

Remote control via Web server

ТЕХНОЛОГИИ ПОСТРОЕНИЯ И ИСПОЛЬЗОВАНИЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ

Удаленное управление через Веб-сервер

Цель работы: Рассмотреть средства для удаленного управления аппаратными ресурсами компьютера Raspberry Pi через Веб-сервер.

Задача работы: Установить связь между удаленным компьютером и устройствами, подключенными к порту GPIO компьютера Raspberry Pi. Связь осуществляется через Веб-сервер Raspberry Pi.

Приборы и принадлежности: Персональный компьютер, компьютер Raspberry Pi (версия B), маршрутизатор.

ВНИМАНИЕ! В конце работы находятся исправления и дополнения к последним версиям ПО.

ВВЕДЕНИЕ

Существует несколько способов подключения к компьютеру с другого устройства. Например, к Raspberry Pi можно подключиться через USB, RS-232, I²C и др. последовательные интерфейсы. Но подключение через Веб-сервер Raspberry Pi имеет то преимущество, что может быть выполнено с любого устройства имеющего браузер. Для Web подключений используются языки программирования HTML5 и JavaScript которые применяются также и для построения удобных пользовательских интерфейсов используемых как для наблюдения за состоянием объекта так и для перевода объектов в заданное состояние. В этой работе рассматриваются варианты построения Web связи между удаленным компьютером и устройствами, подключенными к порту GPIO компьютера Raspberry Pi.

В разделе “Общие сведения” освещаются следующие вопросы.

- Установка HTTP Веб-сервера Apache на RPi
- Получение параметров сетевого соединения RPi
- Установка **root** прав пользователя
- Установка диспетчера задач
- Установка менеджера файлов

- Установка утилит доступа к GPIO
- Атрибуты файла (директории) и их установка
- Автоматический запуск команд при старте RPi
- Структура html файла Веб-сервера для связи с GPIO через браузер
- CGI скрипты Web соединений
- Подключение Python программ к Веб-браузеру в CGI формате
- Запуск HTML файлов RPi с удаленного Веб-клиента
- Считывание скриптов RPi через Веб-браузер
- Сообщение об ошибках Веб-браузера

ОБЩИЕ СВЕДЕНИЯ

Установка HTTP Веб-сервера Apache на RPi

Для управления компьютером Raspberry Pi через сеть необходимо установить на RPi Веб-сервер. Установка наиболее популярного HTTP Веб-сервера “Apache” выполняется командой

```
sudo apt-get install apache2 php5 libapache2-mod-php5
```

Для проверки работы сервера в браузер удаленного компьютера (смартфона или другого устройства с браузером) введите IP адрес Raspberry Pi. При успешном соединении удаленный браузер выводит сообщение "It Works!". Источник сообщения “It works! ...” находится в файле RPi: `var/www/index.html`.



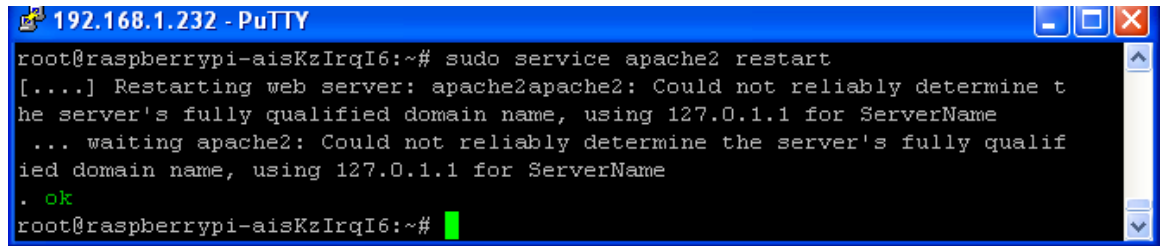
Примечание:

- При обращении к серверу только по IP адресу (без указания имени файла) выполняется программа `/var/www/index.html`.
- Для обращения к файлам каталога `/var/www/`, например, к `remote_led.php`, необходимо ввести их сетевой адрес, например, `http://192.168.1.232/remote_led.php`.
- Перезапуск Веб-сервера Apache 2 выполняется командой

```
$ sudo /etc/init.d/apache2 restart
```

или

```
$ sudo service apache2 restart
```



```
192.168.1.232 - PuTTY
root@raspberrypi-aisKzIrqI6:~# sudo service apache2 restart
[...] Restarting web server: apache2apache2: Could not reliably determine t
he server's fully qualified domain name, using 127.0.1.1 for ServerName
... waiting apache2: Could not reliably determine the server's fully qualif
ied domain name, using 127.0.1.1 for ServerName
. ok
root@raspberrypi-aisKzIrqI6:~# █
```

- Для остановки сервера Apache 2 необходимо введите команду

```
$ sudo /etc/init.d/apache2 stop
```

После остановки сервера RPi удаленный браузер выводит сообщение:

Connection closed by remote server ...

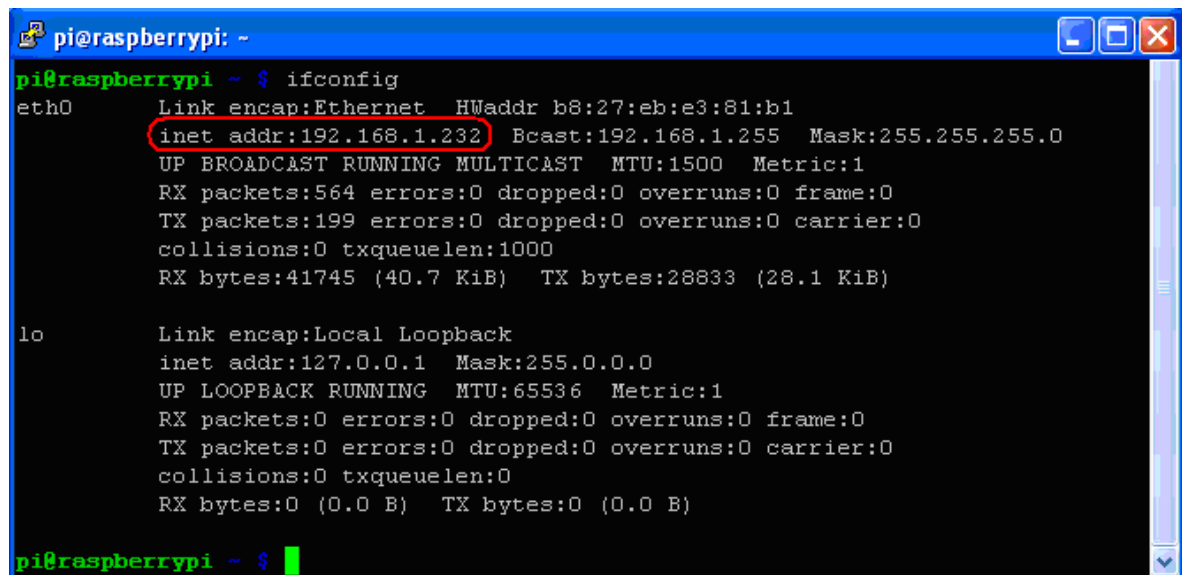
- Запуск Веб-сервера выполняется командой

```
$ sudo /etc/init.d/apache2 start
```

Получение параметров сетевого соединения RPi

Параметры сетевого соединения компьютера Raspberry Pi выводит на экран команда

```
ifconfig
```



```
pi@raspberrypi: ~
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:e3:81:b1
          inet addr:192.168.1.232  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:564 errors:0 dropped:0 overruns:0 frame:0
          TX packets:199 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:41745 (40.7 KiB)  TX bytes:28833 (28.1 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  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:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

pi@raspberrypi ~ $ █
```

IP адрес Ethernet соединения находится в секции **eth0**. В этом примере IP адрес: 192.168.1.232. Если используется беспроводное соединение (Wi-Fi) его адрес можно найти в секции **'wlan0'**.

Установка **root** прав пользователя

Команда начинающаяся словом **sudo** выполняется как **root** команда. Установка **root** прав пользователю на весь сеанс работы с компьютером выполняется в следующем порядке.

