

smartIO MVB

Technische Daten und Beschreibung

Das smartIO MVB ist ein MVB-Class 0 Device, mit dem rückwirkungsfrei der MVB Bus eines Schienenfahrzeugs gemäß EN61375-3 ausgelesen werden kann. Als Class 0 Device nimmt das Modul nicht an der MVB-Kommunikation teil und besitzt selbst keine Vorrichtung, um Signale oder Telegramme auf dem Bus abzusetzen.

Das Modul ist als eigenständiges smartIO zum Anschluss an einen USB-2.0-Bus oder baugleich als interne Baugruppe im smartRAIL MVB verfügbar.

Der MVB-Bus ist mechanisch stabil über zwei 9-polige D-SUB Stecker durch das Modul geführt. Die kapazitive Kopplung der Signale von der A- und B-Leitung auf die Eingänge des Moduls

erfolgt ausschließlich über parallelgeführte Leiterbahnen. So ist die galvanische Trennung der Potentialebenen stets gewährleistet.

Das Modul besitzt auf der Seite des MVB-Busses zwei interne Abschlusswiderstände für MVB-EMD-Busse, die über die Pin-Paare 6-7 und 8-9 am D-SUB-Stecker verfügbar sind und über einen Brücken-Stecker, wie er in EN 61375-3, §4.5.6.4 *EMD terminator strapping* vorgeschlagen ist, genutzt werden können. Für MVB-ESD-Busse ist auf Anfrage eine Bestückungsvariante verfügbar, die zusätzlich zwei potentialfreie, kurzschlussfeste, unabhängige Spannungsversorgungen 5V für die externen Terminierung gemäß §4.4.5.3 *ESD connector of the terminator* beinhaltet.

Das kapazitiv aufgenommene MVB-Signal wird in einer analogen Signalvorverarbeitung aufbereitet und wieder in ein digitalisiertes Signal gewandelt. Ein nachgeschalteter FPGA

- erkennt das Framing der MVB-Telegramme,
- führt die Manchester-Dekodierung der seriellen Bits aus,
- prüft die Redundanz eines Telegramms auf den A/B-Leitungen,
- erkennt fehlerhafte Telegramme,
- filtert optional Telegramme nach Port-ID oder
- Messages nach Sendern oder Empfängern und
- überträgt die vollständige Kommunikation zeitgestempelt über eine USB-2.0 Schnittstelle an die Host CPU zur weiteren Verarbeitung.



1) smartIO MVB integriert im smartRAIL MVB

Messfunktionen

Einführung

Die Funktion des MVB-Busses ist in EN 61375-3 beschrieben und wird hier nur in Teilen, stark vereinfacht als kurze Einführung wiederholt. Für Details zu Timing, Framing, Signalpegeln und Dateninhalten sei deshalb auf diese Norm verwiesen.

Die Kommunikation auf dem MVB-Bus wird von mindestens einem MVB-Master gesteuert, in dem dieser kurze Anfragetelegramme an die „Slave“-Teilnehmer verschickt. Diese antworten mit den angeforderten Dateninhalten, die sich grob in drei Kategorien einteilen lassen:

1. Ports: Enthalten zyklisch ausgetauschte Prozessdaten: Messwerte, Statusinformationen, Steuersignale, kurze Texte, etc. und sind in folgenden Größen festgelegt: 16, 32, 64, 128 und 256 bits. Ports sind einem Absender eindeutig zugeordnet, alle Teilnehmer auf dem Bus können die Werte empfangen und mitlesen
2. Messages: Enthalten in Telegrammen von 256-bit Länge eine Kommunikation zwischen mindestens zwei Geräten zum Austausch ereignisbasierter Daten. Broadcast Messages sind möglich.
3. Management: Für die Signalisierung von Ereignissen, die Übergabe der Masterfunktion an einen redundanten zweiten Master, die Abfrage von Gerätezuständen, etc. gibt es ebenfalls Anfrage-Telegramme.

Die Norm sieht vor, dass die Kommunikation immer auf zwei Busleitungen A und B parallel ausgeführt wird. Dabei wird eine Busleitung in Abhängigkeit von Fehlerzuständen als Vertrauenswürdig deklariert.

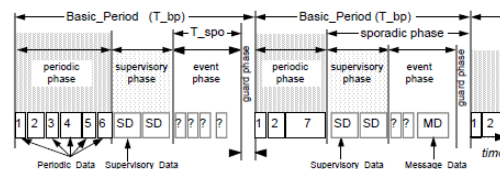


Figure 49 – Basic Periods

Die Kommunikation wird in einem Basistakt von $T_{bp} = 1\text{ms}$ durchgeführt. In diesem Intervall werden zunächst mehrere Ports und organisatorische Daten ausgetauscht, bevor Geräte nach vorliegenden Ereignissen abgefragt werden. Ein reservierter Anteil von ca. 30% bleibt für Message-Kommunikation. Ein einzelner Port wird in der Regel deutlich seltener als jede 1ms ausgetauscht (z.B. 100ms, 200ms, 500ms), sodass durch eine geschickte Verteilung aller Ports auf die tausend Zeitscheiben einer Sekunde sehr viele Daten ausgetauscht werden können.

