

УДАЛЕННОЕ УПРАВЛЕНИЕ КАМЕРОЙ НА СЕРВОПРИВОДАХ

В работе изучается возможность создание системы для удаленного управления видеокамерой с помощью сервоприводов и пересылки полученных изображений на систему обработки этих изображений с целью принятия решения по управлению роботом. В качестве системы управления используется одноплатный мини-компьютер Raspberry Pi 2 под управлением операционной системы RASPBIAN(Linux). Позиционирование камеры выполняется Фреймворком WebIOPi, передача видео - потока программой mjpg-streamer.

Ключевые слова: Raspberry Pi, mjpg-streamer, Python, JavaScript, WebIOPi, servo SG90, camera, Internet of Things.

Введение. Создание автономных транспортных средств (роботов) связано с решением целого комплекса задач. Одной из задач является создание системы управления видеокамерой с помощью сервоприводов и пересылки полученных изображений со скоростью потокового видео на систему обработки этих изображений с целью принятия решения. Решение задачи управления камерой возможно с использованием микроконтроллеров, реализованных, например, в проектах Ардуино. Однако для формирования и передачи видео-потока через сеть необходимы вычислительные мощности, которые доступны лишь микрокомпьютерам. В настоящее время широкое распространение получают одноплатные мини-компьютеры, которые работают под управлением операционной системы Linux. Не смотря на их размеры с кредитную карточку, они способны решать аналогичные задачи, программы для которых широко распространены под Linux.

Постановка задачи. В работе изучается вопрос о возможности практического построения системы удаленного управления камерой, входящей в состав мини компьютера Raspberry Pi 2 с помощью сервоприводов, вращающих камеру в двух плоскостях. Для управления сервоприводами по сети TCP/IP предлагается использовать Фреймворк WebIOPi, а формирование видео потока с камеры на удаленный браузер - с помощью программы mjpg-streamer для видео захвата и трансляции в сеть.

На рис. 1 представлена практическая реализация устройства. На рис. 2 показано фото картинки с камеры для одного из ее положений и кнопок управления сервоприводами с индикацией положения. В строку ввода Input step вводится шаг поворота сервопривода в градусах. Рассмотрен самый простой случай - загружены два окна браузера Chrome. Первое окно подключено к web - серверу, который запускается при запуске mjpg-streamer(port 9000). Второе окно подключено к web - серверу Фреймворка WebIOPi (port 8000).

Результаты. Решение задачи выполнялось поэтапно.

1. Для вращения камеры использованы сервоприводы SG90, которые устанавливаются на серво-кронштейны для обеспечения вращения в двух плоскостях на углы +90...-90 градусов. Сервоприводы подключены к GPIO порту Raspberry Pi 2. Подключение сервоприводов иллюстрируется рисунком 3. Для стабильной работы установки целесообразно сервоприводы подключать к отдельному от компьютера источнику питания 5В. Управляющие выходы сервоприводов подключены к выводам GPIO23 и GPIO24 для вращения камеры влево-вправо и вверх-вниз соответственно.

