CircuiTikZ

version 0.6 (2016/06/06)

Massimo A. Redaelli (m.redaelli@gmail.com)
Stefan Lindner (stefan.lindner@fau.de)
Stefan Erhardt (stefan.erhardt@fau.de)

June 6, 2016

Contents

1 Introduction

1.1 About

CircuiTi*k*Z was initiated by Massimo Redaelli in 2007, who was working as a research assistant at the Polytechnic University of Milan, Italy, and needed a tool for creating exercises and exams. After he left University in 2010 the development of CircuiTi*k*Z slowed down, since LaTeX is mainly established in the academic world. In 2015 Stefan Lindner and Stefan Erhardt, both working as research assistants at the University of Erlangen-Nürnberg, Germany, joined the team and now maintain the project together with the initial author.

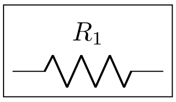
The use of CircuiTi*k*Z is, of course, not limited to academic teaching. The package gets widely used by engineers for typesetting electronic circuits for articles and publications all over the world.

This documentation is somewhat scant. Hopefully the authors will find the leisure to improve it some day.

1.2 Loading the package

LaTeX	ConTeXt[1]
`\usepackage{circuitikz}`	`\usemodule[circuitikz]`

TikZ will be automatically loaded.
CircuiTikZ commands are just TikZ commands, so a minimum usage example would be:




```
\tikz \draw (0,0) to[R=$R_1$] (2,0);
```

1.3 Requirements

- `tikz`, version ≥ 2;
- `xstring`, not older than 2009/03/13;
- `siunitx`, if using `siunitx` option.

1.4 Incompatible packages

TikZ's own `circuit` library, which is based on CircuiTikZ, (re?)defines several styles used by this library. In order to have them work together you can use the `compatibility` package option, which basically prefixes the names of all CircuiTikZ `to[]` styles with an asterisk.

So, if loaded with said option, one must write `(0,0) to[*R] (2,0)` and, for transistors on a path, `(0,0) to[*Tnmos] (2,0)`, and so on (but `(0,0) node[nmos] {}`). See example at page 55.

1.5 License

Copyright © 2007–2016 Massimo Redaelli. This package is author-maintained. Permission is granted to copy, distribute and/or modify this software under the terms of the LaTeX Project Public License, version 1.3.1, or the GNU Public License. This software is provided 'as is', without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

1.6 Feedback

The easiest way to contact the authors is via the official Github repository: `https://github.com/circuitikz/circuitikz/issues`

2 Incompabilities between version

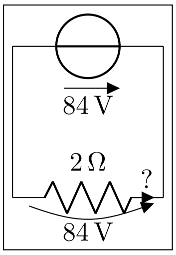
Here, we will provide a list of incompabilitys between different version of circuitikz. We will try to hold this list short, but sometimes it is easier to break with old syntax than including a lot of switches and compatibility layers.

- Since v0.5.1: The parts pfet,pigfete,pigfetebulk and pigfetd are now mirrored by default. Please adjust your yscale-option to correct this.
- Since v0.5: New voltage counting direction, here exists an option to use the old behaviour

[1] ConTeXt suppurt was added mostly thanks to Mojca Miklavec and Aditya Mahajan.

3 Package options

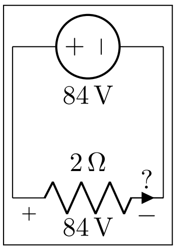
Circuit people are very opinionated about their symbols. In order to meet the individual gusto you can set a bunch of package options. The standard options are what the authors like, for example you get this:




```
\begin{circuitikz}
  \draw (0,0) to[R=2<\ohm>, i=?, v=84<\volt>] (2,0) --
     (2,2) to[V<=84<\volt>] (0,2)
     -- (0,0);
\end{circuitikz}
```

Feel free to load the package with your own cultural options:

LaTeX	ConTeXt
`\usepackage[american]{circuitikz}`	`\usemodule[circuitikz][american]`




```
\begin{circuitikz}
  \draw (0,0) to[R=2<\ohm>, i=?, v=84<\volt>] (2,0) --
     (2,2) to[V<=84<\volt>] (0,2)
     -- (0,0);
\end{circuitikz}
```

Here is the list of all the options:

- `europeanvoltages`: uses arrows to define voltages, and uses european-style voltage sources;
- `americanvoltages`: uses − and + to define voltages, and uses american-style voltage sources;
- `europeancurrents`: uses european-style current sources;
- `americancurrents`: uses american-style current sources;
- `europeanresistors`: uses rectangular empty shape for resistors, as per european standards;
- `americanresistors`: uses zig-zag shape for resistors, as per american standards;
- `europeaninductors`: uses rectangular filled shape for inductors, as per european standards;
- `americaninductors`: uses ”4-bumps” shape for inductors, as per american standards;
- `cuteinductors`: uses my personal favorite, ”pig-tailed” shape for inductors;
- `americanports`: uses triangular logic ports, as per american standards;
- `europeanports`: uses rectangular logic ports, as per european standards;
- `americangfsurgearrester`: uses round gas filled surge arresters, as per american standards;
- `europeangfsurgearrester`: uses rectangular gas filled surge arresters, as per european standards;
- `european`: equivalent to `europeancurrents`, `europeanvoltages`, `europeanresistors`, `europeaninductors`, `europeanports`, `europeangfsurgearrester`;

- `american`: equivalent to `americancurrents`, `americanvoltages`, `americanresistors`, `americaninductors`, `americanports`, `americangfsurgearrester`;
- `siunitx`: integrates with `SIunitx` package. If labels, currents or voltages are of the form `#1<#2>` then what is shown is actually `\SI{#1}{#2}`;
- `nosiunitx`: labels are not interpreted as above;
- `fulldiode`: the various diodes are drawn *and* filled by default, i.e. when using styles such as `diode`, `D`, `sD`, …Other diode styles can always be forced with e.g. `Do`, `D-`, …
- `strokediode`: the various diodes are drawn *and* stroke by default, i.e. when using styles such as `diode`, `D`, `sD`, …Other diode styles can always be forced with e.g. `Do`, `D*`, …
- `emptydiode`: the various diodes are drawn *but not* filled by default, i.e. when using styles such as `D`, `sD`, …Other diode styles can always be forced with e.g. `Do`, `D-`, …
- `arrowmos`: pmos and nmos have arrows analogous to those of pnp and npn transistors;
- `noarrowmos`: pmos and nmos do not have arrows analogous to those of pnp and npn transistors;
- `fetbodydiode`: draw the body diode of a FET;
- `nofetbodydiode`: do not draw the body diode of a FET;
- `fetsolderdot`: draw solderdot at bulk-source junction of some transistors;
- `nofetsolderdot`: do not draw solderdot at bulk-source junction of some transistors;
- `emptypmoscircle`: the circle at the gate of a pmos transistor gets not filled;
- `lazymos`: draws lazy nmos and pmos transistors. Chip designers with huge circuits prefer this notation;
- `straightlabels`: labels on bipoles are always printed straight up, i.e. with horizontal baseline;
- `rotatelabels`: labels on bipoles are always printed aligned along the bipole;
- `smartlabels`: labels on bipoles are rotated along the bipoles, unless the rotation is very close to multiples of 90°;
- `compatibility`: makes it possibile to load CircuiTi*k*Z and Ti*k*Z circuit library together.
- `oldvoltagedirection`: Use old(erronous) way of voltage direction having a difference between european and american direction
- `betterproportions`[2]: nicer proportions of transistors in comparision to resistors;

The old options in the singular (like `american voltage`) are still available for compatibility, but are discouraged.

Loading the package with no options is equivalent to my own personal liking, that is to the following options:
`[nofetsolderdot,nooldvoltagedirection,europeancurrents,europeanvoltages,americanports,americanres`

In ConTEXt the options are similarly specified: `current=european|american`, `voltage=european|american`, `resistor=american|european`, `inductor=cute|american|european`, `logic=american|european`, `siunitx=true|false`, `arrowmos=false|true`.

[2]May change in the future!

4 The components

Here follows the list of all the shapes defined by CircuiTi*k*Z. These are all `pgf` nodes, so they are usable in both `pgf` and Ti*k*Z.

Drawing normal components

Normal components (monopoles, multipoles) can be drawn at a specified point with this syntax, where `#1` is the name of the component:

```
\begin{center}\begin{circuitikz} \draw
   (0,0) node[#1,#2] (#3) {#4}
; \end{circuitikz} \end{center}
```

Explanation of the parameters:
`#1`: component name[3] (mandatory)
`#2`: list of comma separated options (optional)
`#3`: name of an anchor (optional)
`#4`: text written to the text anchor of the component (optional)

> **Note for Ti*k*Z newbies:** Nodes must have curly brackets at the end, even when empty. An optional anchor (`#3`) can be defined within round brackets to be addressed again later on. And please don't forget the semicolon to terminate the `\draw` command.

Drawing bipoles/two-ports

Bipoles/Two-ports (plus some special components) can be drawn between two points using the following command:

```
\begin{center}\begin{circuitikz} \draw
   (0,0) to[#1,#2] (2,0)
; \end{circuitikz} \end{center}
```

Explanation of the parameters:
`#1`: component name (mandatory)
`#2`: list of comma separated options (optional)
Transistors and some other components can also be placed using the syntax for bipoles. See section 6.6.

> If using the `\tikzexternalize` feature, as of Ti*k*z 2.1 all pictures must end with `\end{tikzpicture}`. Thus you *cannot* use the `circuitikz` environment.
>
> Which is ok: just use the environment `tikzpicture`: everything will work there just fine.

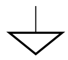
4.1 Monopoles

- Ground (node[`ground`]{})

- Reference ground (node[`rground`]{})

[3]For using bipoles as nodes, the name of the node is `#1shape`.

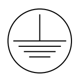
- Signal ground (node[sground]{})



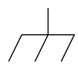
- Thicker ground (node[tground]{})

- Noiseless ground (node[nground]{})

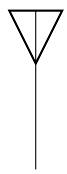
- Protective ground (node[pground]{})



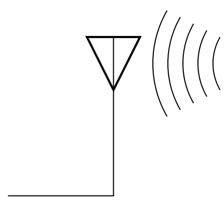
- Chassis ground[4] (node[cground]{})



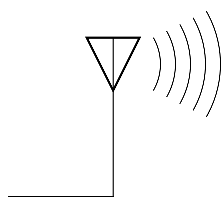
- Antenna (node[antenna]{})



- Receiving antenna (node[rxantenna]{})

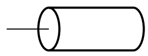


- Transmitting antenna (node[txantenna]{})



[4]These last three were contributed by Luigi «Liverpool»)

- Transmission line stub (node[`tlinestub`]{})



- VCC/VDD (node[`vcc`]{})

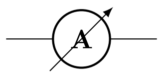
- VEE/VSS (node[`vee`]{})

- match (node[`match`]{})

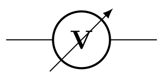
4.2 Bipoles

4.2.1 Instruments

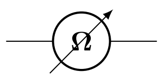
- Ammeter (`ammeter`)



- Voltmeter (`voltmeter`)



- Ohmmeter (`ohmmeter`)

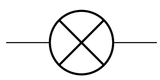


4.2.2 Basic resistive bipoles

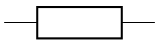
- Short circuit (`short`)

- Open circuit (`open`)

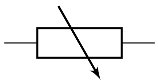
- Lamp (`lamp`)



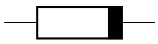
- Generic (symmetric) bipole (`generic`)



- Tunable generic bipole (`tgeneric`)

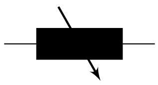


- Generic asymmetric bipole (`ageneric`)

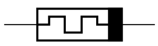


- Generic asymmetric bipole (full) (`fullgeneric`)

- Tunable generic bipole (full) (`tfullgeneric`)



- Memristor (`memristor`, or `Mr`)



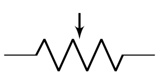
4.2.3 Resistors and the like

If (default behaviour) `americanresistors` option is active (or the style `[american resistors]` is used), the resistor is displayed as follows:

- Resistor (`R`, or `american resistor`)

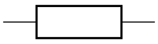
- Variable resistor (`vR`, or `variable american resistor`)

- Potentiometer (`pR`, or `american potentiometer`)

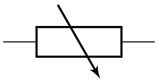


If instead `europeanresistors` option is active (or the style `[european resistors]` is used), the resistors, variable resistors and potentiometers are displayed as follows:

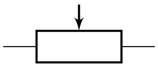
- Resistor (`R`, or `european resistor`)



- Variable resistor (`vR`, or `european variable resistor`)

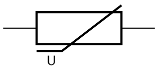


- Potentiometer (`pR`, or `european potentiometer`)



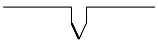
Other miscellaneous resistor-like devices:

- Varistor (`varistor`)

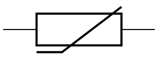


- Photoresistor (`phR`, or `photoresistor`)

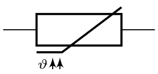
- Thermocouple (`thermocouple`)



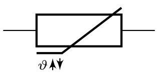
- Thermistor (`thR`, or `thermistor`)



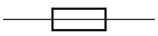
- PTC thermistor (`thRp`, or `thermistor ptc`)



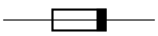
- NTC thermistor (`thRn`, or `thermistor ntc`)



- Fuse (`fuse`)

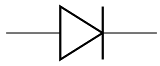


- Asymmetric fuse (`afuse`, or `asymmetric fuse`)

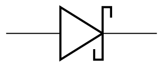


4.2.4 Diodes and such

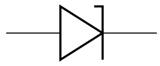
- Empty diode (`empty diode`, or `Do`)



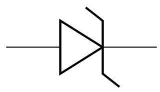
- Empty Schottky diode (`empty Schottky diode`, or `sDo`)



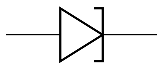
- Empty Zener diode (`empty Zener diode`, or `zDo`)



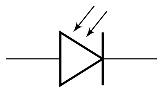
- Empty ZZener diode (`empty ZZener diode`, or `zzDo`)



