NEWSLETTER

Vol. 29 – No. 6
November – December 2017

Digital Consultants
Rick Sinclair
Curtis Jeung

Current Central Station 3 Version – 1.3.1 (0)
Current Central Station 2 Version – 4.2.1 (0)
Current Mobile Station 2 Version – 2.7

We had a busy couple of weeks in October and early November. Märklin, Inc. exhibited at two shows within a few of weeks of each other. We demonstrated the Central Station 3 and new locomotives at the Rocky Mountain Hobby-Expo in Denver, Colorado and then Trainfest in Milwaukee, Wisconsin. Our next show is the Amherst Railway Society Railroad Hobby Show & EuroEast, January 27-28, 2018, in West Springfield, Massachusetts.

To all of our Digital Club members: We hope you have a safe and joyful holiday season… playing with trains!

About this newsletter
Our first article, finishes up the programming of the mSD3 decoder with the mDecoder Tool for those interested in programming sounds. In our second article, Curtis offers some ideas on sensor usage and explains some of the limitations of use.

mDecoder Tool Part 2

Recap

In the last article, I explained how to set up and program an mLD3 (Märklin Locomotive Decoder 3) with the mDecoder Tool. This article will show how to set up and program an mSD3 (Märklin Sound Decoder 3), including how to set up the sound files. I will not be covering the decoder information or the motor setting. Please refer to the last newsletter for these (Sept – Oct 2017, Vol. 29 No 5).

I should point out, if there is a locomotive profile of the desired locomotive or one that is close, the profile can easily be downloaded from the Märklin server. If one is not available, it will need to be built. In this article, a locomotive sound file was built from scratch because there was no profile for this type of locomotive to download.

Now, assuming that the steps were followed from my last article, I will start with the “Sound” tab in the programmer.

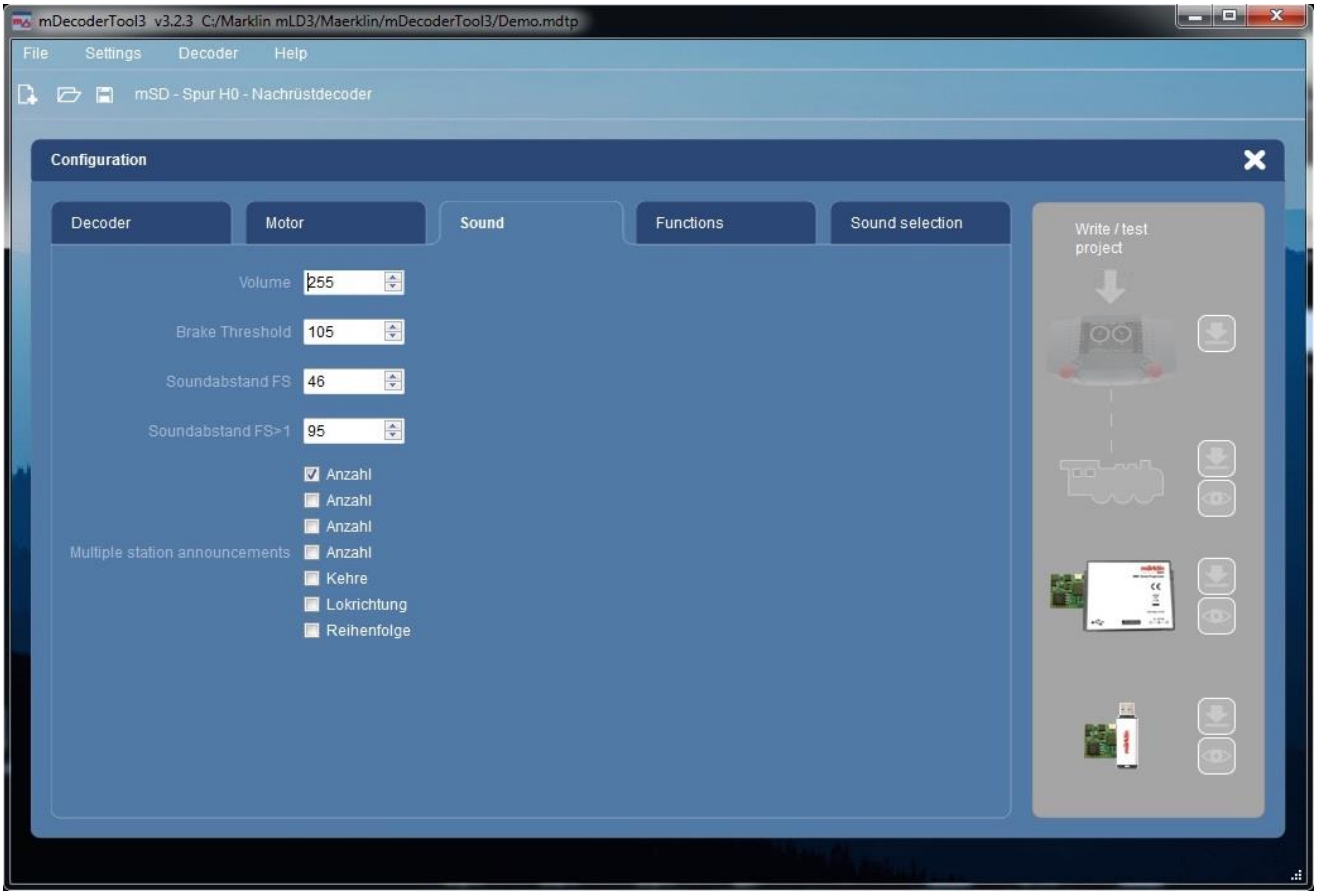

Fig. 1 Sound tab

The Sound Tab

In the “Sound” tab, the over-all volume can be set, as well as the brake threshold and the random sound interval. Like other modifications, I usually just leave them alone until I see a need to change them (Fig. 1).

After any changes are made, proceed to the “Sound Selection” tab (Fig. 2).

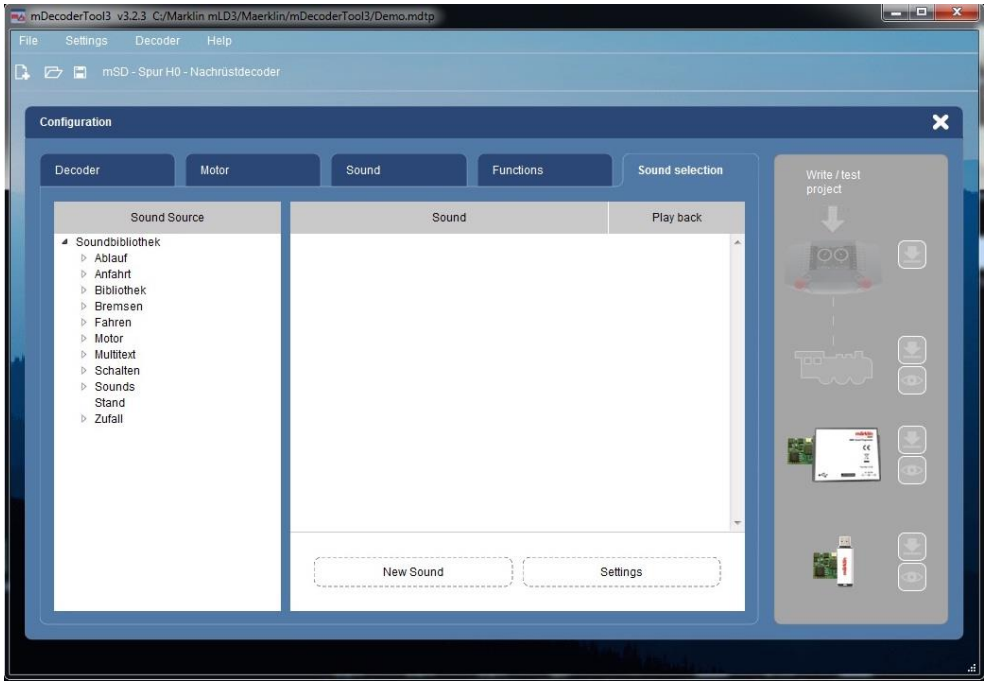

Fig. 2 Sound Selection tab

Sound Set-Up

Click "New Sound" to start making a sound list. For the first sound, select "Driving Noise" (Fahrgeräusch). This and subsequent selections serve as a placeholder until the desired number of sounds are selected (Fig. 3).

