

9 Addendum

9.1 Changes in STP V5.2

9.1.1 General Notes

After some delay a new version of STP is now available. As with previous updates, a majority of the new functions have been included on demand and based on proposals made by customers.

The main focus of this version is on the area of routes, especially the new concept of **variables** brings lots of flexibility and variety to the operation of your layout.

In addition the introduction of the new Zimo central unit **MX10** is a key aspect of STP V5.2. As special feature there is no extra CAN adapter in the PC required as the MX10 forwards CAN bus data using an USB or Ethernet connection.

9.1.2 Installation of the Update

9.1.2.1 Setup operation

The handling of the setup is in general identical to the previous versions, only the design was updated.

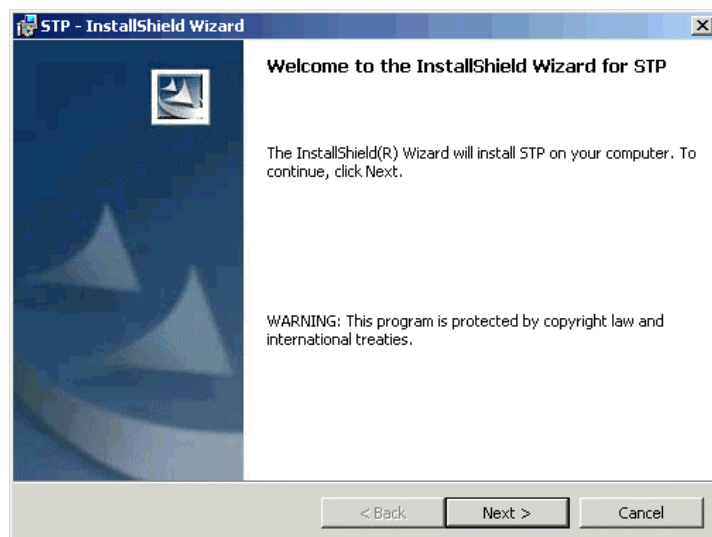


Fig. 1: STP Setup Wizard

Starting with Windows Vista Administrator privileges are required for the installation (especially for installation of the STP CANKey).

The current version of STP supports all current Windows operating systems (**XP, Vista, Windows 7**) in their 32- and 64 bit versions. Older Windows versions may work but have not been tested. A quick test on **Windows 8** has also been made and STP worked properly, when using a CANKey with Windows 8 additional steps may be required. Windows RT (for tablets based on ARM CPU) is currently not supported.

The installation of the **STP CANKey** is explained in the manual which is included in its delivery. Please note that in case the CANKey is used it has to be installed before the first start of STP. The required driver software is available on the STP installation CD. When using the Zimo MX10 base unit no CANKey is required, as the connection to the CAN bus is handled by this command station. In this case no driver have to be installed as they are included in the Windows operating system by default.

Note: If STP is to be installed in a new directory (i.e. not the one of the previous STP installation), by default the **ZIMO MX10** is selected as CAN interface in the **Global Interface Setup** dialog box (see "Zimo MX10" on page 3). By doing so, STP can also be started when the command station is turned off. In case the same directory as for the previous version is selected for installation, all settings will remain unchanged.

9.1.2.2 Registration of a Full License Version (only when using a Zimo MX10 Base Unit)

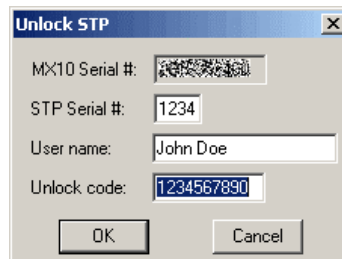
The verification check if an STP full version is licensed for a specific customer previously has been done using the STP CANKey (or for still older versions using the STP Program Key on the parallel printer port), i.e. STP could not be started without an attached CANKey.

Because no CANKey is required when using the MX10 base unit (but it can be used anyway, see above), now there is a new way of license verification which uses the (unique) serial number of the MX10 central station.

When STP V5.2 is started and an MX10 is configured as CAN interface (see "Zimo MX10" on page 3), the program can be used even if the central station is turned off. The **Online** mode (i.e. the mode where the layout is actually controlled by STP) is limited to six minutes, afterwards the program terminates.

To unlock the program for an unlimited Online mode, proceed as follows:

1. Turn the MX10 on and connect it to the computer
2. Start STP and check if the MX10 is set up as CAN interface in the **Global Interface Setup** dialog box using the correct parameters (see "Zimo MX10" on page 3)
3. Select the menu entry **Register...** in the STP **Help** menu



4. Note the number shown under **MX10 Serial #** (in case **N/A** is shown there, please wait a moment as the serial number is currently requested from the MX10).

- Send this serial number together with your name and the STP serial number (shown on the STP installation CD) to this e-mail address: info@stp-software.at
Then close the windows above by clicking at **Cancel**.
- Within one or two days you should get the remaining data which have to be entered in the window above via e-mail. Then close the window by clicking at **OK** to save the data permanently.

9.1.3 Additionally Supported Hardware

9.1.3.1 Zimo MX10

At the time of writing of this document the ethernet interface on the MX10 was not yet operational.

If STP is installed into a new directory, the MX10 will be selected by default as CAN bus interface. Communication with the PC can be done via Ethernet or USB interface, in the latter case a "virtual" serial interface (e.g. COM5) will be created on the PC when the central station is connected for the first time.

In the **Global Interface Setup** of STP (in the **Configuration** menu) you have to define to which interface the MX10 is connected.

