

trainmap

Schematic railway and transit maps with TikZ

in the style of FGC / Rodalies de Catalunya line diagrams

Pol Sardà Palla

psardapalla48@gmail.com

Version 1.0.0 • 2026/08/28

Abstract

trainmap is a small L^AT_EX package, built on TikZ, for drawing schematic railway and transit line diagrams *declaratively*. Instead of positioning every node by hand you describe a line as a sequence of stations and terminals, and the package advances an internal cursor, draws the track segments, places the station symbols and rotated labels, and even folds very long lines onto several rows. It ships with the official line colours of *Rodalies de Catalunya* and *FGC*. This manual documents version 1.0.0.

Contents

1	Introduction and purpose	1
2	Installation	2
2.1	Next to your document (quickest)	2
2.2	Into your personal <code>texmf</code> tree (available everywhere)	2
2.3	Loading it	3
3	A first map	3
4	Command reference	3
4.1	The <code>trainmap</code> environment	3
4.2	Declaring the trunk	3
4.3	Stations	4
4.4	Tracks, branches and forks	4
5	Configuration keys	5
5.1	Predefined colours	5
6	Worked examples	5
6.1	Correspondence chips	5
6.2	Several lines and a fork (<code>mode=fgc</code>)	6
6.3	Long lines: the serpentine layout	6
7	Licence	6

1 Introduction and purpose

Transit line diagrams follow a very regular visual grammar: a coloured track, tick marks for ordinary stations, filled discs for terminals, capsules where several services meet, and labels set

at a slant so long station names do not collide. Reproducing that grammar by hand in *TikZ* is repetitive and error-prone — every coordinate has to be computed manually.

trainmap captures the grammar in a handful of commands. You declare the tracks that form the *trunk*, then call `\addstation`, `\addterminal` and `\addinterchange` in order. Each call advances a horizontal cursor, draws the connecting track, and renders the station. Branches and forks are drawn with curved connectors (`\forkup`, `\forkdown`, `\joinline`, ...), and lines that are too long to fit on one row fold automatically into a *serpentine* layout.

The package offers two presentation styles, selected with the `mode` key:

mode=fgc (default)

Every track of the trunk is drawn at once; each terminal carries a coloured *badge* with its line code below the disc. This is the classic multi-line FGC schematic.

mode=rodalies

A single line is drawn, and its code appears on a square plate at each end of the whole route. Suited to long single lines, usually combined with serpentine wrapping.

2 Installation

The package consists of two files that **must always travel together**:

File	Role
<code>trainmap.sty</code>	$\text{\LaTeX}$ wrapper — enables <code>\usepackage{trainmap}</code> .
<code>tikzlibrarytrainmap.code.tex</code>	The implementation, loaded by <code>\usetikzlibrary{trainmap}</code> .

2.1 Next to your document (quickest)

Copy *both* files into the same folder as your `.tex` file and compile with `pdflatex`. $\text{\LaTeX}$ searches the document's own directory first, so nothing further is needed.

2.2 Into your personal `texmf` tree (available everywhere)

```
# 1. Locate your personal texmf root
kpsewhich -var-value TEXMFHOME
#   Linux/macOS: usually ~/texmf
#   Windows/MiKTeX: usually C:\Users\<user>\AppData\Roaming\MiKTeX

# 2. Create the destination and copy BOTH files
mkdir -p "$(kpsewhich -var-value TEXMFHOME)/tex/latex/trainmap"
cp trainmap.sty tikzlibrarytrainmap.code.tex \
  "$(kpsewhich -var-value TEXMFHOME)/tex/latex/trainmap/"

# 3. Refresh the filename database
mktexlsr           # TeX Live
initexmf --update-fndb # MiKTeX (or use the MiKTeX Console)
```

After a database refresh, `\usepackage{trainmap}` works from any document. When the package is installed from CTAN by your distribution's package manager this is all handled for you.