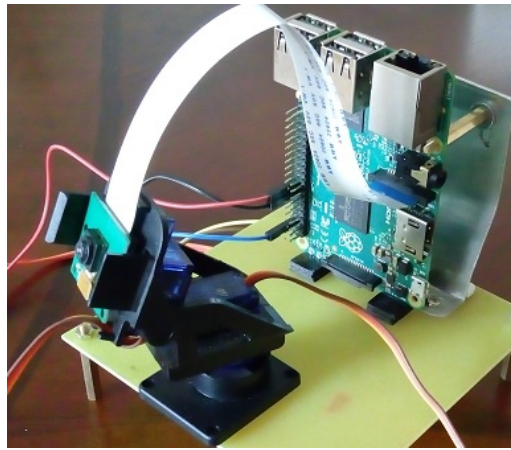


Рис. 1. Фото камеры на сервоприводах с подключенным к ней Raspberry Pi 2

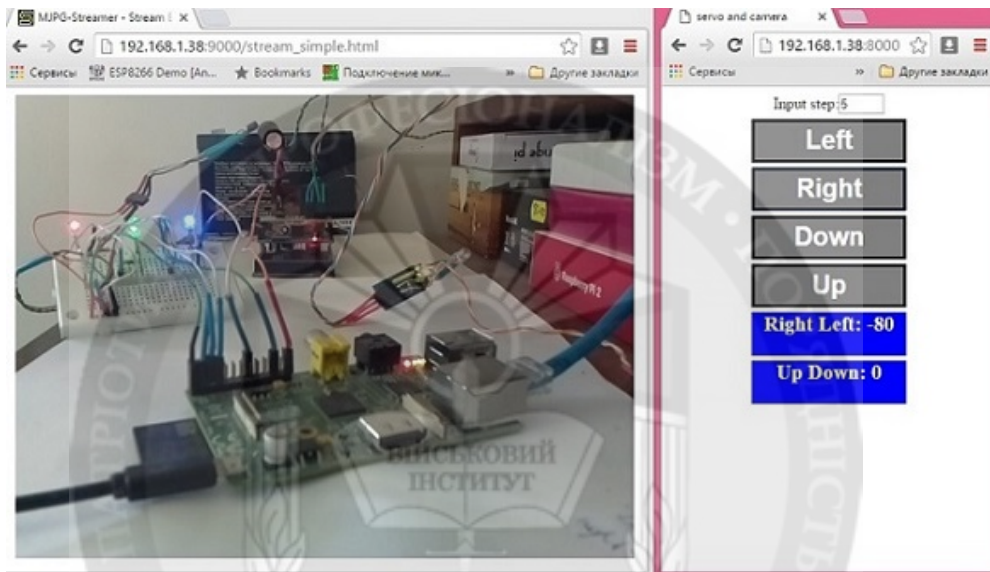


Рис. 2. Иллюстрация интерфейса при работе с камерой

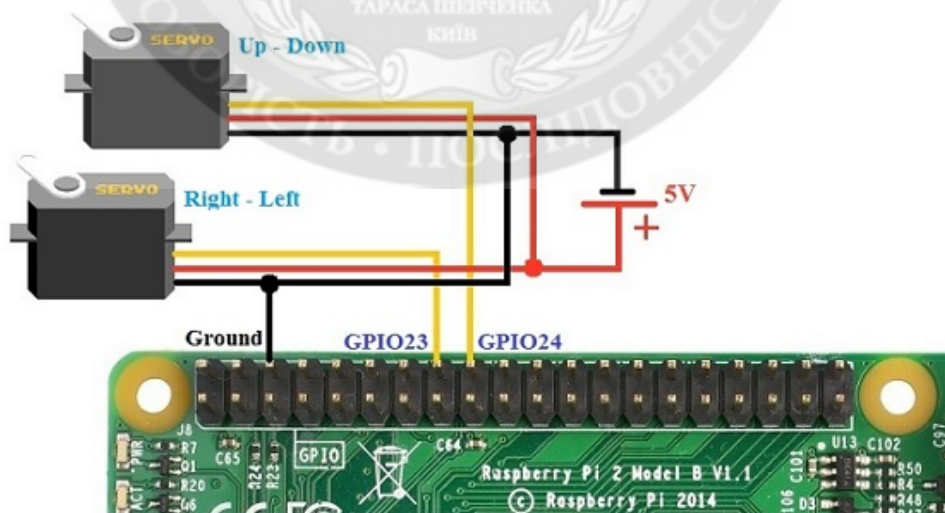


Рис. 3. Иллюстрация подключения сервоприводов к Raspberry Pi 2

2. Удаленное управление сервоприводами выполняется с помощью Фреймворка WebIOPi. WebIOPi представляет пакет программ, специально разработанный для Raspberry Pi для удаленного управления устройствами. Совместно с Raspberry Pi он реализует

технологии Internet of Things (Интернет вещей). В частности WebIOPi имеет следующие необходимые для решения задачи возможности:

- Встроенный Web - сервер, реализованный на языке Python
- Библиотеки Javascript / HTML для создания Web-интерфейса
- Библиотеки Python для работы с подключенными к нему устройствами(например, сервоприводами)

Перед установкой WebIOPi предполагается, что на Raspberry Pi 2 уже установлена операционная система RASPBIAN, и ему присвоен IP адрес 192.168.1.38.