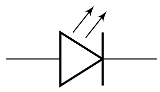
- Empty tunnel diode (`empty tunnel diode`, or `tDo`)



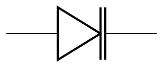
- Empty photodiode (`empty photodiode`, or `pDo`)



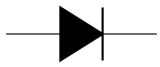
- Empty led (`empty led`, or `leDo`)



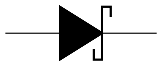
- Empty varcap (`empty varcap`, or `VCo`)



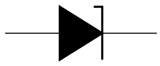
- Full diode (`full diode`, or `D*`)



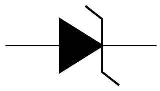
- Full Schottky diode (`full Schottky diode`, or `sD*`)



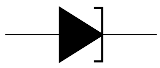
- Full Zener diode (`full Zener diode`, or `zD*`)



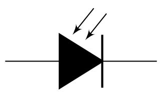
- Full ZZener diode (`full ZZener diode`, or `zzD*`)



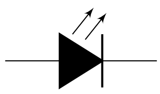
- Full tunnel diode (`full tunnel diode`, or `tD*`)



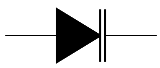
- Full photodiode (`full photodiode`, or `pD*`)



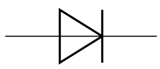
- Full led (`full led`, or `leD*`)



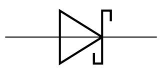
- Full varcap (`full varcap`, or `VC*`)



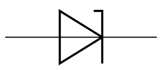
- Stroke diode (`stroke diode`, or `D-`)



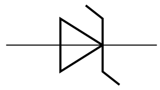
- Stroke Schottky diode (`stroke Schottky diode`, or `sD-`)



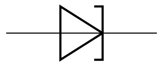
- Stroke Zener diode (`stroke Zener diode`, or `zD-`)



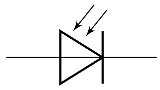
- Stroke ZZener diode (`stroke ZZener diode`, or `zzD-`)



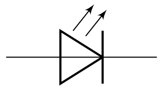
- Stroke tunnel diode (`stroke tunnel diode`, or `tD-`)



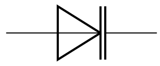
- Stroke photodiode (`stroke photodiode`, or `pD-`)



- Stroke led (`stroke led`, or `leD-`)



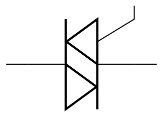
- Stroke varcap (`stroke varcap`, or `VC-`)



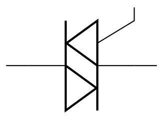
4.2.5 Other tripole-like diodes

The following tripoles are entered with the usual command of the form

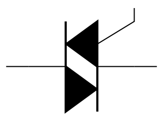
- Standard triac (shape depends on package option) (`triac`, or `Tr`)



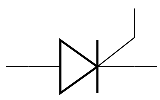
- Empty triac (`empty triac`, or `Tro`)



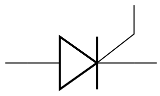
- Full triac (`full triac`, or `Tr*`)



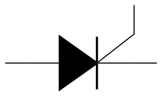
- Standard thyristor (shape depends on package option) (`thyristor`, or `Ty`)



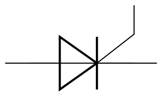
- Empty thyristor (`empty thyristor`, or `Tyo`)



- Full thyristor (`full thyristor`, or `Ty*`)



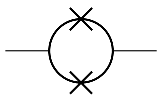
- Stroke thyristor (`stroke thyristor`, or `Ty-`)



See chapter 6.1.3 for information how access the third connector

> The package options `fulldiode`, `strokediode`, and `emptydiode` (and the styles `[full diodes]`, `[stroke diodes]`, and `[empty diodes]`) define which shape will be used by abbreviated commands such that `D`, `sD`, `zD`, `zzD`, `tD`, `pD`, `leD`, `VC`, `Ty`,`Tr`(no stroke symbol available!).

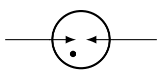
- Squid (`squid`)



- Barrier (`barrier`)

- European gas filled surge arrester (`european gas filled surge arrester`)

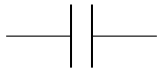
- American gas filled surge arrester (`american gas filled surge arrester`)



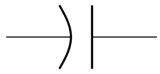
> If (default behaviour) `europeangfsurgearrester` option is active (or the style `[european gas filled surge arrester]` is used), the shorthands `gas filled surge arrester` and `gf surge arrester` are equivalent to the european version of the component.
>
> If otherwise `americangfsurgearrester` option is active (or the style `[american gas filled surge arrester]` is used), the shorthands the shorthands `gas filled surge arrester` and `gf surge arrester` are equivalent to the american version of the component.

4.2.6 Basic dynamical bipoles

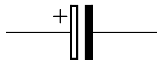
- Capacitor (`capacitor`, or `C`)



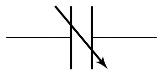
- Polar capacitor (`polar capacitor`, or `pC`)



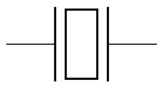
- Electrolytic capacitor (`ecapacitor`, or `eC,elko`)



- Variable capacitor (`variable capacitor`, or `vC`)



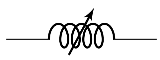
- Piezoelectric Element (`piezoelectric`, or `PZ`)



If (default behaviour) `cuteinductors` option is active (or the style `[cute inductors]` is used), the inductors are displayed as follows:

- Inductor (`L`, or `cute inductor`)

- Variable inductor (`vL`, or `variable cute inductor`)



If `americaninductors` option is active (or the style `[american inductors]` is used), the inductors are displayed as follows:

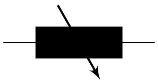
- Inductor (`L`, or `american inductor`)

- Variable inductor (`vL`, or `variable american inductor`)

Finally, if `europeaninductors` option is active (or the style `[european inductors]` is used), the inductors are displayed as follows:

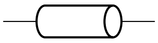
- Inductor (`L`, or `european inductor`)

- Variable inductor (`vL`, or `variable european inductor`)



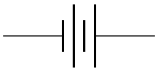
There is also a transmission line:

- Transmission line (`TL`, or `transmission line, tline`)

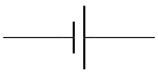


4.2.7 Stationary sources

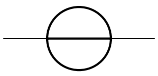
- Battery (`battery`)



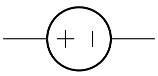
- Single battery cell (`battery1`)



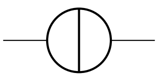
- Voltage source (european style) (`european voltage source`)



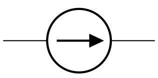
- Voltage source (american style) (`american voltage source`)



- Current source (european style) (`european current source`)



- Current source (american style) (`american current source`)



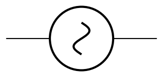
If (default behaviour) `europeancurrents` option is active (or the style `[european currents]` is used), the shorthands `current source`, `isource`, and `I` are equivalent to `european current source`. Otherwise, if `americancurrents` option is active (or the style `[american currents]` is used) they are equivalent to `american current source`.

Similarly, if (default behaviour) `europeanvoltages` option is active (or the style `[european voltages]` is used), the shorthands `voltage source`, `vsource`, and `V` are equivalent to `european voltage source`. Otherwise, if `americanvoltages` option is active (or the style `[american voltages]` is used) they are equivalent to `american voltage source`.

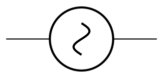
4.2.8 Sinusoidal sources

Here because I was asked for them. But how do you distinguish one from the other?!

- Sinusoidal voltage source (`sinusoidal voltage source`, or `vsourcesin, sV`)

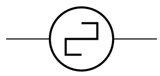


- Sinusoidal current source (`sinusoidal current source`, or `isourcesin, sI`)

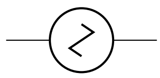


4.2.9 Special sources

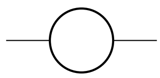
- Square voltage source (`square voltage source`, or `vsourcesquare, sqV`)



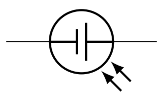
- Triangle voltage source (`vsourcetri`, or `tV`)



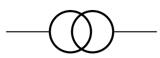
- Empty voltage source (`esource`)



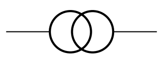
- Photovoltaic-voltage source (`pvsource`)



- Double Zero style current source (`ioosource`)

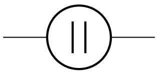


- Double Zero style voltage source (`voosource`)

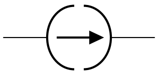


4.2.10 DC sources

- DC voltage source (`dcvsource`)

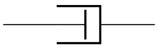


- DC current source (`dcisource`)



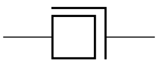
4.2.11 Mechanical Analogy

- Mechanical Damping (`damper`)



- Mechanical Stiffness (`spring`)

- Mechanical Mass (`mass`)



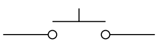
4.2.12 Switch

- Switch (`switch`, or `spst`)

- Closing switch (`closing switch`, or `cspst`)

- Opening switch (`opening switch`, or `ospst`)

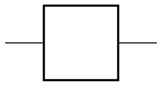
- Push button (`push button`)



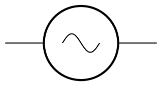
4.2.13 Block diagram components

Contributed by Stefan Erhardt.

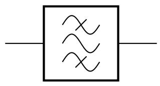
- generic two port[5] (`twoport`)



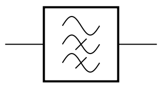
- vco (`vco`)



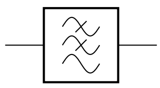
- bandpass (`bandpass`)



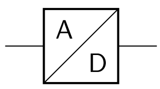
- highpass (`highpass`)



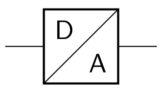
- lowpass (`lowpass`)



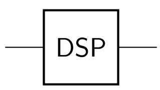
- A/D converter (`adc`)



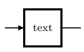
- D/A converter (`dac`)

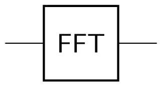


- DSP (`dsp`)

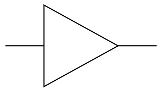


- FFT (`fft`)

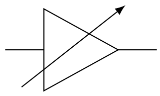
[5]To specify text to be put in the component: `twoport[t=text]`): 



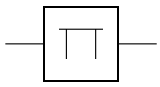
- amplifier (`amp`)



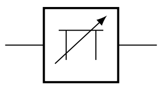
- VGA (`vamp`)



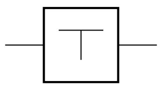
- π attenuator (`piattenuator`)



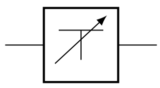
- var. π attenuator (`vpiattenuator`)



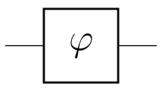
- T attenuator (`tattenuator`)



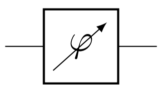
- var. T attenuator (`vtattenuator`)



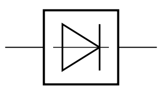
- phase shifter (`phaseshifter`)



- var. phase shifter (`vphaseshifter`)



- detector (`detector`)

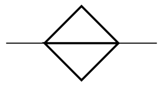


4.3 Tripoles

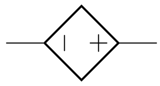
4.3.1 Controlled sources

Admittedly, graphically they are bipoles. But I couldn't…

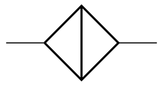
- Controlled voltage source (european style) (`european controlled voltage source`)



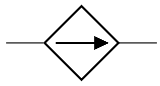
- Controlled voltage source (american style) (`american controlled voltage source`)



- Controlled current source (european style) (`european controlled current source`)

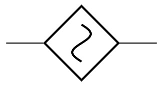


- Controlled current source (american style) (`american controlled current source`)

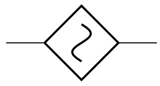


> If (default behaviour) `europeancurrents` option is active (or the style `[european currents]` is used), the shorthands `controlled current source`, `cisource`, and `cI` are equivalent to `european controlled current source`. Otherwise, if `americancurrents` option is active (or the style `[american currents]` is used) they are equivalent to `american controlled current source`.
>
> Similarly, if (default behaviour) `europeanvoltages` option is active (or the style `[european voltages]` is used), the shorthands `controlled voltage source`, `cvsource`, and `cV` are equivalent to `european controlled voltage source`. Otherwise, if `americanvoltages` option is active (or the style `[american voltages]` is used) they are equivalent to `american controlled voltage source`.

