

XML-Schnittstelle Digital-S-Inside 2



© 2015 Thomas Falkner, Schwabach
18.01.2016

Version 1.05

Allgemeine Einführung

Diese Dokumentation beschreibt die XML-Schnittstelle des Digital-S-Inside Services. Die Aufgabe des Services ist es, Modellbahnhardware, wie z.B. die DiCoStation oder ein HSI-88, über das Netz zur Verfügung zu stellen. Ein Service kann mehrere Modellbahngeräte verwalten. Im weiteren Sprachgebrauch wird jedes angeschlossenes System als Digitalsystem bezeichnet. Jedes Digitalsystem erhält eine eindeutige System-Id, über die es angesprochen werden kann. Wird keine System-Id angegeben, so wird das erste Digitalsystem angesprochen, zu dem der Befehl passt. Das erste Digitalsystem hat die System-Id 1.

Funktionsgruppen

Da nicht alle Digitalsysteme alle Digital-Funktionen bereitstellen, gibt es Funktionsgruppen. Der Service unterstützt folgende Funktionsgruppen:

control	Steuern von Lok-Decodern
switch	Schalten von Magnetartikeldecodern
feedback	Einlesen von Rückmeldekontakten
program	Programmierung von Lok-Decodern
power	Die Gleisspannung lässt sich ein- und ausschalten

Welche Funktionsgruppen das jeweilige Digitalsystem unterstützt, kann mit dem Befehl „Query System“ ermittelt werden.

Netzwerk

Die Schnittstelle kommuniziert über das TCP/IP-Protokoll. Standardmäßig hört der Service am Port 51400. Somit kann der Service z.B. mit einem Telnet-Client getestet werden. Der Service arbeitet asynchron. Das heißt, dass Ereignisse, z.B. die Änderung eines Rückmeldekontaktes, sofort und unaufgefordert übertragen werden. Wenn man daher einen Befehl absendet und dann auf dessen Antwort wartet, kann es passieren, dass dazwischen ein Ereignis gemeldet wird. Das aufrufende Client-Programm muss damit rechnen und zurechtkommen.

XML

Alle Befehle und Antworten sind in XML. Somit ist eine Unabhängigkeit vom Betriebssystem und von der Programmiersprache gewährleistet. Damit der Service das Ende eines Befehls erkennt, muss dieser mit einer LF oder CR/LF abgeschlossen werden. Damit ist es auch möglich, dass der Service mit einem Telnet-Client getestet werden kann. Dies bedeutet aber auch, dass der XML-Befehl kein CR und LF enthalten darf. Gewöhnlich lassen sich XML-Builder so einstellen, dass sie ein flaches, unformatiertes XML ausgeben. Andernfalls müssen die CR und LF nachträglich manuell ausgefiltert werden.

Protokolle

Lok-Protokolle

Bei der Funktionsgruppe Control können folgende Protokolle verwendet werden:

Bezeichnung	Format	Adressen	Fahrstufen	Funktionen	
M1	Motorola		80	14	F0
M2	Motorola		80	14	F0-F4*
M3	Motorola		255	28	F0-F4*
M4	Motorola		255	14	F0-F4*
M5	Motorola		80	27	F0-F4*
D0	DCC		127	14	F0-F28
D1	DCC		127	28	F0-F28
D2	DCC		127	126	F0-F28
D3	DCC		16127	28	F0-F28
D4	DCC		16127	126	F0-F28

X0 Xlight 16383 126 F0-F15

*F5-F8 über zweite Adresse

Die Anzahl der Fahrstufen ist ohne die Fahrstufe 0 angegeben.

Magnetartikeldecoder-Protokolle:

Bei der Funktionsgruppe Switch können folgende Protokolle verwendet werden:

Bezeichnung	Format	Magnetartikeladressen
M	Motorola	320
D	DCC	2048

Message

Meldung	Message
Syntax	<Message Id="mi" type="type">text</Message>
Variablen	mi = Id der Nachricht type = Art der Nachricht (0 = Meldung; 1 = Warnung; 2 = Fehler) text = Text der Nachricht
Beschreibung	Der DSI-Service kann diverse Statusmeldungen an den aufrufenden Prozess schicken. Die Meldungen haben alle eine eindeutige Id. Der Client könnte die Nachrichten z.B. in einem Statusfenster anzeigen und somit den Benutzer über den Servicestatus informieren.
Beispiel	</Message Id="0001" type="0">Der Dienst ist bereit.</Message>