1. Откройте удаленный терминал, например, **PuTTY**.
2. Установите соединение с RPi. Введите login: **pi** и password: **raspberrypi**

3. Введите команды

```
sudo -i  
passwd root
```

затем дважды введите новый пароль для root пользователя

4. Перезагрузите компьютер RPi

```
reboot
```

Примечание: Можно перейти к пользователю «root» сразу вместе со всеми правами, не выходя из «pi» введя команду `su - root` и пароль для «root».

5. Снова откройте удаленный терминал PuTTY и введите login: **root** и его новый password.

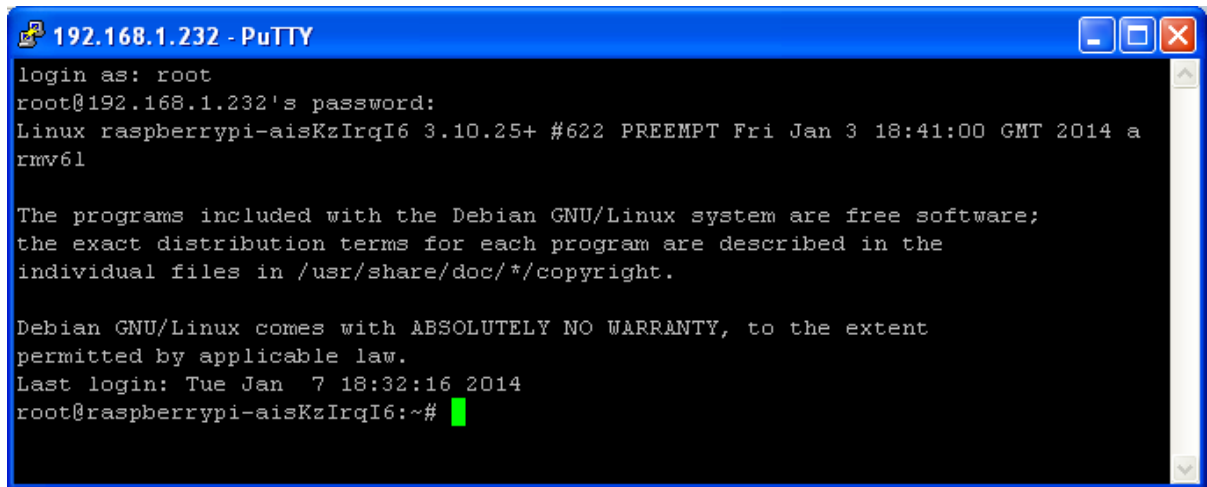


Рис. 1. Вид интерфейса пользователя с Root правами.

Внимание! Для минимизации случайных повреждений операционной системы и ее настроек не используйте права **root** без необходимости.

6. Получив права «root» пользователя обновите список всех пакетов из репозитория: `apt-get update` и обновите пакеты: `apt-get upgrade`.

Чтобы команды `sudo` конкретного пользователя не запрашивали пароль при их запуске нужно отредактировать файл `sudoers`.

Для этого, после строки файла

```
root ALL=(ALL) ALL
```

необходимо добавить

```
username ALL=(ALL) NOPASSWD: ALL
```

Например, для пользователя `pi` это добавление будет выглядеть так

```
pi ALL=(ALL) NOPASSWD: ALL
```

Установка диспетчера задач

Для мониторинга загруженности ресурсов операционной системы полезно установить диспетчер задач. Установка запускается командой

```
sudo apt-get install htop
```

После установки диспетчера можно вызвать командой

```
htop
```

Выход: F10.

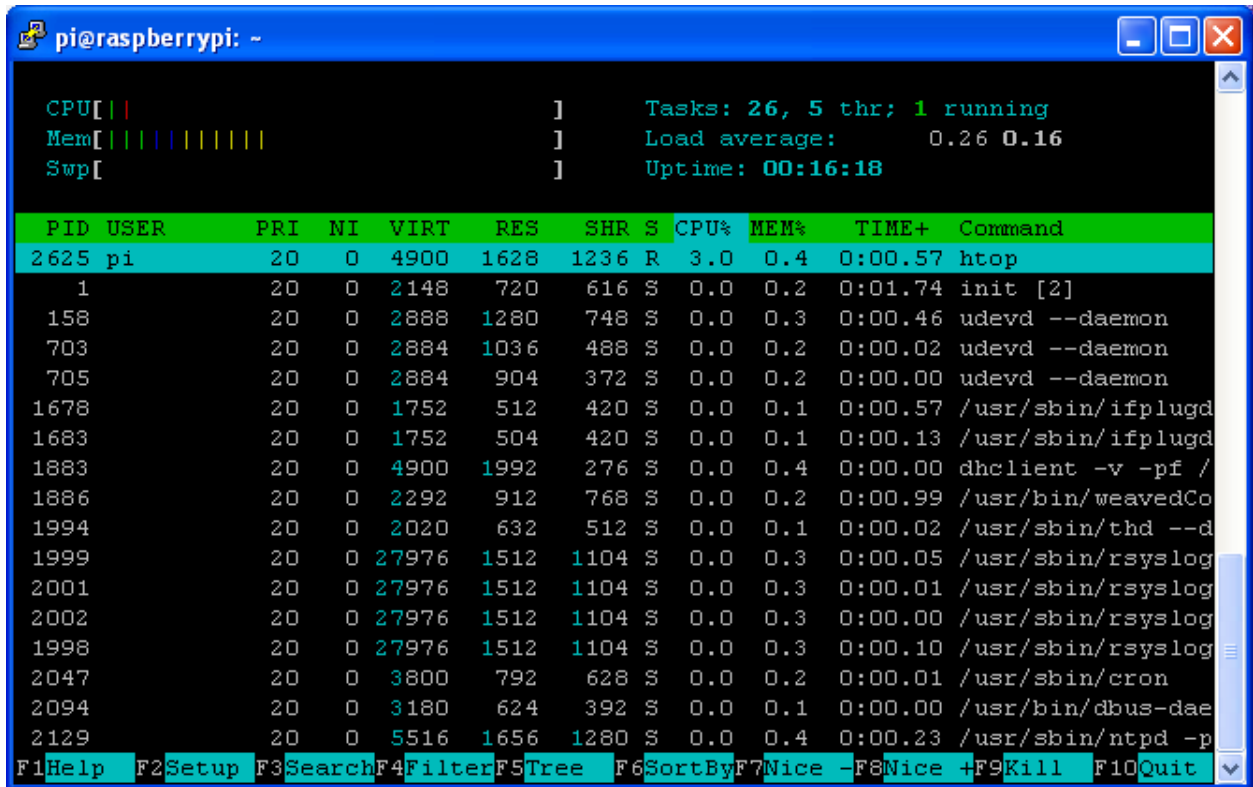


Рис. 2. Интерфейс диспетчера задач.

Установка менеджера файлов

Установка менеджера файлов MC (Midnight Commander) для поиска, редактирования, просмотра и других важных манипуляций с файловой системой выполняется командой

```
sudo apt-get install mc
```

