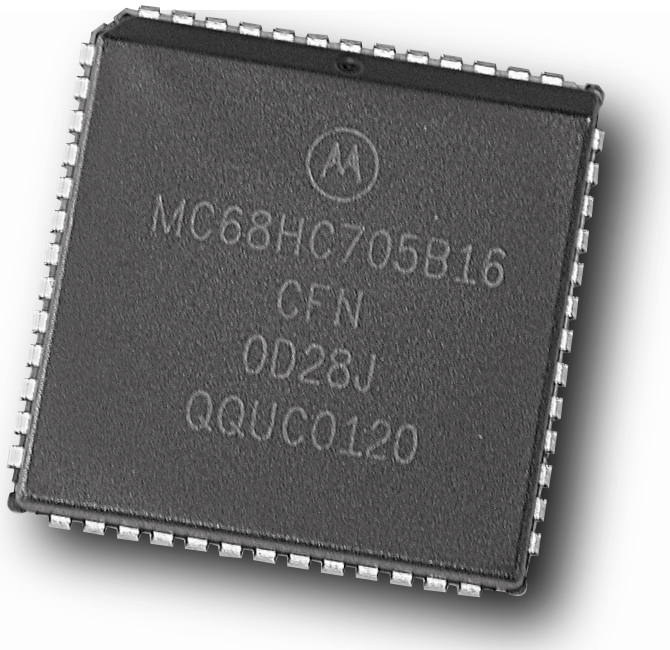
EEDTS Pro Control Unit Protocol

protocol extensions

By S. van de Vries



Without question, the EEDTS Pro protocol has been the subject of most discussions related to EEDTS Pro, due to the fact that it differs from the old EEDTS and the Motorola protocol.

Furthermore, many users and software suppliers have found it difficult to convert addresses and date to the trits coding required by the EDTS controller.

Consequently, the protocol has been extended to include the command set listed in **Table 1**.

Naturally, the existing command set (as described in the EEDTS Pro book) is still fully supported, and the new control unit is downwards compatible with the old version.

The Märklin protocol formed the starting point for this move, but since that protocol does not have any commands for the extra functions, an extra byte has been specifically added (in particular, for the locomotive control commands).

General command structure

The EEDTS Pro control unit communicates with the PC via the RS232 port using fixed settings of 9600 baud with no parity, eight data bits and one stop bit. In order to ensure correct data transfers, a design has been used in which a response byte is sent back for every byte that is sent to the control unit.

A control unit instruction may consists of 1, 2, 3 or 4 bytes, with the first byte being the command byte.

In general, the command byte is sent back *verbatim* as an indication that the instruction was correctly received, except in the case of the return commands (single-byte control unit instructions), for which return information is sent back directly in order to achieve a high data rate.

If the control unit cannot send the information to the track (for instance, if the booster is out of service due to a short circuit), the value '65' is sent back following the command byte, in order to inform the PC of this situation.

In order to inform the control unit in turn that the booster is out of service, a small modification to the booster interface is necessary. This will be described in the following instalment, along with the addressing of the keyboard and stand-alone controller.

Locomotive command

The first byte of the locomotive command can have a value of 0–15. This value represents the speed, bearing in mind that in the old format, speed level '1' is the 'reverse' command (in the Märklin protocol, it is '15').

The second byte that is sent represents the locomotive address. The first 80 addresses correspond to the 80 available addresses in the Märklin format. However, since there is room for 256 addresses in the new format, the command has been extended to 256 addresses.

The only existing 'standard' for this sequence is to be found at Uhlenbock, which is why this sequence has been chosen.

The third byte provides information regarding the format, direction of travel and extra functions. The meanings of the individual bits in the old format are shown in **Table 2**, while the same information for the new format is shown in **Table 3**. For convenience, the values are also shown in decimal notation.

It is now possible to generate old-format functions. An important difference between the two tables is that in the old format, all four functions are modified by each byte, while in the 'new style' table a separate byte is used to set each function.

In the old format, if it is not desired to modify the extra func-

Table 1. The extended command set.

