

Typesetting Electrical Wiring Diagrams for *Relay* Control Systems with \LaTeX and TikZ

Luis Paulo Laus
e-mail: laus@utfpr.edu.br

Version: 1.1, Version date: 2018-02-19

1 Abstract

This package contains a collection of symbols for typesetting electrical wiring diagrams for relay control systems. The symbols are meant to be in agreement with international standard IEC 60 617 which is world-wide adopted with perhaps the exception of USA. It extends and modify, when needed, the TikZ-library `circuits.ee.IEC`. A few non-standard symbols are also included mainly to be used in presentations, particularly with `beamer` package.

2 Introduction

Relay control system has been gradually replaced by PLC (Programable Controllers) software. However, small automation systems are usually implemented using relay control systems and the descriptions of PLC connection to external components are done using the same symbols as in relay control systems. Relay control systems are traditionally introduced in the very begin of industrial automation disciplines. This package provides symbols to represent those systems accordingly to IEC 60 617.

Very unfortunately, the author has no access to the current version of that standard. The standards consulted during the elaboration of this package are:

- IEC 60 617-2/1996 Graphical symbols for diagrams – Part 2: Symbol elements, qualifying symbols and other symbols having general application;
- IEC 60 617-4/1996 Graphical symbols for diagrams – Part 4: Passive components;
- IEC 60 617-8/1996 Graphical symbols for diagrams – Part 8: Measuring instruments, lamps and signalling devices;
- IEC 60 617-7/1996 Graphical symbols for diagrams – Part 7: Switchgear, controlgear and protective devices.

All of the aforementioned standards have been withdrawn and correspond to edition 2 of IEC 60 617. The up-to-date edition, at the moment this document was written, is IEC 60 617:2012 DB. The author trust that the symbols in this package have not been modified between versions, but without access to the new edition it is impossible to say for sure. One indication that improves confidence in the up-to-dateness of this package is the fact that the current editions is based in editions 2 and 3 and the blunt fact that symbols, once adopted, tend to remain unchanged over time. There are also other libraries with free access¹ that show symbols with the same form they appear herein. Please, report any inaccuracy and out-to-date symbols.

You can also help if you have access to the current standard and can verify and report the accuracy of the information presented herein. Particularly regarding the dimensions employed in the graphs. Some of them were devised using soundings geometrical rules, but many (like circle diameters) were estimated using poor resolution graphics.

The international standard IEC 60 617 is worldwide adopted with perhaps the exception of USA which uses NEMA – National Electrical Manufacturers Association. If you are interested in NEMA standard you should refer to `tikz-ladder` package. This package is meant to be used for PLC programming, but can also be used for relay control systems.

¹See, for instance, <https://symbols.radिकासoftware.com/stencil/IEC.html> and <https://www.tracepartsonline.net/> and search for “CEI IEC”.

3 Relay Library

TikZ Library `circuits.ee.IEC.relay`

```
\usepgflibrary{circuits.ee.IEC.relay} %  $\TeX$  and plain  $\TeX$  and pure pgf
\usepgflibrary[circuits.ee.IEC.relay] % Con $\TeX$ t and pure pgf
\usetikzlibrary{circuits.ee.IEC.relay} %  $\TeX$  and plain  $\TeX$  when using TikZ
\usetikzlibrary[circuits.ee.IEC.relay] % Con $\TeX$ t when using TikZ
```

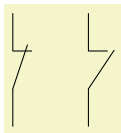
This library provides graphics for electrical wiring diagrams for relay control systems according to the international standard IEC 60 617. The library was written to extend the TikZlibrary `circuit.ee.IEC`. The reader is urged to read the Section “Circuit Libraries” of TikZ manual. This library defines the following key:

`/tikz/circuit ee IEC relay` (no value)

This key should be passed as an option to a picture or a scope that contains a diagram. It will do some internal setups.

`/tikz/activated` (no value)

Adding this key to a contact symbol will “activate the contact” so that a make contact will close, a break contact will open and a change-over contact will change. The “activated” symbols are not in accordance with IEC 60 617 because contact shall be drawn in the deactivated position, but sometimes we need to illustrate how a circuit works and this comes in handy. The main purpose is for BEAMER presentations, see Section 7.2 for details and further examples. Compare deactivated (normal) and activated break contact:



```
\tikz[circuit ee IEC relay]
\draw (0,0) to[break contact] (0,1.5)
(1,0) to[break contact,activated] (1,1.5);
```

`/tikz/term=[$\langle options \rangle$] $\langle angle \rangle$: $\langle text \rangle$` (no default)

This key is meant to place text information about the upper terminal of the component. It has nearly the same effect as the `label` key, only the placement position is change and following style is used additionally and automatically:

`/tikz/every term` (style, initially `font=\scriptsize`)

Set this style to configure the styling of term labels. Since this key is *not* used with normal labels, it provides an easy way of changing the way info labels look without changing other labels.

The $\langle options \rangle$ and $\langle angle \rangle$ are passed directly to the `label` command. For a detailed discussion of the `label` option refer to the TikZ manual.

The use of this key is exemplified in Section 6.

`/tikz/term'=[$\langle options \rangle$] $\langle angle \rangle$: $\langle text \rangle$` (no default)

This key is similar to `term`, but it is meant to place text information about the lower terminal of the component. It has nearly the same effect as the `label` key, only the placement position is change and following style is used additionally and automatically:

`/tikz/every term'` (style, initially `font=\scriptsize`)

Set this style to configure the styling of `term'` labels. Since this key is *not* used with normal labels, it provides an easy way of changing the way info labels look without changing other labels.

The $\langle options \rangle$ and $\langle angle \rangle$ are passed directly to the `label` command. For a detailed discussion of the `label` option refer to the TikZ manual.

The use of this key is exemplified in Section 6.

`/tikz/term''=[$\langle options \rangle$] $\langle angle \rangle$: $\langle text \rangle$` (no default)

This key is similar to `term`, but it is meant to place text information about the upper left terminal of the component mainly for `change over` contact. It has nearly the same effect as the `label` key, only the placement position is `change` and following style is used additionally and automatically:

`/tikz/every term''` (style, initially `font=\scriptsize`)

Set this style to configure the styling of `term''` labels. Since this key is *not* used with normal labels, it provides an easy way of changing the way info labels look without changing other labels.

The `<options>` and `<angle>` are passed directly to the `label` command. For a detailed discussion of the `label` option refer to the TikZ manual.













The use of this key is exemplified in Section 6.

4 Convention

In this library, the `mid` anchor is used to place annotations, which are secondary symbols placed on top of another symbol, on contacts and the `north` or `south` anchor on `relay coil`. The key `info`, as described in Section “Circuit Libraries” of TikZ manual, is used to place the component identification. Observe that the `info` has to be provide for the annotation, if any, or for the symbol itself if the annotation is placed at right side of the symbol. The keys `term`, `term'` and `term''` are used to place terminal numbers by the side of component terminals. See Section 6 for details and examples.

5 Symbols

The symbols in this library are:

<i>key</i>	<i>IEC 60 617 number</i>	<i>Appearance</i>
<code>/tikz/resistor</code>	04-01-01	
<code>/tikz/fuse</code>	07-21-01	
<code>/tikz/inductor IEC relay graphic</code>	04-A3-01	
<code>/tikz/bulb</code>	08-10-01	
<code>/tikz/magnetic sensor</code>	02-08-02	
<code>/tikz/thermic sensor</code>	02-08-01	
<code>/tikz/make contact</code>	07-02-01	
<code>/tikz/break contact</code>	07-02-03	
<code>/tikz/change over contact</code>	07-02-04	
<code>/tikz/relay coil</code>	07-15-01	
<code>/tikz/proximity sensor coil</code>	07-19-01	
<code>/tikz/mechanical interlock</code>	02-12-11	

Three symbols, resistor, inductor and bulb are already defined in `TikZ-library circuits.ee.IEC`. They are redefined in this package because their proportion was not in agreement with IEC 60 617. The symbols inductor is listed in Annex A of IEC 60 617 implying it is deleted and should appear in old diagrams or documentation explaining it is obsolete. To use this symbol set the option `set inductor graphic = inductor IEC relay graphic` either globally with `tikzset` or locally as option of a picture or a scope.

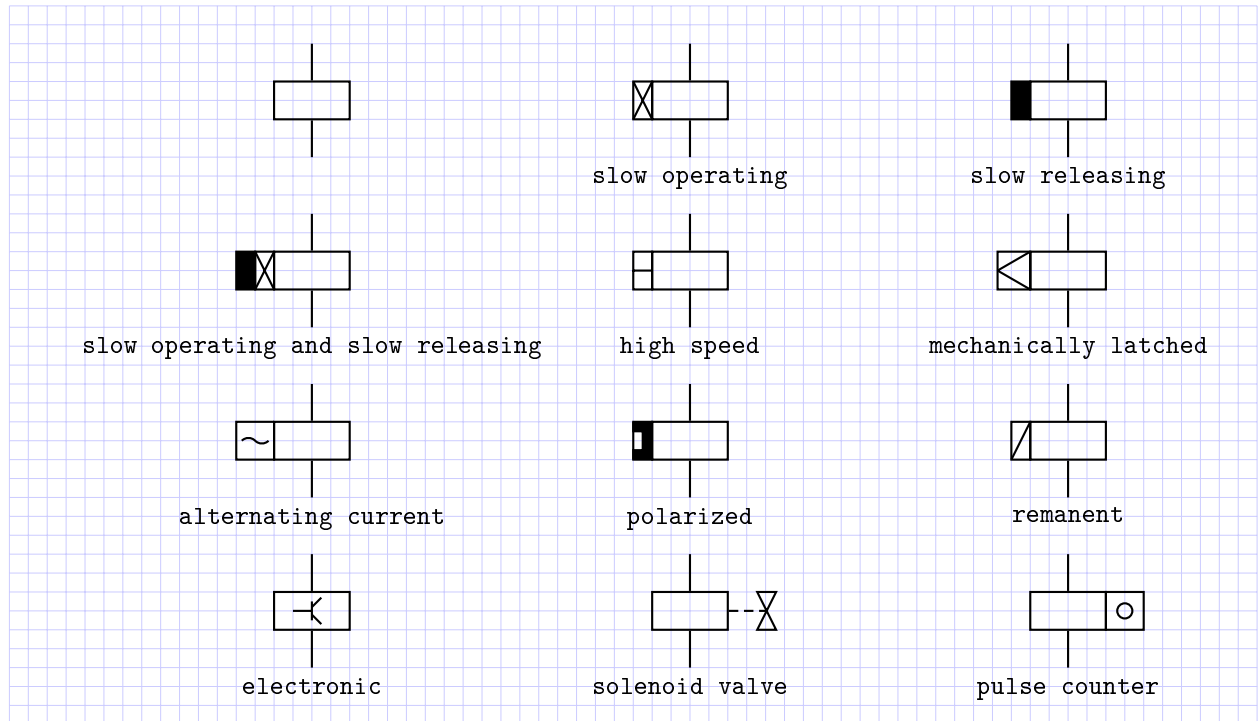
“Change over contact” is short for “change-over break before make contact”. The symbols `make contact`, `break contact` and `change over contact` can be activated (the contact changes the position) and this feature is needed for creating presentations. See Section 7.2 for details.

