Belegung

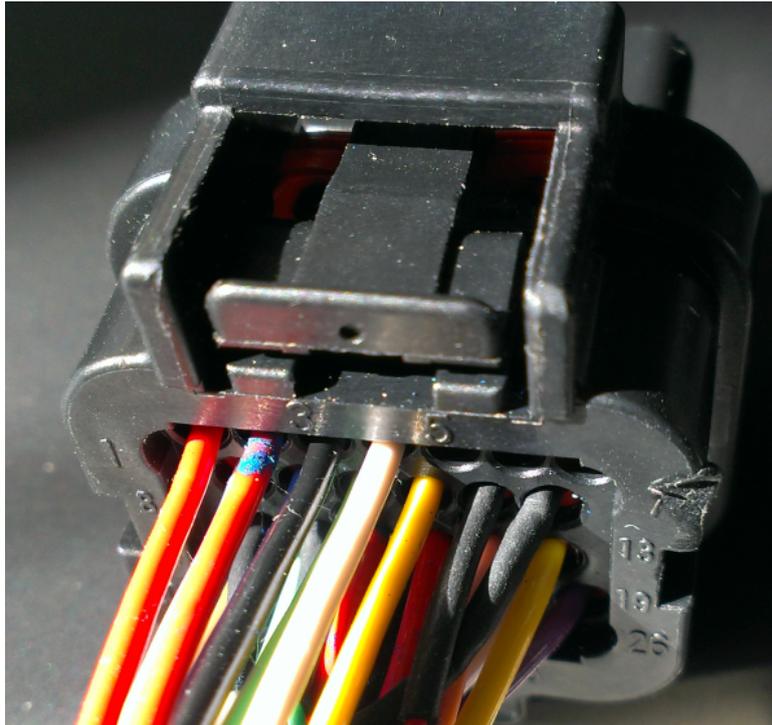


Abbildung 2.3: A3s Display Stecker

3 Software

Die jeweils aktuelle Programmiersoftware findet sich auf der Internetseite von topcon-electronics im Bereich Service – Kundenbereich – Software download.

Für viele Thematiken existieren youtube Videos unter dem Suchbegriff „Projektor tool“. Zudem ist die Suchfunktion der Hilfe mit vielen Erklärungen und zusätzlichen Verweisen auf Videos ein sehr gutes Nachschlagewerk.

Für das bestehende Vorlagenprojekt werden einige Thematiken erklärt.

3.1 Prozessdatenspeicherung

Da jedes Display einen Linux Rechner impliziert besteht die Möglichkeit, zyklisch oder ereignisbezogen, Prozessdaten auf der Festplatte zu speichern. Vorgehen: Projekt am Projektnamen aktivieren – rechte Maustaste – Properties – Variable Logging. Dort die zu schreibenden Variablen auswählen.

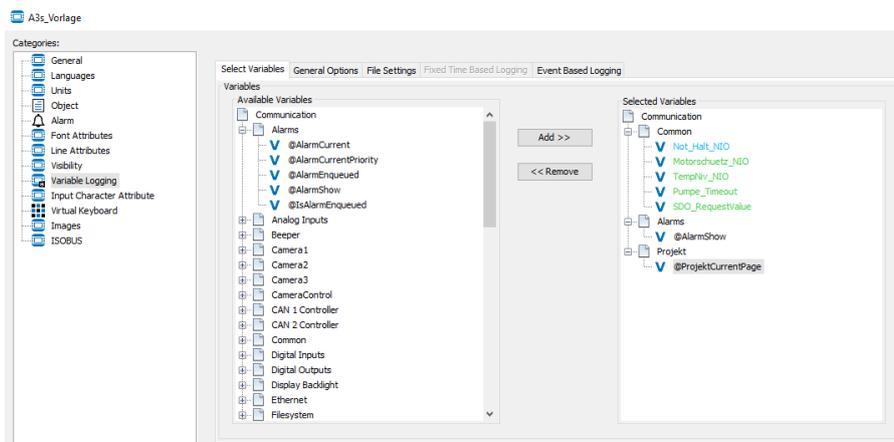


Abbildung 3.1: Variablen speichern

Im Reiter „General Options“ zyklische oder ereignisbasierende Aufzeichnung einstellen. Bei „File Settings“ Aufzeichnungsart und maximale Dateigröße definieren.

