





## << emotas - your embedded communication partner <<

**In den Bereichen Automatisierungstechnik und LEV-Entwicklung ist emotas einer der führenden Anbieter von Kommunikationstechnologien.**

Thematische Schwerpunkte des Produktportfolios sind CAN, CANopen (FD), J1939 (FD) und EnergyBus. Neben eigenen Produkten entwickelt emotas auch individuell maßgeschneiderte Embedded-Software-Lösungen im Kundenauftrag. Das Leistungsspektrum reicht dabei von Beratung, über Entwicklung und Inbetriebnahme vor Ort bis hin zur Wartung der Software. Außerdem bietet emotas Schulungen und Consulting zu den Kommunikationsprotokollen CANopen, CANopen FD, EnergyBus, J1939 (FD) und zu individuellen Themen an.

### << Technologien <<

emotas ist Mitglied des CAN in Automation e.V. (CiA) und Entwicklungspartner des EnergyBus e.V.. Wir arbeiten in verschiedenen Arbeitsgruppen und in den Gremien zur Standardisierung von CiA und EnergyBus e.V. mit. Die emotas Produkte entsprechen somit stets dem neuesten Stand der technischen Entwicklung.

**CANopen**

**CANopen<sup>FD</sup>**

**J1939**

**EnergyBus**

**UDS**

**ISO-TP**

### emotas steht für

- // Kompetente Beratung
- // Exzellenter Service
- // Hoher Qualitätsstandard
- // Made in Germany

### << Branchen <<

emotas-Produkte kommen in verschiedensten Branchen zum Einsatz:

- Automatisierungstechnik
- Medizintechnik
- LEV-Komponenten
- Batterie und Ladetechnik
- Aufzüge *und weitere...*

### << Partnerschaft <<

Die Partnerschaft mit unseren Kunden ist uns ein wichtiges Anliegen. Wir wissen: Die Zukunft des Unternehmens können wir nur gemeinsam mit unseren Kunden erfolgreich gestalten. Mit starkem Service, kompetenter Beratung und hochwertigen Produkten bietet emotas seinen Kunden maximalen Nutzen.

### << Service <<

Besten Service mit direktem Telefonkontakt zu den Entwicklern des Produkts oder Projekts – jederzeit.

## << Inhaltsverzeichnis <<

### CAN

|                         |   |
|-------------------------|---|
| // RawCAN Stack         | 6 |
| // ISO-TP Client Server | 6 |

6

### CANopen (FD)

7

|                                 |    |
|---------------------------------|----|
| // CANopen (FD) Einführung      | 7  |
| // CANopen Protokoll Stack      | 8  |
| // CANopen FD Protokoll Stack   | 10 |
| // CANopen DeviceDesigner       | 12 |
| // CANopen NetworkDesigner      | 14 |
| // CANopen DeviceExplorer       | 16 |
| // CANopen Process DataLinker   | 18 |
| // CANopen Bootloader           | 20 |
| // CAN UpdateManager            | 21 |
| // CANinterpreter               | 22 |
| // CANopen Gateway CiA309       | 24 |
| // CAN Multiprotokoll Stack     | 25 |
| // CANopen Schulungen - Termine | 27 |
| // CAN Hardware                 | 28 |

### J1939 (FD)

|                             |    |
|-----------------------------|----|
| // J1939 Protokoll Stack    | 30 |
| // J1939 FD Protokoll Stack | 31 |
| // J1939 DeviceDesigner     | 32 |
| // J1939 CANinterpreter     | 33 |
| // J1939 Bootloader         | 34 |

30

## UDS

**35**

|                                 |    |
|---------------------------------|----|
| // UDS Server Stack-Erweiterung | 35 |
| // UDS Client Stack-Erweiterung | 35 |
| // UDS Bootloader               | 36 |

## EnergyBus

**37**

|                           |    |
|---------------------------|----|
| // EnergyBus Einführung   | 37 |
| // EnergyBus Framework    | 38 |
| // EnergyBus DeviceWizard | 40 |
| // EnergyBus Tool         | 42 |

## Dienstleistungen

**43**

|                                  |    |
|----------------------------------|----|
| // Embedded Software Entwicklung | 43 |
| // CANopen Integration           | 44 |
| // EnergyBus Integration         | 45 |
| // Inbetriebnahme-Unterstützung  | 46 |
| // Schulungen                    | 47 |
| // 10 Gründe für emotas          | 48 |
| // Partner                       | 49 |
| // Über emotas                   | 50 |

## << RawCAN Stack <<

**Der RawCAN Stack ist ein leistungsfähiger CAN-Stack im Quellcode. Er wurde in ANSI-C erstellt und ist MISRA-C konform. Mithilfe der CAN Software können einfach und schnell CAN-Geräte entwickelt werden oder Geräte um CAN erweitert werden. Größere Datenpakete können mithilfe der integrierten ISO-TP-Funktionalität übertragen werden**

Verglichen mit den einfachen CAN-Routinen der Mikrocontrollerhersteller bietet unserer RawCAN Stack und der eigens von uns entwickelten CAN-Treiber Verlässlichkeit und zusätzliche Funktionen wie beispielsweise:

- Zyklischen Senden von CAN-Telgrammen
- Empfangsüberwachung zyklische CAN-Telegramme
- Inhibitzeiten für ereignisgesteuerte Nachrichten
- Flexible Sende-Queue mit zeitoptimiertem Senden aus dem Interrupt
- Empfangsqueue variabler Länge

### << RawCAN Stack Highlights <<

- ANSI-C kompatibler CAN Source Code Stack
- MISRA-C-konform
- Zeitsparendes Entwicklungstool  
CAN DeviceDesigner inklusive
- verfügbar für viele CAN-Controller und CPU-Typen
- komfortables User-Interface
- in weiten Teilen konfigurier- und skalierbar

Eine flexible Benutzerschnittstelle erlaubt die mühelose Auswertung von empfangenen Daten und die Nutzung von CAN-Diensten im Netzwerk.

Für die flexible Anbindung an verschiedene CAN-Controller und CPU-Typen ist der CANopen Stack mit einem definierten Treiberinterface ausgestattet. Damit ist neben der einfachen Möglichkeit der Anpassung an vorhandene CPU Typen wie u.a. ARM- und Cortex-Varianten, auch der unkomplizierte Austausch der Hardwareplattform möglich. Die Verwendung verschiedener Echtzeitbetriebssysteme (RTOS) ist möglich.

Für möglichst geringen Ressourcenverbrauch ist der RawCAN Stack in weiten Teilen konfigurier- und skalierbar. Die Einstellungen für diese Features werden mit dem grafischen Konfigurationstool CAN DeviceDesigner vorgenommen. Das Tool ermöglicht die Erstellung der CAN-Signalkonfiguration.

### << ISO-TP <<

Sowohl ISO-TP-Server und ISO-TP-Client-Funktionen sind im RawCAN Stack integriert. Mehrere Instanzen können parallel zur Übertragung größerer Datenpakete von bis zu 4 KB genutzt werden.

#### # Anforderungen/Voraussetzung

- ▶ ANSI-C Compiler

#### # Lieferumfang/Lizenzierung

- ▶ CAN-Stack Source Code (ANSI-C kompatibel)
- ▶ sofort lauffähige Beispielapplikationen
- ▶ Anwenderhandbuch und Referenzhandbuch(Doxygen)
- ▶ Standortlizenz ohne RuntimeGebühren (Royalty free; inkl. 24 Monate Telefon und E-MailSupport)
- ▶ Projektlizenz (Royalty free; inkl. 6 Monate Telefon und E-MailSupport)
- ▶ Named-User-Lizenz (Projekt) oder Floating Lizenz mit Dongle des CAN DeviceDesigner
- ▶ optionaler Wartungsvertrag
- ▶ optionaler Integrationssupport

#### # Bestellinformation

- ▶ 5042-10 | RawCAN Pack
- ▶ 5042-10-ML | RawCAN Pack Multiline

## << CANopen (FD) - Einführung <<

**CANopen (FD) - das industrielle Kommunikationsprotokoll basierend auf CAN, hat seinen Ursprung in der Automatisierungstechnik. CANopen ist international standardisiert (EN 50325-4) und hat viele Anwendungsfelder erschlossen. Maschinensteuerungen, medizinische Geräte, Schienen- und Offroad-Fahrzeuge, Elektroräder, Gebäudeautomation sowie konventionelle und erneuerbare Energieversorgung gehören dazu.**

### << Überblick <<

CANopen (FD) bietet sehr flexible Kommunikationsmechanismen, welche im CANopen-Kommunikationsprofil CiA-301 (CiA-1301) definiert sind. Dieses Kommunikationsprofil sowie weitere Applikations-, Interface- und Geräteprofile werden vom CAN in Automation e.V. und seinen Mitgliedern entwickelt und gepflegt. Dabei fließen stets aktuelle Anforderungen ein. emotas ist aktiv an der Weiterentwicklung des Kommunikationsprofils und weiterer Profile beteiligt. Mit den dabei gewonnenen Erfahrungen entwickeln wir unsere CANopen Produkte kontinuierlich weiter und passen die Produkte an Kundenwünsche an.



### << CANopen DeviceExplorer <<

Der CANopen DeviceExplorer ist das vielseitige Tool von emotas für Entwicklungs-, Test-, Diagnose und Serviceaufgaben. Dieses Tool stellt CANopen-Masterfunktionen bereit, die die Inspektion und Konfiguration von CANopen-Geräten erlauben. Informationen über das jeweilige CANopen-Gerät werden aus dem elektronischen Datenblatt des Geräts entnommen oder direkt aus dem Gerät gelesen. Mithilfe der standardisierten DCF-Dateien (Device configuration files) können

Gerätekonfigurationen gespeichert und geladen werden. Die Verwaltung ganzer CANopen-Netzwerke in Projektdateien ist möglich. Verschiedene PlugIns erweitern den Funktionsumfang.

### << CANopen Protokoll Stack <<

Der emotas CANopen (FD) Protokoll Stack ist eine Software-Bibliothek, die alle Kommunikationsdienste entsprechend des CANopen (FD) Kommunikationsprofils bereitstellt. Er ist in ANSI-C erstellt und MISRA-C konform. Mithilfe der CANopen (FD) Software können einfach und schnell CANopen (FD)-konforme Geräte entwickelt oder Geräte um CANopen (FD) erweitert werden. Der Protokollstack ist in den Ausbaustufen Slave, Master/Slave und Manager verfügbar.

### << CANopen DeviceDesigner <<

Der CANopen DeviceDesigner ist ein praktisches Tool zum schnellen und kostensparenden Entwurf von CANopen (FD)-Geräten. Mit wenigen Klicks fügt der Anwender das Objektverzeichnis des Geräts zusammen und spezifiziert die Eigenschaften der Objekte. Der CANopen DeviceDesigner erzeugt daraus das Objektverzeichnis und Initialisierungsfunktionen, das elektronische Datenblatt, sowie eine Gerätedokumentation. Zusätzlich nimmt der CANopen DeviceDesigner die optimale Konfiguration des CANopen Stack (FD) und CAN Treiber vor.

### << CANopen Training und Consulting <<

Für Anfänger wie Profis gleichermaßen geeignet sind die von emotas angebotenen Schulungen zu CAN und CANopen (FD) durch individuelle Absprache der Themen. Von CAN- und CANopen (FD)-Grundlagen über Geräte- und Applikationsprofile bis hin zu spezifischen Fragen zu Ihrer Anwendung sind viele Themen möglich. Die Schulungen richten sich an Entwicklungsingenieure und Entscheider, die sich in die Technologie einarbeiten möchten, eine Grundlage für die Arbeit mit unseren Produkten schaffen wollen oder die technischen Möglichkeiten von CANopen kennenlernen möchten. Zur Lösung komplexer Fragen und Problemstellungen bietet emotas Ihnen sein CANopen-Know-How in Form von individuellem Consulting an.

### << CANopen Inbetriebnahme/Integration <<

emotas übernimmt die Kommunikationsanbindung Ihrer Geräte und unterstützt Sie bei der Inbetriebnahme für einen schnellstmöglichen Start Ihres Projekts. Notwendig sind dafür nur Ihre Hardware, Ihre Entwicklungsumgebung, ein Entwickler und das langjährige Know-How unserer Mitarbeiter. Die Übernahme der Hardware erfolgt bei einem gemeinsamen Termin in Ihrem Hause oder an unserem Standort in Merseburg.

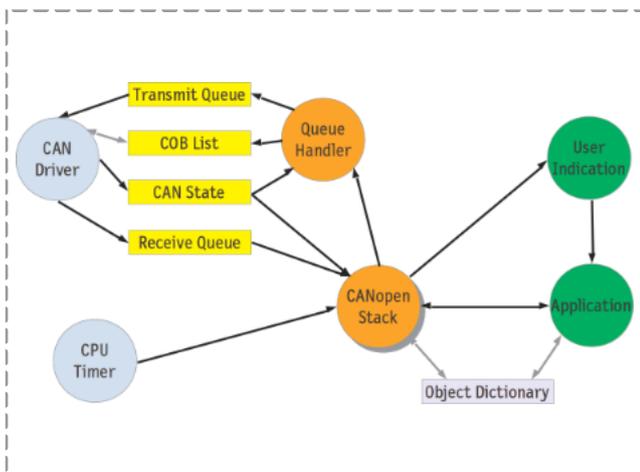
# << CANopen Protokoll Stack <<

**CANopen® Protokoll Stacks sind Software Bibliotheken, die alle Kommunikationsdienste entsprechend des CANopen Kommunikationsprofils CiA 301 in der aktuellen Version bereitstellen. Verschiedene Ausbaustufen sind erhältlich. Die CANopen Protokoll Stacks sind in ANSI-C erstellt und werden laufend mit dem aktuellen CANopen Conformance Test des CiA e.V. auf Konformität zum Standard geprüft. Der strukturierte Aufbau der Software unterstützt die schnelle und komfortable Entwicklung von CANopen- konformen Geräten**

Die flexible Benutzerschnittstelle erlaubt die einfache Auswertung empfangener Daten und die unkomplizierte Nutzung von CANopen Diensten im Netzwerk. Neben den Standard Diensten des CiA 301 und CiA 302 sind Erweiterungsmodul für verschiedene Kommunikations- und Applikationsprofile verfügbar.

Das definierte Treiberinterface gewährleistet hohe Flexibilität, weil die Adaption des CANopen Protokoll Stacks an vorhandene oder neue CAN-Controller oder CPU-Typen leicht möglich ist. Damit ist ein unkomplizierter Austausch der Hardwareplattform gewährleistet. Die Verwendung verschiedener Echtzeitbetriebsysteme (RTOS) sowie Linux (SocketCAN, can4linux), Windows und QNX ist möglich.

Um Speicher-Ressourcen zu schonen sind CANopen Protokoll Stacks von emotas konfigurier- und skalierbar. Die Einstellungen für diese Features werden mit dem grafischen Konfigurationstool CANopen DeviceDesigner vorgenommen. Das Tool ermöglicht auch die Erstellung des Objektverzeichnisses und die Erzeugung der EDS-Files, die somit immer konsistent sind. So können Änderungen schnell und problemlos eingepflegt werden.



## << CANopen Erweiterungsmodul <<

Für eine Vielzahl von CANopen Geräte- oder Applikationsprofilen stehen Erweiterungsmodul bereit. Aktuell sind Erweiterungen für folgende Profile bei emotas verfügbar:

- ▶ **CiA401** - Geräteprofil für digitale u. analoge I/O- Module
- ▶ **CiA402** - Zustandsmaschine für Antriebe
- ▶ **CiA404** - Geräteprofil für Sensoren und Regler
- ▶ **CiA406** - Geräteprofil für Encoder
- ▶ **CiA413** - Geräteprofil für Nutzfahrzeuge
- ▶ **CiA418** - Geräteprofil für Batterien
- ▶ **CiA419** - Geräteprofil für Ladegeräte
- ▶ **CiA437** - Applikationsprofil f. Photovoltaik-Komponenten
- ▶ **CiA443** - Applikationsprofil für SIIIS Level-2 Geräte
- ▶ **CiA447** - Applikationsprofil für Zubehörteile an Spezialfahrzeugen (Taxameter, Rundumleuchten)
- ▶ **CiA454** - Applikationsprofil für Energie Management Systeme u.a. in LEVs (EnergyBus)
- ▶ *und weitere...*

## Flying Master

Der Ausfall des aktuellen CANopen Master kann mit dem Flying Master Mechanismus erkannt und die Master Funktionalität automatisch auf ein anderes Gerät übertragen werden.

## Redundanz

Nutzung von 2 CAN-Kanälen zur Übertragung von Prozess- und Konfigurationsdaten. Beim Ausfall eines CAN-Kanals kann der zweite weiter genutzt werden. Diese Erweiterung wird häufig bei maritimen Applikationen eingesetzt.

## Safety

Bietet Unterstützung für die CANopen-Safety-Kommunikation mit SRDOs (Safety Relevant Data Objects). Damit können Applikationen nach SIL-3 erstellt werden.

## MultiLine

Ermöglicht die Verwendung des CANopen Stacks mit mehreren CAN-Schnittstellen. Dies ist insbesondere für Gateway-Applikationen interessant.

## << CANopen Dienstmerkmale <<

| Dienstmerkmal                     | Basic Slave | Master/Slave   | Manager        |
|-----------------------------------|-------------|----------------|----------------|
| SDO Server                        | 2           | 128            | 128            |
| SDO Client                        |             | 128            | 128            |
| SDO expedited/<br>segmented/block | ●/●/-       | ●/●/●          | ●/●/●          |
| PDO Producer                      | 32          | 512            | 512            |
| PDO Consumer                      | 32          | 512            | 512            |
| PDO Mapping                       | static      | static/dynamic | static/dynamic |
| MPDO Dest.Mode                    |             | ●              | ●              |
| MPDO Src. Mode                    |             | ●              | ●              |
| SYNC Producer                     |             | ●              | ●              |
| SYNC Consumer                     | ●           | ●              | ●              |
| Time Producer                     |             | ●              | ●              |
| Time Consumer                     |             | ●              | ●              |
| Emergency Producer                | ●           | ●              | ●              |
| Emergency Consumer                |             | 127            | 127            |
| Guarding Master                   |             |                | ●              |
| Guarding Slave                    | ●           | ●              | ●              |
| Bootup Handling                   |             | ●              | ●              |
| Heartbeat Producer                | ●           | ●              | ●              |
| Heartbeat Consumer                | 1           | 127            | 127            |
| NMT Master                        |             | ●              | ●              |
| NMT Slave                         | ●           | ●              | ●              |
| Master Bootup CiA302-2            |             |                | ●              |
| Flying Master CiA302-2            |             | ○              | ●              |
| Configuration Manager<br>CiA302-3 |             |                | ●              |
| SRD CiA302-5                      |             | ○              | ●              |
| Redundancy CiA302-6               |             | ○              | ○              |
| CANopen Router CiA302-7           |             | ○              | ○              |
| LED CiA3032                       | ●           | ●              | ●              |
| Safety                            | ○           | ○              | ○              |
| LSS CiA305                        | ●           | ●              | ●              |
| ISO-TP                            |             | ●              | ●              |
| Multiline                         |             | ○              | ○              |
| C#-API-Wrapper Windows            | ○           | ○              | ○              |

● inklusive ○ optional

## # Unterstützte Plattformen

Die emotas CANopen Protokoll Stacks können mit und ohne Betriebssystem verwendet werden. Sie unterstützen eine Vielzahl von CANController und CPU-Typen.

