Утверждаю

Генеральный директор

ООО «ДЕЦИМА»

\_\_\_\_\_\_\_\_\_\_\_ А.А. Шкляев

“\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2017

Программный комплекс КОТМИ-14

МЦВК. Модуль центрального вычислителя-коммуникатора

Руководство пользователя

Лист утверждения

ЯКШГ.00067-01 91 01-03 32 - ЛУ

|  |  |  |
| --- | --- | --- |
|  |  | Руководитель разработки |
|  |  | Начальник лаборатории |
|  |  | \_\_\_\_\_\_\_\_\_\_\_А.В. Тумаков |
|  |  | “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2017 |
|  |  |  |
|  |  | Ответственный исполнитель |
|  |  | Ведущий инженер-программист |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_М.Ю. Дьяченко |
|  |  | “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2017 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Утвержден

ЯКШГ.00067-01 91 01-03 32 - ЛУ

2017

**Программный комплекс КОТМИ-14**

**МОДУЛЬ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА-КОММУНИКАТОРА**

**Руководство пользователя**

**ЯКШГ.00067-01 91 01-03 32**

**Листов**

**АННОТАЦИЯ**

Данный документ является руководством пользователя (системного администратора) для модуля центрального вычислителя-коммуникатора комплекса КОТМИ-14 (далее по тексту – Мцвк).

Данный модуль служит для приема и передачи сигналов и команд на объектах электроэнергетики.

В документе приводятся необходимые сведения по обращению, проверке и настройке Мцвк на требуемые условия эксплуатации.

Для эксплуатации программно-аппаратного комплекса Мцвк предполагаются базовые знания операционной системы Linux, локальных вычислительных сетей, а также программных утилит типа telnet (ftp, pscp, ...).

**СОДЕРЖАНИЕ**

**Лист**

[1. Общие сведения о программе 4](#_Toc522183015)

[1.1. Функциональное назначение и область применения 4](#_Toc522183016)

[1.2. Состав дистрибутива 4](#_Toc522183017)

[1.3. Требования к инструментальной машине 4](#_Toc522183018)

[2. Настройка МЦВК 6](#_Toc522183019)

[2.1. Установка ПО 6](#_Toc522183020)

[2.2. Настройка сетевых интрефейсов Мцвк 8](#_Toc522183021)

[2.3. Настройка автоматического запуска Сервера Ввода Вывода, при старте Мцвк 11](#_Toc522183022)

[3. Установка и настройка инструментальной ОС на виртуальную машину 13](#_Toc522183023)

[4. Bonding 18](#_Toc522183024)

[5. Yocto 21](#_Toc522183025)

[6. Yocto 29](#_Toc522183026)

# Общие сведения о программе

## Функциональное назначение и область применения

## Состав дистрибутива

В состав распространяемого ПО для Мцвк входит:

* Дистрибутив программного пакета «Сервер ввода-вывода» — 1 шт

Дистрибутив программного пакета «Сервер ввода-вывода» содержит следующие компоненты:

* Сервер ввода-вывода
* Монитор сервера ввода-вывода
* Конфигуратор сервера ввода-вывода
* Конвертер шаблонов ввода-вывода

## Требования к инструментальной машине

Настройка Мцвк выполняется при помощи инструментальной машины. Это может быть ноутбук или любой другой персональный компьютер с операционной системой Windows. Подключение инструментальной машины и Мцвк выполняется через сетевые адаптеры устройств напрямую, или через коммутатор локальной вычислительной сети.

На инструментальной машины должен находится рабочий каталог с устанавливаемым программным обеспечением.

Перечень файлов:

***RdxServer*** – исполняемый модуль Сервера Ввода Вывода.

***RdxCopy*** – утилита копирования.

***RdxServer.ini*** – файл содержащий параметры запуска Сервера Ввода Вывода.

***libScdMdx.so, libScdMdx.so.1, libScdMdx.so.1.0, libScdMdx.so.1.0.0*** – библиотека Сервера Ввода Вывода.

***libRdxMdx.so, libRdxMdx.so.1, libRdxMdx.so.1.0, libRdxMdx.so.1.0.0*** – библиотека протокола Mdx.

***libRdxMonitor.so, libRdxMonitor.so.1, libRdxMonitor.so.1.0, libRdxMonitor.so.1.0.0*** – библиотека протокола Monitor.

***libRdxBridge.so, libRdxBridge.so.1, libRdxBridge.so.1.0, libRdxBridge.so.1.0.0*** – библиотека протокола Bridge.

***libRdxIec101.so, libRdxIec101.so.1, libRdxIec101.so.1.0, libRdxIec101.so.1.0.0*** – библиотека протокола Iec101.

***libRdxIec104.so, libRdxIec104.so.1, libRdxIec104.so.1.0, libRdxIec104.so.1.0.0*** – библиотека протокола Iec104.

***libRdxModbus.so, libRdxModbus.so.1, libRdxModbus.so.1.0, libRdxModbus.so.1.0.0*** – библиотека протокола Modbus.

***libRdxCan.so, libRdxCan.so.1, libRdxCan.so.1.0, libRdxCan.so.1.0.0*** – библиотека протокола Can.

***libRdxSnmp.so, libRdxSnmp.so.1, libRdxSnmp.so.1.0, libRdxSnmp.so.1.0.0*** – библиотека протокола Snmp.

***configName.bin, configName.xml*** – конфигурационный файл.

***telnet.exe, pscp.exe*** – вспомогательные утилиты.

# Настройка МЦВК

## Установка ПО

Статические предустановленные IP адреса комплекса Мцвк.

