

CircuiTikZ

version git:6c1ca8c (2017/03/25)

Massimo A. Redaelli (m.redaelli@gmail.com)

Stefan Lindner (stefan.lindner@fau.de)

Stefan Erhardt (stefan.erhardt@fau.de)

March 25, 2017

Contents

1	Introduction	1
1.1	About	1
1.2	Loading the package	1
1.3	Requirements	1
1.4	Incompatible packages	2
1.5	License	2
1.6	Feedback	2
2	Incompabilities between version	2
3	Package options	2
4	The components	4
4.1	Monopoles	5
4.2	Bipoles	7
4.2.1	Instruments	7
4.2.2	Basic resistive bipoles	7
4.2.3	Resistors and the like	8
4.2.4	Diodes and such	10
4.2.5	Other tripole-like diodes	12
4.2.6	Basic dynamical bipoles	14
4.2.7	Stationary sources	15
4.2.8	Sinusoidal sources	16
4.2.9	Special sources	16
4.2.10	DC sources	17
4.2.11	Mechanical Analogy	17
4.2.12	Switch	17
4.2.13	Block diagram components	18
4.3	Tripoles	20
4.3.1	Controlled sources	20
4.3.2	Transistors	21
4.3.3	Electronic Tubes	25
4.3.4	Block diagram	25
4.3.5	Switch	26
4.3.6	Electro-Mechanical Devices	26
4.4	Double bipoles	27

4.5	Logic gates	28
4.5.1	American Logic gates	28
4.5.2	European Logic gates	29
4.6	Amplifiers	30
4.7	Support shapes	31
5	Usage	32
5.1	Labels and Annotations	32
5.2	Currents	33
5.3	Voltages	35
5.3.1	European style	35
5.3.2	American style	36
5.4	Nodes	36
5.5	Special components	37
5.6	Integration with <code>siunitx</code>	39
5.7	Mirroring and Inverting	40
5.8	Putting them together	40
5.9	Line joins between Path Components	41
6	Not only bipoles	41
6.1	Anchors	41
6.1.1	Logical ports	41
6.1.2	Transistors	42
6.1.3	Other tripoles	43
6.1.4	Operational amplifier	44
6.1.5	Double bipoles	45
6.2	Input arrows	46
6.3	Labels and custom twoport boxes	46
6.4	Box option	47
6.5	Dash optional parts	47
6.6	Transistor paths	47
7	Customization	48
7.1	Parameters	48
7.2	Components size	49
7.3	Colors	50
8	FAQ	52
9	Examples	52
10	Changelog	58
	Index of the components	62

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 CircuiTikZ 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.

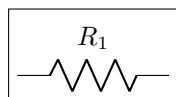
This documentation is somewhat scant. Hopefully the authors will find the leisure to improve it some day.

1.2 Loading the package

L ^A T _E X	ConT _E Xt ¹
<code>\usepackage{circuitikz}</code>	<code>\usemodule[circuitikz]</code>

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[*Tmos] (2,0)`, and so on (but `(0,0) node[nmos] {}`). See example at page 58.

1.5 License

Copyright © 2007–2017 Massimo Redaelli. This package is author-maintained. Permission is granted to copy, distribute and/or modify this software under the terms of the L^AT_EX 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>

¹ConT_EXt support was added mostly thanks to Mojca Miklavac and Aditya Mahajan.

2 Incompabilities between version

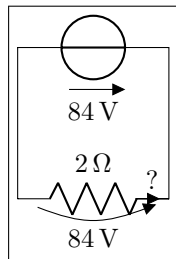
Here, we will provide a list of incompatibilities between different versions of circuitikz. We will try to hold this list short, but sometimes it is easier to break with old syntax than including a lot of switches and compatibility layers. You can check the used version at your local installation using the macro `\pgfcircversion{}`.

- Since v0.7?: The label behaviour at mirrored bipoles has changes, this fixes the voltage drawing, but perhaps you have to adjust your label positions.
- 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

For older projects, you can use an older version locally using the git-version and picking the correct commit from the repository (branch `gh-pages`).

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:



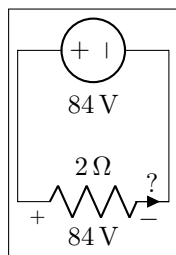
```

1 \begin{circuitikz}
2   \draw (0,0) to[R=2<\ohm>, i=?, v=84<\volt>] (2,0) --
3     (2,2) to[V<=84<\volt>] (0,2)
4     -- (0,0);
5 \end{circuitikz}

```

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

L ^A T _E X	ConT _E Xt
<code>\usepackage[american]{circuitikz}</code>	<code>\usemodule[circuitikz][american]</code>



```

1 \begin{circuitikz}
2   \draw (0,0) to[R=2<\ohm>, i=?, v=84<\volt>] (2,0) --
3     (2,2) to[V<=84<\volt>] (0,2)
4     -- (0,0);
5 \end{circuitikz}

```

Here is the list of all the options:

- `europenvoltages`: uses arrows to define voltages, and uses european-style voltage sources;
- `straightvoltages`: uses arrows to define voltages, and uses straight voltage arrows;
- `americanvoltages`: uses `-` and `+` to define voltages, and uses american-style voltage sources;
- `europencurrents`: uses european-style current sources;

- `americancurrents`: uses american-style current sources;
- `europereistors`: uses rectangular empty shape for resistors, as per european standards;
- `americanresistors`: uses zig-zag shape for resistors, as per american standards;
- `europainductors`: 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;
- `europainports`: uses rectangular logic ports, as per european standards;
- `americangfsurgearrester`: uses round gas filled surge arresters, as per american standards;
- `europengfsurgearrester`: uses rectangular gas filled surge arresters, as per european standards;
- `european`: equivalent to `europereistors`, `europainductors`, `europainports`, `europengfsurgearrester`;
- `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}`;
- `nosunitx`: 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 possible to load CircuiTikZ and TikZ circuit library together.
- **oldvoltagedirection**: Use old(erroneous) way of voltage direction having a difference between european and american direction
- **betterproportions**²: nicer proportions of transistors in comparison 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`.

4 The components

Here follows the list of all the shapes defined by CircuiTikZ. These are all `pgf` nodes, so they are usable in both `pgf` and `TikZ`.