Darunter: Bosch, Infineon, Intel, NXP, ST Microelectronics, Texas Instruments, Atmel, Renesas, Microchip.

Die Liste der unterstützten Hardwareplattformen wird ständig erweitert:

<https://www.emotas.de/info/controller>

Sollte Ihre Plattform dennoch einmal nicht verfügbar sein, ist die Entwicklung weiterer Treiberpakete problemlos möglich.

## # Lieferumfang

- ▶ CANopen Protokoll Sourcecode in ANSI-C (MISRA-konform)
- ▶ CANopen Treiber
- ▶ CANopen DeviceDesigner (1 Floating Lizenz mit Dongle)
- ▶ sofort lauffähige Beispielapplikationen
- ▶ Benutzer und Referenzhandbuch
- ▶ optionaler Wartungsvertrag
- ▶ optionaler Integrationssupport

## # Lizenzierung

- ▶ Standortlizenz ohne Runtime-Gebühren (Royalty free; inkl. 24 Monate Telefon und E-Mail-Support)
- ▶ Projektlizenz (Royalty free; inkl. 6 Monate Telefon und E-Mail-Support)

## # Bestellinformation

- ▶ 5004-10 | CANopen Slave Pack
- ▶ 5004-11 | CANopen Master/Slave Pack
- ▶ 5004-12 | CANopen Manager Pack
- ▶ 2004-xx | CANopen Treiber
- ▶ 1004-4x | CANopen Profile
- ▶ 0140-10 | Integrationssupport

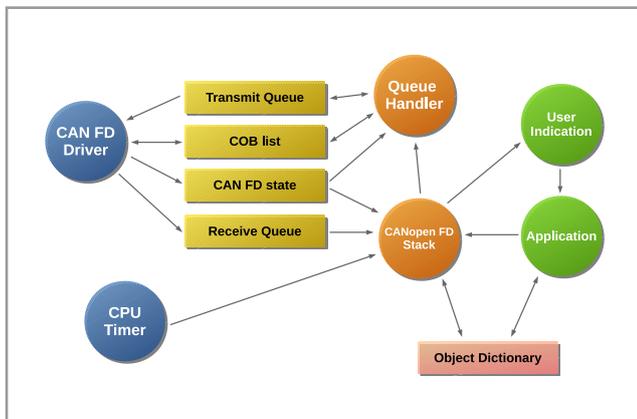
CANopen Beispiel-Implementierungen zum Download: [www.emotas.de/download](http://www.emotas.de/download)

## «« CANopen Protokoll Stack für CAN FD/CAN ««

**CANopen® FD Protokoll Stacks sind Software Bibliotheken, die alle Kommunikationsdienste entsprechend des CANopen Kommunikationsprofils CiA 1301 in der aktuellen Version bereitstellen. Verschiedene Ausbaustufen sind erhältlich. Die CANopen Protokoll Stacks sind in ANSI-C erstellt und werden laufend auf Konformität zum Standard geprüft. Der strukturierte Aufbau der Software unterstützt die schnelle und komfortable Entwicklung von CANopen- konformen Geräten.**

Die flexible Benutzerschnittstelle erlaubt die einfache Auswertung empfangener Daten und die unkomplizierte Nutzung von CANopen FD Diensten im Netzwerk. Neben den Standard-Diensten des CiA 1301 sind Erweiterungsmodule für verschiedene Kommunikations- und Applikationsprofile verfügbar.

Das definierte Treiberinterface gewährleistet hohe Flexibilität, weil die Adaption des CANopen FD Protokoll Stacks an vorhandene oder neue CAN-Controller oder CPU-Typen leicht möglich ist. Damit ist ein unkomplizierter Austausch der Hardwareplattform gewährleistet. Die Verwendung verschiedener Echtzeitbetriebssysteme (RTOS) sowie Linux (SocketCAN, can4linux), Windows und QNX ist möglich.



Um Speicher-Ressourcen zu schonen, sind die CANopen FD Protokoll Stacks von emotas konfigurierbar und skalierbar. Die Einstellungen für diese Features werden mit dem grafischen Konfigurationstool CANopen DeviceDesigner vorgenommen. Das Tool ermöglicht auch die Erstellung des Objektverzeichnisses und die Erzeugung der EDS-Files, die somit immer konsistent sind. So können Änderungen schnell und problemlos eingepflegt werden.

Die CANopen FD Protokoll Stacks stellen auch alle Dienste für Classical CANopen entsprechend CiA 301 zur Verfügung, so dass die erstellten Geräte zur Laufzeit wahlweise das Classical CANopen oder das CANopen FD Protokoll nutzen können.

Im Classical CANopen gelten dann natürlich die entsprechenden Einschränkungen wie 8 statt 64 Byte PDO Mapping Daten und die Nutzung von SDO anstatt dem USDO.

Alle Protokoll Stacks können auch als Mehrlinien Version für die Nutzung von mehreren CAN Schnittstellen bereitgestellt werden.

### «« CANopen Erweiterungsmodule ««

Für eine Vielzahl von Classical CANopen Geräte- oder Applikationsprofilen stehen Erweiterungsmodule bereit. Für CANopen FD ist die Spezifikation für diese Geräte- oder Applikationsprofilen noch nicht abgeschlossen. Sobald diese erfolgt ist, werden diese bereitgestellt.

Folgende Profile für Classical CANopen sind aktuell bei emotas verfügbar:

- ▶ **CiA 401** - Geräteprofil für digitale/analoge I/O-Module
- ▶ **CiA 402** - Zustandsmaschine für Antriebe
- ▶ **CiA 404** - Geräteprofil für Sensoren und Regler
- ▶ **CiA 406** - Geräteprofil für Encoder
- ▶ **CiA 413** - Geräteprofil für Nutzfahrzeuge
- ▶ **CiA 418** - Geräteprofil für Batterien
- ▶ **CiA 419** - Geräteprofil für Ladegeräte
- ▶ **CiA 422** - CleANopen, Geräteprofil für Abfallsammelfahrzeuge
- ▶ **CiA 437** - Applikationsprofil f. Photovoltaik-Komponenten
- ▶ **CiA 443** - Applikationsprofil für SIIS Level-2 Geräte
- ▶ **CiA 447** - Applikationsprofil für Zubehörteile an Spezialfahrzeugen (Taxameter, Rundumleuchten)
- ▶ **CiA 454** - Applikationsprofil für Energie Management Systeme u.a. in LEVs (EnergyBus)

Der emotas CANopen FD Protokoll Stack ist auch für die Entwicklung von Standard CANopen Geräten geeignet.

## << CANopen Dienstmerkmale <<

| Dienstmerkmal                   | Basic Slave | Master/Slave   | Manager        |
|---------------------------------|-------------|----------------|----------------|
| SDO Server*                     | 2           | 128            | 128            |
| SDO Client*                     |             | 128            | 128            |
| SDO expedited/segmented/block*  | ●/●/-       | ●/●/●          | ●/●/●          |
| USDO Connections**              | 2           | 512            | 512            |
| PDO Producer                    | 32          | 512            | 512            |
| PDO Consumer                    | 32          | 512            | 512            |
| PDO Mapping                     | static      | static/dynamic | static/dynamic |
| MPDO Dest.Mode*                 |             | ●              | ●              |
| MPDO Src. Mode*                 |             | ●              | ●              |
| SYNC Producer                   |             | ●              | ●              |
| SYNC Consumer                   | ●           | ●              | ●              |
| Time Producer                   |             | ●              | ●              |
| Time Consumer                   |             | ●              | ●              |
| Emergency Producer              | ●           | ●              | ●              |
| Emergency Consumer              |             | 127            | 127            |
| Guarding Master*                |             |                | ●              |
| Guarding Slave                  | ●           | ●              | ●              |
| Bootup Handling                 |             | ●              | ●              |
| Heartbeat Producer              | ●           | ●              | ●              |
| Heartbeat Consumer              | 1           | 127            | 127            |
| NMT Master                      |             | ●              | ●              |
| NMT Slave                       | ●           | ●              | ●              |
| Master Bootup CiA302-2*         |             |                | ●              |
| Flying Master CiA302-2*         |             | ○              | ●              |
| Configuration Manager CiA302-3* |             |                | ●              |
| SRD CiA302-5*                   |             | ○              | ●              |
| Redundancy CiA302-6*            |             | ○              | ○              |
| CANopen Router CiA302-7*        |             | ○              | ○              |
| LED CiA303                      | ●           | ●              | ●              |
| Safety*                         | ○           | ○              | ○              |
| LSS CiA305                      | ●           | ●              | ●              |
| Networking/Routing*             | ○           | ○              | ○              |
| ISO-TP                          |             | ●              | ●              |
| Multiline                       |             | ○              | ○              |
| C#-API-Wrapper Windows          | ○           | ○              | ○              |

\* Classic CAN Protokoll Stack \*\* CAN FD Protokoll Stack

● inklusive ○ optional

## # Unterstützte Plattformen

Die emotas CANopen Protokoll Stacks können mit und ohne Betriebssystem verwendet werden. Sie unterstützen eine Vielzahl von CAN-Controller und CPU-Typen.

Darunter: Bosch, Infineon, Intel, NXP, ST Microelectronics, Texas Instruments, Atmel, Renesas, Microchip.

Die Liste der unterstützten Hardwareplattformen wird ständig erweitert:

<https://www.emotas.de/info/controller>

Sollte Ihre Plattform dennoch einmal nicht verfügbar sein, ist die Entwicklung weiterer Treiberpakete problemlos möglich.

## # Lieferumfang

- ▶ CANopen Protokoll Sourcecode in ANSI-C (MISRA-konform)
- ▶ CANopen Treiber
- ▶ CANopen DeviceDesigner (1 Floating Lizenz mit Dongle)
- ▶ sofort lauffähige Beispielapplikationen
- ▶ Benutzer und Referenzhandbuch
- ▶ optionaler Wartungsvertrag
- ▶ optionaler Integrationssupport

## # Lizenzierung

- ▶ Standortlizenz ohne Runtime-Gebühren (Royalty free; inkl. 24 Monate Telefon und E-Mail-Support)
- ▶ Projektlizenz (Royalty free; inkl. 6 Monate Telefon und E-Mail-Support)

## # Bestellinformation

- ▶ 5004-10-FD | CANopen FD Slave Pack
- ▶ 5004-11-FD | CANopen FD Master/Slave Pack
- ▶ 5004-12-FD | CANopen FD Manager Pack
- ▶ 5004-15-FD-ML | CANopen FD Slave Pack MultiLine
- ▶ 5004-16-FD-ML | CANopen FD Master/Slave Pack MultiLine
- ▶ 5004-17-ML | CANopen Manager Pack Multiline
- ▶ 1004-4x-FD | CANopen FD Profile
- ▶ 2004-xx-FD | CANopen FD Treiber

CANopen Beispiel-Implementierungen zum Download: [www.emotas.de/download](http://www.emotas.de/download)

# << CANopen DeviceDesigner <<

Für den schnellen und kostensparenden Entwurf von CANopen-Geräten ist der CANopen DeviceDesigner das optimale Tool. Mit wenigen Klicks fügt der Anwender aus vordefinierten Profilen das Objektverzeichnis des Geräts zusammen und spezifiziert die Eigenschaften der Objekte. Der CANopen DeviceDesigner erzeugt daraus das Objektverzeichnis und Initialisierungsfunktionen, das elektronische Datenblatt sowie eine Gerätedokumentation. Zusätzlich konfiguriert der CANopen DeviceDesigner anhand der Geräteeigenschaften den CANopen Stack und den CANopen Treiber optimal.

Der CANopen DeviceDesigner führt den Anwender durch die Konfiguration der verwendeten Hardware (Mikrocontroller). Dabei sind nur wenige Eigenschaften wie CPU Typ, CANController Typ und einige Timing-Parameter wie die Taktfrequenz des Mikrocontrollers auszuwählen.

Das Objektverzeichnis wird entsprechend der Profile und Eingaben als C-Quellcode-Datei generiert, die in die Applikation als Interface zum Protokollstack eingebunden wird. Entsprechend der Konfiguration kann auf die Objekte des Objektverzeichnisses direkt als C-Variable aus der Applikation zugegriffen werden. Alternativ ist aus der Applikation ein Zugriff über Funktionen via Index und Subindex möglich. Auch die Weiterverwendung bestehender Applikationsvariablen ist möglich.

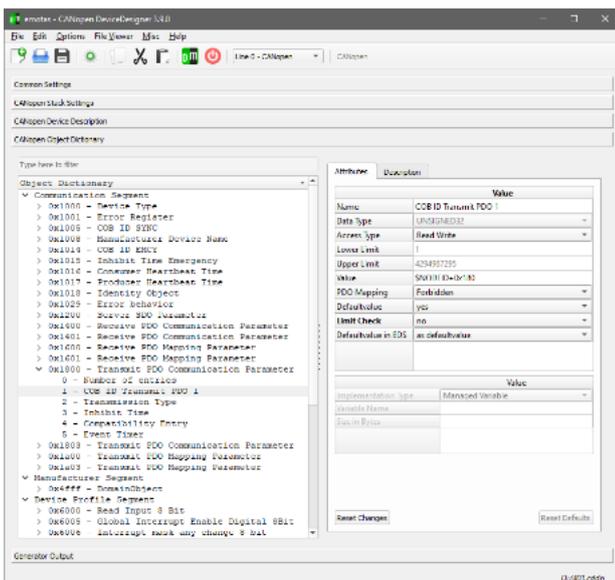
## << Geräteprofile <<

Für die Kommunikations- und Geräteprofile des CiA stehen Profildateien zur Verfügung.

Eine Profildatei umfasst Vorlagen für alle Objekttypen eines Profils mit den Standardeigenschaften und einer Erläuterung zu jedem Objekt. Die Objekte können im CANopen DeviceDesigner importiert, vervielfältigt und an individuelle Eigenschaften der Geräte angepasst werden. Die Entwicklungszeit verkürzt sich bei der Verwendung vordefinierter Objekte signifikant.

Fehler durch manuelle Eingabe der Daten entfallen.

optimale Anbindung der Applikation an den Stack mit dem CANopen DeviceDesigner



Folgende Profildateien sind aktuell für den CANopen DeviceDesigner verfügbar:

- ▶ **CiA 301** - CANopen application layer/ communication profile
- ▶ **CiA 302** - CANopen additional application layer functions
- ▶ **CiA 401** - Generic I/O Devices
- ▶ **CiA 402** - Drives and motion control
- ▶ **CiA 404** - Measuring devices /closed-loop controllers
- ▶ **CiA 418** - Battery modules
- ▶ **CiA 419** - Battery Charger
- ▶ **CiA 437** - Grid-based photovoltaic
- ▶ **CiA 443** - SIIIS Level 2 devices
- ▶ **CiA 447** - Special purpose car add-on Devices
- ▶ **CiA 454** - Energy Management / EnergyBus

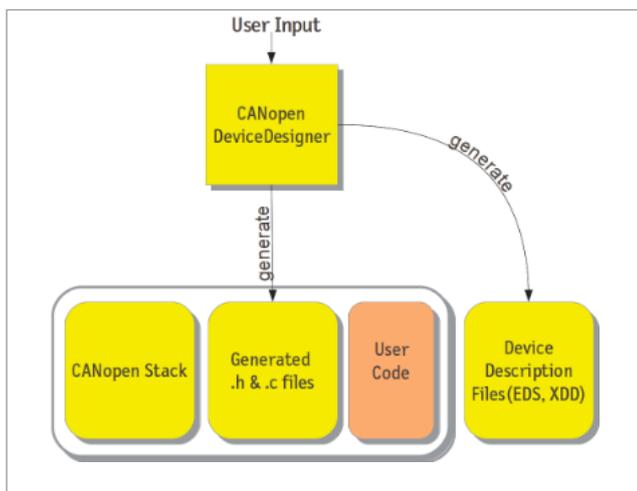
## << Programmausgaben <<

Aus den Benutzereingaben werden .h-Konfigurationsdateien, .c-Applikationscode sowie erforderliche Gerätebeschreibungsdateien in verschiedenen Ausgabeformaten erzeugt. Die eingegebenen Daten werden vor der Generierung auf Plausibilität geprüft und Warnungen und Fehlermeldungen entsprechend angezeigt.

Die Konfigurationsdateien konfigurieren den CAN-open Stack und aktivieren durch das Setzen von Compiler-Schaltern nur die jeweils benötigten Funktionalitäten der Stacks. Passgenau generierte C-Dateien beinhalten das CANopen-Objektverzeichnis, PDO-Mapping-Code und Schnittstellen zur Applikation.

Der CANopen Protokollstack mit den generierten Dateien ist so auf die Anwendung zugeschnitten und hinsichtlich des Speicherbedarfs an RAM und Flash für die jeweilige Anwendung optimiert.

## << Design Flow <<



## << Generierte Dateien <<

- Objektverzeichnis
- CANopen Stack-Konfiguration
- CANopen Stack-Initialisierung
- Electronic Datasheet (EDS nach CiA 306)
- XML Device Description (XDD nach CiA 311)
- Objektverzeichnisdokumentation in HTML
- Objektaufzählung im CSV-Format zum Import in Tabellenkalkulationsprogramme
- Beschreibung der Objekte im txt-Format

## # Anforderungen

- ▶ Windows 10 oder höher
- ▶ diverse Linux-Varianten (getestet mit (K)Ubuntu)
- ▶ CANopen Protokoll Stack von emotas

## # Lieferumfang

- ▶ Software zum Download
- ▶ Handbuch im PDF-Format
- ▶ Profildatenbanken für verschiedene CiA-Profile

## # Lizenzierung

- ▶ Named User Lizenz/Node Locked Lizenz (bei Projektlizenz inklusive zum Stack)
- ▶ Floating Lizenz mit Dongle (bei Standortlizenz inklusive zum Stack)
- ▶ Firmenlizenzen auf Anfrage

## # Bestellinformation

- ▶ 1006-10 | CANopen DeviceDesigner Windows
- ▶ 1006-11 | CANopen DeviceDesigner Linux
- ▶ 1006-10-D | CANopen DeviceDesigner Floating Lizenz Windows
- ▶ 1006-11-D | CANopen DeviceDesigner Floating Lizenz Linux

Die Named User Lizenz erlaubt die Installation und ausschließliche Nutzung des Programms durch einen namentlich genannten Benutzer auf beliebigen Computern unter Windows und unter Linux.

