märklín DIGITAL CLUB

# NEWSLETTER

**Vol. 34 – No. 3**
**May - June 2022**

**Digital Consultants**
**Rick Sinclair**
**Curtis Jeung**

**Current Central Station 3 Version – 2.4.0 (5)**
**Current Central Station 2 Version – 4.3.0 (5)**
**Current Mobile Station 2 Version – 3.148**

---

Finally, we are back to attending train shows! In early April, we did our first show in two years. We attended the Rocky Mountain Train Show in Denver, which coincidentally was our last show in 2020. The show was very well attended, and we were able to talk to many Märklin, LGB and TRIX fans in the area. Now, we are looking forward to the National Garden Railway Convention in Denver. We have been preparing for this show for weeks and it should be well attended.

Our first article discusses how to stop a train with a signal and a brake module, and our second covers the Central Station 3/3+ Block Macro.

## Stopping a Train - Controllers

While the topic of controlling trains has been covered, we never fully explained the different ways to make trains stop and start without turning the speed control knob. The automatic starting and stopping adds a sense of realism.

In digital, or analog, operation, a train can be stopped with a signal, an m84 or a Brake Module. I will explain the digital process here using a digital signal as an example.

A digital signal can be controlled in two ways. One is the use of the 72760 Signal Control Box. This will allow manual signal control with the control box. The other way is to control the signal with a digital controller, i.e., Mobile Station or a Central Station.

While the Central station is by far the easiest way to control a signal, the Signal Control Box is very convenient since the box will always be in the same spot and it will be easily reached. To change a signal with the Mobile Station a person must navigate the “Turnout/Signal” screen to find the desired turnout or signal. Caution, this could take a little time if you are in an emergency situation.

## Stopping a Train - Signal Set-up

The set-up will basically be the same for any controller or signal. It requires a little wiring and track insulators.

For C-Track and Trix track, the 74030 insulators are required. K-Track will need a 7500 Ground Terminal Clip, two 7504 Third Rail Terminal Clips and two 7522 insulators per signal. LGB would need two 10260 Insulated Rail Joiners along with 50160 Track Feed Wires or 50161 Track Feed Terminals. 1-Gauge would use two 59090 Insulated Rail Joiners and two 59096 or 56031 Feeder Clips per signal.

For simplicity, I will use C-Track as an example since that is the track system that most people use. Track insulators must be installed to insulate a “Stop Section”. There must be insulators at both ends of this stop section. The power for this section will be cut to stop the train. Figure 1 shows that insulators must be installed on the center rail contact (red rail for all track systems).

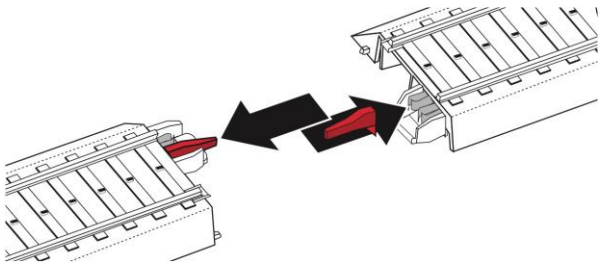

Fig. 1 C-Track insulators

In figure 2, the wiring diagram from the signal manual shows the basic wiring. Please note: There are standard red and brown track feed wires inserted into one socket of the decoder to power the signal and there are two red wires that go into another socket of the decoder that will receive and supply track power for the “stop section”. It doesn’t matter which of the two red wires goes to the track supply or the stop section, so you can’t get them wrong.

