

JMRI

Die Unterstützung für BiDiB in [JMRI](#) wird von Eckart gemacht.

Siehe [Forums-Thread](#).

Verbindungen

Anschlüsse

Derzeit unterstützt JMRI die Verbindung über USB, BiDiB über TCP, und es gibt einen BiDiB-Simulator, der durch eine XML-Datei konfiguriert wird.

Öffnen Sie die Registerkarte Verbindungen, wählen Sie „BiDiB“ aus dem Auswahlfeld Systemhersteller und wählen Sie aus den folgenden Setups. Der Systembuchstabe für BiDiB-Verbindungen ist „B“.

BiDiB unterstützt Hotplugging auf dem BiDiBus, dies wird auch in JMRI-BiDiB unterstützt. Geht ein Knoten verloren, werden alle Objekte, die darauf verweisen als ungültig markiert aber nicht entfernt. Wenn ein neuer Knoten gefunden wurde, werden alle ungültigen Objekte überprüft, ob sie wieder aktiviert werden können.

Serial über USB

Um eine neue Verbindung zu erstellen, kann entweder der Portname (z. B. ttyUSB0 unter Linux oder COM1 unter Windows) ausgewählt oder die UID des Root-Knotens direkt eingegeben werden. Da UIDs weltweit eindeutig sind und sich für eine bestimmte Hardware nie ändern, kann der passende Port automatisch gefunden werden, indem alle verwendbaren Ports gescannt werden. Unter Windows werden alle Ports COMx gescannt, unter Linux und macOS kann ein Namensfilter verwendet werden (Standard ist „ttyUSB*“). Beim Herstellen der Verbindung durch Auswahl eines Ports aus der Liste der verfügbaren Ports wird die vom Gerät ausgelesene UID angezeigt und gespeichert - sofern es sich bei dem Gerät um einen BiDiB-Knoten handelt. Dazu kann eine Checkbox (Autoscan) aktiviert werden - wenn JMRI neu gestartet wird, kann der Port durch Scannen der verfügbaren Ports gefunden werden. Wenn also mehr als eine BiDiB-Verbindung verwendet wird, ist die Port-Bezeichnung des USB-Geräts nicht relevant.