**Evaluierungsversionen des Tools auf Anfrage verfügbar.**

## << CANopen NetworkDesigner <<

Der emotas CANopen NetworkDesigner ermöglicht den Entwurf eines vollständigen CANopen-Netzwerks inklusive aller Geräte und Kommunikationsbeziehungen. Der entsprechende Quellcode für alle Komponenten wird automatisch durch das Tool generiert.

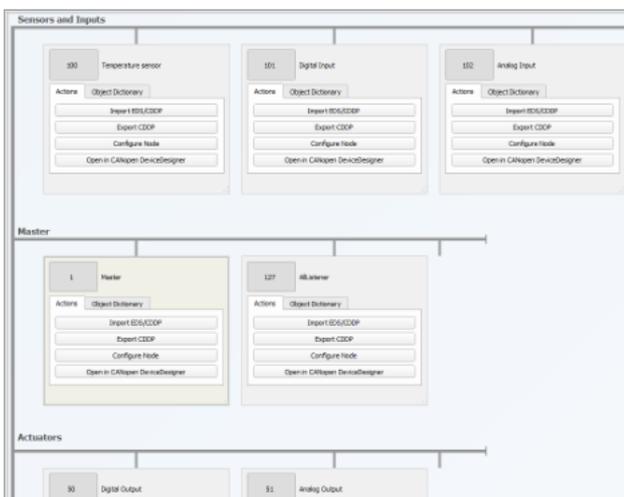
### << Geräte- und Netzwerkentwurf <<

Im CANopen NetworkDesigner werden Geräte, deren Signale und Kommunikationsbeziehungen (per PDO und MPDO) definiert. Das Tool erzeugt daraus den passenden Quellcode für alle Komponenten im Netzwerk. Der CANopen NetworkDesigner sorgt in Kombination mit dem emotas CANopen Stack für eine CANopen-konforme Realisierung des Entwurfs. Heartbeat Producer- und Consumer-Beziehungen sowie die Verknüpfung von Emergency Producer und Consumer erfolgen ebenfalls automatisiert.

### << Objektverknüpfung <<

Bei der Konfiguration werden in einer Tabellenansicht die Objekte der Producer und Consumer übersichtlich dargestellt. Dabei treten die CANopen-Details für den Anwender in der Hintergrund und anstelle der CANopen-Notation Objekt 0x6041, SubIndex 1 wird der Name des Signals beispielsweise 'Control Word' angezeigt. Mit einem Klick können nun Objekte der Producer und Consumer verbunden werden. Ungültige Verbindungen werden im Vorfeld ausgeschlossen, da nur Verbindungen angezeigt werden, welche entsprechend ihren CANopen-Datentypen übereinstimmen.

Die Vergabe bestimmter Prioritäten für einzelne Verknüpfungen ist möglich.



### << PDO Linking <<

Entsprechend der Einstellungen der Anwender berechnet das Tool selbstständig die PDO-Konfiguration mit dem Mapping und den COB-IDs der CANopen-Geräte. Eine manuelle Konfiguration der PDO-Parameter durch Experten ist zusätzlich möglich. Dabei sichert der CANopen NetworkDesigner die Gültigkeit der Konfiguration.

.....  
 CANopen NetworkDesigner –  
 Netzwerkdesign der neuesten Generation  
 .....

### << Simulation <<

Eine Simulation der CANopen-Geräte auf dem PC ist ebenfalls integriert. Dabei können noch nicht vorhandene Geräte und ihre Prozesseingänge (z.B. digitale Eingänge, Temperatursensoren,...) komfortabel simuliert und mit Werten stimuliert werden. In der simulierten Applikation wird der gleiche CANopen-Protokoll-Stack wie in der späteren realen Applikation verwendet. Damit muss für die Simulation keine separate Skriptsprache erlernt werden und Algorithmen wie z.B. Regler und andere Signalverarbeitung können in C erstellt und später auf dem realen Gerät nachgenutzt werden.

### << Netzwerkanalyse <<

Aus dem CANopen NetworkDesigner können Projektdateien in den CANopen DeviceExplorer importiert

## # Anforderungen

- ▶ Windows 10 oder höher
- ▶ Diverse Linux-Varianten (getestet mit (K)Ubuntu)

## # Lieferumfang

- ▶ CANopen NetworkDesigner zum Download
- ▶ Handbuch im PDF-Format

## # Lizenzierung

- ▶ Named User Lizenz
- ▶ Floating Lizenz mit Dongle
- ▶ Firmenlizenzen auf Anfrage

## # Bestellinformation

- ▶ 1027-10 | CANopen NetworkDesigner  
Named User Lizenz Windows
- ▶ 1027-11 | CANopen NetworkDesigner  
Named User Lizenz Linux
- ▶ 1027-10-D | CANopen NetworkDesigner  
Floating Lizenz Windows
- ▶ 1027-11-D | CANopen NetworkDesigner  
Floating Lizenz Linux

## << CANopen DeviceExplorer <<

Für Entwicklungsaufgaben, Test-, Diagnose- und Serviceaufgaben ist der CANopen DeviceExplorer das geeignete Tool. Er stellt CANopen-Masterfunktionen bereit und erlaubt damit den Test und die Konfiguration von CANopen-Geräten.

Informationen über das jeweilige CANopen-Gerät werden aus dem elektronischen Datenblatt des Geräts (EDS oder XDD-Format) entnommen oder direkt aus dem Gerät gelesen. Mithilfe standardisierter DCF-Dateien (Device configuration files) können Gerätekonfigurationen gespeichert und geladen werden. Die Verwaltung ganzer CANopen-Netzwerke in Projektdateien ist ebenfalls möglich. Individuelle Test- und Steuerungsprogramme lassen sich mit geringem Aufwand mithilfe des Script PlugIn erstellen.

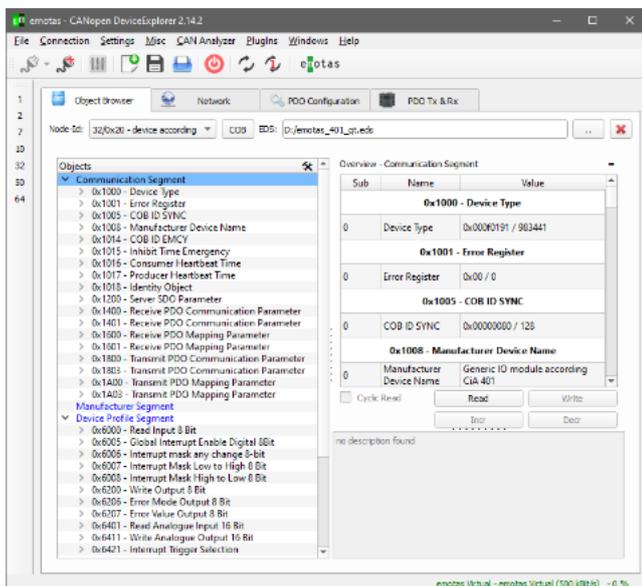
### << Objektverzeichnis <<

Die Objekte des Geräts werden im übersichtlichen Objektbrowser dargestellt. Auf jedes Objekt kann – entsprechend den Zugriffsrechten – lesend und schreibend zugegriffen werden. Dabei unterstützt der CANopen-DeviceExplorer sämtliche SDO-Dienste. Die Objektwerte sind je nach Datentyp dezimal, hexadezimal, binär oder im ASCII-Format darstellbar. Große Daten (z.B. Domain-Blöcke) können direkt aus Dateien gelesen oder in Dateien geschrieben werden. SDO-Zugriffe auf Objekte, die nicht in der EDS-Datei verzeichnet sind, sind auch möglich.

### << Network Management <<

Mit dem CANopen DeviceExplorer kann das Netzwerk nach vorhandenen Geräten gescannt werden. Heartbeat und Node-Guarding der Geräte im Netzwerk können konfiguriert und überwacht werden. Das Senden aller CANopen-NMT Kommandos an die Geräte ist möglich und die NMT-Zustände der Geräte können überwacht werden.

.....  
CANopen DeviceExplorer – unentbehrlich für Ihre  
CANopen-Geräteentwicklung  
.....



### << PDO-Konfiguration <<

Das Tool unterstützt die einfache Konfiguration des PDO-Mappings per Drag&Drop. PDOs lassen sich mit einem Klick senden, empfangen und visualisieren. Zu Test- und Diagnosezwecken kann die SYNC-Nachricht zum Senden zyklischer PDOs einmalig oder zyklisch mit und ohne SYNC-Counter gesendet werden. Der CANopen DeviceExplorer kann den Empfang von timer-getriggerten PDOs überwachen.

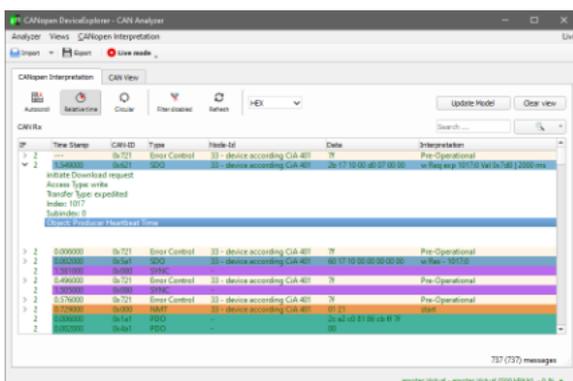
## << PlugIns – Erweiterungsmodule <<

Der CANopen DeviceExplorer ist als **Basic** oder **Full**-Version erhältlich und kann durch verschiedene optionale PlugIns erweitert werden. Die folgenden PlugIns für die entsprechenden Versionen sind derzeit verfügbar:

| PlugIns/Variante             | Basic | Full | separat |
|------------------------------|-------|------|---------|
| CANopen FD Support           | ●     | ●    |         |
| Object Browser (U)SDO Client | ●     | ●    |         |
| NMT master                   | ●     | ●    |         |
| PDO configuration            | ●     | ●    |         |
| PDO visualization            | ●     | ●    |         |
| Multi-Data Plotter           | ●     | ●    |         |
| DCF editor                   | ●     | ●    |         |
| configuration download       | ●     | ●    |         |
| CAN analyzer                 | ●     | ●    |         |
| CAN replayer                 | ●     | ●    |         |
| CANopen Interpreter          | ●     | ●    |         |
| LSS Master                   |       |      | ●       |
| CAN/CANopen Scripting        |       | ●    |         |
| SRDO Configuration           |       |      | ●       |
| Process DataLinker           |       |      | ●       |
| J1939 Interpreter            |       |      | ●       |
| ISO-TP Interpreter           |       | ●    |         |
| EnergyBus Interpreter        |       |      | ●       |
| CiA 402 Control              |       | ●    |         |
| Multi-Interface-Support      |       | ●    |         |

### weitere PlugIns:

- ▶ 1007-20 | CDE PlugIn - Scripting
- ▶ 1007-21 | CDE PlugIn LSS
- ▶ EnergyBus PlugIns auf Anfrage



## # Anforderungen

- ▶ Windows 10 oder höher
- ▶ Linux-Varianten getestet mit (K)Ubuntu
- ▶ unterstütztes CAN-Interface

## # CAN Schnittstellen

### Windows®

- ▶ IXXAT
- ▶ SYS TEC USB-CANmodul
- ▶ PEAK PCAN-USB
- ▶ Kvaser CAN-Interfaces
- ▶ EMS Wünsche CAN-Interfaces
- ▶ Vector CAN-Interfaces (XL-Driver)

### Linux®

- ▶ can4linux
- ▶ SocketCAN

Unter Linux werden alle CAN-Schnittstellen mit SocketCAN-API oder can4linux-API unterstützt.

## # Lieferumfang

- ▶ CANopen DeviceExplorer zum Download
- ▶ CANopen Interpreter PlugIn
- ▶ Handbuch im PDF-Format

## # Lizenzierung

- ▶ Named User Lizenz für CANopen DeviceExplorer
- ▶ Floating Lizenz mit Dongle
- ▶ Firmenlizenzen auf Anfrage

## # Bestellinformation

- ▶ 5007-10/-11 | CANopen DeviceExplorer Basic Windows/Linux
- ▶ 5007-50/51 | CANopen DeviceExplorer Full Windows/Linux
- ▶ 5007-10-D/11-D | CANopen DeviceExplorer Basic Floating Lizenz Windows/Linux
- ▶ 5007-50-D/51-D | CANopen DeviceExplorer Full Floating Lizenz Windows/Linux

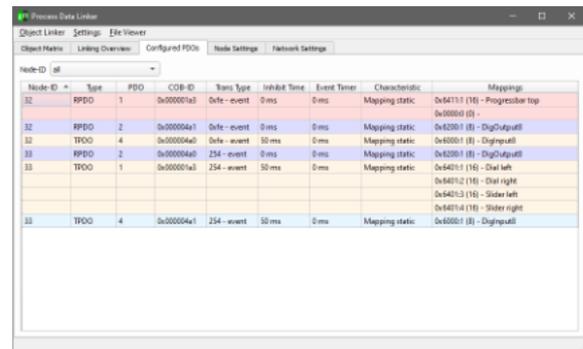
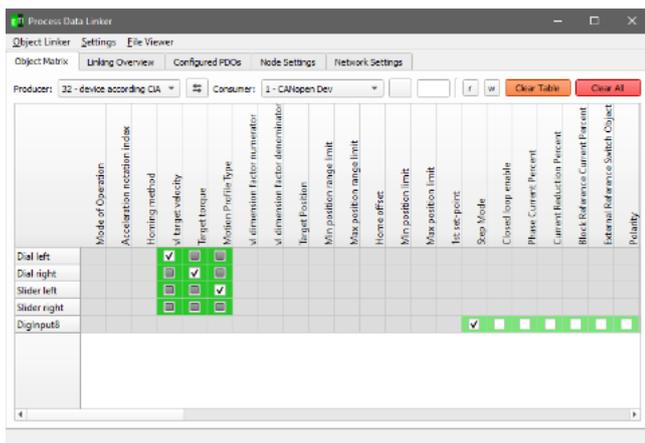
Testen Sie den CANopen DeviceExplorer.  
Eine Demo finden Sie auf unserer Website oder auf  
Anfrage unter [service@emotas.de](mailto:service@emotas.de)

## << Process DataLinker <<

Der Process DataLinker ist eine leistungsfähige Erweiterung zum CANopen DeviceExplorer zur Parametrierung der Kommunikations-Verbindungen in CANopen-Netzwerken. Mit wenigen Klicks können die Verknüpfungen zwischen den Geräten definiert werden, woraus der Process DataLinker automatisch die PDO-Konfiguration erstellt. Es werden keine Detailkenntnisse von CANopen mehr benötigt und damit wird das PDO-Linking deutlich vereinfacht.

Mit dem CANopen DataLinker PlugIn steht eine leistungsfähige Erweiterung zum DeviceExplorer bereit. Das PlugIn dient zur Parametrierung der Kommunikationsverbindungen in CANopen-Netzwerken. Mit wenigen Klicks können die Verknüpfungen zwischen den Geräten definiert werden. Der CANopen DataLinker erstellt daraus automatisch die PDO-Konfiguration. Es werden keine Detailkenntnisse von CANopen mehr benötigt, was das PDO-Linking deutlich vereinfacht.

Zusätzlich ist es möglich, Prioritäten für einzelne Verknüpfungen festzulegen.



### << PDO – Linking <<

Entsprechend den Einstellungen des Anwenders berechnet der CANopen DataLinker selbständig die PDO-Konfiguration mit dem Mapping und den COB-IDs der CANopen Geräte. Sonderfälle wie statisches Mapping und fixe COB-IDs werden dabei berücksichtigt. Die manuelle Konfiguration der PDO-Parameter als Möglichkeit für Experten ist ebenfalls möglich. Dabei überprüft der CANopen DataLinker die Gültigkeit der vorgenommenen Konfiguration.

### << Objektverknüpfungen <<

Die Objekte der Producer und Consumer werden bei der Konfiguration in einer Tabellenansicht dargestellt. Übersichtlichkeit wird dabei zusätzlich durch den Verzicht auf die übliche CANopen Notation (0x6041, Subindex 1) erreicht. Stattdessen wird ein Objektname (z.B. Control Word) angezeigt. Mit einem Klick lassen sich Objekte der Producer und Consumer miteinander verbinden. Die Erstellung ungültiger Verbindungen ist ausgeschlossen, da nur Verbindungen mit entsprechend übereinstimmenden CANopen-Datentypen angezeigt werden.

.....  
 CANopen DataLinker – unentbehrlich für die CANopen-  
 Netzwerkkonfiguration  
 .....

### << Timer Konfiguration <<

Per Heartbeat lassen sich PDO-spezifische Timing Parameter wie in Inhibit Time oder Event Timer oder SYNC Timings und die Knotenüberwachung konfigurieren.

## << CANopen PLC Support <<

CANopen SPS nach CiA-405 werden durch den CANopen DataLinker direkt unterstützt. Netzwerkvariablen können für jede SPS angelegt und mit den Objekten der anderen Geräte verknüpft werden. Mehrere SPS in einem Projekt sind möglich. Die Variablendefinitionen nach IEC61131 werden direkt durch den CANopen DataLinker erzeugt.

```
File Viewer: /o/1007/software/cde/plcproject/plc12/pdo_var_1.txt
(* IEC61131-3 Variable Definition File node 1 *)
VAR_EXTERNAL
  NV_U8_INPUT_1: BYTE; (* 0xa040:1 *)
  NV_U8_INPUT_2: BYTE; (* 0xa040:2 *)
  NV_I16_INPUT_1: INT; (* 0xa0c0:17 *)
  NV_I16_INPUT_2: INT; (* 0xa0c0:18 *)
  NV_U8_OUTPUT_1: BYTE; (* 0xa4c0:53 *)
  NV_U8_OUTPUT_2: BYTE; (* 0xa4c0:54 *)
  N3_A13: INT; (* 0xa540:22 *)
  N3_A14: INT; (* 0xa540:31 *)
END_VAR
(* generated by emtas CANopen Device Explorer - Process Data Linker *)
```

## << Geräteparametrierungen <<

Herstellerspezifische Objekte und gerätespezifische Objekte der Geräte können über den CANopen DeviceExplorer konfiguriert werden. Diese Einstellungen werden in den CANopen DataLinker und die erzeugten Konfigurationsdateien (DCF) übernommen. Zusätzlich können alle Konfigurationsdaten direkt über CAN in die Geräte geschrieben werden.

### # Anforderungen

- ▶ CANopen DeviceExplorer
- ▶ unterstütztes CAN-Interface

### # CAN Schnittstellen

#### Windows®

- ▶ IXXAT
- ▶ SYS TEC USB-CANmodul
- ▶ PEAK PCAN-USB
- ▶ Kvaser CAN-Interfaces
- ▶ EMS Wünsche CAN-Interfaces
- ▶ Vector CAN-Interfaces (XL-Driver)

#### Linux®

- ▶ can4linux
- ▶ SocketCAN

Unter Linux werden alle CAN-Schnittstellen mit SocketCAN-API oder can4linux-API unterstützt.

