

Dr. Bob Davidov

Импорт и экспорт МатЛАБ данных через Raspberry Pi

Цель работы: Ознакомиться с пакетом МатЛАБ для ввода / вывода данных и управления ресурсами компьютера Raspberry Pi.

Задача работы: Построить несколько вариантов связи МатЛАБ с компьютером Raspberry Pi для ввода / вывода данных и построения фрагментов распределенных систем.

Приборы и принадлежности: Персональный компьютер с MatLAB 2014a, компьютер Raspberry Pi с Linux ОС Raspbian Wheezy, цифровые и I²C устройства, согласователь уровней для COM соединений.

ВВЕДЕНИЕ

В среде МатЛАБ имеется приложение обеспечивающее связь и совместную работу МатЛАБ и Simulink с компьютером Raspberry Pi. Такой инструмент позволяет использовать Raspberry Pi (**RPi**) в качестве интерфейса связи МатЛАБ с внешней средой при решении задач моделирования, управления, идентификации, синтеза систем, наблюдения, тестирования и других. Мощные специализированные средства МатЛАБ могут быть задействованы и для оперативной разработки алгоритмов автономной работы Raspberry Pi.

В работе рассматриваются следующие вопросы.

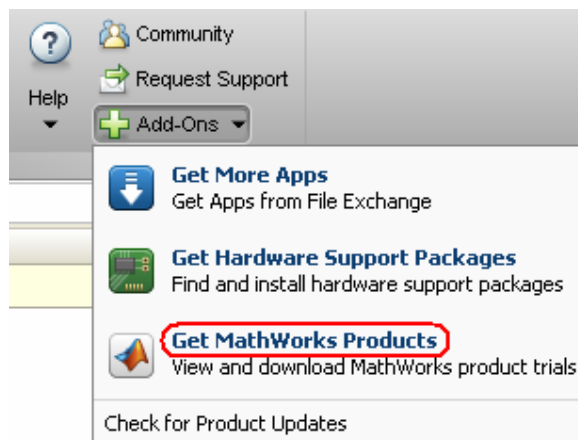
- Установка пакета МатЛАБ для работы с Raspberry Pi и установка ОС Raspbian Wheezy на SD карту RPi для взаимодействия с пакетом МатЛАБ
- Получение сетевых параметров для статического IP соединения
- Запуск примеров МатЛАБ для работы с Raspberry Pi
- Установка соединения МатЛАБ – Raspberry Pi
- Подключение аппаратных ресурсов Raspberry Pi к МатЛАБ. Команды для работы с периферией RPi.
- Интерфейс RPi Linux команд
- Работа с файлами компьютера Raspberry Pi
- Пакет библиотеки Simulink для работы с RPi

ОБЩИЕ СВЕДЕНИЯ

УСТАНОВКА ПАКЕТА МАТЛАБ ДЛЯ РАБОТЫ С RASPBERRY PI И УСТАНОВКА ОС RASPBIAN WHEEZY НА SD КАРТУ RPI ДЛЯ ВЗАИМОДЕЙСТВИЯ С ПАКЕТОМ МАТЛАБ

Установка пакета МатЛАБ для работы с Raspberry Pi выполняется через Интернет. Взаимодействие МатЛАБ с RPi осуществляется через Linux операционную среду (ОС): Raspbian Wheezy, которая может быть установлена на SD карту компьютера Raspberry Pi в

процессе установки пакета МатЛАБ. Установка пакета может быть выполнена через МатЛАБ Toolstrip > Add-Ons > [Get Hardware Support Packages](#)

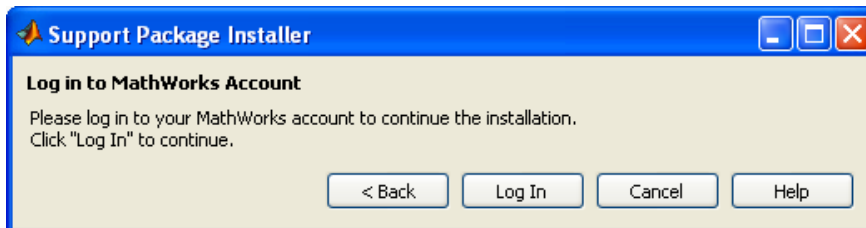


или следующим образом.

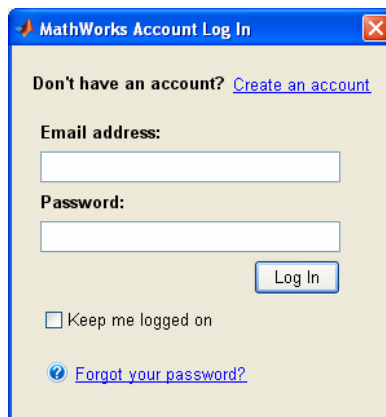
1. С официального сайта МатЛАБ <http://www.mathworks.com/hardware-support/raspberry-pi-simulink.html> загрузите установочный файл пакета МатЛАБ для работы с RPi.



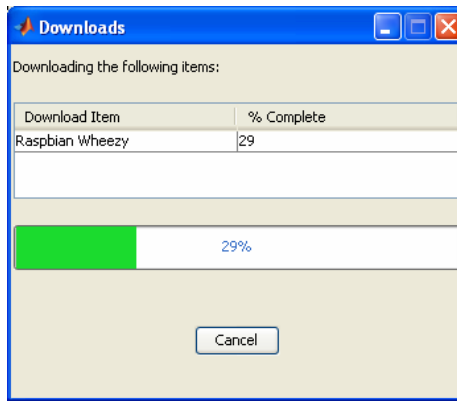
2. Запустите МатЛАБ и настройте его на каталог с загруженным файлом.
3. В среде МатЛАБ запустите установочный файл.
4. Дойдите до следующего указателя и нажмите Log In.



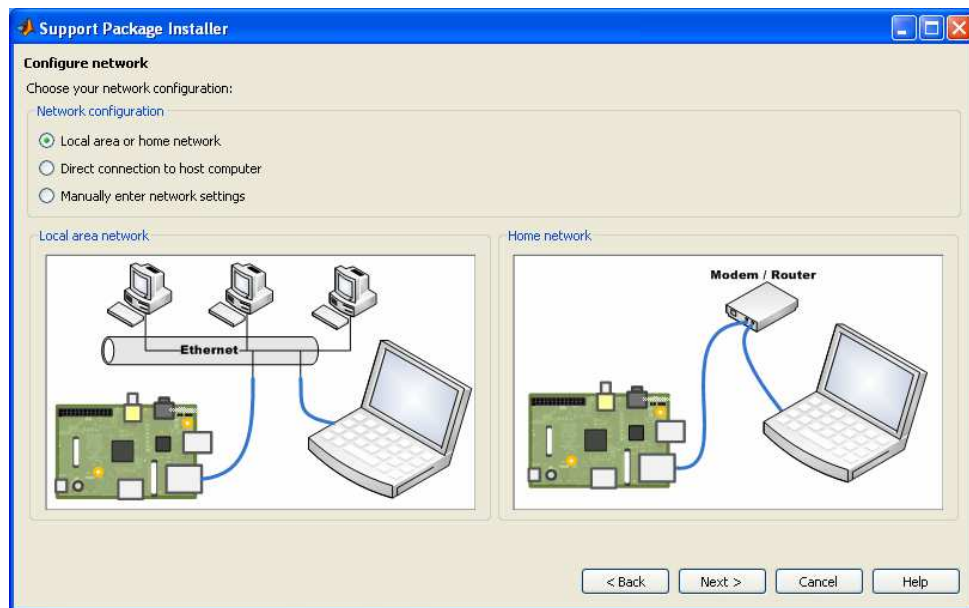
5. Зарегистрируйтесь на сайте МатЛАБ и/или введите параметры вашей учетной записи.



6. Выполняйте следующие шаги по установке пакета и дождитесь окончания загрузки Linux Операционной Среды (ОС): Raspbian Wheezy.



7. Выберите вариант подключения Raspberry Pi к хост компьютеру с МатЛАБ. Это необходимо для последующей начальной настройки Raspbian Wheezy на SD карту RPi.

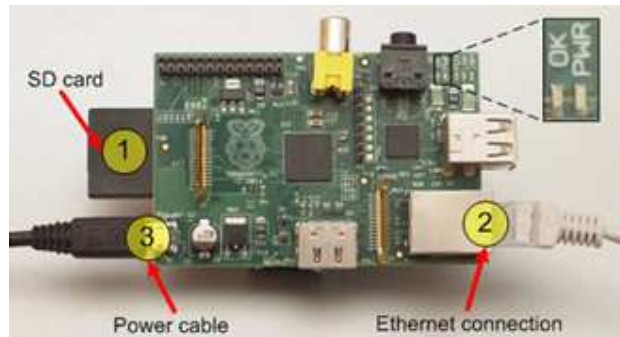


8. Для установки Linux операционной среды Raspbian Wheezy на SD карту (объёмом не менее 4 ГБ) вставьте карту в считывающее устройство компьютера.
Внимание! Если на SD карте имеются данные, то они будут удалены при установке Raspbian Wheezy.

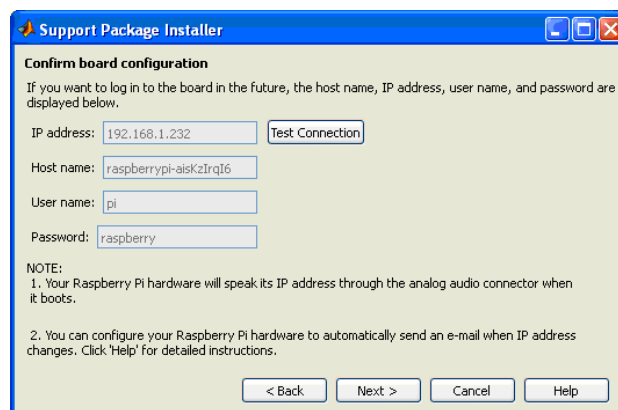


9. Запустите установку Raspbian Wheezy (~ 2.8 ГБ) на SD карту RPi.

10. После установки ОС вставьте SD карту в приемник Raspberry Pi, подключите компьютер к сети и подайте питание на компьютер RPi.



11. Перейдите к следующему пункту для сканирования установщиком сетевых устройств. При успешном сканировании установщик определяет IP адреса компьютера RPi. Сохраните IP адрес, Host name, user name и password.



12. Для завершения установки нажмите **Finish**.

