

Modellstell(netz)werk

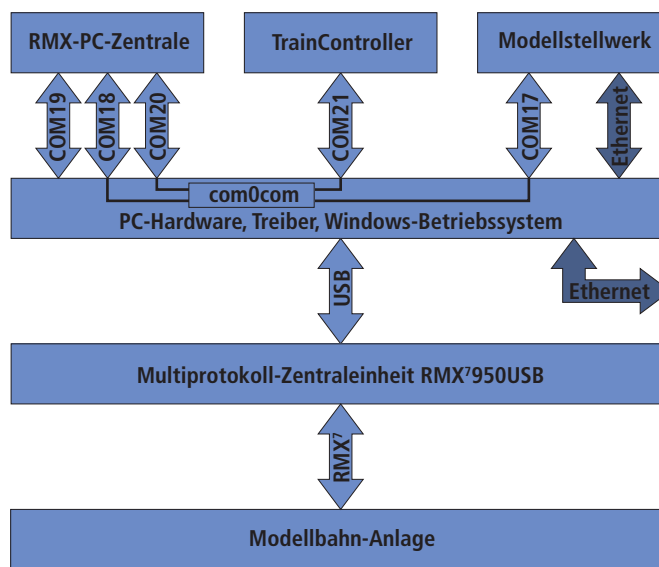
Statt „nur“ eine Schnittstelle in eine Zentraleinheit zu bauen, ergänzte man bei rautenhaus digital die Hardware – Zentraleinheit nebst USB-Schnittstelle –, um eine dazu maßgeschneiderte PC-Software. Sie ist keine Software zur Automatisierung von Abläufen, sondern mehr ein Kommunikations-Server als Bindeglied zwischen Modellbahn und Steuerungsprogrammen. Dr. Bernd Schneider testete, was sich mit den „virtuellen Interfaces“ der RMX-PC-Zentrale anstellen lässt.

Es ist wie so oft im Leben: Man ist solange zufrieden, bis man etwas Schöneres/Neueres/Schnelleres/ ... sieht. So war es beim Autor erst mit dem DCC-System, dem für anspruchsvolle Steuerungsaufgaben dann doch die Echtzeitfähigkeit fehlte und das daher beim Autor erst durch das Selectrix- und dieses durch das RMX-System ersetzt wurde. Nun sind echtzeitfähige Steuerung mit der Weiterverwendung der DCC-Lokdecoder kombiniert.

Ebenso war es bei dem liebgewonnenen Steuerungsprogramm. Längst nicht alle Funktionen und Möglichkeiten des TrainControllers wurden genutzt – die sehr realistisch wirkende Oberfläche des Programms „Modellstellwerk“ hatte es dem Autor jedoch angetan. Aber dafür alle Steuerungsaufgaben vom TrainController in einer anderen Programmlogik noch einmal formulieren?

Anforderungsanalyse

Ideal wäre es, wenn beide Programme mit denselben Daten von der Modellbahn-Anlage versorgt werden könnten, sie also quasi parallel an das Interface angeschlossen wären. Benötigt wird folglich eine Komponente, die mit einem



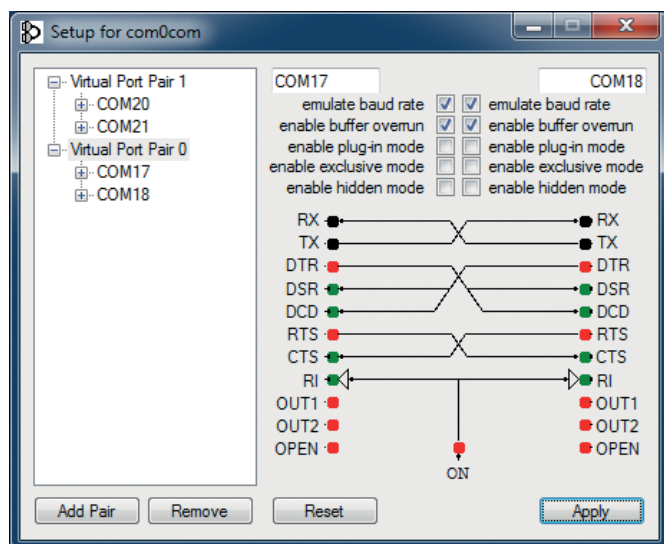
Die Modellbahn wird mit dem RMX-System betrieben, die RMX'950USB dient dabei als Zentraleinheit. Sie ist per USB-Schnittstelle mit einem Windows-PC verbunden (PC1 im Schaubild auf der nächsten Seite). Auf diesem läuft die Software RMX-PC-Zentrale, der TrainController und – seit neuestem – Modellstellwerk. Die RMX-PC-Zentrale hält über COM19 die