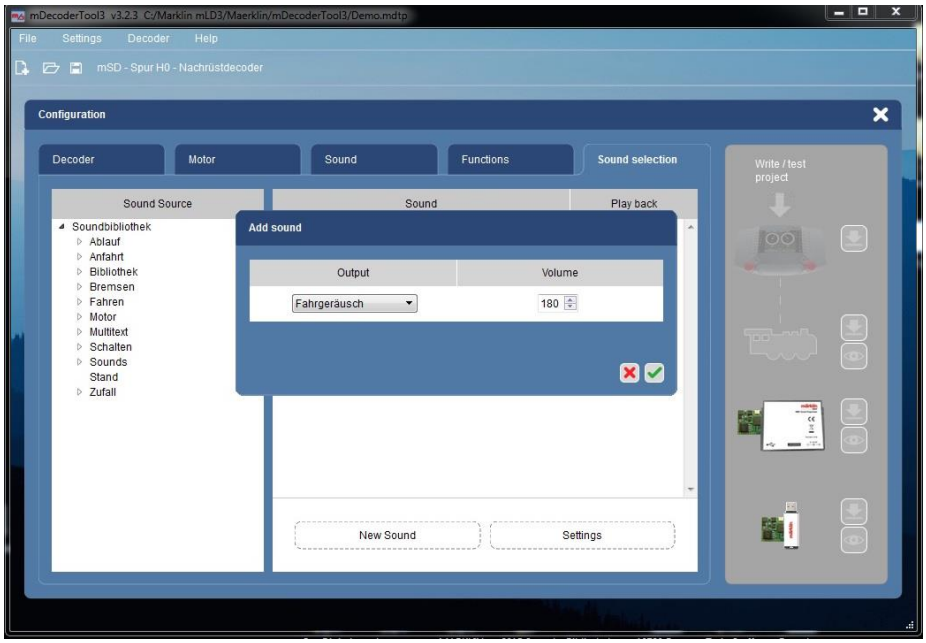

Fig. 3 Adding a sound placeholder

Next, use "New Sound" to select another sound placeholder, such as "Sound without output." The “Sound without output” is important for sounds that depend on the driving sound,

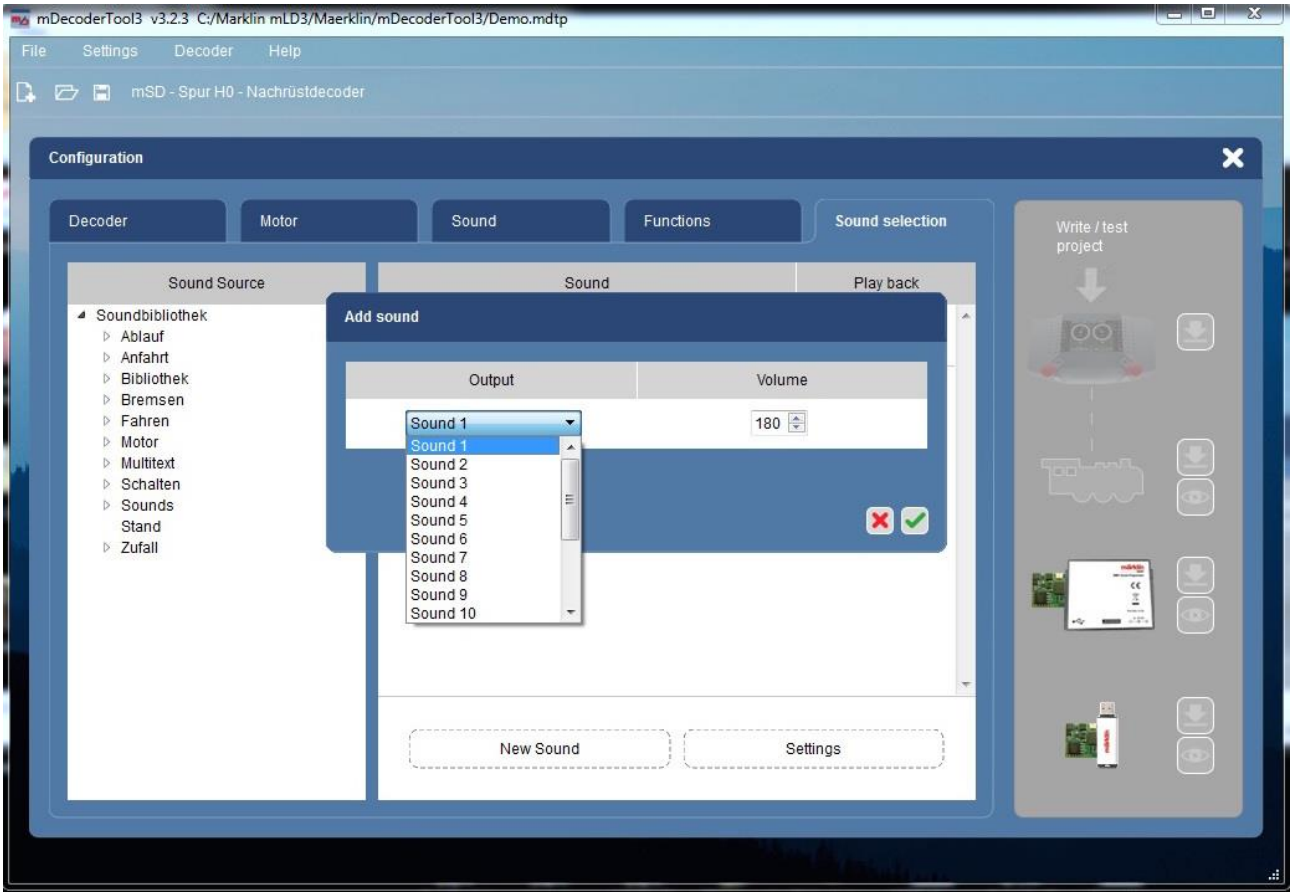

Fig. 4 Adding more placeholders

including sounds from the following categories: Driving, braking and shifting. Additional sounds can be added with the "New Sound" button. The numbering of the sounds is used afterwards for the setting of the functions and for the function mapping (Fig. 4).