Примечание: Установка пакета Raspbian Wheezy из среды МатЛАБ может быть выполнена командой МатЛАБ

Targetupdater

ПОЛУЧЕНИЕ СЕТЕВЫХ ПАРАМЕТРОВ ДЛЯ СТАТИЧЕСКОГО IP СОЕДИНЕНИЯ

Для статического соединения

- IP адрес каждого сетевого устройства должен быть уникальным,
- маска сети (Subnet mask) должна быть единой для всех сетевых устройств,
- обычно используется общий шлюз (Gateway) для всех сетевых устройств

Для просмотра сетевых параметров наберите в командной строке основного компьютера `start > run > cmd > ipconfig`.

Если известен IP адрес одного из устройств, например, 192.168.1.XXX, то адреса других

```
C:\WINDOWS\system32\cmd.exe

Ethernet adapter Local Area Connection:

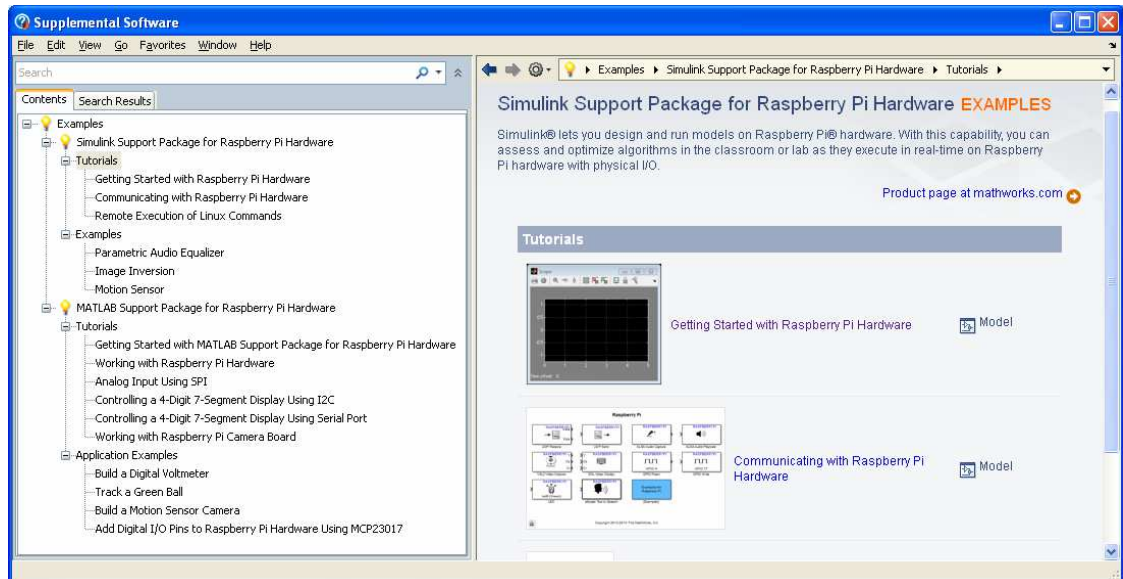
    Connection-specific DNS Suffix  . :
    IP Address. . . . .               : 192.168.1.59
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 192.168.1.1
```

сетевых устройств лежат в диапазоне 192.168.1.3 .. 192.168.1.254, а маска сети для них остаётся такой же: 255.255.255.0

ЗАПУСК ПРИМЕРОВ МАТЛАБ ДЛЯ РАБОТЫ С RASPBERRY PI

После установки пакета МатЛАБ можно посмотреть примеры работы с Raspberry Pi список который открывается командой

```
raspi_examples
```



УСТАНОВКА СОЕДИНЕНИЯ МАТЛАБ – RASPBERRY PI

Подключение МатЛАБ к Raspberry Pi создается через дескриптор – идентификатор объекта (handle), например,

```
>> муpi = raspi
```

где муpi – идентификатор соединения.

Для подключения к Raspberry Pi по IP адресу, имени и паролю, необходимо использовать команду, например,

```
>> gaspi('192.168.1.232','pi','raspberrу') % или
```

```
>> муpi = gaspi('192.168.1.232','pi','raspberrу')
```

При успешном соединении на экран выводятся параметры RPi периферии:

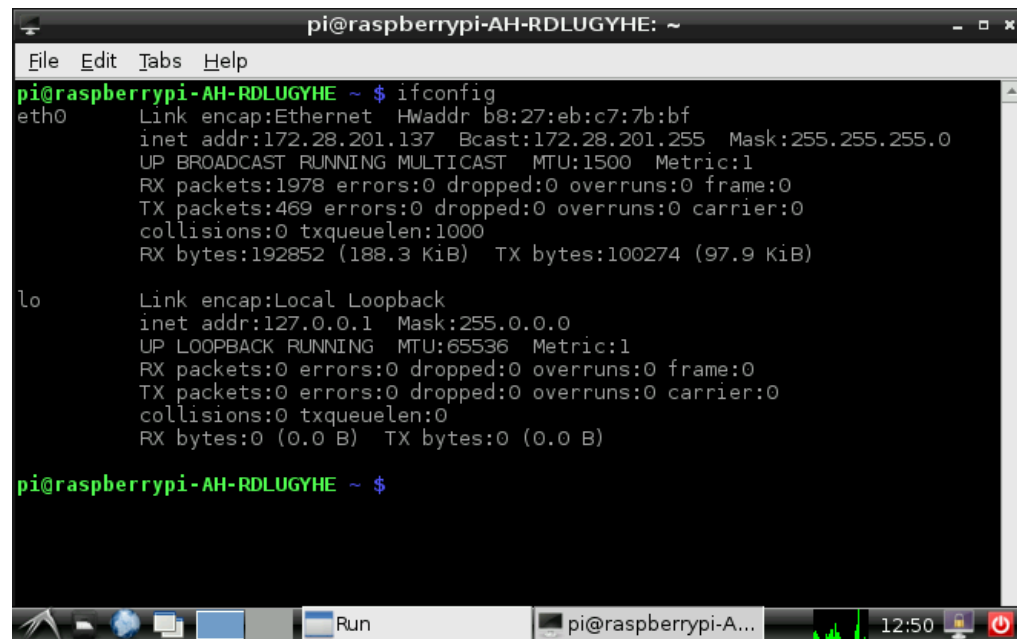
```
ans =
raspi with properties:
```

```
DeviceAddress: '192.168.1.232'  
Port: 18725  
BoardName: 'Raspberry Pi Model B Rev 2'  
AvailableLEDs: {'led0'}  
AvailableDigitalPins: [4 7 8 9 10 11 14 15 17 18 22 23 24 25 27 30 31]  
AvailableSPIChannels: {}  
AvailableI2CBuses: {'i2c-0' 'i2c-1'}  
I2CBusSpeed: 100000
```

Примечание: IP адрес Raspberry Pi подключенного к сети через SSH сервер можно найти программой сканированием сетевых устройств, например, [ipscan23](#).

Для озвучивания IP адреса Raspberry Pi необходимо подключить наушники к audio разъёму RPi, сделать рестарт и прослушать IP адрес.

Для отображения IP адреса на дисплее необходимо подключить клавиатуру и мышь к USB портам, монитор – напрямую или через адаптер к HDMI порту или к выходу S-video. После загрузки графической оболочки ОС откройте [start](#) меню на Raspbian Wheezy desktop и выберите [Select Accessories > LXTerminal](#). Терминал покажет параметры сетевого соединения, например, как показано ниже.



```
pi@raspberrypi-AH-RDLUGYHE: ~  
File Edit Tabs Help  
pi@raspberrypi-AH-RDLUGYHE ~$ ifconfig  
eth0  Link encap:Ethernet  Hwaddr b8:27:eb:c7:7b:bf  
      inet addr:172.28.201.137  Bcast:172.28.201.255  Mask:255.255.255.0  
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
      RX packets:1978 errors:0 dropped:0 overruns:0 frame:0  
      TX packets:469 errors:0 dropped:0 overruns:0 carrier:0  
      collisions:0 txqueuelen:1000  
      RX bytes:192852 (188.3 KiB)  TX bytes:100274 (97.9 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-AH-RDLUGYHE ~$
```

ПОДКЛЮЧЕНИЕ АППАРАТНЫХ РЕСУРСОВ RASPBERRY PI К МАТЛАБ.

Из среды МатЛАБ можно подключиться к Raspberry Pi, например, для

- управления состоянием светодиода платы RPi;
- считывания и установки состояния выводов разъема GPIO;
- подключения к устройствам которые подключены к Raspberry Pi через
 - последовательный порт RS-232
 - I²C интерфейс
 - SPI интерфейс
- записи видео и получения снимков при помощи камеры подключенной к RPi
- подачи Linux команд оболочки Raspberry Pi.