exklusive Verbindung zur Zentraleinheit, erzeugt aber auf COM18 (SX) und COM20 (RMX) zwei virtuelle Interfaces. Per Software-Null-Modem (com0com) werden diese auf COM21 (RMX) dem TrainController und auf COM17 (SX) dem Modellstellwerk weitergeleitet. Modellstellwerk offeriert seinerseits weitere Verbindungsmöglichkeiten per Netzwerk („Ethernet“).

Interface verbunden ist, dessen Daten aber auf mehrere Ausgänge weitergibt.

Bei der Weitergabe von Daten vom Interface zu den Ausgängen wäre dies noch vergleichsweise einfach zu bewerkstelligen, da die Daten ja „nur“ vervielfältigt werden müssen.

Ungleich komplizierter ist der umgekehrte Weg, bei dem mehrere Ausgänge zu Eingängen werden. Diese müssen dann ihre Daten nicht nur konsolidiert an das Interface weitergeben, sondern sich auch alle gegenseitig auf den jeweils aktuellen Stand bringen.



Im Null-Modem-Emulator com0com werden die COM-Ports 17 und 18 sowie 20 und 21 miteinander verbunden. Daten, die auf COM17 ausgegeben werden, werden auf COM18 empfangen – und umgekehrt. Ein Programm, das jetzt mit COM17 verbunden wird, verhält sich so, als wäre es bspw. mit einem 2-Bus-Interface SLX852 verbunden.



COM18 wird in der RMX-PC-Zentrale mit dem virtuellen SX-Interface, hier vom Typ SLX852, verbunden. Die beiden Busse des virtuellen Interface – „virtueller SX0“ und „virtueller SX1“ werden mit den internen Bussen V SX2 und RMX1 verbunden. Damit werden Schaltbefehle direkt zum und Belegtmeldungen direkt vom RMX1-Bus übertragen.

Leider ist eine solche Lösung mit keiner der in den Digitalsystemen verwendeten Schnittstellen direkt möglich: Die „alte“ serielle Schnittstelle (RS232), die „neue“ USB-Schnittstelle (USB = Universal Serial Bus) und auch die „ganz neuen“ Netzwerk-Schnittstellen nehmen immer nur eine Punkt-zu-Punkt-Verbindung zwischen einem Sender und einem Empfänger vor. Diese können ihre Rollen zwar tauschen, die Pärchen bleiben aber fest.

Virtualisierung

Das RMX-System von rautenhaus digital kann natürlich ohne PC-Unterstützung laufen, jedoch bietet die PC-Unterstützung mit der Software RMX-PC-Zentrale eine Reihe zusätzlicher Funktionen. Im hier betrachteten Fall helfen vor allem die virtuellen Busse und die virtuellen Interfaces der RMX-PC-Zentrale.

Ein reales Interface – wie das in der RMX7950USB verbaute – korrespondiert im PC mit zwei Bussen: dem RMX0-Bus, der für die Ansteuerung der Lokdecoder zuständig ist, und dem RMX1-Bus, der die Aufgaben des Schaltens und Meldens übernimmt.