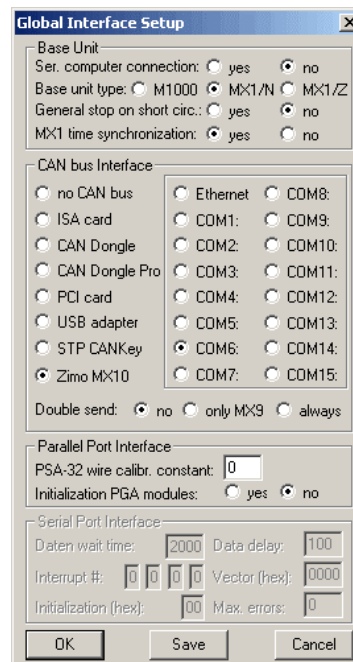
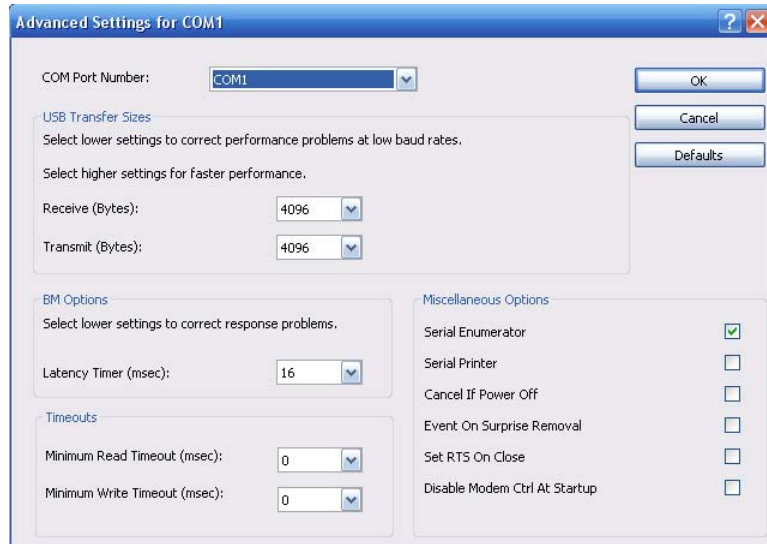


Fig. 2: Global Interface Setup

If the MX10 is connected via USB, a COM port has to be selected here to which the MX10 is attached to (COM1 – COM15). To find out which COM port the MX10 is using, proceed as follows:

- Turn the MX10 on and connect it to an unused USB port of the PC using a proper USB cable
- Open the Windows Device Manager (Windows XP: Start / Performance and Maintenance / System icon / System property window, Hardware tab / Device Manager, Windows Vista / 7: Start / Control Panel / System and Security / Device Manager (under System heading))
- Click at the **+** sign at the left next to **Ports (COM & LPT)**
- Now all serial ports of the PC are shown together with their port numbers (in brackets).

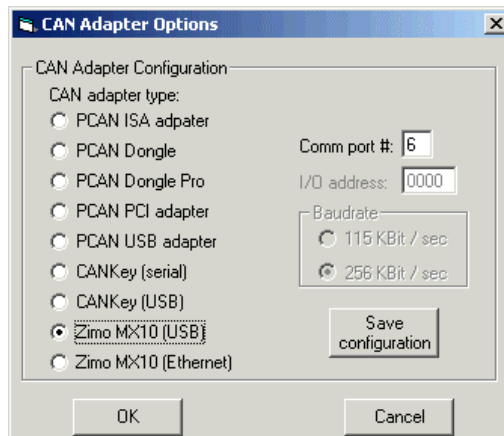
5. Look in this list for the port which corresponds to the one the MX10 is attached to (if multiple ports are shown, unplug an re-plug the USB cable of the MX10, the port which disappears for a short period is the one we are looking for); if the COM port number is in the region COM1 – COM15, we are done – this is the COM port which has to be selected in the **Global Interface Setup** dialog box (close the **Global Interface Setup** window afterwards using the **Save** button!)
6. If the COM port number is larger than 15, **right-click** at the **USB Serial Port** entry of the MX10
7. In the menu which now will appear select **Properties**
8. Go to **Port Settings** and click at **Advanceded...**



9. You can now see the COM port number on the top of the window. If it is not in the range COM1 – COM15, click at the triangle on the right next to the COM port number and select an available port from the drop-down list in the range COM1 - COM15.
10. In case all ports in this range are marked with **(already used)**, select a port which you are sure no actual device is currently connected to (preferred range COM1 – COM15). Answer the following security question by clicking at **Yes**
11. The now chosen COM port number has to be selected in the **Global Interface Setup** dialog box.

Alternativeley the COM port used by the MX10 can be detected using the **CAN-figurator** software which is installed together with STP. To do so, proceed as follows:

1. Start **CANfigurator** (Start / Programs / STP / CANfigurator)
2. Click at **Find CAN Adapter...**
3. CANfigurator now checks all available ports if the MX10 can be detected. As soon as this is the case, a specific message is shown
4. Under **Tools / Options...** the detected COM port number can be checked (**Comm port #:**):



5. By clicking at **Save configuration** the detected interface type and parameters can be forwarded to STP. At the next start of STP these settings will be used
6. If the port number of the COM port is not in the range COM1 – COM15, please proceed using the Windows Device Manager as described above (COM port number has to be changed).

Note: If you want to use the STP CANKey even when an MX10 base unit is available (e.g. because you have several PCs connected to the CAN bus and only one can use the MX10 to communicate with) the firmware in the STP CANKey has to be V1.4 or higher (the firmware version can be detected using the **CAN configurators Find CAN Adapter...** function). CANKeys with older firmware cannot handle the extended CAN identifiers used by the MX10. For an update of the CANKey please contact the manufacturer.

