The **Engtlc** package

Version 3.1

Claudio Fiandrino

February 18, 2012

Contents

Chapter 1

Introduction

This package has been realized in four different periods: in the first one I wrote the macros for measurement units and general symbols; in the second one I extended the units part and I added some other symbols such as the impedance and admittance ones. In the third version, in which Alessio Sanna gave me precious suggestions, I inserted byte and chip units and the symbols for electrical and magnetic fields. I finished it on 18/12/2009 and on 13/01/2010 it had been published on CTAN.

In that third version there were some mistakes due of my inexperience in LaTeX programming. Thanks to Enrico Gregorio, who send me a list of them, I decided to correct them in a new version and added some new symbol commands.

This fourth version, thanks to Claudio Beccari for the precious help, complies to ISO regulations and introduces *alias* commands in english, as explained in section 3.2. Moreover some new commands have been added: probability symbols, commands to define signals in temporal, frequency, discrete and z-transform domain are the major novelties.

The purpose to create engtlc is very simple: it can be used by all those who work in electrical engineering or telecommunication area; indeed, engtlc is the acronym of *Engineering Telecommunications*.

Where does it help? It helps in writing a document with LaTeX, of course; in my personal experience I observed that LaTeX code isn't very user friendly when you have to repeat over and over again some words where the code is a little bit different every time.

If you have to write the probability of x you can write the LaTeX code like:

```
$\mathcal{P}(x)$
```

$\mathcal{P}(x)$

If, after some time, you have to write the probability of the event A:

```
$\mathcal{P}(\text{A})$
```

$\mathcal{P}(\text{A})$

or you can copy the previous code and change the argument.

With engtlc it is easier; you simply write:

`$\prob{`⟨*argument*⟩`}$`

where ⟨*argument*⟩ is the argument you have to put like x or A.

The following chapters are structured in this way:

- ⋆ the procedures to install the package and its dependencies are explained in chapter 2;
- ⋆ the specific new commands introduced by engtlc are shown in chapter 3.

Chapter 2

How to install **engtlc**

If it is not already intalled in your TEX distribution, you can find the package in my website http://claudiofiandrino.altervista.org within the works section; alternatively you can download it from the official LATEX packages website http://tug.ctan.org/tex-archive/macros/latex/contrib/engtlc.

After downloading the *.zip* file, decompress it and move all the files into the `/texmf/tex/latex/engtlc/` in your home directory (if you don't have a personal `texmf` tree in your home directory, create one). After moving the unzipped files in the aforementioned folder, rembeber to refresh the filename database if your TEX distribution and operating system require it.

The home folder, shown with `~`, is a normal way of indicating it on Linux platforms. On Macs the `texmf` personal tree should be rooted in `~/Library`; on Windows platforms, from Vista and newer releases, the concept of "home" is now well known; for previous version you should root your personal `texmf` tree somewhere such as `C:\Documents and Settings\`⟨*your username*⟩.

Remember: This package is dependent on some external packages such as textcomp, amsmath, amssymb. The package does not expressly load them in order to avoid conflicts among different sets of options the user might want to specify. Therefore the user is responsable for loading them with the options s/he likes best. Since these packages are generally loaded by the user for other purposes, this should not be an inconvenience. Anyway, if you do not load amsmath and amssymb, you will receive an error: this because some commands need this packages. On the contrary, if you do not load textcomp, you will see a warning.

Chapter 3

Commands introduced by engtlc

Let's start to examine engtlc commands.

In the first section several commands for writing units of measurement are shown, while in the second section several new symbol commands are listed.

3.1 Units of measurement

I believe that the possibility of introducing units of measurement in a short and consistent way might be very useful. Of course there are other packages that allow to do the same, but possibly with longer constructs. For example with siunuitx if you have to use "kbit/s" you have to type in:

Code	Visualization
`\si{\kibi\bit\per\second}`	Kib/s

So I decided to create shorter codes that can replace the longer ones shown above.