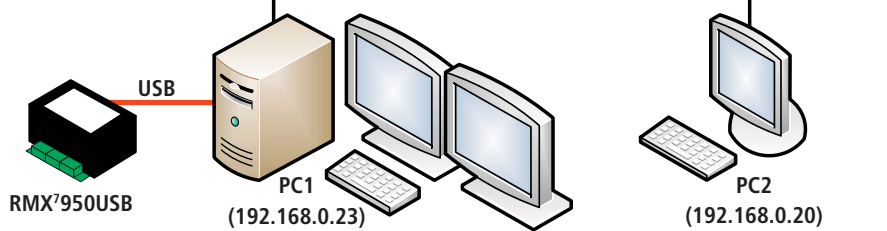
Zudem verfügt die RMX-PC-Zentrale über sogenannte virtuelle Interfaces, die in der Lage sind, Inhalte eines Busses wieder auf eine serielle Schnittstelle nach außen zu geben. Anstatt nun die Daten, die auf dieser seriellen Schnittstelle ankommen, über ein Null-Modem-Kabel einer anderen seriellen Schnittstelle zuzuführen, wird dies alles durch eine entsprechende Software emuliert („nachgebildet“).

Die physischen seriellen Schnittstellen am PC und auch die Null-Modem-Kabel werden dabei durch ein Pro-

gramm nachgebildet: com0com ist das Helferlein, das dies ermöglicht. Ein (Steuerungs-)Programm, das mit der seriellen Schnittstelle verbunden wird, erkennt keinen Unterschied zwischen dem Betrieb an einem realen, physischen Interface oder an einem virtuellen, durch Software emulierten Interface.

So stellt sich die in diesem Beitrag verwendete Netzwerk-Struktur dar: Zwei PCs sind über einen Switch miteinander verbunden. An diesem ist auch ein WLAN-Zugriffspunkt angeschlossen, der das Netzwerk um ein Funknetzwerk für Mobilgeräte erweitert. Die Verbindung zum DSL-Modem ist nicht zwingend erforderlich.

PC1 besitzt den in der ersten Abbildung gezeigten Aufbau, PC2 wird als reiner Client betrieben und kommuniziert über das Modell-



stellwerk-eigene Protokoll per Netzwerk mit PC1 (Port 10080). Auch die mobilen Endgeräte kommunizieren mit Modellstellwerk auf PC1, optional per SRCP (Port 4303) oder das http-Protokoll (Port 80).

gramm nachgebildet: com0com ist das Helferlein, das dies ermöglicht.

Ein (Steuerungs-)Programm, das mit der seriellen Schnittstelle verbunden wird, erkennt keinen Unterschied zwischen dem Betrieb an einem realen, physischen Interface oder an einem virtuellen, durch Software emulierten Interface.

So können die realen RMX-Busse über ein virtuelles Interface wieder ausgegeben werden. Aus diesem virtuellen Interface kann nun ein weiteres Programm – im hier betrachteten Fall das Steuerungsprogramm TrainController – gespeist werden.

Werden Befehle vom TrainController ausgegeben, so gelangen sie über das virtuelle Interface zur RMX-PC-Zentra-

le, die sie an das reale RMX-Interface weiterleitet. Dieser Prozess läuft auf dem PC so schnell ab, dass die Verzögerung an der Modellbahn-Anlage nicht wirklich messbar ist.

... und der zweite Streich

Die gerade beschriebene Konfiguration ist quasi Standard beim RMX-System. Nun verfügt die RMX-PC-Zentrale aber nur über ein virtuelles RMX-Interface – wie soll jetzt das zweite Steuerungsprogramm (Modellstellwerk) mit der Anlage verbunden werden?

Was für RMX-Busse geht, geht auch für SX-Busse. Es wird ein zweites virtuelles Interface konfiguriert, das ein SX0-/SX1-Bus-Pärchen weiterleitet –

Da sich RMX0- und SX0-Bus auf der Protokollebene unterscheiden, kann der SX0-Bus eines virtuellen Interface nicht direkt mit dem