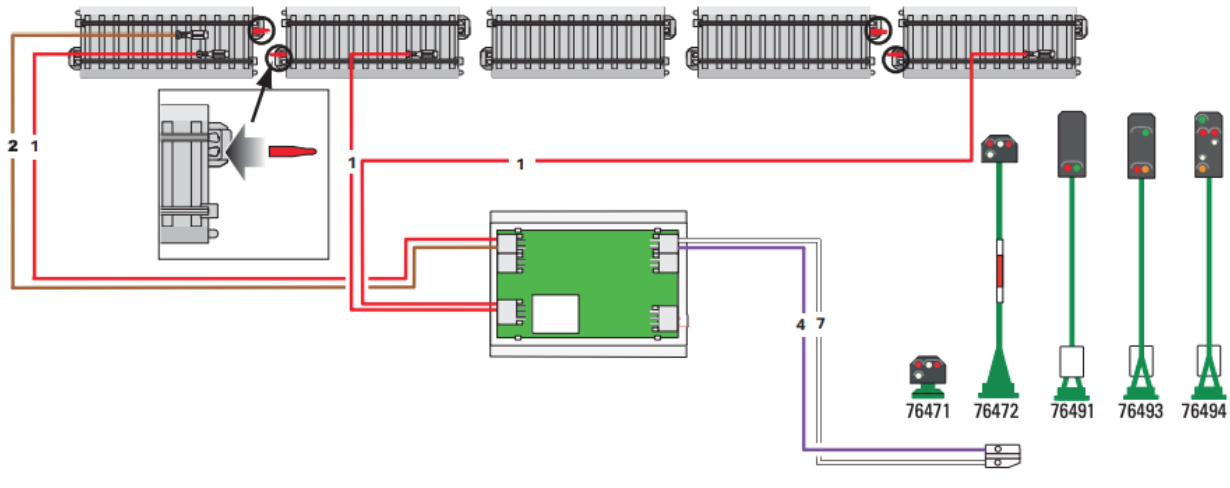


Fig. 2 – Signal wiring

This is the basic set-up for either the 72760 Signal Control Box, a Mobile Station, or Central Station. To connect a signal to the 72760 Signal Control box for manual control only, just follow the instructions that come with the 72760.

With this basic wiring for a “stop section” a signal will control track power within the stop section. Now the signal can be operated manually with the 72760 Signal Control Box, or once entered into the controller, it can be controlled by Mobile Station or Central Station.

## 72442 Braking Module

The next step for realism would be adding a “Brake Module”. What a brake module does is slow a train to a stop rather than just cutting the power and instantly stopping the train. This module triggers the braking delay in the locomotive decoder to bring it to a slow stop.

One small note is that Märklin named this unit “Signal Module” on the unit itself and the catalog calls it a “Braking Module”. They are the same thing.

A brake module will be set up with three separate track sections instead of one. For C-Track this will require eight track insulators (one bag of item 74030) per brake module. If many brake modules are to be set up, I would suggest buying a box of 74030 insulators that includes five bags.

Figure 3 shows the wiring for the three brake sections. Section 1 is where the locomotive decoder gets the trigger to slow down, Section 2 is the actual slow down section, this should be at least 2 sections of straight track or more if desired. Section 3 is the emergency stop. This section will not have any power when the unit is activated so the train cannot overrun the brake section and keep going.

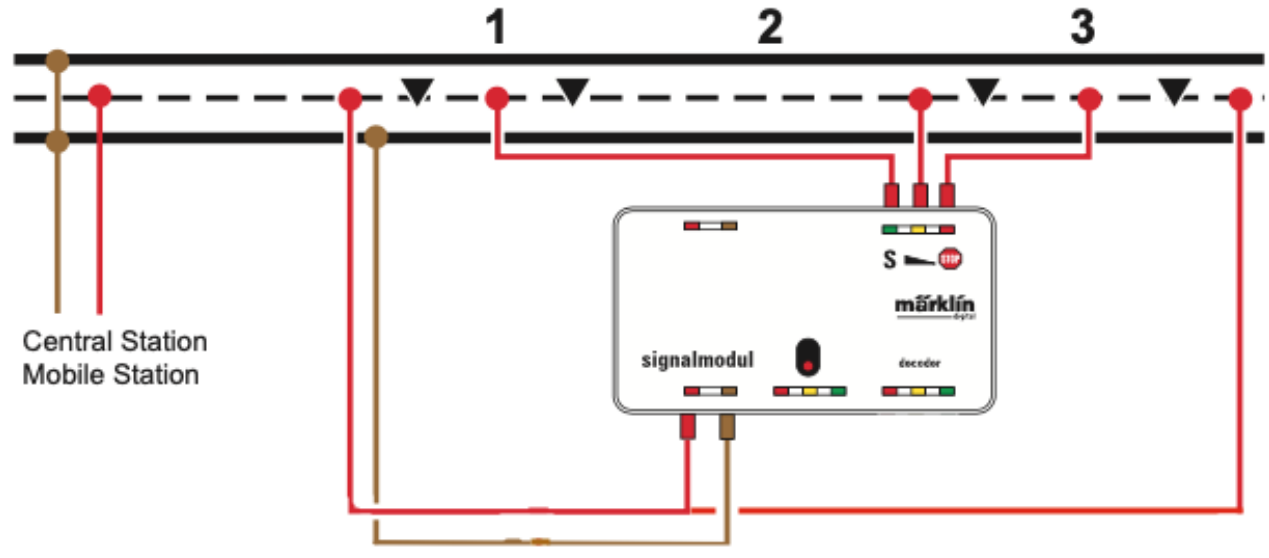


Fig. 3 - Brake Module wiring

Again, there are a couple of things to note. The triangles shown on the track in figure 3 indicate where to place the track insulators. Also, the brake module needs track power to supply the unit itself so it can allow track power to flow when the unit is not activated (green signal for example). There is no digital decoder inside the brake module, so it still needs a “switch” to activate it.

