

Dezentrale Reaktionen

Mit der Einführung der Intelli-Drive 2-Digitaldecoder baut Uhlenbrock die mit Lissy begonnene Idee der dezentralen Steuerung und Automatisierung weiter aus: Intellimatic ist eine programmierbare, decoderinterne Ablaufsteuerung, die das Fahrzeug auch ohne Hilfe des Computer auf Situationen des Anlagenbetriebs reagieren lässt.

Die Möglichkeiten, Fahrzeugdecoder an die individuellen Anforderungen und Eigenschaften anzupassen, entwickeln sich ständig weiter. Erlaubten die ersten Decoder nur das Einstellen der Adresse, so kamen schnell Anpassungen des Fahrverhaltens wie Anfahrstufe bzw. Mindestgeschwindigkeit, Höchstgeschwindigkeit sowie das Beschleunigungs- und Bremsverhalten hinzu.

Das Einstellen der Eigenschaften – unter Modellbahnern oft als Programmierung bezeichnet – erfolgt durch das Einschreiben von Werten in bestimmte Speicherzellen des Decoders, die mit CV für configuration variable bezeichnet werden. Jede CV hat eine Nummer, die quasi die Adresse der Speicherzelle repräsentiert.

Mit steigender Anzahl an CVs sinkt die Einfachheit, Einstellungen vornehmen zu können. Die Fahrstufentabelle eines Decoders mit 128 Fahrstufen über einen Handregler zu füllen ist weder einfach noch angenehm: So entstanden spezielle Anwendungen zur „Programmierung“ der Decoder. Teilweise arbeiten diese Programme mit der Zentrale oder einem speziellen Programmieradapter zusammen.

IDE für Intellimatic

Programmierer bezeichnen ihre Arbeitsumgebung mit IDE (integrated development environment, integrierte Entwicklungsumgebung). Nach dieser Philosophie entwarf Uhlenbrock Digi-Test als eine Kombination aus Programmiergerät für Fahrzeugdecoder,



das per USB an den Windows-Computer angeschlossen wird und der Software gleichen Namens.

Über DigiTest lassen sich alle Decodereigenschaften einstellen; neben der Programmierung aller CVs wird sowohl das erweiterte Funktionsmapping wie auch die Intellimatic-Ablaufsteuerung unterstützt.

Intellimatic

Mit Intellimatic bezeichnet Uhlenbrock die Möglichkeit, (einfache) Abläufe im Decoder zu hinterlegen, beispielsweise über ein zeitlich gestaffeltes Aufrufen von Funktionen, die das Aufrufen einer Lokomotive vor Fahrtantritt simulieren und mit nur einem Tastendruck zu starten sind.

Eine Ablaufsteuerung in Intellimatic besteht aus einem oder mehreren sogenannten Einzelabläufen. Ein Einzelablauf beschreibt das gewünschte Verhalten des Fahrzeugs in einer bestimmten Situation. Während ein Einzelablauf ausgeführt wird, hat die Intellimatic die Kontrolle über das Fahrzeug und bestimmt Fahrtrichtung, Geschwindigkeit und die Funktionen. Befehle der Zentrale – und damit auch einer eventuell vorhandenen Computersteuerung – werden währenddessen ignoriert, was beim Entwurf der Abläufe beachtet werden muss.

Die Abläufe werden in einer einfachen, visuellen Programmiersprache formuliert. Diese Programmiersprache besteht aus acht Befehlen und vier Elementen für den Aufbau von Kontrollstrukturen.

Aktuell unterstützen zwei Decoder-Typen Intellimatic: Die 22 x 15 x 3,8 mm großen PluX22-Lokdecoder 74560 und 75570 leisten 1,2 A am Motorausgang, die kleineren Next18-Decoder 73235 messen 9,5 x 7,8 x 2,4 mm und leisten 0,8 A am Motorausgang. Alle Decoder können in DCC-, Motorola- und Selectrix-Systemen betrieben werden, der 74570 unterstützt auch mfx. Die Decoder finden sich auch bei Piko als 56400 bis 56406.