	Command byte	Normal response	Short circuit	Second byte	Response	Third byte	Response	Fourth byte	Response
Locomotive command	0 - 15	0 - 15	65	0 - 255	0 - 255	See Tables 2 & 3	Third byte		
Last changed	190	1..64	65						
Return-signalling module	192 - 255	0 - 255	65						
Turnout (switch) reset	32	32	65						
Turnout (switch) to the left	33	33	65	0 - 255	0 - 255				
Turnout (switch) to the right	34	34	65	0 - 255	0 - 255				
Program manual controller	40	40	65	16 - 23 24 - 31	0 - 255 Speed	0 - 255	Previous address	0 - 255	Previous status
Read manual controller	41	41	65	16 - 23 24 - 31	0 - 255 Speed	0	Address	0	Status

tions, a '0' or a '3' is sent (the 'n' in the table indicates that F1–F4 do not change). The new format also includes the 'reverse' and 'forward' commands.

The observant reader may have already noticed that two bits (b1 and b2) are reserved for the F0 function and that these bits always have the same values in the table ('00' or '01'). The values '01' and '10' are also allowed, but they are reserved for assignment to future functions.

Return commands

Last changed

The 'last changed' command ('190') will doubtless prove to be useful for real-time control in layouts having a large number of return-signalling units. The response byte contains the address of the return signaller whose input state has changed (with respect to the time at which this unit was last read). This makes it possible to quickly determine which unit must be read.

Return-signalling modules

The return-signalling units can be read using commands 192–255.

Command '192' is used to read the first unit, '193' is used to read the second unit and so on, up to '255' for the 64th EEDTS return signalling unit (or 9–16 of the 32nd Märklin S88).

The value that the control unit sends back provides a binary representation of the inputs (in the case of a detector module, this is the locomotive address).

Turnout (switch) commands

The turnout (switch) commands are exactly the same as the Märklin commands. For instance, command '33-(1–4)' activates the 'green' output of a k73, while command '34-1–4)' activates the 'red' output. Even the peculiarity that '33-0' and '34-0' represent the final two outputs of decoder 64 has been implemented. Turnout activation is rescinded using command '32'.

These commands limit the number of decoders to 64. If a larger number of decoders must be supported, recourse can be made to

Table 2. The old Motorola format.

b8	b7	b6	b5	b4	b3	b2	b1	Decimal	F0	F1	F2	F3	F4
0	0	0	0	0	0	0	0	0	0	n	n	n	n
0	0	0	0	0	0	1	1	3	1	n	n	n	n
0	1	0	0	0	1	0	0	68	0	0	0	0	0
0	1	0	0	0	1	1	1	71	1	0	0	0	0
0	1	0	0	1	0	0	0	72	0	1	0	0	0
0	1	0	0	1	0	1	1	75	1	1	0	0	0
0	1	0	0	1	1	0	0	76	0	1	1	0	0
0	1	0	0	1	1	1	1	79	1	1	1	0	0
0	1	0	1	0	0	0	0	80	0	0	0	1	0
0	1	0	1	0	0	1	1	83	1	0	0	1	0
0	1	0	1	0	1	0	0	84	0	1	0	1	0
0	1	0	1	0	1	1	1	87	1	1	0	1	0
0	1	0	1	1	0	0	0	88	0	1	1	1	0
0	1	0	1	1	0	1	1	91	1	1	1	1	0
0	1	0	1	1	1	0	0	92	0	0	0	0	1
0	1	0	1	1	1	1	1	95	1	0	0	0	1
0	1	1	0	0	0	0	0	96	0	1	0	0	1
0	1	1	0	0	0	1	1	99	1	1	0	0	1
0	1	1	0	0	1	0	0	100	0	0	1	0	1
0	1	1	0	0	1	1	1	103	1	0	1	0	1
0	1	1	0	1	0	0	0	104	0	1	1	0	1
0	1	1	0	1	0	1	1	107	1	1	1	0	1
0	1	1	0	0	1	0	0	108	0	0	0	1	1
0	1	1	0	0	1	1	1	111	1	0	0	1	1
0	1	1	0	1	1	0	0	112	0	1	0	1	1
0	1	1	0	1	1	1	1	115	1	1	0	1	1
0	1	1	1	1	1	0	0	116	0	1	1	1	1
0	1	1	1	1	1	1	1	119	1	1	1	1	1

the normal EEDTS command set, which can address up to 240 decoders.

Programming manual controllers

Program manual controller

Command '40' can be used to program the manual controllers. The command byte ('40') is followed by a second byte that can have a value of 16–23. The value '16' selects controller 1, '17' selects controller 2 and so on, up to '23' for controller 8.