## Brake Module Activation

Basically, there are two ways to activate a brake module. Analog or digital. Analog control would be through a 72720 Control Box. This will be a standard box on a control panel that is easy to activate. The wires will be run from the control box to the brake module (Fig. 4). The control box can be powered with either a secondary power source (as shown) or track power.

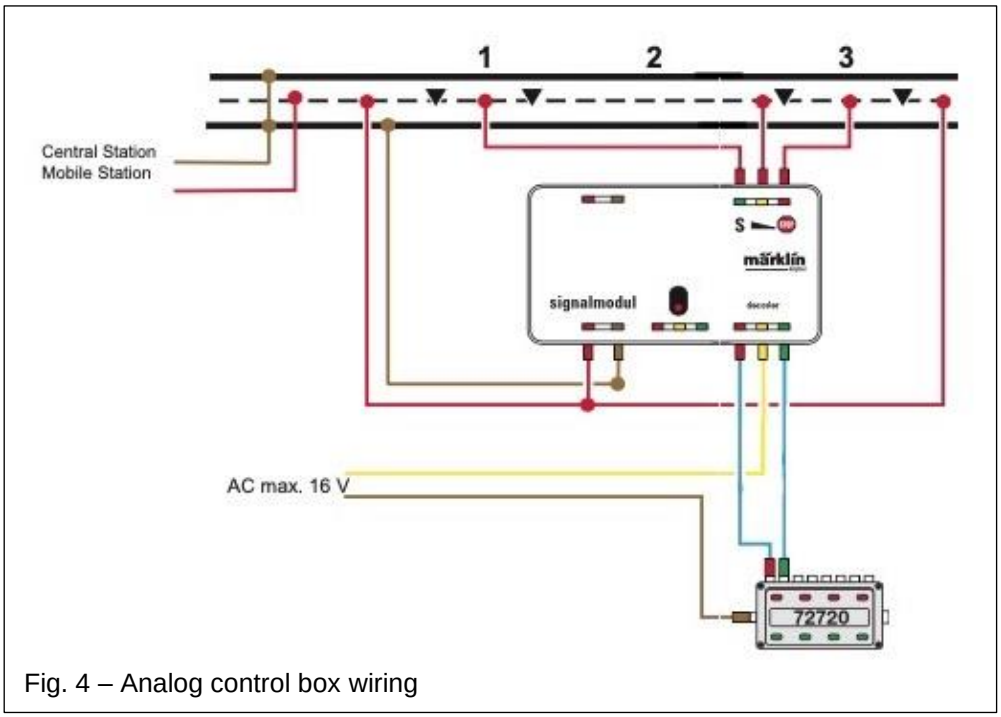


Fig. 4 – Analog control box wiring

Digital control has two ways to activate the brake module. The simplest way is to plug the digital signal into the brake module. With this set-up the signal will activate the brake module when the signal is red and deactivate the brake module when the signal is green. Figure 5 shows the wiring for signal activation.