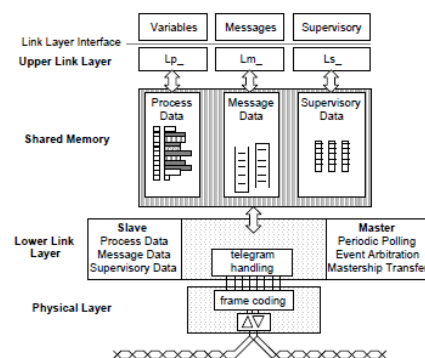


Figure 66 – Link Layer Layering

Typischer Weise wird gemäß EN 61375-3 eine Implementierung für die Anbindung an den MVB-Bus gewählt, mit der die Prozessdaten einzelner Ports in einem Speicher in der MVB-Hardware hinterlegt und gezielt abgefragt werden können. Wie in einem Postfach legt der MVB-FPGA oder ASIC die Daten eines Ports in diesem Speicher ab und von der Applikation werden gewünschte Ports in einem eigenen, unabhängigen Zeitraster ausgelesen. Zu jedem Port gibt es die Information, wie alt die vorliegenden Daten sind, jedoch keinen Zähler für die Anzahl der empfangenen Telegramme. Für den Austausch von Message-Daten müssen diese Systeme aktiv an der Kommunikation teilnehmen und benötigen im Teilnehmerkreis eine eigne Kennung.

Dieser Ansatz ist für Systeme der Automatisierung, wie z.B. Türsteuerungen, Klimaanlage, durchaus sinnvoll. Für ein Monitoring System, dessen Hauptaufgabe darin besteht, den vollständigen Datenverkehr aufzuzeichnen und auch Message-Daten zwischen beliebigen Teilnehmern mitzuhören, eignet sich dieser Ansatz nicht. Es ist nicht gewährleistet, dass jede einzelne Nachricht zu einem Port auch erfasst wird, kurzzeitige Zustandsänderungen können verloren gehen. Des Weiteren zeigt die Erfahrung, dass häufig die verfügbare Hardware nicht genügend Rechenleistung aufweist, um alle 4096 Ports ständig aus dem *Shared Memory* abzufragen und ggf. daraus wieder Ereignisse zu generieren.

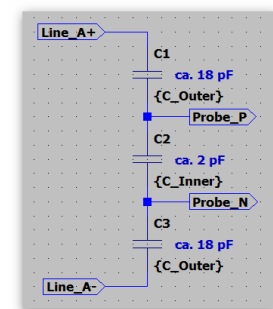
smartIO MVB – Vollständiges Bus Monitoring

Das von optiMEAS zum Patent angemeldete smartIO MVB verfolgt aus den oben genannten Gründen einen etwas anderen Ansatz.

Abnahme der physikalischen Bussignale

Die Leitungspaare A+/- und B+/- des MVB Busses werden nur durch das smartIO Modul geführt. Außer zu den zwei 9-poligen D-SUB-Steckern besteht keine Verbindung zu einem weiteren Bauteil. Die Leitungspaare werden jeweils auf den Außenseiten der Platine so übereinanderliegend und parallel geführt, dass sich zwischen den Leitern ein elektrisches Feld ausbilden kann. Dieses entspricht dem Feld, dass das Signal auch in einem typischen MVB-Kabel ausbilden würde und stellt damit keine außergewöhnliche Belastung dar. In den Innenlagen der Platine sind weitere Leiterbahnen so angeordnet, dass die Änderungen in dem elektrischen Feld erfasst und verstärkt werden können.

Das Ersatzschaltbild zu diesem Aufbau ist eine Reihenschaltung von 3 Kapazitäten zwischen den Signalleitungen, davon haben die Äußeren eine Größe von ca. 18pF und die innere ca. 2pF. Dies ergibt zusammen eine Kapazität von ca. 1.6pF. Verglichen mit dem Kapazitätsbelag eines typischen MVB-Kabels¹ von 46pF/m entspricht dies einer zusätzlichen Kabellänge von 35mm. Die Isolation zwischen MVB Bus und Signalaufbereitung/-auswertung ist durch das Platinenmaterial zwischen den Lagen gewährleistet.

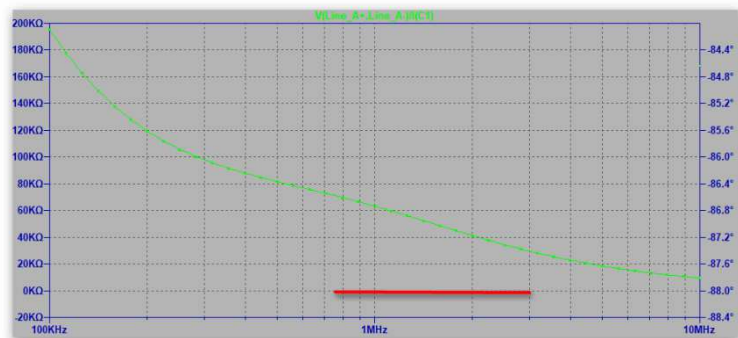


Aufgrund des elektrischen Aufbaus des Signalabgriffs ist sofort ersichtlich, dass smartIO MVB nicht dafür entwickelt wurde, aktiv an der Kommunikation auf dem MVB teilzunehmen. Es gibt keinerlei Sendeeinheit, die elektrisch auf den Bus einwirken könnte oder die Kommunikation auf dem Bus beeinflussen kann. Dadurch unterscheidet sich das Modul signifikant von denen anderer Hersteller, bei denen eine Sendeeinheit zur Rückwirkungsfreiheit nur deaktiviert wird, jedoch technisch die vollständige elektrische Kopplung der Einheit gemäß EN-Norm realisiert ist.