Таблица 1. Команды для работы с периферией RPi

Устройство	Примеры команд	Описание	
RPi	<code>mypi = raspi;</code>	Соединение с RPi	
	<code>clear mypi;</code>	Уничтожение соединения	
	<code>mypi.BoardName</code>	Отображение версии RPi	
	<code>system(mypi, 'sudo shutdown -r now');</code>	Выключение RPi	
GPIO Цифровой ввод / вывод	<code>showPins(mypi)</code>	Отображение местоположения контактов разъема GPIO	
	<code>mypi = raspi</code> <code>mypi.AvailableDigitalPins</code>	Отображение доступных контактов ввода / вывода	
	<code>configureDigitalPin(mypi,4,'input')</code>	Конфигурирование цифрового канала, например 4-го, на ввод	
	<code>configureDigitalPin(mypi,7,'output')</code>	Конфигурирование цифрового канала на вывод	
	<code>buttonPressed = readDigitalPin(mypi,23);</code>	Чтение входа GPIO 23	
Светодиод RPi	<code>mypi = raspi</code> <code>mypi.AvailableLEDs</code>	Отображение состояния светодиода	
	<code>showLEDs(mypi)</code>	Отображение местоположения светодиода	
	<code>writeLED(mypi,'led0',1);</code> <code>writeLED(mypi,'led0',true)</code>	Включение светодиода	
	<code>writeLED(mypi,'led0',0)</code> <code>writeLED(mypi,'led0',false)</code>	Выключение светодиода	
	Перезагрузка ОС	Установка СД в начальное состояние	
	COM port	<code>system(mypi, 'rpi-serial-console status')</code>	Проверка статуса COM порта
RS-232 (+3.3 В TTL) GPIO 14 и 15	<code>myserialdevice = serialdev(mypi,'%dev/ttyAMA0')</code> <code>myserialdevice = serialdev with properties:</code> BaudRate: 115200 DataBits: 8 Parity: 'none' StopBits: 1 Timeout: 10	Подключение COM порта	
	<code>myserialdevice = serialdev(mypi,'%dev/ttyAMA0',115200,8,'none',2)</code> или <code>myserialdevice.Timeout = 20</code> или <code>myserial = serialdev(mypi,'%dev/ttyAMA0',9600);</code>	Изменение параметров порта	
	<code>write(myserialdevice,[10 12],uint16')</code>	Запись (передача) двух байт целого uint16 в последовательный порт	
	<code>output = read(myserialdevice,100);</code>	Пример чтения (приема данных) 100 байт последовательного порта	
	<code>clear myserialdevice;</code>	удаление COM соединения	
	<code>system(mypi, 'sudo rpi-serial-console disable');</code> <code>system(mypi, 'sudo shutdown -r now');</code> <code>clear mypi;</code>	Отключение COM порта перезагрузкой ОС	
	<code>system(mypi, 'sudo rpi-serial-console enable');</code> <code>system(mypi, 'sudo shutdown -r now');</code> <code>clear mypi;</code>	Использование COM порта и после перезагрузки ОС	
	I²C	<code>mypi = raspi</code> <code>mypi.AvailableI2CBuses</code>	Отображение возможных I ² C каналов
	GPIO 2 (SDA) GPIO 3 (SDL)	<code>mypi.I2CBusSpeed</code>	Отображение I ² C скорости
Скорость по дефолту: 100000	<code>disableI2C(mypi);</code> <code>enableI2C(mypi,400000);</code>	Изменение скорости I ² C	
	<code>scanI2CBus(mypi,'i2c-1')</code>	Санирование адресов I ² C устройств	
	<code>i2cdevice = i2cdev(mypi,'i2c-1','0x49')</code>	Создание связи с I ² C устройством через идентификатор объекта	
	<code>read(i2cdevice,1)</code>	Чтение I ² C устройства	

бод	<code>write(i2cdevice,4092);</code>	Запись значения в I ² C устройство
	<code>enableI2C(myPi)</code>	Установка I ² C конфигурации
	<code>disableI2C(myPi)</code>	Отмена I ² C конфигурации
SPI GPIO 11 (SPI0_SCLK) GPIO 10 (SPI0_MOSI) GPIO 9 (SPI0_MISO) GPIO 8 (SPI0_CE0) GPIO 7 (SPI0_CE1)	<code>myPi = raspi</code> <code>myPi.AvailableSPIChannels</code>	Отображение состояния SPI
	<code>enableSPI(myPi);</code> <code>myPi.AvailableSPIChannels</code>	Создание SPI интерфейса
	<code>myspidevice = spidev(myPi,'CE1')</code>	Создание соединения с SPI устройством
	<code>out = writeRead(myspidevice,[hex2dec('08') hex2dec('D4')])</code>	Для чтения данных из SPI, отправьте фиктивные значения
	<code>myPi = raspi</code> <code>myPi.AvailableSPIChannels</code>	Отображение состояния светодиода
	<code>out = writeRead(myspidevice,[hex2dec('08') hex2dec('D4')])</code>	Запись данных в SPI, по возможности, используйте данные возвращаемые устройством
	<code>disableSPI(myPi)</code>	Закрытие SPI канала
Видеокамера	<code>mycam =</code> <code>cameraboard(myPi,'Resolution','1280x720')</code>	Создание соединения с камерой
	<code>for ii = 1:10</code> <code>img = snapshot(mycam);</code> <code>imagesc(img);</code> <code>drawnow;</code> <code>end</code>	Импортирование и отображение 10 кадров на хост компьютер
	<code>mycam.Rotation = 180;</code>	Изменение ориентации изображения
	<code>record(mycam,'myvideo.mp4',10)</code>	Запись 10 сек видео
	<code>stop(mycam)</code>	Остановка процесса записи
	<code>getFile(myPi,'myvideo.mp4','C:\MATLAB')</code>	Копирование видео RPi на хост компьютер
	<code>deleteFile(myPi,'myvideo.mp4')</code>	Удаление видеофайла

Соединение остается активным до момента очистки всех дескрипторов которые используют соединение.

Нельзя создать новое соединение пока активно текущее соединение. Соединение, например, `myserial` остается активным, даже если очистить `myPi`. Для создания нового соединения, например, `mynewpi` необходимо удалить (`clear`) как `myPi` так и `myserial`.

Чтобы определить местоположение светодиода платы RPi используйте `showLEDs(myPi)` функцию. Команда `myPi.AvailableLEDs` позволяет считать состояние светодиода.

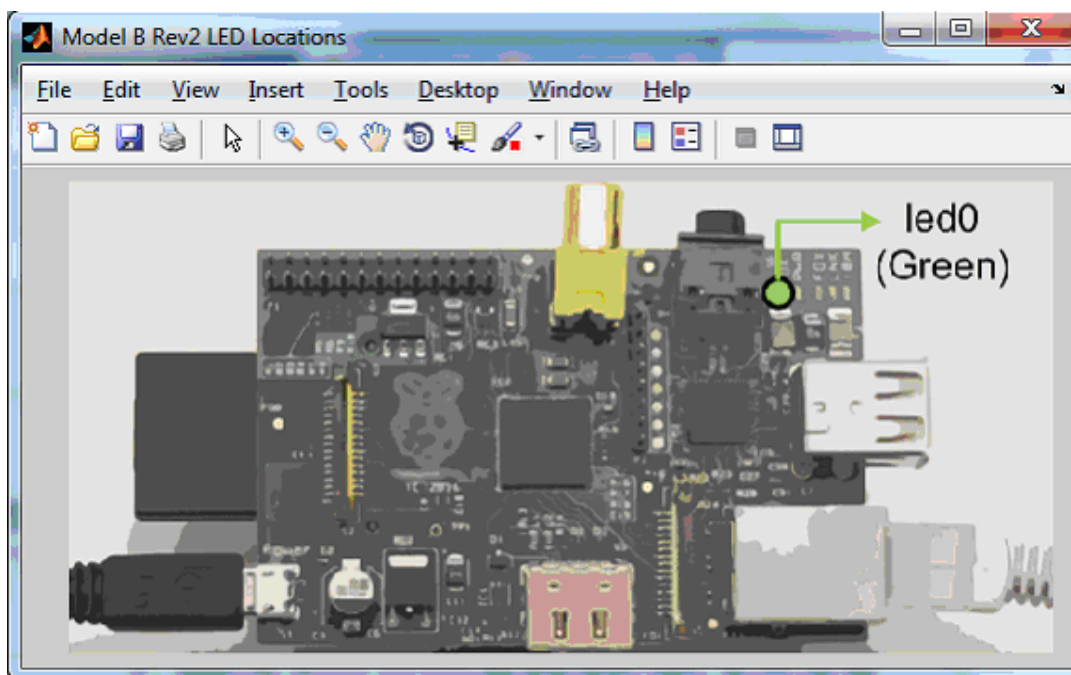


Рис. 1. Положение управляемого светодиода платы Raspberry Pi.

	3.3 V	1	2	5 V	
I2C (SDA)	GPIC 2	3	4	5 V	
I2C (SCL)	GPIC 3	5	6	Ground	
	GPIC 4	7	8	GPIC 14	UART (TXD)
	Ground	9	10	GPIC 15	UART (RXD)
	GPIC 17	11	12	GPIC 18	PWM аппаратный
	GPIC 27	13	14	Ground	
	GPIC 22	15	16	GPIC 23	
	3.3 V	17	18	GPIC 24	
SPI (MOSI)	GPIC 10	19	20	Ground	
SPI (MISO)	GPIC 09	21	22	GPIC 25	
SPI (SCLK)	GPIC 11	23	24	GPIC 08	SPI (CE0)
	Ground	25	26	GPIC 07	SPI (CE1)

Рис. 2. Контакты основного порта GPIO Raspberry Pi (Model B, Rev 2). Для цифрового ввода/вывода при соответствующей конфигурации можно использовать следующие выводы основного разъема GPIO: 2, 3, 4, 7, 8, 9, 10, 11, 14, 15, 17, 18, 22, 23, 24, 25, 27). Рядом с основным разъёмом GPIO на плате разведены (но не распаяны) ещё четыре пары GPIO выводов (с GPIO 28, 29, 30, 31), дополнительно дающие I²C и I²S интерфейсы (см. Рис. 4).

Выводы порта GPIO Raspberry Pi (Model B, Rev 2) после загрузки ОС имеют следующие начальные состояния.

- SPI интерфейс (GPIO выводы: 7, 8, 9, 10, 11) – отключен.
- I²C интерфейс (GPIO выводы: 2, 3, 28, 29) – включен.

Если вывод не сконфигурирован, то команда чтения контакта GPIO переводит его в состояние ввода. Соответственно, запись в цифровой порт – переводит контакт в состояние вывода.

Запись логической единицы устанавливает на выходе контакта +3.3 В. Запись нуля подключает контакт к земле – нулевому потенциалу.

Для изменения направления передачи уже сконфигурированного вывода необходимо использовать команду `configureDigitalPin` (см. Таблица 1).

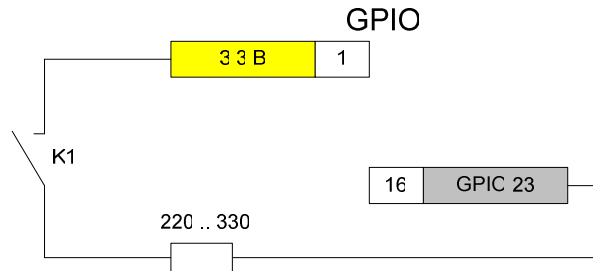


Рис. 3. Пример подключения кнопки к порту GPIO платы Raspberry Pi.

