

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.

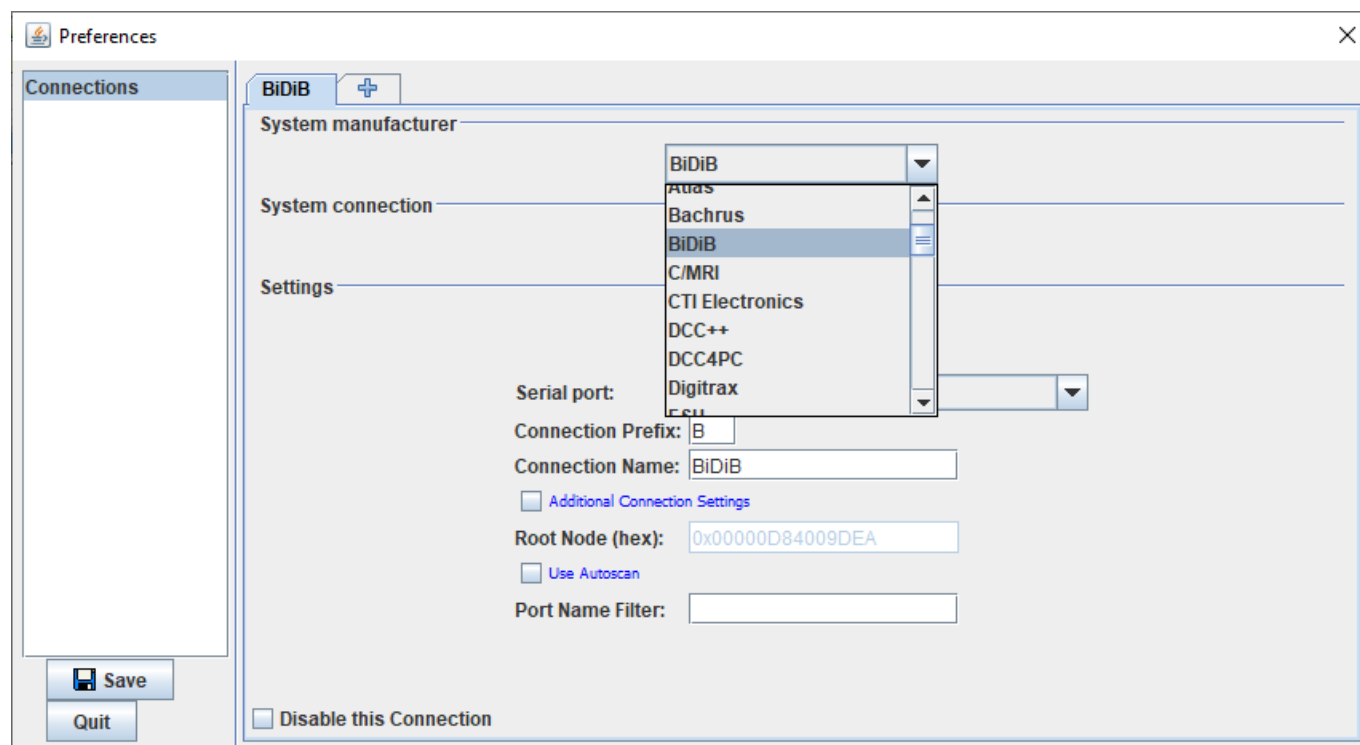
BiDiB über TCP

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



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:

Add/Edit Signal Mast

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 <input type="checkbox"/> Disable Aspect	Hp0/Sh1 Aspect ID: 1 <input type="checkbox"/> Disable Aspect
Hp1+Vr0 Aspect ID: 2 <input type="checkbox"/> Disable Aspect	Hp1+Vr1 Aspect ID: 3 <input type="checkbox"/> Disable Aspect
Hp1+Vr2 Aspect ID: 4 <input type="checkbox"/> Disable Aspect	Hp2+Vr0 Aspect ID: 5 <input type="checkbox"/> Disable Aspect
Hp2+Vr1 Aspect ID: 6 <input type="checkbox"/> Disable Aspect	Hp2+Vr2 Aspect ID: 7 <input type="checkbox"/> Disable Aspect

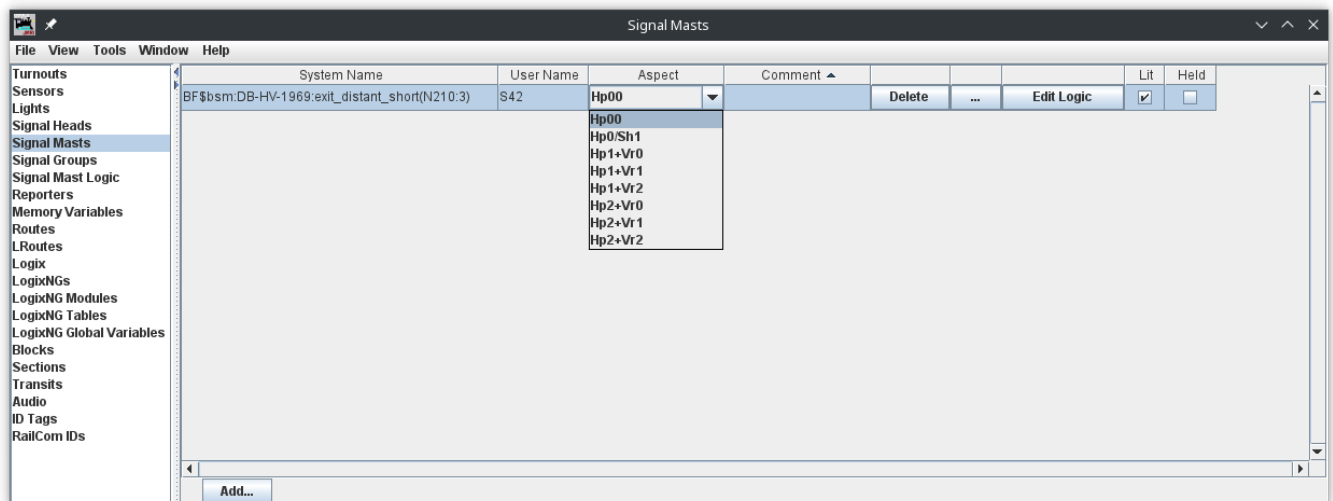
Copy Aspect IDs from Mast:

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 Aspekts des Signal ausprobieren. Dieser Signal Mast kann nun in den Panels verwendet werden.

From:

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

Permanent link:

<https://forum.opendcc.de/wiki/doku.php?id=steuerungsprogramme:jmri&rev=1683646282>

Last update: **2023/05/09 17:31**