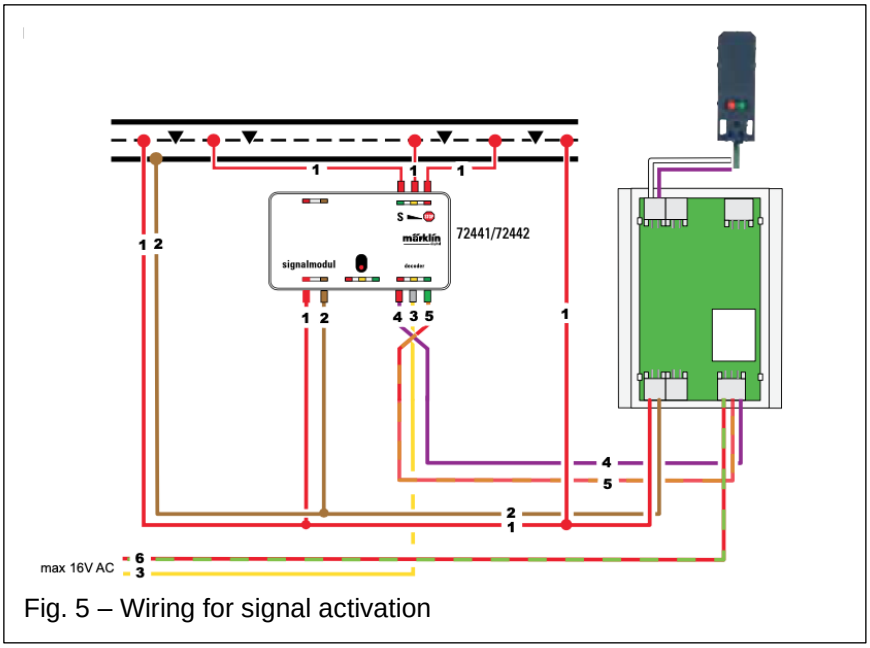


Fig. 5 – Wiring for signal activation

## Central Station Activation

For those of you looking for that “next level” of realism, there is another way to connect the brake module. This method is intended for automatic routing that a Central Station is capable of.

If an m83 (60832) is used to control the brake module, it is possible to separate the activation of the signal and the activation of the brake module. So, the signal can turn green, and the train can sit there for a few seconds with a timing delay, then the brake module can be deactivated, and the train can proceed. Our theory behind this is that a train must leave on a schedule. So just because the signal is green, doesn’t mean the train can go, the engineer must leave at the time the train is scheduled to leave. We like to think of this as the next level of realism that nobody will ever notice! Figure 6 shows the wiring for activation of a brake module with an m83, a digital signal would just use track power with no other wiring.

Something else to remember is any accessory that can be activated digitally can be controlled with a digital controller. This means that a Central Station can turn it on and off. Once this is achieved the activation of the accessory can be put into an automatic route. This also means that analog devices can be controlled digitally and subsequently put into an automatic route (see newsletter article July - Aug. 2021). So, in short, by using an m83, it is possible to use old analog signals on a digital layout with a brake module in automatic routing, an M84 can also be used the same way for other accessories that just need an “on and off”.

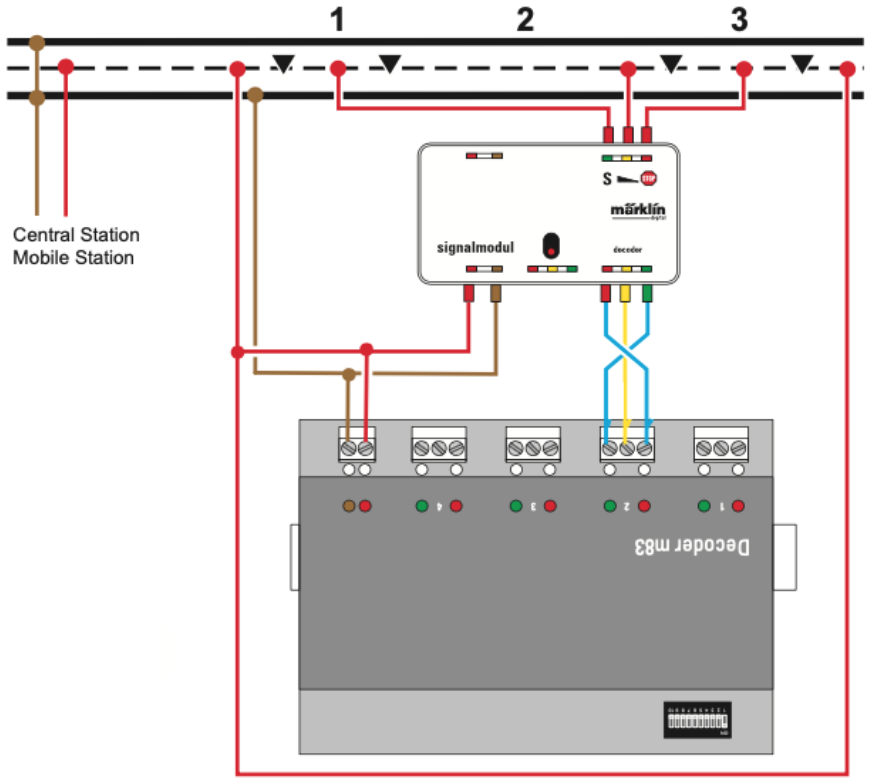


Fig. 6 – m83 control of brake module

## Locomotive Stopping Distance

Depending on the speed of the locomotive, it might run too far through the “slow down” section. The braking delay can be set in the locomotive to be longer or shorter and most likely need to be adjusted to stop in the correct length of track.

## A Few Last Notes

There are more ways to activate a brake module, for instance the use of an m84 can be used with a circuit track or reed contact. This would essentially be “analog” routing which I covered in the March – April 2019 newsletter and concluded in the May – June 2019 newsletter. This is more for Mobile Station users who want to do automatic routing.

