

CPL V4

**Lastenheft &**

**Entwicklungshandbuch**

Stand: 09. Mai 2025

Littwin Systemtechnik GmbH & Co. KG  
Bürgermeister-Brötje Str. 28  
D-26180 Rastede  
[www.littwin-systemtechnik.de](http://www.littwin-systemtechnik.de)

E-Mail: [kontakt@littwin-systemtechnik.de](mailto:kontakt@littwin-systemtechnik.de)  
Telefon: 04402 / 972577-0

## Inhalt

1.	Einleitung .....	4
1.1.	Abkürzungen .....	5
2.	Mikrocontroller .....	6
2.1.	RTOS .....	7
2.2.	Belegung Mikrocontroller: .....	8
3.	Software Allgemein .....	9
3.1.	Externer Quellcode .....	10
3.2.	Bootprogramm CPLboot.bex .....	11
3.3.	Hauptprogramm CPL40.bex: .....	12
3.4.	Tasks .....	13
3.4.1.	DOS 1 .....	13
3.4.2.	DOS 2 .....	13
3.5.	Watchdog .....	14
3.6.	Telnet .....	14
4.	Hardwareanbindung .....	16
5.	RS485 Schnittstelle .....	17
5.1.	RS485 intern .....	17
5.2.	RS485 extern .....	17
6.	OPCUA Schnittstelle .....	18
6.1.	OPCUA Objekte .....	19
6.1.1.	CPL .....	19
6.1.2.	Kabelüberwachungen .....	19
6.2.	OPCUA Events .....	22
7.	Datenlogger .....	22
7.1.	Datenbanktabellen .....	24
7.1.1.	Datenvorhaltung .....	24
8.	Webschnittstelle .....	25
8.1.	CGI-Funktionalität .....	25
8.2.	Webdatenpunkte Lesen .....	26
8.3.	Webdatenpunkte Schreiben .....	36
8.4.	JSON Daten anfordern .....	45
8.4.1.	Messwerte .....	45
8.4.2.	Meldungen .....	47
9.	Webseiten .....	48
9.1.1.	Menüleiste (auf allen Seiten) .....	48
9.1.2.	Startseite .....	48
9.1.3.	Systemeinstellung .....	48
9.1.4.	Systemeinstellungen erweitert .....	48
9.1.5.	Digitale Eingänge .....	48
9.1.6.	Kabelüberwachung .....	48
9.1.7.	Analoge Eingänge .....	49

---

9.1.8. Meldungsseite.....	49
10. NTP-Client .....	50

**Markierungen:**

Gelb hinterlegt	Offen, muss noch umgesetzt werden
Cyan hinterlegt	Optional, kann später umgesetzt werden

## 1. Einleitung

Aufgrund neuer Sicherheitsanforderungen wird eine neue Version des Kommunikationsmoduls CPL im TALAS System benötigt.

Die neue CPL V4 soll neue Kommunikationsprotokolle wie z.B. OPCUA verschlüsselt übertragen und der integrierte Webserver muss HTTPS unterstützen.

Dabei soll die CPL so weit wie möglich abwärtskompatibel mit der CPL V3.5 sein.

Es müssen 32 verschiedene Module in max. 4 BGT über den bekannten Systembus angeschlossen werden können.

Die Bedienung / Konfiguration soll über HTTPS und OPCUA realisiert werden. FTP und telnet sind theoretisch möglich, sind aber aufgrund der Sicherheit zu vernachlässigen. Evtl. nur zur Inbetriebnahme oder Wartung vor Ort möglich.

Da die neuesten Kü bereits über einen ungenutzten CAN-Bus verfügen sollte dieser später ebenfalls von der CPL unterstützt werden.

Optional (später) sollen folgende weitere Features integriert werden:

- Zutrittskontrolle TAB und TAT über einen separaten externen Systembus.
- XIO-PM über einen externen Systembus.
- Zweite Ethernet Schnittstelle für die Anbindung von externen zu überwachenden Geräten oder für eine Wartung via Laptop.
- Display zur Anzeige sämtlicher Informationen wie KÜ705.
- Funktion eines M-Bus-Master oder Modbus-Master.
- SNMP V3 Funktionalität

## 1.1. Abkürzungen

Abk.	Bedeutung
BGT	Baugruppenträger
KÜ	Kabelüberwachung (Cu und LWL)
FW	Fernwirk
DP	Datenpunkt
CPL	Zentrales Fernwirkmodul CPL
Cu	Kupfer
LWL	Glasfaser
KÜSS	Störspannungsüberwachung
XIO-PM	Erweiterungsmodul der CPL mit 8 analogen Eingängen
KVz	Kabelverzweiger
TAB	Zutrittskontrolle Basiseinheit (TALAS-Access-Basis)
TAT	Zutrittskontrolle Türsteuerung (TALAS-Access-Tür)
OoS	Out of Service. Meldung ist temporär abgeschaltet.
EMS	Energie-Management-System



## 2. Mikrocontroller

Der Embedded Controller ist ein Anybus IPC@CHIP SC145:

Prozessor:	Cortex A7 core @ 528 MHz Prozessor.
Ethernet:	2x
CAN-Bus:	2x
Serieller Bus:	7x
USB:	2x
I/O:	3x SPI 78x GPIO 4x analog 3x I <sup>2</sup> C 6x PWM
RAM	128 Mbyte Alternativ SC165: 512 Mbyte (nicht verfügbar)
Flash Memory	64 MByte (ca. 32 MByte frei) Alternativ SC165: 256 Mbyte (224 Mbyte frei)
Operating System	IPC@CHIP TROS-LNX (basiert auf Linux)
Operating Temperatur	-25°C bis 80°C



## 2.1. RTOS

Es muss das RTOS ab der Version 2.12 auf dem Mikrocontroller installiert sein.

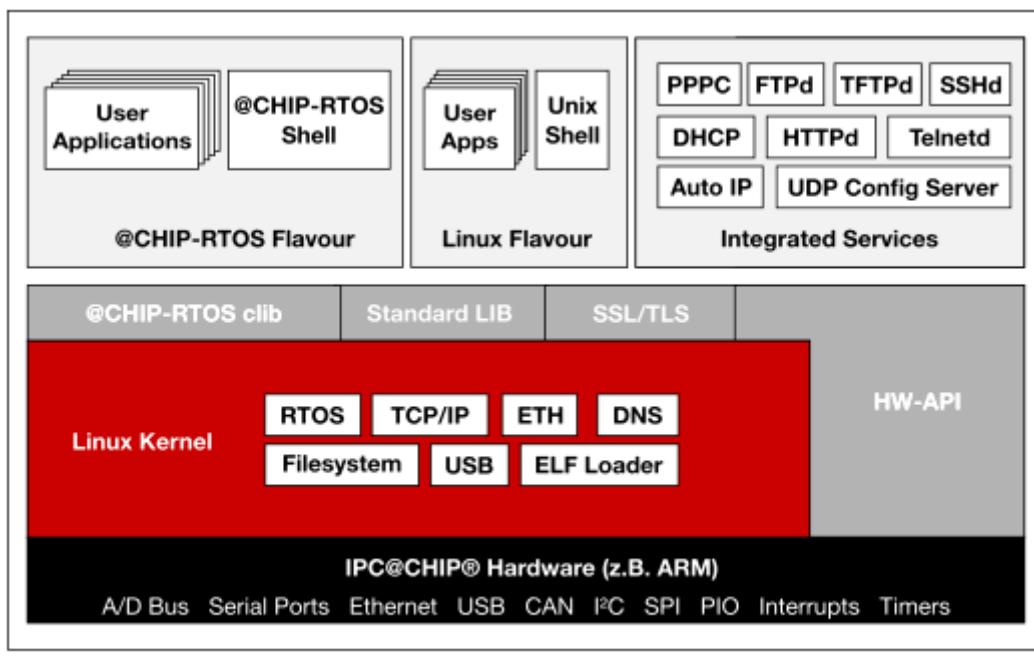
Quelle RTOS:

<https://littwinsystemtechnik.sharepoint.com/:f/r/sites/LittwinSystemtechnik/Freigegebene%20Dokumente/Projekte/K%C3%9CDB%20Netz/CPLV4/RTOS%20LNX?csf=1&web=1&e=fNrJlJ>

Über Telnet und dem Kommando „FLM“ gelangt man (bei Bedarf) zur Linux-Oberfläche. Von der Linux-Oberfläche gelangt man über den Befehl „exit“ wieder zurück zum RTOS.