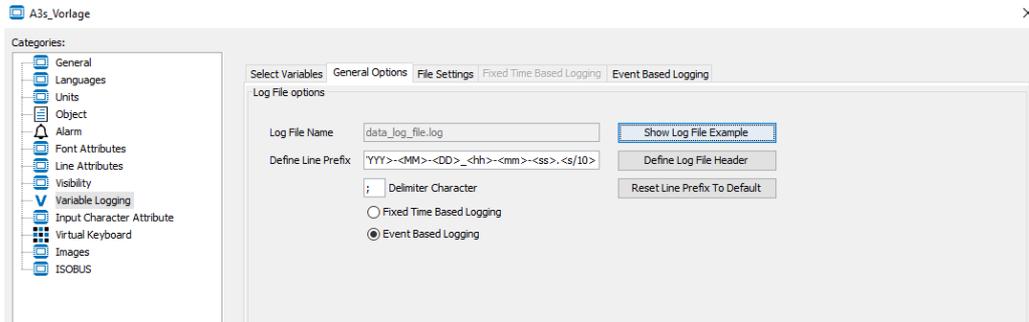


Abbildung 3.2: Variablen speichern Optionen

Im Reiter „Event Based Loggings“ können zu den zuvor definierten Variablen die Kriterien, wann aufgezeichnet wird, festgelegt werden.

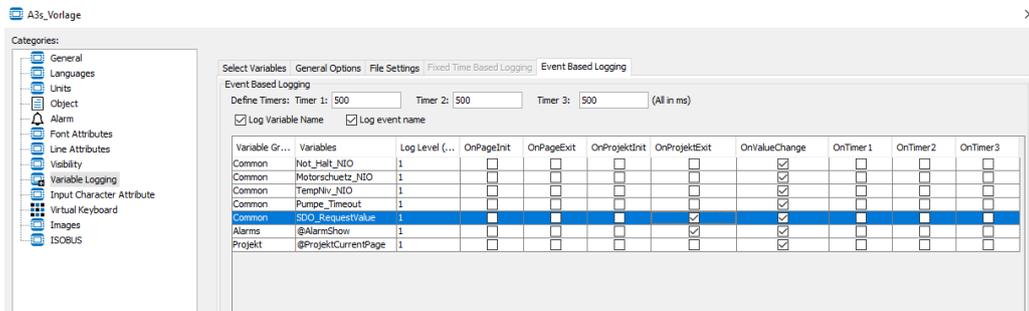


Abbildung 3.3: Variablen speichern ereignisbezogen

Mit dem Event einer Taste kann dann ein Skript aufgerufen werden, welches bei eingestecktem USB Speichermedium eine Kopie der log-Datei auf das Speichermedium kopiert. Die Variable `usb_verbunden` prüft zunächst, ob ein Speichermedium mit dem Display verbunden ist.

Beispiel:

Beispielhafter Quellcode Logdatei kopieren:



```

1 |
3 | var usb_verbunden = getVariableValue("@USBMemStatus");
5 | if (usb_verbunden === 1) {
7 |     var usb_path = getVariableValue("@USBMountPath");
   |     setVariableValue("@LogCopyPath", usb_path);
   |     setVariableValue("@LogCopyAllNow", 1);
   | }

```

Zum Logdatei verschieben anstelle kopieren müsste Zahl 2 bei `@LogCopyAllNow` stehen.

3.2 Schriftarten definieren

Zur Verwendung farbiger Schriften (z.B. für Fehlermeldungen oder Warnungen) sind diese vorher für das gesamte Projekt zu definieren. Schriften gehören zu den sogenannten „Projekt Properties“ (siehe auch Hilfe, Suchbegriff „font“)

Projekt am Projektnamen aktivieren – rechte Maustaste – Properties

Font Attributes – im Feld „Enter name“ einen Namen des neuen Schriftsatzes eingeben. Diese Schriftart kann nachfolgend in den jeweiligen Eigenschaften der Grafikelemente ausgewählt werden.

