



www.kb-agava.ru

АГАВА

Настоящая газовая автоматика

ООО КБ «АГАВА»

620026, г. Екатеринбург, ул. Бажова, 174, 3 эт.

т/ф 343/262-92-76 (78,87)

agava@kb-agava.ru; <http://www.kb-agava.ru>

Промышленный контроллер
АГАВА ПК-30

Быстрый старт

Редакция 1.1

Екатеринбург

2017

Оглавление

1 Введение.....	3
1.1 Используемые термины и сокращения:.....	3
2 Назначение.....	4
3 Подготовка к работе с прибором	5
3.1 Загрузка и установка необходимого программного обеспечения	5
4 Первое включение и настройка контроллера.....	6
4.1 Установка времени и даты.....	6
4.2 Настройка сетевого доступа	7
4.3 Подключение к прибору и просмотр файловой системы.....	7
5 Разработка пользовательского прикладного программного обеспечения с использованием компиляторов C/C++	9
5.1 Запуск и настройка виртуальной машины	9
5.2 Разработка простого приложения с использованием системы сборки make в среде Linux	10
5.3 Запуск приложений на целевом контроллере.....	11

1 Введение

Данное руководство предназначено для быстрого ознакомления основными принципами использования изделия.

1.1 Используемые термины и сокращения:

SSH – Secure Shell;

VM – виртуальная машина;

ОЗУ – оперативное запоминающее устройство;

ОС – операционная система;

ПЛК – программируемый логический контроллер (промышленный контроллер);

ПО – программное обеспечение;

ФС – файловая система.

2 Назначение

Промышленный контроллер АГАВА ПК-30 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 приложений в прибор и его отладка производятся через интерфейс Ethernet.

3 Подготовка к работе с прибором

3.1 Загрузка и установка необходимого программного обеспечения

Для первичного ознакомления с контроллером необходимо загрузить и установить на основной персональный компьютер следующее программное обеспечение:

- терминальная программа (Terminal, Hyperterminal, PuTTY и др.);
- SFTP-клиент (Filezilla, WinSCP и др.)

Для разработки программного обеспечения необходимо загрузить и установить на основной персональный компьютер следующее программное обеспечение:

- VMWare Player или VMWare Workstation;
- образ виртуальной машины AGAVA SDK 30;
- текстовые редакторы, Microsoft Visual Studio;
- дополнение VisualGDB для Visual Studio.

4 Первое включение и настройка контроллера

1. Подключить консоль через порт RS232 с помощью нуль-модемного кабеля из комплекта поставки, запустить терминальную программу.

Параметры терминала для консоли:

- скорость (бит/с): 115200;
- биты данных: 8;
- четность: Нет;
- стоповые биты: 1;
- управление потоком: Нет.

2. Подать питание 220 В на разъем X5, контакты 1, 3. Начнется загрузка ПО контроллера.

После появления приглашения

```
Arago Project http://arago-project.org agava6432.30 ttyS0
Agava 2017.03 agava6432.30 ttyS0
agava6432 login:
```

ввести имя пользователя «root».

4.1 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);
- 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 контроллера с серверами точного времени.

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

4.2 Настройка сетевого доступа

Для использования сетевых ресурсов необходимо настроить подключение к сети Ethernet. По умолчанию прибор настроен на получение сетевых настроек по DHCP. Просмотреть IP-адрес и другую сетевую конфигурацию можно из консоли, набрав команду:

```
ifconfig
```

Разово задать IP-адрес контроллера, действующий до перезагрузки контроллера, можно с помощью команды:

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

Задать статический IP-адрес, который будет устанавливаться автоматически каждый раз при загрузке контроллера, можно в файле `/etc/systemd/network/10-eth.network`:

```
[Network]
DHCP=no
Address=192.168.10.32/24
Gateway=192.168.10.10
```

4.3 Подключение к прибору и просмотр файловой системы

При загруженной ОС, подключенной и настроенной сети доступ к системной консоли можно получить по SSH, а так же с помощью консоли на порту RS-232, как описано в п. 4.

Права администратора для входа по SSH:

- логин: root;
- пароль отсутствует.

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

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

- `/run/media/mmcblk[0,1]p[0-3]` для SD-карты;
- `/run/media/sda[1,4]` для и USB-flash.

