МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Отчёт по лабораторной работе № 2 по курсу «Архитектура мобильных платформ»**

«Использование широтно-импульсной модуляции на одноплатном компьютере Raspberry Pi 3»

Вариант №3

ВЫПОЛНИЛ студент группы 16-ИТ-3

Виноградова А.Д.

ПРОВЕРИЛ преподаватель

Лукьянов А.О.

Полоцк, 2019 г.

**ЦЕЛЬ РАБОТЫ:** Ознакомиться с принципами формирования ШИМ-сигнала. Научиться использовать интерфейс ввода/вывода общего назначения на Raspberry Pi 3 в режиме ШИМ.



**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ:**

1. **Дайте определение понятия ШИМ.**

ШИМ или PWM (широтно-импульсная модуляция) –это способ управления подачей мощности к нагрузке. Управление заключается в изменении длительности импульса при постоянной частоте следования импульсов.

1. **Как связана скважность с коэффициентом заполнения импульса?**

Отношение полной длительности периода импульса ко времени включения (положительной части импульса) называется скважностью импульса. Величина обратная скважности называется коэффициентом заполнения импульса

**3. Сколько каналов аппаратного ШИМ поддерживает Raspberry Pi 3 и на каких GPIO?**

ШИМ является дополнительной функцией портов GPIO 12, 32, 33, 35. Чтобы воспользоваться аппаратными ШИМ, необходимо установить соответствующий порт в режим дополнительных функций и задать процессору параметры ШИМ-сигнала.

Соответствие порта GPIO и режима дополнительных функций следующее:

\* GPIO 12 (PWM channel 0) – режим Alt5;

\* GPIO 32 (PWM channel 0) – режим Alt0;

\* GPIO 33 (PWM channel 1) – режим Alt0;

\* GPIO 35 (PWM channel 1) – режим Alt5.

**4. Перечислите преимущества и недостатки аппаратного и программного ШИМ.**

*Достоинство аппаратной реализации ШИМ сигнала* - это низкая загрузка микроконтроллера (прерывание вызывается один раз в период ШИМ сигнала), простота использования и точность (если в системе мало прерываний). *Из недостатков можно отметить* - ограниченное разрешение счетчиков, невысокая частота, ограниченное число каналов, на которых можно генерировать ШИМ сигналы. Хотя существуют специальные микроконтроллеры специально "заточенные" для генерации большого количества ШИМ сигналов.

*Преимущество программной реализации* - простота, неограниченное количество каналов, неограниченное разрешение. Конечно, условно неограниченное, с учетом доступной памяти. Недостатки программной реализации - высокая загрузка микроконтроллера. Прерывания должны вызываться на каждый инкремент счетчика и каждый раз нужно проверять не достиг ли он одного из крайних значений. Также программная реализация имеет меньшую точность (большее дрожание фронтов сигнала) и еще меньшую частоту (из-за первого недостатка).

Однако, несмотря на это, программная реализация ШИМа тоже имеет место быть, если требуется генерировать постоянный аналоговый сигнал или переменный, но с невысокой частотой.

**5. Опишите механизм передачи параметров командной строки.**

При запуске программы из командной строки, ей можно передавать дополнительные параметры в текстовом виде. В программе эти параметры из командной строки можно получить через аргументы функции main при использовании её в следующей форме

void main(int argc, char \*\*argv) {

...

}

Первым аргументом argc является число переданных функции параметров. Второй аргумент argv – массив строк – переданные параметры. Так как параметры у функции могут быть любыми, то они передаются как строки, и уже сама программа должна их разбирать и приводить к нужному типу.

Первым аргументом (argv[0]) всегда является имя программы. При этом имя выводится в зависимости от того, откуда была запущена программа [4].

**6. Что называется коэффициентом заполнения импульса?**

Величина обратная скважности называется коэффициентом заполнения импульса.

**7. Что называется скважностью ШИМ?**

Отношение полной длительности периода импульса ко времени включения (положительной части импульса) называется скважностью импульса.

**8. В каком виде передаются параметры командной строки при запуске программы?**

При запуске программы из командной строки, ей можно передавать дополнительные параметры в текстовом виде.