Also, an m84 will stop a train also by cutting track power. Think of it like the first example, a stop section controlled by a signal, except without the signal. This technique would be used where a train would need to stop but nobody would see it slow down first, like in a tunnel or hidden storage yard.

As one would guess there are many combinations of components to make a train stop. I have just tried to cover the basics here and encourage people to think about what technique will work best for them, and with the controller they are using.

**Enjoy your hobbies!**
**Rick Sinclair**

---

## Central Station 3/3+ Block Macro

In the CS3 software update for version 2.4.0 (5), a new Block Macro was added to the event list. In this article, I will walk through this new addition. Then give some thoughts and comparisons to the previous blocking method used in my previous articles.

To view the Block Macro, you will need to enable the 'Events - Extended Mode' functionality of your CS3 first. This setting is found in the CS3's 'Settings> Track:Protocols and operations' panel. This allows you to view the 'Add Macro' option in the Events menu.

The Block Macro has five preset containers. (Fig. 1) These are all conditional containers. In other words, they are checks where the CS3 evaluates the status of each container. I will explain the contents of each container first.

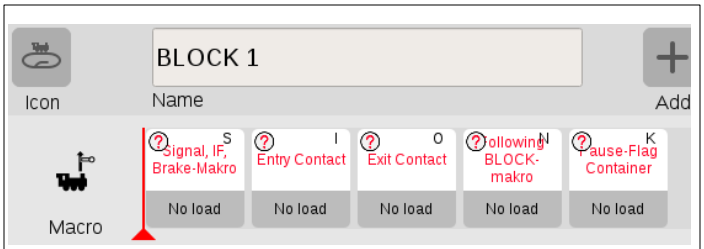

*Fig. 1: The five containers in a Block Macro*

"Signal, IF, Brake-Makro" (SIB container)- The first container allows you to nest one of 3 elements: A Signal, IF macro, or the Brake-Makro. For the signal, you would place and set a signal into this container. The next two are Macro events (IF, Brake-Macro). These are additional Event macros that you may use for braking/slow down or whatever IF condition you may need. For example, if an assigned Brake-Macro is in operation, this condition could be set to be 'true' and then the Block event would proceed to the next container.

"Entry Contact", "Exit Contact" - These containers will host assigned contact points that will indicate the entry point of a block, and the exit point of a block.

"Following BLOCK-makro" (FB container) - This container will contain next block on your route that follows the current block in edit. This indicates that you may need to have your track routine in a pre-defined sequence. You'll need to have a 'following' block macro added into your CS3's events inventory to load this container. In the Fig. 1 example, you would need a 'BLOCK 2' macro in place to add into Block 1's 'Following BLOCK-makro' container.

"Pause-Flag Container" (PF container) - Of the five containers in the Block Macro, the Pause-Flag Container requires some discovery. Through experimentation, I found that this container is meant to hold a contact, where nothing else was able to load. The PF container is also found in the Brake-Macro. Its function is basically to halt the macro if the setting in this container is determined to be true.

**Notes about the containers**

The "Entry Contact", "Exit Contact" and "Pause-Flag Containers" are the only containers that you can configure while in the container edit window. It is where you can adjust the condition to read if the contact is on or off. (Fig. 2)

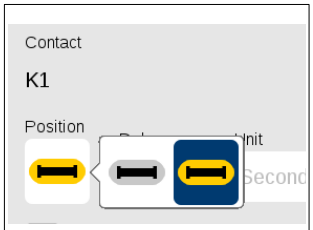

*Fig. 2: Contact options*

The SIB container will have an un-editable set configuration. For Signals (or any appropriate accessory/article), the container will change and evaluate only the stop aspect of the signal or stop track. I even tried this with a turnout, which turned to the red/curved aspect (Fig. 3). You cannot load in a 'go' signal, the container will switch it to 'stop'.

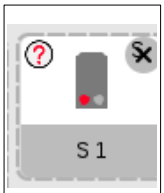

*Fig. 3: Set condition of the 1st container*

If loading a Brake macro, the assumption is that it is stop as well (being a braking macro). Loading an IF macro is more confusing as to how it is supposed to function. Like the others it is un-editable, but the way the IF macro works, it suggests there are two settings with two output results (Fig. 4: The 'on' or 'off' IF container + the 'THEN' or 'ELSE' result). This is a notable contrast to the signal or brake macro container entries.