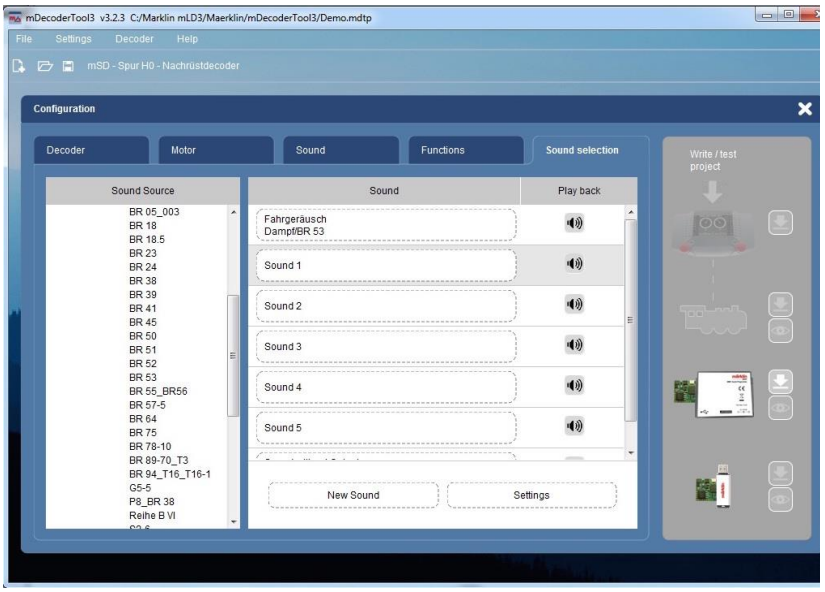

Fig. 5 Drag and drop desired sounds

Assigning Sounds

Once the desired number of sound placeholders has been selected, the individual sounds can be assigned. Choose from the sound source, for example, "Motor" for “driving” sounds. Then select the desired locomotive noise. Drag and drop the sound into the placeholder "Driving noise" (Fahrgeräusch).

Since I downloaded the full sound library, I am using the “Bibliothek” -> BR 53. There is no BR 96 so I chose the BR 53 because it has a similar wheel arrangement (Fig. 5).

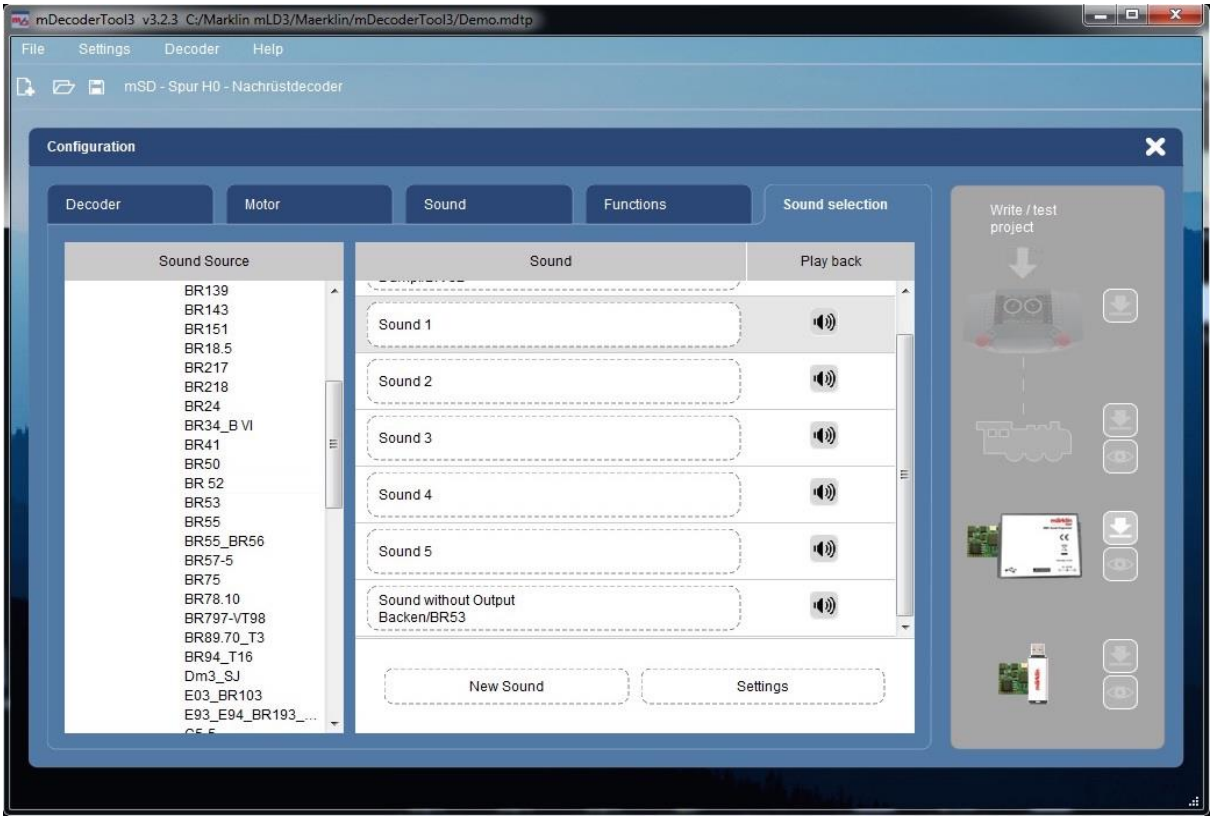

Fig. 6 Sound added to “Sound without output”

In the "Sound without output" field, a sound must be set from the categories of approach, brakes (for example "braking/BR 53") or switching. It's possible to create several fields of "Sound without output" and fill it with sounds from the above categories. Sound without output does not count toward the maximum 28 functions since there will be no function button assigned to them (Fig. 6). If all desired sounds are placed, switch to the "Functions" tab.

Function Set-Up

From the "Function" tab (Fig. 7), the sounds will be assigned to a "Function" button. Select the desired functions in the "Function Buttons". Use the "+" symbols in the right column "Actions" to add an action to the selected function key. The wrench icon is used to configure the respective action. The functions F0, F1, F2 and F4 are pre-assigned with examples; these can be skipped or adjusted.

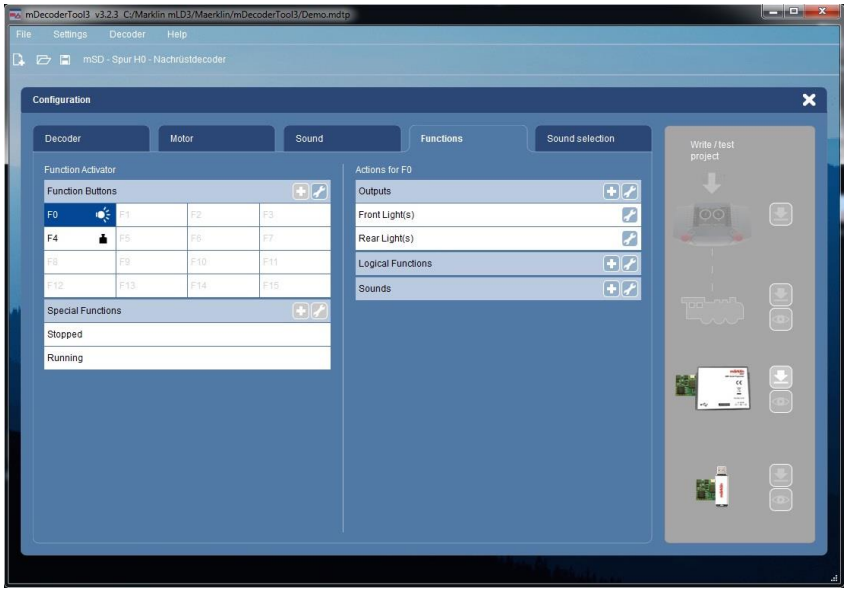

Fig. 7 Function tab

A double-click in the function box will allow the icon and the operating mode to be set (on/off or momentary) and there is also a choice to have it on in analog mode.

By right-clicking on the function, the field can be reset.

By right-clicking on an action, it can be deleted individually.