Response

Antwort	Response
Syntax	<pre><Response Errors="e" System="id"> <Message Id="mi">text</Message> </Response></pre>
Variablen	<p>id = Id des Systems (0, wenn der Service betroffen ist)</p> <p>e = Wenn Fehler aufgetreten = 1; wenn kein Fehler = 0</p> <p>text = Fehlermeldung</p> <p>mi = Id der Fehlermeldung</p>
Beschreibung	<p>Jeder Befehl an den DSI-Service wird mit einer Antwort (Response) quittiert. War der Befehl erfolgreich, so ist das Errors-Attribut 0 und es werden in der Regel in dem Response-Tag die Daten des Befehls bestätigt. Im Fehlerfall ist das Errors-Attribut ungleich 0.</p>
Beispiel	<p>Einfache Antwort nach erfolgreich ausgeführten Befehl:</p> <pre><Response Errors="0" System="1"/></pre> <p>Antwort mit Zusatzinformationen nach erfolgreich ausgeführten Power-Befehl:</p> <pre><Response Errors="0" System="1"> <Power>1</Power> </Response></pre> <p>Antwort nach nicht erfolgreichen Befehl:</p> <pre><Response Errors="1" System="1"> <Message Id="1">Alles kaputt.</Message> </Response></pre>

Service / Digitalsystem

Befehl	Shutdown
Syntax	<Shutdown/>
Antwort	<Response Errors="0" System="0">
Variablen	-
Beschreibung	Der DSI-Service wird beendet. Dies ist für Remote-Clients nur erlaubt, wenn kein aufrufender Prozess den Service kontrolliert.
Beispiel	<Shutdown/>

Befehl	Restart
Syntax	<Restart/>
Antwort	<Response Errors="0" System="0">
Variablen	-
Beschreibung	Der DSI-Service wird neu gestartet. Dies ist für Remote-Clients nur erlaubt, wenn kein aufrufender Prozess den Service kontrolliert.
Beispiel	<Restart/>

Befehl	Restart System
Syntax	<Restart System="id"/>
Antwort	<Response Errors="0" System="id">
Variablen	id = Id des Systems
Beschreibung	Ein Digitalsystem wird neu gestartet.
Beispiel	<Restart System="1"/>

Befehl	Query System (alle)
Syntax	<Query><System/></Query>
Antwort	<pre> <Response Errors="0" System="0"> <SystemList><System Id="1" Type="1" Name="DiCoStation" Port="USB1" Active="true" DemoMode="false" SerialNumber="0"> <Version Text="USB 1.05 Release Windows 7 (64) / DiCoStation 1.09"/> <FunctionGroups> <FunctionGroup Name="power"/> <FunctionGroup Name="control" DefaultProtocol="M2"> <LocoProtocol MaxAddress="80" MaxSpeedStep="14" Functions="1" AbsoluteDirection="0">M1</LocoProtocol> <LocoProtocol MaxAddress="80" MaxSpeedStep="14" Functions="5" AbsoluteDirection="1">M2</LocoProtocol> <LocoProtocol MaxAddress="255" MaxSpeedStep="28" Functions="5" AbsoluteDirection="1">M3</LocoProtocol> <LocoProtocol MaxAddress="255" MaxSpeedStep="14" Functions="5" AbsoluteDirection="1">M4</LocoProtocol> <LocoProtocol MaxAddress="80" MaxSpeedStep="27" Functions="5" AbsoluteDirection="1">M5</LocoProtocol> <LocoProtocol MaxAddress="127" MaxSpeedStep="14" Functions="29" AbsoluteDirection="1">D0</LocoProtocol> <LocoProtocol MaxAddress="127" MaxSpeedStep="28" Functions="29" AbsoluteDirection="1">D1</LocoProtocol> <LocoProtocol MaxAddress="127" MaxSpeedStep="126" Functions="29" AbsoluteDirection="1">D2</LocoProtocol> <LocoProtocol MaxAddress="16127" MaxSpeedStep="28" Functions="29" AbsoluteDirection="1">D3</LocoProtocol> <LocoProtocol MaxAddress="16127" MaxSpeedStep="126" Functions="29" AbsoluteDirection="1">D4</LocoProtocol> </FunctionGroup> <FunctionGroup Name="switch" DefaultProtocol="M"> <DecoderProtocol MaxAddress="320">M</DecoderProtocol> <DecoderProtocol MaxAddress="2048">D</DecoderProtocol> </FunctionGroup> </FunctionGroups> </System><System Id="2" Type="2" Name="HSI-88" Port="USB1" Active="true" DemoMode="false"> <Version Text="V. 1.09 / HSI-88-USB / 1.05r Win 7 (64) / (c) 2011 LDT & Falkner"/> <FunctionGroups> <FunctionGroup Name="feedback"> <FeedbackBus MinAddress="1" MaxAddress="0" Bus="left"/> <FeedbackBus MinAddress="1" MaxAddress="0" Bus="middle"/> <FeedbackBus MinAddress="1" MaxAddress="16" Bus="right"/> </FunctionGroup> </FunctionGroups> </System> </SystemList></Response> </pre>