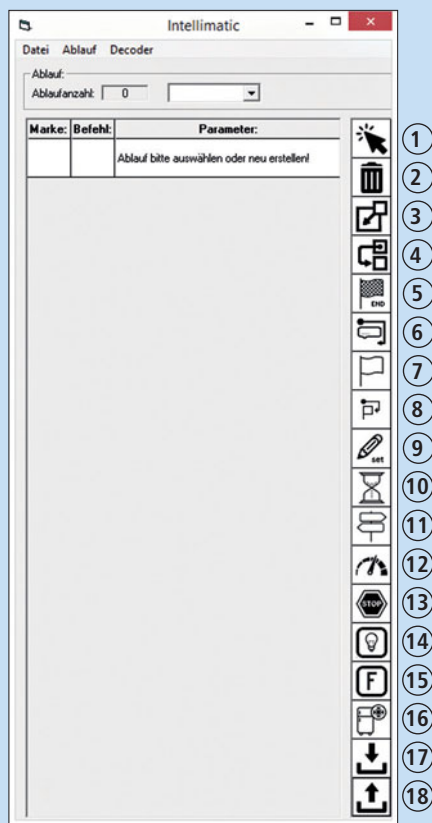
Befehle

Der verfügbare Satz an Befehlen ist recht überschaubar: Es können die Ausgänge des Decoders ein-, aus- oder umgeschaltet werden ⑭ (siehe Seite 23). Gleiches gilt für den Zustand der Funktionen ⑮. Hiermit lassen sich alle Ausgänge und Funktionen (etwa Geräusche) in Abläufe einbinden.

Hinzu kommen Befehle zur Steuerung der Fahrtrichtung ① und Geschwindigkeit ⑫ sowie das Auslösen eines Nothalts ⑬.

Zur Einbettung der Intellimatic-Abläufe in andere Steuerungsabläufe – zum Beispiel durch Lissy oder ein Computerprogramm gesteuert – dienen die drei untersten Schaltflächen: ⑯ kann Fahrstufe, Fahrtrichtung und Funktionen „einfrieren“, sprich, der Decoder behält auch nach Beendigung des Ablaufs so lange die durch die Intellimatic bestimmten Werte bei, bis diese durch eine wirkliche Änderung von außen überschrieben werden.

Wird beispielsweise in einem Intellimatic-Ablauf die Fahrtrichtung umgekehrt, so würde diese Einstellung bei Beendigung des Ablaufs direkt wieder



So stellt sich das Editor-Fenster für die Intellimatic-Abläufe beim Programmstart dar: Die am rechten Rand angeordneten Schaltflächen ① bis ④ dienen zur Steuerung des Editors, wobei Schaltfläche ③ als die wohl wichtigste dem Einfügen neuer Zeilen dient. Beim Verändern von Abläufen ist Schaltfläche ④ hilfreich, sie ermöglicht das Verschieben von Programmzeilen. Die Schaltflächen ⑤ bis ⑧ fügen Kontrollstrukturen in den Ablauf ein, die Schaltflächen ⑨ bis ⑱ sind die eigentlichen Befehle.

mit dem ursprünglichen Wert überschrieben und wiederum gewechselt – was in den allermeisten Fällen unerwünscht sein dürfte. Mit dem Einfrieren-Befehl kann dies so lange verhindert werden, bis die Fahrtrichtung über das Gleissignal von der Zentrale geändert wird.

Soll bei Abläufen – beispielsweise eine Pendelzugsteuerung – aus dem Stillstand nicht mit einer fest im Ablauf vorgegebenen Geschwindigkeit, sondern mit der bei der Übernahme durch den Intellimatic-Ablauf eingestellten Geschwindigkeit gefahren werden, so kann diese Geschwindigkeit beziehungsweise Fahrstufe mit ⑰ gespeichert und mit ⑱ wieder an den Decoder übermittelt werden. Der Brems- und Beschleunigungsvorgang selbst wird durch die im Decoder vorgegebenen Einstellungen bestimmt.

Kontrollstrukturen

Kontrollstrukturen steuern den Programmablauf. Sie legen fest, wann ein Ablauf startet und endet ⑤, zudem erlauben sie die Formulierung von Verzweigungen („wenn-dann“). Dazu ist einerseits ein Überprüfen von Bedingungen ⑥, andererseits das Springen zu einer bestimmten Stelle im Ablauf durch einen Sprungbefehl ⑧ erforderlich.

Stellen im Ablauf, an die gesprungen werden soll, werden durch Sprungmarken ⑦ definiert. Durch einen Sprungbefehl kann nur genau eine Sprungmarke angesprungen werden, jedoch ist es möglich, dass mehrere Sprungbefehle die gleiche Sprungmarke – beispielsweise die Startmarke – anspringen, um einen Ablauf erneut durchlaufen zu lassen.