9.1.3.2 Littfinski TT-DEC Turntable Module

Besides the turntable module already supported by STP (the "MXDS" from Zimo which is no longer available), now turntable modules made by Littfinski Datentechnik ("TT-DEC") can be used with STP. The module is available in two versions (TT-DEC-R for the turntable from Roco and TT-DEC for the Fleischmann and Märklin turntable), for STP the interface is identical so only one module type for them is available in the **Function Modules** window. Except for the DCC address of the module (52 or 56) no further settings are required there.

The module address which has been set in the **Function Modules** window is then assigned to an element of type turntable in the **Change Board Element** dialog window. Using the **Layout** button the positions of the different track connections can be defined. More details about that can be found in the STP manual or in the online help under **Parameters for Element Type Turntable**.

For installation, connection and programming of the module please follow the instructions of the manufacturer. It is recommended to perform the setup of the track exits on the module using a handheld controller and not with STP. The operation of the turntable should be possible using the handheld before trying to control it using STP. It is important that the track exits are numbered in clockwise direction starting with track 1 and that exits which are exactly offset by 180° (i.e. on the opposite side of an already existing track exit) are also counted as shown on page 14 of the TT-DEC-R instruction manual.

It should be noted that the Littfinski module – in contrast to the Zimo MXDS module - is not capable of turntable position feedback. Due to that when activating **Online** mode in STP, the turntable will always be moved to position 1 to get a defined initial state.

The Littfinski module is not capable of turn direction optimization, the turn direction can only be set globally.

The control of the turntable in **Online** mode is identical to the one of the Zimo turntable module (see **Turntable Operation**) except that the **Calibration** button is not used and instead of the selection **short** and **long** now the turn direction is set using the **CW** (clockwise) and **CCW** (counter-clockwise) options.

9.1.4 General Changes

9.1.4.1 Engine Export from ADaPT

The software product ADaPT is available from the same manufacturer as STP and is used for train administration and decoder programming.

Starting with Version 3.13 of the decoder programming tool **ADaPT** it is possible to align and transfer the information from the existing engine database with the train information and engine groups available in STP (manageable with **Configuration / Train Numbers...**).

To do so a new entry **Export database to STP...** is now available in ADaPT's **File** menu. When this function is activated a file selection dialog window is shown where the position of the file STP_LOCO.DAT (in which STP stores its engine information) has to be given. This file is located in the STP installation directory. After clicking at **OK** all engines from ADaPT are copied into this file, already existing engines are updated. Existing engines which are not present in ADaPT will not be deleted. The changes will be considered by STP at the next start. STP should not be active during the export operation.

If a filter is active in ADaPT (i.e. only a selected number of engines is visible) only these engines are considered for export. At the end of the operation the number of forwarded engines is shown.

9.1.4.2 Extended Program Limits

The following maximum settings and restrictions have been adapted in STP 5.2:

Description	Max. value
Number of track elements in switchboard	7999
Number of turnouts in switchboard	1023
Number of signals in switchboard	1023
Number of route buttons	966
Number of track sections in switchboard	1008
Number of simultaneously active routes	127
Number of engine groups	99
Number of turnout / signal modules	255
Number of simultaneously open windows (except main window)	9
Number of j commands (change engine data) per route file	1000

9.1.5 Dispatcher Panel Display

9.1.5.1 Dispatcher Panel Windows

Up to now the display of the switchboard in STP was limited to a single window with given zoom (enlargement) factor. Starting with STP V5.2 now up to nine additional windows can be opened in STP's **Online** mode, which can show selected parts of the panels using different zoom values ("Spyglass" functionality). The operation of the switchboard can be done from any window, all changes are shown in all windows.

The definition of the windows is done using the **Settings...** entry in the new **Windows** menu (not available in Online mode).

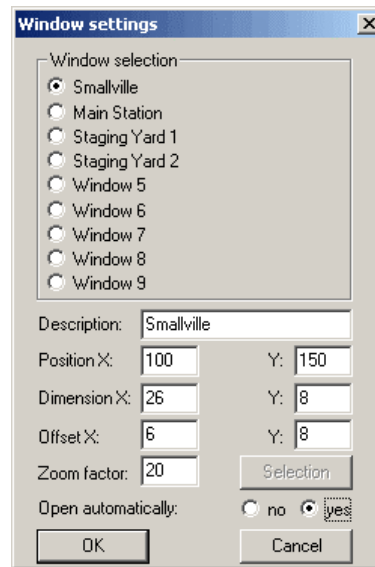


Fig. 3: Window settings

*If a window is moved in **Online** mode, the new position is saved internally. When the **Settings** window is opened again, this new position can be saved permanently by clicking at **OK**.*

The sum of dimension and offset should not exceed the current board size.

In the upper part of the dialog window you click at the window to be defined, then all data belonging to that window are shown and can be changed in the lower part of the dialog. The **Description** is later shown in the title bar of the window. **Position** gives the offset of the window in X- and Y-axis relative to the upper left corner of the screen in pixels, i.e. the position where the window will appear on the screen.

Under **Dimension** the size of the window in number of elements is given and **Offset** specifies at which element in the main window the upper left corner of the window starts (0 means at the top resp. left border).

If a selection rectangle is drawn in **Change** mode before the **Settings...** window is opened, the **Selection** button can be used to fill the fields for **Dimension** and **Offset** with the values of the rectangle selected before.

Using the **Zoom factor** the enlargement of the window can be defined where a value of **10** means a 1:1 display, **20** means double size and **5** half size (= reduction).

The selection **Open automatically** when set to **yes** causes that the currently selected window will be opened immediately when entering **Online** mode.