Für die SD-Karte wird kein Treiber mehr, wie vorher beim SC24, benötigt.

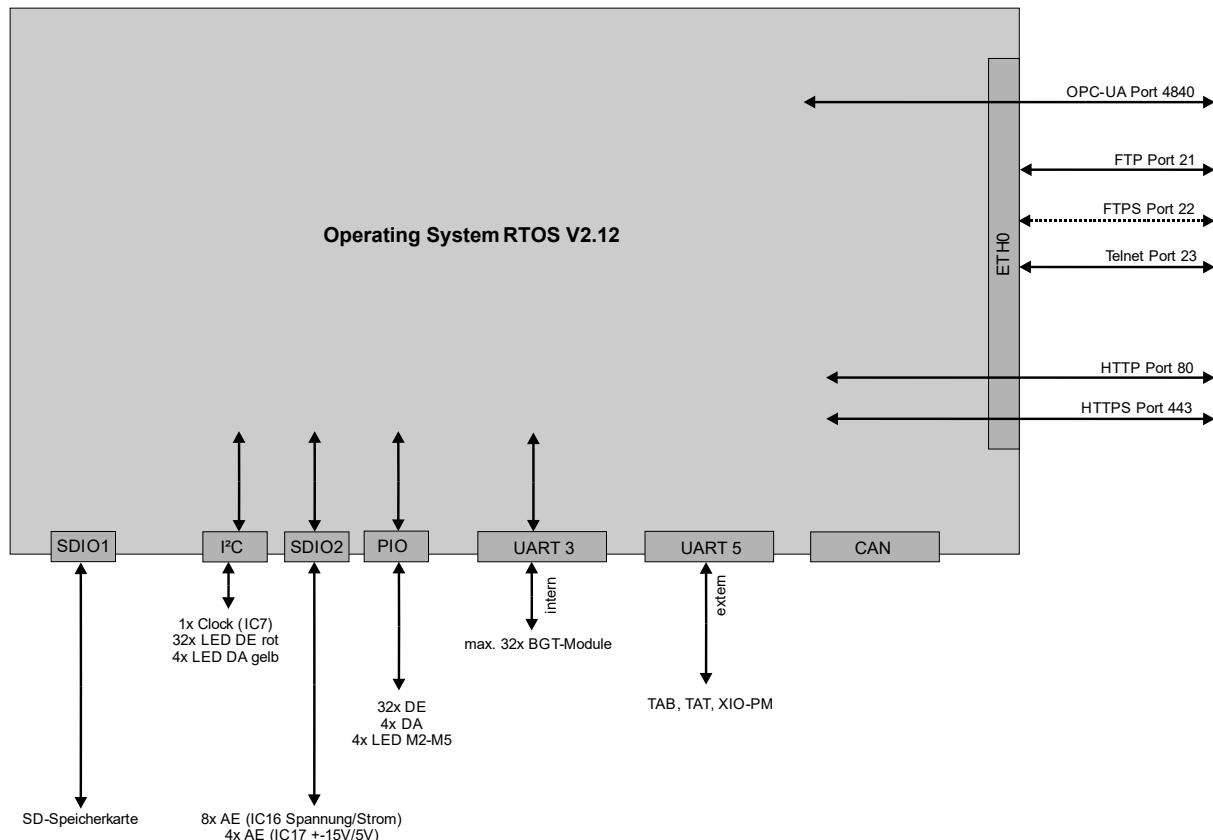
Für die erste Inbetriebnahme (Einstellung IP-Adresse) kann weiterhin das Programm „Chiptool“ von Beck wie beim SC24 verwendet werden.





## 2.2. Belegung Mikrocontroller:

Platine V4.b

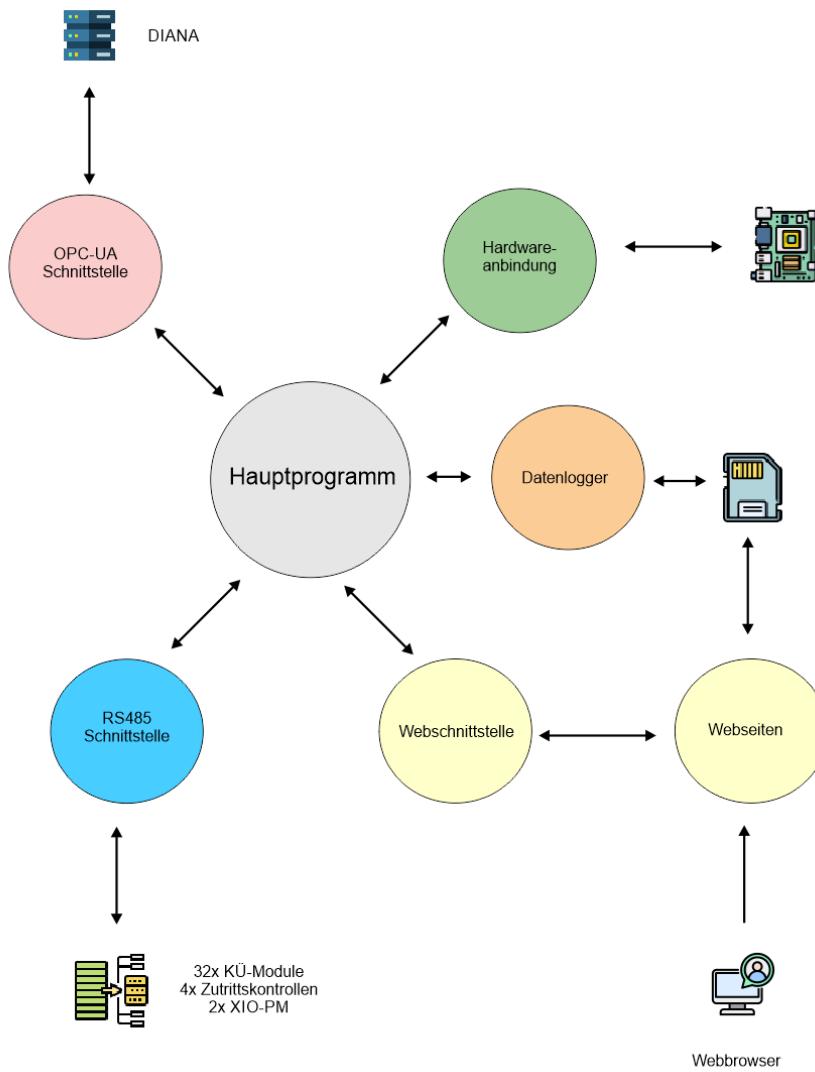




### 3. Software Allgemein

Es wird die Entwicklungsumgebung „ONE – Workbench for IPC@CHIP“ verwendet (basiert auf Eclipse)

Download unter <https://www.hms-networks.com/de/p/abe01001-b-chip-sc145-0002-codesys-v3-variant?tab=tab-support>



Das Hauptprogramm ist auf Laufwerk B im Hauptverzeichnis (CPL40.bex). Die Datenbankdatei data.db befindet sich auf Laufwerk im Ordner Database.



### 3.1. Externer Quellcode

Für die Umsetzung der Funktionen wurden drei externe Projekte in das Ansi-C99-Programm integriert:

#### 1.) Open62541

OPC UA Server. Entwickelt unter anderem vom Fraunhofer ISODB und der Technischen Universität Dresden.

Integrierte Version: 1.3.6 (Mai 2023)

Aktuell verfügbare Version: 1.4.11 (März 2025)

Neue für uns evtl. nützliche Funktionen in der Version 1.4:

Aes256-Sha256-RsaPss unterstützt.

X509 Zertifikationen

OpenSSL 3.0

Integration von NodesetLoader (zur Laufzeit ein Nodeset.xml laden)

#### 2.) Mbed-TLS

<https://www.trustedfirmware.org/projects/mbed-tls/>

Das Projekt implementiert kryptografische Primitive, X.509-Zertifikationsmanipulation sowie die Protokolle SSL/TLS und DTLS. Diese werden im open62541 für eine verschlüsselte opcua-Übertragung benötigt.