### # Lieferumfang

- ▶ CANopen Datalinker PlugIn Software zum Download

### # Lizenzierung

- ▶ Named User Lizenz
- ▶ Floating Lizenz mit Dongle
- ▶ Firmenlizenzen auf Anfrage

### # Bestellinformation

- ▶ 5007-25 | CANopen DeviceExplorer inkl. Process DataLinker PlugIn

## << CANopen Bootloader <<

Der CANopen Bootloader ist mit dem Fokus auf geringen Speicherbedarf entwickelt. Er unterstützt nur die für einen Bootloader zwingend notwendigen CANopen-Dienste und Objekte entsprechend dem CANopen-Profil CiA 302-3.

Der emotas Bootloader kommt mit 16KB Flash aus und ist dennoch CANopen-kompatibel. So kann die Aktualisierung der Firmware durch Standard CANopen-Master oder CANopen Konfigurationstools erfolgen.

kleiner Speicherbedarf - großer Nutzen

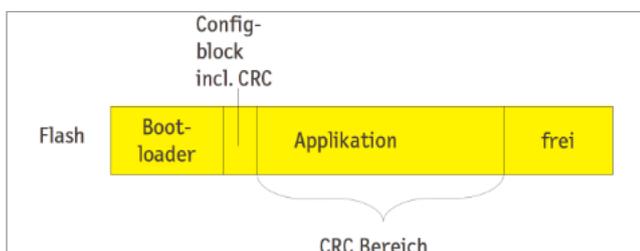
Verfügbar ist der Bootloader für verschiedene 16- und 32-Bit-Mikrocontroller. Durch die portable Treiberanbindung ist eine Anpassung an andere Zielsysteme einfach möglich. Erweiterungen, wie Layer Settings Service (LSS) zur Konfiguration der Knotennummer, sind verfügbar. Herstellerspezifischer Programmcode z.B. zur Signierung der Firmwaredaten kann schnell und unkompliziert integriert werden.

### << CANopen Dienste <<

NMT Slave - Bootloader dauerhaft in Preoperational  
 Heartbeat - Producer Timer wird fest zur  
 Compile-Zeit definiert  
 SDO Server - expedited/segmentierter/Block Transfer  
 SDO Abort - General Error  
 Boot-Up - Boot-Up-Nachricht beim Start

### << Flashlayout <<

Das Kommandozeilentool zur CRC-Erstellung berechnet die CRC aus der Binärdatei der Applikation. Es stellt einen Konfigurationsblock mit CRC an den Anfang der Binärdatei. Vom Bootloader wird die Binärdatei mit Konfigurationsblock inklusive CRC und Applikation in den Flash geschrieben.



Durch die Lieferung als Sourcecode lassen sich auch eigene Dateien in den Konfigurationsblock einfügen und der Applikation zugänglich machen.

### # Anforderungen

- ▶ ANSI-C Compiler
- ▶ Mikrocontroller mit In-Application-Programming Unterstützung

### # Lieferumfang

- ▶ Bootloader Source Code (ANSI-C kompatibel)
- ▶ Anpassung für CPU, ersten internen CAN Controller und internen Flash
- ▶ EDS-Datei des Bootloaders
- ▶ sofort lauffähige Beispiel-Applikation
- ▶ Benutzerhandbuch im PDF-Format
- ▶ Kommandozeilentool zur Checksummen-Generierung für MS Windows® und Linux
- ▶ inkl. 1 Named User Lizenz (Projekt) oder Floating Lizenz (Standort) für CAN UpdateManager

### # Lizenzierung

- ▶ Standortlizenz ohne RuntimeGebühren (Royalty free; inkl. 24 Monate Telefon und E-MailSupport)
- ▶ Projektlizenz (Royalty free; inkl. 6 Monate Telefon und E-MailSupport)

### # Bestellinformation

- ▶ 5001-10 | CANopen Bootloader generisch

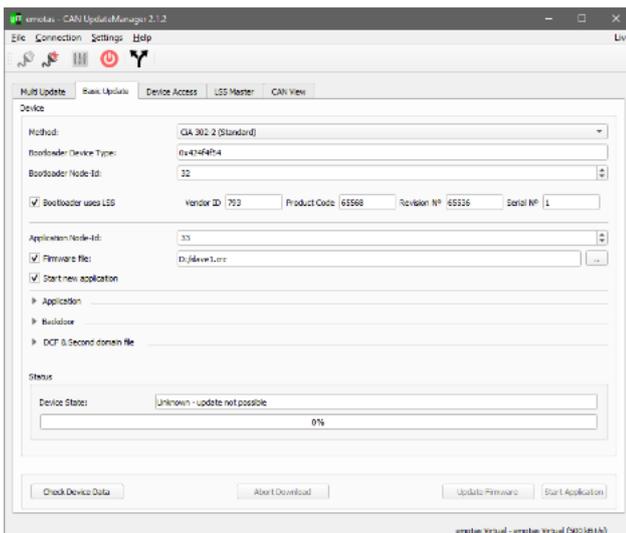
## << CAN UpdateManager <<

Der CAN UpdateManager ist für Firmware-Downloads und Netzwerk-Updates entwickelt. Größter Wert wurde dabei auf einfache Bedienbarkeit gelegt. Der UpdateManager unterstützt sowohl Update-Mechanismen nach CiA 302-2 als auch nach verschiedenen Applikationsprofilen inklusive dem Zugriff auf Passwortobjekte.

Für Entwickler und Integratoren von CANopen-Bootloadern bietet der integrierte CANopen-Master mit SDO-Client (expedited, segmented und Block-Transfer) und NMT-Master direkten Zugriff auf das CANopen-Gerät. Bei der Entwicklung kann so schrittweise validiert werden, ob sich Gerät und Bootloader erwartungsgemäß verhalten. Parametrierungen von CANopen-Geräten sind mit dem Tool leicht möglich. Die CAN-Kommunikation zwischen UpdateManager und dem Gerät lässt sich zu Diagnosezwecken aufzeichnen.

### Firmware Download und Netzwerk Update in einem Tool

Neben direktem Firmware-Download zu einem Gerät können Firmware-Dateien für alle Geräte eines Netzwerks zusammen mit Konfigurationsdaten in ein Update-Package gelegt werden. Der Service-Techniker vor Ort startet hierzu das Tool mit dem Update-Package, um alle Geräte eines Netzwerks automatisch zu aktualisieren.



### # Anforderungen

- ▶ Windows 10 oder höher
- ▶ Linux-Varianten getestet mit (K)Ubuntu
- ▶ unterstütztes CAN-Interface

### # CAN Schnittstellen

#### Windows®

- ▶ IXXAT
- ▶ SYS TEC USB-CANmodul
- ▶ PEAK PCAN-USB
- ▶ Kvaser CAN-Interfaces
- ▶ EMS Wünsche CAN-Interfaces
- ▶ Vector CAN-Interfaces (XL-Driver)

#### Linux®

- ▶ can4linux
- ▶ SocketCAN

Unter Linux werden alle CAN-Schnittstellen mit SocketCAN-API oder can4linux-API unterstützt.

### # Lieferumfang

- ▶ Software zum Download
- ▶ Handbuch im PDF-Format

### # Lizenzierung

- ▶ Named User Lizenz inkl. einer Projektlizenz
- ▶ Floating Lizenz mit Dongle inkl. einer Standortlizenz
- ▶ Firmenlizenzen für Service-Techniker oder Buyout-Versionen auf Anfrage

### # Bestellinformation

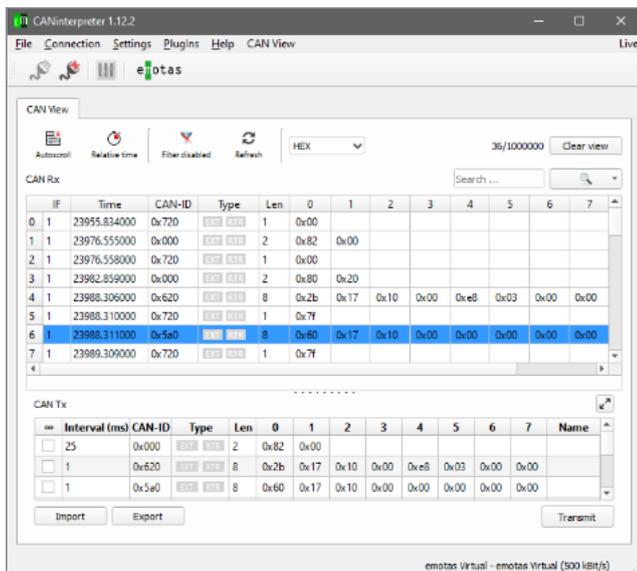
- ▶ 5002-10/11(-D) | CAN UpdateManager Basic  
Windows/Linux  
PlugIns: CANopen
- ▶ 5002-50/51(-D) | CAN UpdateManager Full  
Windows/Linux  
PlugIns: CANopen, J1939 (CAM11/CAM21), UDS

## << CANinterpreter <<

Das vielseitig einsetzbare Monitoring-Tool CANinterpreter dient zur Überwachung und Aufzeichnung des Datenverkehrs und zur Analyse von CAN-Netzwerken. Verschiedene PlugIns erweitern bei Bedarf den Funktionsumfang.

### << Grundfunktionen <<

- Online-Überwachung des Busverkehrs CAN-Daten-Interpretation nach Anwendervorgaben
- Aufzeichnung der CAN-Telegramme
- Flexible CAN-ID spezifische Filterung/Triggerung
- einmaliges oder zyklisches Senden von CAN Nachrichten/Sequenzen
- mit einem geeigneten Interface ist die Verwendung für CAN FD möglich



### << PlugIns/Erweiterungsmodule <<

Durch optionale PlugIns lässt sich der Funktionsumfang erweitern. Die Erstellung zusätzlicher PlugIns nach Kundenwunsch ist möglich. Anwendungsfälle hierfür sind z.B. Service- oder Inbetriebnahmetools für Geräte und Anlagen.

### Scripting PlugIn

Das PlugIn basiert auf QtScript/JavaScript, erweitert um CAN und CANopen spezifische Kommandos. Mithilfe des Tools können eigene Benutzer-oberflächen z.B. für Test- und Steuerungs-applikationen erstellt werden. Der Entwurf erfolgt über einen grafischen UI-Designer. Lauffähige Beispiele liegen der Auslieferung bei.

### PlugIn EnergyBus Visualisierung

Das PlugIn ist speziell für die Anwendung mit EnergyBus entwickelt. Es überwacht den Datenverkehr bei EnergyBus (CiA 454) und zeigt die Zustände und Prozessdaten aller EnergyBus-Geräte im Netzwerk an.

### PlugIn CANopen Interpretation

Die Interpretation von CANopen-Telegrammen anhand der CANopen-Dienste wird mit diesem PlugIn möglich. Anstelle der Datenbytes "40 10 17 00" zeigt das PlugIn z.B. an, dass das Heartbeat-Producer-Objekt (0x1017) per SDO gelesen werden soll. Für alle CANopen-Dienste sind entsprechende Interpretationen verfügbar. Dateninhalte der PDOs können anhand des Mappings angezeigt werden. Eine Filterung der Telegramme ist nach CANopen-Knotennummern und CANopen-Diensten möglich.

### PlugIn J1939

Das J1939-PlugIn ist eine optionale Erweiterung des CANopen DeviceExplorers bzw. des CANinterpreters und stellt Funktionen zur Interpretation und zum Senden von J1939-Nachrichten bereit. Einzelne SPN-Werte können hierbei grafisch in Charts dargestellt werden.

### PlugIn UDS

Das UDS-PlugIn ist optional für den CANopen DeviceExplorer bzw. den CANinterpreter erhältlich und stellt Funktionen zur Interpretation und zum Senden von UDSonCAN-Nachrichten bereit.

## # Anforderungen

- ▶ Windows 10 oder höher
- ▶ diverse Linux-Varianten (getestet mit (K)Ubuntu)
- ▶ unterstütztes CAN-Interface

## # CAN Schnittstellen

### Windows®:

- ▶ IXXAT
- ▶ SYS TEC USB-CANmodul
- ▶ PEAK PCAN-USB
- ▶ Kvaser CAN-Interfaces
- ▶ EMS Wünsche CAN-Interfaces
- ▶ Vector CAN-Interfaces (XL-Driver)

### Linux®:

- ▶ can4linux
- ▶ SocketCAN

Unter Linux werden alle Hardware-Schnittstellen mit SocketCAN-API oder can4linux-API unterstützt

## # Lieferumfang

- ▶ CANinterpreter Software zum Download
- ▶ Handbuch im PDF Format

## # Lizenzierung

- ▶ Named User Lizenz
- ▶ Floating Lizenz mit Dongle

## # Bestellinformation

- ▶ 5017-10/11 | CANinterpreter Windows/Linux
- ▶ 5017-10-D/11-D | CANinterpreter Floating Lizenz Windows/Linux
- ▶ 1007-20 | PlugIn - Scripting
- ▶ 1007-22 | PlugIn CANopen Interpreter
- ▶ 1007-27 | PlugIn J1939
- ▶ 1017-30 | PlugIn UDS
- ▶ EnergyBus PlugIns auf Anfrage

## << CANopen Gateway CiA-309 <<

Das universelle CANopen Gateway gemäß CiA309-3 erlaubt über die zuverlässige und erprobte CANopen-Technologie den Ethernet-Zugriff bis hinunter auf die Feldebene.

### << Grundfunktionen <<

Das CiA309-3-Gateway ist ein universelles TCP/IP-CANopen-Gateway entsprechend der CiA-Spezifikation 309-3. Es ist als Binärversion für Linux verfügbar und kann mit jedem Linux, das eine can4linux oder SocketCAN CAN-Schnittstelle hat, verwendet werden. Zudem steht das Gateway auch im Quellcode zur Verfügung und kann damit auf alle Plattformen portiert werden, die eine CAN-Schnittstelle und einen TCP/IP-Stack mit BSD-Sockets besitzen. Die CANopen-Komponente des Gateways besteht aus dem bewährtem CANopen Master-Stack von emotas.

### << CANopen-Dienste <<

Es werden die in der Spezifikation 309-3 definierten CANopen-Dienste durch das Gateway unterstützt:

- SDO Client (Datentypen von 8..32 Bit, Strings und Domains)
- PDO Consumer
- PDO Producer
- NMT Kommandos
- Node Guarding Master
- Heartbeat Consumer
- Heartbeat Producer
- Emergency Consumer

Weitere CiA309-3 Kommandos zur Konfiguration des Gateways sind ebenfalls verfügbar.

Mit der Verwendung des CANopen-TCP-Gateways nach CiA309-3 können CANopen-Geräte einfach in TCP-Netzwerke integriert werden. Weitergehende Applikationen, die über den Funktionsumfang der CiA309-3-Spezifikation hinaus gehen, können mit der Source Code Variante auf Basis des CANopen Master Stacks entwickelt werden.

### # Anforderungen

- ▶ Raspberry Pi
- ▶ Microsoft Windows 7, 8,10
- ▶ diverse Linux-Varianten (getestet mit (K)Ubuntu)

### # Lieferumfang

- ▶ CANopen TCP Gateway Software zum Download
- ▶ Handbuch im PDF-Format

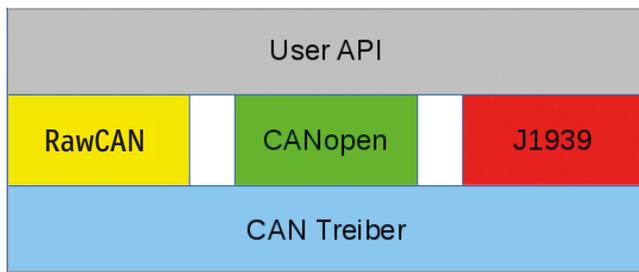
### # Bestellinformation

- ▶ 1021-xx | CiA 309 CANopen TCP Gateway

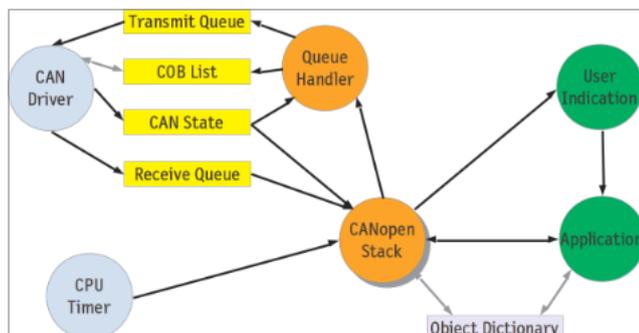
## << CAN Multiprotokoll Stack <<

Der CAN Multi-Protokoll Stack ist eine Software-Bibliothek mit der CAN-, CANopen- und J1939-Applikationen für ein oder mehrere CAN-Linien zusammen in einem Gerät implementiert werden können. Ein grafisches Konfigurationstool unterstützt die schnelle und komfortable Entwicklung und ist im Lieferumfang enthalten.

Der CAN Multi Protokoll Stack ist in ANSI-C erstellt und somit auf vielen Systemen einsetzbar. Der strukturierte Aufbau der Software und die bereitgestellte flexible Benutzerschnittstelle erlaubt eine einfache Auswertung empfangener Daten und die unkomplizierte Nutzung von CAN-, CANopen- und J1939- Diensten im Netzwerk.

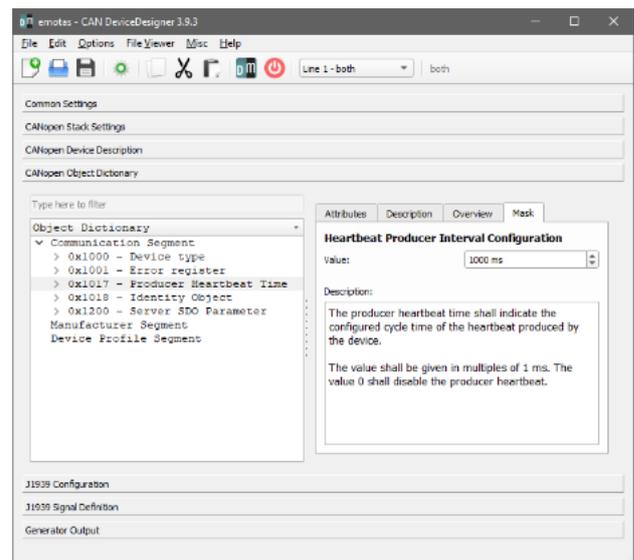


Die CANopen-Implementierung stellt alle Kommunikationsdienste entsprechend des CANopen Kommunikationsprofils CiA 301 V4.2 bereit und ist in verschiedenen Ausbaustufen erhältlich. Sie wird mit dem jeweils aktuellen CANopen Conformance Test des CiA e.V. auf Konformität zum Standard geprüft. Neben den Standard-Diensten des CiA 301 sind Erweiterungsmodule für verschiedene Kommunikations- und Applikationsprofile verfügbar.



Ein definiertes Treiberinterface gewährleistet die hohe Flexibilität des CAN Multi-Protokoll Stacks. Es ermöglicht eine einfache Adaption an vorhandene oder neue CAN-Controller oder CPU-Typen. Der Austausch der Hardwareplattform ist somit unkompliziert durchführbar.