¹ Z.B. HELUKABEL, RADOX Databus, 120 Ohm XM S

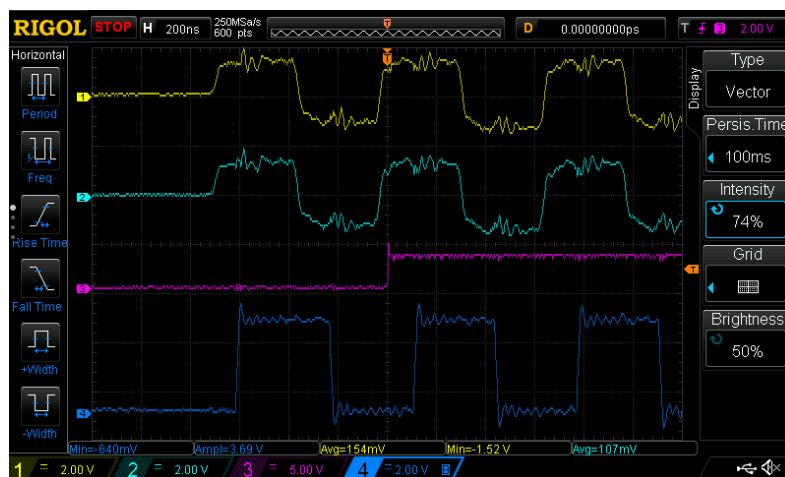
Der MVB-Bus soll eine Impedanz von $120\Omega \pm 10\%$ in dem Frequenzbereich von 0,75MHz bis 3,0MHz (0,5 bis 2,0 x BR) aufweisen. Die sehr hohe Eingangsimpedanz des smartIO MVB Moduls von $>35k\Omega$ im angegebenen Frequenzbereich führt zu einer Änderung der Impedanz auf dem Bus von

- $<0,5\Omega$ @ 3MHz ($<0.4\%$)
- $<0,21\Omega$ @ 0,75MHz ($<0.2\%$)



und liegt damit in jedem Fall deutlich innerhalb der Toleranzgrenze der Bus-Parameter. Die Impedanz des smartIO MVB ist unabhängig davon, ob das Module ordnungsgemäß funktioniert oder im schlechtesten Fall in den ersten aktiven Bauteilen (Operationsverstärker) im Signalpfad ein Defekt vorliegt.

Bei Messungen eines realen MVB-Signals (EMD) mit und ohne smartIO MVB waren keinerlei Unterschiede im Signalverlauf, den Flanken oder Amplituden erkennbar. Das Oszillogramm zeigt exemplarisch die Signalpegel der Leitung A+ (gelb) und B+ (cyan), das vom FPGA erkannte Framing-Signal (magenta, Triggerquelle) und das rekonstruierte bit-Signal (blau) im Betrieb mit dem smartIO MVB.



Mit dem zuvor gezeigten Aufbau und Verhalten des smartIO MVB am MVB-Bus ist sichergestellt, dass das Modul durch seinen Anschluss an einen bestehenden MVB-Bus keine betriebsbeeinflussende Wirkung auf den Bus ausüben kann:

- Das Modul nimmt in keiner Form an der MVB-Kommunikation teil, bleibt also für alle Teilnehmer unsichtbar.
- Das Modul beeinflusst das elektrische Signal auf den MVB-Leitungen weder in Form noch Amplitude so, dass für anderer Teilnehmer die Auswertung der Signalflanken und -pegel erschwert wird.
- Die Änderung der Impedanz der MVB-Leitung durch Einschleifen des Moduls liegt weit unterhalb der in Norm zulässigen Impedanz-Toleranz von 10% und kann somit vernachlässigt werden.
- Durch den minimalistischen, isolierten Aufbau des Signalabgriffs ohne zusätzliche Bauteile ist die Betriebssicherheit des MVB-Busses auch im Störfall der Modulelektronik gegeben.

Digitalisierung und Dekodierung

Da im Wesentlichen eine kapazitive Kopplung zum Bussignal vorliegt, kann smartIO MVB nur Änderungen auf dem MVB Bus erfassen. Diese Änderungen werden gefiltert und verstärkt und letztlich als Differenzsignal einem RS422 Receiver zugeführt, der das Signal digitalisiert und dem FPGA bereitstellt. Der korrekten Erkennung des Startbits kommt dabei eine besondere Bedeutung zu.

