МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Отчёт по лабораторной работе № 1 по курсу «Архитектура мобильных платформ»**

«Использование интерфейса ввода/вывода общего назначение на одноплатном компьютере Raspberry Pi 3»

Вариант №10

ВЫПОЛНИЛ студент группы 16-ИТ-3

Яблонский А.С.

ПРОВЕРИЛ преподаватель

Лукьянов А.О.

Полоцк, 2019 г.

**ЦЕЛЬ РАБОТЫ:** Освоить навыки использования интерфейса общего назначения одноплатного компьютера.

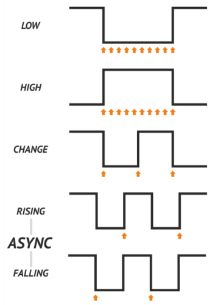


**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ:**

1. Дайте определение понятию «Интерфейс ввода/вывода общего назначения».

Интерфейс ввода/вывода общего назначенияb— интерфейс для связи между компонентами компьютерной системы, к примеру микропроцессором и различными периферийными устройствами.

1. Перечислите режимы работы прерываний.

Каждый из 28 выводов в режиме INPUT может генерировать прерывания — по спаду, по фронту, по единице, по нулю, по изменению сигнала, а также в асинхронном режиме по фронту и по спаду: 

1. Какие электрические характеристики имеют GPIO Raspberry Pi 3?

Raspberry Pi 3 имеет 40-пиновую рейку GPIO. Однако называть все 40-пинов выводами GPIO некорректно, так как 12 из них представляют из себя линии питания 3.3 В, 5 В и общий вывод GND (земля)

Также 27 (BCM 0) и 28 (BCM 1) выводы используются для конфигурации EEPROM Raspberry Pi для работы с HAT-устройствами (Hardware Attached on Top — устройства поверхностного монтажа, по сути — обычные платы расширения) и использование этих выводов крайне не рекомендуется. Тем не менее, они являются полноценными GPIO-выводами.

Фактически получается, что GPIO-выводов не 40, а 28.

Максимальный выходной ток каждого вывода не должен превышать **16 мА**. Суммарный выходной ток всех выводов не должен превышать **50 мА**. 5-вольтовые линии могут давать больший ток, который остается после питания Raspberry Pi 3 и других периферийных устройств (клавиатуры, мыши) — до **500 мА**.

1. Каким образом выполняется управление GPIO Raspberry Pi 3?

Raspberry Pi представляет собой плату размером чуть больше кредитной карты, на которой распаян ARM-процессор, чипы оперативной памяти, слот под microSD-карту, а также Ethernet-порт, HDMI, 3,5мм аудиовыход и USB-порты для подключения периферийных устройств. Кроме того, как и на Arduino, на плате Raspberry Pi имеется GPIO-интерфейс. Все это работает под управлением адаптированного под ARM-архитектуру дистрибутива \*unix.

1. Опишите структуру регистров GPIO Raspberry Pi 3.

Режим, в котором работает каждый отдельный разряд порта GPIO, управляется полностью программным способом.

Процессор BCM2835 имеет 41 32-разрядный регистр, которые полностью определяют режим и состояние портов GPIO. В частности, для установки единичного значения на выводе, запрограммированном на работу как выход, необходимо записать единичный бит в соответствующий разряд одного из двух регистров установки битов GPIO Pin Output Set Registers (GPSETn). Чтобы установить выход в ноль, следует выставить единичный бит в регистрах сброса битов GPIO Pin Output Clear Registers (GPCLRn). Такая схема позволяет независимо устанавливать и сбрасывать любой бит GPIO без необходимости чтения текущего состояния выводов.

Аналогично, когда разряды GPIO работают на чтение, то узнать уровень входного сигнала можно, прочитав значение одного из двух портов GPIO Pin Level Registers (GPLEVn), каждый бит которого отображает текущее состояние входного разряда.

Регистры, отвечающие за работу с GPIO, расположены по адресам 0x7E200000—0x7E2000B0, которые отображаются на физическую память с адресами, начинающимися с 0x20200000 [4].

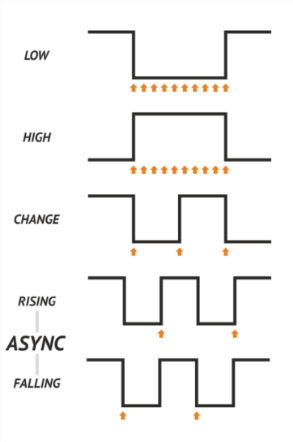
1. Назовите номиналы подтягивающих и стягивающих резисторов на выводах GPIO Raspberry Pi 3.

Номиналы сопротивлений не постоянны и равны:

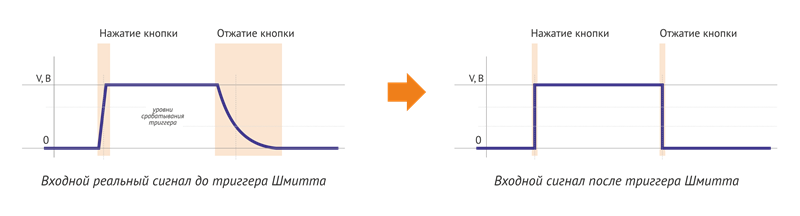
* для подтягивающего резистора 50 КОм — 65 КОм
* для стягивающего резистора 50 КОм — 60 КОм

1. Охарактеризуйте работу прерываний в различных режимах.

