

Kickstart-Benutzerhandbuch

Stand: Kickstart V1.9.40

Doc Rev 1.0.12 – 31.01.2014

Copyright 2013 ebm-papst

Alle Rechte vorbehalten. Kein Teil des Werkes darf in irgendeiner Form oder mit irgendwelchen Mitteln ohne die schriftliche Genehmigung des Verlags reproduziert werden.

Marken

Produkte, auf die in diesem Dokument verwiesen wird, können entweder Marken und/oder eingetragene Marken ihrer jeweiligen Eigentümer sein. Der Verlag und der Autor erheben keinen Anspruch auf diese Marken.

Microsoft und Windows sind entweder eingetragene Marken oder Marken der Microsoft Corporation in den Vereinigten Staaten und/oder anderen Ländern.

Haftungsausschluss

Obwohl bei der Erstellung dieses Dokuments größte Sorgfalt angewendet wurde, übernehmen der Verleger und der Autor keine Verantwortung für Fehler oder Auslassungen oder für Schäden, die aus der Nutzung der in diesem Dokument enthaltenen Informationen bzw. durch die Nutzung von begleitenden Programmen und Quellcode entstehen. In keinem Fall ist der Verlag und der Autor haftbar für entgangenen Gewinn oder sonstige kommerzielle Schäden, die angeblich oder tatsächlich direkt oder indirekt durch dieses Dokument verursacht wurden.

Änderungen vorbehalten.

Die jeweils aktuelle Version dieses Betriebshandbuchs finden Sie auf der Internetseite von ebm-papst: www.ebmpapst.com

Inhaltsverzeichnis

1	Über dieses Handbuch	4			
1.1	Verwendete Symbole	4			
1.2	Verwendete Begriffe	4			
2	Sicherheit	6			
2.1	Bestimmungsgemäßer Gebrauch	6			
2.2	Sicherheitshinweise	6			
3	Kickstart-Übersicht	7			
4	Installation	9			
5	Kickstart bedienen	10			
5.1	Erste Schritte	10			
5.2	Hauptbildschirm und Symbolleiste	11			
5.3	Benutzerebene	12			
5.4	Projekte und Vorlagen	13			
5.5	Bearbeiten des Modulbaums	14			
6	Kickstart Module	15			
6.1	CAN Interface	15			
6.1.1	CAN Monitor	15			
6.1.2	CAN Sender	16			
6.2	Drive01 / CANopen Drive	17			
6.2.1	Setup	18			
6.2.2	Object Editor	18			
6.2.3	Script	19			
6.2.4	Firmware	19			
6.3	Motor Access / CANopen Node	20			
6.3.1	CANopen Node – Edit Mode	21			
6.3.2	CANopen Node - UI Format	21			
6.4	Network Manager	22			
6.5	Scope	23			
6.5.1	Scope – Erste Schritte	23			
6.5.2	Konfiguration	24			
6.5.3	Aufnahme von Daten	25			
6.5.4	Plot	25			
6.5.5	Trigger	26			
			6.6	UI-Panel	27
			6.6.1	UI-Panel – Erste Schritte	27
			6.6.2	UI Format-Definition	28
			6.6.3	Knoten-IDs und Panels mit mehreren Knoten	31
			6.6.4	Projekt and UI-Dateien / Erweiterte Benutzeroberflächen	31

1 Über dieses Handbuch

1.1 Verwendete Symbole

Die folgenden Symbole werden in diesem Dokument verwendet:



Wie in dieser Bedienungsanleitung und den Gefahrenhinweisen auf den Produkten erwähnt, können Tod, schwere Körperverletzung oder erhebliche Sachschäden die Folge sein, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Wichtige Informationen über das Produkt oder Teile der Bedienungsanleitung.

1.2 Verwendete Begriffe

Fachpersonal

Personal, das mit der Installation, Montage, Inbetriebnahme und dem Betrieb des Produkts vertraut und für die Durchführung der jeweiligen Aktivitäten qualifiziert ist.

CAN (CAN-Bus)

Controller Area Network. Feldbus-Design nach ISO 11898-1.

CANopen

Auf CAN basierendes höherschichtiges Protokoll für den Einsatz in bewegungsorientierten Maschinensteuerungs-Netzwerken, unterhalten von www.can-cia.org.

CAN-Schnittstelle

Zubehör für die Verbindung zwischen PC und CAN-Bus-Netzwerk. Normalerweise ist dies eine CAN-USB-Schnittstelle, aber es können auch andere Optionen verfügbar sein (z. B. CAN-NET über Ethernet-TCP).

Knoten

CANopen-Gerät, identifiziert durch eine **Knoten-ID** im Bereich von 1 bis 127.

CANopen-Antrieb

EC-Positionierantrieb, der mit Kickstart konfiguriert und betrieben werden kann, z.B. ECI-Serie von ebm-papst.

Python

Eine Programmier- und Skriptsprache unter der vom OSI bestätigten Open-Source-Lizenz, verwaltet unter www.python.org. Kickstart verwendet derzeit die [Python-Implementierung 2.6.7](#).

Datenobjekt

Ein Geräteparameter oder Wert, der mithilfe von Kickstart ausgelesen und/oder modifiziert werden kann. Bei CANopen-Geräten wird ein Datenobjekt durch die **Knoten-ID** sowie den **Objektindex** und **-subindex** referenziert. Kickstart verwendet das folgende Standardformat für eine Datenobjekt-ID::

127.1000h.00h (hier: Knoten-ID = 127, Objektindex 1000h und Subindex 0h)

Die Werte von CANopen-Objektdateien werden mithilfe von **SDO**- oder **PDO**-Transfers erhalten.

1 Über dieses Handbuch

SDO – Service Data Object

CANopen SDO-Transfers werden für Lese- und Schreibvorgänge von Geräteparametern und Objekten mit niedriger Priorität verwendet. Die Spezifikationen **SDO-Upload** (vom Gerät lesen) und **SDO-Download** (auf Gerät schreiben) definieren verschiedene Standardmethoden für das Lesen und Schreiben verschiedener Arten von Daten, einschließlich optimierter Transfers für große Datenblöcke.

PDO – Process Data Object

CANopen-Geräte produzieren oder verbrauchen PDO-CAN-Frames. Sie werden für die schnelle Übertragung von kritischen Prozessdaten verwendet. Bevor Sie den PDO-Transfer verwenden können, muss das CANopen-Gerät konfiguriert werden, d. h. der Inhalt des PDO-Objekts muss definiert werden (PDO-Mapping). Es sind zwei Arten von PDOs für CANopen definiert:

Sende-PDO (TPDO) – Vom Gerät produzierte Prozessdaten, z. B. Status- und Positionsinformationen.

Empfangs-PDO (RPDO) – Vom Gerät konsumierte Prozessdaten, z. B. Steuerbefehle.

XDD/XDC

XML-Gerätebeschreibungsdatei (.**xdd**)/XML-Gerätekonfigurationsdatei (.**xdc**).

XDD-Dateien enthalten ein Wörterbuch der Gerätefunktionen und -parameter und ersetzen die klassischen .eds-Dateien (Electronic DataSheet).

XDC-Dateien enthalten zusätzlich Parameterwerte für die Gerätekonfiguration und ersetzen das .dcf-Dateiformat (Device Configuration File).

Qt Quick/QML

[Qt Quick](#) ist eine neue Benutzeroberflächen-Technologie (UI-Technologie) und ein Entwicklungssystem zum Erstellen von attraktiven Benutzeroberflächen innerhalb kürzester Zeit.

QML ist eine deklarative Benutzeroberflächensprache, die auf Javascript basiert. Sie beschreibt, wie Elemente der Benutzeroberfläche aussehen und wie sie sich verhalten. Kickstart-Projekte können von der [Qt Quick](#)-Technologie Gebrauch machen, um maßgeschneiderte Bildschirme und attraktive Endnutzeranwendungen zu erstellen.

RS485-Schnittstelle

Die RS485-Schnittstelle stellt ein leitungsgebundenes Bussystem für serielle Datenübertragungen dar. Die Übertragung geschieht grundsätzlich über 2 Adern. Über diese Adern wird das digitale Signal in differentieller Form übertragen. Auf diesem Weg werden Gleichtaktstörungen effektiv unterdrückt.

RS485-Kommunikation

Die Kommunikation zwischen Benutzer und Antriebssoftware findet über sogenannte Telegramme statt. Jedes Telegramm beinhaltet spezifizierte Daten, die empfangen oder gesendet werden müssen. Die Antriebssoftware ignoriert Telegramme, die nicht an sie adressiert sind.

2 Sicherheit

2.1 Bestimmungsgemäßer Gebrauch

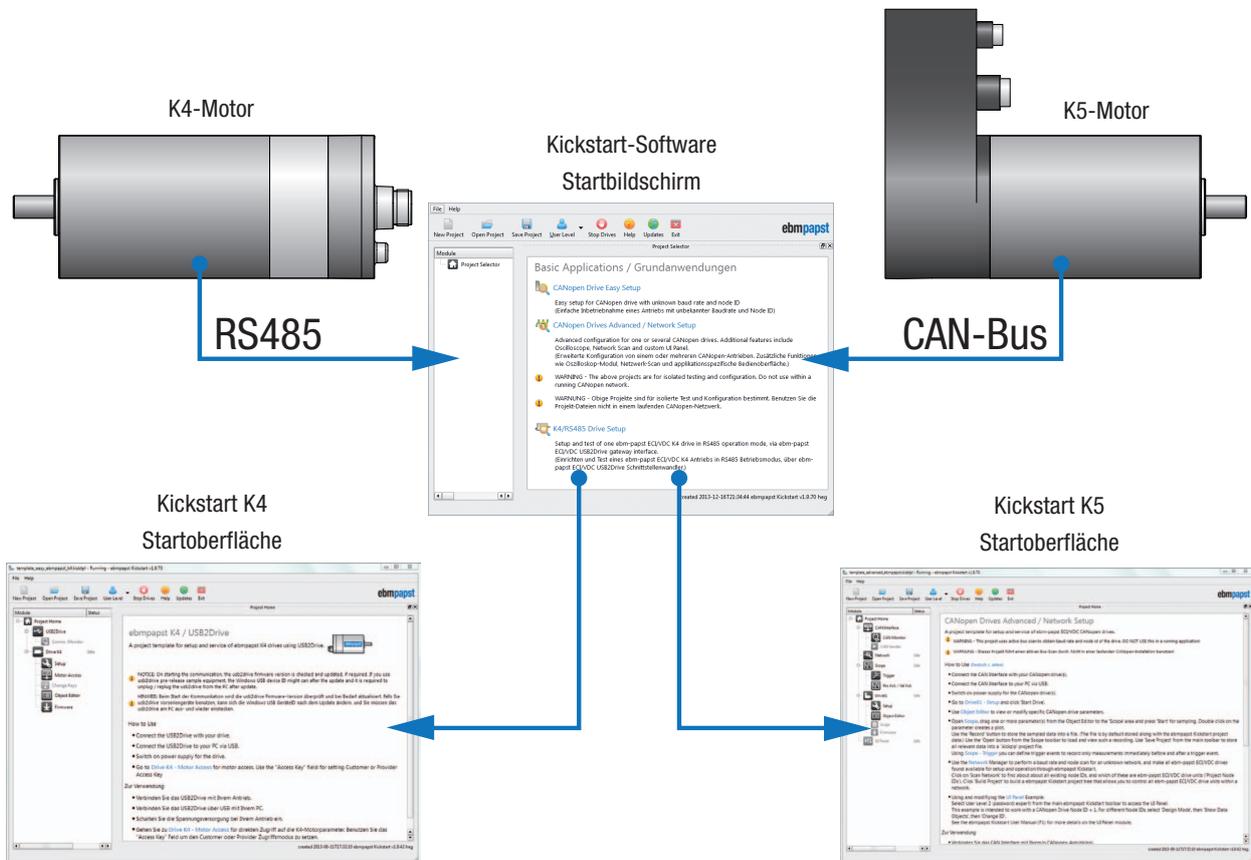
Diese Software ist für die Inbetriebnahme und den Service von Positionierantrieben und anderen bewegungsorientierten Maschinensteuerungen bestimmt. Diese Software greift über ein hochschichtiges Anwendungsprotokoll, wie die CANopen application oder das RS485-Protokoll, auf die Geräte zu.

