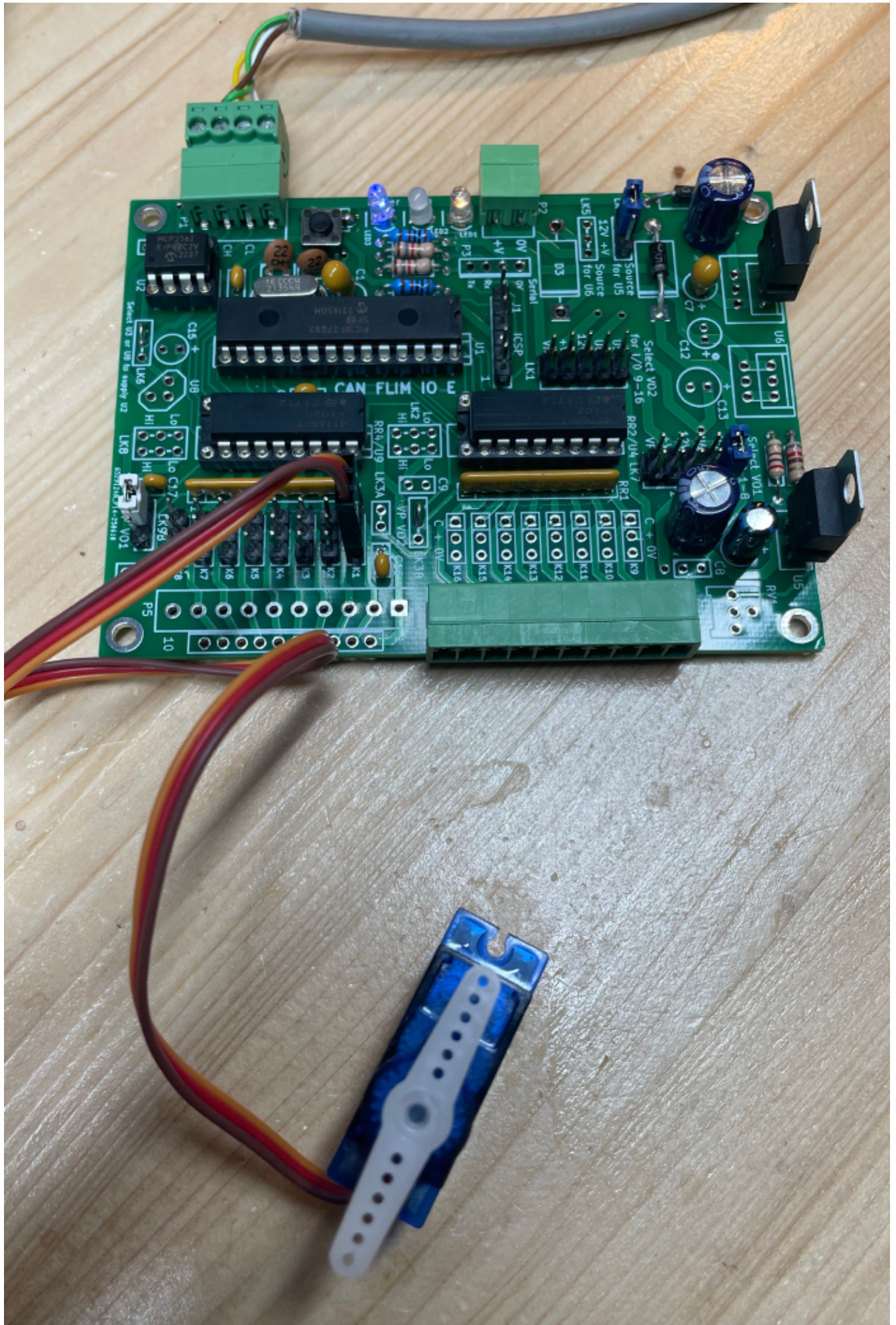


CANMIO-SERVO

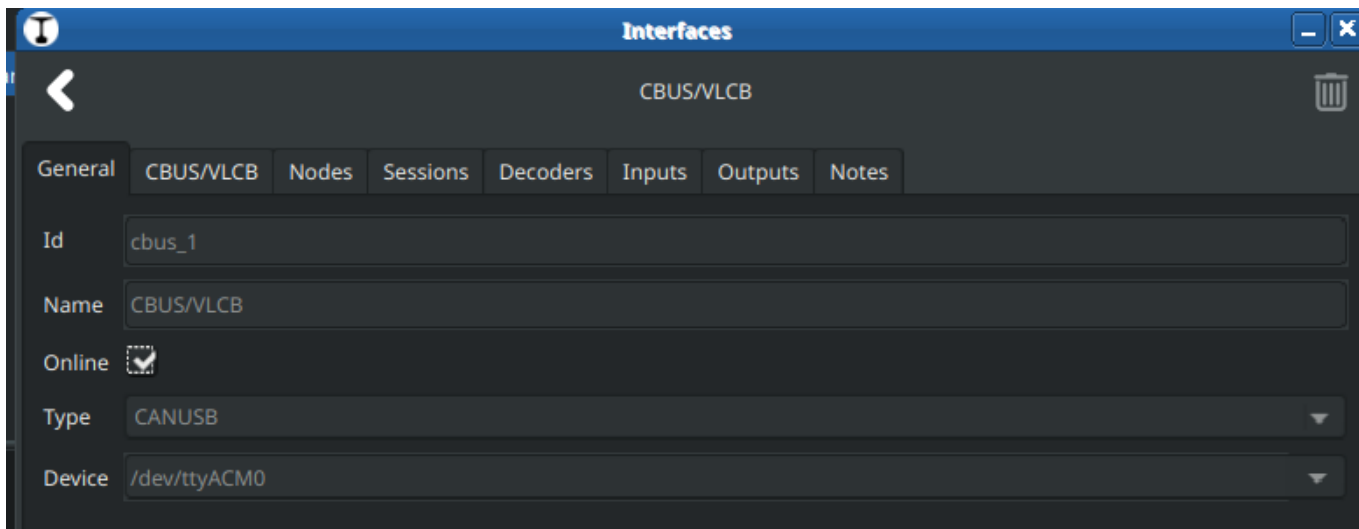
Bei dem Node CANMIO-SERVO handelt es sich um einen Ein- und Ausgabe Baustein für CBUS/VLCB, der 8 Eingangskanäle und 8 Kanäle für die Steuerung von Servos hat. Die Firmware basiert auf dem [Univeral-VLCB](#) Code von Ian Hogg.