Im FPGA werden nach dem Startbit die Telegrammköpfe von Master oder Slave erwartet, bevor mit der Manchester-Dekodierung der Dateninhalte und der im Telegramm enthaltenen Prüfsummen begonnen wird. Für jedes Telegramm werden folgende Fehler geprüft:

- Header-Fehler: unbekannter Telegrammkopf empfangen
- Time-Out-Fehler: plötzlich abbreißender Bit-Strom
- Symbol-Fehler: 1-1 oder 0-0 Symbole im Manchester-Code
- CRC-Fehler: Fehler in der oder den 16-bit Prüfsummen, die im MVB-Protokoll spätestens nach 64 bit eingefügt sind

In dem Kommunikationsverlauf zwischen Master und Slave werden zusätzlich folgende Fehlerbedingungen erkannt:

- Time-Out-Fehler: Slave antwortet nicht oder zu spät
- Skewing-Fehler: Telegramme auf A-/B-Leitung unzulässig weit gegeneinander verschoben
- Redundanz-Fehler: Anfrage oder Antwort ist nur auf einer Busleitung verfügbar

Der FPGA bewertet jedes redundant empfangene Telegrammpärchen und leitet dasjenige mit dem zuverlässigsten Dateninhalt zur Auswertung weiter. Folgende Filtermöglichkeiten sind im FPGA einstellbar:

- Monitoring nur auf Leitung A oder B (Solo), Monitoring auf A und B (Redundanzbetrieb)
- Unterdrückung aller Frames mit Fehlerzustand
- Filtern von Port-Telegrammen nach Port-Nummer (white list)
- Filtern von Message-Telegrammen nach Sender / Empfänger und Telegrammtyp (white list)

Alle nicht ausgefilterten oder unterdrückten Telegramme werden mit einem hochauflösenden Zeitstempel versehen und zusammen mit der Status- und Fehlerinformation an die auswertende CPU über die USB-Schnittstelle übertragen. Damit können ohne Einschränkung bis zu 100% des auf dem MVB auftretenden Datenverkehrs transparent ausgewertet und aufgezeichnet werden. Und dies zeitgenau mit vielen zusätzlichen Statusinformationen, die auch eine Diagnose der MVB-Kommunikation zulassen.

Übertragung an die Host-CPU

Die vom FPGA bereitgestellten Daten werden binär über eine USB-2.0 Schnittstelle an die Host-CPU übertragen (z.B. smartRAIL), wo in Software eine weitere Auswertung der Dateninhalte und Statusinformationen erfolgen kann. Ein spezieller Treiber ist hierfür nicht erforderlich. Das Modul meldet sich mit seinem FTDI-Chip am Betriebssystem an und stellt für die Funktion eine virtuelle, „serielle“ Schnittstelle (COM:, /dev/tty*) zur Verfügung.

Durch verschiedene Ringspeicher wird ein hoher Datendurchsatz ermöglicht. Bei der Simulation eines vollständig ausgelasteten MVB-Busses mit vielen kurzen MVB-Telegrammen werden Datenraten bis 500kByte/s erreicht, bei Messungen auf realen Fahrzeugen liegt dieser Wert eher bei 150–250kByte/s. Je nach Auslastung und Echtzeitfähigkeit der CPU kann es trotzdem zu Pufferüberläufen kommen, wenn die CPU die Daten von der Schnittstelle nicht schnell genug entgegennehmen kann. Hier ist beim Softwaredesign einer Auswerte-Software besondere Aufmerksamkeit erforderlich.

Die einzelnen Datenpakete sind im Datenstrom jeweils mit einer eindeutigen 32-bit Kennung versehen, sodass eine Synchronisierung auf den Datenstrom jederzeit gegeben ist.

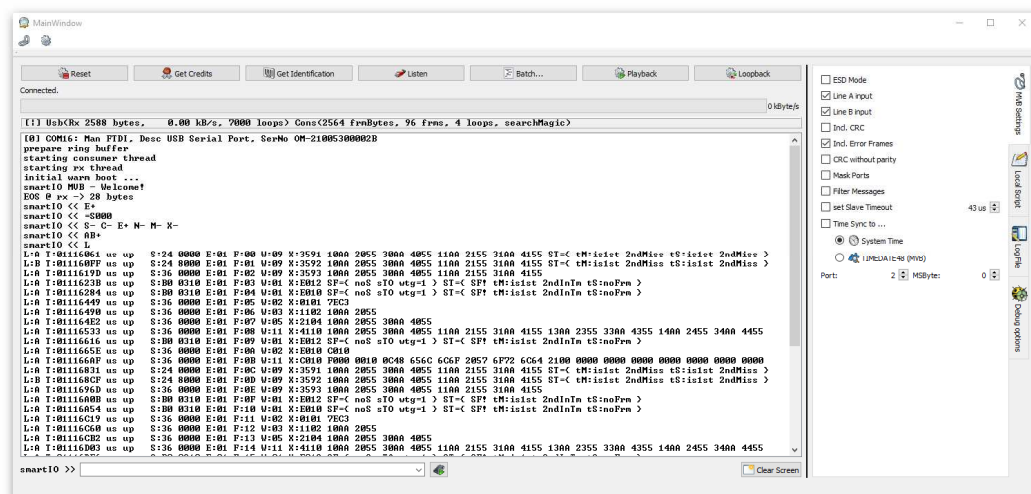
Software: Auswertung und Aufzeichnung

Folgende Softwarekomponenten zur Auswertung und Aufzeichnung des Datenverkehrs sind verfügbar:

- **MVB-Plugin** für smartCORE (Standard)
Konfiguration für den Regelbetrieb, Dekodierung einzelner Datenkanäle aus Ports, Aufzeichnung der Message-Kommunikation
- **mvbdump**: Unix-Kommandozeilen-Tool (auf Anfrage)
Darstellung / Dump aller MVB Telegramme mit Statusinformationen, einfache Diagnosefunktionen und MVB-Simulation zur Software-Entwicklung

```
/sde/mvb # ./mvbdump -h
usage: mvbdump <options>
mandatory:
  -o <dumpFile>
  -f <format>      (def = 0 [bin], 1, 2, ...)
recommended:
  -i <inputBatch>  (load a test program sequence)
  -c <c-sequence>  (immediate, direct commands)
  -t <minutes>     (def = 0: interactive mode)
optional:
  -L              (list all devices)
  -r <readDumpFile>
  -u <usbDevice>  (def = '', find by udev)
  -s              (synch to local time)
  -p <port>       (synch to TIMEDATE48 in port)
  -b <offset>     (synch to TIMEDATE48 at byte offset)
```

- **mvb_test**: Windows Anwendung (auf Anfrage) zur Inbetriebnahme eines smartIO MVB, einfache Diagnosefunktionen und MVB-Simulation zur Software-Entwicklung



Die Applikationen können alle den MVB Datenverkehr als „Dump“ in einem ASCII-basierten Format mitschreiben. Die Dateien werden verständlicher Weise sehr schnell mehrere Gigabyte groß, sodass die Arbeit mit geeigneten Analysewerkzeugen² eher aus der Softwareentwicklung³ empfohlen wird. Zum fortlaufenden Mitschreiben der Telegramme und Übertragung über eine Mobilfunkstrecke ist dieses Format ebenfalls nicht vorgesehen.

Das nachfolgende Bild zeigt den Ausschnitt eines so erstellten Dumps, in dem folgende Informationen enthalten sind:

² Explizit: nicht EXCEL

³ Unter Windows z.B. Bare-Grep und Bare-Tail in der Freeware- oder Pro-Variante


```
(34.0 MB) - BareTailPro
File Edit View Preferences Help
Open Highlighting Follow Tag ANSI C:\Users\opt\MEAS GmbH\Auftragsabwicklung - Dokumente\Offene Aufträge (34.0 MB)

L:R  T:09.40.02.378642 S:09 0000 E:01 F:SE W:02 Q:F010 <stat> R:S780 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.378693 S:09 0000 E:01 F:SF W:02 Q:F010 <stat> R:S780 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.378744 S:09 0000 E:01 F:W0 W:02 Q:F010 <stat> R:S780 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.378795 S:09 0000 E:01 F:L1 W:02 Q:F010 <stat> R:S780 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.378846 S:09 0000 E:01 F:E2 W:02 Q:E021 <evnt> R:C021 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.378897 S:09 4000 E:01 F:R3 W:02 M:0108 <v15 ST= [ noS sTO } ST=[ SF! tM1:A.2.miss tS:1.A.2.noFr }
L:R  T:09.40.02.378948 S:09 4000 E:01 F:R4 W:02 M:0108 <v15 ST= [ noS sTO } ST=[ SF! tM1:A.2.miss tS:1.A.2.noFr }
L:R  T:09.40.02.378999 S:09 4000 E:01 F:R5 W:02 M:0108 <v15 ST= [ noS sTO } ST=[ SF! tM1:A.2.miss tS:1.A.2.noFr }
L:R  T:09.40.02.379050 S:09 4200 E:01 F:R6 W:02 M:0108 <v15 ST= [ noS sTO } ST=[ SF! tM1:A.2.miss tS:1.A.2.noFr }
L:R  T:09.40.02.379200 S:09 31F0 E:01 F:EE W:01 X:S780 SF=[ Sym noM sTO wtg=F } ST=[ SF! tM1:A.2.noFr tS:1.A.2.miss }
L:R  T:09.40.02.379251 S:09 0000 E:01 F:EE W:02 X:0010 <mmd> R:E001 ST=[ tM1:A.2.miss tS:1.A.2.noFr }
L:R  T:09.40.02.380540 S:09 0300 E:01 F:P7 W:01 Q:J23E <p128> SF=[ noS sTO } ST=[ SF! tM1:A.2.miss tS:1.A.2.noFr }
L:R  T:09.40.02.380681 S:09 0300 E:01 F:R8 W:01 Q:T2196 <p 64> SF=[ noS sTO } ST=[ SF! tM1:A.2.miss tS:1.A.2.noFr }
L:R  T:09.40.02.380764 S:09 0000 E:01 F:R9 W:09 Q:J221E <p128> R:DAS8 0001 0002 0000 0000 0000 0000 0000 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.380924 S:09 0000 E:01 F:EA W:01 X:385E SF=[ Rdr noM nos } ST=[ SF! tM1:A.2.noFr tS:1.A.2.noFr }
L:R  T:09.40.02.381093 S:09 0300 E:01 F:FA W:01 Q:E014 <sEvT> SF=[ noS sTO } ST=[ SF! tM1:A.2.miss tS:1.A.2.noFr }
L:R  T:09.40.02.381540 S:09 0000 E:01 F:EB W:09 Q:J314E <p128> R:0000 0000 0000 0000 0000 0000 00DD B726 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.381675 S:09 0000 E:01 F:EC W:05 Q:E2182 <p 64> R:795E 0000 0006 6800 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.381758 S:09 0000 E:01 F:ED W:11 Q:434A <p256> R:0000 0000 0000 0000 FFFF FFFF FFFF FFFF 0000 0000 0000 0000 0000 0000 000E 0D4C ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.381989 S:09 0000 E:01 F:EE W:11 Q:434A <p256> R:E003 1676 0000 0000 0000 000B 00C4 0000 0000 0000 ASAO 0000 0033 000E 9FE2 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.382220 S:09 0000 E:01 F:EF W:11 Q:C021 <mdrg> M1 m1 R:010 p18 T:021 c:08 D:00 32 00 32 41 00 3B 70 } ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.382450 S:09 0000 E:01 F:F0 W:09 Q:J319E <p128> R:0000 0000 0000 0000 0000 0000 00AE 5A1A ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.382671 S:09 0000 E:01 F:F1 W:11 Q:436A <p256> R:00E0 0E74 0000 0000 0000 0000 00AC 0A92 0000 0000 ASAO FFFF 00E2 00A4 F6FC ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.382912 S:09 0000 E:01 F:F2 W:11 Q:436A <p256> R:FFFF FFFF 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 00A4 443F ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.383236 S:09 0000 E:01 F:F3 W:05 Q:E221E <p 64> R:012B 0400 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 }
L:R  T:09.40.02.383392 S:09 0300 E:01 F:F4 W:01 Q:J931O <evT> SF=[ noS sTO } ST=[ SF! tM1:A.2.miss tS:1.A.2.noFr }
L:R  T:09.40.02.383540 S:09 0000 E:01 F:F5 W:09 Q:J334E <p128> R:0000 0000 0000 0000 0000 0000 000E EB47 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.383615 S:09 4100 E:01 F:F6 W:01 Q:J01E <v15 ST= [ noS sTO } ST=[ SF! tM1:A.2.miss tS:1.A.2.noFr }
L:R  T:09.40.02.383767 S:09 1100 E:01 F:F7 W:10 X:X000 0011 0000 0101 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 03CC 7C87 SF=[ noM } ST=[ SF! tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.383895 S:09 31C0 E:01 F:F7 W:04 X:X000 0501 0130 0603 SE=[ Sym noM sTO wtg=C } ST=[ SF! tM1:A.2.noFr tS:1.A.2.miss }
L:R  T:09.40.02.384062 S:09 0000 E:01 F:F7 W:02 Q:E010 <evT> R:C010 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.384540 S:09 0000 E:01 F:F8 W:09 Q:J369E <p128> R:0000 0000 0000 0000 0000 0000 00A4 3065 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.384685 S:09 0000 E:01 F:F9 W:11 Q:4421 <p256> R:0000 0011 0000 0101 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0008 0002 03B7 88B2 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.384921 S:09 0000 E:01 F:FA W:11 Q:4431 <p256> R:0000 0010 0000 0101 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0003 03CC ED24 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.385157 S:09 0000 E:01 F:FB W:02 Q:E021D <p 16> R:DA57 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.385208 S:09 0300 E:01 F:FC W:01 Q:E011 <sEvT> SF=[ noS sTO } ST=[ SF! tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.385540 S:09 0000 E:01 F:FD W:09 Q:J343E <p128> R:0000 0000 0000 0000 0000 0000 000C BAB7 ST=[ tM1:A.2.miss tS:1.A.2.miss }
L:R  T:09.40.02.385680 S:09 0000 E:01 F:FE W:09 Q:J342E <p128> R:0000 0000 0000 0000 0000 0000 00B7 850B ST=[ tM1:A.2.miss tS:1.A.2.miss }
```