2.3 Loading it

There are two equivalent ways to load the library:

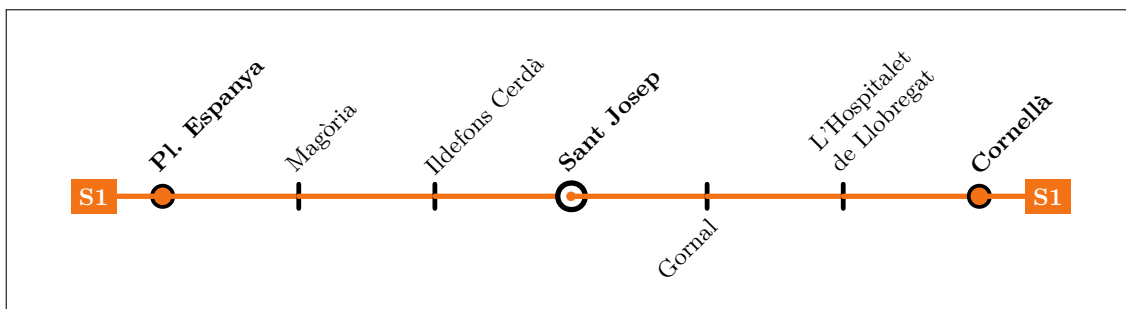
```
\usepackage{trainmap}           % as a LaTeX package
% or, if TikZ is already loaded:
\usetikzlibrary{trainmap}      % as a TikZ library
```

3 A first map

The following complete document draws a single Rodalies-style line. The `mode=rodalies` key puts a square `[S1]` plate at each end; station names are set at 45°; `\addinterchange` marks a transfer stop; and `\\` breaks a long name.

```
\documentclass[border=10pt]{standalone}
\usepackage{trainmap}
\begin{document}
\begin{trainmap}[mode=rodalies, station step=2.0, badge color=tmS1]
  \settrunk{0/tmS1}           % the single S1 track (orange)
  \addterminal{Pl. Espanya}{S1} % first end -> sets the line code
  \addstation{Magoria}
  \addstation{Ildefons Cerdà}
  \addinterchange{Sant Josep} % a correspondence / transfer stop
  \addstation[label position=below]{Gornal}
  \addstation[L'Hospitalet\\de Llobregat]
  \addterminal{Cornellà}{S1} % last end -> closing [S1] plate
\end{trainmap}
\end{document}
```

It produces:



4 Command reference

4.1 The trainmap environment

```
\begin{trainmap}[<keys>] ... \end{trainmap}
```

Wraps a `tikzpicture`, fixes the drawing unit to 1 cm, and resets the internal cursor and trunk state. Any `/trainmap/` key (see Section 5) may be given in `<keys>` as a map-wide default. You may freely add ordinary TikZ code (`\node`, `\draw`, `\coordinate`) inside the environment; that is how the examples place off-trunk terminals.

4.2 Declaring the trunk

```
\settrunk{<y>/<colour>, <y>/<colour>, ...}
```

Declares the parallel tracks that currently make up the common trunk. Each entry is a vertical offset in centimetres and the colour of that track. The tracks are drawn automatically between consecutive stations. `\settrunk` may be **called again at any point** to change the active bundle — for example after a line branches away it can be dropped from the trunk, or a new line added.

4.3 Stations

Command	What it draws
<code>\addstation[<keys>]{<name>}</code>	A standard station : a perpendicular tick crossing every track of the trunk, plus a rotated label.
<code>\addterminal[<keys>]{<name>}{<line>}</code>	A terminal : a filled disc in the badge colour, the name in bold. In <code>fgc</code> mode a badge with <code><line></code> is drawn below the disc; in <code>rodalies</code> mode the code appears on the end plates instead.
<code>\addinterchange[<keys>]{<name>}</code>	An interchange : a rounded capsule grouping all trunk tracks, with a coloured dot per line, and a bold label.

Any `<name>` may contain `\\` to break it across lines. Every station call **advances the cursor** by `station step` centimetres before drawing, and publishes the reached position in the macros `\tmlastx` and `\tmlasty`. Capture them to anchor a branch:

```
\addstation[name=gracia]{Gracia}
\edef\gx{\tmlastx}           % \gx now holds Gracia's x coordinate
```

Passing `name=<id>` additionally creates three reusable coordinates: `(<id>)` at the top track, `(<id>-mid)` at the band centre, and `(<id>-bot)` at the bottom track.

4.4 Tracks, branches and forks

The `<from>` and `<to>` arguments are passed **with their parentheses**, so they accept explicit coordinates, named nodes or `calc` expressions interchangeably.