Engtlc units of measurement must used only within math environments `$ $`, `[ ]` or within any kind *equation like* ones.

Be careful: if in your text you use a unit command and forget that the space after a control sequence is absorbed by the control sequence itself, you observe this mistake:

Code	Visualization
`8 \cm and`	8 cmand

Time units

Unit	Equivalent code	Visualization
hours	`\ho`	h
seconds	`\s`	s
milliseconds	`\ms`	ms
microseconds	`\us`	µs
nanoseconds	`\ns`	ns
picoseconds	`\ps`	ps

Length units

Unit	Equivalent code	Visualization
micrometres	`\um`	µm
millimetres	`\mm`	mm
centimetres	`\cm`	cm
decimetres	`\dm`	dm
metres	`\m`	m
kilometres	`\km`	km

Current units

Unit	Equivalent code	Visualization
microampere	`\uA`	µA
milliampere	`\mA`	mA
ampere	`\A`	A

Voltage units

Unit	Equivalent code	Visualization
microvolt	`\uV`	µV
millivolt	`\mV`	mV
volt	`\V`	V
megavolt	`\MV`	MV

Resistance units

Unit	Equivalent code	Visualization
milliohm	`\mohm`	mΩ
ohm	`\ohm`	Ω
kilohm	`\kohm`	kΩ
megaohm	`\Mohm`	MΩ

Conductance units

Unit	Equivalent code	Visualization
picosiemens	`\pSi`	pS
nanosiemens	`\nSi`	nS
microsiemens	`\uSi`	µS
millisiemens	`\mSi`	mS
siemens	`\Si`	S
kilosiemens	`\kSi`	kS
megasiemens	`\MSi`	MS

Capacity units

Unit	Equivalent code	Visualization
femtofarad	`\fFa`	fF
picofarad	`\pFa`	pF
nanofarad	`\nFa`	nF
microfarad	`\uFa`	µF
millifarad	`\mFa`	mF
farad	`\Fa`	F

Inductance units

Unit	Equivalent code	Visualization
femtohenry	`\fHe`	fH
picohenry	`\pHe`	pH
nanohenry	`\nHe`	nH
microhenry	`\uHe`	µH
millihenry	`\mHe`	mH
henry	`\He`	H

Level units

Unit	Equivalent code	Visualization
dB	`\dB`	dB
dBm	`\dBm`	dBm

Power units

Unit	Equivalent code	Visualization
microwatt	`\uW`	µW
milliwatt	`\mW`	mW
watt	`\W`	W
kilowatt	`\kW`	kW
megawatt	`\MW`	MW

Frequency units

Unit	Equivalent code	Visualization
hertz	`\Hz`	Hz
kilohertz	`\kHz`	kHz
megahertz	`\MHz`	MHz
gigaahertz	`\GHz`	GHz
terahertz	`\THz`	THz

Bit, byte, and chip units according to the ISO regulations

Unit	Equivalent code	Visualization
bit	`\bit`	bit
kibibit	`\kbit`	Kib
mebibit	`\Mbit`	Mib
byte	`\Byte`	B
kibibyte	`\kByte`	KiB
mebibyte	`\MByte`	Mib
gibibyte	`\GByte`	GiB
tebibyte	`\TByte`	TiB
bit per second	`\bits`	bit/s
kibibit per second	`\kbits`	Kib/s
mebibit per second	`\Mbits`	Mib/s
byte per second	`\Bytes`	B/s
kibibyte per second	`\kBytes`	KiB/s
mebibyte per second	`\MBytes`	MiB/s
gibibyte per second	`\GBytes`	GiB/s
tebibyte per second	`\TBytes`	TiB/s
chip per second	`\chips`	chip/s
kibichip per second	`\kchips`	Kichip/s
mebichip per second	`\Mchips`	Michip/s
chip su bit per second	`\chipsubit`	chip/bit

3.2 Symbols

Several symbol macros are listed in this section. All these macros are supposed to be used only in math mode. Each macro has an English and an Italian contracted name and the examples will show both of them: in the first line there will be the English version and in the second line the Italian one; if only one line is shown, there is no difference between the English and the Italian names.

