

УДК 004.522

А.А. МЯСИЦЕВ

Хмельницький національний університет

## УПРАВЛЕНИЕ ЧЕРЕЗ ИНТЕРНЕТ ЭЛЕКТРОПИТАНИЕМ И ДАТЧИКАМИ МИНИ КОМПЬЮТЕРОМ RASPBERRY PI

*Разработано устройство для управления через Интернет электропитанием оборудования и получения информации с датчиков. Система управления реализована на базе одноплатного миникомпьютера Raspberry Pi. В качестве программного обеспечения для построения web-сервера, управления выводами порта GPIO, датчиками, для работы со скриптами на Python использован Фреймворк WebIOPi. Разработанное устройство позволяет отслеживать показания датчиков в течении длительного времени, записывая измеряемые значения в файл и графически отображая их. Реализована возможность подключения устройства к сети Интернет без использования реального IP-адреса благодаря использованию сервиса Weaved.*

*Ключевые слова: Raspberry Pi, Фреймворк WebIOPi, Weaved, микроконтроллер AVR, Wiznet W5100, датчик DS18B20.*

А.А. MYASISCHEV

Khmelnysky National University

### MINI COMPUTER OF THE RASPBERRY PI USE FOR CONTROL OF SUPPLY POWER AND SENSORS VIA INTERNET

*The device is created for controlling of power supply of equipment and receiving information from sensors via the Internet. Control system was created on base of mini computer Raspberry Pi. As the software for building of the web-server, for control of the ports of GPIO, for control of the sensors, and for work with scripts in Python is used WebIOPi Framework. The developed device allows you to monitor data with sensors for a long time, create records of the measured values to a file and graphically displaying them. We can to connect this device to the Internet without using a real IP-address by using Weaved service.*

*Keywords: Raspberry Pi, framework WebIOPi, Weaved, microcontroller AVR, Wiznet W5100, DS18B20 sensor.*

#### Постановка задачи

В настоящее время достаточно остро стоит задача дистанционного обслуживания систем электропитания на удаленных объектах. Например, в развивающихся компаниях с небольшой штатной численностью сотрудников, которые вынуждены совмещать две-три специализации. Именно перед ними, в первую очередь возникает задача поиска экономичного решения организации удаленного управления электропитанием. Параллельно этой задаче устройства удаленного управления должны иметь возможность работать с датчиками для анализа состояния удаленного объекта с целью принятия решения или удаленного информирования персонала о состоянии объекта. Например, изменение температуры, влажности, давления на удаленном объекте требует принятия решения обслуживающим персоналом или автоматического срабатывания систем дистанционного обслуживания электропитанием. Эти задачи также возникают при создании "Умного дома", "Умной фирмы", "Умного офиса" и др. Создание систем управления по TCP/IP сети требует построения устойчиво работающего web-сервера, который имеет функции управления. Возможно построение такого сервера на базе микроконтроллеров, например фирмы AVR и контроллера сети Wiznet W5100 с встроенными в него Ethernet и TCP/IP протоколами на аппаратном уровне[1]. Также можно использовать и одноплатные мини компьютеры, работающие под управлением операционной системы Linux. В работе изучена возможность и представлены практические результаты связанные с созданием системы управления электропитанием через Интернет, которая должна решать задачи:

1. Удаленно включать и выключать три электророзетки. Первая розетка служит для подключения мощных потребителей энергии (например электронагревателя), две другие – стандартные бытовые устройства – это лампы накаливания, компьютер, кондиционер и др.

2. Определять температуру помещения, рисовать график изменения температуры для любого интервала времени.

3. Управление устройством и получение информации о температуре должно выполняться удаленно через браузер (Chrome).

4. Должно быть предусмотрено управление устройством и в случае отсутствия у него реального IP-адреса (DNS имени). Необходимо лишь подключение к интернет, например через стандартный ADSL модем с установленным Nat (что в модемах установлено по умолчанию).

5. Интерфейс связи с устройством через браузер должен быть такой, как показано на рисунке 1. При включенной электророзетке кнопка должна изменять цвет с черной на зеленую.

Розетка для подключения мощного источника (обогревателя) должна управляться временем. Необходимо предусмотреть время включения и выключения обогревателя, которые должны вводиться в браузер (рис. 1). На рис. 2 показано фото экспериментального устройства.