By clicking at **OK** the current settings of all windows will be stored permanently and are also available when STP is started the next time.

Note: The window settings are stored globally, i.e. they are not saved in the switchboard file and so are identical for all dispatcher panels.

When in **Online** mode by selecting an entry from the **Windows** menu the corresponding window will be opened. To close a window just click at the proper part in the title area of the window.

9.1.5.2 Restore Window Position

When terminating STP the current position of the STP main window and the pan position (i.e. the currently visible section of the main window) are saved and automatically restored when STP is started the next time.

9.1.5.3 Display of Communication State of Zimo MX9 Modules

Especially with larger layouts the monitoring of the different components of the digital system becomes more important because a failure of a module or cable can result in an elaborate search after possible reasons of the breakdown.

To simplify the trackdown of such problems it now is possible to use elements of type **Track** and the color **Add. indicator red** to display in red / green if the communication state to an MX9 track section module is still working. The assignment of the element to the module is done by the track section number of the element, which has to correspond to the module to be monitored.

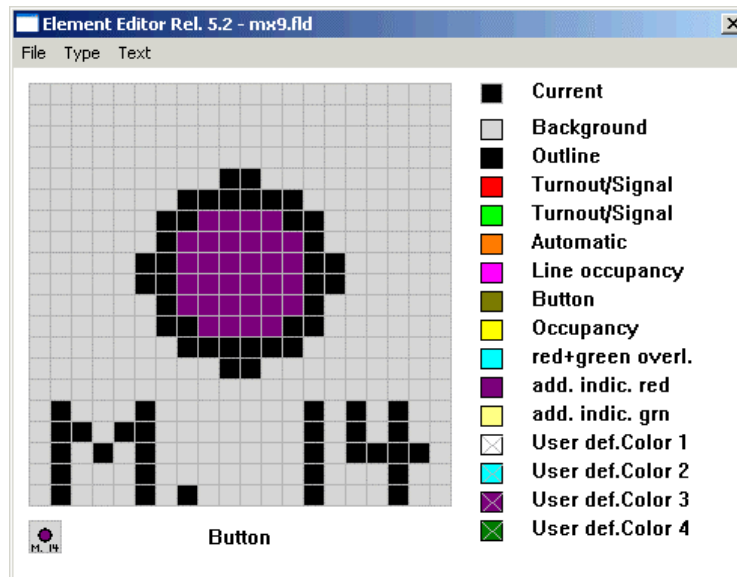


Fig. 4: Element for MX9 communication status (example)

As an example, to monitor three MX9 modules with addresses 901, 902 and 903, create three track elements which have track sections 1, 17 and 33 assigned.

Note: These elements can of course also be used as "real" track elements, in most cases they are located together in a separate part of the panel similar to special buttons. They then consist only of a status display (i.e. drawn in the Element Editor using the color **Add. indicator red**).

9.1.6 Routes

9.1.6.1 Variables in Routes

Problem Statement

In spite of the numerous possibilities which STP provides for route programming there are cases which cannot be solved or need complicated combinations of commands. Examples are the definition of cyclic operations (i.e. main station – staging yard – main station), but after three cycles a different direction has to be driven. Or a train should get informations "attached" when leaving a station to which track at the target station it should be routed (this currently can only be done using the time table).

In order to perform such tasks it has to be possible to execute routes or parts of routes only under certain circumstances or conditions. These conditions can now be defined using variables.

Concept

The "engine relation" of variables is no direct binding but is more a logical concept, i.e. the usage of such a variable normally is in association with a specific engine.

Variables are placeholders or "storages" for numbers (whole numbers, positive or negative) and can contain values between +32767 and -32768. Each variable has a number under which it can be accessed, currently 11999 variables (variable 1 to variable 11999) are available, where each variable internally is assigned to an engine address (i.e. variable 1 corresponds to the engine with address 1 etc.). Additionally there are 999 "free" variables which are not engine-bound. Each variable can be set to a given value, can be increased or decreased.

In routes now these variables can be checked, i.e. if they are larger than a given value or equal to zero etc. and then – in case the condition is true – cancel the activation of the route. For the example from above with the cyclic movement that should be performed three times this would mean that e.g. in the exit route a certain variable always is increased by one and in the entry route there is a check if the value of this variable is larger than three. If not, the entry route will be set, otherwise it will not be activated and any entry automatism (when present) will select the next route from the list of available routes.

When starting STP all variables are initialized to the value 0, when activating **Online** mode also all variables except for the "free" variables are reset to 0.

Route Commands for Variables

Variables can be accessed in routes in two ways:

- To set or change a value: route command **#**
- To compare with values for a check whether the route should be activated: route command **?**

The variables themselves are specified by their number (1-11999) with two possible extensions:

- "Free" variables are identified by a leading **F** (e.g. **F112**)
- Engine related variables can also be addressed "indirectly" by - instead of the variable number - specifying a track section number and the address of the engine in that track section is the variable number to be used; for this an **S** is put in front of the variable number (e.g. **S25** means the variable with the number of the address of the engine in track section 25 – if there is e.g. engine 103, variable 103 is addressed)

The general syntax to **set a variable** is:

"#<var>:<op>value", where <var> is the variable to change (1-11999, S1-S1008, F1-F999), <op> the assignment operator (=, +, -, !) and value the value at which the variable is to set, to increment or to decrement.

The assignment operators have the following meaning:

= ... the variable is set to this value

+ ... the variable is incremented by this value

- ... the variable is decremented by this value

! ... the variable is inverted (if it contains a value different from 0 it is set to 0, if it contains the value 0, it is set to 1).