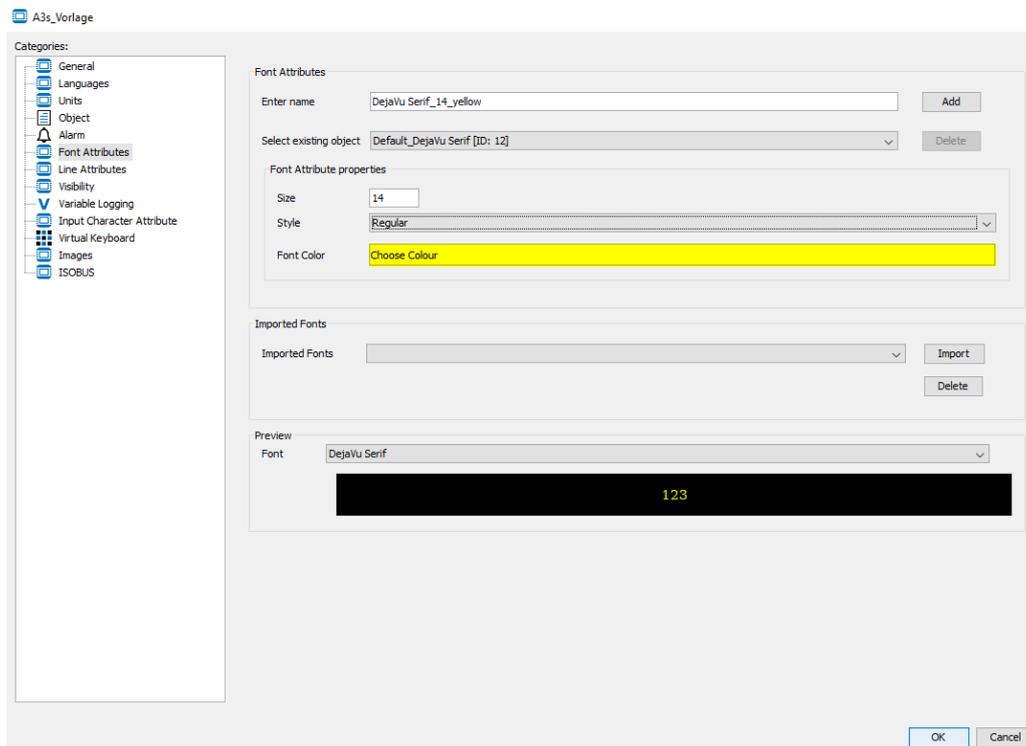


Abbildung 3.4: Schriftart definieren

3.3 CAN ID

Zum Datenaustausch über CAN muss zunächst die Baudrate eingestellt werden. Dazu unter Communication – Port Configuration – Properties – Bus Speed die Übertragungsrate einstellen.

CAN-ID definieren:

Communication – Protocols – CAN Freestyle – Configure Mappings. Bei Neudefinition Type: Receive oder Transmit auswählen. Dies erscheint dann unten im Feld Type. Unter CAN ID die entsprechende ID eintragen. Im nächsten Schritt wird die genaue Variablenzuordnung im Reiter „Configure Mapping“ definiert.

Achtung: Gibt es eine Abweichung z.B. in der definierten Bytelänge in „Configure Mapping“ zur realen CAN ID funktioniert das Einlesen nicht.