Assignment of Function and Sound Actions

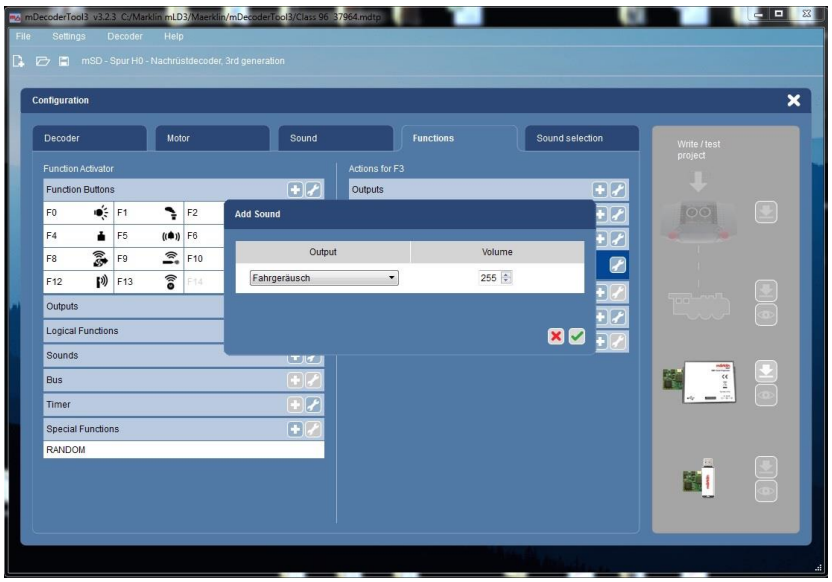

Fig. 8 Sound assignment

To assign a sound to a function button, click on the desired function box, (F3 for example), then select the desired action by pressing + in the "Sounds" row. Then, select the desired sound (Sound 2 for example). I chose the running sounds (Fahrgeräusch Dampf/BR 53) to be the steam sounds, then if necessary, adjust the volume (Fig. 8).

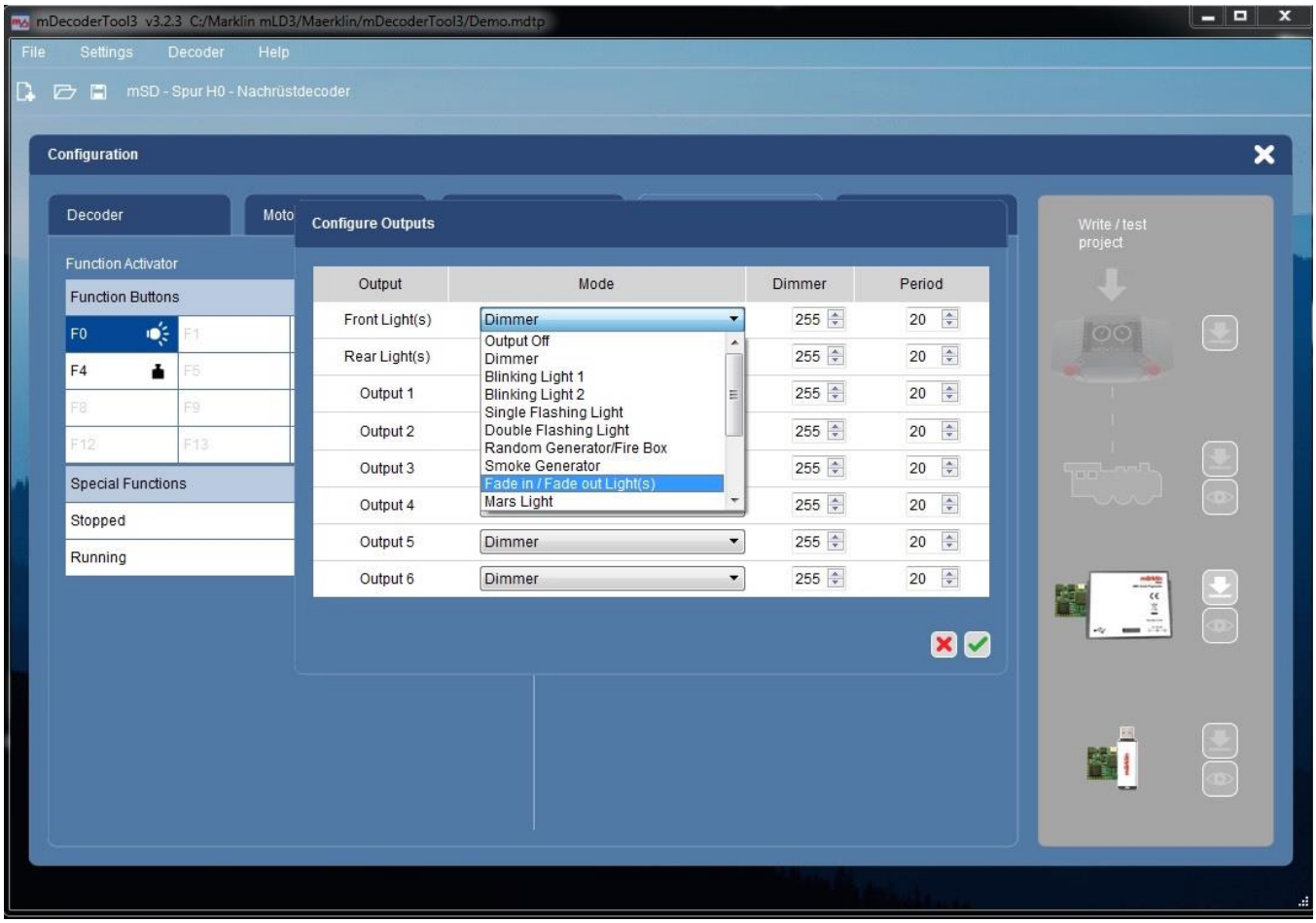

Fig. 9 Physical output edit

By selecting the wrench icon, the characteristics of the physical outputs can be adjusted. For example, I selected “Fade in/Fade out Light(s)” for my headlights and then “Smoke Generator” for the smoke unit (Output1) (Fig. 9).

Once all of the settings have been edited the profile can be uploaded to the decoder. The options are in the right-hand window. Press this button to start transmission to the selected device. If the device is not connected, this arrow is gray.

Decoder Installation

I think it is important to show the installation of this decoder. As I said, the decoder in my class 96 (37964) failed. Once I went through the steps to program the decoder, it was a simple matter of plugging in the 21-pin decoder. The harder part was the speaker.

A Square Peg in a Round Hole

The new Märklin speakers are awesome; they have great sound in a small space. The problem is, the old speaker was round and so was the housing. The new speakers are rectangular in shape (Fig. 10). I tried using the old speaker with the new decoder, but the sound was less than satisfactory. Obviously, I would need to make some modifications.

Since the speaker housing was also the decoder mount and the front light socket, the small speaker that came with the mSD3 sound decoder seemed like it would fit under the decoder like the original speaker, but I would have to cut the housing to make it fit.

I used my hobby knife and started cutting a hole for the speaker (Fig. 11). Once I had a rough hole, I started carefully trimming for a close fit (Fig. 12-13). It was important not to cut too much or this housing would bind the locomotive on curves and it might have trouble on the layout.