Variablen	-
Beschreibung	Es wird eine Liste aller angeschlossenen Digitalssysteme und deren Eigenschaften geliefert.
Beispiel	<Query><System/></Query>

Befehl	Query System (einzeln)
Syntax	<Query System="id"><System/></Query>
Antwort	<pre> <Response Errors="0" System="2"> <System Id="2" Type="2" Name="HSI-88" Port="USB1" Active="true" DemoMode="false"><Version Text="V. 1.09 / HSI-88-USB / 1.05r Win 7 (64) / (c) 2011 LDT & amp; Falkner"/> <FunctionGroups> <FunctionGroup Name="feedback"> <FeedbackBus MinAddress="1" MaxAddress="0">left</FeedbackBus> <FeedbackBus MinAddress="1" MaxAddress="0">middle</FeedbackBus> <FeedbackBus MinAddress="1" MaxAddress="16">right</FeedbackBus> </FunctionGroup> </FunctionGroups></System></Response> </pre>
Variablen	id = Id des Systems
Beschreibung	Es werden Informationen zum ausgewählten Digitalssystem geliefert.
Beispiel	<Query System="2"><System/></Query>

Befehl	Query Version (Service)
Syntax	<Query><Version/></Query>
Antwort	<Response Errors="0" System="0"><Version Text="Digital-S-Inside">2.00</Version></Response>
Variablen	-
Beschreibung	Es wird die Version des DSI-Service zurückgegeben.
Beispiel	<Query><Version/></Query>

Befehl	Query Version (System)
Syntax	<Query System="id"><Version/></Query>
Antwort	<pre><Response Errors="0" System="1"> <Version Text="USB 1.02 Release Windows XP / DiCoStation 1.05"/> </Response></pre>
Variablen	id = Id des Systems
Beschreibung	Es wird die Version des ausgewählten Digitalsystems geliefert.
Beispiel	<Query System="1"><Version/></Query>

Befehl	Query LocoProtocol
Syntax	<pre><Query System="id"> <LocoProtocol>lp</LocoProtocol> </Query></pre>
Antwort	<pre><Response Errors="0" System="1"> <LocoProtocol MaxAddress="80" MaxSpeedStep="14" Functions="1" AbsoluteDirection="0">M2</LocoProtocol> </Response></pre>
Variablen	<p>id = Id des Systems</p> <p>lp = Lok-Protokoll: M1, M2, M3, M4, M5, D0, D1, D2, D3, D4, X0</p>
Beschreibung	<p>Die unterstützten Grenzwerte eines Lok-Protokolls können sich je nach Zentrale unterscheiden. Daher können sie je System abgerufen werden.</p> <p>Es werden folgende Informationen über das angegebene Lok-Protokoll ausgegeben:</p> <ul style="list-style-type: none"> – Die maximal zulässige Adresse – Die maximale Fahrstufe – Die maximale Anzahl der Schaltfunktionen – Absolute Richtungsangabe

Beispiel	<pre><Query System="1"> <LocoProtocol>M2</LocoProtocol> </Query></pre>
----------	--

