

Diagramme mit TikZ: Ein Beispiel

Uwe Ziegenhagen

Seit einiger Zeit beschäftige ich mich mit der Erzeugung elektronischer musik und wollte dazu gern ein Blockdiagramm erstellen, das den Signalweg in einem Synthesizer zeigt. Dazu habe ich TikZ genutzt, der Weg zum fertigen Diagramm kann vielleicht auch für einige als Anregung dienen, eigene Zeichnungen mit TikZ anzufertigen.

Grundlagen und Vorlage

Klassische Synthesizer bestehen üblicherweise aus einer Reihe von miteinander verschalteten VCOs (»Voltage-Controlled Oscillators«, spannungsgesteuerten Oszillatoren), VCFS (»Voltage-Controlled Filters«, spannungsgesteuerten Filtern) und VCAs (»Voltage-Controlled Amplifiers«, spannungsgesteuerten Verstärkern), die durch sogenannte ADSR-Hüllkurven gesteuert werden. Visualisiert man diese in einem Blockdiagramm, so ergibt sich ein Bild wie in Abbildung 1, das von der Webseite von Kent Lundberg (MIT) genommen wurde. Diese Abbildung soll die Vorlage für unsere heutige TikZ-nische Reise sein.

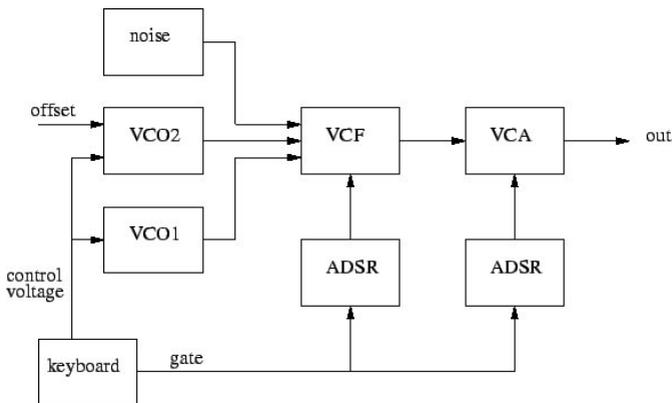


Abb. 1: Vorlage, Quelle: <http://web.mit.edu/klund/www/weblatex/node2.html>

Umsetzung in TikZ

Um eine höher aufgelöste Version der Grafik zu bekommen und damit auch die Möglichkeit zu erhalten, einzelne Teile grafisch betonen zu können, habe ich mich für den »Nachbau« mittels TikZ entschieden. Die Umsetzung soll dabei so nah wie sinnvoll am Original sein.

Mein erster Schritt bei der Umsetzung als TikZ-Zeichnung ist stets, erst einmal ein Gitter zu zeichnen, das als Basis für die darauf folgenden Positions-Arbeiten dient. TikZ kennt dazu die `grid` Option des `\draw` Befehls, Farbe, Schrittweite und Liniendicke dabei können leicht angepasst werden.

Außerdem setze ich gleich einen »Node« (»Knoten«) in die Zeichnung. Wer mit dem Begriff nichts anfangen kann, der findet in der Dokumentation unter <https://tikz.dev/tikz-shapes> die passende Erläuterung, die ich hier gern frei übersetzt wiedergeben möchte: Nodes sind in ihrer einfachsten Form –gegebenenfalls sogar leere –Zeichenketten, die an Koordinaten gesetzt werden. Die Koordinaten kann man dabei fest vorgeben, wie ich es in diesem Beispiel tue, man kann sie aber auch errechnen über die Nähe zu anderen Nodes, ein solches Beispiel werde ich vielleicht in einem Folgeartikel nachreichen. Man kann den Nodes auch Namen geben – was spätere Zeichenoperationen deutlich vereinfacht – sie können Vorder- und Hintergrundfarben sowie eine Form haben, kurz gesagt, es sind wunderbare kleine Konstrukte, die uns das Zeichnen erleichtern.

```
\begin{tikzpicture}
\draw[step=1cm,lightgray,thin] (0,0) grid (10,5);
\node at (2,3) (noise) {Noise};
\end{tikzpicture}
```