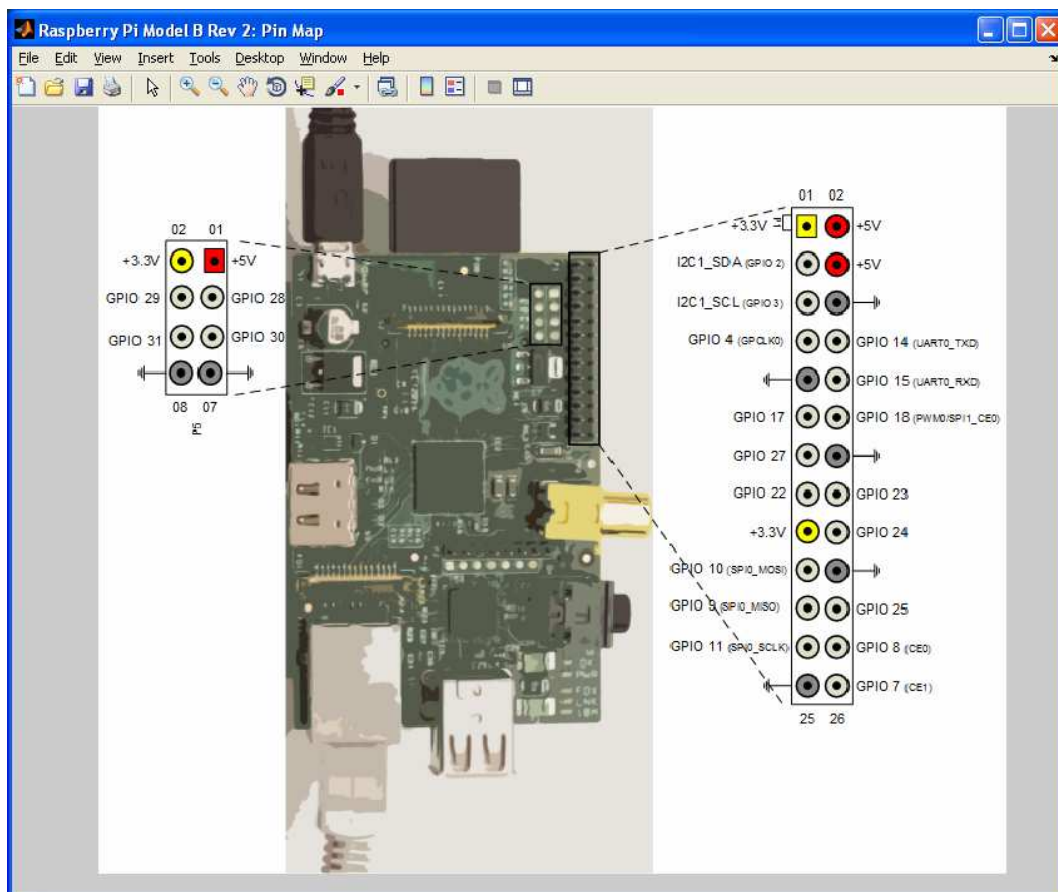


Рис. 4. Отображение местоположения контактов GPIO на плате RPi. Вызывается командой `showPins(мури)`.

Последовательный порт Raspberry Pi

Канал RS-232 (+3.3 В TTL) для последовательной передачи данных может быть сконфигурирован на выводах

- GPIO 14 (UART0_TXD) – передача данных периферийному устройству (RxD)
- GPIO 15 (UART0_RXD) – прием данных (TxD)

По умолчанию в ОС Raspbian Wheezy порт COM не установлен. Для работы с ним необходимо сконфигурировать показанные выше выводы порта GPIO

Внимание! Не подключайте 3.3 В последовательный порт Raspberry Pi к 12 В RS-232 каналу без согласователя напряжения.

I²C интерфейс Raspberry Pi

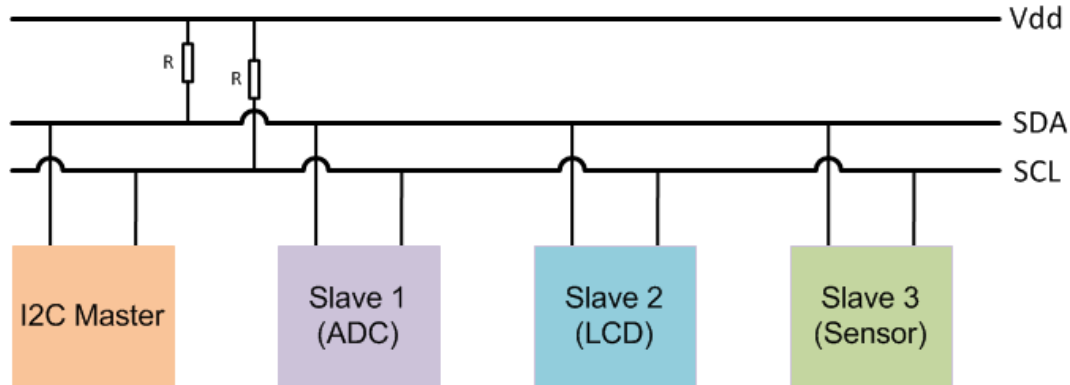


Рис. 5. Подключение устройств к шине I²C. Команды для работы с I²C представлены в Таблица 1.

В зависимости от модели RPi имеет один или два I²C канала последовательной передачи данных.

Одно из устройств, подключенных к I²C каналу является мастером (ведущим), а все остальные устройства выполняют функции ведомых. Канал I²C содержит две линии: линию данных SDA и линию синхроимпульсов (SCL) которые через резисторы (1.8кОм) “подтягиваются” к напряжению питания +3.3В.

SPI интерфейс Raspberry Pi

Serial Peripheral Interface (SPI) – полнодуплексный интерфейс, обеспечивающий параллельный прием и передачу данных – протокол высокоскоростной связи с периферией, например, дисплеем, датчиком, флеш картой.

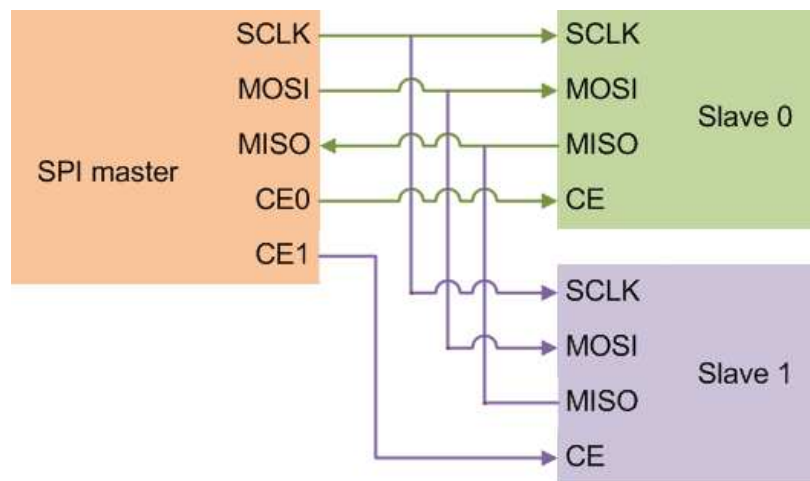


Рис. 6. Подключение устройств через SPI интерфейс. Команды для работы с SPI представлены в Таблица 1.

SPI мастер может обеспечивать связь с двумя периферийными устройствами

Состав SPI шины

- GPIO 11 (SPI0_SCLK) импульсы синхронизации
- GPIO 10 (SPI0_MOSI) выходные данные для периферии.
- GPIO 9 (SPI0_MISO) принимаемые данные от периферии.
- GPIO 8 (SPI0_CE0) активизация одного периферийного устройства.
- GPIO 7 (SPI0_CE1) активизация другого периферийного устройства.

RPI поддерживает следующие режимы работы SPI интерфейса:

- 0, 1 или 2;
- 8 бит слово данных;
- Скорости данных: 500 000, 1 000 000, 2 000 000, 4 000 000, 8 000 000, 16 000 000, 32 000 000 бод

Работа с камерой Raspberry Pi

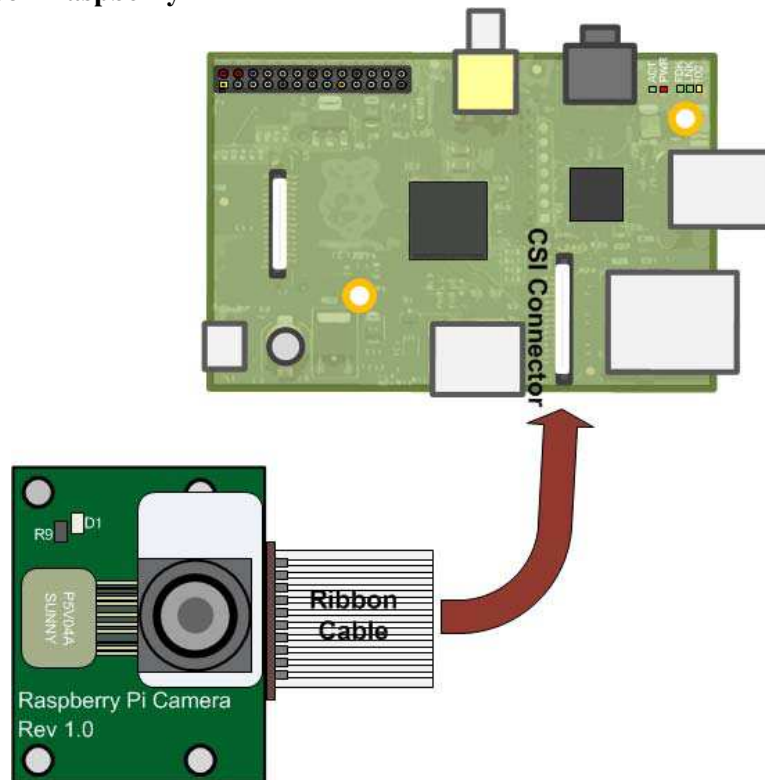


Рис. 7. Подключение видекамеры к CSI разъёму Raspberry Pi. Команды для работы с видекамерой представлены выше в Таблица 1.