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 name³ (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 TikZ 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}
```

²May change in the future!

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

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 Tikz 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]{})



- Signal ground (node[sground]{})



- Thicker ground (node[tground]{})



- Noiseless ground (node[nground]{})



- Protective ground (node[pground]{})



- Chassis ground⁴ (node[cground]{})

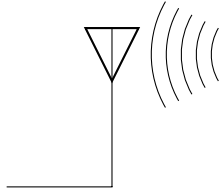


- Antenna (node[antenna]{})

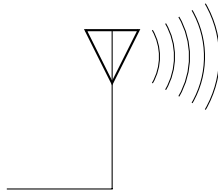


⁴These last three were contributed by Luigi «Liverpool»

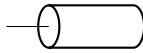
- Receiving antenna (node[**rxantenna**]{})



- Transmitting antenna (node[**txantenna**]{})



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



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



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



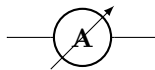
- match (node[**match**]{})



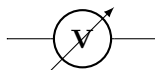
4.2 Bipoles

4.2.1 Instruments

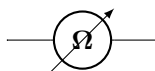
- Ammeter (**ammeter**)



- Voltmeter (**voltmeter**)



- Ohmmeter (**ohmmeter**)



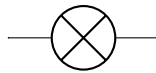
4.2.2 Basic resistive bipoles

- Short circuit (`short`)

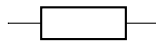


- Open circuit (`open`)

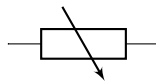
- Lamp (`lamp`)



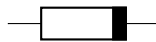
- Generic (symmetric) bipole (`generic`)



- Tunable generic bipole (`tgeneric`)



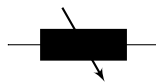
- Generic asymmetric bipole (`ageneric`)



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



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



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



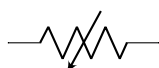
4.2.3 Resistors and the like

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`)

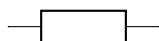


- Potentiometer (`pR`, or `american potentiometer`)

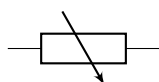


If instead `europeanresistors` option is active (or the style `[european resistors]` is used), the resistors, variable resistors and potentiometers are displayed as follows:

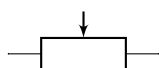
- Resistor (`R`, or `european resistor`)



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

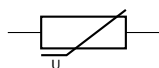


- Potentiometer (`pR`, or `european potentiometer`)

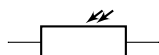


Other miscellaneous resistor-like devices:

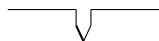
- Varistor (`varistor`)



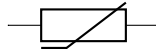
- Photoresistor (`phR`, or `photoresistor`)



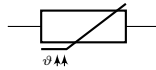
- Thermocouple (`thermocouple`)



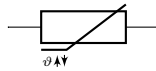
- Thermistor (`thR`, or `thermistor`)



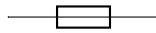
- PTC thermistor (`thRp`, or `thermistor ptc`)



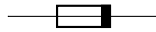
- NTC thermistor (`thRn`, or `thermistor ntc`)



- Fuse (`fuse`)

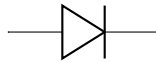


- Asymmetric fuse (`afuse`, or `asymmetric fuse`)

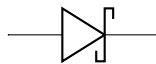


4.2.4 Diodes and such

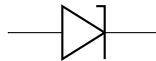
- Empty diode (`empty diode`, or `Do`)



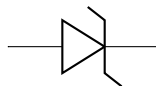
- Empty Schottky diode (`empty Schottky diode`, or `sDo`)



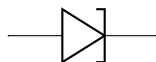
- Empty Zener diode (`empty Zener diode`, or `zDo`)



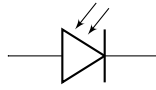
- Empty ZZener diode (`empty ZZener diode`, or `zzDo`)



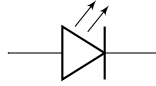
- Empty tunnel diode (`empty tunnel diode`, or `tDo`)



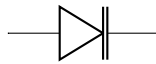
- Empty photodiode (`empty photodiode`, or `pDo`)



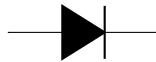
- Empty led (`empty led`, or `leDo`)



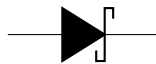
- Empty varcap (`empty varcap`, or `VCo`)



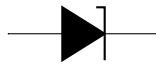
- Full diode (`full diode`, or `D*`)



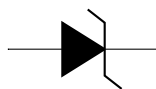
- Full Schottky diode (`full Schottky diode`, or `sD*`)



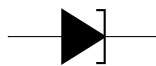
- Full Zener diode (`full Zener diode`, or `zD*`)



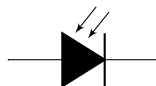
- Full ZZener diode (`full ZZener diode`, or `zzD*`)



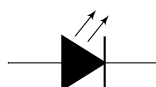
- Full tunnel diode (`full tunnel diode`, or `tD*`)



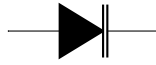
- Full photodiode (`full photodiode`, or `pD*`)



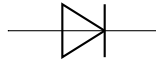
- Full led (`full led`, or `leD*`)



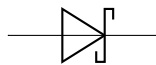
- Full varcap (`full varcap`, or `VC*`)



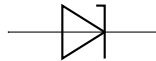
- Stroke diode (`stroke diode`, or `D-`)



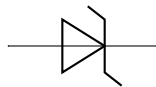
- Stroke Schottky diode (`stroke Schottky diode`, or `sD-`)



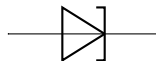
- Stroke Zener diode (`stroke Zener diode`, or `zD-`)



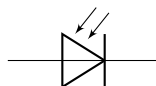
- Stroke ZZener diode (`stroke ZZener diode`, or `zzD-`)



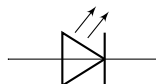
- Stroke tunnel diode (`stroke tunnel diode`, or `tD-`)



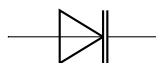
- Stroke photodiode (`stroke photodiode`, or `pD-`)



- 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) `europeangfsgearrester` 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.

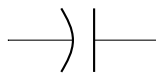
If otherwise `americangfsgearrester` option is active (or the style `[american gas filled surge arrester]` is used), the shorthands `gas filled surge arrester` and `gf surge arrester` are equivalent to the american version of the component.

4.2.6 Basic dynamical bipoles

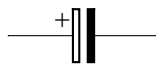
- Capacitor (`capacitor`, or `C`)



- 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`)



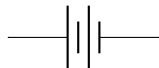
There is also a transmission line:

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

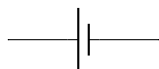


4.2.7 Stationary sources

- Battery (`battery`)



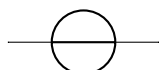
- Single battery cell (`battery1`)



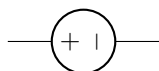
- Single battery cell (`battery2`)



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



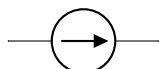
- Voltage source (american style) (`american voltage source`)



- 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`, `vsources`, 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 (`pvsources`)



- Double Zero style current source (`ioosources`)

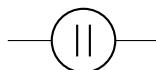


- Double Zero style voltage source (`voosources`)



4.2.10 DC sources

- DC voltage source (`dcvsources`)

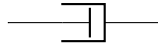


- DC current source (`dcisources`)



4.2.11 Mechanical Analogy

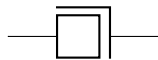
- Mechanical Damping (damper)



- Mechanical Stiffness (spring)



- Mechanical Mass (mass)



4.2.12 Switch

- Switch (switch, or spst)



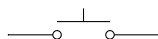
- Closing switch (closing switch, or cspst)



- Opening switch (opening switch, or ospst)



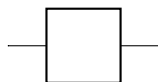
- Push button (push button)



4.2.13 Block diagram components

Contributed by Stefan Erhardt.

- generic two port⁵ (twoport)



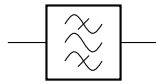
- vco (vco)



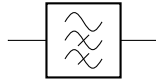
- bandpass (bandpass)

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

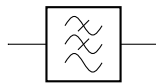




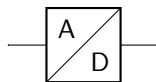
- highpass (**highpass**)



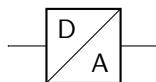
- lowpass (**lowpass**)



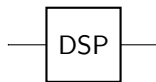
- A/D converter (**adc**)



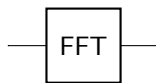
- D/A converter (**dac**)



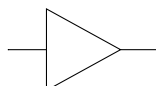
- DSP (**dsp**)



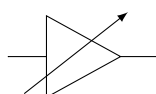
- FFT (**fft**)



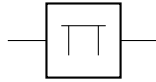
- amplifier (**amp**)



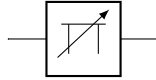
- VGA (**vamp**)



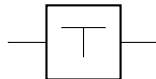
- π attenuator (**piattenuator**)



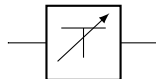
- var. π attenuator (`vpiattenuator`)



- T attenuator (`tattenuator`)



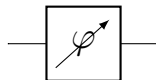
- var. T attenuator (`vtattenuator`)



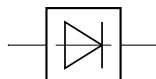
- phase shifter (`phaseshifter`)



- var. phase shifter (`vphaseshifter`)



- detector (`detector`)

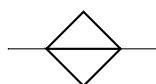


4.3 Tripoles

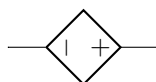
4.3.1 Controlled sources

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

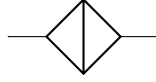
- Controlled voltage source (european style) (`european controlled voltage source`)



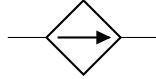
- Controlled voltage source (american style) (`american controlled voltage source`)



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



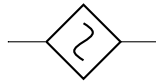
- Controlled current source (american style) (`american controlled current source`)



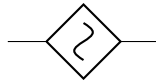
If (default behaviour) `europeancurrents` option is active (or the style `[european currents]` is used), the shorthands `controlled current source`, `cisource`, and `cI` are equivalent to `european controlled current source`. Otherwise, if `americancurrents` option is active (or the style `[american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

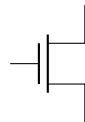


- Controlled sinusoidal current source (`controlled sinusoidal current source`, or `controlled isourcesin`, `cisourcesin`, `csI`)

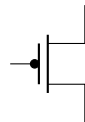


4.3.2 Transistors

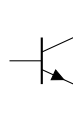
- NMOS (`node[nmos]{}`)



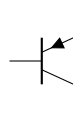
- PMOS (`node[pmos]{}`)



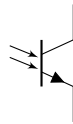
- NPN (node[npn]{})



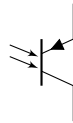
- PNP (node[pnp]{})



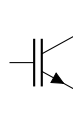
- NPN (node[npn,photo]{})



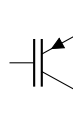
- PNP (node[pnp,photo]{})



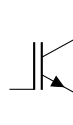
- NIGBT (node[nigbt]{})



- 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`:

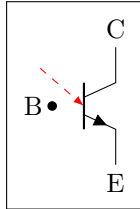


```

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

```

The Base/Gate connection of all transistors can be disabled by using the options `nogate` or `nobase`, respectively. The Base/Gate anchors are floating, but there is an additional anchor "no-gate"/"nobase", which can be used to point to the unconnected base:



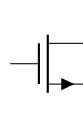
```

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

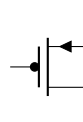
```

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 style/emptycircle}`

- 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]{})



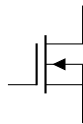
- NIGFETE (node[nigfete,solderdot]{})



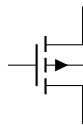
- NIGFETEBULK (node[nigfetebulk]{})



- NIGFETD (node[nigfetd]{})



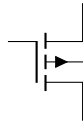
- PFET (node[pfet]{})



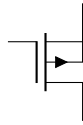
- PIGFETE (node[pigfete]{})



- PIGFETEBULK (node[pigfetebulk]{})

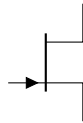


- PIGFETD (node[pigfetd]{})



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

- NJFET (node[njfet]{})

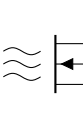


- PJFET (node[pjfet]{})



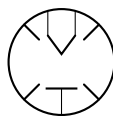
ISFET

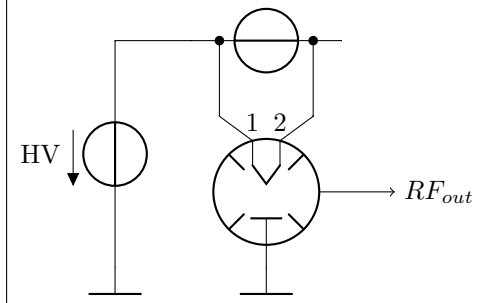
- ISFET (node[isfet]{})



4.3.3 Electronic Tubes

- Magnetron (node[magnetron]{})





```

1 \begin{circuitikz}
2 \draw (0,-2)node[rground](gnd){} to[
   voltage source,v<={HV}]++(0,3)
   --++(1,0)to[V,n=DC]++(2,0);
3 \draw (2,-1) node[magnetron,scale=1](
   magn){};
4 \draw (DC.left)++(-0.2,0)to [short,*-]
   ++(0,-1) to [short] (magn.cathode1);
5 \draw (DC.right)++(0.2,0)to [short,*-]
   ++(0,-1) to [short] (magn.cathode2);
6 \draw (magn.anode) to [short] (magn.
   anode|-gnd) node[rground]{};
7 \draw (magn.cathode1)node[above]{$1$};
8 \draw (magn.cathode2)node[above]{$2$};
9 \draw[->](magn.east) --++(1,0)node[
   right]{$RF_{out}$};
10 \end{circuitikz}

```

4.3.4 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]{})



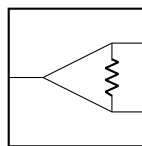
- OSCILLATOR (node[oscillator]{})



- CIRCULATOR (node[circulator]{})



- WILKINSON DIVIDER (node[wilkinson]{})



4.3.5 Switch

- SPDT (node[spdt]{})



- Toggle switch (toggle switch)

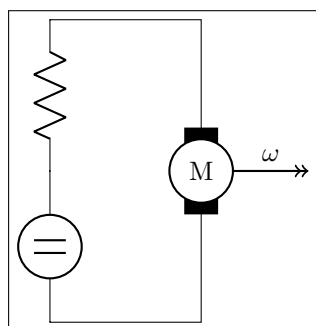


4.3.6 Electro-Mechanical Devices

- MOTOR (node[elmech]{M})



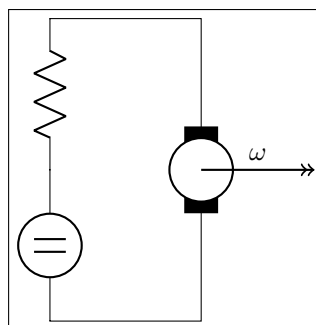
- GENERATOR (node[elmech]{G})



```

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

```

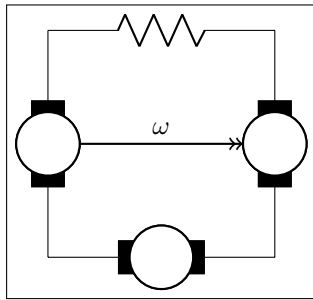