Вызов менеджера файлов осуществляется командой

```
mc
```

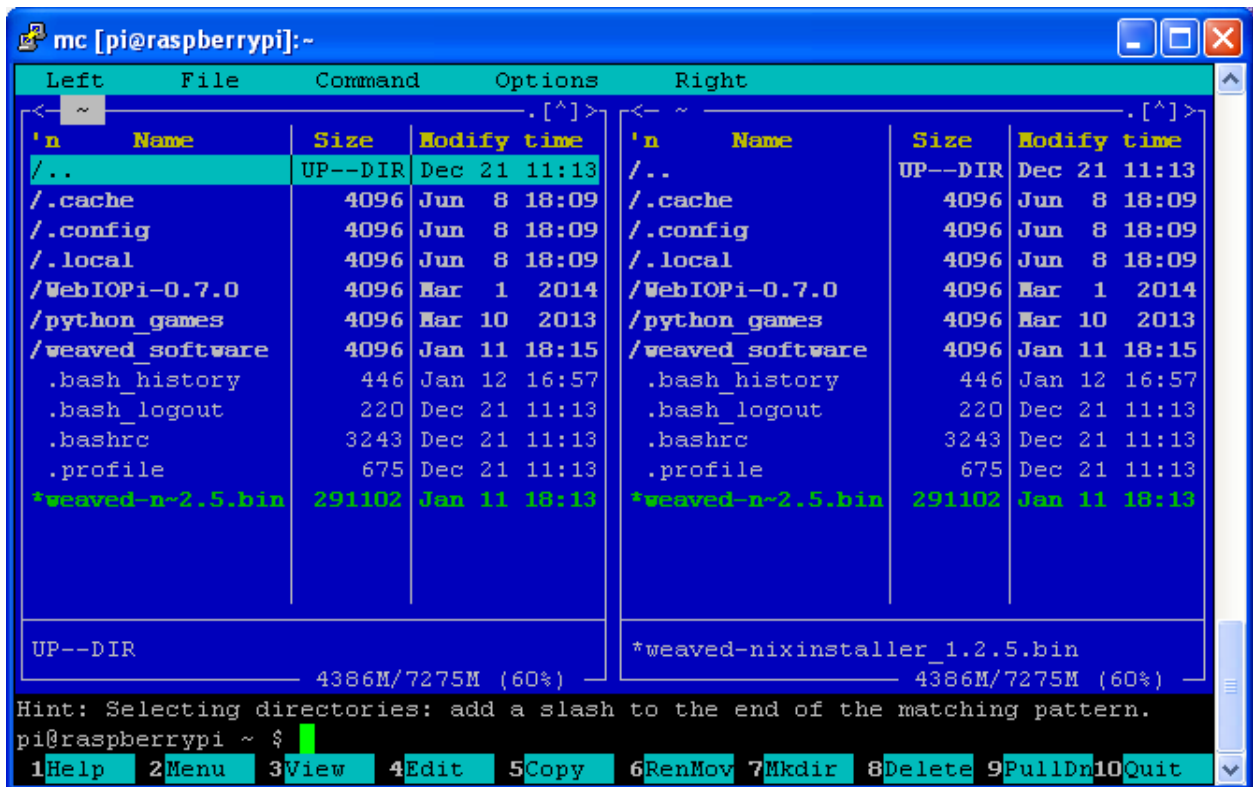


Рис. 3. Интерфейс менеджера файлов.

Выход: F10

Установка утилит доступа к GPIO

WiringPi – это библиотека, которая содержит утилиты для простого доступа к GPIO. Для управления Wiring Pi необходимо установить систему управления исходным кодом Git используя следующие команды.

```
sudo apt-get install git-core
sudo git clone git://git.drogon.net/wiringPi
```

далее, перейдя в папку Wiring Pi

```
cd wiringPi
```

произведите сборку пакета

```
sudo ./build
```

Пакет установлен.

Чтобы это проверить, введите команду запроса версии

```
gpio -v
```

```

pi@raspberrypi: ~/wiringPi
pi@raspberrypi ~/wiringPi $ sudo gpio -v
gpio version: 2.26
Copyright (c) 2012-2015 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
  Type: Model B, Revision: 2, Memory: 512MB, Maker: Sony
pi@raspberrypi ~/wiringPi $ █

```

Рис. 4. Проверка наличия утилит доступа к GPIO.

Номера и состояние контактов GPIO можно распечатать командой

`gpio readall`

```

pi@raspberrypi: ~/wiringPi
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
  Type: Model B, Revision: 2, Memory: 512MB, Maker: Sony
pi@raspberrypi ~/wiringPi $ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |   3.3v   |      |  |    1    | 2 |      |   5v     |     |     |
|  2  |  8  | SDA.1    | IN   | 1 |    3    | 4 |      |   5V     |     |     |
|  3  |  9  | SCL.1    | IN   | 1 |    5    | 6 |      |   0v     |     |     |
|  4  |  7  | GPIO. 7  | IN   | 1 |    7    | 8 |  1  | ALTO TxD | 15  | 14  |
|     |     |   0v     |      |  |    9    |10 |  1  | ALTO RxD | 16  | 15  |
| 17  |  0  | GPIO. 0  | IN   | 0 |   11   |12 |  0  | IN  GPIO. 1 | 1  | 18  |
| 27  |  2  | GPIO. 2  | IN   | 0 |   13   |14 |      |   0v     |     |     |
| 22  |  3  | GPIO. 3  | IN   | 0 |   15   |16 |  0  | IN  GPIO. 4 | 4  | 23  |
|     |     |   3.3v   |      |  |   17   |18 |  0  | IN  GPIO. 5 | 5  | 24  |
| 10  | 12  | MOSI     | IN   | 0 |   19   |20 |      |   0v     |     |     |
|  9  | 13  | MISO     | IN   | 0 |   21   |22 |  0  | IN  GPIO. 6 | 6  | 25  |
| 11  | 14  | SCLK     | IN   | 0 |   23   |24 |  1  | IN  CEO    |10  |  8  |
|     |     |   0v     |      |  |   25   |26 |  1  | IN  CE1    |11  |  7  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 28  | 17  | GPIO.17  | IN   | 0 |   51   |52 |  0  | IN  GPIO.18 |18  | 29  |
| 30  | 19  | GPIO.19  | IN   | 0 |   53   |54 |  0  | IN  GPIO.20 |20  | 31  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
pi@raspberrypi ~/wiringPi $ █

```

Рис. 5. Номера и состояние контактов GPIO.

Атрибуты файла (директории) и их установка

Атрибуты файла, количество ссылок, принадлежность пользователю, принадлежность группе, размер файла, время последней модификации, имя файла / каталога можно запросить командой

```
ls -al /sys/devices/virtual/gpio/gpio24
```

Атрибуты файла имеют следующие обозначения:

```
r = читаемый  
w = записываемый  
x = загружаемый
```

гwx права устанавливаются для владельца файлов (**Owner**), для пользователей входящих в группу (**Group**) и других (**Other**) пользователей, например,

```
-rwxr-xr-x
```

означает, что файл является загружаемым для всех пользователей, Вы можете писать и читать, а пользователи группы могут только читать.

Атрибуты файла устанавливаются командой **chmod**, например,

```
sudo chmod 755 hello.py
```

где каждый знак числа устанавливает **rwX** атрибуты в двоичном коде для **Owner**, **Group** и **Other** пользователей, например, 7 устанавливает **rwX** атрибуты, 6– **rw_**, а 5– **r_x** и т.д.

Автоматический запуск команд при старте RPi

Unix система управляет состояниями вводов/выводов порта GPIO путём записи соответствующих значений в специальные файлы выводов. По дефолту этих файлов в RPi нет, и для чтения/записи порта необходимо сначала создать такой файл, затем установить направление передачи данных и, в соответствии с установленным режимом, записывать или считывать данные. Созданные во время сессии файлы автоматически удаляются после перезагрузки. Чтобы установить связь с выводами GPIO в процессе загрузки операционной системы соответствующие команды необходимо включить в файл автозагрузки **/etc/rc.local** .

Ниже дан пример включения/выключения светодиода подключенного к выводу GPIO 24 (18-й физический вывод порта GPIO).

1. Программой putty.exe удаленного компьютера установите связь с RPi через терминал.

2. Откройте файл редактором nano

```
sudo nano /etc/rc.local
```

3. Для автоматического создания файла вывода GPIO24 и его настройки на режим вывода добавьте следующие строки в rc/local.

```
echo 24 > /sys/class/gpio/export  
echo out > /sys/class/gpio/gpio24/direction
```

Примечание. Последняя команда задает направление передачи данных (на вывод данных) и устанавливает логический ноль (низкое напряжение) на выводе.

4. Перезагрузите систему.

```
sudo reboot
```

После перезагрузки системы должен включиться светодиод.

5. Записывая логическую единицу / ноль на вывод GPIO24 наблюдайте за состоянием светодиода, подключенного к 18-му контакту.

```
echo 1 > /sys/class/gpio/gpio24/value  
echo 0 > /sys/class/gpio/gpio24/value
```

Если установлена wiringPI, то вместо echo 1 и echo 2 можно использовать следующие команды для записи в файл /sys/class/gpio/gpio24:

```
$ gpio write 5 1  
$ gpio write 5 0
```

Примечание: 5-й вывод по wiring нумерации соответствует 24 выводу по BCM нумерации.

