

Лабораторная работа №6.

Тема: "Знакомство с Arduino и Proteus". SPI, I2C интерфейсы.

Цель работы: освоить базовые навыки работы с Arduino IDE и Proteus.

Теоретическая часть

1. Интерфейс SPI (Serial Peripheral Interface)

SPI — высокоскоростной **синхронный** последовательный интерфейс для связи микроконтроллеров с периферийными устройствами (датчиками, дисплеями, SD-картами и др.).

Основные особенности SPI:

- **Синхронная передача** (есть тактовый сигнал).
- **Полнодуплексный режим** (одновременная передача и прием).
- **4 основных сигнала:**
 - **SCK (Serial Clock)** – тактовый сигнал от ведущего (Master).
 - **MOSI (Master Out Slave In)** – передача данных от Master к Slave.
 - **MISO (Master In Slave Out)** – передача данных от Slave к Master.
 - **SS (Slave Select)** – выбор ведомого устройства (активный низкий уровень).
- **Поддержка нескольких Slave** (каждому нужен отдельный SS).
- **Высокая скорость** (до десятков МГц).

SPI в Arduino

- В Arduino Uno аппаратный SPI использует пины:
 - **SCK (D13), MOSI (D11), MISO (D12), SS (D10).**
- Для работы используется библиотека **SPI.h**.
-

2. Интерфейс I²C (Inter-Integrated Circuit)

I²C — **двухпроводной** последовательный интерфейс для связи нескольких устройств по одной шине.

Основные особенности I²C:

- **Асинхронный** (нет общего тактового сигнала, но есть синхронизация).
- **Двухпроводная шина:**
 - **SDA (Serial Data)** – линия данных.
 - **SCL (Serial Clock)** – тактовый сигнал.
- **Адресная система** (каждое устройство имеет уникальный адрес).
- **Поддержка множества устройств** (до 128 на одной шине).

- **Меньшая скорость**, чем у SPI (обычно 100 кГц или 400 кГц).

I²C в Arduino

- В Arduino Uno аппаратный I²C использует пины:
 - **SDA (A4), SCL (A5).**
- Для работы используется библиотека **Wire.h**.

Сравнение SPI и I²C

Характеристика	SPI	I ² C
Скорость	Высокая (до 10+ МГц)	Низкая (100–400 кГц)
Количество линий	4 (SCK, MOSI, MISO, SS)	2 (SDA, SCL)
Тип передачи	Полнодуплексная	Полудуплексная
Адресация	Через SS (выбор чипа)	Адресная (7/10 бит)
Сложность	Проще для высоких скоростей	Проще для множества устройств

Практическое задание:

Задание 1. Работа с SPI (пример: управление сдвиговым регистром 74HC595).

Цель: научиться передавать данные через SPI.

Компоненты:

- Arduino Uno.
- Сдвиговый регистр **74HC595**.
- 8 светодиодов + резисторы (220 Ом).

Схема подключения:

- **SCK (D13) → SH_CP (11) 74HC595**
- **MOSI (D11) → DS (14) 74HC595**
- **SS (D10) → ST_CP (12) 74HC595**

Код:

```
#include <SPI.h>
```

```

void setup() {
  SPI.begin(); // Инициализация SPI
  pinMode(10, OUTPUT); // SS (активный LOW)
}

void loop() {
  digitalWrite(10, LOW); // Активация регистра
  SPI.transfer(0b10101010); // Отправка байта (зажигаем светодиоды)
  digitalWrite(10, HIGH); // Деактивация
  delay(500);
}

```

Вывод: Светодиоды должны мигать в заданном порядке.

Задание 2. Работа с I²C (пример: чтение данных с датчика BMP280).

Цель: научиться читать данные по I²C.

Компоненты:

- Arduino Uno.
- Датчик **BMP280** (I²C).

Схема подключения:

- **SDA (A4) → SDA BMP280**
- **SCL (A5) → SCL BMP280**

Код:

```

#include <Wire.h>
#include <Adafruit_BMP280.h>

Adafruit_BMP280 bmp;

void setup() {
  Serial.begin(9600);
  if (!bmp.begin(0x76)) { // Адрес датчика
    Serial.println("Датчик не найден!");
    while (1);
  }
}

void loop() {
  Serial.print("Температура: ");
  Serial.println(bmp.readTemperature());
  delay(1000);
}

```

Вывод: В Virtual Terminal должны выводиться значения температуры.