This list may look short, but several other symbols are possible combining these symbols with annotation.

In fact, it is annotation that empowers this library.

5.1 Annotation for relay coils

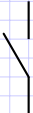
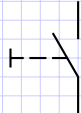
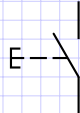
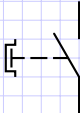
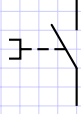
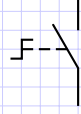
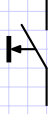
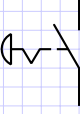


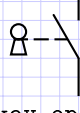
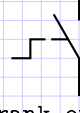
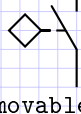


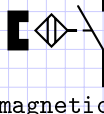

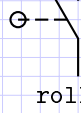
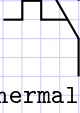
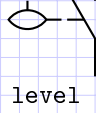
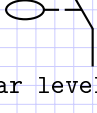
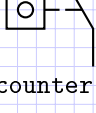
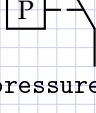
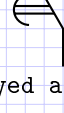
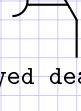
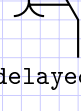
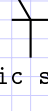

The relay coil can be used alone or combined with annotations to form:



Note that symbol relay coil is used to produce the graphic for other devices which in fact are not relays at all.

5.2 Annotation for contacts

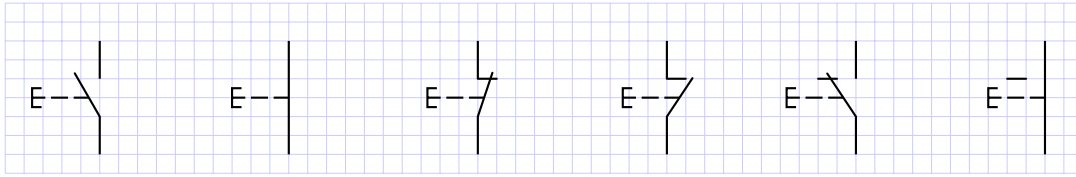
The contacts can be used alone or combined with annotations to form:

			
	manually operated	push button	protected manually operated
			
pull switch	turn switch	var push button	emergency switch
			
pedal operated	lever operated	key operated	crank operated
			
removable handle	proximity switch	touch switch	magnetic switch
			
position switch	roller	thermal switch	level switch
			
var level switch	counter switch	pressure switch	delayed activation
			
delayed deactivation	delayed both	static switch	circuit breaker

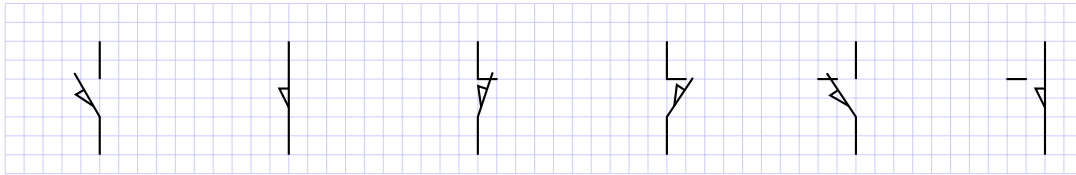
Temperature sensitive switch is indicated by either the Greek capital letter Θ or by the operating temperature conditions written at the right side of the contact and it can be achieved using the info' key. It is also usual to indicate pressure sensitive switch with roman letter P or by the operating pressure conditions in the same way.

5.3 Annotation self-placement

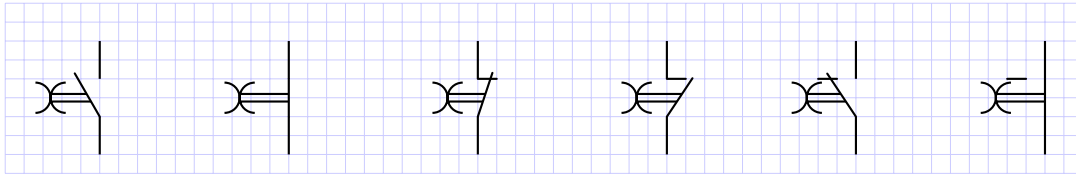
For contacts, the `mid` anchor is used to place the annotation at correct position. For instance, a push button applied to all kinds of contacts produces:



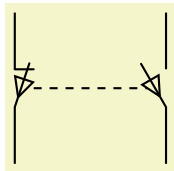
In the case of a position switch, delayed activation, delayed deactivation, delayed both and static switch, the orientation is also adjusted with the help of input anchor:



and



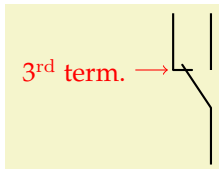
Annotations can be mirrored using the apostrophe ('), e.g., a position switch, mechanically operated in both directions with two separate circuits would be:



```
\begin{tikzpicture}[circuit ee IEC relay,thick]
\draw
(0,0) to[break contact={name=sw1,position switch,position switch'}] ++(0,2)
(2,0) to[make contact={name=sw2,position switch,position switch'}] ++(0,2);
\draw[dashed] (sw1.south) -- (sw2.north);
\end{tikzpicture}
```

5.4 Change over contact

Change over contact, or more exactly, “change-over break before make contact” is the only contact in this library that has three terminals. The bottom and top right terminal are connected in the usual way, namely, using the command `to`. To allow the user access to the top left terminal, an anchor named² `output 1` is automatically placed. To use this anchor, the contact has to be named. The reference is available in the same and subsequent paths. Therefore,



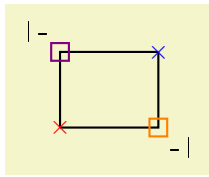
```
\begin{tikzpicture}[circuit ee IEC relay,thick]
\draw (0,0) to[change over contact={name=sw1}] ++(0,2) coordinate(n1)
(sw1.output 1) -- (sw1.output 1 |- n1);
\draw[<[sep=2pt]]-,red,thin] (sw1.output 1) -- ++(-0.5,0)
node[left]{3$~{\mathrm{rd}}$ term.};
\end{tikzpicture}
```

where `(n1)` is used to store the coordinates of the second point making it possible to draw a vertical line with the same vertical level³. The second part of the `\draw` command just draws a line from `(sw1.output 1)`

²The same name convention adopted in TikZ-library `circuits.ee.IEC` is used here: the symbol default orientations make the component appear as rotated 90° clockwise. The terminals on the left (bottom in the normal orientation) are named inputs and anchors with names starting with `input` are automatically placed. The terminals on the right (top in the normal orientation) are the outputs and anchors with names starting with `output` are automatically placed.

³In this case, the second point coordinates are known to be `(0,2)`, but in general it is a good idea to use the command `coordinate` to store the information for future use. This allows for the contact to be moved easily when needed.

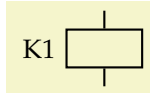
1) to (sw1.output 1 |- n1), a point with x coordinate equal to (sw1.output 1) _{x} and y coordinate equal to (n1) _{y} . Refer to Section “Coordinates at Intersections” of TikZ manual for details. This example also uses relative points; refer to Section “Special Syntax for Specifying Points” of TikZ manual for details. The use of relative coordinates and anchor allows the easy repositioning of the elements. Keep in mind that:



```
\begin{tikzpicture}[thick]
\coordinate (n1) at (0,0); \coordinate(n2) at (1.3,1);
\draw node[red]{\times} node[above left]{\verb!|-!} -- (n2)
(n1) -- (n1 |- n2) node[draw=violet]{} node[below right]{\verb!|-!} -- (n2)
node[blue]{\times};
\end{tikzpicture}
```

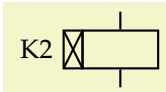
6 Labels

You can place information at the left side of a symbol using `info` and at the right side with `info'`. Note that, if the symbol has an annotation at its left side and you want to use `info`, you have to use `info` in the annotation. The same is valid for `info'` if the annotation is at the right side of the symbol. For instance, a relay coil named K1 is labelled as:



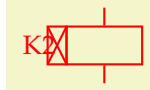
```
\tikz[circuit ee IEC relay,thick]
\draw (0,0) to[relay coil={info=K1}] (0,1);
```

but a slow operating relay coil has to be labelled in the `slow operating` key as:



```
\tikz[circuit ee IEC relay,thick]
\draw (0,0) to[relay coil={slow operating={info=K2}}] (0,1);
```

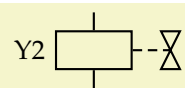
failing to do so produces incorrect labelling position as:



```
\tikz[circuit ee IEC relay,thick,red]
\draw (0,0) to[relay coil={slow operating,info=K2}] (0,1);
```

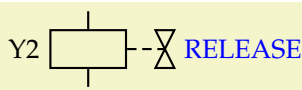
In the last case, the `info` is applied to `relay coil`, not to the annotation `slow operating`.

On the other hand, if the annotation is placed at right of the symbol, like a solenoid valve, the `info` is placed in the symbol itself (not in the annotation):



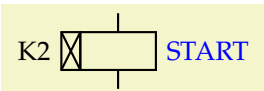
```
\tikz[circuit ee IEC relay,thick]
\draw (0,0) to[relay coil={solenoid valve,info=Y2}] (0,1);
```

The same reasoning applies to the right side using `info'` key: if there is an annotation at the right side, use the `info'` inside the annotation key:



```
\tikz[circuit ee IEC relay,thick]
\draw (0,0)
to[relay coil={solenoid valve={info'={blue}RELEASE}},info=Y2] (0,1);
```

but if there is no annotation or the annotation is on the left side, use the `info'` inside the symbol:



```
\tikz[circuit ee IEC relay,thick]
\draw (0,0)
to[relay coil={slow operating={info=K2},info'={blue}START}] (0,1);
```

It is also possible to label the terminals for assembling reference. There are three specially designed keys for it:

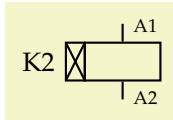
term places a label at the top right side of the symbol. It is meant to identify the top terminal. Because of TikZ convention, an anchor named `output` is placed on this terminal;

term' places a label at the bottom right side of the symbol. It is meant to identify the bottom terminal. Because of TikZ convention, an anchor named `input` is placed on this terminal;

term” places a label at the top left side of the symbol. It is meant to identify the top left terminal of a change over contact. Because of TikZ convention, an anchor named output 1 is placed on this terminal. Key **term**” places the label such that a line to output 1 can be drawn vertically or horizontally without crossing the label.