Für möglichst geringen Ressourcenverbrauch sind die Protokoll Stacks konfigurier- und skalierbar. Die Konfiguration wird mit dem im Lieferumfang enthaltenen grafischen Softwaretool CAN DeviceDesigner vorgenommen.



Benutzeroberfläche CAN DeviceDesigner

Der CAN DeviceDesigner hilft beim Entwurf der CANopen-Geräte und der erforderlichen Dienste. Durch vordefinierte Profile und Objekte können auch das Objektverzeichnis, J1939 Dienstekennung und das EDS-File des Gerätes schnell und sicher erzeugt werden.

## << CANopen Dienstmerkmale <<

| Dienstmerkmal                      | Basic Slave | Master/Slave   | Manager        |
|------------------------------------|-------------|----------------|----------------|
| SDO Server*                        | 2           | 128            | 128            |
| SDO Client*                        |             | 128            | 128            |
| SDO expedited/<br>segmented/block* | ●/●/-       | ●/●/●          | ●/●/●          |
| USDO Connections**                 | 2           | 512            | 512            |
| PDO Producer                       | 32          | 512            | 512            |
| PDO Consumer                       | 32          | 512            | 512            |
| PDO Mapping                        | static      | static/dynamic | static/dynamic |
| MPDO Dest.Mode*                    |             | ●              | ●              |
| MPDO Src. Mode*                    |             | ●              | ●              |
| SYNC Producer                      |             | ●              | ●              |
| SYNC Consumer                      | ●           | ●              | ●              |
| Time Producer                      |             | ●              | ●              |
| Time Consumer                      |             | ●              | ●              |
| Emergency Producer                 | ●           | ●              | ●              |
| Emergency Consumer                 |             | 127            | 127            |
| Guarding Master*                   |             |                | ●              |
| Guarding Slave                     | ●           | ●              | ●              |
| Bootup Handling                    |             | ●              | ●              |
| Heartbeat Producer                 | ●           | ●              | ●              |
| Heartbeat Consumer                 | 1           | 127            | 127            |
| NMT Master                         |             | ●              | ●              |
| NMT Slave                          | ●           | ●              | ●              |
| Master Bootup CiA302-2*            |             |                | ●              |
| Flying Master CiA302-2*            |             | ○              | ●              |
| Configuration Manager<br>CiA302-3* |             |                | ●              |
| SRD CiA302-5*                      |             | ○              | ●              |
| Redundancy CiA302-6*               |             | ○              | ○              |
| CANopen Router CiA302-7*           |             | ○              | ○              |
| LED CiA303                         | ●           | ●              | ●              |
| Safety*                            | ○           | ○              | ○              |
| LSS CiA305                         | ●           | ●              | ●              |
| Networking/Routing*                | ○           | ○              | ○              |
| ISO-TP                             |             | ●              | ●              |
| Multiline                          |             | ○              | ○              |
| C#-API-Wrapper Windows             | ○           | ○              | ○              |

\* Classic CAN Protokoll Stack \*\* CAN FD Protokoll Stack

● inklusive ○ optional

| Dienstmerkmal                                 | J1939                     |
|-----------------------------------------------|---------------------------|
| Knotenadressvergabe                           | fest oder Adress Claiming |
| Zeitüberwachung von Nachrichten               | ●                         |
| Zyklisches Senden/Empfangen                   | ●                         |
| Transportprotokoll (TP) für große Datenmengen | ●                         |

## # Unterstützte Plattformen

Der emotas CAN Multi-Protokoll Stack kann mit und ohne Betriebssystem verwendet werden.

Er unterstützt eine Vielzahl von CAN-Controllern und CPU-Typen: Bosch, Infineon, Intel, NXP, ST Microelectronics, Texas Instruments, Atmel, Renesas, Microchip uva.

Die Liste der unterstützten Hardwareplattformen wird ständig erweitert:

<https://www.emotas.de/info/controller>

Sollte Ihre Plattform noch nicht verfügbar sein, sprechen Sie uns an. Die Entwicklung weiterer Treiberpakete ist problemlos möglich.

## # Lieferumfang

- ▶ CANopen/J1939/RawCAN Protokoll Sourcecode (ANSI-C kompatibel)
- ▶ Multiprotokoll Treiber
- ▶ CAN DeviceDesigner
  - 1 Named User Lizenz (Projekt Lizenz)
  - 1 Floating Lizenz mit Dongle (Site Lizenz)
- ▶ sofort lauffähige Beispielapplikationen
- ▶ Benutzer- und Referenzhandbuch
- ▶ optionaler Wartungsvertrag
- ▶ optionaler Integrationsupport

## # Lizenzierung

- ▶ Standortlizenz ohne RuntimeGebühren (Royalty free; inkl. 24 Monate Telefon und E-MailSupport)
- ▶ Projektlizenz (Royalty free; inkl. 6 Monate Telefon und E-MailSupport)

## # Bestellinformation

- ▶ 5031-10 | Multiprotokoll J1939/CANopen Slave Stack
- ▶ 5004-11 | MP J1939/CANopen Master/Slave Stack
- ▶ 5004-12 | MP J1939/CANopen Manager Stack

Alle Multiprotokoll-Stack-Varianten sind auch als FD, Multiline oder FD & Multiline verfügbar.

## << CANopen Schulungen 2025 <<

Unsere CANopen-Schulungen vor Ort richten sich vornehmlich an Entwicklungsingenieure und Systemintegratoren. Sie erhalten Einblicke in die Funktions- und Arbeitsweise des CAN-basierenden, höheren Protokolls CANopen.

Für Entscheidungsträger hingegen ist die Schulung eine Möglichkeit, CANopen kennenzulernen und zu entscheiden, ob es für ihre Anwendungen geeignet ist.

### Kommende CANopen-Schulungen in Merseburg:

>> Donnerstag, 27. März & Donnerstag, 13. November

### online CANopen-Schulungen:

>> Donnerstag, 24. April & Dienstag, 9. Dezember

#### << Schulungsinhalte <<

- Grundlagen CAN
- Einführung zu CANopen und CANopen FD
- Gerätemodell und Objektverzeichnis
- Kommunikationsdienste (PDO, SDO, USDO, NMT, Emergency, usw.)
- Überblick über verschiedene CANopen Erweiterungen (CANopen Manager, Layer setting services (LSS), CANopen Redundancy)
- Demonstration von CANopen Diensten an realen Geräten
- Ressourcenbedarf von CANopen-Geräten
- Überblick über die CANopen Geräte- und Applikationsprofile
- Implementierungshinweise zu CANopen

**jeweils 9-17 Uhr | €370,- pro Teilnehmer**

Frühbuchern gewähren wir bis 5 Wochen vor dem Termin 15% Rabatt.

Gern erstellen wir für Sie individuelle In-House- oder remote-Schulungen, wobei wir Ihre gewünschten Fachthemen und den jeweiligen Kenntnisstand der Teilnehmer berücksichtigen. Wir beraten Sie gern.

## << CAN Hardware (Auswahl) <<

**CAN-Interface-Hardware erlaubt den Zugriff von PC-Anwendungen auf CAN-Netzwerke. Wählen Sie für Ihre Anwendung aus den Bereichen J1939, CANopen (FD) oder EnergyBus das optimale CAN-Interface entsprechend dem Leistungsbedarf. Für die Anwendung der Interfaces mit unseren emotas Software Tools fragen Sie uns – wir beraten Sie gern.**

Ein Wechsel zwischen verschiedener CAN-Hardware ist problemlos möglich. Die meisten CAN-Interfaces stehen für viele unterschiedliche Systemarchitekturen zur Verfügung. Die Hersteller der von uns angebotenen Hardwareprodukte bieten höchste Qualität und vielfach einen weitreichenden Support für Ihre speziellen Anwendungsgebiete.

### SYS TEC electronic GmbH

#### << USB-CANmodul1 <<

- High-Speed-CAN Interface
- 11 bit ID und 29 bit ID CAN Unterstützung
- 9-pin D-SUB Steckanschluss
- Diagnose-LEDs
- CAN-Interface nach ISO 11898-2



### EMS Wünsche

#### << EtherCAN 3.0 <<

- Ethernet/CAN Gateway, 2xCAN, RS232, USB Host, SD Card Slot
- CAN jeweils galvanisch getrennt
- 454 MHz ARM9 CPU (Freescale I.MX287)
- Web Server
- 24 V Versorgung
- Plattform für lokale Steuerungsanwendungen



## Kvaser AB

### << Kvaser U100 <<

- USB-CAN/CAN FD Interface, einkanalig
- 11 bit ID und 29 bit ID CAN Unterstützung
- Galvanische Trennung
- 9-pin D-Sub Steckanschluss



## Kvaser AB

### << Kvaser Leaf v3 <<

- unterstützt CAN FD, bis 8 Mbit/s
- 11 bit ID und 29 bit ID CAN Unterstützung
- Galvanische Trennung
- 9-pin D-SUB Anschluss USB „A“
- unterstützt SocketCAN



## << J1939 Stack <<

**Der J1939 Protokollstack ermöglicht eine einfache und schnelle Entwicklung von J1939 Geräten. Der Stack umfasst die gesamte Funktionalität, die für eine Integration von J1939 in eigene Geräte notwendig ist.**

Der J1939 Protokollstack ermöglicht mit der vorhandenen Funktionalität für die Integration von J1939 in eigene Geräte die einfache und schnelle Entwicklung von J1939 Geräten. Die komfortable API erlaubt die einfache Auswertung von empfangenen Daten. Der J1939 Stack ist mit einem definierten Treiberinterface ausgestattet. Das Treiberinterface gewährleistet hohe Flexibilität, weil die Adaption des Protokoll Stacks an vorhandene oder neue CAN-Controller oder CPU-Typen leicht möglich ist. Damit ist ein unkomplizierter Austausch der Hardwareplattform gewährleistet. Die Verwendung verschiedener Echtzeitbetriebssysteme (RTOS) sowie Linux (SocketCAN, can4linux), Windows und QNX ist möglich. Mehrere Beispiele gehören zum Lieferumfang, sodass der Einstieg in die Nutzung des J1939 Stacks leicht fällt. Für die Nutzung der Funktionen steht zudem das Referenzhandbuch mit Beschreibung der Funktionsparameter bereit.

### << Konfiguration <<

Um den Ressourcenverbrauch gering zu halten, ist der J1939 Stack weitgehend konfigurier- und skalierbar. Die vielfältigen Möglichkeiten zur Konfiguration erlauben die individuelle Anpassung für das Kundenprodukt. Die Einstellung für diese Features erfolgt mit dem grafischen Konfigurationstool CAN DeviceDesigner. Das Tool ermöglicht die Konfiguration der PGNs und das Mapping der SPNs über eine integrierte Datenbank.

### << Eigenschaften <<

- Transportprotokolle (BAM und CMTD) für große Datenmengen
- Einstellung der Knotenadresse: fest oder über Address Claiming
- Zeitüberwachung von zyklischen J1939-Nachrichten
- Zyklisches Senden und Empfangen von J1939 Nachrichten
- Address Claiming Protokoll wird in vollem Umfang unterstützt
- Dynamische Konfiguration der PGN und SPNs zur Laufzeit

### # Unterstützte Plattformen

Der emotas J1939 Protokoll Stack kann mit und ohne Betriebssystem verwendet werden. Er unterstützt eine Vielzahl von CAN-Controllern und CPU-Typen: Bosch, Infineon, Intel, NXP, ST Microelectronics, Texas Instruments, Atmel, Renesas, Microchip uva.

Die Liste der unterstützten Hardwareplattformen wird ständig erweitert:

<https://www.emotas.de/info/controller>

Sollte Ihre Plattform noch nicht verfügbar sein, sprechen Sie uns an. Die Entwicklung weiterer Treiberpakete ist problemlos möglich.

### # Lieferumfang

- ▶ Protokoll Sourcecode in ANSI-C (MISRA-konform)
- ▶ 1939 Treiber
- ▶ J1939 DeviceDesigner
  - 1 Named User Lizenz (Projekt Lizenz)
  - 1 Floating Lizenz mit Dongle (Site Lizenz)
- ▶ sofort lauffähige Beispielapplikationen
- ▶ Benutzer- und Referenzhandbuch
- ▶ optionaler Wartungsvertrag
- ▶ optionaler Integrationsupport

### # Lizenzierung

- ▶ Standortlizenz ohne RuntimeGebühren (Royalty free; inkl. 24 Monate Telefon und E-MailSupport)
- ▶ Projektlizenz (Royalty free; inkl. 6 Monate Telefon und E-MailSupport)

### # Bestellinformation

- ▶ 5028-10 | J1939 Stack Pack
- ▶ 5028-10-ML | J1939 Stack Pack MultiLine
- ▶ 2028-xx | J1939 Treiber
- ▶ 2028-xx-ML | J1939 Treiber MultiLine
- ▶ 0100-40 | Integrationsupport

## « J1939 FD Stack »

**Der emotas J1939 FD Stack beinhaltet den klassischen J1939 Stack. Sowohl zur Kompilier- als auch zur Laufzeit kann zwischen J1939 und J1939 FD gewählt werden.**

Der J1939 FD Protokollstack ermöglicht mit der vorhandenen Funktionalität für die Integration von J1939 FD in eigene Geräte die einfache und schnelle Entwicklung von J1939 FD Geräten nach J1939-22. Eine komfortable API erlaubt die einfache Auswertung von empfangenen Daten. Der J1939 FD Stack ist mit einem definierten Treiberinterface ausgestattet. Das Treiberinterface gewährleistet hohe Flexibilität, weil die Adaption des Protokoll Stacks an vorhandene oder neue CAN-Controller oder CPU-Typen leicht möglich ist. Damit ist ein unkomplizierter Austausch der Hardwareplattform gewährleistet. Die Verwendung verschiedener Echtzeitbetriebssysteme (RTOS) sowie Linux (SocketCAN, can4linux), Windows und QNX ist möglich. Mehrere Beispiele gehören zum Lieferumfang, sodass der Einstieg in die Nutzung des J1939 FD Stacks leicht fällt. Für die Nutzung der Funktionen steht zudem das Referenzhandbuch mit Beschreibung der Funktionsparameter bereit.

### « Konfiguration »

Um den Ressourcenverbrauch gering zu halten, ist der J1939 FD Stack in weiten Teilen konfigurier- und skalierbar. Es werden nur die jeweils für die Applikation benötigten Programmteile übersetzt und eingebunden. Dabei unterstützt die J1939 FD Library alle relevanten J1939 Transportprotokolle und optional Diagnosedaten (Diagnostic Messages). Vordefinierte SLOT-Funktionen unterstützen zudem bei der Konvertierung der Messwerte in das Übertragungsformat und umgekehrt. Durch die komplette Implementierung der von J1939 FD definierten Kommunikationsmechanismen können sich die Entwickler vollständig auf die Applikation konzentrieren.

### « Eigenschaften »

- Transportprotokolle (BAM und CMTD) für große Datenmengen
- feste Knotenadresse oder Address Claiming
- Zyklisches Senden/Empfangen und Zeitüberwachung von J1939-Nachrichten
- Dynamische Konfiguration der PGN und SPNs zur Laufzeit
- MPG-Transport

### # Unterstützte Plattformen

Der emotas J1939 FD Protokoll Stack kann mit und ohne Betriebssystem verwendet werden. Er unterstützt eine Vielzahl von CAN-Controllern und CPU-Typen: Bosch, Infineon, Intel, NXP, ST Microelectronics, Texas Instruments, Atmel, Renesas, Microchip uva. Die Liste der unterstützten Hardwareplattformen wird ständig erweitert:  
<https://www.emotas.de/info/controller>

Sollte Ihre Plattform noch nicht verfügbar sein, sprechen Sie uns an. Die Entwicklung weiterer Treiberpakete ist problemlos möglich.

### # Lieferumfang

- ▶ Protokoll Sourcecode in ANSI-C (MISRA-konform)
- ▶ J1939 FD Treiber
- ▶ J1939 DeviceDesigner
  - 1 Named User Lizenz (Projekt Lizenz)
  - 1 Floating Lizenz mit Dongle (Site Lizenz)
- ▶ sofort lauffähige Beispielapplikationen
- ▶ Benutzer- und Referenzhandbuch
- ▶ optional Optional Multi-Line Support für Gateways oder Mehrkanalgeräte
- ▶ optionale UDS Server Erweiterung verfügbar
- ▶ optionaler Wartungsvertrag
- ▶ optionaler Integrationssupport

### # Lizenzierung

- ▶ Standortlizenz ohne RuntimeGebühren (Royalty free; inkl. 24 Monate Telefon und E-MailSupport)
- ▶ Projektlizenz (Royalty free; inkl. 6 Monate Telefon und E-MailSupport)

### # Bestellinformation

- ▶ 5028-10-FD | J1939 FD Stack Pack
- ▶ 5028-10-FD-ML | J1939 FD Stack Pack MultiLine
- ▶ 2028-xx-FD | J1939 FD Treiber
- ▶ 2028-xx-FD-ML | J1939 FD Treiber Multiline

## << J1939 DeviceDesigner <<

**Der J1939 DeviceDesigner ist ein einfach zu bedienendes Tool zum schnellen und kostensparenden Entwurf von J1939-Geräten. Mit wenigen Klicks fügt der Anwender aus vordefinierten Datenbanken die PGN- und SPN-Konfiguration des Gerät zusammen und spezifiziert die Eigenschaften der Signale.**

Der J1939 DeviceDesigner erzeugt daraus die PGN- und SPN-Konfiguration und Initialisierungsfunktionen in C sowie eine Gerätedokumentation.

Zusätzlich konfiguriert der J1939 DeviceDesigner anhand der Geräteeigenschaften den J1939 Stack und das J1939 Treiberpaket optimal.

### << PGN- und SPN-Konfiguration <<

Die PGN- und SPN-Konfiguration wird entsprechend der integrierten Datenbank und Eingaben als C-Quellcode-Datei generiert, welche in die Applikation als Interface zum Protokollstack mit eingebunden wird. Entsprechend der Konfiguration kann auf die Variablen der PGNs direkt als C-Variable aus der Applikation zugegriffen werden.

### << Stack Konfiguration <<

Entsprechend den Einstellungen und der Definition der PGNs und SPNs wird eine Konfigurations- und eine Initialisierungsdatei als C-Quellcode-Datei erstellt, die sicherstellt, dass nur die tatsächlich verwendeten Dienste des J1939-Stacks kompiliert und initialisiert werden. Somit ist der Speicherbedarf des J1939 Stacks jeweils ressourcen-optimal konfiguriert.

### << Gerätedokumentation <<

Für die PGN-Konfiguration mit allen Einstellungen und Beschreibungen sowie für die weiteren geräte-spezifischen Einstellungen wird eine Gerätedokumentation im HTML- und Text-Format erstellt.

Die generierte Dokumentation spiegelt die aktuellen Einstellungen und die aktuelle Implementierung wider und ist somit stets auf dem neuesten Stand. Die Verwendung der generierten Dokumentation als Teil der Anwenderdokumentation des Geräts oder als Import in Textverarbeitungsprogramme ist möglich.

