

handstroke: L^AT_EX Package for Hand-Written Underline*

Yegor Bugayenko
yegor256@gmail.com

2026-07-13, 0.4.0

1 Introduction

A plain, ruler-straight underline blends into the text and is easy to miss. This package draws the line as if it were sketched by hand: it wavers a little, stands apart from the typeset letters, and catches the reader's eye where an ordinary rule would go unnoticed. Reach for the `\handstroke` command whenever a word or a phrase truly deserves attention:

Don't forget to
encrypt the data
before sending them
over the network, it's
important!

```
1 \documentclass{article}
2 \usepackage[paperwidth=14em]{geometry}
3 \usepackage[colored]{handstroke}
4 \pagestyle{empty}
5 \begin{document}\raggedright
6 Don't forget to \handstroke{encrypt}
7 the data before sending them over the
8 network, it's \handstroke{important}!
9 \end{document}
```

`\handstroke` The `\handstroke` command takes a single argument and underlines it, much like the familiar `\u1`. The only difference is the shape of the line: instead of a straight rule, `\handstroke` sketches an uneven handstroke that looks hand-drawn and stands out on the page.

By default the handstroke is drawn in gray. Load the package with the `colored` option to paint it in color instead, and pass `color=<name>` to choose which one (orange by default), as in `\usepackage[colored, color=red]{handstroke}`.

A single stroke may also be recolored on its own, without touching the rest of the document, by giving `\handstroke` an optional `color=<name>` argument. This overrides both the gray default and whatever the package-wide `color` option asked for, but only for that one line:

*The sources are in GitHub at [yegor256/handstroke](https://github.com/yegor256/handstroke)

```

1 \documentclass{article}
2 \usepackage[paperwidth=8em]{geometry}
3 \usepackage{handstroke}
4 \pagestyle{empty}
5 \begin{document}\raggedright
6 This word is
7 \handstroke[color=red]{urgent},
8 while this one is
9 \handstroke[color=blue]{calm}.
10 \end{document}

```

2 Package Options

colored The `colored` package option makes the line be colored, with the help of the [xcolor](#) package.

color The `color` package option (`orange` by default) defines the color of the line.

synonym The `synonym` package option defines a shorter alias for the `\handstroke` command, so that `\usepackage[synonym=hs]{handstroke}` lets you write `\hs{word}` instead of `\handstroke{word}`:

```

1 \documentclass{article}
2 \usepackage[paperwidth=10em]{geometry}
3 \usepackage[synonym=hs]{handstroke}
4 \pagestyle{empty}
5 \begin{document}\raggedright
6 How about \hs{this}?
7 \end{document}

```

The chosen name need not be new. When it already belongs to an existing command, such as `\emph`, that command is renewed to draw a handstroke, rather than aborting with an error.

emph The `emph` package option is a shortcut for `synonym=emph`: it renews the standard `\emph` command so that `\emph{word}` draws a handstroke.

ul The `ul` package option is a shortcut for `synonym=ul`: it lets you write `\ul{word}` in place of `\handstroke{word}`.

They combine freely: `[synonym=hs,ul,emph]` makes `\hs`, `\ul` and `\emph` all draw handstrokes at once.

3 Implementation

First, we parse package options with the help of the [pgfopts](#) package. Every `synonym` appends one more name to `\handstroke@synonyms`, so the options accumulate and several aliases can be requested at once. The `emph` and `ul` options are styles that expand to `synonym=emph` and `synonym=ul`, sharing that same machinery:

```

1 \RequirePackage{pgfopts}
2 \def\handstroke@synonyms{}
3 \pgfkeys{

```

```

4 /handstroke/.cd,
5 colored/.store in=\handstroke@colored,
6 color/.store in=\handstroke@color,
7 color/.default=orange,
8 color,
9 synonym/.code={%
10 \g@addto@macro\handstroke@synonyms{\handstroke@alias{#1}}},
11 emph/.style={synonym=emph},
12 emph/.default=,
13 ul/.style={synonym=ul},
14 ul/.default=,
15 }
16 \ProcessPgfPackageOptions{/handstroke}

```

Then, we load the `xcolor` package, if coloring is required via the `colored` package option:

```

17 \makeatletter
18 \ifdefined\handstroke@colored
19 \RequirePackage{xcolor}
20 \fi
21 \makeatother