- Controlled sinusoidal voltage source (`controlled sinusoidal voltage source`, or `controlled vsourcesin, cvsourcesin, csV`)

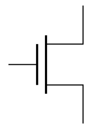


- Controlled sinusoidal current source (`controlled sinusoidal current source`, or `controlled isourcesin, cisourcesin, csI`)

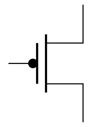


4.3.2 Transistors

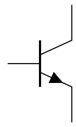
- NMOS (node[`nmos`]{})



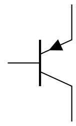
- PMOS (node[`pmos`]{})



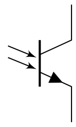
- NPN (node[`npn`]{})



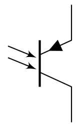
- PNP (node[`pnp`]{})



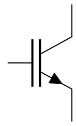
- NPN (node[`npn,photo`]{})



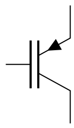
- PNP (node[`pnp,photo`]{})



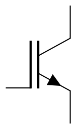
- NIGBT (node[`nigbt`]{})



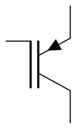
- PIGBT (node[`pigbt`]{})



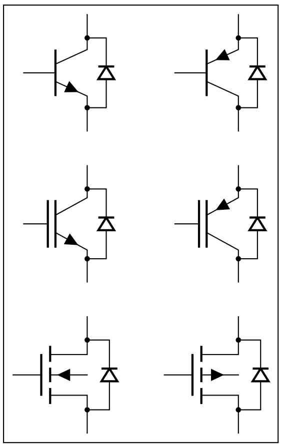
- Lnigbt (node[`Lnigbt`]{})



- Lpigbt (node[`Lpigbt`]{})

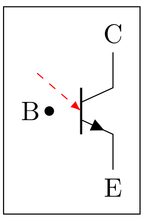


For all transistors a bodydiode(or freewheeling diode) can automatically be drawn. Just use the global option bodydiode, or for single transistors, the tikz-option bodydiode:



```
\begin{circuitikz}
  \draw (0,0) node[npn,bodydiode](npn){}++(2,0)node[pnp,
      bodydiode](npn){};
  \draw (0,-2) node[nigbt,bodydiode](npn){}++(2,0)node[
      pigbt,bodydiode](npn){};
  \draw (0,-4) node[nfet,bodydiode](npn){}++(2,0)node[
      pfet,bodydiode](npn){};
\end{circuitikz}
```

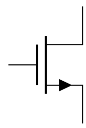
The Base/Gate connection of all transistors can be disable by using the options *nogate* or *nobase*, respectively. The Base/Gate anchors are floating, but there an additional anchor "nogate"/"nobase", which can be used to point to the unconnected base:



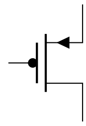

```
\begin{circuitikz}
  \draw (2,0) node[npn,nobase](npn){};
  \draw (npn.E) node[below]{E};
  \draw (npn.C) node[above]{C};
  \draw (npn.B) node[circ]{} node[left]{B};
  \draw[dashed,red,-latex] (1,0.5)--(npn.nobase);
\end{circuitikz}
```

If the option `arrowmos` is used (or after the command `\ctikzset{tripoles/mos style/arrows}` is given), this is the output:

- nmos (node[`nmos`]{})

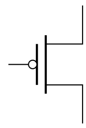


- PMOS (node[pmos]{})



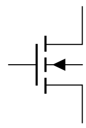
To draw the PMOS circle non-solid, use the option emptycircle or the command \ctikzset{tripoles/pmos styl

- PMOS (node[pmos,emptycircle]{})

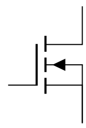


NFETs and PFETs have been incorporated based on code provided by Clemens Helfmeier and Theodor Borsche. Use the package options fetsolderdot/nofetsolderdot to enable/disable solderdot at some fet-transistors. Additionally, the solderdot option can be enabled/disabled for single transistors with the option ”solderdot” and ”nosolderdot”, respectively.

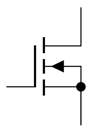
- NFET (node[nfet]{})



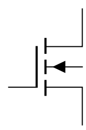
- NIGFETE (node[nigfete]{})



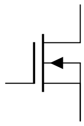
- NIGFETE (node[nigfete,solderdot]{})



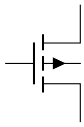
- NIGFETEBULK (node[nigfetebulk]{})



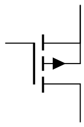
- NIGFETD (node[nigfetd]{})



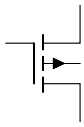
- PFET (node[pfet]{})



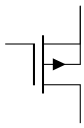
- PIGFETE (node[pigfete]{})



- PIGFETEBULK (node[pigfetebulk]{})

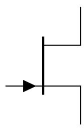


- PIGFETD (node[pigfetd]{})

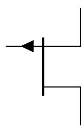


NJFET and PJFET have been incorporated based on code provided by Danilo Piazzalunga:

- NJFET (node[njfet]{})

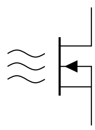


- PJFET (node[pjfet]{})



ISFET

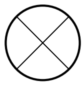
- ISFET (node[isfet]{})



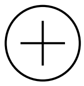
4.3.3 Block diagram

These come from Stefan Erhardt's contribution of block diagram components. Add a box around them with the option `box`.

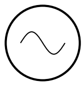
- MIXER (node[`mixer`]{})



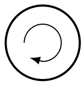
- ADDER (node[`adder`]{})



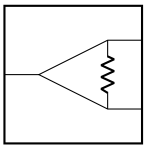
- OSCILLATOR (node[`oscillator`]{})



- CIRCULATOR (node[`circulator`]{})

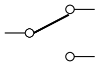


- WILKINSON DIVIDER (node[`wilkinson`]{})

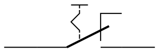


4.3.4 Switch

- SPDT (node[`spdt`]{})



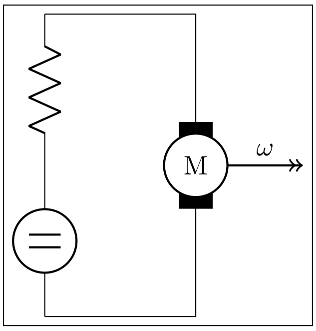
- Toggle switch (`toggle switch`)



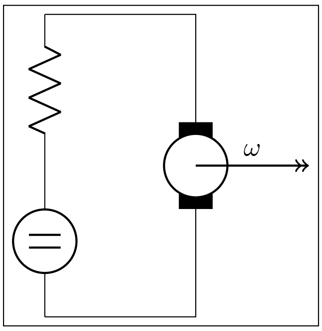
4.3.5 Electro-Mechanical Devices

- Motor (node[**elmech**]{M})

- Generator (node[**elmech**]{G})

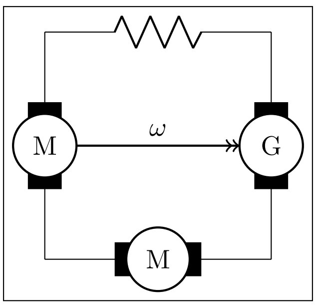



```
\begin{circuitikz}
\draw (2,0) node[elmech](motor){M};
\draw (motor.north) |-(0,2) to [R] ++(0,-2) to[
    dcvsource]++(0,-2) -| (motor.bottom);
\draw[thick,->>](motor.right)--++(1,0)node[midway,
    above]{$\omega$};
\end{circuitikz}
```




```
\begin{circuitikz}
\draw (2,0) node[elmech](motor){};
\draw (motor.north) |-(0,2) to [R] ++(0,-2) to[
    dcvsource]++(0,-2) -| (motor.bottom);
\draw[thick,->>](motor.center)--++(1.5,0)node[midway,
    above]{$\omega$};
\end{circuitikz}
```

The symbols can also be used along a path, using the transistor-path-syntax(T in front of the shape name, see section 6.6). Don´t forget to use parameter n to name the node and get acces to the anchors:

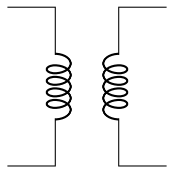



```
\begin{circuitikz}
\draw (0,0) to [Telmech=M,n=motor] ++(0,-3) to [
    Telmech=M] ++(3,0) to [Telmech=G,n=generator]
    ++(0,3) to [R] (0,0);
\draw[thick,->>](motor.left)--(generator.left)node[
    midway,above]{$\omega$};
\end{circuitikz}
```

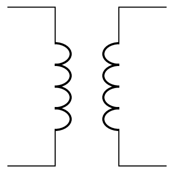
4.4 Double bipoles

Transformers automatically use the inductor shape currently selected. These are the three possibilities:

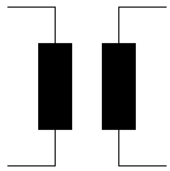
- Transformer (cute inductor) (node[`transformer`]{})



- Transformer (american inductor) (node[`transformer`]{})

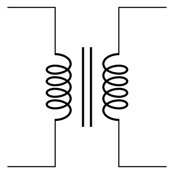


- Transformer (european inductor) (node[`transformer`]{})

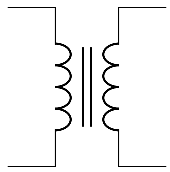


Transformers with core are also available:

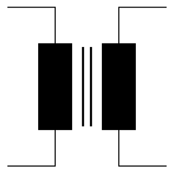
- Transformer core (cute inductor) (node[`transformer core`]{})



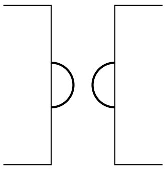
- Transformer core (american inductor) (node[`transformer core`]{})



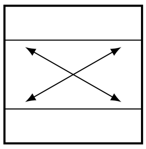
- Transformer core (european inductor) (node[`transformer core`]{})



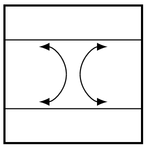
- Gyrator (node[`gyrator`]{})



- Coupler (node[`coupler`]{})

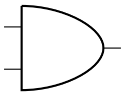


- Coupler, 2 (node[`coupler2`]{})

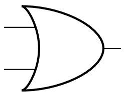


4.5 Logic gates

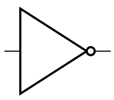
- American AND port (node[`american and port`]{})



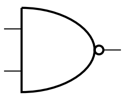
- American OR port (node[`american or port`]{})



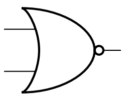
- American NOT port (node[`american not port`]{})



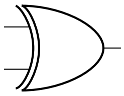
- American NAND port (node[`american nand port`]{})



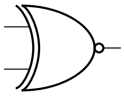
- American NOR port (node[`american nor port`]{})



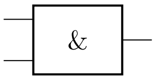
- American XOR port (node[`american xor port`]{})



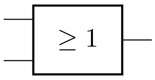
- American XNOR port (node[`american xnor port`]{})



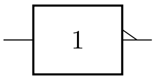
- European AND port (node[`european and port`]{})



- European OR port (node[`european or port`]{})

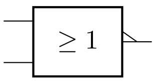


- European NOT port (node[`european not port`]{})

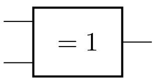


- European NAND port (node[`european nand port`]{})

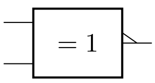
- European NOR port (node[`european nor port`]{})



- European XOR port (node[`european xor port`]{})



- European XNOR port (node[`european xnor port`]{})

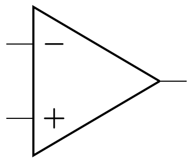


If (default behaviour) `americanports` option is active (or the style `[american ports]` is used), the shorthands `and port`, `or port`, `not port`, `nand port`, `not port`, `xor port`, and `xnor port` are equivalent to the american version of the respective logic port.

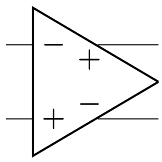
If otherwise `europeanports` option is active (or the style `[european ports]` is used), the shorthands `and port`, `or port`, `not port`, `nand port`, `not port`, `xor port`, and `xnor port` are equivalent to the european version of the respective logic port.

4.6 Amplifiers

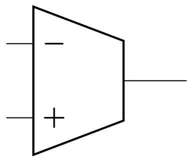
- Operational amplifier (node[`op amp`]{})



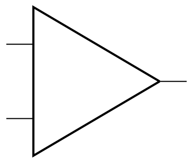
- Fully differential operational amplifier[6] (node[`fd op amp`]{})



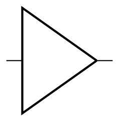
- transconductance amplifier (node[`gm amp`]{})



- Plain amplifier (node[`plain amp`]{})



- Buffer (node[`buffer`]{})



[6] Contributed by Kristofer M. Monisit.

4.7 Support shapes

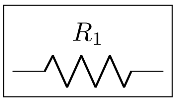
- Arrows (current and voltage) (node[`currarrow`]{})

- Arrow to draw at its tip, useful for block diagrams. (node[`inputarrow`]{})

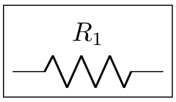
- Connected terminal (node[`circ`]{})

- Unconnected terminal (node[`ocirc`]{})

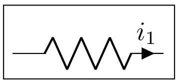
5 Usage



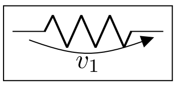

```
\begin{circuitikz}
  \draw (0,0) to[R, l=$R_1$] (2,0);
\end{circuitikz}
```



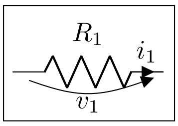

```
\begin{circuitikz}
  \draw (0,0) to[R=$R_1$] (2,0);
\end{circuitikz}
```



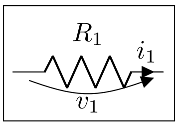

```
\begin{circuitikz}
  \draw (0,0) to[R, i=$i_1$] (2,0);
\end{circuitikz}
```




```
\begin{circuitikz}
  \draw (0,0) to[R, v=$v_1$] (2,0);
\end{circuitikz}
```

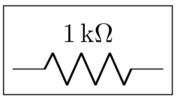



```
\begin{circuitikz}
  \draw (0,0) to[R=$R_1$, i=$i_1$, v=$v_1$] (2,0);
\end{circuitikz}
```



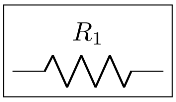

```
\begin{circuitikz}
  \draw (0,0) to[R=$R_1$, i=$i_1$, v=$v_1$] (2,0);
\end{circuitikz}
```

Long names/styles for the bipoles can be used:

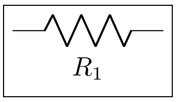



```
\begin{circuitikz}\draw
 (0,0) to[resistor=1<\kilo\ohm>] (2,0)
;\end{circuitikz}
```

5.1 Labels

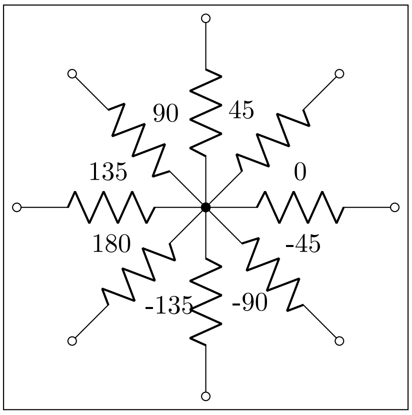



```
\begin{circuitikz}
  \draw (0,0) to[R, l^=$R_1$] (2,0);
\end{circuitikz}
```

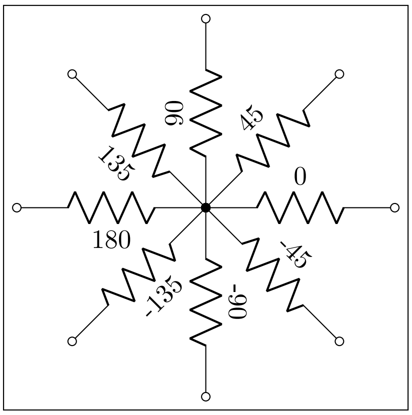



```
\begin{circuitikz}
  \draw (0,0) to[R, l_=$R_1$] (2,0);
\end{circuitikz}
```