RMX0-Bus verbunden werden (siehe Abbildung rechts unten, vorhergehende Seite). Stattdessen muss für jede RMX0-Systemadresse einzeln angegeben werden, ob und wenn ja, mit welcher SX-Adresse welches SX-Busses sie korrespondiert. Dies erfolgt logischerweise in der Pflegemaske der Lokdatenbank, denn der RMX0-Bus dient ja allein der Ansteuerung der Lokdecoder. Im Dialogbereich „Erweiterter Translater“ ist dabei für jede Lokadresse, die über das virtuelle Interface angesprochen werden soll, eine Adresse für den erweiterten Translater einzutragen. Zusätzlich ist der bzw. sind die Busse auszuwählen, für den diese Übersetzungsfunktion aktiv sein soll – im hier beschriebenen Fall ist es der VSX2-Bus. Je nach Anzahl der Lokdecoder und deren Adressen kann es sinnvoll sein, die kurze Adresse und die Translater-Adresse identisch zu wählen. Ist dies nicht möglich, muss ein geeignetes, leicht merkbares Schema gefunden oder eben eine Liste geführt werden. Für Lokdecoder, die nicht über das virtuelle Interface angesprochen werden (müssen), bleibt der erweiterte Translater deaktiviert.



sich also wie ein SLX852 von rautenhaus digital verhält. Dieses Interface bedient dann die Software Modellstellwerk.

Zuletzt müssen nun in der RMX-PC-Zentrale der RMX0-Bus mit dem virtuellen SX0-Bus und der RMX1-Bus mit dem virtuellen SX1-Bus in der Art verknüpft werden, dass die Bus-Pärchen jeweils die gleichen Daten führen. Da die Protokolle auf dem RMX1-Bus und dem SX1-Bus identisch sind, kann hier ein direktes Bus-Mapping erfolgen.

Zwischen den unterschiedlichen Protokolle führenden RMX0- und SX0-Bussen müssen die Daten dagegen jeweils in das andere Protokoll übersetzt werden. Dies erledigt die Funktion „Erweiterter Translater“ der RMX-PC-Zentrale.

Ergebnis 1

Mit der bis hierher beschriebenen Konfiguration laufen jetzt beim Autor die RMX-PC-Zentrale sowie die Steuerungsprogramme TrainController und Modellstellwerk parallel auf einem PC (Intel Core2Duo, 2.16 GHz, 8 GByte RAM, Windows 7), ohne sich gegenseitig technisch zu behindern.

Bei der Benutzung der Steuerungsprogramme sollte der Anwender aber Eigendisziplin beweisen. Werden beiden Programmen Steuerungsaufgaben – bspw. Zug- oder Fahrplanautomatiken – übertragen, so können sie sich sehr wohl gegenseitig stören.

Im vorliegenden Fall übernimmt TrainController alle Steuerungsaufgaben, Modellstellwerk wird lediglich auf einem separaten Monitor angezeigt und dient dem Stellen von Fahrstraßen bei manuellem Fahren und Steuern sowie der Anzeige der Stellische im Betrieb.

Dabei ergab sich dann schnell der Wunsch, diese Eingabe- und Anzeigemöglichkeiten an einer weiteren Stelle verfügbar zu haben.

Modellstellwerk-Schnittstelle

Das Programm Modellstellwerk verfügt über eine eingebaute Netzwerkschnittstelle, über die Modellstellwerk mit Programminstanzen auf anderen Computern in Verbindung treten kann.

Maximal fünf Modellstellwerk-Programme auf fünf Rechnern können als Clients mit dem zentralen Modellstellwerk als Server auf einem sechsten PC zusammenarbeiten. Während die Clients ausschließlich der Eingabe von Stellbefehlen und der Ausgabe der Wei-



chen- und Signalstellungen dienen, übernimmt der Server die zentrale Steuerung und die Weitergabe aller Befehle an das Digitalsystem.

Die Installation auf dem Client ist einfach: Das Programm wird über die Installationsroutine entpackt, danach werden die auf dem Server-Rechner verwendeten Dateien zur Anlagenbeschreibung auf den Client kopiert. Wichtig ist, dass Änderungen immer auf dem Server vorgenommen werden und von dort die geänderten Dateien wieder zu den Clients kopiert werden.