Befehl	Query DecoderProtocol
Syntax	<pre><Query System="id"> <DecoderProtocol>dp</DecoderProtocol> </Query></pre>
Antwort	<pre><Response Errors="0" System="1"> <DecoderProtocol MaxAddress="320">M</DecoderProtocol> </Response></pre>
Variablen	<p>id = Id des Systems</p> <p>dp = Magnetartikeldecoder-Protokoll: M, D</p>
Beschreibung	<p>Die unterstützten Grenzwerte eines Magnetartikeldecoder-Protokolls können sich je nach Zentrale unterscheiden. Daher können sie je System abgerufen werden.</p> <p>Es werden folgende Informationen über das angegebene Magnetartikeldecoder-Protokoll ausgegeben:</p> <p>- Maximale Magnetartikeladresse</p>
Beispiel	<pre><Query System="1"> <DecoderProtocol>M</DecoderProtocol> </Query></pre>

Power

Befehl	Set Power
Syntax	<pre><Set System="id"> <Power Booster="bn">pwr</Power> </Set></pre>
Antwort	<pre><Response Errors="0" System="1"> <Power>1</Power> </Response></pre>
Variablen	<p>id = Id des Systems</p> <p>bn = Nummer des Boosters (optional)</p> <p>pwr = 1 für Gleisspannung an; 0 für Gleisspannung aus</p>
Beschreibung	Der Befehl schaltet die Gleisspannung ein oder aus. Die Angabe einer Boosternummer ist für künftige Erweiterungen vorgesehen.
Beispiel	<pre><Set System="1"> <Power>1</Power> </Set></pre>

Befehl	Query Power
Syntax	<pre><Query System="id"> <Power Booster="bn"/> </Query></pre>
Antwort	<pre><Response Errors="0" System="id"> <Power>pwr</Power> </Response></pre>
Variablen	id = Id des Systems

	bn = Nummer des Boosters (optional) pwr = 1 für Gleisspannung an; 0 für Gleisspannung aus
Beschreibung	Der Befehl ermittelt, ob die Gleisspannung ein oder aus ist. Die Angabe einer Boosternummer ist für künftige Erweiterungen vorgesehen.
Beispiel	<Query System="1"> <Power/> </Query>

Loco

Befehl	Set Loco
Syntax	<Set System="id"> <Loco Address="adr" Protocol="lp"> <Speed Direction="dir" Base="bs">v</Speed> <EmergencyStop>es</EmergencyStop> <ToggleImpulse>ti</ToggleImpulse> <Function Id="fn0">f0</Function> ... <Function Id="fnx">fx</Function> </Loco> </Set >
Antwort	<Response Errors="0" System="id"> <Loco Address="adr" Protocol="lp"> <Speed Direction="dir" Base="bs">v</Speed> <EmergencyStop>es</EmergencyStop>

	<pre> <ToggleImpulse>ti</ToggleImpulse> <Function Id="fn0">f0</Function> ... <Function Id="fnx">fx</Function> </Loco> </Response> </pre>
Variablen	<p>id = Id des Systems</p> <p>lp = Lok-Protokoll</p> <p>dir = Fahrtrichtung (forwards = vorwärts; rearwards = rückwärts)</p> <p>bs = Basis auf die sich die Geschwindigkeit bezieht (optional)</p> <p>v = Geschwindigkeit</p> <p>es = Nothalt; 1 = ein; 0 = aus</p> <p>ti = Fahrtrichtungsumschaltimpuls; 1 = ein / 0 = aus</p> <p>fnx = Nummer der Schalfunktion</p> <p>fx = Wert der Schalfunktion; 1 = ein / 0 = aus</p>
Beschreibung	<p>Der Befehl setzt die Geschwindigkeit und / oder Schalfunktionen eines Lok-Decoders. Werte die nicht angegeben werden, bleiben unverändert. Für Protokolle mit absoluter Fahrtrichtungsangabe kann das Attribute "Direction" angegeben werden. Für andere kann ein Fahrtrichtungsumschaltimpuls mit dem Tag "ToggleImpulse" ein- bzw. ausgeschaltet werden. Die Geschwindigkeit wird in Fahrstufen entsprechend des gewählten Protokolls erwartet. Alternativ kann auch optional eine andere Geschwindigkeitsbasis angegeben werden. Im Beispiel fährt die Lok mit 33% der angegebenen Basis von 100%.</p>
Beispiel	<pre> <Set System="1"> <Loco Address="45" Protocol="M2"> <Speed Direction="forwards" Base="100">33</Speed> <Function Id="0">0</Function> <Function Id="3">1</Function> </pre>