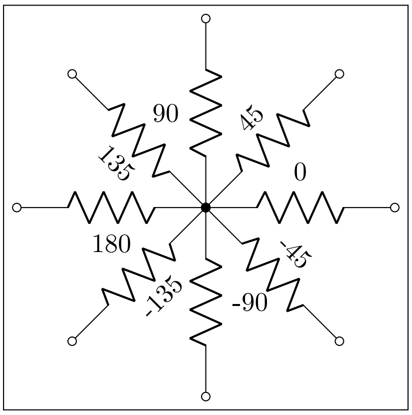
The default orientation of labels is controlled by the options `smartlabels`, `rotatelabels` and `straightlabels` (or the corresponding `label/align` keys). Here are examples to see the differences:




```
\begin{circuitikz}
\ctikzset{label/align = straight}
\def\DIR{0,45,90,135,180,-90,-45,-135}
\foreach \i in \DIR {
  \draw (0,0) to[R=\i, *-o] (\i:2.5);
}
\end{circuitikz}
```




```
\begin{circuitikz}
\ctikzset{label/align = rotate}
\def\DIR{0,45,90,135,180,-90,-45,-135}
\foreach \i in \DIR {
  \draw (0,0) to[R=\i, *-o] (\i:2.5);
}
\end{circuitikz}
```



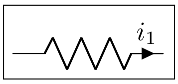

```
\begin{circuitikz}
\ctikzset{label/align = smart}
\def\DIR{0,45,90,135,180,-90,-45,-135}
\foreach \i in \DIR {
  \draw (0,0) to[R=\i, *-o] (\i:2.5);
}
\end{circuitikz}
```

5.2 Currents

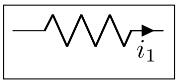
The counting direction of currents and voltages have changed with version 0.5, for compability reasons there is a option to use the olddirections(see options). For the new scheme, the following rules apply:

- **Normal bipoles:** currents and voltages are counted positiv in drawing direction.
- **Current Sources:** current is counted positiv in drawing direction, voltage in opposite direction
- **Voltage Sources:** voltage is counted positiv in drawing direction, current in opposite direction

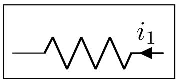
With this convention, the power at loads is positive and negative at sources.



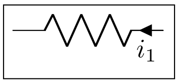

```
\begin{circuitikz}
  \draw (0,0) to[R, i^>=$i_1$] (2,0);
\end{circuitikz}
```



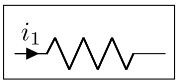

```
\begin{circuitikz}
  \draw (0,0) to[R, i_>=$i_1$] (2,0);
\end{circuitikz}
```



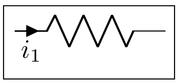

```
\begin{circuitikz}
  \draw (0,0) to[R, i^<=$i_1$] (2,0);
\end{circuitikz}
```



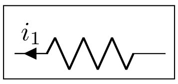

```
\begin{circuitikz}
  \draw (0,0) to[R, i_<=$i_1$] (2,0);
\end{circuitikz}
```



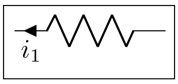

```
\begin{circuitikz}
  \draw (0,0) to[R, i>^=$i_1$] (2,0);
\end{circuitikz}
```




```
\begin{circuitikz}
  \draw (0,0) to[R, i>_=$i_1$] (2,0);
\end{circuitikz}
```

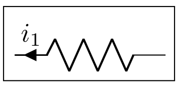



```
\begin{circuitikz}
  \draw (0,0) to[R, i<^=$i_1$] (2,0);
\end{circuitikz}
```

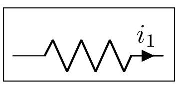



```
\begin{circuitikz}
  \draw (0,0) to[R, i<_=$i_1$] (2,0);
\end{circuitikz}
```

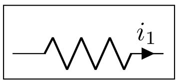
Also



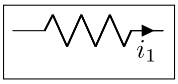

```
\begin{circuitikz}
  \draw (0,0) to[R, i<=$i_1$] (2,0);
\end{circuitikz}
```



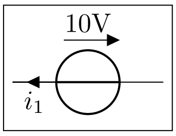

```
\begin{circuitikz}
  \draw (0,0) to[R, i>=$i_1$] (2,0);
\end{circuitikz}
```



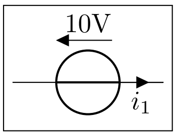

```
\begin{circuitikz}
  \draw (0,0) to[R, i^=$i_1$] (2,0);
\end{circuitikz}
```



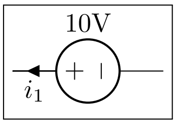

```
\begin{circuitikz}
  \draw (0,0) to[R, i_=$i_1$] (2,0);
\end{circuitikz}
```



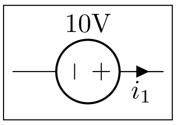

```
\begin{circuitikz}
  \draw (0,0) to[V=10V, i_=$i_1$] (2,0);
\end{circuitikz}
```




```
\begin{circuitikz}
  \draw (0,0) to[V<=10V, i_=$i_1$] (2,0);
\end{circuitikz}
```




```
\begin{circuitikz}[american]
  \draw (0,0) to[V=10V, i_=$i_1$] (2,0);
\end{circuitikz}
```



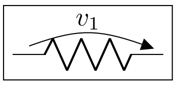

```
\begin{circuitikz}[american]
  \draw (0,0) to[V<=10V, i_=$i_1$] (2,0);
\end{circuitikz}
```

5.3 Voltages

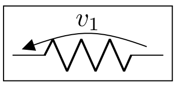
See introduction note at Currents (chapter 5.2, page 32)!

5.3.1 European style

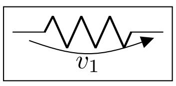
The default, with arrows. Use option `europeanvoltage` or style `[european voltages]`.



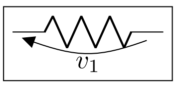

```
\begin{circuitikz}[european voltages]
  \draw (0,0) to[R, v^>=$v_1$] (2,0);
\end{circuitikz}
```



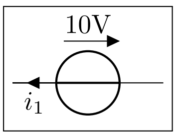

```
\begin{circuitikz}[european voltages]
  \draw (0,0) to[R, v^<=$v_1$] (2,0);
\end{circuitikz}
```



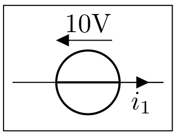

```
\begin{circuitikz}[european voltages]
  \draw (0,0) to[R, v_>=$v_1$] (2,0);
\end{circuitikz}
```



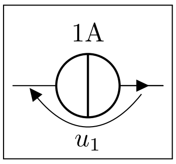

```
\begin{circuitikz}[european voltages]
  \draw (0,0) to[R, v_<=$v_1$] (2,0);
\end{circuitikz}
```



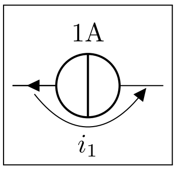

```
\begin{circuitikz}
  \draw (0,0) to[V=10V, i_=$i_1$] (2,0);
\end{circuitikz}
```




```
\begin{circuitikz}
  \draw (0,0) to[V<=10V, i_=$i_1$] (2,0);
\end{circuitikz}
```



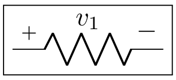

```
\begin{circuitikz}
  \draw (0,0) to[I=1A, v_=$u_1$] (2,0);
\end{circuitikz}
```



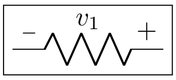

```
\begin{circuitikz}
  \draw (0,0) to[I<=1A, v_=$i_1$] (2,0);
\end{circuitikz}
```

5.3.2 American style

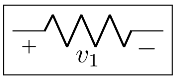
For those who like it (not me). Use option `americanvoltage` or set `[american voltages]`.



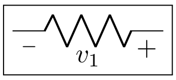

```
\begin{circuitikz}[american voltages]
  \draw (0,0) to[R, v^>=$v_1$] (2,0);
\end{circuitikz}
```



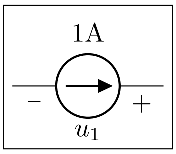

```
\begin{circuitikz}[american voltages]
  \draw (0,0) to[R, v^<=$v_1$] (2,0);
\end{circuitikz}
```



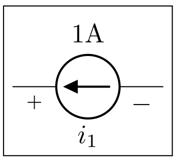

```
\begin{circuitikz}[american voltages]
  \draw (0,0) to[R, v_>=$v_1$] (2,0);
\end{circuitikz}
```




```
\begin{circuitikz}[american voltages]
  \draw (0,0) to[R, v_<=$v_1$] (2,0);
\end{circuitikz}
```

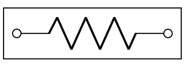



```
\begin{circuitikz}[american]
  \draw (0,0) to[I=1A, v_=$u_1$] (2,0);
\end{circuitikz}
```

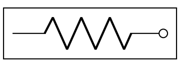



```
\begin{circuitikz}[american]
  \draw (0,0) to[I<=1A, v_=$i_1$] (2,0);
\end{circuitikz}
```

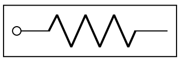
5.4 Nodes



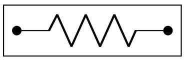
```
\begin{circuitikz}
  \draw (0,0) to[R, o-o] (2,0);
\end{circuitikz}
```



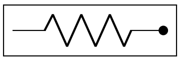
```
\begin{circuitikz}
  \draw (0,0) to[R, -o] (2,0);
\end{circuitikz}
```



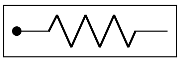
```
\begin{circuitikz}
  \draw (0,0) to[R, o-] (2,0);
\end{circuitikz}
```



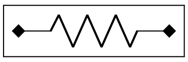
```
\begin{circuitikz}
  \draw (0,0) to[R, *-*] (2,0);
\end{circuitikz}
```



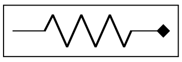
```
\begin{circuitikz}
  \draw (0,0) to[R, -*] (2,0);
\end{circuitikz}
```



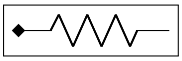
```
\begin{circuitikz}
  \draw (0,0) to[R, *-] (2,0);
\end{circuitikz}
```



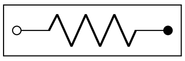
```
\begin{circuitikz}
  \draw (0,0) to[R, d-d] (2,0);
\end{circuitikz}
```



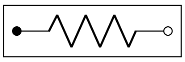
```
\begin{circuitikz}
  \draw (0,0) to[R, -d] (2,0);
\end{circuitikz}
```



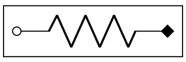
```
\begin{circuitikz}
  \draw (0,0) to[R, d-] (2,0);
\end{circuitikz}
```



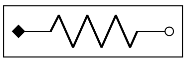
```
\begin{circuitikz}
  \draw (0,0) to[R, o-*] (2,0);
\end{circuitikz}
```



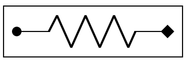
```
\begin{circuitikz}
  \draw (0,0) to[R, *-o] (2,0);
\end{circuitikz}
```



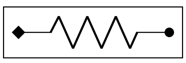
```
\begin{circuitikz}
  \draw (0,0) to[R, o-d] (2,0);
\end{circuitikz}
```



```
\begin{circuitikz}
  \draw (0,0) to[R, d-o] (2,0);
\end{circuitikz}
```



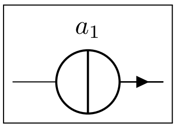
```
\begin{circuitikz}
  \draw (0,0) to[R, *-d] (2,0);
\end{circuitikz}
```



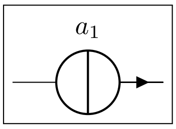
```
\begin{circuitikz}
  \draw (0,0) to[R, d-*] (2,0);
\end{circuitikz}
```

5.5 Special components

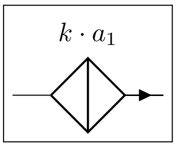
For some components label, current and voltage behave as one would expect:



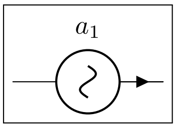

```
\begin{circuitikz}
  \draw (0,0) to[I=$a_1$] (2,0);
\end{circuitikz}
```



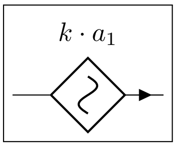

```
\begin{circuitikz}
  \draw (0,0) to[I, i=$a_1$] (2,0);
\end{circuitikz}
```




```
\begin{circuitikz}
  \draw (0,0) to[cI=$k\cdot a_1$] (2,0);
\end{circuitikz}
```

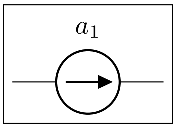



```
\begin{circuitikz}
  \draw (0,0) to[sI=$a_1$] (2,0);
\end{circuitikz}
```

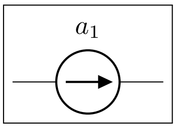



```
\begin{circuitikz}
  \draw (0,0) to[csI=$k\cdot a_1$] (2,0);
\end{circuitikz}
```

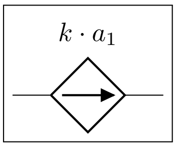
The following results from using the option `americancurrent` or using the style `[american currents]`.



