|  |
| --- |
| Министерство образования Республики Беларусь  УО «Полоцкий государственный университет им. Ефросинии Полоцкой» |

|  |  |
| --- | --- |
|  | Факультет информационных технологий  Кафедра технологий программирования |

|  |
| --- |
| ОТЧЕТ |
| **к лабораторной работе №2**  **Дисциплина**: «Архитектура мобильных платформ» |

|  |  |
| --- | --- |
| Выполнил: | Студент группы 19-ИТ-2  Гилимович А.Н. |

|  |  |
| --- | --- |
| Проверил: | Преподователь  Панасик А.А. |

|  |
| --- |
| Полоцк, 2022 г. |

**ИСПОЛЬЗОВАНИЕ ШИРОТНО-ИМПУЛЬСНОЙ МОДУЛЯЦИИ НА ОДНОПЛАТНОМ КОМПЬЮТЕРЕ RASPBERRY PI 3**

**Вариант 1**

**Цель работы:** изучить принципы формирования ШИМ-сигнала. Освоить использование интерфейс ввода/вывода общего назначения на Raspberry Pi 3 в режиме ШИМ.

**Задание**

Модифицировать пример так, чтобы в качестве параметров командной строки программа принимала два аргумента: команда (установка уровня или скважности ШИМ) и значение (вкл./выкл. или значение скважности). Без указания параметров программа должна выполнить действие, согласно индивидуальному варианту.

**Ход работы**

Изучив теоритический материал преступил к выполнению заданий.

**Листинг 1** – задание

#include <bcm2835.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define PIN RPI\_GPIO\_P1\_12 //12

#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\_OUTP);

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)

{

printf("set\n");

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)

{

printf("shim\n");

bcm2835\_gpio\_fsel(PIN, BCM2835\_GPIO\_FSEL\_ALT5);

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

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

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

int direction = 1;

int data = 1;

int count = 0;

while (count < 5)

{

if (data == 1)

direction = 1;

else if (data == atoi(argv[2])){

count++;

direction = 0;}

else if (data == RANGE-1)

{

count++;

direction = -1;}

data += direction;

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

bcm2835\_delay(1);}}

else

{

bcm2835\_gpio\_fsel(PIN, BCM2835\_GPIO\_FSEL\_OUTP); // установка порта на вывод

unsigned int t\_on, t\_off; // продолжительность вкл. и выкл. состояния

int d = 5, i, j, flag=0;

// d- коэффициент заполнения в процентах

//i и j, вспомогательные переменные для организации циклов

//flag- если = 0 светодиод затухает, если = 1 разгорается

int a=10; // количество полных рабочих циклов

while (a)

{

for (j=50; j!=0; j--) // изменяем коэффициент заполнения от 100% до 0%

{

t\_on=50\*d; //находим время включения

t\_off=50\*(100-d); //находим время выключения

// если светодиод затухает, уменьшаем коэффициент заполнения

if (flag==0) d=d-1;

// если светодиод разгорается, увеличиваем коэффициент заполнения

if (flag==1) d=d+1;

// передаём 10 импульсов на светодиод с рассчитанными параметрами

for (i=10; i!=0; i--)

{

bcm2835\_gpio\_write(PIN, LOW);

delayMicroseconds(t\_on);

bcm2835\_gpio\_write(PIN, HIGH);

delayMicroseconds(t\_off);}

// если светодиод выключен, начинаем его включать

if (d==0) flag=1;

// если светодиод достиг максимума свечения, начинаем его гасить

if (d==100) flag=0;}

a--;}}

bcm2835\_close();

return 0;

}

**Вывод:** изучили принципы формирования ШИМ-сигнала. Освоили использование интерфейса ввода/вывода общего назначения на Raspberry Pi 3 в режиме ШИМ.