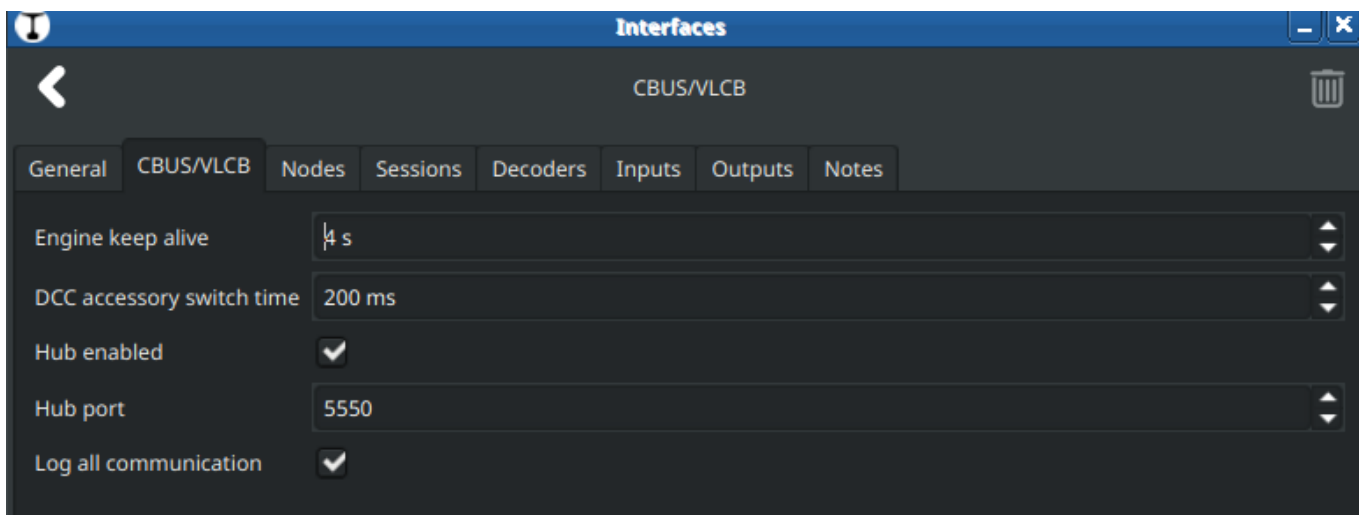
Für die ersten Versuche hatte ich mir einen Satz leere Leiterplatten für CANVOUT bei der [MERC](#) bestellt. Die Leiterplatten können je nach Anwendungsfall unterschiedlich bestückt werden. Dies hier ist die Bestückung für ein Servomodul zur Ansteuerung von 8 Servos und mit 8 Eingängen für Rückmeldungen. Mit einer etwas anderen Bestückung ergibt sich ein [I/O Modul](#) mit 8 Eingängen und 8 Ausgängen.

Wie alle Module kann auch das Servo Modul mit der MMC Software (alternativ auch mit FCU, aber nur unter Windows) konfiguriert werden. Auch JMRI unterstützt hier einiges. Ich arbeite mit Traintastic, die Software stellt einen Hub zur Verfügung, so dass mit nur einem [CANUSB4-ISO](#) diverse Software arbeiten kann, die eine Netzwerkverbindung unterstützt. Die genannten Programme können das alle.

So ist Traintastic für ein CANUSB Interface eingestellt:



Damit andere Programme auch mit dem CAN Bus kommunizieren können, wird hier der Hub konfiguriert, der über Port 5550 auf localhost (127.0.0.1) zu erreichen ist.



In der MMC Software wird diese Verbindung als Netzwerk Schnittstelle eingetragen:

Edit connection details CLOSE

Mode

Network

Auto will connect to the first CANUSB or CANUSB4 it finds

SerialPort will connect to the serial port entered below

Network will use the address & port entered below

serialPort

Forces MMC to use this specific serial port

Windows example: COM3

Linux example: ttyUSB2

Leave blank if 'SerialPort' not selected

Host

127.0.0.1

Name or IP address of machine to use for CAN connection

Leave blank if 'Network' not selected

Host Port

5550

Port number of machine for CAN connection

5550 is the default assigned port number

unused if 'Network' not selected

VLCB 'FCU Compatibility mode'

In VLCB nodes, disables some enhancements for FCU compatibility

Leave unchecked unless FCU also being used at the same time as MMC

save

Wenn MMC gestartet wird, sind so alle konfigurierten Nodes sichtbar:

| Node | CAN ID | Name | Group | Module | Version | Mode | Status | Stored Events | Space | Backup | Actions |
|-------|--------|-----------------|-------|---------|---------|------|--------|---------------|-------|--------|---|
| 256 | 2 | CANMIO | IO | CANMIO | 4d7 | FLJM | OK | 4 | 251 | OK | Events Name Parameters Variables VLCB Advanced Delete |
| 258 | 1 | CANMIO Servo | SERVO | CANMIO | 4d7 | FLJM | OK | 21 | 234 | OK | Events Name Parameters Variables VLCB Advanced Delete |
| 65534 | 127 | CANCMDB (65534) | | CANCMDB | 4f B12 | FLJM | OK | | | OK | Events Name Parameters Variables CBUS Advanced Delete |
| 65535 | 0 | CANCAB (65535) | | CANCAB | 4h | SLIM | OK | | | OK | Events Name Parameters Variables CBUS Advanced Delete |