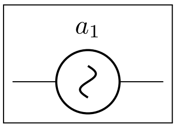

```
\begin{circuitikz}[american currents]
  \draw (0,0) to[I=$a_1$] (2,0);
\end{circuitikz}
```



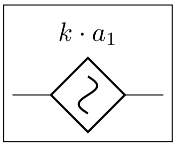

```
\begin{circuitikz}[american currents]
  \draw (0,0) to[I, i=$a_1$] (2,0);
\end{circuitikz}
```




```
\begin{circuitikz}[american currents]
  \draw (0,0) to[cI=$k\cdot a_1$] (2,0);
\end{circuitikz}
```

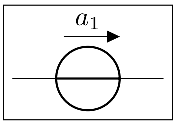



```
\begin{circuitikz}[american currents]
  \draw (0,0) to[sI=$a_1$] (2,0);
\end{circuitikz}
```

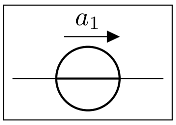



```
\begin{circuitikz}[american currents]
  \draw (0,0) to[csI=$k\cdot a_1$] (2,0);
\end{circuitikz}
```

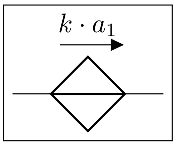
The same holds for voltage sources:



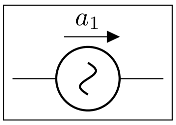

```
\begin{circuitikz}
  \draw (0,0) to[V=$a_1$] (2,0);
\end{circuitikz}
```



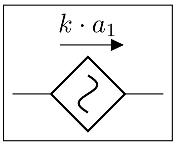

```
\begin{circuitikz}
  \draw (0,0) to[V, v=$a_1$] (2,0);
\end{circuitikz}
```




```
\begin{circuitikz}
  \draw (0,0) to[cV=$k\cdot a_1$] (2,0);
\end{circuitikz}
```

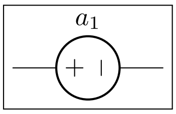



```
\begin{circuitikz}
  \draw (0,0) to[sV=$a_1$] (2,0);
\end{circuitikz}
```

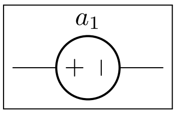



```
\begin{circuitikz}
  \draw (0,0) to[csV=$k\cdot a_1$] (2,0);
\end{circuitikz}
```

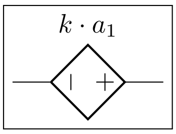
The following results from using the option `americanvoltage` or the style `[american voltages]`.



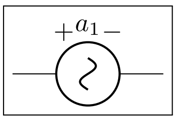

```
\begin{circuitikz}[american voltages]
  \draw (0,0) to[V=$a_1$] (2,0);
\end{circuitikz}
```



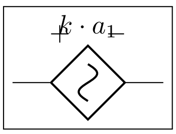

```
\begin{circuitikz}[american voltages]
  \draw (0,0) to[V, v=$a_1$] (2,0);
\end{circuitikz}
```




```
\begin{circuitikz}[american voltages]
  \draw (0,0) to[cV=$k\cdot a_1$] (2,0);
\end{circuitikz}
```



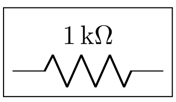

```
\begin{circuitikz}[american voltages]
  \draw (0,0) to[sV=$a_1$] (2,0);
\end{circuitikz}
```



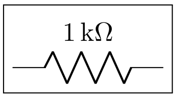

```
\begin{circuitikz}[american voltages]
  \draw (0,0) to[csV=$k\cdot a_1$] (2,0);
\end{circuitikz}
```

5.6 Integration with `siunitx`

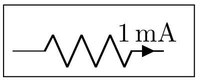
If the option `siunitx` is active (and *not* in ConTEXt), then the following are equivalent:



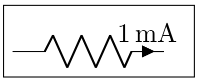

```
\begin{circuitikz}
  \draw (0,0) to[R, l=1<\kilo\ohm>] (2,0);
\end{circuitikz}
```



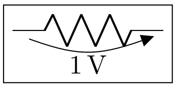

```
\begin{circuitikz}
  \draw (0,0) to[R, l=$\SI{1}{\kilo\ohm}$] (2,0);
\end{circuitikz}
```



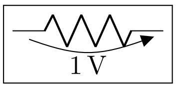

```
\begin{circuitikz}
  \draw (0,0) to[R, i=1<\milli\ampere>] (2,0);
\end{circuitikz}
```




```
\begin{circuitikz}
  \draw (0,0) to[R, i=$\SI{1}{\milli\ampere}$] (2,0);
\end{circuitikz}
```

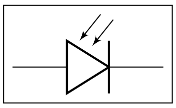



```
\begin{circuitikz}
  \draw (0,0) to[R, v=1<\volt>] (2,0);
\end{circuitikz}
```

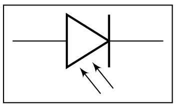



```
\begin{circuitikz}
  \draw (0,0) to[R, v=$\SI{1}{\volt}$] (2,0);
\end{circuitikz}
```

5.7 Mirroring

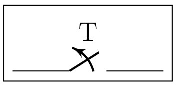


```
\begin{circuitikz}
  \draw (0,0) to[pD] (2,0);
\end{circuitikz}
```

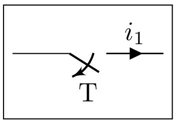


```
\begin{circuitikz}
  \draw (0,0) to[pD, mirror] (2,0);
\end{circuitikz}
```

At the moment, placing labels and currents on mirrored bipoles works:

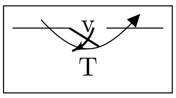



```
\begin{circuitikz}
  \draw (0,0) to[ospst=T] (2,0);
\end{circuitikz}
```




```
\begin{circuitikz}
  \draw (0,0) to[ospst=T, mirror, i=$i_1$] (2,0);
\end{circuitikz}
```

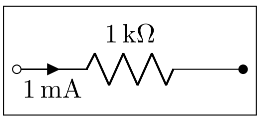
But voltages don't:



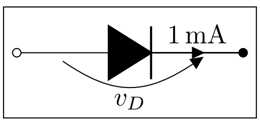

```
\begin{circuitikz}
  \draw (0,0) to[ospst=T, mirror, v=v] (2,0);
\end{circuitikz}
```

Sorry about that.

5.8 Putting them together



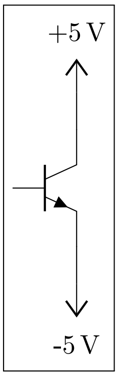

```
\begin{circuitikz}
  \draw (0,0) to[R=1<\kilo\ohm>,
     i>_=1<\milli\ampere>, o-*] (3,0);
\end{circuitikz}
```




```
\begin{circuitikz}
  \draw (0,0) to[D*, v=$v_D$,
     i=1<\milli\ampere>, o-*] (3,0);
\end{circuitikz}
```

6 Not only bipoles

Since only bipoles (but see section 6.6) can be placed ”along a line”, components with more than two terminals are placed as nodes:



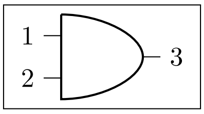

```
\begin{circuitikz}
\draw (0,0) node[npn](npn) at (0,0) {};
\draw (npn.C) --++(0,0.5) node[vcc]{+5\,\textnormal{V}};
\draw (npn.E) --++(0,-0.5) node[vee]{-5\,\textnormal{V}};
\end{circuitikz}
```

6.1 Anchors

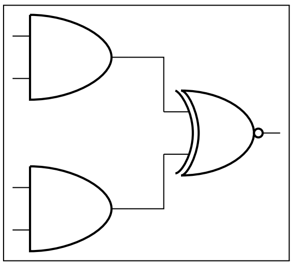
In order to allow connections with other components, all components define anchors.

6.1.1 Logical ports

All logical ports, except NOT, have two inputs and one output. They are called respectively `in 1`, `in 2`, `out`:

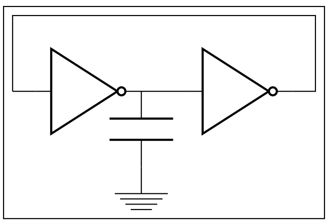



```
\begin{circuitikz} \draw
  (0,0) node[and port] (myand) {}
  (myand.in 1) node[anchor=east] {1}
  (myand.in 2) node[anchor=east] {2}
  (myand.out) node[anchor=west] {3}
;\end{circuitikz}
```



```
\begin{circuitikz} \draw
  (0,2) node[and port] (myand1) {}
  (0,0) node[and port] (myand2) {}
  (2,1) node[xnor port] (myxnor) {}
  (myand1.out) -| (myxnor.in 1)
  (myand2.out) -| (myxnor.in 2)
;\end{circuitikz}
```

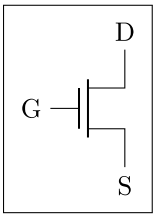
In the case of NOT, there are only `in` and `out` (although for compatibility reasons `in 1` is still defined and equal to `in`):



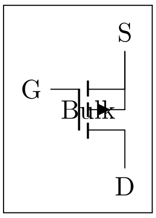
```
\begin{circuitikz} \draw
  (1,0) node[not port] (not1) {}
  (3,0) node[not port] (not2) {}
  (0,0) -- (not1.in)
  (not2.in) -- (not1.out)
  ++(0,-1) node[ground] {} to[C] (not1.out)
  (not2.out) -| (4,1) -| (0,0)
;\end{circuitikz}
```

6.1.2 Transistors

For NMOS, PMOS, NFET, NIGFETE, NIGFETD, PFET, PIGFETE, and PIGFETD transistors one has `base`, `gate`, `source` and `drain` anchors (which can be abbreviated with `B`, `G`, `S` and `D`):

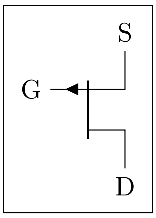



```
\begin{circuitikz} \draw
  (0,0) node[nmos] (mos) {}
  (mos.gate) node[anchor=east] {G}
  (mos.drain) node[anchor=south] {D}
  (mos.source) node[anchor=north] {S}
;\end{circuitikz}
```



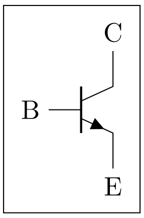

```
\begin{circuitikz} \draw
  (0,0) node[pigfete] (pigfete) {}
  (pigfete.G) node[anchor=east] {G}
  (pigfete.D) node[anchor=north] {D}
  (pigfete.S) node[anchor=south] {S}
  (pigfete.bulk) node[anchor=east] {Bulk}
;\end{circuitikz}
```

Similarly NJFET and PJFET have `gate`, `source` and `drain` anchors (which can be abbreviated with `G`, `S` and `D`):

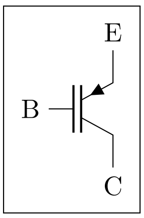



```
\begin{circuitikz} \draw
  (0,0) node[pjfet] (pjfet) {}
  (pjfet.G) node[anchor=east] {G}
  (pjfet.D) node[anchor=north] {D}
  (pjfet.S) node[anchor=south] {S}
;\end{circuitikz}
```

For NPN, PNP, NIGBT, and PIGBT transistors the anchors are `base`, `emitter` and `collector` anchors (which can be abbreviated with `B`, `E` and `C`):

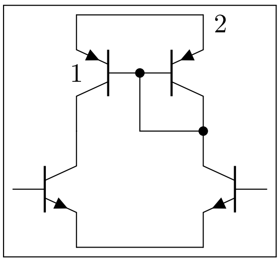



```
\begin{circuitikz} \draw
  (0,0) node[npn] (npn) {}
  (npn.base) node[anchor=east] {B}
  (npn.collector) node[anchor=south] {C}
  (npn.emitter) node[anchor=north] {E}
;\end{circuitikz}
```



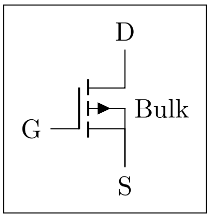

```
\begin{circuitikz} \draw
  (0,0) node[pigbt] (pigbt) {}
  (pigbt.B) node[anchor=east] {B}
  (pigbt.C) node[anchor=north] {C}
  (pigbt.E) node[anchor=south] {E}
;\end{circuitikz}
```

Here is one composite example (please notice that the `xscale=-1` style would also reflect the label of the transistors, so here a new node is added and its text is used, instead of that of `pnp1`):

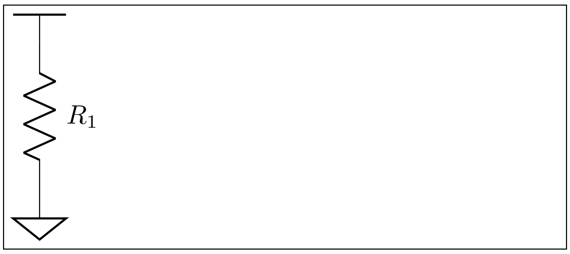



```
\begin{circuitikz} \draw
 (0,0) node[pnp] (pnp2) {2}
 (pnp2.B) node[pnp, xscale=-1, anchor=B] (pnp1) {}
   (pnp1) node {1}
 (pnp1.C) node[npn, anchor=C] (npn1) {}
 (pnp2.C) node[npn, xscale=-1, anchor=C] (npn2) {}
 (pnp1.E) -- (pnp2.E) (npn1.E) -- (npn2.E)
 (pnp1.B) node[circ] {} |- (pnp2.C) node[circ] {}
;\end{circuitikz}
```

Similarly, transistors and other components can be reflected vertically:



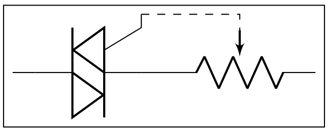

```
\begin{circuitikz} \draw
 (0,0) node[pigfete, yscale=-1] (pigfete) {}
 (pigfete.bulk) node[anchor=west] {Bulk}
 (pigfete.G) node[anchor=east] {G}
 (pigfete.D) node[anchor=south] {D}
 (pigfete.S) node[anchor=north] {S}
;\end{circuitikz}
```




```
 \begin{circuitikz}
     \draw (0,2)
         node[rground, yscale=-1]
             {}
     to[R=$R_1$] (0,0)
         node[sground] {};
  \end{circuitikz}
```

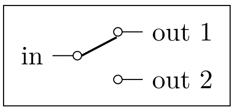
6.1.3 Other tripoles

When inserting a thrystor, a triac or a potentiometer, one needs to refer to the third node–gate (`gate` or `G`) for the former two; wiper (`wiper` or `W`) for the latter one. This is done by giving a name to the bipole:

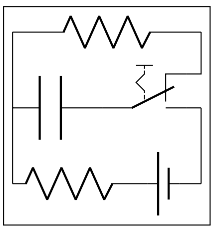


```
\begin{circuitikz} \draw
 (0,0) to[Tr, n=TRI] (2,0)
      to[pR, n=POT] (4,0);
 \draw[dashed] (TRI.G) -| (POT.wiper)
;\end{circuitikz}
```

As for the switches:

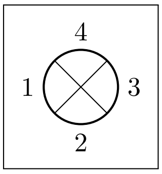



```
\begin{circuitikz} \draw
 (0,0) node[spdt] (Sw) {}
 (Sw.in) node[left] {in}
 (Sw.out 1) node[right] {out 1}
 (Sw.out 2) node[right] {out 2}
;\end{circuitikz}
```



