QSpice C-Block Code Generator (QCodeGen)

User Notes

Document Revisions:

2024.02.13 – Initial Version.

The QCodeGen program is based on potentially-flawed reverse-engineering. While I believe it to be reliable at the time of release, changes to QSpice may break it in the future.

Overview

QCodeGen is a C-Block component code generator for QSpice similar to the built-in QSpice code generator. Improvements over the built-in generator include:

- Generation of code without overwriting existing *.cpp component code files.
- Selective copying of generated code fragments to the clipboard to paste into existing code.
- Improved error analysis for malformed string attributes. See Appendix x.
- Code generation using user-defined custom code templates.
- (Future Plan: Include a size parameter for the passed uData array using custom templates. Maybe add feature to insert a sentinel value directly into the schematic, verify it's presence in code?)
- (Future Plan: Add a feature to convert bit vector port types to/from bit fields.)

Installing/Uninstalling QCodeGen

The most current version of the QCodeGen source code and binaries are available on my GitHub QSpice repository: https://github.com/robdunn4/QSpice.

Installation

Binaries

The QCodeGen executable, required library files, and initial code templates are provided in a *.zip file. There is no special installation program – simply unzip into a folder of your choosing.

Note that the executable (QCodeGen.exe) is compiled for 64-bit Windows. If you need a 32-bit version, you can compile from the sources.

Source Code

The source code was developed using the Qt framework (Qt Creator 12.0.2, Community (free) version) and MSVC 2022. You can get Qt at https://www.qt.io.

Uninstallation

There is no dedicated uninstallation program. Simple delete the folder/subfolders from wherever you installed them.

If you want to remove all traces of QCodeGen:

- QCodeGen saves configuration data (preferences/settings) in the user's home directory: "C:\Users\[user name]\AppData\Roaming\Thin Air Enterprises\QCodeGen.ini".
- I believe that the Qt applications automatically save some general configuration information in the Windows Registry, stuff like the last folder opened in a QFileDialog. I could be wrong. Anyway, you'll need to see the Qt site for more information if this is an issue.

License

QCodeGen is copyrighted software made available under the GNU GPL3 license.

Using QCodeGen

Default template includes all features that can be selected in QSpice code generator. Whitespace may not be identical.

Preferences settings.

Custom Code Templates

Custom code templates make it easy to generate alternate versions. In fact, the distribution includes a "better" version with minor modifications to the default.

When generating code, QCodeGen replaces certain text patterns in the template with code. There are two types of pattern keywords: Embedded and Block.

Embedded keywords may appear anywhere within the template. Block keywords must appear on a line by themselves, i.e., no other text besides white-space may appear on the line.

Keyword Pattern	Type	Replaced With
~~%%author_name%%~~	~~Embedded~~	~~The “Author Name” value from the preferences settings.~~
~~%%date_generated%%~~	~~Embedded~~	~~The current date/time in ??? format.~~
~~%%copyright_text%%~~	~~Embedded~~	~~The copyright text from the preferences settings.~~
%%inst_struct%%	Embedded	The component name as taken from the Symbol Properties “1st Attribute” text. This is typically used for naming the instance data structure but could be used anywhere the DLL name (with unaltered case) is needed.
%%eval_func%%	Embedded	Same as %%inst_struct%% but forced to lower case. This is required to be the name of the evaluation function.
%%undef%%	Block	“#undef” pre-processor directives for each of the uData variable names.
%%udata_all%%	Block	uData[] array element declarations. Typically used in the evaluation function.
%%trunc_udata_output%%	Block	uData[] array elements for only output ports. Intended for use in the Trunc() function.
%%trunc_udata_save_output%%	Block	Code for temporary copies of uData[] array output port elements. Intended for use in the Trunc() function.
%%trunc_udata_restore_output%%	Block	Code to restore uData[] array output port elements from temporary copies. Intended for use in the Trunc() function.

Note: Over-struck items not initially implemented.

Appendix X – Error Analysis

The QSpice code generator will produce invalid code under certain conditions. In particular, a malformed string attribute will corrupt the uData pin/attribute definitions. For example, if the component contains the following string attributes as viewed in the Symbol Properties pane:

Text Order	**Content**
2nd Attribute	`char* mystring1=”some text”`
3rd Attribute	`char* mystring2=”more text`
4th Attribute	`char* mystring3=”even more text”`

Because the mystring2 parameter isn’t properly closed, then QSpice will not generate code for the mystring3 or any subsequent string attributes.

QCodeGen provides warnings for this and other potential issues. See Appendix ? For a list of warnings.

> Note: While the warnings should prevent errors during component development, QSpice generates it’s internal version of the uData parameters each time the simulation is run. If a parameter is subsequently changed and malformed, the runtime array will not match the compiled code. There is no easy fix for this since QSpice doesn’t pass a size parameter at runtime. Hopefully, such problems would corrupt the netlist sufficiently to call attention to such problems...

Final Notes

More?

The union uData declaration includes both “char* str” and “unsigned char* bytes.” Not sure if the latter is used by the Qspice template generator or if it’s just there for convenience for those who know about it. Might be for bit vectors of some arbitrary width. Open question.

Conclusion

If you made it this far, well, congratulations and thanks for sharing the ride. I hope that you find the information useful.

Please let me know if you find problems or suggestions for improving this documentation.

--robert