```

Then, we load the `tikz` package, together with a few required libraries that provide the coordinate arithmetic and the random-steps decoration used below:

```

22 \RequirePackage{tikz}
23 \usetikzlibrary{calc}
24 \usetikzlibrary{decorations.pathmorphing}

```

\handstroke Finally, we define the `\handstroke` command itself. It typesets its argument inside a `tikz` node and then draws a wavering rectangle path just beneath the baseline, so that only the lower edge shows and reads as an underline. The optional first argument accepts a `color=<name>` key that overrides the ink for this one stroke only:

```

25 \makeatletter
26 \pgfkeys{
27 /handstroke/stroke/.cd,
28 color/.store in=\handstroke@ink,
29 }
30 \newcommand\handstroke[2] [] {%
31 \ifdefined\handstroke@colored%
32 \let\handstroke@ink\handstroke@color%
33 \else\def\handstroke@ink{gray!80}\fi%
34 \pgfkeys{/handstroke/stroke/.cd, #1}%
35 \tikz[baseline=(text.base)] {
36 \node[inner sep=0em, outer sep=0em] (text) {#2};
37 \begin{scope}
38 \path[use as bounding box]
39 (text.east) rectangle (text.west);
40 \draw[decorate,
41 decoration={random steps,
42 amplitude=0.05em, segment length=0.3em},
43 line join=round,
44 line cap=round,
45 line width=.08em,

```

```

46         draw=\handstroke@ink]
47         ($(text.west) + (-0.2em, -.46em)$)
48         rectangle
49         ($(text.east) + (+0.2em, -.46em)$);
50     \end{scope}
51 }%
52 }
53 \makeatother

```

Once `\handstroke` exists, we walk the collected names and let each one alias it. Because `\let` overwrites whatever meaning a name carried before, an existing command such as `\emph` is renewed rather than aborting with an error:

```

54 \makeatletter
55 \newcommand\handstroke@alias[1]{%
56     \expandafter\let\csname#1\endcsname\handstroke}
57 \handstroke@synonyms
58 \makeatother
59 \endinput

```

Change History

v0.1.0	General: Initial version	2	be passed to the <code>\handstroke</code> command to recolor a single line.	2	
v0.2.0	General: The package was renamed from <code>stroke</code> to <code>handstroke</code> for the sake of disambiguation and better clarity.	2	v0.4.0	General: The <code>synonym</code> package option defines a shorter alias for the <code>\handstroke</code> command, renewing the target when it already exists, such as <code>\emph.</code>	2
v0.3.0	General: The <code>color</code> option can now				

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

B		G		N	
<code>\begin</code>	<i>37</i>	<code>\g@addto@macro</code>	<i>10</i>	<code>\node</code>	<i>36</i>
C		H		P	
<code>\color</code>	<i>2</i>	<code>\handstroke</code> ...	<i>1, 25, 56</i>	<code>\path</code>	<i>38</i>
<code>\colored</code>	<i>2</i>	<code>\handstroke@alias</code>	<i>10, 55</i>	<code>\pgfkeys</code>	<i>3, 26, 34</i>
<code>\csname</code>	<i>56</i>	<code>\handstroke@color</code> .	<i>6, 32</i>	R	
D		<code>\handstroke@colored</code>		<code>\RequirePackage</code>	<i>1, 19, 22</i>
<code>\def</code>	<i>2, 33</i>		<i>5, 18, 31</i>	S	
<code>\draw</code>	<i>40</i>	<code>\handstroke@ink</code> ...		<code>\synonym</code>	<i>2</i>
E			<i>28, 32, 33, 46</i>		
<code>\emph</code>	<i>2</i>	<code>\handstroke@synonyms</code>			
<code>\end</code>	<i>50</i>		<i>2, 10, 57</i>		
<code>\endcsname</code>	<i>56</i>	I		T	
<code>\endinput</code>	<i>59</i>	<code>\ifdefined</code>	<i>18, 31</i>	<code>\tikz</code>	<i>35</i>
<code>\expandafter</code>	<i>56</i>	M		U	
		<code>\makeatletter</code>	<i>17, 25, 54</i>	<code>\ul</code>	<i>2</i>
		<code>\makeatother</code> .	<i>21, 53, 58</i>	<code>\usetikzlibrary</code> .	<i>23, 24</i>