3.2.1 General symbols

End of exercise

The command of end exercise puts a black square flush to the right side of the text block.

Code	Visualization
`\exerend`	■
`\finees`	■

Spaced "implies" command

This command is very similar to `\implies`, but it puts before and after the symbol a space that can be chosen by the user by means of an optional argument; by default this space is 0.5 cm; you can change the unit of measure for the spacing by assigning a value to the length register `\Implspace`, for example:

```
\setlength{\Implspace}{3mm}
```

The syntax is therefore:

`\Spimplies[`⟨*optional space in units of* `\Implspace`⟩`]`
`\frecciadex[`⟨*optional space in units of* `\Implspace`⟩`]`

In the following table, keeping in mind that `\Spimpiles` and `\frecciadex` are synonymous, in the first line the optional argument has been used, while in the second line the default spacing has been used, so as to evaluate the difference; in both cases the length register `\Implspace` was not reset and contains the value of 0.5 cm.

Code	Visualization
`A\Spimplies[0.3] B`	$A \Longrightarrow B$
`A\frecciadex B`	$A \Longrightarrow B$

Notice that, by inserting in the optional argument the default length, you will obtain the same result:

Codice	Visualizzazione
`A\Spimplies B`	$A \Longrightarrow B$
`A\frecciadex[0.5] B`	$A \Longrightarrow B$

Vertical "implies" command

Code	Visualization
`\Downimplies`	$\Downarrow$
`\frecciadown`	$\Downarrow$

White noise command

Code	Visualization
`\noisevar`	$\frac{N_0}{2}$
`\varianzarumore`	$\frac{N_0}{2}$

Fourier Transform

Command for the Fourier transform of x.

Code	Visualization
`\fourier{x}`	$\mathcal{F}\{x\}$

Inverse Fourier Transform

This command is very similar to the previous one.

Code	Visualization
`\invfourier{x}`	$\mathcal{F}^{-1}\{x\}$

Real part

Code	Visualization
`\bfRe{x}`	$\mathbf{Re}\{x\}$
`\partereale{x}`	$\mathbf{Re}\{x\}$

Imaginary part

Code	Visualization
`\bfIm{x}`	$\mathbf{Im}\{x\}$
`\parteimm{x}`	$\mathbf{Im}\{x\}$

Information element

Code	Visualization
`\Info{x}`	$I\left(x\right)$

Signals in different domains

The following four macros define signal functions, by using as a convention lower case letters for signals in the time domain and for discrete sequences while signals in frequency and z-transform domains have automatically the upper case letter.

Code	Visualization
`\signt{f}`	$f(t)$
`\signf{g}`	$G(f)$
`\signn{h}`	$h(n)$
`\signz{k}`	$K(z)$

Analytic signal

Code	Visualization
`\analytic{x}`	$\mathring{x}$
`\analitic{x}`	$\mathring{x}$

Unit vector

Code	Visualization
`\unitvec{x}`	$\hat{x}$
`\versore{x}`	$\hat{x}$

Vector

Code	Visualization
`\vector{x}`	$\overrightarrow{x}$
`\vettore{x}`	$\overrightarrow{x}$

Cosine wave of a given frequency

Code	Visualization
`\cosine{f_0}`	$\cos\left(2\pi f_0 t\right)$
`\coseno{f_0}`	$\cos\left(2\pi f_0 t\right)$

Sine wave of a given frequency

Code	Visualization
`\sine{f_0}`	$\sin(2\pi f_0 t)$
`\seno{f_0}`	$\sin(2\pi f_0 t)$

Energy

Code	Visualization
`\energy{m}`	$\mathcal{E}_m$
`\energia{m}`	$\mathcal{E}_m$

Module

Code	Visualization
`\Abs{x}`	$\lvert x \rvert$
`\modulo{x}`	$\lvert x \rvert$

Exponential with ISO compliant natural base

The ISO regulations require that the base of natural logarithms and of the exponential function be set in roman type.