**9. Какие способы реализации ШИМ есть на Raspberry Pi 3?**

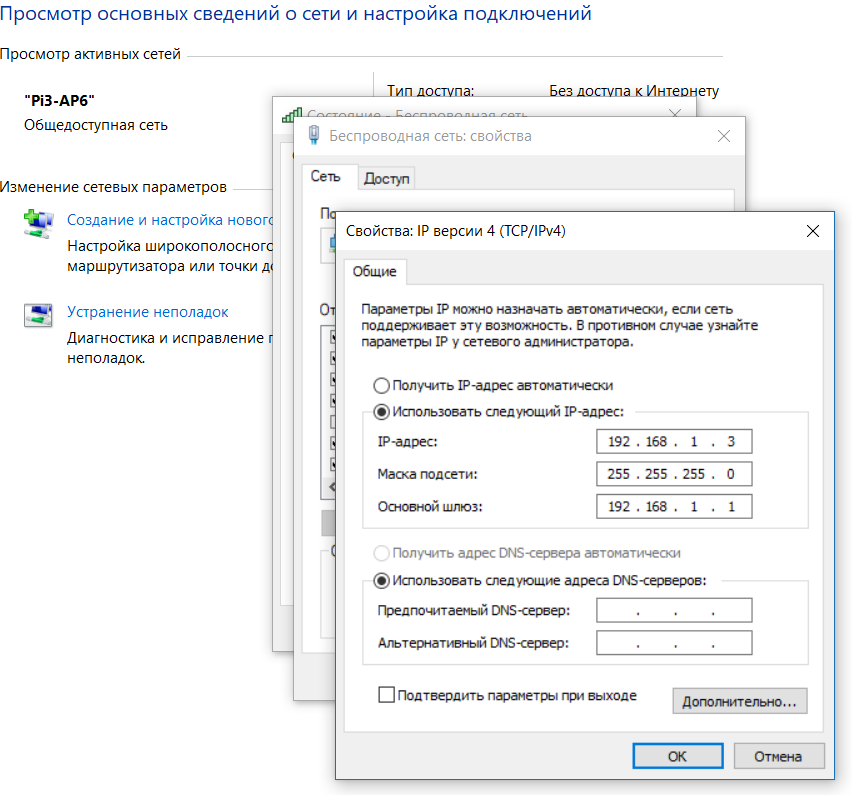
Программное и аппаратное формирование ШИМ-сигнала

**10. Для чего используется ШИМ?**

Широтно-импульсная модуляция позволяет регулировать яркость подсветки жидкокристаллических дисплеев сотовых телефонов, смартфонов, ноутбуков. ШИМ реализована в сварочных аппаратах, в автомобильных инверторах, в зарядных устройствах и т. д. Любое зарядное устройство сегодня использует при своей работе ШИМ.