2.2 Sicherheitshinweise



Beziehen Sie sich für alle Inbetriebnahme- und Serviceaufgaben auf die technische Dokumentation Ihres Antriebs oder Automatisierungsgeräts, der oder das von dieser Software gesteuert wird. Alle Arbeiten müssen von Fachpersonal durchgeführt werden, das mit den Antriebseinheiten oder Geräten vertraut ist.

3 Kickstart-Übersicht



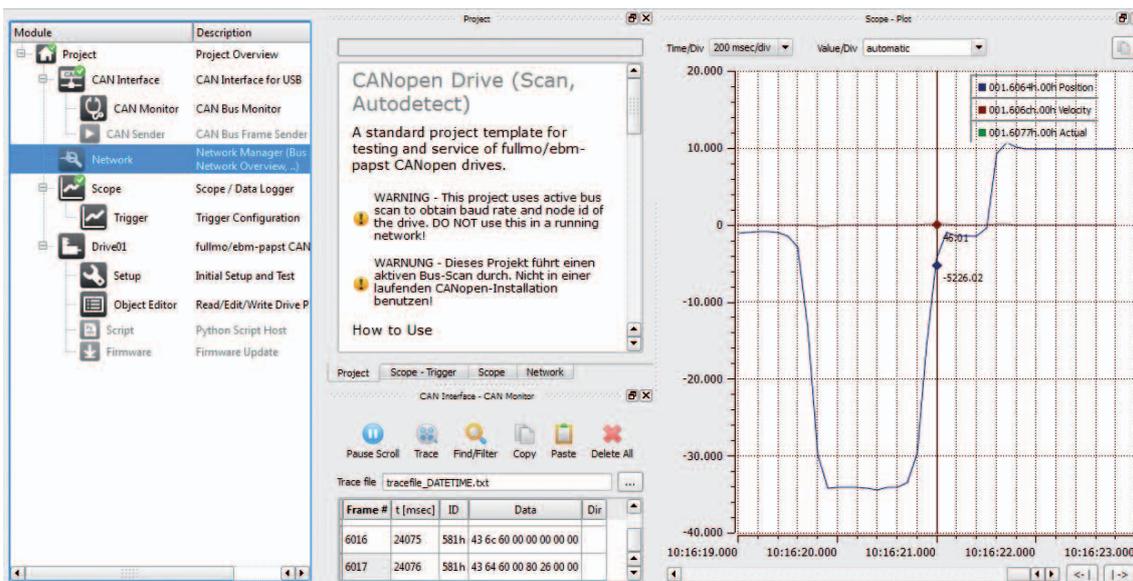
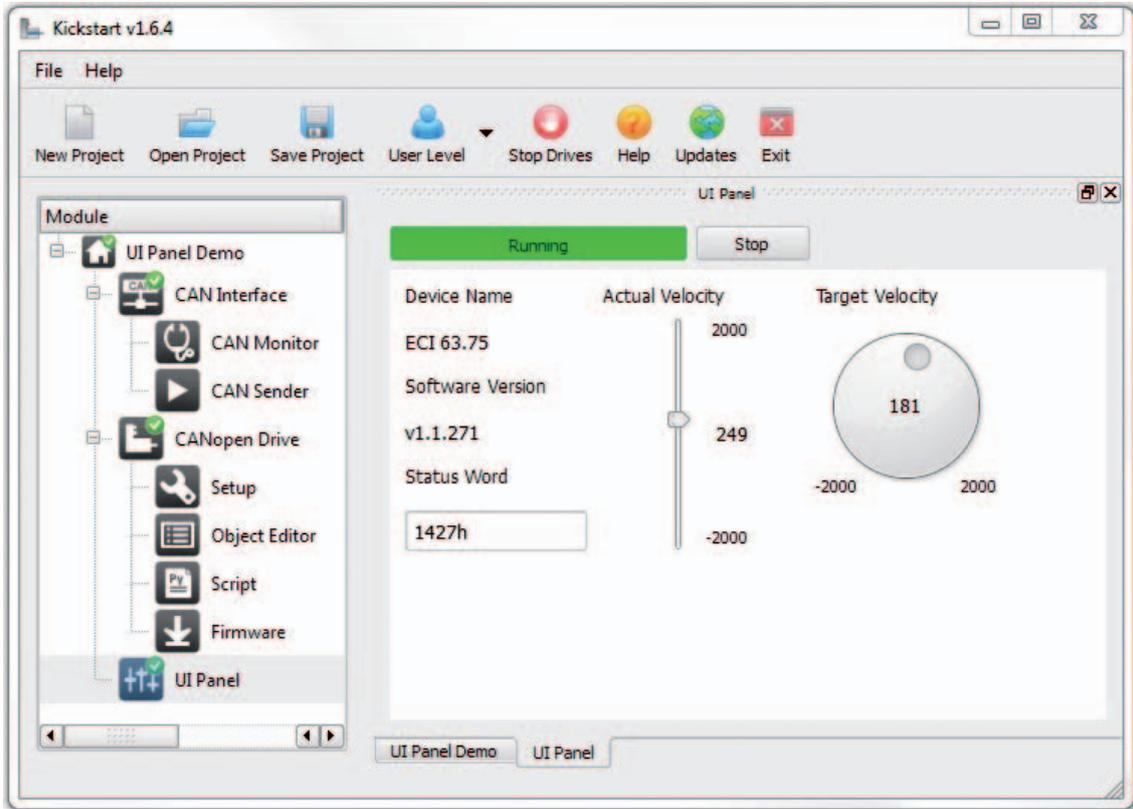
Kickstart ist eine modulare PC-Softwareplattform, die für eine Vielzahl von Konfigurations-, Inbetriebnahme- und Serviceaufgaben verwendet werden kann.

Sie ist sehr flexibel und verwendet anwendungsspezifische Projekt- und Vorlagendateien, um dem Benutzer genau die Funktionen und Steuerungen zur Verfügung zu stellen, die für eine bestimmte Aufgabe benötigt werden. Zum Beispiel könnte ein Außendiensttechniker mit einem Kickstart-Projekt arbeiten, das ausschließlich anwendungsspezifische Serviceaufgaben durchführt, z. B. ein Firmware-Update für einen Antrieb. Unterdessen könnten Entwickler und Anwendungstechniker mit einer skalierbaren, voll flexiblen Schnittstelle arbeiten, die die gesamte Funktionalitätspalette von Kickstart nutzt. Zum Beispiel:

- Inbetriebnahme und Konfiguration von Feldbusknoten wie Antriebseinheiten oder Controller.
- Feldbus-Überwachung (z. B. Generieren einer CAN-Bus-Ablaufverfolgungsdatei oder eines Bus-Monitors mit Such- und Filterfunktionen).
- Die Kommunikation über die RS485 läuft im Hintergrund ab. Der Anwender muß sich lediglich auf die definierten Objekte konzentrieren. Die Umsetzung auf das Protokoll erfolgt in der externen Konverterbox USB-CAN-RS485-Adapter, wie im Handbuch beschrieben.
- Anzeige von Scope-Daten: Live-Datenanzeige von einem oder mehreren Antriebsparametern, Protokollierung in einer Datei, Auslöser bei bestimmten Ereignissen.
- Maßgeschneiderte UI-Panels: Erstellungs-, Anzeige- und Bedienelemente für eine bestimmte Anwendung, z. B. Maschinensteuerung. Erstellen Sie ganz einfach per Drag & Drop ein einfaches Panel aus vordefinierten Schiebereglern und Drehscheiben.

3 Kickstart-Übersicht

Kickstart-Projekte bestehen aus Standardmodulen, welche als Bausteine dienen. Zum Beispiel die Module „CAN (USB) Interface“ oder „Drive01 / CANopen Drive“. Für spezialisierte Anwendungen (Scope, UI-Panels usw.) stehen zusätzliche Projektvorlagen zur Verfügung, die weitere Module enthalten.



4 Installation

Die Anwendungssoftware Kickstart wird in der Regel als ZIP-Archiv oder als selbstextrahierende Archivdatei (.exe) zur Verfügung gestellt.

Doppelklicken Sie die .exe Datei, um die Anwendung in ein Verzeichnis auf Ihrem Computer oder auf einem Wechselmedium zu kopieren, z. B. einen USB-Speicherstick.

Starten Sie anschließend die Software Kickstart durch Doppelklick auf **Kickstart.exe**.



HINWEIS

Es ist keine Installation (Setup) erforderlich, aber stellen Sie bitte sicher, dass alle Dateien und Ordner aus dem ursprünglichen .zip-Archiv in ein Verzeichnis auf Ihrem PC oder auf dem Wechselmedium extrahiert worden sind.



HINWEIS

Für Hardware-Zubehör wie die CAN-Schnittstelle (USB-CAN) ist möglicherweise die zusätzliche Installation von Gerätetreibern erforderlich. Die Treiberinstallation auf *Windows 8*, *Windows 7* und anderen aktuellen *Windows*-Versionen geschieht vollautomatisch, wenn Sie das Gerät zum ersten Mal anschließen. Bitte folgen Sie den *Windows*-Anweisungen.



HINWEIS

Die jeweils aktuellste Version der Kickstart.exe Datei steht auf der Internetseite von ebm-papst (www.ebmpapst.com) zum Download bereit.



HINWEIS

Komprimierte Dateien müssen mit einer dementsprechender Software wieder entpackt werden. Informationen hierzu finden Sie auf www.winzip.de.

5 Kickstart bedienen

5.1 Erste Schritte



WARNUNG

Verwenden Sie Kickstart nicht innerhalb eines Automatisierungsnetzwerks, in dem die CAN- oder CANopen-Kommunikation von Kickstart den normalen Betrieb stören könnte.



WARNUNG

Eine falsche Verkabelung kann elektronische Bauteile beschädigen, einschließlich des USB-CAN-RS485-Adapters, der CAN-RS485-Schnittstelle und des PCs, der mit der CAN-Schnittstelle verbunden ist.

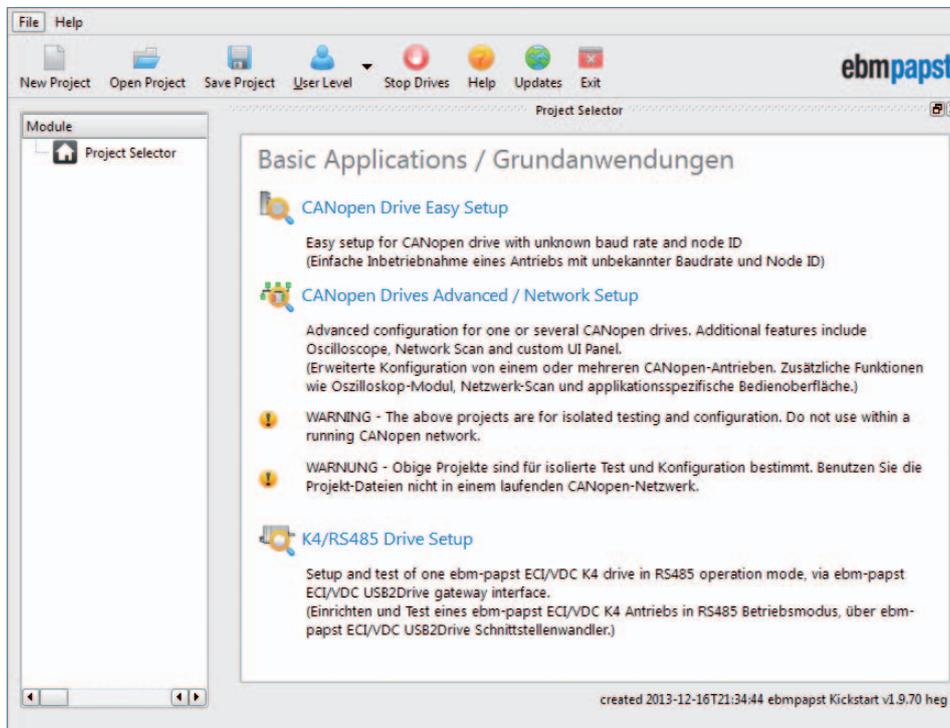


HINWEIS

Der USB-CAN-RS485-Adapter wird als Zubehör für die ebm-papst PC-Software „Kickstart“ benötigt, um den PC mit dem K4 Antrieb zu verbinden. Der Adapter kann unter der Material-Nr. 914 0000 400 bestellt werden.