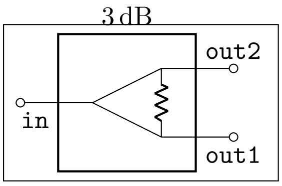
```
\begin{circuitikz} \draw
(0,0) to[C] (1,0) to[toggle switch , n=Sw] (2.5,0)
  -- (2.5,-1) to[battery1] (1.5,-1) to[R] (0,-1) -| (0,0)
 (Sw.out 2) -| (2.5, 1) to[R] (0,1) -- (0,0)
;\end{circuitikz}
```

The ports of the mixer and adder can be addressed with numbers or `west`/`south`/`east`/`north`:




```
\begin{circuitikz} \draw
  (0,0) node[mixer] (mix) {}
  (mix.1) node[left] {1}
  (mix.2) node[below] {2}
  (mix.3) node[right] {3}
  (mix.4) node[above] {4}
;\end{circuitikz}
```

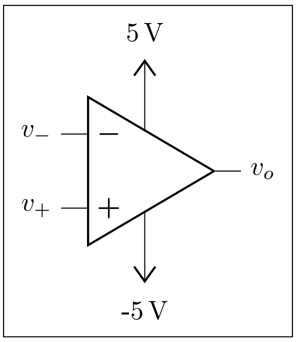
The Wilkinson divider has:




```
\begin{circuitikz} \draw
  (0,0) node[wilkinson] (w) {\SI{3}{dB}}
  (w.in) to[short,-o] ++(-0.5,0)
  (w.out1) to[short,-o] ++(0.5,0)
  (w.out2) to[short,-o] ++(0.5,0)
  (w.in) node[below left] {\texttt{in}}
  (w.out1) node[below right] {\texttt{out1}}
  (w.out2) node[above right] {\texttt{out2}}
  ;
\end{circuitikz}
```

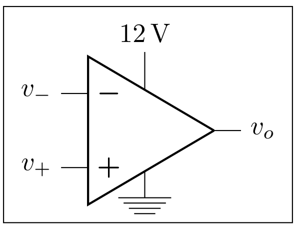
6.1.4 Operational amplifier

The op amp defines the inverting input (`-`), the non-inverting input (`+`) and the output (`out`) anchors:



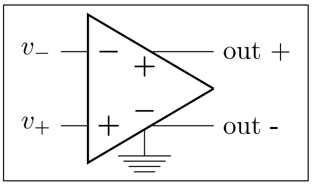

```
\begin{circuitikz} \draw
  (0,0) node[op amp] (opamp) {}
  (opamp.+) node[left] {$v_+$}
  (opamp.-) node[left] {$v_-$}
  (opamp.out) node[right] {$v_o$}
  (opamp.up) --++(0,0.5) node[vcc]{5\,\textnormal{V}}
  (opamp.down) --++(0,-0.5) node[vee]{-5\,\textnormal{V
      }}
;\end{circuitikz}
```

There are also two more anchors defined, `up` and `down`, for the power supplies:




```
\begin{circuitikz} \draw
  (0,0) node[op amp] (opamp) {}
  (opamp.+) node[left] {$v_+$}
  (opamp.-) node[left] {$v_-$}
  (opamp.out) node[right] {$v_o$}
  (opamp.down) node[ground] {}
  (opamp.up) ++ (0,.5) node[above] {\SI{12}{\volt}}
    -- (opamp.up)
;\end{circuitikz}
```

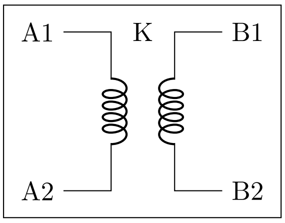
The fully differential op amp defines two outputs:



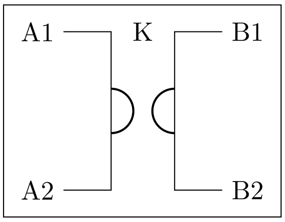

```
\begin{circuitikz} \draw
  (0,0) node[fd op amp] (opamp) {}
  (opamp.+) node[left] {$v_+$}
  (opamp.-) node[left] {$v_-$}
  (opamp.out +) node[right] {out +}
  (opamp.out -) node[right] {out -}
  (opamp.down) node[ground] {}
;\end{circuitikz}
```

6.1.5 Double bipoles

All the (few, actually) double bipoles/quadrupoles have the four anchors, two for each port. The first port, to the left, is port `A`, having the anchors `A1` (up) and `A2` (down); same for port `B`. They also expose the `base` anchor, for labelling:

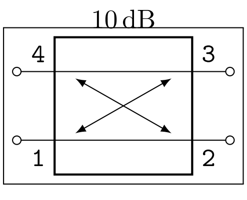



```
\begin{circuitikz} \draw
  (0,0) node[transformer] (T) {}
  (T.A1) node[anchor=east] {A1}
  (T.A2) node[anchor=east] {A2}
  (T.B1) node[anchor=west] {B1}
  (T.B2) node[anchor=west] {B2}
  (T.base) node{K}
;\end{circuitikz}
```

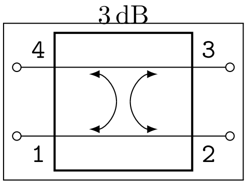



```
\begin{circuitikz} \draw
  (0,0) node[gyrator] (G) {}
  (G.A1) node[anchor=east] {A1}
  (G.A2) node[anchor=east] {A2}
  (G.B1) node[anchor=west] {B1}
  (G.B2) node[anchor=west] {B2}
  (G.base) node{K}
;\end{circuitikz}
```

However:




```
\begin{circuitikz} \draw
  (0,0) node[coupler] (c) {\SI{10}{dB}}
  (c.1) to[short,-o] ++(-0.5,0)
  (c.2) to[short,-o] ++(0.5,0)
  (c.3) to[short,-o] ++(0.5,0)
  (c.4) to[short,-o] ++(-0.5,0)
  (c.1) node[below left] {\texttt{1}}
  (c.2) node[below right] {\texttt{2}}
  (c.3) node[above right] {\texttt{3}}
  (c.4) node[above left] {\texttt{4}}
  ;
\end{circuitikz}
```

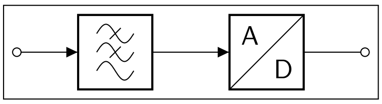



```
\begin{circuitikz} \draw
  (0,0) node[coupler2] (c) {\SI{3}{dB}}
  (c.1) to[short,-o] ++(-0.5,0)
  (c.2) to[short,-o] ++(0.5,0)
  (c.3) to[short,-o] ++(0.5,0)
  (c.4) to[short,-o] ++(-0.5,0)
  (c.1) node[below left] {\texttt{1}}
  (c.2) node[below right] {\texttt{2}}
  (c.3) node[above right] {\texttt{3}}
  (c.4) node[above left] {\texttt{4}}
  ;
\end{circuitikz}
```

6.2 Input arrows

Two ports

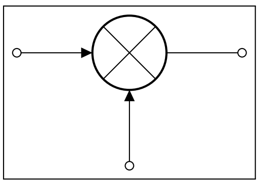
With the option `>` you can draw an arrow to the input of the block diagram symbols.




```
\begin{circuitikz} \draw
  (0,0) to[short,o-] ++(0.3,0)
  to[lowpass,>] ++(2,0)
  to[adc,>] ++(2,0)
  to[short,-o] ++(0.3,0);
\end{circuitikz}
```

Multi ports

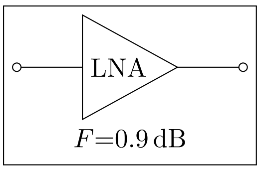
Since inputs and outputs can vary, input arrows can be placed as nodes. Note that you have to rotate the arrow on your own:



```
\begin{circuitikz} \draw
  (0,0) node[mixer] (m) {}
  (m.1) to[short,-o] ++(-1,0)
  (m.2) to[short,-o] ++(0,-1)
  (m.3) to[short,-o] ++(1,0)
  (m.1) node[inputarrow] {}
  (m.2) node[inputarrow,rotate=90] {};
\end{circuitikz}
```

6.3 Labels and custom twoport boxes

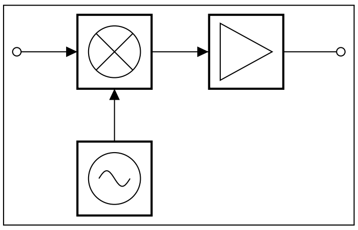
Some twoports have the option to place a normal label (`l=`) and a inner label (`t=`).




```
\begin{circuitikz}
  \ctikzset{bipoles/amp/width=0.9}
  \draw (0,0) to[amp,t=LNA,l_=$F{=}0.9\,$dB,o-o] ++(3,0);
\end{circuitikz}
```

6.4 Box option

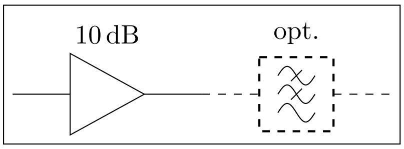
Some devices have the possibility to add a box around them. The inner symbol scales down to fit inside the box.



```
\begin{circuitikz} \draw
 (0,0) node[mixer,box,anchor=east] (m) {}
   to[amp,box,>,-o] ++(2.5,0)
 (m.west) node[inputarrow] {} to[short,-o]
     ++(-0.8,0)
 (m.south) node[inputarrow,rotate=90] {} --
   ++(0,-0.7) node[oscillator,box,anchor=north] {};
\end{circuitikz}
```

6.5 Dash optional parts

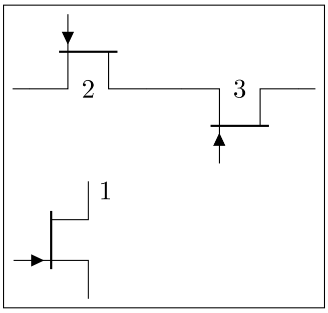
To show that a device is optional, you can dash it. The inner symbol will be kept with solid lines.




```
\begin{circuitikz}
  \draw (0,0) to[amp,l=\SI{10}{dB}] ++(2.5,0);
  \draw[dashed] (2.5,0) to[lowpass,l=opt.]
      ++(2.5,0);
\end{circuitikz}
```

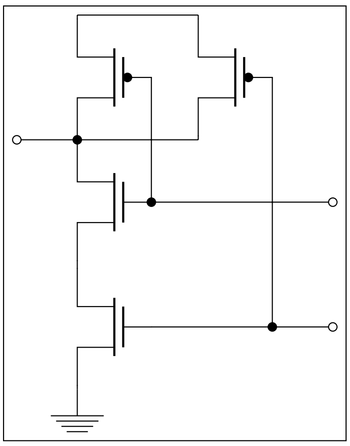
6.6 Transistor paths

For syntactical convenience transistors can be placed using the normal path notation used for bipoles. The transitor type can be specified by simply adding a "T" (for transistor) in front of the node name of the transistor. It will be placed with the base/gate orthogonal to the direction of the path:




```
\begin{circuitikz} \draw
 (0,0) node[njfet] {1}
 (-1,2) to[Tnjfet=2] (1,2)
   to[Tnjfet=3, mirror] (3,2);
;\end{circuitikz}
```

Access to the gate and/or base nodes can be gained by naming the transistors with the `n` or `name` path style:



```
\begin{circuitikz} \draw[yscale=1.1, xscale=.8]
 (2,4.5) -- (0,4.5) to[Tpmos, n=p1] (0,3)
    to[Tnmos, n=n1] (0,1.5)
    to[Tnmos, n=n2] (0,0) node[ground] {}
 (2,4.5) to[Tpmos,n=p2] (2,3) to[short, -*] (0,3)
 (p1.G) -- (n1.G) to[short, *-o] ($(n1.G)+(3,0)$)
 (n2.G) ++(2,0) node[circ] {} -| (p2.G)
 (n2.G) to[short, -o] ($(n2.G)+(3,0)$)
 (0,3) to[short, -o] (-1,3)
;\end{circuitikz}
```

The `name` property is available also for bipoles, although this is useful mostly for triac, potentiometer and thyristor (see 4.2.5).

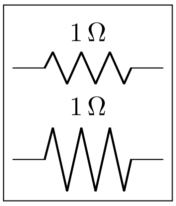
7 Customization

7.1 Parameters

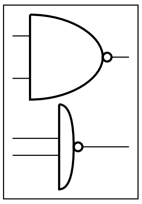
Pretty much all CircuiTi*k*Z relies heavily on `pgfkeys` for value handling and configuration. Indeed, at the beginning of `circuitikz.sty` a series of key definitions can be found that modify all the graphical characteristics of the package.

All can be varied using the `\ctikzset` command, anywhere in the code.