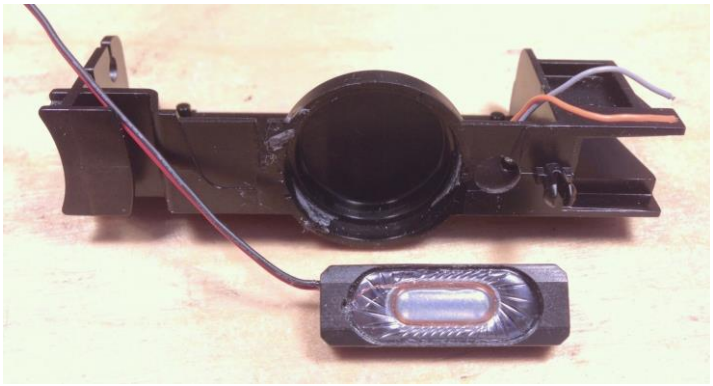

Fig. 10 Old housing and new speaker

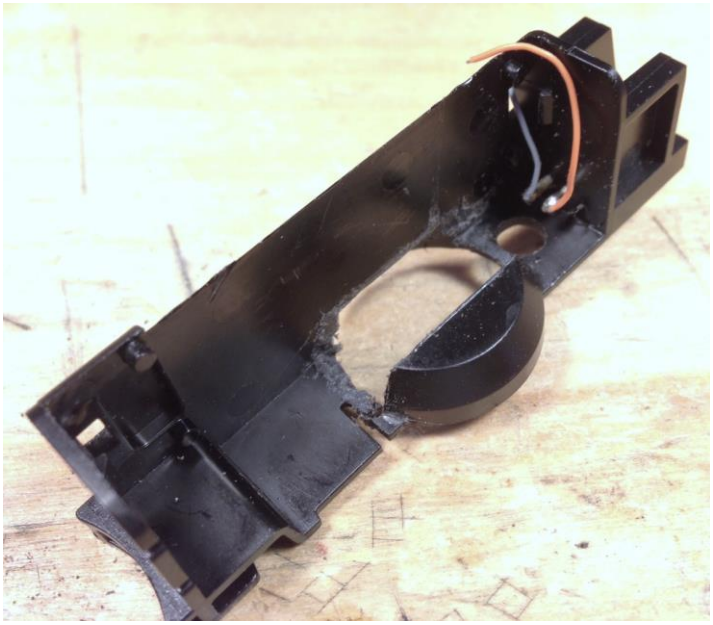

Fig. 11 Trimmed housing

Fig. 12 Rough fit of speaker - bottom

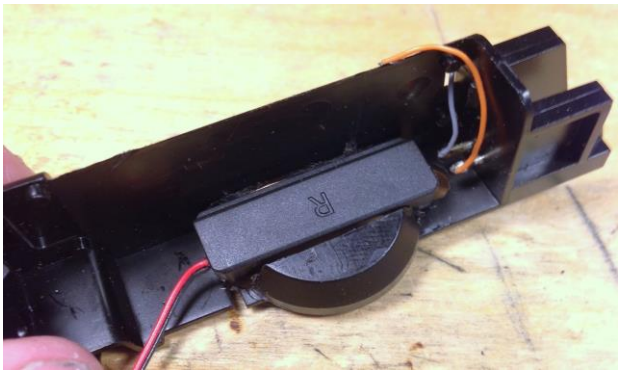

Fig. 13 Rough fit of speaker - top

I used very little adhesive to secure the speaker, because I'm not sure if I can get the decoder/speaker housing as a spare part in the future if the speaker fails. The final assembly went smoothly and the locomotive sounds great (Fig. 14). One more project off my bench!

Enjoy your hobbies!

Rick Sinclair

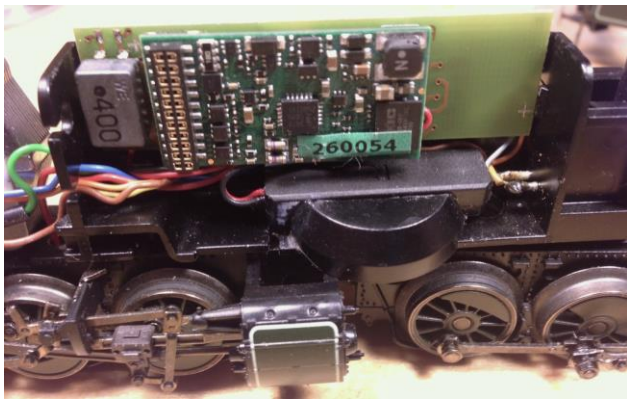

Fig. 14 Finished install

Practical Track Events and Sensor Use

In many of my articles, I have written about the processes of enabling automation on your layout, mostly from the technical standpoint of hardware and software setup. This involved creating contact tracks or feedback sensors, which send a unique signal to your Central Station. The CS will then activate the required automation script according to instructions written by you. In this month's article, I would like to offer some ideas on sensor usage and explain some of the limitations of use.

Basic Sensor Types

There are three commonly used sensor types: contact, circuit and magnetic reed. Reed sensors are momentary connections and commonly found in 2-rail track operations (i.e. LGB and Trix). Märklin's 3-rail track uses circuit tracks that are trigged by the center slider (or ski) and its connection is also momentary. Contact track (aka track occupation sensors) make connections that are constant. They are easily integrated into 3-rail layouts. For the purpose of this article, 2-rail users can cross-reference most reed sensor applications as circuit tracks, this is due in part to its momentary nature.

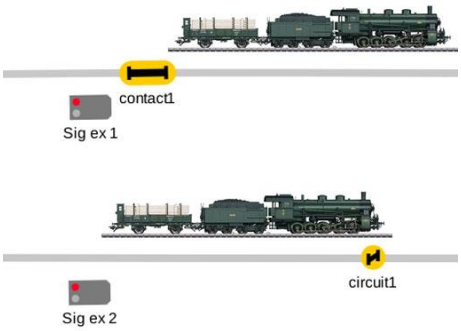

Fig. 1 – Basic sensors and prescribed locations. Circuit tracks often need length of train placement.

It is important to remember the distinction between constant and momentary, because it can directly affect their placement on the layout. Fig. 1 illustrates the primary difference in placement with a simple mainline block setup. 'contact1' is placed to indicate complete train clearance of 'Sig ex 1', before switching to the 'Stop' aspect. 'circuit1' is placed similarly to have complete clearance of 'Sig ex 2'. However, one drawback of 'circuit1's' placement is that it becomes a limiter on the length of train allowed. For example, if you add cars to the train shown in the 'circuit1' example, you can see that it would not have complete clearance beyond 'Sig ex 2'.

Basic Track Blocks

A track block is defined in this article as a line of track that only allows for single train occupation. In other words, you aren't allowed to have two trains within a single track block. To control this, I use sensors to tell me when a train has cleared a block behind it (allowing a following train to proceed).

In Fig. 2, I illustrate a vacated block (track between 'Sig ex 0' and 'Sig ex 1') and a corresponding control sensor's location ('contact1'). The lower track is the setup using a circuit track ('circuit1'). Note that 'circuit1' is spaced ahead of 'Sig ex 2' to insure the train has cleared the preceding block. This is because the contact ski is typically under the locomotive.

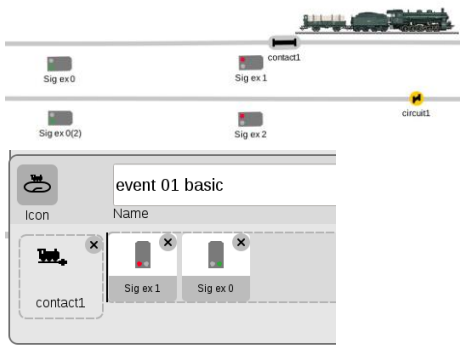

Fig. 2 - Basic block event script steps.

The lower half of Fig. 2 shows the event script for ‘contact1’. The event script for ‘circuit1’ would be identical. The script operates by stating, when ‘contact1’ is activated, switch ‘Sig ex 1’ to red aspect, then switch ‘Sig ex 0’ to green aspect. The event steps are activated in the order in which they are displayed. One detail about the Event script, that is not shown, is that ‘contact1’ is set for ‘Departure’ trigger. There is no such setting available for magnetic reed or circuit tracks, and I will explain this trigger in the next example.

Prototyping the Signal Change

If you examine the signal change procedure found in Fig. 2, you will notice that both signal lights (‘Sig ex 1’ & ‘Sig ex 0’) change at the same time, when the train has cleared the previous track block. In prototypical operation, ‘Sig ex 1’ would have changed to red aspect as soon as the locomotive has begun to pass. To operate the signals prototypically, I have now split the event script from Fig. 2 into separate events with an alteration in the Trigger settings. These settings are shown in Fig. 3. To open the ‘Settings contact1’ window, click on the ‘contact1’ icon to the left of the event sequence (upper half, Fig 3).