- Verbinden Sie zunächst den **USB-CAN-RS485-Adapter** mit Ihrer Anlage, bevor Sie diese an den PC anschließen.
- Schalten Sie Stromversorgung der CAN-Knoten ein.
- Schließen Sie die CAN-Schnittstelle mithilfe des mitgelieferten USB-Kabels an Ihren PC an.
- Starten Sie die Softwareanwendung Kickstart.

Kickstart präsentiert eine Startseite, auf der Sie Ihre spezifische Anwendung oder ein Projekt auswählen können.



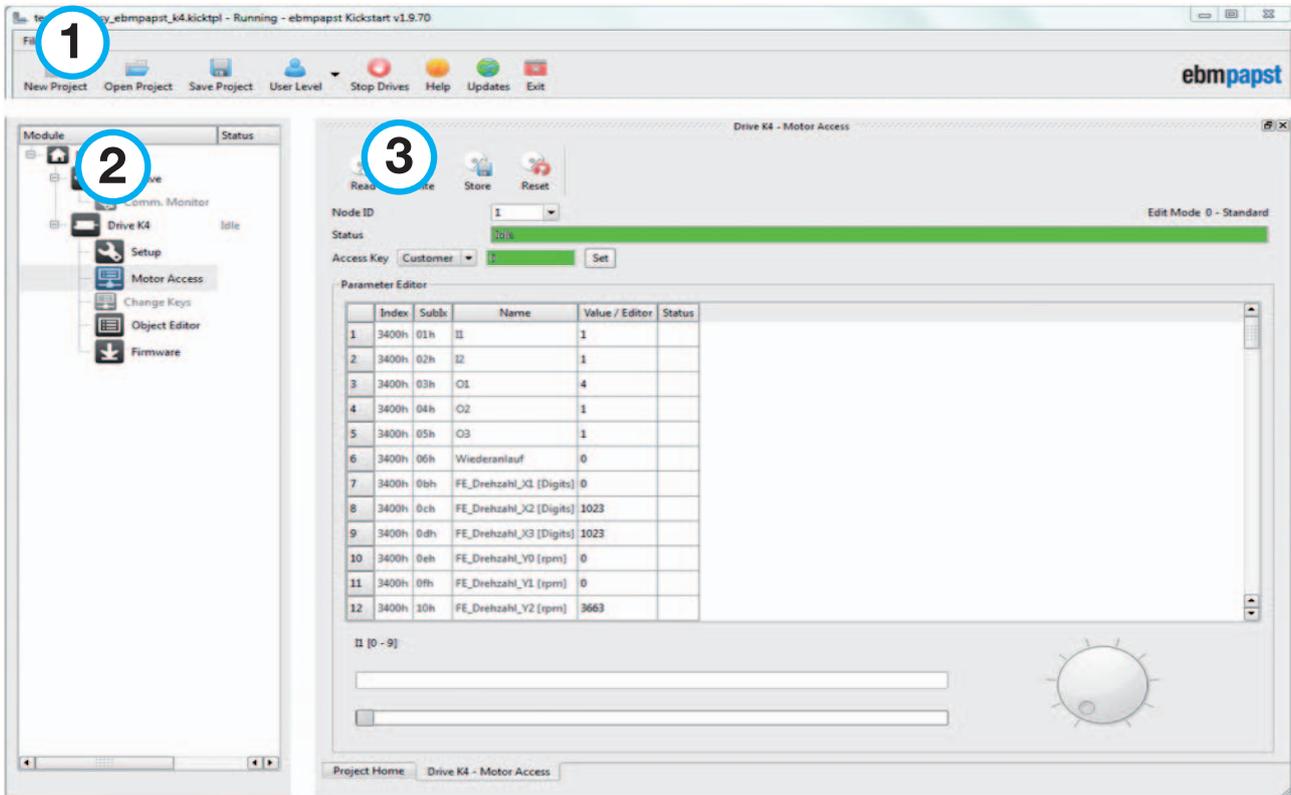
Kickstart-Hauptbildschirm und Projektauswahl

- Wählen Sie ein Projekt oder eine Anwendung aus der **Projektauswahl**-Liste auf der rechten Seite aus.
- Alternativ können Sie  **Open Project** verwenden, um eine Kickstart-Projektdatei (**.kickzip** oder **.kickpro**) zu laden, die Sie für Ihre spezifische Anwendung erhalten haben.

5 Kickstart bedienen

5.2 Hauptbildschirm und Symbolleiste

Der Kickstart-Hauptbildschirm ist in drei Bereiche unterteilt:



Kickstart – Hauptbildschirm: Anordnung der Bereiche

1 Menü- und Symbolleisten-Bereich

Der Menü- und Symbolleisten-Bereich ist für alle Kickstart-Projekte gleich.

☐ Funktionen der Hauptsymbolleiste

New Project

Öffnet die Startprojekt-Vorlage in Kickstart. Normalerweise können Sie über eine Projektauswahl Ihre spezifische Anwendung auswählen.

Open Project

Öffnet ein bestehendes Kickstart-Projekt (**.kickzip** oder **.kickpro**) oder eine Vorlage (**.kicktpl**).

Save Project

Speichert das aktuell geöffnete Projekt einschließlich aller Konfigurationen und zusätzlichen Dateien, die innerhalb des Projekts verwendet werden.

User Level

Ändert die Benutzerebene. Möglicherweise ist ein Passwort erforderlich.

5 Kickstart bedienen

Stop Drives

Versetzt alle im Modulbereich aufgeführten Antriebseinheiten sowie die Kommunikation zwischen Antrieb und dem Computer zurück in den Ruhezustand.



Die Funktion **Stop Drives** dient nicht als Not-Aus-Vorrichtung. Sie kann keine Sicherheits- oder Not-Funktionalität für Ihre Antriebsanwendung durchführen. Bitte vergewissern Sie sich, dass externe Not-Aus-Funktionen nach den Sicherheitsvorschriften und -bestimmungen Ihrer Antriebsanwendung wirksam sind.

2 Modulbereich

Dieser Bereich zeigt die Kickstart-Module, die im aktuell geöffneten Projekt verfügbar sind. Ausgegraute Elemente sind Module, die für den Zugriff eine höhere Benutzerebene erfordern.

3 Inhaltsbereich

Durch Anklicken eines Modulsymbols im Modulbereich 2 wird eine detaillierte Ansicht des Moduls oder einer bestimmten Funktion (z. B. „CAN Monitor“) im Inhaltsbereich 3 geöffnet. Der Inhaltsbereich ermöglicht das Andocken und Stapeln mehrerer Modulansichten. Beim Speichern eines Projekts wird die Anordnung des Inhaltsbereichs ebenfalls gespeichert und beim nächsten Öffnen des Projekts wiederhergestellt.

5.3 Benutzerebene

Benutzerebenen können in Kickstart-Projekten verwendet werden, um den Zugriff auf Module zu verhindern, die für den normalen Betrieb nicht erforderlich sind bzw. die mit zusätzlichen Vorsichtsmaßnahmen verwendet werden müssen.

Der Passwortschutz soll nicht den Zugriff durch bestimmte Benutzer verbieten, sondern vielmehr als Hinweis dienen, dass Sie sich jetzt auf eine höhere Ebene begeben. Die standardmäßigen Benutzerebenen und Passwörter sind:

Benutzerebene	Passwort
0 – Monitoring	–
1 – Running	–
2 – Running With Precaution	expert



Verschiedene Versionen der Kickstart-Software verwenden möglicherweise verschiedene Benutzerebenen und Passwörter.



Die höchsten User Level-Aufgaben können zusätzliches Expertenwissen über die gesteuerten Geräte erfordern oder gar ein Gerät unbrauchbar machen oder andere Schäden verursachen.

Ändern der erforderlichen Benutzerebene für ein Modul

Die für ein bestimmtes Modul mindestens erforderliche User Level wird in der Kickstart-Projektdatei gespeichert. Gehen Sie folgendermaßen vor, um diese Einstellungen zu ändern:

- Speichern Sie Ihr Projekt als unkomprimierte **.kickpro**-Projektdatei, wie in [_](#) beschrieben.
- Bearbeiten Sie die **.kickpro**-Datei mit einem standardmäßigen Text-Editor oder einem entsprechenden XML-Editor. Suchen Sie nach den Attributen **userLevel = "..."** und ändern Sie diese entsprechend.
- Um Ihre veränderte Projektdatei wieder in eine **.kickzip**- oder **.kicktpl**-Datei umzuwandeln, befolgen Sie bitte die Anweisungen in [_](#).

5 Kickstart bedienen

5.4 Projekte und Vorlagen

Alles, was Sie im Modul- und Inhaltsbereich von Kickstart sehen, ist in einer Kickstart-Vorlage oder -Projektdatei definiert. Die folgenden Dateiformate werden verwendet:

☐ **.kickpro Unkomprimierte Projektdateien**

Eine Kickstart **.kickpro**-Datei definiert alles, was Sie im Modul- und Inhaltsbereich von Kickstart sehen können: CAN-Schnittstellen, CANopen-Antriebe, Scope-Module usw.

Die **.kickpro**-Datei ist eine standardmäßige XML-Datei, die von erfahrenen Benutzern bearbeitet werden kann, um bestimmte Anwendungen aufzubauen (z. B. mit drei Antriebseinheiten statt einer arbeiten, die für jedes Modul benötigten Benutzerebenen ändern, zusätzliche HTML-Dokumentation erstellen).



Stellen Sie bei der manuellen Bearbeitung von **.kickpro**-Dateien sicher, dass Sie eine Kopie der Originaldateien aufbewahren. Prüfen Sie sorgfältig, dass Ihre geänderte Datei eine gültige **.xml**-Datei ist. Ein guter XML-Editor kann bei der Validierung des XML-Formats helfen und ungültige Änderungen verhindern, die die Struktur unterbrechen könnten.

Die **.kickpro**-Datei wird oft von zusätzlichen Dateien begleitet, die erforderlich sind, um die vordefinierten Aufgaben des Projekts auszuführen. Diese Dateien können umfassen::

- **.css** – Dateien für die Formatierung von Dokumentation.
- **.py** – Dateien für [Python](#)-Skriptcode, der von Kickstart ausgeführt oder kompiliert und in einer Antriebseinheit geladen wird.
- **.bin** – Firmwaredateien für die Aktualisierung der Anwendungsfirmware der Antriebseinheit.

☐ **.kickzip Komprimierte Projektdateien**

.kickzip – Dateien sind komprimierte Dateiarchive mit folgendem Inhalt:

- Eine **.kickpro**-Datei mit dem gleichen Namen wie das **.kickzip**-Archiv.
- Zusätzliche Hilfsdateien (**.css**, **.py**, **.bin**, ...) wie oben beschrieben.

Beim Öffnen eines **.kickzip**-Projekts erstellt Kickstart einen temporären Ordner und extrahiert alle Dateien in dieses Verzeichnis.