- порт eth0 – 192.168.1.11, 192.168.3.11

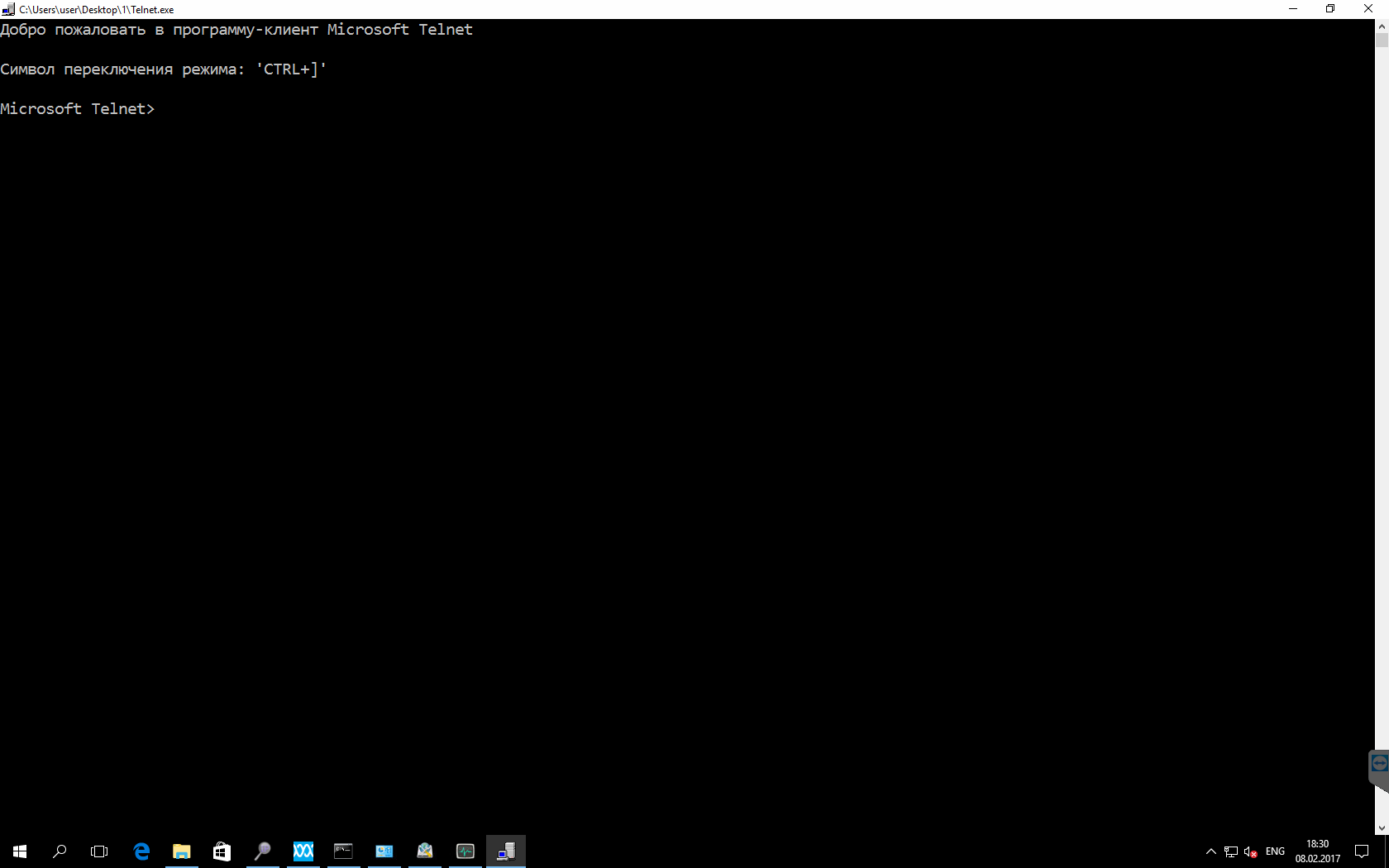
- порт eth1 – 192.168.4.11

Пользователь операционной системы Linux: root. Пароль отсутствует.

2.1.1. На инструментальной машине создать дополнительный IP адрес с подсетью 192.168.1.ххх. Например: 192.169.1.2.

2.1.2. Подключить сетевой адаптер *eth0* контроллера Мцвк к сетевому адаптеру инструментальной машины.

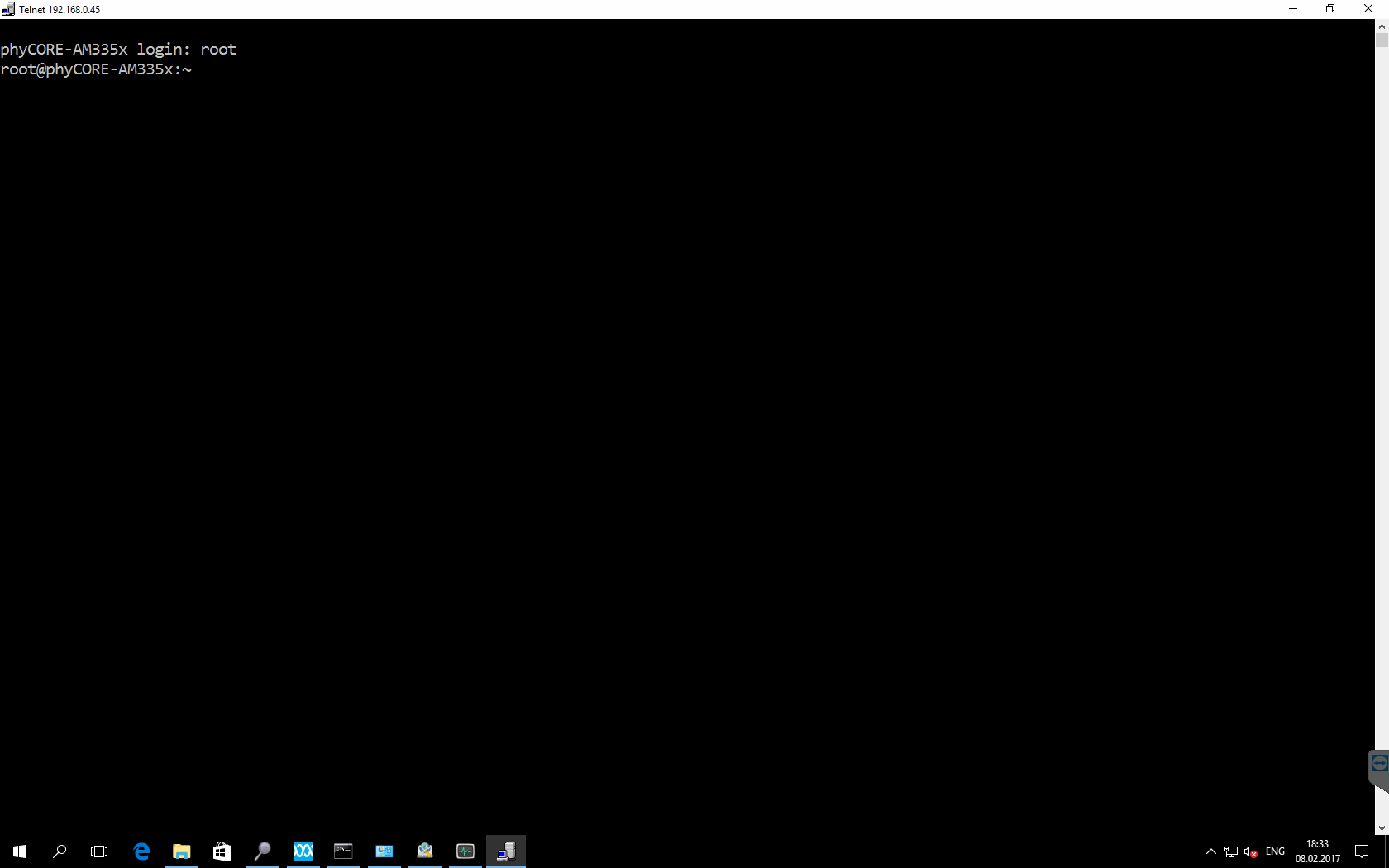
2.1.3. На инструментальной машине запустить утилиту *telnet*.