Структура html файла Веб-сервера для связи с GPIO через браузер

Варианты HTML запросов для связи с GPIO портом компьютера Raspberry Pi представлены на следующем рисунке. Передача данных шлюзу осуществляется в формате

```
имя=значение&имя1=значение1&...
```

Здесь "имя" это название параметра, а "значение" его содержимое. Два метода GET и POST используются для передачи данных в таком формате. При использовании метода GET данные передаются серверу вместе с URL:

```
http://.../cgi-bin/test.cgi?имя=значение&имя1=значение1&...
```

При использовании метода POST данные посылаются внутри самого HTTP запроса.

Использование метода GET с получением данных из переменной окружения намного легче, чем считывание данных из потока при использовании метода POST.

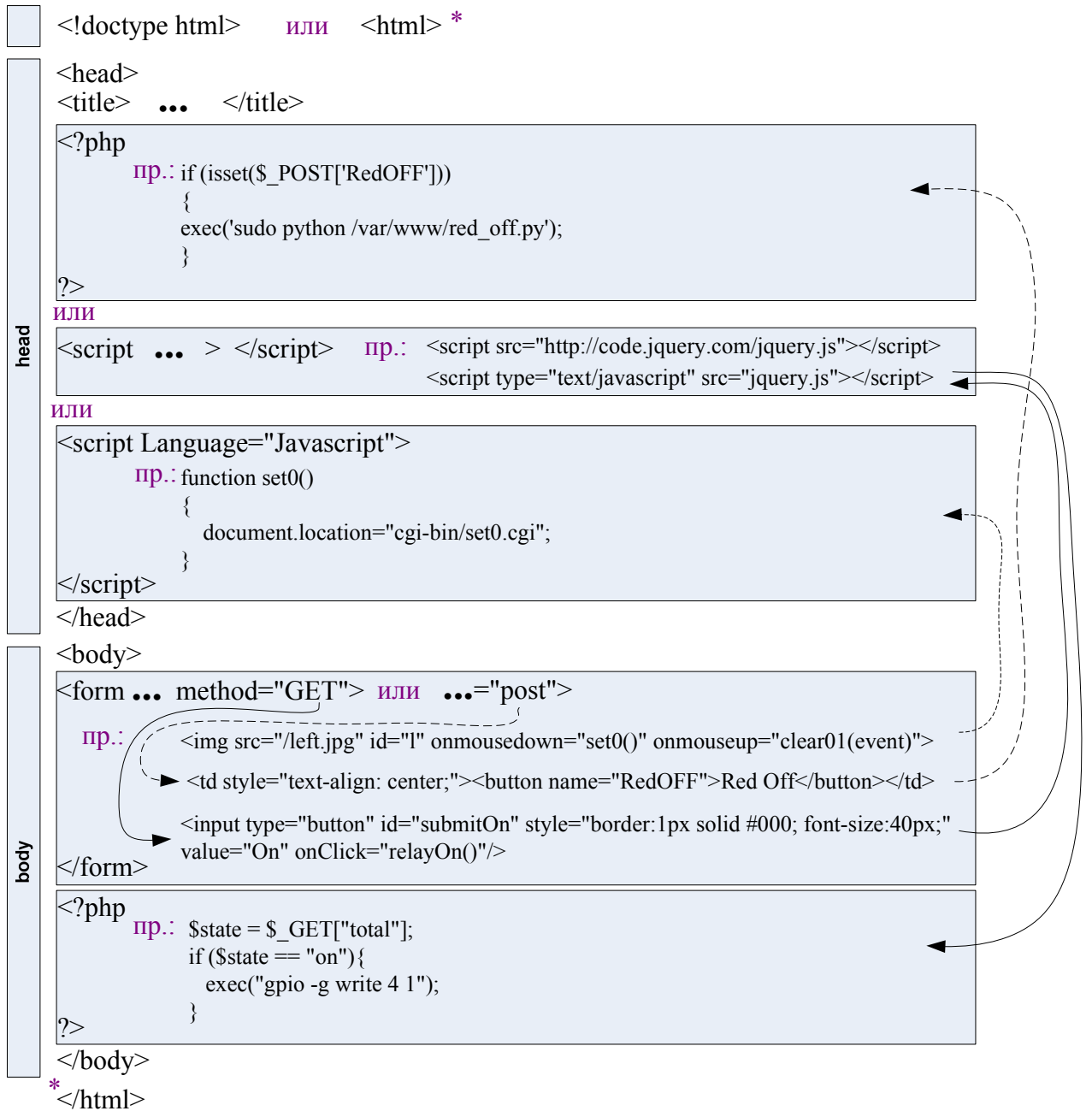


Рис. 6. Варианты структур HTML файла.

cat: /sys/class/gpio/gpio24/value | sed -n 1: No such file or directory

cat: s/(.*)/1/p: No such file or directory

sed: -e expression #1, char 1: missing command

CGI скрипты для Web соединений

CGI является стандартом интерфейса, который служит для связи внешней программы с веб-сервером. CGI скрипты, как правило, помещают в директорию `/usr/lib/cgi-bin/` сервера.

CGI протокол разработан таким образом, чтобы можно было использовать любой язык программирования, который может работать со стандартными устройствами ввода/вывода. Скрипт можно выполнить и в виде командного файла. CGI скрипты в Bash выполняются также как и скрипты интерпретатора Python. В программе должен быть указатель на то, какой интерпретатор должен выполнять алгоритм и заголовок, указывающий на тип выводимых данных содержимого, предназначенного для Веб-браузера.

Скрипты должны иметь атрибут загружаемого файла. Атрибуты устанавливаются командой `chmod`. Скрипты выполняются когда их запрашивает браузер. Все выводимое на печать командой `print` отображается в браузере.

Для запуска CGI (Common Gateway Interface) скриптов [1] Веб-браузером и их выполнения интерпретатором команд `bash` необходимо установить опционные компоненты сервера Apache2

```
sudo apt-get install apache2
```

Для проверки связи CGI скриптов с Веб-браузером создайте в каталоге `/usr/lib/cgi-bin` редактором `nano` следующий пример CGI файла `hello.cgi` :

```
#!/bin/bash
echo -e "Content-type: text/html\n\n"
echo "<h1>Hello World</h1>"
```

Первая строка сообщает, что файл должен быть выполнен интерпретатором `bash`. Эта строка не передается браузеру, запрашиваемому эту страницу. Вторая строка – HTTP заголовок который заносится в `stdout` и, таким образом, передается браузеру. Этот заголовок сообщает браузеру, что следующая строка должна быть интерпретирована как HTML код. Заголовок заканчивается пустой строкой создаваемой двумя `\n` или строкой `echo`.

Установите атрибут загрузки `cgi` файлу.

```
sudo chmod +x /usr/lib/cgi-bin/hello.cgi
```

Введите на удаленном браузере адрес `hello.cgi` RPi файла.

```
http://192.168.1.232/cgi-bin/hello.cgi
```

В ответ браузер выдает сообщение `cgi` файла.



Примечание: Список ошибок выполнения скрипта запускаемого Веб-браузером можно найти в файле `/var/log/apache2/error.log`.

Недостатком CGI технологии является запуск веб-сервером отдельной программы для обработки запроса при каждом обращении к скрипту, что требует достаточно много ресурсов системы и процессорного времени.

Подключение Python программ к Веб-браузеру в CGI формате

Для того, чтобы выполнять скрипты Python в Apache2 необходимо обновить его настройки.