Die Konfiguration der Nodes kann über MMC gespeichert werden. Mit "Name" wird ein Name für den Node festgelegt, mit "Parameters" können die Parameter des Nodes angezeigt werden und mit "Variables" können die Funktionen definiert werden. Bei der Konfiguration beginnt man in der Regel mit Variables:

[GLOBALS](#)
[CHANNEL 1](#)
[CHANNEL 2](#)
[CHANNEL 3](#)
[CHANNEL 4](#)
[CHANNEL 5](#)
[CHANNEL 6](#)
[CHANNEL 7](#)
[CHANNEL 8](#)
[CHANNEL 9](#)
[CHANNEL 10](#)
[CHANNEL 11](#)
[CHANNEL 12](#)
[CHANNEL 13](#)
[CHANNEL 14](#)
[CH](#)

Produced startup event Delay ¹

0.1 second steps, starting at 2

2.0 seconds

Heartbeat Delay ²

not yet implemented

range 0 to 255 milliseconds

0

Time delay between response messages ⁵

1 millisecond steps

10 milliseconds

Multi Servo speed ³

If >234 moves this amount every 100ms. If <= 234 number of 20ms periods per step

range 0 to 255 milliseconds

234

Pullups for channels 1-8 ⁶

- Channel 1
- Channel 2
- Channel 3
- Channel 4
- Channel 5
- Channel 6
- Channel 7
- Channel 8

Pullups for channels 9-16 ⁷

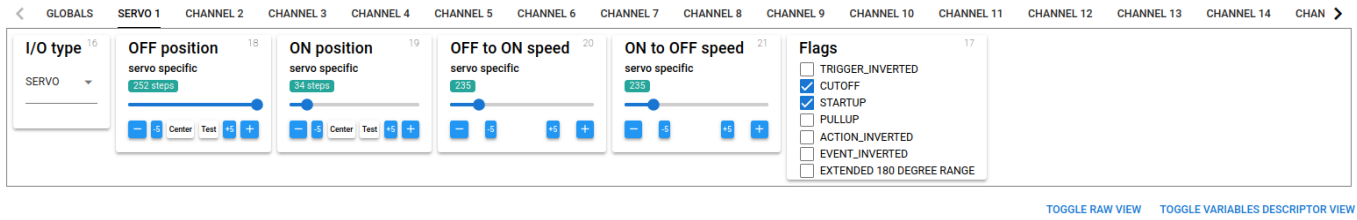
- Channel 9
- Channel 10
- Channel 11
- Channel 12
- Channel 13
- Channel 14
- Channel 15
- Channel 16

Module flags ¹⁰

Prevent default event creation

[TOGGLE RAW VIEW](#)
[TOGGLE VARIABLES DESCRIPTOR VIEW](#)

Es gibt immer globale Variablen und für jeden Kanal des Nodes spezifische Variablen für den Kanal.



Beim Servo Node können diverse Einstellungen vorgenommen werden. Wichtig ist, das der I/O Type auf SERVO oder auf BOUNCE steht. SERVO wird eher für Weichen verwendet, BOUNCE für Flügelsignale oder Schrankenanlagen, wo ein leichtes Einwippen erfolgen soll. Für einen Servo können die Start- und Endposition eingestellt werden, so wie zwei Geschwindigkeiten, von Start nach Ende und Ende nach Start. Es gibt noch ein paar Triggerwerte, für unterschiedliche Funktionen.