Beim Speichern eines **.kickzip**-Projekts erstellt Kickstart zunächst eine aktualisierte **.kickpro**-Datei im temporären Ordner und anschließend die komprimierte **.kickzip**-Datei mit allen Dateien aus dem temporären Ordner, einschließlich Unterverzeichnissen.



Sie können die **.kickpro**-Datei und alle Hilfsdateien extrahieren, indem Sie das Menü **File >**  **Save As...** verwenden und **Kickstart Uncompressed Project (.kickpro)** anstelle des standardmäßigen **.kickzip**-Dateiformats wählen.



Um ein komprimiertes **.kickzip**-Dateiarchiv zu erstellen, müssen Sie zunächst ein standardmäßiges **.zip**-Dateiarchiv erstellen, das die **.kickpro**-Datei und alle erforderlichen Hilfsdateien enthält. Ändern Sie anschließend die Dateierweiterung von **.zip** zu **.kickzip**.

5 Kickstart bedienen

☐ .kicktpl Vorlagendateien

Eine Kickstart-Installation liefert eine Reihe von vordefinierten Projekten für spezifische Aufgaben, die als **.kicktpl**-Vorlagendateien gespeichert sind. Die Projektauswahl und die in [Erste Schritte](#) definierten Projekte sind Beispiele für solche Vorlagen.

Das Öffnen einer **.kicktpl**-Datei ist wie das Öffnen einer **.kickzip**-Datei, außer dass zum Speichern des Projekts kein Dateiname gewählt ist. Wenn Sie das Projekt zum ersten Mal mit  **Save Project** speichern, werden Sie aufgefordert, einen Speicherort und einen Namen für das neue Projekt anzugeben – dies wird anschließend als **.kickzip**-Datei gespeichert.



HINWEIS

Sie können eigene Vorlagen erstellen, indem Sie eine **.kickzip**-Datei nehmen und einfach ihre Erweiterung zu **.kicktpl** ändern.

5.5 Bearbeiten des Modulbaums

Um den Modulbaum im Kickstart-Hauptbildschirm zu bearbeiten, **klicken Sie mit der rechten Maustaste** auf eine Modulkachel und wählen Sie eine Aktion aus dem angezeigten Kontextmenü.



HINWEIS

Benutzerebene 2 ist erforderlich, um das Kontextmenü erreichbar zu machen.

Sie können:

- Mit **Cut**, **Copy** oder **Paste** ein Modul im Modulbaum ausschneiden, kopieren oder einfügen. Wenn Sie Module zwischen verschiedenen Projekten kopieren möchten, öffnen Sie eine zweite Instanz von Kickstart mit dem zweiten Projekt.
- Mit **Delete** ein Modul und seine untergeordneten Elemente aus dem Projektbaum löschen.
- Über **Rename** ein Modul umbenennen.

Die verschiedenen Kickstart-Module haben Master/Slave-Beziehungen, die im Modulbaum nicht sichtbar sind. Ein CANopen Drive-Modul verwendet beispielsweise immer ein CAN Interface-Modul für die CAN-Bus-Kommunikation. Wenn Sie ein solches Slave-Modul umbenennen, z. B. von **CAN Interface** zu **CAN-USB**, werden die Master/Slave-Verbindungen automatisch aktualisiert. Wenn Sie jedoch ein Modul per **Paste** in ein Projekt einfügen, ist Kickstart möglicherweise nicht in der Lage, automatisch die korrekten Slaves für dieses Modul zu wählen. In diesem Fall ist es erforderlich, dass Sie die ursprüngliche unkomprimierte Projektdatei (**.kickpro**-Datei) betrachten und die damit verbundenen **<slave>**-Informationen manuell bearbeiten.



HINWEIS

Zusätzliche Leitfäden für Entwickler bezüglich der Struktur der **.kickpro**-Datei sind auf Anfrage erhältlich. Bitte kontaktieren Sie Ihren Anwendungssupport.



WARNUNG

Die Bearbeitungsaktionen im Modulbaum sind nicht reversibel. Vergewissern Sie sich, dass Sie über eine Sicherungskopie des Projekts verfügen, bevor Sie seine Inhalte verändern.

6 Kickstart Module

6.1 CAN Interface

Je nach Vorlage (Template) sind unterschiedliche Module / Modulnamen vorhanden, d.h. die Module haben nicht immer alle hier beschriebenen Funktionen.

Das  **CAN Interface**-Modul steuert die Kommunikationsschnittstelle zum CAN-Netzwerk. Dies ist normalerweise eine  CAN-USB-Schnittstelle mit den folgenden Parametern:

- **Kommunikationsport**

Die Standardeinstellung (**first found**) sollte für die meisten Anwendungen geeignet sein. Wenn Sie einen COM-Anschluss manuell wählen müssen (z. B. „COM6“), öffnen Sie den Windows-Geräte-Manager, um die richtige COM-Anschlussnummer herauszufinden, die mit der  CAN-USB-Schnittstelle verwendet werden soll.

- **Baud Rate**

Die Einstellung (`scan`) wird verwendet, wenn Sie eine Verbindung zu einem CAN-Netzwerk oder Gerät mit unbekannter Baudrate herstellen. Der Baudrate-Scan setzt aktiv die Knoten zurück und versucht, Zugriff auf alle vorhandenen CAN-Knoten zu erhalten.



Der Baudrate-Scan setzt die CANopen NMT-Kommunikation zurück, was laufende Anwendung stört. Verwenden Sie den Baudrate-Scan niemals in in einem Automatisierungsnetzwerk oder in einer Antriebsanwendung im Normalbetrieb.



Der Baudrate-Scan oder vorherige Zugriffsversuche mit unpassender Baudrate können das Netzwerk unzugänglich machen. Das heißt, dass ein oder alle CAN-Knoten oder die Steuerung der CAN-Schnittstelle in einen „Bus-Off“-Zustand übergegangen ist und nicht mehr kommuniziert. Schlägt der Baudrate-Scan ohne ersichtlichen Grund fehl, so schalten Sie bitte alle CAN-Knoten und anschließend die CAN-Schnittstelle selbst aus und wieder ein.

6.1.1 CAN Monitor

Verwenden Sie das Modul  **CAN Monitor**, um die aktuelle Kommunikation auf dem CAN-Bus anzuzeigen oder um sie mit der Funktion **Record** als Textdatei zu protokollieren.

Mit der Funktion **Find/Filter** können Sie nach spezifischen CAN-Telegramme suchen bzw. eine gefilterte Version des CAN-Bus-Verkehrs anzeigen. Die Dropdownliste in **Find/Filter** enthält einige Voreinstellungen für nützliche Filteroptionen, die zeigen, wie der Filter arbeitet.

Filter Bedingungen

`id = <MyCobID>` – nur CAN-Frames mit einer bestimmten COB-ID anzeigen.

`id != <MyCobID>` – nur Frames anzeigen, die diese COB-ID NICHT enthalten.

`dir = tx` – nur Frames anzeigen mit Übermittlung durch Kickstart.

`index = <ObjIdx>`, `subindex = <ObjSubIdx>` – nur CANopen-Frames anzeigen, die im Zusammenhang stehen mit einem SDO mit der angegebenen `ObjIdx` und `ObjSubIdx`.

`data = 80*` – nur Frames anzeigen, bei denen die CAN-Daten mit hex 80 beginnt.

Multiple Bedingungen

Sie können mehr als einen Filterzustand in einem Filter verwenden, z. B.

`dir = tx, index = <ObjIdx>`, `subindex = <ObjSubIdx>` - nur Frames aus Kickstart anzeigen, die `ObjIdx` und `ObjSubIdx` enthalten. Nur CAN-Frames, die alle Filterbedingungen erfüllen (`dir`, `index` and `subindex`) werden angezeigt.

6 Kickstart Module

Einen Filter invertieren

Wenn Sie Ihren Filter mit dem Schlüsselwort **exclude** beginnen, wird alles ausgeblendet, was die Filterbedingungen erfüllt, z. B.

```
exclude id = 581h, index=606c, index=6041 – nur SDO-Verkehr für Objekte 606c und 6041 für CANopen-Knoten 1 anzeigen.
```

Kommentare

Verwenden Sie ein #, um zusätzliche Freitextkommentare über Ihren Filter hinzuzufügen, z. B.

```
id != 81-ff # hide emergency msg
```

Auto Filter

Wenn Sie **Auto Filter** drücken, wird eine Filterbedingung hinzugefügt, um den CAN-Frametyp mit dem häufigsten Auftreten zu unterdrücken. Die wiederholte Verwendung von **Auto Filter** ist ein bequemer Weg, um eine fast ruhige Anzeige zu erhalten, in der nur seltene CAN-Frames (z. B. Kontroll-PDOs oder Notfälle) angezeigt werden.

Überwachte Frames aufzeichnen

Verwenden Sie die Schaltfläche **Record**, um mit der Aufzeichnung von CAN-Frames in einer Textdatei zu beginnen. Standardmäßig wird die nachverfolgte Datei im Ordner der Projekte mit dem Datums- und Zeitstempel des Aufzeichnungsbeginns gespeichert. Vergewissern Sie sich, dass Sie das Projekt speichern, damit Sie auch in Zukunft auf die aufgenommene Datei zugreifen können.

Um eine aufgezeichnete Datei zu laden, verwenden Sie die Schaltfläche **Open** – damit werden alle CAN-Frames aus der aufgezeichneten Datei in den CAN-Monitor geladen. Optional können Sie **Delete All** verwenden, um den Monitor vor dem Öffnen der Datei zu löschen.



HINWEIS

Filterung wirkt sich nur auf die Bildschirmanzeige aus, nicht auf den eigentlichen CAN-Framepuffer oder Ablaufverfolgungsdatei-Aufnahmen. Nach dem Deaktivieren von **Find / Filter** wird wieder der vollständige CAN-Verkehr angezeigt.



HINWEIS

Das CAN-Monitor-Display erfasst nur eine begrenzte Menge von CAN-Frames im Speicher. Verwenden Sie bitte die Funktion **Record**, um eine unbegrenzte Anzahl von CAN-Frames in einer standardmäßigen Textdatei (.txt) aufzuzeichnen. Sie können eine CAN-Ablaufverfolgungsdatei mit einem beliebigen Texteditor oder mithilfe der Schaltfläche **Open** wieder öffnen. Dadurch werden alle Frames in den Monitor geladen

6.1.2 CAN Sender

Das Modul  **CAN Sender** kann für einfache CAN-Testanwendungen verwendet werden, die das Senden eines oder mehrerer fester CAN-Frames erfordern.

Es ist kein vollständiges Skriptmodul, sondern eine einfache Makrosequenz für das schnelle und einfache Testen.

Verfügbare Befehle:

```
send <ID>, <data Size>, <data Hex>
```

```
delay <milliseconds>
```

```
wait <ID>, <data size>, <data Hex>[, <timeoutMillisec>]
```

ID und data size können als Hex-Ausdrücke zur Verfügung gestellt werden:

```
send 0x7e5, 2, 04 01
```

oder

```
send 7e5h, 0x2, 04 01
```

wait unterstützt die Platzhalterzeichen * und ? für die <data hex>-Teile. Ein zusätzliches Zeitlimit kann festgelegt werden.

6 Kickstart Module

Beispiel:

```
wait 581h,8,4b 41 60 00 ?? ?4 *,1000
```

(Wartet auf ein Statuswort, bei dem die höchsten vier Bits „4“ sind. Gibt nach einer Sekunde auf.)

Die folgenden Makros werden innerhalb der Befehle `send` und `wait` unterstützt:

`canid(<expression>)` – Berechnet den Ausdruck und gibt eine CAN-ID im Hex-Format wieder.

`canunsigned32(<expression>)` or `caninteger32(<expression>)` – Berechnet den Ausdruck und gibt einen 4-Byte-CAN-Datenblock wieder

`canunsigned16(<expression>)` or `caninteger16(<expression>)` – Wie oben, jedoch mit 2 Byte.