По дефолту Python сконфигурирован на выполнение скриптов с расширением .cgi. Для того чтобы Apache выполнял .py файлы необходимо отредактировать конфигурационный файл.

```
sudo nano /etc/apache2/sites-enabled/000-default
```

Добавьте строку

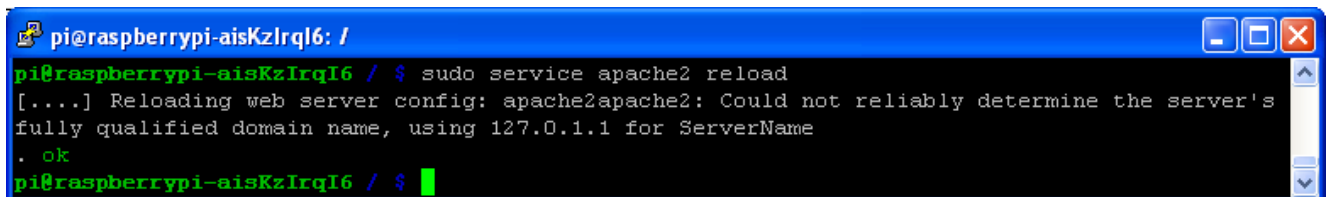
```
AddHandler cgi-script .py
```

В секцию Directory файла **000-default**:

```
<Directory "/usr/lib/cgi-bin">
...
</Directory>
```

Перезагрузите конфигурационный файл Веб-сервера Apache:

```
sudo service apache2 reload
```



```
pi@raspberrypi-aisKzIrqI6: /
pi@raspberrypi-aisKzIrqI6 / $ sudo service apache2 reload
[...] Reloading web server config: apache2apache2: Could not reliably determine the server's
fully qualified domain name, using 127.0.1.1 for ServerName
. ok
pi@raspberrypi-aisKzIrqI6 / $
```

Для проверки связи Python программ с Веб-браузером создайте редактором **nano** в разделе **/usr/lib/cgi-bin** файл **hello_py.cgi** включающий импортирование модулей **cgi** and **cgitb**.

```
#!/usr/bin/env python

import cgi
import cgitb

cgitb.enable()

print "Content-type: text/html\n\n"

print "<h1>Hello World</h1>"
```

Установите атрибут загрузки файлу **hello_py.cgi**.

```
$ sudo chmod +x hello_py.cgi
```

Введите на удаленном браузере адрес **hello_py.cgi** RPi файла.

http://192.168.1.232/cgi-bin/hello_py.cgi

В ответ браузер выдает следующее сообщение cgi файла.



А теперь, создайте в разделе `/usr/lib/cgi-bin` файл `hello.py`

```
#!/usr/bin/env python

print "Content-type: text/html\n\n"
print "<h1>Hello World</h1>"
```

Установите атрибут загружаемый файл.

```
$ sudo chmod +x hello.py
```

И проверьте связь с Python программой через удаленный браузер.

<http://192.168.1.232/cgi-bin/hello.py>

Запуск HTML файлов RPi с удаленного Веб-клиента

Файл `index.html` каталога `/var/www/` запускается с удаленного браузера после ввода IP адреса RPi, например,

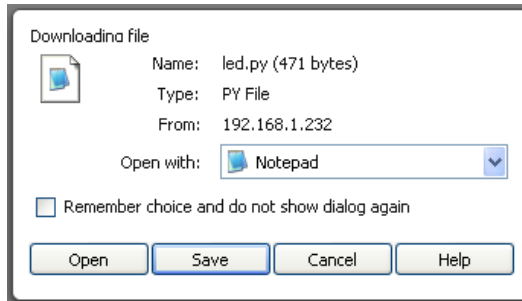
<http://192.168.1.232>

Для запуска других html страниц каталога `/var/www/` необходимо адрес RPi дополнить именем html файла, например,

http://192.168.1.232/index_2.html

Считывание скриптов RPi через Веб-браузер

Для просмотра или копирования файлов каталога `/var/www/` необходимо Веб-клиенту указать адрес RPi и имя файла, например, <http://192.168.1.232/pin.py>



Сообщение об ошибках Веб-браузера

Список ошибок Web соединений в </var/log/apache2/error.log>. Вот пример распечатки содержимого </var/log/apache2/error.log>:

```
nano /var/log/apache2/error.log
```

```
[Wed Apr 30 12:02:56 2014] [notice] Apache/2.2.22 (Debian) PHP/5.4.4-14+deb7u8 configured -- resuming normal operations
[Wed Apr 30 12:29:29 2014] [error] [client 192.168.1.59] PHP Notice: Use of undefined constant php - assumed 'php' in /var/www/remote_relay.php on
line 16, referer: http://192.168.1.232/remote_relay.php?total=off
[Wed Apr 30 13:14:45 2014] [error] [client 192.168.1.59] PHP Notice: Undefined index: total in /var/www/remote_control.php on line 18
[Wed Apr 30 13:14:52 2014] [error] [client 192.168.1.59][Wed Apr 30 13:23:37 2014] [notice] caught SIGTERM, shutting down
[Wed Apr 30 13:23:39 2014] [notice] Apache/2.2.22 (Debian) PHP/5.4.4-14+deb7u8 configured -- resuming normal operations
[Wed Apr 30 13:23:55 2014] [error] [client 192.168.1.59] PHP Notice: Use of undefined constant php - assumed 'php' in /var/www/remote_led.php on
line 15
[Wed Apr 30 13:24:51 2014] [notice] Apache/2.2.22 (Debian) PHP/5.4.4-14+deb7u8 configured -- resuming normal operations
[Wed Apr 30 23:06:52 2014] [error] [client 192.168.1.59] File does not exist: /var/www/favicon.ico, referer: http://192.168.1.232/remote_led.php
[Wed Apr 30 23:24:05 2014] [error] [client 192.168.1.59] PHP Notice: Undefined index: total in /var/www/remote_relay.php on line 18
[Wed Apr 30 23:25:29 2014] [notice] caught SIGTERM, shutting down
[Wed Apr 30 23:27:30 2014] [error] [client 192.168.1.59] PHP Notice: Use of undefined constant php - assumed 'php' in /var/www/remote_relay.php on
line 16
[Wed Apr 30 23:32:24 2014] [notice] Apache/2.2.22 (Debian) PHP/5.4.4-14+deb7u8 configured -- resuming normal operations
[Wed Apr 30 23:33:44 2014] [notice] caught SIGTERM, shutting down
[Wed Apr 30 23:33:56 2014] [error] [client 192.168.1.59] PHP Notice: Undefined index: total in /var/www/remote_relay.php on line 18
```

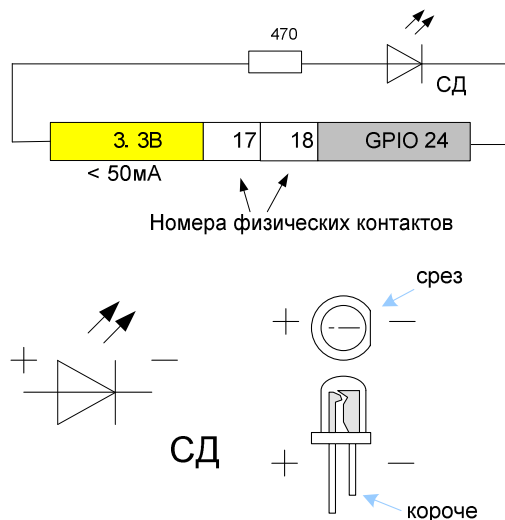
ПРИМЕРЫ ПОЛУЧЕНИЯ ПРОВЕРЕННЫХ РЕЗУЛЬТАТОВ И ВАРИАНТЫ ДЛЯ САМОКОНТРОЛЯ

Для взаимодействия с компьютером Raspberry Pi через веб-интерфейс в заданиях используются следующие языки программирования.

- HTML (Hyper Text Markup Language) – язык гипертекстовой разметки для размещения всех необходимых элементов (блоков) на странице;
- CSS (Cascading Style Sheets) – каскадные таблицы стилей для описания документа, написанного с использованием языка разметки HTML.
- Язык программирования PHP для построения функционала работы с компьютером Raspberry Pi.

Задание 1. Управление светодиодом подключенного к выводу порта GPIO RPi через браузер удаленного компьютера с применением языков программирования PHP и Java Script.

1. Подключите к порту GPIO компьютера Raspberry Pi светодиод как показано на рисунке.
ВНИМАНИЕ: Компьютер должен быть обесточен.



2. Подключите RPi к локальной сети как показано на рисунке.

