

Track Environment

Inputs

Name	Type	Range	Short description
TrackSpeeds	int[48]	[-255 , 255]	The speed per track. Track names can be used as indices. (e.g. KH_ST_0)
Points	int[30]	{BRANCH, STRAIGHT}	The numbered values of the switchable intersection points. (e.g. 23)
TrackSignals	bool[40][2]	{true, false}	The signals on each track. If the track contains signals it's always two. Track names can be used as indices. (e.g. KH_ST_1)

Outputs

Name	Type	Range	Short description
TrainPositions	int[15]	[0 , 47]	The position, that a given train currently occupies. (e.g. Train TRAIN_KH_ST_0 occupies KH_ST_0 on startup)
TrainSpeeds	int[15]	[-255 , 255]	The speed, that a given train currently occupies (e.g. Train TRAIN_KH_ST_0 has speed 100)
ForwardNext	int[48][5]	[0 , 48] (-1 means invalid)	The next lines in the main direction. Each track can have at most 5 possible lines. -1 for all entries, that are not valid lines. (e.g. ForwardNext[KH_ST_1][0] == KH_ST_6, ForwardNext[KH_ST_1][1] == -1,...)
ReverseNext	int[48][5]	[0 , 48] (-1 means invalid)	Same behaviour as ForwardNext, but in the secondary direction. If the secondary direction is not permitted all values are -1.
Error	string		Error messages. (e.g. If a required track signal did not trigger)

Requirements

For two directly connected Tracks the IDs of the Points that need to be passed (if any) including their direction (branch, straight).

Idea: A big matrix in the form of PointPositions[48][48][2][2]. So for each set of tracks the points (if any) and as value either branch or straight.

```
PointPositions[KH_LN_5][KH_LN_7][0] = {14, BRANCH}
PointPositions[KH_LN_5][KH_LN_7][1] = {null, null}

PointPositions[KH_ST_4][KH_ST_0][0] = {1, BRANCH}
PointPositions[KH_ST_4][KH_ST_0][1] = {2, BRANCH}

PointPositions[KH_LN_5][KH_LN_6][0] = {null, null}

...
```