Для работы 40 выводов порта GPIO целесообразно установить Фреймворк WebIOPi версии 0.7.1:

```
$ wget http://sourceforge.net/projects/webiopi/files/WebIOPi-0.7.1.tar.gz
$ tar xvzf WebIOPi-0.7.1.tar.gz
$ cd WebIOPi-0.7.1
```

Устанавливаем patch для работы с 40 выводами порта GPIO Raspberry Pi 2:

```
$ wget https://raw.githubusercontent.com/doublebind/raspi/master/webiopi-pi2bplus.patch
$ patch -p1 -i webiopi-pi2bplus.patch
$ sudo ./setup.sh
```

Для автоматического старта WebIOPi после перезагрузки компьютера выполняется команда (действительно для образа системы 2015-05-05-raspbian-wheezy.img):

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

А более поздних версий выполняются следующие команды:

```
$ cd /etc/systemd/system/
$ sudo wget https://raw.githubusercontent.com/doublebind/raspi/master/webiopi.service
$ sudo systemctl start webiopi
$ sudo systemctl enable webiopi
```

После установки выполняется редактирование конфигурационного файла config:

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

Для этого настраиваются порты GPIO 23,24 как выходы с низким стартовым уровнем. В секции [GPIO] записываем:

```
23 = OUT 0
24 = OUT 0
```

В секции [SCRIPTS] указываем имя и расположение файла - скрипта на Питоне, который используется для управления сервоприводами:

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

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

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

В секции [REST] прописываются активные порты(только эти порты будут работать):

```
gpio-export = 23,24
```

После этого выполняется перезагрузка Raspberry Pi 2:

```
sudo reboot
```

Для входа в web - server по адресу http://192.168.1.38:8000 используется логин - "webiopi" , пароль - "raspberrypi"

3. Настройка Фреймворка WebIOPi. Построение файлов сценария на Python и HTML файла с JavaScript.

Принимается, что сервоприводы поворачивают камеру Right-Left на угол -90...+90 градусов и Up-Down на угол -90...+90 градусов соответственно. Исходное состояние при старте или рестарте WebIOPi для Right-Left 0 градусов и Up-Down также 0 градусов.

В скрипте scrpt.py используются встроенные в WebIOPi функции:

GPIO.setFunction(SERVO1, GPIO.PWM) - устанавливает режим работы вывода SERVO1, как ШИМ вывод.

GPIO.pwmWriteAngle(SERVO1, 0) - для сервопривода, подключенного к выводу SERVO1 устанавливает угол 0 градусов.

Задержка, формируемая командой webiopi.sleep(0.25) и последующее изменение режима работы вывода SERVO1 (GPIO.setFunction(SERVO1, GPIO.OUT)) необходимы для предотвращения "дрожания" сервопривода (см. литературу по принципу работы сервопривода).

Аналогично для вывода SERVO2.

Итогом является файл-сценарий на языке Python - /home/pi/myproject/scrpt.py:

```
import webiopi
import datetime
GPIO = webiopi.GPIO
SERVO1=23
SERVO2=24
SERRL = 0 # Angle rotation Right-Left
STE = 0 # Step angle rotation
SERUD = 0 # Angle rotation Up-Down

SERRL=SERRL+STE
if (abs(SERRL)>=90):
    SERRL=90
GPIO.pwmWriteAngle(SERVO1, SERRL)
webiopi.sleep(0.25)
GPIO.setFunction(SERVO1, GPIO.OUT)
return gets()

@webiopi.macro
def ri(on):
    global SERRL,STE
    STE = int(on)
    GPIO.setFunction(SERVO1, GPIO.PWM)
    SERRL=SERRL-STE
    if (abs(SERRL)>=90):
        SERRL=-90
    GPIO.pwmWriteAngle(SERVO1, SERRL)
    webiopi.sleep(0.25)
    GPIO.setFunction(SERVO1, GPIO.OUT)
    return gets()

# setup function is automatically called at WebIOPi startup
def setup():
# set the GPIO used by the light to output
    GPIO.setFunction(SERVO1, GPIO.PWM)
    GPIO.setFunction(SERVO2, GPIO.PWM)
    GPIO.pwmWriteAngle(SERVO1, 0)
    GPIO.pwmWriteAngle(SERVO2, 0)
    webiopi.sleep(0.25)
    GPIO.setFunction(SERVO1, GPIO.OUT)
    GPIO.setFunction(SERVO2, GPIO.OUT)

# destroy function is called at WebIOPi shutdown
def destroy():
    GPIO.digitalWrite(SERVO1, GPIO.OUT)
    GPIO.digitalWrite(SERVO2, GPIO.OUT)

@webiopi.macro
def steps():
    global SERRL,STE
    return "%d" % SERRL

@webiopi.macro
def steps1():
    global SERUD,STE
    STE = int(on)
    GPIO.setFunction(SERVO2, GPIO.PWM)
    SERUD=SERUD+STE
    if (abs(SERUD)>=90):
        SERUD=90
    GPIO.pwmWriteAngle(SERVO2, SERUD)
    webiopi.sleep(0.25)
    GPIO.setFunction(SERVO2, GPIO.OUT)
    return gets()
```

```
global SERUD,STE
return "%d" % SERUD
```

```
@webiopi.macro
def gets():
    global SERRL
    return "%d" % STE
```

```
@webiopi.macro
def lef(on):
    global SERRL,STE
    STE = int(on)
    GPIO.setFunction(SERVO1, GPIO.PWM)
```

