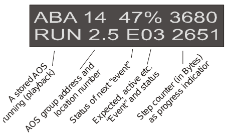
AOS-specific information displayed by the command station:

Depending on the CV #101 setting (see below), useful information during playback of an operating sequence is shown by the MX1 display either temporarily (about 20 sec, default) or permanently (CV #101, Bit 0 =1).


ABA 14 47% 3680
RUN 2.5 E03 2651
A stored AOS running (playback)
AOS group address and location number
Status of next "event"
Expected, active etc. "Event" and status
Step counter (in Bytes) as progress indicator

The upper line shows common data about the AOS memory (as during a recording); the lower line is all about the "event" handling: it shows whether the timing during playback matches the sample run and whether the "events" are being recognized properly. For this purpose, the "events" are shown with prefixes:

N... = This is the next expected "event"; the playback of stored commands is being executed as planned (that is according to the recorded time stamps).

W...= The "event" is imminent that is no more commands are to be executed ahead of this "event". If the train is late (compared to the sample run), the playback is delayed until the "event" is triggered in order to re-synchronize the sequence.

S... = The "event" was triggered earlier than expected (the train was faster than during the sample run); the playback speed is increased to re-synchronize the sequence; that is the commands are processed faster.

B...= The next "event" is locked because it is also the past "event", and multiple triggers by axles of the train are expected (see "IMPORTANT…" on the previous page).

E... = "Event" is being triggered.

X00 = This operating sequence contains no "events" (no synchronization possible).

MX1 – Configuration variables for Automatic Operating Sequences (AOS):

These parameters are absolutely valid for MX2 and MX21 cabs; with an MX31 cab and future software version an individual setting for every operating sequence is possible. See MX31 manual!

CV #100: Simulation of overdue "events". In case a stored "event" has not been triggered, the sequence is aborted after 1 minute by default (CV #100 = 0); most often though an accident will happen before the 1 minute has elapsed (collision at the end of the run or similar). With

CV #100, Bits 0 – 5 = 0….63 a time is defined after which a reaction takes place if an "event" is overdue, namely

when Bit 7, 6 = 00 the operating sequence is being aborted;

when Bit 7, 6 = 01 (decimal + 64) the "event" is being simulated.

For example: If CV #100 = 69, the overdue "event" (compared to the recorded "event") is generated artificially after 5 seconds and the operating sequence will continue.

CV #101: pertains to the accompanying information displayed by the MX1 when executing a stored operating sequence; by default the information is displayed only briefly and thereafter switches back to the normal MX1 volt and amp meter display. This does not affect the display during a sample run recording, which is always displayed permanently:

Bit 0 = 0: Display is used "normally" (Volt and amp meter ….), not for AOS
Bit 0 = 1: The last activated sequence is continuously being displayed and updated.
Bit 1 = 0: Unlock after next event triggered or with change in direction (or unlock switch)
Bit 1 = 1: Unlock after next event triggered and change in direction or after two triggered events (or unlock switch)

17. The "old" (ASCII-oriented) Interface

This section is only of interest to users that want to operate the layout with an external computer but don't want to use the ready made software "STP".

The serial interface of the MX1 command station allows the control of trains and accessories by a computer as well as the polling of current route and switch data from the computer.

The interface at the MX1 is a 9-pin Sub-D socket; the connection with a computer is done with a 1:1 serial cable.

The following described protocol was adopted from the previous product of the MX1 "model 2000" command station described here and ensures compatibility to already existing applications. A more powerful (alternative) protocol is in preparation (binary instead of ASCII).

Data transmission parameters:

9600 bit /s (default) - 8 bit - no parity - number of stop-bits 1 or 2.
(higher speed selectable with CV #12).

Computer to MX1 command format:

COMMAND ID -	INFORMATION BYTES	- CARRIAGE RETURN
(defines the command)	(command data)	(always last byte of a command)

The COMMAND ID is an ASCII letter (i.e. S, F, K etc.), which identifies the kind of command and also determines the meaning of the information byte that follows.

With the command ID "S" only one INFORMATION BYTE follows in the form of another ASCII letter (S, A, E).

All other command ID's can be followed by a row of INFORMATION BYTES, which are however not coded as binary but rather as two ASCII characters that represent the hexadecimal value (function "HEX$" in BASIC).

At the end of each command, a CARRIAGE RETURN code must be present ("CHR$(13)" in BASIC).