Code	Visualization
`\rmexp{x}`	e^{x}
`\ex{x}`	e^{x}

ISO compliant imaginary unit, engineering style

Similarly the ISO regulations require the imaginary unit to be typeset in roman type; electrical and telecommunications engineers use the letter "j" in order to avoid any possible confusion with the standard current symbol i; the roman type avoids also any possible confusion with the standard symbols for the current density j.

Code	Visualization
`\iu\omega`	$\mathrm{j}\omega$

Module with exponent

Code	Visualization
`\AbsPow{x}{2}`	$\lvert x \rvert^2$
`\moduloexp{x}{2}`	$\lvert x \rvert^2$

Function value for a certain value of the independent variable

Code	Visualization
`\for{f(x)}{x_0}`	$f(x)\big\vert_{x_0}$

A ratio in dB

A special application of the previous command `\for`.

Code	Visualization
`\indB{\dfrac{C}{I}}`	$\left.\dfrac{C}{I}\right\vert_{\mathrm{dB}}$

Maximum

Code	Visualization
`\Max{x}`	$\max\{x\}$
`\massimo{x}`	$\max\{x\}$

Minimum

Code	Visualization
`\Min{x}`	$\min\{x\}$
`\minimo{x}`	$\min\{x\}$

Speed of light

Code	Visualization
`\clight`	$3 \cdot 10^8$
`\valc`	$3 \cdot 10^8$

Logarithm with specified base

Code	Visualization
`\Log{2}{x}`	$\log_2 x$
`\loga{2}{x}`	$\log_2 x$

Integral

An integral on the whole real domain, from $-\infty$ to $+\infty$:

Code	Visualization
`\infint{x\diff x}`	$\int_{-\infty}^{+\infty} x \,\mathrm{d}x$
`\intinf{x\diff x}`	$\int_{-\infty}^{+\infty} x \,\mathrm{d}x$

Notice the use of the differential sign `\diff` that complies with the ISO regulations that require the differential sign be set in roman type with the proper spacings on the left and on the right.

Dirac's Delta with independent variable

Code	Visualization
`\deltain{x}`	$\delta\left(x\right)$

3.2.2 Reflectances

The reflectance symbols used in electrical and magnetic field theory courses are described; you have two kind of commands, the generic ones, and the specific ones where you can specify the point of measure.

Generic reflectances

Code	Visualization
`\Vgamma`	${}^{\mathrm{V}}\Gamma$
`\gammatens`	${}^{\mathrm{V}}\Gamma$
`\Cgamma`	${}^{\mathrm{I}}\Gamma$
`\gammacorr`	${}^{\mathrm{I}}\Gamma$

Specific reflectances

Code	Visualization
`\Vgammain{A}`	${}^{\mathrm{V}}\Gamma_{\mathrm{A}}$
`\gammatensin{A}`	${}^{\mathrm{V}}\Gamma_{\mathrm{A}}$
`\Cgammain{A}`	${}^{\mathrm{I}}\Gamma_{\mathrm{A}}$
`\gammacorrin{A}`	${}^{\mathrm{I}}\Gamma_{\mathrm{A}}$

Since the voltage reflectance (gamma) is used more often than the current one, it may be input with a shorter command:

Code	Visualization
`\gammain{A}`	Γ_{A}

The Kurokawa reflectance

Code	Visualization
`\gammak`	${}^{\mathrm{k}}\Gamma$

3.2.3 Usage examples

Here are some examples that clearly show the usefulness of engtlc:

1. Code
   `\AbsPow{\gammak}{2}`
   Visualization
   $\left|{}^{\mathrm{k}}\Gamma\right|^2$

2. Code
   `\bfRe{\fourier{\AbsPow{x}{2}}}`
   Visualization
   $\mathbf{Re}\left\{\mathcal{F}\left\{|x|^2\right\}\right\}$

   The standard code to write the above expression is:

```
$\textbf{Re}\left\lbrace \mathcal{F}\left\lbrace
\left\vert x \right\vert^{2} \right\rbrace \right\rbrace$
```

3.2.4 Wave length symbols

There are three different types of symbols:

In the open space

Code	Visualization
`\lbvt`	λ_0

In a guide filled with generic dielectric

Code	Visualization
`\lbg`	λ_g

In an empty guide

Code	Visualization
`\lbgvt`	λ_{g_0}

3.2.5 Impedance and admittance symbols

With engtlc you can also write any kind of impedance and admittance.

Generic impedance and admittance

To input an impedance or admittance at a specific port or line section A use:

Code	Visualization
`\z{A}`	Z_{A}
`\y{A}`	Y_{A}

To input a normalized impedance or admittance at a specific port or line section A use:

Code	Visualization
`\znorm{A}`	z_{A}
`\ynorm{A}`	y_{A}

Characteristic impedance and admittance

To input the symbol for characteristic impedance or admittance use:

Code	Visualization
`\zinf`	Z_{∞}
`\yinf`	Y_{∞}

In a guide there are several mode dependent characteristic impendances: you may use another command in order to distinguish them; for example if you want to label with 2 a second impedance you may use:

Code	Visualization
`\zinf[2]`	$Z_{\infty 2}$
`\zinfn{2}`	$Z_{\infty 2}$
`\yinf[2]`	$Y_{\infty 2}$
`\yinfn{2}`	$Y_{\infty 2}$

To avoid confusion pay attention when you input a numbered impedance and you use the `\zinf` or `\yinf` commands: you must use either the optional argument command or the n-extended command:

Impedance with optional argument	Visualization
`\zinf[2]`	$Z_{\infty 2}$
n-extended impedance Code	Visualization
`\zinfn{2}`	$Z_{\infty 2}$
Generic impedance Code	Visualization
`\zinf{2}`	$Z_{\infty}2$

The last code is not wrong under the LaTeX point of view, but probably the output is not the expected one...

The open space impedance and admittance is input by means of:

Code	Visualization
`\zvt`	Z_0
`\yvt`	Y_0

3.2.6 Some more usage examples

With engtlc commands the available power can be written with the code:

```
\[\availpow=\frac{\AbsPow{V}{2}}{\bfRe{4\cdot\z{G}}}\]
```

that yields:

$$P_{\mathrm{disp}} = \frac{|V|^2}{4 \cdot \mathbf{Re}\{Z_{\mathrm{G}}\}}$$

Without the engtlc commands the code would be:

```
\[P_{\mathrm{disp}}=\frac{\left| V \right|^2}%
{ 4\cdot\textbf{Re}\left\{ Z_\mathrm{G}\right\}}\]
```

3.2.7 Power symbols

Here we have the commands to input power symbols; the following commands may be used in both text and math modes.

Power into a port

To write the power entering into a specific port or line/guide section A use:

Code	Visualization
`\powerin{A}`	P_{A}
`\potin{A}`	P_{A}

Available Power

Code	Visualization
`\availpow`	P_{disp}
`\potdisp`	P_{disp}

Power supply

Code	Visualization
`\potDC`	P_{DC}
`\potCC`	P_{CC}

For compatibility reasons, there is also the possibility of exploit `\potalim` the old name for the symbol.