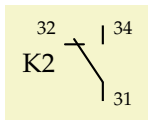
Each one of these three keys can be individually styled using the style, e.g., `every term'/.style`.

A slow operating relay coil:



```
\tikz[circuit ee IEC relay,thick]
\draw (0,0) to[relay coil={slow operating={info=K2},term=A1,term'=A2}] (0,1);
```

A change over contact of the same relay:

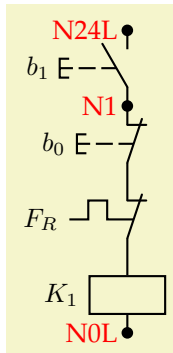


```
\tikz[circuit ee IEC relay,thick]
\draw (0,0) to[change over contact={info=K2,
term=34,term'=31,term''=32}] (0,1);
```

7 Design Guidance

This section brings some recommendations that reflect the way I produce diagrams for relay control systems. It may or may not work for you. Feel free to e-mail me if you have better ideas.

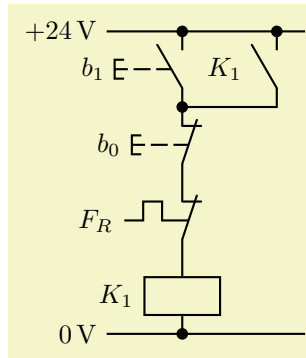
The first thing to consider is that, as the manual says, “TikZ ist kein Zeichenprogramm” which translates to “TikZ is not a drawing program”. You shall start with a draft of you diagram and then codify it using TikZ. Generally, a handmade pencil sketch will do. In this draft, you shall use node names for future reference. For instance, the contact named N1 below is used as a connection point for the latching contact of relay K_1 ; contacts N0L and N24L are used for power rails (left side).



```
\begin{tikzpicture}[circuit ee IEC relay,thick]
\draw (0,0)
node[contact,name=N0L]{}
to [relay coil={info=$K_1$}] ++(0,1)
to [break contact={thermal switch={info=$F_R$}}] ++(0,1)
to [break contact={push button={info=$b_0$}}] ++(0,1)
node[contact,name=N1]{}
to [make contact={push button={info=$b_1$}}] ++(0,1)
node[contact,name=N24L]{};
\node[left,red] at(N1) {N1};
\node[left,red] at(N0L) {N0L};
\node[left,red] at(N24L) {N24L};
\end{tikzpicture}
```

The second thing to consider is that tikz-relay uses the `\tikzcircuitssizeunit` to keep all figures proportional. Therefore, when you consider any dimension related to symbol size it is good idea to set that dimension in respect to `\tikzcircuitssizeunit`, i.e., using `\tikzcircuitssizeunit` as the unit of length. The default value of `\tikzcircuitssizeunit` is 7 pt or approximately 2.46 mm and it can be set by the circuit symbol unit key among several other keys. You can think of `\tikzcircuitssizeunit` as the module size M in IEC 60 617. Most symbols presented in the IEC 60 617 are draw using $M = 2.5$ mm. This can be achieved placing the command `\tikzset{circuit symbol unit=2.5mm}` in the document preamble which changes `\tikzcircuitssizeunit` to 7.11317pt document wide (not too different from the default value though). You can establish your own length unit and set x and y to that length unit. In this way you will be working on a grid; if it is too big or too small you will have to change a single declaration.

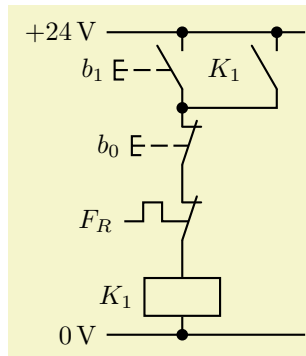
Finishing up the previous example, the x grid is set to five `\tikzcircuitssizeunit` to allow horizontal space for labels and y grid is set to four `\tikzcircuitssizeunit` because the symbols used in control circuit schematics are two `\tikzcircuitssizeunit` tall, so there will be one `\tikzcircuitssizeunit` above and below the symbol. Starting at N1, the latching contact of K_1 is placed one horizontal length unit aside b_1 .



```
\begin{tikzpicture}[circuit ee IEC relay,thick,
x=5\tikzcircuitssizeunit,y=4\tikzcircuitssizeunit]
\draw (0,0)
node[contact,name=NOL]{}
to [relay coil={info=$K_1$}] ++(0,1)
to [break contact={thermal switch={info=$F_R$}}] ++(0,1)
to [break contact={push button={info=$b_0$}}] ++(0,1)
node[contact,name=N1]{}
to [make contact={push button={info=$b_1$}}] ++(0,1)
node[contact,name=N24L]{};
\draw (N1) -- ++(1,0)
to [make contact={info=$K_1$}] ++(0,1)
node[contact,name=N24R]{};
% power rails
\draw (N24L) ++(-0.8,0) node[left]{$\unit{+24}{V}$} -- (N24R) -- ++(0.3,0)
(NOL) ++(-0.8,0) node[left]{$\unit{0}{V}$} -- (NOL -| N24R) -- ++(0.3,0);
\end{tikzpicture}
```

Power rails are placed using named nodes. This is not really necessary since the beginning and end of the power rails lines can be determined easily because we are working on a grid.

Because of the direction chosen when the symbols were first defined in TikZ, we start from the bottom moving to the top of the diagram. This is not mandatory, although changing the direction requires that all symbols and annotations are negatively scale, for instance, the last example would be:



```
\begin{tikzpicture}[circuit ee IEC relay,thick,
x=5\tikzcircuitssizeunit,y=4\tikzcircuitssizeunit,
every circuit symbol/.style={scale=-1},
every circuit annotation/.style={scale=-1}]
\draw (0,0)
node[contact,name=N24L]{}
to [make contact={push button={info=$b_1$}}] ++(0,-1)
node[contact,name=N1]{}
to [break contact={push button={info=$b_0$}}] ++(0,-1)
to [break contact={thermal switch={info=$F_R$}}] ++(0,-1)
to [relay coil={info=$K_1$}] ++(0,-1)
node[contact,name=NOL]{};
\draw (1,0)
node[contact,name=N24R]{}
to [make contact={info=$K_1$}] ++(0,-1) -- (N1);
% power rails
\draw (N24L) ++(-0.8,0) node[left]{$\unit{+24}{V}$} -- (N24R) -- ++(0.3,0)
(NOL) ++(-0.8,0) node[left]{$\unit{0}{V}$} -- (NOL -| N24R) -- ++(0.3,0);
\end{tikzpicture}
```

where `every circuit symbol/.style={scale=-1}` and `every circuit annotation/.style={scale=-1}` were used to scale the symbols and annotations, respectively. This example produces the very same diagram, but we start at the top and work down to the bottom.

7.1 Some tricks

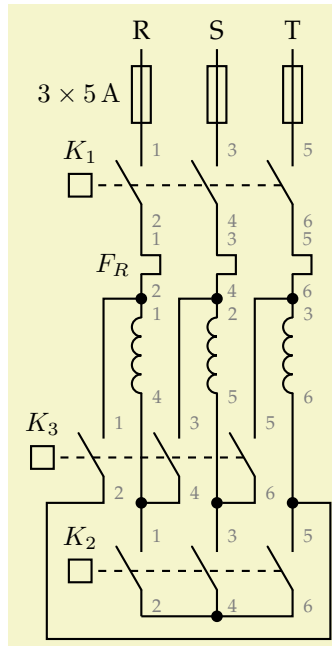
Power circuits schematics use a more diverse variety of symbol height than control circuit schematics. Thus, it is better to set the grid to a smaller value. This allows for a higher control over the symbols distance. For each symbol, consider its height and allow some extra space between symbols.

Some symbols like relay coils for power circuits schematics are not available, thus we can draw a square by the side of the relay contacts connecting it to the right most contact using a dashed line. The mid anchor is perfect for that. Actually, we shall start at the mid anchor of the right most contact and draw a dashed line to the left using relative coordinates, at the end of the dashed line we place a square node with no text inside, but with a label above (or left) identifying the relay coil.

The inductors look a bit odd: the semicircles end in very sharp corners. It is better to change the line joint to bevel.

You can use different styles for `info` and `info'`, and also for `term`, `term'` and `term''`, but it is usually a bad idea. It is normally best to set a style for one of these keys and copy it to the others.

For example, a star-delta motor starter has power circuit schematic like:



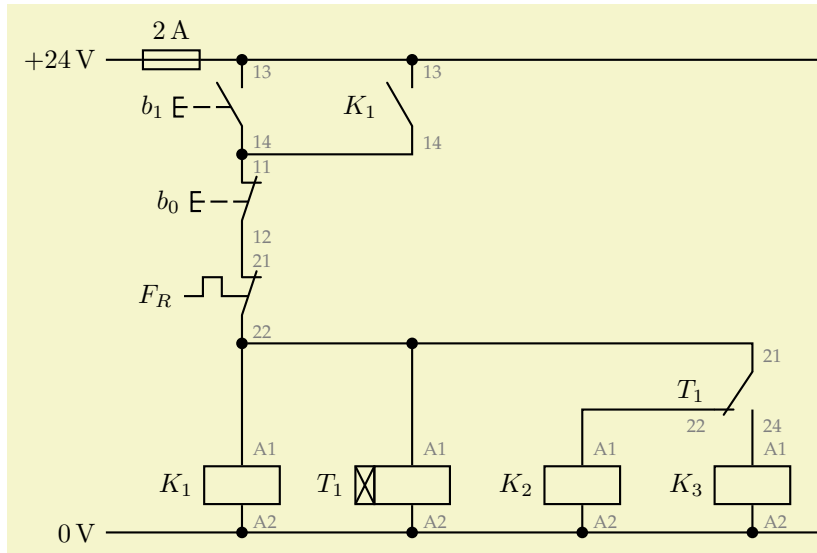
```
\begin{tikzpicture}[circuit ee IEC relay,thick,
x=|tikzcircuitssizeunit,
y=1.2|tikzcircuitssizeunit,
every inductor/.style={line join=bevel},
every term/.style={gray,font=\scriptsize},
every term'/.style=every term,
every term''/.style=every term]
\draw (0,0)
to[make contact={term=1,term'=2}] ++(0,4)
-- ++(0,1)
node[contact,name=N1R]{}
-- ++(0,4)
to[inductor={name=LR,term=1,term'=4}] ++(0,5)
node[contact,name=N2R]{}
to[thermic sensor={info=$F_R$,term=1,term'=2}] ++(0,3)
to[make contact={term=1,term'=2}] ++(0,4)
to[fuse={info={$\unit[3]{times5}{A}$}}] ++(0,4)
node[above]{R};
\draw (0,0) -- ++(4,0) node[contact]{}
to[make contact={term=3,term'=4}] ++(0,4)
-- ++(0,1)
node[contact,name=N1S]{}
-- ++(0,4)
to[inductor={name=LS,term=2,term'=5}] ++(0,5)
node[contact,name=N2S]{}
to[thermic sensor={term=3,term'=4}] ++(0,3)
to[make contact={term=3,term'=4}] ++(0,4)
to[fuse] ++(0,4)
node[above]{S};
\draw (4,0) -- ++(4,0)
to[make contact={name=K2T,term=5,term'=6}] ++(0,4)
-- ++(0,1)
node[contact,name=N1T]{}
-- ++(0,4)
to[inductor={name=LT,term=3,term'=6}] ++(0,5)
node[contact,name=N2T]{}
to[thermic sensor={term=5,term'=6}] ++(0,3)
to[make contact={name=K1T,term=5,term'=6}] ++(0,4)
to[fuse] ++(0,4)
node[above]{T};
\draw
(N1R) -- ++(2,0)
to[make contact={term=3,term'=4}] ++(0,4) |- (N2S)
(N1S) -- ++(2,0)
to[make contact={name=K3T,term=5,term'=6}] ++(0,4) |- (N2T)
(N1T) -| ++(2,-6) -| ++(-15,6) -- ++(3,0)
to[make contact={term=1,term'=2}] ++(0,4) |- (N2R);
\draw[dashed](K1T.mid) --++(-10,0) % relay coils
node[left,draw,solid,minimum size=3mm,label={[above]:$K_1$}]{};
\draw[dashed](K2T.mid) --++(-10,0)
node[left,draw,solid,minimum size=3mm,label={[above]:$K_2$}]{};
\draw[dashed](K3T.mid) --++(-10,0)
node[left,draw,solid,minimum size=3mm,label={[above]:$K_3$}]{};
\end{tikzpicture}
```