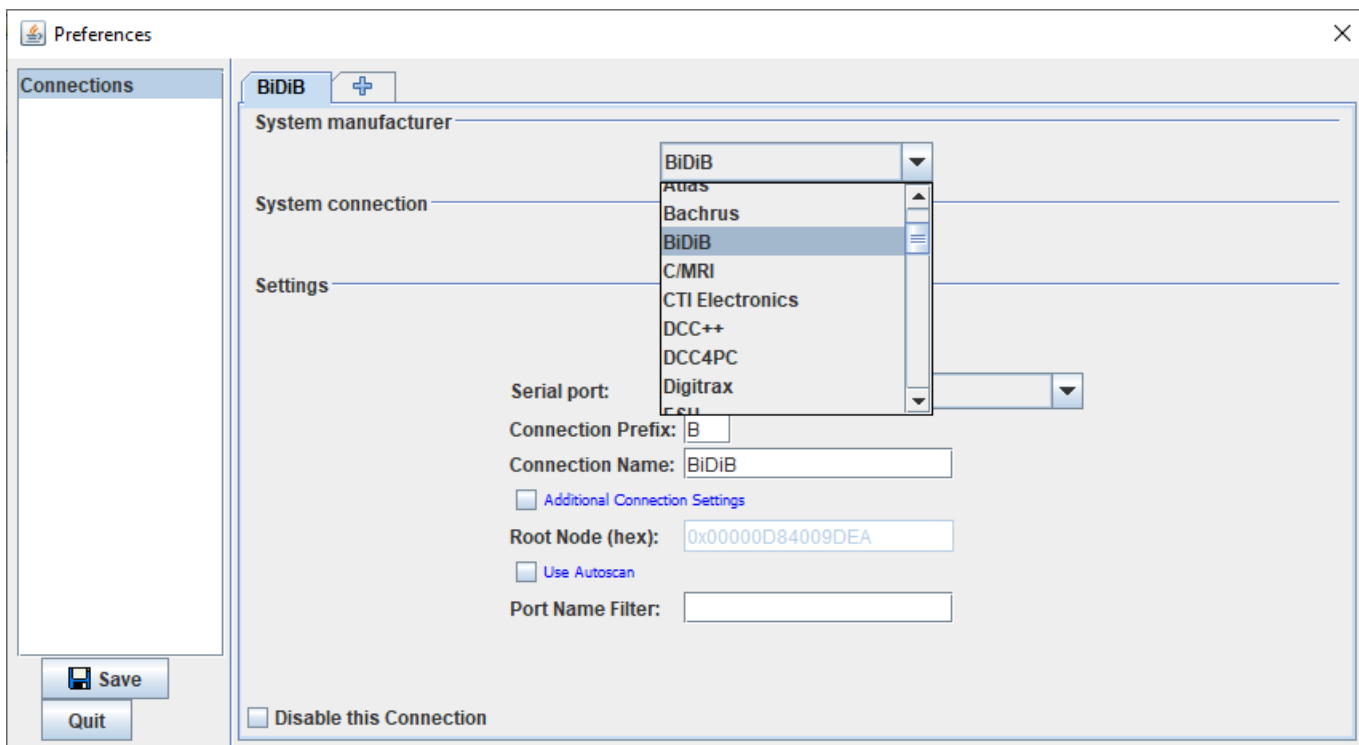
BiDiB über TCP

In diesem Fall ist die Hardware nicht direkt mit JMRI verbunden, sondern JMRI verbindet sich mit einem TCP-Server, der sich wiederum mit der Hardware verbindet. Dies kann nützlich sein, wenn ein anderes Programm als Hauptsteuerprogramm verwendet wird und JMRI nur als CV-Programmierer (Decoder Pro). Die Parameter sind die IP-Adresse und die Portnummer des Servers, zu dem eine Verbindung hergestellt werden soll. Die Standardportnummer ist 62875.

Simulator

Es steht ein BiDiB-Simulator zur Verfügung. Dieser Verbindungstyp hat ein Feld für eine XML-Datei mit der simulierten Konfiguration. Der Speicherort für diese Datei ist das Profilverzeichnis.

Einstellungen



BiDiB Namensgebung

JMRI-Systemnamen bestehen im Allgemeinen aus einem Systemverbindungspräfix Xn, wobei X der Systemverbindungsbuchstabe ist und n entweder leer oder bei Mehrfachverbindungen eine Zahl ist, gefolgt von dem Typbuchstaben („T“ für Weichen, „L“ für Lampen, „S“ für Detektoren und „R“ für Melder (Lokadressenrückmeldung, Reporter) und die Systemadresse.

Für BiDiB ist der Systemverbindungsbuchstabe „B“, die Systemadresse ist ein String mit der BiDiB-Adresse.

Die BiDiB-Adresse hat generell die Form:

Knoten:Adresse

Ein Knoten auf dem BiDiBus wird entweder durch seine weltweit eindeutige Kennung (UID) identifiziert, die eine 40-Bit-Hexadezimalzahl ist, oder durch einen im Knoten konfigurierten Benutzernamen (mit dem BiDiBWizard). Die hexadezimale UID muss mit einem „X“ beginnen, Benutzernamen müssen mit einem Buchstaben beginnen (besser nicht mit „X“) und dürfen nur andere Buchstaben, Zahlen, Unterstriche („_“) oder Bindestriche („-“) enthalten. Dies ist eine Einschränkung von JMRI für BiDiB, nicht für BiDiB selbst. Die Knotenspezifikation ist unabhängig von Groß- und Kleinschreibung.

Der Knotenteil der Adresse kann weggelassen werden (einschließlich Doppelpunkt), in diesem Fall wird automatisch das Interface (Root Node, d. h. der direkt mit JMRI verbundene BiDiB-Knoten) verwendet.