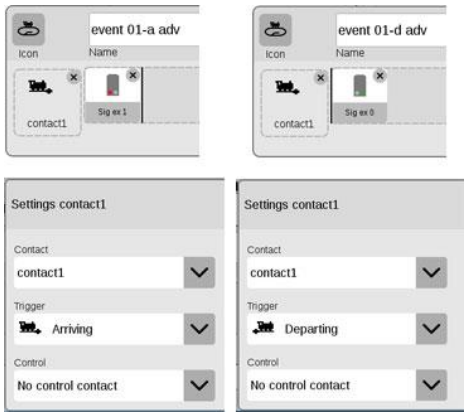

Fig. 3 - Arriving and Departing trigger assignment.

The lower half of Fig. 3 shows the Trigger settings for the respective event script. The options are ‘Arriving’ and ‘Departing’. The ‘Arriving’ contact is triggered when a train’s wheels make contact with the contact track. The ‘Departure’ contact is triggered when the same train’s wheels disconnect from the contact track.

To illustrate the effects of the scripts in Fig. 3, follow the sequence shown in Fig. 4. The top shows the signals prior to contact with ‘contact1’. The middle shows the effect of ‘contact1’s arrival script. The bottom shows the effect of ‘contact1’s departure script.

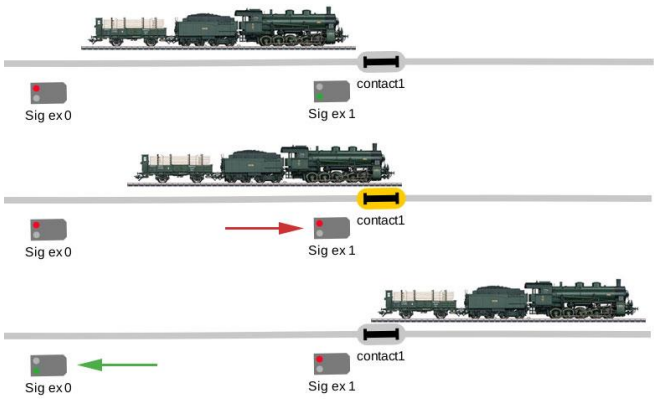

Fig. 4 – Track events with differing trigger assignments.

In this example, the separation of events into arrival and departure scripts is basic and simple. As mentioned previously, this setting is not available with reed or circuit tracks. You would need two circuit tracks to simulate what is accomplished with the single contact track. The ability to separate these two triggers allows us to create more advanced and complex event scripts. In the remainder of this article, I will show you some more advanced and technical examples of the trigger settings and contact placements.

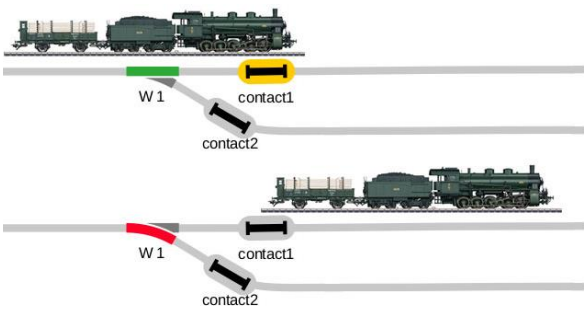

Fig. 5 - Re-routing with a departure trigger event.

Alternating Route

In this example, I show how you can re-route trains to an alternate line. This approach can be used to redirect trains to an open line or an alternating route. The thing to remember is that the 'control' train is used to re-direct the train behind it. This can create a sense of randomness to your track automation, because you aren't directly controlling a train's destination. Fig. 5 shows the placement of the contact tracks after a turnout. It illustrates how a departure (from 'contact1') insures that the train has sufficiently cleared the turnout ('W1') before switching to the alternate line. Fig. 6 shows the event scripts for each of the contacts (right hand script for 'contact1'). They are both set for 'Departure' triggers. Both events are programmed to change the turnout to the opposite track.

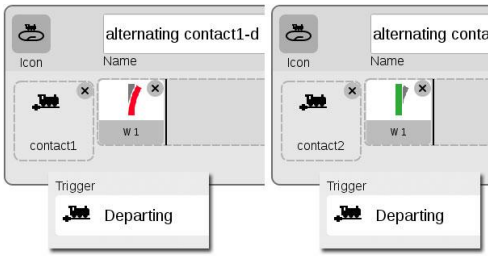

Fig. 6 - Event scripts for the two opposing contacts.

Track Crossings

In a train crossing, we separate the arriving and departing trigger across two different contacts to control 'Sig ex 1'. As in the previous example, 'contact2' uses a departing trigger to insure the crossing train has fully passed the intersection (Figs. 7 & 8).

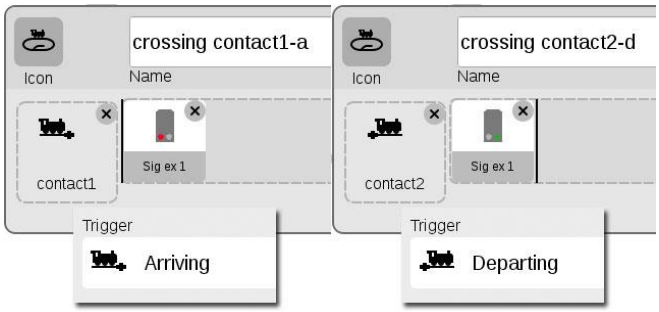

Fig. 7 - Two contact scripts used in a simple track crossing.

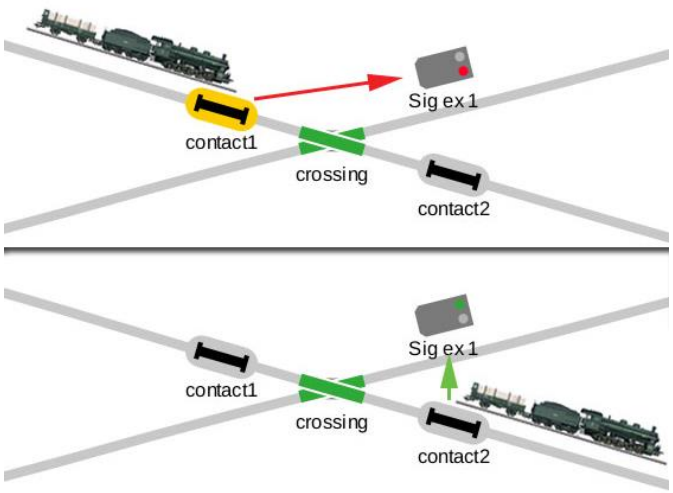

Fig. 8 - Signal controlled by two contacts on track.

Depot Stop and Bypass

In the next example, I combine a couple of events within a single block. The example itself is to show how to create a regular depot stop (off the main line) and also allow a freight train bypass the depot. Due to its complexity, I will split up the events for clarity. The example itself is only a single approach to the effect and has been simplified to highlight the specific control mechanisms.

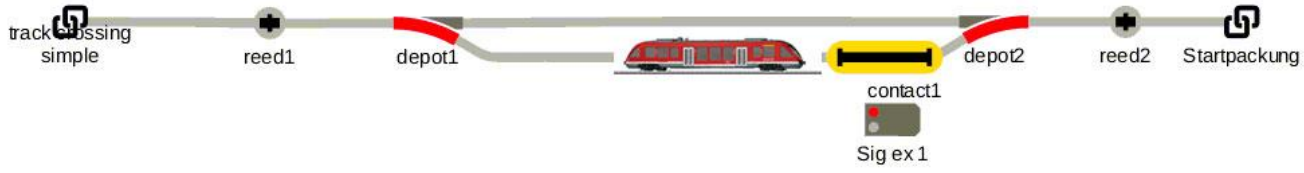

Fig. 9 - Setup of sensors for use in a depot w/ bypass. Note 'contact1' reaches before and after 'Sig ex 1.'