	<pre> </Loco> </Set > </pre>
--	--

Befehl	Query Loco
Syntax	<pre> <Query System="id"> <Loco Address="adr" Protocol="lp"/> </Query> </pre>
Antwort	<pre> <Response Errors="0" System="id"> <Loco Address="adr" Protocol="lp"> <Speed Direction="dir" Base="bs">v</Speed> <EmergencyStop>es</EmergencyStop> <ToggleImpulse>ti</ToggleImpulse> <Function Id="fn0">f0</Function> ... <Function Id="fnx">fx</Function> </Loco> </Response> </pre>
Variablen	<p>id = Id des Systems</p> <p>lp = Lok-Protokoll</p> <p>dir = Fahrtrichtung (forwards = vorwärts; rearwards = rückwärts)</p> <p>bs = Basis auf die sich die Geschwindigkeit bezieht (optional)</p> <p>v = Geschwindigkeit</p> <p>es = Nothalt; 1 = ein; 0 = aus</p> <p>ti = Fahrtrichtungsumschaltimpuls; 1 = ein / 0 = aus</p> <p>fnx = Nummer der Schaltfunktion</p>

	fx = Wert der Schaltfunktion; 1 = ein / 0 = aus
Beschreibung	Der Befehl fragt die Geschwindigkeit und die Schaltfunktionen eines Lok-Decoders ab.
Beispiel	<pre><Query System="1"> <Loco Address="45" Protocol="M2"/> </Query></pre>

Decoder

Befehl	Set Decoder
Syntax	<pre><Set System="id"> <Decoder Address="adr" Protocol="dp" Port="po" Timeout="t">v</Decoder> </Set ></pre>
Antwort	<pre><Response Errors="0" System="id"> <Decoder Address="adr" Protocol="dp" Port="po" Timeout="t">v</Decoder> </Response></pre>
Variablen	<p>id = Id des Systems</p> <p>adr = Adresse des Magnetartikels</p> <p>dp = Decoder-Protokoll: M, D</p> <p>po = Port der angesprochen werden soll</p> <p>t = Schaltdauer im Millisekunden</p> <p>v = Zustand des Ports. 1 = ein / 0 = aus</p>
Beschreibung	Der Befehl dient zum Schalten eines Magnetartikeldecoders. Die Adresse ist die fortlaufende Adresse des Magnetartikels. Bei Decodern die mehrere Magnetartikel ansprechen, muss entsprechend umgerechnet werden. Als Port ist in der Regel nur der Anschluss 0 für gebogen/rot und 1 für gerade/grün sinnvoll. Wird der

	Ausgang auf 1 gesetzt, kann optional eine Schaltdauer angegeben werden, nach der, der Ausgang automatisch wieder ausgeschaltet wird. Wird keine Schaltdauer angegeben, muss der Client den Ausgang manuell ausschalten.
Beispiel	<pre><Set System="1"> <Decoder Address="37" Protocol="M" Port="0" Timeout="1000">1</Decoder> </Set ></pre>

Befehl	Query Decoder
Syntax	<pre><Query System="id"> <Decoder Address="adr" Protocol="dp"/> </Query></pre>
Antwort	<pre><Response Errors="0" System="id"> <Decoder Address="adr" Protocol="dp" Port="po">v</Decoder> </Response></pre>
Variablen	<p>id = Id des Systems</p> <p>adr = Adresse des Magnetartikels</p> <p>dp = Decoder-Protokoll: M, D</p> <p>po = Port der zuletzt gesetzt wurde</p> <p>v = Zustand des Ports. 1 = ein / 0 = aus</p>
Beschreibung	Der Befehl fragt den Zustand eines Magnetartikels ab. Dabei gibt das Attribut "Port" an, welcher Port als letztes eingeschaltet wurde. Somit weis man, ob z.B. eine Weiche auf gerade oder gebogen steht. Der Wert v gibt an, ob der Ausgang im Moment noch an- oder ausgeschaltet ist.
Beispiel	<pre><Query System="1"> <Decoder Address="37" Protocol="M"/></pre>

	</Query>
--	----------

Feedback

Der Feedback-Befehl unterstützt mehrere Rückmeldebusse je Digitalsystem. In der Antwort des Befehls "Query System" wird die Information über vorhandene Rückmeldebusse geliefert. Bei einem HSI-88 USB sind das z.B. "left", "middle" und "right". Bei einer anderen Zentrale könnte das z.B. "s88" und "loconet" sein.