Radiation Power

Code	Visualization
`\irrpow`	P_{irr}
`\potirr`	P_{irr}

Dissipated Power

Code	Visualization
`\disspow`	P_{diss}
`\potdiss`	P_{diss}

Incident power

Code	Visualization
`\incpow`	P_{inc}
`\potinc`	P_{inc}

Pay attention that, in this version, all shortcuts describing power symbols, a part from the first one `\powerin{A}`, have an optional argument: it is used to indicate explicitly the point in which you can measure the power. To describe, for example, the dissipated power for a general transmitting device, you should use `\disspow[tx]` which give as a result P_{diss}^{tx}. Be careful: if you put { } instead of [ ] you obtain a different, undesired result. Indeed, `\irrpow{rx}` becomes $P_{\mathrm{irr}}rx$.

3.2.8 Electrical and magnetic field symbols

With regards to electrical and magnetic fields, engtlc commands are not so well established, as they depend on every particular school, whether they are typeset or handwritten, etcetera. For informal handouts the underlined vector convention is often used. We distinguish:

- ⋆ electric and magnetic field as a function of a vector position $\overrightarrow{r}$ and time;

⋆ electric and magnetic field phasors as a function of a vector position $\overrightarrow{r}$.

In several electric and magnetic fields courses in order to denote A as a vector, instead of using arrows "accents" or bold symbols, the convention $\underline{A}$ is used:

Time dependent fields		Field phasors	
Code	Visualization	Code	Visualization
`\Efield`	$\underline{\mathcal{E}}(\underline{r},t)$	`\phasorEfield`	$\underline{E}(\underline{r})$
`\campoe`	$\underline{\mathcal{E}}(\underline{r},t)$	`\campoefas`	$\underline{E}(\underline{r})$
`\Hfield`	$\underline{\mathcal{H}}(\underline{r},t)$	`\phasorHfield`	$\underline{H}(\underline{r})$
`\campoh`	$\underline{\mathcal{H}}(\underline{r},t)$	`\campohfas`	$\underline{H}(\underline{r})$

Pay attention: we distinguish the symbols of energy `\energy{`⟨*argument*⟩`}` and electric field `\campoe` because the second one is a vector and it does not have subscript symbols.

3.2.9 Probability symbols

Probability

The probability of event A, or of variable x, is input as:

Code	Visualization
`\prob{\text{A}}`	$\mathcal{P}\left(\mathrm{A}\right)$
`\prob{x}`	$\mathcal{P}\left(x\right)$

Expected value

Code	Visualization
`\expval{x}`	$\mathbb{E}\left[x\right]$
`\valatt{x}`	$\mathbb{E}\left[x\right]$

Variance

Code	Visualization
`\var{x}`	$\mathrm{Var}\left[x\right]$

Joint probability

This macro inserts a well separated comma between two stochastic variables:

Code	Visualization
`\comma`	$\mathcal{P}\left(x\,,\,y\right)$

Conditional probability

These macros insert a vertical bar separator between two stochastic variables:

Code	Visualization
`\given`	$\mathcal{P}\left(x \,|\, y\right)$
`\dato`	$\mathcal{P}\left(x \,|\, y\right)$

Aknowledgements

I wish to thank prof. Enrico Gregorio, who very kindly sent me a complete list of errors in my third package edition and documentation.

I wish to thank also prof. Claudio Beccari for many useful suggestions and the adaptation of several commands to the ISO regulations.

Chapter 4

LPPL Licence

engtlc is distributed under LPPL Licence: LaTeX Project Public Licence.

```
%% engtlc.sty
%% Copyright 2010-2012 Claudio Fiandrino
%
%  This work may be distributed and/or modified under the
%  conditions of the LaTeX Project Public License, either version 1.3
%  of this license or (at your option) any later version.
%  The latest version of this license is in
%  http://www.latex-project.org/lppl.txt
%  and version 1.3 or later is part of all distributions of LaTeX
%  version 2005/12/01 or later.
%
%  This work has the LPPL maintenance status ‘maintained’.
%
%  The Current Maintainer of this work is Fiandrino Claudio.
%
%  This work consists of the file engtlc.sty.
```