In Fig. 9, I have placed ‘Sig ex 1’ on the depot stop. On the track, I have extended the length of my contact track to reach in front of and behind ‘Sig ex 1’. In a normal track block contact, I would only have the contact track behind the signal. Note: The graphic shows a longer than normal contact track. This has been edited and no such extended contact icon exists within the CS.

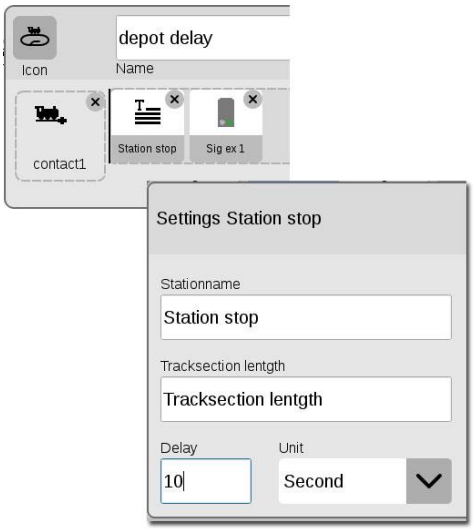

Fig. 10 - Delay timing script step added to event.

Fig. 10 shows the ‘Arriving’ trigger event script. The script is defined as, “When the train arrives, it starts a 10-second timer. When the timer counts down, the signal is then set to green and the train can proceed.” The reason why ‘contact1’ was extended is due to ‘Sig ex 1’ being set to stop a train before it reaches the signal. If my contact was only placed behind ‘Sig ex 1,’ then the train would never activate the script, which starts the delay timer, which was added to the first step in the event list. To add a delay timer step in your script, click on the ‘Text’ of your event script window.

Normally, I would reserve the ‘Arriving’ trigger to change the signal light to red aspect. Because it has already been triggered to start the timer and switch ‘Sig ex 1’ to green aspect, I have another script for ‘contact1’ using the ‘Departing’ trigger, which will reset ‘Sig ex 1’ back to red aspect. (I won’t repeat the script image here, as it is being used in previous examples.) I could have accomplished the same effect using two different contact tracks, but as you can see, it is just as effective to extend the one contact. Fig. 11 shows how to set up circuit tracks instead. In this case, ‘circuit1’ should be placed prior to any stop tracks.

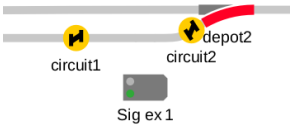

Fig. 11 - Circuit track placement in depot example.

The second part of this example is the ‘Bypass’ event. If you notice in Fig. 9, I have placed two magnetic reed sensors on the main line (‘reed1` and ‘reed2’). This allows me to bypass the depot by placing magnets on trains that do not need to stop. Fig. 13 (left) shows that ‘reed1’ will activate the event to change turnouts ‘depot1’ & ‘depot2’ to straight pass through. Fig. 13 (right) uses ‘reed2’ to reset the turnouts to detour through the depot line. Fig. 12 illustrates the event for each reed sensor.

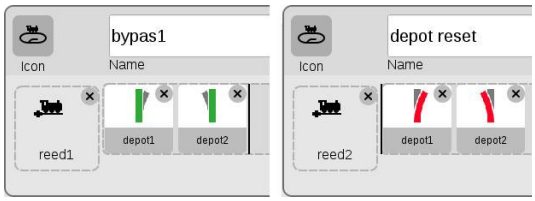

Fig. 12 - Bypass and re-set event scripts for Fig. 13.

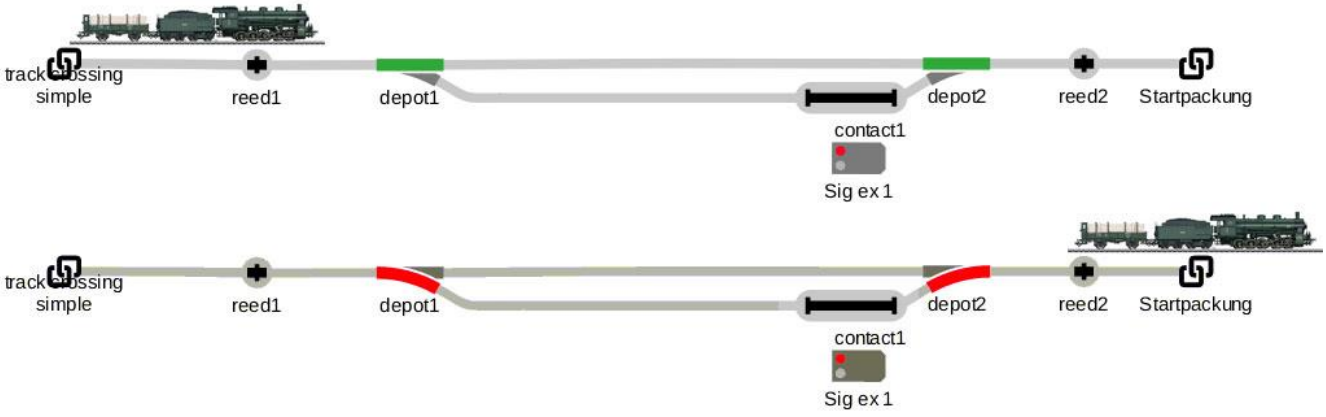

Fig. 13- Bypass event track results.

Simple Shuttle

To create a simple shuttle that traverses back and forth between two end points, you'll be creating two simple event scripts. The Central Station 2 has a pre-set that you would assign to a single locomotive to do just that. However, this has now been removed from the CS3 controller. While this may seem a loss, I can guarantee there is a better way that is more robust in functionality and use. You can even apply this method to the CS2, which makes the shuttle function obsolete. I will explain the simple shuttle operation first.

For shuttle operations, you'll need to set up contact tracks covering the end points of each line. The event scripts will use 'Arriving' triggers. To explain the operation: The shuttle train will arrive at the end terminal, it will stop, pause, switch directions (as a simple direction change), pause again, then accelerate in the opposite direction back to the main line. Conceptually, the sensor action will look like Fig. 14. Remember that we will be using 'Arriving' triggers.

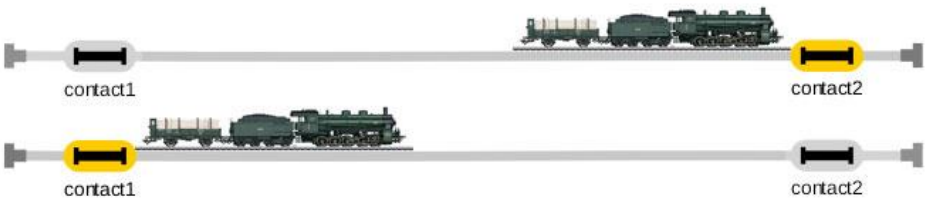

Fig. 14 - Simple shuttle contact placement. It illustrates arriving triggers.

The script events for each contact are essentially identical. I say essentially, because while you can add bells and whistles (literally) I'll only give you the basics. There are two things you should notice about these script events.