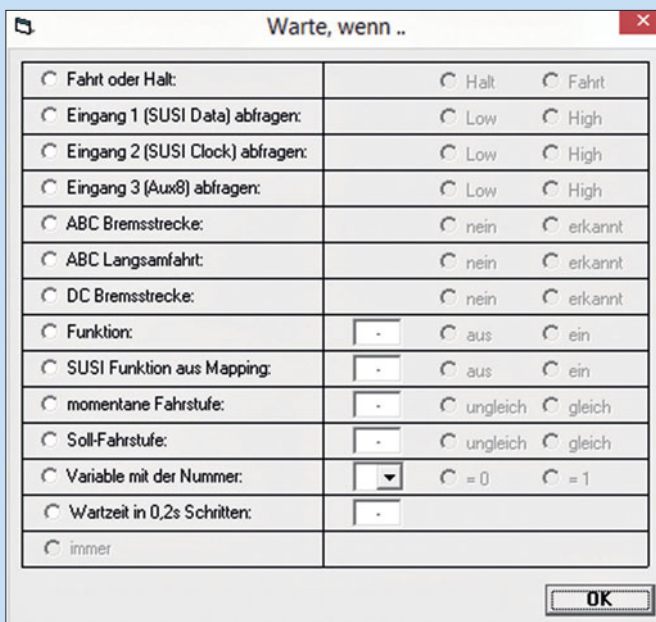
Über die Verzweigung ist es also möglich, einen Abschnitt eines Ablaufs mehrfach zu durchlaufen oder den Ablauf situationsabhängig zu variieren.

Soll der Ablauf pausieren, die Kontrolle weiterhin aber bei der Intellimatic bleiben, so ist die Warte-Anweisung ⑩ zu verwenden. Erst bei Erreichen einer Ende-Anweisung ⑤ geht die Kontrolle an die Zentrale zurück beziehungsweise der Decoder reagiert wieder auf die Gleissignale.

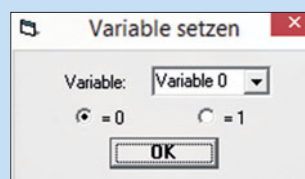
Ereignisse und Variablen

Abläufe werden in der Intellimatic durch Ereignisse gestartet, kontrolliert und beendet. Ein Ereignis kann die Änderung der Geschwindigkeit oder der Fahrtrichtung, das Erreichen einer bestimmten Fahrstufe, das Ein- oder Ausschalten einer Funktion oder die Einfahrt in eine Bremsstrecke sein.

In komplexen Abläufen besteht mitunter die Notwendigkeit, Zustände des



Dieser Dialog zeigt alle möglichen Ereignisse beziehungsweise Bedingungen, auf die Intellimatic-Abläufe in den Befehlen Start, Ende ⑤, Warten ⑩ und Springe ⑥ reagieren können. Die unterste Option „immer“ ist nicht bei allen Befehlen sinnvoll, daher wird sie wie hier beim Warte-Befehl ausgegraut. „Immer warten“ würde das Fahrzeug für immer im Ablauf gefangen halten.



Während die Abfrage von Variablen in Kontrollstrukturen

(siehe oben) erfolgt, werden die Werte im Programm gesetzt. Die Variablen 0 bis 7 können jeweils den Zustand „0“ oder „1“ speichern. Die Bedeutungen der Variablen wie auch der Werte können vom Ersteller definiert werden. Dabei ist es empfehlenswert, diese Bedeutungen für eine mögliche Fehlersuche oder spätere Erweiterungen zu dokumentieren. Eine Textdatei oder Notizen auf Papier reichen dazu völlig aus.

Ablaufs zu speichern ⑨. Hierzu können in acht Variablen binäre Werte abgelegt werden. Die Werte wirken wie Schalter, die in die Stellung 0 oder 1 gebracht werden können und auf deren Stellung im weiteren Verlauf desselben wie auch aus anderen Abläufen reagiert werden kann.

Auf diese Weise kann ein einzelner Ablauf ein „Gedächtnis“ bekommen und sich beispielsweise bei erneuter Ausführung anders verhalten als bei der ersten Ausführung.

Die Variablen ermöglichen auch eine Kommunikation zwischen zwei oder mehr Abläufen: So kann ein Ablauf auf eine Situation reagieren, die ein ande-

rer, gegebenenfalls bereits beendeter Ablauf vorgefunden hat.