(продолжение в следующем столбце)

```
@webiopi.macro
def ups(on):
    global SERUD,STE
    STE = int(on)
    GPIO.setFunction(SERVO2, GPIO.PWM)
    SERUD=SERUD-STE
    if (abs(SERUD)>=90):
        SERUD=-90
    GPIO.pwmWriteAngle(SERVO2, SERUD)
    webiopi.sleep(0.25)
    GPIO.setFunction(SERVO2, GPIO.OUT)
    return gets()
```

Далее представлен html файл с WebIOPi Javascript library формирующий web - страничку (правая часть рис. 2), которая передают на сервер команды при нажатии на кнопки, получает от сервера информацию о состоянии кнопок, выводит значение углов поворота камеры, т.е. выполняет взаимодействие со скриптом на Python. Файл index.html находится в каталоге /home/pi/myproject/html. Из-за большого размера этого файла он не будет здесь представлен. Ниже в частности показан фрагмент JavaScript для создания кнопки поворота серво-привода вправо, вызова функции pi скрипта и добавления этой кнопки в to the controls box using a jQuery function:

```
// Create a button to call lef macro
var sendButton1 = webiopi().createButton("sendButton1", "Left", function() {
    // Arguments sent to the macro
    var servo = $("#inputOn").val();
    // Call the macro
    webiopi().callMacro("lef", servo, update);
});
// Append the button to the controls box using a jQuery function
$("#controls").append(sendButton1);
```

4. Ниже рассмотрим установку и настройку программы MJPG-Streamer. Выше отмечалось, что эта программа необходима для формирования видео-потока и удаленной передачи его на вычислительную систему анализа изображения и принятия решения о поведении робота. MJPG-Streamer используют потоковый протокол под названием Motion JPEG или сокращенно MJPEG, представляющий собой просто поток отдельных JPEG фотографий, следующей одна за другой. Достоинством его является то, что большинство современных браузеров могут проигрывать MJPEG потоки.

Для установки MJPG-Streamer на Raspberry Pi 2 выполним последовательность действий, описанных ниже.

Предположим, наш текущий каталог /home/pi. Создадим каталог:

```
sudo mkdir /opt/mjpg-streamer
```

Установим библиотеку libjpeg62-dev:

```
sudo apt-get install libjpeg62-dev
```

Устанавливаем cmake:

```
sudo apt-get install cmake
```

Загружаем mjpg-streamer с плагином raspicam:

```
git clone https://github.com/jacksonliam/mjpg-streamer.git ~/mjpg-streamer
```

Заходим в директорию:

```
cd ~/mjpg-streamer/mjpg-streamer-experimental
```

Выполняем компиляцию

```
make clean all
```

Перемещаем файлы и удаляем рабочую директорию:

```
sudo mv ~/mjpg-streamer/mjpg-streamer-experimental /opt/mjpg-streamer
```

```
sudo rm -rf ~/mjpg-streamer
```

Создаем файл /opt/mjpg-streamer/start_stream.sh для старта MJPG-Streamer:

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

```
if pgrep mjpg_streamer > /dev/null
```

```
then
```

```
    echo "mjpg_streamer already running"
```

```
else
```

```
LD_LIBRARY_PATH=/opt/mjpg-streamer/ /opt/mjpg-streamer/mjpg_streamer -i
```

```
"input_raspicam.so -fps 10 -q 50 -x 640 -y 480" -o "output_http.so -p 9000 -w /opt/mjpg-streamer/www" > /dev/null 2>&1&
```

```
    echo "mjpg_streamer started"
```

```
fi
```

Создаем файл /opt/mjpg-streamer/stop_stream.sh для останова MJPG-Streamer:

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

```
if pgrep mjpg_streamer
```

```
then
```

```
    kill $(pgrep mjpg_streamer) > /dev/null 2>&1
```

```
    echo "mjpg_streamer stopped"
```

```
else
```

```
    echo "mjpg_streamer not running"
```

```
fi
```

Таким образом для запуска необходима команда:

```
/opt/mjpg-streamer/stop_stream.sh
```

Для останова:

```
/opt/mjpg-streamer/stop_stream.sh
```

Просмотреть видео-поток можно, набрав в браузере адрес:

```
http://192.168.1.38:9000/stream_simple.html
```

Получим изображение, как в левом окне рисунка 2.