Mit der Universal-VLCB Firmware für den Node sind einige Events vorhanden, aber Events können auch frei programmiert werden. Events sind Ereignisse, wenn diese Eintreten wird eine Nachricht auf dem Bus erzeugt. Es wird dabei unterschieden in erzeugte (produced) und konsumierte (consumed) Ereignisse. Das kann z.B. beim Servo Node so aussehen:

| Identifier | Name | Group | Event node | Event number ↓ | Index | Type | Source | Actions |
|------------|---------------------------|-------|------------|----------------|-------|------|--------------|--|
| 0102C353 | Servo 1 Straight | SERVO | 258 | 50003 | 18 | long | stored event | Name Variables Teach send ON send OFF Delete |
| 0102C352 | Servo 1 Mid Position | SERVO | 258 | 50002 | 17 | long | stored event | Name Variables Teach send ON send OFF Delete |
| 0102C351 | Servo 1 Turnout | SERVO | 258 | 50001 | 1 | long | stored event | Name Variables Teach send ON send OFF Delete |
| 01022712 | Button 5 Servo 1 Turnout | SERVO | 258 | 10002 | 21 | long | stored event | Name Variables Teach send ON send OFF Delete |
| 01022711 | Button 4 Servo 1 Straight | SERVO | 258 | 10001 | 20 | long | stored event | Name Variables Teach send ON send OFF Delete |

Servo 1 soll eine einfache Weiche steuern. Eine Weiche hat eine Stellung für geradeaus (Straight) und abzweigend (Turnout). Wenn ein Servo sich bewegt, können Ereignisse als "Producer" erzeugt werden. Bei dem hier konfigurierten Servo 1 ist das Event 50003 (Straight), 50002 (Mittelposition) und 50001 (Turnout). Diese 3 Events sind "Producer" Events, die anderen beiden Events 10001 und 10002 sind Consumer Events. Die Consumer Events können von anderen Nodes oder von Traintastic (und zum Test auch von MMC, FCU) erzeugt werden. Zum Test werden die Events 10002 und 10001 von dem CANMIO-IO Node erzeugt, mit den Tastern 4 und 5 (besser den Eingängen 4 und 5 an denen Taster angeschlossen sind).

Wird der Taster 4 am CANMIO-IO Node betätigt, wird ein Event 10001 an Node 258 (CANMIO-SERVO) gesandt. Entsprechend stellt der CANMIO-SERVO Node das Servo 1 auf geradeaus. Mit dem Taster 5 wird entsprechend auf den Abzweig gestellt, in dem das Event 10002 an Node 258 geschickt wird.

Auf dem Bus sieht das am Beispiel des Events 10001 so aus:

```

2026-04-19 21:18:38.105947 cbus_1 D2002: RX: ACON node=258 event=10002 [90
01 02 27 12]
2026-04-19 21:18:38.203049 cbus_1 D2002: RX: ACOF node=258 event=50003 [91
01 02 C3 53]
2026-04-19 21:18:40.452841 cbus_1 D2002: RX: ACON node=258 event=50002 [90
01 02 C3 52]
2026-04-19 21:18:42.723794 cbus_1 D2002: RX: ACON node=258 event=50001 [90
01 02 C3 51]

```

Beim Betätigen des Tasters am Eingang 4 vom CANMIO-IO Modul wird das Ereignis 10002 an Node 258 gesendet, wobei ACON bedeutet, das der Taster eingeschaltet ist. Der Servo 1 steht in der