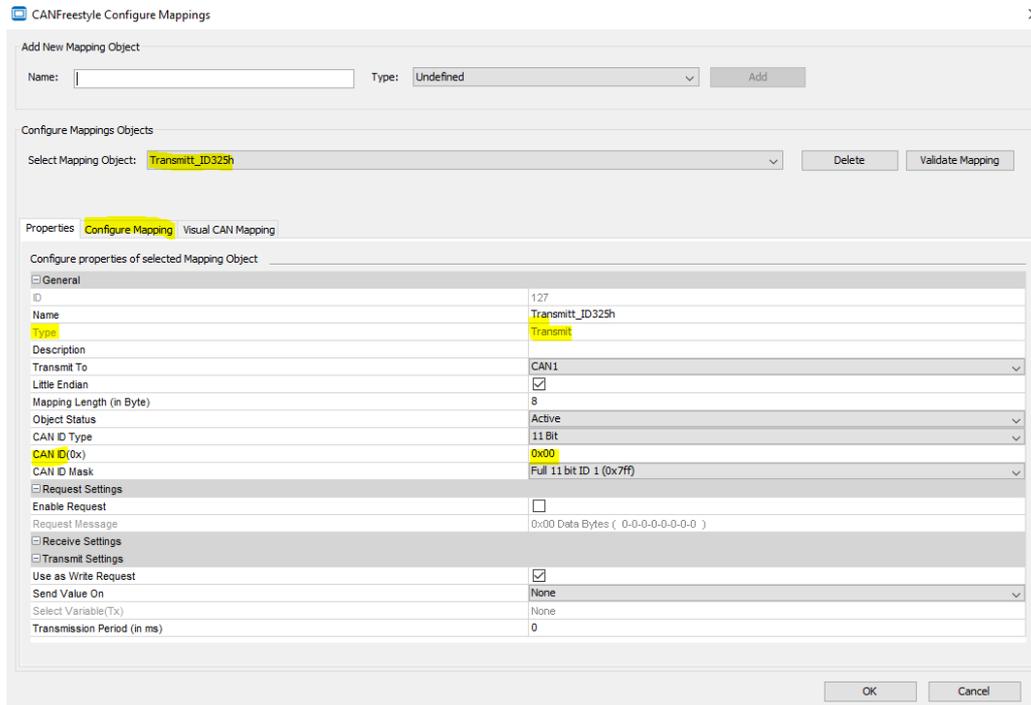


Abbildung 3.5: CAN ID definieren

3.4 Analogen Wert einlesen

Die einfachste Variante ist, wenn der analoge Wert von der Steuerung auf eine CAN Nachricht geschrieben wird. In diesem Fall muss lediglich eine neue Nachricht vom Typ „Recieve“ definiert und eine Variable in „Configure Mapping“ zugeordnet werden.

3.5 User Parameter schreiben und lesen

3.5.1 Schreiben

Beispiel CAN IO:

Voraussetzung ist die Bekanntheit der Node-ID des CAN-IO. Gesendet wird über SDO-Protokoll auf Index 2002h. Im Beispiel ist Node-ID des CAN IO = 1. Daraus folgt: Senden auf 601h. Mit nachfolgendem Befehl wird User Parameter Nr. 25 beschrieben.

`sendCANMessage(1, 0x601, 8, 0x2B, 0x02, 0x20, 26, ValLow, ValHigh, 0, 0);`

Wichtig: User Parameter beginnen mit 0. Deshalb muss in der CAN Message um 1 erhöht werden.

3.5.2 Lesen

Beispiel SDO Input auf ID 581h

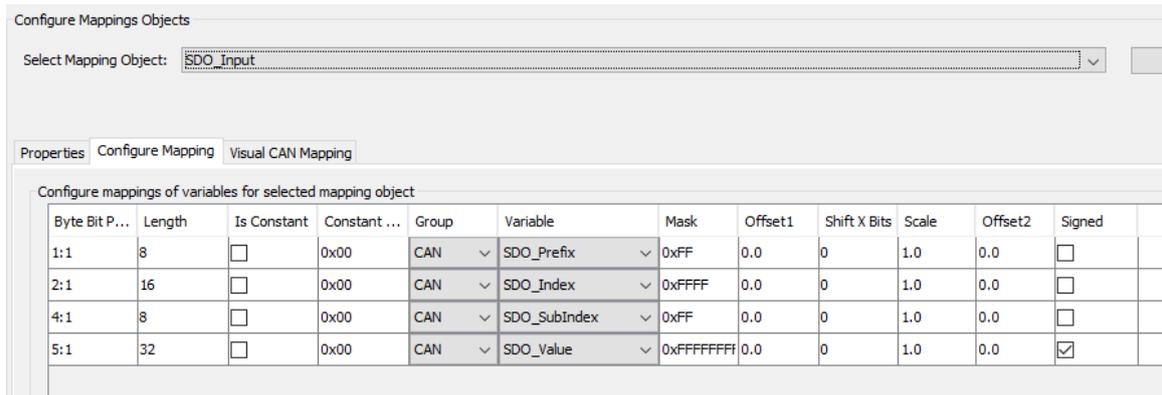


Abbildung 3.6: CAN ID lesen Konfiguration 1

In diesem Beispiel wird sowohl bei der Variablen „SDO SubIndex“ als auch „SDO_Value“ das Einlese-Script mit dem Event „OnChangeByOwner“ zum Einlesen neuer Werte ausgeführt. Je nach Anwendungsfall kann es unvorteilhaft sein, wenn ein Script mehrfach innerhalb kurzer Zeit ausgeführt wird. Problematisch ist, dass innerhalb einer CAN Nachricht ein minimaler Zeitunterschied zwischen dem Eintreffen der Nachrichten der einzelnen Bytes existiert.