Посредством утилиты *telnet*, выполняются операции на целевой машине, то есть на Мцвк.

2.1.4. На целевой машине ввести команду: >open 192.168.1.11

2.1.5. На целевой машине ввести имя пользователя: root.



2.1.6. На целевой машине создать каталог для лог-файлов:

>mkdir logs

2.1.7. На целевой машине создать каталог для файлов конфигурации:

>mkdir configurations

2.1.8. На инструментальной машине открыть командную строку cmd.

2.1.9. На инструментальной машине перейти в рабочий каталог.

2.1.10. При помощи утилиты pscp скопировать с инструментальной машины рабочие файлы на Мцвк.

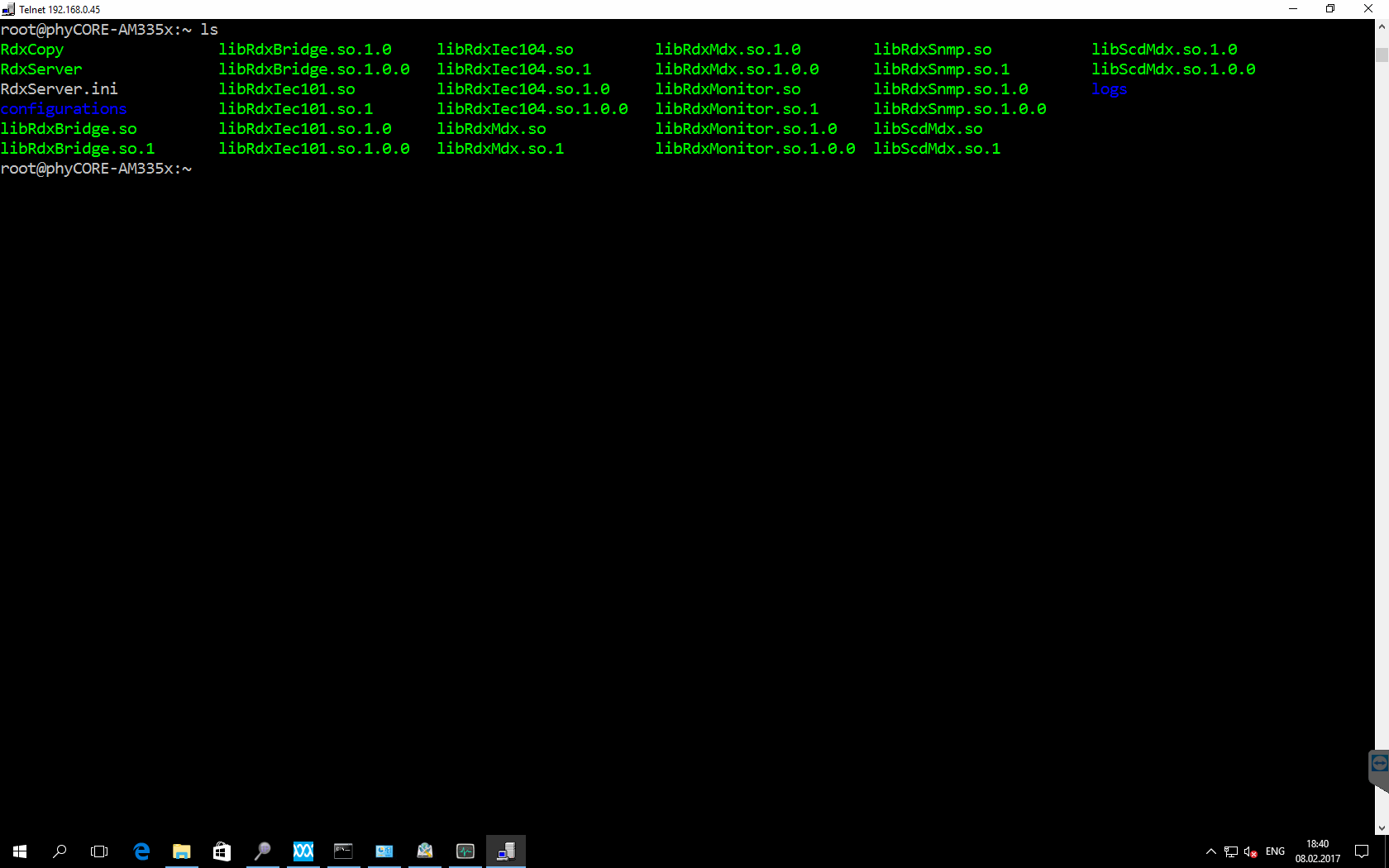
>pscp Rdx\* root@192.168.3.11:/home/

>pscp lib\* root@192.168.3.11:/home/

>pscp config\* root@192.168.3.11:/home/configurations/

2.1.9. На целевой машине проcмотреть содержимое каталога home/, при помощи команды ls:

>ls



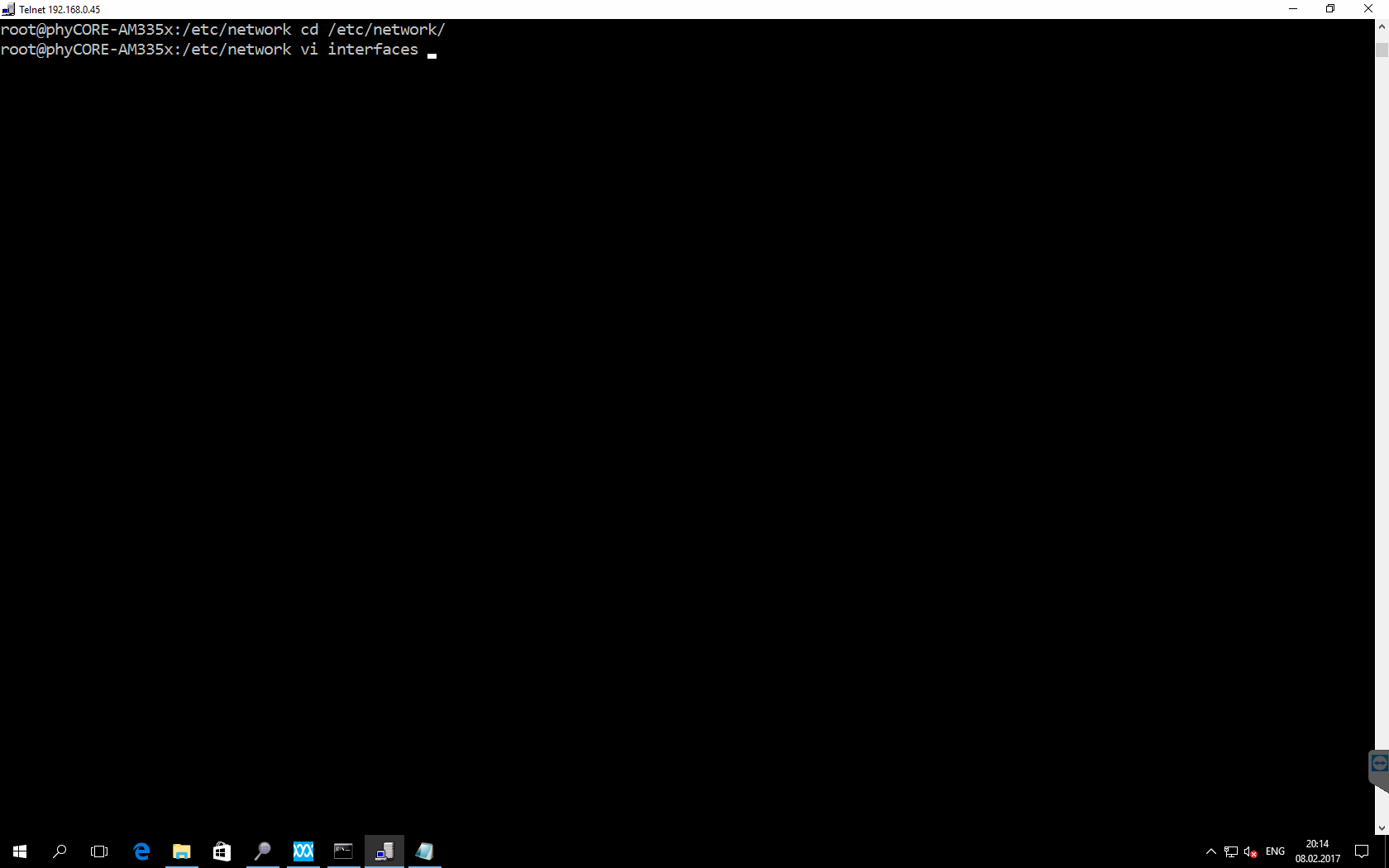
## Настройка сетевых интрефейсов Мцвк

2.2.1. На целевой машине перейти в системный каталог /etc/network/

>cd /etc/network/

2.2.2. На целевой машине ,при помощи тестового редактора vi, отредактировать файл interfaces

>vi interfaces



2.2.3. Файл interfaces должен иметь следующее содержание:

*# /etc/network/interfaces*

*auto lo eth0 eth1 can0*

*# The loopback interface*

*iface lo inet loopback*

*# eth0*

*iface eth0 inet static*

*address 192.168.0.45*

*netmask 255.255.255.0*

*# eth1*

*iface eth1 inet static*

*address 192.168.4.11*

*netmask 255.255.255.0*

*# onboard CAN*

*iface can0 inet static*

*address 127.42.23.180*

*netmask 255.255.255.0*

*pre-up /etc/network/can-pre-up*

где,

192.168.0.45 - Ip адрес адаптера eth0;

255.255.255.0 – маска подсети адаптера eth0

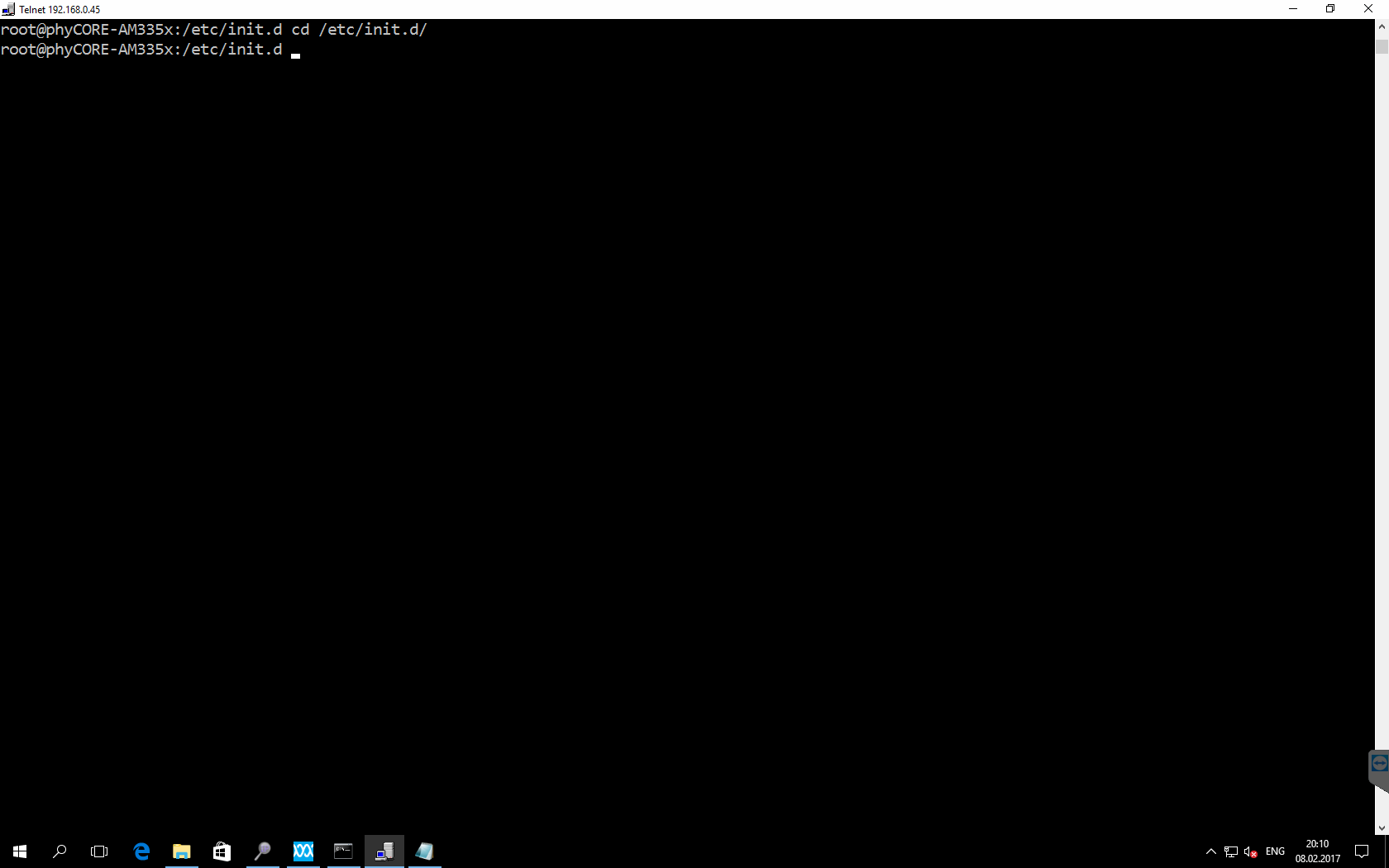
192.168.4.11 - Ip адрес адаптера eth1;

255.255.255.0 – маска подсети адаптера eth1.