Die Variablen behalten bei der Trennung vom Gleissignal ihren eingestellten Wert nicht, sondern werden bei Betriebsbeginn stets mit dem Wert 0 initialisiert. Dies ist beim Entwurf von Abläufen zu berücksichtigen, da beispielsweise nach einer Stromunterbrechung nicht mehr auf die vorherigen Werte zurückgegriffen werden kann.






Entwurf von Abläufen

Schaut man sich die Möglichkeiten an, die die verfügbaren Befehle bieten, so entsteht unweigerlich eine Fülle von Ideen, gerade in Kombination mit den vielfältigen Einstelloptionen und dem erweiterten Funktionsmapping. Diese stehen jedoch mitunter auch in Konkurrenz zu einem Ablauf: Soll beim Bremsen nicht nur das Bremsenquietschen ertönen, sondern auch die Bremscheiben (mit LEDs) zum Glühen gebracht werden, so kann dies per Ablauf oder eben durch das Funktionsmapping erfolgen.

Anders sieht es aus, wenn – beispielsweise und wenig vorbildnah – die Bremscheiben beim ersten Bremsen nicht glühen sollen, sondern erst beim zweiten Bremsen. Benötigt wird hier eine Variable, die zunächst den Wert „0“ hat und im Ablauf für das erste Bremsen auf „1“ gesetzt wird. Die Variable hat somit die Bedeutung „Wurde schon mal gebremst?“, wobei der Wert 0 „nein“ und der Wert 1 „ja“ repräsentiert.

Das Starterereignis für den Ablauf ist das Einfahren in eine Bremsstrecke. Ist der Wert der Variablen 0 „1“, so muss die Bremscheiben-LED (hier Funktion 13) aktiviert werden, andernfalls (also falls die Variable 0 den Wert „0“ hat) soll das Einschalten der Funktion 13 übersprungen werden. Dieses Verhalten entspricht exakt dem Sprungbefehl.

Danach wird Variable 0 auf „1“ gesetzt. Hat sie bereits diesen Wert, ändert sich durch das neuerliche Setzen nichts. Der Ablauf endet beim Verlassen der Bremsstrecke.

Marke:	Befehl:	Parameter:
		wenn ABC Bremsstrecke erreicht
		wenn Variable 0 = 0
		Funktion 13 = ein
		Variable 0 = 1
		wenn keine DC Bremsstrecke

Ablauf „Bremsen glühen“ beim wiederholten Bremsen

Nun unterstützen die Decoder sowohl ABC- als auch DC-Bremsstrecken. Soll das Verhalten bei beiden Bremsstrecken identisch sein, so müsste die Startbedingung ein logisches ODER im Sinne von „wird in eine ABC-Bremsstrecke eingefahren ODER wird in eine DC-Bremsstrecke eingefahren“ enthalten. Dies kann mit den verfügbaren Sprachmitteln nicht direkt umgesetzt werden, jedoch kann diese Funktionalität erreicht werden, indem der gleiche Ablauf mit dem alternativen Starterereignis „DC-Bremsstrecke erkannt“ erneut erstellt wird.

Ein anderes Anwendungsszenario ist das Abrufen einer ganzen Kette von Aktionen mit nur einem Tastendruck. Dieses Szenario wird oft mit „Makro-Funktion“ bezeichnet. Durch das Aneinanderreihen der gewünschten Einzelschritte ergibt sich der entsprechende Ablauf, wobei die Schritte bei Bedarf durch eine Wartezeit voneinander abgetrennt werden können.

Werden in solchen Makros Befehle für Fahrtrichtung und Geschwindigkeit eingebaut, so lassen sich beispielsweise Pendelzugsteuerungen aufbauen: Die Enden der Pendelzugstrecke weisen einen Bremsabschnitt auf, in dem das Fahrzeug zum Halten kommt. Das Halten startet einen Ablauf, in dem zunächst etwas gewartet wird (im Beispiel rechts 20 Sekunden), bevor die Fahrtrichtung gewechselt wird und damit auch gleichzeitig die Stirnbeleuchtung wechselt. Nach 10 Sekunden Wartezeit wird der Abfahrtspiff (F4) ein- und direkt wieder ausgeschaltet. Nach erneuter Wartezeit wird die Fahrt mit Fahrstufe 8 fortgesetzt – aufgrund des Fahrtrichtungswechsels aber in die Gegenrichtung. Damit ist der Ablauf „immer“ – also ohne auf weitere Ereignisse zu warten – beendet.