If a value in the range of 16–23 is chosen, the address modification is temporary and is no longer valid after a power-down. In order to achieve a permanent address modification, a value in the range of 24–31 must be chosen ('21' for controller 1 etc., up to '31' for controller 8).

Once the control unit has received the second byte, it will send back the controller status. This can lie in the range of '0' (control rotated fully to the left) to '255' (control rotated fully to the right).

The third byte determines the address that must be set for the controller, while the fourth byte indicates the format of the controller and the functions that are already enabled or disabled in the locomotive (or in the PC software). In this way, for example, the control unit knows that if the F3 function is enabled, it must send the 'F0 off' command to the locomotive when the F0 button is pressed on the manual controller.

The bits in the fourth byte have the following meanings:

b8	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**
s/h	o/n	f/r	F4	F3	F2	F1	F0

For example, if the fourth byte is assigned a value of '36' (binary '00100100'), the controller settings are as follows:

s/h = 0 format selection:
  1 = format set by software
  0 = format set by hardware
o/n = 0 format:
  1 = old format
  0 = new format
f/r = 0 direction of travel:
  1 = forward
  0 = reverse
F4 = 0 (F4 off)
F3 = 0 (F3 off)
F2 = 1 (F2 on)
F1 = 0 (F1 off)
F0 = 0 (F0 off)

If the s/h bit is '1', this means that the format (old or new) is not determined by the selection diode in the manual controller, but instead by the o/n bit. If the s/h bit is '0', the format is determined by the selection diode in the manual controller.

After the third (address) byte has been sent to the control unit, the control unit sends back the address set for the manual controller (at the time that the command was sent). Following the fourth byte, the control unit sends back the previously read controller status in the response byte. The response data can be used by the EEDTS Pro software, for example, to restore the original values when control is passed from a software controller to a manual controller.

The eight manual controllers cannot be disabled and will continually send information to the track. You should therefore avoid having more than one controller set to the same address (except for address '0') or having the PC send commands to addresses that are being used by manual controllers.

In order to avoid problems arising from the manual controllers, it is a good idea to first set all manual controllers (or all manual controllers that are not actively in use) to address '0' when starting up your own program.

Table 3. The new Motorola format.

b8	b7	b6	b5	b4	b3	b2	b1	Decimal	Description	
0	0	0	0	0	1	0	0	4	F1 off	F0 off
0	0	0	0	0	1	1	1	7	F1 off	F0 on
0	0	0	0	1	0	0	0	8	F1 on	F0 off
0	0	0	0	1	0	1	1	11	F1 on	F0 on
0	0	0	0	1	1	0	0	12	F2 off	F0 off
0	0	0	0	1	1	1	1	15	F2 off	F0 on
0	0	0	1	0	0	0	0	16	F2 on	F0 off
0	0	0	1	0	0	1	1	19	F2 on	F0 on
0	0	0	1	0	1	0	0	20	F3 off	F0 off
0	0	0	1	0	1	1	1	23	F3 off	F0 on
0	0	0	1	1	0	0	0	24	F3 on	F0 off
0	0	0	1	1	0	1	1	27	F3 on	F0 on
0	0	0	1	1	1	0	0	28	F4 off	F0 off
0	0	0	1	1	1	1	1	31	F4 off	F0 on
0	0	1	0	0	0	0	0	32	F4 on	F0 off
0	0	1	0	0	0	1	1	35	F4 on	F0 on
0	0	1	0	0	1	0	0	36	Reverse	F0 off
0	0	1	0	0	1	1	1	39	Reverse	F0 on
0	0	1	0	1	0	0	0	40	Forward	F0 off
0	0	1	0	1	0	1	1	43	Forward	F0 on

Read manual controller

Command '41', which is used to read out a manual controller, is closely related to the 'program manual controller' command. However, the 'read manual controller' command only reads data and does not change any controller settings. In this case as well, controller status is returned following the second byte.

The values following the third and fourth bytes are the same as for the 'program manual controller' command, with the understanding that here it does not matter what value is sent by the PC, since the control unit does not do anything with these values.

This completes our presentation of the most important changes in the new control unit protocol. The new microcontroller is available from Readers Services under order number **010088-41** and can simply be fitted on the existing control unit circuit board.

(010088-2)