



Proview on Raspberry PI

2014 02 09

Copyright SSAB EMEA AB 2014

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

Table of Contents

- Introduction.....4
- Development.....4
- Install the cross compiler.....4
- Install Proview.....4
- Create a project.....5
- Configure the project.....5
- Build the project.....8
- RPI installation.....8

Introduction

Raspberry PI is a small singleboard computer developed to promote the the teaching of basic computer sience in scools. This document describes how build a Proview project for raspbian on Raspberry PI. The reader expects to have some knowlege of how to create an build projects in Proview.

Development

There are two ways to develop a system for Raspberry PI. On way is to install the Proview develoment package on an RPI board with raspbian installed. The installation and creation of a projects follows the same procedure as for an ordinary debian system. It will work for small projects but is not ideal for larger projects.

The other way is to use an ordinary pc with ubuntu or debian, and to install the rpi cross compiler to generate code for the Raspberry PI. In this case you have to have access to the Proview runtime archives, cross compiled for RPI, and they are available in the pwrrpi package. This document will describe how to install the cross compiler, and configure a Proview project for Raspberry PI.

Install the cross compiler

On the development station, the cross compiler for RPI should be installed.

The cross complier is fetched from github.com with git.

Install git

```
> sudo apt-get install git
```

Download the cross compiler to /usr/local/rpi.

This is done as super user.

```
> sudo su
> mkdir /usr/local/rpi
> cd /usr/local/rpi
> git clone git://github.com/raspberrypi/tools.git
```

When this is written in Februari 2014, the latest version contains a severe bug, so checkout a previous version

```
> git checkout 9c3d7b6ac692498dd36fec2872e0b55f910baac1
```

Log out as super user

```
> exit
```

Install Proview

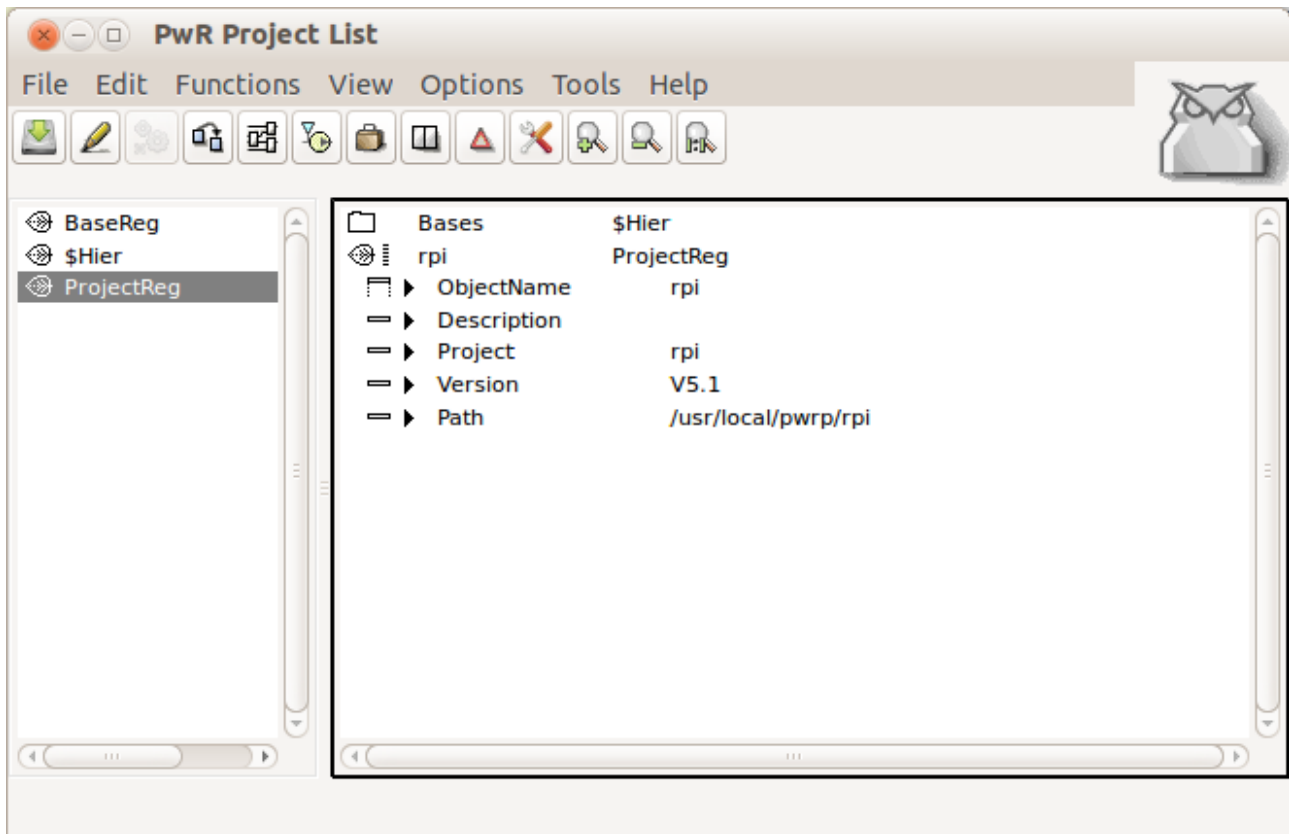
Download and install the Proview develoment package, pwr51, and the corresponding Proview RIP

