Creating bipole circuit components for circuitikz.

John Kormylo

Before proceeding further, you should have read An easy way to create you own circuitikz symbols and the first half of Creating multipole components for circuitikz (the second half will make more sense later). If you decide to go with the easy method, you may want to copy the source code here for some new components which will be very useful for that goal.

To make life more difficult, LaTeX restricts the use of @ in macro names in some arcane manner. Consequently, you will need to add the following code before you begin your component definitions.

```
\makeatletter
% used to process styles for to-path
\def\TikzBipolePath#1#2{\pgf@circ@bipole@path{#1}{#2}}

% restore size value for bipole definitions
\pgf@circ@Rlen = \pgfkeysvalueof{/tikz/circuitikz/bipoles/length}
\makeatother

\newlength{\ResUp} \newlength{\ResDown}
\newlength{\ResLeft} \newlength{\ResRight}
```

If you copy a component definition from the .sty files, it will sometimes use registers `\pgf@circ@res@up`, `\pgf@circ@res@down`, `\pgf@circ@res@left` and/or `\pgf@circ@res@right`. One can replace them with the above lengths, which can be computed using:

```
 \pgfextractx{\ResRight}{\northeast}
 \pgfextracty{\ResUp}{\northeast}
 \pgfextractx{\ResLeft}{\southwest}
 \pgfextracty{\ResDown}{\southwest}
```

Wait until you need them since they change values.

Circuitikz contains a database using the `\pgfkeys` utility which has all sorts of information about specific components. It is also used to let `to[myname]` know what to do with `myname`. And as with the multipole components, you will be creating a new shape. Fortunately, `\pgfcircdeclarebipole` will do almost all of the work. You just have to draw the component using graphics primitives.

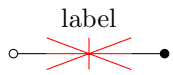
The first example is an invisible resistor, which I named `empty`. One can create it using the following:

```
% empty (resistor)
\ctikzset{bipoles/empty/height/.initial=.30}
\ctikzset{bipoles/empty/width/.initial=.80}
\ctikzset{bipoles/empty/voltage/distance from node/.initial=.4}
\ctikzset{bipoles/empty/voltage/bump b/.initial=2}

\pgfcircdeclarebipole{} % no extra anchors
 {\ctikzvalof{bipoles/empty/height}} % offset to label
 {empty} % name of component
 {\ctikzvalof{bipoles/empty/height}} % height of box
 {\ctikzvalof{bipoles/empty/width}} % width of box
 {} % draw nothing
```

```
\def\emptypath#1{\TikzBipolePath{empty}{#1}} % create empty style
\tikzset{empty/.style = {\circuitikzbasekey, /tikz/to path=\emptypath, l=#1}}
```

and it looks like this:



The red lines were added to show where the compass anchors are.

The first section defines everything one needs to know about this component. Basically, I copied everything used by the `generic` component. The next section creates the new shape, and the last creates the new style.

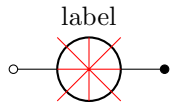
The only thing really complicated with bipoles is the American/European style conflict. For example, the are five different bipoles named `generic`.

The next example is an empty circle, which is based on the `lamp` component.

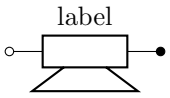
```
% Circle (lamp)
\ctikzset{bipoles/circle/height/.initial=.60}
\ctikzset{bipoles/circle/width/.initial=.60}

\pgfcircdeclarebipole{}
 {\ctikzvalof{bipoles/circle/height}}
 {circle}
 {\ctikzvalof{bipoles/circle/height}}
 {\ctikzvalof{bipoles/circle/width}}
 {
  \pgfsetlinewidth{\pgfkeysvalueof{/tikz/circuitikz/bipoles/thickness}\pgfstartlinewidth}
  \pgfextracty{\ResUp}{\northeast}
  \pgfextractx{\ResLeft}{\southwest}
  \pgfpathellipse{\pgfpointorigin}{\pgfpoint{0}{\ResUp}}{\pgfpoint{\ResLeft}{0}}
  \pgfusepath{draw}
 }
\def\circlepath#1{\TikzBipolePath{circle}{#1}}
\tikzset{circle/.style = {\circuitikzbasekey, /tikz/to path=\circlepath, l=#1}}
```

It looks like this:



Again, the red lines show the locations of the compass anchors. Note that the terminal points are visible, as was **not** the case for the `mystery` component.

And, of course, I can't quit before I create `speaker`.

```
% speaker
\ctikzset{bipoles/speaker/height/.initial=.30}
\ctikzset{bipoles/speaker/width/.initial=.80}

\pgfcircdeclarebipole{}
 {\ctikzvalof{bipoles/speaker/height}}
 {speaker}
 {\ctikzvalof{bipoles/speaker/height}}
 {\ctikzvalof{bipoles/speaker/width}}
 {
  \pgfsetlinewidth{\pgfkeysvalueof{/tikz/circuitikz/bipoles/thickness}\pgfstartlinewidth}
  \pgfextractx{\ResRight}{\northeast}
  \pgfextractx{\ResLeft}{\southwest}
  \pgfextracty{\ResDown}{\southwest}
  \pgfpathrectanglecorners{\southwest}{\northeast}
  \pgfpathmoveto{\pgfpoint{0.5\ResRight}{\ResDown}}
  \pgfpathlineto{\pgfpoint{1.25\ResRight}{2.5\ResDown}}
  \pgfpathlineto{\pgfpoint{1.25\ResLeft}{2.5\ResDown}}
  \pgfpathlineto{\pgfpoint{0.5\ResLeft}{\ResDown}}
  \pgfusepath{draw} %draw speaker
 }
\def\speakerpath#1{\TikzBipolePath{speaker}{#1}}
\tikzset{speaker/.style = {\circuitikzbasekey, /tikz/to path=\speakerpath, l=#1}}
```