Alternativ:

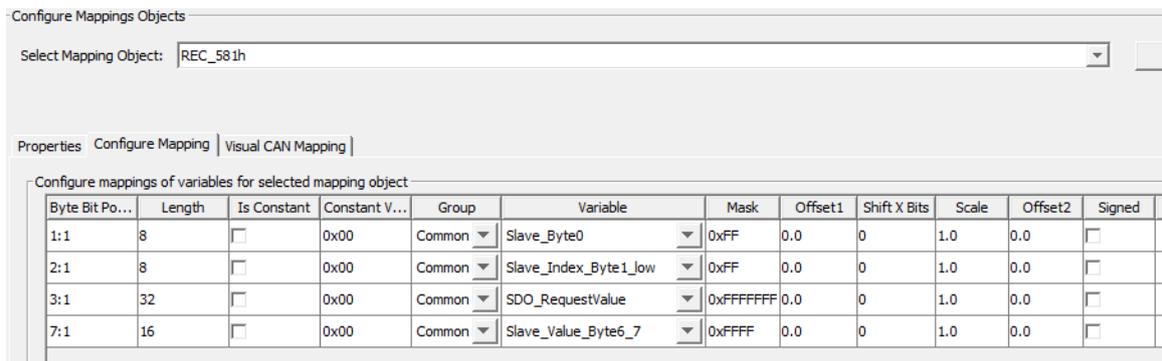


Abbildung 3.7: CAN ID lesen Konfiguration 2

Ausführung des Scripts mit dem Event „OnChangeByOwner“. Soll mit jedem Telegramm das Script ausgeführt werden, ist bei Properties der Variable „Slave_Value_Byte6_7“ die Option „Force Writing“ zu aktivieren.

3.6 Zyklisches Ausführen eines Scripts

Das zyklische Ausführen eines Scripts kann auf drei Arten geschehen.

- Variante 1 – Projekteigenschaften
Das Projekt markieren – rechte Maustaste – Events – OnProjektRepeat. Hier kann ein Script in einem festen Zeitraster ausgeführt werden. Das Zeitraster ist in der Einheit 100ms im Feld

Periode verstellbar. Im Beispiel ist 500ms ausgewählt. Es fehlt noch die Auswahl einer Aktion und im Falle von "Execute Script" die Auswahl des entsprechenden Scripts.

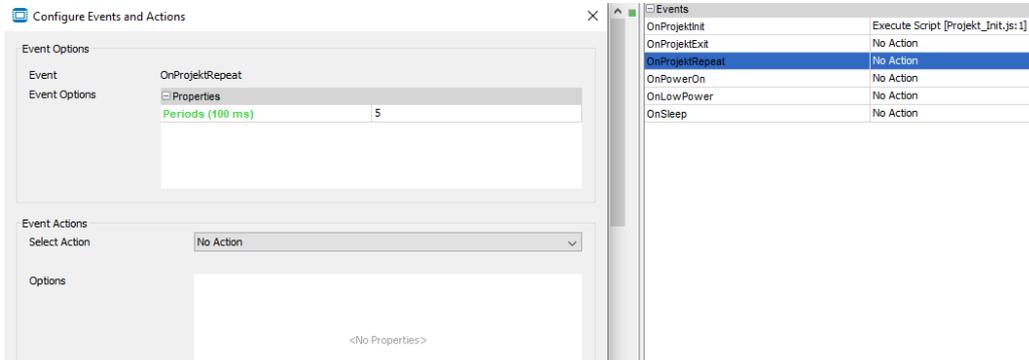


Abbildung 3.8: Variante 1 - Zyklisches ausführen eines Code

- Variante 2 – Page
Gleiches Prozedere wie bei Variante Projekt. Die periodische Wiederholung geschieht, solange die Seite aufgerufen ist.
- Variante 3 – Systemvariable
Mit RTC (real time clock) Variablen (z.B. @RTC_Seconds, @RTC_Minutes) Events – „OnChangeByOwner“ kann ein Script ausgeführt werden. Im Gegensatz zu Variante 1 und 2 funktioniert Variante 3 nicht im Simulationsmodus, weil das OnValueChange-Ereignis für @RTC von der Hardware des Displays ausgelöst wird.