Integrierte Version: 2.2.1

Aktuell verfügbare Version: 3.6.3 (März 2025)

#### 3.) SQLite

<https://www.sqlite.org/index.html>

SQLite ist eine C-Bibliothek, die eine kleine, schnelle, eigenständige, hochzuverlässige und vollfunktionale SQL-Datenbank-Engine implementiert. SQLite ist die weltweit am häufigsten verwendete Datenbank-Engine.

Integrierte Version: 3.46.1 (August 2024)

Aktuell verfügbare Version: 3.49.1 (Februar 2025)



### 3.2. Bootprogramm CPLboot.bex

Damit während des Betriebes das Hauptprogramm CPL40.bex mir einer neuen Version überschrieben werden kann wird beim Starten der CPL erst ein kleines Programm mit Namen CPLboot.bex vom Laufwerk A gestartet (Autostart.bat). Es überprüft ob bereits (fälschlicherweise) bereits Programme auf der CPL laufen. Wenn ja -> Reboot, wenn nicht dann startet es nach 90 Sekunden das Hauptprogramm. In dieser Zeit kann das Hauptprogramm auf Laufwerk B kopiert oder geändert werden.

Während des Betriebes überwacht das CPLboot Programm die laufenden Tasks (21 Stück). Sollte ein Task durch eine Störung nicht mehr aktiv sein wird die CPL neu gestartet.



### 3.3. Hauptprogramm CPL40.bex:

Datei	Funktion (Autor)	Bemerkung
CPL40.c	Hauptprogramm (Kai)	Initialisierung Tasks
CPL_V4_IO.c	Hardwareanbindung (Heinz)	Uhrenbaustein I/O's LED's Spannungsüberwachung
DATABASE.c	Datenspeicherung DB (Kai)	Datenlogger
FILE.c	Dateizugriff SD-Karte (Kai)	
RS485.c	Interner Systembus (Kai)	Abfrage Module (Kü) Steuerung Module
WEBSERVER.c	Webschnittstelle (Kai)	Datenpunkte über HTTP lesen / schreiben
OPCUA.c	OPCUA Funktionalität (Kai)	OPCUA Server initialisieren OPCUA Server konfigurieren OPCUA Datenpunkte erstellen / vererben OPCUA schreiben / lesen
Open62541.c	OPCUA Funktionsbibliothek (Heinz)	SDK-Sourcecode vom OPC-UA Schnittstelle von <a href="https://www.open62541.org/">https://www.open62541.org/</a> Zusammengestellt von Heinz inkl. der Integration des Nodeset von der Deutschen Bahn AG
SQLite3.c	Datenbankbibliothek	SDK-Sourcecode von SQLite <a href="https://www.sqlite.org/">https://www.sqlite.org/</a> in der Version 3.46.1
Ntp.c	Zeitserver	Basiert auf einer Vorgabe von Beck IPC



### 3.4. Tasks

#### 3.4.1. DOS 1

Kleines CPL Bootprogramm.

#### 3.4.2. DOS 2

Hauptprogramm

Taskname	Priorität (Speicher)	Bemerkung
SYST	26 (16384 kByte)	Unbenutzt
R485	27 (16384 kByte)	Lesen / Schreiben interner Systembus (Kabelüberwachungen)
OPC1	25 (16384 kByte)	OPCUA Kommunikation
IOX1	27 (16384 kByte)	Hardwareansteuerung I/O
EVAL	30 (16384 kByte)	Hardwareauswertung
DABA	39 (16384 kByte)	Datenbank
WEBC	40 (32768 kByte)	Webschnittstelle



### 3.5. Watchdog

Der Watchdog ist im Modus 3 (RTOS) aktiviert.

### Hinweis:

Modus 1 (User) war während der Entwicklung nicht kompatibel mit der Datenbank wenn eine Abfrage länger als eine Sekunde dauerte. Hier evtl. später Abhilfe mit DB-Abfrage mit einem Callback. Er wird nur im Main-Task zweimal pro Sekunde in der While-Schleife getriggert. Wenn der Watchdog 1000ms nicht getriggert wird, wird ein Reboot durchgeführt.

### 3.6. Telnet

Im Quellcode können alle Tasks eine Message ans Telnet senden. Dafür wurde die Funktion `print_task_message(string quelle, string message, int typ)` erstellt. Über den Typ können folgende Formatierungen übergeben werden:

<b>Typ</b>	<b>Typedef</b>	<b>Farbe</b>
Info	MES_TYP_INFO	Grün
Warnung	MES_TYP_WARN	Gelb
Fehler	MES_TYP_ERROR	Rot
System	MES_TYP_SYSTEM	Magenta
Debug	MES_TYP_DEBUG	Blau

Beim Starten der CPL erscheint über telnet der Status:

Danach erscheinen die Meldungen in der Form **Zeitstempel | Quelle | Typ Meldung:**



```
07:56:07|SNTP |info  New Time NTP 1: 16.10.2024 07:56:07 Timez.: UTC+2
07:56:07|SNTP |system Setting system clock
07:56:07|MAIN |system Heartbeat 16.10.24 Firm 2 Serial 77461 RTOS 531
07:57:25|WEBSER|info  File B:\WEBS\Service\Start.ACP open OK
07:57:25|WEBSER|info  File close
07:57:25|WEBSER|info  File B:\WEBS\Service\Start.ACP open OK
07:57:26|WEBSER|info  File close
07:57:28|WEBSER|info  Unknown placeholder in URL: OFF_1=1
07:57:28|WEBSER|info  File B:\WEBS\Service\KUEconfig.ACP open
07:57:28|WEBSER|info  File close
07:57:32|WEBSER|info  File B:\WEBS\Service\kueData.js open
07:57:32|WEBSER|info  File close
07:57:32|WEBSER|info  File B:\WEBS\4000values\slot8.json open
07:57:39|WEBSER|info  Unknown placeholder in URL: slot=8
07:57:39|WEBSER|debug Grenzwert Iso Slot 8 = 0
07:57:39|RS485|info  New float 16.00 for slot 9
07:57:39|RS485|info  Answer 9
07:57:39|WEBSER|info  File B:\WEBS\Service\kueDetail.HTML open
07:57:39|WEBSER|info  File close
07:57:39|WEBSER|info  File B:\WEBS\Service\kueData.js open
07:57:39|WEBSER|info  File close
07:57:39|WEBSER|info  File B:\WEBS\4000values\slot8.json open
08:01:07|SNTP |info  New Time NTP 1: 16.10.2024 08:01:07 Timez.: UTC+2
08:01:07|SNTP |system Setting system clock
08:01:07|MAIN |system Heartbeat 16.10.24 Firm 2 Serial 77461 RTOS 531
```



## 4. Hardwareanbindung

### **LED-Frontblende:**

Betrieb: Immer An. Festverdrahtet mit 5V

Online: Aus = OPCUA Server außer Betrieb  
Blink langsam = OPCUA Server bereit  
Blink schnell = mindestens ein OPCUA-Client mit Server verbunden  
Dauer an = Softwareabsturz

Meldung: Aktuell keine Funktion

Alarm: Aktuell keine Funktion

Link: Blink nur beim Starten der CPI kurz auf. Soll noch von MIK angepasst werden  
(Anschaltung statisch auf Platine)



## 5. RS485 Schnittstelle

### 5.1. RS485 intern

Systembus intern:

Schnittstelle:	UART 3
Baudrate:	9600
Parität	0
Wortlänge:	8
Stopbit:	1
TX enable:	Aus
Flowcontrol:	Aus
Modus:	Highactive
DMA:	Aus

In der Datei rs485i.c wird regelmäßig jeder Slot überprüft, ob ein neues Modul gesteckt worden ist. In dieser Funktion wird ein Zähler zurückgesetzt. Passiert dies 10 Minuten nicht wird die CPL neu gestartet.

Hintergrund ist das es in der Entwicklungsphase vorgekommen ist das die KÜ's nicht mehr abgefragt worden sind (Pfeile auf den Display zeigten keine Kommunikation auf dem Systembus)

### 5.2. RS485 extern

Noch nicht vorhanden



## 6. OPCUA Schnittstelle

Die CPL ist ein OPCUA-Server. Es können mehrere OPCUA-Clients eine Verbindung zum Server herstellen. Eine Verbindung kann benutzerabhängig, passwortgeschützt und verschlüsselt sein.

Jede Verbindung kann über den Benutzer unterschiedliche Rechte besitzen. Ein Client kann Objekte abonnieren, den Abfrageintervall selbst bestimmen und vom Server zur Verfügung gestellte Methoden aufrufen.