```

1 \begin{circuitikz}
2 \draw (2,0) node[elmech](motor){};
3 \draw (motor.north) |-(0,2) to [R] ++(0,-2) to[
  dcvsource]++(0,-2) -| (motor.bottom);
4 \draw[thick,->>] (motor.center)--++(1.5,0)node[midway,
  above]{$\omega$};
5 \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 access to the anchors:



```

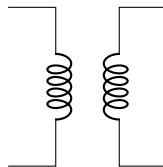
1 \begin{circuitikz}
2 \draw (0,0) to [Telmech=M,n=motor] ++(0,-3) to [
    Telmech=M] ++(3,0) to [Telmech=G,n=generator]
    ++(0,3) to [R] (0,0);
3 \draw[thick,->>] (motor.left)--(generator.left)node[
    midway,above]{$\omega$};
4 \end{circuitikz}

```

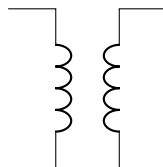
4.4 Double bipoles

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

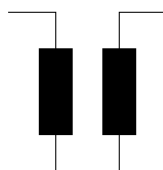
- Transformer (cute inductor) (node[transformer]{})



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

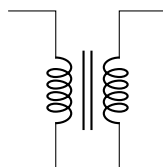


- Transformer (european inductor) (node[transformer]{})

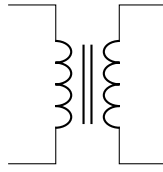


Transformers with core are also available:

- 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]{})



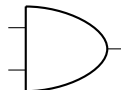
- Coupler, 2 (node[coupler2]{})



4.5 Logic gates

4.5.1 American Logic gates

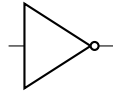
- American AND port (node[american and port]{})



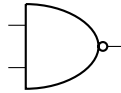
- American OR port (node[american or port]{})



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



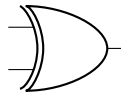
- American NAND port (node[american nand port]{})



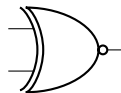
- American NOR port (node[american nor port]{})



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



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



4.5.2 European Logic gates

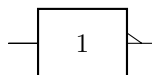
- European AND port (node[european and port]{})



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



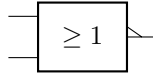
- European NOT port (node[european not port]{})



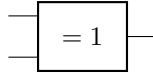
- European NAND port (node[european nand port]{})



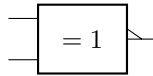
- European NOR port (node[`europaan nor port`]{})



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



- European XNOR port (node[`europaan 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 `europaanports` option is active (or the style `[europaan 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.

- Non-Inverting SCHMITTTRIGGER (node[`schmitt`]{})



- Inverting SCHMITTTRIGGER (node[`invschmitt`]{})



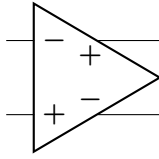
4.6 Amplifiers

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

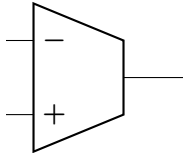


- Fully differential operational amplifier⁶ (node[`fd op amp`]{})

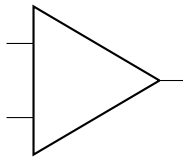
⁶Contributed by Kristofer M. Monisit.



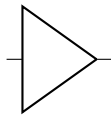
- transconductance amplifier (node[gm amp]{})



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



- Buffer (node[buffer]{})



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]{})



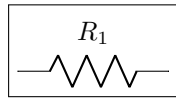
- Unconnected terminal (node[ocirc]{})



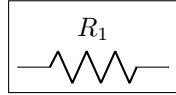
- Diamond-style terminal (node[diamondpole]{})



5 Usage



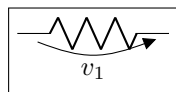
```
1 \begin{circuitikz}
2   \draw (0,0) to[R, l=$R_1$] (2,0);
3 \end{circuitikz}
```



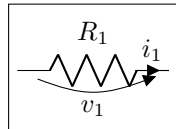
```
1 \begin{circuitikz}
2   \draw (0,0) to[R=$R_1$] (2,0);
3 \end{circuitikz}
```



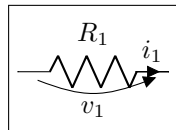
```
1 \begin{circuitikz}
2   \draw (0,0) to[R, i=$i_1$] (2,0);
3 \end{circuitikz}
```



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

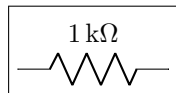


```
1 \begin{circuitikz}
2   \draw (0,0) to[R=$R_1$, i=$i_1$, v=$v_1$] (2,0);
3 \end{circuitikz}
```



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

Long names/styles for the bipoles can be used:

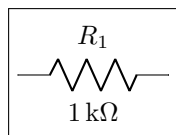


```
1 \begin{circuitikz}\draw
2   (0,0) to[resistor=1<\kilo\ohm>] (2,0)
3 ;\end{circuitikz}
```

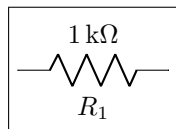
5.1 Labels and Annotations

Since Version 0.7, beside the original label (l) option, there is a new option to place a second label, called annotation (a) at each bipole. Up to now this is a beta-test and there can be problems. For example, up to now this option is not compatible with the concurrent use of voltage labels.

The position of (a) and (l) labels can be adjusted with `_` and `^`, respectively.



```
1 \begin{circuitikz}
2   \draw (0,0) to[R, l=$R_1$, a=1<\kilo\ohm>] (2,0);
3 \end{circuitikz}
```



```
1 \begin{circuitikz}
2   \draw (0,0) to[R, l_=$R_1$, a^=1<\kilo\ohm>] (2,0);
3 \end{circuitikz}
```

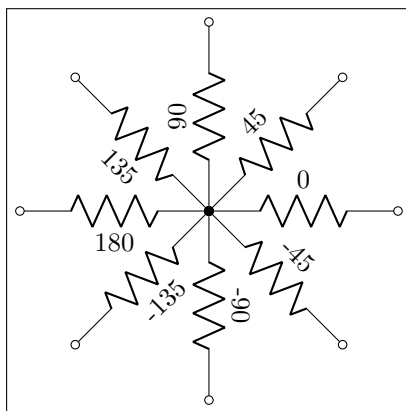
The default orientation of labels is controlled by the options `smartlabels`, `rotatelabels` and `straightlabels` (or the corresponding `label/align` keys). Here are examples to see the differences:



```

1 \begin{circuitikz}
2 \ctikzset{label/align = straight}
3 \def\DIR{0,45,90,135,180,-90,-45,-135}
4 \foreach \i in \DIR {
5   \draw (0,0) to[R=\i, *-o] (\i:2.5);
6 }
7 \end{circuitikz}

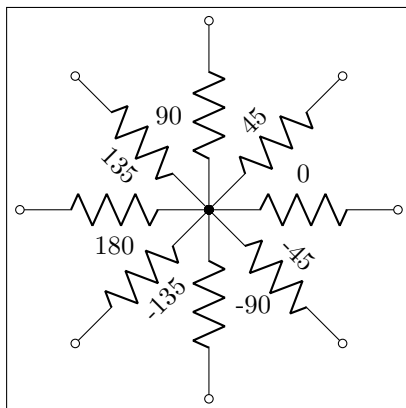
```



```

1 \begin{circuitikz}
2 \ctikzset{label/align = rotate}
3 \def\DIR{0,45,90,135,180,-90,-45,-135}
4 \foreach \i in \DIR {
5   \draw (0,0) to[R=\i, *-o] (\i:2.5);
6 }
7 \end{circuitikz}

```



```

1 \begin{circuitikz}
2 \ctikzset{label/align = smart}
3 \def\DIR{0,45,90,135,180,-90,-45,-135}
4 \foreach \i in \DIR {
5   \draw (0,0) to[R=\i, *-o] (\i:2.5);
6 }
7 \end{circuitikz}

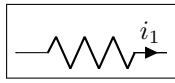
```

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.



```

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

```



```

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

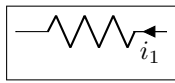
```



```

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

```



```

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

```



```

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

```



```

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

```



```

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

```

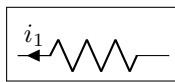


```

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

```

Also



```

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

```



```

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

```



```

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

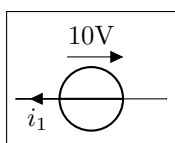
```



```

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

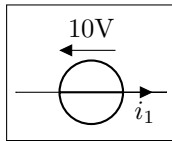
```



```

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

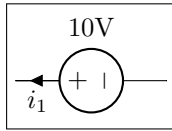
```



```

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

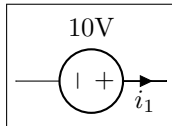
```



```

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

```



```

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

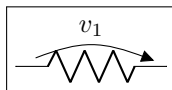
```

5.3 Voltages

See introduction note at Currents (chapter 5.2, page 33)!

