## Лабораторная работа 4: Работа с интерфейсом I2C и GPIO

### Цель

* Изучить интерфейс I2C и его принципы работы.
* Подключить устройство с интерфейсом I2C к ESP32.
* Написать программу для работы с I2C, используя конкретный пример (датчик температуры).
* Изучить работу с GPIO на ESP32.
* Подключить различные устройства к GPIO пинам.
* Написать программу для управления GPIO пинами.

### Оборудование

* Плата ESP32
* Датчик температуры (например, I2C BMP180)
* Светодиоды
* Кнопки
* Соединительные провода
* Плата макетная

### Теоретическая часть

#### Интерфейс I2C

I2C (Inter-Integrated Circuit) - это двухпроводной последовательный интерфейс связи, используемый для подключения различных устройств к микроконтроллеру. Он позволяет подключать к одной шине несколько устройств, используя всего два провода:

* **SDA (Data):** Линия данных, по которой передаются данные между устройствами.
* **SCL (Clock):** Тактовая линия, которая синхронизирует передачу 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 I2C:**

* Простота подключения
* Низкая стоимость
* Поддержка множества устройств на одной шине
* Экономия ресурсов (использует всего два проводника)

#### Работа с I2C на ESP32

ESP32 имеет два аппаратных интерфейса I2C, которые можно использовать для подключения различных устройств. Для работы с I2C на ESP32 необходимо использовать библиотеку Wire.

**Пример подключения датчика BMP180:**

1. Подключите SDA датчика к SDA ESP32.
2. Подключите SCL датчика к SCL ESP32.
3. Подключите питание датчика (3,3 В) к питанию ESP32.
4. Подключите землю датчика к земле ESP32.

**Пример кода для работы с датчиком BMP180:**

#include <Wire.h>  
#include "Adafruit\_BMP180.h"  
  
#define BMP180\_ADDRESS 0x77  
  
Adafruit\_BMP180 bmp180;  
  
void setup() {  
 Serial.begin(115200);  
  
 // Инициализация I2C  
 Wire.begin();  
  
 // Инициализация датчика BMP180  
 if (!bmp180.begin()) {  
 Serial.println("Ошибка при инициализации датчика BMP180!");  
 while (1);  
 }  
  
 // Считывание данных с датчика  
 bmp180.setTemperatureCompensation(true);  
 bmp180.readTemperature(bmp180.temperature);  
}  
  
void loop() {  
 // Вывод температуры  
 Serial.print("Температура: ");  
 Serial.print(bmp180.temperature);  
 Serial.println(" °C");  
  
 delay(1000);  
}

#### GPIO (General Purpose Input/Output)

GPIO - это программируемые пины ESP32, которые можно использовать для различных целей, таких как:

* **Ввод:** Считывание состояния кнопок, датчиков и других устройств.
* **Вывод:** Управление светодиодами, реле, сервоприводами и другими устройствами.

**Работа с GPIO на ESP32:**

Для работы с GPIO на ESP32 необходимо использовать функции pinMode(), digitalWrite(), digitalRead() из библиотеки Arduino.h.

**Пример подключения кнопки:**

1. Подключите один вывод кнопки к GPIO пину ESP32 (например, GPIO 21).
2. Подключите другой вывод кнопки к земле ESP32.
3. (Дополнительно) Подключите резистор (10 кОм) между кнопкой и питанием для защиты от перегрузки.

**Пример кода для считывания состояния кнопки:**

const int buttonPin = 21; // GPIO пин, к которому подключена кнопка  
  
void setup() {  
 pinMode(buttonPin, INPUT\_PULLUP); // Настройка пина как входа с подтягивающим резистором  
 Serial.begin(115200);  
}  
  
void loop() {  
 int buttonState = digitalRead(buttonPin); // Считывание состояния кнопки  
  
 if (buttonState == HIGH) {  
 Serial.println("Кнопка нажата!");  
 } else {  
 Serial.println("Кнопка не нажата.");  
 }  
  
 delay(100);  
}

### Задачи

* **Использование I2C для управления несколькими устройствами:** Подключите к ESP32 несколько устройств с интерфейсом I2C (например, датчики, дисплеи) и напишите программу для их управления.
* **Использование GPIO для управления RGB-светодиодом:** Подключите RGB-светодиод к ESP32 и напишите программу для управления его цветом.
* **Реализация системы сигнализации:** Подключите к ESP32 датчик движения и зуммер. Напишите программу, которая будет включать зуммер при обнаружении движения.
* **Создание системы управления домом:** Подключите к ESP32 датчики температуры, влажности, освещенности и другие устройства. Напишите программу для мониторинга данных с датчиков и управления устройствами (например, включение/выключение кондиционера, освещения).