Shape of the components (on a per-component-class basis)

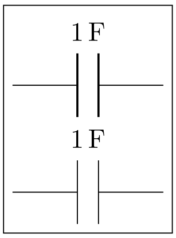



```
\tikz \draw (0,0) to[R=1<\ohm>] (2,0); \par
\ctikzset{bipoles/resistor/height=.6}
\tikz \draw (0,0) to[R=1<\ohm>] (2,0);
```



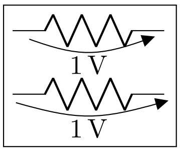
```
\tikz \draw (0,0) node[nand port] {}; \par
\ctikzset{tripoles/american nand port/input height=.2}
\ctikzset{tripoles/american nand port/port width=.2}
\tikz \draw (0,0) node[nand port] {};
```

Thickness of the lines (globally)

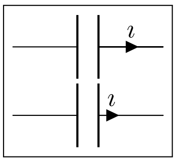



```
\tikz \draw (0,0) to[C=1<\farad>] (2,0); \par
\ctikzset{bipoles/thickness=1}
\tikz \draw (0,0) to[C=1<\farad>] (2,0);
```

Global properties Of voltage and current



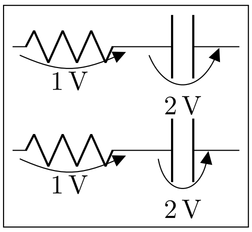

```
\tikz \draw (0,0) to[R, v=1<\volt>] (2,0); \par
\ctikzset{voltage/distance from node=.1}
\tikz \draw (0,0) to[R, v=1<\volt>] (2,0);
```




```
\tikz \draw (0,0) to[C, i=$\imath$] (2,0); \par
\ctikzset{current/distance = .2}
\tikz \draw (0,0) to[C, i=$\imath$] (2,0);
```

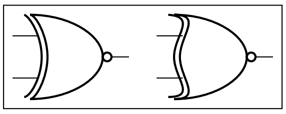
However, you can override the properties `voltage/distance from node`[7], `voltage/bump b`[8] and `voltage/european label distance`[9] on a per-component basis, in order to fine-tune the voltages:

[7] That is, how distant from the initial and final points of the path the arrow starts and ends.
[8] Controlling how high the bump of the arrow is — how curved it is.
[9] Controlling how distant from the bipole the voltage label will be.




```
\tikz \draw (0,0) to[R, v=1<\volt>] (1.5,0)
      to[C, v=2<\volt>] (3,0); \par
\ctikzset{bipoles/capacitor/voltage/%
    distance from node/.initial=.7}
\tikz \draw (0,0) to[R, v=1<\volt>] (1.5,0)
      to[C, v=2<\volt>] (3,0); \par
```

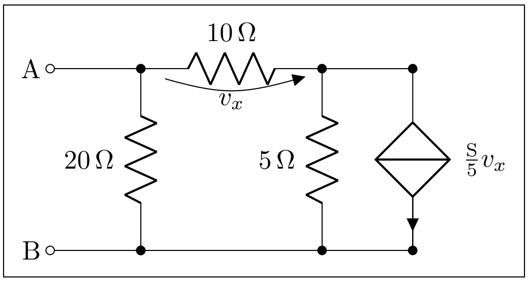
Admittedly, not all graphical properties have understandable names, but for the time it will have to do:



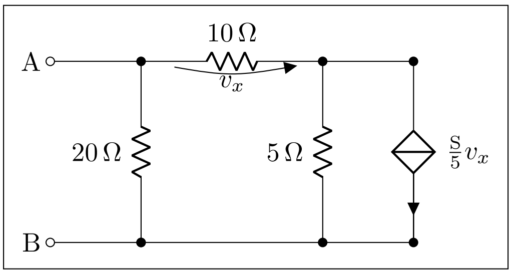
```
\tikz \draw (0,0) node[xnor port] {};
\ctikzset{tripoles/american xnor port/aaa=.2}
\ctikzset{tripoles/american xnor port/bbb=.6}
\tikz \draw (0,0) node[xnor port] {};
```

7.2 Components size

Perhaps the most important parameter is `\circuitikzbasekey/bipoles/length`, which can be interpreted as the length of a resistor (including reasonable connections): all other lenghts are relative to this value. For instance:




```
\ctikzset{bipoles/length=1.4cm}
\begin{circuitikz}[scale=1.2]\draw
 (0,0) node[anchor=east] {B}
      to[short, o-*] (1,0)
      to[R=20<\ohm>, *-*] (1,2)
      to[R=10<\ohm>, v=$v_x$] (3,2) -- (4,2)
      to[cI=$\frac{\si{\siemens}}{5} v_x$, *-*] (4,0) -- (3,0)
      to[R=5<\ohm>, *-*] (3,2)
 (3,0) -- (1,0)
 (1,2) to[short, -o] (0,2) node[anchor=east]{A}
;\end{circuitikz}
```

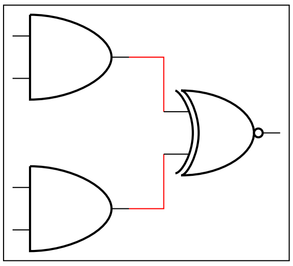



```
\ctikzset{bipoles/length=.8cm}
\begin{circuitikz}[scale=1.2]\draw
  (0,0) node[anchor=east] {B}
       to[short, o-*] (1,0)
       to[R=20<\ohm>, *-*] (1,2)
       to[R=10<\ohm>, v=$v_x$] (3,2) -- (4,2)
       to[cI=$\frac{\siemens}{5} v_x$, *-*] (4,0) -- (3,0)
       to[R=5<\ohm>, *-*] (3,2)
  (3,0) -- (1,0)
  (1,2) to[short, -o] (0,2) node[anchor=east]{A}
;\end{circuitikz}
```

7.3 Colors

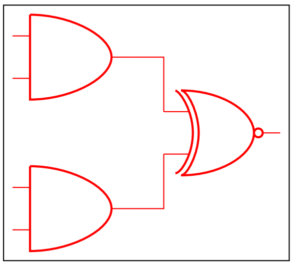
The color of the components is stored in the key `\circuitikzbasekey/color`. CircuiTi*k*Z tries to follow the color set in Ti*k*Z, although sometimes it fails. If you change color in the picture, please do not use just the color name as a style, like `[red]`, but rather assign the style `[color=red]`.

Compare for instance



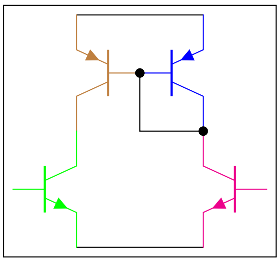
```
\begin{circuitikz} \draw[red]
  (0,2) node[and port] (myand1) {}
  (0,0) node[and port] (myand2) {}
  (2,1) node[xnor port] (myxnor) {}
  (myand1.out) -| (myxnor.in 1)
  (myand2.out) -| (myxnor.in 2)
;\end{circuitikz}
```

and



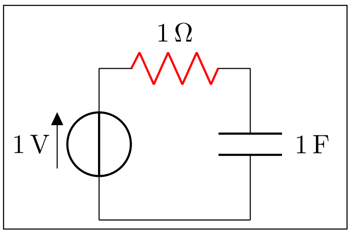
```
\begin{circuitikz} \draw[color=red]
  (0,2) node[and port] (myand1) {}
  (0,0) node[and port] (myand2) {}
  (2,1) node[xnor port] (myxnor) {}
  (myand1.out) -| (myxnor.in 1)
  (myand2.out) -| (myxnor.in 2)
;\end{circuitikz}
```

One can of course change the color *in medias res*:



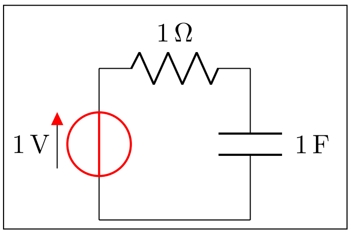
```
\begin{circuitikz} \draw
  (0,0) node[pnp, color=blue] (pnp2) {}
  (pnp2.B) node[pnp, xscale=-1, anchor=B, color=brown] (pnp1) {}
  (pnp1.C) node[npn, anchor=C, color=green] (npn1) {}
  (pnp2.C) node[npn, xscale=-1, anchor=C, color=magenta] (npn2) {}
  (pnp1.E) -- (pnp2.E) (npn1.E) -- (npn2.E)
  (pnp1.B) node[circ] {} |- (pnp2.C) node[circ] {}
;\end{circuitikz}
```

The all-in-one stream of bipoles poses some challanges, as only the actual body of the bipole, and not the connecting lines, will be rendered in the specified color. Also, please notice the curly braces around the `to`:



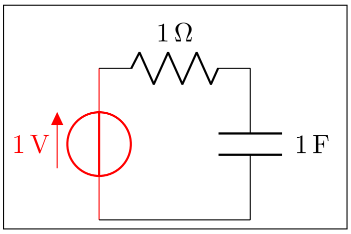

```
\begin{circuitikz} \draw
  (0,0) to[V=1<\volt>] (0,2)
     { to[R=1<\ohm>, color=red] (2,2) }
       to[C=1<\farad>] (2,0) -- (0,0)
;\end{circuitikz}
```

Which, for some bipoles, can be frustrating:




```
\begin{circuitikz} \draw
  (0,0){to[V=1<\volt>, color=red] (0,2) }
       to[R=1<\ohm>] (2,2)
       to[C=1<\farad>] (2,0) -- (0,0)
;\end{circuitikz}
```

The only way out is to specify different paths:




```
\begin{circuitikz} \draw[color=red]
  (0,0) to[V=1<\volt>, color=red] (0,2);
  \draw (0,2) to[R=1<\ohm>] (2,2)
       to[C=1<\farad>] (2,0) -- (0,0)
;\end{circuitikz}
```

And yes: this is a bug and *not* a feature...

8 FAQ

Q: When using `\tikzexternalize` I get the following error:

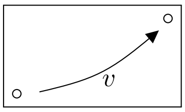
```
! Emergency stop.
```

A: The Ti*k*Z manual states:

> Furthermore, the library assumes that all LaTeX pictures are ended with `\end{tikzpicture}`.

Just substitute every occurrence of the environment `circuitikz` with `tikzpicture`. They are actually pretty much the same.

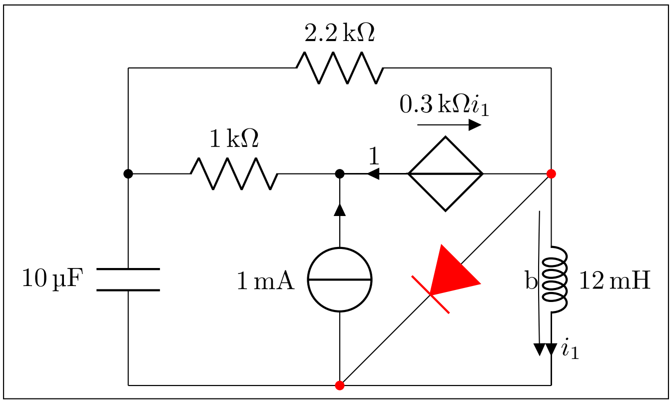
Q: How do I draw the voltage between two nodes?
A: Between any two nodes there is an open circuit!



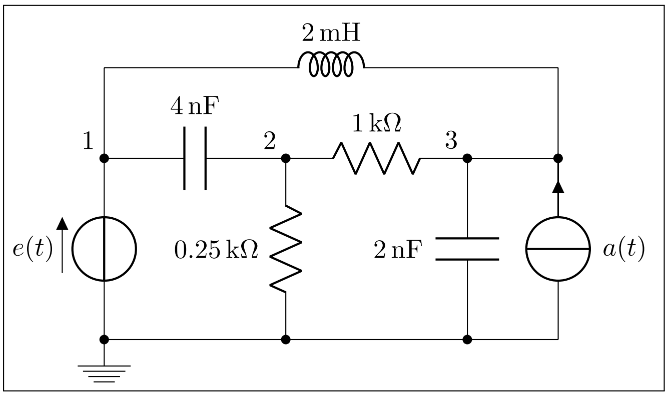

```
\begin{circuitikz} \draw
  node[ocirc] (A) at (0,0) {}
  node[ocirc] (B) at (2,1) {}
  (A) to[open, v=$v$] (B)
;\end{circuitikz}
```

Q: I cannot write `to[R = $R_1=12V$]` nor `to[ospst = open, 3s]`: I get errors.
A: It is a limitation of the Ti*k*Z parser. Use `to[R = $R_1{=}12V$]` and `to[ospst = open{,} 3s]` instead.

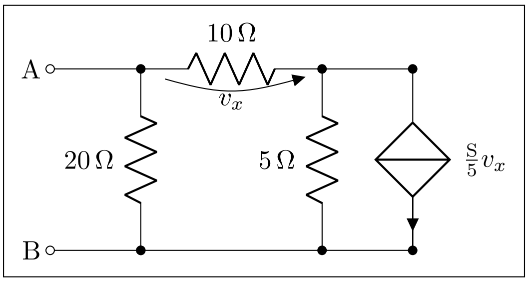
9 Examples



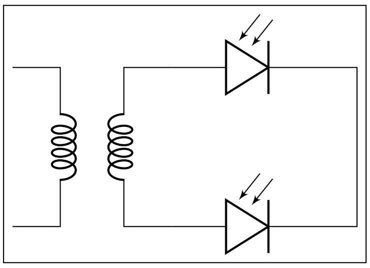

```
\begin{circuitikz}[scale=1.4]\draw
  (0,0) to[C, l=10<\micro\farad>] (0,2) -- (0,3)
        to[R, l=2.2<\kilo\ohm>] (4,3) -- (4,2)
        to[L, l=12<\milli\henry>, i=$i_1$,v=b] (4,0) -- (0,0)
  (4,2) { to[D*, *-*, color=red] (2,0) }
  (0,2) to[R, l=1<\kilo\ohm>, *-] (2,2)
        to[cV, i=1,v=$\SI{.3}{\kilo\ohm} i_1$] (4,2)
  (2,0) to[I, i=1<\milli\ampere>, -*] (2,2)
;\end{circuitikz}
```



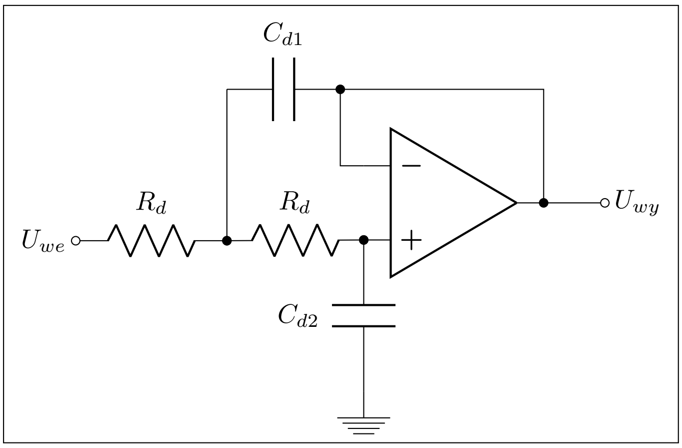

```
\begin{circuitikz}[scale=1.2]\draw
 (0,0) node[ground] {}
      to[V=$e(t)$, *-*] (0,2) to[C=4<\nano\farad>] (2,2)
      to[R, l_=.25<\kilo\ohm>, *-*] (2,0)
 (2,2) to[R=1<\kilo\ohm>] (4,2)
      to[C, l_=2<\nano\farad>, *-*] (4,0)
 (5,0) to[I, i_=$a(t)$, -*] (5,2) -- (4,2)
 (0,0) -- (5,0)
 (0,2) -- (0,3) to[L, l=2<\milli\henry>] (5,3) -- (5,2)

{[anchor=south east] (0,2) node {1} (2,2) node {2} (4,2) node {3}}
;\end{circuitikz}
```