Damit das Gleissignal die Fahrtrichtungsumkehr bei Beenden des Ablaufs nicht wieder rückgängig macht, wird die vom Ablauf erzeugte Fahrtrichtung über den gleichnamigen Befehl „eingefroren“.










Wird wie beim Ablauf „Bremsen glühen“ eine Variable eingeführt, die je nach Fahrtrichtung den Wert wechselt, so kann die Hinfahrt in anderer Fahrstufe erfolgen als die Rückfahrt. In diesem Fall ist aber neben der Fahrtrichtung auch die Fahrstufe einzufrieren.

Bei Abläufen dieser Art ist zu bedenken, dass der Decoder entweder der Intellimatic oder dem Gleissignal gehorcht. Zur Abarbeitung zählt auch das

Eine häufige Anwendung dürfte das Zusammenfassen mehrerer aufeinanderfolgender

Einzelfunktionen zu einem Ablauf („Makro“) sein, hier beispielhaft mit den Ausgängen 1 bis 4. Am realen Modell könnte ein Ablauf wie dieser das Aufrüsten einer Lokomotive wie einer


















V 200 darstellen: Der Lokführer öffnet und schließt die Tür, schaltet das Licht im Maschinenraum ein, nach einer kurzen Zeit auch im Führerstand. Etwas Zeit vergeht, dann wird erst Motor 1 und wiederum etwas später Motor 2 gestartet. Ausgelöst wird der gesamte Ablauf komfortabel mit nur einem Tastendruck, in diesem Fall F28.

Marke:	Befehl:	Parameter:
		wenn F28 ein
		Ausgang 1 = ein
		warte 1 Sek.
		Ausgang 2 = ein
		warte 1 Sek.
		Ausgang 3 = ein
		warte 1 Sek.
		Ausgang 4 = ein
		Ablauf Endel

Marke:	Befehl:	Parameter:
		wenn Lok. steht
		warte 20 Sek.
		Fahrtrichtung umkehren
		Einfrieren von Fahrtrichtung
		warte 10 Sek.
		Funktion 4 = ein
		Funktion 4 = aus
		warte 6 Sek.
		Fahrstufe = 8
		Macro Endel

Dieser Ablauf realisiert eine einfache Pendelzugsteuerung direkt im Fahrzeugdecoder: Das Fahrzeug wechselt nach jedem Halt die Fahrtrichtung und ein Abfahrtspiff ertönt, bevor die Fahrt fortgesetzt wird.

Nun erfolgt die Hinfahrt mit einer anderen Fahrstufe als die Rückfahrt. Je nach Wert der Variablen 2 erfolgt eine Verzweigung, die sowohl den Wert der Variablen wechselt als auch die Fahrstufe vorgibt und dann zum Einfrieren und zum Ende des Ablaufs springt bzw. direkt zu diesem Befehl kommt.

Marke:	Befehl:	Parameter:
		wenn Lok. steht
		warte 20 Sek.
		Fahrtrichtung umkehren
		warte 10 Sek.
		Funktion 4 = ein
		Funktion 4 = aus
		warte 6 Sek.
		wenn Variable 2 = 0
		springe nach Marke1
		Variable 2 = 1
		Fahrstufe = 11
		springe nach Marke2
		1
		Variable 2 = 0
		Fahrstufe = 8
		Einfrieren von Fahrstufe, Fahrtrichtung
		Macro Endel



Sollen in Abläufen vorgenommene Einstellungen auch nach Ende des Ablaufs erhalten bleiben und nicht durch die am Gleis anliegenden Informationen überschrieben werden, so müssen die erfolgten Einstellungen mit dem Befehl „Einfrieren“ festgehalten werden. Die Einstellungen werden erst dann wieder vom Gleissignal bestimmt, wenn sich dieses gegenüber dem Ursprungszustand bei Beendigung des Ablaufs ändert, in diesem Fall also die Fahrstufe geändert wird.

Marke:	Befehl:	Parameter:
START	wenn IST-Fahrstufe gleich 0	
Light	Ausgang 7 = ein	
End	Ablauf Endel	

Marke:	Befehl:	Parameter:
START	wenn IST-Fahrstufe ungleich 0	
Light	Ausgang 7 = aus	
End	Ablauf Endel	

Links der korrekte Ablauf für das Verhalten „Solange das Fahrzeug steht, ist die Führerstandsbeleuchtung eingeschaltet.“ Das aktive Ausschalten (rechts) der Führerstandsbeleuchtung ist unnötig, da der Ausgang ohne Einfrieren der Einstellung in den vorherigen Zustand kommt, sobald der Ablauf endet.