5.3.1 European style

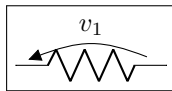
The default, with arrows. Use option `europeanvoltage` or style `[european voltages]`.



```

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

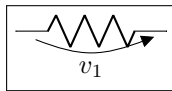
```



```

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

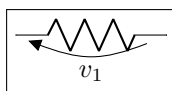
```



```

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

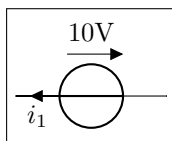
```



```

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

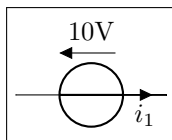
```



```

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

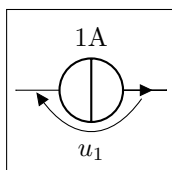
```



```

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

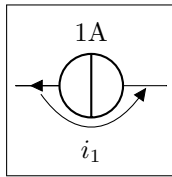
```



```

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

```



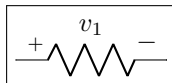
```

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

```

5.3.2 American style

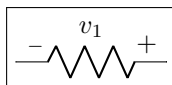
For those who like it (not me). Use option `americanvoltage` or set `[american voltages]`.



```

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

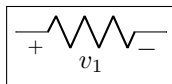
```



```

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

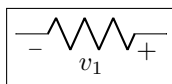
```



```

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

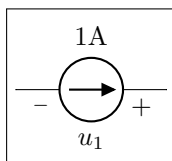
```



```

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

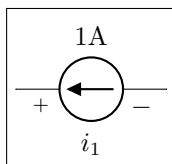
```



```

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

```



```

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

```

5.4 Nodes



```

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

```



```

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

```



```

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

```



```

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

```



```

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

```



```

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

```



```

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

```



```

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

```



```

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

```



```

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

```



```

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

```



```

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

```



```

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

```



```

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

```



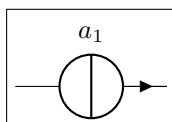
```

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

```

5.5 Special components

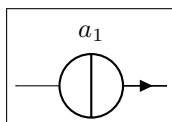
For some components label, current and voltage behave as one would expect:



```

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

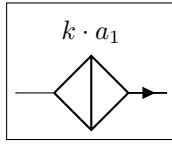
```



```

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

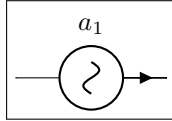
```



```

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

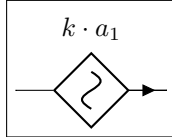
```



```

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

```

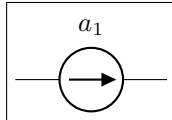


```

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

```

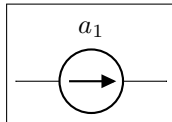
The following results from using the option `americancurrents` or using the style `[american currents]`.



```

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

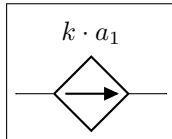
```



```

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

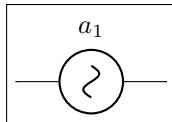
```



```

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

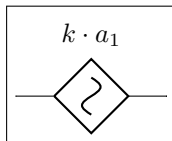
```



```

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

```

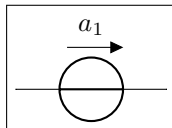


```

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

```

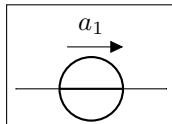
The same holds for voltage sources:



```

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

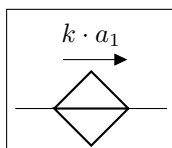
```



```

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

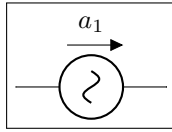
```



```

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

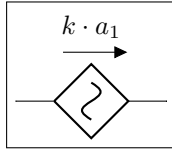
```

```

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

```

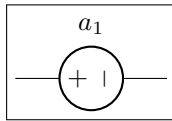


```

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

```

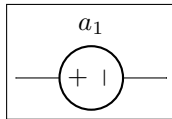
The following results from using the option `americanvoltage` or the style `[american voltages]`.



```

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

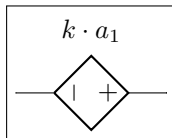
```



```

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

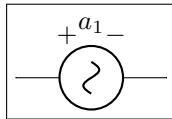
```



```

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

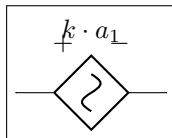
```



```

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

```



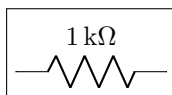
```

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

```

5.6 Integration with siunitx

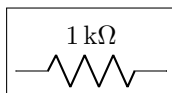
If the option `siunitx` is active (and *not* in ConT_EXt), then the following are equivalent:



```

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

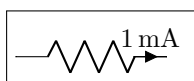
```



```

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

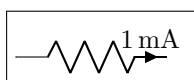
```



```

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

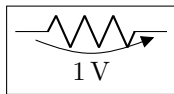
```



```

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

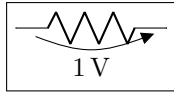
```



```

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

```



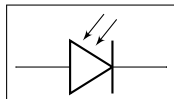
```

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

```

5.7 Mirroring and Inverting

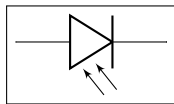
Bipole paths can also mirrored and inverted (or reverted) to change the drawing direction.



```

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

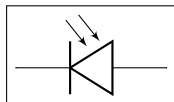
```



```

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

```

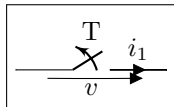


```

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

```

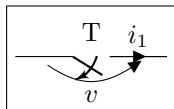
Placing labels, currents and voltages works also, please note, that mirroring and inverting does not influence the positioning of labels and voltages. Labels are by default above/right of the bipole and voltages below/left, respectively.



```

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

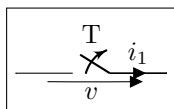
```



```

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

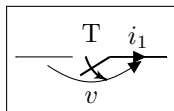
```



```

1 \begin{circuitikz}
2   \draw (0,0) to[ospst=T, invert, i=\i_1$, v=\v$] (2,0);
3 \end{circuitikz}

```

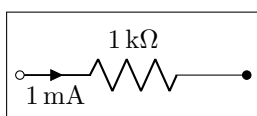


```

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

```

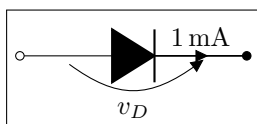
5.8 Putting them together



```

1 \begin{circuitikz}
2   \draw (0,0) to[R=1<\kilo\ohm>,
3     i>_1<\milli\ampere>, o-*] (3,0);
4 \end{circuitikz}

```



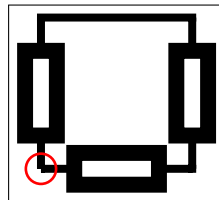
```

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

```

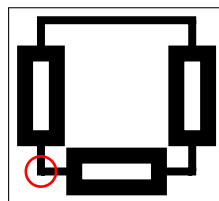
5.9 Line joins between Path Components

Line joins should be calculated correctly, if they were on the same path and if the path is not closed. For example, the following path is not closed correctly (*-cycle* does not work here!):



```
1 \begin{tikzpicture}[line width=3pt,european]
2 \draw (0,0) to[R]++(2,0)to[R]++(0,2)
3   --++(-2,0)to[R]++(0,-2);
4 \draw[red,line width=1pt] circle(2mm);
5 \end{tikzpicture}
```

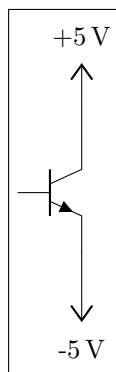
To correct the line ending, there are support shapes to fill the missing rectangle. They can be used like the support shapes(*,o,d) using a dot (.) on one or both ends of a component (have a look at the last resistor in this example):



```
1 \begin{tikzpicture}[line width=3pt,european]
2 \draw (0,0) to[R]++(2,0)to[R]++(0,2)
3   --++(-2,0)to[R,.]++(0,-2);
4 \draw[red,line width=1pt] circle(2mm);
5 \end{tikzpicture}
```

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:



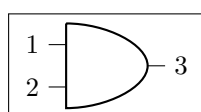
```
1 \begin{circuitikz}
2 \draw (0,0) node[npn] (npn) at (0,0) {};
3 \draw (npn.C) --++(0,0.5) node[vcc]{+5\,\textnormal{V}};
4 \draw (npn.E) --++(0,-0.5) node[vee]{-5\,\textnormal{V}};
5 \end{circuitikz}
```

6.1 Anchors

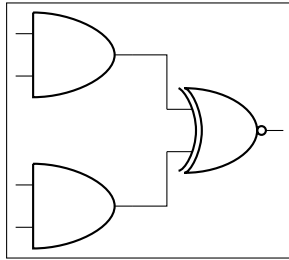
In order to allow connections with other components, all components define anchors.

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:



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

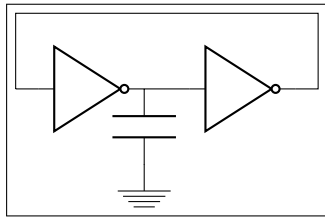


```

1 \begin{circuitikz} \draw
2   (0,2) node[and port] (myand1) {}
3   (0,0) node[and port] (myand2) {}
4   (2,1) node[xnor port] (myxnor) {}
5   (myand1.out) -| (myxnor.in 1)
6   (myand2.out) -| (myxnor.in 2)
7 ;\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**):



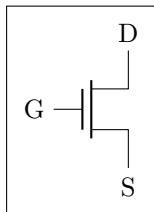
```

1 \begin{circuitikz} \draw
2   (1,0) node[not port] (not1) {}
3   (3,0) node[not port] (not2) {}
4   (0,0) -- (not1.in)
5   (not2.in) -- (not1.out)
6   ++(0,-1) node[ground] {} to[C] (not1.out)
7   (not2.out) -| (4,1) -| (0,0)
8 ;\end{circuitikz}

```

6.1.2 Transistors

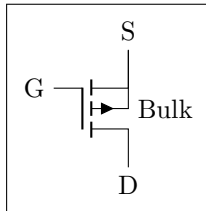
For NMOS, PMOS, NFET, NIGFETE, NIGFETD, PFET, PIGFETE, and PIGFETD transistors one has **base**, **gate**, **source** and **drain** anchors (which can be abbreviated with **B**, **G**, **S** and **D**):



```

1 \begin{circuitikz} \draw
2   (0,0) node[nmos] (mos) {}
3   (mos.gate) node[anchor=east] {G}
4   (mos.drain) node[anchor=south] {D}
5   (mos.source) node[anchor=north] {S}
6 ;\end{circuitikz}

```

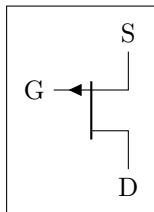


```

1 \begin{circuitikz} \draw
2   (0,0) node[pigfete] (pigfete) {}
3   (pigfete.G) node[anchor=east] {G}
4   (pigfete.D) node[anchor=north] {D}
5   (pigfete.S) node[anchor=south] {S}
6   (pigfete.bulk) node[anchor=west] {Bulk}
7 ;\end{circuitikz}

```

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

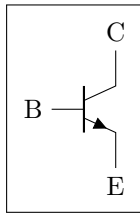


```

1 \begin{circuitikz} \draw
2   (0,0) node[pjfet] (pjfet) {}
3   (pjfet.G) node[anchor=east] {G}
4   (pjfet.D) node[anchor=north] {D}
5   (pjfet.S) node[anchor=south] {S}
6 ;\end{circuitikz}

```

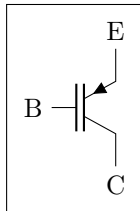
For NPN, PNP, NIGBT, and PIGBT transistors the anchors are **base**, **emitter** and **collector** anchors (which can be abbreviated with **B**, **E** and **C**):



```

1 \begin{circuitikz} \draw
2   (0,0) node[npn] (npn) {}
3   (npn.base) node[anchor=east] {B}
4   (npn.collector) node[anchor=south] {C}
5   (npn.emitter) node[anchor=north] {E}
6 ;\end{circuitikz}

```

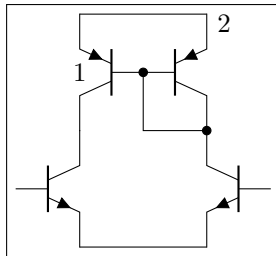