## Настройка автоматического запуска Сервера Ввода Вывода, при старте Мцвк

2.3.1. На целевой машине перейти в системный каталог /etc/init.d/

>cd /etc/init.d/



2.3.2. На целевой машине, при помощи тестового редактора vi, создать скрипт rdxserver

>vi rdxserver

2.3.3. Скрипт rdxserver должен иметь следующее содержание:

*#!/bin/sh*

*#*

*# /home/RdxServer*

*#*

*export LD\_LIBRARY\_PATH=/home/:$LD\_LIBRARY\_PATH*

*case "$1" in*

*start)*

*cd /home/*

*./RdxServer*

*;;*

*stop)*

*;;*

*\*)*

*exit 1*

*;;*

*esac*

*exit 0*

2.3.4. На целевой машине перейти в системный каталог /etc/rc.d/

>cd /etc/rc.d/

2.3.5. На целевой машине создать символьную ссылку на скрипт

>ln -s ../init.d/rdxserver S99rdxserver

2.3.5. На целевой машине перейти в системный каталог /etc/

>cd /etc/

2.3.6. На целевой машине, при помощи тестового редактора vi, отредактировать файл profile.environment

>vi profile.environment

2.3.7. В этот файл надо добавить в конец строку:

*export LD\_LIBRARY\_PATH=/home/:$LD\_LIBRARY\_PATH*

# Установка и настройка инструментальной ОС на виртуальную машину

* 1. Установка на VMware.
     1. Создание виртуальной машины.

1. Запустить VMware Player;
2. В меню Player – File выбрать “New Virtual Machine…”;
3. В появившемся окне выбрать “Installer disk image file (iso)” и указать путь к файлу SO-498v5.iso и нажать кнопку Next;
4. В следующем меню выбрать Guest Operating System – Linux, Version – Ubuntu, и нажать Next;
5. В следующем меню указать название виртуальной машины и указать путь к папке, в которой будут храниться данные машины, нажать Next;
6. Maximum disk size указать не менее 25GB, выбрать Store virtual disk as a single file и нажать Next;
7. В следующем окне проверить данные и нажать Finish;
8. В программе VMware появится новая виртуальная машина.
   * 1. Установка ОС на виртуальную машину.
9. Запустить новую виртуальную машину;
10. В появившемся окне выбрать русский язык и нажать «Установить PHYliveDVD»;
11. В следующем меню нажать продолжить, в следующем выбрать «стереть диск и установить PHYliveDVD», в следующем меню нажать «Установить сейчас»;
12. В следующих меню выбрать город, Раскладка клавиатуры – «Ангрийская (США)», ввести имя и задать пароль пользователя;
13. После завершения установки, нажать кнопку Перезагрузить, и после появления белых букв на черном фоне нажать клавишу Enter.
    * 1. Установка VMware Tools на виртуальную машину.
14. В меню Player - Manage выбрать "Install VMware Tools";
15. Скопировать дистрибутив в домашний каталок командой "cp /media/VMware\ Tools/VMwareTools-10.0.0-2977863.tar.gz .";
16. Распаковать дистрибутив командой "tar -zxf VMwareTools-10.0.0-2977863.tar.gz";
17. Сделать распакованный каталог текущим командой "cd vmware-tools-distrib";
18. Запустить установку командой "sudo ./vmware-install.pl", по ходу установки нажимать кнопку Enter ао аремя всех вопросов;
19. Перезапустить машину;
20. Создать общую папку: в меню Player - Manage выбрать "Virtual Machine Settings...". В появившемся окне выбрать вкладку Options. Выбрать пункт "Shared Folders". Справа выбрать пункт "Always enabled". Ниже нажать кнопку Add. В появившемся меню указать имя и путь к общей папке. В виртуальной машине она будет доступна в /mnt/hgfs.
    1. Установка на VirtualBox.
       1. Создание виртуально машины.
21. Запустить Oracle VM VirtualBox, нажать кнопку Создать;
22. В появившемся окне ввести имя машины, тип – Linux, Версия – Ubuntu (64 bit) если машина 32, то Версия - Ubuntu и нажать Next;
23. Указать объем памяти для виртуально машины, желательно не менее 1 ГБ и нажать Next;
24. Выбрать «Создать новый виртуальный диск» и нажать Создать;
25. В появившемся окне выбрать VDI (VirtualBox Disk Image) и нажать Next;
26. Выбрать фиксированный виртуальный диск и нажать Next;
27. Указать путь и имя виртуального жесткого диска и задать объем не менее 22 ГБ, нажать Создать;
28. В Oracle VM VirtualBox появится новая виртуальная машина.
    * 1. Настройка виртуальной машины.
29. Выбрать созданную виртуальную машину и нажать кнопку Настройки;
30. В меню «Носители» выбрать Контроллер: IDE – Пусто, далее в правой части экрана нажать кнопку в виде диска справа от надписи «Привод: вторичный мастер IDE», нажать кнопку «Выбрать образ оптического диска» и указать пусть к файлу SO-498v5.iso;
31. В меню «Общие папки» нажать кнопку добавить общую папку (синяя папка с зеленым плюсом);
32. В появившемся меню выбрать путь к папке, для обмена между виртуальной машиной и реальной, ввести имя папки «vbox», а также установить галочку Авто-подключение, и нажать Ок;
33. Нажать кнопку Ок.
    * 1. Установка ОС на виртуальную машину.
34. Запустить новую виртуальную машину;
35. В появившемся окне выбрать русский язык и нажать «Установить PHYliveDVD»;
36. В следующем меню нажать продолжить, в следующем выбрать «стереть диск и установить PHYliveDVD», в следующем меню нажать «Установить сейчас»;
37. В следующих меню выбрать город, Раскладка клавиатуры – «Ангрийская (США)», ввести имя и задать пароль пользователя;
38. После завершения установки, нажать кнопку Перезагрузить, и после появления белых букв на черном фоне нажать клавишу Enter.
    * 1. Настройка гостевой ОС

* Открыть терминал и выполнить следующие команды: ‘sudo apt-get update’ ‘sudo apt-get install linux-headers-3.2.0-52-generic’
* Затем выполнить команду ‘sudo usermod –aG vboxsf <имя пользователя>’;
* В меню Устройства выбрать «Установить дополнения гостевой ОС»;
* В появившемся окне нажать "Запустить", затем ввести пароль;
* В появившейся консоли после установки когда появится надпись “Press Return to close the Windows” нажать Enter;
* Перезагрузить машину;
* Теперь должны стать доступны все разрешения экрана, а так же общая папка с хост машиной /media/sf\_vbox.