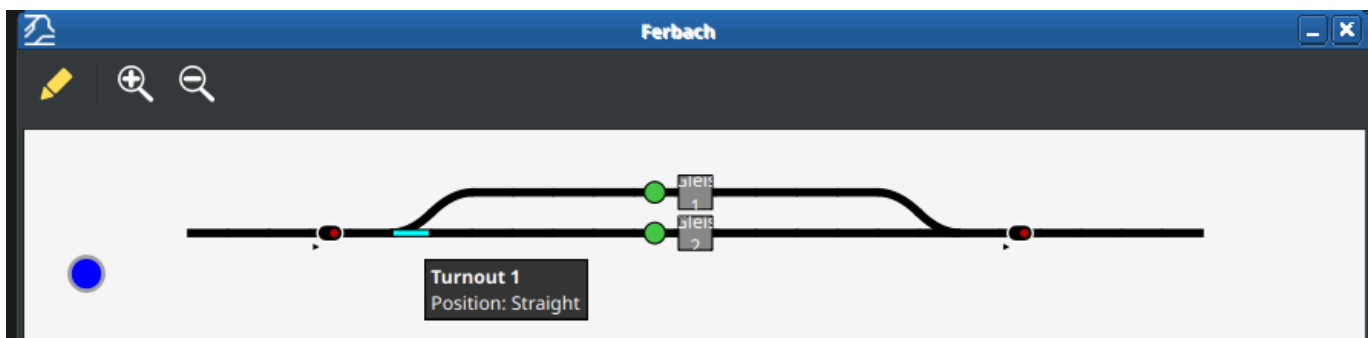
Position 'Straight', daher erfolgt als erstes die Meldung mit dem Event 50003 und zwar ACOF (Accessory Off), weil der Servo die 'Straight' Position verlässt. Ist das Servo in der Mittelstellung wird Event 50002 gesendet. Wenn die gewünschte Endstellung erreicht ist, wird Event 50001 gesendet, mit ACON (Accessory On), weil das Servo in der gewünschten Stellung angekommen ist.

Entsprechend sieht das beim Betätigen von Taster 5 genau andersherum aus:

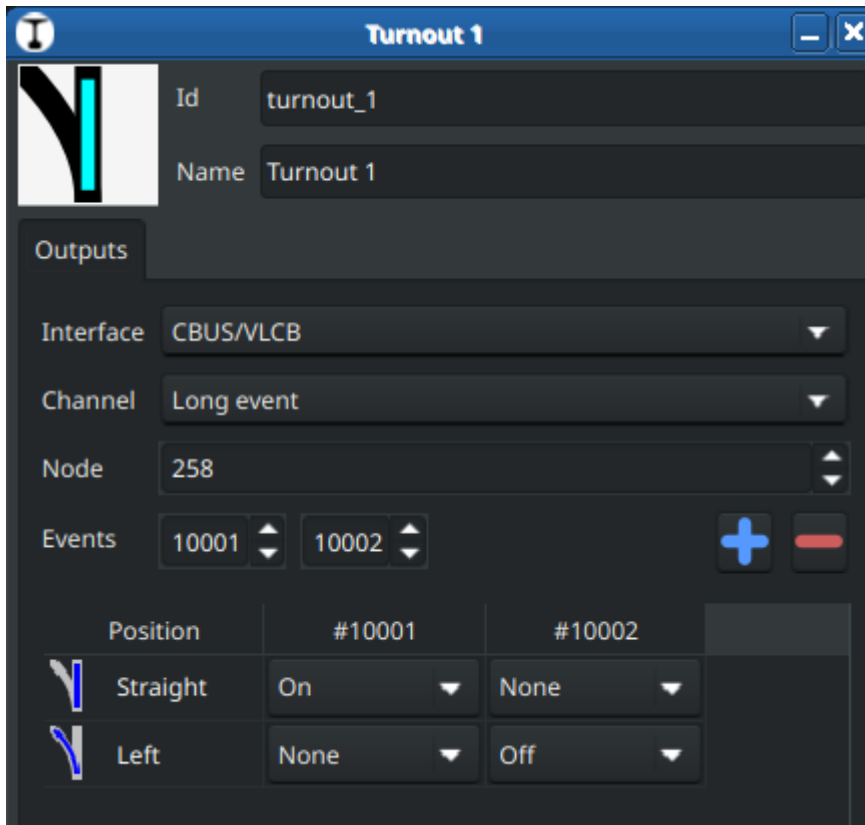
```
2026-04-19 21:25:07.964646 cbus_1 D2002: RX: ACON node=258 event=10001 [90 01 02 27 11]
2026-04-19 21:25:08.043531 cbus_1 D2002: RX: ACOF node=258 event=50001 [91 01 02 C3 51]
2026-04-19 21:25:10.293052 cbus_1 D2002: RX: ACOF node=258 event=50002 [91 01 02 C3 52]
2026-04-19 21:25:12.564153 cbus_1 D2002: RX: ACON node=258 event=50003 [90 01 02 C3 53]
```

