

ПРОМЫШЛЕННЫЙ КОНТРОЛЛЕР
АГАВА ПК-40

РУКОВОДСТВО ПРОГРАММИСТА

АГСФ.421445.006 РП

Редакция 1.4

Екатеринбург

2018

Оглавление

1. Введение	5
1.1. Используемые термины и сокращения:	5
2. Назначение.....	6
3. Настройка прибора.....	7
3.1. Установка времени и даты	7
3.2. Доступ к файлам контроллера	8
3.2.1. Настройка сетевого доступа	8
4. Аппаратные ресурсы контроллера.....	10
4.1. Звуковой извещатель.....	10
4.2. Светодиоды.....	10
4.3. EEPROM	10
4.4. RTC	10
4.5. NAND flash (для исполнения с NAND flash)	11
4.6. Подсветка TFT	11
4.7. Touchscreen.....	11
4.8. Последовательные порты.....	11
4.9. Ethernet.	12
4.10. USB OTG	12
4.11. Фреймбуфер и виртуальная консоль.	12
5. Настройка виртуальной машины	13
5.1. Вход в ВМ	13
5.2. Настройка сетевого взаимодействия	13
6. Разработка пользовательского прикладного программного обеспечения с использованием компиляторов C/C++	14
6.1. Разработка простого приложения с использованием системы сборки make в среде Linux	14
6.1.1. Создание Makefile	14
6.1.2. Создание текста программы.	14
6.1.3. Сборка приложения	14
6.2. Разработка простого приложения с использованием средств Visual Studio и VisualGDB в среде Windows	15
6.2.1. Создание проекта VisualGDB	15
6.2.2. Наполнение проекта кодом	24
6.2.3. Сборка приложения	26
6.2.4. Отладка приложения в среде Visual Studio.....	26
7. Разработка графического приложения с использованием Qt.....	29
7.1. Создание проекта в Qt Creator	29
7.2. Дополнительная настройка проекта	32
7.2.1. Размещение приложения на целевом устройстве.....	32
7.2.2. Создание целевого устройства	32
7.2.3. Смена целевого устройства в комплекте сборки	34

7.3. Сборка и отладка приложения.....	34
8. Обновление базовых программных компонентов ПК	35
8.1. Обновление компонентов загрузчика.....	35
8.2. Обновление компонентов ОС Linux.....	36
8.3. Обновление корневой файловой системы	36
8.4. Обновление прикладного (пользовательского) программного обеспечения	37
8.4.1. Корневая файловая система версий до 2018.3	37
8.4.2. Корневая файловая система версий 2018.3 и позднее	37
9. Список рекомендуемой литературы	39
Приложение 1. Описание реализации протокола Modbus для субмодулей	40
Регистры, общие для всех модулей	40
Регистры субмодуля R – Релейные выходы	40
Регистры субмодуля DI – Дискретные входы.....	40
Регистры субмодуля AI – Аналоговые входы	41
Регистры субмодуля DO – Дискретные выходы – открытый коллектор.....	42
Регистры субмодуля SIM – Симисторные выходы.....	43
Регистры субмодуля AIO – Аналоговые входы и выходы	43
Регистры субмодуля TMP – измерение температуры.....	45

1. Введение

Руководство по эксплуатации содержит сведения, необходимые для обеспечения правильной эксплуатации и полного использования технических возможностей промышленного контроллера АГАВА ПК-40, далее по тексту ПРИБОР, ПЛК или КОНТРОЛЛЕР.

Разработка приложений для контроллеров серии АГАВА ПК предполагает использование SDK в виде виртуальной машины с установленным программным обеспечением.

1.1. Используемые термины и сокращения:

SDK – Software development kit – комплект средств разработки приложений

SSH – Secure Shell – протокол защищенного подключения

ВМ – виртуальная машина

ПЛК – программируемый логический контроллер (промышленный контроллер);

ОС – операционная система;

ПО – программное обеспечение;

ОЗУ – оперативное запоминающее устройство;

ФС – файловая система.

2. Назначение

Промышленный контроллер АГАВА ПК-40 предназначен для создания систем автоматизированного управления технологическим оборудованием в различных областях промышленности, жилищно-коммунального и сельского хозяйства.

Логика работы контроллера определяется потребителем в процессе программирования контроллера. Программирование осуществляется с помощью различных средств разработки с использованием компиляторов C/C++.

Загрузка проекта в прибор и его отладка производятся через интерфейс Ethernet.

3. Настройка прибора

На уровне операционной системы прибор имеет файловые ресурсы и системную консоль. В файлах содержится необходимая информация для работы ОС и пользовательского прикладного программного обеспечение. Консоль служит для интерактивного взаимодействия с ОС (выполнения команд ОС и т.п.).

Файловая система состоит из системной ФС и монтируемой ФС, которая доступна как на чтение, так и для записи¹, имеющая следующие точки монтирования:

- /run/media/mmcblk* для SD-карты;
- /run/media/sda* для и USB-флеш;

Системная консоль находится на последовательном порте RS-232. Параметры терминала для консоли следующие:

- Скорость (бит/с): 115200
- Биты данных: 8
- Четность: Нет
- Стартовые биты: 1
- Управление потоком: Нет

Соединение контроллера с персональным компьютером по интерфейсу RS-232 производится нуль-модемным кабелем.

При загруженной ОС, подключенной и настроенной сети доступ к системной консоли можно получить по SSH.

Права администратора для входа по SSH:

- Логин: root
- Пароль отсутствует

3.1. Установка времени и даты

Для установки времени и даты следует воспользоваться командой:

```
date MMDDhhmmYYYY
```

где

- MM – месяц (1-12);
- DD – число (1-31);
- hh – часы (0-23);

¹ SD-карта может быть заблокирована на запись при установке на ней переключателя в соответствующее положение.

- mm – минуты (0-59);
- YYYY – год.

Для сохранения установленного времени и даты в часы реального времени воспользуйтесь командой:

```
hwclock -w
```

При подключении контроллера к сети Ethernet и наличии выхода в Интернет, происходит синхронизация времени с серверами точного времени.

Часовой пояс устанавливается в файле /etc/profile путем задания переменной окружения TZ. Например, export TZ="STD-5" (для Екатеринбурга).

3.2. Доступ к файлам контроллера

Доступ к файлам и ресурсам контроллера при загруженной ОС (в т.ч. запущенной системы исполнения CODESYS) можно получить следующими способами:

- через системную консоль на порте RS-232;
- через системную консоль SSH-сервиса;
- через sftp-сервер;

Для доступа к файлам контроллера через sftp-сервер следует пользоваться Unix-совместимым sftp-клиентом. Под ОС Windows это может быть, например, WinSCP, Total Commander и т.п.

Для того чтобы иметь доступ к SD-карте через USB-интерфейс, необходимо загрузить модуль *g_mass_storage.ko*. При подключении компьютера к разъему USB-OTG прибора, на компьютере появится съемный диск с содержимым SD-карты. Также загрузка данного модуля необходима для подключения к разъему USB-OTG других устройств (USB-флеш и т.п.).

```
modprobe g_mass_storage file=/dev/mmcblk0
```

3.2.1. Настройка сетевого доступа

По умолчанию Прибор настроен на получение сетевого адреса от DHCP-сервера.

Просмотреть IP-адрес и другую сетевую конфигурацию для всех интерфейсов Ethernet можно из консоли, набрав команду:

```
ifconfig
```

```
root@agava6432_40:~# ifconfig
eth0      Link encap:Ethernet  HWaddr A8:1B:6A:48:71:93
          inet  addr:192.168.10.215  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::aa1b:6aff:fe48:7193%132688/64 Scope:Link
                  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
RX packets:865 errors:0 dropped:16 overruns:0 frame:0
TX packets:98 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:75565 (73.7 KiB) TX bytes:13702 (13.3 KiB)
Interrupt:173

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1%132688/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
        RX packets:6 errors:0 dropped:0 overruns:0 frame:0
        TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:540 (540.0 B) TX bytes:540 (540.0 B)

usb0    Link encap:Ethernet HWaddr 46:10:3A:B3:AF:D9
        inet addr:192.168.7.1 Bcast:192.168.7.3 Mask:255.255.255.252
          UP BROADCAST MULTICAST MTU:1500 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

root@agava6432_40:~#
```

Задать статический IP-адрес интерфеса eth0 можно в файле /etc/systemd/network/10-eth.network, например:

```
[Network]
DHCP=no
Address=192.168.10.100/24
Gateway=192.168.10.10
```

Для редактирования файла можно воспользоваться встроенным фаловым менеджером mc, либо установить подключение к Прибору через USB интерфейс (адрес ПЛК - 192.168.7.1) по протоколу SFTP и изменить файл имеющимся в ОС текстовым редактором.

4. Аппаратные ресурсы контроллера

4.1. Звуковой извещатель

Драйвер регистрируется как устройство `/sys/devices/beeper`. Для подачи звукового сигнала введите в терминале команду:

```
echo -en "\07" > /dev/tty5
```

4.2. Светодиоды

Драйвер регистрируется как устройство `/sys/class/leds`. В каталоге присутствуют устройства для управления соответствующими светодиодами:

```
agava:work:green - зеленый светодиод "Работа";  
agava:fault:red - красный светодиод "Авария";  
agava:programm:green - зеленый светодиод "Программа";  
agava:work_usr:red - красный пользовательский светодиод " Работа " (по-умолчанию настроен на событие mmc0 - управление от драйвера SD-карты);  
agava: fault_usr:green - зеленый пользовательский светодиод " Авария ";  
agava:prog_usr:red - красный пользовательский светодиод " Программа " (по-умолчанию настроен на событие usb-host - управление от драйвера хоста USB);
```

Для включения, например, светодиода "Программа", из терминала нужно подать команду

```
echo 1 > /sys/class/leds/agava:programm:green/brightness.
```

Для выключения:

```
echo 0 > /sys/class/leds/agava:programm:green/brightness.
```

4.3. EEPROM

Регистрируется как устройство `/sys/bus/i2c/devices/0-0050/fram`.

Для проверки записи необходимо выполнить команды:

```
cd /sys/bus/i2c/devices/0-0050  
echo "TEST" > eeprom  
head -c 300 eeprom | hexdump -C
```

Убедиться, что записалась строка "TEST".

4.4. RTC.

Установить системное время командой:

```
date -s "2016-06-28 17:47:00"
```

Записать время в RTC командой:

```
hwclock -w
```

Выключить контроллер. Через 1 минуту включить контроллер, командой date убедиться, что установлено правильное системное время.

4.5. NAND flash (для исполнения с NAND flash)

Для проверки функционирования NAND flash нужно выполнить команду:

```
nandtest -p 1 /dev/mtd9
```

Убедиться, что тест прошел без ошибок (Finished pass 1 successfully).

4.6. Подсветка TFT

Для уменьшения уровня подсветки экрана выполните команду:

```
echo 1 > /sys/devices/platform/backlight/backlight/brightness
```

Для уменьшения уровня подсветки экрана выполните команду:

```
echo 8 > /sys/devices/platform/backlight/backlight/brightness
```

4.7. Touchscreen

Для проверки тачскрина, подключенном к встроенному в процессор контроллеру, выполнить команду:

```
evtest /dev/input/touchscreen0
```

или

```
evtest /dev/input/by-path/platform-TI-am335x-tsc-event
```

Нажимая на экран, можно убедиться, что регистрируются соответствующие события.

Для калибровки тачскрина при использовании графической системы Weston, необходимо удалить файлы, выполнив следующие команды:

```
rm /etc/udev/rules.d/ws-calibrate.rules  
rm /run/media/mmcblk0p1/ws-calibrate.rules  
reboot
```

4.8. Последовательные порты

Последовательные порты регистрируются в системе как устройства /dev/ttyS0 - /dev/ttyS3.

Устройство /dev/ttyS0 - локальная шина для связи с модулями.

Устройство /dev/ttyS1 - RS-232 (системная консоль).

Устройство /dev/ttyS2 - порт RS-485

Устройство /dev/ttys3 - порт RS-485

Из терминала системной консоли можно выполнить команду установки скорости порта 1200 бит/сек:

```
stty -F /dev/ttys0 1200
```

или 230400 бит/сек:

```
stty -F /dev/ttys0 230400
```

Для включения и проверки аппаратного контроля управления потоком RTS/CTS подайте команду:

```
stty crtscts -F /dev/ttys0
```

Выключение аппаратного контроля:

```
stty -crtsts -F /dev/ttys0
```

4.9. Ethernet.

Работа порта Ethernet поддерживается операционной системой, пользователю достаточно использовать интерфейс eth0 или другой, если была произведена смена имени интерфейса.

4.10.USB OTG

Порт USB OTG функционирует при загруженном модуле гаджета g_mass_storage.ko. Для загрузки модуля необходимо выполнить команду:

```
modprobe g_mass_storage file=/dev/mmcblk0
```

После чего подключить контроллер кабелем miniUSB/USB к компьютеру, убедиться, что на компьютере появится новый диск - SD-карта, установленная в контроллере.

Для проверки режима USB host необходимо подключить к контроллеру кабель OTG, в который установить USB-флеш. Убедиться, что флеш определяется контроллером.

4.11.Фреймбуфер и виртуальная консоль.

Фреймбуфер регистрируется как устройство /dev/fb0. Текст на экран выводится через виртуальную консоль /dev/tty0.

Для проверки работы фреймбуфера и виртуальной консоли выполнить команду:

```
echo '12345' > /dev/tty0
```

При этом на ЖКИ-экране контроллера должен появиться текст 12345.