`canunsigned8(<expression>)` or `caninteger8(<expression>)` – Wie oben, jedoch mit 1 Byte.

Beispiel:

```
nodeId = 1
velocity = 1000
send canid(0x600 + nodeId),8,23 ff 60 00 canunsigned32(velocity)
```

Eine einfache `do ... loop`-Syntax, die für Endlosschleifen verwendet werden kann. Oder verwenden Sie `do...loop until found` in Kombination mit einem `wait`-Makro. .

Example:

```
do
  do
    ; request status
    send canid(0x600 + nodeId),8,40 41 60 00 00 00 00 00
    ; wait for target reached (status word upper four bits = 4)
    ; using 0.5 seconds timeout
    wait canid(0x580 + nodeId),8,4b 41 60 00 ?? ?4 ?? ??,500
  loop until found
  ; target has been reached. wait until drive moves again
  do
    send canid(0x600 + nodeId),8,40 41 60 00 00 00 00 00
    wait canid(0x580 + nodeId),8,4b 41 60 00 ?? ?0 ?? ??,500
  loop until found
loop
```

6.2 Drive01 / CANopen Drive

Je nach Vorlage (Template) sind unterschiedliche Module / Modulnamen vorhanden, d.h. die Module haben nicht immer alle hier beschriebenen Funktionen.

Das  **Drive 01 / CANopen Drive**-Modul bietet einfache Konfigurations-, Test- und Serviceaufgaben für CANopen-Positionierantriebe, die von dieser Kickstart-Anwendung unterstützt werden. Details zu den aktuell unterstützten Antriebsmodelle finden Sie in den Kickstart-Projekten/Vorlagen, die in Ihrer spezifischen Kickstart-Softwareversion enthalten sind.



Die folgenden Bereiche in Drive01 / CANopen Drive-Panels bieten einen grundlegenden Überblick über die verfügbaren Funktionen. Einzelheiten können vom tatsächlich verwendeten Antriebsmodell abhängen und sind im Bereich Kickstart-Projekte/Vorlagen in der Anwendung selbst dokumentiert.

6 Kickstart Module

6.2.1 Setup

Das  **Setup**-Panel wird für verschiedene Konfigurations- und einfache Testaufgaben verwendet:



Gehen Sie bei den Ersteinrichtungsfunktionen vorsichtig vor. Es wird dringend empfohlen, Ersteinrichtungsfunktionen mit nur einer an Kickstart angeschlossenen Antriebseinheit durchzuführen und nicht innerhalb einer vollen CANopen-Installation mit mehreren Antriebsknoten.

Initial Setup

Zuweisen oder Ändern der CAN Node ID oder der CAN Baud Rate.

Folgende Panels sind nach dem Starten eines Antriebsknotens mit der Schaltfläche **Start Drive** verfügbar:

- **Test Velocity**
Schalten Sie den Antrieb in den Betriebsmodus „Profile Velocity Mode“ und führen Sie grundlegende Geschwindigkeitssteuerung durch (target velocity, acceleration und deceleration einstellen).
- **Test Position**
Schalten Sie den Antrieb in den Betriebsmodus „Profile Position Mode“ und führen Sie grundlegende Positionierungssteuerung durch. (target position einstellen und Positionierung ermöglichen; zusätzliche Profilpositionsparameter wie Geschwindigkeit einstellen).
- **Homing**
Grundlegende Nullpunktsuchmethoden testen, einschließlich Homing Mode 35, um die aktuelle Position auf Null (neue Referenzposition) zurückzusetzen, ohne den Antrieb zu bewegen.

6.2.2 Object Editor

Das Modul  **Object Editor** wird zum Lesen, Bearbeiten und Schreiben von Geräteparametern verwendet. Hier können Sie auch jedes beliebige Datenobjekt definieren, das in anderen Modulen wie CANopen-Knoten, Scope oder UI-Panel verwendet werden kann.

Das Panel Object Editor besteht aus zwei Teilen:

- **Object Dictionary (Device Description)**
Der obere Teil des Panels Object Editor wird verwendet, um ein Wörterbuch der verfügbaren Parameter für ein Gerät anzuzeigen. Die Beschreibungen der Parameter werden durch eine XDD / XDC-Konfigurationsdatei definiert. Verwenden Sie die Funktion  **Open Dict** um eine andere / neue Konfigurationsdatei zu laden.
- **Object Editor**
Ziehen Sie Parameter (einzeln oder mehrfach) aus dem Object Dictionary in die Tabelle Object Editor, um sie zum Lesen, Bearbeiten und Schreiben verfügbar zu machen.

Objekteditor Grundfunktionen



Read

Lesen Sie die ausgewählten Werte aus dem Gerät (CANopen Upload). Wenn in der Tabelle Object Editor nichts ausgewählt ist, werden alle Werte gelesen.



Edit

Verwenden Sie diese Funktion bei einem Element oder doppelklicken Sie es in der Spalte Value / Editor. Die Werte werden nicht sofort auf das Gerät geschrieben, außer wenn  **Synch Mode** verwendet wird.



Write

Schreiben Sie die ausgewählten Werte auf das Gerät (CANopen Download). Wenn in der Tabelle Object Editor nichts ausgewählt ist, werden alle schreibbaren Werte heruntergeladen.

6 Kickstart Module

Store

Speichern Sie die geschriebenen Parameter dauerhaft. Das heißt, dass das CANopen-Gerät diese Werte nach einem Spannungsreset weiterhin verwenden. Diese Funktion wird nach dem Schreiben der geänderten Werte mithilfe der  **Write**-Funktion verwendet.

Reset

Setzen Sie alle Geräteparameter auf die Werkseinstellungen zurück.

Synch Mode

Lesen Sie schreibgeschützte Parameter in regelmäßigen Abständen. Schreibbare Parameter werden einmal gelesen. Danach wird jede Änderung, die durch den Benutzer eingegeben wird, sofort auf das Gerät heruntergeladen.

Default

Kopieren Sie für die ausgewählten Zeilen den Wert aus der Spalte Default des Wörterbuchs in die Spalte Value. Wenn keine Zeilen ausgewählt sind, werden alle Objekte auf die Standardwerte aktualisiert. Um diese Standardeinstellungen in das Gerät zu schreiben, verwenden Sie  **Write**.

Skalieren

Bei der Verwendung auf Benutzerebene 2 können Sie die Spalte **Scaling** verwenden, um zu definieren, wie die Werte skaliert werden sollen, bevor sie in der Spalte **Value** angezeigt werden.

Es werden die gleichen Formate unterstützt, die in Skalierung für den Scope verwendet werden.



Für **Lesen- / Schreibwerte** sind nur einfache Skalierungsfaktoren wie **0,25** zulässig. Bei **Write** wird der eingegebene Wert in den ursprünglichen, rohen, „unskalierten“ Wert zurückgewandelt. Bei der Verwendung von Skalierungsausdrücken wie **=x/20+10** wird der Objektwert schreibgeschützt und kann nicht mehr bearbeitet oder auf das Gerät geschrieben werden.

6.2.3 Script

Das Modul  **Script** wird verwendet, um eingebettete [Python](#)-Skripte zu bearbeiten, zu kompilieren und in eine Antriebseinheit herunterzuladen, die sie unterstützt.



Einzelheiten zum eingebetteten Python-Modul, Beispielcode und Anwendungen finden Sie in der Dokumentation für Ihren CANopen-Antrieb oder kontaktieren Sie den Anwendungssupport.

6.2.4 Firmware

Mit dem Modul  **Firmware** können Sie die Anwendungsfirmware Ihres Geräts mit einer neuen Softwareversion aktualisieren, die im **.bin**-Dateiformat zur Verfügung gestellt wird.



Stellen Sie vor dem Start eines Firmwaredownloads sicher, dass die Node ID korrekt ist und dass auf das Gerät zugegriffen werden kann, z. B. mit Hilfe des Panels Antriebs-Setup.



Bitte stellen Sie sicher, dass Sie eine korrekte **.bin**-Datei verwenden, die zu Ihrem spezifischen Gerät passt. Kickstart führt vor dem Herunterladen der Datei keine zusätzlichen Kontrollen durch. Mit einer falschen oder unpassenden **.bin**-Datei kann Ihr Antrieb beschädigt werden, was es unmöglich macht, beim zweiten Versuch eine korrekte **.bin**-Datei herunterzuladen.

6 Kickstart Module

6.3 Motor Access / CANopen Node

Je nach Vorlage (Template) sind unterschiedliche Module / Modulnamen vorhanden, d.h. die Module haben nicht immer alle hier beschriebenen Funktionen.

Das Modul  **CANopen Node** bietet einen Bildschirm zur Grundkonfiguration für jede Art von CANopen-Knoten, z. B. einen Drittanbieter-Servoantrieb, ein I/O-Modul oder ein Sensorgerät.

Die Tabelle **Parameter Editor** enthält eine Liste der CANopen-Objekte und ihrer Werte, ähnlich dem [Object Editor](#).

Für das ausgewählte Objekt werden unter der Tabelle zusätzliche Steuerelemente zum Anzeigen und Bearbeiten angezeigt. Diese Elemente lassen sich durch Verändern der Spalte UI-Format in **Design Mode** konfigurieren.



HINWEIS

Das Aussehen und Verhalten des **Parameter Editor** ist abhängig von der Einstellung des Bearbeitungsmodus. Einige der nachfolgend beschriebenen Funktionen sind nur für bestimmte **Edit Mode**-Einstellungen relevant.

+ Parameter Editor functions



Read

Lesen Sie den Wert eines Parameters oder der gesamten Liste (je nach Bearbeitungsmodus).



Write

Schreiben Sie den Wert des ausgewählten Parameters oder der gesamten Liste auf das Gerät (je nach Bearbeitungsmodus).



Store

Speichern Sie die geschriebenen Parameter dauerhaft. Das heißt, dass das CANopen-Gerät diese Werte nach einem Spannungsreset weiterhin verwenden. Diese Funktion wird nach dem Schreiben der geänderten Werte mithilfe der  **Write**-Funktion verwendet.



Reset

Setzen Sie alle Geräteparameter auf die Werkseinstellungen zurück.



Design Mode

Ermöglicht das Hinzufügen und Entfernen von Parameter aus der Liste sowie das Bearbeiten der Spalten **UI Format** und **Comment**, um die Steuerelemente für Anzeige und Bearbeitung für dieses Element zu verändern.

Im **Design Mode** können Sie:

- Weitere Objekte hinzufügen: Ziehen Sie Datenobjekte aus dem [Object Editor](#) und legen Sie sie in den Parameter Editor. Alternativ können Sie per **Copy** Datenobjekte aus dem [Object Editor](#) kopieren und mit **Paste** in der Parametertabelle einfügen.
- Ändern Sie **Name**, **UI Format** und **Comment** des Parameters. Das **UI Format** konfiguriert die UI-Elemente, die für den Parameter angezeigt werden, wie in der Dokumentation des UI-Formats beschrieben. **Comment** definiert den zusätzlichen Beschreibungstext, der unter dem Namen im Bereich der UI-Elemente angezeigt wird.
- Verwenden Sie Skalierungsfaktoren und -formeln in der gleichen Weise wie im [Object Editor](#).
- Ändern Sie den Bearbeitungsmodus.



HINWEIS

Benutzerebene 2 ist für den Zugriff auf **Design Mode** erforderlich.

6 Kickstart Module

6.3.1 CANopen Node – Edit Mode

Der **Edit Mode** wirkt sich auf das Aussehen und Verhalten des Moduls CANopen-Knoten aus. Er kann vom **Design Mode** abgewandelt werden.



Benutzerebene 2 ist für den Wechsel zu **Design Mode** erforderlich.