Example of opening the interface and sending a loco command in BASIC:

```
OPEN "COM1:9600,N,8,1,CS,DS,CD" FOR OUTPUT AS #2
 PRINT#2, "F"; "N"; HEX$(Loco address); HEX$(Speed step);
                   HEX$(Data byte1); HEX$(Data byte1); CHR$(13);
```

"F" is the command ID for loco commands; "Loco address" and "Data byte" are in decimal form (i.e. loco address = 123, the data byte is a combination of speed and functions, see below); CHR$(13) is the concluding carriage return.

ATTENTION: A leading "0" must be entered with values of <16!

All information bytes (loco address, speed step, data bytes etc. as described in the following commands) must be sent to the MX1 command station in the form of two ASCII characters. The "HEX$" function however sends in many BASIC versions only one character for values smaller than 16 (hexadecimal characters "0" to "F"). A "0" must be added as a leading character in such cases; if possible with corresponding BASIC commands or by defining a special function that always produces two characters.

Note for the creation of commands in other than a BASIC environment:

Sending the respective command ID (ASCII) does not present a problem. Coding of information bytes is as follows: a loco address of "123" for example corresponds to the hexadecimal figure "7B", which is being sent as the two ASCII characters "7" and "B" (only capital letters are allowed as hexadecimal characters!). The same applies to data bytes: for example, the "L function ON" and the speed step 14 in a loco command result in a hexadecimal figures "1" and "E", which are send as ASCII characters. ATTENTION: always two ASCII characters per information byte (add a leading zero if necessary).

Description of individual command types:

NOTE: In case of a MX1 command station sending out several data formats (DCC, MOTOROLA), the respective loco or accessory address must contain a data format prefix:
"N" for DCC (for ZIMO and other NMRA-DCC compatible decoders) / **"M"** for MOTOROLA (Märklin decoder).

Broadcast stop and track power OFF/ON commands: **Command ID "S"**

The entire command consists of only the command ID "S" and another letter, which represents its content, as well as the obligatory carriage return:

"SS"; CHR$(13) - Broadcast stop ON (stops all trains).

"SA"; CHR$(13) - Track power OFF.

"SE"; CHR$(13) - Track power and broadcast stop OFF (normal operation commences).

Loco command: **Command ID "F"**

Besides the command ID "F", the loco command consists of the appropriate loco address and the data bytes, which contain the information about speed, direction and functions.

"F"; data format prefix ("N","M") HEX$(loco address);
HEX$(speed step); HEX$(data byte 1), HEX$ (data byte 2),
HEX$ (data byte 3), CHR$(13)

Data format prefix-	only "N", "M" allowed, see note above.	
Loco address -	decimal value (1 to 255); use HEX$(0) with single digit values	
Speed step -	Speed step of the currently active speed step systems	
Data byte 1 -	Bit 7	MAN-Bit
	Bit 6	
	Bit 5	Direction (0 = forward, 1 = reverse)
	Bit 4	Headlights ON/OFF (= DCC-Function 0)
	Bit 3/2	DCC – speed step system (01 = 14 speed steps, 10 = 28 steps, 11 = 127 steps;
	Bit 1	Deceleration time "BZ" (definable with command "B") ON/OFF
	Bit 0	Acceleration time "AZ" (definable with command "A") ON/OFF
Data byte 2 -	Bit 0-7	Functions 1-8 ON/OFF
Data byte 3 -	Bit 0-3	Functions 9-12 ON/OFF

How the decimal value of the data byte that is entered to the above HEX$-function is calculated, explains this example: "MAN" turned OFF, lights ON, direction reverse, speed step system "0-28", "AZ" and "BZ" OFF results in the binary word "00111000", which in turn is the equivalent to the decimal value of 0 x 128 + 0 x 64 + 1 x 32 + 1 x 16 + 1 x 8 + 0 x 4 + 0 x 2 + 0 x 1 = 56.

Function inversion command: **Command ID "U"**

This command allows inversion of direction and/or function outputs; this is sometimes useful in computer applications.

"U"; Address prefix, HEX$(loco address); HEX$(data byte 1); HEX$(data byte 2); CHR$(13)

Data format prefix - only "N" and "M" or "Z" are allowed

Loco address - decimal value (1 to 255),

Data byte 1 - bit assignment like command "F", inversion with"1" (except 3/2)

Data byte 2 - bit assignment like command "F", in version with"1"

Data byte 3 - bit assignment like command "F", in version with"1"

Acceleration/deceleration (AZ/BZ) - command: **Command ID "B"**

Adjustment of "AZ" and "BZ", which are the acceleration and deceleration times controlled by the MX1 command station. Not to be confused with the configuration variables #3 and #4 in the decoders! "AZ" and "BZ" is independent of these configuration variables and it can also be used for non-DCC decoders. Note that "AZ" and "BZ" is turned ON and OFF with the "U" command.