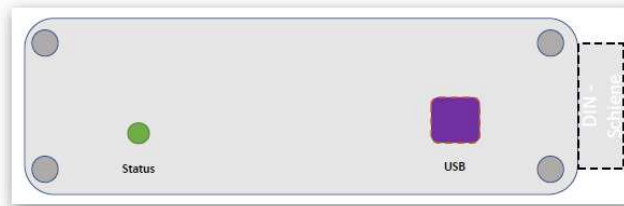
- L: Telegramm von Leitung A / B
- T: Zeitstempel des Telegramms mit μ s-Auflösung
- S: Status-Informationen zu Timing, Inhalt, CRCs (siehe auch SF und ST)
- E: Summenzähler für Übertragungsfehler zur Host-CPU
- F: Telegrammzähler
- Q: Anfrage-Kommando des Masters und <...> dekodierter Befehl
- R: Slave-Antwort (Ports)
- M: Slave-Antwort (Messages)
- X: Nicht identifizierte oder fehlerhafte Telegrammfragmente
- SF: Status Flags zum Telegramm
- ST: Timing und Redundanz-Status

Es sei darauf hingewiesen, dass auf dem MVB-Bus durchaus verschiedene Fehlerzustände erwartet oder auch notwendig sind. Diese treten insbesondere mit der Abfrage von Event-Zuständen auf, bei denen entweder keine Antwort kommt, sofern keine Events in den Subsystemen anliegen, oder sogar mehrere Antworten miteinander kollidieren, wenn mehrere Subsysteme ihre Ereignisse melden wollen. smartIO MVB erlaubt, fehlerhafte Frames ebenfalls an die Host CPU zu leiten und so ein vollständiges Bild der Vorgänge auf dem MVB zu erhalten.

Schnittstellen des Moduls

Spannungsversorgung, Status

Das smartIO MVB wird in der Variante eines eigenständigen IO Moduls über den externen USB-2.0 Anschluss Typ-B versorgt, in der Variante als integriertes Modul im smartRAIL-MVB über den internen USB-2.0 Anschluss von der CPU-Platine des smartRAIL. Der „externe“ Anschluss liegt in diesem Fall nicht zugänglich hinter der Frontblende und wird ausschließlich für die Werks-Inbetriebnahme verwendet. In keinem Fall dürfen beide Anschlüsse gleichzeitig verbunden sein.



