How To Parse the KiCad Netlist

Doc. Version 2017-07-01-1

Author: Mario Blunk

Abstract: *Guideline to implement software that parses the KiCad netlist or similar files with bracket orientated hierarchic structures.*

Keywords: EDA, KiCad, netlist, device, name, value, package, pin, net, algorithm, flow-chart, hierarchic, bracket, section, subsection, keyword, argument, open source, boundary scan, JTAG, list, map, programming language, Ada

We appreciate every feedback that helps to improve this document !
Please send your comments to info(@)blunk-electronic.de !

Thank you !

Table of Contents

1 About KiCad

Instead of many words, just have a look at http://kicad-pcb.org/ .

2 The Problem to Solve

A given KiCad netlist is to be imported into the *Boundary Scan Test System M-1*. The result of the import process shall be a map of devices (or device list) and a map of nets (or a list of nets). Both maps serve as source to create the system internal skeleton netlist for further test generation.

3 Versions

The approach outlined below has been developed and tested with KiCad Version 4.0.4-stable. This version exports a netlist from Eeschema (the schematic editor) in format version D.

3.1 File Structure

The KiCad netlist stores project data, devices and nets in sections. A section starts with an opening round bracket. Right after the opening bracket the section name follows. The section name is a keyword that indicates the information stored inside the particular section. The text after the keyword, which is not enclosed in brackets, is called the argument.

Subsections are structured the same way and may be nested down to an unlimited level in hierarchy.

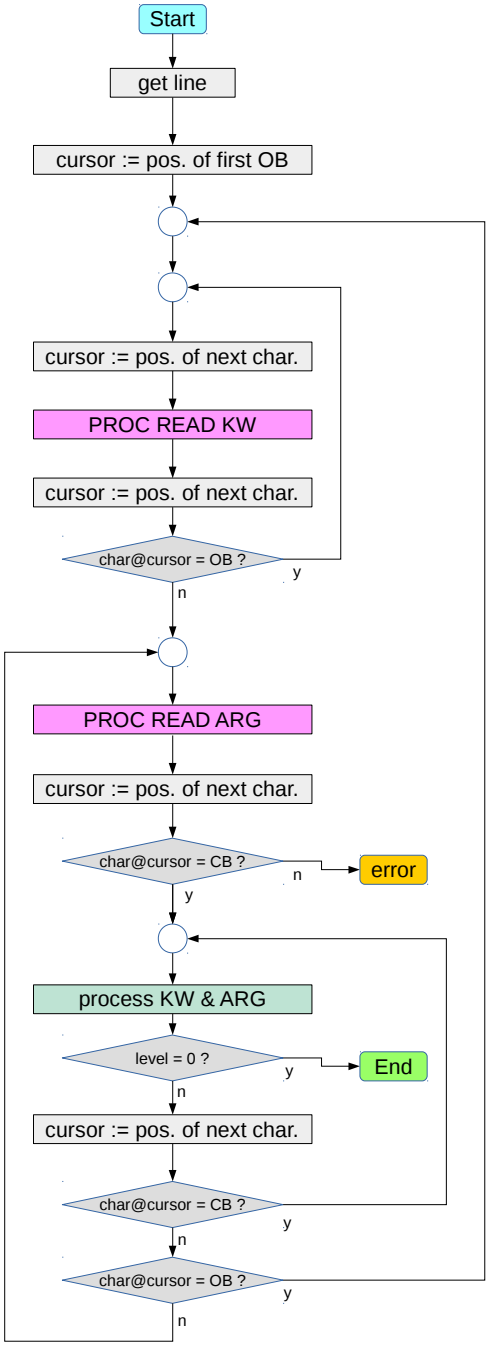
3.2 Netlists Used

The dummy netlist used for developing the following approach and its testing can be found at:

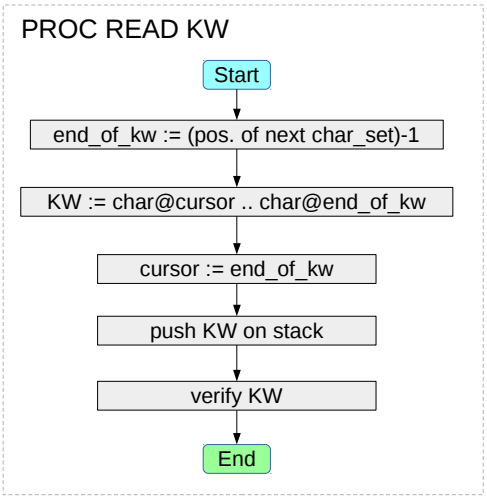
https://github.com/Blunk-electronic/M-1/blob/master/uut/mmu/cad/transmission_line.net

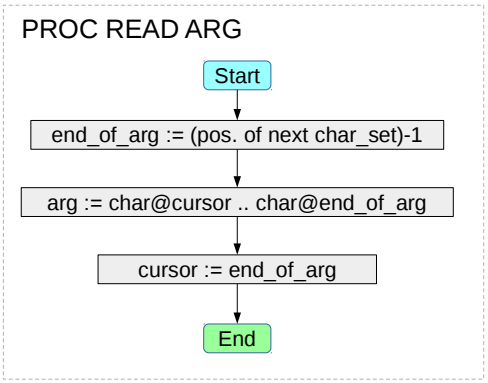
4 The Basic Algorithm

For the start we ignore the fact that the KiCad netlist file has many lines. We assume all characters are rowed up in a single long line.