### # Anforderungen

- ▶ Windows 10 oder höher
- ▶ diverse Linux-Varianten (getestet mit (K)Ubuntu)
- ▶ J1939 Protokollstack von emotas

### # Lieferumfang

- ▶ Software zum Download
- ▶ Handbuch im PDF-Format
- ▶ PGN/SPN Datenbank

### # Lizenzierung

- ▶ Named User Lizenz (Projektlizenz)
- ▶ Floating Lizenz mit Dongle (Standortlizenz)
- ▶ Firmenlizenzen auf Anfrage

### # Bestellinformation

- ▶ 1030-10/11 | J1939 DeviceDesigner Windows/Linux
- ▶ 1030-10-D/11-D | J1939 DeviceDesigner Floating Lizenz Windows/Linux

Evaluierungsversionen des Tools sind auf Anfrage bei emotas verfügbar.

## << J1939 CANinterpreter <<

Das vielseitig einsetzbare Monitoring-Tool CANinterpreter dient zur Überwachung und Aufzeichnung des Datenverkehrs und zur Analyse von CAN-Netzwerken. Mit dem J1939 PlugIn können Nachrichten entsprechend dem J1939 interpretiert und versendet werden.

### << Grundfunktionen <<

- Online-Überwachung des Busverkehrs
- CAN-Daten Interpretation entsprechend J1939
- Aufzeichnung der CAN-Telegramme
- Flexible CAN ID-spezifische Filterung/Triggerung
- einmaliges oder zyklisches Senden von CAN Nachrichten/Sequenzen

### << J1939 PlugIn <<

Die J1939-Interpretation interpretiert alle CAN-Telegramme entsprechend dem J1939 Standard. Alle Nachrichten werden mit Namen sowie Quell- und Zieladresse dargestellt. Die PGNs und die darin enthaltenen SPNs werden detailliert mit allen enthaltenen Werten als physikalische Größen mit Wert und Einheit interpretiert. Neben der im Bild gezeigten linearen Ansicht der Telegramme ist auch ein J1939 Objektview verfügbar, welches alle Signale mit ihren aktuellen Werten darstellt.

Importmöglichkeiten für proprietäre PGNs ergänzen die Interpretation. Zudem werden auch die BAM- und TP-Nachrichtensequenzen vollständig analysiert. Mit dem Monitoring-Tool lassen sich J1939-Telegramme senden und anfordern. Darüberhinaus können ebenfalls Address Claiming Telegramme gesendet werden. Somit ist der CANinterpreter mit J1939-Interpretation die ideale Ergänzung für Entwickler und Nutzer von J1939-Geräten. Durch optionale PlugIns lässt sich der Funktionsumfang erweitern. Die Erstellung zusätzlicher PlugIns auf Kundenwunsch ist möglich. Anwendungsfälle hierfür sind z.B. Service- oder Inbetriebnahmetools für Geräte und Anlagen.

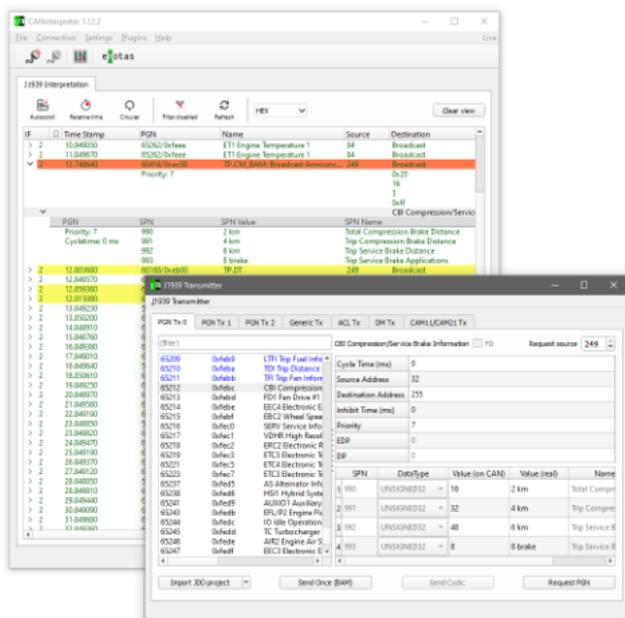
### << Scripting PlugIn <<

Das PlugIn basiert auf QtScript/JavaScript, erweitert um CAN und J1939-spezifische Kommandos. Mithilfe des Tools können eigene Benutzeroberflächen z.B. für Test- und Steuerungsapplikationen erstellt werden. Der Entwurf erfolgt über einen grafischen UI-Designer. Lauffähige Beispiele liegen der Auslieferung bei.

### << PlugIn J1939 Transmitter <<

Dieses PlugIn ermöglicht das Senden von diversen J1939-Nachrichten:

- Standard Daten-PGNs <= 8 Byte Länge
- Request PGN und Address Claiming
- Broadcast PGNs > 8 Byte Länge mittels BAM
- ausgewählte Diagnose-Nachrichten
- J1939 CAM11/CAM21-Nachrichten



### # Bestellinformation

- ▶ 5017-10/11 | CANinterpreter J1939 Windows/Linux
- ▶ 5017-10-D/11-D | CANinterpreter J1939 Floating Lizenz Windows/Linux

## << J1939 Bootloader (CAM11/CAM21) <<

Eine wachsende Zahl von J1939 Geräten benötigt heute Bootloader, um die Firmware im Feld auszutauschen. Die Verwendung eines kompletten J1939-Stacks ist für Bootloader wegen des großen Flash-Speicherbedarfs oft ungeeignet. Der J1939 Bootloader (CAM11/CAM21) von emotas wurde daher mit dem Fokus auf geringen Speicherbedarf, bei gleichzeitiger Berücksichtigung hoher Ausfallsicherheit entwickelt.

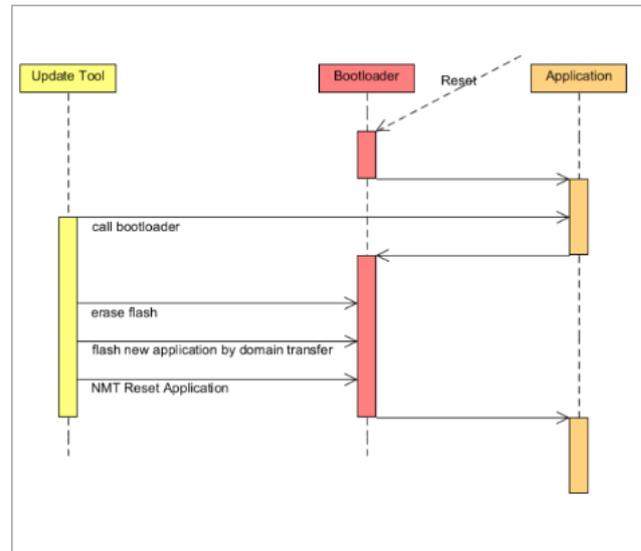
Der Bootloader unterstützt nur die für einen Bootloader notwendigen Dienste. Der emotas Bootloader kommt dadurch mit nur 16 KiB Flash aus und ist dennoch J1939 kompatibel. Verfügbar ist der Bootloader für verschiedene 16- und 32-Bit-Mikrocontroller und er kann einfach an andere Zielsysteme angepasst werden.

Herstellerspezifischer Programmcode kann problemlos integriert werden, da der Bootloader im C Source Code geliefert wird. Somit ist es für die Gerätehersteller z.B. möglich, Zugriffsschutzmechanismen, Verschlüsselung der Firmware oder eine Backdoor in den Bootloader zu integrieren.

Die Applikation, welche mit dem Bootloader übertragen werden soll, muss geringfügig angepasst werden. Dazu müssen die Linker-Einstellungen so geändert werden, dass der Resetvektor auf den Bootloader verweist und der Platz nicht von der Applikation beansprucht wird. Zudem muss ein persistenter RAM (4 Byte) zum Datenaustausch zwischen Bootloader und Applikation bereitgestellt werden, wobei auch alternative Möglichkeiten zur Kommunikation zwischen der Applikation und dem Bootloader unterstützt werden. Der Binärdatei der Applikation wird zudem ein Datenblock vorangestellt, welche die CRC und weitere Information beinhaltet. Hierfür ist ebenfalls der entsprechende Platz im Flash freizuhalten.

### << Highlights <<

- ▶ J1939 kompatibel
- ▶ segmentierter Transfer oder Blocktransfer via CAM11/CAM21
- ▶ geringer Ressourcenbedarf



### # Anforderungen/Voraussetzung

- ▶ ANSI-C Compiler

### # Lieferumfang/Lizenzierung

- ▶ J1939 Bootloader (CAM11/CAM21) Source Code (ANSI-C kompatibel)
- ▶ Treiber für CAN-Controller, CPU und Flashzugriff
- ▶ sofort lauffähige Beispiel-Applikationen
- ▶ Benutzer- und Referenzhandbuch als PDF
- ▶ standortbezogene Firmenlizenz oder Projektlizenz ohne Laufzeitkosten, inkl. 6-24 Monate kostenloser Support je nach Lizenz
- ▶ inkl. 1 Named User Lizenz (Projekt) oder Floating Lizenz (Standort) für CAN UpdateManager

### # Bestellinformation

- ▶ 5039-10 | J1939 (CAM11/CAM21) Bootloader generisch

# << UDS Server/Client Stackerweiterung <<

Unified Diagnostic Services (UDS) ist ein Kommunikationsprotokoll zur Diagnose von ECUs im automobilen Umfeld, welches in ISO 14229 standardisiert ist. Unsere UDS Server/Client Stackerweiterung implementiert UDS auf CAN und setzt dabei auf ISO-TP auf. ISO-TP ist ein internationaler Standard (ISO 15765-2) zur Übertragung von Daten über CAN. Das Protokoll umfasst dabei die OSI-Schichten 3 und 4. Es können dabei Datenpakete von bis zu 4095 Bytes Länge segmentiert übertragen werden.

## << UDS Server/Client Stack für UDSonCAN <<

Der UDS Server/Client Stack ist im C-Quellcode verfügbar und entspricht den MISRA-C:2004-Regeln.  
Der Stack implementiert aktuell folgende UDS-Dienste:

- 10h DiagnosticSessionControl
- 11h ECUReset
- 14h ClearDiagnosticInformation
- 19h Read DTC Information
- 22h ReadDataByIdentifer
- 23h ReadMemoryByAddress
- 27h Security Access
- 28h CommunicationControl
- 2Eh WriteDataByIdentifer
- 2Fh IOControlByIdentifer
- 3Dh WriteMemoryByAddress
- 31h RoutineControl
- 34h Request Download
- 35h Request Upload
- 36h Transfer Data
- 37h Request Transfer Exit
- 3eh TesterPresent
- 85h ControlDTCSetting

Mit sowohl funktionaler als auch physikalischer ISO-TP Adressierung.

Die UDS Server/Client Stack Erweiterung kann mit dem J1939-Stack, dem CANopen-Stack, dem RawCAN-Stack und auch mit dem CAN-MultiProtokollstack beliebig kombiniert werden und ist für eine Vielzahl an Targets verfügbar – von 8 bit Microcontrollern, über die beliebigen ARM Cores, bis zu diversen PC-Plattformen.

## <<UDS Interpretation <<

Eine UDS Interpretation ist optional für unseren CANopen DeviceExplorer verfügbar. Zudem können ausgewählte UDS-Client-Nachrichten mit dem Tool gesendet werden.

### # Anforderungen/Voraussetzung

- ▶ CANopen Stack, J1939 Stack, RawCAN-Stack  
CAN MultiProtokollstack
- ▶ ANSI-C Compiler

### # Lieferumfang/Lizenzierung

- ▶ UDS Server Stack Erweiterung Source Code (ANSI-C kompatibel)
- ▶ Beispielapplikationen
- ▶ Benutzer- und Referenzhandbuch als PDF
- ▶ Firmenlizenz basierend auf Stack-Lizenzform
- ▶ optionaler Wartungsvertrag
- ▶ optionaler Integrationsupport

### # Bestellinformation

- ▶ 1033-10 | UDS Server Addon
- ▶ 1033-10-ML | UDS Server Addon Multiline
- ▶ 1033-11 | UDS Client Addon
- ▶ 1033-11-ML | UDS Client Addon Multiline

## << UDS Bootloader <<

**Eine wachsende Zahl von ECUs benötigt heute Bootloader, um die Firmware im Feld auszutauschen. Die Verwendung eines kompletten UDS-Stacks ist für Bootloader wegen des großen Flash-Speicherbedarfs oft ungeeignet. Der UDS Bootloader von emotas wurde daher mit dem Fokus auf geringen Speicherbedarf, bei gleichzeitiger Berücksichtigung hoher Ausfallsicherheit, entwickelt.**

Der Bootloader unterstützt nur die für einen Bootloader notwendigen Dienste. Der emotas UDSonCAN Bootloader kommt dadurch mit nur 32 KiB Flash aus und ist dennoch UDS kompatibel. Verfügbar ist der Bootloader für verschiedene 16- und 32-Bit-Mikrocontroller und er kann einfach an andere Zielsysteme angepasst werden.

Herstellerspezifischer Programmcode kann problemlos integriert werden, da der Bootloader im C Source Code geliefert wird. Somit ist es für die Gerätehersteller z.B. möglich, Zugriffsschutzmechanismen, Verschlüsselung der Firmware oder eine Backdoor in den Bootloader zu integrieren.

Die Applikation, welche mit dem Bootloader übertragen werden soll, muss geringfügig angepasst werden. Dazu müssen die Linker-Einstellungen so geändert werden, dass der Resetvektor auf den Bootloader verweist und der Platz nicht von der Applikation beansprucht wird. Zudem muss ein persistenter RAM (4 Byte) zum Datenaustausch zwischen Bootloader und Applikation bereitgestellt werden, wobei auch alternative Möglichkeiten zur Kommunikation zwischen der Applikation und dem Bootloader unterstützt werden. Der Binärdatei der Applikation wird zudem ein Datenblock vorangestellt, welche die CRC und weitere Information beinhaltet. Hierfür ist ebenfalls der entsprechende Platz im Flash freizuhalten.

### << UDS-Dienste und Einschränkungen <<

- 10h DiagnosticSessionControl
- 11h ECUReset
- 22h ReadDataByIdentifier
- 27h SecurityAccess
- 31h RoutineControl
- 34h RequestDownload
- 36h TransferData (download only)
- 37h RequestTransferExit

Falls weitere UDS-Dienste nötig sind, können wir gern den UDS-Bootloader in Ihrem Auftrag um weitere UDS-Dienste und spezifische "Update-Sequenzen" erweitern.

### << Highlights <<

- ▶ UDS kompatibel
- ▶ Datentransfer via ISO-TP
- ▶ geringer Ressourcenbedarf

### # Anforderungen/Voraussetzung

- ▶ ANSI-C Compiler

### # Lieferumfang/Lizenzierung

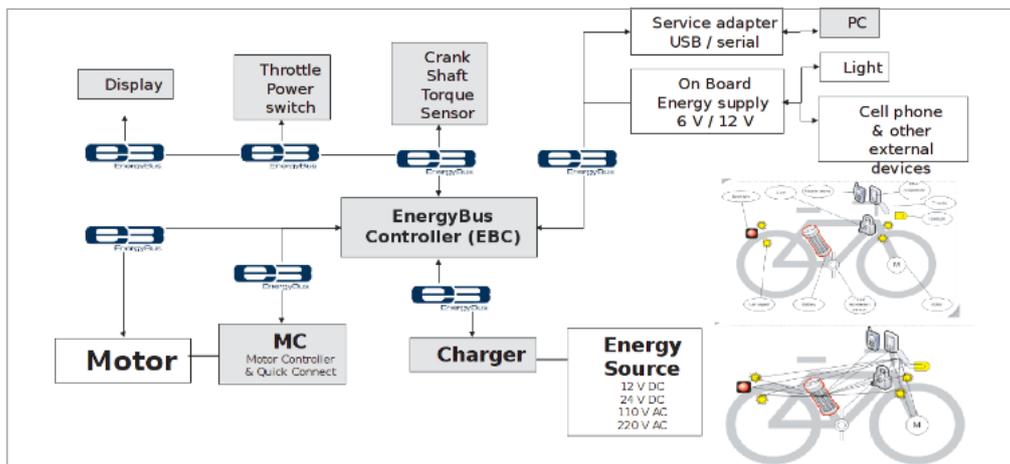
- ▶ UDS Bootloader Source Code (ANSI-C kompatibel)
- ▶ Treiber für CAN-Controller, CPU und Flashzugriff
- ▶ sofort lauffähige Beispiel-Applikationen
- ▶ Benutzer- und Referenzhandbuch als PDF
- ▶ standortbezogene Firmenlizenz oder Projektlizenz ohne Laufzeitkosten incl. 6-24 Monate kostenloser Support je nach Lizenz

### # Bestellinformation

- ▶ 5041-10 | UDS Bootloader generisch
- ▶ inkl. 1 Named User Lizenz (Projekt) oder Floating Lizenz (Standort) für CAN UpdateManager

## << EnergyBus - Einführung <<

EnergyBus ist ein offener Standard für alle elektrischen Komponenten an Elektrofahrzeugen und LEVs – Light-Electric-Vehicles. Durch die Standardisierung wird die Kompatibilität und Sicherheit von LEVs erhöht. EnergyBus besteht aus einer dafür entwickelten Steckverbinder-Familie und einem auf CANopen basierendem Kommunikationsprotokoll.



### << EnergyBus-Kommunikation <<

Zum Datenaustausch setzt EnergyBus auf das bewährte CANopen-Protokoll. Die Spezifika von EnergyBus wurden von den Mitgliedern des EnergyBus e.V. gemeinsam mit dem CAN in Automation e.V. als ein CANopen Applikationsprofil entwickelt und als CiA-454 veröffentlicht. Jedes EnergyBus-Netzwerk erfordert einen EnergyBus-Controller (EBC) als CANopen-Master. Die verschiedenen Komponenten (Akku, Antrieb, Display, Ladeschlosskabel, usw.) sind als CANopen-Slaves mit spezifischen Parametern und Verhalten definiert. Im Applikationsprofil sind neben den übertragenen Daten und Zykluszeiten auch Mechanismen zur Erkennung von Geräten beim Austausch von Komponenten und Kompatibilitätsprüfungen definiert.

### << EnergyBus e.V. <<

Die Standardisierung von EnergyBus wird durch den EnergyBus e.V. geleitet. Zu den Mitgliedern des Vereins zählen verschiedene namhafte Unternehmen aus den Branchen Batteriemangement, Fahrradverleih, Energieversorgung, Antriebstechnik sowie Elektrofahrzeugkomponentenhersteller aus dem In- und Ausland sowie engagierte Privatpersonen.



Der EnergyBus e.V. mit Sitz in Thüringen wurde 2007 in Taiwan gegründet. Seit 2012 ist emotas Software-Entwicklungspartner des EnergyBus-Vereins. emotas engagiert sich aktiv an der Weiterentwicklung des EnergyBus Kommunikationsprotokolls.