Control circuit schematics can use a much bigger grid space because the symbol heights are the same, i.e. always two units of length. When setting the x unit of length, you need to consider the space occupied by labels and annotations.

Sometimes we need to place a symbol upside-down. This is the case of the star-delta motor starter control circuit schematic in which a change-over contact is used to change from K_2 to K_3 . To achieved the correct orientation, a negative y scaling is set by `yscale=-1`. This contact is named `t1` for future reference, in this case, to access the output `1` and connect it to K_2 . If you think it is against IEC 60 617 rules to drawn the change-over contact upset-down you will end up with a terrible layout because of the position of the common terminal.

The relay coil K_3 is also placed using top-down direction and, consequently, it has to be negatively scaled (mirrored). In this case, we need to mirror the symbol in both x and y direction so the terminal and symbol identification will be correctly located. So, `scale=-1` is used.

The power rails are drawn using grid coordinates. A fuse is place in the top power rail.



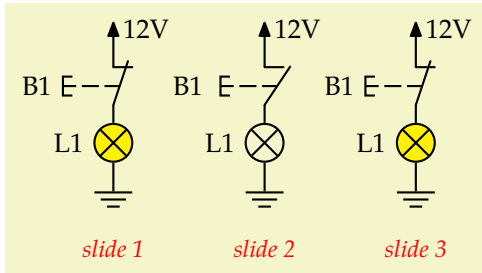
```
\begin{tikzpicture}[circuit ee IEC relay,thick,
  x=9|tikzcircuitssizeunit,
  y=5|tikzcircuitssizeunit,
  every term/.style={gray,font=\scriptsize},
  every term'/.style=every term,
  every term''/.style=every term]
\draw (0,0)
  node[contact]{}
  to [relay coil={info=$K_1$,term=A1,term'=A2}] ++(0,1) -- ++(0,1)
  node[contact,name=N1]{}
  to [break contact={thermal switch={info=$F_R$},term=21,term'=22}] ++(0,1)
  to [break contact={push button={info=$b_0$},term=11,term'=12}] ++(0,1)
  node[contact,name=N2]{}
  to [make contact={push button={info=$b_1$},term=13,term'=14}] ++(0,1)
  node[contact,name=N3]{};
\draw (N2) -- ++(1,0)
  to [make contact={info=$K_1$,term=13,term'=14}] ++(0,1)
  node[contact]{};
\draw (1,0)
  node[contact]{}
  to [relay coil={slow operating={info=$T_1$,term=A1,term'=A2}] ++(0,1) -- ++(0,1)
  node[contact,name=N4]{} --(N1);
\draw (N4) -- ++(2,0)
  to [change over contact={yscale=-1,info=$T_1$,name=t1,term=24,term'=21,term''=22}] ++(0,-1)
  to [relay coil={scale=-1,info=$K_3$,term=A1,term'=A2}] ++(0,-1)
  node[contact]{};
\draw (2,0)
  node[contact]{}
  to [relay coil={info=$K_2$,term=A1,term'=A2}] ++(0,1) |- (t1.output 1);
\draw(-0.8,0) node[left]{\unit[0]{V}} -- +(4.2,0) % power rails
  (-0.8,5) node[left]{\unit[+24]{V}} to [fuse={info={\unit[2]{A}}}] (N3) -- +(3.4,0);
\end{tikzpicture}
```

7.2 Beamer presentation with overlay

Perhaps the most important feature of BEAMER is its capacity of creating a series of slides based in one slide for, e.g., showing one concept step by step. It is called *overlay*. Unfortunately, BEAMER overlay and TikZ present some compatibility issues when overlay macros are typed in the options list of a TikZ command. This can be solved setting two TikZ keys that takes advantage of the fact that `\pgfkeysalso` doesn't change the path.

```
\tikzset{ % alt and visible (overlay)
  alt/.code args={<#1>#2#3}{%
    |alt<#1>{\pgfkeysalso{#2}}{\pgfkeysalso{#3}}
  },
  visible/.code args={<#1>#2}{%
    |alt<#1>{\pgfkeysalso{#2}}{}
  }
}
```

Once these keys are in place, they can be used instead of `\visible` and `\alt` macros with a syntax slightly different. The following example creates three slides to illustrate how a push button works with a break contact and, when pushed, break contact is activated. Below, the result is shown side by side:



```
\begin{tikzpicture}[circuit ee IEC relay,thick,
y=3|tikzcircuitssizeunit]
\node[ground={point down,name=gnd}] at(0,0) {};
\draw(gnd)
to ++ (0,0.5)
to[bulb={info=L1,visible={<1,3>{fill=yellow}}}] ++ (0,1)
to[break contact={push button={info=B1},
alt={<2>{activated}}}] ++ (0,1)
-- ++ (0,0.5) node[current direction={point up}]{}
node[right]{12V};
\end{tikzpicture}
```

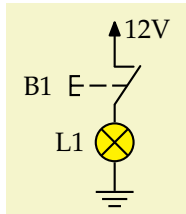
During the presentation you can say that “pressing B1 the lamp L1 turns off, releasing it turns L1 on again”. One cool trick is to move forward and backward between slides 2 and 3.

To draw this example, we start with the schematics and once it is ready the animation part is done by placing keys `visible` and `alt` in suitable locations. The filling of L2 is visible only in slide 1 and 3. B1 appears differently depending on the slide. For slides 1 and 3, B1 is a normal break contact (closed), but in slide 2 it is an “activated” break contact which is open. Note that key `bulb` receives or not the key `fill`. Similarly, the key `break contact` receives or not the key `activated` (the key `visible` could be used instead of `alt`).

When developing “animations”, you may need to know the overlay number. It can be printed, say in the frame title, using macro `\overlaynumber`; type this in the document preamble:

```
\makeatletter
\newcommand*{\overlaynumber}{\number\beamer@slideinframe}
\makeatother
```

If you need to generate a handout, the `visible` argument will be drawn as well as the *first* argument of `alt` no matter what is the overlay specification (the thing between `<` and `>` signs). Therefore, the above example would appear like:



```
\begin{tikzpicture}[circuit ee IEC relay,thick,
y=3|tikzcircuitssizeunit]
\node[ground={point down,name=gnd}] at(0,0) {};
\draw(gnd)
to ++ (0,0.5)
to[bulb={info=L1,fill=yellow}] ++ (0,1)
to[break contact={push button={info=B1,activated}}] ++ (0,1)
-- ++ (0,0.5) node[current direction={point up}]{}
node[right]{12V};
\end{tikzpicture}
```

and it does not match any of the previous slides⁴. Keep in mind that you shall finish the schematic in the regular form (without any activated contact) and then activated can be inserted always as the second argument of `alt`; the first argument shall be left empty, e.g., `alt{<1,3>{}{activated}}`. Alternatively, you can use, e.g., `alt{<1,3>{activated=false}{activated=true}}`. The slide numbers enlisted in the overlay specification shall correspond to slides in which the contact is *not* activated.

The “correct” diagram that works both in beamer and handout modes is:

```
\begin{tikzpicture}[circuit ee IEC relay,thick,
y=3|tikzcircuitssizeunit]
\node[ground={point down,name=gnd}] at(0,0) {};
\draw(gnd)
to ++ (0,0.5)
to[bulb={info=L1,visible={<1,3>{fill=yellow}}}] ++ (0,1)
to[break contact={push button={info=B1},
alt={<1,3>{}{activated}}}] ++ (0,1)
-- ++ (0,0.5) node[current direction={point up}]{}
node[right]{12V};
\end{tikzpicture}
```

The handout will show a lighted lamp in series with a disactivated break contact. If you do not want that the lamp appears lighted, use `alt={<2>{}{fill=yellow}}` instead of `visible`. So, the lamp will not be lighted in slide 2 (implying it will in slides 1 and 3) and it will not be lighted in the handout as well.

⁴It is just wrong.

A complete, though minimalistic, example is provided in a separated file⁵: “BeamerAnimation.tex”, which is listed in Appendix A for your convenience. In this example, a four-step sequencer is animated. A total of twenty slides are generated based in a single schematic. The frame title includes the macro `\overlaynumber` which generates the slide number and shall be removed once the presentation is ready. So, all slides would look like the same slide, but with animation on it.

7.3 Adjusting the diagram size

The diagram size can be adjusted setting `x` and `y` units. Mainly, you change those to control the space between symbols, not the diagram size in total⁶. If your diagram is too big, consider to use a landscape page. For presentations, adjustments can be done using `resizebox`.

It is not a good idea to change the diagram size setting `\tikzcircuitssizeunit` using `circuit` symbol unit. `\tikzcircuitssizeunit` shall be set document wise in the document preamble and kept unchanged.