Каждый из 28 выводов в режиме *INPUT* может генерировать прерывания — по спаду, по фронту, по единице, по нулю, по изменению сигнала, а также в асинхронном режиме по фронту и по спаду:



Также все выводы в режиме *INPUT* имеют входную фильтрацию на триггере Шмитта (преобразовывают аналоговый сигнал в цифровой с резкими переходами между состояниями):



1. Назовите системные файлы, используемые для управления GPIO Raspberry Pi 3.

Программировать поведение GPIO можно на большом количестве различных языков — Pascal, Ruby, Perl, Java (Pi4J), C, C++, C#, WiringPi, Basic и т.д.

Также можно записать номер GPIO в файл *./export* в подкаталоге */sys/class/gpio/*, и система создаст файл со структурой GPIO согласно номеру входа.

Создание файла доступа GPIO:

*echo 12 > /sys/class/gpio/export*

1. Перечислите регистры управления GPIO Raspberry Pi 3.

Режим, в котором работает каждый отдельный разряд порта GPIO, управляется полностью программным способом.

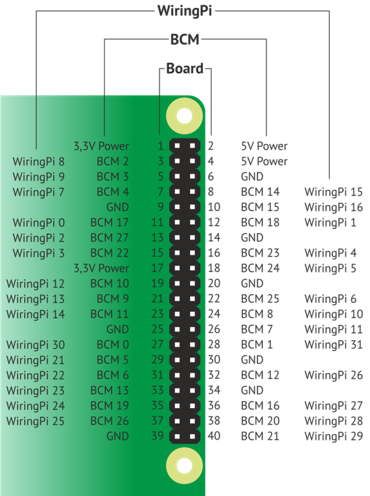
Процессор BCM2835 имеет 41 32-разрядный регистр, которые полностью определяют режим и состояние портов GPIO. В частности, для установки единичного значения на выводе, запрограммированном на работу как выход, необходимо записать единичный бит в соответствующий разряд одного из двух регистров установки битов GPIO Pin Output Set Registers (GPSETn). Чтобы установить выход в ноль, следует выставить единичный бит в регистрах сброса битов GPIO Pin Output Clear Registers (GPCLRn). Такая схема позволяет независимо устанавливать и сбрасывать любой бит GPIO без необходимости чтения текущего состояния выводов.

Аналогично, когда разряды GPIO работают на чтение, то узнать уровень входного сигнала можно, прочитав значение одного из двух портов GPIO Pin Level Registers (GPLEVn), каждый бит которого отображает текущее состояние входного разряда.

Регистры, отвечающие за работу с GPIO, расположены по адресам 0x7E200000—0x7E2000B0, которые отображаются на физическую память с адресами, начинающимися с 0x20200000.

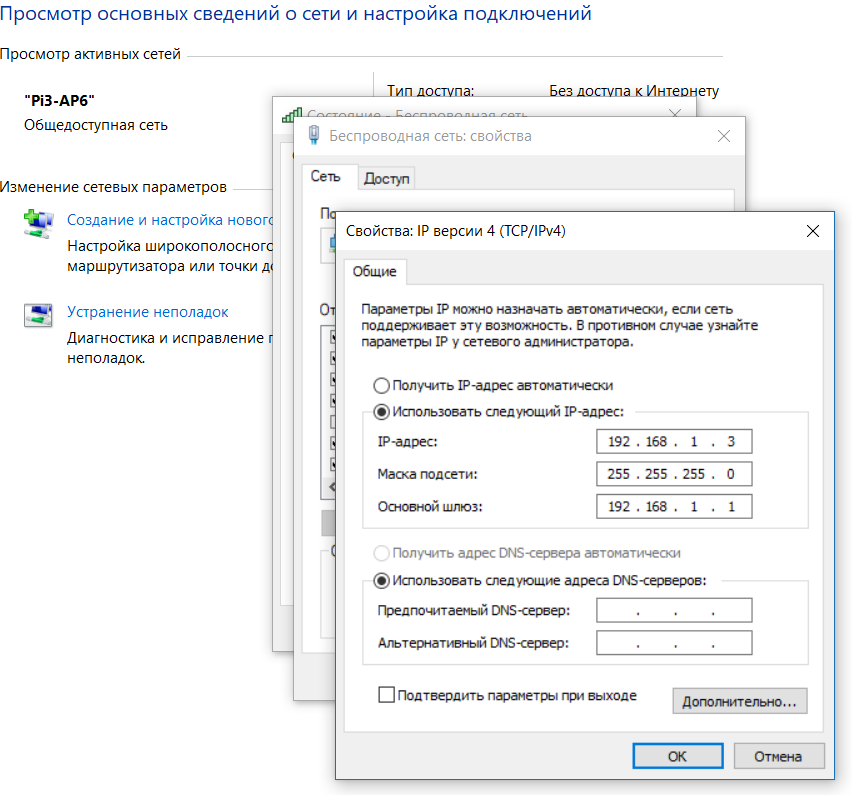
1. Перечислите и охарактеризуйте системы нумерации выводов Raspberry Pi 3.

