

# pinouTikz

v1.0.1

User's manual

Robert BLAZEK

[robert.h.blazek@gmail.com](mailto:robert.h.blazek@gmail.com)

---

## ***Abstract***

This package which requires  $\varepsilon$ -TeX, provides macros manipulating strings of tokens. For a basic use, tokens can be alphanumeric chars, but the macros can also be useful for manipulating tokens, i.e. TeX code.

---

# Contents

|          |                     |          |
|----------|---------------------|----------|
| <b>1</b> | <b>Introduction</b> | <b>1</b> |
| 1.1      | Description         | 1        |
| 1.2      | Motivation          | 1        |
| <b>2</b> | <b>The macros</b>   | <b>1</b> |
| 2.1      | The pinout diagrams | 1        |
| 2.1.1    | \PDIP               | 1        |

## 1 Introduction

### 1.1 Description

This package defines macros for generating symbolic pinout diagrams for different package classes, such as DIP, PLCC, etc.

### 1.2 Motivation

Whoever has ever had to do with FPGA or MCUs (whether for living and leisure), it's just natural he or she might have been in a need to document some pins. So was my case and since I failed in finding any package in  $\text{\LaTeX}$  to suit my needs, I opted for creating one myself.

I hope others will find it as useful as it was to me and my colleagues.

This is my first latex package documentation ever - and since I hate reinventing the wheel - this manual has been based upon that of **xstrings** - with the courtesy of the author, of course.

## 2 The macros

For a better understanding, let's see first the macros with the simpler arguments possible. No special catcode, no exotic token, no control sequence neither: only alphanumeric chars will be contained in the arguments.

In the following chapters, all the macros will be presented this way:

- a short description of the operation;
- the operation under special conditions. For each conditions considered, the operation described has priority on that (those) below;
- finally, several examples are given. I tried to find them most easily comprehensible and most representative of the situations met in normal use.

**Important:** in the following, a  $\langle number \rangle$  can be an integer written with numeric chars, a counter, or the result of an arithmetic operation made with the command `\numexpr`.

All the macros of pinouTikz are displayed in `red`.

### 2.1 The pinout diagrams

#### 2.1.1 \PDIP

`\PDIP( $\langle pinnumber \rangle$ ){ $\langle pinarray \rangle$ }`

- $\langle pinnumber \rangle$  thethe number of pins of a DIP package and should be an even number.
- $\langle pinarray \rangle$  is a list of pins - position of every pin's description is defined by its position in this array.

```
1 \begin{figure}
2   \centering
3   \PDIP(4){%
4     1/E,2/B,3/NC,4/C}
5   \caption{NPN-Transistor, 4-pin PDIP package} \label{fig:X_DIP4}
6 \end{figure}
```

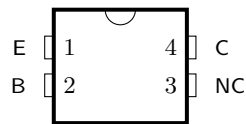


Figure 1: NPN-Transistor, 4-pin PDIP package

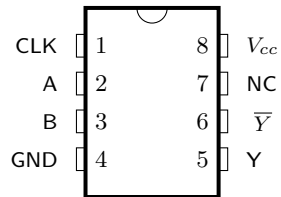


Figure 2: TTL logic chip, 8-pin PDIP package

```

1 \begin{figure}
2   \centering
3   \PDIP(8){%
4     1/CLK,
5     2/A,
6     3/B,
7     4/GND,
8     5/Y,
9     6/\FormatPinLabel{~Y~},
10    7/NC,
11    8/\$V_{cc}$%
12  }
13  \caption{TTL logic chip, 8-pin PDIP package} \label{fig:X_DIP8}
14 \end{figure}

```

```

1 \begin{figure}
2   \centering
3   \PDIP(14){%
4     1/A1, %1
5     2/B1, %2
6     3/\FormatPinLabel{\#1 AND/~OR~/GPIO1}, %3
7     4/Y1, %4
8     5/C1, %5
9     6/\FormatPinLabel{\#2 AND/~OR~/GPIO2}, %6
10    7/GND, %7
11    8/PCLK, %8
12    9/PDAT, %9
13    10/A2, %10
14    11/B2, %11
15    12/\FormatPinLabel{\#2 INV/~SME~/GPIO3}, %12
16    13/C2, %13
17    14/\$V_{cc}$% %14
18  \caption{Generic programmable TTL logic chip, 14-pin PDIP package} \label{fig:X_DIP
19  14}
20 \end{figure}

```

★  
★ ★

That's all, I hope you will find this package useful!

Please, send me an [email](#) if you find a bug or if you have any idea of improvement...

Robert Blazek

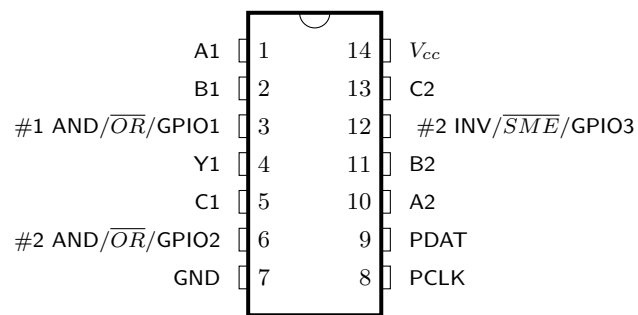


Figure 3: Generic programmable TTL logic chip, 14-pin PDIP package