Weitere Informationen finden Sie auf  
[www.energybus.org](http://www.energybus.org)

### << CANopen Anforderungen <<

Zur Integration von EnergyBus in Komponenten ist ein CANopen Stack oder das EnergyBus Framework notwendig. Für den EBC und externe Ladegeräte sind zudem CANopen Master Funktionen erforderlich. Für alle Geräte ist die Unterstützung von LSS (Layer Setting Services - CiA 305) zur automatischen Adressierung vorgeschrieben und die Verwendung eines Bootloaders empfohlen. Aufgrund der flexiblen PDO-Konfiguration zur Laufzeit muss die CANopen-Implementierung dynamische Objektverzeichniseinträge unterstützen. Alle diese Anforderungen erfüllt das EnergyBus Framework von emotas. Neben den erforderlichen CANopen-Funktionen beinhaltet das Framework zusätzlich weitere EnergyBus-spezifische Funktionen.

## << EnergyBus Framework <<

Das EnergyBus Framework stellt ein Applikationsinterface für EnergyBus-Geräte, z.B. Elektrofahräder bereit. Die Software ist in ANSI-C programmiert und konform zum Standard CiA 454. Der Einsatz des EnergyBus Frameworks, das auf dem bewährten CANopen Stack basiert, erlaubt die Integration von CANopen-Kommunikationsdiensten über ein anwenderfreundliches User-Interface.

Die Software ist modular aufgebaut. Je nach Anforderung lassen sich die Module für verschiedene Gerätetypen wie Anzeige, Antrieb, Batterie, Load Monitoring Modul und EBC (EnergyBus Controller) zusammenstellen. Der modulare Aufbau ermöglicht die einfache und schnelle Integration der verschiedenen Gerätetypen. Die einzelnen Geräte können auch mit dem EnergyBus Controller kombiniert werden.

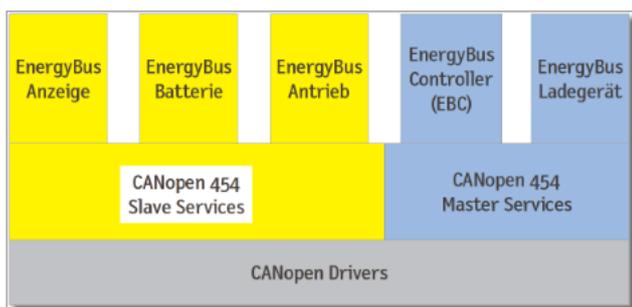
Alle EnergyBus-relevanten Objektverzeichnis-Daten werden abhängig vom Gerätetyp in einem Datenblock bereitgestellt, der von der Applikation gelesen und geschrieben werden kann. Die Applikation wird bei Änderung der Objekte über registrierbare Callback-Funktionen informiert. Das Handling der Kommunikation und der EnergyBus-Zustandsmaschine sowie die Knotennummernvergabe erfolgt automatisch im Hintergrund.

Die Konfiguration erfolgt über den EnergyBus DeviceWizard. Dieses Tool erstellt neben dem Objektverzeichnis auch die zugehörigen EDS-Files und eine Dokumentation des Gerätes.

### << Slave Module <<

Aktuell werden durch das EnergyBus Framework folgende EnergyBus-Komponententypen als Slaves unterstützt:

- Anzeige (HMI)
- Akku
- Antrieb (Motor Controller, MCU)
- Security Device
- Load Monitoring Device

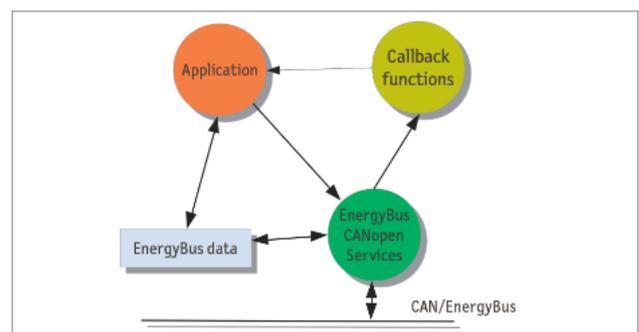


### << Externes Ladegerät <<

Für externe Ladegeräte steht ein separates Modul im Framework zur Verfügung, da ein externes Ladegerät sowohl Master- als auch Slave-Funktionalität beinhaltet. Je nach angeschlossenen Komponenten übernimmt das Modul nur Slave Funktionalitäten (wenn eine EBC im System vorhanden ist), oder die Master Funktionalität im Netzwerk z.B. LSS, Knotennummernvergabe, NMT Kommandos (wenn kein EBC vorhanden ist).

### << EnergyBus Controller <<

Der EnergyBus Controller (EBC) ist das zentrale Steuerelement des Energybus-Netzwerks und arbeitet immer als Master (LSS, NMT und Steuerungsmaster). Dabei ist er u.a. auch für die Vergabe der Knotennummern, die Umschaltung der NMT- und EnergyBus State-Machine und die Steuerung der einzelnen Geräte verantwortlich. Er kann als separates Gerät ausgeführt werden oder mit anderen Komponenten in einem Gerät kombiniert werden. Häufig ist die Kombination mit Antrieb (Motor Controller).



## << Dienstmerkmale <<

| Dienstmerkmal                      | EB Slave Device | EB EBC  | EB ext. Ladegerät |
|------------------------------------|-----------------|---------|-------------------|
| Knotennummer                       | per LSS         | 1       | 127               |
| Virtuelle Geräte                   | 1               | 127     | 127               |
| SDO Server                         | 1               | 1       | 1                 |
| SDO Client                         |                 | 128     | 128               |
| SDO expedited/segmented/block*     | ●/●/○           | ●/●/○   | ●/●/○             |
| PDO Producer                       | 20              | 512     | 512               |
| PDO Consumer                       | 4               | 512     | 512               |
| PDO Mapping                        | static          | dynamic | dynamic           |
| SYNC Producer                      |                 | ●       | ● <sup>1</sup>    |
| SYNC Consumer                      | ●               |         | ● <sup>2</sup>    |
| Emergency Producer                 | ●               | ●       | ●                 |
| Emergency Consumer                 |                 | 127     | 127               |
| Bootup Handling                    |                 | ●       | ●                 |
| Heartbeat Producer                 | ●               | ●       | ●                 |
| Heartbeat Consumer                 | 2               | 127     | 127               |
| NMT Master                         |                 | ●       | ● <sup>1</sup>    |
| NMT Slave                          | ●               |         | ● <sup>2</sup>    |
| LSS CiA305/Fastscan Master         | ●               | ●       | ● <sup>1</sup>    |
| LSS CiA305/Fastscan Slave Handling | ●               |         |                   |

● inklusive ○ optional

1 - nur OHNE EBC im Netzwerk

2- nur MIT EBC im Netzwerk

## # Unterstützte Plattformen

Das EnergyBus Framework kann mit und ohne Betriebssystem verwendet werden. Es unterstützt eine Vielzahl von CAN-Controllern und CPU-Typen. Die Liste der unterstützten Hardwareplattformen wird ständig erweitert:

<https://www.emotas.de/info/controller>

Sollte Ihre Plattform noch nicht verfügbar sein, sprechen Sie uns an. Die Entwicklung weiterer Treiberpakete ist möglich.

## # Lieferumfang

- ▶ EnergyBus LEV Framework ANSI-C Sourcecode
- ▶ sofort einsetzbare Beispiele
- ▶ Handbuch im PDF-Format

## # Lizenzierung

- ▶ Standortlizenz ohne RuntimeGebühren (Royalty free; inkl. 24 Monate Telefon und E-MailSupport)
- ▶ Projektlizenz (Royalty free; inkl. 6 Monate Telefon und E-MailSupport)
- ▶ inkl. EnergyBus DeviceWizard  
Projektlizenz: 1 Named User  
Standortlizenz: 1 Floating Lizenz mit Dongle

## # Bestellinformation

- ▶ 1014-10 | EnergyBus LEV Framework Slave Communication Modul
- ▶ 1014-11 | EB LEV Framework Master/Slave Communication Modul
- ▶ 1014-20 | EnergyBus LEV Framework EBC Modul
- ▶ 1014-21 | EnergyBus LEV Framework Anzeige Modul
- ▶ 1014-22 | EnergyBus LEV Framework Batterie Modul
- ▶ 1014-23 | EnergyBus LEV Framework Antrieb Modul
- ▶ 1014-24 | EnergyBus LEV Framework Voltage Converter Modul

## << EnergyBus DeviceWizard <<

Zur Konfiguration, Parametrierung und Skalierung des EnergyBus Framework steht der EnergyBus DeviceWizard zur Verfügung. Ein Assistent führt den Anwender durch die Auswahl der EnergyBus-Geräte (Batterie, Ladegerät, Antrieb, Anzeige oder EnergyBus-Controller(EBC)). In weiteren Schritten werden die Eigenschaften der einzelnen Komponenten definiert. Detailkenntnisse von CANopen und des EnergyBus-Applikationsprofils sind nicht zwingend nötig und die tiefe Einarbeitung in die Kommunikationsprotokolle entfällt.

Durch die Konfiguration der verwendeten Hardware (Mikrocontroller) wird der Anwender vom EnergyBus DeviceWizard geführt. Dabei sind nur CPU-Typ, CAN-Controller-Typ und einige Timing-Parameter z.B. die Taktfrequenz des Mikrocontrollers auszuwählen und der Treiber wird entsprechend den Einstellungen konfiguriert.

### << Programmausgaben <<

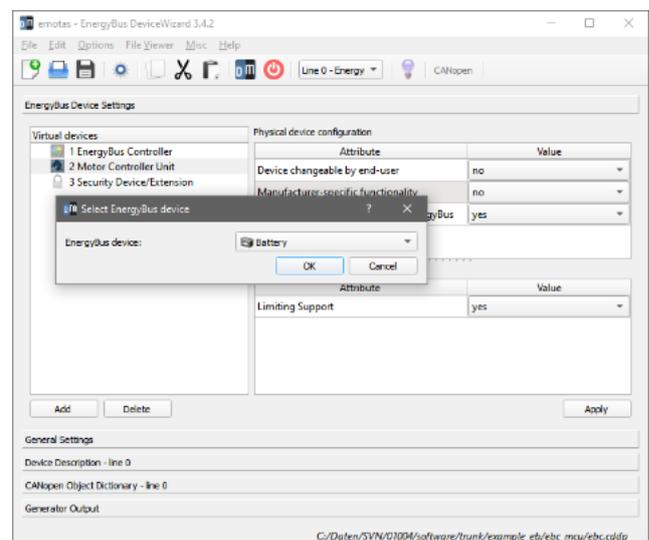
Aus den Benutzereingaben werden .h-Konfigurationsdateien, .c-Applikationscode sowie erforderliche Gerätebeschreibungsdateien in verschiedenen Ausgabeformaten erzeugt. Vor der Generierung prüft der EnergyBus DeviceWizard die Plausibilität der eingegebenen Daten und liefert ggf. entsprechende Warnungen und Fehlermeldungen. Die Konfigurationsdateien stellen das EnergyBus Framework ein und aktivieren durch das Setzen von Compiler-Schaltern nur die jeweils benötigten Funktionalitäten des Frameworks.

.....  
 optimale Konfiguration des EnergyBus Framework  
 mit dem EnergyBus DeviceWizard  
 .....

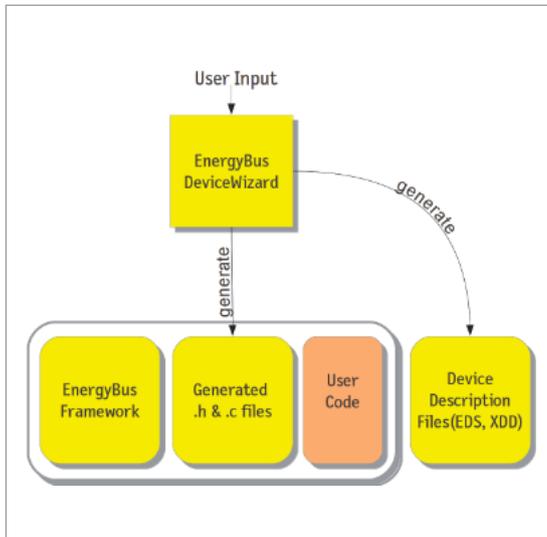
Die passgenau generierten C-Dateien beinhalten das CANopen-Objektverzeichnis, den PDO-Mapping-Code und die Schnittstelle zur Applikation. Das EnergyBus Framework ist so mit den generierten Dateien auf die Anwendung zugeschnitten und hinsichtlich des Speicherbedarfs an RAM und Flash optimiert.

### << Experten-Modus <<

Der Experten-Modus bietet für CANopen-erfahrene Anwender zusätzlich die Möglichkeit, Objekte im herstellerspezifischen Teil des Objektverzeichnisses oder Geräteprofilsegment anzulegen. Dabei können CANopen-Objekte auch aus CODB- oder EDS-Dateien importiert werden. In diesem Modus kann die CANopen-Funktionalität des generierten Applikationscodes erweitert werden, soweit das im Rahmen des Funktionsumfangs des EnergyBus Frameworks möglich ist.



## << Design Flow <<



## << Generierte Dateien <<

- Objektverzeichnis
- EnergyBus Framework-Konfiguration
- EnergyBus Framework-Initialisierung
- Electronic Datasheet (EDS nach CiA 306)
- XDD according to CiA 1311
- Objektverzeichnisdokumentation im HTML-Format
- Auflistung der Objekte im CSV-Format zum Import in Tabellenkalkulationsprogramme
- Beschreibung der Objekte im txt-Format

## # Anforderungen

- ▶ Windows 10 oder höher
- ▶ diverse Linux-Varianten (getestet mit (K)Ubuntu)
- ▶ EnergyBus Framework von emotas

## # Lieferumfang

- ▶ EnergyBus DeviceWizard Software zum Download
- ▶ Handbuch im PDF-Format

## # Lizenzierung

- ▶ Named User Lizenz
- ▶ Floating Lizenz mit Dongle
- ▶ Firmenlizenzen auf Anfrage
- ▶ EnergyBus Framework inkl.1 Floating Lizenz

## # Bestellinformation

- ▶ 1013-10/11 | EnergyBus DeviceWizard Windows/Linux
- ▶ 1013-10-D/11-D | EnergyBus DeviceWizard Floating Lizenz Windows/Linux

Evaluierungsversionen des Tools sind auf Anfrage bei emotas verfügbar.

## << EnergyBus Tool <<

Zur Analyse und für das Monitoring von EnergyBus-Komponenten in LEVs dient das EnergyBus Tool. In der Basisversion zeigt das Tool alle Komponenten und ihre aktuellen Zustände und Betriebsdaten an. Über PlugIns kann das Tool um eine Vielzahl zusätzlicher Funktionen zur Überwachung von EnergyBus-Komponenten ergänzt werden.

Die aktuellen Gerätezustände können in XML-Dateien exportiert werden und später wieder eingelesen werden. Zudem können diese Dateien, die den aktuellen Zustand eines EnergyBus-Netzwerkes (z.B. eines Elektrofahrads) beinhalten, zu Service-Zwecken versendet werden. Der Gerätezustandsbericht kann auch als PDF-Datei erzeugt werden.

### << Optionale Erweiterungen <<

- EnergyBus-Nachrichten-Interpretation
- EnergyBus-Nachrichten-Filterung
- CAN-Rohdatenexport
- Firmware-Update der Komponenten
- EBC-Simulation

| HMI control unit/display            |                                                                                                                                                                                                                                                                                                  |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Device Parameters (fixed)</b>    |                                                                                                                                                                                                                                                                                                  |
| Vendor ID                           | 10066329                                                                                                                                                                                                                                                                                         |
| Product code                        | 1                                                                                                                                                                                                                                                                                                |
| Device information                  | EBC controller supported<br>Passive device<br>Device not changeable by end-user<br>Manufacturer-specific functionality supported<br>Masterless mode supported<br>Bootloader active; no application software is running                                                                           |
| Nominal voltage VDN 1               | 48.00 V                                                                                                                                                                                                                                                                                          |
| Security properties                 | Mechanical locking with charge&lock cable supported   Softlock is supported                                                                                                                                                                                                                      |
| Key class                           | External device; temporary keys required                                                                                                                                                                                                                                                         |
| <b>Actual Values</b>                |                                                                                                                                                                                                                                                                                                  |
| Device control command              | No action                                                                                                                                                                                                                                                                                        |
| Device status                       | nothing active                                                                                                                                                                                                                                                                                   |
| EnergyBus FSA state                 | Compatibility Check                                                                                                                                                                                                                                                                              |
| Device electronic temperature VDN 1 | 23.00 °C                                                                                                                                                                                                                                                                                         |
| Implemented specification versions  | 454.1.2.0<br>454.2.2.0<br>454.3.2.0<br>454.4.2.0<br>454.5.2.0<br>454.6.2.0<br>301.0.4.2<br>302.1.4.1<br>302.3.4.1<br>305.0.2.2<br>0.0.0.10                                                                                                                                                       |
| Security Status                     | No locking mechanism supported<br>Device is not locked locally<br>Manipulation detected<br>Device owns current encryption key<br>Device owns initial encryption key<br>Locking of all virtual devices in the EMS requested<br>All virtual devices in the EMS that support softlocking are locked |

| Property              | Value                                                                       |
|-----------------------|-----------------------------------------------------------------------------|
| Vendor ID             | 10066329                                                                    |
| Product code          | 269746211                                                                   |
| Revision number       | 0.2                                                                         |
| Nominal voltage VDN 1 | 48.00 V                                                                     |
| Security properties   | Mechanical locking with charge&lock cable supported   Softlock is supported |
| Key class             | External device; temporary keys required                                    |

### # Anforderungen

- ▶ Microsoft Windows 7, 8,10 oder diverse Linux-Varianten getestet mit (K)Ubuntu
- ▶ unterstütztes CAN-Interface

### # CAN Schnittstellen

#### Windows®

- ▶ IXXAT
- ▶ SYS TEC USB-CANmodul
- ▶ PEAK PCAN-USB
- ▶ Kvaser CAN-Interfaces
- ▶ EMS Wünsche CAN-Interfaces
- ▶ Vector CAN-Interfaces (XL-Driver)

#### Linux®

- ▶ can4linux
- ▶ SocketCAN

### # Lieferumfang

- ▶ EnergyBus Tool Software zum Download
- ▶ Handbuch im PDF-Format

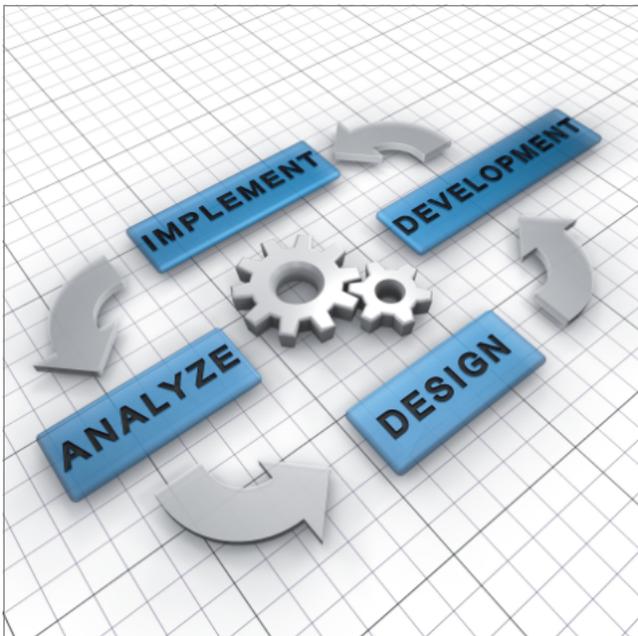
## << Embedded Software-Entwicklung <<

**Embedded Software-Entwicklung mit fundiertem Know-How ist neben dem Produktangebot ein weiterer Schwerpunkt von emotas. Für unterschiedlichste Mikrocontroller von 8 bis 64 Bit, von ARM bis DSP, von single- bis multicore – emotas unterstützt Sie bei Ihrer Geräteentwicklung. Von Anfang an.**

Die Mitarbeiter von emotas haben langjährige Erfahrung in der Software-Entwicklung für embedded-Geräte. In der Medizintechnik, Automatisierungstechnik, in regenerativer oder konventioneller Energietechnik und weiteren Branchen. Neben dem Know-how und der Erfahrung, um die oftmals komplexen Programmierungen durchzuführen, bietet emotas eine breite Produktpalette an Software-Tools, die dies erleichtert.