The FB container is the only container where you can access the optional pull-down menu with two options: Self block, Central block. (Fig. 5)

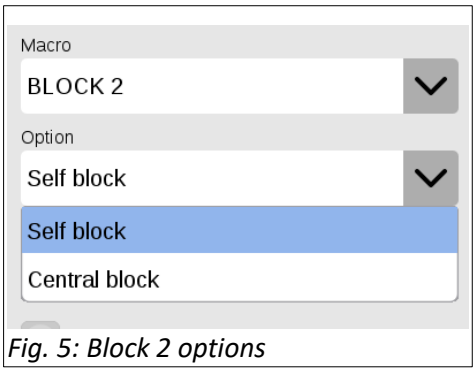

*Fig. 5: Block 2 options*

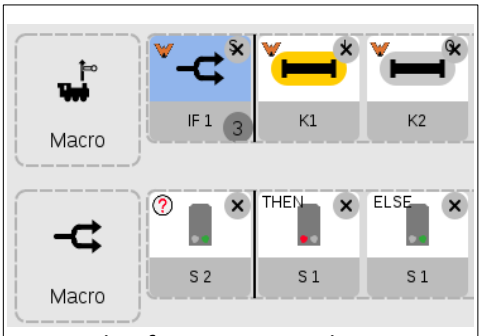

*Fig. 4: The If container can be set one of two ways, each with the same two results*

**Making sense of the Block Macro**

All the containers within the Block Macro are 'evaluative' or 'conditional.' In other words, you cannot set any actions to occur within them (like changing a signal or stop block to go). Unfortunately, the release of the Block Macro has not included any update to the CS3's internal help assistant. The following is some of my procedural thinking on how the BM can be implemented. The First thing I wanted to sort out was where they can be placed.

**Block Macro directly in the event procedures**

A Block Macro can be placed directly into an event line as a step object. In this manner, it will show the same edit options described in the FB container description above. If the Macro proves to be true, then the following steps would act (Fig. 6). The difficulty with this is that each of the 5 containers would need to prove true for the block to be true. If not, then you would need to sort out what container was not true.

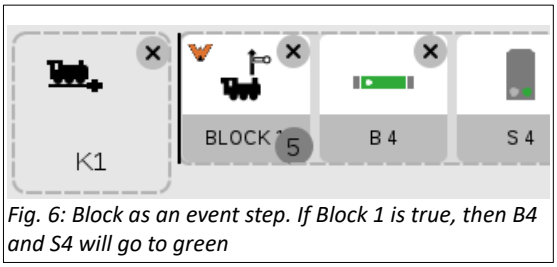

*Fig. 6: Block as an event step. If Block 1 is true, then B4 and S4 will go to green*

To compound the problem, each block macro has a nested block macro (in the FB container). Would each *nested* block also need to prove true? If so, then it is difficult to understand how these are read in. For example: Block 1 would have Block 2 nested, but the nested Block 2 would also have a nested Block 3, 3 would have a nested block 4, etc.

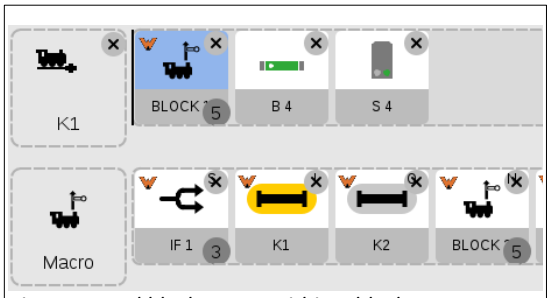

*Fig. 7: Nested block macro within a block macro (bottom right)*

**Block Macro in an IF Macro**

A BM in an IF macro would have the same flaw in evaluation as the event procedure described above (nested blocks. The difference between the two is that an IF macro can offer an alternative action if the Macro was to show as a 'false' instead of 'true' result.

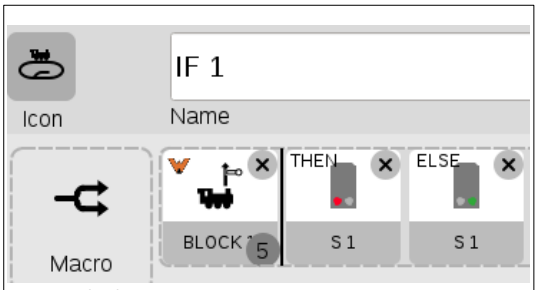

*Fig. 8: Block Macro in an IF Macro container*

The alternative actions are placed in the 'THEN' and 'ELSE' containers of the If Macro. A 'false' would result in the 'ELSE' action, a 'true' would result in the 'THEN' action (Fig. 8). If you have more than one action for each of these containers, you will need to create events with the desired sequence of actions, then place the events (nested) into the 'THEN' and 'ELSE' containers.