Der Adresse auf dem Knoten kann ein Adresstyp als Buchstabe vorangestellt werden, und – nur für Portadressen – kann ein Porttyp als Buchstabe folgen. Die Nummer ist erforderlich und die beiden Buchstaben sind optional. Werden sie weggelassen, werden geeignete Standardeinstellungen in Abhängigkeit von den Fähigkeiten des Knotens ausgewählt: Nur eine Zentrale hat DCC-Adressen, Weichen sind bevorzugt Accessories und Lichter sind bevorzugt Ports (digitale Pins des Geräts). Die allgemeine Form einer Adresse auf einem Knoten ist also:

xny

wobei x der optionaler Adresstyp ist, n die numerische Adresse und y der optionale BiDiB-Porttyp.

In JMRI werden in deutscher Sprache Belegtmelder als Detektoren (engl. Sensors) bezeichnet, Railcom Rückmelder als „Melder“.

Die Bezeichnung „Weichen“ (engl. Turnouts) in JMRI bezieht sich nicht unbedingt auf tatsächliche Weichen, sondern eher auf ein einen generellen I/O-Port der Hardware, der die beiden Zustände An und Aus einnehmen kann. In BiDiB sollten Weichen aber nicht direkt über die Ports gesteuert werden, sondern als Accessories angesprochen werden. Diese Auswahl geschieht über den Adresstyp-Buchstaben.

Liste der Adresstypen:

Buchstabe	Beschreibung	Standardtyp für:
t	DCC Accessory Adresse („track“)	Weichen und Belegtmelder wenn der Knoten eine Command Station (CS) ist. Ungültig bei Knoten ohne CS Funktion.
a	(non-DCC) Accessory Adresse	Weichen wenn der Knoten keine Command Station ist und der Knoten BiDiB Accessories unterstützt.
f	Belegtmelder Nummer	Detektoren, wenn der Knoten BiDiB Belegtmelder unterstützt. Melder (Reporters) sind nur für Knoten relevant, die Belegtmelder unterstützen, daher immer Standard
p	Portnummer	Lampen (LEDs). Für Detektoren, wenn der Knoten keine Belegtmelder unterstützt.

Liste der BiDiB Porttypen:

Buchstabe	Bezeichnung	Beschreibung
S	SWITCHPORT	Einfacher digitaler Ausgang. Entweder EIN oder AUS.
L	LIGHTPORT	Steuerung eines digitalen Ausgangs mit diversen Funktionen wie Dimmen, Blinken etc. Gesteuert durch einen Zustandswert (ähnlich wie bei Signalen).
V	SERVOPORT	Steuerung eines angeschlossenen Servos. Start- und Endposition werden im Knoten konfiguriert, ebenso die Umlaufgeschwindigkeit.
U	SOUNDPORT	derzeit nicht in JMRI unterstützt.
M	MOTORPORT	derzeit nicht in JMRI unterstützt.
A	ANALOGPORT	derzeit nicht in JMRI unterstützt.
B	BACKLIGHTPORT	derzeit nicht in JMRI unterstützt.
P	SWITCHPAIRPORT	derzeit nicht in JMRI unterstützt.

Buchstabe	Bezeichnung	Beschreibung
I	INPUTPORT	Standard für Sensoren. Für andere ungültig.

Beispiele (das Interface (Rootnode) wird als Zentrale angenommen und hat die UID 0d68001234 und „MeinKnoten“ ist der Nutzernamen):