The general syntax to **check a variable** is:

"?*var*>:<*op*>*value*", where <*var*> is the variable to check (1-11999, S1-S1008, F1-F999), <*op*> the comparison operator (=, >, <, !) and *value* the value against which the check should be made.

The comparison operators have the following meaning:

= ... the variable is checked if it is identical to the given value

> ... the variable is checked if it is larger than the given value

< ... the variable is checked if it is smaller than the given value

! ... the variable is checked if it is different to the given value

If the comparison command is put at the beginning of a route definition, the route will not be set at all when the condition is true and a possible check from an automatism will perform as if the route was not activatable. If the comparison command is located further back in the route definition, the route is executed up to the comparison command, but all following commands of the route will (in case the condition is true) not be executed.

If the condition is fulfilled (true), the route will not be set or it will be interrupted.

As a support when testing variable commands, the current values of variables can be checked using the menu command **Info / Variable info**.

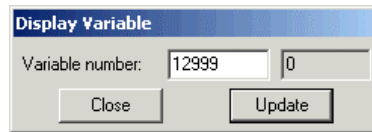


Fig. 5: Display Variable

The "free" variables can be displayed by entering their number +12000 (i.e. for variable F103 enter 12103 here). By clicking at **Update** the currently valid content of the variable will be shown.

Examples

#17:=4 ... Set the engine-related variable 17 to the value 4

#F1:=3 ... Set the "free" variable 1 to the value 3

#S23:+2 ... Increase the engine-related variable, whose number corresponds to the engine address in track section 23 by 2

?S23:>3 ... Check if the engine-related variable, whose number corresponds to the engine address in track section 23, is larger than 3 – if true, do not activate this route or interrupt it

?17:=5 ... Check if the engine-related variable 17 has the value 5 – if true, do not activate this route or interrupt it

9.1.6.2 Execute Commands when a Route is Cleared

Using the new route commands **n** and **p** now commands can also be executed when a route is cleared. The commands work similar to the route connection commands **K** and **k**, i.e. the parameters specify the start and end buttons of another route that will be set (**n**) or cleared (**p**) when the route containing the **n** or **p** command is cleared.

Using these commands e.g. a reversion of the travel direction using direct commands or the switch-off of illuminations or track power ((**Px/4**, *x* = track section number) are made quite simple.

Format: "na/b", where *a* = start button of route to be activated, *b* = target button
"pa/b", where *a* = start button of route to be cleared, *b* = target button

Example: n11/12 ...Activates route S:11/12 when the route is cleared
p25/16 ... When route is cleared, also clears route S:11/12 (in case it is active at that time)

9.1.6.3 Handover of Trains to other Zimo Base Stations

Concept

Such modules that allow the handover of trains between two digital command systems are available from AMW (amw.huebsch.at)

The possibility for handover of trains between Zimo base stations is intended for very large layouts which cannot be operated by a single Zimo system. To accomplish this, trains have to be switched over using overlapping track sections that have to be addressable from one and the other digital system (switched by a relay controlled by turnout commands from routes, so there always is only one connection active).

When forwarding trains beyond these borders, STP has to read the current engine information (speed step, light, additional functions, ...) from the start base unit and send it to the target unit, so that the train is known there and the information in the target digital system is updated accordingly.

Two (or more) PCs are needed which are connected via Ethernet (TCP) and on which STP is running, each connected to "its" command station. The Zimo base units are not directly connected in any way.

Using a new route command **@x/y** STP reads out the address(es) of the train in section **y**, determines the current data of this train(s) (such has speed, function state, direction) and sends the data to PC **x** using a TCP connection. The STP instance on PC **x** receives that information and forwards it to its own Zimo base unit, whereafter this settings are also available on this system.

Settings

For this configuration some network knowledge is required.

To allow the PCs to communicate over TCP/IP several settings have to made in the STP.INI configuration file:

```
TCPFwdPort=1339
```

On this port STP "listens" for TCP messages of other STP systems. The port is openend when Online mode is activated and closed when this mode is finished. The port number (the value above is recommended, but can be changed if required) has to be opened in a Firewall if required.

```
TCPFwdAddr_1=192.168.0.10
```

```
TCPFwdAddr_2=192.168.0.23
```

```
...
```

The TCP addresses of all PCs to which engine information should be sent have to entered in the STP.INI file in the format shown above. For the PC with STP network address 1 as "TCPFwdAddr_1", for the PC with network address 2 as "TCPFwdAddr_2" etc. (up to "TCPFwdAddr_14"). Using the route command **@1/37** the information for the engine(s) in section 32 will be sent to PC 1 (192.168.0.10, port 1339 in the example above).

Usage

For the handover of engine data between systems the new route command **@x/y** (see above) is used. If the command is found in a route, a connection to PC **x** (1-14) is established. If the connection has already been made (by a previous **@** command to the same target PC) it is reused. If no connection can be established to PC **x**, a corresponding error message will be shown (e.g. because the target PC is not reachable or not on Online mode). When the next **@** command is encountered, a new attempt to create a connection is performed.

Format: "@a/b", where *a* = STP network address of target PC (1-14), *b* = track section number from which address(es) of engine(s) should be read.

Example: @2 / 25 ... Send the data of the engine(s) from track section 25 (direction, speed step, ...) to the PC with STP network address 2

Important: If no engine address is known for track section *y* at the moment the route containing this command is executed, no data will be sent over the TCP connection. Up to four engine addresses can be sent for track section *y*.