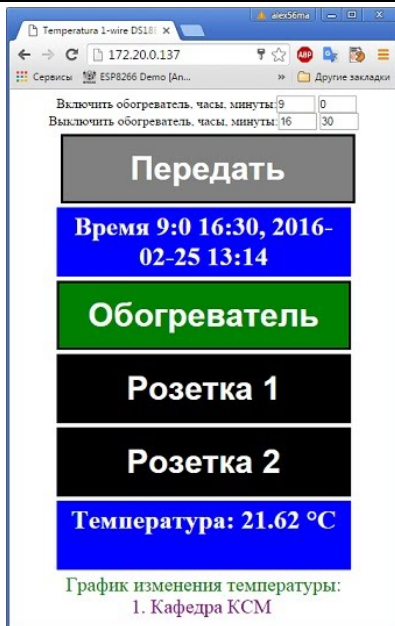


Рис. 1. Интерфейс с Интернет розеткой



Рис. 2. Фото системы управления электропитанием

### Изложение основного материала работы

Рассмотрим удаленное управление устройствами и датчиками через компьютерную сеть (Интернет) с помощью одноплатных мини компьютеров. В настоящее время наиболее популярными и покупаемыми являются Raspberry Pi 2, Banana Pi 2 и Orange Pi PC (рис. 3).



Рис. 3. Фото одно платных мини компьютеров

Эти компьютеры объединяют следующие характеристики:

1. Малые размеры, с кредитную карточку.
2. Четырехядерный процессор, который работает на всех компьютерах примерно на частоте 1–1.2 ГГц.
3. Оперативная память 1 Гбайт.
4. SD card вместо жесткого диска для загрузки операционной системы и программ.
5. Ethernet порт для подключения к сети.
6. HDMI выход для подключения монитора или телевизора.
7. USB порты для подключения, например клавиатуры, мыши, флешь памяти.
8. Операционная система Linux.
8. И главное – 40-а пиновый GPIO порт, к которому подключаются устройства, датчики, которыми надо управлять.

Одной из основных задач был выбор компьютера для удаленного управления. Для этого рассматривались параметры:

1. Стоимость (на 25.02.2016, сайт <http://ru.aliexpress.com> с доставкой):
    - Raspberry Pi 2 - \$36.99;
    - Banana Pi 2 (BPI-M2 A31S) - \$50.21;
    - Orange Pi pc - \$18.99.
  2. Быстродействие процессора + памяти: по вычислительным тестам с использованием 4-х ядер
    - Banana Pi 2 (BPI-M2 A31S);
    - Orange Pi pc;
    - Raspberry Pi 2.
- При использовании одного ядра для вычислительных работ (задача не распараллелена)
- Orange Pi pc;

- Banana Pi 2 (BPI-M2 A31S);
- Raspberry Pi 2.

Отмечают, что у Orange Pi работает устойчиво 3 ядра, 4-е ядро не всегда запускается.

3. Техническая поддержка и наличие отлаженного программного обеспечения:

- Raspberry Pi 2;
- Banana Pi 2;
- Orange Pi pc.

У Orange Pi pc пока нет главного условия для управления устройствами – это программной поддержки порта GPIO.

Ранее отмечалось, что удаленное управление устройствами может выполняться с помощью микроконтроллеров (дана ориентировочная цена):

1. Arduino Mega256 с Ethernet Shield w5100 – \$12–15;
2. Arduino nano с контроллером сети enc28j60 – \$8-9;
3. ESP8266-12 – \$2-3.

Однако исследования показали, что в локальной сети микроконтроллеры работают неплохо, а в глобальной сети при потерях пакетов управление становится ненадежным [1]. Мини компьютеры работают под управлением ОС Linux, у которой сетевые протоколы отлажены хорошо. Для мини компьютеров можно делать высокую степень защиты для входа в управляемую систему. У микроконтроллеров для надежно работающих протоколов и защиты от взлома недостаточно ресурсов. Рассмотрим последовательность решения задачи с использованием самой дешевой модели Raspberry Pi B[2].