Задание 3. Управление OLED-дисплеем через I2C

Цель: научиться управлять OLED-дисплеем с использованием I2C.

Компоненты Proteus:

- Arduino Uno.
- OLED Display (например, SSD1306, поддерживающий I2C).
- потенциометр, например, 10 кОм.

Описание:

Подключите OLED-дисплей:

- SDA подключите к A4 (SDA) на Arduino.
- SCL подключите к A5 (SCL) на Arduino.
- Vcc подключите к +5V.
- GND подключите к GND.

Подключите потенциометр:

- Левый вывод подключите к +5V.
- Правый вывод подключите к GND.
- Средний вывод (движок) подключите к A0.

Напишите программу, которая:

- Использует библиотеку Adafruit_SSD1306 для работы с OLED-дисплеем.
- Считывает значение с потенциометра.
- Отображает значение на OLED-дисплее.

Задание 4. Работа с EEPROM через I2C

Цель: научиться сохранять и считывать данные из внешней EEPROM памяти.

Компоненты Proteus:

- Arduino Uno.
- 24LC256 EEPROM (EEPROM с интерфейсом I2C).
- кнопка
- резистор 10 кОм — для подтяжки кнопки.

Описание:

Подключите EEPROM:

- SDA подключите к A4 (SDA) на Arduino.
- SCL подключите к A5 (SCL) на Arduino.
- Vcc подключите к +5V.
- GND подключите к GND.

Подключите кнопку:

- Один вывод кнопки подключите к цифровому пину D2.
- Второй вывод кнопки подключите к GND.
- Добавьте подтягивающий резистор (10 кОм) между D2 и +5V.

Напишите программу, которая:

- При нажатии кнопки записывает текущее время (в секундах) в EEPROM.
- Считывает и отображает сохранённое значение при следующем запуске программы.

Примечание: Используйте библиотеку Wire для работы с EEPROM.

Задание 5. Управление светодиодной лентой через SPI

Цель: научиться управлять устройствами через SPI.

Компоненты Proteus:

- Arduino Uno.
- WS2812 RGB LED Strip (или аналогичная светодиодная лента с SPI-интерфейсом).

Описание:

Подключите светодиодную ленту:

- DIN подключите к цифровому пину D11 (MOSI) на Arduino.
- Vcc подключите к +5V.
- GND подключите к GND.

Напишите программу, которая:

- Использует библиотеку FastLED или Adafruit_NeoPixel для управления светодиодной лентой.
- Меняет цвета светодиодов по кругу.

Задание 6. Передача данных между двумя Arduino через SPI

Цель: научиться передавать данные между двумя микроконтроллерами через SPI.

Компоненты Proteus:

- два Arduino Uno.
- потенциометр, например, 10 кОм.
- светодиод.
- резистор 220 Ом.

Описание:

Подключите первый Arduino (Master):

- MOSI подключите к MOSI второго Arduino.
- MISO подключите к MISO второго Arduino.
- SCK подключите к SCK второго Arduino.
- SS подключите к SS второго Arduino.
- GND подключите к GND второго Arduino.

Подключите второй Arduino (Slave):

- Светодиод подключите к D8 через резистор 220 Ом.
- Подключите потенциометр к первому Arduino:
- Левый вывод подключите к +5V.
- Правый вывод подключите к GND.
- Средний вывод (движок) подключите к A0.

Напишите программы:

- Для первого Arduino (Master):
- Считывайте значение с потенциометра.
- Передавайте его на второй Arduino через SPI.

Для второго Arduino (Slave):

- Принимайте значение через SPI.
- Управляет яркостью светодиода с помощью analogWrite().

Задание 7. Часы-будильник с системой мониторинга погоды.

Компоненты:

- Arduino Nano.
- DS1307 (часы реального времени).
- Датчик температуры и влажности (DHT11/DHT22).
- Датчик давления (BMP180).
- BUZZER (сигнал будильника).

Задачи: Вывод времени и погоды на LCD. Управление будильником через кнопки.