Command	What it draws
<code>\track[<o>]{<from>}{<to>}{<c>}</code>	A straight track segment.
<code>\branch[<o>]{<from>}{<to>}{<c>}</code>	A gentle, near-horizontal curved branch.
<code>\forkup[<o>]{<from>}{<to>}{<c>}</code>	A branch peeling upwards (alias <code>\forkline</code>).
<code>\forkdown[<o>]{<from>}{<to>}{<c>}</code>	A branch peeling downwards .
<code>\joinline[<o>]{<from>}{<to>}{<c>}</code>	A branch merging back into the trunk.

Here `<c>` is the colour and `<o>` an optional list of raw TikZ options appended to the `\draw` (e.g. `[dashed]`).

5 Configuration keys

All keys live under `/trainmap/`. Set them on the environment for a map-wide default, or in a command's `[...]` to override a single element.

Key	Default	Meaning
<code>mode</code>	<code>fgc</code>	<code>fgc</code> (all lines) or <code>rodalies</code> (single line).
<code>line code</code>	(empty)	Code on the end plates in <code>rodalies</code> mode.
<code>line color</code>	<code>black</code>	Fallback colour of a single track.
<code>track width</code>	<code>2pt</code>	Stroke width of a track.
<code>station color</code>	<code>black</code>	Colour of the station tick mark.
<code>station width</code>	<code>1.8pt</code>	Stroke width of the tick mark.
<code>terminal size</code>	<code>8pt</code>	Diameter of the terminal disc.
<code>terminal border</code>	<code>1.4pt</code>	Border width of the terminal disc.
<code>badge color</code>	<code>black</code>	Background colour of the line badge.
<code>capsule width</code>	<code>12pt</code>	Outer width of the interchange capsule.
<code>capsule inner</code>	<code>8pt</code>	Inner (white) width of the capsule.
<code>capsule color</code>	<code>black</code>	Border colour of the capsule.
<code>station step</code>	<code>1.6</code>	Horizontal distance between stations (cm).
<code>track sep</code>	<code>0.16</code>	Suggested vertical gap between tracks (cm).
<code>max width</code>	<code>0</code>	Wrap to a new row past this x (cm); 0 = off.
<code>row sep</code>	<code>2.6</code>	Vertical distance between wrapped rows (cm).
<code>label angle</code>	<code>45</code>	Rotation of the station labels (degrees).
<code>label position</code>	<code>above</code>	<code>above</code> or <code>below</code> the track.
<code>corr</code>	(empty)	List CODE/colour, ... of correspondence chips.

5.1 Predefined colours

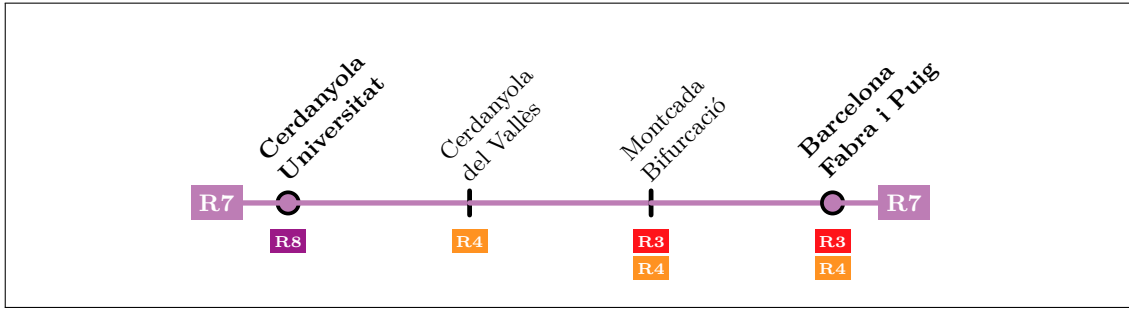
The package defines the official line colours of *Rodalies de Catalunya* (`tmR1`, `tmR2`, ..., `tmR17`, plus `tmRL1`, `tmRL2`, `tmRG1`, `tmRT1`, `tmRT2`) and *FGC* (`tmL6`, `tmL7`, `tmL12`, `tmS1`, `tmS2`, `tmL8`, `tmR5`, `tmR50`, `tmR6`, `tmR60`). You may of course use any `xcolor` colour, such as `red` or `blue!60!black`.