1. Установка операционной системы Raspbian. Для этого с сайта <https://www.raspberrypi.org/downloads/raspbian/> скопируется образ операционной системы RASPBAN JESSIE, например на компьютер под управлением Windows 8.1. Далее разархивированный файл с помощью утилиты Win32DiskImager (<http://sourceforge.net/projects/win32diskimager>) устанавливается на SD card компьютера. После этого SD card переносится на компьютер Raspberry Pi и подключается к источнику питания. Предварительно к компьютеру необходимо подключить монитор, клавиатуру, мышь и кабель Ethernet. Компьютер автоматически загружается и на экране появляется меню предварительной настройки, которое формируется файлом rasp-config. Опции этого файла можно посмотреть по ссылке <https://www.raspberrypi.org/documentation/configuration/rasp-config.md>

2. Назначение Raspberry Pi статического IP-адреса. Raspberry Pi выполняет здесь функцию web-сервера, поэтому он должен иметь статический IP-адрес. Для этого меняется содержимое файла `/etc/network/interfaces` на

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
    address 172.20.0.137
    netmask 255.255.0.0
    gateway 172.20.200.1
    dns-nameservers 8.8.8.8
```

- полностью удаляется из системы dhcpd5, выполнением команды:

```
sudo apt-get purge dhcpd5
```

3. Установка Фреймворка Webiopi. Webiopi представляет собой пакет программ, специально разработанный для Raspberry Pi для удаленного управления устройствами. Совместно с Raspberry Pi он реализует технологию Internet of Things (Интернет вещей) и позволяет создавать различные пользовательские приложения. В качестве примера на рисунке 4 представлены основные возможности этого Фреймворка в соответствии с его описанием на официальном сайте <http://webiopi.trouch.com/> [3], которые сводятся к следующим:

- Встроенный web-сервер, реализованный на языке Python;
- Встроенная поддержка более чем 30 устройств с интерфейсами UART, SPI, I2C, 1-Wire;
- Библиотеки Javascript / HTML для создания Web-интерфейса;
- Библиотеки Python / Java для создания приложений под Android;
- Поддержка протокола CoAP предназначенный для управления и взаимодействия между простыми электронными устройствами через сеть.

Webiopi имеет открытый код, который может быть изменен пользователем. Это позволяет увеличить количество задач для решения. Для настройки пакета под конкретную задачу изменяется файл конфигурации. Например, в этот файл записываются GPIO pins, к которым подключены устройства. Если используются датчики, их также заносят в конфигурационный файл. Однако необходимо в некоторых случаях также включить и драйвер устройства (например, для температурного датчика ds18b20). Рассмотрим установку версии 0.6 Webiopi. С этой версией устойчиво работает Raspberry Pi model B, который имеет 26 пиновый порт GPIO. Для установки WebIOPi, входим в Raspberry Pi через программу Putty (логин – pi, пароль – raspberry) и в терминале вводим поочередно следующие команды:

```
$ wget http://webiopi.googlecode.com/files/WebIOPi-0.6.0.tar.gz
$ tar xvzf WebIOPi-0.6.0.tar.gz
```

```
$ cd WebIOPi-0.6.0
$ sudo ./setup.sh
```

После завершения установки активируется автозапуск WebIOPi для того, что бы каждый раз после включения Raspberry Pi не выполнять запуск приложения вручную. Для этого выполняем команду:

```
$ sudo update-rc.d webiopi defaults
```

и перезапускаем Raspberry Pi:

```
sudo reboot
```



Рис. 4. Возможности Фреймворка Webiopi

Для проверки работы Webiopi с любого компьютера в локальной сети вводим в браузере сетевой адрес, присвоенный Raspberry Pi с указанием порта 8000. Например:

```
http://172.20.0.137:8000/app/gpio-header
```

Для доступа к WebIOPi необходимо в открывшейся форме ввести логин и пароль. По умолчанию используется логин «webiopi», пароль – «raspberrypi». При нормальной работе браузер отобразит интерфейс программы Webiopi с номерами пинов порта GPIO и их назначением, как показано на рис. 5.

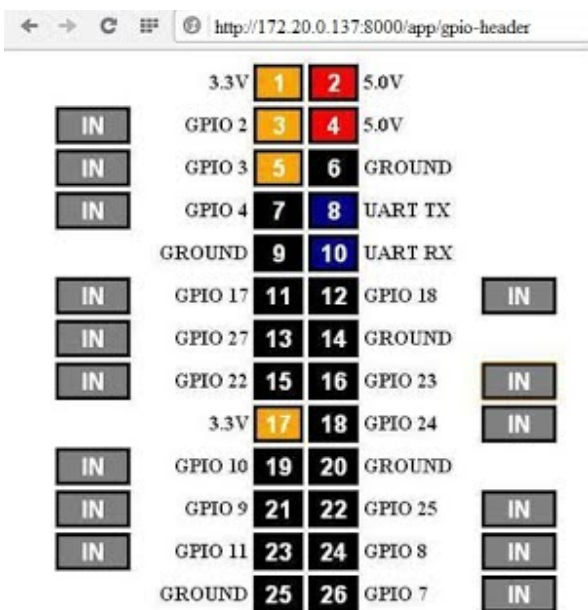


Рис. 5. Интерфейс Фреймворка Webiopi

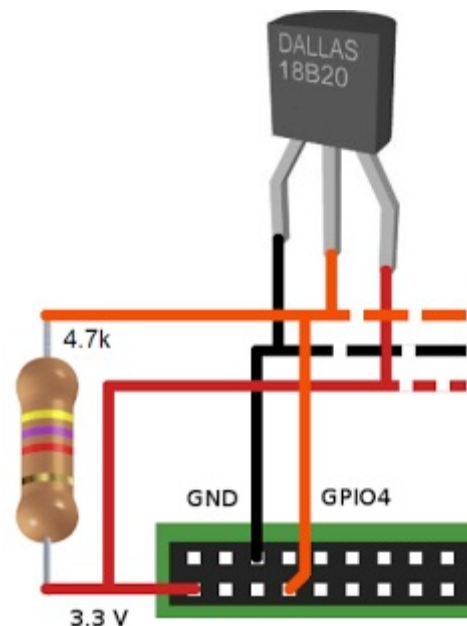


Рис. 6. Иллюстрация подключения DS18B20 к GPIO4

Из рисунка видно, что по умолчанию все пины GPIO работают на вход (IN). Для переключения на выход, необходимо мышкой нажать на кнопку IN. Значение этого пина измениться на OUT. Черный цвет пина означает низкий уровень порта, желтый – высокий. Если кликнуть мышкой на номер порта, его значение переключиться с низкого на высокий и черный цвет поменяется на желтый. В этом случае произойдет включение(или выключение) устройства, подключенного к этому пину.

Для настройки Webiopi под сформулированную здесь задачу необходимо выбрать пины порта GPIO для подключения розеток и датчика температуры DS18B20 и прописать их в конфигурационном файле Webiopi:

- GPIO23 – розетка для подключения мощного источника (обогревателя);
- GPIO9 – розетка 1;

GPIO11 – розетка 2.

Датчик температуры подключен к GPIO4, который реализует шину 1-wire (рис. 6).

Редактируем конфигурационный файл config:

```
sudo nano /etc/webiopi/config
```

Настраиваем порты GPIO 9, 11, 23 как выходы с высоким стартовым уровнем. В секции [GPIO] записываем:

```
9 = OUT 1
11 = OUT 1
23 = OUT 1
```

Для того, чтобы работал датчик температуры в секции [DEVICES] прописываем:

```
tmp0 = DS18B20
```

В секции [SCRIPTS] указываем имя и расположение файла-скрипта на Питоне, необходимый для считывания данных с датчика температуры и контроля наступления включения или выключения обогревателя в заданное время:

```
myproject = /home/pi/myproject/python/script.py
```

В секции [HTTP] указываем имя и расположение html файла:

```
doc-root = /home/pi/myproject/html
```

и номер порта работы встроенного в Webiopi web-сервера:

```
port = 80
```

В секции [REST] прописываем активные порты (только они будут работать):

```
gpio-export = 9, 11, 23
```

и разрешаем вводить изменение с браузера величин GPIO (низкий и высокий уровень пина) но запрещаем менять функции GPIO (нельзя менять OUT на IN):

```
gpio-post-value = true
gpio-post-function = false
```

Для того, чтобы работал датчик DS18B20, необходимо в файл /boot/config.txt вставить строку

```
dtoverlay=w1-gpio.
```

Если к GPIO4 подключено несколько датчиков DS18B20, то каждый из них должен иметь свое имя.

Для определения имени переходим в каталог: `cd /sys/bus/w1/devices`

Распечатываем файлы в этом каталоге:

```
ls
28-00000272b04c 28-000002d9a4f5 w1_bus_master1
```

Первые два имени определяют два датчика DS18B20. Заносим их в /etc/webiopi/config в секцию [DEVICES]:

```
tmp0=DS18B20 slave:28-00000272b04c
tmp1=DS18B20 slave:28-000002d9a4f5
```

Изменение пароля Webiopi выполняется командой

```
$ sudo webiopi-passwd
```

Далее выполняем перезагрузку компьютера командой `reboot`. Подключиться к web-серверу webiopi теперь можно по адресу: <http://172.20.0.137/app/gpio-header>. В браузере появится изображение аналогичное рисунку 5, но активными будут только порты gpio9, 11, 23. Проверить работоспособность датчика температуры можно, подключившись по адресу: <http://172.20.0.137/app/devices-monitor>. В браузере должно появиться значение температуры от датчика. Для перезагрузки webiopi после внесения изменений в конфигурационный файл, в скрипт на Python, html файл, необходимо выполнить команду: `/etc/init.d/webiopi restart`. Сообщения об ошибках при запуске Webiopi находятся в файле `/var/log/webiopi`. Его можно распечатать так: `cat /var/log/webiopi`.

4. Создание файла `index.html`. Этот html файл с WebIOPi Javascript library должен сформировать web-страничку, показанную на рисунке 1, передавать на сервер команды при нажатии на кнопки, получать от сервера информацию о состоянии кнопок, выводить значение даты, температуры и взаимодействовать со скриптом на Python. Файл `index.html` размещается в каталоге `/home/pi/myproject/html`, который предварительно создается. Из-за большого размера этого файла он не будет здесь представлен. Ниже для примера показан фрагмент JavaScript для создания двух кнопок управления розетками, добавления этой кнопки в controls box с использованием jQuery function:

```
// Create a "Розетка 1" labeled button1 for GPIO 9
var button1 = webiopi().createGPIOButton(9, "Розетка 1");
// Append button to HTML element with ID="controls1" using jQuery
$("#controls1").append(button1);
// Create a "Розетка 2" labeled button2 for GPIO 11
var button2 = webiopi().createGPIOButton(11, "Розетка 2");
// Append button to HTML element with ID="controls2" using jQuery
$("#controls2").append(button2);
```

А также пример вызова макроса каждые 5 секунд для запроса и получения температуры с датчика DS18B20:

```
// Вызов макроса изменения температуры с интервалом 5 сек.
```

```
setInterval ("callMacro_getTmp0()", 5000);{ }
// Запрос температуры
function callMacro_getTmp0(){ webiopi().
  callMacro("getTmp0", [], macro_getTmp0_Callback); }
// Получение температуры
function macro_getTmp0_Callback(macro, args, data)
  { $("#celsius_0").text("Температура: "+data+" °C"); }
```

Полную распечатку файла index.html можно найти в источнике [3].

### 5. Создание файла-скрипта на Python - script.py

Этот файл должен находиться в каталоге /home/pi/myproject/python, который должен быть предварительно создан. Задачей скрипта является получение данных с температурного датчика и передача их в файл index.html, запись файла /home/pi/myproject/html/data\_18B20.txt с значениями температур и времени их определения для построения графика, анализа времени включения и выключения розетки для мощного оборудования и ее включения или отключения. Ниже приведена распечатка файла /home/pi/myproject/python/script.py

```
import webiopi
import datetime
import time
from webiopi import deviceInstance
from time import strftime
GPIO = webiopi.GPIO
LIGHT = 23 # GPIO pin
HOUR_ON = 0 # Turn Light ON at 13:00
HOUR_OFF = 0 # Turn Light OFF at 14:00
MIN_ON = 0
MIN_OFF = 0
celsius_0 = 0,0
cel_0=0,0
num=1
# setup function is automatically called at
WebIOpi startup
def setup():
  # set the GPIO used by the light to output
  GPIO.setFunction(LIGHT, GPIO.OUT)
  # retrieve current datetime
  now = datetime.datetime.now()
  # test if we are between ON time and tun
the light ON
  if ((now.hour >= HOUR_ON) and (now.hour
<= HOUR_OFF) and ((now.minute >= MIN_ON) and
(now.minute <= MIN_OFF))):
    GPIO.digitalWrite(LIGHT, GPIO.LOW)
@webiopi.macro
def getTmp0():
  global celsius_0
  tmp0 = webiopi.deviceInstance("tmp0")
  celsius_0 = tmp0.getCelsius() # получение
температуры
  print (celsius_0)
  return "%.2f" % celsius_0 # возврат данных
температуры в HTML
  # с округлением до сотых
@webiopi.macro
def tihour():
  now = datetime.datetime.now()
  return "Время {0}, {1}".format("%d:%d
%d:%d" % (HOUR_ON, MIN_ON, HOUR_OFF, MIN_OFF),
strftime("%Y-%m-%d %H:%M"))
(продолжение в следующем столбце)
```