Web-Interface

Zusätzlich verfügt Modellstellwerk über einen eingebauten Webserver, sodass Funktionen auch per Web-Browser von jedem am Netzwerk angeschlossenen Rechner aufgerufen werden können. Dazu reicht die Eingabe von http:// mit der IP-Adresse des Rechners, auf dem

Modellstellwerk empfängt seine Daten über die Schnittstelle COM17 – die wiederum mit COM18 verbunden ist, auf der die RMX-PC-Zentrale die Daten eines virtuellen 2-Bus-Interface vom Typ SLX852 ausgibt. Dieses Interface wird in Modellstellwerk als einzige Zentraleinheit eingetragen, folglich ist sie auch die Standard-Zentrale für alle Steuerungsaufgaben. Im unteren Drittel des Einstellungsdialogs können die zusätzlichen Schnittstellen aktiviert werden. Ist beim oberen Feld „Port“ eine 0 eingetragen, so ist keine Kommunikation mit einer anderen Modellstellwerk-Instanz möglich, ansonsten ist die Nummer die Port-Nummer, die auch beim Client eingetragen ist. Das Web-Interface ist ebenso aktiviert (Standard-Port 80), wie der SRCP-Server, der auf Port 4303 angesprochen wird.

Modellstellwerk läuft. Die Möglichkeiten im Webserver sind zwar beschränkt, können aber durchaus produktiv auf solchen Client-Geräten eingesetzt werden, auf denen keine andere Möglichkeit besteht.

Die mit Modellstellwerk gelieferten Vorlagen können nach eigenen Wünschen verfeinert werden, sodass mit ein wenig HTML-Kenntnissen gänzlich neue Oberflächen geschaffen werden können. Standardmäßig läuft der Webserver auf Port 80; wird dieser verändert, muss die neu vergebene Portnummer beim Aufruf per Doppelpunkt an die IP-Adresse angehängt werden.

SRCP

Richtigerweise lenkte der Autor von Modellstellwerk die Arbeit in die Einbettung von SRCP, anstatt das Web-Interface weiter auszubauen. Mit SRCP besteht die Möglichkeit, Digitalsysteme

Die Abbildung links zeigt den bereits bekannten Einstellungsdialog von Modellstellwerk, hier aber von der Programminstanz, die als Client auf PC2 laufen soll. Als Zentrale muss PC-Netzwerk gewählt werden, die weiteren Dialogfelder werden daraufhin deaktiviert.

In der Feldgruppe „PC-Netzwerk“ ist dann die Adresse des Computers einzutragen, auf dem die per Interface mit der Modellbahn verbundene Instanz von Modellstellwerk als Server läuft – dies ist im hier dargestellten Szenario PC1 mit der IP-Adresse 192.168.0.23. Die Port-Nummer ist identisch zur Einstellung auf dem Server zu wählen. Ist alles korrekt eingestellt und der



Dialog geschlossen, wird die Verbindung aufgebaut. Auf dem Server kann dann im Dialog „Netzwerk-Server“ eine Liste der mit dem Server verbundenen Clients angezeigt werden. Die Abbildung rechts unten zeigt die IP-Adresse von PC2 (Client) und seinen Rechner- und Domain-Namen in zwei Zeilen. War der Verbindungsaufbau nicht erfolgreich, so sollte als erstes Modellstellwerk auf dem Server einmal geschlossen und neu gestartet werden, damit der dort eingebaute Web-Server mit den ggf. geänderten Einstellungen arbeitet.



Client und Server

Die Client-Server-Technologie ist vergleichsweise alt. Schon früh diente sie dazu, Aufgaben in einem Netzwerk von Computern arbeitsteilig zu bewältigen. Meist speichert der Server alle notwendigen Daten und stellt einen mehr oder minder großen Teil der Verarbeitungslogik (Server-Programm) bereit. Clients greifen dann über diese Verarbeitungslogik auf Daten zu und verändern sie bspw. durch die eigene Verarbeitungslogik oder die des Servers. Den Clients kommt häufig in erster Linie die Aufgabe der Interaktion mit dem Anwender zu, also das Entgegennehmen von Eingaben und das Anzeigen von Ergebnissen. Die ganzen Dienste im WWW sind auf diese Weise aufgebaut. Daten (z.B. Web-Seiten) sind auf den Web-Servern gespeichert, die von den Clients per Web-Browser abgerufen und auf dem Bildschirm des Clients angezeigt werden können.

mit einem einheitlichen Befehlssatz anzusprechen – egal, welches System oder von welchem Hersteller. Somit abstrahiert SRCP vom eingesetzten Digitalsystem.