HINWEIS

Name	Description
0 - Standard	Das Lese-/Schreibverhalten ist ähnlich dem Object Editor , Read / Write ist nur im aktuell ausgewählten Objekt möglich.
1 - List Read	Keine Write / Edit -Funktionalität, Read liest immer die gesamte Liste der Parameter, nicht nur ein einzelnes Objekt.
2 - List Read/Write	Keine Store / Reset -Funktionalität. Read / Write wirkt sich immer auf die gesamte Liste der Parameter aus.
3 - Auto Store	So wie EditMode = 2 , aber Write führt nach dem Schreiben der Werte immer eine Store -Aktion aus, um die Parameteränderung dauerhaft zu machen.

6.3.2 CANopen Node - UI Format



Benutzerebene 2 ist erforderlich, um zum **Design Mode** zu wechseln, der die Spalte **UI Format** zur Bearbeitung zur Verfügung stellt.

HINWEIS

Die Spalte **UI Format** in der Tabelle **Parameter Editor** konfiguriert die Bearbeitungssteuerung für jedes Datenobjekt.

Benutzerdefinierte **UI Format**-Einstellungen für jedes Objekt ermöglichen eine saubere und intuitive Spontanvisualisierung seines Wertes. Beispielsweise ein Dreh- und ein Schieberegler für Ganzzahldaten innerhalb eines wohl definierten Min./Max.-Bereichs oder ein Kombinationsfeld für Multiple-Choice-Parameter.



Die nachfolgend beschriebenen **UI Format**-Einstellungen sind nicht identisch mit den für das **UI-Panel** verwendeten. Wenn Sie jedoch ein Datenobjekt aus dem **UI-Panel** oder dem [Object Editor](#) ziehen und es im **CANopen Node**-Panel ablegen, werden entsprechende Konvertierungen und Ergänzungen angewendet.

HINWEIS

Der **UI Format**-Wert ist eine Liste von Schlüsselwert-Attributpaaren, getrennt durch Semikolons („;“). Nachfolgend sehen Sie eine Beschreibung der verfügbaren Attribute:

type

Definiert die Art der zu visualisierenden Daten.

type	Beschreibung
integer	Ganzzahlwerte werden als ein Textfeld, eine Drehscheibe und ein Schieberegler angezeigt. Siehe das Attribut range , um Min./Max.-Werte zu definieren.
text	Texte werden nur mithilfe eines Textfelds angezeigt.
combo	Ein Kombinationsfeld. Dies ist nützlich für Objekte, die eine begrenzte Anzahl verschiedener Werte mit verschiedenen Bedeutungen annehmen. Die verfügbaren Werte und ihre Beschreibungen werden über das Attribut options definiert. Unten sehen Sie weitere Informationen und ein Beispiel.

6 Kickstart Module

range

Definiert die minimalen und maximalen Werte. Zum Beispiel:

```
range:0,100
```

```
range:-100,100
```

options

Verwendet mit `type: combo`. Dies spezifiziert eine Liste verfügbarer Optionen und ihrer entsprechenden Werte.

Format:

```
options:<value_1>,<text _1>|<value_2>,<text_2> | ... (etc.)
```

Hier ist beispielsweise ein Kombinationsfeld für die Auswahl aus 4 verschiedenen Werten (0, 1, 3 or 6):

```
type:combo; options:0,Reset|1,Profile Position|3,Velocity Mode|6,Homing Mode
```

Beispielhafte UI Format-Definitionen

UI Format	Beschreibung
<code>type:text</code>	Ein einfaches Etikett oder ein Textfeld. Dies kann verwendet werden, um einen beliebigen Wert anzuzeigen oder zu bearbeiten (falls schreibbar).
<code>type:integer; range:0,100</code>	Ein Textfeld, ein Schieberegler und eine Drehscheibe, die Werte zwischen 0 und 100 akzeptieren.
<code>type:combo; options:0,Reset 1,Profile Position 3,Velocity Mode 6,Homing Mode</code>	Ein Kombinationsfeld mit den spezifizierten Optionen (Reset, Profile Position, Velocity Mode, Homing Mode) und entsprechenden Werten (0, 1, 3, 6).

6.4 Network Manager

Je nach Vorlage (Template) sind unterschiedliche Module / Modulnamen vorhanden, d.h. die Module haben nicht immer alle hier beschriebenen Funktionen.

Verwenden Sie den  **Network Manager**, um einen Baudrate- und Knoten-Scan für ein unbekanntes Netzwerk durchzuführen und ein Kickstart-Modul Aufbau und Betrieb für jeden gefundenen und unterstützten Laufwerkknoden einzurichten.

Network Scan

Verwenden Sie dieses Verfahren für einen erfolgreichen Scan:

- Verbinden Sie die **CAN-Schnittstelle** mit dem CANopen-Netzwerk.
- Schließen Sie die CAN-Schnittstelle über USB an Ihren PC an.
- Schalten Sie Stromversorgung der CANopen-Knoten ein.
- Achten Sie darauf, dass im CAN Interface-Modul die Option `baud rate` entweder auf (Scan) oder auf die tatsächliche Baudrate Ihres Netzwerks eingestellt ist.
- Wählen Sie  **Scan Network** um mehr über die folgenden Punkte zu erfahren:
 - **Scanned Node IDs** – alle vorhandenen Knoten-IDs in diesem Netzwerk,
 - **Configurable Node IDs** – unterstützte Laufwerkknoden, die verfügbar sind für Setup und Betrieb durch Kickstart.
- Wählen Sie  **Build Project** um ein Kickstart-Modul für jede der IDs zu erstellen, die im Textfeld **Configurable Node IDs** aufgeführt sind.



WARNUNG

Verwenden Sie niemals einen Baudrate-Scan oder ähnliche Konfigurationsaufgaben bei laufenden Maschinen oder in einer CANopen-Anwendung bei normalem Betrieb. Siehe auch die zusätzlichen Informationen in der Modulbeschreibung der **CAN-Schnittstelle**.

6 Kickstart Module

Node Monitor

Sie können den Node Monitor für eine einfache Netzwerkübersicht verwenden, welche einige grundlegende Informationen über jede Node ID zeigt, die im Textfeld **Scanned Node IDs** aufgeführt ist.

- Stellen Sie sicher, dass die Liste **Scanned Node IDs** auch wirklich die Liste von Knoten ist, die Sie überwachen möchten.
- Klicken Sie  **Node Mon. Config**, um eine Liste von Parametern zu erhalten, die für jeden Knoten gelesen werden sollen. Sie können zusätzliche **Datenobjekte** aus dem **Object Editor-Modul** ziehen und sie in der Tabelle **Node Monitoring Configuration** ablegen.
- Verwenden Sie in der Spalte **UI Format** das Schlüsselwort **updates:**, um festzulegen, wie die Werte eingeholt werden sollen, z. B.
 - **updates:r** – nur einmal lesen beim Starten von **Node Monitor**
 - **updates:r2000** – alle 2000 Millisekunden lesen
(siehe den Bereich UI-Panel / UI-FormatDefinition für weitere Informationen zum Spezifizierer updates:).
- Klicken Sie auf  **Node Monitor**, um die Netzwerkübersicht / Knotenüberwachung zu starten.



HINWEIS

Node Monitor verwendet zum Auslesen der Werte aus den Netzknoten SDO-Transfer. Bitte stellen Sie sicher, dass dies nicht in Konflikt mit anderen Operationen im CANopen-Netzwerk steht, z. B. **Firmware-Downloads**.

6.5 Scope

Je nach Vorlage (Template) sind unterschiedliche Module / Modulnamen vorhanden, d.h. die Module haben nicht immer alle hier beschriebenen Funktionen.

Das Modul  **Scope** erlaubt dem Benutzer, Daten in skalierbaren Plots anzuzeigen. Die Daten können in einer Datei erfasst und später durchgesehen werden.

6.5.1 Scope – Erste Schritte

- Ziehen Sie ein oder mehrere Datenobjekte aus dem **Object Editor** und legen Sie sie in der **Scope**-Konfigurationstabelle ab.
- Stellen Sie sicher, dass das Feld Probenahme auf den Modus **Software** gestellt ist. Verwenden Sie das Feld **Rate**, um die Abtastrate (Proben/ms) einzustellen. Verwenden Sie das Feld Skalierung, um die Werte mithilfe eines Faktors oder eines grundlegenden mathematischen Ausdrucks zu ändern.
- Verwenden Sie die Symboleistenschaltflächen **Start** und **Stop**, um Probenahme zu aktivieren oder zu deaktivieren.
- Um ein Plot-Panel für ein überwacht Datenobjekt zu erstellen, doppelklicken Sie seine Zeile in der Konfigurationstabelle. Der Plot zeigt die neuesten verfügbaren Daten an. Um einen Plot für mehrere Datenobjekte zu erstellen, verwenden Sie **Create Plot** und wählen Sie das Datenobjekt (oder mehrere Objekte), das in den Plot aufgenommen werden soll.
- Verwenden Sie das Modul Auslöser, um Auslöser zu konfigurieren und zwischen ihnen umzuschalten.
- Klicken Sie **Record**, um eine Aufnahme zu starten und alle abgefragten Daten in einer Datei zu speichern.
- Um eine bestehende Aufnahme zu laden, verwenden Sie die Schaltfläche **Open** und wählen Sie eine **.csv**-Datei. Sie können Auslöser festlegen, um die aufgezeichneten Daten zu filtern.
- Um wieder zurück zur Live-Protokollierung von Scope und Daten zu gelangen, verwenden Sie die Schaltfläche **Live Mode**.



HINWEIS

Vergewissern Sie sich beim Öffnen einer **.csv**-Aufnahmedatei, dass die passende **.xml**-Datei im gleichen Ordner vorhanden ist.

6 Kickstart Module

6.5.2 Konfiguration

Nachdem Sie ein **Datenobjekt** aus dem [Object Editor](#) in die Konfigurationstabelle gezogen haben, können Sie die folgenden Felder verwenden, um die Datenverarbeitung innerhalb des Moduls **Scope** zu konfigurieren.

Sampling and Rate

Dies legt fest, wie die Werte für Datenprotokollierung und Scope erhalten werden.

Sampling	Beschreibung
Off	Die Probennahme ist deaktiviert oder inaktiv.
Software	Kickstart sendet SDO-Upload-Anfragen an den Knoten in einem festen Zeitintervall von Rate Millisekunden.
TPDO<no>.<byteStart>.<size>	Das Objekt ist über Sende-PDO <no> (1..4) verfügbar. Innerhalb des PDO-Rahmens beginnen die erforderlichen Daten bei Byte-Position <byteStart> (0..7) und sind <size> (1, 2 oder 4) Bytes groß. Beispiel: TPDO3.0.2 (Der Wert ist ein 16-Bit-Wert aus den ersten beiden Bytes von Sende-PDO 3.)
RPDO<no>.<byteStart>.<size>	Wie TPDO, aber für Empfangs-PDO



HINWEIS

Die Methoden TPDO und RPDO sind passive Probenahmemethoden. Sie führen nicht die eigentliche Geräte-PDO-Konfiguration (PDO-Mapping) durch und ändern nicht den Zustand des CAN-Knotens per NMT. Sie können sicher innerhalb einer laufenden CANopen-Anwendung verwendet werden.

Scaling

Erlaubt die Skalierung und Verarbeitung der Rohdatenwerte. Sie können einen einfachen Faktor oder auch komplexere Formeln verwenden, einschließlich Offsets oder gar nicht-linearer Transformationen. Beispiele:

Beispiele:

Scaling	Beschreibung
0.25	Skalieren Sie den Wert um 0,25 (d. h. der angezeigte Wert ist viermal kleiner als der Rohdatenwert)..
100	Skalieren Sie den Wert um 100 (d. h. der angezeigte Wert wird mit 100 multipliziert).
$= x / 65536 + 10$	Dividieren Sie Rohdatenwerte um 65536 und addieren Sie anschließend 10.
$= 1 / x$	Nicht-lineare Ausdrücke wie dieser sind ebenfalls möglich. Achten Sie jedoch auf ungültige Ergebnisse, z. B. „Division durch Null“. Kickstart zeigt diese Werte als „0“ an.



