

Лабораторная работа №4.

Тема: "Знакомство с Raspberry Pi и Proteus". Работа с аналоговыми устройствами.

Цель работы:

Освоить базовые навыки работы с Raspberry Pi и Proteus.

Raspberry Pi не имеет встроенных аналоговых входов, поэтому для работы с аналоговыми датчиками требуется дополнительное оборудование. Вот краткое описание основных способов подключения аналоговых датчиков:

1. Использование внешнего АЦП (Аналогово-Цифрового Преобразователя).

Примеры АЦП:

MCP3008 (SPI): 10-битный АЦП с 8 каналами.

ADS1115 (I2C): 16-битный АЦП с 4 каналами.

PCF8591 (I2C): 8-битный АЦП с 4 каналами.

Подключение:

- Подключите аналоговый выход датчика к входу АЦП.
- Подключите интерфейс АЦП (SPI или I2C) к GPIO-пинам Raspberry Pi.
- Напишите программу для считывания данных через SPI или I2C.

Преимущества:

- Высокая точность.
- Поддержка нескольких каналов.
- Совместимость с большинством аналоговых датчиков.

2. Использование готовых модулей с АЦП.

Примеры модулей:

- Модули с интегрированным АЦП (например, датчики температуры с цифровым выходом DS18B20).
- Модули с интерфейсом I2C/SPI (например, INA219 для измерения тока и напряжения).

Подключение:

- Подключите модуль к Raspberry Pi через GPIO (I2C, SPI или UART).
- Используйте библиотеки для работы с модулем.

Преимущества:

- Простота использования.
- Встроенные функции обработки данных.
-

3. Использование Arduino как промежуточного устройства.

Принцип работы:

- Arduino считывает аналоговые сигналы с датчиков и передает данные на Raspberry Pi через UART, I2C или USB.

Подключение:

- Подключите аналоговые датчики к Arduino.

- Настройте связь между Arduino и Raspberry Pi (например, через UART или I2C).
- Напишите программы для обоих устройств.

Преимущества:

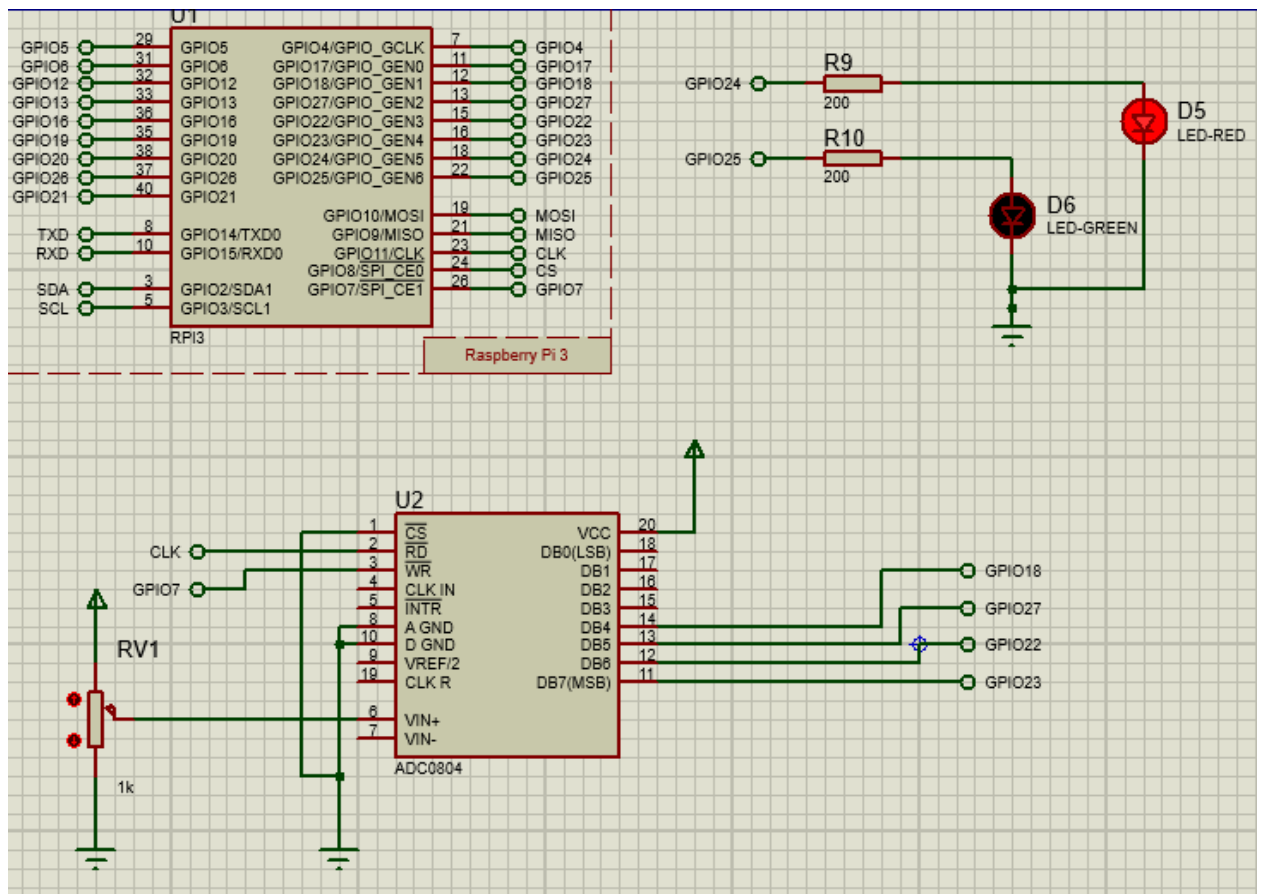
- Гибкость.
- Возможность обработки сложных сигналов.