3.7 Mehrsprachigkeit

Projektor tool stellt ein System für Mehrsprachigkeit zur Verfügung. Die einzelnen Sprachen können je Projekt unterschiedlich gewählt werden. Zu beachten ist, dass die Änderung der Sprache immer auch Auswirkungen auf die Darstellung und damit auf die gewählte Schriftart hat. Jede Sprache hat einen Index, über den die Sprache zur Laufzeit verändert werden kann.

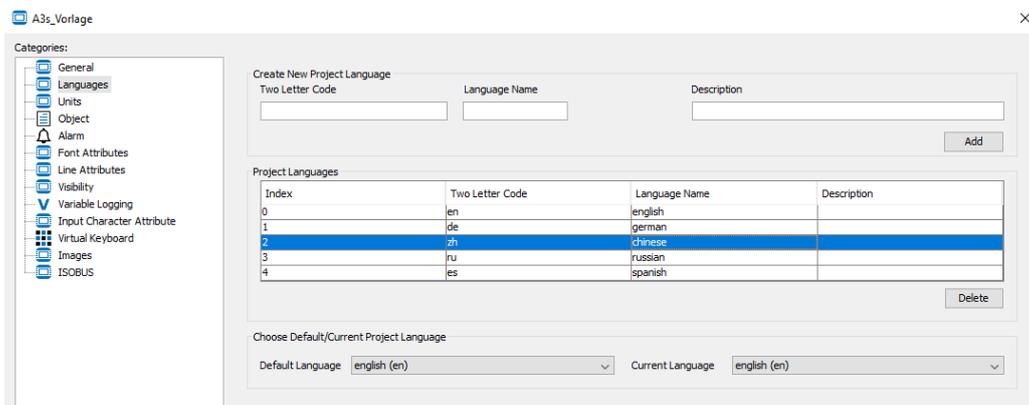


Abbildung 3.9: Definition Index der Sprache

Hat man sich für Mehrsprachigkeit entschieden, so ist bei jedem String Element unter der Eigenschaft „Dependency“ der Wert „Language“ auszuwählen. Dies hat zur Folge, dass bei den Eigenschaften eines String Field unter „Preview Value“ nicht mehr direkt der Text eingegeben wird, sondern über das Symbol rechts mit den drei Punkten ein Feld zur Eingabe der in den Projekteigenschaften vordefinierten Sprache erscheint.

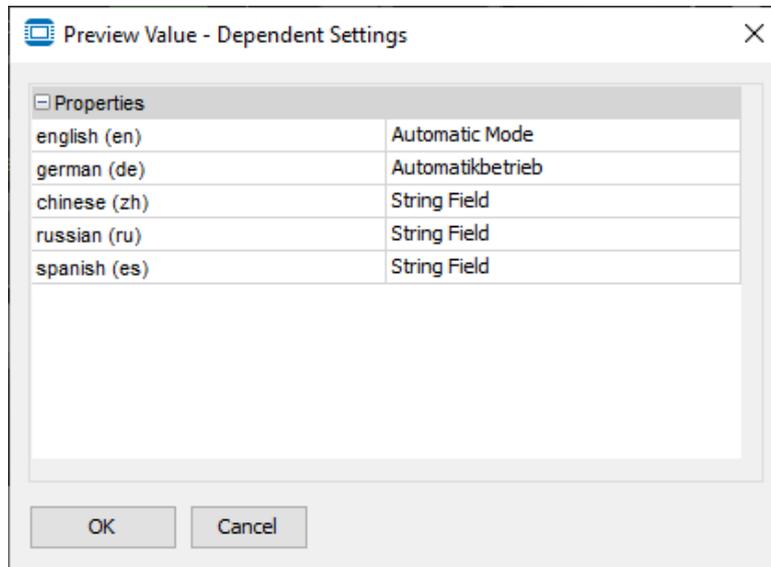


Abbildung 3.10: Eintrag der Sprache

Zur vereinfachten Übersetzung kann eine .csv-Datei exportiert, die übersetzte Sprache eingetragen und wieder importiert werden. Der Menüpunkt dazu ist unter Tools – Import Export Language CSV zu finden. Die Referenzvariable zum Umstellen der Sprache lautet „@SettingsLanguageEnum“. Beispielsweise kann über ein „Numeric Field“ und „Set As Input“ zur Laufzeit die Sprache umgestellt werden. Je nach Festlegung in den Eigenschaften des Projektes unter der Rubrik Language bestimmt der Indexwert die Sprache.