8 Shapes

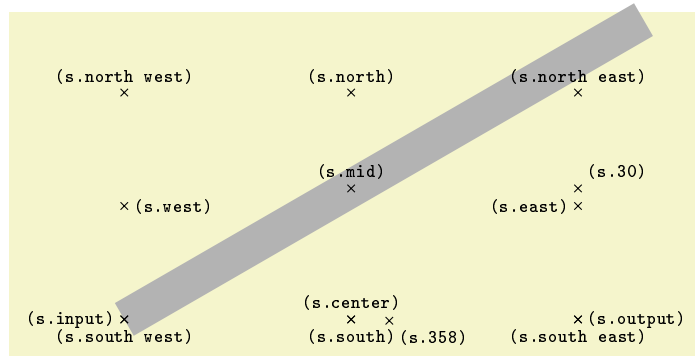
This is rather technical material, but it is here anyway and you may use it to modify or extend the library. It that case, if you need help, please do not hesitate in contacting the author.

Several anchors are defined for each symbol (or, more exactly, shapes). The mechanism TikZ uses to place anchors and define borders is sophisticated. It takes into account the line width (or an optional additional space defined by `outer sep`⁷) to place the anchors just on the line border of the symbol. Something called “anchor border” is also defined in the same manner. Anchor border is use to place labels around the symbol and interrupt the line directed to the symbol. Remember that a symbol is usually placed by command to which interrupts the path line the symbol is placed on. It makes perfect sense for symbols that encloses some area, like `proximity sensor coil` and `relay coil` and even for symbols that, although they do not enclose an area, defines some intuitive border, like `magnetic sensor` and `thermic sensor`. But it makes little sense for contacts.

Contacts can be considered an especial case of symbols. They have no defined border, no intuitive notion of inside and outside. The TikZ-library `circuits` places them in a category called “wires”. The normal anchors (`north`, `south`, `south west`, etc.) have no practical use. Even so, they need a `center` anchor, used to place the symbol, and a `mid` anchor, used to place annotations. They also need an `anchor border` in order to interrupt the line during placement (and also for label positioning). The `anchor border` controls the gap that appears on the line. Thus, it is defined as the symbol size. In other words, the symbol size is an invisible border a bit smaller than the actual symbol because the inclined line which represents the movable contact have an addendum that protrudes beyond the symbol border.

Outer separation is also undesired when placing contact. If the normal treatment were dispensed for contacts, setting the outer separation to any positive number would disrupt the line connected to the symbol. Instead of setting the outer separation to zero, the shape code that produces contacts simply disregards the outer separation when drawing the component.

All symbols below look rotated 90° because TikZ uses this orientation to place the symbol in a line.



⁵This file should be in the same folder you have found this document, if not try <http://repositorios.cpai.unb.br/ctan/graphics/pgf/contrib/tikz-relay/doc/BeamerAnimation.tex>.

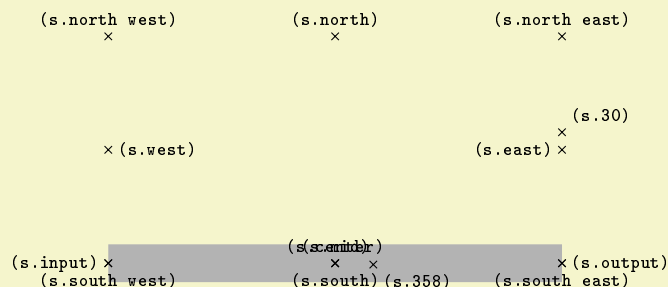
⁶The total size will change, but it is more like a collateral damage than intentional deed.

⁷See Section “Common Options: Separations, Margins, Padding and Border Rotation” of TikZ manual.

```

\begin{tikzpicture}[circuit ee IEC relay]
  \node[name=s, shape=make contact IEC relay, shape example, inner xsep=1cm, inner ysep=1cm, minimum
width=6cm, minimum height=3cm]{};
  \foreach \anchor/\placement in {center/above, 30/above right, 358/below right,
north/above, north east/above, north west/above,
south/below, south east/below, south west/below,
east/left, west/right, mid/above, input/left,output/right}
  \draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)} node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}

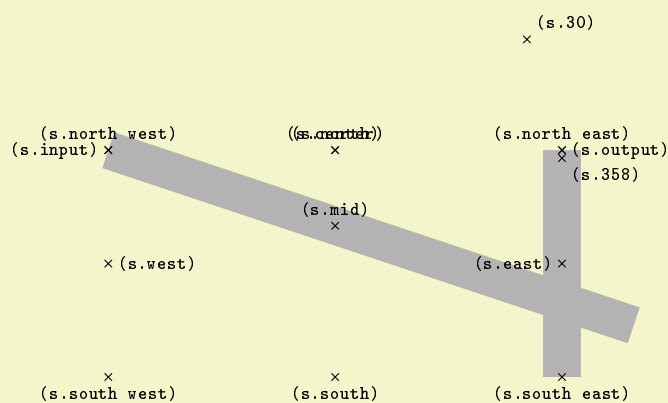
```



```

\begin{tikzpicture}[circuit ee IEC relay]
  \node[name=s, activated, shape=make contact IEC relay, shape example, inner xsep=1cm, inner
ysep=1cm, minimum width=6cm, minimum height=3cm]{};
  \foreach \anchor/\placement in {center/above, 30/above right, 358/below right,
north/above, north east/above, north west/above,
south/below, south east/below, south west/below,
east/left, west/right, mid/above, input/left,output/right}
  \draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)} node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}

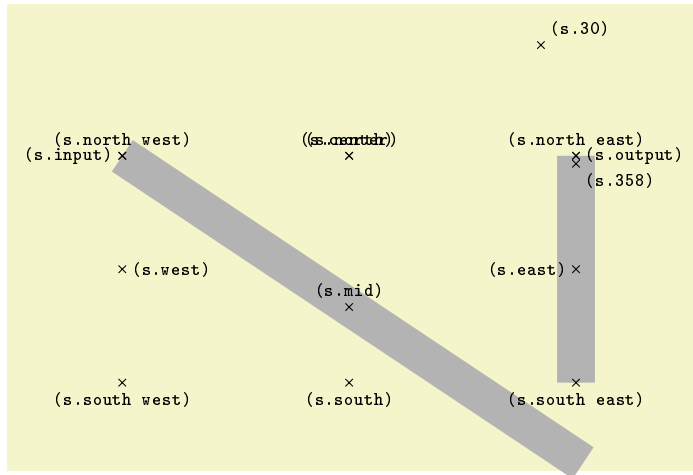
```



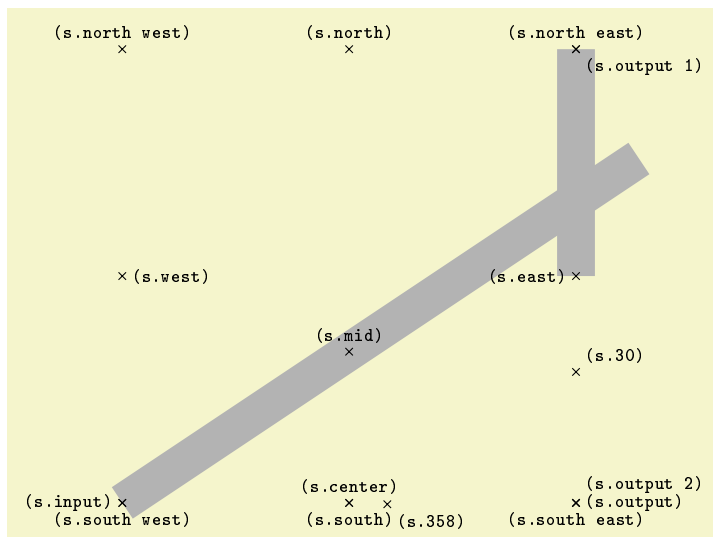
```

\begin{tikzpicture}[circuit ee IEC relay]
  \node[name=s, shape=break contact IEC relay, shape example, inner xsep=1cm, inner ysep=1cm, minimum
width=6cm, minimum height=3cm]{};
  \foreach \anchor/\placement in {center/above, 30/above right, 358/below right,
north/above, north east/above, north west/above,
south/below, south east/below, south west/below,
east/left, west/right, mid/above, input/left,output/right}
  \draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)} node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}

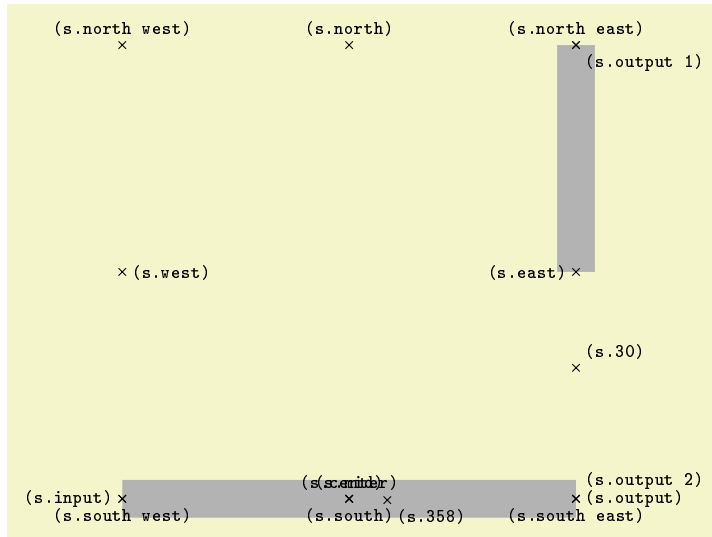
```



