

vmm_automated_software Project ID: 86695 39 Commits 1 Branch 0 Tags 1.1 MB Files 1.1 MB Storage

VTS software by Danny adapted for Roodmicrotec

Auto DevOps It will automatically build, test, and deploy your application based on a predefined CI/CD configuration. Enable in settings

master vmm_automated_software / + History Find file Web IDE Clone

setup working at Roodmicrotec change of art on threshold George Iakovidis authored 5 months ago 5dc38614

README Add LICENSE Add CHANGELOG Add CONTRIBUTING Add Kubernetes cluster Set up CI/CD

Table with 3 columns: Name, Last commit, Last update. Rows include qt, vts, and README.md.

README.md

vmm_automated_software based on https://github.com/dantrim/vmm_testing_software

This repository contains the software implementing the testing functionality for the VMM3a ASIC, referred to as VTS (VMM Testing Software).

- Base Requirements: Python >= 3.6, A recent version of Qt 5 (preferably Qt 5.13), ROOT6, Boost >=1.60

Installation

- Installing pre-requisites: Python3 (macOSx, linux), Qt

- ROOT: See detailed instructions from VERSO

- Boost: See detailed instructions from VERSO, Catalina has issues on compiling. Use the following code to comile:

```
cd boost_1_60_0
./bootstrap.sh --with-libraries=atomic,chrono,date_time,exception,filesystem,system,thread,timer --prefix=<install-d
./b2 install toolset=clang architecture=x86 address-model=64 threading=multi --layout=tagged --prefix=<install-d
```

Installing VTS

VTS Server backend Follow these instructions:

```
git clone https://gitlab.cern.ch/giakov/vmm_automated_software.git
```

Edit vts.pro, specifically the lines with boostinclude and boostlib to point to your locations of boosts include/ and lib/ directories. Nothing else in vts.pro needs to be changed.

Next, we compile the VTS source code:

```
cd vmm_automated_software/
cd vts/
mkdir build
cd build/
qmake ../vts.pro
make 2>&1 |tee compile.log
```

If this compilation succeeds, a binary named vts_server should be located in vmm_automated_software/vts/build/.

VTS user interface and python virtual env Next, we obtain the python dependencies for running the user interface and top-level VTS interface, which is all written in python. The python implementation of Qt, PySide2 is used for this. We grab all of the dependencies by setting up a virtual environment.

```
$ cd vmm_automated_software/
$ cd vts/
$ python3 -m venv vtsenv
(vtsenv) $ source vtsenv/bin/activate
(vtsenv) $ pip3 install -r requirements.txt
(vtsenv) $ deactivate
$
```

Once this succeeds you should be able to, for example, run the following command without fail, in a new shell:

```
$ cd vmm_automated_software/
$ cd vts/
$ source vtsenv/bin/activate
(vtsenv) $ python
(vtsenv) $ >>> from PySide2 import QtCore
(vtsenv) $ >>> [Ctrl]+d
(vtsenv) $ deactivate
$
```

VTS Configuration for your machine If you have followed the last two steps, then all of the software required for VTS is installed successfully. Now we need to edit the configuration for VTS to be able to run.

- 1. Edit vts/config/vts_default.json so that the paths up to the "vmm_automated_software/" are correct for your machine

That's it!

Running VTS

Each time you want to run VTS, you must return to the virtual environment:

```
$ cd vmm_automated_software/vts/
$ source vtsenv/bin/activate
(vtsenv) $ cd python/
(vtsenv) $ python vts.py --gui
```

At which point the VTS user interface will appear on your screen.

Once done with VTS, to exit the virtual environment simply do:

```
(vtsenv) $ deactivate
$
```

Modifying the .ui file and converting it back to .py

Under the general directory there is a test project inside the qt directory. This is used to create the ui but then it needs to be converted to a python gui. To do so:

```
$ cd vmm_automated_software/vts/python/interface
$ pyside2-uc ../../qt/mainwindow.ui -o vts_mainwindow.py
```

That will regenerate the .py file. In order to load the logo again modify the line self.label_10.setPixmap(QPixmap(u":/python/vts_logo.png")) with self.label_10.setPixmap(QPixmap("interface/resources/vts_logo.png"))