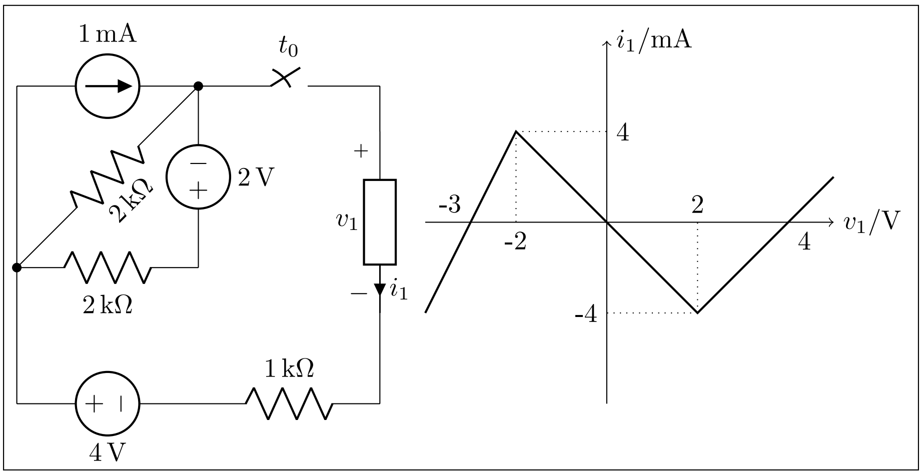

```
\begin{circuitikz}[scale=1.2]\draw
 (0,0) node[anchor=east] {B}
      to[short, o-*] (1,0)
      to[R=20<\ohm>, *-*] (1,2)
      to[R=10<\ohm>, v=$v_x$] (3,2) -- (4,2)
      to[cI=$\frac{\siemens}{5} v_x$, *-*] (4,0) -- (3,0)
      to[R=5<\ohm>, *-*] (3,2)
 (3,0) -- (1,0)
 (1,2) to[short, -o] (0,2) node[anchor=east]{A}
;\end{circuitikz}
```



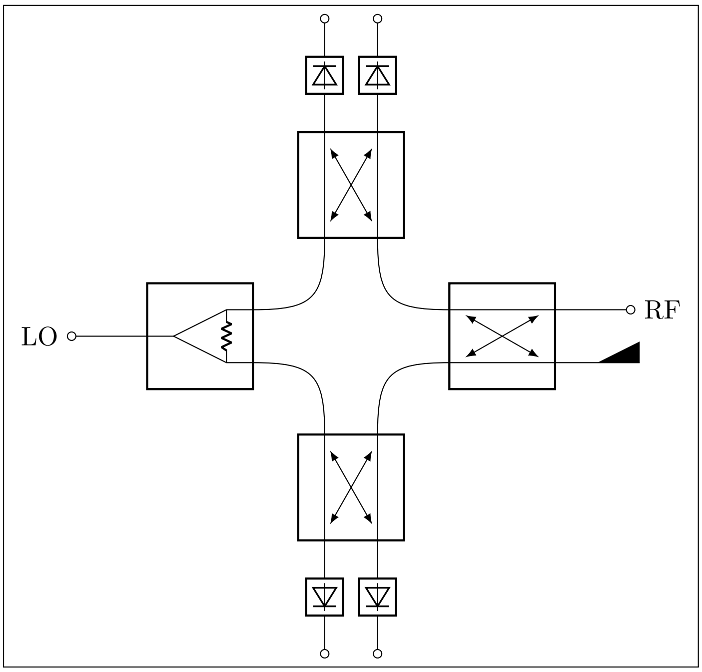
```
\begin{circuitikz}[scale=1]\draw
  (0,0) node[transformer] (T) {}
  (T.B2) to[pD] ($(T.B2)+(2,0)$) -| (3.5, -1)
  (T.B1) to[pD] ($(T.B1)+(2,0)$) -| (3.5, -1)
;\end{circuitikz}
```



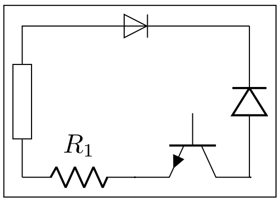

```
\begin{circuitikz}[scale=1]\draw
  (5,.5) node [op amp] (opamp) {}
  (0,0) node [left] {$U_{we}$} to [R, l=$R_d$, o-*] (2,0)
  to [R, l=$R_d$, *-*] (opamp.+)
  to [C, l_=$C_{d2}$, *-] ($(opamp.+)+(0,-2)$) node [ground] {}
  (opamp.out) |- (3.5,2) to [C, l_=$C_{d1}$, *-] (2,2) to [short] (2,0)
  (opamp.-) -| (3.5,2)
  (opamp.out) to [short, *-o] (7,.5) node [right] {$U_{wy}$}
;\end{circuitikz}
```




```
\begin{circuitikz}[scale=1.2, american]\draw
 (0,2) to[I=1<\milli\ampere>] (2,2)
      to[R, l_=2<\kilo\ohm>, *-*] (0,0)
      to[R, l_=2<\kilo\ohm>] (2,0)
      to[V, v_=2<\volt>] (2,2)
      to[cspst, l=$t_0$] (4,2) -- (4,1.5)
      to [generic, i=$i_1$, v=$v_1$] (4,-.5) -- (4,-1.5)
 (0,2) -- (0,-1.5) to[V, v_=4<\volt>] (2,-1.5)
      to [R, l=1<\kilo\ohm>] (4,-1.5);

  \begin{scope}[xshift=6.5cm, yshift=.5cm]
   \draw [->] (-2,0) -- (2.5,0) node[anchor=west] {$v_1/\volt$};
   \draw [->] (0,-2) -- (0,2) node[anchor=west] {$i_1/\SI{}{\milli\ampere}$} ;
   \draw (-1,0) node[anchor=north] {-2} (1,0) node[anchor=south] {2}
        (0,1) node[anchor=west] {4} (0,-1) node[anchor=east] {-4}
        (2,0) node[anchor=north west] {4}
        (-1.5,0) node[anchor=south east] {-3};
   \draw [thick] (-2,-1) -- (-1,1) -- (1,-1) -- (2,0) -- (2.5,.5);
   \draw [dotted] (-1,1) -- (-1,0) (1,-1) -- (1,0)
        (-1,1) -- (0,1) (1,-1) -- (0,-1);
  \end{scope}
\end{circuitikz}
```




```
\begin{circuitikz}[scale=1]
   \ctikzset{bipoles/detector/width=.35}
   \ctikzset{quadpoles/coupler/width=1}
   \ctikzset{quadpoles/coupler/height=1}
   \ctikzset{tripoles/wilkinson/width=1}
   \ctikzset{tripoles/wilkinson/height=1}
   %\draw[help lines,red,thin,dotted] (0,-5) grid (5,5);
   \draw
   (-2,0) node[wilkinson](w1){}
   (2,0) node[coupler] (c1) {}
   (0,2) node[coupler,rotate=90] (c2) {}
   (0,-2) node[coupler,rotate=90] (c3) {}
   (w1.out1) .. controls ++(0.8,0) and ++(0,0.8) .. (c3.3)
   (w1.out2) .. controls ++(0.8,0) and ++(0,-0.8) .. (c2.4)
   (c1.1) .. controls ++(-0.8,0) and ++(0,0.8) .. (c3.2)
   (c1.4) .. controls ++(-0.8,0) and ++(0,-0.8) .. (c2.1)
   (w1.in) to[short,-o] ++(-1,0)
   (w1.in) node[left=30] {LO}
   (c1.2) node[match,yscale=1] {}
   (c1.3) to[short,-o] ++(1,0)
   (c1.3) node[right=30] {RF}
   (c2.3) to[detector,-o] ++(0,1.5)
   (c2.2) to[detector,-o] ++(0,1.5)
   (c3.1) to[detector,-o] ++(0,-1.5)
   (c3.4) to[detector,-o] ++(0,-1.5)
   ;
\end{circuitikz}
```




```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{circuits.ee.IEC}
\usetikzlibrary{positioning}

\usepackage[compatibility]{circuitikz}
\ctikzset{bipoles/length=.9cm}

\begin{document}
 \begin{tikzpicture}[circuit ee IEC]
  \draw (0,0) to [resistor={name=R}] (0,2)
   to[diode={name=D}] (3,2);
  \draw (0,0) to[*R=$R_1$] (1.5,0) to[*Tnpn] (3,0)
    to[*D](3,2);
 \end{tikzpicture}
\end{document}
```

10 Changelog

The major changes among the different circuitikz versions are listed here. See `https://github.com/circuitikz/circuitikz/commits` for a full list of changes.

- Version 0.6 (2016-06-06)
  - Added Mechanical Symbols (damper,mass,spring)
  - Added new connection style diamond, use (d-d)
  - Added new sources voosource and ioosource (double zero-style)
  - All diode can now drawn in a stroked way, just use globel option "strokediode" or stroke instead of full/empty, or D-. Use this option for compliance with DIN standard EN-60617
  - Improved Shape of Diodes:tunnel diode, Zener diode, schottky diode (bit longer lines at cathode)
  - Reworked igbt: New anchors G,gate and new L-shaped form Lnigbt, Lpigbt
  - Improved shape of all fet-transistors and mirrored p-chan fets as default, as pnp, pmos, pfet are already. This means a backward-incompatibility, but smaller code, because p-channels mosfet are by default in the correct direction(source at top). Just remove the 'yscale=-1' from your p-chan fets at old pictures.
- Version 0.5 (2016-04-24)
  - new option boxed and dashed for hf-symbols
  - new option solderdot to enable/disable solderdot at source port of some fets
  - new parts: photovoltaic source, piezo crystal, electrolytic capacitor, electromechanical device(motor, generator)
  - corrected voltage and current direction(option to use old behaviour)
  - option to show body diode at fet transistors
- Version 0.4

- minor improvements to documentation
- comply with TDS
- merge high frequency symbols by Stefan Erhardt
- added switch (not opening nor closing)
- added solder dot in some transistors
- improved ConTeXt compatibility

- Version 0.3.1
  - different management of color…
  - fixed typo in documentation
  - fixed an error in the angle computation in voltage and current routines
  - fixed problem with label size when scaling a tikz picture
  - added gas filled surge arrester
  - added compatibility option to work with Tikz's own circuit library
  - fixed infinite in arctan computation
- Version 0.3.0
  - fixed gate node for a few transistors
  - added mixer
  - added fully differential op amp (by Kristofer M. Monisit)
  - now general settings for the drawing of voltage can be overridden for specific components
  - made arrows more homogeneous (either the current one, or latex' bt pgf)
  - added the single battery cell
  - added fuse and asymmetric fuse
  - added toggle switch
  - added varistor, photoresistor, thermocouple, push button
  - added thermistor, thermistor ptc, thermistor ptc
  - fixed misalignment of voltage label in vertical bipoles with names
  - added isfet
  - added noiseless, protective, chassis, signal and reference grounds (Luigi «Liverpool»)
- Version 0.2.4
  - added square voltage source (contributed by Alistair Kwan)
  - added buffer and plain amplifier (contributed by Danilo Piazzalunga)
  - added squid and barrier (contributed by Cor Molenaar)
  - added antenna and transmission line symbols contributed by Leonardo Azzinnari
  - added the changeover switch spdt (suggestion of Fabio Maria Antoniali)
  - rename of context.tex and context.pdf (thanks to Karl Berry)
  - updated the email address
  - in documentation, fixed wrong (non-standard) labelling of the axis in an example (thanks to prof. Claudio Beccaria)
  - fixed scaling inconsistencies in quadrupoles

  - fixed division by zero error on certain vertical paths
  - introduced options straighlabels, rotatelabels, smartlabels
- Version 0.2.3
  - fixed compatibility problem with label option from tikz
  - Fixed resizing problem for shape ground
  - Variable capacitor
  - polarized capacitor

- ConTeXt support (read the manual!)
- nfet, nigfete, nigfetd, pfet, pigfete, pigfetd (contribution of Clemens Helfmeier and Theodor Borsche)
- njfet, pjfet (contribution of Danilo Piazzalunga)
- pigbt, nigbt
- *backward incompatibility* potentiometer is now the standard resistor-with-arrow-in-the-middle; the old potentiometer is now known as variable resistor (or vR), similarly to variable inductor and variable capacitor
- triac, thyristor, memristor
- new property "name" for bipoles
- fixed voltage problem for batteries in american voltage mode
- european logic gates
- *backward incompatibility* new american standard inductor. Old american inductor now called "cute inductor"
- *backward incompatibility* transformer now linked with the chosen type of inductor, and version with core, too. Similarly for variable inductor
- *backward incompatibility* styles for selecting shape variants now end are in the plural to avoid conflict with paths
- new placing option for some tripoles (mostly transistors)
- mirror path style

• Version 0.2.2 - 20090520
  - Added the shape for lamps.
  - Added options `europeanresistor`, `europeaninductor`, `americanresistor` and `americaninductor`, with corresponding styles.
  - FIXED: error in transistor arrow positioning and direction under negative `xscale` and `yscale`.

• Version 0.2.1 - 20090503
  - Op-amps added
  - added options arrowmos and noarrowmos, to add arrows to pmos and nmos

• Version 0.2 - 20090417 First public release on CTAN
  - *Backward incompatibility*: labels ending with `:`*angle* are not parsed for positioning anymore.
  - Full use of Ti*k*Z keyval features.
  - White background is not filled anymore: now the network can be drawn on a background picture as well.
  - Several new components added (logical ports, transistors, double bipoles, …).
  - Color support.
  - Integration with `{siunitx}.`
  - `Voltage, american style.`
  - `Better code, perhaps. General cleanup at the very least.`

• Version 0.1 - 2007-10-29 First public release

Index of the components