9.1.6.4 Modify Name of Train and Engine Group

When a locomotive gets a different train attached in a station it is often necessary to change the train name and possibly also change the engine group as the locomotive now carries a passenger train and no longer a freight train. Until now you had to do these changes manually either using the **Configuration / Train Numbers** menu entry or by using the context menu of a track section (right mouse button on a track section in **Online** mode).

Using the new route command **j** now by specifying a track section number, **Engine group**, **Short name** as well as **Description** of a locomotive can be changed from within a route.

Format: "js/g "short" "long"", where *s* = track section where the engine is located, *g* = new engine group, "short" = short name, "long" = description. Short name and description may be omitted, but when description is specified also the short name has to be present.

Example: j34/5, "R1475", "Commuter train" ... sets the engine group of the locomotive in section 34 to 5, the short name to R1475 and the description to Commuter train.

Note: All changes made this way (i.e. by route command **j**) are only temporary, i.e. after STP is restarted, all data is restored as it was originally entered under **Configuration / Train Numbers**. If several engines are present in the specified track section, only the data of the first (detected) engine is modified. Up to 1000 **j** commands are allowed in one route definition file.

9.1.6.5 Clearing of Memory Routes

The clearing of a memorized route (see "Memory Routes" in chapter "Special functions") was only possible by using the special button assigned for that purpose. Now the new route command **r** allows to erase a route from route memory (if present).

Format: "ra/b", where *a* = start button of the memory route to be cleared, *b* = target button

Example: r11/12 ... erases the route S:11/12 from the list of memorized routes (if this route is memorized)

9.1.6.6 Playback of Sound files on Multiple Speakers

In modern PCs a multi channel sound card is present, i.e. to use this function only the appropriate number of speakers have to be attached.

Starting with STP V5.2 it is possible to direct a sound to a specific speaker if multiple speakers are connected to the sound card of the PC. In addition the restriction that a new sound that is activated mutes any sound that was active before has been removed, i.e. multiple sounds can be played back simultaneously (the number is only limited by the sound card in use).

For that purpose it is possible to mount several speakers under the layout and then use the **G** route command to direct certain sounds to specific locations of the layout e.g. a whistle sound when reaching a grade crossing or a tunnel or station announcements when the train enters a platform.

To address a specific speaker the second parameter of the **G** route command is now re-defined, it specifies the number of the sound output of the sound card (1 – 9). Which speakers these numbers 1 – 9 are related to can be checked in the sound settings of Windows. The number 0 means no specific speaker and defaults to the standard sound output (normally two-speaker stereo).

The standard assignment of speakers in Window is as follows:

Channel	Assignment
1	1 = Front left
2	2 = Front right
3	3 = Front center
4	4 = Subwoofer
5	5 = Back left
6	6 = Back right
7	7 = Back left + right
8	7 = Back center
9	9 = Differently, depending on sound hardware

Format: "Gsound, a, b", where *sound* = name of WAV file, *a* = track section number for train specific sound assignment or 0, *b* = speaker / channel number (0-9)

Example: Gwhistle, 0, 3 ... Plays the file *whistle.wav* on speaker / channel 3

Note: Only use sound files with one channel (mono) for this purpose.

9.1.6.7 Keyboard Shortcuts for STP Wizards

For the wizards available in STP (route wizard, line wizard, automatic wizard) now keyboard shortcuts are available for the Wizard dialogs:

Keyboard shortcut	Function
Alt+W	Arrow up
Alt+X	Delete
Alt++	+ (Automatic wizard)
Alt+Q	Edit
Alt+S	Arrow down

9.1.6.8 Communication with Third Party Software

In specific application scenarios it may be useful to use data from STP in other programs, especially information about the current location of trains on the layout or when specific routes are activated.

Examples for such applications could be a specific illumination software which directs spot beams to the location where a specific train is located or a camera tracing software which monitors a train along the layout using multiple cameras.

STP can now forward this information to other programs using the TCP protocol, either on the same PC or on another one connected via Ethernet.

Programming as well as network skills are assumed for the creation of software which communicates with STP.

Note: The remaining part of this chapter is a step-by-step explanation of this information exchange. The content is targeted against developers of software which want to use this data and is not required or relevant for the understanding or operation of STP.

1. The IP address (129.168.x.y) of the server PC (the PC where the application which receives the train position information from STP) and the port number (example: 1338) have to be set in the STP.INI file as shown in the following example:

```
TCPMonAddr=192.168.0.5
```

```
TCPMonPort=1338
```

This server PC can be the same PC where STP is running on ("Mon" stand for "Monitoring").

2. The server program has to started before STP online mode is activated. Note that the port has to be made available if a firewall is running.

3. If the STP.INI settings shown above are made, whenever STP enters online mode it connects to the server application and sends a message like

```
[ 01 ] I : INIT
```

followed by the linefeed character <lf>. The general format of the telegrams is [<adr>] <type> : <data> <lf> where <adr> is the STP network address of the PC (01-14), <type> is I for informations or R for the route command (see below) and <data> is data depending on telegram type.

There are two types of information telegrams:

```
[ 01 ] I : INIT
```

```
[ 01 ] I : EXIT
```

which are sent whenever STP enters (INIT) or leaves (EXIT) **Online** mode. In the second case STP closes the connection to the server application.

4. Whenever a train moves along a route or line and it crosses a section boundary and the train address(es) are known, a message like this is sent:

```
[ 01 ] S : 0036 / 0032 ( 00003 , 00000 , 00000 , 00000 )
```

'S' indicates the message type (Section change), followed by from / to section (train comes from section 36 and now enters section 32) and up to four train addresses (in this case only one - 3, the remaining three are set to 0).