Die Funktion des smartIO MVB wird durch die Blink-Codes der Status-LED angezeigt. Dabei bedeuten:

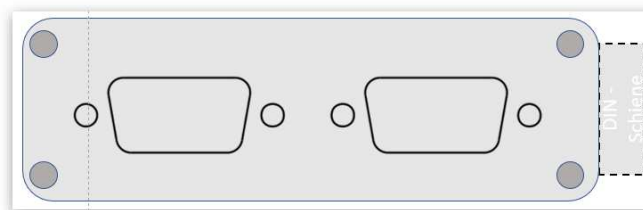
Blink-Code	Bedeutung
blau:  	Stand-By, Warte auf Steuerbefehle
blau: 	Steuerzeichen von der Host-CPU empfangen
rot:  / 	Fehler oder Fehlerereignis in USB-Kommunikation, Pufferüberlauf
grün:  	Lesen der MVB-Leitung A oder B, einfach
grün:    	Lesen der MVB-Leitungen A und B, redundant

Legende: : fade in, : fade out, : on, : off, : permanent on, : flash, : break

Das MVB Softwaremodul im smartCORE ist so ausgeführt, dass Fehlerzustände selbsttätig erkannt und bedient werden. Nach dem Einschalten ist die Anzeige eines kurzzeitigen Fehlerzustands normal, ebenso die Anzeige des Stand-By-Modus, bis die Software die Kontrolle über das Modul übernimmt. Im Betrieb sollte, je nach Konfiguration, ein wiederholt verglimmendes grünes Leuchtsignal oder ein grüner Lichtimpuls mit anschließendem Verglimmen angezeigt werden. Einzelne MVB-Botschaften, der MVB-Datenverkehr oder MVB-Fehler werden nicht angezeigt.

MVB Anschluss

Auf der Seite der Feldbusanschlüsse sind am smartIO MVB die beiden 9-poligen MVB Anschlüsse zu finden. An den MVB Anschlüssen sind die Pins 1 bis 5 zwischen [X9] und [X10] direkt miteinander verbunden. Die Pins 6 bis 9 bieten standardmäßig eine Option zur normgerechten Terminierung eines MVB-EMD Busses oder als Bestückungsvariante auch eines MVB-ESD Busses an. Die Gewindebolzen zur Arretierung der Stecker sind mit einem UNC4-40 Gewinde ausgeführt. Ein Umrüstsatz auf M3 Gewinde ist unter Art.-Nr. 20745 verfügbar. Bei größeren Stückzahlen kann dieser bereits bei der Produktion berücksichtigt werden.

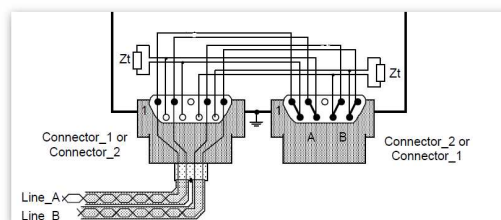


Pin	Signal	Beschreibung EMD (Standard)	Beschreibung ESD (Option ⁴)
1	A+	Datenleitung Bus A, positiv	
2	A-	Datenleitung Bus A, negativ	
3	[TxE]	[ohne Verwendung]	[TxE, ohne Verwendung]
4	B+	Datenleitung Bus B, positiv	
5	B-	Datenleitung Bus B, negativ	
6	Term/GND	Terminierung-A, positiv ($Z=120\Omega$)	Versorgung Term.-A, GND
7	Term/GND	Terminierung-A, negativ	Versorgung Term.-B, GND
8	Term/5V	Terminierung-B, positiv ($Z=120\Omega$)	Versorgung Term.-A, 5V, 100mA
9	Term/5V	Terminierung-B, negativ	Versorgung Term.-B, 5V, 100mA

Terminierung

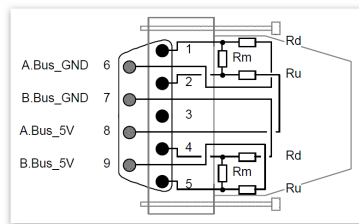
Zur Terminierung der MVB-Busleitungen am smartIO ist gemäß Standard ein externer D-SUB-Endstecker mit geeigneter Beschaltung notwendig. Dieser kann an beiden Anschlüssen aufgesteckt werden.

Für EMD-Terminierung sind 4 Brücken im Endstecker gefordert, mit denen die interne Abschlussimpedanzen auf die Busleitungspaare A und B geschaltet werden:



⁴ Als Bestückungsoption verfügbar, dann per Jumper auf dem Modul für EMD/ESD konfigurierbar

Für die ESD-Terminierung, die als Bestückungsoption auf Anfrage verfügbar ist, liegen Abschlussimpedanz und die Widerstände für die Potentialvorspannung der Busleitungspaare A und B im Endstecker. Das smartIO liefert zwei potentialfreie Spannungen mit 5V, die mit bis zu 100mA belastbar sind.