development package pwrpi51.

Create a project

Login as user pwrp and start the administrator to create an rpi project.

In this example the project name is set to rpi.



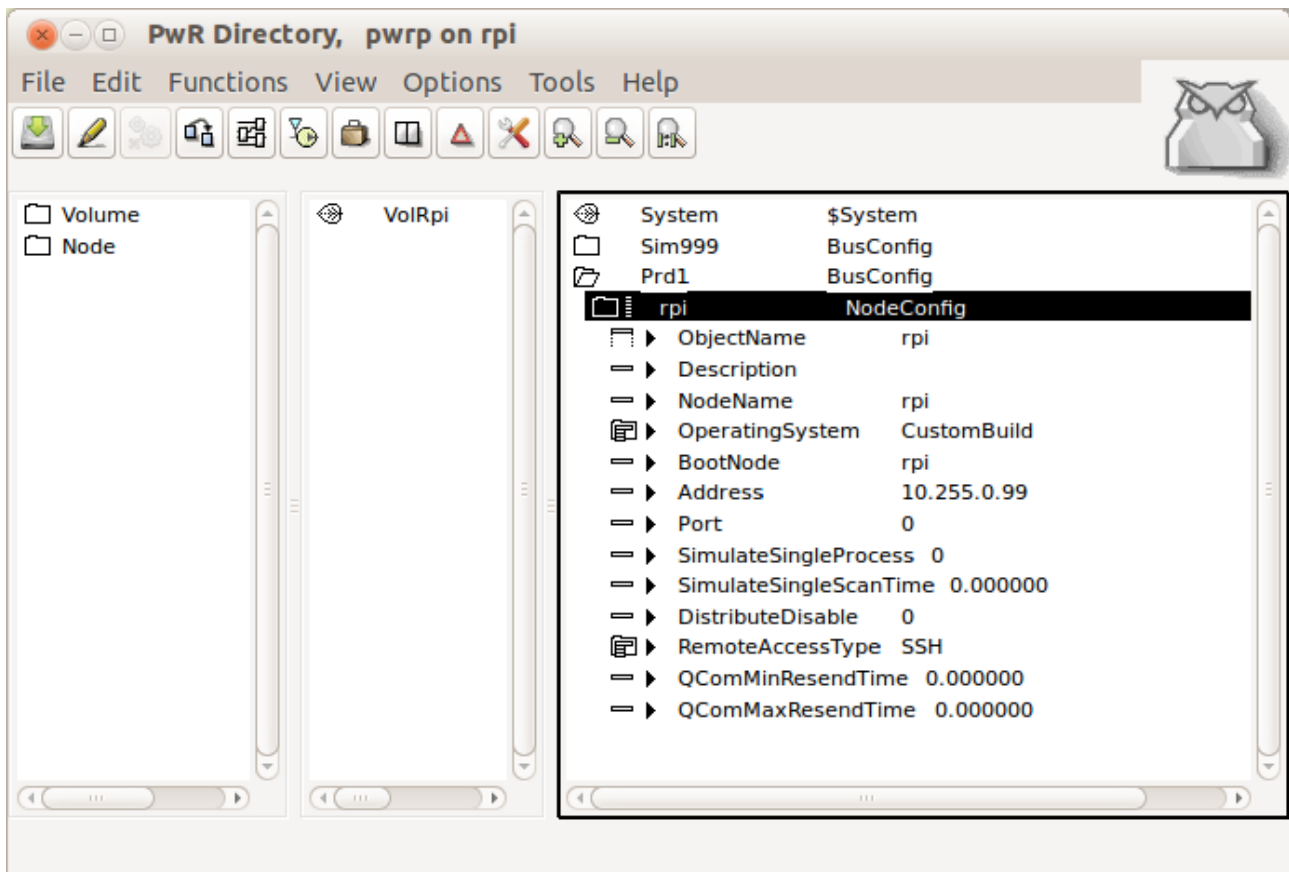
Open the new project by leaving edit mode and activate 'Open Project' in the popup menu for the ProjectReg object.