```

1 \begin{circuitikz} \draw
2   (0,0) node[pigbt] (pigbt) {}
3   (pigbt.B) node[anchor=east] {B}
4   (pigbt.C) node[anchor=north] {C}
5   (pigbt.E) node[anchor=south] {E}
6 ;\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 `npn1`):

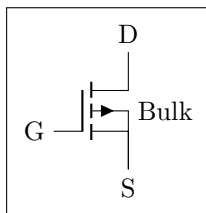


```

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

```

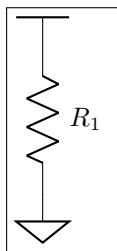
Similarly, transistors and other components can be reflected vertically:



```

1 \begin{circuitikz} \draw
2   (0,0) node[pigfete, yscale=-1] (pigfete) {}
3   (pigfete.bulk) node[anchor=west] {Bulk}
4   (pigfete.G) node[anchor=east] {G}
5   (pigfete.D) node[anchor=south] {D}
6   (pigfete.S) node[anchor=north] {S}
7 ;\end{circuitikz}

```



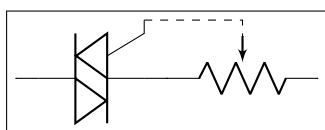
```

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

```

6.1.3 Other tripoles

When inserting a thyristor, 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:

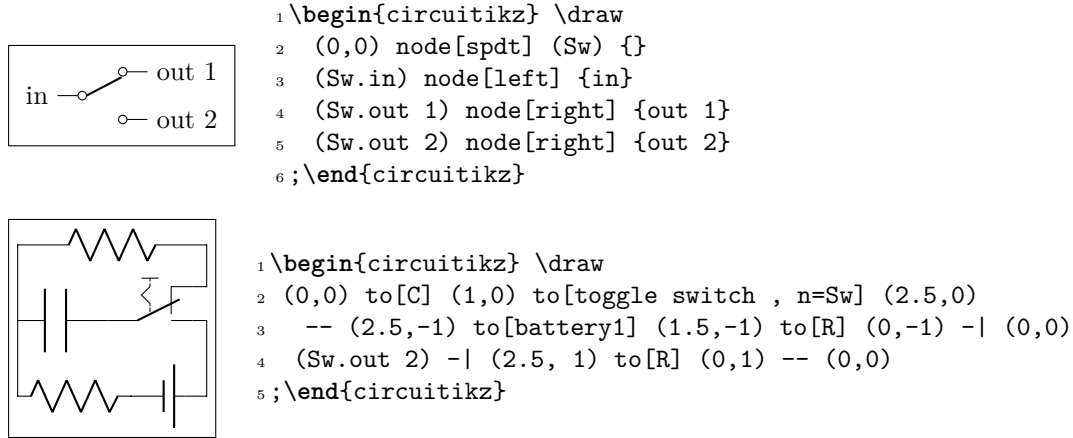


```

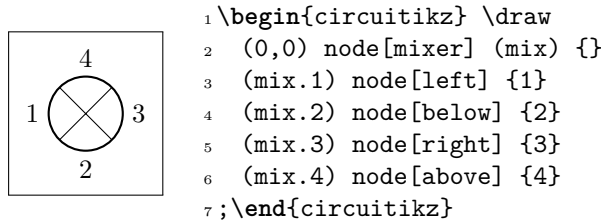
1 \begin{circuitikz} \draw
2   (0,0) to[Tr, n=TRI] (2,0)
3     to[pR, n=POT] (4,0);
4   \draw[dashed] (TRI.G) -| (POT.wiper)
5 ;\end{circuitikz}

```

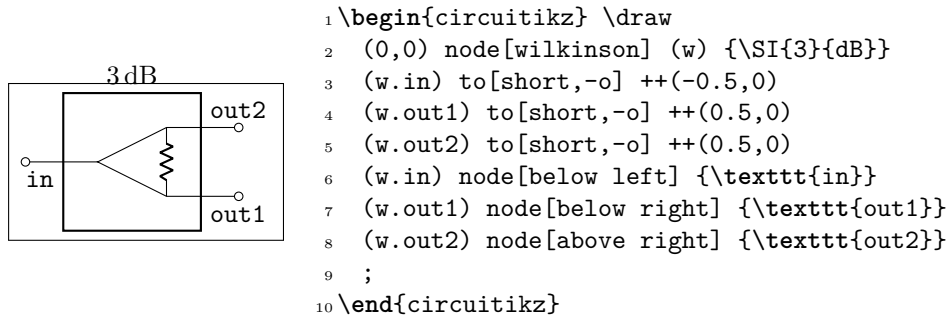
As for the switches:



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

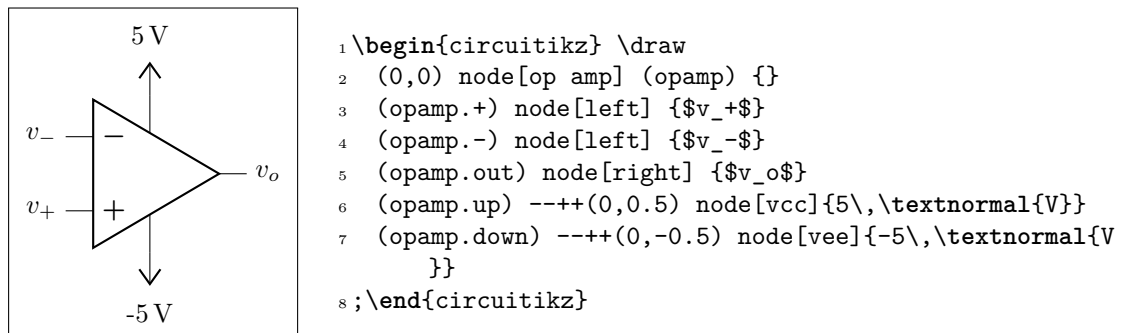


The Wilkinson divider has:

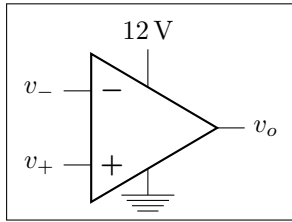


6.1.4 Operational amplifier

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



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

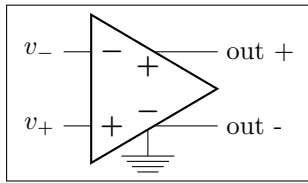


```

1 \begin{circuitikz} \draw
2   (0,0) node[op amp] (opamp) {}
3   (opamp.+) node[left] {$v_+$}
4   (opamp.-) node[left] {$v_-$}
5   (opamp.out) node[right] {$v_o$}
6   (opamp.down) node[ground] {}
7   (opamp.up) ++ (0,.5) node[above] {\SI{12}{\volt}}
8   -- (opamp.up)
9 ;\end{circuitikz}

```

The fully differential op amp defines two outputs:



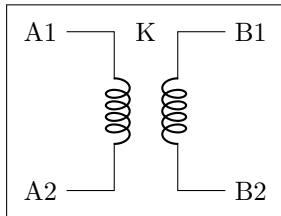
```

1 \begin{circuitikz} \draw
2   (0,0) node[fd op amp] (opamp) {}
3   (opamp.+) node[left] {$v_+$}
4   (opamp.-) node[left] {$v_-$}
5   (opamp.out +) node[right] {out +}
6   (opamp.out -) node[right] {out -}
7   (opamp.down) node[ground] {}
8 ;\end{circuitikz}

```

6.1.5 Double bipoles

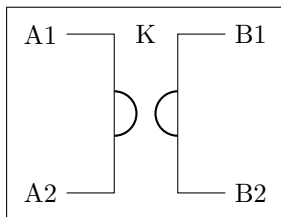
All the (few, actually) double bipoles/quadrupoles have the four anchors, two for each port. The first port, to the left, is port A, having the anchors A1 (up) and A2 (down); same for port B. They also expose the **base** anchor, for labelling:



```

1 \begin{circuitikz} \draw
2   (0,0) node[transformer] (T) {}
3   (T.A1) node[anchor=east] {A1}
4   (T.A2) node[anchor=east] {A2}
5   (T.B1) node[anchor=west] {B1}
6   (T.B2) node[anchor=west] {B2}
7   (T.base) node{K}
8 ;\end{circuitikz}

```

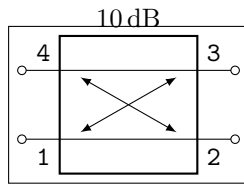


```

1 \begin{circuitikz} \draw
2   (0,0) node[gyrator] (G) {}
3   (G.A1) node[anchor=east] {A1}
4   (G.A2) node[anchor=east] {A2}
5   (G.B1) node[anchor=west] {B1}
6   (G.B2) node[anchor=west] {B2}
7   (G.base) node{K}
8 ;\end{circuitikz}

```

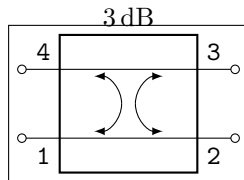
However:



```

1 \begin{circuitikz} \draw
2   (0,0) node[coupler] (c) {\SI{10}{dB}}
3   (c.1) to[short,-o] ++(-0.5,0)
4   (c.2) to[short,-o] ++(0.5,0)
5   (c.3) to[short,-o] ++(0.5,0)
6   (c.4) to[short,-o] ++(-0.5,0)
7   (c.1) node[below left] {\texttt{1}}
8   (c.2) node[below right] {\texttt{2}}
9   (c.3) node[above right] {\texttt{3}}
10  (c.4) node[above left] {\texttt{4}}
11 ;
12 \end{circuitikz}

```



```

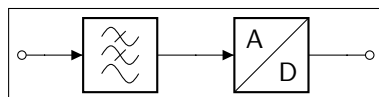
1 \begin{circuitikz} \draw
2   (0,0) node[coupler2] (c) {\SI{3}{dB}}
3   (c.1) to[short,-o] ++(-0.5,0)
4   (c.2) to[short,-o] ++(0.5,0)
5   (c.3) to[short,-o] ++(0.5,0)
6   (c.4) to[short,-o] ++(-0.5,0)
7   (c.1) node[below left] {\texttt{1}}
8   (c.2) node[below right] {\texttt{2}}
9   (c.3) node[above right] {\texttt{3}}
10  (c.4) node[above left] {\texttt{4}}
11 ;
12 \end{circuitikz}

```

6.2 Input arrows

Two ports

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



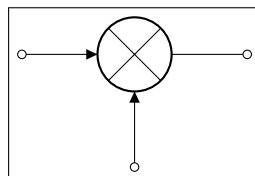
```

1 \begin{circuitikz} \draw
2   (0,0) to[short,o-] ++(0.3,0)
3   to[lowpass,>] ++(2,0)
4   to[adc,>] ++(2,0)
5   to[short,-o] ++(0.3,0);
6 \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:



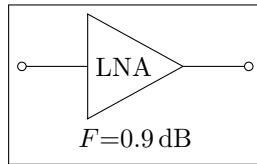
```

1 \begin{circuitikz} \draw
2   (0,0) node[mixer] (m) {}
3   (m.1) to[short,-o] ++(-1,0)
4   (m.2) to[short,-o] ++(0,-1)
5   (m.3) to[short,-o] ++(1,0)
6   (m.1) node[inputarrow] {}
7   (m.2) node[inputarrow,rotate=90] {};
8 \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=`).