Befehl	Query FeedbackBus
Syntax	<pre><Query System="id"> <FeedbackBus Bus="bus"/> </Query></pre>
Antwort	<pre><Response Errors="0" System="id"> <FeedbackBus Bus="bus" MinAddress="min" MaxAddress="max"> <Feedback Address="1">1</Feedback> <Feedback Address="2">0</Feedback> ... </FeedbackBus> </Response></pre>
Variablen	<p>id = Id des Systems</p> <p>bus = Bezeichnung des Busses z.B. "right"</p> <p>min = kleinste gültige Adresse</p> <p>max = größte gültige Adresse</p>
Beschreibung	Der Befehl fragt den gültigen Adressbereich eines Rückmeldebusses ab und gibt den Status aller Rückmeldekontakte des Busses zurück.
Beispiel	<Query System="2">

	<pre><FeedbackBus Bus="right"/> </Query></pre>
--	--

Befehl	Query Feedback
Syntax	<pre><Query System="id"> <Feedback Address="adr" Bus="bus"/> </Query></pre>
Antwort	<pre><Response Errors="0" System="id"> <Feedback Address="adr" Bus="bus">value</Feedback> </Response></pre>
Variablen	<p>id = Id des Systems</p> <p>adr = fortl. Adresse des Kontaktes</p> <p>bus = Busbezeichnung</p> <p>value = Status des Rückmeldekontaktes</p>
Beschreibung	Der Befehl fragt den Zustand eines Rückmeldekontaktes ab.
Beispiel	<pre><Query System="2"> <Feedback Address="13" Bus="right"/> </Query></pre>

Events

Event	Event Feedback
Syntax	<pre><Event System="id"> <Feedback Address="adr" Bus="bus">v</Feedback></pre>

	</Event>
Variablen	id = Id des Systems adr = fortl. Adresse des Kontaktes bus = Busbezeichnung v = Zustand des Rückmeldekontakts
Beschreibung	Die Änderung eines Rückmeldekontaktes wird gemeldet.
Beispiel	<Query System="2"> <Feedback Address="13" Bus="right">1</Feedback> </Query>

Event	Event ShortCircuit
Syntax	<Event System="id"> <ShortCircuit Booster="bn">v</ShortCircuit> </Event>
Variablen	id = Id des Systems bn = Nummer des Boosters (optional) v = Kurzschlußzustand 1 = ein / 0 = aus
Beschreibung	Ein Kurzschluß oder dessen Beseitigung wird gemeldet.
Beispiel	<Event System="1"> <ShortCircuit>1</ShortCircuit> </Event>

Event	Event Loco
Syntax	<Event System="id"> <Loco Address="adr" Protocol="lp"> <Speed Direction="dir" Base="base">v</Speed>

	<pre> <EmergencyStop>es</EmergencyStop> <ToggleImpulse>ti</ToggleImpulse> <Function Id = "fn0">f0</Function> ... <Function Id = "fnx">fx</Function> </Loco> </Event> </pre>
Variablen	<p>id = Id des Systems</p> <p>lp = Lok-Protokoll</p> <p>dir = Fahrtrichtung (forwards = vorwärts; rearwards = rückwärts)</p> <p>bs = Basis auf die sich die Geschwindigkeit bezieht (optional)</p> <p>v = Geschwindigkeit</p> <p>es = Nothalt; 1 = ein; 0 = aus</p> <p>ti = Fahrtrichtungsumschaltimpuls; 1 = ein / 0 = aus</p> <p>fnx = Nummer der Schaltfunktion</p> <p>fx = Wert der Schaltfunktion; 1 = ein / 0 = aus</p>
Beschreibung	<p>Änderungen der Geschwindigkeit oder der Schaltfunktionen eines Lok-Decoders werden gemeldet. Nur die sich geänderten Werte werden übermittelt.</p>
Beispiel	<pre> <Event System="1"> <Loco Address="45" Protocol="M2"> <Speed Direction="forwards" Base="14">9</Speed> <EmergencyStop>0</EmergencyStop> <ToggleImpulse>0</ToggleImpulse> <Function Id = "0">0</Function> <Function Id = "3">1</Function> </pre>