```
def loop():
  global cel_0
  global num
  # retrieve current datetime
  now = datetime.datetime.now()
  # toggle light ON all days at the correct
time
  if ((now.hour == HOUR_ON) and (now.minute ==
MIN_ON) and (now.second == 0)):
    if (GPIO.digitalRead(LIGHT) ==
GPIO.HIGH):
      GPIO.digitalWrite(LIGHT, GPIO.LOW)
      # toggle light OFF
      if ((now.hour == HOUR_OFF) and (now.minute
== MIN_OFF) and (now.second == 0)):
        if (GPIO.digitalRead(LIGHT) ==
GPIO.LOW):
          GPIO.digitalWrite(LIGHT, GPIO.HIGH)
  if (num==300):
    tmp0 = webiopi.deviceInstance("tmp0")
    cel_0 = tmp0.getCelsius()
    f =
open('/home/pi/myproject/html/data_18B20.txt',
'a')
    data_entry =
"{0},{1}\n".format(strftime("%Y-%m-%d
%H:%M:%S"), "%.2f" % cel_0)
    f.write(data_entry)
    f.close()
    num=0
  num+=1
  # gives CPU some time before looping again
time.sleep(1)
@webiopi.macro
def getLightHours():
  return "%d;%d;%d;%d" % (HOUR_ON, HOUR_OFF,
MIN_ON, MIN_OFF)
@webiopi.macro
def setLightHours(on, off, onm, offm):
  global HOUR_ON, HOUR_OFF, MIN_ON, MIN_OFF
  HOUR_ON = int(on)
  HOUR_OFF = int(off)
  MIN_ON = int(onm)
  MIN_OFF = int(offm)
  return getLightHours()
# destroy function is called at WebIOpi shutdown
def destroy():
  GPIO.digitalWrite(LIGHT, GPIO.HIGH)
```

### 6. Построение графика температуры

График температуры строиться с помощью библиотеки dygraph, файл которой dygraph-combined-dev.js необходимо скопировать в каталог /home/pi/myproject/html с сайта <http://dygraphs.com>. Эта библиотека для прорисовки графика использует файл /home/pi/myproject/html/data\_18B20.txt, который формирует каждые 300 секунд скрипт на Python. Построение графика будет выполняться браузером при обращении по адресу:

<http://172.20.0.137/temp.html>.

Текст файла temp.html для формирования графика, показанного на рисунке 7, представлен в источнике [4].

7. Подключение к Интернет устройства, если оно не имеет реального ip-адреса или доменного имени, но имеет выход в Интернет(через модем, router, межсетевой экран). Одним из способов получения

доступа к Raspberry Pi как к устройству интернет вещей является использование сервиса Weaved [5]. Он предлагает следующие услуги:

SSH – позволяет войти в Raspberry Pi с любой точки мира по SSH;

Web (http) on port 80 – можно просматривать web-страницы с любой точки мира, размещенные на Raspberry Pi;

WebIOPi – позволяет управлять пинами GPIO порта Raspberry Pi, используя разработанное пользователем программное обеспечение.

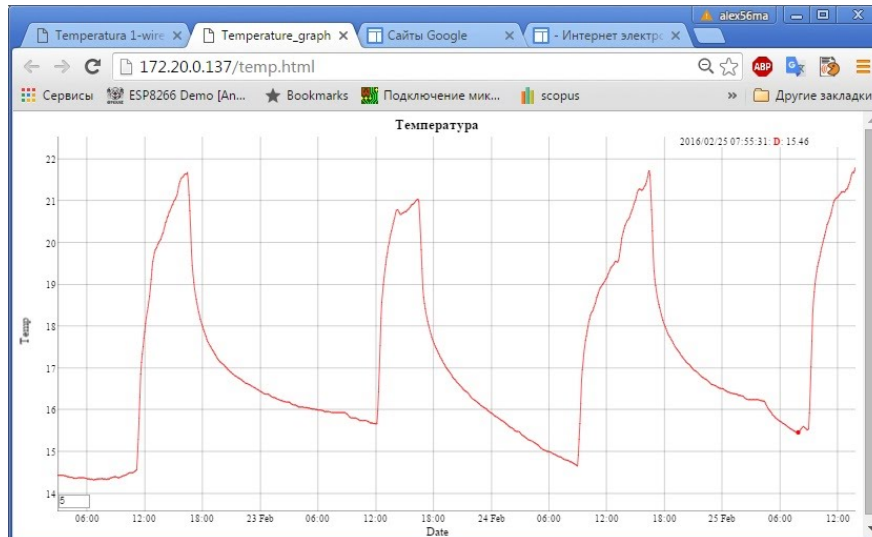


Рис. 7. График температуры, который генерируется при нажатии на ссылку "Кафедра КСМ" рисунка 1

Перед установкой Weaved создается каталог `/home/pi/myproject/my`, и выполняются следующие действия:

- На сайте <https://developer.weaved.com/portal/login.php> регистрируем аккаунт;
- Подключаем Raspberry Pi к Интернет;
- Загружаем Weaved Software на Raspberry Pi:

```
wget https://github.com/weaved/installer/raw/master/binaries/weaved-nixinstaller_1.2.13.bin
```

- Делаем файл `weaved-nixinstaller_1.2.13.bin` исполняемым:

```
chmod +x weaved-nixinstaller_1.2.13.bin
```

- Запускаем программу установки:

```
./weaved-nixinstaller_1.2.13.bin
```

- Выбираем услугу. При первом запуске программы будет предложено установить одну из услуг: SSH на порт 22, Web (HTTP) на 80-й порт, WebIOPi на порт 8000, VNC на порт 5091 (протестирован с `tightvncserver`), или пользовательский TCP на выбранном порту.

Выбираем здесь 2-ю услугу, т.е. Web (HTTP) на 80-й порт.

- Далее вводится информация для входа в Weaved (ввести аккаунт, который был получен на сайте Weaved).

Вводится имя своего устройства, например `webiopi80`.

## New Service Plans Available

Go to [My Account](#) to choose a new service plan.

## Your current list of services

Click on service names to connect. Your account allows for 10 registered services and 30 minute connections on up to 1 concurrent service(s). You can [Upgrade Now](#) to get more services, longer connection times and more concurrent connections.

Name	Type	Status	
<a href="#">pi2_22</a>	SSH	online	<a href="#">Share</a>   <a href="#">Settings</a>
<a href="#">ssh22pi</a>	SSH	online	<a href="#">Share</a>   <a href="#">Settings</a>
<a href="#">webiopi80</a>	HTTP	online	<a href="#">Share</a>   <a href="#">Settings</a>

Рис. 8.

- Выполняется проверка, было ли создано новое устройство: для этого по адресу <https://developer.weaved.com/portal/login.php> вводим свой аккаунт. После входа должна появиться следующая страничка, где указано имя созданного устройства:

Если нажать на ссылку на имени устройства, то должна появиться страничка как на рис. 8, только адрес будет примерно такой: <https://eksgaxwj.p19.weaved.com/>. Таким образом подключение к Raspberry Pi было выполнено через промежуточный сервер, который предоставлен сервисом Weaved.

8. Подключение Raspberry Pi к релейному модулю. На рис. 9 представлена иллюстрация подключения релейного модуля к пинам порта GPIO Raspberry Pi и электророзеткам. Для повышения мощности нагрузки два реле подключены параллельно к розетки повышенной мощности (например, для подключения обогревателя). Приведенное на рисунке подключение работоспособно, если реле срабатывают от напряжения 5 вольт. Необходимо отметить, что реле срабатывает при подаче на вход модуля нулевого напряжения.

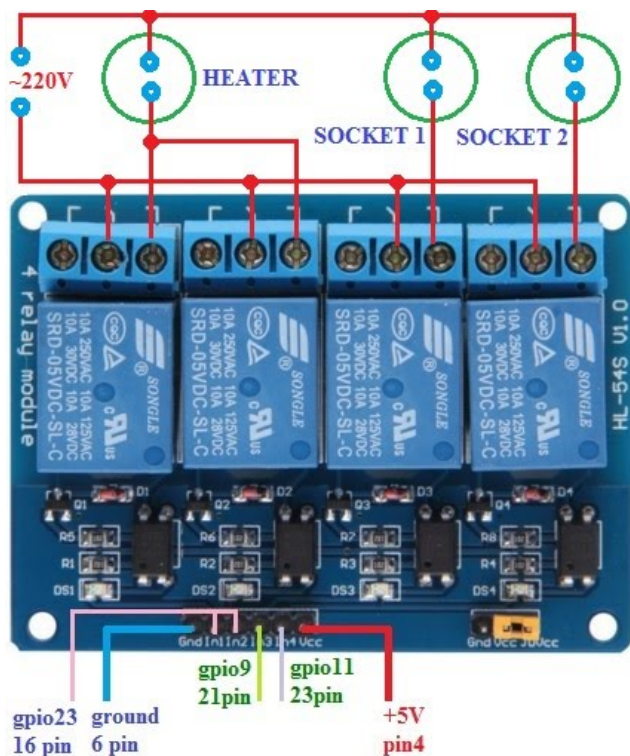


Рис. 9. Подключение релейного модуля к пинам Raspberry Pi и розеткам