HINWEIS

Die Skalierung wird nur auf den angezeigten Wert angewendet, nicht auf die aufgezeichneten Rohdaten. Das heißt, dass Sie zu einem späteren Zeitpunkt (beim Laden der Daten mit Open) einen anderen Skalierungsausdruck anwenden oder spontan alle Daten skalieren können.

6 Kickstart Module

6.5.3 Aufnahme von Daten

Verwenden Sie in der Symbolleiste die Schaltfläche **Record**, die Aufnahme aller überwachten Daten in eine **.csv**-Datei zu starten oder zu anzuhalten.

Damit der Dateiname das Datum und die Uhrzeit des Aufnahmebeginns enthält, verwenden Sie den Platzhalter **DATETIME**, der später durch den richtigen Zeitstempel ersetzt wird. Zum Beispiel: Der Dateiname `Messungen_DATETIME.csv` wird als `Messungen_2012-02-16T15-01-32.csv` gespeichert.

Um die aufgezeichneten Daten anzuzeigen, verwenden Sie **Open**.



HINWEIS

Es werden nur diejenigen Datenobjekte aufgezeichnet, die für die Probenahme in der Konfigurationstabelle des Moduls Scope konfiguriert wurden. Wenn Auslöser aktiv sind, werden nur die Daten innerhalb des Auslöserereignis-Fensters aufgezeichnet.



HINWEIS

Die **.csv**-Datei der aufgezeichneten Daten wird von einer **.xml**-Datei begleitet, die Informationen zu den aufgezeichneten Datenobjekten enthält. Stellen Sie beim Kopieren der **.csv**-Datei an einen anderen Ort sicher, dass die **.xml**-Datei enthalten ist, damit die Daten auch angesehen werden können.

6.5.4 Plot

Ein **Plot**-Panel zeigt Daten mit begrenztem Gültigkeitsbereich in einem skalierbaren Plot. Mehrere Datenobjekte können durch mehrere Kurven im gleichen Plotfenster dargestellt werden.

Sie können die Skala **Value / Div** (Y-Achse) und die Skala **Time / Div** (X-Achse) verändern, um die Ansicht zu vergrößern oder zu verkleinern.

Standardmäßig zeigt der Plot die aktuellsten verfügbaren Daten. Um frühere Daten anzuzeigen, verwenden Sie die Bildlaufleiste unter dem Plot.



HINWEIS

Alle erstellten Plot-Panels werden innerhalb der Kickstart-Projektdatei gespeichert. Beim erneuten Öffnen des Projekts wird das Panel automatisch wiederhergestellt, aber es werden nur Daten angezeigt, wenn Sie das **Scope** starten oder eine passende Aufnahme laden.

☐ Einen Wert zu einer bestimmten Zeit lesen

Der Wert-Cursor ist eine schmale schwarze vertikale Linie mit einem Zeitstempel. Für jedes Datenobjekt zeigen eine Farbmarkierung und eine schwarze Beschriftung den aktuellen Wert zu diesem bestimmten Zeitpunkt. Sie können den Cursor an eine andere Position verschieben, indem Sie in das Plot-Fenster klicken. Sie können auch den Zeitbereich (und die durch den Cursor markierte Zeit) in kleinen Schritten verschieben, indem Sie die Schaltflächen **<- |** und **| ->** neben der Bildlaufleiste verwenden.

☐ Konfigurieren der Wert / Div-Skala

Das Kombinationsfeld Value / Div konfiguriert den angezeigten Bereich auf der Skala des Wertes, die Y-Achse des Plots. Wählen Sie eine der verfügbaren Voreinstellungen. Es ist auch möglich, eine eigene Skalenkonfiguration einzugeben.

Mögliche Konfigurationsmodi:

Auto – Die Skala wechselt automatisch, um die aktuelle Wertauswahl anzuzeigen.

Min / Max – Die Skala wird durch die eingegebenen Min./Max.-Grenzen definiert.

Units / Div + Offset – Die Skala beginnt beim eingegebenen Offset. Der Wert Einheiten / Div definiert Anzahl von Einheiten zwischen jeder Division, also jeder kleinen Markierung auf der Skala. Die kleinen Markierungen sind diejenigen ohne Wert daneben.

- Es gibt zwei mögliche Formate für die Konfiguration: Standard und Kurzschrift. Während der Eingabe zeigt ein Etikett direkt neben der Eingabesteuerung die normierte (Standard-)Version Ihrer Skala. Wenn Ihre aktuelle Eingabe ungültig ist, wird eine Meldung angezeigt.

6 Kickstart Module

Beispiele für die Wert / Div-Skalae

Scaling Methode	Normalisiertes (standardmäßiges) Format	Kurzschrift-format	Beschreibung
Auto	automatic	a	Die Skala wird, basierend auf den aktuell angezeigten Werten, automatisch eingestellt.
Min/Max	min=0 max=100	0m100	Die angezeigte Skala liegt zwischen 0 und 100
	min=0.25 max=0.75	.25max0.75	Die angezeigte Skala liegt zwischen 0.25 and 0.75
Units/Div + Offset	100/div offset=0	100	Die angezeigte Skala beginnt bei Offset 0 mit 100 Einheiten zwischen den einzelnen Divisionen.
	0.1/div offset=-20	0.1o-20	Der angezeigte Bereich beginnt bei Offset -20 mit 0,1 Einheiten zwischen den einzelnen Divisionen.

6.5.5 Trigger

Ein  **Trigger** ist auf ein Datenobjekt definiert und kann auf eine bestimmte Bedingung eingestellt werden. Sobald die Bedingung erfüllt ist, werden die Daten innerhalb eines Zeitfensters (Auslöserereignis-Fenster genannt) weitergeleitet. Wenn keine Auslöser definiert sind, werden alle Daten direkt weitergeleitet, sobald sie überwacht worden sind.

Trigger Event Window

Zeitraum vor und nach dem Starten eines Auslöserereignisses. Wenn Auslöser definiert wurden, werden nur Daten innerhalb des Ereignisfensters überwacht. Alle Daten, die keinen Auslöser starten oder sich außerhalb eines Auslöserereignis-Fensters befinden, werden ignoriert: Sie erscheinen nicht in Plots und werden bei der Aufnahme nicht in einer Datei gespeichert.

Trigger Konfiguration

Use the **Pre-Trigger Time** and **Post-Trigger Time** fields to define the size of the trigger event window.

Verwenden Sie die Felder **Pre-Trigger Time** und **Post-Trigger Time**, um die Größe des Auslöserereignis-Fensters zu definieren.

Mit der Schaltfläche **Trigger On / Off** wechseln Sie zwischen ausgelöstem und nicht ausgelöstem Modus.

Geben Sie die Auslöserdefinitionen im unten beschriebenen Format in die Definitionstabelle ein. Eine Validierungsanzeige neben den Daten zeigt, ob der eingegebene Auslöser gültig ist. Um einen Auslöser aus der Tabelle zu entfernen, geben Sie eine leere Zeichenfolge ein.

Trigger Definition Format:

(data_object_id) – Auslöser bei jeder Veränderung des Datenobjekt-Wertes

(data_object_id) < | > (trigger level) – Auslöser bei steigender (>) oder fallender (<) Neigung

Examples

002.6041h.00h – Auslöser bei jeder Änderung im Wert des angegebenen Datenobjekts

001.60ffh.00h < 200 – Auslöser, wenn der Wert des angegebenen Datenobjekts unter 200 fällt

001.58ffh.00h > 100 – Auslöser, wenn der Wert des angegebenen Datenobjekts über 200 steigt



Auch bei der Verwendung eines Skalierungsfaktors arbeitet der Auslöser immer mit dem Rohdatenwert, nicht mit den skalierten Ergebnissen.

6 Kickstart Module



HINWEIS

Sie können Auslöser als Nachbearbeitungs- und Datenanalyse-Tool für erfasste Daten verwenden. Richten Sie die Auslöser ein und aktivieren Sie sie, bevor Sie die **.csv**-Datei öffnen. Nach dem Ändern der Auslöseereinstellungen müssen die Aufzeichnungsdaten mithilfe der Schaltfläche **Reload** neu geladen werden.

6.6 UI-Panel

Je nach Vorlage (Template) sind unterschiedliche Module / Modulnamen vorhanden, d.h. die Module haben nicht immer alle hier beschriebenen Funktionen.

Das  **UI-Panel**-Modul kann unter Verwendung der [Qt Quick/QML](#)-Technologie Datenobjekte in konfigurierbaren dynamischen Benutzeroberflächen anzeigen.

Sie können personalisierte Schnittstellen erstellen, die sich für verschiedene Benutzer und Szenarien eignen. Datenobjekte können durch grafische Steuerelemente (Drehzscheibe, Schieberegler, Optionsfeld) in einem Arbeitsbereich dargestellt werden. Sie können das Aussehen und Verhalten jedes Elements verändern und eine Smartphone-ähnliche Benutzeroberfläche für Ihre individuelle Anwendung erstellen.



HINWEIS

Die Anweisungen im Bereich [Erste Schritte](#) zeigen, wie man ein Panel aus einem grundlegenden Satz von vordefinierten UI-Steuerelementen zusammenstellen kann, die jeweils über ein einziges [Object Editor](#) und SDO-Kommunikationsanforderungen kommunizieren, ähnlich den Lese- und Schreibfunktionen im Objekteditor. Für erweiterte Benutzeroberflächen mit Grafiken und Animationen kontaktieren Sie bitte Ihren Anwendungssupport.

6.6.1 UI-Panel – Erste Schritte

- Die Schaltflächen **Start/Stop** steuern, ob die Schnittstelle aktiv ist, d. h. ob sie mit den tatsächlichen Netzknoten (z. B. CANopen-Geräten) kommuniziert und die angegebenen Datenobjekte liest oder schreibt.
- Beginnen Sie die Planung oder Änderung Ihrer Benutzeroberfläche, indem Sie in der Symbolleiste die Schaltfläche  **Design Mode** klicken.



HINWEIS

Benutzerebene 2 ist für den Zugriff auf **Design Mode** erforderlich.

- Bearbeiten Sie im [Object Editor](#) die Spalte **UI Format** der Parameter (Datenobjekte), die Sie für das UI-Panel verwenden möchten, um deren Aussehen und Verhalten zu definieren. Siehe die UI-Format-Definition für Syntax und Beispiele.
- Nachdem der Wert für **UI Format** gesetzt ist, erstellen Sie Ihre visuelle Kontrolle, indem Sie die Datenobjektzeile aus der Tabelle [Object Editor](#) auf den weißen Arbeitsbereich des **UI-Panels** ziehen.
-  **Show Data Objects** zeigt eine Tabelle mit den aktuellen Datenobjekten und der Konfiguration ihrer Steuerelemente. Um das Verhalten eines erstellten Steuerelements zu ändern, bearbeiten Sie die Spalte **UI-Format** in der Objekttable. Die Benutzeroberfläche wird dann mit den neuesten Änderungen aktualisiert. Aussehen und Positionen werden durch die Spalte **Instances** definiert..
- Verwenden Sie die Spalte **Scaling**, um die Rohdaten-Objektwerte zu skalieren. Dies geschieht in der gleichen Weise, wie Scaling in den Modulen [Object Editor](#) und **Scope** verwendet wird. Möglicherweise müssen Sie das Verhalten Ihrer UI-Steuerelemente auf Basis der Skalierung anpassen. Stellen Sie beispielsweise den Bereich für einen Schieberegler so ein, dass er nach der Skalierung die möglichen Minimal- und Maximalwerte umfasst, oder passen Sie die Schrittgröße für einen Drehregler an.
- Um mehrere Instanzen des gleichen Datenobjekts zu erstellen, ziehen Sie das Datenobjekt wieder von der Tabelle in den Arbeitsbereich.