Es wird das SDK **open62541** (siehe auch open62541) vom Fraunhofer IOSB verwendet.

Zum Testen sollte der von der Deutschen Bahn empfohlenen OPCUA-Client „Unified Automation UaExpert - The OPC Unified Architecture Client“ verwendet werden.

Der Port für OPCUA muss 4840 sein.

Der OPCUA-Server muss für die Deutsche Bahn AG folgende Eigenschaften besitzen:

- TLS-Verschlüsselung
- 256bit Verschlüsselung im Modus **Basic256Sha256**
- Security Mode **SignAndEncrypt**
- Authentifizierung erfolgt über einen Namen und ein Passwort. Benutzername: **db** Passwort frei wählbar vom Hersteller
- Das Zertifikat kann selbst signiert werden
- Der OPCUA-Server muss jedes Client-Zertifikat zulassen. Es darf kein „Reject“ von Client Zertifikaten erfolgen.
- Jedes Client-Zertifikat muss ohne vorheriges „Trust“ zugelassen werden (Bei einigen SDKs ist standardmäßig das Trust-Verfahren aktiviert.)

Weitere Vorgaben vom Auftraggeber zur OPCUA-Schnittstelle sind im Anhang A zu finden. Dort sind die Objekte, Namespaces, usw. aus dem Nodeset beschrieben.

Hinweis Kai zur Verschlüsselung:

Laut open62541 muss anscheinend noch ein externes SSL SDK integriert werden. Hierfür kann laut open62541 Open SSL, LibreSSL oder Mbed-TLS verwendet werden. Eine Integration von Mbed-TLS war erfolgreich.

### Nützliche Links

Dokumentation

<https://www.open62541.org/doc/open62541-master.pdf>

Zusatz

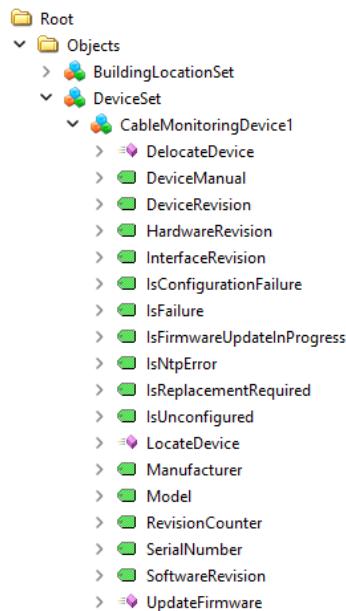
<https://www.open62541.org/doc/0.3/protocol.html>



## 6.1. OPCUA Objekte

### 6.1.1. CPL

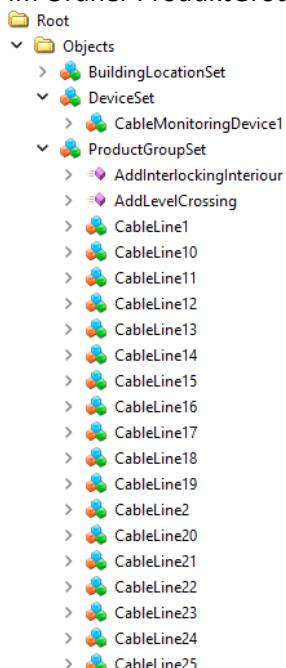
Im Ordner DeviceSet befindet sich die **CPL** unter CableMonitoringDevice1



Etwa die Hälfte der Datenpunkte werden beschrieben.

### 6.1.2. Kabelüberwachungen

Im Ordner ProduktGroupSet befinden sich 32 Kabelüberwachungen:





In jedem Kabelüberwachungsordner befindet sich ein Ordner „CableLineAddIn“ sowie 10 weitere CableLink und 10 weitere CableNodeOrdner. Die jeweils 10 Ordner soll nur der Dokumentation dienen und brauchen laut DB erst einmal nicht gefüllt werden.

Der Ordner CableLineAddIn beinhaltet folgende Datenpunkte / Methoden:

- CableLine
  - CableLineAddIn
    - > ConfigureResistanceMeasurementInterval
    - > ConfigureTdrInterval
    - Id
  - InsulationResistanceThreshold\_AB\_Earth
    - > ResistanceValue
    - > pResistanceThresholdError
    - > pResistanceThresholdFailure
    - InsulationResistance\_AB\_Earth
    - InsulationResistance\_A\_Earth
    - InsulationResistance\_B\_Earth
    - > IsMaintenanceMode
    - > LoopResistance
  - LoopResistanceThreshold
    - > ResistanceValue
    - > pResistanceThresholdError
    - > pResistanceThresholdFailure
    - Name
    - ResistanceMeasurementInterval
  - > SetMaintenanceMode
  - StatusCable
  - TdrInterval
  - > TriggerCableLineDiagnosticsNow
  - > UpsertInsulationResistanceThreshold
  - > UpsertTdrMeasurementReference



Hinweise:

#### **Datenpunkt StatusCable:**

Wert	Bedeutung	Bemerkung
Ok	Ok	Kabel wird überwacht und ist ok
Deviation	Abweichung	Schleifenfehler
Error	Fehler	Isolationsfehler
Failure	Ausfall	Erdschluss oder Aderbruch
Unknown	Unbekannt	Kabelüberwachung nicht betriebsbereit

#### **Datenpunkt ID**

Name der Kabellinie. Ersten 16 Zeichen werden im Display der Kabelüberwachung angezeigt.  
Lesen und Schreiben.

Der gleiche Inhalt wird auch im Hauptverzeichnis „CableLine1“ im Ordner Identification dargestellt. Die Methode SetIdentification besitzt kein Input und kann somit nicht benutzt werden. Hier ist wahrscheinlich ein Fehler im Nodeset der Deutschen Bahn vorhanden

#### **Datenpunkt Name**

Der Datenpunkt Name kann nur gelesen werden. Über die Webseite der CPL kann er beschrieben werden.

#### **Datenpunkt InsulationResistance\_AB\_Eart**

Isolationswiderstand in MΩ. Nur Lesen.

#### **Datenpunkt LoopResistance**

Schleifenwiderstand in kΩ. Nur Lesen.

#### **Datenpunkt ResistanceMeasurementInterval**

Schleifenmessintervall in Sekunden. Aktuell kann ein Bereich von 6h bis 24h eingetragen werden. In Sekunden 21600 bis 86400. Die eingegebenen Sekunden werden auf Stunden gerundet. Lesen und Schreiben.

#### **Methode ConfigureResistanceMeasurementInterval**

Input Argument: Interval = Schleifenmessintervall in Sekunden

Output Argument: MethodeResult = Good or Bad (wenn nicht im Wertebereich 21600 bis 86400)

#### **Ordner InsulationResistanceThreshold\_AB\_Earth Datenpunkt pResistanceError**

Isolationsgrenzwert in MΩ. Wertebereich 1 bis 200 MΩ. Lesen und Schreiben

#### **Ordner LoopResistanceThreshold Datenpunkt pResistanceError**

Unterer Schleifengrenzwert in kΩ. Wertebereich 1 bis 50 kΩ. Lesen und Schreiben



## 6.2. OPCUA Events

Es sind zwei Simple Events programmiert worden.

1.) TDR-Event nach Vorgabe der Deutschen Bahn

Wenn eine neue TDR-Messkurve von einer Kabelüberwachung über den Systembus von der CPL empfangen worden ist wird diese in zwei Float-Arrays über dieses Event an die OPCUA Clients versendet. Eine TDR-Messung kann über ein Call-Befehl der jeweiligen CableLine im Client ausgelöst werden.

2.) CPL System Event

Hier werden einfache Messages versendet wie z.B. Aderbruch, Erdenschluss, DE 01 ein usw.  
Diese Events entsprechen keiner Vorgabe der DB

Beispiel Uni Hamburg

[https://git.informatik.uni-hamburg.de/OpenDISCO/open62541/blob/81a1ed350e2240c4dbed79d41e93bb5f50de0f9d/examples/tutorial\\_server\\_events.c](https://git.informatik.uni-hamburg.de/OpenDISCO/open62541/blob/81a1ed350e2240c4dbed79d41e93bb5f50de0f9d/examples/tutorial_server_events.c)

Doku open62541

[https://www.open62541.org/doc/1.1/tutorial\\_server\\_events.html](https://www.open62541.org/doc/1.1/tutorial_server_events.html)  
[https://www.open62541.org/doc/master/tutorial\\_server\\_alarms\\_conditions.html](https://www.open62541.org/doc/master/tutorial_server_alarms_conditions.html)

Doku uaExpert

<https://documentation.unified-automation.com/uaexpert/1.5.0/html/event.html>

## 7. Datenlogger

Zur Speicherung der Daten wird eine **SQLite** Datenbank in der Version 3.46.1 verwendet.