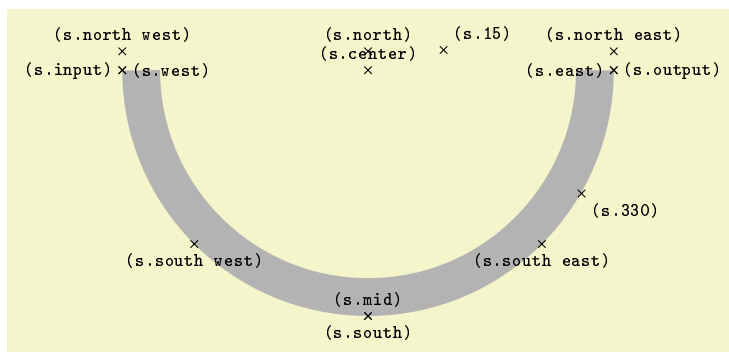
```
\begin{tikzpicture}[circuit ee IEC relay]
  \node[name=s, activated, shape=break contact IEC relay, shape example, inner xsep=1cm, inner
ysep=1cm, minimum width=6cm, minimum height=3cm]{};
  \foreach \anchor/\placement in {center/above, 30/above right, 358/below right,
north/above, north east/above, north west/above,
south/below, south east/below, south west/below,
east/left, west/right, mid/above, input/left,output/right}
  \draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)} node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```



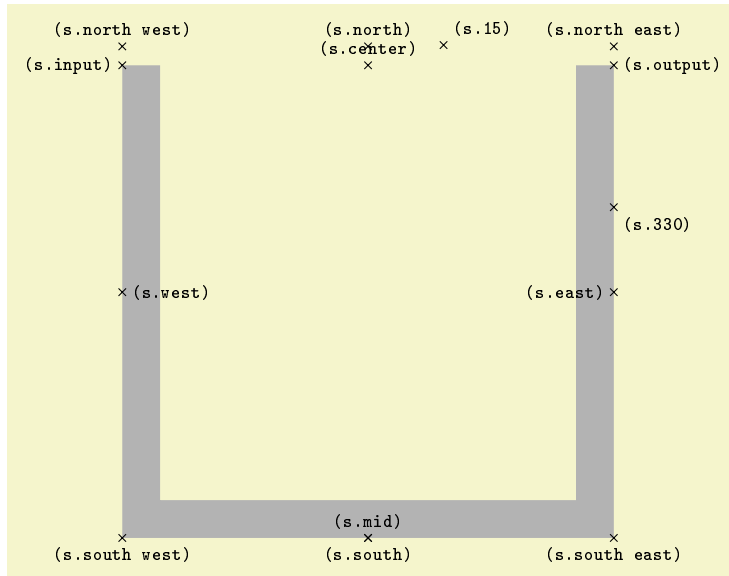
```
\begin{tikzpicture}[circuit ee IEC relay]
  \node[name=s, shape=change over contact IEC relay, shape example, inner xsep=1cm, inner ysep=1cm, minimum
width=6cm, minimum height=6cm]{};
  \foreach \anchor/\placement in {center/above, 30/above right, 358/below right,
north/above, north east/above, north west/above,
south/below, south east/below, south west/below,
east/left, west/right, mid/above, input/left,output/right, output 1/below right, output 2/above right}
  \draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)} node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

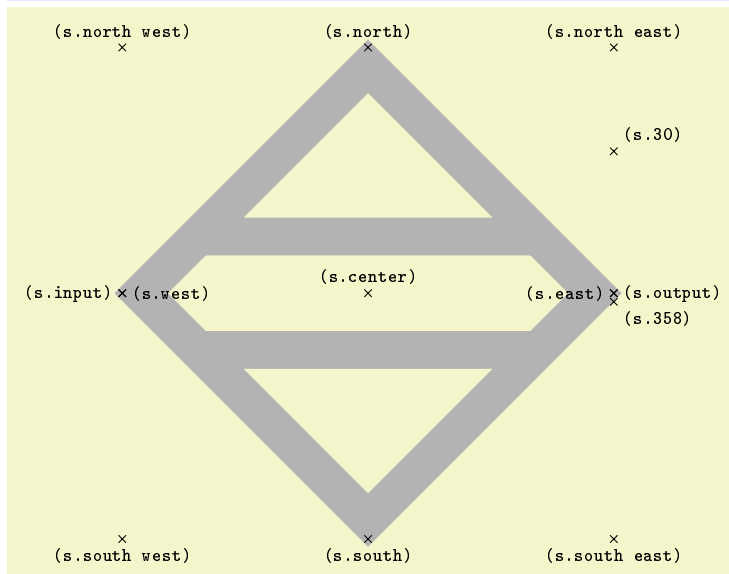
```
\begin{tikzpicture}[circuit ee IEC relay]
  \node[name=s, activated, shape=change over contact IEC relay, shape example, inner xsep=1cm, inner
ysep=1cm, minimum width=6cm, minimum height=6cm]{};
  \foreach \anchor/\placement in {center/above, 30/above right, 358/below right,
    north/above, north east/above, north west/above,
    south/below, south east/below, south west/below,
    east/left, west/right, mid/above, input/left,output/right, output 1/below right, output 2/above right}
  \draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)} node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```



```
\begin{tikzpicture}[circuit ee IEC relay]
  \node[name=s, shape=magnetic sensor IEC relay, shape example, inner xsep=1cm, inner ysep=1cm, minimum
width=6cm, minimum height=6cm]{};
  \foreach \anchor/\placement in {center/above, 15/above right, 330/below right,
    north/above, north east/above, north west/above,
    south/below, south east/below, south west/below,
    east/left, west/right, mid/above, input/left,output/right}
  \draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)} node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```



```
\begin{tikzpicture}[circuit ee IEC relay]
  \node[name=s, shape=thermic sensor IEC relay, shape example, inner xsep=1cm, inner ysep=1cm, minimum
width=6cm, minimum height=6cm]{};
  \foreach \anchor/\placement in {center/above, 15/above right, 330/below right,
north/above, north east/above, north west/above,
south/below, south east/below, south west/below,
east/left, west/right, mid/above, input/left, output/right}
  \draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)} node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```



```
\begin{tikzpicture}[circuit ee IEC relay]
  \node[name=s, shape=proximity sensor coil IEC relay, shape example, inner xsep=1cm, inner ysep=1cm, minimum
width=6cm, minimum height=6cm]{};
  \foreach \anchor/\placement in {center/above, 30/above right, 358/below right,
north/above, north east/above, north west/above,
south/below, south east/below, south west/below,
east/left, west/right, input/left, output/right}
  \draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)} node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

9 Known Issues

This library is base in withdrawn standards and the current standardized symbols might be different. It is also rather incomplete; so many symbols are missing, though in my best knowledge the missing symbols are rarely used. Let me know if you need a symbol, I will do my best to implement it.

If you want to report a bug or have any suggestion, please feel free to send me an e-mail. Contact details are in the first page. Every feedback is important.

10 Final Remarks

This package has been tested and used for more than three years, so I do believe it is mature by now and I decided to share it. On the other hand, I was the only person who used it, therefore idiosyncrasies were not detected.

Any comments, suggestions, request for missing symbols and feedback are welcomed. I will do my best to answer as soon as possible. My contact e-mail is in the first page.

It should be great if someone with experience in writing TikZ libraries could have a look in the code and point out error or improvements to be made.

Appendix A – Beamer Example of a Four Step Sequencer

In this example, the animations is done in two layers: the schematics is drawn on foreground with contact and filling animation; on the background, circles (actually, ellipses since x and y scales are different) are drawn in light blue to highlight with contact or coil that has been activated because several of them changes from one slide to the next. In order to place the ellipses in the precise position, every contact and coil is named.

At first, only one coil and the contacts respective to one step are drawn, then they are copy three times and the information is changed as needed. Finally, the b_s is added. When the schematics is done, the animation part is made adding the `alt` key in the foreground and the `\visible` macro in the background.