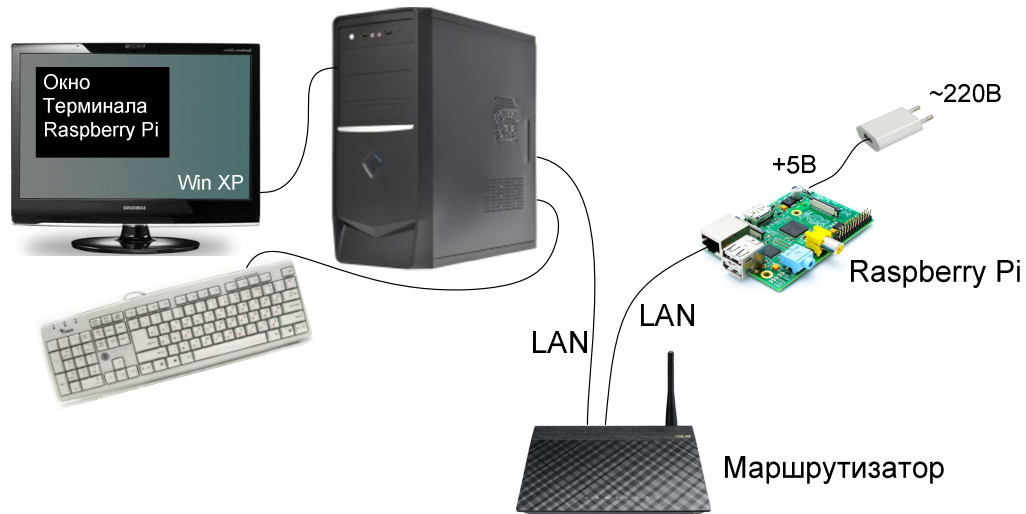
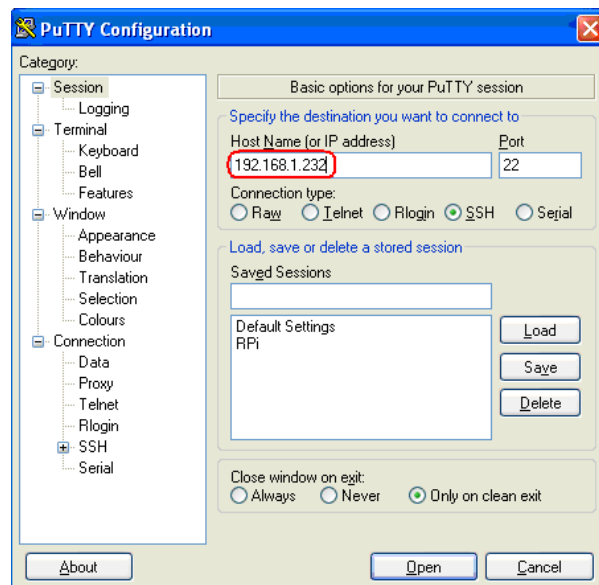


Рис. 7. Подключение основного компьютера и Raspberry Pi к локальной сети через маршрутизатор с SSH сервером. Связь пользователя с RPi поддерживается через программный терминал основного компьютера.

3. Включите Raspberry Pi и основной компьютер.
4. Программой `ipscan23.exe` определите IP адрес RPi в локальной сети. В нашем примере у Raspberry Pi IP адрес 192.168.1.232
5. Запустите терминал `putty.exe` на основном компьютере.



6. Через терминальное окно установите связь с RPi (login: `pi`, password: `raspberrypi`).


```
pi@raspberrypi-aisKzIrlqI6: -
login as: pi
pi@192.168.1.232's password:
Linux raspberrypi-aisKzIrlqI6 3.10.25+ #622 PREEMPT Fri Jan 3 18:41:00 GMT 2014 a
rmv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 30 11:43:12 2014 from 192.168.1.59
pi@raspberrypi-aisKzIrlqI6 ~ $
```

- Установите автоматическое создание файла вывода GPIO24 для работы в режиме цифрового вывода, для этого в файл автозагрузки `/etc/rc.local` добавьте следующие строки.

```
echo 24 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio24/direction
```

```
GNU nano 2.2.6 File: /etc/rc.local Modified
# Speak IP address through default audio device
espeak -s 100 -p 82 --stdout "My IP address is `hostname -I`" | aplay -f $
fi
echo 24 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio24/direction
#chmod 777 /sys/class/gpio/gpio24/value
exit 0
^G Get Help ^C WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

- Перезагрузите систему
`$ sudo reboot`
- Вновь программой `putty.exe` установите соединение с RPi через терминал (login: `pi`, password: `raspberrypi`).
- Проверьте включение/выключение светодиода командами библиотеки `wiringPI`
`$ gpio write 5 1`
`$ gpio write 5 0`

Примечание: при необходимости библиотека `WiringPi` в каталог `/home/pi/` устанавливается следующими командами [2].

```
$ sudo apt-get install python-dev python-setuptools
$ git clone https://github.com/WiringPi/WiringPi-Python.git
```

```
$ cd WiringPi-Python
$ git submodule update --init
$ sudo python setup.py install /home/pi/WiringPi-Python
```

Внимание! В библиотеке WiringPi используется своя нумерация выводов GPIO (см. рис). Так, вывод GPIO24 (BCM нумерация), физический контакт GPIO N18 (BOARD нумерация) имеет номер 5 по версии WiringPi.

11. Перейдите в каталог сервера RPi

```
$ cd /var/www
```

12. В редакторе nano создайте следующий файл remote_led.php, включающий HTML форму с

двумя кнопками и и скрытой (hidden) переменной “total” отражающей состояние кнопок интерфейса. Состояние светодиода, подключенного к 5 выводу GPIO (по нумерации библиотеки WiringPi), должно соответствовать текущему значению переменной “total”, а значит, и нажатию на соответствующую кнопку интерфейса. Управление светодиодом через вывод GPIO выполняет PHP скрипт находящийся в границах `<?php> ... ?>` файла remote_led.php.

```
$ sudo nano remote_led.php
```

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Remote control</title>
  <script type="text/javascript" src="script.js"></script>
</head>
<body>

<form id="relay" action="remote_led.php" method="GET">
  <input type="button" id="submitOn" style="border:1px solid #000; font-size:40px;"
value="On" $
  <input type="button" id="submitOff" style="border:1px solid #000; font-size:40px;"
value="Off"$
  <input type="hidden" name="total" id="total" value="">
</form>

<?php>

  $state = $_GET["total"];

  if ($state == "on"){
    exec("gpio write 5 0");
  }

  if ($state == "off"){
```

```
        exec("gpio write 5 1");
    }
?>

</body>
```

13. Создайте в этом же каталоге файл Java Script – обработчик состояния кнопок который выдаёт PHP файлу состояние "on" или "off" через переменную "total".

```
$ sudo nano script.js
```

```
window.onload = function(){
    buttonOn = document.getElementById('submitOn');
    buttonOn.onClick = relayOn;

    buttonOff = document.getElementById('submitOff');
    buttonOff.onClick = relayOff;
}

function relayOn(){

    hidden = document.getElementById("total");
    hidden.value = "on";

    form = document.getElementById("relay");
    form.method = "GET";
    form.action = "remote_led.php";
    form.submit();
}

function relayOff(){

    hidden = document.getElementById("total");
    hidden.value = "off";

    form = document.getElementById("relay");
    form.method = "GET";
    form.action = "remote_led.php";
    form.submit();
}
```

14. Запустите удаленный браузер и свяжитесь с программой контроля светодиода сервера RPi Apache.

Введите http://192.168.1.232/remote_led.php

Примечание: При необходимости сделайте рестарт сервера RPi Apache.

```
sudo service apache2 restart
```

Используйте браузер **Mozilla Firefox**

Не используйте браузер **Opera**, который не может установить соединение.

15. Проверьте связь светодиода с браузером удаленного компьютера через сервер RPi **Apache**.



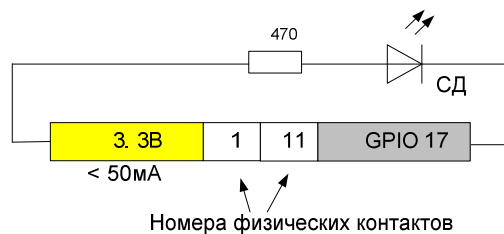
Рис. 8. Смартфон с интерфейсом для управления через беспроводную сеть светодиодом, подключенного к порту GPIO компьютера Raspberry Pi. Такое же соединение можно организовать и с удаленного компьютера, подключенного к сети с Raspberry Pi.