Listing 1: Quellcode für Abbildung 2

Im nächsten Schritt kümmern wir uns um die äußere Form der einzelnen Nodes. Dazu definieren wir in den Optionen des `tikzpicture` eine Stilvorlage, die wir dann einzelnen Nodes einfach zuweisen können. Ich habe mich für eine schwarze, rechteckige Form entschieden, die minimale Breite jedes Node wird auf 20 Millimeter, die minimale Höhe auf 10 Millimeter gesetzt, siehe Listing 2.

```
\begin{tikzpicture}[
box/.style={rectangle,thick,draw=black,minimum width=20mm, minimum height=10mm}]
\draw[step=1cm,lightgray,thin] (0,0) grid (10,5);
\node at (2,3) [box] (noise) {Noise};
\end{tikzpicture}
```

Listing 2: Quellcode für Abbildung 3

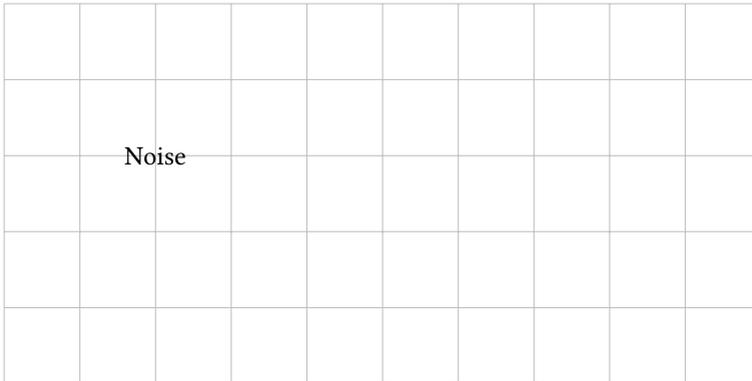


Abb. 2: Ein einzelner Node im Zeichengitter

Damit haben wir alles, um die restlichen Nodes erzeugen und positionieren zu können. Es macht dabei nichts, wenn einzelne Nodes über das Grid hinaus ragen, da in der finalen Version der Zeichnung das Gitter eh nicht enthalten sein wird. Für den zweizeiligen Text im »control voltage« Node habe ich mich eines Kniffs von <https://tex.stackexchange.com/questions/31096/how-can-i-use-linebreak-inside-a-node-in-tikz/31100#31100> bedient, der eine Parbox nutzt, deren Breite mittels `\widthof` Befehl aus dem `calc`-Paket genau auf die Länge des »control«-Worts gesetzt wird.

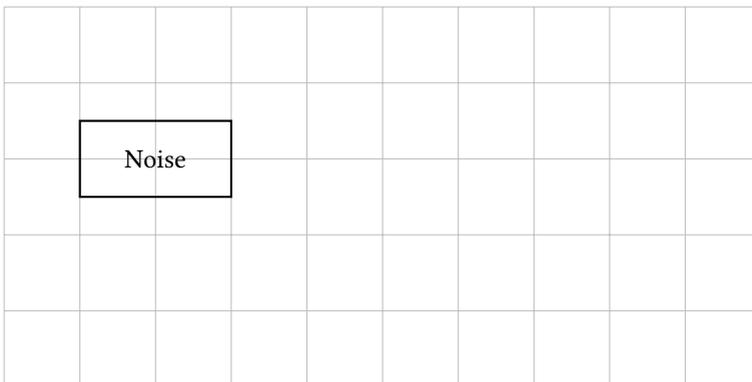


Abb. 3: Gitter und Node, mit Stilvorlage

```

\begin{tikzpicture}[
box/.style={rectangle,thick,draw=black,minimum width=20mm,
minimum height=10mm,align=center},
]
\draw[step=1cm,lightgray,thin] (-1,0) grid (11,7);
\node at (2,6) [box] (noise) {Noise};
\node at (2,4.5) [box] (vco2) {VCO2};
\node at (2,3) [box] (vco1) {VCO1};
\node at (5,4.5) [box] (vcf) {VCF};
\node at (8,4.5) [box] (vca) {VCA};
\node at (5,2.5) [box] (adsr1) {ADSR};
\node at (8,2.5) [box] (adsr2) {ADSR};
\node at (0,1) [box] (keyboard) {Keyboard};

\node at (0,5) (offset) {offset};
\node at (10.5,4.5) (output) {output};
\node at (2,1.25) (gate) {gate};

\node at (-0.75,3) (cv) {\parbox{\widthof{control}}{
control \ \ voltage }};
\end{tikzpicture}