After the \LaTeX code, the slides were included for reference. They shall be found as a separated PDF file in the same folder you have found this document.

```
%% This is file 'BeamerAnimation.tex'
%% Version: 1.0.1
%% Version date: 2018-02-19
%%
%% Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br
%%
%% This package can be redistributed and/or modified under the terms
%% of the LaTeX Project Public License distributed from CTAN
%% archives in directory macros/latex/base/lppl.txt; either
%% version 1 of the License, or (at your option) any later version,
%% with 'The Package' referring to the software 'tikzlibrarysfc.code.tex' and its
%% accompanying documentation and 'The Copyright Holder' referring to the
%% person Luis Paulo Laus.
%%
%%
%% IMPORTANT NOTICE:
%%
%% For error reports, comments or suggestions in case of UNCHANGED
%% versions send mail to:
%% laus@utfpr.edu.br
%%
%%
\documentclass{beamer}
\usepackage{tikz,units}
\usetikzlibrary{backgrounds, circuits.ee.IEC.relay}

\makeatletter
\newcommand*{\overlaynumber}{\number\beamer@slideinframe}
\makeatother

\tikzset{ % alt and visible (overlay)
  alt/.code args={<#1>#2#3}{%
    \alt<#1>\pgfkeysalso{#2}}{\pgfkeysalso{#3}}
  },
  visible/.code args={<#1>#2}{%
    \alt<#1>\pgfkeysalso{#2}}{}
  }
}

\colorlet{LBlue}{blue!20}
\colorlet{LRed}{red!20}

\begin{document}

\begin{frame}{Four-Step Sequencer \overlaynumber{}}
\noindent \begin{center}
\hspace*{-0.05\textwidth}\resizebox{1.1\textwidth}{!}{
\begin{tikzpicture}[circuit ee IEC relay,thick,x=6\tikzcircuitssizeunit,y=5\tikzcircuitssizeunit]

\draw(-1.5,0) node[left]{$\unit[0]{V}$} --+(9,0)
(-1.5,4) node[left]{$\unit[+24]{V}$} --+(9,0);

\draw (0,0)
node[contact]{}
to [relay coil={info=$K_1$,name=k11,alt={<1,2,7-17>{}{fill=LRed}}}] ++(0,1)
```

```

to [break contact={info=$K_2$,name=k24,alt={<1-5,11-20>{{activated}}}] ++(0,1)
node[contact,name=N1]{}
to [make contact={info=$K_4$,name=k43,alt={<1-13,19-20>{{activated}}}] ++(0,1)
to [make contact={push button={info=$b_1$,name=b1,alt={<1-16,20>{{activated}}}] ++(0,1)
node[contact]{};
\draw (N1) — ++(1,0)
to [make contact={info=$K_1$,name=k12,alt={<1,2,7-17>{{activated}}}] ++(0,1) — ++(0,1)
node[contact]{};
\draw (N1) — ++(-1,0)
to [make contact={push button={info=$b_S$,name=bs,alt={<1,4-20>{{activated}}}] ++(0,2)
node[contact]{};
\draw (2,0)
node[contact]{}
to [relay coil={info=$K_2$,name=k21,alt={<1-5,11-20>{{fill=LRed}}}] ++(0,1)
to [break contact={info=$K_3$,name=k34,alt={<1-9,15-20>{{activated}}}] ++(0,1)
node[contact,name=N1]{}
to [make contact={info=$K_1$,name=k13,alt={<1,2,7-17>{{activated}}}] ++(0,1)
to [make contact={push button={info=$b_2$,name=b2,alt={<1-4,8-20>{{activated}}}] ++(0,1)
node[contact]{};
\draw (N1) — ++(1,0)
to [make contact={info=$K_2$,name=k22,alt={<1-5,11-20>{{activated}}}] ++(0,1) — ++(0,1)
node[contact]{};
\draw (4,0)
node[contact]{}
to [relay coil={info=$K_3$,name=k31,alt={<1-9,15-20>{{fill=LRed}}}] ++(0,1)
to [break contact={info=$K_4$,name=k44,alt={<1-13,19-20>{{activated}}}] ++(0,1)
node[contact,name=N1]{}
to [make contact={info=$K_2$,name=k23,alt={<1-5,11-20>{{activated}}}] ++(0,1)
to [make contact={push button={info=$b_3$,name=b3,alt={<1-8,12-20>{{activated}}}] ++(0,1)
node[contact]{};
\draw (N1) — ++(1,0)
to [make contact={info=$K_3$,name=k32,alt={<1-9,15-20>{{activated}}}] ++(0,1) — ++(0,1)
node[contact]{};
\draw (6,0) node[contact]{}
to [relay coil={info=$K_4$,name=k41,alt={<1-13,19-20>{{fill=LRed}}}] ++(0,1)
to [break contact={info=$K_1$,name=k14,alt={<1,2,7-17>{{activated}}}] ++(0,1)
node[contact,name=N1]{}
to [make contact={info=$K_3$,name=k33,alt={<1-9,15-20>{{activated}}}] ++(0,1)
to [make contact={push button={info=$b_4$,name=b4,alt={<1-12,16-20>{{activated}}}] ++(0,1)
node[contact]{};
\draw (N1) — ++(1,0)
to [make contact={info=$K_4$,name=k42,alt={<1-13,19-20>{{activated}}}] ++(0,1) — ++(0,1)
node[contact]{};
\begin{pgfonlayer}{background}
\visible<2-3>{
\draw[fill=LBlue,LBlue](bs) circle (0.4);
}
\visible<3-6,18-20>{
\draw[fill=LBlue,LBlue](k11) circle (0.4);
\draw[fill=LBlue,LBlue](k12) circle (0.4);
\draw[fill=LBlue,LBlue](k13) circle (0.4);
\draw[fill=LBlue,LBlue](k14) circle (0.4);
}
\visible<5-7>{
\draw[fill=LBlue,LBlue](b2) circle (0.4);
}
\visible<6-10>{
\draw[fill=LBlue,LBlue](k21) circle (0.4);
\draw[fill=LBlue,LBlue](k22) circle (0.4);
\draw[fill=LBlue,LBlue](k23) circle (0.4);
\draw[fill=LBlue,LBlue](k24) circle (0.4);
}
\visible<9-11>{
\draw[fill=LBlue,LBlue](b3) circle (0.4);
}
\visible<10-14>{
\draw[fill=LBlue,LBlue](k31) circle (0.4);
\draw[fill=LBlue,LBlue](k32) circle (0.4);
\draw[fill=LBlue,LBlue](k33) circle (0.4);
}

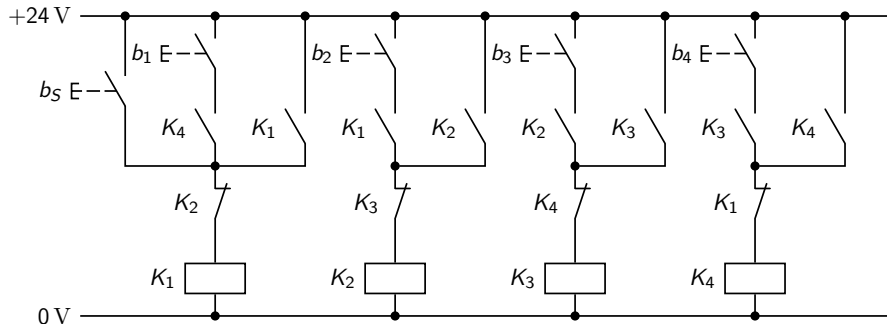
```

```

\draw[fill=LBlue,LBlue](k34) circle (0.4);
}
\visible<13-15>{
\draw[fill=LBlue,LBlue](b4) circle (0.4);
}
\visible<14-18>{
\draw[fill=LBlue,LBlue](k41) circle (0.4);
\draw[fill=LBlue,LBlue](k42) circle (0.4);
\draw[fill=LBlue,LBlue](k43) circle (0.4);
\draw[fill=LBlue,LBlue](k44) circle (0.4);
}
\visible<17-19>{
\draw[fill=LBlue,LBlue](b1) circle (0.4);
}
\end{pgfonlayer}
\end{tikzpicture}}
\par\bigskip
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br
\end{center}
\end{frame}
\end{document}

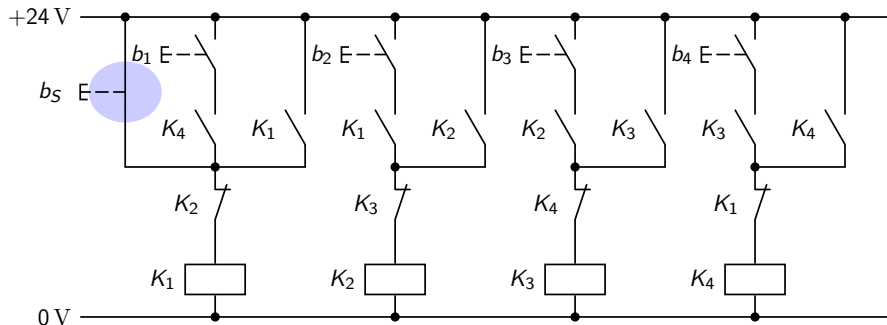
```

Four-Step Sequencer 1



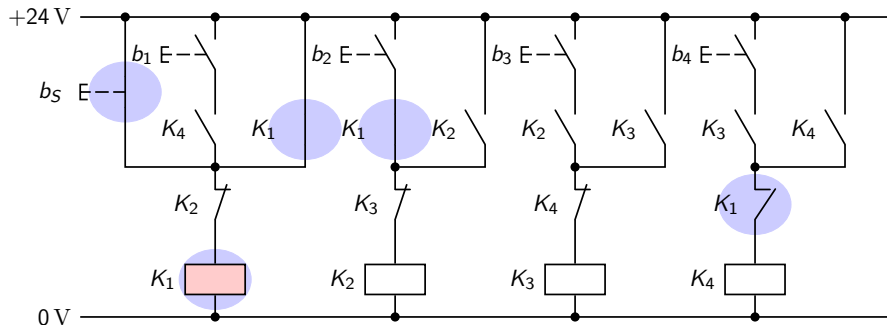
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 2



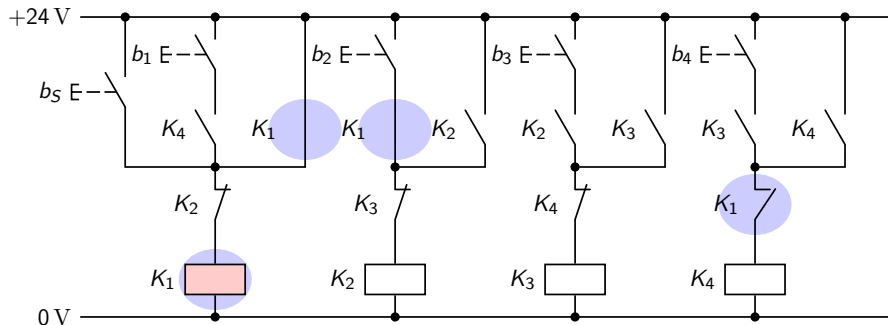
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 3



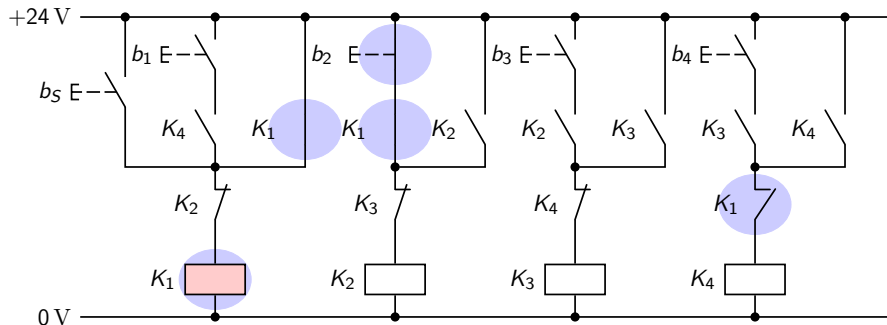
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 4



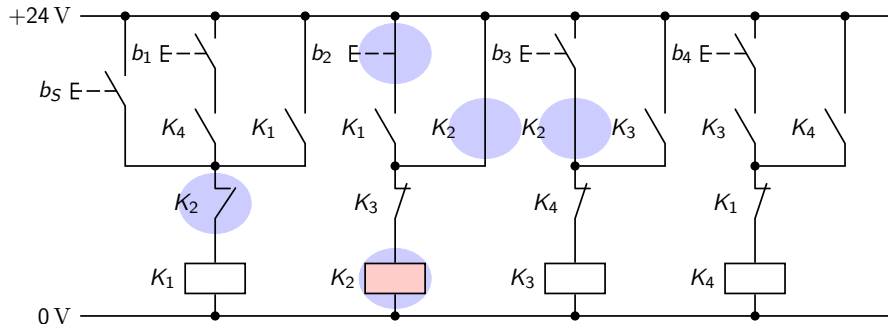
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 5



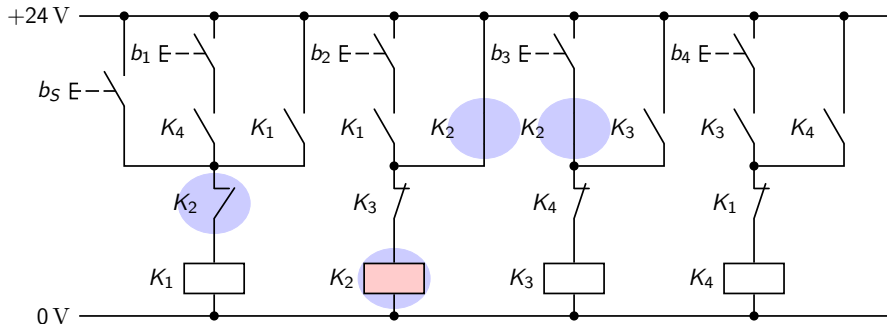
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 7



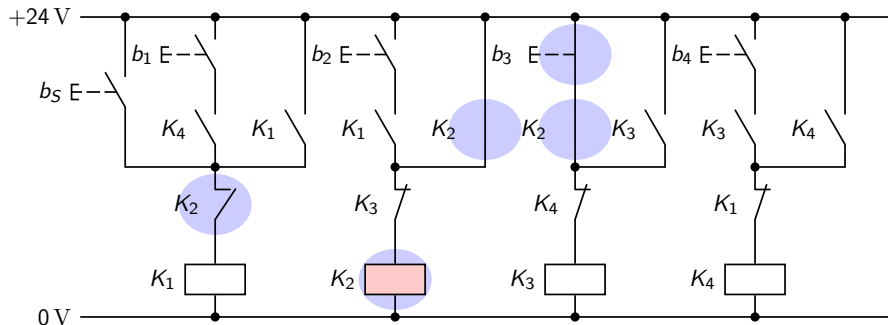
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 8