Informationen zur Datenbank: [HTTPS://www.sqlite.org](https://www.sqlite.org)

Damit die SQLite Datenbank auf dem Mikrocontroller funktionsfähig ist musste das Multithreading und der „Large File Support“ deaktiviert werden.

```
#define SQLITE_DISABLE_LFS 1
#define SQLITE_THREADSAFE 0
(Datei sqlite3.c Zeile 24 und 25)
```

Die Datenbank verwendet nur eine Datei (data.db) zur Speicherung der Daten. Diese Datei muss aufgrund ihrer möglichen Größe auf dem Laufwerk B gespeichert werden. Dies funktioniert nur wenn das Hauptprogramm ebenfalls auf dem Laufwerk B installiert ist.

Die Datenbank läuft nur mit der folgenden Dateisystemeinstellung für UNIX (siehe Datei database.c):

```
sqlite3_vfs_register(sqlite3_vfs_find("unix-none"), 1);
```

Zur Überprüfung der Datenbank kann die Datei data.db vom Laufwerk B:\database der CPL heruntergeladen werden und mit einem Viewer (z.B. DB-Browser für SQLite) auf einem Windowsrechner geöffnet werden.

## 7.1. Datenbanktabellen

Tabelle	Inhalt
Systems	CPL-Systemeinstellungen
Module	Kabelüberwachungseinstellungen (CableLine)
SimpleEvents	Meldungshistorie
	IdDatapoint: 0-1000: System (z.B. 10=Reboot) 2000-2031: digitale Eingänge 10000-10099: Kü 01 Datenpunkte 10100-10199: Kü 02 Datenpunkte ... 13100-13199: Kü 32 Datenpunkte
Measurements	Messwerte der Module
	Modul: 0-31 Kabelüberwachungen 100-116 analoge Eingänge 200-231 digitale Eingänge
	Typ: 1 = analoger Messwert 2 = digitaler Zustand 3 = Isolationswiderstand 4 = Schleifenwiderstand
Spalte	Measurement: akuteller Messwert Minimum: kleinster Messwert innerhalb des Loggerintervalls Maximum: größter Messwert innerhalb des Loggerintervalls
Tdr_Master	Auflistung aller Tdr-Messungen (mit idTDR autoincrement)
Tdr_Measure	Messwerte der einzelnen Tdr-Messungen (mit idTDR Key)
CableLinks	Namen und Länge der bis zu 9 Kabelstrecken einer Kabelüberwachung CableLine
CableNodes	Namen der bis zu 10 Knotenpunkte einer Kabelüberwachung CableLine
Input_AE	Name, Faktor, Offset und Loggerintervallzeit pro analogen Eingang
Input_DE	Name, Filterzeit, Invertierung und Zählereinstellung pro digitalen Eingang
Sqlite_sequence	DB interne Daten

### 7.1.1. Datenvorhaltung

Die Datenbank wird alle 30 Minuten überprüft. Es werden alle Daten (Messages, Measurements und Tdr) gelöscht die **älter als ein Jahr** sind. (Ausnahme TDR-Referenzkurve)



## 8. Webschnittstelle

Der Webserver wird ausschließlich via HTTPS aufgerufen werden. Hierzu ist in der CHIP.INI Datei auf Laufwerk A folgende Einstellung nötig:

```
[WEB]
ENABLE=1
DRIVE=1
ROOTDIR=WEB
MAINPAGE=INDEX.HTM

SSL_ENABLE=1
SSL_DRIVE=1
SSL_ROOTDIR=WEBS
SSL_VERSION=3
SSL_CACERT0=Littwin-cacert.pem
SSL_KEYFILE=cplv4-key.pem
SSL_SERVERCERT=cplv4-cert.pem
```

Die SSL-Dateien Littwin-cacert.pem, cplv4-key.pem und cplv4-cert.pem wurden von Florian Freytag generiert und befinden sich im TEAMS-Ordner ... TODO

Die Webseiten müssen auf das Laufwerk B (Speicherplatte) im Ordner WEBS gespeichert werden.

Der Webserver ist mit einem Passwort geschützt. Beim ersten Aufruf der Seite CPL wird im Browser ein Benutzer und ein Passwort abgefragt.

In der Entwicklungszeit ist dies **Littwin / Littwin**

Offen ist noch eine Zertifizierung über den Browser damit die Verbindung als gesichert anerkannt wird.

### 8.1. CGI-Funktionalität

Der Quellcode der Webseiten kann HTML und JavaScript enthalten. Die aktuellen Werte einiger Variablen kann über Platzhalter in der Form <%xxxdd%> in den Quellcode geschrieben werden. Immer wenn die Seite neu geladen wird, werden die Platzhalter durch die aktuellen Werte ersetzt. Dies geschieht aber nur beim Aufruf einer xxx.acp Seite, wenn vor dem Seitennamen CPL? gesetzt wird.

Beispiel **CPL?KUEdetail.ACp** ruft die Webseite KUEdetail.ACp auf welche HTML und JavaScript enthalten kann.

Mit dem Aufruf der Seite <https://10.10.0.222/CPL?offline.htm> wird man abgemeldet und die Anmeldeseite erscheint.



Im Verzeichnis B:\webs\service\ befinden sich die Testseiten von Kai die bei der Entwicklung der Webschnittstelle verwendet worden sind. Hiermit lassen sich alle Platzhalter und Events testen.

Hinweis: Eine ACP-Seite mit Platzhalter darf max. nach dem parsen 10.000 Zeichen enthalten. Werden mehr Zeichen für die Variablen benötigt ist das Aufteilen in mehreren Dateien eine Lösung. So können z.B. die Variablen in einzelne .js Dateien aufgeteilt und auf der HTML Seite wieder einbunden werden.

## 8.2. Webdatenpunkte Lesen

Im Script kann nun eine Reihe von Platzhalter gesetzt werden:

Aus der Zeile

```
var kueIso=[<%=KIM80%>,<%=KIM81%>,<%=KIM82%>,<%=KIM83%>];
```

wird nach dem Parsen z.B.

```
var kueIso=[200.00,200.00,200.00,-1.00,-1.00,11.00,-1.00,11.00,-1.00,-1.00,-1.00,
```

Tipp: Bei großen Seiten ist es von Vorteil, wenn die Seite normal als HTML-Seite aufgerufen wird. Innerhalb der Seite sollte dann z.B. ein separates JavaScript als Datei via CPL?Datei eingefügt werden. Dadurch muss der Webserver nicht die ganze HTML-Datei nach Platzhalter durchsuchen. Beim automatischen Reload der Daten (Aktualisierung) muss nur die kleine JavaScript Datei periodisch im Hintergrund neu geladen werden.

Allgemein:

Kontext	Platzhalter	Ausgabeformat
MAC Adresse Ethernet Schnittstelle	SEMxx xx=Schnittstelle (1, 2)	%u %u %u %u %u %u
IP-Adresse	SEI01	%s
Subnetz (Netzmaske)	SES01	%s
Gateway	SEG01	%s
Zeitstempel der internen Uhr	SCL01	%02i.%02i.%02i %02i:%02i:%02i Uhr
Name CPL, ID	SAN01	%s Max. 20 Zeichen



Applikationsversion	SAV01	%f
HTML-Meldungsliste BETA	SAM01	<tr><td>idMessage</td><td>id... </tr> Spalte 1: Meldungs-ID Spalte 2: Datenpunkt-ID Spalte 3: Zeitstempel Spalte 4: Meldungstext Spalte 5: Zustand (1/0)
OPCUA Zustand Server	SOS01	%i 0=Server außer Betrieb 1=Server betriebsbereit
OPCUA Anzahl verbundener Clients	SOC01	%i Anzahl aktiver Clients
OPCUA Name Nodeset	SON01	%s
NTP: Aktiv	STA	%i 0=aus 1=ein
NTP: IP Adresse Server 1	STP01	%s
NTP: IP Adresse Server 2	STP01	%s
NTP: IP Adresse Server 3	STP01	%s
NTP: Zeitzone	STT01	%i 0=UTC +2=Berlin
Applikationsversion	SAV	%s String Versionsnummer
Startzeit CPL	SSS	%s String Zeitstempel Startzeitpunkt