```

Listing 3: Quellcode für Abbildung 4

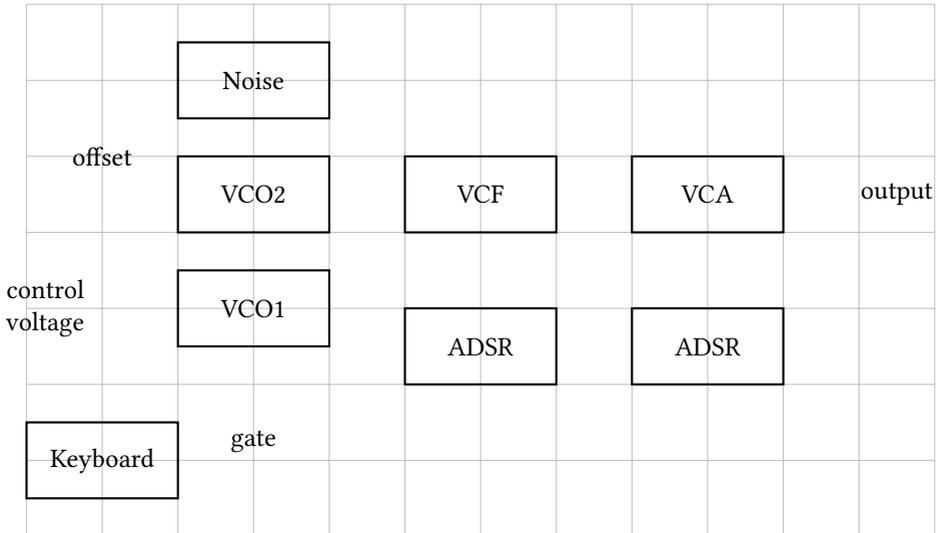


Abb. 4: Grafik mit den finalen Nodes

Bevor wir jetzt im nächsten Schritt die einzelnen Nodes untereinander mit Pfeilen verbinden, machen wir einen kurzen Exkurs dazu, was TikZ uns dafür bereitstellt, Listing 4 zeigt dazu einige der Möglichkeiten. Neben den direkten Linien (mit `--`) gibt es auch eckige Verbindungen (mit `|-` für »erst vertikal, dann horizontal« bzw. `-|` für »erst horizontal, dann vertikal«). Wir können auch die Winkel angeben, mit denen die Pfeile aus einem beziehungsweise in einen Node kommen (`g` und `h`), auch Bezier-Kurven kann man nutzen (`i` und `j`, die beiden Kontrollpunkte sind mit einem `*` markiert). Das letzte Beispiel (`k` und `l`) nutzt dann noch einige der vorgegebenen Anker in den einzelnen Nodes, die über die Himmelsrichtungen adressiert werden. Siehe dazu auch die Grafik in der Anleitung unter <https://tikz.dev/library-shapes>.

```
\begin{tikzpicture}[
box/.style={rectangle,thick,draw=black,
minimum width=10mm, minimum height=10mm,align=center}]
\draw[step=1cm,lightgray,thin] (0,0) grid (8,7);
\node at (1,1) [box] (a) {a};
\node at (3,2) [box] (b) {b};
\draw [thick,->] (a) -- (b);
\node at (1,3) [box] (c) {c};
\node at (3,4) [box] (d) {d};
\draw[thick,->] (c) |- (d);
```