Die erhöhte Komplexität von Embedded-Software-Entwicklung liegt in der Anforderung des perfekten Zusammenspiels von Hard- und Software. Daher sind alle unsere Embedded Software-Entwickler auch elektrotechnisch ausgebildet.

Expertenkenntnisse in der Programmierung mit C auf Mikrocontrollern und Assemblern, sowie beim Einsatz von Echtzeitbetriebssystemen und modernen Entwicklungswerkzeugen zeichnen emotas und seine Entwickler aus. Bei uns haben Sie von Anfang an einen persönlichen Ansprechpartner und jederzeit direkten Kontakt zum Entwickler des Projekts.



.....  
 Kontaktieren Sie uns für eine erste Diskussion  
 möglicher Lösungen.  
 .....

Vom ersten Gespräch bis zum erfolgreichen Abschluss Ihres Projekts und darüber hinaus – Sie erhalten die passgenaue Unterstützung für alle Entwicklungsschritte:

- gemeinsames Erstellen des Pflichtenheftes
- Entwurf und Design
- Implementierung
- Test
- Integration
- Wartung und Updates

## «« CANopen Integration ««

**Sie entwickeln ein neues Gerät oder Ihre Kunden wünschen eine CANopen-Schnittstelle für ein vorhandenes Gerät? emotas unterstützt Sie mit geballtem CANopen-Know-How und führt die Integration des CANopen Stacks in Ihre Geräte aus. Ihre Anforderungen sind dabei das Maß der Dinge.**

emotas übernimmt die Kommunikationsanbindung Ihrer Geräte. Unsere Entwickler haben langjährige Erfahrungen in der Integration von Feldbus- und insbesondere CANopen-Stacks in embedded-Geräte. Ob Medizintechnik, Automatisierungstechnik, regenerative oder konventionelle Energietechnik oder weitere Branchen – wir kennen die jeweiligen gerätespezifischen Anforderungen, die CANopen mit den Geräteprofilen stellt.

emotas führt die Entwicklungsarbeiten am eigenen Standort durch. Die Übernahme der Hardware erfolgt in einem gemeinsamen Termin - bei Ihnen im Haus oder bei emotas. Dies kann auch mit einer CANopen-Schulung und einer Einweisung in den CANopen Stack und emotas CANopen Tools verbunden werden.



Durch die Mitarbeit in verschiedenen Standardisierungsarbeitsgruppen des CiA e.V. ist emotas stets auf dem aktuellen Stand. Gemeinsam mit Ihrem Know-How finden wir die ideale Lösung für Ihre Anwendung.

.....  
**Profiwissen für kurze Entwicklungszeiten**  
 .....

emotas steht Ihnen bei allen Entwicklungsschritten zur Seite. Wir begleiten Sie vom ersten Gespräch bis zum Ende des Produktlebenszyklus:

- Auswahl des geeigneten Geräteprofils und der passenden Objekte
- gemeinsame Erstellung des Pflichtenheftes
- Entwurf und Design
- Implementierung
- Integration des CANopen-Stacks
- Vorbereitung und Durchführung des Conformance-Tests
- Anpassung an Bootloader
- Wartung und Updates

### «« Durchgeführte Projekte ««

- digitale und analoge I/O-Geräte
- modulare Gerätefamilien
- Positions- und Neigungssensoren
- Antriebe
- maritime Sensoren
- Safety-Sensoren/Aktoren
- medizinische Geräte
- SIIIS Level-2 Sensoren
- Lift-Anwendungen

#### **Ihre Vorteile mit emotas:**

- // **CANopen-Integration durch erfahrene CANopen-Entwickler**
- // **Schnelle Time-To-Market**
- // **Kalkulierbare Entwicklungskosten**
- // **Konzentration auf Kernkompetenzen**

## << EnergyBus Integration <<

**Sie entwickeln neue Komponenten für Leichtfahrzeuge oder haben bereits ein solches Gerät? Sie oder Ihre Kunden wünschen eine EnergyBus-Schnittstelle? Mit unserem Know-How im Bereich EnergyBus finden Sie bei uns beste Unterstützung. Die Integration der erforderlichen EnergyBus-Software in Ihre Geräte führen wir nach Ihren Anforderungen aus.**

emotas übernimmt für Sie die Kommunikationsanbindung Ihrer Geräte. Unsere Entwickler haben Erfahrungen in der Integration von CANopen- und EnergyBus-Stacks in LEV-Komponenten: wir kennen die jeweiligen gerätespezifischen Anforderungen, die der EnergyBus-Standard stellt.

Die Entwicklung führt emotas am eigenen Standort durch. Der enge Kontakt zu Ihnen ist dabei unerlässlich – wir halten Sie auf dem Laufenden. Zur Übernahme der Hardware ist ein gemeinsamer Termin sinnvoll, bei Ihnen im Haus oder bei emotas. Dieser kann mit einer Einweisung in das EnergyBus Framework und die EnergyBus Tools oder mit einer Schulung verbunden werden.



Durch die Mitarbeit in der EnergyBus-Arbeitsgruppe und als Software-Entwicklungspartner des EnergyBus e.V. ist emotas stets auf dem aktuellen Stand. Wir finden die ideale Lösung für Ihre Anwendung.

.....  
**Kompetenter Service für Ihre Anforderungen**  
 .....

emotas bietet Ihnen Unterstützung für alle Entwicklungsschritte vom ersten Gespräch bis zum Ende des Produktlebenszyklus:

- Auswahl des geeigneten Geräteprofils und der passenden Objekte
- gemeinsames Erstellung des Pflichtenheftes
- Entwurf und Design
- Implementierung
- Integration des EnergyBusFrameworks
- Vorbereitung und Durchführung des Conformance Tests
- Anpassung an Bootloader
- Wartung und Updates

### << Projektbeispiele <<

- Batterie und Ladegerät
- Motor Controller
- EnergyBus Controller
- HMI/Display

### Ihre Vorteile mit emotas:

- // **EnergyBus-Integration durch erfahrene EnergyBus-Entwickler**
- // **Schnelle Time-To-Market**
- // **Kalkulierbare Entwicklungskosten**
- // **Konzentration auf Kernkompetenzen**

## << Inbetriebnahme-Unterstützung <<

Sie möchten emotas-Kommunikationsstacks in Ihr Gerät integrieren und schnell starten? Mit der Inbetriebnahme-Unterstützung von emotas stellen Sie sicher, dass die Kommunikation (CAN) der Protokollstacks auf Ihrer Hardware innerhalb kurzer Zeit funktioniert.

Notwendig sind dafür nur Ihre Hardware, Ihre Entwicklungsumgebung, ein Entwickler Ihres Unternehmens und die langjährige Erfahrung der emotas-Experten.

Bei der gemeinsamen Inbetriebnahme wird eines der zahlreichen emotas-Beispielprojekte für übliche Evaluierungsboards genutzt. Die Übernahme der Software auf Kundenhardware funktioniert damit problemlos.

Kleinere „Hürden“ wie zum Beispiel Taktfrequenzeinstellungen, Registeroffsets, Transceiver-Konfigurationen oder ähnliches lassen sich gemeinsam schnell überwinden. Nebenbei erhalten Sie viele Tipps und Hinweise zum eingesetzten Protokoll, der internen Wirkungsweise und zu den Schnittstellen des emotas-Protokoll-Stacks.

Anschließend sind Sie selbständig in der Lage, die Verknüpfung des Kommunikationsstacks mit Ihrer Applikation vorzunehmen.

Die Kombination der Inbetriebnahme mit einer Schulung ist aus unserer Erfahrung heraus die ideale Voraussetzung für einen schnellen und erfolgreichen Projektstart.



### Ihre Vorteile mit emotas:

- // **Anpassung und Integration der Stacks durch unsere Softwareentwickler**
- // **Schnelle Time-To-Market**
- // **Kalkulierbare Entwicklungskosten**
- // **Konzentration auf Kernkompetenzen**

## << Schulungen <<

**Der Bedarf an Schulungen ist besonders hoch, wenn man sich neu mit einer Thematik beschäftigt. Deshalb bietet emotas Seminare in den Bereichen CAN/CANopen, J1939 und EnergyBus an. Durch individuelle Absprache der Themen eignen sich die Schulungen für Anfänger und für Profis, die tiefer ins Detail gehen wollen.**

### << Zielgruppe <<

Die Seminare richten sich an Entwicklungsingenieure, die sich in die Technologien einarbeiten möchten, eine Grundlage für die Arbeit mit emotas-Produkten schaffen wollen oder näheres zu bestimmten Geräte- und Applikationsprofilen wissen möchten. Zudem sind Entscheidungsträger angesprochen, die die technischen Möglichkeiten der Technologien kennenlernen wollen.

### << Inhalte <<

Unsere Schulungen sind individuell auf Ihre Anforderungen und (Vor-) Kenntnisse zugeschnitten. Daher stehen nachfolgende Inhalte beispielhaft:

- Einführung zu CAN
- Einführung zu CANopen (FD)/EnergyBus/ J1939
- Erläuterungen zu emotas Stacks und Erweiterungsmodulen
- Überblick über Geräte- und Applikationsprofile
- Detaillierte Informationen über ein Geräte- oder Applikationsprofil
- Inbetriebnahme anhand von Anwendungsbeispielen

Gern erstellen wir für Sie individuelle Seminarpläne zu verschiedenen Themenbereichen und stimmen diese auf die Kenntnisse Ihrer Mitarbeiter ab.

### << Unsere Trainer <<

Unsere Schulungsleiter haben langjährige Erfahrung im Wissenstransfer und haben sich als Co-Trainer bei Schulungen des CAN in Automation e.V. (CiA) bereits vielfach bewährt. Durch die Bearbeitung vieler Projekte in den verschiedenen Technologien und die Mitarbeit an der CANopen- und EnergyBus- Standardisierung in verschiedenen Arbeitsgruppen des CiA sind die Schulungsleiter stets auf dem aktuellen Stand.

### << Unterlagen <<

Die verwendeten Schulungsunterlagen werden allen Teilnehmern zum Seminar bereitgestellt.

### << Schulungsort-/zeit <<

Wir führen unsere Seminare gern als In-House-Schulung oder in unseren Schulungsräumen durch. Je nach Inhalt sind die Seminare für ein oder zwei Tage konzipiert. Eine Kombination mit einem Integrationsupport kann Ihnen einen schnellen und sinnvollen Start in ein neues Projekt ermöglichen. Die Terminfindung erfolgt in Absprache.



## << 10 Gründe für emotas <<

**Die Zahl der Softwareunternehmen in der Automatisierungsbranche ist groß. Einen geeigneten Partner zu finden, der die individuellen Anforderungen zuverlässig umsetzen kann, fällt da mitunter nicht leicht. Wir geben Ihnen 10 Gründe, sich für die Produkte von emotas zu entscheiden.**

1. Das emotas-Team verfügt über hohe Kompetenz und langjährige Erfahrung im Bereich CAN/CANopen.
2. emotas ist aktiv und beteiligt sich an der Standardisierung im CiA. Zudem ist emotas Mitglied im Technical Committee des CiA e.V. - also immer auf dem neuesten Stand der Entwicklungen.
3. emotas-CANopen-Sourcecode wird auf MISRA-Konformität getestet.
4. emotas-Produkte aller Bereiche werden ständig weiterentwickelt und an aktuelle Standards angepasst.
5. Kunden- und anwendungsbezogene Schulungen zu Stacks und Tools haben Praxisbezug – für Ihren schnellen Einstieg ins Projekt.
6. emotas bietet komfortable grafische Konfigurations- und Installationstools im Lieferumfang der Stacks und eine umfangreiche Toolkette für CANopen, EnergyBus, J1939 und UDS.
7. Kunden profitieren von einem schnellen und kostenlosen First-Level-Support mit direktem Kontakt zu den Entwicklern, denn Service ist uns ein Anliegen.
8. emotas ist Entwicklungspartner des EnergyBus-Vereins und vernetzt mit vielen weiteren Partnern. So können wir Sie bestens beraten.
9. Anwenderfreundliche Lizenzbedingungen gewähren die Nutzung des Sourcecodes für verschiedenste Projekte (ohne Stückzahlizenzen).
10. Unsere Kunden bleiben up-to-date mit individuellen Wartungsverträgen zum fairen Preis – langfristiger Support und Software Updates sind Ihnen so sicher.



<< Partner (Auswahl) <<



**KVASER**



**IntervalZero**



**resolut**

## << Über emotas <<

Die emotas embedded communication GmbH wurde von den drei Software-Entwicklern Andreas Boebel, Torsten Gedenk und Steffen Rose als Unternehmen für Embedded-Software-Entwicklungen gegründet. Sie hat ihren Sitz in zentraler Lage in Mitteldeutschland (Sachsen-Anhalt). Hier entwickelt emotas innovative und hochwertige Produkte für die Industrielle Kommunikation. Stets im Zentrum aller Projekte steht der Kunde, für den emotas ein zuverlässiger Partner ist.

emotas – your embedded solution partner

Thematische Schwerpunkte des Produktportfolios sind CAN, CANopen, CANopen FD, EnergyBus, J1939 und UDS. Neben diesem Produktangebot entwickelt emotas maßgeschneiderte Embedded-Software-Lösungen im Kundenauftrag. Das Leistungsspektrum reicht hier von der ersten Beratung, über die Entwicklung und Inbetriebnahme vor Ort bis hin zur Wartung der Software. Immer im Mittelpunkt steht dabei der Service-Gedanke und der enge Kontakt zum Kunden. Ergänzend bietet emotas Schulungen zu den Kommunikationsprotokollen CANopen, J1939 und UDS sowie zu individuellen Themen an.

emotas ist Mitglied des CAN in Automation e.V. und Entwicklungspartner des EnergyBus e.V. Durch diese Mitarbeit in Arbeitsgruppen und Gremien zur Standardisierung bei CiA und EnergyBus e.V. sind die Mitarbeiter der emotas stets up-to-date. emotas-Produkte entsprechen immer dem neuesten Stand der technischen Entwicklungen.

Die gute und partnerschaftliche Zusammenarbeit mit unseren Kunden ist uns ein wichtiges Anliegen. Wir wissen, dass die erfolgreiche Zukunft der emotas nur gemeinsam mit unseren Kunden gestaltet werden kann. Durch professionellen Service, kompetente Beratung und hochwertige Produkte bieten wir Ihnen maximalen Nutzen und stellen so die Weichen für eine langjährige und erfolgreiche Zusammenarbeit.



## << Kontakt <<

Tel. +49 3461-79416-0  
Fax: +49 3461-79416-10  
service@emotas.de  
<http://www.emotas.de>

**emotas embedded communication GmbH**  
**Fritz-Haber-Str. 9**  
**D-06217 Merseburg**

## «« Haftungsausschluss ««

Die von emotas embedded communication GmbH gelieferten Programme und Dokumentationen werden mit größter Sorgfalt erstellt, in verschiedenen Szenarien getestet und geprüft. Dennoch kann emotas embedded communication GmbH keine Gewähr oder Haftung übernehmen, dass die Software für einen bestimmten Einsatzzweck geeignet und fehlerfrei ist. Technische Daten und Beschreibungen sind keine zugesicherten Eigenschaften im rechtlichen Sinne.

Folgeschäden, welche aufgrund der Benutzung der Software auftreten, sind von jeglicher Haftung oder juristischer Verantwortung ausgeschlossen.

Sämtliche Rechte an den Programmen und Dokumentationen liegen bei emotas embedded communication GmbH. Die Vervielfältigung oder eine auszugsweise Weitergabe an Dritte ist nur mit schriftlicher Genehmigung der emotas embedded communication GmbH gestattet.

Die allgemeinen Geschäftsbedingungen können Sie bei emotas anfordern. Sie sind Bestandteil jedes Angebots der emotas embedded communication GmbH.

Nicht explizit genannte Marken- und Produktnamen sind Warenzeichen der jeweiligen Unternehmen. Technische Änderungen, Irrtümer und Druckfehler vorbehalten.

### Copyright

2025 emotas embedded communication GmbH  
 Fritz-Haber-Str. 9  
 D-06217 Merseburg  
 Deutschland

fon +49 3461 79416-0  
 fax +49 3461 79416-10

service@emotas.de  
 www.emotas.de

### «« Bildnachweise ««

- 3d small people – repairers © Anatoly Maslennikov – Fotolia.com – 3d small people – team © Anatoly Maslennikov – Fotolia.com – Computer learning © kaboliczech – Fotolia.com – Four main steps for a software process cycle © Yabresse – Fotolia.com - 3d small people - info icon © Anatoly Maslennikov - Fotolia.com - Abbildungen CAN Hardware © beim angegebenen Hersteller - AURA-Verleihung © emotas embedded communication GmbH

### «« Warenzeichen ««

Alle verwendeten Warenzeichen und Marken sind Eigentum der jeweiligen Inhaber. CiA®, CANopen® and CANopen FD® are registered Community Trademarks of CAN in Automation e.V.









UDS PGN Master  
Slavetack ISO-TP Tools  
SPN Data Transmission  
Address Claiming Cont  
roller Area Network CA  
NUDS Data Link Layer C  
AN Standardization Tr  
ansport Protocol PGN  
Messages CANopen RAW Car

## << Kontakt

emotas embedded communication GmbH  
Fritz-Haber-Str. 9  
D-06217 Merseburg

mail [service@emotas.de](mailto:service@emotas.de)  
web [www.emotas.de](http://www.emotas.de)  
fon +49 3461 79416-0  
fax +49 3461 79416-10 <<