Systemname	Typ	Adressteil	resultierende Knoten UID	resultierende Adresse	Bemerkungen
BSX0d68001234:20	Detektor	X0d68001234:20	0d68001234	f20 - Belegmelder 20	BiDiB Belegmelder 20 des (Interface-) Knoten 0d68001234
BSMyNode:42	Detektor	MyNode:42	0d68001234	f42 - Belegmelder 42	BiDiB Belegmelder 42 des (Interface-) Knoten 0d68001234 mit dem Nutzernamen „MyNode“
BT5	Weiche	5	0d68001234	t5 - DCC Adresse 5	Weiche mit der DCC address 5 über das Interface (CS Knoten) 0d68001234
BTN201:22	Weiche	N201:22	0d68006789	a22 - Accessory Adresse 22	Weiche mit der BiDiB Accessory Adresse 22 des Knoten 0d68006789 mit dem Nutzernamen „N201“ (nicht die Command Station)
B1LLC6:8L	Lampe (licht)	LC6:8	0d68004321	p8L - Port 8	Lampe am Ausgangsport 8 (typbasierendes Adressierungsmodell) des Knoten 0d68004321 mit dem Nutzernamen „LC6“ auf der zweiten BiDiB Verbindung (B1)
BLN201:15	Lampe (Licht)	N201:15	0d68006789	p15L - Port 15	Licht am Ausgangsport 15 (flaches Adressierungsmodell) des Knotens 0d68006789 mit dem Namen „N201“, Port muss als LIGHT konfiguriert werden
BSX0d68006789:3	Detektor	0d68006789:3	0d68006789	p3I - Port 3	Detektor am Eingangsport 3 (flaches Adressierungsmodell) des Knotens 0d68006789 (ohne Belegmelder), Port muss als INPUT konfiguriert werden

Systemname	Typ	Adressteil	resultierende Knoten UID	resultierende Adresse	Bemerkungen
BR42	Melder (Reporter)	42	0d68001234	f42 - Belegmelder 42	Railcom-Rückmeldung über Belegmelder 42 des Interface Knotens 0d68001234

Detektoren

BiDiB Belegtmelder und Eingangsports werden Detektoren zugeordnet.

Weichen

BiDiB-Accessories sind das Äquivalent zu Weichen. Sowohl Signallampen als auch tatsächliche Weichen können sich auf JMRI-Weichen beziehen. Signale sollen aber über „Signalmasten“ bedient werden (siehe weiter unten).

Lampen

BiDiB-Ports werden normalerweise Lampen (Lichter, Lights) zugeordnet. BiDiB-Ports unterstützen entweder ein flaches Adressierungsmodell oder ein typbasiertes Adressierungsmodell. Das Adressierungsmodell wird durch die Firmware des Knotens bestimmt. Beim flachen Adressierungsmodell werden alle verfügbaren Ports einfach von 0 bis zur Anzahl der Ports minus 1 nummeriert. Jeder Port muss im Knoten (z. B. mit dem BiDiBWizard) als einer der oben genannten Porttypen konfiguriert werden, der von der Hardware selbst unterstützt wird. In diesem Fall ist der Buchstabe des Porttyps in der JMRI-Systemadresse nicht erforderlich, da der Typ vom Knoten gelesen wird.

Wenn der Knoten das typbasierte Adressierungsmodell verwendet, sind Porttypen in der Firmware fest kodiert und können nicht geändert werden. Jeder Porttyp hat einen eigenen Adressbereich beginnend bei 0 bis zur Anzahl der Ports für diesen Typ minus 1. In diesem Fall ist der Porttyp Teil der Portadresse und der Porttypbuchstabe wird in der BiDiB-Adresse benötigt. Siehe die Beispiele oben.

BiDiB Signalmasten

Es wird ein Signalmasttyp für BiDiB-Accessories bereitgestellt, bei dem ein JMRI-Aspekt direkt einem BiDiB-Accessories-Aspekt zugeordnet werden kann. Die Signalbilder können programmiert werden in den BiDiB-Knoten mit Makros über den BiDiB-Wizard. Andere Signalmasttypen (z. B. der Matrix-Typ) können „Weichen“ zugeordnet werden.

Melder (Reporters)

Belegtmelder, die die Rückmeldung von Lokadressen unterstützen, melden diese Adresse an einen JMRI-Melder. Jede Melderadresse hat eine Belegtmelderadresse mit der gleichen Nummer.

JMRI BiDiB Tools

Wenn JMRI über dieses System mit einer Anlage verbunden ist, wird ein BiDiB-Menü angezeigt:

- **Traffic Monitor**
- **Start/Stopp BiDiB über TCP Server**

Der Server hört auf Port 62875 auf eingehende Clients, leitet alle Nachrichten des Clients an die Systemverbindung weiter und sendet alle von der Verbindung eingehenden Antwortnachrichten an alle verbundenen Clients. Das ist so etwas wie das Gegenteil der oben erwähnten „BiDiB over TCP“-Systemverbindung.