	<pre> </Loco> </Event> </pre>
--	---

Event	Event Decoder
Syntax	<pre> <Event System="id"> <Decoder Address="adr" Protocol="dp" Port="po">v</Decoder> </Event> </pre>
Variablen	<p>id = Id des Systems</p> <p>adr = Adresse des Magnetartikels</p> <p>dp = Decoder-Protokoll: M, D</p> <p>po = Port der zuletzt gesetzt wurde</p> <p>v = Zustand des Ports. 1 = ein / 0 = aus</p>
Beschreibung	Eine Änderung an einem Magnetartikeldecoder wird gemeldet.
Beispiel	<pre> <Event System="1"> <Decoder Address="37" Protocol="M" Port="0">1</Decoder> </Event> </pre>

Event	Event Power
Syntax	<pre> <Event System="id"> <Power Booster="bn">pwr</Power> </Event> </pre>
Variablen	<p>id = Id des Systems</p> <p>bn = Nummer des Boosters (optional)</p> <p>pwr = 1 für Gleisspannung an; 0 für Gleisspannung aus</p>

Beschreibung	Es wird gemeldet, dass die Gleisspannung ein- oder ausgeschaltet wurde.
Beispiel	<pre><Event System="1"> <Power>1</Power> </Event></pre>

Decoder Programmierung

Es wird die Programmierung mit folgenden Protokollen unterstützt:

MM (Märklin Motorola)

DCC PoM (Programming on the Main)

Befehl	Set ProgramMode (aktivieren)
Syntax	<pre><Set System="id"> <ProgramMode Address="adr" Protocol="p" Track="t">1</ProgramMode> </Set></pre>
Antwort	<pre><Response Errors="0" System="id"> <ProgramMode Address="adr" Protocol="p" Track="t">1</ProgramMode> </Response></pre>
Variablen	<p>id = Id des Systems</p> <p>adr = Adresse des Lok-Decoders</p> <p>p = Decoder-Protokoll: M, D</p> <p>t = Gleisanschluss (0 = Hauptgleis; 1 = erstes Programmiergleis)</p>
Beschreibung	Der Befehl aktiviert den Programmiermodus. Mit der DiCoStation wird nur das Hauptgleis unterstützt.
Beispiel	<pre><Set System="1"> <ProgramMode Address="37" Protocol="D" Track="0">1</ProgramMode> </Set></pre>

Befehl	Set ProgramMode (deaktivieren)
Syntax	<pre><Set System="id"> <ProgramMode>0</ProgramMode> </Set></pre>
Antwort	<pre><Response Errors="0" System="id"> <ProgramMode>0</ProgramMode> </Response></pre>
Variablen	id = Id des Systems
Beschreibung	Der Befehl deaktiviert den Programmiermodus.
Beispiel	<pre><Set System="1"> <ProgramMode>0</ProgramMode> </Set></pre>

Befehl	Set CvRegister
Syntax	<pre><Set System="id"> <CvRegister RegisterAddress="cv">v</CvRegister> </Set></pre>
Antwort	<pre><Response Errors="0" System="id"> <CvRegister RegisterAddress="cv" ProgramAddress="a">v</CvRegister> </Response></pre>
Variablen	id = Id des Systems cv = Nummer des Cv-Registers v = Wert der gesetzt werden soll a = Adresse des Lok-Decoders
Beschreibung	Der Befehl schreibt einen Wert in ein CV-Register. Dazu muss der

	Programmiermodus aktiviert sein. Wird durch das Ändern des CV-Registers 1, die Adresse des Lok-Decoders geändert, wird anschliessend unter der neuen Adresse weiterprogrammiert.
Beispiel	<pre><Set System="1"> <CvRegister RegisterAddress="1">47</CvRegister> </Set></pre>