1. Установка Toolchains.
   1. Установить ptxdist-2014.12.0 (необходим для сборки Toolchain).
2. Запустить терминал;
3. Скопировать дистрибутив в домашнюю папку командой "cp /mnt/hgfs..../ptxdist-2014.12.0.tar.bz2 .";
4. Распаковать архив командой "tar -xvjf ptxdist-2014.12.0.tar.bz2";
5. Сделать текущим распакованный каталог командой "cd ptxdist-2014.12.0";
6. Запустить скрипт конфигурации "./configure";
7. Начать процесс сборки командой "make";
8. Установить командой "sudo make install".
   1. Установка OSELAS.Toolchain.
9. Запустить терминал;
10. Скопировать дистрибутив в домашнюю папку командой "cp /mnt/hgfs..../OSELAS.Toolchain-2014.12.1.tar.bz2 .";
11. Распаковать архив командой "tar -xvjf OSELAS.Toolchain-2014.12.1.tar.bz2";
12. Сделать текущим распакованный каталог командой "cd OSELAS.Toolchain-2014.12.1";
13. Ввести команду "ptxdist select ptxconfigs/arm-cortexa8-linux-gnueabihf\_gcc-4.9.2\_glibc-2.20\_binutils-2.24\_kernel-3.16-sanitized.ptxconfig";
14. Ввести команду "ptxdist go".
    1. Установить Toolchain-2014.12.1 используемым для проекта.
15. Сменить текущий каталог командой "cd /opt/PHYTEC\_BSPs/phyCOME-AM335x-PD13.1.2";
16. Удалить ссылку на текущие Toolchain командой "rm selected\_toolchain";
17. Создать ссылку на новые Toolchain командой "ls -d /opt/OSELAS.Toolchain-2014.12.1/arm-cortexa8-linux-gnueabihf/gcc-4.9.2\_glibc-2.20\_binutils-2.24\_kernel-3.16-sanitized/bin/ ./selected\_toolchain";
18. Сменить версию Toolchain в проекте, запустив меню командой "ptxdist platformconfig";
19. В появившемся меню зайти в architecture -> toolchain;
20. В строке "check for specific toolchain vendor" указать "OSELAS.Toolchain-2014.12.1";
21. В строке "check for specific gcc version" указать "4.9.2";
22. Выйти из меню нажав Exit.
23. Установка qt для МЦВК.
24. Скопировать правила сборки qt в проект командой "cp /mnt/hgfs..../rules/\* /opt/PHYTEC\_BSPs/phyCOME-AM335x-PD13.1.2/rules";
25. Скопировать конфиг qt командой "sudo cp -R /mnt/hgfs..../qt5 /usr/local/lib/ptxdist-2013.01.0/config";
26. Сменить текущий каталог командой "cd /opt/PHYTEC\_BSPs/phyCOME-AM335x-PD13.1.2";
27. Запустить меню сборки командой "ptxdist menuconfig";
28. В разделе "Graphics & Multimedia - qt" отметить звездочкой с помощью клавиши пробел "qt5";
29. В меню "qt5 - Modules" отметить QtSerialPort;
30. Выйти из меню и запустить сборку командой "ptxdist go";
31. Установка и настройка qtcreator на инструментальную машину.
    1. Установка qt5 (нужна для сборки qtcreator).
32. Скопировать дистрибутив qt5 в домашнюю папку «cp /opt/PHYTEC\_BSPs/phyCORE-AM335x-PD13.1.2/src/qt-everywhere-opensource-src-5.4.2.tar.xz .»;
33. Распаковать архив «tar –xpJf qt-everywhere-opensource-src-5.4.2.tar.xz»;
34. Сменить текущую директорию «cd qt-everywhere-opensource-src-5.4.2»;
35. Установить необходимые компоненты для установки qt «sudo apt-get install libxcb1 libxcb1-dev libx11-xcb1 libx11-xcb-dev libxcb-keysyms1 libxcb-keysyms1-dev libxcb-image0 libxcb-image0-dev libxcb-shm0 libxcb-shm0-dev libxcb-icccm4 libxcb-icccm4-dev libxcb-sync0 libxcb-sync0-dev libxcb-xfixes0-dev libxrender-dev libxcb-shape0-dev libxcb-randr0-dev libxcb-render-util0 libxcb-render-util0-dev libxcb-glx0-dev»;
36. Запустить скрипт конфигурации "./configure";
37. Начать процесс сборки командой "make";
38. Установить командой "sudo make install".
    1. Смена текущей версии qt по умолчанию.
39. Удалить ссылку на старую версию qt «sudo rm /usr/bin/qmake»;
40. Создать ссылку на новую версию qt «sudo ln -s /usr/local/Qt-5.4.2/bin/qmake /usr/bin/»;
41. Сделать новую ссылку исполняемым файлом «sudo chmod +x /usr/bin/qmake».
    1. Установка qtcreator.
42. Распаковать дистрибутив qtcreator «tar -zxf»;
43. Сменить текущую директорию «cd qt-creator-opensource-src-3.4.1/»;
44. Запустить qmake для проектра «qmake»;
45. Начать процесс сборки командой "make";
46. Установить командой "sudo make install";
47. Запустить qtcreator командой «qtcreator».
    1. Настройка qtcreator.
       1. Создание удаленного устройства.
48. Открыть меню Инструменты - Параметры... - Устройства (слева);
49. Нажать кнопку Добавить...;
50. Выбрать обычное Linux-устройство;
51. Ввести любое название;
52. Ввести IP адрес МЦВК;
53. Ввести имя пользователя – root
54. Выбрать способ авторизации - пароль, пароль пользователя пустой.
    * 1. Настройка компилятора.
55. Открыть меню Инструменты - Параметры... - Сборка и Запуск;
56. Во вкладке Компиляторы добавить – особый;
57. Дать любое название;
58. Задать путь к компилятору /opt/OSELAS.Toolchain-2014.12.1/arm-cortexa8-linux-gnueabihf/gcc-4.9.2-glibc-2.20-binutils-2.24-kernel-3.16-sanitized/bin/arm-cortexa8-linux-gnueabihf-gcc;
59. Задать путь к make /usr/bin/make;
60. Задать ABI - arm linux generic elf 32bit.
    * 1. Настройка qmake.
61. Открыть меню Инструменты - Параметры... - Сборка и Запуск;
62. Во вкладке Qt Versions нажать добавить;
63. путь к qmake - /opt/PHYTEC\_BSPs/phyCORE-AM335x-PD13.1.2/platform-phyCORE-AM335x/sysroot-cross/bin/qt5/qmake.
    * 1. Создание комплекта.
64. Открыть меню Инструменты - Параметры... - Сборка и Запуск;
65. Во вкладке Комплекты нажать добавить;
66. Дать любое название;
67. Задать тип устройства - обычное Linux-устройство;
68. Устройство - созданное ранее;
69. Компилятор - созданный ранее;
70. Профиль qt - созданный ранее;
71. Для создания проектов использовать созданный комплект.

# Bonding

Для объединения нескольких (2-х) физических сетевых интерфейсов в один логический используется модуль bonding.

Подробнее можно прочитать тут http://www.adminia.ru/linux-bonding-obiedinenie-setevyih-interfeysov-v-linux/ .

