


[Project \(/qemu/qbox\)](#)[Activity \(/qemu/qbox/activity\)](#)[Repository \(/qemu/qbox/tree/qbox\)](#)[Pipelines \(/qemu,](#)

QBox

QBox or "QEMU in a box" is QEMU with a TLM-2.0 like interface

 passed ([/qemu/qbox/commit/a29cad68cd81b9100f6b60ea6f7bc01ae8cfc8ee/builds](#)) **a29cad68**
([/qemu/qbox/commit/a29cad68cd81b9100f6b60ea6f7bc01ae8cfc8ee](#)) qbox: Modified ARM events
implementation ([/qemu/qbox/commit/a29cad68cd81b9100f6b60ea6f7bc01ae8cfc8ee](#)) · 3 months ago by
Clement Deschamps ([/u/clement](#))

QBox



(<https://static.greensocs.com/logos/qbox-h300.png>)

Overview

QBox or "QEMU in a box" is QEMU with a TLM-2.0 like interface. QEMU is a system platform emulator based on JIT technology which is an industry benchmark in terms of execution speed. It can be used to simulate a wide variety of platforms, based on an equally wide variety of CPU architectures. **QBox allows the powerful JIT based CPU simulations to be exploited within a TLM-2.0 context, such that it can, for instance be instanciated without TLM models in a SystemC simulation environment.**

QBox is sufficiently flexible that some of the basic system components can be included within QBox, so only the device models of interest need to be modelled in SystemC.

The result is that QBox is a flexible platform simulation environment perfect for either adding new components and designing device drivers, or it can be used within the context of a much wider system as merely one of many components.

QBox's are **shared libraries that contain QEMU based CPU models**. Each QBox contains a specific CPU core. QBox's are C objects that export a TLM-2.0 like interface. The GreenSocs Library contains TLM2C (<https://git.greensocs.com/tlm/tlm2c>) bridges to connect C TLM-2 style models to SystemC TLM-2.0. This can be used with QBox in order to allow a QBox to execute with your existing systems SystemC simulation library.

For more information about QBox's and training on how to use QBox's and the GreenSocs SystemC library please contact us (<https://www.greensocs.com/contact>).

QBox also supports the use of:

- Eclipse connection (Use Eclipse for the full debug environment)
- Guest O/S debug (Connect to the running O/S via QEMU's GDB server port)

QBox CPU models

- ARM ARM926
- ARM Cortex-M3
- ARM Cortex-A9

Code status

build success (<https://git.greensocs.com/qemu/qbox/commits/qbox>)

Requirements

QBox requires:

- git > 1.7
- tlm2c (<https://git.greensocs.com/tlm/tlm2c>) > 1.0
- glib2 dev
- pixman dev

Official OS support:

- Linux :
 - Ubuntu 12.04 and 14.04
 - CentOS 6 and 7
- Windows with MinGW
- OS X : soon

Linux

Ubuntu / Debian :

```
$ sudo apt-get install libglib2.0-dev libpixman-1-dev
```

CentOS / Fedora :

```
$ sudo yum install glib2-devel pixman-devel
```

Windows

Please try to install packages in paths without spaces.

- MinGW (<http://sourceforge.net/projects/mingw/>) with MSYS

With MinGW Installer, install these following packages:

MinGW

- gcc
- gcc-g++
- libz-dev
- pthreads-w32-dev

MSYS

- flex
- bison

You also need:

- pkg-config
- glib
- libpixmap

You can compile these packages using sources and Google or you can download binaries (fastest).

pkg-config, glib and libpixmap can be found here: GTK Download (<http://www.gtk.org/download/win32.php>).

Then, install packages (copy / paste files) in your MinGW installation.

Download

Be sure to clone the repository and include dtc submodule. After cloning the repository, you should run:

```
$ git submodule update --init dtc
```

Configure

```
$ ./configure --target-list=arm-softmmu --library --qbox --with-tlm2c=[TLM2C_PATH] --pref
```

Please fill options:

- [TLM2C_PATH] : Specify TLM2C (<https://git.greensocs.com/tlm/tlm2c>) build path
- [PREFIX_PATH] : Specify build path
- This will result in generating the Makefile to build QBox.

Example

```
./configure --target-list=arm-softmmu --library --qbox --with-tlm2c=`pwd`/tlm2c/build --p
```

Build

```
make
```

or (fastest)

```
make -j [NUMBER_OF_PARALLELS_JOBS]
```

This action will result in the compilation of QBox.

Installation

```
$ make install
```

Configure

Some options are available to configure QBox through GSParam (a part of GreenLib):

Name	Description	Default value
CPU.gdb_port	GDB server port. Must be > 0 to enable GDB server	0 (= disable)
CPU.extra_arguments	Pass extra arguments to QEMU	<i>None</i>
CPU.kernel_cmd	Kernel command (equivalent to --append on QEMU)	<i>None</i>
CPU.rootfs	Rootfs path (equivalent to --initrd on QEMU)	<i>None</i>
CPU.quantum	Quantum	<i>None</i>
CPU.kernel	Kernel path (equivalent to --kernel on QEMU)	<i>None</i>
CPU.dtb	DTB path (equivalent to --dtb on QEMU)	<i>None</i>

TODO

- Coming Soon... Multi-host thread support (multi-core platforms simulated on multi-host threads for efficacy and speed)
- Mac OS X support

Known Issues

- An problem has been reported with gcc 4.8.5 which causes an TLS error with more than 4 instances of qbox.

Contributing

QBox is an open source, community-driven project. If you'd like to contribute, please feel free to fork project and open merge request or to send us (<https://www.greensocs.com/contact>) patch.