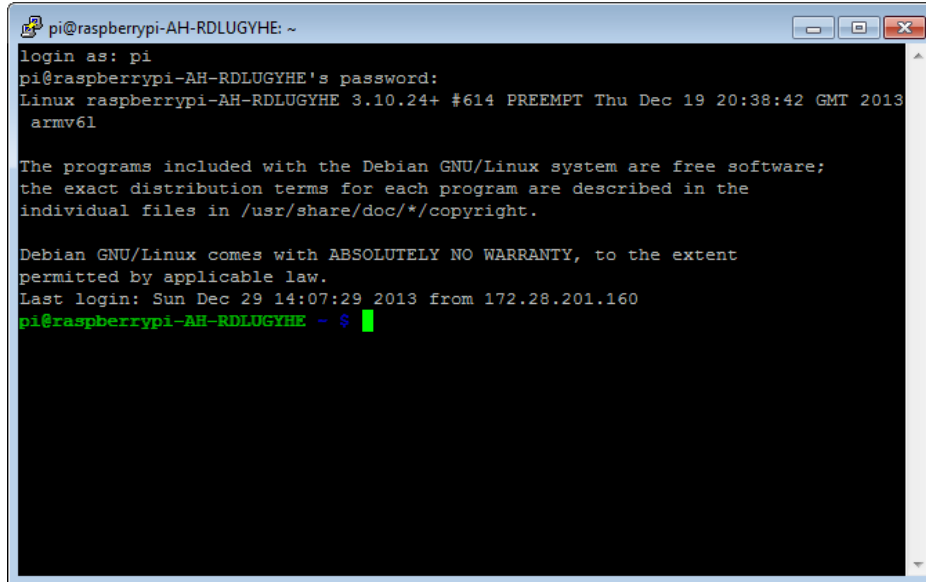
ИНТЕРФЕЙС RPI LINUX КОМАНД

Команды Linux передаются вторым аргументом команды `system`. Например, системная команда детального просмотра содержимого каталога RPi (атрибуты файлов, количество ссылок, имя пользователя, имя группы, размер файла, время последнего изменения файла и имя файла/директории) `'ls -al'` вводится как

```
system(myri,'ls -al')
```

`openShell` - функция открывающая SSH терминал который подключен к интерфейсу командной строки Linux, например,

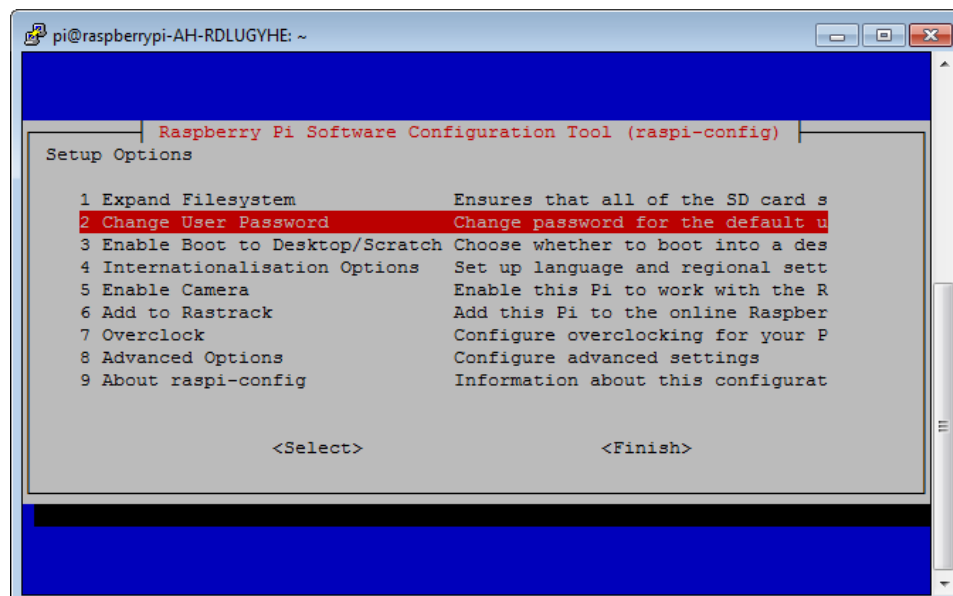
```
myri = raspi()  
openShell(myri)
```



```
pi@raspberrypi-AH-RDLUGYHE: ~  
login as: pi  
pi@raspberrypi-AH-RDLUGYHE's password:  
Linux raspberrypi-AH-RDLUGYHE 3.10.24+ #614 PREEMPT Thu Dec 19 20:38:42 GMT 2013  
armv6l  
  
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: Sun Dec 29 14:07:29 2013 from 172.28.201.160  
pi@raspberrypi-AH-RDLUGYHE ~ $
```

Хорошей мерой обеспечения безопасности является установка своего пароля. Это можно сделать во втором пункте настройки конфигурации компьютера RPi которая вызывается командой RPi.

```
sudo raspi-config
```



```
pi@raspberrypi-AH-RDLUGYHE: ~  
Raspberry Pi Software Configuration Tool (raspi-config)  
Setup Options  
  
1 Expand Filesystem           Ensures that all of the SD card s  
2 Change User Password       Change password for the default u  
3 Enable Boot to Desktop/Scratch Choose whether to boot into a des  
4 Internationalisation Options Set up language and regional sett  
5 Enable Camera              Enable this Pi to work with the R  
6 Add to Rastrack            Add this Pi to the online Raspber  
7 Overclock                  Configure overclocking for your P  
8 Advanced Options           Configure advanced settings  
9 About raspi-config         Information about this configurat  
  
<Select>                       <Finish>
```

После установки нового пароля переподключите RPi:

```
clear mypi
mypi = raspi('192.168.1.232','pi','newpassword')
```

РАБОТА С ФАЙЛАМИ КОМПЬЮТЕРА RASPBERRY PI

Через МатЛАБ соединение можно

- загружать файлы RPi в хост компьютер,
- выгружать файлы в RPi и
- удалять файлы RPi

Для загрузки файла RPi в хост компьютер наберите команды просмотр каталога `ls` и загрузка файла `getFile`:

```
system(mypi,'ls')
getFile(mypi,'/home/pi/.profile')
```

ПО дефолту `getFile` сохраняет файлы в текущей папке МатЛАБ. Для загрузки файла в другой каталог, например, C:\Users\myusername\Desktop используется команда

```
getFile(mypi,'/home/pi/profile','C:\Users\myusername\Desktop')
```

Для передачи файла в RPi используется команда `putFile`, например,

```
putFile(mypi,'C:\Users\myusername\Desktop\profile','/home/pi/')
```

Для удаления RPi файла используется команда `deletFile`, например,

```
deletFile(mypi,'myvideo.mp4')
```

ПАКЕТ БИБЛИОТЕКИ SIMULINK ДЛЯ РАБОТЫ С RPI

В библиотеке Simulink имеется “Simulink Support Package for Raspberry Pi Hardware” который включает пакет блоков и примеры подключения к устройствам и портам Raspberry Pi.

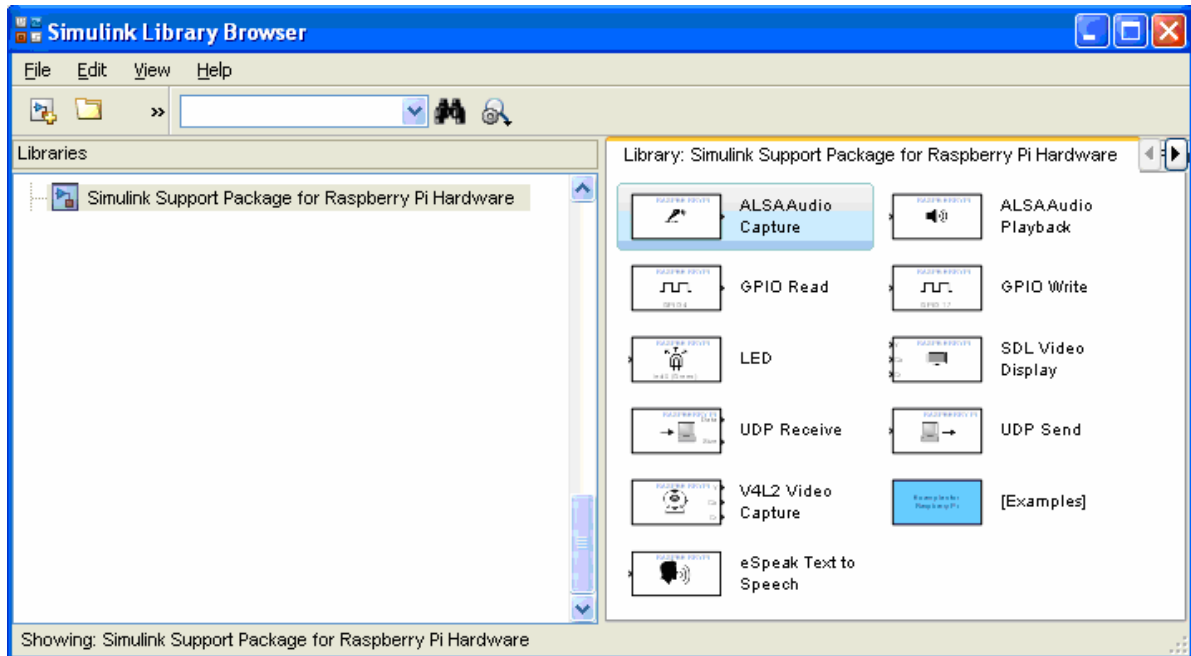
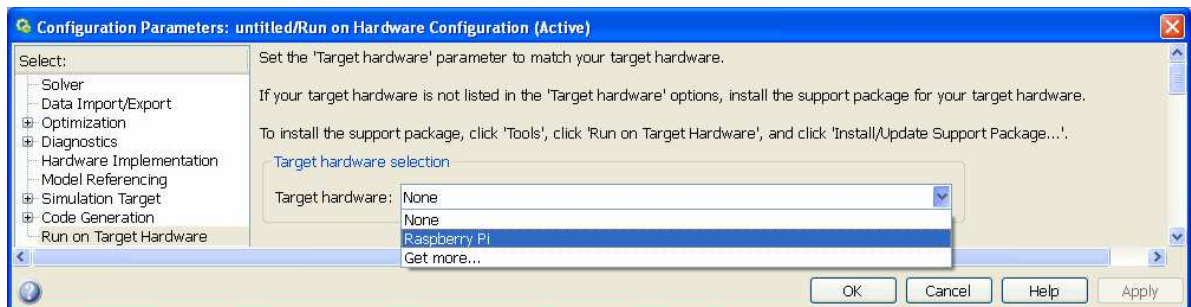


Рис. 8. Пакет библиотеки Simulink Support Package for Raspberry Pi Hardware включает блоки цифрового приема / передачи данных через выходы GPIO, управления светодиодом платы RPi, прием / передачу Audio и UDP данных, прием видео сигнала.