16. Завершите сеанс работы с Raspberry Pi командой

```
sudo shutdown now -h
```

Задание 2. Управление светодиодом подключенного к выводу порта GPIO RPi через браузер удаленного компьютера с применением языков программирования PHP и CSS [7].

1. Подключите светодиод к другим контактам порта GPIO компьютера Raspberry Pi как показано на рисунке. **ВНИМАНИЕ:** Компьютер должен быть обесточен.



2. Включите Raspberry Pi и основной компьютер.

3. Установите связь с Raspberry Pi через терминал Putty.
4. Убедитесь в том, что выполнен раздел “Установка утилит доступа к GPIO” главы ОБЩИЕ СВЕДЕНИЯ.
5. Проверьте работу светодиода.

Введите команду

```
gpio mode 0 out
```

Светодиод должен включиться. Выключите светодиод командой

```
gpio mode 0 in
```

6. В директории `/var/www` удалите html файл: `index.html` и создайте следующий новый файл с расширением `.php`: `index.php`:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Главная</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <?php
      phpinfo();
    ?>
  </body>
</html>
```

7. Установите сетевое соединение с Raspberry Pi (IP адрес, например, 192.168.1.232) через браузер компьютера локальной сети.

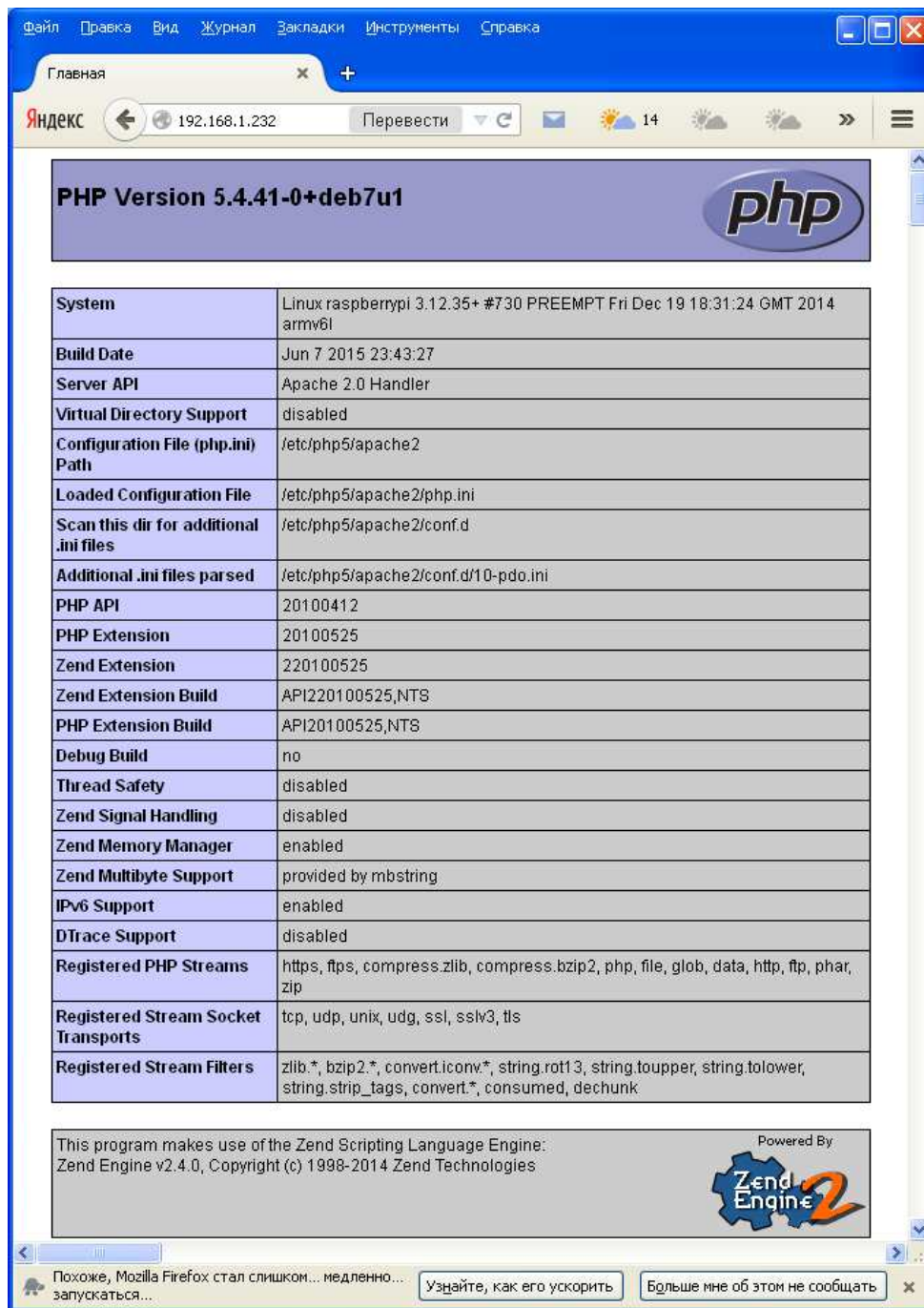


Рис. 9. Результат успешного выполнения PHP кода в файле `index.php`. Веб-сервер функционирует.

- Разработайте интерфейс при помощи языков HTML и CSS, а также запрограммируйте интерфейс, написав обработчик, который будет работать напрямую с GPIO реагируя на запросы пользователя.

Пример файла **index.php**:

```
<!-- Обработчик формы -->
<?php
if(isset($_POST["on"])) {
```

```

    $val=trim(@shell_exec("/usr/local/bin/gpio mode 0 out"));
    $i="<span id='true'>Светодиод включен</span>";
} else if(isset($_POST["off"])) {
    $val=trim(@shell_exec("/usr/local/bin/gpio mode 0 in"));
    $i="<span id='false'>Светодиод выключен</span>";
}
?>
<!DOCTYPE html>
<html>
  <head>
    <title>Главная</title>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/styles.css" type="text/css" />
  </head>
  <body>
    <!-- Оболочка -->
    <fieldset id="main">
      <legend>GPIO Control</legend>
      <form method="post" action="index.php">
        <input type="submit" name="on" value="Включить" />
        <input type="submit" name="off" value="Выключить" />
        <?php
          echo $i;
        ?>
        <p class="copyright">Copyright &copy; 2015 Мазуров Денис</p>
      </form>
    </fieldset>
  </body>
</html>

```

Пример файла **styles.css**:

```

* {
  box-sizing:border-box;
  outline:none;
}
#main {
  border:1px solid black;
  margin:200px auto;
  width:235px;
  height:120px;
}
input {
  width:100px;
  height:35px;
  margin-bottom:10px;
}
input:hover {
  background-color:rgba(217, 217, 217, 0.27);
  cursor:pointer;
  box-shadow:0 0 5px #000000;
}
#true {
  background-color:rgba(197, 228, 146, 0.71);
  margin-top:10px;
  padding:5px;
  border-radius:5px;
}

```