Note that due to occupancy detection and processing it is possible to receive (spurious) messages with the same from and to section like

```
[ 01 ] S : 0032 / 0032 ( 00003 , 00000 , 00000 , 00000 )
```

5. The route command @ may also be specified in the format @0 / <info>:

```
@0 / 10
```

Whenever this command is encountered when setting a route, the server gets a message like:

```
[ 01 ] R : 0052 / 0053 ( 10 )
```

'R' stands for message type Route command, followed by start and end button number of the route followed by the <info> value (allowed range 0-32767). The number after the @ is zero (0), other values are currently reserved for the STP to STP communication, see chapter "Handover of Trains to other Zimo Base Stations" on page 11.

6. If the server is stopped during online mode or not available when entering online mode, an error message is shown and no further attempts are made to contact the server until STP online mode is activated again.

The STP installation now includes a subfolder **VB** which contains a simple Visual Basic example program (usable with VB version 6 and up). It demonstrates the communication using the protocol explained above.

9.1.6.9 Tabular Overview of all Route Commands

Id	Description	Param. 1	Param. 2	Par. 3 ¹⁾	Examp.
@	Train data forwarding to another PC	PC network address or 0	Section number / data	-	@2/25
#	Setting of variables	Variable number	Assignment operator ¹⁸⁾	Value	#F17=4
?	Comparison of variables	Variable number	Comparison operator ¹⁹⁾	Value	?S23>10
A	Section speed step	Track section number	Speed step (0-7) ²⁾	Speed step (0-7) ³⁾	A17/3
a	Section illumination on panel	Lamp address	Section number	-	a7.123, 47
B	Number of occupied sections (activation)	Number of sections	-	-	B2
b	Route illumination for button	Button number	0=off, 1=on	-	b13/1
C	Route clearing by section occ. / clear	Section number	Clearing event (0-2) ⁴⁾	-	C27/0
D	Direct command (loco addr. related)	Decoder address	Function number ⁵⁾	0=off, 1= on, 2=tog.	D3/15, 2
D	Direct loco decoder command	Decoder address ¹⁰⁾	Command byte ⁶⁾	-	D14/31
E	Direct command (track sect. related)	Track section number	Function number ⁵⁾	0=off, 1= on, 2=tog.	E42/15, 2
E	Direct decoder command (section)	Section number	Command byte ⁶⁾	-	E27/31
F	Number of free sections (release)	Number of sections	-	-	F3
f	Release on single section	0=off, 1=on	-	-	f1
G	Sounds	Sound file name	Section number ¹¹⁾	Speaker number	Gtrain
g	Launch external program	Program name	Parameter 1 ⁷⁾	Parameter 2 ⁷⁾	H18/0
H	Route type	Route type ⁸⁾	-	-	H2
h	Load / save current state	0=load, 1=save	-	-	h0
I	Activate / deactivate automation function	Button number	0 – 4 ⁹⁾	-	I27/3
I	Memory route clear section	Track section number	-	-	i27
J	Engine group inclusion	Track sections	Engine group	-	J3+4/2+1
j	Change engine data	Track section number	Engine group	Short name, description	j27/1, "R175", "BR103"
K	Route connection	Start button	Target button	-	K11/14
k	Route connection (clear route)	Start button	Target button	-	k11/14
L	Track illumination	Track element number	0=off, 1=on	-	L285/1

M	Section illumination	Section number	0=off, 1=on	-	M27/1
N	Route number	Route # (1-999999)	-	-	N6
n	Route for clearing (activation)	Start button	Target button	-	n11/14
O	Test for occupied section	Section number	-	-	O27
o	Track section occupancy check	Track section number	0 – 3 ¹²⁾	-	o24/1
P	Protective track section speed step	Track section number	Speed step (0-7) ²⁾	Speed step (0-7) ³⁾	P18/0
p	Route for clearing (clearing)	Start button	Target button	-	p11/14
Q	Sound control	Button number	-	-	Q27
R	Overlapping sections	Number of sections	-	-	R3
r	Clear memory route	Start button	Target button	-	r11/14
S	Signal set command (train route)	Signal number	Signal state (0-3) ¹³⁾	Sig. aspect, reset ¹⁴⁾	S5/1, 3, 2
s	Signal indicator (not in route)	Signal number	0=off, 1=on	-	s16/1
T	Signal set command (shunting route)	Signal number	Signal state (0-3) ¹³⁾	Sig. aspect, reset ¹⁴⁾	T6/0
t	Set STP clock	hh:mm			t15:30
U	Turnout release / unlock	Turnout number	Section number	-	U19/27
u	Turnout lock-change	Turnout number	Type of change ²⁰⁾	-	u19/1
V	Signal release / unlock	Signal number	Section Number	-	V15/28
v	Signal lock-change	Signal number	Type of change ²⁰⁾	-	v19/1
W	Turnout set command	Turnout number	Position (0-3) ¹⁵⁾	Track section number	W27/1, 4 2
w	Turnout indicator (not in route)	Turnout number	0=off, 1=on	-	w20/1
X	Distant signal dependency	Distant signal number	Main signal number	-	X16/15
x	Restriction for direct command	Address / engine group	Restriction (1-4) ¹⁶⁾	-	x611/1
Y	Engine group exclusion	Track sections	Engine group	-	Y3+4/4+3
Z	Turnout settle delay	Settle time (1/10 sec) ¹⁷⁾	-	-	Z50
z	Time delay	Delay time (1/10 sec) ¹⁷⁾	-	-	z30

Table 1: Route elements (overview)

Footnotes:

¹⁾ Parameter 3 is always optional, so it can be omitted.