**Отчет:**

* Опишите **схему подключения** устройств к ESP32.
* **Представьте код** программ для работы с I2C и GPIO.
* **Опишите результаты** выполнения расширенных задач.
* **Сделайте выводы** по проделанной работе.

## Примеры кода для лабораторной работы:

### 1. Использование I2C для управления несколькими устройствами

**Пример:** Подключение двух датчиков температуры I2C BMP180 к ESP32.

**Схема подключения:**

* **Датчик 1:**
  + SDA - SDA ESP32
  + SCL - SCL ESP32
  + VCC - 3.3V ESP32
  + GND - GND ESP32
* **Датчик 2:**
  + SDA - SDA ESP32
  + SCL - SCL ESP32
  + VCC - 3.3V ESP32
  + GND - GND ESP32

**Код:**

#include <Wire.h>  
#include "Adafruit\_BMP180.h"  
  
#define BMP180\_ADDRESS1 0x77 // Адрес датчика 1  
#define BMP180\_ADDRESS2 0x76 // Адрес датчика 2  
  
Adafruit\_BMP180 bmp180\_1; // Объект для датчика 1  
Adafruit\_BMP180 bmp180\_2; // Объект для датчика 2  
  
void setup() {  
 Serial.begin(115200);  
  
 // Инициализация I2C  
 Wire.begin();  
  
 // Инициализация датчиков BMP180  
 if (!bmp180\_1.begin(BMP180\_ADDRESS1)) {  
 Serial.println("Ошибка при инициализации датчика BMP180 1!");  
 while (1);  
 }  
  
 if (!bmp180\_2.begin(BMP180\_ADDRESS2)) {  
 Serial.println("Ошибка при инициализации датчика BMP180 2!");  
 while (1);  
 }  
  
 // Считывание данных с датчиков  
 bmp180\_1.setTemperatureCompensation(true);  
 bmp180\_2.setTemperatureCompensation(true);  
}  
  
void loop() {  
 // Считывание температуры с датчика 1  
 bmp180\_1.readTemperature(bmp180\_1.temperature);  
 Serial.print("Температура датчика 1: ");  
 Serial.print(bmp180\_1.temperature);  
 Serial.println(" °C");  
  
 // Считывание температуры с датчика 2  
 bmp180\_2.readTemperature(bmp180\_2.temperature);  
 Serial.print("Температура датчика 2: ");  
 Serial.print(bmp180\_2.temperature);  
 Serial.println(" °C");  
  
 delay(1000);  
}

**Описание:**

* В коде определены два объекта Adafruit\_BMP180, один для каждого датчика.
* При инициализации датчиков используются разные адреса (0x77 и 0x76).
* В цикле loop() считываются данные температуры с каждого датчика и выводятся в последовательный порт.

### 2. Использование GPIO для управления RGB-светодиодом

**Схема подключения:**

* **RGB-светодиод:**
  + Anode (красный) - GPIO пин ESP32 (например, GPIO 13)
  + Anode (зеленый) - GPIO пин ESP32 (например, GPIO 12)
  + Anode (синий) - GPIO пин ESP32 (например, GPIO 11)
  + Катод - GND ESP32

**Код:**

const int redPin = 13; // GPIO пин для красного канала  
const int greenPin = 12; // GPIO пин для зеленого канала  
const int bluePin = 11; // GPIO пин для синего канала  
  
void setup() {  
 pinMode(redPin, OUTPUT);  
 pinMode(greenPin, OUTPUT);  
 pinMode(bluePin, OUTPUT);  
}  
  
void loop() {  
 // Установка красного цвета  
 digitalWrite(redPin, HIGH);  
 digitalWrite(greenPin, LOW);  
 digitalWrite(bluePin, LOW);  
 delay(1000);  
  
 // Установка зеленого цвета  
 digitalWrite(redPin, LOW);  
 digitalWrite(greenPin, HIGH);  
 digitalWrite(bluePin, LOW);  
 delay(1000);  
  
 // Установка синего цвета  
 digitalWrite(redPin, LOW);  
 digitalWrite(greenPin, LOW);  
 digitalWrite(bluePin, HIGH);  
 delay(1000);  
  
 // Установка желтого цвета  
 digitalWrite(redPin, HIGH);  
 digitalWrite(greenPin, HIGH);  
 digitalWrite(bluePin, LOW);  
 delay(1000);  
  
