Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

**Факультет Пиикт**

Дисциплина

‘Системы ввода-вывода’

**Домашнее задание № 1**

Вариант №-2

*Выполнил:*

Студент группы P3333

Шикунов Максим Евгеньевич 369065 поток 1.5  
Марков Максим Александрович 368491 поток 1. 5  
Гуменник Петр Олегович 368064 поток 1.4  
Анисимов Максим Дмитриевич 379791 поток 1.5

*Преподаватель:*

Быковский Сергей Вячеславович



Санкт-Петербург, 2025

**Вариант**

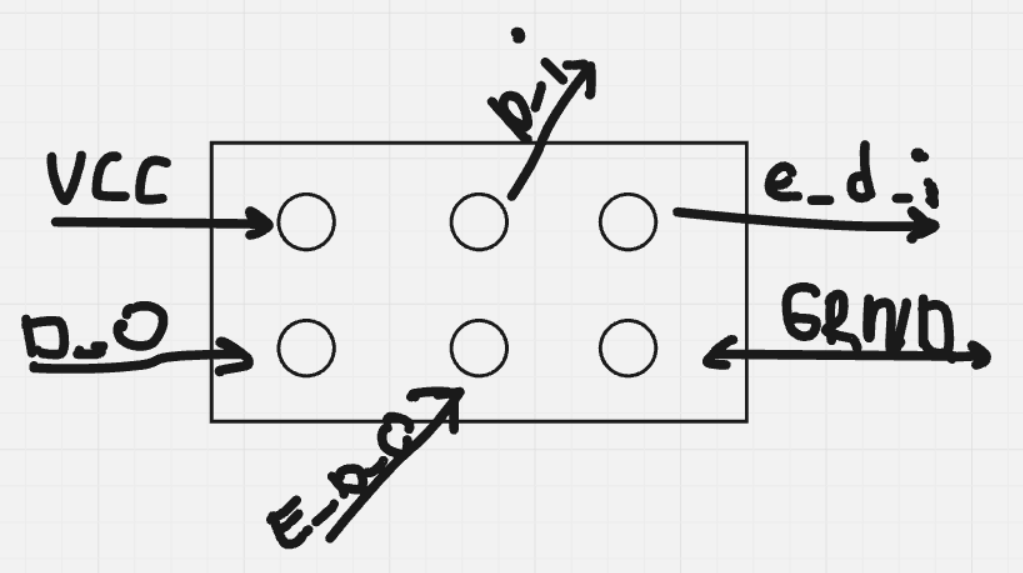
Изображение выглядит как текст, Шрифт, белый

Контент, сгенерированный ИИ, может содержать ошибки.

**Этап 1. Проектирование портов ввода вывода**



Для наших нужд мы решили выбрать провод Molex 6 pin, так как нам нужно два канала по которым будет передаваться данные (один принимать, другой передавать), два канала которые указывают на начало/конец передачи данных, а также напряжение и земля



* VCC – канал напряжения (однонаправленный в slave)
* D\_i – канал, по которому отправляются данные (однонаправленный в slave)
* E\_d\_i – канал, по которому передается сигнал о начале/остановке передаче данных (однонаправленный в slave)
* D\_o – канал, по которому принимаются данные (однонаправленный в master)
* E\_d\_o – канал, по которому передается сигнал о начале/остановке приеме данные (однонаправленный в master)
* Grnd – земля (двунаправленный)

Разделили напряжение и землю для того, чтобы на всякий не перепутать их с другими каналами. В данном проводе они будут иметь другой цвет провода. Провода будут разбиты по три также для удобства (в каждой троице будет пара канала передачи + канал сигнала о передачи).

Топология по подключения устройств будет “точка-точка”. Сделано это из логики, что оно имеет ряд плюсов:

* Минимум компонентов
* Нет промежуточных узлов, которые могут выйти из строя
* Высокая скорость передачи
* Более короткие соединения (из-за чего лучше помехоустойчивость, лучше безопасность и др.)

**Этап 2. Проектирование протокола передачи данных**



Так выглядит наш пакет данных, в нем находятся:

* SB – начальный байт
* Slave адрес – 8 байт, которые передают адрес по которому надо совершить какое-то действие (Максимальная адресация – 8 байт)
* R – байт, который говорит читать или записывать информацию
* Count байт – байт который передает, сколько количество последующих полезных байт с данными
* F – байт начала данных в пакете
* Данные – полезные байты с информацией, которую надо записать по адресу (Максимально данных – 255 байт)
* Контрольная сумма – 2 байта, которые отвечают за целостность пакета
* N – байт конца данных в пакете
* P – байт конца пакета

Изображение выглядит как текст, диаграмма, линия, число

Контент, сгенерированный ИИ, может содержать ошибки.

Вот так выглядит сигналы передачи данных по каналам:

1. По каналам, которые отвечают за саму передачу данных (пакетов), будут передаваться последовательно биты (соответственно на фронте 1, на спаде 0)
2. По каналам, которые отвечают за сигнализацию о том, что передаются данные, в нем передается постоянно 0 если нет передачи данных и 1 если передаются данные

Подсчет эффективной пропускной способности:

При скорости в 1 Мбит/c:

**Этап 3. Описание сценариев использования и протокола транспортного уровня**

Сценарии использования и прикладные области:

* Различные датчики на производстве к ПЛК (температура, давление и т. д.)
* Управление реле и шаговыми двигателями
* Промышленная автоматизация в целом

Скорость передачи не такая уж и большая, из-за чего можно использовать в передаче низкочастотных команд

* Различные медицинские устройства
* Телекоммуникационные системы по передаче информаций между устройствами

Так как скорость не особо большая, то данный провод не подходит для высокоскоростных систем

Протоколы безопасности:

1. У нас будет реализован протокол по подтверждению о том, что приемщик готов принимать пакеты данных. Отправитель делает первый запрос: передает slave адрес, в R-байте передает код, означающий что сейчас начнется передача данных. Count байт равен 0. Дальше ждем ответ, где в slave адресе будет записан тот же адрес и в R-байте будет записан код того, что устройство готово к принятию (ну или другой код если не готово). Count байт равен 0. Дальше отправляем данные. После чего ожидаем опять ответ от приемщика с slave адрес и R-байтом, в котором будем записан код успеха или код ошибки, которая произошла.
2. Также реализована контрольная сумма наших данных, которая считается hash функцией по нашем данным во время отсылки данных, а потом проверяется данная контрольная сумма при приеме данных. Если она не совпадает, то возвращается код ошибки, которые означает что контрольная сумма не совпала и надо отправить данные снова.