SRCP wird von Modellstellwerk auf dem generell für SRCP reservierten Port 4303 ausgegeben. SRCP-Clients können über diesen Port Befehle an den SRCP-Server übergeben oder von ihm Informationen abfragen. Voraussetzung dafür ist natürlich, dass die Clients per Netzwerk – kabel- oder funkbasiert – mit dem Server verbunden sind. Aktuell sind über 20 verschiedene (Open Source-) Clients verfügbar, die von einer einfachen Loksteuerung per PC bis zur Android-App reichen.

Ergebnis 2

Mit der realisierten Konfiguration ist es nun möglich, die Modellbahn von mehreren (Bildschirm-)Stellwerken aus zu bedienen – dank Modellstellwerk sogar in verschiedenen Designs (EStw, Sp-DrS60 gezeichnet oder fotorealistisch, SpDrL60).

Zusätzlich ist die Einbindung mobiler Endgeräte (Smartphones) über SRCP-fähige Apps – wie beispielsweise die kostenlos verfügbare App von Michael Blank – sogar spontan möglich. Auch die in MIBA 5/2012 ab S. 39 vorgestellte Möglichkeit, ein Tablet als mobilen Touchscreen-Bildschirm einzusetzen, besteht weiterhin.

Protokoll: SRCP

Unter Protokoll wird eine Sammlung von Regeln verstanden, nach denen eine Kommunikation abläuft. In den Regeln können sowohl physische als auch logische Eigenschaften festgelegt werden, bspw. welche Zeichen zur Kommunikation verwendet werden dürfen, welche Bedeutung die daraus zusammengesetzten Nachrichten haben und in welcher Reihenfolge sie zu verwenden sind.

Das Simple Railroad Command Protocol (SRCP) entstand im Jahr 2000 aus einer im Internet intensiv geführten Diskussion über eine einheitliche, herstellerneutrale Ansprache digitaler Steuerungssysteme. Nach einer anfänglichen alleinigen Unterstützung des Protokolls aus dem Open Source-Umfeld finden sich nun auch zunehmend kommerziell orientierte Produkte, die das SRCP integrieren, zum Beispiel das Steuerungsprogramm Modellstellwerk oder die kürzlich von JSS Elektronik vorgestellten Zentraleinheiten MBS2 und MBS3.

Fazit

Die RMX-PC-Zentrale fungiert hier als „Modellbahn-Server“, der eine trennende Ebene zwischen Steuerungsprogramm(en) und Modellbahn-Interface(s) einzieht und den Datenaustausch zwischen allen Client-Komponenten koordiniert. Somit ist auch die Benutzeroberfläche der RMX-PC-Zentrale ein Client.

Derzeit „spricht“ jedes Digitalsystem eine eigene Sprache, dementsprechend müssen die Steuerungsprogramme die Sprachen der Digitalsysteme sprechen, die unterstützt werden sollen. Dies wird erst dann zu einem technischen Problem, wenn mehrere (spezialisierte) Programme gemeinsam arbeiten sollen, bspw. Bedienfunktionen auf Mobilgeräten genutzt werden sollen. Diese Zusammenarbeit zwischen den Programmen („Interoperabilität“) wird durch das Fehlen einer gemeinsamen, systemübergreifenden Sprache erschwert.

Wünschenswert wäre aus Anwendersicht die Verwendung einer gemeinsamen standardisierten, aber offenen Sprache, die der Kommunikation zwischen Modellbahn-Server (oder Interface), Steuerungsprogrammen und Apps auf Mobilgeräten etc. dient. Mit SRCP (Simple Railroad Command Protocol) ist eine solche Sprache entworfen worden, die sich bereits in einer Reihe von Anwendungen bewährt hat.

Einen Schritt in diese Richtung hat die aktuelle Version von Modellstellwerk getan: Hier ist SRCP zum festen Bestandteil geworden, wodurch Modellstellwerk zum kommunikativen Mittelpunkt der Modellbahnsteuerung werden kann. Generell sollte der SRCP-Server möglichst nah am oder sogar im Interface angesiedelt sein, was jedoch im betrachteten Szenario nicht möglich ist. *Dr. Bernd Schneider* 