## Kabelüberwachung Allgemein:

Kontext	Platzhalter	Ausgabeformat
Status 0=Offline 1=Online	KSOxx xx=Slot 80-83 = BGT 1 bis 4	%d %d,%d,%d,%d,%d,%d,%d,%d
ID, Name (max. 16 Zeichen)	KSIxx xx=Slot 80-83 = BGT 1 bis 4	'%s' '%s','%s','%s','%s','%s','%s','%s','%s'
Aderbruch 0=kein Alarm 1=Aderbruch	KSCxx xx=Slot 80-83 = BGT 1 bis 4	%d %d,%d,%d,%d,%d,%d,%d,%d
Erdschluss 0=kein Alarm 1=Erdschluss	KSGxx xx=Slot 80-83 = BGT 1 bis 4	%d %d,%d,%d,%d,%d,%d,%d,%d
Version Faktor 100	KSVxx xx=Slot 80-83 = BGT 1 bis 4	%d %d,%d,%d,%d,%d,%d,%d,%d
Messspannungs-ausfall 0=ok, 1=96V Fehler	KSSxx xx=Slot 81-84 = BGT 1 bis 4	%d %d,%d,%d,%d,%d,%d,%d
Kein Kabel (Funktion noch nicht geklärt)	KSNxx xx=Slot 80-83 = BGT 1 bis 4	%d %d,%d,%d,%d,%d,%d,%d,%d
Event Bootvorgang 0=Normalbetrieb 1=Im Bootvorgang	KSBxx xx=Slot 80-83 = BGT 1 bis 4	%d %d,%d,%d,%d,%d,%d,%d,%d
Event Schleifenmessung 0=Normalbetrieb 1=Schleifenmessung aktiv	KSXxx xx=Slot 80-83 = BGT 1 bis 4	%d %d,%d,%d,%d,%d,%d,%d,%d
Event TDR-Messung 0=Normalbetrieb 1=TDR-Messung aktiv	KSYxx xx=Slot 80-83 = BGT 1 bis 4	%d %d,%d,%d,%d,%d,%d,%d,%d
Event Abgleich 0=Normalbetrieb 1=Abgleich in Aktion	KSZxx xx=Slot 80-83 = BGT 1 bis 4	%d %d,%d,%d,%d,%d,%d,%d,%d
Knoten ID (Node ID)	KNyxx y=Knoten (0-9) xx=Slot (01-32)	'%s'
Teilstrecke ID (Link ID)	KPyxx y=Teilstrecke (0-8) xx=Slot (01-32)	'%s'



Teilstrecke Länge (Link Length)	KQyxx y=Teilstrecke (0-8) xx=Slot (01-32)	%f
Teilstrecke Status (Link Status)	KUyxx y=Teilstrecke (0-8) xx=Slot (01-32)	%i



## Kabelüberwachung Isolationsmessung:

Kontext	Platzhalter	Ausgabeformat
Messwert Einheit MOhm	KIM xx=Slot 80-83 = BGT 1 bis 4	%f03.2  %f03.2,%f03.2,%f03.2,%f03.2,%f03.2,%f03.2,%f03.2,%f03.2
Gültig 0=Nein (eingefroren) 1=Aktueller Messwert	KIV xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d
Messwertalarm 0=kein Alarm 1=Isolationswert zu gering	KIAxx xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d
Grenzwert Einheit MOhm	KIGxx xx=Slot 80-83 = BGT 1 bis 4	%f03.2  %f03
Verzögerung Einheit sek.	KIDxx xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d
Speicherintervall für Isolations- und Schleifenwiderstand	KILxx xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d
Overflow 0=Messwert<=200 1=Messwert>200	KIWxx xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d
Messreihe letzte 4000 Messungen Isolation (ein Monat)	KIZxx xx=Slot	JSON {Timestamp, Isolation, Gültigkeit} Daten werden in einen File-Rohling im Ordner 4000values übergeben. Hinweis: Es können keine weiteren Platzhalter in der Datei benutzt werden!
Messreihe letzte 4000 Messungen Schleifenwert (ein Monat)	KISxx xx=Slot	JSON {Timestamp, Schleifenwert, Gültigkeit} Daten werden in einen File-Rohling im Ordner 4000values übergeben. Hinweis: Es können keine weiteren Platzhalter in der Datei benutzt werden!

## Kabelüberwachung Schleifenmessung:

Kontext	Platzhalter	Ausgabeformat
Messwert Einheit kOhm	KRM xx=Slot 80-83 = BGT 1 bis 4	%f02.3  %f02.3,%f02.3,%f02.3,%f02.3,%f02.3,%f02.3,%f02.3
Unterer Messwertalarm 0=kein Alarm 1=Messwert<Grenzw.	KRA xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d



Unterer Grenzwert Einheit kOhm	KRGxx xx=Slot 80-83 = BGT 1 bis 4	%f02.3  %f02.3,%f02.3,%f02.3,%f02.3,%f02.3,%f02.3,%f02.3,%f02.3
Oberer Grenzwert Einheit kOhm	KRHxx xx=Slot 80-83 = BGT 1 bis 4	%f02.3  %f02.3,%f02.3,%f02.3,%f02.3,%f02.3,%f02.3,%f02.3,%f02.3
Schleifenintervall Einheit Stunden h	KRIxx xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d,%d



## Kabelüberwachung TDR:

Kontext	Platzhalter	Ausgabeformat
Entfernung Bruchstelle Einheit Meter	KTFxx xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d,%d
Dämpfung Kabel Einheit?	KTDxx xx=Slot 80-83 = BGT 1 bis 4	%f02.1  %f02.1,%f02.1,%f02.1,%f02.1,%f02.1,%f02.1,%f02.1
Kabelgeschwindigkeit Einheit m/µs	KTSxx xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d,%d
Pulsweite der letzten Messung	KTPxx xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d,%d
Verstärkung der letzten Messung	KTAxx xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d,%d
Trigger der letzten Messung	KTExx xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d,%d
TDR-Funktion aktiviert 0=ausgeschaltet 1=eingeschaltet	KTXxx xx=Slot 80-83 = BGT 1 bis 4	%d  %d,%d,%d,%d,%d,%d,%d,%d
Zeitpunkt letzte TDR- Messung	KTLxx xx=Slot 80-83 = BGT 1 bis 4	'%s'  '%s','%s','%s','%s','%s','%s','%s','%s'
Letzte TDR-Messung als Messreihe <b>Nicht mehr verwenden. Dafür aus der TDR Liste die letzte Messung nehmen siehe 8.4.1</b>	KTZxx xx=Slot	JSON {Entfernung, Pegel} Hinweis: Es können keine weiteren Platzhalter in der Datei benutzt werden!
Messreihe der Referenz	KTRxx xx=Slot	JSON {Entfernung, Pegel} Hinweis: Es können keine weiteren Platzhalter in der Datei benutzt werden!



## Digitale Eingänge:

Kontext	Platzhalter	Ausgabeformat
Zustand	DESxx xx =Nr Eingang 1-32 80-83 = 8er Blöcke 80 = DE01 bis DE08 81 = DE09 bis DE16 82 = DE17 bis DE24 83 = DE25 bis DE32	%i (1/0) %i,%i,%i,%i,%i,%i,%i
Name	DENxx xx =Nr Eingang 1-32 80-83 = 8er Blöcke	'%s' '%s','%s','%s','%s','%s','%s','%s','%s'
Zählerstand	DECxx xx = Nr Eingang 1-32 80-83 = 8er Blöcke	%f %f,%f,%f,%f,%f,%f,%f,%f
Filterzeit	DEFxx xx = Nr Eingang 1-32 80-83 = 8er Blöcke	%f %f,%f,%f,%f,%f,%f,%f,%f
Gewichtung	DEGxx xx = Nr Eingang 1-32 80-83 = 8er Blöcke	%i %i,%i,%i,%i,%i,%i,%i,%i
Invertierung	DEIx xx = Nr Eingang 1-32 80-83 = 8er Blöcke	%i (0/1) %i,%i,%i,%i,%i,%i,%i
Zähler aktiv	DEZxx xx = Nr Eingang 1-32 80-83 = 8er Blöcke	%i %i,%i,%i,%i,%i,%i,%i
Out of service	DEAxx xx = Nr Eingang 1-32 80-83 = 8er Blöcke	%i %i,%i,%i,%i,%i,%i,%i