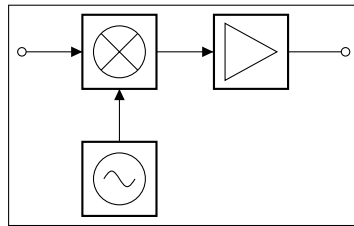
```

1 \begin{circuitikz}
2   \ctikzset{bipoles/amp/width=0.9}
3   \draw (0,0) to[amp,t=LNA,l_=$F{=}0.9\,$dB,o-o] ++(3,0);
4 \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.



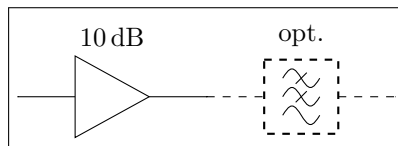
```

1 \begin{circuitikz} \draw
2   (0,0) node[mixer,box,anchor=east] (m) {}
3   to[amp,box,>,-o] ++(2.5,0)
4   (m.west) node[inputarrow] {} to[short,-o]
5   ++(-0.8,0)
6   (m.south) node[inputarrow,rotate=90] {} --
7   ++(0,-0.7) node[oscillator,box,anchor=north] {};
8 \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.



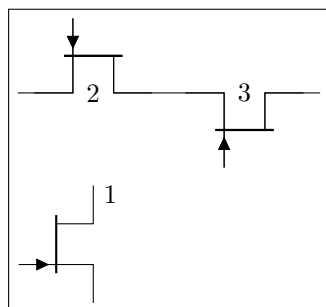
```

1 \begin{circuitikz}
2   \draw (0,0) to[amp,l=\SI{10}{dB}] ++(2.5,0);
3   \draw[dashed] (2.5,0) to[lowpass,l=opt.]
4   ++(2.5,0);
5 \end{circuitikz}

```

6.6 Transistor paths

For syntactical convenience transistors can be placed using the normal path notation used for bipoles. The transistor 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:

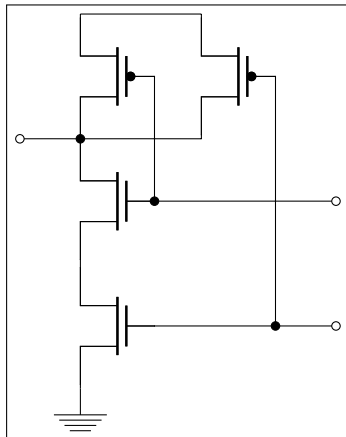


```

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

```

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



```

1 \begin{circuitikz} \draw[yscale=1.1, xscale=.8]
2   (2,4.5) -- (0,4.5) to[Tpmos, n=p1] (0,3)
3     to[Tnmos, n=n1] (0,1.5)
4     to[Tnmos, n=n2] (0,0) node[ground] {}
5   (2,4.5) to[Tpmos,n=p2] (2,3) to[short, -*] (0,3)
6   (p1.G) -- (n1.G) to[short, *-o] ($(n1.G)+(3,0)$)
7   (n2.G) ++(2,0) node[circ] {} -| (p2.G)
8   (n2.G) to[short, -o] ($(n2.G)+(3,0)$)
9   (0,3) to[short, -o] (-1,3)
10;\end{circuitikz}

```

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

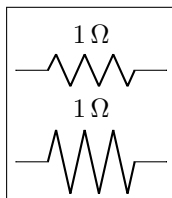
7 Customization

7.1 Parameters

Pretty much all `CircuitikZ` 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.

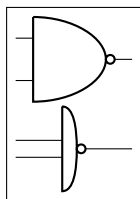
Shape of the components (on a per-component-class basis)



```

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

```

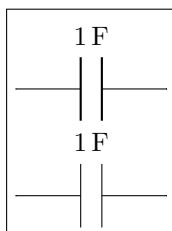


```

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

```

Thickness of the lines (globally)

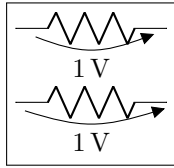


```

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

```

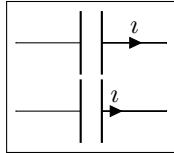
Global properties Of voltage and current



```

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

```

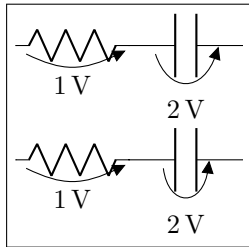


```

1\tikz \draw (0,0) to[C, i=${\imath}$] (2,0); \par
2\ctikzset{current/distance = .2}
3\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:

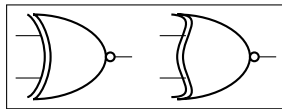


```

1\tikz \draw (0,0) to[R, v=1<\volt>] (1.5,0)
2      to[C, v=2<\volt>] (3,0); \par
3\ctikzset{bipoles/capacitor/voltage/%
4      distance from node/.initial=.7}
5\tikz \draw (0,0) to[R, v=1<\volt>] (1.5,0)
6      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:



```

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

```

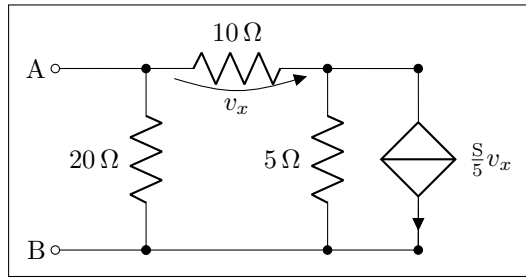
7.2 Components size

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

⁷That is, how distant from the initial and final points of the path the arrow starts and ends.

⁸Controlling how high the bump of the arrow is — how curved it is.

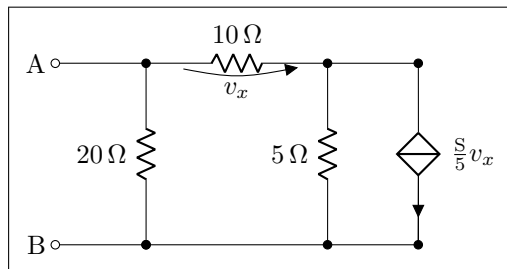
⁹Controlling how distant from the bipole the voltage label will be.



```

1 \ctikzset{bipoles/length=1.4cm}
2 \begin{circuitikz}[scale=1.2]\draw
3   (0,0) node[anchor=east] {B}
4     to[short, o-*] (1,0)
5     to[R=20<\ohm>, *-] (1,2)
6     to[R=10<\ohm>, v=$v_x$] (3,2) -- (4,2)
7     to[cI=$\frac{\si{siemens}}{5} v_x$, *-] (4,0) -- (3,0)
8     to[R=5<\ohm>, *-] (3,2)
9   (3,0) -- (1,0)
10  (1,2) to[short, -o] (0,2) node[anchor=east]{A}
11 ;\end{circuitikz}

```



```

1 \ctikzset{bipoles/length=.8cm}
2 \begin{circuitikz}[scale=1.2]\draw
3   (0,0) node[anchor=east] {B}
4     to[short, o-*] (1,0)
5     to[R=20<\ohm>, *-] (1,2)
6     to[R=10<\ohm>, v=$v_x$] (3,2) -- (4,2)
7     to[cI=$\frac{\siemens}{5} v_x$, *-] (4,0) -- (3,0)
8     to[R=5<\ohm>, *-] (3,2)
9   (3,0) -- (1,0)
10  (1,2) to[short, -o] (0,2) node[anchor=east]{A}
11 ;\end{circuitikz}

```

7.3 Colors

The color of the components is stored in the key `\circuitikzbasekey/color`. CircuiTikZ tries to follow the color set in TikZ, 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



and

```

1 \begin{circuitikz} \draw[red]
2   (0,2) node[and port] (myand1) {}
3   (0,0) node[and port] (myand2) {}
4   (2,1) node[xnor port] (myxnor) {}
5   (myand1.out) -| (myxnor.in 1)
6   (myand2.out) -| (myxnor.in 2)
7 ;\end{circuitikz}

```



```

1 \begin{circuitikz} \draw[color=red]
2   (0,2) node[and port] (myand1) {}
3   (0,0) node[and port] (myand2) {}
4   (2,1) node[xnor port] (myxnor) {}
5   (myand1.out) -| (myxnor.in 1)
6   (myand2.out) -| (myxnor.in 2)
7 ;\end{circuitikz}

```

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

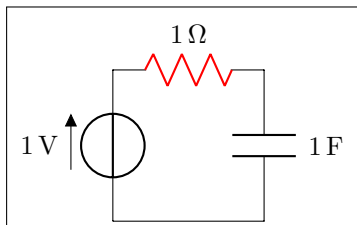


```

1 \begin{circuitikz} \draw
2   (0,0) node[pnp, color=blue] (pnp2) {}
3   (pnp2.B) node[pnp, xscale=-1, anchor=B, color=brown] (pnp1) {}
4   (pnp1.C) node[npn, anchor=C, color=green] (nnp1) {}
5   (pnp2.C) node[npn, xscale=-1, anchor=C, color=magenta] (nnp2) {}
6   (pnp1.E) -- (pnp2.E) (nnp1.E) -- (nnp2.E)
7   (pnp1.B) node[circ] {} |- (pnp2.C) node[circ] {}
8 ;\end{circuitikz}

```

The all-in-one stream of bipoles poses some challenges, 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`:

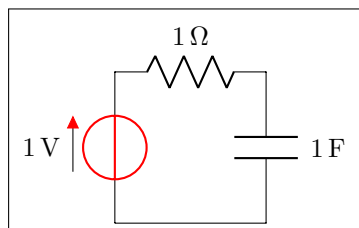


```

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

```

Which, for some bipoles, can be frustrating:

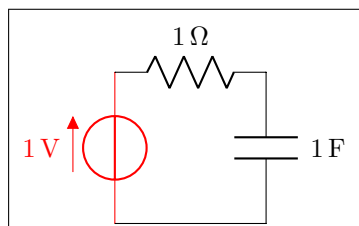


```

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

```

The only way out is to specify different paths:



```

1 \begin{circuitikz} \draw[color=red]
2   (0,0) to[V=1<\volt>, color=red] (0,2);
3   \draw (0,2) to[R=1<\ohm>] (2,2)
4         to[C=1<\farad>] (2,0) -- (0,0)
5 ;\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.

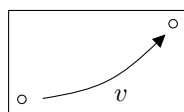
A: The TikZ manual states:

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!