 // Установка фиолетового цвета  
 digitalWrite(redPin, HIGH);  
 digitalWrite(greenPin, LOW);  
 digitalWrite(bluePin, HIGH);  
 delay(1000);  
  
 // Установка голубого цвета  
 digitalWrite(redPin, LOW);  
 digitalWrite(greenPin, HIGH);  
 digitalWrite(bluePin, HIGH);  
 delay(1000);  
  
 // Установка белого цвета  
 digitalWrite(redPin, HIGH);  
 digitalWrite(greenPin, HIGH);  
 digitalWrite(bluePin, HIGH);  
 delay(1000);  
  
 // Выключение всех цветов  
 digitalWrite(redPin, LOW);  
 digitalWrite(greenPin, LOW);  
 digitalWrite(bluePin, LOW);  
 delay(1000);  
}

### 3. Реализация системы сигнализации

**Подключение:**

* Датчик движения (например, HC-SR501)
  + VCC - 5V ESP32
  + GND - GND ESP32
  + OUT - GPIO пин ESP32 (например, GPIO 14)
* Зуммер
  + Положительный вывод - GPIO пин ESP32 (например, GPIO 15)
  + Отрицательный вывод - GND ESP32

**Код:**

const int motionPin = 14; // GPIO пин датчика движения  
const int buzzerPin = 15; // GPIO пин зуммера  
  
void setup() {  
 pinMode(motionPin, INPUT);  
 pinMode(buzzerPin, OUTPUT);  
}  
  
void loop() {  
 int motionState = digitalRead(motionPin);  
  
 if (motionState == HIGH) {  
 digitalWrite(buzzerPin, HIGH); // Включить зуммер  
 delay(1000);  
 digitalWrite(buzzerPin, LOW); // Выключить зуммер  
 delay(1000);  
 } else {  
 digitalWrite(buzzerPin, LOW); // Зуммер выключен  
 }  
}

### 4. Создание системы управления домом

**Подключение:**

* Датчики температуры, влажности, освещенности и другие устройства подключаются в соответствии с их спецификациями.
* Для управления устройствами (кондиционер, освещение) можно использовать реле, подключаемые к GPIO пинам ESP32.

**Код:**

// ... код для чтения данных с датчиков (реализуется в соответствии с используемыми датчиками)  
  
void loop() {  
 // Считывание данных с датчиков  
 // ...  
  
 // Управление устройствами в зависимости от считанных данных  
 // Например:  
 if (temperature > 25) {  
 // Включить кондиционер  
 // ...  
 } else {  
 // Выключить кондиционер  
 // ...  
 }  
  
 if (lightLevel < threshold) {  
 // Включить освещение  
 // ...  
 } else {  
 // Выключить освещение  
 // ...  
 }  
  
 // ...  
}

**Схема подключения:**

**Подключение датчика BMP180:**

### Вариант 1 (ESP32 работает от 3.3 В):

+-------------------------------------------------+  
| |  
| ESP32 |  
| |  
| 3.3V <--> BMP180 (VCC) |  
| A4 <--> BMP180 (SDA) |  
| A5 <--> BMP180 (SCL) |  
| GND <--> BMP180 (GND) |  
| |  
+-------------------------------------------------+

### Вариант 2 (ESP32 работает от 5 В):

+-------------------------------------------------+  
| |  
| ESP32 |  
| |  
| 5V <--> BMP180 (VCC) |  
| A4 <--> BMP180 (SDA) |  
| A5 <--> BMP180 (SCL) |  
| GND <--> BMP180 (GND) |  
| |  
+-------------------------------------------------+

\*\* Подключение RGB-светодиода:\*\*

+-------------------------------------------------+  
| |  
| ESP32 |  
| |  
| GPIO 13 <--> RGB-светодиод (красный анод) |  
| GPIO 12 <--> RGB-светодиод (зеленый анод) |  
| GPIO 11 <--> RGB-светодиод (синий анод) |  
| GND <--> RGB-светодиод (катод) |  
| |  
+-------------------------------------------------+

\*\* Подключение датчика движения и зуммера:\*\*

### Датчик движения:

+-------------------------------------------------+  
| |  
| ESP32 |  
| |  
| 5V <--> HC-SR501 (VCC) |  
| GPIO 14 <--> HC-SR501 (OUT) |  
| GND <--> HC-SR501 (GND) |  
| |  
+-------------------------------------------------+

### Зуммер:

+-------------------------------------------------+  
| |  
| ESP32 |  
| |  
| GPIO 15 <--> Зуммер (положительный вывод) |  
| GND <--> Зуммер (отрицательный вывод) |  
| |  
+-------------------------------------------------+