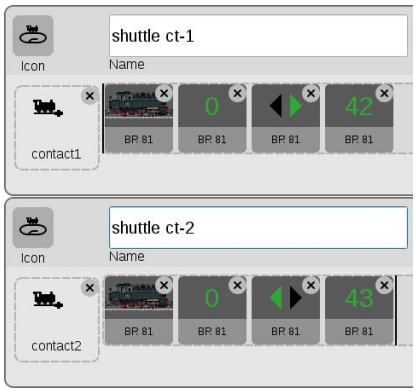

Fig. 15 - Simple shuttle events and contact assignments

First, the direction change that I've assigned to each contact (the third block on each event script, Fig. 15). The locomotive that you use on the shuttle may accelerate toward the end of the line instead of back toward the main line. To change this, click on the step itself to enter the step edit window Fig. 16. You can change the direction step by clicking on the 'Value' button. I've added a 2 second delay which will create the pause prior to the final acceleration step.

The second item to notice about these event scripts is I've assigned a locomotive to an automated script. As a warning, these are the **only** types of automation event scripts that you should currently assign locomotives to. I will explain this warning at the end of the article.

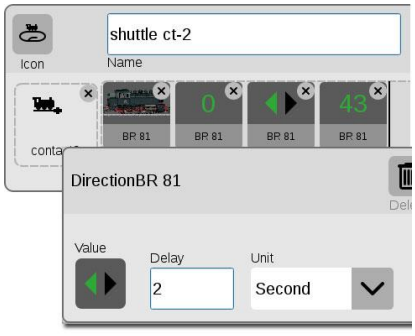

Fig. 16 - Event Step edit window w/ delay timer setting.

What has been a very nice improvement for this type of shuttle script over the CS2, is the ease in which you can change locs. You won't have to recreate the script event if you wish to use another shuttle train. To change locs, click on the locomotive step to pull up the step edit window. Next, click on the 'Locomotive' pull down menu to reveal a full list of all the locomotives in your inventory. Simply select the loc that you wish to run as a shuttle. Remember to change this in the complementary script for the other end of the line.

All of the examples here show how a singular trigger event can activate simple procedures from signals, turnouts, and loco re-routing. I do wish to mention a couple of concepts that for clarity, I did not mention in the examples. First, the signal light controls I use can be separated into signal and stop sections. It is possible to write your scripts where a brake module can be triggered independently from the signal. Second, except for the shuttle train example, notice how none of the sensors address a specific locomotive. Even the magnetic reed sensor in the 'Depot Stop and Bypass' example addresses a specific locomotive. As I mentioned earlier, I will explain this important fact at the end of the article.

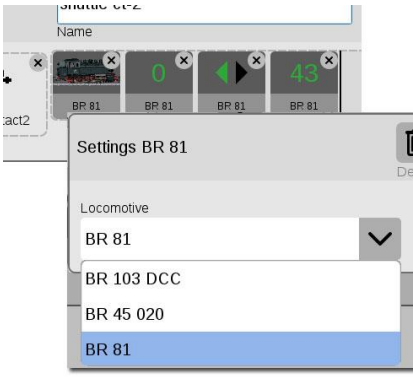

Fig. 17 - Locomotive re-assignment in step editor.

These several examples show how you can set up sensors and create event scripts for automatic control around your layout. Going beyond the practical application of having such controls, I hope you can take some of the methods or concepts and be inspired to create new functionality to your layout. Using the three different sensor mechanisms (contact, circuit and reed), you can see it is possible to mix sensor types at your choosing. But, you should be aware of the advantages and limitations of each. This article covers some beginner level sensor and event scripts. In my next article, I will show you some intermediate to advanced level scripting examples that you can use.

The Myth of Automating Locomotive Events

I think most of us with train depots or stations have envisioned our trains approaching at a reduced speed, blowing a whistle or ringing a bell as it comes to a full stop. Then, often reversing the process as it accelerates back to the main line. These sequences of operation can easily be scripted into an event using any of the Central Station controllers.

Fig. 18 shows an event script for a locomotive entry into a depot station. Notice the matched sequence of the event steps in relation to the train's position on the track. Attempts to automate such an event using track sensors will eventually fail. The displayed example will work as it is, but it won't work on a full working layout.

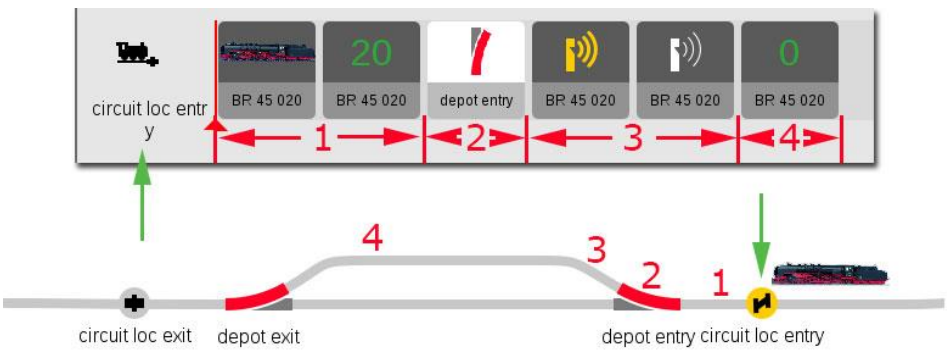

Fig. 18 - An ideal locomotive event script.

The primary reason for this failure is that the CS3 is currently blind to the location of the locomotive. In example Fig. 18, note how the activation of 'circuit loc entry' would seem correct, because it is the only locomotive used in proximity of the sensor. The specific loco is addressed by this specific event sensor.

Now, let's examine what would happen when you add another locomotive. I will display an entire loop to clarify the results (Fig. 19). Our second locomotive is now the locomotive that activates the sensor (green arrow). However, we wrote our event to activate the specific locomotive in our setup. Our event steps proceed accordingly, but the location of the programmed locomotive doesn't match where the event should occur (steps 1, 3 & 4). The track change in step 2 will function properly, because it is in accordance with the train that triggers the 'circuit loc entry' sensor.

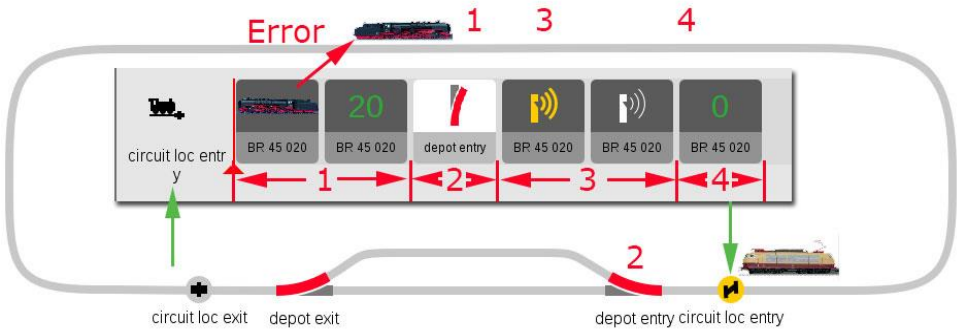

Fig. 19 – Automation if locomotive causes a location error.

Going back to the 'Simple Shuttle' in the main article, it was okay to assign a locomotive to the automated script because in a simple shuttle, there is only one locomotive that can activate both sensors. In the end, the general rule of thumb for track triggered events is, "You can program track events for automation, but you can't automate locomotive events." On the bright side (and as a former programmer), I know that this capability can be implemented with the current hardware available in both the CS2 and CS3, but it will require some additional software development.

Curtis Jeung

Upcoming appearances:

Amherst Railway Society Railroad Hobby Show & EuroEast
Eastern States Exposition Fairgrounds, Mallory Bldg
1305 Memorial Ave
West Springfield, Massachusetts
January 27 – 28, 2018

Märklin Enthusiasts of America (MEA) Spring Meet
Steamtown
Scranton, Pennsylvania
May 6, 2018

To contact Rick and Curtis for help with your Märklin Digital, technical and product related questions:

Phone: 650-569-1318 Hours: 6:00am – 9:00pm PST. Monday through Friday.

E-mail: digital@marklin.com