Configure the project

Run the configurator wizard and apply the default configuration.

Before leaving the directory volume, make following modifications.

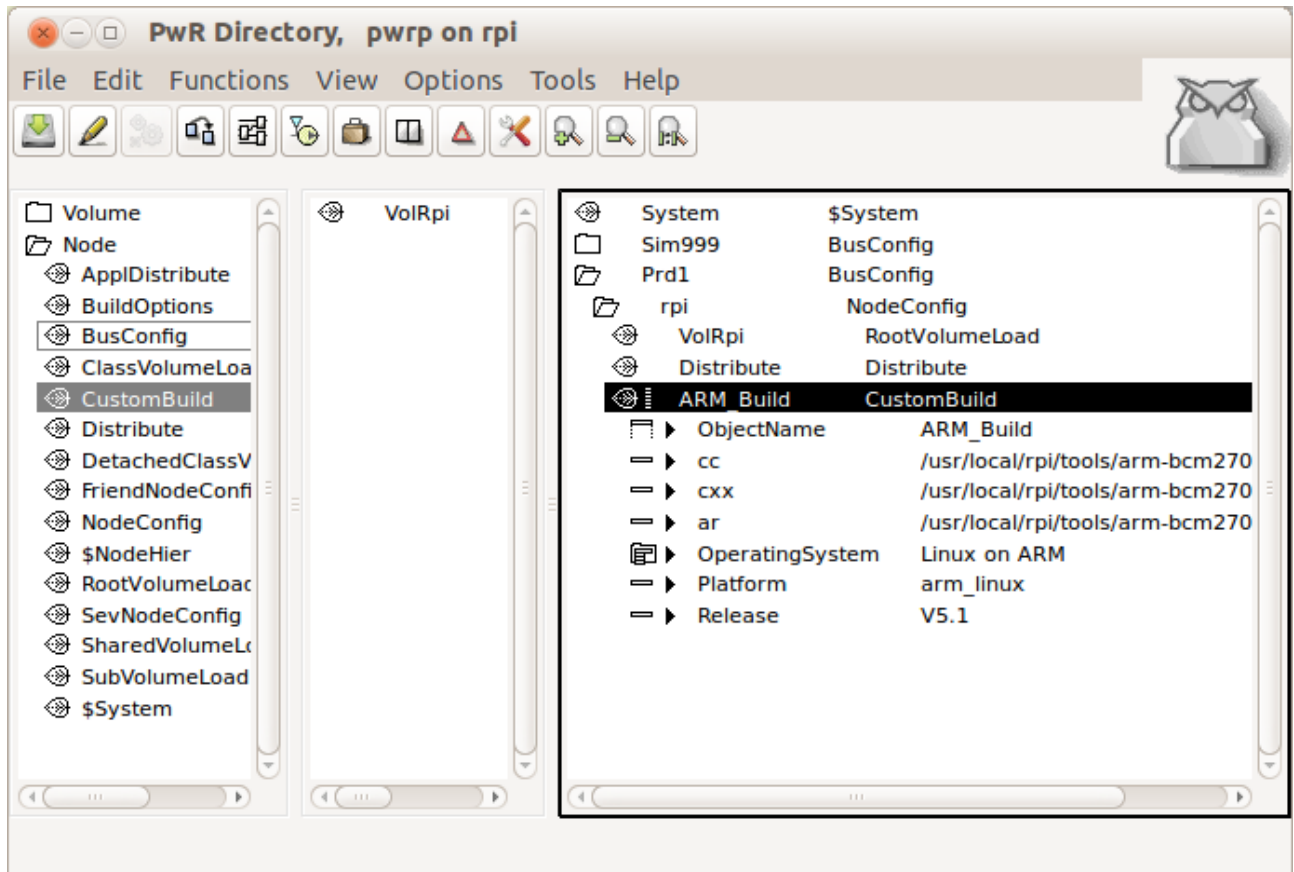
Change the OperatingSystem of the NodeConfig object for the Raspberry PI node to CustomBuild. Also insert the IP Address if this is not previously done, and the node name of the RPI in NodeName.