Haftungsausschluss

Informationen in diesem Dokument werden ausschließlich in Verbindung mit HAWE Produkten bereitgestellt. HAWE Hydraulik SE und ihre Tochtergesellschaften ("HAWE") behalten sich das Recht vor, Änderungen, Korrekturen, Modifikationen oder Verbesserungen auf dieses Dokument und die hier beschriebenen Produkte und Dienstleistungen jederzeit ohne vorherige Ankündigung vorzunehmen.

Alle HAWE Produkte werden gemäß den Verkaufsbedingungen von HAWE verkauft.

Die Käufer sind allein verantwortlich für die Auswahl, Abgrenzung und Verwendung der hier beschriebenen HAWE Produkte und Dienstleistungen und HAWE übernimmt keine Haftung für die Auswahl, Abgrenzung oder Verwendung der hier auch im Zusammenhang beschriebenen HAWE Produkte und Dienstleistungen.

Unter diesem Dokument wird keine Lizenz gewährt, weder ausdrücklich noch stillschweigend, durch Streichung oder anderweitig, noch zu irgendwelchen Rechten an geistigem Eigentum. Wenn sich ein Teil dieses Dokuments auf Produkte oder Dienstleistungen Dritter bezieht, gilt dies nicht als Lizenzgewährung von HAWE für die Nutzung solcher Produkte oder Dienstleistungen von Drittanbietern oder eines darin enthaltenen geistigen Eigentums oder als Garantie für die Verwendung in jeglicher Art von Produkten oder Dienstleistungen Dritter oder geistigem Eigentum, die darin enthalten sind.

WENN NICHT ANDERS IN DEN VERKAUFS- UND VERKAUFSBEDINGUNGEN VON HAWE ANGEZEIGT, LEHNT HAWE JEGLICHE AUSDRÜCKLICHE ODER STILLSCHWEIGENDE GEWÄHRLEISTUNG HINSICHTLICH DER VERWENDUNG UND / ODER DES VERKAUFS VON HAWE PRODUKTEN EINSCHLIESSLICH DER GARANTIE DER MARKTGÄNGIGKEIT, DER EIGNUNG FÜR EINEN BESTIMMTEN ZWECK (UND DEREN GLEICHWERTIGEN GESETZGEBUNGEN JEGLICHER GERICHTSBARKEIT) ODER VERLETZUNG VON PATENT-, URHEBERRECHTS- ODER ANDEREM GEISTIGEN EIGENTUMSRECHTS AB.

SOFERN NICHT AUSDRÜCKLICH VON EINEM BEVOLLMÄCHTIGTEN VERTRETER VON HAWE SCHRIFTLICH GENEHMIGT, SIND HAWE PRODUKTE NICHT FÜR DIE VERWENDUNG IN MILITÄR-, LUFTFAHRT-, RAUMFAHRT-, RETTUNGS-, ODER LEBENSERHALTUNGSSYSTEMEN ENTWORFEN, BERECHTIGT ODER GEWÄHRLEISTET, NOCH IN PRODUKTEN ODER SYSTEMEN, WO EIN FEHLER ODER EINE FEHLFUNKTION PERSÖNLICHE VERLETZUNGEN, TOD, SCHWERE EIGENTUMS- ODER UMWELTSCHÄDEN VERURSACHEN KANN.

Die Informationen in diesem Dokument ersetzen alle zuvor bereitgestellten Informationen und ersetzen diese.

HAWE Hydraulik SE

Einsteinring 17 | 85609 Aschheim/München | Postfach 11 55 | 85605 Aschheim/München | Germany
Tel +49 89 379100-1000 | Fax +49 89 379100-91000 | info@hawe.de | www.hawe.com