Модуль используется в режиме active-backup.

Политика активный-резервный. Только один сетевой интерфейс из объединённых будет активным.

Другой интерфейс может стать активным, только в том случае, когда упадёт текущий активный интерфейс.

При такой политике MAC адрес bond интерфейса виден снаружи только через один сетевой порт, во избежание появления проблем с коммутатором.

Эта политика применяется для отказоустойчивости.

Для использования модуля необходимо:

- сам модуль ядра bonding;

- утилита ifenslave для управления модулем.

Модули присутствуют в последней сборке.

Для настройки bonding на МЦВК необходимо следующее:

- в настроечном файле сети (/etc/network/interfaces) удалить настройки для физических интерфейсов eth0 и eth1.

- в настроечном файле сети добавить интерфейс bond0, который и будет логическим узлом.

Для него необходимо прописать скрипты инициализации pre-up и post-up:

auto bond0

iface bond0 inet static

address x.x.x.x

netmask x.x.x.x

pre-up /etc/network/bond-pre-up

post-up /etc/network/bond-post-up

скрипт pre-up выполняется перед поднятием интерфейса bond, скрипт post-up после поднятия.

- создать скрипт pre-up «/etc/network/bond-pre-up», добавить ему атрибут запускаемости (chmod +x).

В скрипте необходимо задавать режим (режим задается до поднятия интерфейса):

echo active-backup > /sys/class/net/bond0/bonding/mode #задание режима работы bond

- создать скрипт post -up «/etc/network/bond-post-up», добавить ему атрибут запускаемости (chmod +x).

В скрипте необходимо настроить физические интерфейсы, с которыми будет работать bond0 и интервал, с которым будет проверяться состояние линии

на наличие отказов:

ifenslave bond0 eth0 eth1 #задание физических интерфейсов, с которыми будет работать bond

echo 1000 > /sys/class/net/bond0/bonding/miimon #задание времени в миллисекундах проверки состояния линии

После перезапуска системы виртуальный интерфейс bond0 должен появиться и корректно работать.

Чтобы задать bond0 интерфейсом по умолчанию для отправки в другие подсети, нужно добавить в скрипт post-up следующую строку:

route add default dev bond0

Для ручного выбора физического интерфейса, через который должен работать bond используется команда:

ifenslave -c bond0 <физический интерфейс eth0 или et

# Yocto

Создание образа системы

(https://www.phytec.de/documents/l-813e-5-yocto-reference-manual/#Get\_phyLinux)

1. Создать диск 70Гб.

2. Имя диска - yocto.

3. Запустить Терминал.

4. cd /media/пользователь/yocto

5. sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential chrpath socat libsdl1.2-dev xterm

6. wget ftp://ftp.phytec.de/pub/Software/Linux/Yocto/Tools/phyLinux

7. chmod +x phyLinux

8. ./phyLinux init

9. Выбрать 1, 17, 7

10. В файле build/conf/local.conf убрать # в строках

#SYSTEMD\_AUTO\_ENABLE\_pn-phytec-qtdemo = "disable"

#DL\_DIR ?= "<your\_directory>/yocto\_downloads"

#SSTATE\_DIR ?= "<your\_directory>/yocto\_sstate"

11. source sources/poky/oe-init-build-env

12. bitbake -c fetchall phytec-qt5demo-image

13. ( du -sh downloads/ )

14. bitbake phytec-qt5demo-image

Создание SDK

(https://www.phytec.de/documents/l-813e-5-yocto-reference-manual/#Using\_the\_SDK)

1. ( cd deploy/images/phycore-am335x-1/ )

2. ( source sources/poky/oe-init-build-env )

3. Создание SDK установщика

bitbake -c populate\_sdk phytec-qt5demo-image

4. Установка SDK

cd deploy/sdk/

./phytec-yogurt-glibc-i686-phytec-qt5demo-image-cortexa8hf-neon-toolchain-AM335x-PD16.2.0.sh

SDK has been successfully set up and is ready to be used.

Each time you wish to use the SDK in a new shell session, you need to source the environment setup script e.g.

$ . /opt/phytec-yogurt/AM335x-PD16.2.0/environment-setup-cortexa8hf-neon-phytec-linux-gnueabi

Настройка Qt Creator’а

(https://wiki.phytec.com/display/public/PRODUCTINFO/How+to+Build+Qt+Application+with+PHYTEC+Yocto+BSPs)

1. . /opt/phytec-yogurt/AM335x-PD16.2.0/environment-setup-cortexa8hf-neon-phytec-linux-gnueabi

2. qtcreator

3. Открыть меню Инструменты - Параметры... - Устройства (слева);

4. Нажать кнопку Добавить...;

5. Выбрать обычное Linux-устройство;

6. Ввести название yocto;

7. Ввести IP адрес Мцвк;

8. Ввести имя пользователя – root

9. Выбрать способ авторизации - пароль, пароль пользователя пустой.

10. Открыть меню Инструменты - Параметры... - Сборка и Запуск;

11. Во вкладке Qt Versions нажать Добавить;

12. Путь к qmake - /opt/phytec-yogurt/AM335x-PD16.2.0/sysroots/i686-yogurtsdk-linux/usr/bin/qt5/qmake.

13. Во вкладке Компиляторы добавить – особый;

14. Дать название yocto;

15. Задать путь к компилятору /opt/phytec-yogurt/AM335x-PD16.2.0/sysroots/i686-yogurtsdk-linux/usr/bin/arm-phytec-linux-gnueabi/arm-phytec-linux-gnueabi-gcc;

16. Задать путь к make /usr/bin/make;

17. Задать ABI - arm linux generic elf 32bit.

18. Во вкладке Debbugers добавить – особый;

19. Дать название yocto;

20. Задать путь /opt/phytec-yogurt/AM335x-PD16.2.0/sysroots/i686-yogurtsdk-linux/usr/bin/arm-phytec-linux-gnueabi/arm-phytec-linux-gnueabi-gdb;

21. Во вкладке Комплекты добавить – особый;

22. Дать название yocto;

23. Задать тип устройства - обычное Linux-устройство;

24. Устройство — yocto;

25. Sysroot - /opt/phytec-yogurt/AM335x-PD16.2.0/sysroots/cortexa8hf-neon-phytec-linux-gnueabi

26. Компилятор — yocto;

27. Debbuger - yocto;

28. Профиль qt — Qt 5.6.2 (qt5);

29. Для того чтобы удаленная отладка работала, необходимо на мцвк, в файл /etc/ssh/sshd\_config, добавить линию:

Ciphers +aes128-cbc

30. Для создания проектов использовать созданный комплект. Собрать сервер ввода-вывода.

Qtcreator запускать из консоли.