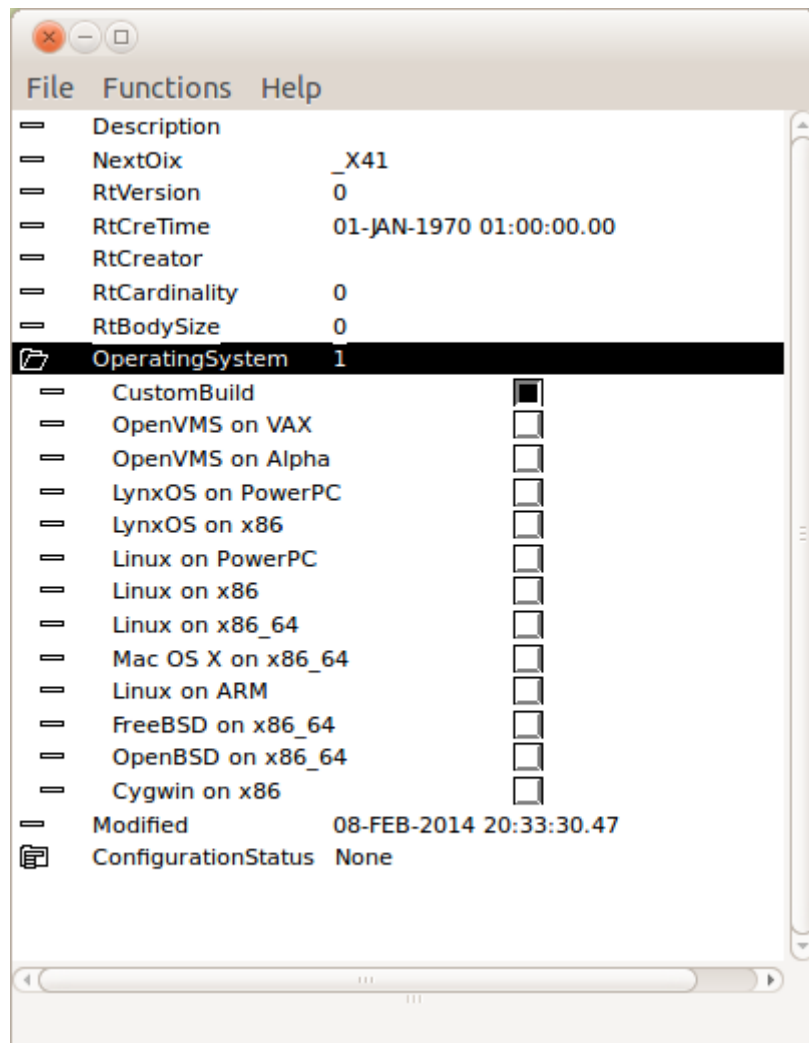
Add a CustomBuild object below the NodeConfig object.

Set

- OperatingSystem to Linux on ARM
- Platform to arm_linux
- Release to the current Preview release.
- CC= /usr/local/rpi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi-hf-raspbian/bin/arm-linux-gnueabi-hf-gcc
- CXX= /usr/local/rpi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi-hf-raspbian/bin/arm-linux-gnueabi-hf-g++
- ar= /usr/local/rpi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi-hf-raspbian/bin/arm-linux-gnueabi-hf-ar



Open the rootvolume, in the example VolRpi, execute the basic configuration with the wisard, and then open volume attributes from File/Volume Attributes in the menu. Set operatingSystem to CustomBuild.



Build the project

Build the RPI node by activating the build button in the configurator tool bar, and select the rpi node in the list.

RPI installation

Install raspbian on the RPI board, and then the pwrrt raspbian package. Finally distribute the RPI project and startup Proview.