Для настройки модели на работу с RPi блоками необходимо выбрать Меню модели > Tools > Run on Target Hardware > Prepare to Run (или Options) > Run on Target Hardware > Target Hardware > Raspberry Pi > Apply > OK



ПРИМЕРЫ ПОЛУЧЕНИЯ ПРОВЕРЕННЫХ РЕЗУЛЬТАТОВ И ВАРИАНТЫ ДЛЯ САМОКОНТРОЛЯ

Задание 1. Работа с Raspberry через командную строку МатЛАБ

1. Соедините RPi и основной компьютер через нуль-модемный кабель: подключите кабель к RPi через преобразователь напряжений 3.3В/12В и к COM порту основного компьютера (Рис. 9) или к USB порту через адаптер Uport1100 (Рис. 10).

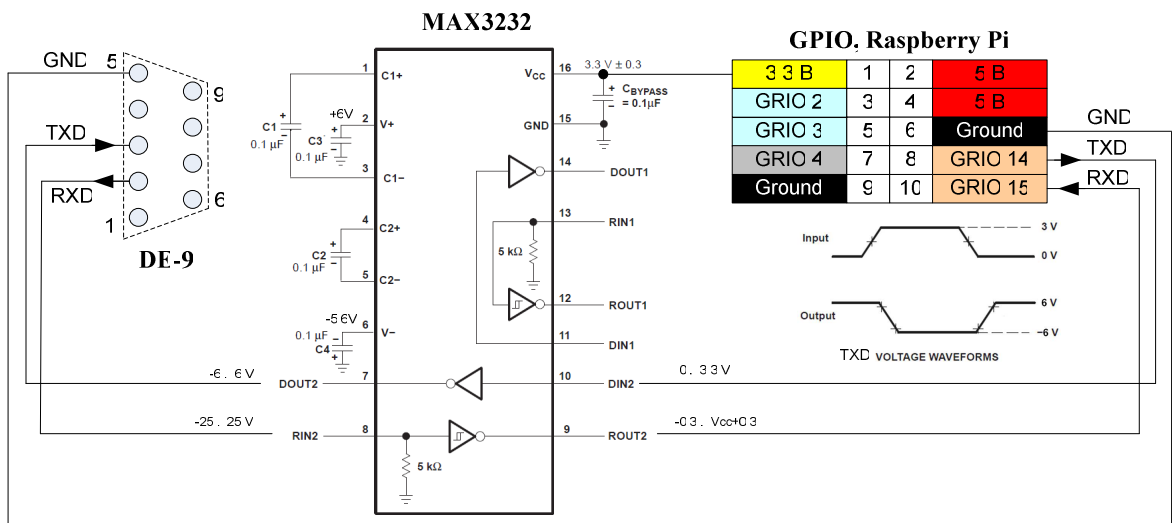


Рис. 9. Схема согласования уровней напряжения порта GPIO (0/3.3 В) и устройства с RS – 232 интерфейсом (-12/+12 В), подключаемого к разъёму DE-9.

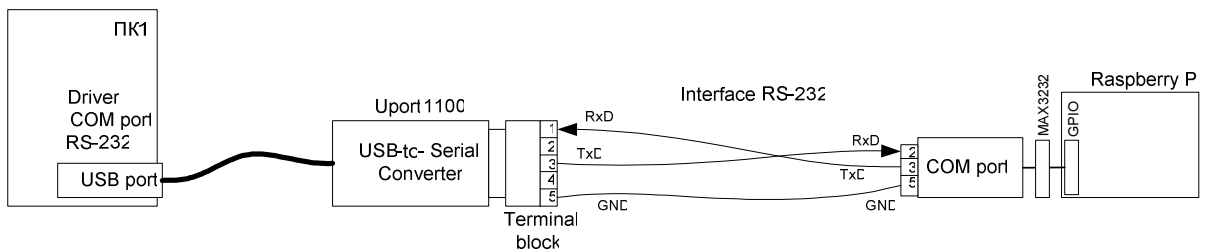


Рис. 10. Схема подключения Raspberry Pi к ПК через USB / COM порты и USB/RS-232 адаптер Uport 1100.

2. Загрузите МатЛАБ.
3. Сделайте соединение с Raspberry Pi, например,


```
>> mypi = raspi      % или
>> mypi = raspi('192.168.1.232','pi','rasberry')
```
4. Рассмотрите список полученных параметров RPi
5. Найдите местоположение светодиода на плате RPi состоянием которого можно управлять.


```
>> showLEDs(mypi)
```
6. Включите светодиод на плате RPi


```
>> writeLED(mypi,'led0',1); % или
>> writeLED(mypi,'led0',true)
```
7. Выключите светодиод


```
>> writeLED(mypi,'led0',0); % или
>> writeLED(mypi,'led0',false)
```
8. Проверьте статус COM порта


```
>> system(mypi, 'rpi-serial-console status')
```
9. Настройте конфигурацию выводов разъема GPIO на работу в режиме RS-232 (COM порта).


```
>> myserialdevice = serialdev(myri,'/dev/ttyAMA0')
```

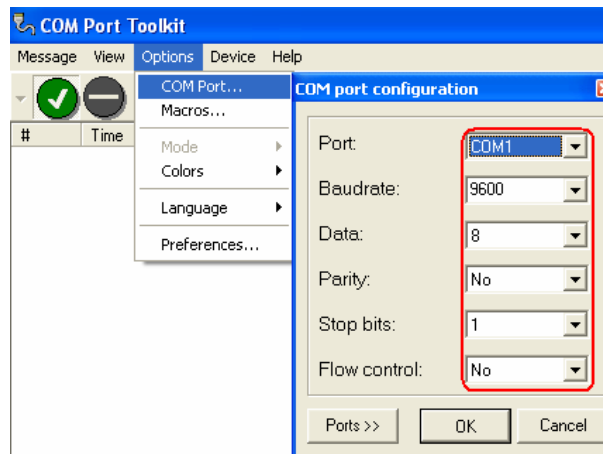
10. Установите скорость порта 9600 бод..

```
>> clear myserialdevice;
```

```
>> myserialdevice = serialdev(myri,'/dev/ttyAMA0',9600);
```

11. На основном компьютере загрузите утилиту COM Port Toolkit для обмена данными с RPi через COM порт.

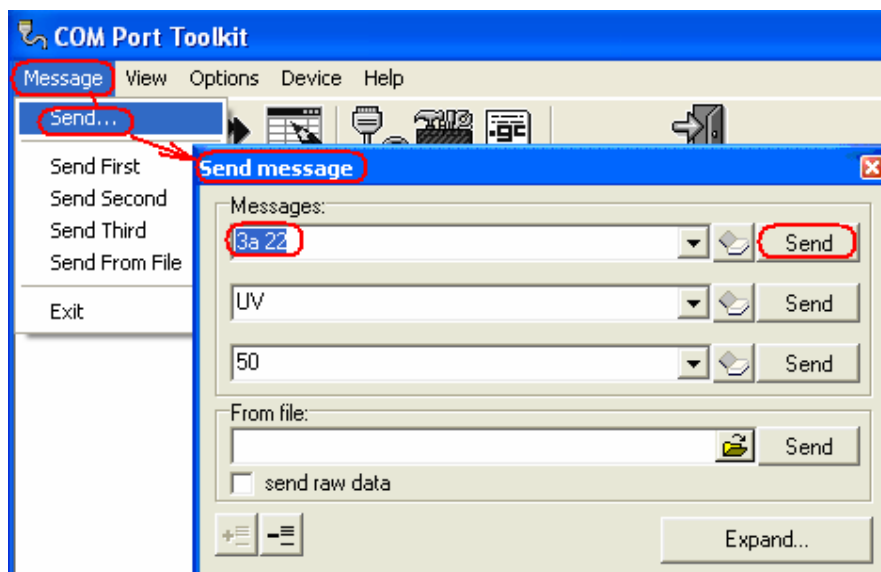
12. Настройте параметры COM соединения утилиты так, чтобы они совпадали с соответствующими параметрами RPi.



13. Используя панель передачи данных утилиты (Меню утилиты > Message > Send > Send Message) и окно отображения принятых данных а также соответствующие команды МатЛАБ для RPi убедитесь в работоспособности канала последовательной передачи данных между основным компьютером и RPi.

```
>> write(myserialdevice,[10 12], 'uint16')
```

```
>> output = read(myserialdevice,100);
```



14. Введите команду отключения COM порта после перезагрузки RPi

```
>> system(myri, 'sudo rpi-serial-console disable');
```

15. Удалите COM соединение myserial

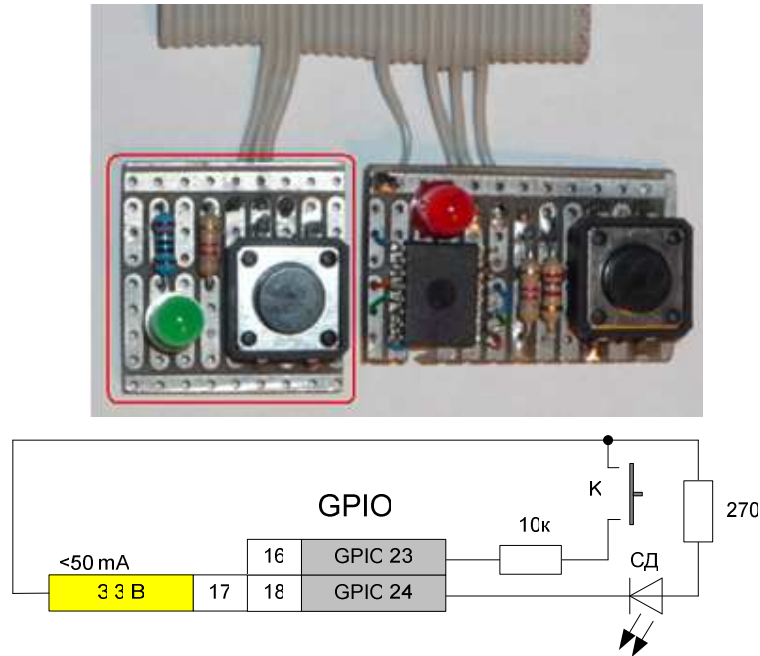
```
>> clear myserialdevice;
```

16. Выключите Raspberry Pi

```
>> system(myPi, 'sudo shutdown -r now');
```

Задание 2. Ручное переключение выводов GPIO в режиме терминала.

1. При выключенном компьютере RPi подключите к выводам разъёма GPIO (Рис. 4) светодиод и механический замыкатель контактов (кнопка К) как показано на Рис. 11.



Примечание. Не забывайте, что у выводы GPIO не защищены от короткого замыкания и высокого (> 3.3 В) напряжения .