"B"; data format prefix; HEX$(loco address); HEX$(data byte); CHR$(13)

Data format prefix - only "N" and "M" or "Z" are allowed, see note above.
Loco address - decimal value (1 to 255)

Data byte -	Bit 0-3	BZ (0 - 15)
	Bit 4-7	AZ (0 - 15)

Shuttle train command: **Command ID "P"**

This command allocates particular locomotives to the shuttle train inputs 1 to 4 of the command station (see the cab operating manual for details on shuttle train operations).

"P"; data format prefix; HEX$(loco address); HEX$(data byte); CHR$(13)

Data format prefix - only "N" and "M" or "Z" are allowed, see note above.
Loco address - decimal value (1 to 255)

Data byte -	Bit 0-3	Shuttle train inputs 1-4, forward
	Bit 4-7	Shuttle train inputs 1-4, reverse

Query of extend address indexing **Command ID "E"**

Locos with extended addresses (128 - 10239) are accessed with the usual commands listed above, but instead of the address itself, an automatically assigned internal index address (128 - 255) is used; which first needs to be ascertained with the help of the "E" query:

“E”; HEX$(High-byte of the loco address); HEX$(Low-byte); CHR$(13)

The command station then supplies an index message; such a message is delivered for addresses >128 after each loco command to validate their assignment (unused loco addresses can lose their initial index address if required).

Commands for accessory DECODERS: Command ID “M”

An accessory command is similar in its structure as a loco command; the data byte contains the actuation information of 4 (in the DCC and MOTOROLA format) or 8 (in the ZIMO data format) turnouts or signals.

“M”; data format prefix; HEX$(Accessory decoder address); HEX$(Data byte); CHR$(13)

Data format prefix - only “N” and “M” or “Z” are allowed
Accessory decoder address - decimal value (0 to 255); for MOTOROLA only to 63.

Data byte (DCC-data format)	Bit 7	9. Address bit (Address range 0 to 510 !)
(The same for MOTOROLA-	Bit 3	Switches ON or OFF (1=ON, 0=OFF)
format but without Bit 7)	Bit 2-0	Number of the output to be switched

NOTE: If a switch command cannot be executed because another accessory command is currently being processed, the MX1 answers with

“???; CHR$(13)”

The switch command must be repeated (in intervals of about 500 msec) until the negative answer is no longer returned.

Commands for accessory MODULES: Command ID “N”

In contrast to the loco and accessory DECODER’s, the switch commands here are combined with the command ID “N”. There are therefore two implementations of the command, depending on the “switch type”.

“N”; HEX$(Module address); HEX$(Data byte); CHR$(13)

Module address -		800 to 863
Data byte as command -	Bit 0-5	Number of the MX8 output
	Bit 6	Position (0 = left, 1 = right)
	Bit 7	= 1 (ID for command byte)
Data byte for query	Bit 0-7	= 0 (ID for query)

Status query: Command ID “Z”

The command station is queried about the track voltage, broadcast stop and the available data formats (DCC, MOTOROLA, ZIMO).

“Z”; CHR$(13)

Query the command station memory: Command ID “A”

With this command the command station delivers information about current loco and accessory data for a specific locomotive or accessory address.

“A”; F/W selection; Data format prefix; HEX$(Loco or accessory address); CHR$(13)

F/W selection - “F” = Loco address, “W” = Accessory address
Accessory address - if the DCC data format is used, a second data byte must be sent, of which only Bit 7 as the highest DCC address is being evaluated (address range up to 510).

Commands in conjunction with routes: Command ID “W”

These commands relate to routes (turnout ladders) that are defined and called up with the cab (see cab manual). These route definitions are stored in the MX1 command station; the route memory can be read out, re-installed or deleted via the MX1 computer interface. This allows the route definitions to be saved in an external computer and if needed to be restored again (i.e. after loss of the MX1 memory or for transfer to another command station).

“WCLEAR”; CHR$(13) - Delete all routes.
“WO”; CHR$(13) - Send all stored route definitions to the computer.
“WI”; CHR$(13) - Reload routes from the computer.

If the data structure of the “WI” command is known (will be added later), it will be possible to define routes in the computer and send them to the command station.

Commands in conjunction with lines: Command ID “O”

These commands relate to lines (block control) that are defined and called up with the cab (see cab manual). These line definitions are stored in the MX1 command station; the memory can be read out, re-installed or deleted via the MX1 computer interface. This allows the line definitions to be saved in an external computer and if needed to be restored again (i.e. after loss of the MX1 memory or for transfer to another command station).