²⁾ 0="H", 1="U", 2="L", 3="F", 4="A", 5="HU", 6="UL", 7="LF"

³⁾ State after contact has been reached: 0 = "H", 1 = "U", 2 = "L", 3 =

- "F", 4 = "A", 5 = no action
- 4) 0 = release when occupied, 1 = release when free, 2 = release when contact reached
 - 5) 0 = Lamps, 1 - 12 = F1 - F12, 14 = MAN, 15 = drive direction
 - 6) Speed step (0-15) + light (0=off, 1=on) * 16 + direction (0=forward, 1=backward) * 32 + Z function (0=off, 1=on) * 64 + MAN function (0=off, 1=on) * 128
 - 7) Parameters are passed to the program, range 0 - 32767
 - 8) 0 = general, 1 = only main route, 2 = only shunting route, 3 = route without lock nor release, 4 = permanent route
 - 9) 0 = Deactivate automation (red), 1 = trigger automation (green), 2 = always trigger automation, 3 = trigger active automation (brown), 4 = trigger automation when cancellation section ("S") is occupied (brown)
 - 10) DCC track data format: 0-10234, Zimo format: 11000-11255, Motorola format: 12000-12080
 - 11) When using "%d" in the sound file name and specifying track sections from where to obtain train addresses; if "%d" is contained in the file name at least section number 1 (parameter 2) has to be given.
 - 12) 0 = Section has to be clear, 1 = section has to be occupied, 2 = like 0 but "OR" logic with next "o" command, 3 = like 1, but "OR" logic with next "o" command
 - 13) 0 = red , 1 = green, 2 = red protective signal, 3 = green protective signal
 - 14) This in fact are two parameters in one, signal aspect (1-10) and reset condition (0 - 3, 0 = default, signal reset when following section gets occupied, 1 = reset when signal track section gets free and 3 = reset when route is cleared; the reset parameter may be omitted, if the reset parameter is specified, also the signal aspect has to be given; both parameters are delimited by commas)
 - 15) 0 = straight , 1 = deviated (bend) , 2 = straight protective turnout, 3 = deviated (bend) protective turnout
 - 16) 1 = the following direct commands are only valid for loco addr. in p. 1
2 = the following direct commands are not valid for loco addr. in p. 1
3 = the following direct commands are only valid for eng. group in p. 1
4 = the following direct commands are not valid for eng. group in p. 1
 - 17) Value range 0 - 6000, command may appear multiple times in route.
 - 18) = (set), + (increase), - (decrease), ! (invert)
 - 19) = (is identical), > (larger), < (smaller), ! (is different)
 - 20) 0 = perform lock, 1 = perform unlock, 2 = invert current lock state

Please note that parameter 3 may be omitted for the commands "A", "P", "S", "T" and "W", for command "g" also parameters 2 and 3.

9.1.7 Layout Operation

9.1.7.1 Open ADaPT Engine Throttles

The **Train number** dialog window, which can be opened in **Online** mode by clicking at an occupied track section, now contains an additional button **Cab**. If the decoder programming and engine database software ADaPT (release 3.13 or higher) is installed on your PC and if it is started, this button is available (i.e. not greyed out) and – when clicked – opens an ADaPT engine throttle for the engine address shown in the **Train number** dialog window, so that the corresponding locomotive can be controlled with it.

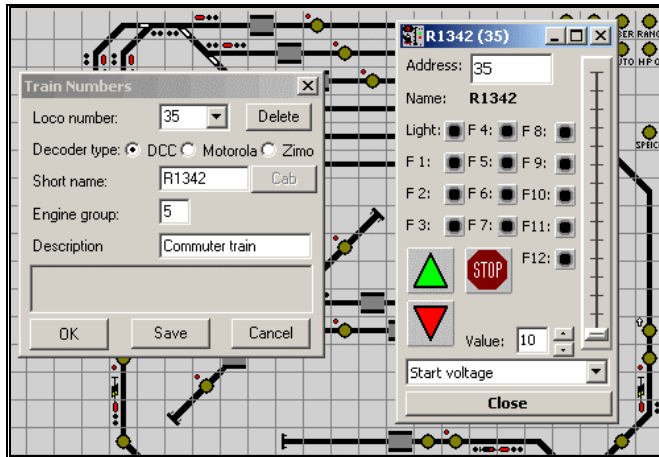


Fig. 6: ADaPT engine throttle in STP

It should be noted that the Zimo MX10 base unit currently does not support a simultaneous usage by two applications, i.e. the parallel operation of STP and ADaPT is not possible as only one program can communicate with this central station at a time. As soon as the Ethernet interface of the MX10 is operational, a common operation (STP using Ethernet, ADaPT using USB) should be possible.

Alternatively STP can also communicate with the layout using the STP CAN-Key, which leaves the USB interface of the MX10 available for ADaPT.

Note: By using the functionality explained in chapter "Engine Export from ADaPT" on page 6, which is available in ADaPT Rel. 3.1.3 and above, the train information in STP (**Configuration / Train numbers**) can be aligned with the ADaPT engine database.

9.1.7.2 Display of "external" Routes

In network mode, i.e. when operating several STP systems on a single layout connected by a common CAN bus, all activated routes are displayed on all PCs. This way it is not possible to distinguish which routes have been activated at your local PC and which on a remote PC, so a specific color is now introduced for "external" routes set by a remote PC.

By default these routes are now shown in color 14 (light yellow). If this color should be changed, open the file **STP.INI** with a text editor (e.g. Windows Notepad) and add a line like

```
ShowExtRoutes = 7
```

(optimally at the end of the file), where 7 is the selected color (see **Color Overview Table**), 0 = off (i.e. no different color for external routes).