## Analoge Eingänge:

Kontext	Platzhalter	Ausgabeformat
Messwert  Hinweis: AE16 (-96V) benötigt die aktuelle Backplane	AAVxx xx = Nr. 1-8 oder 9-16 für Systemspannungen  9: +15V 11: +5V 15: -15V 16: -98V Messspannung  17: ADC Temperatur 18: CPU Temperatur	%0.6f
Name	ACNxx xx=Eingang 1-18	,%s'
Einheit	ACUxx xx=Eingang 1-18	,%s'
Faktor	ACFxx xx=Eingang 1-18	%f
Offset	ACOxx xx=Eingang 1-18	%f
Gewichtung	ACSxx xx=Eingang 1-18	%i 0-1000
Logger Intervall	ACLxx xx=Eingang 1-18	%i [minuten]



## Digitale Ausgänge:

Kontext	Platzhalter	Ausgabeformat
Aktueller Zustand	DASx x=Eingang 1-4	%i 0 / 1
Name	DANx x=Eingang 1-4	,%s'



### 8.3. Webdatenpunkte Schreiben

Zum Senden von Befehlen oder Schreiben von Parameter muss der entsprechende Wert über eine URL zum Webserver gesendet werden:

Beispiel:

```
location.href="CPL?KUEdetail.ACP&KTT1=1;
```

KTT1=1 starten zum Beispiel als Befehl die TDR-Messung der KÜ im Slot 1 im Automatikmodus (siehe Tabelle unten)

Allgemein:

Kontext	Platzhalter	Eingabeformat
Name CPL	SNNA=%s	%s Name der CPL Max. 20 Zeichen
IP-Adresse	SEI01=%s	%s=IP-Adresse
Subnetz (Netzmaske)	SEI02=%s	%s=Subnetz
Gateway	SEI03=%s	%s=Gateway
Zeitstempel der internen Uhr	CLK%i=%iA-%iB-%iC	%i=0=Datum A=Jahr B=Monat C=Tag  %i=1=Uhrzeit A=Stunde B=Minute C=Sekunde
Datenbank vollständig leeren	DEDB=%i	1=leeren
Datum stellen	CLK00=%i-%i-%i	%i=Jahr (0 bis 40) %i=Monat (1 bis 12) %i=Tag (1 bis 31) Beispiel 24-10-18 für 18. Oktober 2024
Uhrzeit stellen	CLK01=%i-%i-%i	%i=Stunde (0 bis 23) %i=Minute (0 bis 59) %i=Sekunde (0 bis 59) Beispiel 17-15-58 für 17:15:58 Uhr



Datenbank Konfiguration löschen	DBC1=1	1=leeren
Datenbank Meldungen löschen	DBC2=1	1=leeren
Datenbank Messwerte Logger löschen	DBC3=1	1=leeren
NTP: IP Adresse Zeitserver	SNIP%i=%s	%i=Nummer NTP Server (1 bis 3) %s=String IP Adresse
NTP: Zeitzone	SNTZ=%i	%i=Timezone (0 bis 9)
NTP: Aktive	SNAC=%i	%i=0 aus %i=1 ein
Neustart der CPL Reboot	BOOT=1	4 Sekunden verzögert



## Kabelüberwachung Allgemein:

Kontext	Platzhalter	Eingabeformat
ID, Name	KID%i= %[^\n] %i=Slot	String
Display einschalten	KSD%i=1 %i=Slot 99=alle	
Firmwareupdate	KSU%i=1 %i=Slot 99=alle	
Backup Konfiguration alle Module	KSBI=1	
Restore Konfiguration	KSR%i=1 %i=Slot	
Node ID	KNA%i=%i=%s %i=Slot (0 – 31) %i=Node (0 – 9) %s= String Text ID	
Link ID	KNB%i=%i=%s %i=Slot (0 – 31) %i=Link (0 – 8) %s= String Text ID	
Link Länge	KNC%i=%i=%f %i=Slot (0 – 31) %i=Link (0 – 8) %s= Float Länge	



## Kabelüberwachung Isolationsmessung:

Kontext	Platzhalter	Eingabeformat
Grenzwert Einheit MOhm	KL_%i=%f %i=Slot	Float
Verzögerung Einheit sek.	KD_%i=1 %i=Slot	Float
Speicherintervall für Isolations- und Schleifenwiderstand	KLO%i=%i %i =Slot %i =Minuten	

## Kabelüberwachung Schleifenmessung:

Kontext	Platzhalter	Eingabeformat
Unterer Grenzwert Einheit kOhm	KR_%i=%f %i=Slot	Float
Oberer Grenzwert Einheit kOhm		
Schleifenintervall Einheit Stunden h	KRI%i=%i %i=Slot	Integer
Messung starten	KS_%i=1 %i=Slot	



## Kabelüberwachung TDR-Messung:

Kontext	Platzhalter	Eingabeformat
TDR-Messung im Automatikbetrieb starten	KT%i=1 %i=Slot	
TDR-Messung mit eingestelltem Parameter (Verstärkung, Pulsw W und Trigger) starten	KT%i=0 %i=Slot	
Eine TDR-Messreihe als Referenz markieren	KTR%i=%i %i=Slot %i=idTDR (siehe Kapitel 8.4 TDM=x)	
TDR-Funktion 0=ausgeschaltet 1=eingeschaltet	KTX%i=%i %i=Slot	0=ausschalten 1=einschalten
Dämpfung in dB	KTD%i=%f %i=Slot	
Kabelgeschwindigkeit in m/µs	KTS%i=%i %i=Slot	
Trigger	KTE%i=%i %i=Slot	



## Digitale Eingänge:

Kontext	Platzhalter	Eingabeformat
Name	DEN%i=%s %i = Nr DE (1-32) %s = String Name	32 Zeichen
Invertierung	DEI%i=%i %i = Nr DE (1-32) %i = 1/0 (ein/aus)	
Filterzeit	DEF%i=%f %i = Nr DE (1-32) %f = ms	
Gewichtung	DEG%i=%i %i = Nr DE (1-32) %i = 0-1000	
Zählerstand	DEC%i=%f %i = Nr DE (1-32) %f = Zählerstand	
Zähler aktiv	DEZ%i=%i %i = Nr DE (1-32) %i = 1/0 (ein/aus)	
Out of service	DEA%i=%i %i = Nr DE (1-32) %i = 1/0 (offline/online)	



## Digitale Ausgänge:

Kontext	Platzhalter	Bemerkung
Schalten	DAS%i=%i %i = Eingang 1-4 %i = Schalten 1 / 0	
Name	DAN%i=%s %i = Eingang 1-4 %s = String Name	32 Zeichen



## Analoge Eingänge:

Kontext	Platzhalter	Eingabeformat
Name	ACN%i=%s %i = Nr AE (1-8) %s = String Name	32 Zeichen
Einheit	ACU%i=%s %i = Nr AE (1-8)	24 Zeichen
Faktor	ACF%i=%f %i = Nr. AE (1-8) %f = Faktor	
Offset	ACO%i=%f %i = Nr. AE (1-8) %f = Offset	
Gewichtung	ACG%i=%i %i = Nr. AE (1-18) %i = 0-1000	
Loggerinterval	ACL%i=%i %i = Nr. AE (1-18) %i = Loggerinterval in Minuten	