Выводы.

1. Представлена практическая реализация удаленного управления сервоприводами видео камеры по сети TCP/IP. Управление выполнено на базе Фреймворка WebIOPi.

2. Показана возможность использования MJPG-Streamer для удаленной передачи видеопотока на вычислительную систему для принятия решения управления роботом с установленной камерой. В качестве системы управления использован одноплатный мини-компьютер Raspberry Pi 2 под управлением операционной системы Linux (RASPBIAN).

3. Установлено, що сервоприводи очень чувствительны к броскам напряжения. Например, при включении мощных потребителей (обогревателя) сервоприводы "дергаются" и меняется положение камеры.

4. Замечено "подвисание" WebIOPi при управлении камерой. После рестарта по команде /etc/init.d/webiopi restart, управление камерой возобновляется.

ЛИТЕРАТУРА:

1. WebIOPi - The Raspberry Pi Internet of Things Framework. [Electronic resource]. - Mode of access: <http://webiopi.trouch.com/>, 2016.

2. Мясішев А.А. Управление камерой на сервоприводах с помощью Raspberry Pi 2 и Фреймворка WebIOPi. - Mode of access: https://sites.google.com/site/webstm32/servo_camera, 2016.

3. Raspberry Pi camera board video streaming. [Electronic resource]. - Mode of access: <https://miguelmota.com/blog/raspberry-pi-camera-board-video-streaming/>. 2013.

4. RASPBERRY PI CAMERA MODULE. - Mode of access: <https://www.raspberrypi.org/documentation/raspbian/applications/camera.md>. 2015.

REFERENCES:

1. WebIOPi - The Raspberry Pi Internet of Things Framework. [Electronic resource]. - Mode of access: <http://webiopi.trouch.com/>, 2016.

2. Mjasishhev A.A. Upravlenie kameroy na servoprivodah s pomoshh'ju Raspberry Pi 2 i Frejmvorka WebIOPi. - Mode of access: https://sites.google.com/site/webstm32/servo_camera, 2016.

3. Raspberry Pi camera board video streaming. [Electronic resource]. - Mode of access: <https://miguelmota.com/blog/raspberry-pi-camera-board-video-streaming/>. 2013.

4. RASPBERRY PI CAMERA MODULE. - Mode of access: <https://www.raspberrypi.org/documentation/raspbian/applications/camera.md>. 2015.

Без рецензії.

д.т.н., проф. Мясішев О.А., к.т.н. Ленков Є.С., к.т.н., с.н.с. Охрамович М.М.
ВІДДАЛЕНЕ УПРАВЛІННЯ КАМЕРОЮ НА СЕРВОПРИВОДАХ

У роботі вивчається можливість створення системи для віддаленого керування відеокамерою за допомогою сервоприводів і пересилання отриманих зображень на систему обробки цих зображень з метою прийняття рішення по управлінню роботом. В якості системи управління використовується одноплатний міні-комп'ютер Raspberry Pi 2 під управлінням операційної системи RASPBIAN (Linux). Позиціонування камери виконується фреймворками WebIOPi, передача відео - потоку програмою mjpg-streamer.

Ключові слова: Raspberry Pi, mjpg-streamer, Python, JavaScript, WebIOPi, servo SG90, camera, Internet of Things.

**Prof. Myasishev A.A., Ph.D. Lenkov Y.S., Ph.D. Ohramovych M.M.
REMOTE CAMERA CONTROL ON SERVO DRIVES**

We study the possibility of creating a system for remote control of a video camera with the help of servo and sending the received images on image-processing system in order to decide on the robot control. As a control system used single-board mini-computer Raspberry Pi 2 which running operating system RASPBIAN (Linux). For positioning the camera is used program WebIOPi, for transmission of video is used program mjpg-streamer.

Keywords: Raspberry Pi, mjpg-streamer, Python, JavaScript, WebIOPi, servo SG90, camera, Internet of Things.