Konfiguration

Die Konfiguration des smartIO MVB erfolgt immer aus der ansteuernden Software heraus, sobald diese die Verbindung mit dem Modul aufgenommen hat. Dazu stehen verschiedene Befehle im ASCII Format zur Verfügung, die auf Anfrage erhältlich sind. In der Regel ist die Konfiguration vollständig im MVB-Plugin des smartCOREs abgebildet oder wird durch die genannten Software-Tools vorgenommen.

Besondere Zulassungen und Erklärungen

Für die *smartRAIL-Ausführung* werden neben der CE Konformitätserklärung auch die nachfolgenden Zertifizierungen zur EN50155 vorgenommen:

EG-Konformitätserklärung		Das CE-Zeichen zeigt die Übereinstimmung mit der • EMV-Richtlinie, • RoHS 2011/65/EU (08.06.2011) und der • Niederspannungsrichtlinie an.
Bahnanwendungen – Elektronische Einrichtungen auf Bahnfahrzeugen. EN 50155:2008	   	Die Systeme erfüllen die Norm für folgende Eigenschaften: • <i>Umgebungsbedingungen:</i> ○ AX (2000m) EN50125-1 §4.2.1 ○ TX EN50155 §4.1.2 • <i>Klima</i>⁵ ○ Kälte EN50155 §13.4.5.2 ○ Trockene Wärme EN50155 §13.4.5.3 ○ Feuchte Wärme EN50155 §13.4.5.7 zyklisch • <i>Schwingen</i> IEC61373 §8 + 9 • <i>Schocken</i> IEC61373 §10 • <i>EMV, Isolation</i> EN50121-3-2 EN 61000-3-2/3 EN 55016-2-1/2 • <i>Brandschutz</i> EN45545-2 <i>Stückprüfungen gemäß</i> EN50155 §12.2 • <i>Sichtprüfung</i> EN50155 §12.2.1 • <i>Isolation</i> EN50155 §12.2.9.1 • <i>Stehspannung</i> EN50155 §12.2.9.2

⁵ Bereits nach EN 50155:2017

Technische Daten

Versorgungsspannung / Umgebungsbedingungen

Symbol	Parameter	Bemerkung	Min	Typ	Max	Einheit
V _{CC}	Versorgungsspannung	aus USB		5		V DC
	Überspannungsschutz		nein			
I _{CC}	Stromaufnahme	@ 5V, ohne I _{Term,ESD}		200		mA
	Steckverbinder		USB Typ-B			
T _{operating}	Betriebstemperatur	EN 50155 / Bereich TX	-40		85	°C
	Relative Luftfeuchte	Nano-Beschichtung, 50°C	5		95	%
	Gehäuse		Aluminium			
L	Maße: Länge	ohne Stecker / Füße / Clip		124		mm
B	Breite			85		mm
H	Höhe			35		mm
m	Gewicht			250		g
	Verschraubung	Frontplatten gegen Korpus	M3, Edelstahl VA2, Sicherungsscheiben			
	Montage (Standard)	Option Montagefüße oder Tragschiene (EN 50022)				
	Montage (RAIL)		TS 35			
	Kühlung		passiv			
	Schutzklasse	[ISO 20653 - 2013]	IP54			
	Stehspannungsprüfung, 60s RAIL	MVB zu USB		1.7		kV

MVB Schnittstelle

Symbol	Parameter	Bemerkung	Min	Typ	Max	Einheit
	Typ / Anzahl	MVB		1		
	Übertragungsrate			1.5		Mbit/s
	Norm		DIN EN 61375-3-1			
	MVB-Bustyp	nur lesend	EMD, ESD ⁶			
	MVB-Gerätekategorie		Class 0			
	Steckverbinder		D-SUB-9, UNC4-40 ⁷			
	Kodierung		Standard			
	Ausführung	Geräteseitig	male/female			
C _{Input}	Koppel-Kapazität			18		pF
Z _{Input}	Eingangsimpedanz	3 MHz...100 kHz	30	50	250	kΩ
Z _{Term,EMD}	MVB-Terminierung ⁸	EMD (Standard, R-Intern)		120		Ω

⁶ Aktive Terminierung für ESD als Bestückungsoption auf Anfrage erhältlich

⁷ Ein Umrüstsatz auf M3 Gewindebolzen ist verfügbar (Art.-Nr.: 20745)

Symbol	Parameter	Bemerkung	Min	Typ	Max	Einheit
$U_{\text{Term,ESD}}$		ESD ⁹ (Option, R-Netz-Extern)	4.9	5	5.1	V DC
$I_{\text{Term,ESD}}$					100	mA
	Status-LED	Status, Activity, Command	RGB			
	Interpretation	Single-Line / Redundant-Line	Ports, Messages, Events, Status			

USB Schnittstelle (intern für smartRAIL – MVB)

Symbol	Parameter	Bemerkung	Min	Typ	Max	Einheit
	Typ / Anzahl	USB 2.0		1		
	Übertragungsrate			480		Mbit/s
	Datenrate			150	500	kByte/s
	IO Puffer			4		kByte
	USB-Chip		FT2232			
	Treiber	Standard Seriell-Treiber	Windows, Linux, YoctoLinux			

Kontakt

optiMEAS Measurement and Automation Systems GmbH
 Am Houiller Platz 4
 64381 Friedrichsdorf
 +49 (6172) 997712-0
 info@optimeas.de



⁸ Über Terminierungsstecker nach Norm für EMD / ESD zu aktivieren

⁹ Als Bestückungsoption erhältlich