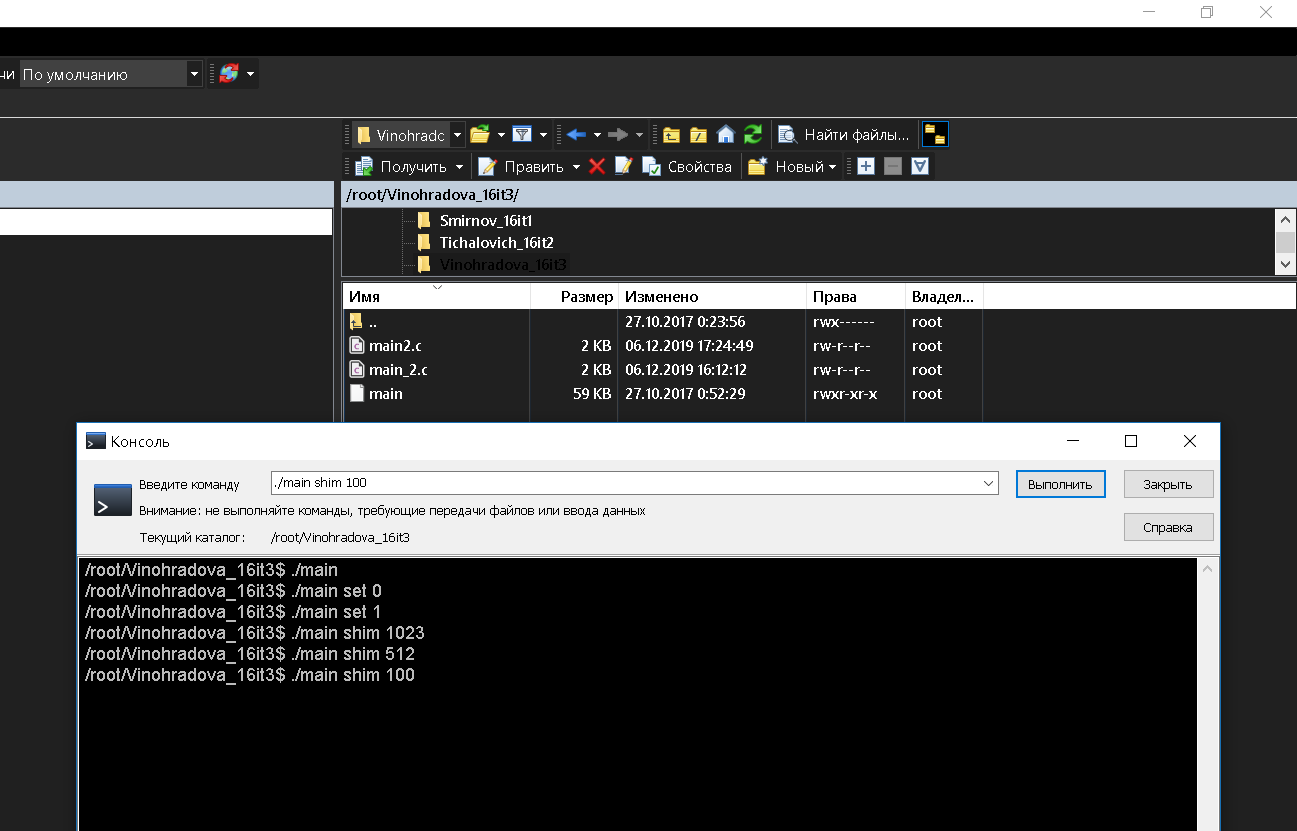
**ОПИСАНИЕ ПРОДЕЛАННОЙ РАБОТЫ:**

1. Подключилась 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 настроила сеть:



**Рисунок 1** – Настройка сети

1. Создала каталог, в названии которого указала свою фамилию и номер группы.
2. Создала файл main2.c с помощью команды nano main2.c;
3. Скомпилировала получившийся файл с исходным кодом и запустила полученный исполняемый файл.



**Рисунок 2** – Запуск скомпилированного файла

**РЕЗУЛЬТАТЫ ВЫПОЛЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ**:

**Листинг приложения**

#include <bcm2835.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define PIN RPI\_V2\_GPIO\_P1\_33

#define PWM\_CHANNEL 0

#define RANGE 1024

int main(int argc, char \*\*argv)

{

int direction = 1, data = 1, count = 0;

if (!bcm2835\_init())

return 1;

bcm2835\_gpio\_fsel(PIN, BCM2835\_GPIO\_FSEL\_ALT0);

bcm2835\_pwm\_set\_clock(BCM2835\_PWM\_CLOCK\_DIVIDER\_16);

bcm2835\_pwm\_set\_mode(PWM\_CHANNEL, 1, 1);

bcm2835\_pwm\_set\_range(PWM\_CHANNEL, RANGE);

if(argc > 2 && strcmp(argv[1], "set") == 0)

{

bcm2835\_gpio\_fsel(PIN, BCM2835\_GPIO\_FSEL\_OUTP);

int state = atoi(argv[2]);

bcm2835\_gpio\_write(PIN, state);

bcm2835\_close();

}

else if(argc > 2 && strcmp(argv[1], "shim") == 0)

{

bcm2835\_pwm\_set\_range(PWM\_CHANNEL, RANGE);

while (count < 1)

{

if (data == 1)

direction = 1;

else if (data >= atoi(argv[2])-1)

{

count++;

direction = -1;

}

data += direction;

bcm2835\_pwm\_set\_data(PWM\_CHANNEL, data);

}

}

else

{

bcm2835\_pwm\_set\_range(PWM\_CHANNEL, RANGE);

while (count < 1)

{

if (data == 1)

direction = 1;

else if (data >= RANGE\*0.2-1)

{

count++;

direction = -1;

}

data += direction;

bcm2835\_pwm\_set\_data(PWM\_CHANNEL, data);

bcm2835\_delay(1);

}

}

bcm2835\_close();

return 0;

}

**Вывод:** в результате выполнения лабораторной работы я ознакомилась с принципами формирования ШИМ-сигнал и научилась использовать интерфейс ввода/вывода общего назначения на Raspberry Pi 3 в режиме ШИМ.