Notes:
A: OB means opening bracket
B: CB means closing bracket
C: char_set is a set of characters
containing space,horizontal tab, line feed, OB
D: KW means keyword
E: ARG means argument
F: next char means the next
non-blank character (0..9,a..z,A..Z,OB/CB)



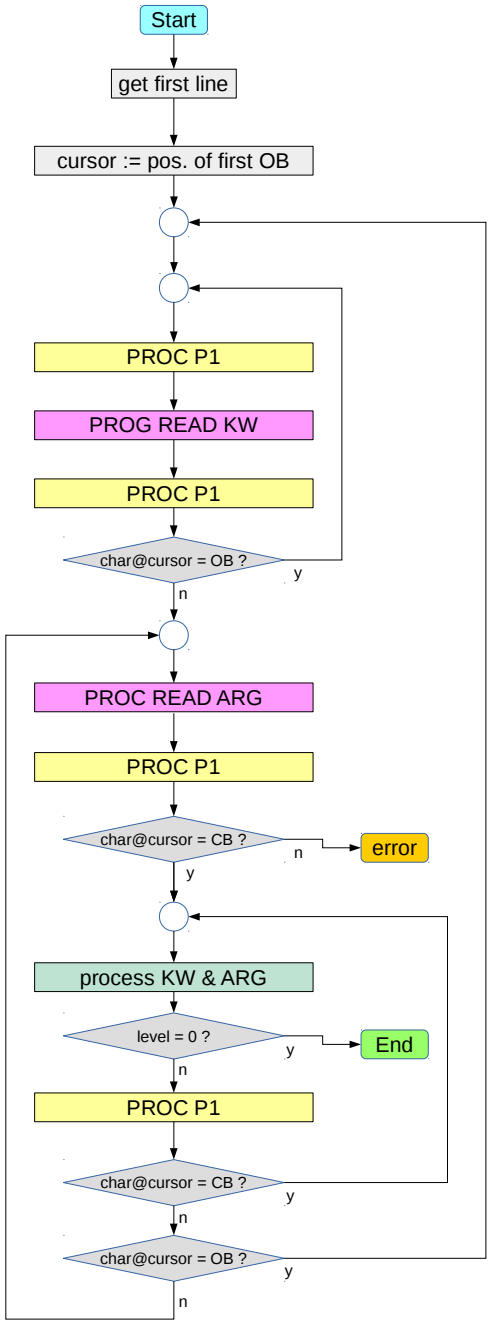


The light green box labeled with “process KW & ARG” is the procedure where we first pop the latest keyword from stack and then do the actual work with the keyword and its argument, for example inserting devices, pins and nets in lists or maps. After this procedure we must check the stack depth. If the top level has been reached (on the last

closing bracket) the import is complete and the algorithm ends. See implementation details in section 6 Implementation on page 6.

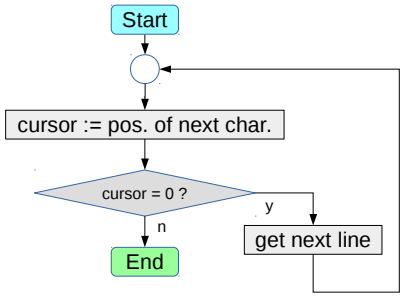
5 Dealing with Lines

The reality is: There are many lines and we must permanently feed the algorithm with new lines if necessary. The box with the action ”cursor := pos of next char” is now replaced by procedure PROC P1 (yellow box). Its purpose is to update the cursor position to the position of the next character starting from the current cursor position AND to fetch a new line if there are no further characters after current cursor position.



Notes:
A: OB means opening bracket
B: CB means closing bracket
C: char_set is a set of characters
containing space,horizontal tab, line feed, OB
D: KW means keyword
E: ARG means argument
F: next char means the next
non-blank character (0..9,a..z,A..Z,OB/CB)

PROC P1



NOTE: cursor = 0 means : no character found

6 Implementation

The approach proposed here can be implemented in any programming language. The realization in Ada 2005 can be found at:

https://github.com/Blunk-electronic/M-1/blob/master/src/impkicad/impkicad.adb

7 Links

- find updates of this document at www.blunk-electronic.de
- Simplify manufacturing fault detection, hardware bring-up, debugging and system tests with *System M-1* the OpenSource Boundary Scan/JTAG Test System at

  http://www.blunk-electronic.de/products.html

8 Disclaimer

This document is believed to be accurate and reliable. I do not assume responsibility for any errors which may appear in this document. I reserve the right to change it at any time without notice, and do not make any commitment to update the information contained herein.

Blunk electronic / Owner : Dipl. Ing. Mario Blunk / Buchfinkenweg 3 / 99097 Erfurt / Germany / Phone +49 361 6022 5184