$ . /opt/phytec-yogurt/AM335x-PD16.2.0/environment-setup-cortexa8hf-neon-phytec-linux-gnueabi

$ qtcreator

Запись образа на Мцвк

1. cd deploy/images/phycore-am335x-1/

2. Вставить SD карту в Usb разъем. Подключить ее к виртуальной машине: "Устройства -> USB -> название\_SD\_карты".

3. Записать на нее образ:

sudo dd if=phytec-qt5demo-image-phycore-am335x-1.sdcard of=/dev/sdc bs=1MB conv=fsync

4. Вставить SD карту в Мцвк. Джампером задать загрузку с SD карты.

5. Подключиться к Мцвк через консольный порт. Загрузить Мцвк. Если Линукс загрузился то значит, что образ создан и записан нормально.

6. Перезагрузить Мцвк и зайти в загрузчик. Для этого в момент загрузки нажать любую клавишу. Приглашение загрузчика выглядит так:

barebox@Phytec phyCORE AM335x:/

7. Прошить Nand образом с SD карты. Для этого выполнить следующую последовательность команд:

barebox@Phytec phyCORE AM335x:/ barebox\_update -t MLO.nand /boot/MLO

barebox@Phytec phyCORE AM335x:/ barebox\_update -t nand /boot/barebox.bin

barebox@Phytec phyCORE AM335x:/ erase /dev/nand0.bareboxenv.bb

barebox@Phytec phyCORE AM335x:/ reset

barebox@Phytec phyCORE AM335x:/ ubiformat /dev/nand0.root

barebox@Phytec phyCORE AM335x:/ ubiattach /dev/nand0.root

barebox@Phytec phyCORE AM335x:/ ubimkvol -t static /dev/nand0.root.ubi kernel 8M

barebox@Phytec phyCORE AM335x:/ ubimkvol -t static /dev/nand0.root.ubi oftree 1M

barebox@Phytec phyCORE AM335x:/ ubimkvol -t dynamic /dev/nand0.root.ubi root 0

barebox@Phytec phyCORE AM335x:/ ubiupdatevol /dev/nand0.root.ubi.kernel /boot/linuximage

barebox@Phytec phyCORE AM335x:/ ubiupdatevol /dev/nand0.root.ubi.oftree /boot/oftree

barebox@Phytec phyCORE AM335x:/ cp /boot/phytec-qt5demo-image-phycore-am335x-1.ubifs /dev/nand0.root.ubi.root

8. Убрать SD карту из Мцвк. Джампером задать загрузку с Nand. Перегрузить Мцвк. Если Линукс загрузился то значит, что загрузчик записан нормально.

Настройка Мцвк

(https://www.phytec.de/documents/l-813e-5-yocto-reference-manual/#How\_to\_change\_the\_Network\_Configuration)

1. Настроить нужные IP адреса. Для этого отредактировать файлы:

/lib/systemd/network/10-eth0.network, /lib/systemd/network/10-eth1.network

2. Перегрузить Мцвк. Теперь можно пользоваться программами putty и WinSCP.

3. Переписать на Мцвк библиотеку Qt5SerialPort. Из каталога SDK:

/opt/phytec-yogurt/AM335x-PD16.2.0/sysroots/cortexa8hf-neon-phytec-linux-gnueabi/usr/lib/l.

В каталог Мцвк: /usr/lib/

4. Переписать на Мцвк, в каталог /home/root/, модули сервера ввода-вывода:

RdxServer – исполняемый модуль Сервера Ввода Вывода.

RdxServer.ini – файл содержащий параметры запуска Сервера Ввода Вывода.

libScdMdx.so, libScdMdx.so.1, libScdMdx.so.1.0, libScdMdx.so.1.0.0 – библиотека Сервера Ввода Вывода.

libRdxMdx.so, libRdxMdx.so.1, libRdxMdx.so.1.0, libRdxMdx.so.1.0.0 – библиотека протокола Mdx.

libRdxMonitor.so, libRdxMonitor.so.1, libRdxMonitor.so.1.0, libRdxMonitor.so.1.0.0 – библиотека протокола Monitor.

libRdxBridge.so, libRdxBridge.so.1, libRdxBridge.so.1.0, libRdxBridge.so.1.0.0 – библиотека протокола Bridge.

libRdxIec101.so, libRdxIec101.so.1, libRdxIec101.so.1.0, libRdxIec101.so.1.0.0 – библиотека протокола Iec101.

libRdxIec104.so, libRdxIec104.so.1, libRdxIec104.so.1.0, libRdxIec104.so.1.0.0 – библиотека протокола Iec104.

libRdxModbus.so, libRdxModbus.so.1, libRdxModbus.so.1.0, libRdxModbus.so.1.0.0 – библиотека протокола Modbus.

libRdxCan.so, libRdxCan.so.1, libRdxCan.so.1.0, libRdxCan.so.1.0.0 – библиотека протокола Can.

libRdxSnmp.so, libRdxSnmp.so.1, libRdxSnmp.so.1.0, libRdxSnmp.so.1.0.0 – библиотека протокола Snmp.

5. Создать символьный ссылки на библиотечные файлы:

>ln -s /home/root/\*dx\* /usr/lib/

6. Создать каталог для лог-файлов:

mkdir logs

7. Создать каталог для файлов конфигурации:

mkdir configurations

8. Переписать на Мцвк, в каталог /home/root/configurations/, конфигурационный файл.

9. Настроить автоматический запуск сервера ввода-вывода, при старте Мцвк.

(https://wiki.dieg.info/update-rc.d)

1. Перейти в системный каталог /etc/init.d/

>cd /etc/init.d/

2. При помощи тестового редактора vi создать файл rdxserverscript1

>vi rdxserverscript1

3. Файл должен иметь следующее содержание:

#!/bin/sh

case "$1" in

start)

cd /home/root/

/home/root/rdxserverscript2 &

;;

stop)

;;

\*)

exit 1

esac

4. Зарегистрировать скрипт rdxserverscript1:

>update-rc.d rdxserverscript1 defaults 99

5. Перейти в каталог /home/root/

>cd /home/root/

6. При помощи тестового редактора vi создать скрипт rdxserverscript2

>vi rdxserverscript2

7. Скрипт должен иметь следующее содержание:

while (true)

do

/home/root/RdxServer

case $? in

0)

exit 0

;;

2)

cp /tmp/kotmi\_rdx/\*.bin /home/root/configurations/

cp /tmp/kotmi\_rdx/\*.xml /home/root/configurations/

cp /tmp/kotmi\_rdx/lib\* /home/root/

cp /tmp/kotmi\_rdx/RdxServer /home/root/

;;

\*)

;;

esac

sleep 1

done

8. Сделать запускаемыми файлы rdxserverscript1, rdxserverscript2, RdxServer, libScd\*, libRdx\*

>chmod +x rdxserverscript1

...

10. Установить время

>date -s "YYMMDDHHMM"

("1802121406" = Mon Feb 12 14:06 : 00 UTC 2018)

11. установить микросхему часов

>hwclock -w -u