“OCLEAR”; CHR$(13) - Delete all lines.
“OO”; CHR$(13) - Send all stored line definitions to the computer.
“”; CHR$(13) - Reload lines from the computer.

If the data structure of the “” command is known (will be added later), it will be possible to define lines in the computer and send them to the command station.

Register-Programming Command ID “R”

For addressing and CV programming of all NMRA-DCC compatible loco decoders (ZIMO and others) and for programmable decoders with the MOTOROLA format (Uhlenbrock decoder); the address is stored in configuration variable (register) 1. The programming is performed on the programming track.

“R”; Data format prefix (“N”or “M”); HEX$(Register number); HEX$(new value); CHR$(13)
“RE” – Terminating the programming mode

The “R” command (same as the “Q”) triggers a register message that serves as acknowledgment!

Register polling (for DCC data format only!) Command ID “Q”

This command prompts the command station to send a register message, which is used for reading out configuration variables of NMRA-DCC compatible loco decoders.

“Q”; HEX$(Register number); CHR$(13)

Programming-on-the-main (for DCC data format only!) Command ID “L”

“Programming on-the-fly” (or “programming-on-the-main”).

“L”; HEX$(Loco address); HEX$(Command bytes); NMRA-Instruction; CHR$(13)

Loco address -	decimal value (1 to 127)	
Data byte -	Bit 0-2	Number of bytes of the NMRA telegram (excl. ECC)
	Bit 3-5	Number of desired telegrams
	Bit 7	= 1: Address is locked out of the normal send cycle. = 0: Address is controlled normally after the desired number of telegrams have been sent.

NMRA-Instruction (up to 5 Bytes) - “On-the-main” – Command according to NMRA - RP 9.2.1

Special command:
Laa00 unlocks address aa without sending new data.

Message formats from MX1 to computer:

Command station status (“Z” query):

“Z”; HEX$(status byte); CHR$(13)

Status byte - Bit 7	MX1-Generation (0 = “old”, 1 = “new”)	
	Bit 6	1 = ZIMO-Data format active (Type and jumper)
	Bit 5	1 = DCC- Data format active (Type and jumper)
	Bit 4	1 = MOTOROLA- Data format active (Type and jumper)
	Bit 0	Broadcast stop (1 = ON, 0 = OFF)
	Bit 1	Track power (1 = ON, 0 = OFF)
	Bit 2	UES-Short circuit protection (1 = ON, 0 = UES OFF)

Memory report (after “A” query):

“A”; F/W-Selection; data format prefix; HEX$(Loco or accessory address);
HEX$(speed step); HEX$(Data byte 1); HEX$(Data byte 2);
HEX$(Data byte 3); HEX$(AZ/BZ); HEX$(Group byte); CHR$(13)

F/W-Selection “F” = Loco address, “W” = Accessory address
Data format prefix - only “N”, “M” allowed
F...- or M...addresses decimal value (1 to 255);
additional Byte for DCC data format with
Bit 7: 9. address byte; Bit 0: 0 = paired bits, 1 = single bits)
Speed step or accessory position – as in “F” or “M” command
Data byte 1 - Format same as data byte 1 in “F” command
Data byte 2 - Format same as data byte 2 in “F” command
AZ/BZ-Values - Format same as data byte in “B” command
Data byte 1 - Bit allocation same as in “F” command
Data byte 2 - Bit allocation same as in “F” command
Data byte 3 - Bit allocation same as in “F” command
or
Group byte - Bit 0 Cab activity (0 = not active, 1 = active, address in display)
Bit 2 Loco decoder feedback (1 = received)

Register report (after “Q” query):

“Q”, HEX$(Error code), HEX$(Register number); HEX$(current value); CHR$(13)

Error code - 0 = successful read-out of register number

Index report (after “E” query or loco command for extended address):

“E”, HEX$(Index addressee), HEX$(loco address high byte); HEX$(low byte); CHR$(13)

Error code - 1 = “Old” ZIMO data format is active (jumper).
3 = Address is < 128 (not extended address)
5 = Index address is not assigned to a loco address.

Error messages in form of an index address (Index address < 128 is an error message!).
0 = no data block available.
1 = “Old” ZIMO data format is active (Jumper).
3 = Address is < 128 (must be used as normal address).

18. The “new” binary Interface Protocol

The binary protocol is available with SW versions 3.xx (2007). Its main purpose is to work together with software products like “TrainController” from Railroad&Co, but can of course also be used with other software.
Note: The “ZIMO geared” layout programs STP and ESTWGJ don’t use this protocol because they are connected directly to the CAN bus, instead of the RS-232 interface.

Description of the binary protocol 3 is covered in a separate document and can be downloaded from the “manuals” pages at www.zimo.at !