```

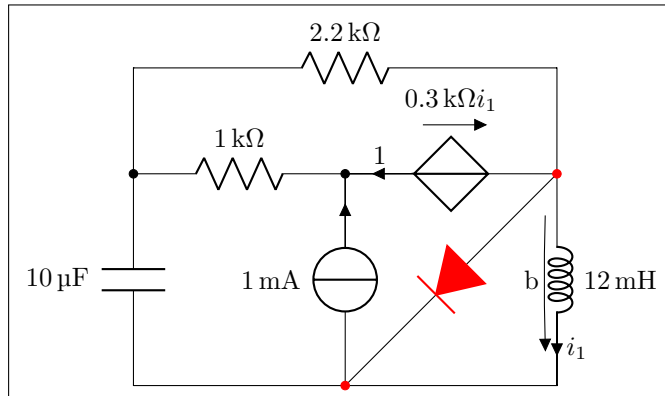
1 \begin{circuitikz} \draw
2   node[ocirc] (A) at (0,0) {}
3   node[ocirc] (B) at (2,1) {}
4   (A) to[open, v=$v$] (B)
5 ;\end{circuitikz}

```

Q: I cannot write `to[R = $R_1=12V$]` nor `to[ospst = open, 3s]`: I get errors.

A: It is a limitation of the TikZ parser. Use `to[R = $R_1{= }12V$]` and `to[ospst = open{,} 3s]` instead.

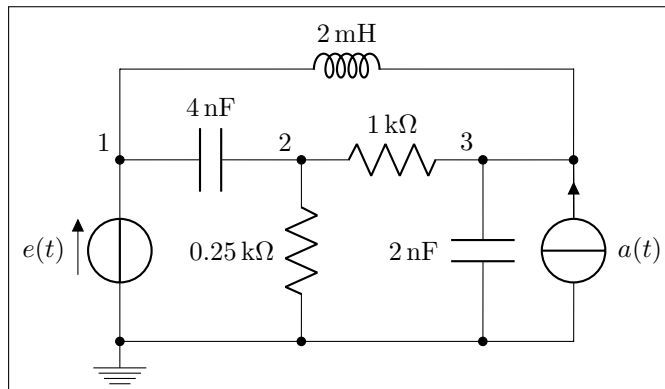
9 Examples



```

1 \begin{circuitikz}[scale=1.4]\draw
2   (0,0) to[C, l=10<\micro\farad>] (0,2) -- (0,3)
3     to[R, l=2.2<\kilo\ohm>] (4,3) -- (4,2)
4     to[L, l=12<\milli\henry>, i=$i_1$,v=b] (4,0) -- (0,0)
5   (4,2) { to[D*, **-, color=red] (2,0) }
6   (0,2) to[R, l=1<\kilo\ohm>, *-] (2,2)
7     to[cV, i=1,v=$\SI{.3}{\kilo\ohm} i_1$] (4,2)
8   (2,0) to[I, i=1<\milli\ampere>, -*] (2,2)
9 ;\end{circuitikz}

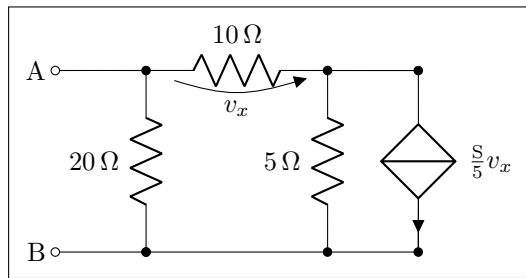
```



```

1 \begin{circuitikz}[scale=1.2]\draw
2   (0,0) node[ground] {}
3     to[V=$e(t)$, **] (0,2) to[C=4<\nano\farad>] (2,2)
4     to[R, l=.25<\kilo\ohm>, **] (2,0)
5   (2,2) to[R=1<\kilo\ohm>] (4,2)
6     to[C, l=2<\nano\farad>, **] (4,0)
7   (5,0) to[I, i_=$a(t)$, -*] (5,2) -- (4,2)
8   (0,0) -- (5,0)
9   (0,2) -- (0,3) to[L, l=2<\milli\henry>] (5,3) -- (5,2)
10
11 {[anchor=south east] (0,2) node {1} (2,2) node {2} (4,2) node {3}}
12 ;\end{circuitikz}

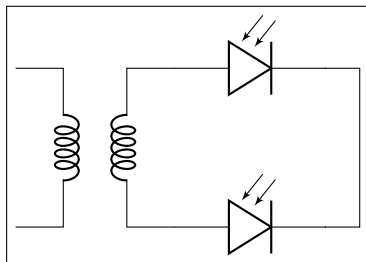
```



```

1 \begin{circuitikz}[scale=1.2]\draw
2   (0,0) node[anchor=east] {B}
3     to[short, o-*] (1,0)
4     to[R=20<\ohm>, *-] (1,2)
5     to[R=10<\ohm>, v=$v_x$] (3,2) -- (4,2)
6     to[cI=$\frac{\siemens}{5} v_x$, *-] (4,0) -- (3,0)
7     to[R=5<\ohm>, *-] (3,2)
8   (3,0) -- (1,0)
9   (1,2) to[short, -o] (0,2) node[anchor=east]{A}
10;\end{circuitikz}

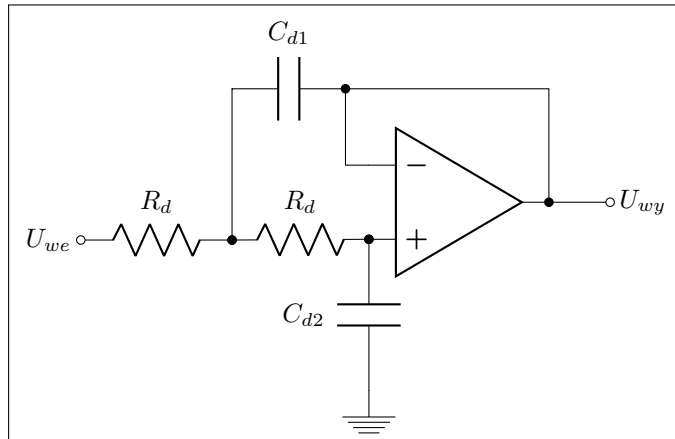
```



```

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

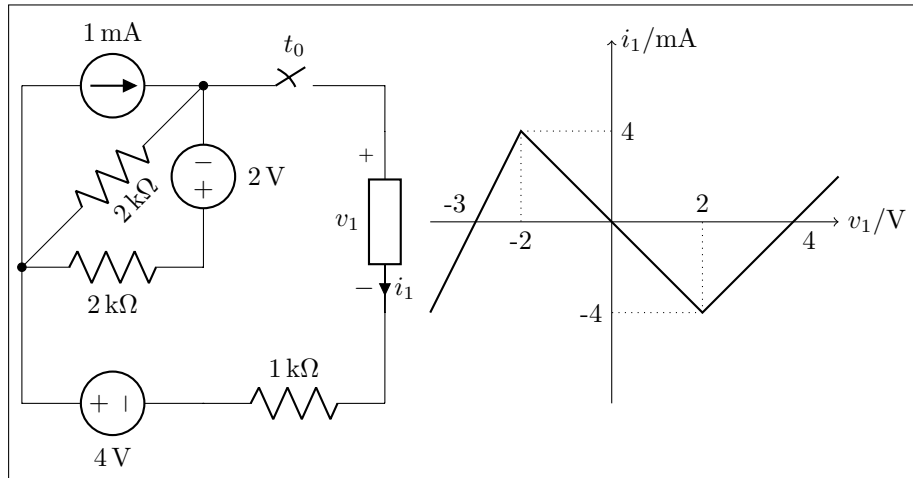
```

```

1 \begin{circuitikz}[scale=1]\draw
2   (5,.5) node [op amp] (opamp) {}
3   (0,0) node [left] {\mathcal{U}_{we}} to [R, l=\mathcal{R}_d, o-*] (2,0)
4   to [R, l=\mathcal{R}_d, *-] (opamp.+)
5   to [C, l=\mathcal{C}_{d2}, *-] ($(opamp.+) + (0,-2)$) node [ground] {}
6   (opamp.out) |- (3.5,2) to [C, l=\mathcal{C}_{d1}, *-] (2,2) to [short] (2,0)
7   (opamp.-) -| (3.5,2)
8   (opamp.out) to [short, *-o] (7,.5) node [right] {\mathcal{U}_{wy}}
9 \end{circuitikz}