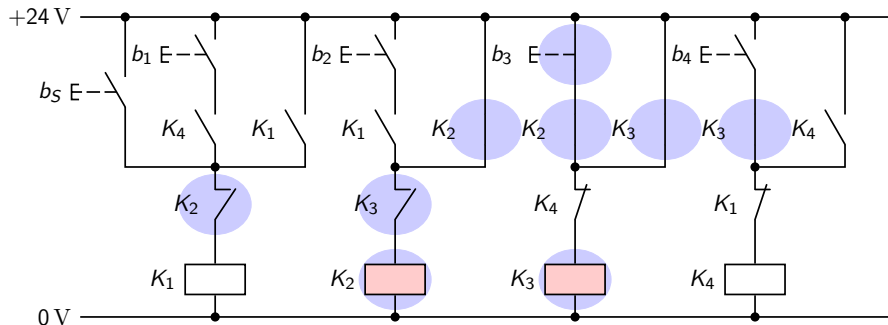
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 9



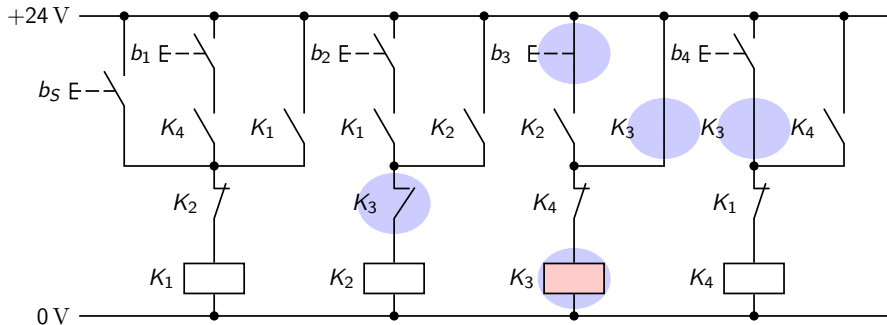
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 10



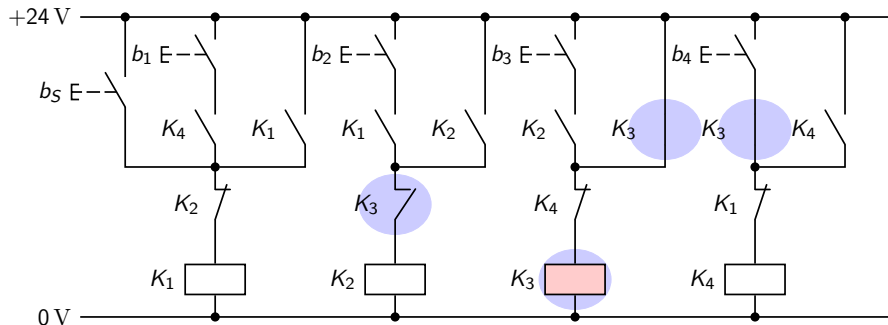
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 11



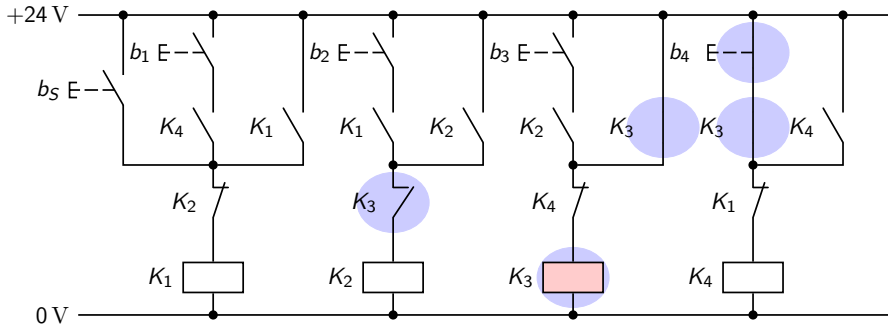
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 12



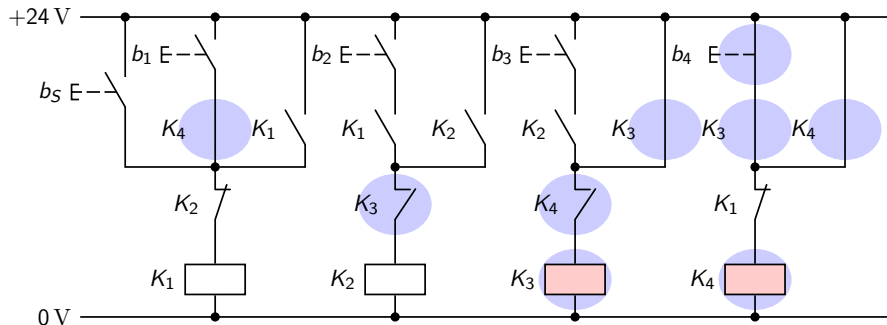
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 13



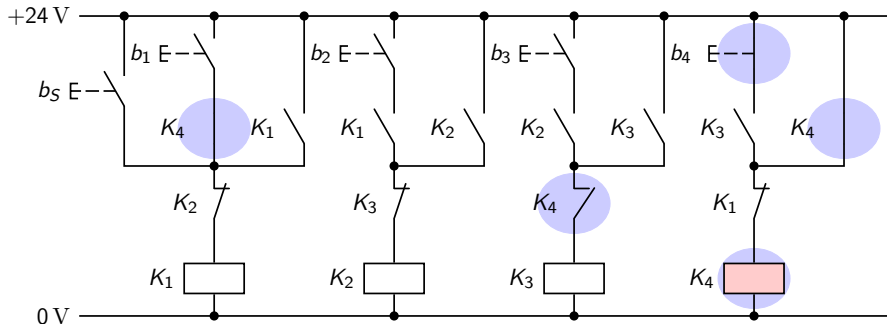
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 14



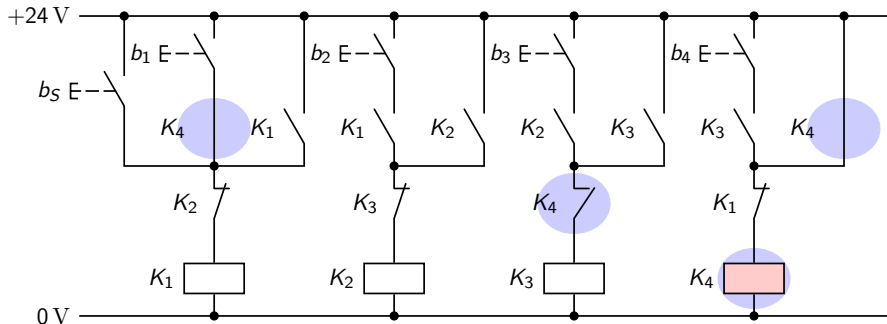
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 15



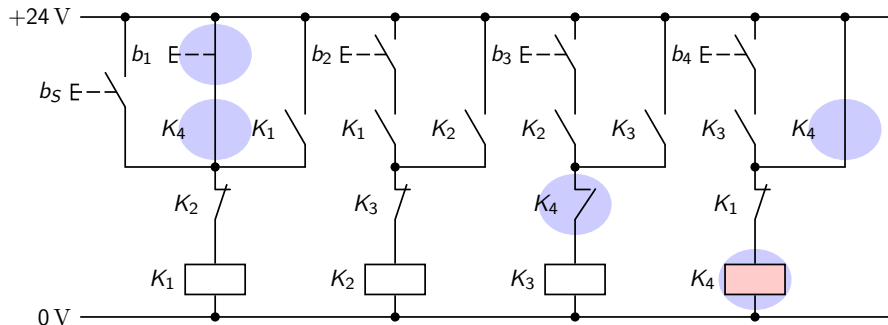
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 16



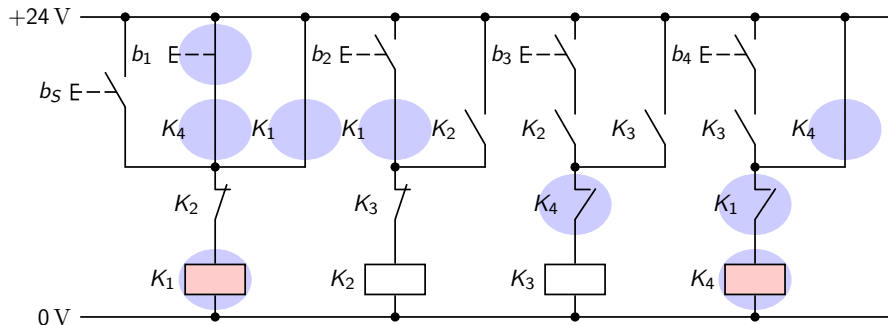
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 17



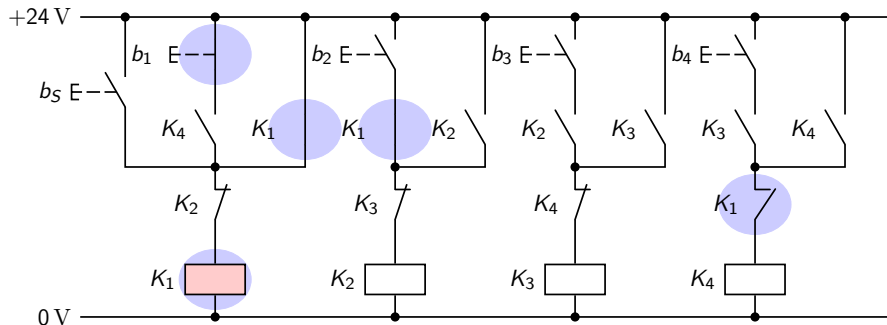
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 18



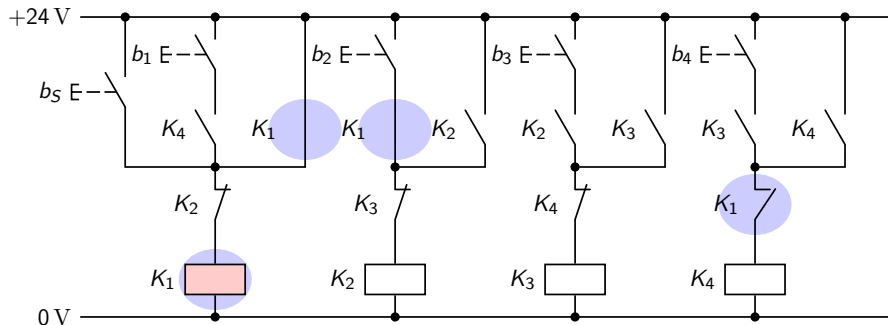
Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 19



Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br

Four-Step Sequencer 20



Copyright (C) 2018 by Luis Paulo Laus, laus@utfpr.edu.br