# RMFT – Reference

Numbers:

All route, sensor, output, turnout or signal ids are limited to 0- 255 ( A UNO does not have enough RAM so the compiler limits this to 0-63 on a UNO device)

The same id may be used for a route, turnout, sensor, output or signal without confusing the software (the same may not be true of the user!).

Its OK to use sensor ids that have no physical item in the layout. These can only be set, tested or reset in the scripts. If a sensor is set on by the script, it can only be set off by the script… so AT(5) SET(5) for example effectively latches the sensor 5 on when detected once.

You can give names to routes turnouts signals and sensors etc using #define or “const byte “ statements.

COMMAND REFERENCE

There are some diagnostic and control commands added to the <tag> language normally used to control the command station over USB, Wifi or Ethermet:

|  |  |
| --- | --- |
| <D TPL ON|OFF> | Turns on/off diagnostic traces for TPL events |
| <t …> | Throttle commands are only accepted for locos that are not currently being controlled by TPL (This not yet implemented) |
| </ PAUSE> | Pauses automation, all locos ESTOP. |
| </ RESUME> | Resumes automation, Locos are restarted at speed when paused. |
| </ STATUS> | Displays TPL running thread information |
| </ SCHEDULE [loco] route> | Starts a new thread to send loco onto route. or Start a non-loco animation route) |
| </ RESERVE id> | Manually reserves a virtual track block. |
| </ FREE id> | Manually frees a virtual track block |
| </ TL Id> | Set turnout LEFT |
| </ TR id > | Set turnout RIGHT |
| </ SET id> | Lock sensor |
| </ RESET id> | Unlock sensor |

Routes and animations.

The TPL system operates on a number of concurrent “threads”. Each thread is following a route through the system and usually has an associated loco that it is driving. Some threads may be driving animations and have no loco attached. The thread keeps track of the position withing the route and the loco speed. A thread may be delayed deliberately or when waiting for a sensor or block section, this does not affect other threads.

At system startup, a single thread is created to follow the first entry in the routes table, with no loco. .

|  |  |
| --- | --- |
| ROUTES | Start of routes table. |
| ROUTE(routeid) | Start if a route  routeid=0-255 |
| AFTER(sensorid) | Waits until sensor reached, then waits until sensor no longer active for 0.5 seconds |
| AT(sensorid) | Waits until sensor reached |
| DELAY(duration) | Waits for duration/10 seconds |
| DELAYMINS(duration) | Waits for a number of minutes |
| DELAYRANDOM(minduration.maxduration) | Waits a random time between minDuration/10 and maxDuration/10 seconds. |
| ENDIF | Marks end of IF block (see IF command) |
| FOFF(func) | Switches loco function off |
| FON(func) | Switches loco function on |
| FOLLOW(routeid) | Continue at ROUTE(routeid) command |
| FREE(blockid) | Frees a previously reserved block. See RESERVE(blockid) |
| FWD(speed) | Drive loco at given speed (0-127) forwards (0=stop, 1=ESTOP) |
| GREEN(signalId) | Sets signal to green |
| IF(sensorId) | Checks if sensor is activated, if NOT then processing skips to the matching ENDIF command (allowing for nested IF/IFNOTs ) |
| IFNOT(sensorId) | Checks if sensor is activated, if it is active then processing skips to the matching ENDIF command (allowing for nested IF/IFNOT/IFRANDOMs ) |
| IFRANDOM(percent) | Randomly decides whether to continue or skip to the matching ENDIF |
| INVERT\_DIRECTION | Causes current loco FWD and REV commands to be reversed (e.g. used if loco is pointing in wrong direction) |
| PAUSE | Sets TPL into paused mode, all animations and locos are stopped and manual control is possible |
| PROGTRACK\_JOIN | See DCCEX cmd <1 JOIN> |
| PROGTRACK\_OFF | See DCC cmd <0 PROG> (Disconnects a JOIN) |
| READ\_LOCO | Reads loco id from prog track and assigns it to current route |
| RED(signalId) | Sets Signal to RED |
| RESERVE(blockId) | Blockid=(0-255)  If block is already reserved by another train, this loco will STOP and wait for the block to become free. block is marked as reserved and this train continues..  When you leave a block that you have reserved, you must FREE it. |
| RESET(sensorId) | Clears a sensor flag (see SET) |
| RESUME | Resumes TPL from PAUSE mode. Locos stopped by PAUSE are restarted. |
| REV(speed) | Move loco in reverse (see FWD) |
| SCHEDULE(routeid) | Starts a new thread at ROUTE(routeid) and transfers current loco to it. |
| SETLOCO(locoid) | Sets the loco id of the current thread. |
| SET(sensorId) | Locks on the software part of a sensor. If a sensor is tested by AT/AFTER/IF etc and the software part is locked on, then the sensor is seen as active without a hardware check.  NOTE: This can be used for debounce. It can also be used for virtual sensors that ONLY exist in software and have no hardware equivalent. Can be used for example to pass information from a travelling train thread to a lineside animation thread. |
| SPEED(speed) | Changes loco speed in current direction. |
| STOP | =SPEED(0) |
| ESTOP | =SPEED(1) DCC emergency stop |
| TL(turnoutId) | Sets turnout LEFT |
| TR(turnoutId) | Sets turnout RIGHT |
|  |  |
| ENDROUTE | Terminates a route thread |
| ENDROUTES | End of ROUTES table, must be last entry. |