## Лабораторная работа 5: Работа с интерфейсом RS-485

**Цель:**

* Изучить интерфейс RS-485 и его принцип работы.
* Подключить устройство с интерфейсом RS-485 к ESP32.
* Написать программу для обмена данными с устройством по RS-485.
* Интегрировать RS-485 с I2C и GPIO для решения комплексной задачи.

## Теоретическая часть

### Интерфейс RS-485

RS-485 (Recommended Standard 485) - это полудуплексный последовательный интерфейс, предназначенный для многоточечной связи. Он позволяет подключать к одной шине несколько устройств, используя всего два провода:

* **A (Data +):** Линия передачи данных.
* **B (Data -):** Линия приема данных.

**Принцип работы:**

1. **Устройство-ведущий:** Инициирует сеанс связи, выбирая адрес устройства-ведомого.
2. **Устройство-ведомое:** Отвечает на запрос ведущего, подтверждая свое присутствие.
3. **Передача данных:** Ведущий отправляет данные ведомому или наоборот, в зависимости от типа операции.
4. **Завершение сеанса:** Ведущий освобождает шину, позволяя другим устройствам инициировать новые сеансы.

**Преимущества RS-485:**

* **Многоточечная связь:** Поддержка до 32 устройств на одной шине.
* **Большая дальность:** До 1200 м при скорости 9600 бит/с.
* **Помехоустойчивость:** Стойкость к электромагнитным помехам.
* **Низкая стоимость:** Простота реализации.

### Подключение RS-485 к ESP32

Для работы с RS-485 на ESP32 необходимо использовать библиотеку RS485.

**Пример подключения устройства с RS-485 к ESP32:**

1. Подключите A (Data +) устройства к A ESP32.
2. Подключите B (Data -) устройства к B ESP32.
3. Подключите питание устройства к 3.3 В ESP32.
4. Подключите землю устройства к земле ESP32.

**Пример кода для обмена данными с устройством по RS-485:**

#include <RS485.h>  
  
#define RS485\_ADDRESS 0x01 // Адрес устройства  
  
RS485 rs485(RS485\_RE, RS485\_DE); // Создание объекта RS485  
  
void setup() {  
 Serial.begin(115200); // Инициализация UART  
 rs485.begin(57600); // Инициализация RS-485 с baud rate 57600  
}  
  
void loop() {  
 // Отправка данных устройству  
 uint8\_t data[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
 rs485.write(RS485\_ADDRESS, data, sizeof(data));  
 Serial.println("Данные отправлены!");  
  
 // Задержка перед приемом данных  
 delay(100);  
  
 // Прием данных от устройства  
 uint8\_t rxData[10];  
 int rxLen = rs485.read(rxData, sizeof(rxData));  
 if (rxLen > 0) {  
 Serial.print("Полученные данные: ");  
 for (int i = 0; i < rxLen; i++) {  
 Serial.print(rxData[i]);  
 if (i < rxLen - 1) {  
 Serial.print(", ");  
 }  
 }  
 Serial.println();  
 }  
  
 delay(1000);  
}

**Интеграция RS-485 с I2C и GPIO:**

RS-485 можно использовать для считывания данных с датчиков, подключенных к I2C, и передачи этих данных на другие устройства по сети RS-485. GPIO можно использовать для управления устройствами на основе полученных данных.

## Задача:

**Задача:** Считывать данные о температуре и влажности с датчика DHT22, подключенного по I2C, и передавать их другому устройству по сети RS-485. По достижении определенных значений температуры или влажности включать светодиод (подключенный к GPIO) с помощью сигнала RS-485.

**Отчёт:**  \* Опишите **схему подключения** устройств. \* **Предоставьте код** для выполнения задачи.

## Пример кода для лабораторной работы :

**Схема подключения:**

* **DHT22:**
  + VCC - 3.3V ESP32
  + GND - GND ESP32
  + DATA - GPIO пин ESP32 (например, GPIO 2)
* **Светодиод:**
  + Anode - GPIO пин ESP32 (например, GPIO 13)
  + Катод - GND ESP32
* **Устройство RS-485:**
  + A (Data +) - A ESP32
  + B (Data -) - B ESP32
  + VCC - 3.3V ESP32
  + GND - GND ESP32

**Код:**

#include <Wire.h>  
#include <DHT.h>  
#include <RS485.h>  
  
#define DHTPIN 2 // GPIO пин для DHT22  
#define DHTTYPE DHT22  
  
DHT dht(DHTPIN, DHTTYPE); // Объект для датчика DHT22  
  
#define RS485\_ADDRESS 0x01 // Адрес устройства RS-485  
#define TEMPERATURE\_THRESHOLD 25  
#define HUMIDITY\_THRESHOLD 60  
  
RS485 rs485(RS485\_RE, RS485\_DE); // Объект RS485  
  
void setup() {  
 Serial.begin(115200);  
 dht.begin();  
 rs485.begin(9600); // Инициализация RS-485 с baud rate 9600  
 pinMode(13, OUTPUT); // Настройка GPIO пина для светодиода  
}  
  
void loop() {  
 // Считывание данных с датчика DHT22  
 float h = dht.readHumidity();  
 float t = dht.readTemperature();  
  
 // Проверка на ошибки чтения  
 if (isnan(h) || isnan(t)) {  
 Serial.println("Ошибка чтения с датчика DHT22!");  
 return;  
 }  
  
 // Формирование сообщения для отправки по RS-485  
 uint8\_t data[5] = {  
 (uint8\_t)t,  
 (uint8\_t)(t \* 10),  
 (uint8\_t)h,  
 (uint8\_t)(h \* 10),  
 0x00 // Флаг тревоги (изначально не установлен)  
 };  
  
 // Отправка данных по RS-485  
 rs485.write(RS485\_ADDRESS, data, sizeof(data));  
  
 // Управление светодиодом  
 if (t > TEMPERATURE\_THRESHOLD || h > HUMIDITY\_THRESHOLD) {  
 data[4] = 0x01; // Установить флаг тревоги  
 digitalWrite(13, HIGH); // Включить светодиод  
 } else {  
 digitalWrite(13, LOW); // Выключить светодиод  
 }  
  
 // Вывод данных на Serial  
 Serial.print("Температура: ");  
 Serial.print(t);  
 Serial.println(" °C");  
 Serial.print("Влажность: ");  
 Serial.print(h);  
 Serial.println(" %");  
  
 delay(2000);  
}

**Схема подключения:**

## Подключение датчика DHT22:

+-------------------------------------------------+  
| |  
| ESP32 |  
| |  
| 3.3V <--> DHT22 (VCC) |  
| GPIO 2 <--> DHT22 (DATA) |  
| GND <--> DHT22 (GND) |  
| |  
+-------------------------------------------------+

## Подключение светодиода:

+-------------------------------------------------+  
| |  
| ESP32 |  
| |  
| GPIO 13 <--> Светодиод (анод) |  
| GND <--> Светодиод (катод) |  
| |  
+-------------------------------------------------+

## Подключение устройства RS-485:

+-------------------------------------------------+  
| |  
| ESP32 |  
| |  
| 3.3V <--> Устройство RS-485 (VCC) |  
| A <--> Устройство RS-485 (A) |  
| B <--> Устройство RS-485 (B) |  
| GND <--> Устройство RS-485 (GND) |  
| |  
+-------------------------------------------------+