5. Настройка виртуальной машины

5.1. Вход в ВМ

Для входа в ВМ используйте следующие реквизиты:

Пользователь: user

Пароль: user

5.2. Настройка сетевого взаимодействия

Для подключения ВМ к локальной вычислительной сети как полноценного ПК может потребоваться изменение IP-адреса.

Для разового задания IP-адреса машины, действующего до перезагрузки, воспользуйтесь командой (потребуется ввод пароля пользователя):

```
sudo ifconfig eth0 <address> netmask 255.255.255.0
```

Для задания постоянного адреса отредактируйте файл /etc/network/interfaces:

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.10.42
    netmask 255.255.255.0
    gateway 192.168.10.10
    dns-nameservers 192.168.10.10
```

6. Разработка пользовательского прикладного программного обеспечения с использованием компиляторов C/C++

Данный раздел предназначен для специалистов, обладающих знаниями языка C/C++, а так же опытом сборки приложений с использованием системы make.

Детальное описание порядка разработки приложений на языках C/C++ приведено в соответствующей литературе (см. раздел 9). Ниже описываются действия, специфичные для контроллеров АГАВА на базе ОС Linux RT 4.4.

Для начала работы необходимо установить виртуальную машину VMWare Player с ОС UBUNTU 14.04, в которой уже установлено все необходимое программное обеспечение:

- Кросскомпилятор C/C++
- SFTP клиент
- SSH Server
- Qt Creator

6.1. Разработка простого приложения с использованием системы сборки make в среде Linux

Разработаем небольшое приложение, использующее аппаратное обеспечение контроллера АГАВА 6432.40.

Приложение будет поочередно включать и выключать все имеющиеся светодиоды, а так же светодиоды подсветки дисплея.

Исходные тексты приложения доступны в виртуальной машине по адресу:
File://home/user/applications/LEDTest

6.1.1. Создание Makefile

Воспользуемся готовым основным makefile, содержащимся в файле "Makefile" в каталоге LEDTest. Описание принципов создания makefile приведено в соответствующей литературе (см. раздел 9). Основной makefile кроме прочих инструкций содержит установку имени файла выходного приложения, а так же имена файлов, содержащих исходные тексты.

Основной makefile дополняют два дополнительных: debug.mak и release.mak для отладочной и релизной версий программы соответственно. В дополнительных makefile указываются пути до кросскомпилиатора, ключи компиляции и линковки.

6.1.2. Создание текста программы.

Напишем основной код приложения на языке C++, и поместим его в файл «LEDTest40.cpp»

6.1.3. Сборка приложения

Для сборки приложения запустим команду make в директории с исходными текстами:

```
user@ubuntu:~/applications/LEDTest40$ make
```

По умолчанию сборка будет выполнена в варианте Debug, то есть с дополнительной отладочной информацией.

При сборке приложения в консоль выводится информация о процессе сборки, ошибках и т.д.:

```
user@ubuntu:~/applications/LEDTest40$ make
/home/user/ti-processor-sdk-linux-rt-am335x-evm-03.03.00.04/linux-
devkit/sysroots/x86_64-arago-linux/usr/bin/arm-linux-gnueabihf-g++ -std=c++11 -
fexceptions -ggdb -ffunction-sections -O0 -DDEBUG=1 -c LEDTest40.cpp -o
Debug/LEDTest40.o -MD -MF Debug/LEDTest40.dep
/home/user/ti-processor-sdk-linux-rt-am335x-evm-03.03.00.04/linux-
devkit/sysroots/x86_64-arago-linux/usr/bin/arm-linux-gnueabihf-g++ -o Debug/LEDTest40 -
-Wl,-gc-sections -Wl,--start-group Debug/LEDTest40.o -Wl,--rpath='$ORIGIN' -Wl,--
end-group
```

Для сборки релизного варианта приложения запустим make с указанием варианта сборки:

```
CONFIG=RELEASE make
```

Сборка приложения в этом варианте будет произведена аналогично предыдущему.

6.2. Разработка простого приложения с использованием средств Visual Studio и VisualGDB в среде Windows

Использование VisualGDB позволит значительно повысить удобство и скорость разработки ПО для контроллеров с ОС Linux, так как редактирование текстов и отладка ведется в среде Visual Studio.

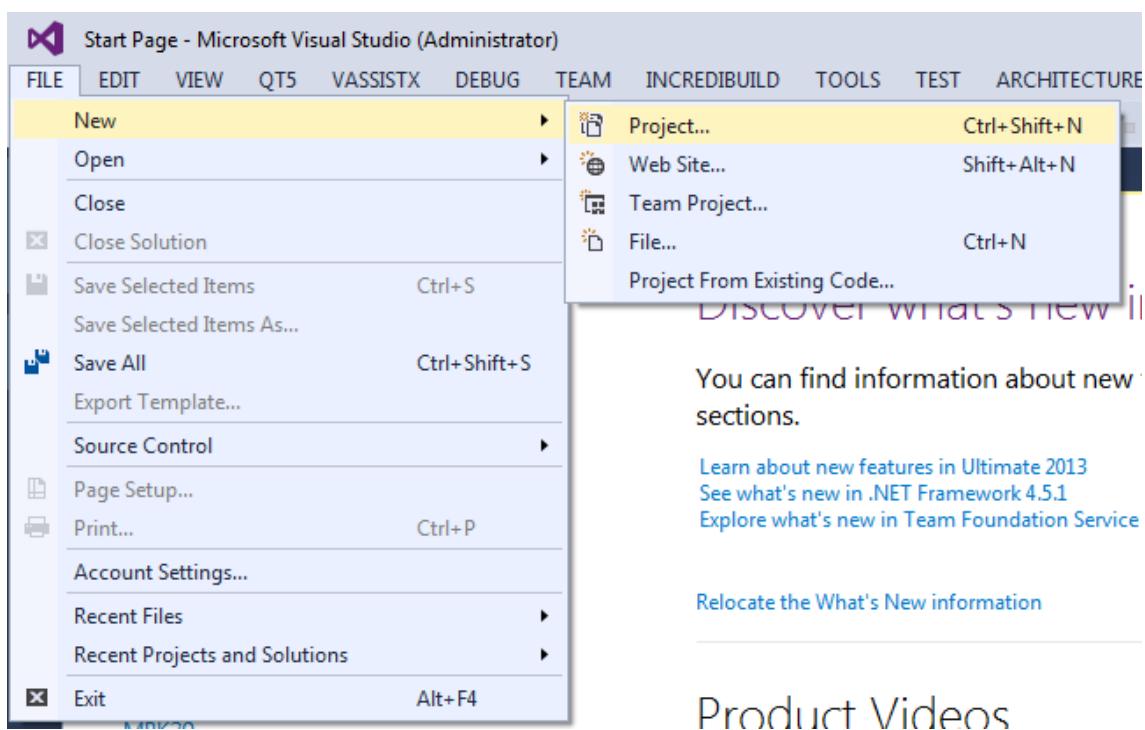
Для начала работы необходимо установить на ПК с Windows программное обеспечение дополнительно к тому, что указано в п. 0:

- VisualGDB
- Visual Studio 2013

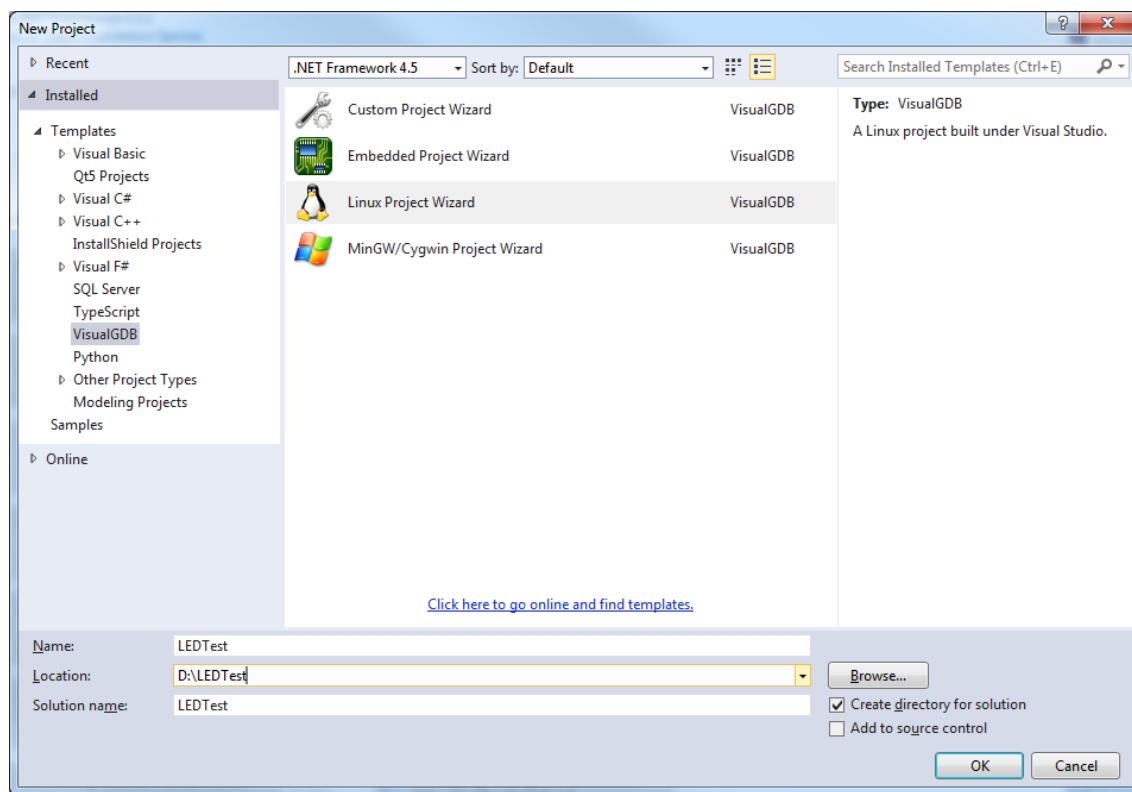
6.2.1. Создание проекта VisualGDB

После установки необходимого ПО создадим проект для VisualGDB, который будет хранить создаваемое нами приложение.

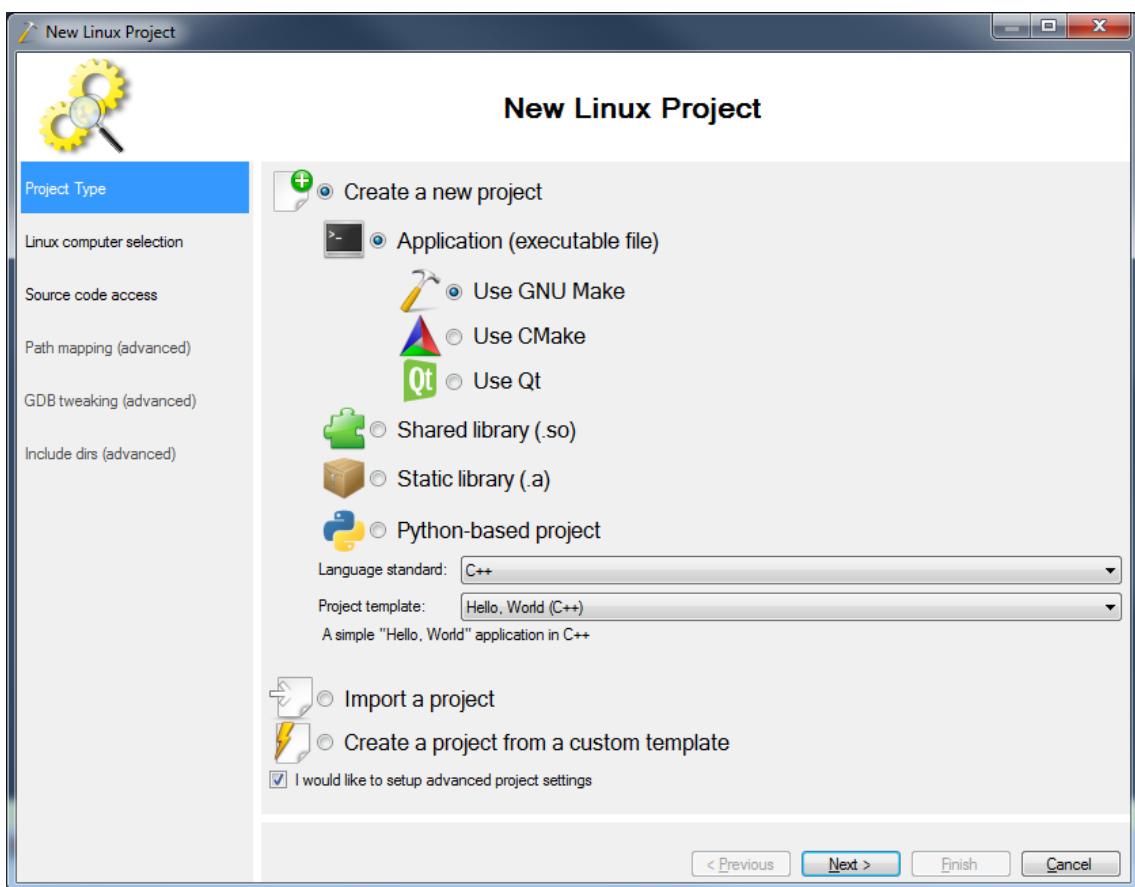
Открываем Visual Studio. И идем в меню File->New->Project



Выбираем [VisualGDB](#) -> Linux Project Wizard. Указываем папку для проекта и имя проекта.

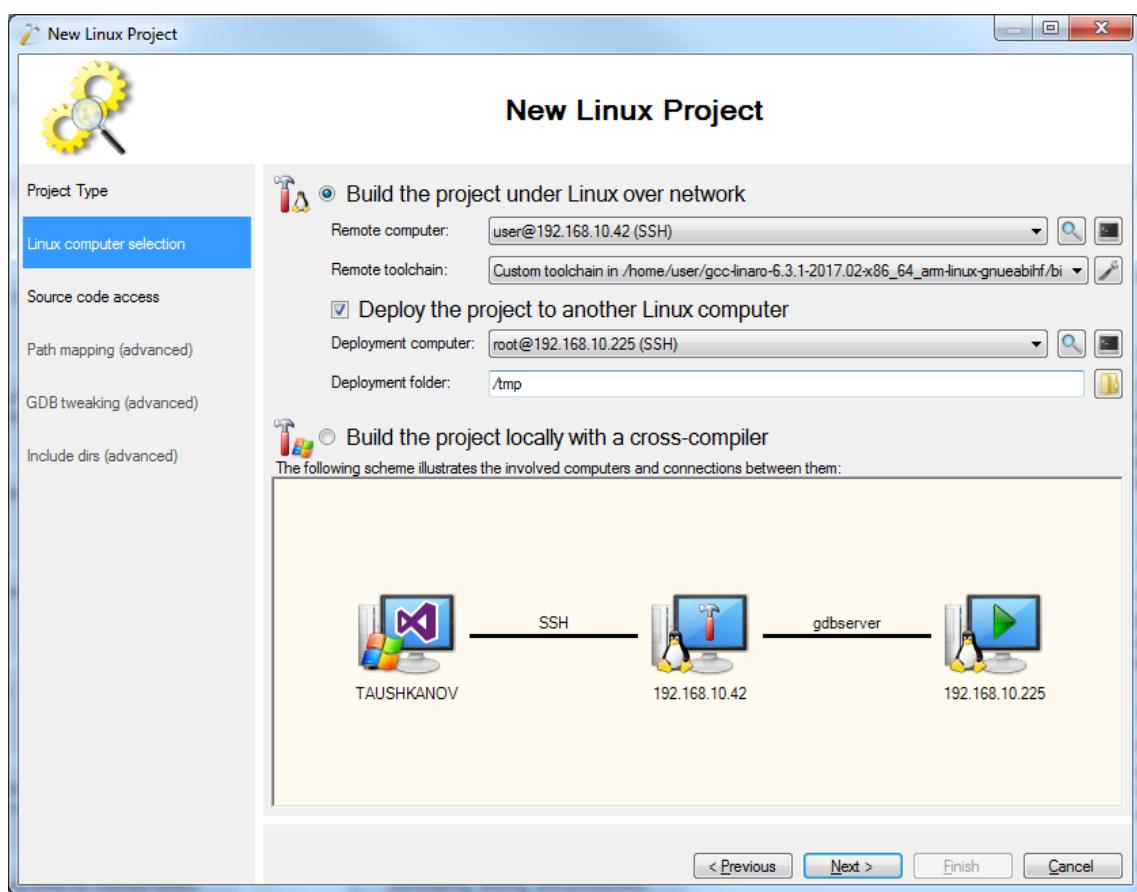


На данном этапе выбранные параметры оставляем без изменения:

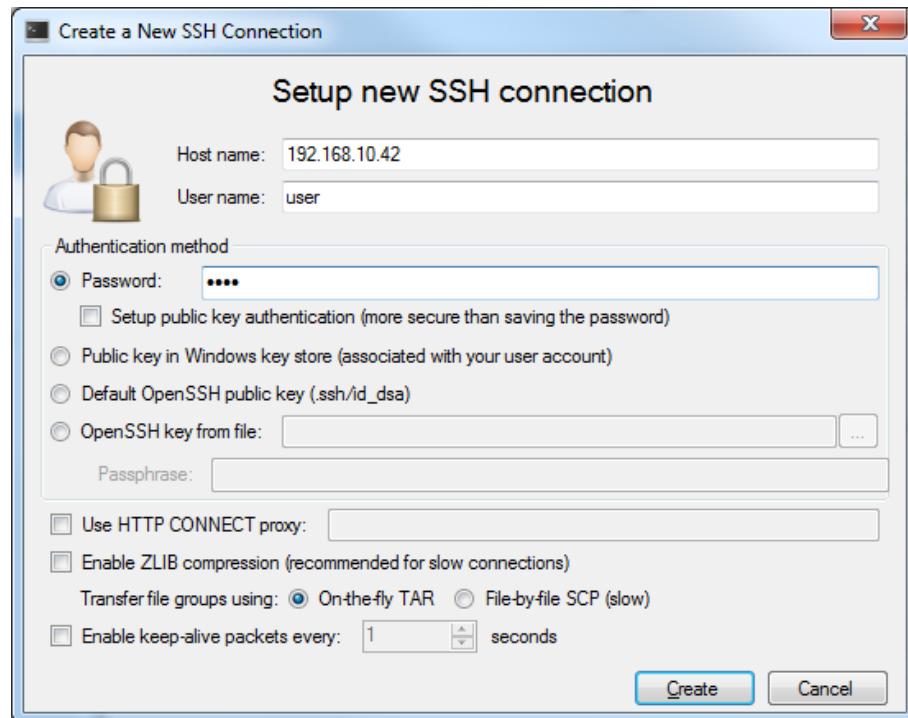


Нажимаем «Next»

Настраиваем параметры, как указано на рисунке ниже:



При настройке поля «Remote computer» выбираем пункт «New connection». Появится окно с параметрами соединения. Заполняем поля, как указано на рисунке ниже:

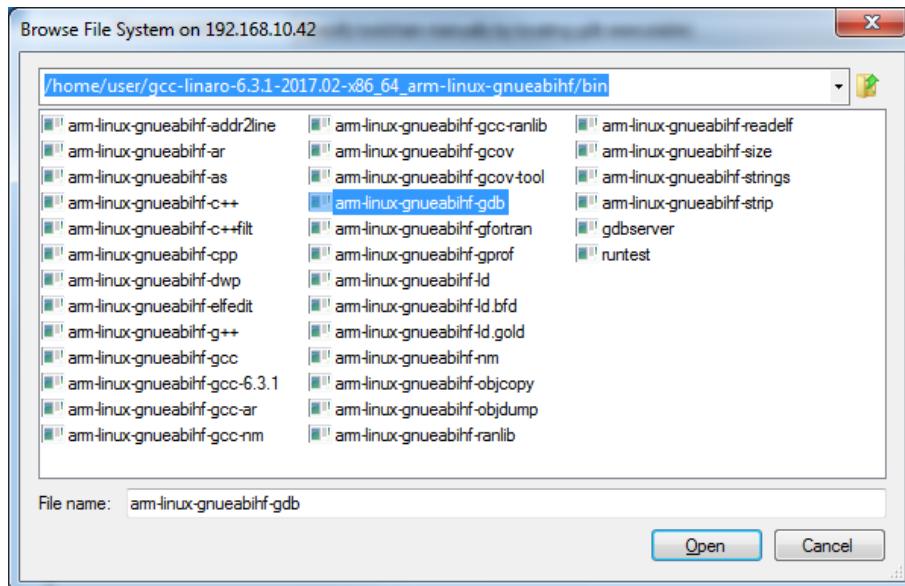


Нажимаем «Create». При первом подключении появится окно:



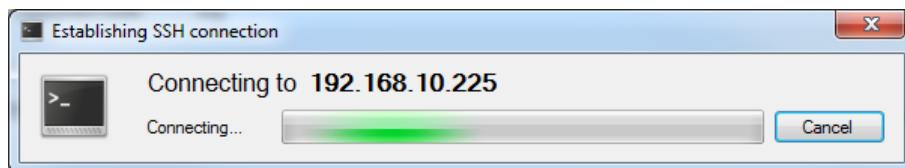
Нажимаем «Remember ...» и еще раз подтверждаем свои намерения.

При заполнении поля «Remote toolchain» выбираем пункт «Specify toolchain manually by locating gdb» и указываем местонахождение gdb в файловой системе виртуальной машины:

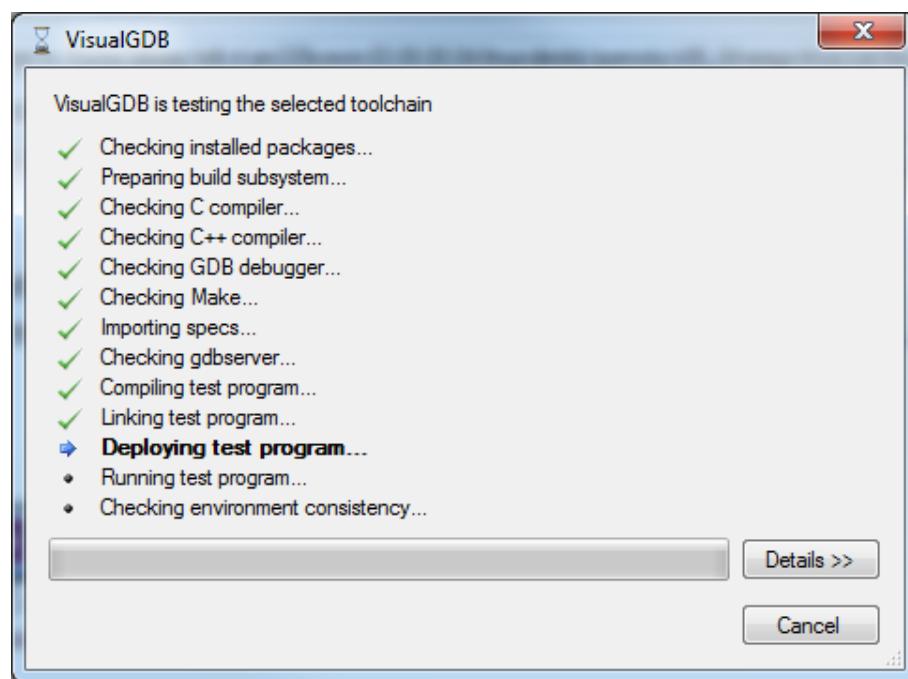


Нажимаем «Open» и возвращаемся к окну настройки соединений. Нажимаем «Next»

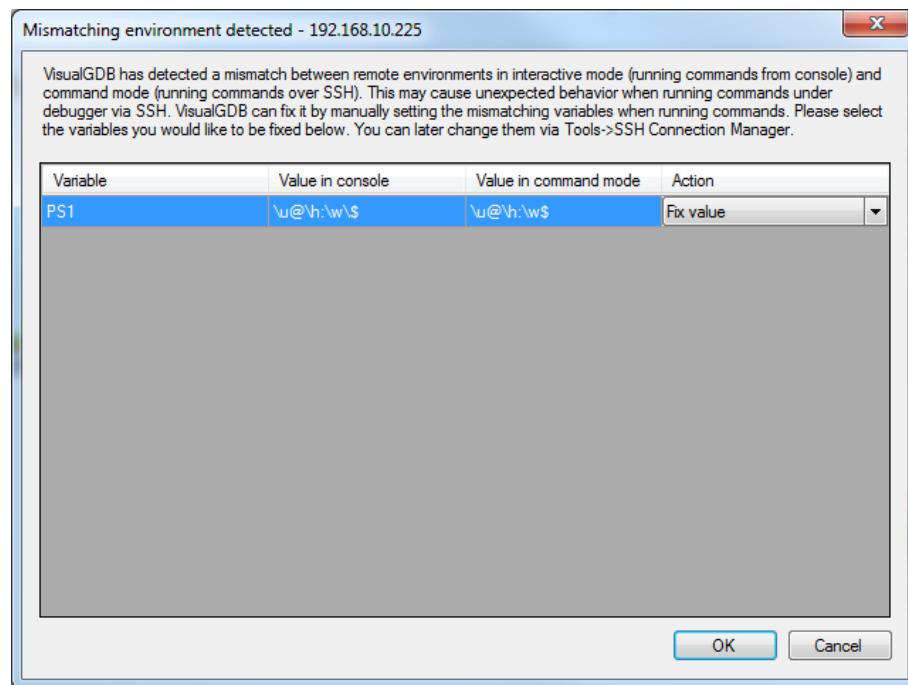
Происходит соединение с виртуальной машиной и целевым устройством. В случае неполадок будет отображено сообщение об отсутствии связи.



Далее идет проверка целевого устройства на наличие всех необходимых компонентов:

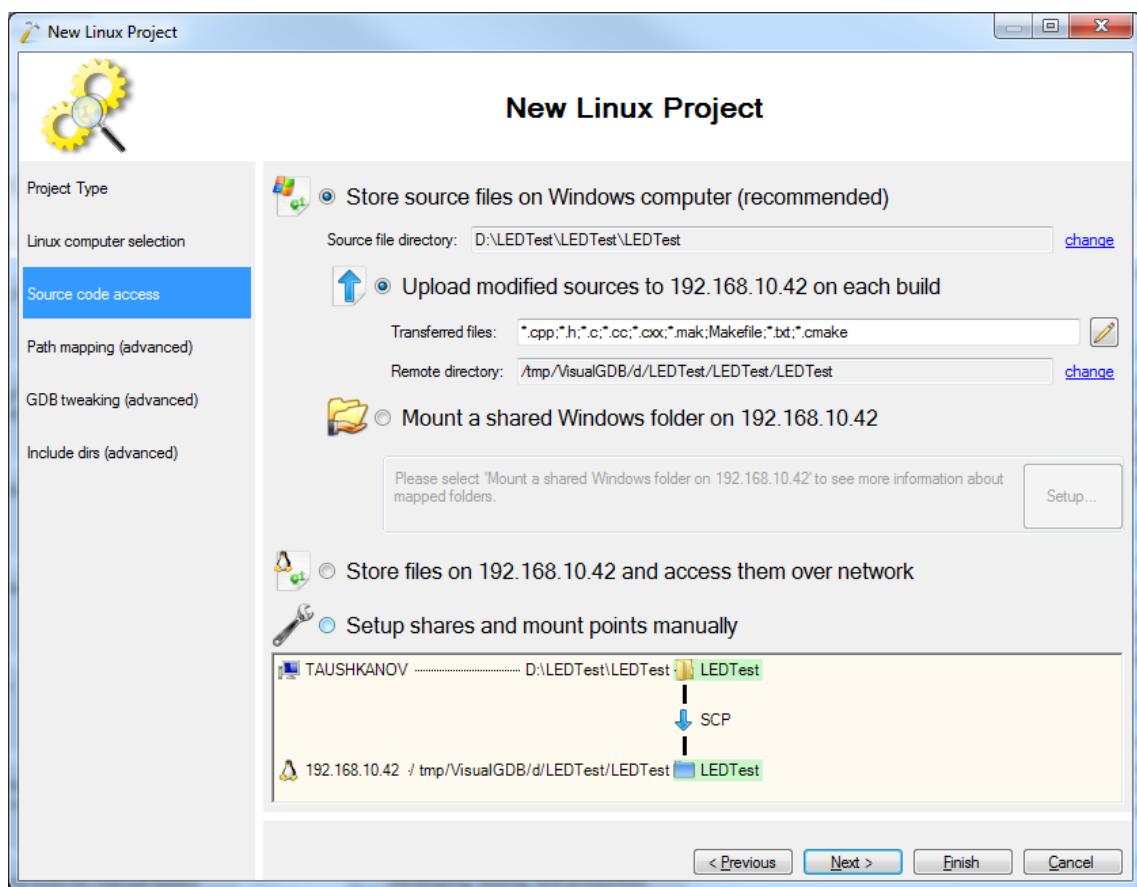


При проверке может обнаружиться несовпадение переменных окружения. В этом случае появится окно:

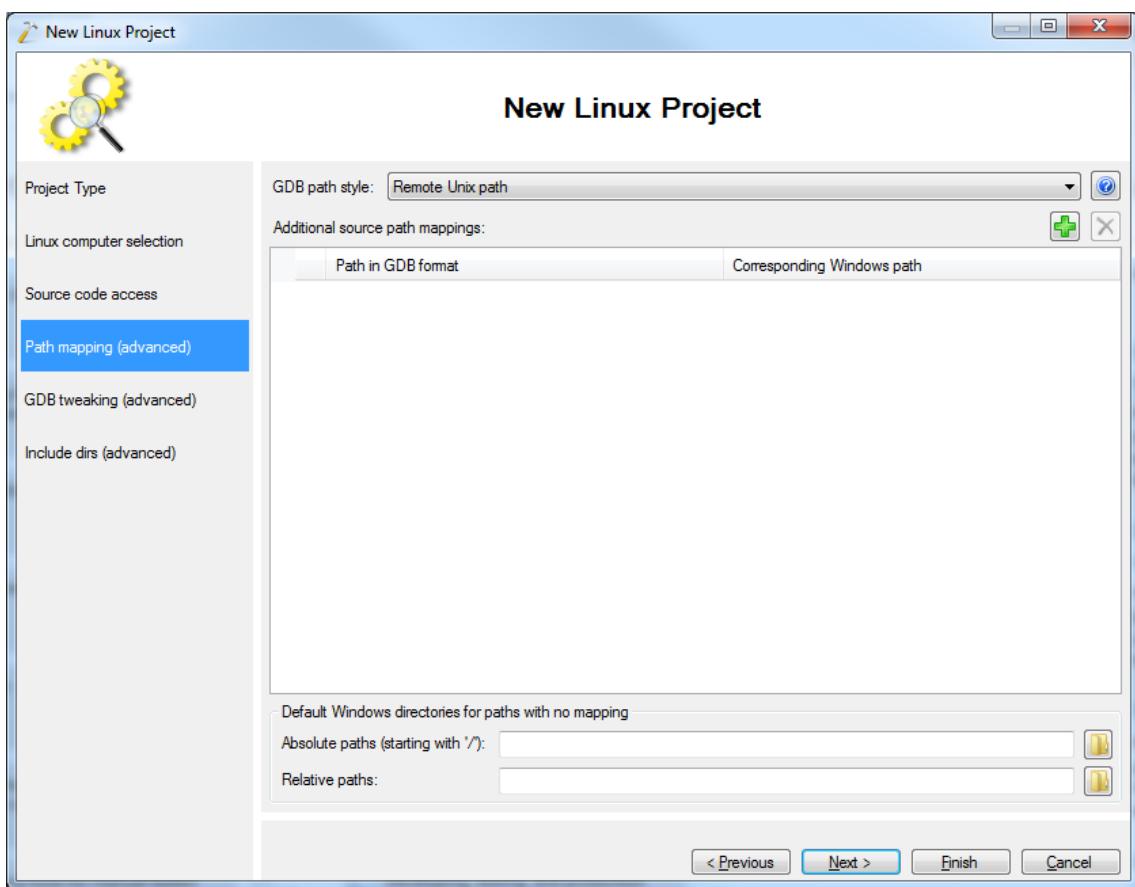


Согласитесь на предложенные варианты исправления и нажмите «OK»

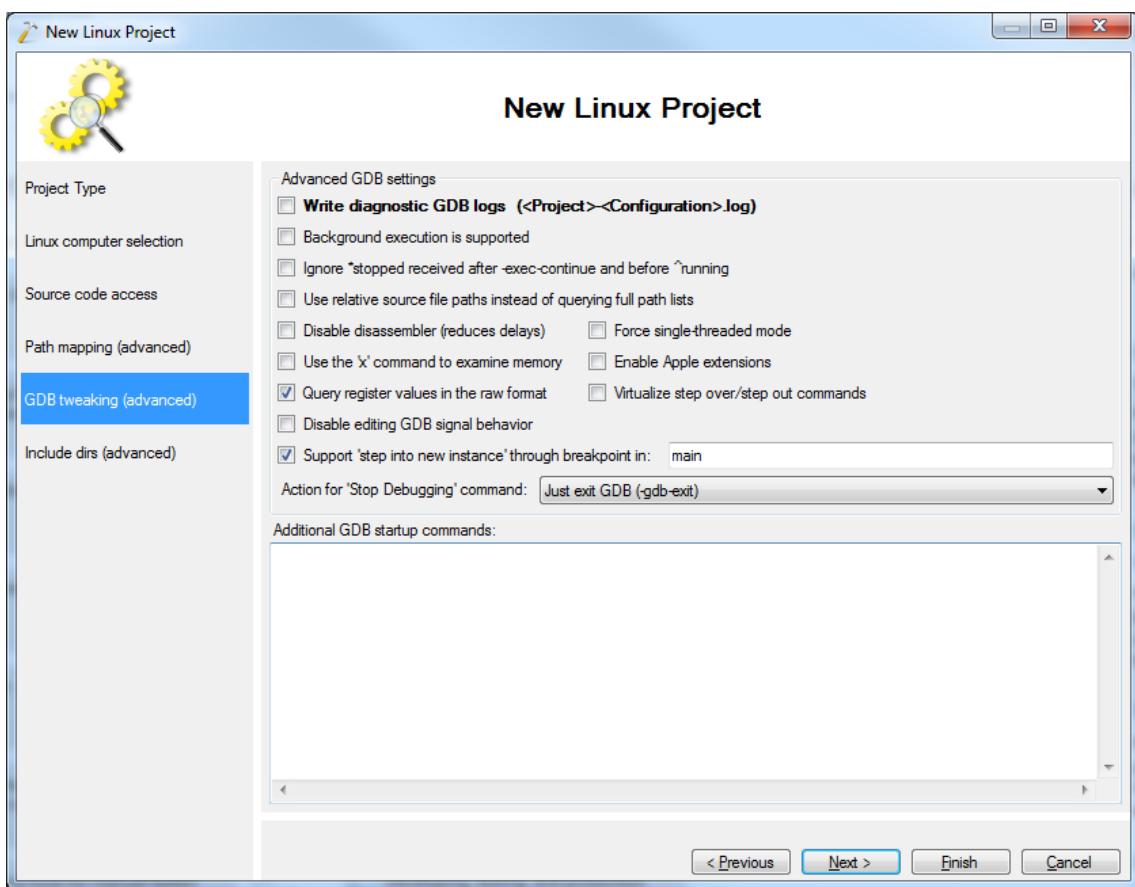
Далее выбираем пункт «Store source files on Windows computer»:



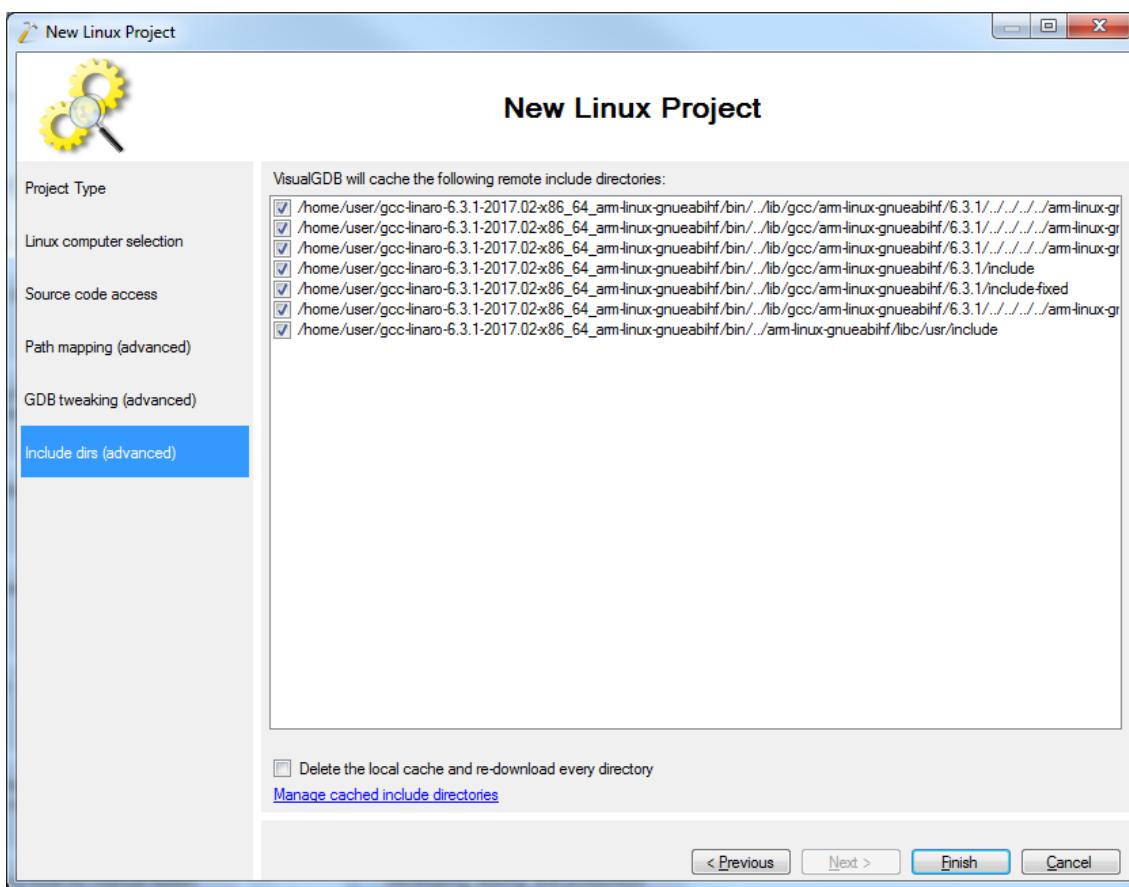
На этапе сопоставления путей можно ничего не менять.



Нажимаем «Next»



Нажимаем «Next»



Нажимаем «Finish». Создание проекта на этом закончено. Далее Visual Studio создаст решение, содержащее один проект, параметры которого мы только что закончили настраивать.

Позднее все настройки проекта можно изменить, выбрав в меню команду «Project-> VisualGDB Project Properties».

Так же при разработке проекта иногда требуется изменить параметры сборки, оптимизации, указать требуемые библиотеки и т.д. Эти параметры необходимо указать в makefile для соответствующего варианта сборки: Debug – debug.mak, Release – release.mak. Файлы makefile можно открыть через окно «Solution explorer».

6.2.2. Наполнение проекта кодом

Перейдем к наполнению проекта программным кодом.

Откроем окно «Solution explorer», содержащее все файлы нашего проекта, откроем файл «LEDTest.cpp». Файл содержит текст примера:

```
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    char sz[] = "Hello, World!";      //Hover mouse over "sz" while debugging to see its contents
    cout << sz << endl;           //===== Put a breakpoint here
    return 0;
}
```

Заменим весь имеющийся текст на текст нашей программы:

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#include <linux/fb.h>
#include <sys/ioctl.h>
#include <errno.h>
#include <string>
#include <string.h>
#include <vector>

using namespace std;
using std::vector;
using std::string;

int g_iMode = 0;

int TestLEDs()
{
    vector<string> arrDevices;

    arrDevices.push_back("/sys/class/leds/agava:work:green/brightness");
    arrDevices.push_back("/sys/class/leds/agava:work_usr:red/brightness");
    arrDevices.push_back("/sys/class/leds/agava:fault_usr:green/brightness");
    arrDevices.push_back("/sys/class/leds/agava:fault:red/brightness");
    arrDevices.push_back("/sys/class/leds/agava:programm:green/brightness");
    arrDevices.push_back("/sys/class/leds/agava:prog_usr:red/brightness");

    char val;
    FILE* f = NULL;

    int fd = -1;

    for (int t = 0; t < arrDevices.size(); t++)
    {
        f = fopen(arrDevices[t].c_str(), "r+");
        if (f != NULL)
        {
            val = 0x31;
            fwrite(&val, sizeof(char), 1, f);
            fflush(f);

            usleep(1000000);

            val = 0x30;
            fwrite(&val, sizeof(char), 1, f);

            fclose(f);
        }
        else
        {
            printf("Error opening led device %s.\n", arrDevices[t].c_str());
        }
    }

    return 0;
}

int main(int argc, char **argv)
{
    printf("LEDTTest: AGAVA6432.40 leds testing program.\n");
    TestLEDs();
    return 0;
}
```

```
}
```

6.2.3. Сборка приложения

Скомпилируем проект: выбираем команду «Build->Build Solution».