**Выводы**

1. Показана практическая возможность создания надежной системы управления электропитанием через Интернет на базе миниатюрного компьютера Raspberry Pi.
2. В качестве программного обеспечения для построения web-сервера, управлением выводами порта GPIO, датчиками, для работы со скриптами на Python использован Фреймворк WebIOPi, который показал эффективность для решения подобных задач.
3. Показана возможность использования разработанного устройства в организациях, которые не могут предоставить реальные IP-адреса или DNS имена, но которые могут предоставить выход в Интернет по технологии с использованием NAT сервера, которая является самой распространенной и универсальной в настоящее время. Это отмечает высокую универсальность разработанного устройства.
4. Разработанное устройство позволяет отслеживать показания датчиков в течении длительного времени, записывая измеряемые значения в файл и графически отображая их. Здесь это показано на примере температурного датчика.
5. Необходимо отметить меньшую стоимость разработанного устройства по сравнению с существующими производственными аналогами (в 2-3 раза) [6].
6. Использование Фреймворка WebIOPi совместно с сервисом Weaved позволяет обеспечить высокую степень защиты от несанкционированного доступа к устройству управления электропитанием.

**Литература**

1. Мясищев А.А. Изучение возможности построения web-сервера на платформе Arduino для удаленного управления / А.А. Мясищев, С.В. Ленков, Г.Б. Жиров // Сучасна спеціальна техніка. – 2013. – № 4(35). – С. 41–50.
2. Raspberry Pi Foundation [Электронный ресурс]. – Режим доступа : <https://www.raspberrypi.org/>, 2016.
3. WebIOPi – The Raspberry Pi Internet of Things Framework [Электронный ресурс]. – Режим доступа : <http://webiopi.trouch.com/>, 2016.
4. Мясищев А.А. Интернет электророзетка на основе мини компьютера Raspberry Pi и Фреймворка WebIOPi. Практика для студентов [Электронный ресурс]. – Режим доступа : [https://sites.google.com/site/webstm32/internet\\_rozetka](https://sites.google.com/site/webstm32/internet_rozetka), 2016.
5. Manage Networked Devices From Anywhere [Электронный ресурс]. – Режим доступа : <http://www.weaved.com/>, 2016.
6. Netping. Устройство мониторинга серверных комнат и удаленного управления электропитанием [Электронный ресурс]. – Режим доступа : <http://www.netping.ru/>, 2016.
7. Комплексная система домашней автоматизации на Raspberry Pi. 2014 [Электронный ресурс]. – Режим доступа : <http://electromost.com/>.

Рецензія/Peer review : 30.5.2016 р. Надрукована/Printed : 6.6.2016 р.  
Статья прорецензирована редакционной коллегией