Event 50001 mit ACOF, weil das Servo 1 die Stellung 'Turnout' verlässt, dann wieder die Mittelstellung mit Event 50002 und die Endstellung mit Event 50003 (Straight). Die Mittelstellung wird bei den Umläufen mit ACON und ACOF gemeldet. ACOF wird beim Event 50002 gemeldet, wenn der Servo von der OFF Position zur ON Position läuft und ACON entsprechend, wenn der Servo von der ON Position zur OFF Position läuft. So kann auch über den Events der Mittelstellung die Laufrichtung des Servos verfolgt werden.

Aber nicht nur die Steuerung über Taster ist möglich, auch ein Mausklick im Stellwerk von Traintastic kann diesen Servo steuern:



Dazu sind für die linke Weiche die Events 10001 und 10002, also die gleichen Events wie bei den Tastern (Eingängen) des CANMIO-IO Moduls eingestellt.



Bei einem Klick auf die Weiche im Stellwerk sendet Traintastic über das CANUSB4-ISO Modul die Events an den Bus, wobei hier direkt Node 258 angesprochen wird. Das muss nicht so sein, denn CBUS/VLCB unterscheidet Long- und Short Events. Long Events haben immer auch eine Nodenummer, es soll also ein spezifischer Node angesprochen werden. Es können 65535 Long Events definiert werden. Bei einem Short Event ist die Nodenummer 0, das ist quasi wie ein Broadcast, es wird ein Event an alle Nodes geschickt und die Nodes, die eine Aktion für diesen Short Event konfiguriert haben, werden darauf entsprechend reagieren. Es kann ebenfalls 65535 Short Events geben.

In einer der nächsten Versionen von Traintastic können auch die vom Node erzeugten Events verarbeitet werden, so dass für die Weichenstellung eine Rückmeldung erfolgt. Dabei sind spiegeln sich in den Events die Servopositionen wieder. Das ist nicht gleichbedeutend mit der Stellung der Weichenzunge, sollte aber eigentlich. Es könnte aber mit etwas mechanischen Aufwand die Weichenzunge selbst überwacht werden und über die Eingänge des CANMIO-SERVO Nodes könnten dann Events erzeugt werden. Ob man diesen Aufwand betreiben möchte, bleibt natürlich jedem selbst überlassen. Das CANMIO-SERVO Modul kann bis zu 4 Positionen des Servos melden, die alle 4 eingestellt werden können.

Ein kleines Video zum Testaufbau gibt es [hier](#)

CBUS/VLCB als Producer/Consumer Bussystem kann ganz ohne Computer arbeiten, z.B. mit kleinen Stellwerkpulten am Anlagenrand, mit Tastern für die Eingabe und LED für die Ausgabe. So wäre mit 2 Tastern und 2 LED die Stellung einer Weiche möglich, inkl. der Rückmeldung über die LED wie die Weiche gerade steht. Mit einer entsprechenden Kombination von Events können so aber auch mit 2 Tastern ganze Fahrstraßen gestellt werden, das erfordert eine etwas andere Firmware für das CANMIO-IO Modul, weil hier Events nur bei gleichzeitiger Betätigung von 2 Tastern ausgelöst werden (CANCOMPUTE heißt das Modul, mit dem Events nach selbst erstellten Regeln mit Verknüpfungen erzeugt werden).

From:

<https://isnix.de/> - **It's boring when it works!**

Permanent link:

<https://isnix.de/doku.php?id=modellbahn:canmio-servo>

Last update: **2026-04-27 21:21**