BiDiB-Wizard

BiDiB-Hardware kann normalerweise in vielen Aspekten konfiguriert werden, einschließlich der Definition von Makros. Dies ist nicht im Umfang von JMRI. Stattdessen gibt es den BiDiB-Wizard. Der BiDiB-Wizard ist ein externes Tool - ebenfalls in Java geschrieben - und verwendet die gleiche zugrunde liegende BiDiB-Bibliothek (jbidibc).

Mit diesem Tool werden alle Knotenparameter konfiguriert, z.B: der Knoten-Nutzername, die Portkonfigurationen und viele andere Parameter. Daher ist kein weiteres Konfigurationstool in JMRI integriert. Der BiDiB-Wizard kann mit dem oben genannten BiDiB-über-TCP-Server verbunden werden.

Signale konfigurieren

Anfangs gab (und gibt) es in JMRI dafür Signal-Heads, dort definiert man einzelne Lampen, die ihrerseits „Turnouts“ steuern, denen dann BiDiB-Ports zugeordnet werden können.

In BiDiB sind für Signale allerdings Accessories vorgesehen, die nicht nur Ein und Aus können, sondern die eine Anzahl von „Aspects“ haben und über Macros eingestellt werden kann, welche Lampen bei welchen Aspect leuchten sollen. Nebenbei wird bei Accessories auch die Übertragung zur Hardware mittels Bestätigungs-Meldungen gesichert. Die Programmierung solcher Accessories und Macros geht mit dem BiDiBWizard, bitte die Doku dazu dort lesen.

In JMRI wurden dann „Signal Masts“ erfunden, die ein eigenes Objekt darstellen, der Systemname wird von JMRI allerdings vorgegeben, enthält aber in Klammern eine Connection-spezifische Adresse (ein Connection ist das Bindeglied zur jeweiligen Hardware, man könnte auch „Treiber“ sagen). Man sieht schon, hier muss es eine explizite Unterstützung für BiDiB geben. Das ist auch der Fall, es gibt ein extra SignalMast Objekt „BiDiB Accessory Aspect“, das eben ein BiDiB-Accessory unterstützt.

Wie legt man also ein Signal Mast an? Man öffnet die Tabelle „Signal Masts“ unter Tools→Tables und klickt „Add“ rechts unten.

Der folgende Dialog erscheint:

Window Help

User Name: S42

Signal System: DB HV 1969

Mast Type: Combined exit + distant signal at short breaking distance

Mast Driver: BiDiB Accessory Aspect

BiDiB Connection: BiDiB

BiDiB Accessory Address: N210:3

Hp00
Aspect ID: 0
 Disable Aspect

Hp0/Sh1
Aspect ID: 1
 Disable Aspect

Hp1+Vr0
Aspect ID: 2
 Disable Aspect

Hp1+Vr1
Aspect ID: 3
 Disable Aspect

Hp1+Vr2
Aspect ID: 4
 Disable Aspect

Hp2+Vr0
Aspect ID: 5
 Disable Aspect

Hp2+Vr1
Aspect ID: 6
 Disable Aspect

Hp2+Vr2
Aspect ID: 7
 Disable Aspect

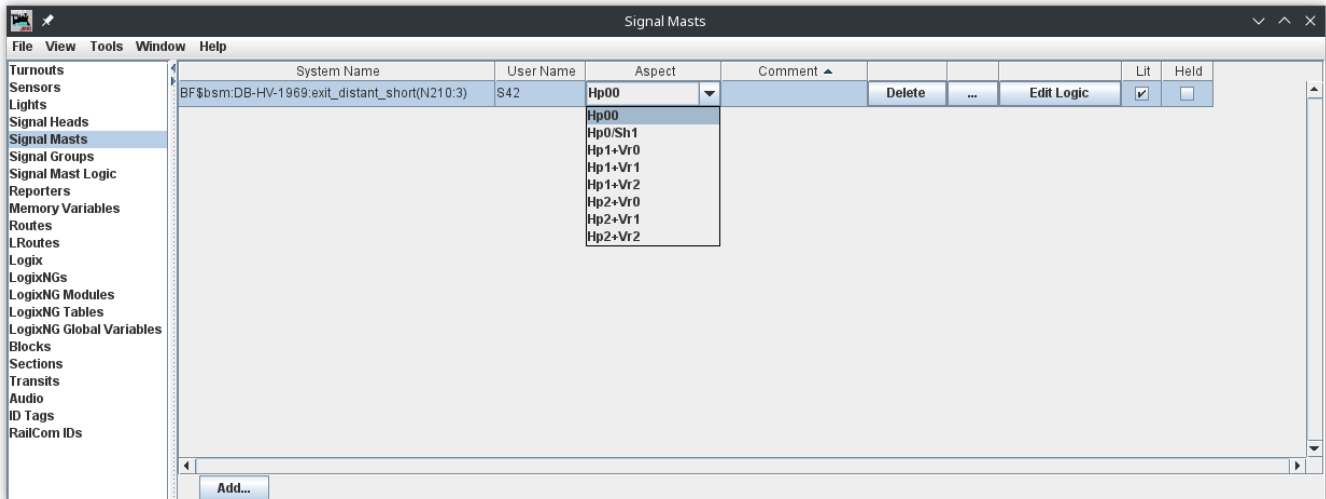
Copy Aspect IDs from Mast:

Cancel Create

User Name ist frei wählbar, als Signal System muss „DB HV 1969“ ausgewählt werden, denn damit werden die Aspekte des Signals definiert. In Mast Type ist dadurch eine Liste erstellt, die die Signaltypen im HV-System darstellt. JMRI kennt also das deutsche HV-System. In diesem Fall ein Ausfahrtsignal mit Vorsignal. Als Mast Driver muss natürlich „BiDiB Accessory Aspect“ gewählt werden, die Connection ist BiDiB und unter Accessory Address wird die BiDiB-Adresse des Accessories eingestellt, wie man sie von den anderen Objekten kennt, in diesem Fall ist es ein BiDiB-Knoten mit dem Username „N210“ und dort Accessory 3 (man kann auch explizit „N210:a3“ schreiben, aber bei Knoten, die Accessories haben, ist „a“ default).

Die folgenden Felder geben die möglichen Aspekte eines solchen Signals an, es ist die Aspekt-Nummer einzutragen, die man mit dem BiDiBWizard in der Hardware definiert hat. Im Beispiel wurden die Aspects einfach von oben nach unten durchnummeriert, nur HP00 sollte immer Aspekt 0 haben.

Der Button „Create“ erzeugt das Signal und die Tabelle enthält einen neuen Eintrag:



Man sieht den etwas kruden Systemname und kann in der Spalte „Aspekt“ gleich mal die verschiedenen Aspekte des Signal ausprobieren. Dieser Signal Mast kann nun in den Panels verwendet werden.

BiDiB Signalmast Systemnamen

Gültige Systemnamen sehen folgendermassen aus:

BF\$bsm:DB-HV-1969:exit_distant_short(N210:3)

wobei die einzelnen Felder bedeuten:

- B - Der Systemverbindungsbuchstabe
- F - Zeigt einen Signalmast an
- \$bsm - Zeigt einen BiDiB Signalmast an
- DB-HV-1969 - Signal System, das deutsche DB HV 1969 System in diesem Fall
- exit_distant_short - Signaltyp, kombiniertes Ausfahrtsignal mit Vorsignal in diesem Fall
- N210:3 - BiDiB Accessory Adresse, Accessory 3 des BiDiB Knotens N210 in diesem Fall

From: <https://forum.opendcc.de/wiki/> - **BiDiB Wiki**

Permanent link: <https://forum.opendcc.de/wiki/doku.php?id=steuerungsprogramme:jmri>

Last update: **2024/02/13 21:59**