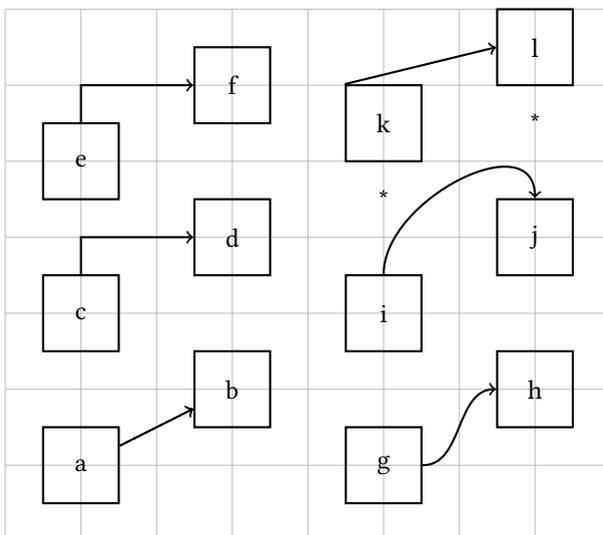


Abb. 5: Exkurs zum Verbinden von Nodes

```

\node at (1,5) [box] (e) {e};
\node at (3,6) [box] (f) {f};
\draw[thick,->] (e) |- (f);
\node at (5,1) [box] (g) {g};
\node at (7,2) [box] (h) {h};
\draw [thick,->] (g) to [out=0,in=180] (h);
\node at (5,3) [box] (i) {i};
\node at (7,4) [box] (j) {j};
\node at (5,4.5) (cp1) {*};
\node at (7,5.5) (cp2) {*};
\draw [thick,->] (i) .. controls (cp1) and (cp2) .. (j);
\node at (5,5.5) [box] (k) {k};
\node at (7,6.5) [box] (l) {l};
\draw [thick,->] (k.north west) -- (l.west);
\end{tikzpicture}

```

Listing 4: Quellcode für Abbildung 5

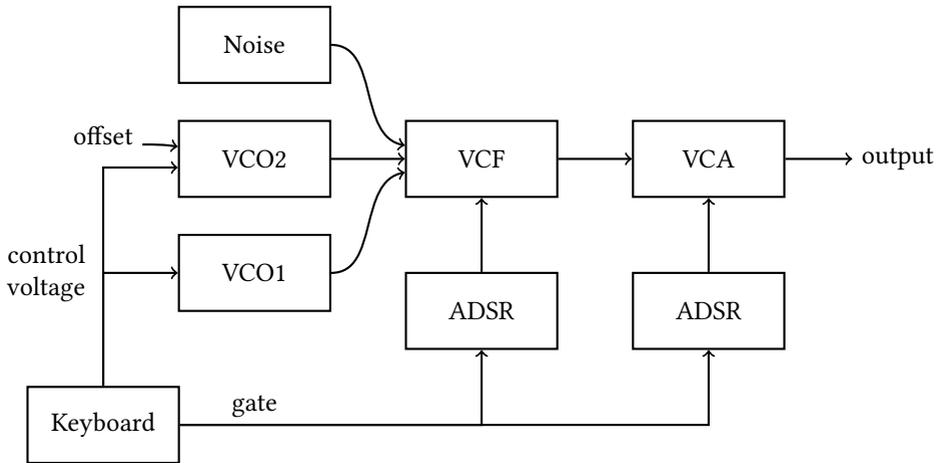


Abb. 6: Finales Diagramm

Um die einzelnen Nodes miteinander zu verbinden, setzen wir einfach weitere `\draw` Befehle, die die Namen unserer Nodes als Parameter erhalten.

```

\draw [thick,->] (vco1) to [out=0,in=190] (vcf);
\draw [thick,->] (vco2) -- (vcf);
\draw [thick,->] (adsr1) -- (vcf);
\draw [thick,->] (adsr2) -- (vca);
\draw [thick,->] (keyboard) -| (adsr1);
\draw [thick,->] (keyboard) -| (adsr2);
\draw [thick,->] (keyboard) |- (vco1);
\draw [thick,->] (keyboard) |- (vco2);
\draw [thick,->] (noise) to [out=0,in=170] (vcf);
\draw [thick,->] (vcf) -- (vca);
\draw [thick,->] (vca) -- (output);
\draw [thick,->] (offset) to [out=0,in=170] (vco2);
  
```

Listing 5: Quellcode für Abbildung 6 (Auszug)

Fazit

TikZ ist toll, so man sich denn mit der manchmal etwas gewöhnungsbedürftigen Syntax angefreundet hat und verstanden hat, wie das mit den Nodes funktioniert.