Для того, чтобы иметь доступ к SD-карте контроллера через USB-интерфейс, необходимо загрузить модуль `g_mass_storage.ko`. При подключении компьютера к разъему USB-OTG прибора, на компьютере

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

появится дополнительный съемный диск с содержимым SD-карты. Также загрузка данного модуля необходима для подключения к разъему USB-OTG других устройств (USB-флеш и т.п.).

Модуль *g_mass_storage.ko* автоматически загружается при старте контроллера. Если автоматическая загрузка была отключена, вручную модуль можно загрузить командой:

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

5 Разработка пользовательского прикладного программного обеспечения с использованием компиляторов C/C++

Для выполнения работ по разработке пользовательского программного обеспечения необходимо:

- запустить виртуальную машину AGAVA SDK30;
- настроить IP-адрес виртуального компьютера на подходящий для использования в локальной сети;
- создать код приложения и переместить его в виртуальную машину, либо воспользоваться готовым примером;
- запустить процесс компиляции;
- поместить скомпилированное приложение в целевой контроллер и запустить его на выполнение.

5.1 Запуск и настройка виртуальной машины

1. Загрузите образ VM с сайта www.kb-agava.ru и запустите его на выполнение. В VM уже установлено все необходимое для работы программное обеспечение:

- кросскомпилятор g++ 5.3;
- SFTP клиент FileZilla;
- SSH Server;
- файловый менеджер Midnight Commander;
- клиент git;
- архиваторы gzip, ar, tar и др.

В операционной системе виртуальной машины создан пользователь user с паролем user.

2. После окончания загрузки ОС введите пароль пользователя для перехода к рабочему столу.
3. Откройте терминал.
4. Задайте IP-адрес машины.

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

```

5.2 Разработка простого приложения с использованием системы сборки make в среде Linux

Скомпилируем небольшое приложение LEDTest, использующее аппаратное обеспечение контроллера АГАВА 6432.30. Приложение будет поочередно включать и выключать все имеющиеся светодиоды, а также светодиоды подсветки дисплея.

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

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

```
user@ubuntu:~/applications/LEDTest$ 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/LEDTest$ 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-gnueabi-g++ -std=c++11 -
fexceptions -ggdb -ffunction-sections -O0 -DDEBUG=1 -c LEDTest.cpp -o
Debug/LEDTest.o -MD -MF Debug/LEDTest.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-gnueabi-g++ -o Debug/LEDTest -
Wl,-gc-sections -Wl,--start-group Debug/LEDTest.o -Wl,--rpath='$ORIGIN' -Wl,--
end-group

```

После завершения сборки исполняемый файл будет помещен в папку Debug:
/home/user/applications/LEDTest/Debug

Проверим получившийся файл с помощью программы readelf:

```
readelf -h /home/user/applications/LEDTest/Debug/LEDTest
```

Результат выполнения команды:

```

ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00
  Class:                               ELF32
  Data:                                   2's complement, little endian
  Version:                               1 (current)
  OS/ABI:                                UNIX - System V
  ABI Version:                           0

```

```
Type: EXEC (Executable file)
Machine: ARM
Version: 0x1
Entry point address: 0x10a1d
Start of program headers: 52 (bytes into file)
Start of section headers: 133484 (bytes into file)
Flags: 0x5000402, has entry point, Version5 EABI, hard-
float ABI
Size of this header: 52 (bytes)
Size of program headers: 32 (bytes)
Number of program headers: 8
Size of section headers: 40 (bytes)
Number of section headers: 38
Section header string table index: 35
```

Результат выполнения содержит информацию о типе файла (elf32 EXEC), операционной системе (UNIX - System V) и целевой платформе (ARM).

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

```
CONFIG=RELEASE make
```

Сборка приложения в этом варианте будет произведена аналогично предыдущему, но исполняемый файл будет помещен в папку Release.

5.3 Запуск приложений на целевом контроллере

Поместите собранное приложение в целевой контроллер: с помощью файлового менеджера Filezilla подключитесь к контроллеру по установленному адресу (порт 22, логин root, пароля нет), скопируйте файл LEDTest в папку /tmp.

Подключитесь к контроллеру через консоль RS232 или по SSH протоколу (логин root, пароля нет), и запустите приложение:

```
/tmp/LEDTest
```

В консоль будет выведен текст, и светодиоды контроллера будут поочередно менять свой цвет согласно заложенному алгоритму.

©1992-2017 г. Конструкторское бюро «АГАВА»

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

АГАВА ПК-30

Все права защищены