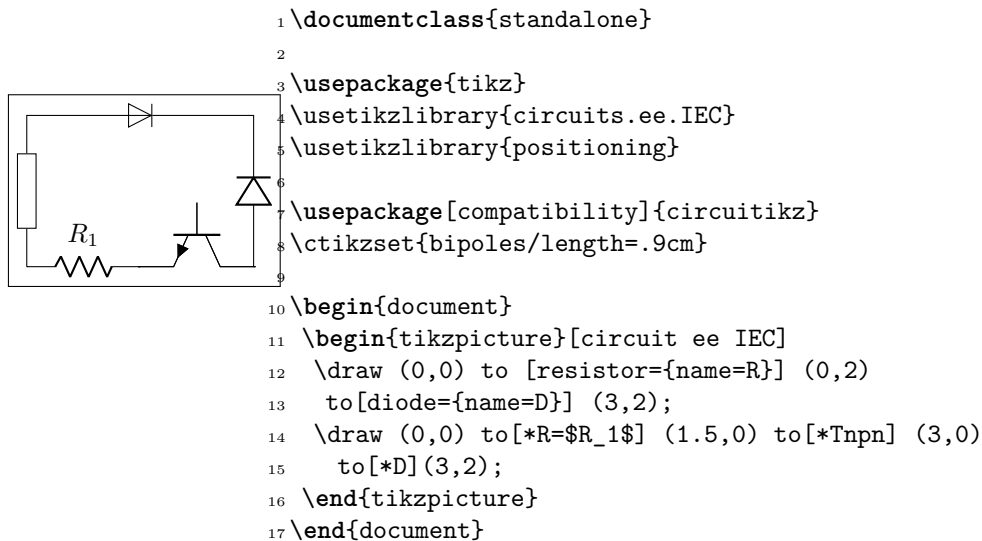
```



```

1 \begin{circuitikz}[scale=1.2, american]\draw
2   (0,2) to[I=1<\milli\ampere>] (2,2)
3     to[R, l_2=2<\kilo\ohm>, *-] (0,0)
4     to[R, l_2=2<\kilo\ohm>] (2,0)
5     to[V, v_2=2<\volt>] (2,2)
6     to[cspst, l=$t_0$] (4,2) -- (4,1.5)
7     to [generic, i=$i_1$, v=$v_1$] (4,-.5) -- (4,-1.5)
8   (0,2) -- (0,-1.5) to[V, v_4=4<\volt>] (2,-1.5)
9     to [R, l=1<\kilo\ohm>] (4,-1.5);
10
11 \begin{scope}[xshift=6.5cm, yshift=.5cm]
12   \draw [->] (-2,0) -- (2.5,0) node[anchor=west] {$v_1/\text{volt}$};
13   \draw [->] (0,-2) -- (0,2) node[anchor=west] {$i_1/\text{mA}$};
14   \draw (-1,0) node[anchor=north] {-2} (1,0) node[anchor=south] {2}
15         (0,1) node[anchor=west] {4} (0,-1) node[anchor=east] {-4}
16         (2,0) node[anchor=north west] {4}
17         (-1.5,0) node[anchor=south east] {-3};
18   \draw [thick] (-2,-1) -- (-1,1) -- (1,-1) -- (2,0) -- (2.5,.5);
19   \draw [dotted] (-1,1) -- (-1,0) (1,-1) -- (1,0)
20             (-1,1) -- (0,1) (1,-1) -- (0,-1);
21 \end{scope}
22 \end{circuitikz}

```

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 git (unreleased)
 - Fixed unwanted line through components if target coordinate is a name of a node
- Version 0.8 (2017-03-08)
 - Allow use of voltage label at a [short]
 - Correct line joins between path components (to[...])
 - New Pole-shape $\cdot\cdot$ to fill perpendicular joins
 - Fixed direction of controlled american current source
 - Fixed incorrect scaling of magnetron
 - Fixed: Number of american inductor coils not adjustable
 - Fixed Battery Symbols and added new battery2 symbol
 - Added non-inverting Schmitttrigger
- Version 0.7 (2016-09-08)
 - Added second annotation label, showing, e.g., the value of an component
 - Added new symbol: magnetron
 - Fixed name conflict of diamond shape with tikz.shapes package
 - Fixed varcap symbol at small scalings
 - New packet-option “straightvoltages, to draw straight(no curved) voltage arrows
 - New option “invert” to revert the node direction at paths
 - Fixed american voltage label at special sources and battery
 - Fixed/rotated battery symbol(longer lines by default positive voltage)
 - New symbol Schmitttrigger

- 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 global 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 Lnigt, Lpigt
 - 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 `europianresistor`, `europianinductor`, `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 `TikZ` 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

- adc, 20
- adder, 27
- afuse, 11
- ageneric, 9
- american and port, 30
- american controlled current source, 22
- american controlled voltage source, 21
- american current source, 17
- american gas filled surge arrester, 15
- american inductor, *see* L
- american nand port, 31
- american nor port, 31
- american not port, 31
- american or port, 30
- american potentiometer, *see* pR
- american resistor, *see* R
- american voltage source, 17
- american xnor port, 31
- american xor port, 31
- ammeter, 8
- amp, 20
- antenna, 7
- asymmetric fuse, *see* afuse

- bandpass, 19
- barrier, 15
- battery, 17
- battery1, 17
- battery2, 17
- buffer, 33

- C, *see* capacitor
- capacitor, 15
- cground, 7
- circ, 33
- circulator, 27
- cisourcesin, *see* controlled sinusoidal current source
- closing switch, 19
- controlled isourcesin, *see* controlled sinusoidal current source
- controlled sinusoidal current source, 22
- controlled sinusoidal voltage source, 22
- controlled vsourcesin, *see* controlled sinusoidal voltage source
- coupler, 30
- coupler2, 30
- csI, *see* controlled sinusoidal current source
- cspst, *see* closing switch
- csV, *see* controlled sinusoidal voltage source
- curarrow, 33
- cute inductor, *see* L

- cvsourcesin, *see* controlled sinusoidal voltage source

- D*, *see* full diode
- D-, *see* stroke diode
- dac, 20
- damper, 19
- dcisource, 18
- dcvsource, 18
- detector, 21
- diamondpole, 33
- Do, *see* empty diode
- dsp, 20

- eC, *see* ecapacitor
- ecapacitor, 15
- elko, *see* ecapacitor
- elmech, 28
- empty diode, 11
- empty led, 12
- empty photodiode, 11
- empty Schottky diode, 11
- empty thyristor, 14
- empty triac, 14
- empty tunnel diode, 11
- empty varcap, 12
- empty Zener diode, 11
- empty ZZener diode, 11
- esource, 18
- european and port, 31
- european controlled current source, 22
- european controlled voltage source, 21
- european current source, 17
- european gas filled surge arrester, 15
- european inductor, *see* L
- european nand port, 31
- european nor port, 32
- european not port, 31
- european or port, 31
- european potentiometer, *see* pR
- european resistor, *see* R
- european variable resistor, *see* vR
- european voltage source, 17
- european xnor port, 32
- european xor port, 32

- fd op amp, 32
- fft, 20
- full diode, 12
- full led, 12
- full photodiode, 12
- full Schottky diode, 12

full thyristor, 14
 full triac, 14
 full tunnel diode, 12
 full varcap, 13
 full Zener diode, 12
 full ZZener diode, 12
 fullgeneric, 9
 fuse, 11

 generic, 9
 gm amp, 33
 ground, 7
 gyrator, 30

 highpass, 20

 inputarrow, 33
 invschmitt, 32
 ioosource, 18
 isfet, 26
 isourcesin, *see* sinusoidal current source

 L, 16
 lamp, 9
 leD*, *see* full led
 leD-, *see* stroke led
 leDo, *see* empty led
 Lnigbt, 23
 lowpass, 20
 Lpigt, 23

 magnetron, 26
 mass, 19
 match, 8
 memristor, 9
 mixer, 27
 Mr, *see* memristor

 nfet, 25
 nground, 7
 nigbt, 23
 nigfetd, 25
 nigfete, 25
 nigfete,solderdot, 25
 nigfetebulk, 25
 njfet, 26
 nmos, 22, 24
 npn, 23
 npn,photo, 23

 ocirc, 33
 ohmmeter, 8
 op amp, 32
 open, 9
 opening switch, 19

oscillator, 27
 ospst, *see* opening switch

 pC, *see* polar capacitor
 pD*, *see* full photodiode
 pD-, *see* stroke photodiode
 pDo, *see* empty photodiode
 pfet, 25
 pground, 7
 phaseshifter, 21
 photoresistor, *see* phR
 phR, 10
 piattenuator, 20
 piezoelectric, 15
 pigbt, 23
 pigfetd, 26
 pigfete, 25
 pigfetebulk, 26
 pjfet, 26
 plain amp, 33
 pmos, 22, 24
 pmos,emptycircle, 24
 pnp, 23
 pnp,photo, 23
 polar capacitor, 15
 pR, 10
 push button, 19
 pvsource, 18
 PZ, *see* piezoelectric

 R, 10
 rground, 7
 rxantenna, 8

 schmitt, 32
 sD*, *see* full Schottky diode
 sD-, *see* stroke Schottky diode
 sDo, *see* empty Schottky diode
 sground, 7
 short, 9
 sI, *see* sinusoidal current source
 sinusoidal current source, 18
 sinusoidal voltage source, 18
 spdt, 28
 spring, 19
 spst, *see* switch
 square voltage source, 18
 squid, 15
 sqV, *see* square voltage source
 stroke diode, 13
 stroke led, 13
 stroke photodiode, 13
 stroke Schottky diode, 13
 stroke thyristor, 14

stroke tunnel diode, 13
stroke varcap, 13
stroke Zener diode, 13
stroke ZZener diode, 13
sV, *see* sinusoidal voltage source
switch, 19

tattenuator, 21
tD*, *see* full tunnel diode
tD-, *see* stroke tunnel diode
tDo, *see* empty tunnel diode
tfullgeneric, 9
tgeneric, 9
tground, 7
thermistor, *see* thR
thermistor ntc, *see* thRn
thermistor ptc, *see* thRp
thermocouple, 10
thR, 11
thRn, 11
thRp, 11
thyristor, 14
TL, 16
tline, *see* TL
tlinestub, 8
toggle switch, 28
Tr, *see* triac
Tr*, *see* full triac
transformer, 29
transformer core, 29, 30
transmission line, *see* TL
triac, 14
Tro, *see* empty triac
tV, *see* vsourcetri
twoport, 19
txantenna, 8
Ty, *see* thyristor

Ty*, *see* full thyristor
Ty-, *see* stroke thyristor
Tyo, *see* empty thyristor

vamp, 20
variable american inductor, *see* vL
variable american resistor, *see* vR
variable capacitor, 15
variable cute inductor, *see* vL
variable european inductor, *see* vL
varistor, 10
vC, *see* variable capacitor
VC*, *see* full varcap
VC-, *see* stroke varcap
vcc, 8
VCo, *see* empty varcap
vco, 19
vee, 8
vL, 16
voltmeter, 8
voosource, 18
vphaseshifter, 21
vpiattenuator, 21
vR, 10
vsourcesin, *see* sinusoidal voltage source
vsourcesquare, *see* square voltage source
vsourcetri, 18
vtattenuator, 21

wilkinson, 27

zD*, *see* full Zener diode
zD-, *see* stroke Zener diode
zDo, *see* empty Zener diode
zzD*, *see* full ZZener diode
zzD-, *see* stroke ZZener diode
zzDo, *see* empty ZZener diode