Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «**Информатика и системы управления (ИУ)**»

КАФЕДРА «**Программное обеспечение ЭВМ и информационные технологии (ИУ7)**»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОМУ ПРОЕКТУ

НА ТЕМУ:

«Драйвер для устройств с интерфейсом GPIO»

Студент группы **ИУ7-72Б** _____ **Иванов В.А.**
(Подпись, дата) (И.О. Фамилия)

Руководитель НИР _____ **Рязанова Н.Ю.**
(Подпись, дата) (И.О. Фамилия)

***2021** г.*

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В данный момент активно ведётся развитие технологий ”умного дома”и ”интернета вещей”. Они направлены на создание общей сети для любого типа домашней техники или механизмов. Целью является их тесная интеграция, создание возможности управления и получения информации об их текущем состоянии для жильца дома.

Разработка подобных ”умных”устройств тесно связана с использованием микроконтроллеров или одноплатных компьютеров. Такие устройства имеют небольшой размер и представляют незначительные вычислительные мощности. Поэтому для них зачастую требуется применение низкоуровневого программирования.

Наиболее распространённым физическим интерфейсом подключения является GPIO - контакты общего назначения, позволяющие подключать к себе широкий спектр различных устройств.

Данная работа посвящена разработке драйвера для взаимодействия с устройствами с интерфейсом GPIO.

1 Аналитическая часть

1.1 Постановка задачи

По заданию требуется разработать драйвер в виде загружаемого модуля ядра, позволяющий управлять устройствами, подключенными с помощью интерфейса GPIO.

Необходимо предоставить пользователю возможность ввода/вывода информации с устройств, управления режимами.

Для достижения данной цели необходимо решить следующие задачи:

1) ознакомиться с основными принципами работы устройств интерфейса GPIO;
2) определить способ управления устройствами;
3) выделить набор действий;
4) реализовать драйвер.

1.2 Принцип работы устройств GPIO

GPIO - интерфейс для связи между компонентами компьютерной системы, к примеру, микропроцессором и различными периферийными устройствами[1]. Контакты GPIO могут выступать как в роли входа, так и в роли выхода — это, как правило, конфигурируется. Обычно они используются для подключения датчиков, переключателей, дисплеев и т.п.

В данной работе будет использоваться одноплатный компьютер Raspberry Pi 2B ввиду отсутствия в распоряжении других компьютеров. Рассмотрим подробнее организацию GPIO на его примере.

На рисунке 1 описывается назначение каждого из контактов (пинов) для этой модели. Можно отметить, что не все из них используются для ввода/вывода. Помимо этого есть контакты предназначенные для подачи определённого напряжения на внешние устройства (3.3V, 5V) или для заземления (GROUND).

Пин GPIO имеет два режима:

- **Вход**. Напряжение подаётся внешним устройством. От +0.0В до +1.8В

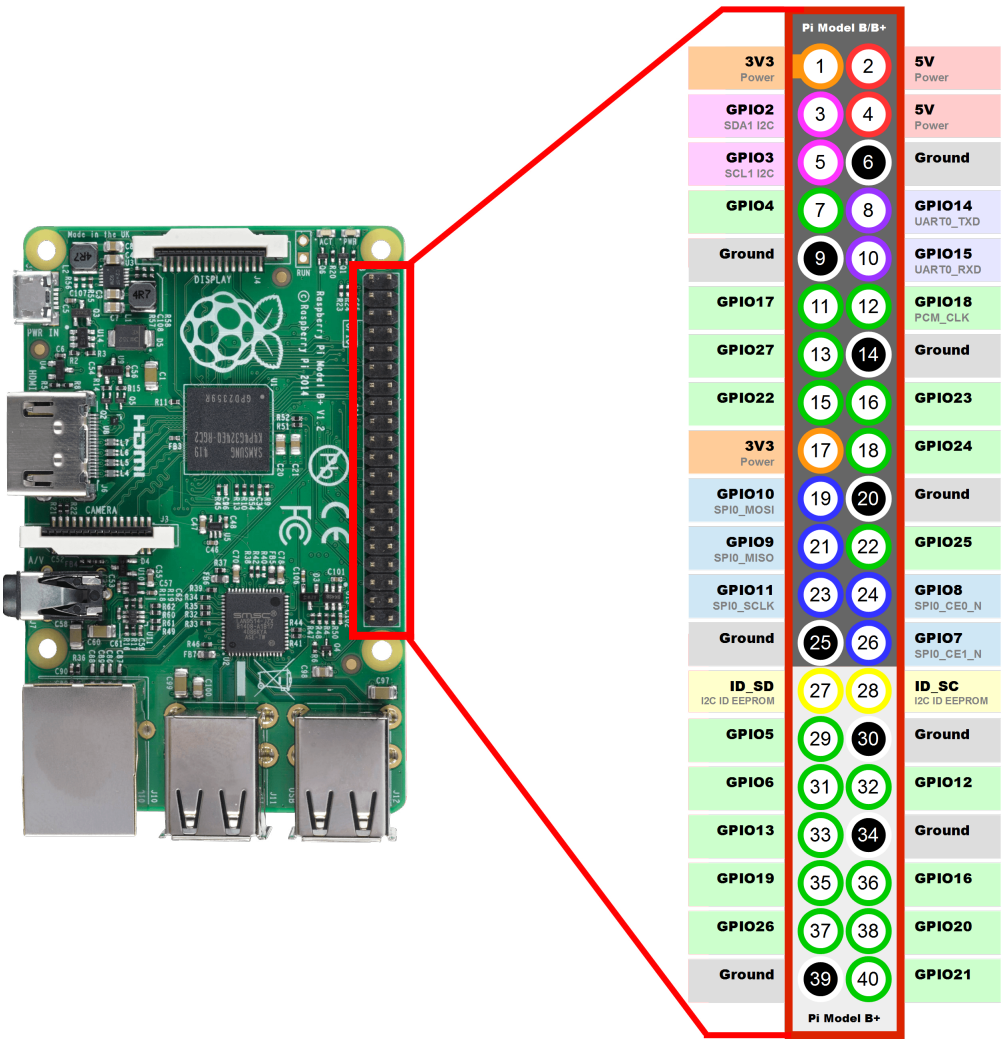

Рисунок 1 – Назначение пинов Raspberry Pi 2B

считается уровнем логического нуля, +1.8В-3.3В - логическая единица.

- **Выход**. Напряжение подаётся самим Raspberry Pi. Уровень логических 0 и 1 аналогичный.

По сути, передача и приём информации осуществляется только считыванием и установлением определённого напряжения. Выходы, отмеченные на схеме зелёным цветом имеют наиболее простой принцип действия: режим ввода/вывода в них устанавливается на всё время подключения устройства, а значение напряжения является относительно постоянным, так как по ним передаётся минимальное количество информации.

В отличие от них, контакты имеющие подпись I2C, SPI или UART используются для последовательный синхронной передачи данных в режиме полного

или полудуплекса. Это означает, что они используются для передачи уже более большого количества данных и могут переключать режим ввода/вывода по несколько тысяч раз за секунду.

Целью данной работы является разработка базового взаимодействия с внешними устройствами, поэтому драйвер будет ориентирован на более простые интерфейсы GPIO.

1.3 Способ управления

Следующей задачей является определение способа чтения и изменения состояния интерфейсов GPIO.

Для работы с пинами используется способ отображения в память (memory maping). Для чтения или изменения состояния устройства требуется взаимодействовать с определённым участком оперативной память, имеющей постоянный физический адрес. Для выполнения подобных действий требуется иметь нулевой уровень привелегий, поэтому программа должна быть реализована в виде загружаемого модуля ядра.

Рассмотрим устройство отображения GPIO в память для Raspberry Pi 2B. В листинге **??** приведены используемые константы.

Листинг 1: struct module

```
1 #define BCM2708_PERI_BASE   0x3F000000
2 #define GPIO_BASE           (BCM2708_PERI_BASE + 0x200000)
3
4 /* GPIO register offsets */
5 #define GPFSEL0     0x0     /* Function Select */
6 #define GPSET0      0x1c    /* Pin Output Set */
7 #define GPCLR0      0x28    /* Pin Output Clear */
8 #define GPLEV0      0x34    /* Pin Level */
9 ...
```

Адрес **BCM2708**$_{P}ERI_{B}ASEmemorymapping, GPIO$**0x200000**.

2 Конструкторская часть

3 Технологическая часть

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. **GPIO [Электронный ресурс] Режим доступа: http://ru.wikipedia.org/wiki/GPIO (дата обращения 10.12.2021).**

2. **Знакомство с GPIO в Raspberry Pi Режим доступа: https://ph0en1x.net/86-raspberry-pi-znakomstvo-s-gpio-perekluchatel-i-svetodiod.htmlgpio-header-layout-pins (дата обращения 10.12.2021).**

ПРИЛОЖЕНИЕ А