В окне вывода появится текст:

```
1>----- Build started: Project: LEDTest, Configuration: Debug Win32 -----
1> VisualGDB: Sending 6 updated source files to build machine...
1> VisualGDB: Run "make CONFIG=Debug" in directory "/tmp/VisualGDB/d/LEDTest/LEDTest" on user@192.168.10.42
(SSH)
1> mkdir Debug
1> /home/user/gcc-linaro-6.3.1-2017.02-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-g++ -ggdb -ffunction-
sections -O0 -DDEBUG=1 -c LEDTest.cpp -o Debug/LEDTest.o -MD -MF Debug/LEDTest.dep
1> /home/user/gcc-linaro-6.3.1-2017.02-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-g++ -o Debug/LEDTest -
-Wl,-gc-sections -Wl,--start-group Debug/LEDTest.o -Wl,--rpath='$ORIGIN' -Wl,--end-group
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

Приведенный текст содержит порядок действий системы при сборке приложения:

- отправка шести файлов с исходными текстами на машину сборки по адресу 192.168.10.42
- сборка приложения в варианте Debug на машине сборки 192.168.10.42

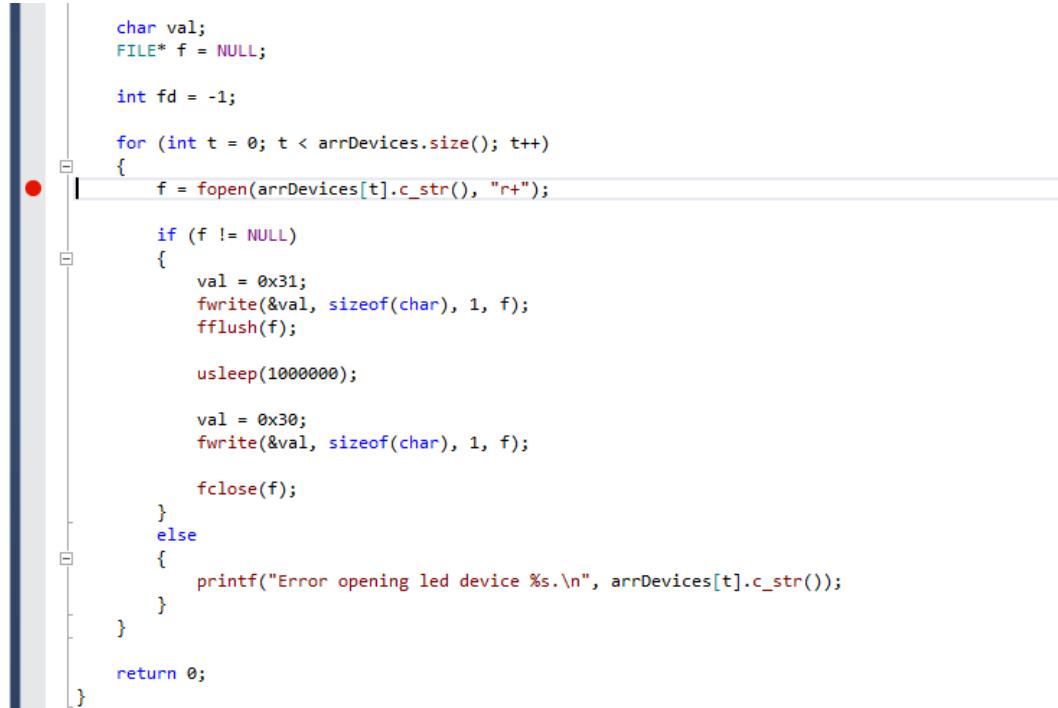
Сборка завершена, одно приложение собрано успешно.

6.2.4. Отладка приложения в среде Visual Studio

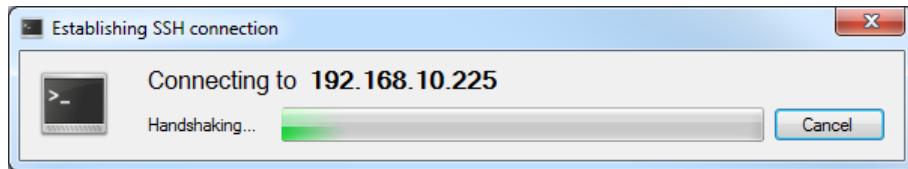
После успешной компиляции приложения можно перейти к отладке.

Поставим точку останова (клавиша F9 по умолчанию) на строке:

```
f = fopen(arrDevices[t].c_str(), "r+");
```



Вызовем команду «Debug->Start debugging with GDB». Появится окно подключения к целевому устройству:



После подключения GDB клиента к GDB серверу на контролере начнется отладка программы и при достижении места с точкой останова выполнение будет приостановлено.

Вывод консоли:

```
Process /tmp/LEDTes40 created; pid = 640
Listening on port 2000
Remote debugging from host 192.168.10.42
LEDTest: AGAVA6432.40 leds testing program.
```

Откроем окно Watch 1, внесем в список отслеживаемых переменных переменные «t», «arrDevices»:

Name	Value
t	0
arrDevices	[6 items] {/sys/class/leds/agava:backlight:red/brightness /sys/class/leds/agava:backlight:green/brightness /sys/class/leds/agava:backlight:blue/brightness /sys/class/leds/agava:work:green/brightness /sys/class/leds/agava:fault:red/brightness /sys/class/leds/agava:programm:green/brightness}

Далее отладку можно продолжить, запустив приложение на выполнение до следующей точки останова, либо продолжить выполнение программы по шагам.

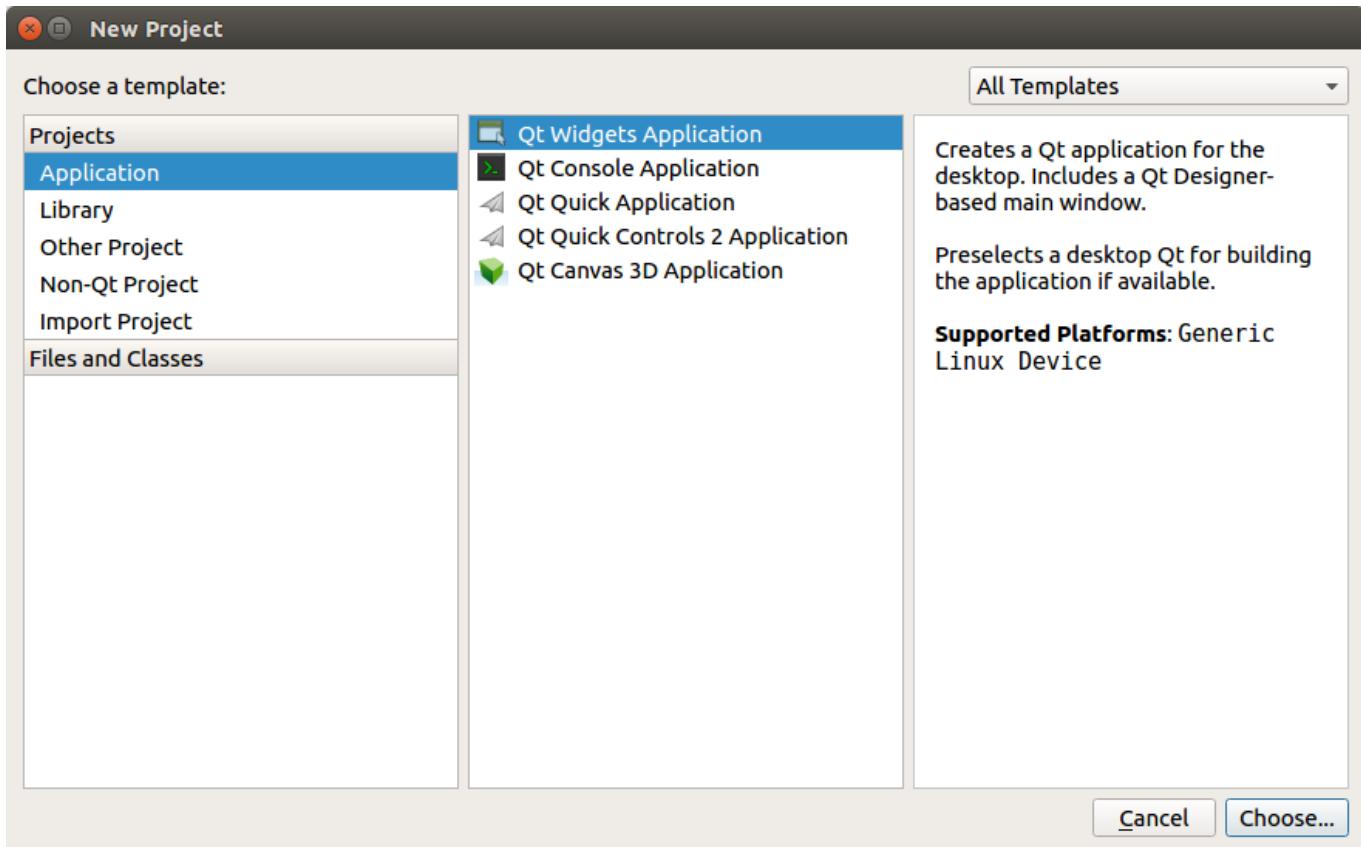
7. Разработка графического приложения с использованием Qt

7.1. Создание проекта в Qt Creator

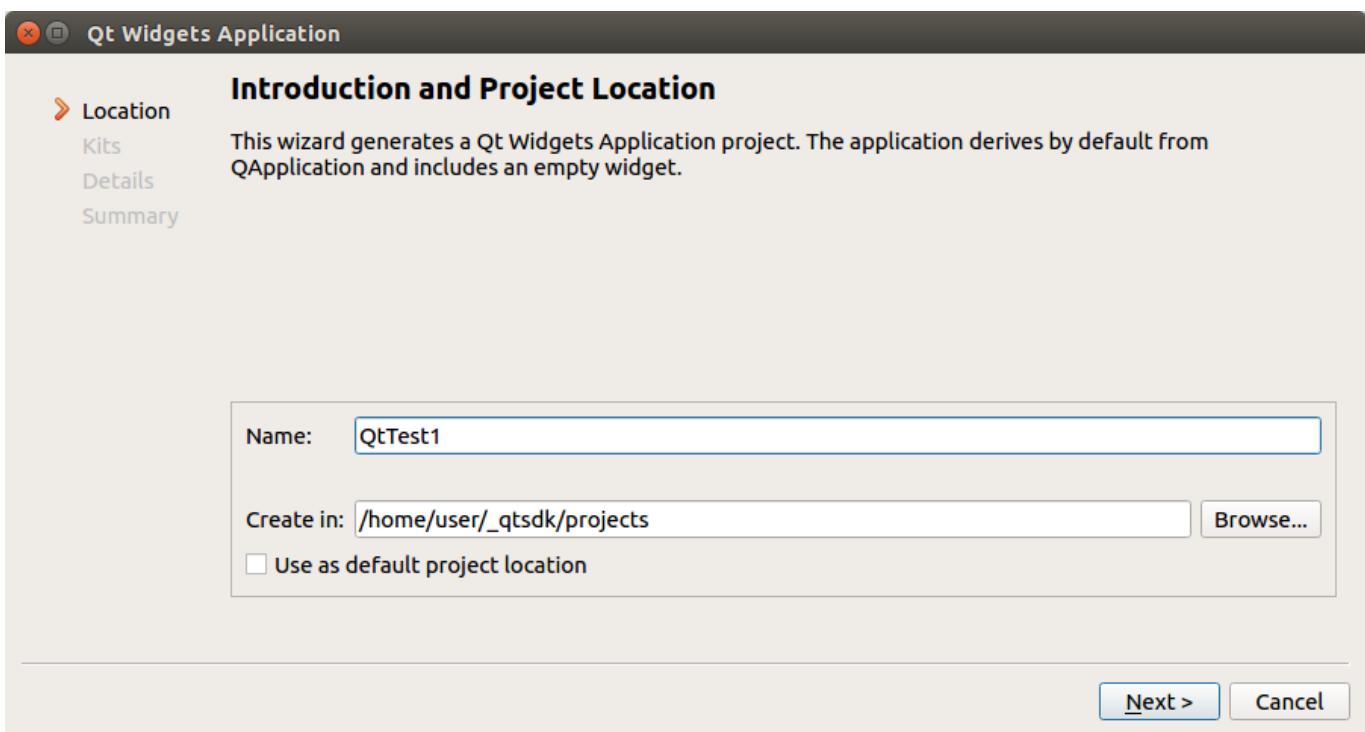
Создадим проект с именем «QtTest1» в среде Qt Creator, который будет хранить создаваемое нами приложение.

Открываем среду разработки. Вызываем команду «New Project».

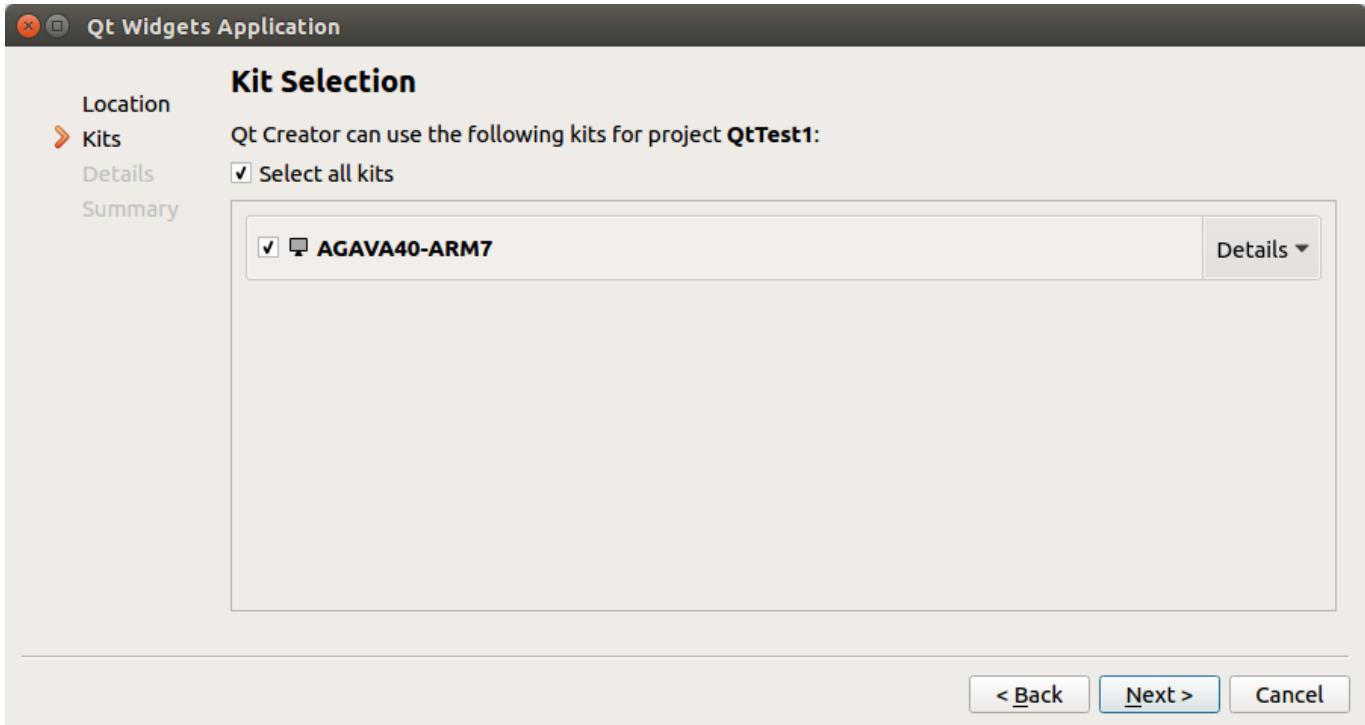
Открывается окно создания проекта, в котором на первом этапе необходимо выбрать тип разрабатываемого приложения. Выбираем вариант «Qt widgets application» и нажимаем кнопку «Choose...»:



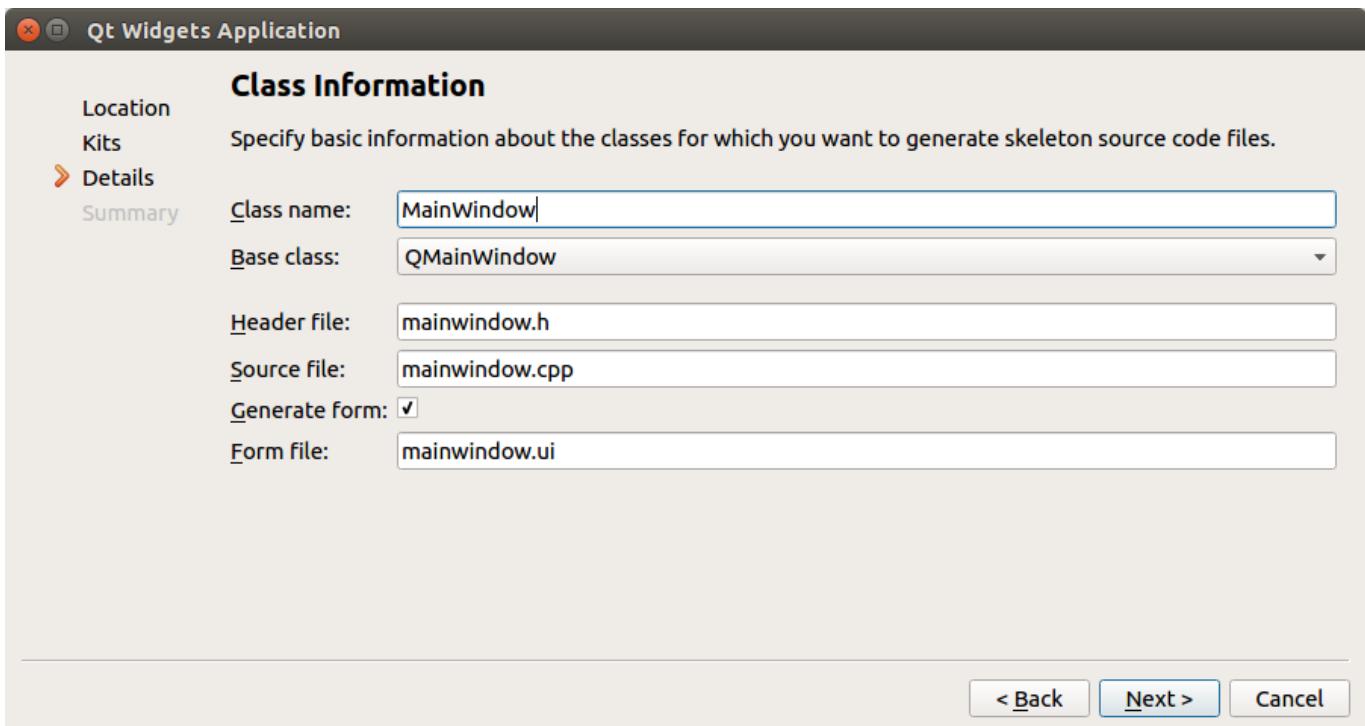
Далее вводим имя нового приложения, а так же путь, по которому он будет храниться и нажимаем кнопку «Next >»:



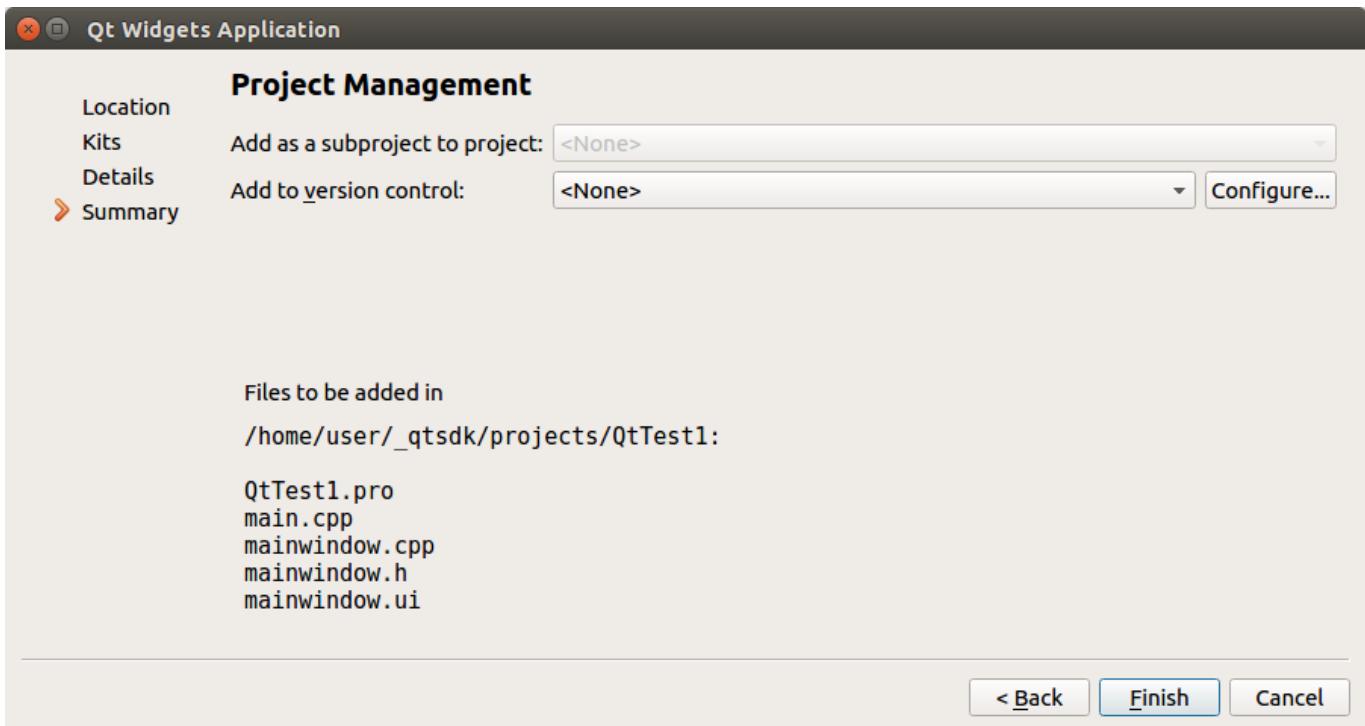
Далее переходим к выбору комплектов сборки приложений. В SDK AGAVA в среде Qt Creator уже создан комплект «AGAVA40-ARM7», необходимый для сборки приложений под контроллер АГАВА ПК40. Оставляем выбор, как указано на рисунке ниже и нажимаем кнопку «Next >»:



Далее нам предоставляется возможность изменить имена создаваемых файлов и классов. Если требуется внести изменения, вносим их и нажимаем кнопку «Next >»:



На последнем этапе создания проекта можно подключить к проекту систему контроля версий. В виртуальной машине не установлены системы контроля версий, поэтому просто нажимаем кнопку «Finish»:



На этом создание проекта можно считать завершенным. Созданный проект можно скомпилировать, но без дополнительной настройки его невозможно разместить и запустить на целевом устройстве.

7.2. Дополнительная настройка проекта

Для добавления возможности компиляции и размещения проекта на целевом устройстве необходимо:

- Задать путь размещения проекта на целевом устройстве
- Создать нужное целевое устройство в Qt Creator для размещения и запуска проектов

7.2.1. Размещение приложения на целевом устройстве

Для определения пути размещения приложения на целевом устройстве необходимо в файле проекта разместить примерно следующие строки, определяющие тип целевого устройства и путь размещения:

```
linux-* {  
    target.path = /tmp  
    INSTALLS += target  
}
```

7.2.2. Создание целевого устройства

Для размещения приложения на целевом устройстве и возможности его запуска/отладки необходимо создать целевое устройство в Qt Creator. Для этого вызовем команду «Tools | Options», и перейдем на вкладку «Devices»:

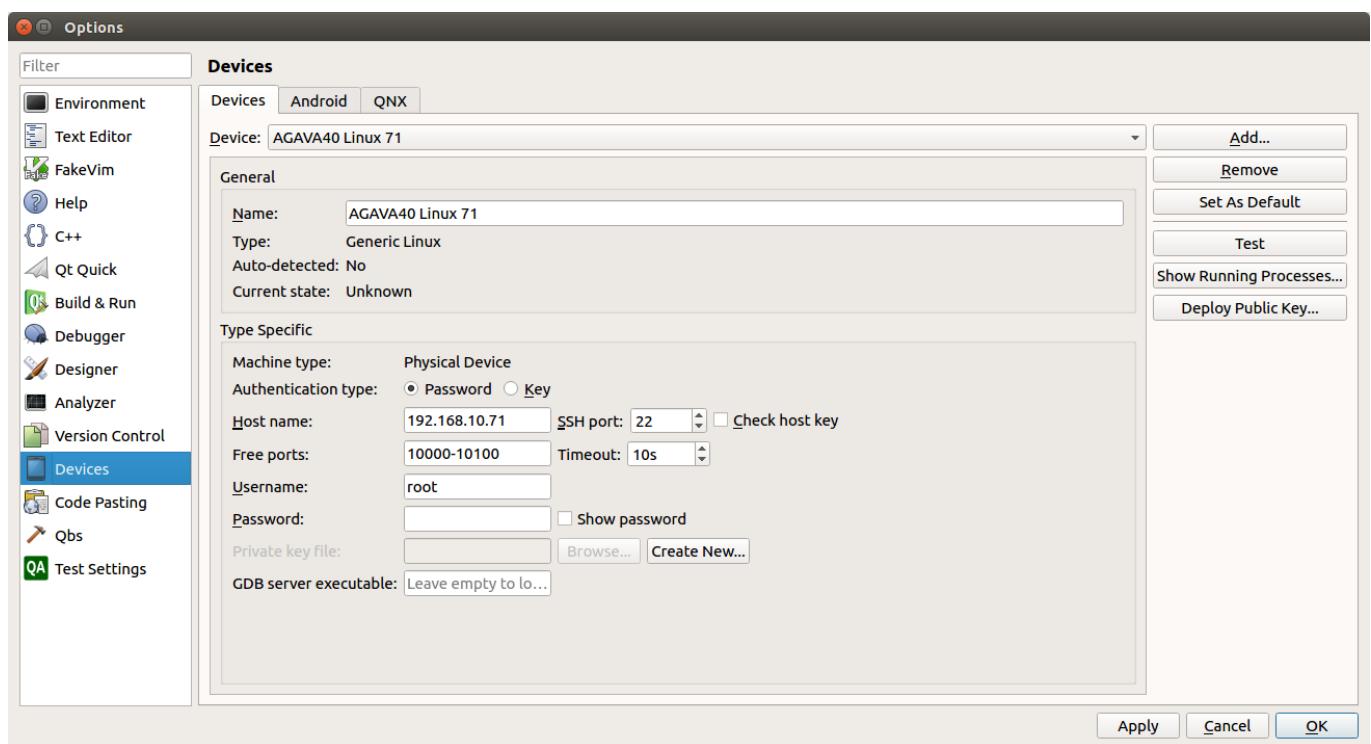
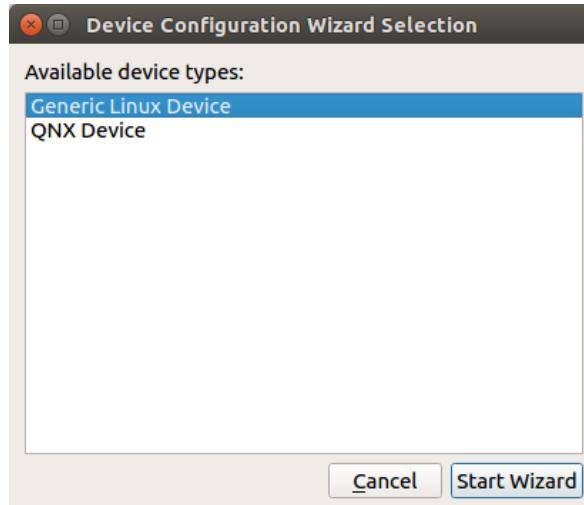


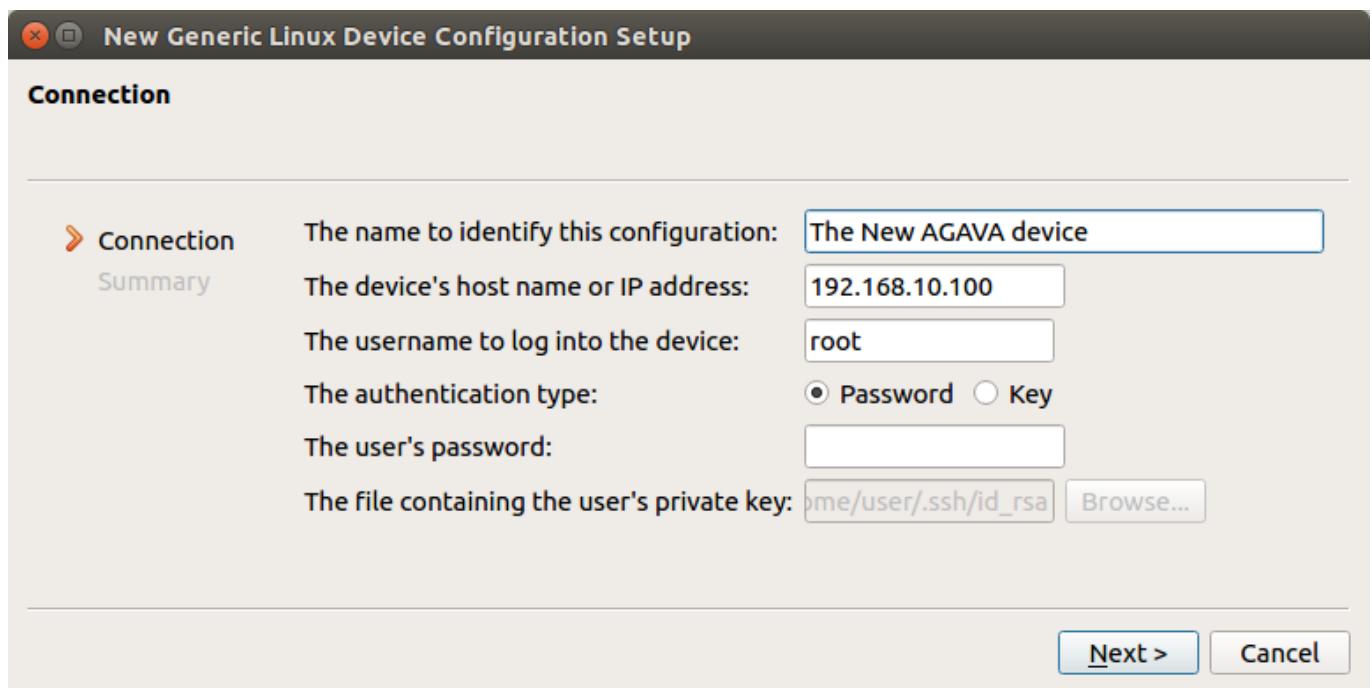
Рис. 1 Окно "Options"

Для создания нового устройства нажмем кнопку «Add».

В появившемся окне выбираем вариант «Generic Linux Device» и нажимаем кнопку «Start Wizard»:



Далее в появившемся окне «New Generic Linux Device Configuration setup» вводим параметры нового устройства, примерно как показано на рисунке ниже:



Нажимаем кнопку «Next >», и на следующем экране подтверждаем создание устройства. После завершения создания нового целевого устройства его можно указать в комплекте сборки как используемое.

7.2.3. Смена целевого устройства в комплекте сборки

При смене адреса целевого устройства или для размещения/отладки приложения на другом целевом устройстве необходимо в используемом комплекте сборки «AGAVA40-ARM7» указать другое целевое устройство. Процедура создания нового целевого устройства описана в п. 7.2.2. После создания устройства в окне «Options» перейдем на вкладку «Build & run», раздел «Kits», и в поле «Device» будет возможность выбрать новое целевое устройство.

7.3. Сборка и отладка приложения

8. Обновление базовых программных компонентов ПК

Прибор поставляется с установленными базовыми программными компонентами. В процессе эксплуатации прибора может возникнуть необходимость их обновления. Файлы программных компонентов могут быть получены через сайт Изготовителя – www.kb-agava.ru, либо предоставлены по запросу.

Базовое программное обеспечение для ПЛК состоит из следующих модулей:

- Загрузчик;
- ОС Linux;
- Корневая файловая система;

Загрузчик служит для загрузки ОС, а также для обновления программных компонентов контроллера и по умолчанию хранится во NAND-памяти ПЛК. Файлы компонентов загрузчика: u-boot.img (образ U-Boot) и MLO (первичный загрузчик). Данные файлы взаимосвязаны и должны применяться только совместно, одной и той же версии. При включении контроллера сначала происходит загрузка первичного загрузчика MLO во внутреннюю память процессора, который выполняет инициализацию необходимого оборудования и загружает основной загрузчик U-Boot, который впоследствии загружает компоненты ОС Linux и передает им управление. Контроллер позволяет выбирать источник загрузки при помощи микропереключателя 1, расположенного на боковой стенке лицевой панели – либо из NAND-памяти – это основной режим загрузки, положение микропереключателя «OFF» – вверх. Либо с SD-карты – это дополнительный режим загрузки для обновления или аварийного восстановления контроллера.

Программные компоненты ОС Linux хранятся в NAND-памяти контроллера и состоят из образа ядра Linux – файл zImage и файла описания устройств am335x-agava_40.dtb. Данные файлы взаимосвязаны и должны применяться только совместно, одной и той же версии.

Корневая файловая система содержит набор каталогов и утилит для нормальной работы ОС, хранится в NAND-памяти и монтируется при загрузке ядра ОС Linux. Имя файла образа для прошивки корневой файловой системы – agava.ubi.

Система исполнения CODESYS состоит из файлов codesyscontrol, codesyscontrol.a и CODESYSControl.cfg, размещена в корневой файловой системе в каталоге /usr/bin/codesys и запускается как сервис при загрузке ОС Linux. Данные файлы взаимосвязаны и должны применяться только совместно, одной и той же версии.

8.1. Обновление компонентов загрузчика

- Подготовить SD-карту с файловой системой FAT(12,16,32). Разместить в ее корневой каталог файлы для обновления MLO и u-boot.img. Установить SD-карту в прибор.
- Подключить ПЛК нуль-модемным кабелем к интерфейсу RS-232 компьютера. На компьютере настроить терминал, в соответствии с параметрами, указанными в п.0, выбрать соответствующий порт.
- Чтобы зайти в загрузчик, необходимо включить ПЛК и сразу нажимать любую клавишу в терминале компьютера до появления в нем строки приглашения:

```
AGAVA6432.40#.
```

- В терминале последовательно выполнить команды

```
run upd_mlo  
run upd_u-boot
```

- Убедиться, что команды выполнились без ошибок.
- Выключить ПЛК, затем включить и повторно зайти в загрузчик как было указано выше. Убедиться, что произошла загрузка обновленной версии загрузчика.
- Обновить переменные окружения нового загрузчика, выполнив команды в терминале

```
env default -f -a  
saveenv  
reset
```

- Убедиться что, произошла полная загрузка контроллера.

8.2. Обновление компонентов ОС Linux

Порядок действия по обновлению компонентов ОС Linux:

- Подготовить SD-карту с файловой системой FAT(12,16,32). В корневом каталоге SD-карты создать папку boot. Разместить в папке файлы для обновления - zImage и am335x-agava_40.dtb. Установить SD-карту в прибор.
- Подключить ПЛК нуль-модемным кабелем к интерфейсу RS-232 компьютера. На компьютере настроить терминал, в соответствии с параметрами, указанными в п.0, выбрать соответствующий порт.
- Чтобы зайти в загрузчик, необходимо включить ПЛК и сразу нажимать любую клавишу в терминале компьютера до появления в нем строки **AGAVA6432.40#**.
- В терминале последовательно выполнить команды

```
run upd_fdt  
run upd_linux  
reset
```

- Убедиться, что команды выполнилась без ошибок и контроллер перезагрузился с обновленной версией Linux.

8.3. Обновление корневой файловой системы

Внимание! При обновлении корневой файловой системы все пользовательские настройки, проекты и иные файлы пользователя не сохраняются. Перед обновлением их необходимо сохранить самостоятельно.

Порядок действий по обновлению файловой системы:

- Подготовить SD-карту с файловой системой FAT(12,16,32). Разместить в корневой каталог файл образа корневой файловой системы agava.ubi. Установить SD-карту в прибор.

- Подключить ПЛК нуль-модемным кабелем к интерфейсу RS-232 компьютера. На компьютере настроить терминал, в соответствии с параметрами, указанными в п.0, выбрать соответствующий порт.
- Для входа в загрузчик необходимо включить ПЛК и сразу начать нажимать любую клавишу в терминале компьютера до появления в нем строки **AGAVA6432.40#**.
- В терминале последовательно выполнить команды

```
run upd_rootfs  
reset
```

- Убедиться, что прошивка выполнилась без ошибок и контроллер перезагрузился с обновленной корневой файловой системой.

8.4. Обновление прикладного (пользовательского) программного обеспечения

8.4.1. Корневая файловая система версий до 2018.3

В операционную систему (до версии 2018.1) встроен сервис обновления agava-uob, выполняющийся при загрузке контроллера.

При старте сервиса выполняется проверка подключенных USB flash на наличие файла update.sh, и если таковой обнаружен, выполняет его.

В скрипте update.sh описываются все действия, необходимые для выполнения обновления ПО.

Ниже приведен пример скрипта, производящего подачу звукового сигнала и перезагрузку контроллера:

```
#!/bin/sh  
#Update script  
  
USB_DIR=`dirname $0`  
#Example usage:  
##$USB_DIR/ledtest  
  
#Make sound:  
echo -en "\07" > /dev/tty5  
  
#Echo message to console and reboot:  
echo "Done! Rebooting" >> /dev/tty0  
reboot
```

8.4.2. Корневая файловая система версий 2018.3 и позднее

9. Список рекомендуемой литературы

Нэйл Мэтью, Ричард Стоунс. Основы программирования в Linux. 4-е издание. БХВ-Петербург, 2009.

Николай Иванов. Программирование в Linux. Самоучитель. БХВ-Петербург, 2007.

Николай Секунов. Программирование на C++ в Linux. БХВ-Петербург, 2004.

Приложение 1. Описание реализации протокола Modbus для субмодулей

Параметры связи по-умолчанию: 115200, 8N1, адрес задается для каждого субмодуля с помощью контактов на кросс-плате.

Все регистры типа Holding Register, (функция чтения (R) 0x03 и 0x04, записи (W) 0x06 и 0x10).

Регистры общие для всех модулей располагаются по адресам от 0 до 999. Регистры, специфичные для каждого типа субмодуля располагаются по адресам, начиная с 1000.

Регистры, общие для всех модулей

Адрес	Описание	Значения	Значения по умолч.	Атрибут
0	Код типа субмодуля	1-65535		R
1	Версия ПО	1-65535		R
2	Число принятых пакетов с ошибкой	0-65535		R/W
3	Число ошибок CRC	0-65535		R/W
4				

Регистры субмодуля R – Релейные выходы

Код типа субмодуля – 3.

Внутреннее наименование – РЛ-2.

Адрес	Описание	Значения	Значения по умолч.	Атрибут
1000	Состояние выходов 1-2 / Установка выходов 1-2 (битовое поле)	0-65535		R/W
1001	Состояние выхода 1 / Установка выхода 1	0-1		R/W
1002	Состояние выхода 2 / Установка выхода 2	0-1		R/W
2000	Начало настроек регистров			

Регистры субмодуля DI – Дискретные входы

Код типа субмодуля – 6.

Внутреннее наименование – У1-4.

Адрес	Описание	Значения	Значения по умолч.	Атрибут
1000	Состояние входов 0-3 (битовое поле) 4 бит – установка антидребезга на 3 канал счетчика	0-15		R/W

	5 бит – установка антидребезга на 4 канал счетчика			
1001	Счетчик 1	0-65535		R/W
1002	Счетчик 2	0-65535		R/W
1003	Состояние входа 1	0-1		R
1004	Состояние входа 2	0-1		R
1005	Состояние входа 3	0-1		R
1006	Состояние входа 4	0-1		R
2000	Начало настроек регистров			

Регистры субмодуля AI – Аналоговые входы

Код типа субмодуля – 4.

Внутреннее наименование – A2-4.

Адрес	Описание	Значения	Значения по умолч.	Атрибут
1000	Значение входа 1 (в ед. измерения, дискрет. 0,01В; 0,01mA)	0-65535		R
1001	Значение входа 2 (в ед. измерения, дискрет. 0,01В; 0,01mA)	0-65535		R
1002	Значение входа 3 (в ед. измерения, дискрет. 0,01В; 0,01mA)	0-65535		R
1003	Значение входа 4 (в ед. измерения, дискрет. 0,01В; 0,01mA)	0-65535		R
1004	Значение входа 1 (АЦП)	0-4095		R
1005	Значение входа 2 (АЦП)	0-4095		R
1006	Значение входа 3 (АЦП)	0-4095		R
1007	Значение входа 4 (АЦП)	0-4095		R
1008	Выбор параметра измерения (0-напряжение, 1-ток)	0-15	0	R/W
2000	Коэф. Измерения напряжения, канал 1, мультипликативный коэффициент	LSB float		R/W
2001		MSB float		R/W
2002	Коэф. Измерения напряжения, канал 1, аддитивный коэффициент	LSB float		R/W
2003		MSB float		R/W
2004	Коэф. Измерения напряжения, канал 2, мультипликативный коэффициент	LSB float		R/W
2005		MSB float		R/W
2006	Коэф. Измерения напряжения, канал 2, аддитивный коэффициент	LSB float		R/W

2007		MSB float		R/W
2008	Коэф. Измерения напряжения, канал 3, мультипликативный коэффициент	LSB float		R/W
2009		MSB float		R/W
2010	Коэф. Измерения напряжения, канал 3, аддитивный коэффициент	LSB float		R/W
2011		MSB float		R/W
2012	Коэф. Измерения напряжения, канал 4, мультипликативный коэффициент	LSB float		R/W
2013		MSB float		R/W
2014	Коэф. Измерения напряжения, канал 4, аддитивный коэффициент	LSB float		R/W
2015		MSB float		R/W
2016	Коэф. Измерения тока канал 1, мультипликативный коэффициент	LSB float		R/W
2017		MSB float		R/W
2018	Коэф. Измерения тока канал 1, аддитивный коэффициент	LSB float		R/W
2019		MSB float		R/W
2020	Коэф. Измерения тока, канал 2, мультипликативный коэффициент	LSB float		R/W
2021		MSB float		R/W
2022	Коэф. Измерения тока канал 2, аддитивный коэффициент	LSB float		R/W
2023		MSB float		R/W
2024	Коэф. Измерения тока, канал 3, мультипликативный коэффициент	LSB float		R/W
2025		MSB float		R/W
2006	Коэф. Измерения тока канал 3, аддитивный коэффициент	LSB float		R/W
2027		MSB float		R/W
2028	Коэф. Измерения тока, канал 4, мультипликативный коэффициент	LSB float		R/W
2029		MSB float		R/W
2030	Коэф. Измерения тока канал 4, аддитивный коэффициент	LSB float		R/W
2031		MSB float		R/W
2032	Температурный коэф. (эталонный)	LSB float		R
2033		MSB float		R
2034	Моментальное значение температурного коэф.	LSB float		R
2035		MSB float		R

Регистры субмодуля DO – Дискретные выходы – открытый коллектор

Код типа субмодуля – 1.

Внутреннее наименование – ОК4.

Адрес	Описание	Значения	Значения по умолч.	Атрибут
1000	Состояние выходов 1-4 / Установка выходов 1-4 (битовое поле)	0-15		R/W
1001	Состояние выхода 1	0-1		R/W
1002	Состояние выхода 2	0-1		R/W
1003	Состояние выхода 3	0-1		R/W
1004	Состояние выхода 4	0-1		R/W
2000	Начало настроек регистров			

Регистры субмодуля SIM – Симисторные выходы

Код типа субмодуля – 2.

Внутреннее наименование – SM2.

Адрес	Описание	Значения	Значения по умолч.	Атрибут
1000	Состояние выходов 1-2 / Установка выходов 1-2 (битовое поле)	0-3		R/W
1001	Состояние выхода 1	0-1		R/W
1002	Состояние выхода 2	0-1		R/W
2000	Начало настроек регистров			

Регистры субмодуля АІО – Аналоговые входы и выходы

Код типа субмодуля – 5.

Внутреннее наименование – А1-2-2.

Адрес	Описание	Значения	Значения по умолч.	Атрибут
1000	Значение входа 1 (в ед. измерения, дискрет. 0,01В; 0,01mA)	0-65535		R
1001	Значение входа 2 (в ед. измерения, дискрет. 0,01В; 0,01mA)	0-65535		R
1002	Значение выхода 1 (в ед. измерения, дискрет. 0,01В; 0,01mA)	0-65535		R/W
1003	Значение выхода 2 (в ед. измерения, дискрет. 0,01В; 0,01mA)	0-65535		R/W
1004	Значение входа 1 (АЦП)	0-4095		R
1005	Значение входа 2 (АЦП)	0-4095		R

1006	Значение входа 1(прямое значение ШИМ)	0-4095		R/W
1007	Значение входа 2 (прямое значение ШИМ)	0-4095		R/W
1008	Выбор параметра измерения (0-напряжение 1-ток)	0-15	0	R/W
				R/W
2000	Коэффициент Измерения напряжения, канал 1, мультипликативный коэффициент	LSB float		R/W
2001		MSB float		R/W
2002	Коэффициент Измерения напряжения, канал 1, аддитивный коэффициент	LSB float		R/W
2003		MSB float		R/W
2004	Коэффициент Измерения напряжения, канал 2, мультипликативный коэффициент	LSB float		R/W
2005		MSB float		R/W
2006	Коэффициент Измерения напряжения, канал 2, аддитивный коэффициент	LSB float		R/W
2007		MSB float		R/W
2008	Коэф. Выхода напряжения, канал 1, мультипликативный коэффициент	LSB float		R/W
2009		MSB float		R/W
2010	Коэф. Выхода напряжения, канал 1, аддитивный коэффициент	LSB float		R/W
2011		MSB float		R/W
2012	Коэф. Выхода напряжения, канал 2, мультипликативный коэффициент	LSB float		R/W
2013		MSB float		R/W
2014	Коэф. Выхода напряжения, канал 2, аддитивный коэффициент	LSB float		R/W
2015		MSB float		R/W
2016	Коэф. Измерения тока канал 1, мультипликативный коэффициент	LSB float		R/W
2017		MSB float		R/W
2018	Коэф. Измерения тока канал 1, аддитивный коэффициент			
2019				
2020	Коэф. Измерения тока, канал 2, мультипликативный коэффициент	LSB float		R/W
2021		MSB float		R/W
2022	Коэф. Измерения тока, канал 2, аддитивный коэффициент			
2023				
2024	Коэф. Выхода тока, канал 1, мультипликативный коэффициент	LSB float		R/W
2025		MSB float		R/W

2026	Коэф. Выхода тока, канал 1, аддитивный коэффициент	LSB float		R/W
2027		MSB float		R/W
2028	Коэф. Выхода тока, канал 2, мультипликативный коэффициент	LSB float		R/W
2029		MSB float		R/W
2030	Коэф. Выхода тока, канал 2, аддитивный коэффициент	LSB float		R/W
2031		MSB float		R/W

Регистры субмодуля TMP – измерение температуры

Код типа субмодуля – 7.

Внутреннее наименование – ТМ-2.

Адрес	Описание	Значения	Значения по умолчанию	Атрибут
1000	Значение входа 1 (в ед. измерения, дискретность 0,1 Ом; 10мкВ)	Ом 0-65535 (мкВ * 10) - 32767 - +32768		R
1001	Значение входа 2 (в ед. измерения, дискретность 0,1Ом; 10мкВ)	Ом 0-65535 (мкВ * 10) - 32767 - +32768		R
1002	Температура холодного спая (дискретность 0.1 градус)	-200 – 800		R
1003	Выбор параметра измерения (битовое поле, 0-сопротивление, 1-напряжение)	0-3	0	R/W
1004	Значение входа 1 (АЦП)	Ом 0-65535 (мкВ * 10) - 32767 - +32768		R
1005	Значение входа 2 (АЦП)	Ом 0-65535 (мкВ * 10) - 32767 - +32768		R
1006				R
1010	Счетчик количества произведенных измерений АЦП	0-65535		R
1200	Разрешение записи/чтения EEPROM	0-65535	0	R/W

	0xAA55 – Разрешить однократную запись/чтение EEPROM Любое другое число – Запрет записи/чтения EEPROM			
1201	Команды работы с EEPROM 0xAA01 – Записать текущие настройки в EEPROM 0xAA02 – Считать настройки из EEPROM После выполнения каждой команды требуется разрешать запись/чтение EEPROM через регистр 1200	0-65535	0	R/W
2000	Канал 1, сопротивление, мультипликативный коэффициент	LSB float		R/W
2001		MSB float		R/W
2002	Канал 1, сопротивление, аддитивный коэффициент	LSB float		R/W
2003		MSB float		R/W
2004	Канал 2, сопротивление, мультипликативный коэффициент	LSB float		R/W
2005		MSB float		R/W
2006	Канал 2, сопротивление, аддитивный коэффициент	LSB float		R/W
2007		MSB float		R/W
2008	Канал 1, напряжение, мультипликативный коэффициент	LSB float		R/W
2009		MSB float		R/W
2010	Канал 1, напряжение, аддитивный коэффициент	LSB float		R/W
2011		MSB float		R/W
2012	Канал 2, напряжение, мультипликативный коэффициент	LSB float		R/W
2013		MSB float		R/W
2014	Канал 2, напряжение, аддитивный коэффициент	LSB float		R/W
2015		MSB float		R/W
2016	Значение входа 1, сопротивление	LSB float		R/W
2017		MSB float		R/W
2018	Значение входа 2, сопротивление	LSB float		R/W
2019		MSB float		R/W
2020	Значение входа 1, напряжение	LSB float		R/W
2021		MSB float		R/W
2022	Значение входа 2, напряжение	LSB float		R/W
2023		MSB float		R/W

Регистры субмодуля DO-6 – Дискретные выходы – открытый коллектор

Код типа субмодуля – 8.

Внутреннее наименование – ОК-6.

Адрес	Описание	Значения	Значения по умолч.	Атрибут
1000	Состояние выходов 1-6 / Установка выходов 1-6 (битовое поле)	0-63	0	R/W
1001	Состояние выхода 1	0-1	0	R/W
1002	Состояние выхода 2	0-1	0	R/W
1003	Состояние выхода 3	0-1	0	R/W
1004	Состояние выхода 4	0-1	0	R/W
1005	Состояние выхода 5	0-1	0	R/W
1006	Состояние выхода 6	0-1	0	R/W
1007	Запуск канала 1 шагового двигателя (сбрасывается при завершении выполнения)	0-1	0	R/W
1008	Запуск канала 2 шагового двигателя (сбрасывается при завершении выполнения)	0-1	0	R/W
1009	Кол-во шагов канала 1 шагового двигателя младшая часть	0 –0xFFFFFFFF	0	R/W
1010	старшая часть		0	R/W
1011	Кол-во шагов канала 2 шагового двигателя младшая часть	0 –0xFFFFFFFF	0	R/W
1012	старшая часть		0	R/W
1013	Частота импульсов 1 канала ШД, в Герцах младшая часть	1- 200000	1	R/W
1014	старшая часть			R/W
1015	Частота импульсов 2 канала ШД, в Герцах младшая часть	1- 200000	1	R/W
1016	старшая часть			R/W

Регистры субмодуля ENI-2 – Модуль энкодера

Код типа субмодуля – 9.

Внутреннее наименование – ENI-2.

Адрес	Описание	Значения	Значения по умолч.	Атрибут
1000	Кол-во шагов канала 1 младшая часть	0 – 0xFFFFFFFF	0	R/W
1001	старшая часть		0	R/W
1002	Кол-во шагов канала 2 младшая часть	0 – 0xFFFFFFFF	0	R/W
1003	старшая часть		0	R/W
1004	Напряжение коммутации 0 – 5 Вольт 1 – 12 Вольт 2 – 24 Вольта	0 - 2	0	R/W
1005	разрешение запуска счета при поступлении импульса с z-контакта первого канала: 0-счетчик ничем не ограничен – счет идет постоянно 1-обнуление счетного канала. Ожидание поступление z – импульса. После поступления z-импульса, этот регистр сбрасывается	0-1	0	R/W
1006	разрешение запуска счета при поступлении импульса с z-контакта второго канала: 0-счетчик ничем не ограничен – счет идет постоянно 1-обнуление счетного канала. Ожидание поступление z – импульса. После поступления z-импульса, этот регистр сбрасывается	0-1	0	R/W

©1992-2017 г. Конструкторское бюро «АГАВА»

Использование приведенных в настоящем документе материалов без официального разрешения КБ «АГАВА» запрещено.

АГАВА ПК-40

Все права защищены