## 8.4. JSON Daten anfordern

### 8.4.1. Messwerte

Kontext	Platzhalter	Ausgabeformat
Alle Messwerte eines Zeitraumes	DIA0=v1;v2;v3;b1;b2;b3;q;t  v1.v2.v3=Datum von (z.B. 25.08.31) b1.b2.b3=Datum bis (z.B. 25.09.01) q=Quelle 0-31: Kü 100-107: AE-Eingänge 109: Systemspannung +15V 111: Systemspannung +5V 115: Systemspannung -15V 116: Systemspannung -98V  T=Typ 1: analoger Messwert 2: digitaler Zustand 3: Isolationswiderstand 4: Schleifenwiderstand	JSON {Timestamp, Messwert, Gültigkeit, Minimum, Maximum}  Max. 4000 Werte. Es werden immer 6 Nachkommastellen angezeigt.  t: Zeitstempel m: aktueller Messwert v: Gültig 0=nein, 1=ja i: Minimum a: Maximum
Für jede <b>Stunde</b> ein Minimum-, ein Maximum- und ein Durchschnittswert	DIA1=y  y = siehe oben	JSON {Timestamp, Minimum, Maximum, Durchschnitt}  Max. 3000 Werte. Es werden nur Werte mit Gültig=1 ausgewertet. Es werden immer 6 Nachkommastellen angezeigt.  t: Zeitstempel i: Minimum a: Maximum g: Durchschnitt
Für jeden <b>Tag</b> ein Minimum-, ein Maximum- und ein Durchschnittswert	DIA2=y  y = siehe oben	JSON {Timestamp, Minimum, Maximum, Durchschnitt}  Max. 3000 Werte. Es werden nur Werte mit Gültig=1 ausgewertet. Es werden immer 6 Nachkommastellen angezeigt.  t: Zeitstempel i: Minimum a: Maximum g: Durchschnitt
Eine Liste mit den letzten 100 TDR-Messungen des Slots x	TDM=x  x = Quelle Modul (0-31)	JSON {idTdr, Zeitstempel, Fehlerstelle in Meter, Pulsweite, Geschwindigkeit, Amp}  id: eindeutige TDR ID der Messreihe t: Zeitstempel d: Entfernung der Bruchstelle p: Pulseweite vom Impuls



		s: eingestellten Kabelgeschwindigkeit a: Amplitude des Impulses
Eine Liste aller Messwerte einer TDR Messung	TDR=x  X=id TDR Messreihe	JSON {Entfernung, Pegel}.  d: Distance, Entfernung p: Pegel in mV?

Soll z.B. von der dritten Kü der Schleifenwert von Januar bis Juli mit stündlichen Werten angezeigt werden muss im URL-Link folgendes stehen:

location.href="..../CPL?seite.ACp&DIA1=2025;01;01;2025;07;31;2;4";

(oder <https://10.10.0.222/CPL?Service/empty.acp&TDM=0>)

Daraufhin werden die JSON-Daten gesendet:

```
[  
{"t":"06-02-2025 14:00:00","i":65.000000,"a":65.000000,"g":65.000000},  
 {"t":"06-02-2025 13:00:00","i":65.000000,"a":65.000000,"g":65.000000},  
 {"t":"06-02-2025 12:00:00","i":65.000000,"a":65.000000,"g":65.000000},  
 {"t":"06-02-2025 11:00:00","i":65.000000,"a":65.000000,"g":65.000000}  
]
```



#### 8.4.2. Meldungen

Kontext	Platzhalter	Ausgabeformat
Meldungen (Events) abrufen	MSS1=v1;v2;v3;b1;b2;b3;f;  v1.v2.v3=Datum von (z.B. 25.08.31) b1.b2.b3=Datum bis (z.B. 25.09.01) f=Filter All: Alle CPL: System DE 1: Digitaler Eingang 1 DE 32: Digitaler Eingang 32 Slot 1: Modul 1 Slot 36: Modul 36	JSON {t, s, c, m, i}  Max. 500 Meldungen.  t: Zeitstempel s: Gewicht 0 (leicht) bis 1000 (schwer) c: Farbe in Hex (0=grün, 1000=rot) m: Meldungstext i: Quelle (Filter)

Beispiellink: <https://10.10.0.222/CPL?Service/ae.ACP&MSS1=2025;01;01;2025;2;28;All;>

Antwort:

```
{"t":"2025-02-25 11:14:25","s":0,"c":"#00d200","m":"Modul 08 Isofehler gehend","i":"Slot 8"}  

,{"t":"2025-02-25 11:14:21","s":500,"c":"#ff0000","m":"Modul 08 Isofehler kommend","i":"Slot 8"}  

,{"t":"2025-02-25 10:59:25","s":0,"c":"#00d200","m":"Modul 08 Isofehler gehend","i":"Slot 8"}  

,{"t":"2025-02-25 10:59:21","s":500,"c":"#ff0000","m":"Modul 08 Isofehler kommend","i":"Slot 8"}  

,{"t":"2025-02-25 10:44:24","s":0,"c":"#00d200","m":"Modul 08 Isofehler gehend","i":"Slot 8"}  

,{"t":"2025-02-25 10:44:20","s":500,"c":"#ff0000","m":"Modul 08 Isofehler kommend","i":"Slot 8"}  

,{"t":"2025-02-25 10:29:25","s":0,"c":"#00d200","m":"Modul 08 Isofehler gehend","i":"Slot 8"}  

,{"t":"2025-02-25 10:29:21","s":500,"c":"#ff0000","m":"Modul 08 Isofehler kommend","i":"Slot 8"}  

,{"t":"2025-02-25 10:28:21","s":700,"c":"#ff0000","m":"Modul 09 Aderbruch kommend","i":"Slot 9"}  

,{"t":"2025-02-25 10:28:18","s":0,"c":"#00d200","m":"Modul 09 online","i":"Slot 9"}, {"t":"2025-02-25 10:28:17","s":10,"c":"#00d200","m":"Modul 08 TDR eingeschaltet","i":"Slot 8"} ... usw.
```

## 9. Webseiten

### 9.1.1. Menüleiste (auf allen Seiten)

- Wer ist angemeldet
- Abmeldebutton
- Name der CPL
- Auswahl (Systemeinstellungen, Digitale Eingänge, Kabelüberwachungen)

### 9.1.2. Startseite

- Meldungsliste (min. 20 letzte Meldungen) (aus JSON)
- Kleine BGT-Ansicht (Klick auf Kabelüberwachung)
- Übersicht Status (Spannungen, OPCUA-Verbindungen)

### 9.1.3. Systemeinstellung

- IP-Adresse
- Netzmaske
- Gateway
- Name der CPL
- Neustart
- Uhrzeit (nur lesen)

### 9.1.4. Systemeinstellungen erweitert

- NTP-Zeitserver (Aktivierung, IP-Adressen, Offset)
- OPCUA-Server (Aktivierung, Benutzer & Passwörter, Verschlüsselung) (noch nicht vorhanden)
- Module (Anzeige Versionen, Update, Reset, Sicherung und Rücksicherung) (noch nicht vorhanden)
- Uhrzeit und Datum schreiben

### 9.1.5. Digitale Eingänge

- 32 Zustände
- Pro Eingang Parameter (Invertierung, Filterzeit, Counteraktivierung)

### 9.1.6. Kabelüberwachung

- Aktuelle Messwerte
- Aktueller Zustand (96V Messspannung, Gültig, usw.)
- Befehl-Button Schleifenmessung starten
- ID, Name, Loggerintervall (einstellbar)
- Einstellungen (Isogrenzwert, Schleifengrenzwert, Verzögerungszeit, Intervallzeit Schleife, usw.)
- Messkurve Isolationswert, Schleifenwert (JSON) inkl. Meldungen
- TDR-Messungen (Ausführen, Messkurve vorherige Messungen und Referenz) (JSON)



- 10 Kabelknoten mit Namen (einstellbar) (offen)
- 9 Kabelstrecken mit Namen & Länge (einstellbar) (offen)

#### 9.1.7. Analoge Eingänge

- 8 externe Eingänge (4 Spannung, 4 Strom)
- 8 interne Eingänge (Systemspannungen 5V, +15V, -15V, -98V, sowie Temperatur Prozessor und AD-Wandler)
- Bei den externen AE: Name, Faktor, Offset, Einheit, Loggerinterval einstellbar.

#### 9.1.8. Meldungsseite

- Alle Meldungen mit Filter Zeitraum und Eingang

## 10. NTP-Client

Die CPL muss seine Zeit über NTP-Server abfragen. Für die Deutsche Bahn sind drei NTP-Server abzufragen:

10.174.50.100  
10.174.17.200  
10.174.36.139

Wenn der erste SNTP-Server nicht erreichbar ist, wird der zweite probiert. Danach der dritte.

Alle 300 Sekunden wird die interne Uhr mit dem SNTP-Server abgeglichen.

Es kann eine Zeitzone angegeben werden. Timezone=2 bedeutet das zwei Stunden auf die UTC aufgerechnet wird

Der NTP-Client kann ein- und ausgeschaltet werden (active)

Der Quellcode ist in der Datei ntp.c

Der Aufruf ist im Main-Programm unter cpl40.c