6 Kickstart Module

6.6.2 UI Format-Definition



HINWEIS

Benutzerebene 2 ist erforderlich für den Zugriff auf die **UI Format**-Definitionen im [Object Editor](#).

Die Spalte **UI Format** der Datenobjekte definiert das Aussehen und Verhalten des erstellten Steuerelements.

Sein Wert ist eine Liste von Schlüsselwert-Attributpaaren, getrennt durch Semikolons („;“), die das Aussehen und Verhalten des Steuerelements beschreiben. Nachfolgend sehen Sie eine Beschreibung aller verfügbaren Attribute:

shape

Definiert die Art der Steuerung, die erstellt wird, wenn das **Datenobjekt** in den UI-Panel-Arbeitsbereich gelegt wird.

Verfügbare Standardformen:

shape:	Beschreibung
label	einfaches Steuerelement, um einen schreibgeschützten Wert anzuzeigen
text	Textfeld-Steuerelement, um einen Wert anzuzeigen oder zu bearbeiten
slider	vertikaler Schieberegler, um einen Wert innerhalb eines bestimmten Bereichs anzuzeigen oder zu ändern
dial	Drehscheibe, um einen Wert innerhalb eines bestimmten Bereichs anzuzeigen oder zu ändern
radio	eine Liste von Optionen wie beispielsweise Optionsfelder, von denen jedes einen bestimmten Daten-Objektwert darstellt (z. B. „3“) und eine passende Beschreibung (z. B. „Geschwindigkeitsmodus“). Siehe die Beschreibung für options für weitere Einzelheiten.
bitfield	eine Liste von Bit-Signalen. Jedes Signal entspricht einem Bit des Ganzzahlwertes des Steuerelements. Siehe die Beschreibungen für options und fieldsize für weitere Einzelheiten.



HINWEIS

Jede shape entspricht einer **.qml**-Datei aus dem **/QML**-Ordner Ihres Kickstart-Projekts, z. B. KickLabel.qml, KickTextInput.qml...

Sie können Ihre eigenen QML/Qt Quick-Elemente entwerfen und verwenden, indem Sie beispielsweise **shape:MyOwnQmlControl.qml** definieren, wobei MyOwnQmlControl.qml eine modifizierte Version ist von **Kick... .qml**-Dateien.



HINWEIS

Sobald ein Steuerelement im Arbeitsbereich erschaffen worden ist, z. B. durch Drag-and-Drop, kann seine Art nicht mehr durch Abwandeln der **shape**-Definition geändert werden. Die **shape**-Definition gilt nur für die Schaffung neuer Elemente im Arbeitsbereich.

Um den Typ eines vorhandenen Steuerelements im Arbeitsbereich zu ändern, müssen Sie den Text in der Spalte **Instances** der Tabelle **Data Objects** ändern.

updates

Legt fest, wie das UI-Steuerelement mit dem aktuellen Datenobjekt kommuniziert. Sein Wert gibt an, ob der Wert des Datenobjekts vom Benutzer geändert werden kann, und auch, wie der Wert aus dem Gerät ausgelesen wird.

6 Kickstart Module

Beispiele:

updates:	Description
<code>r</code>	Der Wert wird nur einmal bei der Initialisierung vom Gerät gelesen.
<code>r100</code>	Der Wert wird periodisch vom Gerät gelesen, in diesem Fall alle 1000 Millisekunden.
<code>r,w</code>	der Anfangswert wird von dem Gerät bei der Initialisierung gelesen und kann anschließend durch den Benutzer geändert werden. Vom Benutzer geänderte Werte werden an das Gerät gesendet.



HINWEIS

Ein vom Benutzer änderbares Steuerelement kann nicht mit periodischem Auslesen kombiniert werden. Beispielsweise ist `updates:r100,w` keine gültige Definition.

range

Definiert den zulässigen Bereich für Eingabesteuerungen. Er wird nur von den Schiebe- und Drehreglern verwendet.

Beispiele:

`range:0,100`

`range:-100,100`



HINWEIS

Wenn Sie ein Objekt aus **Object Dictionary (Device Description)** im Modul [Object Editor](#) ziehen, wird der Bereich aus der ursprünglichen `.xdd / .xdc`-Gerätebeschreibungsdatei berechnet, falls verfügbar. Wenn in der Gerätebeschreibungsdatei kein Wertebereich definiert wurde, wird ein standardmäßiger **Bereichswert** gewählt.

step

Definiert eine gültige positive Schrittgröße für einige der bereichsbasierten Steuerungen, z. B. den Schieberegler oder die Drehscheibe. Der standardmäßige Schrittgröße ist 1. Ändern Sie den Wert, um das Steuerelement so zu konfigurieren, dass es ausgewählte Werte zur angegebenen Schrittgröße rundet.

Beispiele:

`step:1` – the default behavior.

`step:10` – only values that are multiples of 10 will be available.

fieldsize

Wird nur von **bitfield**-Formen verwendet. Definiert die Anzahl der Bit-Signale, die für diesen Ganzzahlwert verwendet werden.

Beispiel:

`fieldsize:8`

Dieser Wert wird als acht einzelne Bit-Signale angezeigt mit einem laufenden numerischen Wert (Bit-Index) zwischen 0 und 7.

Wenn **fieldsize** Null oder nicht definiert ist, zeigt die **bitfield**-Form zeigt nur die Bit-Indizes an, die in **options: definiert sind** (siehe unten).

options

Wird von den Formen **radio** und **bitfield** verwendet. Es spezifiziert eine Liste verfügbarer Optionen und ihrer entsprechenden Werte (bezügl. „Bit-Indizes“ für **bitfield**-Formen).

Die Optionen sind wählbar / anklickbar, wenn die **updates**-Eigenschaft „schreiben“ angibt (`updates:r,w`), und nicht nur „schreibgeschützt“ (`updates:r1000`).

6 Kickstart Module

Format:

options:<value_1>,<text _1>|<value_2>,<text_2> |... (etc.)

Beispiel für ein Auswahl-Steurelement für die Wahl zwischen 4 verschiedenen Werten (0, 1, 3 oder 6):

shape:radio; options:0,Reset|1,Profile Position|3,Velocity Mode|6,Homing Mode

Beispiel für ein Bit-Feld, das 3 digitale Eingangssignale repräsentiert:

shape:bitfield; options:0,Limit Switch Left|1,Limit Switch Right|2,Start

Example UI Format Definitions

UI Format	Beschreibung
shape:label; updates:r	Ein schreibgeschütztes Etikett. Der Wert des Steuerelements wird vom Gerät nur bei der Initialisierung gelesen.
shape:label; updates:r200	Ein schreibgeschütztes Etikett. Der Wert des Steuerelements wird bei der Initialisierung vom Gerät gelesen und anschließend, in regelmäßigen Abständen, alle 200 Millisekunden.
shape:text; updates:r,w	Ein vom Benutzer veränderbares Textfeld, der Wert wird bei der Initialisierung vom Gerät gelesen. Sobald der Benutzer den Wert bearbeitet, wird er auf das Gerät geschrieben.
shape:slider; range:0,100; updates:r,w	Ein vom Benutzer veränderbarer Schieberegler mit einem Bereich zwischen 0 und 100 und Schrittweite 1.
shape:dial; range:-200,200; step:10; updates:r,w	Eine vom Benutzer veränderbare Drehscheibe mit einem Bereich zwischen -200 und 200, wobei alle wählbare Werte ein Vielfaches von 10 sind.
shape:radio; options:0,Off 1,On; updates:r,w	Zwei frei wählbare Optionsfelder: Aus und Ein, entsprechend den Werten 0 und 1.
shape:bitfield; fieldsize:8; updates:rw	Eine Liste von 8 digitalen Ausgangsbits, die durch Anklicken umgeschaltet werden können. Für eine Datenobjekt, das den Namen „Digitalausgang“ hat, wäre die Bezeichnung der Elemente: Digitalausgang 0 Digitalausgang 1 ... Digitalausgang 7
shape:bitfield; options:2,2 - Limit Switch Left 5,5 - Limit Switch Right 7,7 - Start; updates:r1000;	Eine Liste von Eingangsbits, wobei nur drei Bit-Indizes (2, 5 und 7) gezeigt/verwendet werden: 2 – Begrenzungsschalter links 5 – Begrenzungsschalter rechts 7 – Start

6 Kickstart Module

6.6.3 Knoten-IDs und Panels mit mehreren Knoten

Die grundlegenden UI-Panel-Beispiele sind auf Knoten-ID = 001 eingestellt. Wenn Sie also **Datenobjekte** aus dem [Object Editor](#) ziehen, wie zuvor beschrieben, verwenden diese die Node ID aus den [Object Editor](#)-Einstellungen.

Mit  **Design Mode** und mit der Option  **Show Data Objects** aktiviert können Sie Knoten-IDs ändern oder eine Reihe von UI-Steuer-elementen für eine andere Knoten-ID oder ein zweites CANopen-Gerät duplizieren. Verwenden Sie die folgenden Funktionen, die in der Symbolleiste zur Verfügung stehen:

Clone

Verwenden Sie nach der Auswahl einer oder mehrerer Zeilen in der Tabelle **Data Objects** die Option **Clone**, um diese Elemente zu duplizieren. Fügen Sie sie anschließend in den Arbeitsbereich UI-Panel ein unter Verwendung einer anderen **Node ID**.



Die geklonten Elemente erhalten den gleichen **Name**-Text wie die ursprünglichen Elemente. Sie können den Text der Spalte **Name** in der Tabelle **Data Objects** bearbeiten, sodass Ihre UI-Elemente anzeigen, zu welcher Knoten-ID sie gehören. Sie können auch einen anwendungsspezifischen Namen verwenden, wie beispielsweise „Druckformat-Einstellung 1“.

Change ID

Wählen Sie eine oder mehrere Zeilen in der Tabelle **Data Objects** und verwenden Sie anschließend **Change ID**, um die **Node ID** auf einen anderen Wert zu ändern.

6.6.4 Projekt and UI-Dateien / Erweiterte Benutzeroberflächen

Nach dem Zusammenstellen Ihres UI-Panels können Sie die erschaffene Benutzeroberfläche mithilfe der Funktion  **Save Project** aus der Kickstart-Hauptsymbolleiste speichern.

Wenn Sie Ihr UI-Panel ausgehend von einer der Kickstart-Vorlagen erstellt haben, ist die gespeicherte Datei standardmäßig eine komprimierte Projektdatei (**.kickzip**).

Für viele komplexere Anwendungen ist es notwendig, direkt mit den unkomprimierten Dateien im **.kickzip**-Dateiarchiv zu arbeiten. Gehen Sie dafür folgendermaßen vor:

- Speichern Sie das Projekt über das Menü **File >  Save As...** und wählen Sie **Kickstart Uncompressed Project (.kickpro)**, nicht das standardmäßige **.kickzip**-Dateiformat.
- Jetzt können Sie die ursprüngliche **.kickpro**-Datei ggf. mit einem standardmäßigen Text- oder XML-Editor manuell bearbeiten.
- Das **/QML**-Unterverzeichnis enthält alle [Qt Quick/QML](#)-Dateien, die Ihre UI-Panel-Anwendung definieren. Sie können standardmäßige Texteditoren oder die [Qt Quick](#)-Entwicklungswerkzeuge verwenden, um mit diesen Dateien zu arbeiten, z. B. um modifizierte Versionen mit einem anderen Aussehen oder anderen Funktionen zu erstellen.



Zusätzliche Leitfäden für Entwickler über [Qt Quick/QML](#) in Kickstart sind auf Anfrage erhältlich. Bitte kontaktieren Sie Ihren Anwendungssupport.

ebm-papst
St. Georgen GmbH & Co. KG

Hermann-Papst-Straße 1
78112 St. Georgen
Germany
Phone +49 7724 81-0
Fax +49 7724 81-1309
info2@de.ebmpapst.com

ebmpapst

Die Wahl der Ingenieure