**Sorting which elements to use in a Brake Macro**

Once I got a sense of where the BM could be inserted, I tried to understand what happens when I trigger an event with a block macro. My first example is to have a simple action. I used a basic contact section with a single contact which has a preceding signal and stop control. In Fig. 9, the train will be traveling in a left to right direction (track line at top). The 'K1' event I will discuss is also displayed, and the bottom 'Macro' line is the expansion of the BM (in blue).

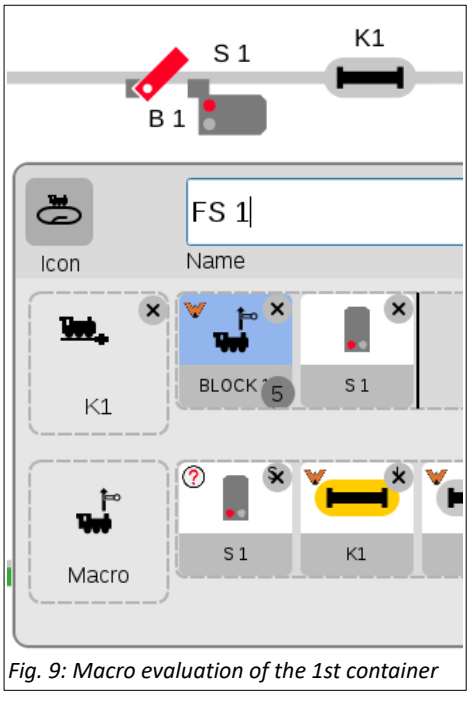

*Fig. 9: Macro evaluation of the 1st container*

For reference, my traditional event step is that when a train contacts contact 'K1' it will set the signal 'S1' to red. This should be familiar with my regular readers, who understand how I build block controls. With the insertion of a Block Macro, the thought process of how this action works becomes more complex. I am going to back track a bit, because before 'K1' can be triggered, the signal 'S1' and 'B1' would be green (the train would need to pass both these elements to reach 'K1', correct?)

**Which contact to trigger the Block-Macro ("BM")**

In this block macro, 'K1' would be triggered and the first step in the BM is to evaluate the condition of 'S1' (asking the CS if 'S1' is red). This is my 1st error, because as you recall, in the BM the first container will only use red as a condition. But to activate the trigger at K1, I would have to have S1 as green. So, let's examine another trigger location (K2 or K4 in Fig. 10).

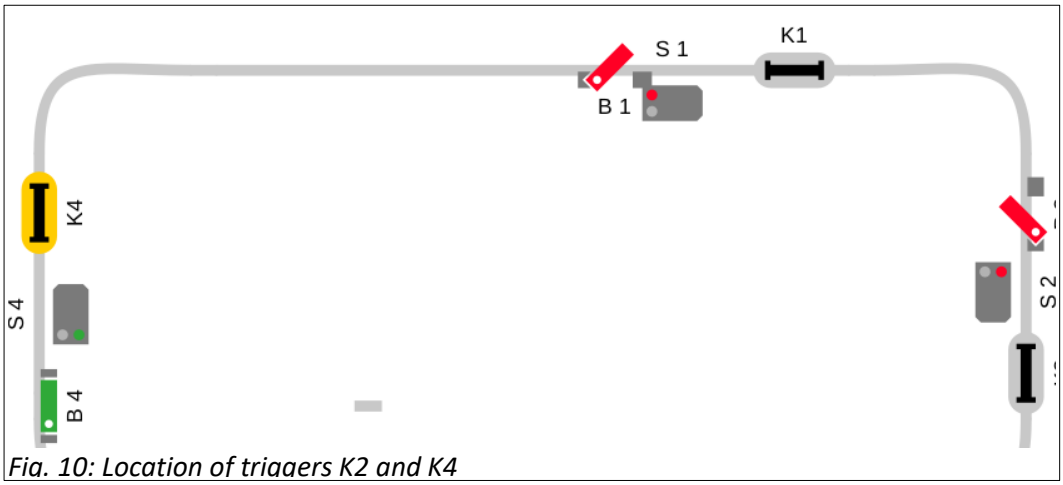


*Fig. 10: Location of triggers K2 and K4*

For the sake of expediency in logic, the contact K2 makes more sense as a trigger. To reason this out, the factors are: The train that contacts K2 may possibly still occupy the block between K1 and K2. This is also the train that should have been the last train to contact K1. K1 should have switched the S1 and B1 to red, which prevents any following train to pass into the occupied block. This sorts out which element needs to be placed in the 1st container of the BM. Fig. 11 resets and displays the full BM elements with K2 as the trigger.