Рис. 11. Электрическая схема подключения внешних устройств к разъему GPIO сконфигурированного на прием / передачу цифровых сигналов. Кнопка “К” специально подключена так, чтобы в отключенном состоянии на входе GPIO 23 был бы не определенный потенциал. Для обозначения этого состояния логическим нулем используется команда `GPIO.setup(16, GPIO.IN, pull_up_down=GPIO.PUD_DOWN`. В этом примере периферия подключена к разъему GPIO через стандартный шлейф-кабель для FDD (34 жилы), показан фрагмент.

2. Загрузите МатЛАБ и сделайте соединение с Raspberry Pi

```
>> myPi = raspi
```

3. Откройте схему расположения контактов GPIO компьютера RPi. Найдите контакты к которым подключена кнопка и светодиод (см. Рис. 2).

```
>> showPins(myPi)
```

4. Откройте SSH терминал и введите login (pi) и password RPi (raspberrypi)

```
>> openShell(myPi)
```

```
pi@raspberrypi-aisKzIrq16: ~
login as: pi
pi@192.168.1.232's password:
Linux raspberrypi-aisKzIrq16 3.10.25+ #622 PREEMPT Fri Jan 3 18:41:00 GMT 2014 a
rmy61

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: Fri Jan 10 12:52:10 2014 from 172.28.195.249
pi@raspberrypi-aisKzIrq16 ~ $ █
```

5. Загрузите интерпретатор Python. Обратите внимание на изменение указателя начала строки: при успешной загрузке интерпретатора вместо \$ появляется >>>.
\$ sudo python
6. Импортируйте библиотеку для работы с GPIO
>>> import RPi.GPIO as GPIO
7. Установите способ нумерации выводов GPIO (BCM или BOARD – номера физических контактов разъема)
>>> GPIO.setmode(GPIO.BOARD)
8. Сконфигурируйте выходы для работы в режиме передачи (OUT) и приёма (IN)
>>> GPIO.setup(18, GPIO.OUT) или
>>> GPIO.setup(18, GPIO.OUT, initial=GPIO.LOW)
>>> GPIO.setup(16, GPIO.IN) или
>>> GPIO.setup(16, GPIO.IN, pull_up_down=GPIO.PUD_UP) или
>>> GPIO.setup(16, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
9. Запишите “1”, а затем “0” в выходной порт. Наблюдайте за состоянием светодиода.
>>> GPIO.output(18, HIGH) или
>>> GPIO.output(18, 1) или
>>> GPIO.output(18, LOW) или
>>> GPIO.output(18, 0)
10. Считайте состояние кнопки
>>> GPIO.input(16)
11. Завершите работу возвращением всех выводов GPIO в исходное состояние
>>> exit()GPIO.cleanup()
12. Выйдите из интерпретатора в консоль: exit() или <Ctrl + D>.

Задание 3. Создание и запуск программы Python через терминал.

1. Для создания кода программы, например, `tsk_5.py` - бесконечного переключения светодиода перейдите в каталог `home/pi/` создайте подкаталог `st_07_09` и откройте в нём текстовый редактор Notepad `nano`.

```
$ cd /home/pi
$ dir
$ mkdir st_07_09
$ cd st_07_09
$ dir
$ nano tsk_3a.py
```

2. Напишите программу на языке Python которая каждую секунду поочередно включает и выключает светодиод (СД) и выводит его состояние на экран. СД подключен к 18-му контакту разъёма GPIO (см. Рис. 11).

```
#!/usr/bin/ python
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setup(18, GPIO.OUT)
GPIO.output(18, True)
LEDState == 0
while 1:
    if LEDState == 0
        GPIO.output(18, False)
        print ('1')
        time.sleep(1)
        LEDState = 1
    else:
        GPIO.output(18,True)
        print ("0")
        time.sleep(0.5)
        LEDState = 0
```

3. Выйдите из редактора: `Ctrl + X` затем `Y`.
4. Запустите файл на исполнение. Проверьте работу программы по миганию светодиода.

```
$ sudo python tsk_3a.py
```

5. Остановите выполнение программы.

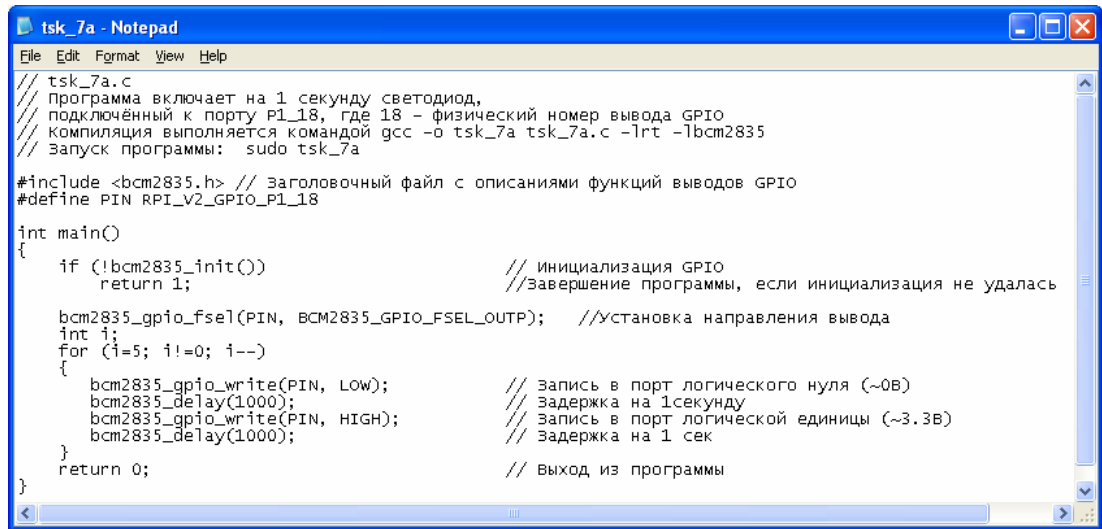
```
<Ctrl C>
```

6. Выключите Raspberry Pi.

```
$ sudo shutdown -h now
```

Задание 4. Копирование и запуск С-программы управления выводами GPIO.

1. В рабочем каталоге хост компьютера при помощи текстового редактора, например, Notepad создайте С-программу переключения светодиода, подключенного к 18 контакту GPIO.



```
tsk_7a.c
// программа включает на 1 секунду светодиод,
// подключённый к порту P1_18, где 18 - физический номер вывода GPIO
// компиляция выполняется командой gcc -o tsk_7a tsk_7a.c -lrt -lbcm2835
// Запуск программы: sudo tsk_7a

#include <bcm2835.h> // Заголовочный файл с описаниями функций выводов GPIO
#define PIN RPI_V2_GPIO_P1_18

int main()
{
    if (!bcm2835_init()) // инициализация GPIO
        return 1; // завершение программы, если инициализация не удалась

    bcm2835_gpio_fsel(PIN, BCM2835_GPIO_FSEL_OUTP); // установка направления вывода
    for (i=5; i!=0; i--)
    {
        bcm2835_gpio_write(PIN, LOW); // запись в порт логического нуля (~0В)
        bcm2835_delay(1000); // задержка на 1 секунду
        bcm2835_gpio_write(PIN, HIGH); // запись в порт логической единицы (~3.3В)
        bcm2835_delay(1000); // задержка на 1 сек
    }
    return 0; // выход из программы
}
```

```
// tsk_4a.c
// Программа включает на 1 секунду светодиод,
// подключённый к порту P1_18, где 18 – физический номер вывода GPIO
// Компиляция выполняется командой gcc -o tsk_4a tsk_4a.c -lrt -lbcm2835
// Запуск программы: sudo tsk_4a

#include <bcm2835.h> // Заголовочный файл с описаниями функций выводов GPIO
#define PIN RPI_V2_GPIO_P1_18

int main()
{
    if (!bcm2835_init()) // Инициализация GPIO
        return 1; // завершение программы, если инициализация не
    удалась

    bcm2835_gpio_fsel(PIN, BCM2835_GPIO_FSEL_OUTP); // установка
    направления вывода
    int i;
    for (i=5; i!=0; i--)
    {
        bcm2835_gpio_write(PIN, LOW); // запись в порт логического нуля (~0В)
        bcm2835_delay(1000); // задержка на 1 секунду
        bcm2835_gpio_write(PIN, HIGH); // запись в порт логической единицы
        (~3.3В)
        bcm2835_delay(1000); // задержка
    }
    return 0; // Выход из программы
}
```

2. Загрузите МатЛАБ и сделайте соединение с Raspberry Pi, например,


```
>> mypi = raspi
```
3. Скопируйте С-программу из рабочего каталога хост-компьютера в каталог /home/pi/st_07_09 RPi


```
>> putFile(mypi,'c:\tmp_MatLAB2014a_Raspberry_Pi\tsk_4a.c','/home/pi/st_07_09')
```
4. Выведите на экран путь текущего каталога RPi


```
>> system(mypi,'pwd')
```
5. Используя Linux команды **cd** и **pwd** и команду МатЛАБ **system** перейдите к каталог /home/pi/st_07_09 и раскройте его содержимое.


```
>> system(mypi,'ls -al')
```
7. Убедившись, что С-программа находится в текущем каталоге /home/pi/st_07_09, скомпилируйте программу.


```
>> system(mypi,'gcc -o tsk_4a tsk_4a.c -lrt -lbcm2835'),
```

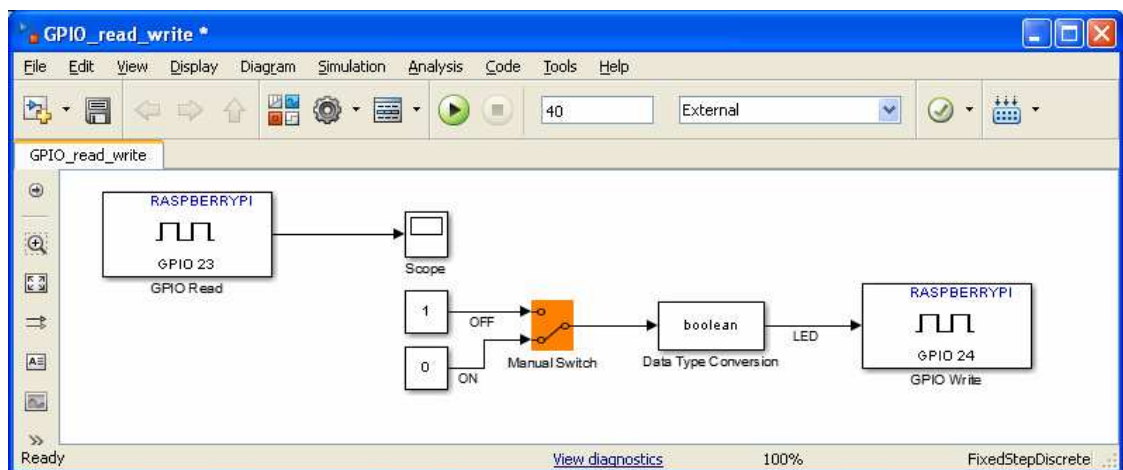
 где gcc- имя компилятора; bcm2835 - библиотека.
 Примечание. Если компилятор не выдал никаких сообщений, то компиляция прошла успешно.
 Если компиляция не прошла, установите на RPi библиотеку **bcm2835-1.17** как показано, например, в работе [3].
6. Проверьте появление исполняемого файла tsk_4a (без расширения) в текущем каталоге командой


```
>> system(mypi,'ls')
```
7. Запустите исполняемый файл. Проверьте работу программы по миганию светодиода.