Warten auf das Ende-Ereignis. Währenddessen ist der Decoder weder per Regler, noch per Computer und Interface ansprechbar.

Anlagenwechsel

Die in der Intellimatic hinterlegten Abläufe sind mit dem Modell verbunden, wechselt also das Modell seinen Einsatzort von der Heim- auf die Clubanlage, sind die Abläufe auch in der neuen Umgebung aktiv. Dies kann im Fall eines „Makros“ erwünscht sein – bei einer Pendelzugsteuerung aber womöglich zu Verwirrung (oder mehr) führen.

Ein generelles Deaktivieren der Intellimatic ist durch Entfernen eines Häkchens in der DigiTest-Oberfläche leicht möglich, einzelne Abläufe können jedoch nicht deaktiviert werden, man muss sie aus dem Decoder löschen. Da das Intellimatic-Programmierfenster nicht nur die Übertragung der Abläufe

zum Decoder unterstützt, sondern auch das Speichern auf dem Windows-PC, lassen sich die Abläufe für ein Fahrzeug leicht in unterschiedlichen Versionen – beispielsweise Club und Zuhause – ablegen und bei Bedarf wieder in den Decoder übertragen.

Fazit

Die Intellimatic hat vor allem bei manuell betriebenen Digitalanlagen ohne Computersteuerung einen hohen Nutzen, indem sie den Modellbahner von Überwachungs- und Steuerungsaufgaben entlastet und den Anlagenbetrieb durch kleine automatisierte, aber trotzdem situationsbezogene Abläufe bereichert.

Auf computergesteuerten/-überwachten Anlagen kann die Intellimatic das zentrale Steuerungssystem unterstützen. Bei den Abläufen ist dann jedoch zu beachten, dass die Fahrzeuge unter Intellimatic-Kontrolle möglicher-

Marke:	Befehl:	Parameter:
START	wenn F28 aus	
Light	Ausgang 4 = aus	
Light	Ausgang 3 = aus	
Light	Ausgang 2 = aus	
Light	Ausgang 1 = aus	
End	Ablauf Endel	

Denkfehler 1: Der „ordentliche“ Programmierer will die zuvor aktivierten Ausgänge (siehe Makro vorhergehende Seite oben) beim Ausschalten von F28 auch wieder deaktivieren. Dies erfolgt jedoch automatisch durch die Zentrale, wenn nach dem Ausschalten von F28 das Startereignis für den Ablauf links nicht mehr erfüllt ist und das Gleissignal die Funktionen des Decoders bestimmt. Der hier gezeigte Ablauf ist somit unnötig.

Light	warte 1 Sek.	
Light	Ausgang 4 = ein	
Light	warte 1 Sek.	
Light	auf F28 aus	
Light	Ausgang 1 = aus	
Light	Ausgang 2 = aus	
Light	Ausgang 3 = aus	

Denkfehler 2: Die beiden Abläufe werden über eine Warte-Anweisung („Warte bis F28 = aus“) miteinander verbunden. In diesem Fall werden die Befehle ab Warte nie erreicht, da der Ablauf beendet wird, wenn F28 deaktiviert wird.

Bei beiden Denkfehlern erfolgt kein Fehlverhalten, es wird jedoch durch nicht benötigte Befehle beziehungsweise komplette Abläufe Programmspeicher verschwendet.

weise „unter dem Radar“ des zentralen Steuerungsprogramms agieren und so von der regulären Überwachung durch die Computersteuerung ausgenommen sind.

Dr. Bernd Schneider

Kurz + knapp

- Uhlenbrock 73235: Next18, 9,5 x 7,8 x 2,4 mm, DCC, Motorola, Selectrix, 0,8A Motor, UVP € 29,90
- Uhlenbrock 74560: PluX22, 22 x 15 x 3,8 mm, DCC, Motorola, Selectrix, 1,2A, UVP € 33,90
- Uhlenbrock 74570: wie 74560, zusätzlich Mfx, UVP € 37,90
- Uhlenbrock 71000 DigiTest: Decoder Test- und Programmiergerät mit Windows-Software, UVP € 159,00