```
#false {  
    background-color:rgba(226, 119, 119, 0.71);  
    margin-top:10px;  
    padding:5px;  
    border-radius:5px;  
}  
.copyright {  
    font-size:12px;  
    float:right;  
    margin-top:15px;  
}
```

9. Через браузер компьютера локальной сети установите соединение с Raspberry Pi.

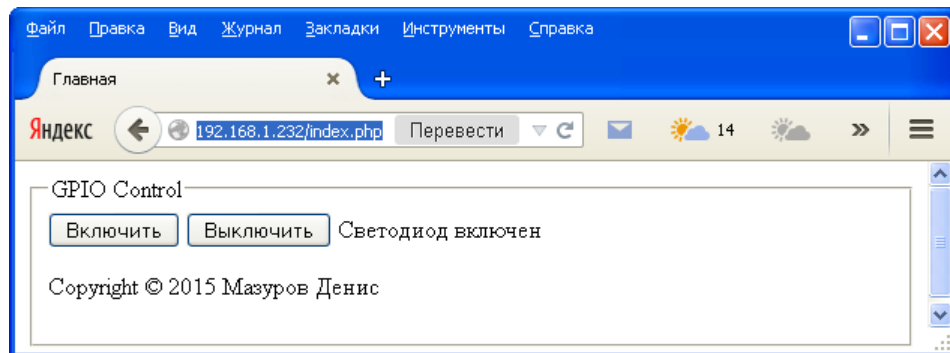


Рис. 10. Демонстрация успешной сетевой связи с Raspberry Pi.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какими средствами можно организовать удаленное управление?
2. Почему с удаленного компьютера нельзя непосредственно вводить команды порта GPIO Raspberry Pi для изменения состояния подключенного к порту светодиода?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Википедия. Common Gateway Interface (CGI)
http://en.wikipedia.org/wiki/Common_Gateway_Interface
2. Установка библиотеки WiringPi - доступа к выводам GPIO, включая аппаратный ШИМ.
<https://github.com/WiringPi/WiringPi-Python>
3. Dr. Bob Davidov. Компьютерные средства систем управления. Raspberry Pi
<http://portalnp.ru/2013/12/1691>
4. Dr. Bob Davidov. Подключение периферии к среде разработки систем управления MatLAB
<http://portalnp.ru/2014/03/1783>
5. Dr. Bob Davidov. Импорт и экспорт MatLAB данных через Raspberry Pi
<http://portalnp.ru/2014/04/1858>
6. Dr. Bob Davidov. Построение RT системы управления на базе компьютера Raspberry Pi
<http://portalnp.ru/2014/04/1867>
7. Мазуров Д.Н. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА к практической работе по дисциплине: «Компьютерные технологии управления в технических системах» на тему: «Удалённое управление через Web-сервер».
8. Dr. Bob Davidov. Компьютерные технологии управления в технических системах
<http://portalnp.ru/author/bobdavidov>

ИСПРАВЛЕНИЯ И ДОПОЛНЕНИЯ В СООТВЕТСТВУЮЩИХ РАЗДЕЛАХ (ЯНВАРЬ 2018 Г)

Установка HTTP Веб-сервера Apache на RPi

- файлы сайта хранятся не в

`/var/www/index.html,`

а в

`/var/www/html/index.html.`

Соответственно, файлы php тоже нужно размещать в директории

`/var/www/html/`

Автоматический запуск команд при старте RPi

- команда

```
gpio write 5 1
```

не работает, нужно добавить ключ

```
gpio -g write 5 1
```

CGI скрипты для Web соединений

Для запуска CGI (Common Gateway Interface) скриптов [1] Веб-браузером и их выполнения интерпретатором команд `bash` необходимо установить опционные компоненты сервера Apache2

```
sudo apt-get install apache2
```

Для проверки связи CGI скриптов с Веб-браузером создайте в каталоге

```
/usr/lib/cgi-bin
```

редактором nano следующий пример CGI файла `hello.cgi`:

```
#!/bin/bash
```

```
echo -e "Content-type: text/html\n\n"
```

```
echo "<h1>Hello World</h1>"
```

Установите атрибут загрузки cgi файлу.

```
sudo chmod +x /usr/lib/cgi-bin/hello.cgi
```

Введите на удаленном браузере адрес `hello.cgi` RPi файла.

```
http://192.168.1.232/cgi-bin/hello.cgi
```

В ответ браузер выдает сообщение cgi файла.

```
HELLO WORD
```

Браузер выдает ошибку, что страница не найдена. В Ubuntu мануале можно найти следующее:

Если вы хотите запускать на сервере cgi-скрипты, **подключите модуль cgi командой**

```
sudo a2enmod cgi
```

С этой командой действительно все начинает работать.

ПРИМЕРЫ ПОЛУЧЕНИЯ ПРОВЕРЕННЫХ РЕЗУЛЬТАТОВ И ВАРИАНТЫ ДЛЯ

САМОКОНТРОЛЯ

- при установке WiringPi есть команда

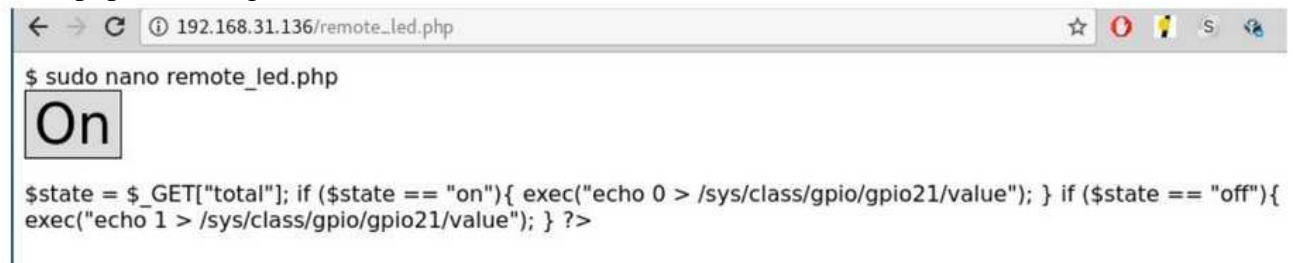
```
git submodule update --init
```

в ней вместо тире должно быть два дефиса:

```
git submodule update --init
```

- снова директорию `/var/www/` нужно изменить на `/var/www/html/`

- сам php код содержит ошибки:



```
$ sudo nano remote_led.php
On
$state = $_GET["total"]; if ($state == "on"){ exec("echo 0 > /sys/class/gpio/gpio21/value"); } if ($state == "off"){
exec("echo 1 > /sys/class/gpio/gpio21/value"); } ?>
```

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Remote control</title>
<script type="text/javascript" src="script.js"></script>
</head>
<body>
<form id="relay" action="remote_led.php" method="GET">
<input type="button" id="submitOn" style="border:1px solid #000; font-size:40px;"
value="On" $ Не хватает > в конце
<input type="button" id="submitOff" style="border:1px solid #000; font-size:40px;"
value="Off$ Не хватает > в конце и закрыть кавычки после off
<input type="hidden" name="total" id="total" value="">
</form>
```

```
<?php> Лишняя >
$state = $_GET["total"];
if ($state == "on"){
exec("gpio write 5 0"); в команду добавить ключ -g, а exec заменить на shell_exec
}
if ($state == "off"){
exec("gpio write 5 1"); в команду добавить ключ -g, а exec заменить на shell_exec
}
?>
</body>
не хватает </html>
```

Примеры программ Олега Медовикова:

1. Минимальный код для работы с простыми кнопками *on* и *off*:

```
<html>
  <?php
    if (isset($_GET['on'])) { shell_exec("gpio -g write 21 1"); }
    if (isset($_GET['off'])) { shell_exec("gpio -g write 21 0"); }
  ?>
  <a href='remote_led.php?on=true'>on</a>
  <a href='remote_led.php?off=true'>off</a>
</html>
```

2. Код с красивыми кнопками, которые привязаны к центру окна:

```
<!DOCTYPE html>
<html lang=en>
  <body>
    <head>
      <meta http-equiv="content-type" content="text/html;charset=utf-8" />
      <meta name="viewport" content="width=device-width, initial-scale=1" />
      <title>Led toggle</title>
    </head>
    <?php
      if (isset($_GET['on'])) { shell_exec("gpio -g write 21 1"); }
      if (isset($_GET['off'])) { shell_exec("gpio -g write 21 0"); }
    ?>
    <style>
      body {
        margin: 0;
        font-size: 200%; }
      #toggle {
        display: table;
        padding: .5rem;
        margin: 20vh auto 0;
        background-color: lightslategray; }
      #toggle span {
        display: block;
        text-align: center;
        color: aliceblue; }
      a {
        display: inline-block;
        padding: .5rem 1rem;
        margin: .5rem;
        background: orange;
        text-decoration: none;
        color: aliceblue;
        border: 1px solid white;
      }
    </style>
    <div id='toggle'>
      <span>Toggle LED</span>
      <a href='remote_led.php?on=true'>on</a>
      <a href='remote_led.php?off=true'>off</a>
    </div>
  </body>
</html>
```