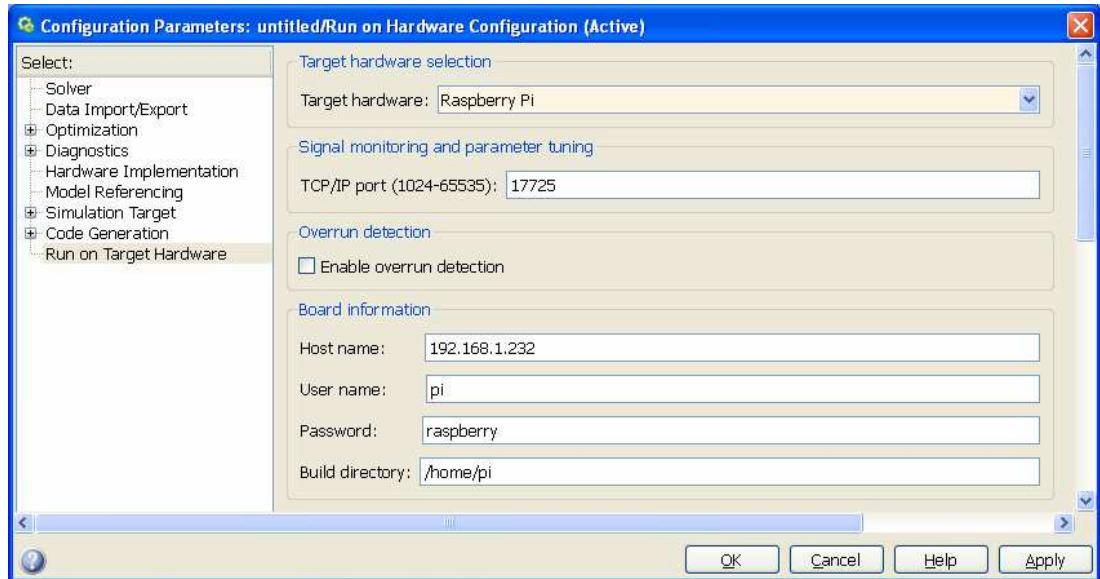
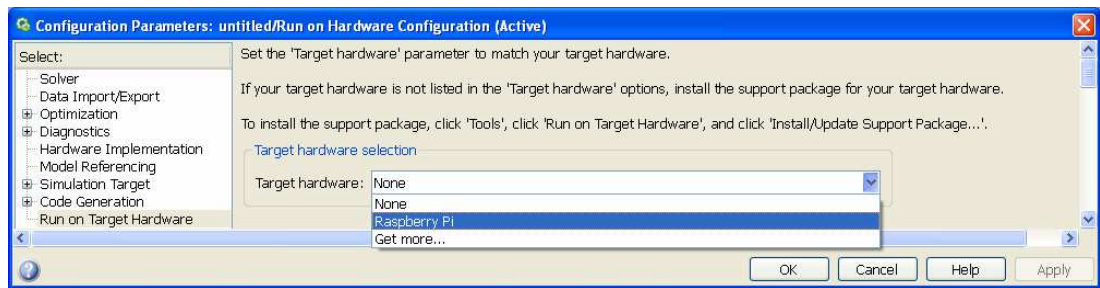

```
>> system(mypi,'sudo ./tsk_4a')
```
8. Клавишами Ctrl C прервите выполнение программы.

Задание 5. Работа Simulink с портом GPIO Raspberry Pi.

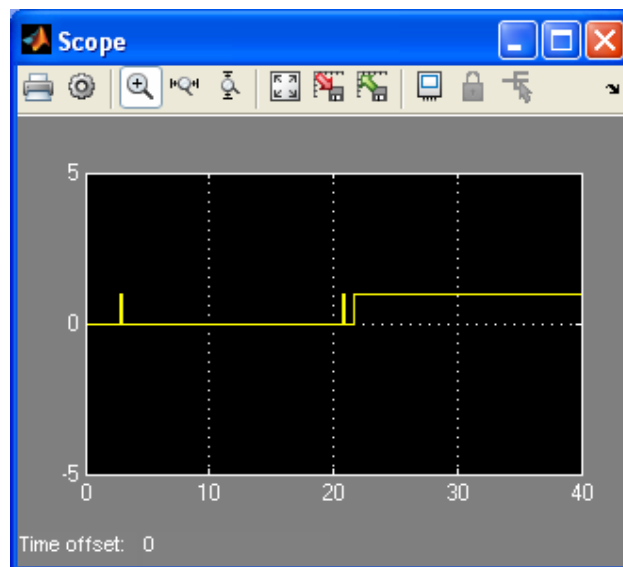
1. Создайте модель отображения состояния кнопки и включения / выключения светодиода подключенных к RPi по схеме Рис. 11.



2. Для настройки модели на работу с RPi выберите Меню модели > Tools >> Run on Target Hardware > Prepare to Run (или Options) > Run on Target Hardware > Target Hardware > Raspberry Pi > Apply > OK



3. Запустите модель и проверьте работу (в реальном времени) связи модели Simulink с периферией подключенной к порту GPIO компьютера RPi.



4. Выключите RPi через командную строку МатЛАБ.

```
>> mypi = raspi  
>> system(mypi, 'sudo shutdown -r now');
```

Задание 6. Связь МатЛАБ с датчиком температуры подключенным к I²C каналу Raspberry Pi.

1. Подключите к порту GPIO датчик температуры как показано на следующей схеме.

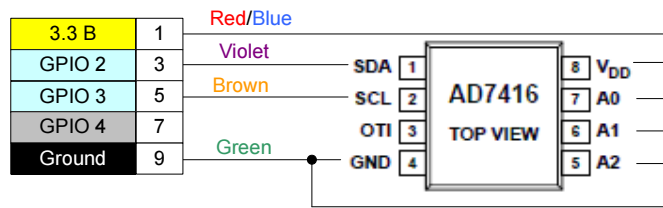


Рис. 12. Подключение датчика температуры к порту GPIO компьютера Raspberry Pi. Адрес датчика может устанавливаться в пространстве: 1001000 ... 1001111. Адрес подключенного датчика: 1001001₂ или 49₁₆ или 56₁₀. Диапазон измерения датчика от -40°C до +125°C.

2. Установите в МатЛАБ соединение с RPi через SSH сервер маршрутизатора.

```
>> mypi = rasp
```
3. Просканируйте I²C канал Raspberry Pi для определения адресов устройств подключенных к I²C.

```
>> scanI2CBus(mypi,'i2c-1')
i2cdevice =
i2cdev with properties:
  Bus: 'i2c-1'
  Address: '0x49'
```
4. Создайте связь через идентификатор объекта с обнаруженным I²C устройством.

```
>> i2cdevice = i2cdev(mypi,'i2c-1','0x49')
```
5. Считайте показания датчика температуры

```
>> read(i2cdevice,1)
```
6. Переконфигурируйте выводы I²C канала разъема GPIO на работу в режиме цифрового ввода / вывода.

```
>> disableI2C(mypi)
```
7. Напишите m-программу считывания показаний температуры каждую секунду в течении минуты и последующего графического отображения полученных показаний. RPi должен включать собственный светодиод при превышении начальной температуры, например, на 2 градуса. Проверьте работу программы.

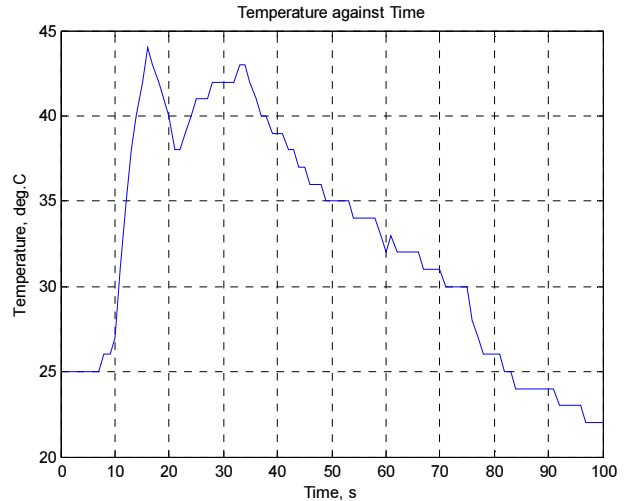

```

temp_ini = read(i2cdevice,1);

for i = 1:60
    temp(i) = read(i2cdevice,1);
    pause(1);
    if temp(i) > temp_ini + 2
        writeLED(myPi, 'led0', 1);
    else
        writeLED(myPi, 'led0', 0);
    end
    i
end

figure
time = 1:length(temp);
plot(time,temp);
grid on
xlabel('Time, s');
ylabel('Temperature, deg.C');
title('Temperature against Time');

```



КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Чем отличается компьютер от контроллера?
2. Перечислите возможные варианты связи МатЛАБ с компьютером Raspberry Pi.
3. Назовите средства RPi которые можно использовать для подключения датчиков и исполнительных устройств систем управления.
4. Почему кнопка в Задании 5 не работает, тогда как эта же кнопка работает в Задании 2?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. OS Raspbian downloads <http://www.raspberrypi.org/downloads>
2. Help MatLAB
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 МатЛАБ <http://portalnp.ru/2014/03/1783>
5. Dr. Bob Davidov. Компьютерные технологии управления в технических системах <http://portalnp.ru/author/bobdavidov>.