**Examining what Entry and Exit contacts to use**

An entry and exit contact would mark the locations of where a train enters a block and exits a block. The entry and exit contacts can be the same contact for two separate blocks (where one train exits, it also enters the next block).

The use of contacts in the BM cannot be momentary (like magnetic reed or circuit 'slider' contacts), because to read if these types of contacts is currently on/occupied, it would need to happen at the exact same time as the activation trigger that evaluates the readout. In other words, the slider circuits

will only show 'on' if the Slider of the loc has stopped on the circuit track. By contrast, a contact track will stay on as long as the train is in contact (this feature is more practical to Marklin 3-rail).

In the instance of K2 activating the Block macro, K2 would naturally be the Entry contact container. This would then clear the way for the CS to move to the next step in the BM queue, the 'Exit container'.

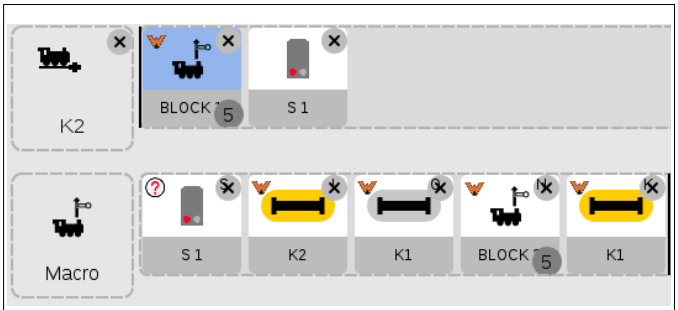

*Fig. 11: K2 trigger, which reads the block 1 macro and sends the S1 action to red*

The Exit container could be the marker at which the train clears the block. I found this harder to define, because I was not sure of how this container may work with the remainder of the containers: Next Block and Pause-flag. In one definition, I would set the exit container as the exit contact from the previous block (K1 of Fig. 10). In a second definition, I would add another contact at the end of block 1 (call it block '1 a'). In the third definition, while 'K2' is an entry marker, it can also double as an 'exit' marker. The question is, which contact is used to show that the block is clear? If K1 were to show as off (true to the setting in Fig. 11), then the system could proceed to the FB container (Block 2 Macro in this example). If K1 was NOT off, then the CS would not proceed to read the FB container. This leads us to another bit of programming speculation. If we examine the final two containers as a working 'whole': the Following Block and Pause-Flag containers, then there appears to be some logic to this arrangement. Within the Following Block edit window there is a Delay setting, which in general, may mean the step will be delayed while a certain condition is still showing. The condition that is displayed in the PF container (K1 is on).

Once all the conditions are met with the Block 1 Macro, then the action steps following BLOCK 1 can take place. This is the S1 to red in the top line of Fig. 11. In the example shown, I was able to place the Block Macro into an appropriately triggered event. I was also able to pin down some of the logic with how it works. I still have questions remaining on the accuracy how the Block Macro is implemented. What is the purpose of having a Following-Block container? What is it reading? The nesting of block macros would seem like it may cause a looping failure when a following block macro is entered into the event. Recall that a Block macro would be entered as a step of another Block Macro. AND, because that nested macro would also have a third Block macro to read, where does this continual nesting end? The purpose of the nested BM is activated as a step in the sequence, but the Following block container seems to act as being triggered based on the Exit and Pause-Fade contacts. Why is this necessary if they could simply be triggered by on track contacts? I kept returning to the idea that the Block Macro can assist with entry and exit points on systems that don't use contact tracks (like two rail systems). These are needed if you need to determine when a train clears a block (train length measuring). However as mentioned, you cannot use a BM to check entry or exit contacts as 'occupied' when the activation of these is momentary.

This has been an initial exploration of the new Block-Macro event. There is some mystery as to what to do on this complex element. It is still unclear if there are benefits to using the Block Macro when compared to much of the documentation that I have offered regarding event programming and block tracking.

Cheers!
Curtis Jeung

## Upcoming appearances:

**National Garden Railway Convention**
June 22nd – 24th, 2022 (Dates we will be in attendance in the Vendor Hall)
Denver, Colorado

**Eurowest**
July 16th – 17th, 2022
San Carlos, California

**NMRA National Train Show**
August 12th – 14th 2022
Collinsville, Illinois

## Next Märklin Digital Webinar:

**Wednesday, June 8th**
11:00 AM Pacific / 12:00 noon Mountain / 1:00 PM Central / 2:00 PM Eastern

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

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

**E-mail: digital@marklin.com**