Существует несколько вариантов нумерации выводов Raspberry Pi: Board (физическая нумерация по порядку) и BCM (нумерация из чипа). Также некоторые библиотеки, например, WiringPi, используют свою нумерацию



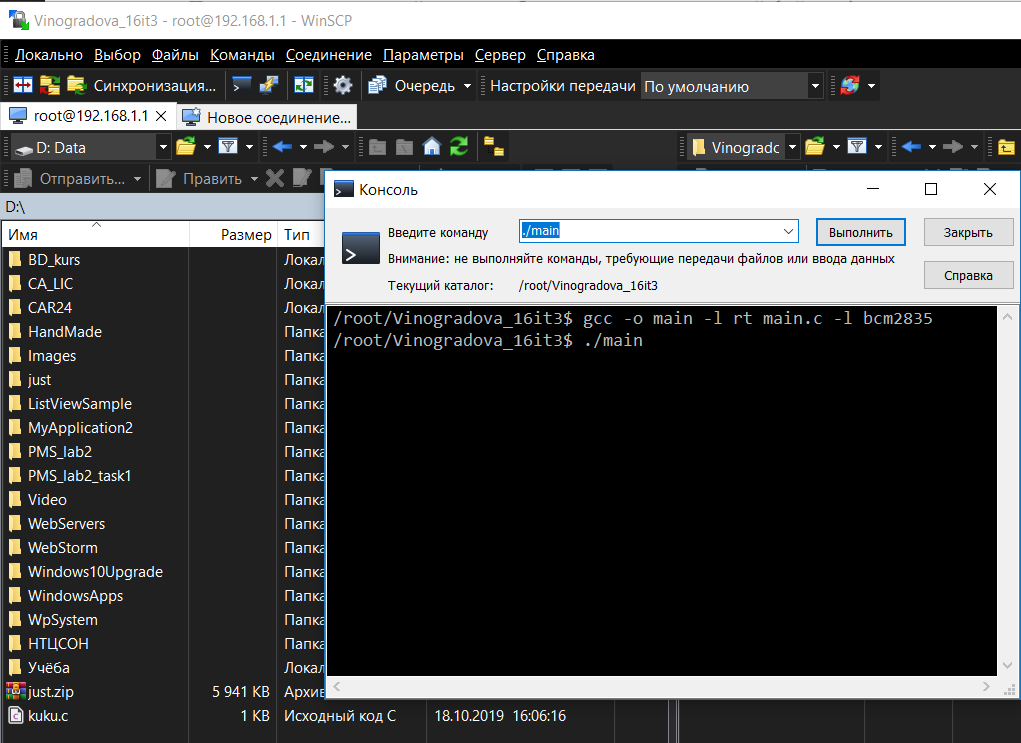
**ОПИСАНИЕ ПРОДЕЛАННОЙ РАБОТЫ:**

1. Подключился к Wi-Fi сети Raspberry Pi; по SFTP к Raspberry Pi по адресу 192.168.1.1 с помощью PuTTY и WinSCP, используя учётную запись пользователя root/root и настроил сеть:



**Рисунок 1** – Настройка сети

1. Создал каталог, в названии которого указал свою фамилию и номер группы.
2. Создала файл main.c с помощью команды nano main.c;
3. Скомпилировал получившийся файл с исходным кодом и запустил полученный исполняемый файл.



**Рисунок 2** – Запуск скомпилированного файла

**РЕЗУЛЬТАТЫ ВЫПОЛЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ**:

**Листинг приложения**

1. #include <bcm2835.h>
2. #define PIN RPI\_V2\_GPIO\_P1\_12
3. #define DURATION\_LONG 1000
4. #define DURATION\_SHORT 300
5. void hightOff(long duration){
6. bcm2835\_gpio\_write(PIN, LOW);
7. bcm2835\_delay(duration);
8. }
9. void hightOn(long duration){
10. bcm2835\_gpio\_write(PIN, HIGH);
11. bcm2835\_delay(duration);
12. }
13. int main(int argc, char \*\*argv){
14. if (!bcm2835\_init()){
15. return 1;
16. }
17. bcm2835\_gpio\_fsel(PIN, BCM2835\_GPIO\_FSEL\_OUTP);
19. hightOn(DURATION\_SHORT);
20. hightOff(DURATION\_SHORT);
22. hightOn(DURATION\_SHORT);
23. hightOff(DURATION\_LONG);
25. hightOn(DURATION\_LONG);
26. hightOff(DURATION\_LONG);
28. bcm2835\_close();
29. return 0;
30. }

**Вывод:** в результате выполнения лабораторной работы я освоил навыки использования интерфейса общего назначения одноплатного компьютера.