6 Worked examples

6.1 Correspondence chips

Any station accepts a `corr` key listing connecting lines as CODE/colour pairs; each becomes a small coloured chip beneath the station.

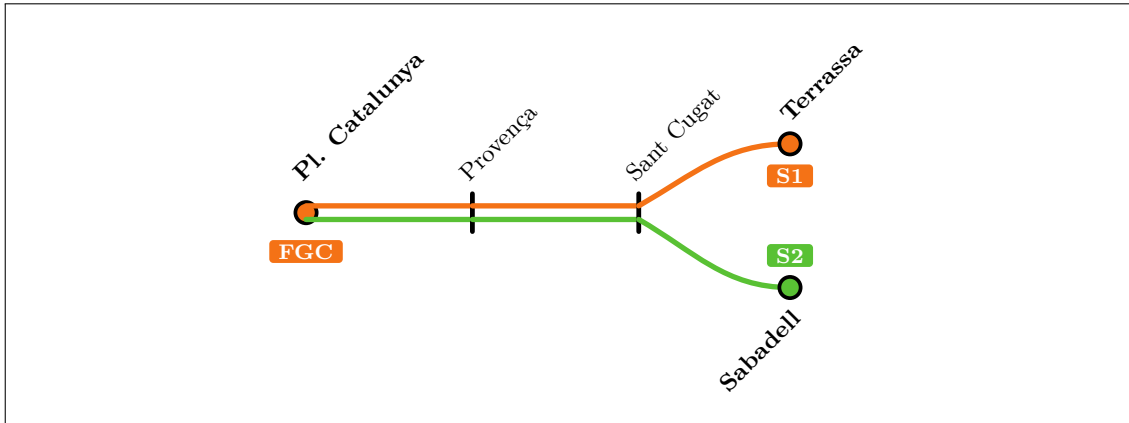
```
\begin{trainmap}[mode=rodalies, station step=2.4, badge color=tmR7]
  \settrunk{0/tmR7}
  \addterminal[corr={R8/tmR8}]{Cerdanyola\\Universitat}{R7}
  \addstation[corr={R4/tmR4}]{Cerdanyola\\del Valles}
  \addstation[corr={R3/tmR3, R4/tmR4}]{Montcada\\Bifurcacio}
  \addterminal[corr={R3/tmR3, R4/tmR4}]{Barcelona\\Fabra i Puig}{R7}
\end{trainmap}
```



6.2 Several lines and a fork (mode=fgc)

In the default `fgc` mode the whole trunk is drawn and each terminal shows its badge. A fork is a pair of curved connectors leaving a captured station position.

```
\begin{trainmap}[mode=fgc, station step=2.2]
  \settrunk{0/tmS2, 0.18/tmS1} % two parallel tracks
  \addterminal[badge color=tmS1]{Pl. Catalunya}{FGC}
  \addstation{Provença}
  \addstation[name=sc]{Sant Cugat}
  \edef\scx{\tmlastx}
  \coordinate (terr) at (\scx+2.0, 1.0);
  \coordinate (saba) at (\scx+2.0,-0.9);
  \forkup {(\scx,0.18)}{(terr)}{tmS1}
  \forkdown{(\scx,0.00)}{(saba)}{tmS2}
  \node[trainmap terminal, fill=tmS1] at (terr) {};
  \node[trainmap terminal, fill=tmS2] at (saba) {};
\end{trainmap}
```



6.3 Long lines: the serpentine layout

When a line has too many stations to fit on one row, enable *serpentine wrapping*: set `max width` (the x at which to wrap) and optionally `row sep`. The cursor turns automatically at the edge, drawing a nested U-turn, and you keep calling `\addstation` as usual. See `examples/rodalies_r4.tex` in the distribution for the full R4 line folded over three rows.

7 Licence

trainmap is distributed under the **L^AT_EX Project Public License**, version 1.3c or (at your option) any later version. The full text is in the file `LICENSE` accompanying the distribution, and

online at <https://www.latex-project.org/lppl.txt>. The package has LPPL maintenance status *maintained*; the Current Maintainer is Pol Sardà Palla.

The project's source repository is at <https://github.com/Anty48/trainmap>.