Недостатки:

- Требуется дополнительное устройство (Arduino).

Наиболее популярный способ — использование внешнего АЦП (например, MCP3008 или ADS1115), так как он обеспечивает высокую точность и простоту подключения. Если вам нужна быстрая и простая реализация, можно использовать RC-цепочку или готовые модули с АЦП.

Пример с использованием внешнего АЦП ADC0804. Если освещенность ниже порогового значения – горит зеленый светодиод, если выше – красный.



ADC0804 — это 8-битный аналого-цифровой преобразователь (АЦП), который широко используется в электронных системах для преобразования аналоговых сигналов в цифровые. Он относится к семейству микросхем, разработанных компанией National Semiconductor, и является одним из наиболее популярных АЦП благодаря своей простоте и надежности.

Основные характеристики ADC0804:

- Разрядность : 8 бит (выходной цифровой код имеет 256 уровней от 0 до 255).
- Тип преобразования : АЦП с последовательным приближением
- Время преобразования : около 100 мкс.
- Интерфейс : Параллельный 8-битный выход данных.
- Напряжение питания : +5 В.

Для запуска преобразования:

- Активируйте сигнал CS (низкий уровень).
- Подайте импульс на вывод WR.
- Дождитесь завершения преобразования (сигнал INTR станет низким).
- Считайте результат с шины данных, активировав сигнал RD.

Подробнее о сигналах:

1. CS (Chip Select) :
 - Для запуска преобразования необходимо установить CS = LOW .
 - Когда CS находится в состоянии LOW, микросхема активна и готова к работе.
2. WR (Write) :
 - После установки CS = LOW, для начала преобразования нужно подать импульс LOW на вывод WR.
 - Этот импульс запускает процесс аналого-цифрового преобразования.
3. INTR (Interrupt Request) :
 - После запуска преобразования вывод INTR переходит в состояние HIGH (высокий уровень), пока идет процесс преобразования.
 - Когда преобразование завершено, вывод INTR автоматически переходит в состояние LOW , сигнализируя о готовности данных.
4. RD (Read) :
 - Чтобы считать результат преобразования, нужно установить RD = LOW. При этом данные становятся доступны на 8-битной шине данных (D0–D7).

Исходный код к схеме:

```
import RPi.GPIO as GPIO
import time

# Настройка GPIO-пинов
DATA_PINS = [18, 27, 22, 23] # GPIO-пины для старших 4 битов данных (D4–D7)
ALARM_PIN = 25
OK_PIN = 24
RD_PIN = 11 # Read
WR_PIN = 7 # Write

# Инициализация GPIO
GPIO.setmode(GPIO.BCM)
for pin in DATA_PINS:
    GPIO.setup(pin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

```

GPIO.setup(RD_PIN, GPIO.OUT)
GPIO.setup(WR_PIN, GPIO.OUT)
GPIO.setup(ALARM_PIN, GPIO.OUT)
GPIO.setup(OK_PIN, GPIO.OUT)

def read_adc():
    # Запуск преобразования
    GPIO.output(WR_PIN, GPIO.LOW) # Запуск преобразования
    time.sleep(0.001)             # Короткая задержка
    GPIO.output(WR_PIN, GPIO.HIGH) # Завершение запуска

    # Ожидание завершения преобразования
    time.sleep(0.001)

    # Чтение данных
    GPIO.output(RD_PIN, GPIO.LOW) # Чтение данных
    data = 0
    for i, pin in enumerate(DATA_PINS):
        data += GPIO.input(pin) << i # Формируем 4-битное значение
    GPIO.output(RD_PIN, GPIO.HIGH) # Завершение чтения
    return data

def update_value(value):
    if value < 5:
        GPIO.output(ALARM_PIN, GPIO.HIGH)
        GPIO.output(OK_PIN, GPIO.LOW)
    else:
        GPIO.output(ALARM_PIN, GPIO.LOW)
        GPIO.output(OK_PIN, GPIO.HIGH)

def main():
    while True:
        adc_value = read_adc() # Считываем значение
        update_value(adc_value) # Обновляем светодиоды
        time.sleep(0.005)      # Задержка для стабильности

```

Задание:

1. Автомат для включения фонарей.

Цель: измерить уровень освещенности с помощью фоторезистора и ADC0804 и при низком уровне включить фонарь.

Компоненты Proteus:

ADC0804

- Raspberry Pi
- Photoresistor (LDR) (фоторезистор)
- Resistor (резистор для делителя напряжения)
- LED (светодиод для индикации низкого уровня света)
- Реле, лампа

Описание:

Создайте делитель напряжения с фоторезистором и резистором. Выведите среднюю точку делителя на вход ADC0804.

Если уровень освещенности ниже заданного порога, включите освещение (лампу через реле).

2. Управление яркостью светодиода через PWM.

Цель: использовать ADC0804 для измерения аналогового сигнала и управлять яркостью светодиода с помощью ШИМ (PWM).

Компоненты Proteus:

- ADC0804
- Raspberry Pi
- Potentiometer (POT-HG)
- LED (светодиод)
- Resistor (резистор для ограничения тока LED)

Описание:

Подключите потенциометр к входу ADC0804.

Считайте аналоговый сигнал с ADC0804 на Raspberry Pi.

Преобразуйте полученное значение в коэффициент заполнения (duty cycle) для ШИМ.

Настройте один из GPIO-пинов RPi на работу в режиме PWM.

Подключите светодиод к этому пину через резистор.

Напишите программу на Python, которая будет изменять яркость светодиода в зависимости от значения, считанного с ADC0804.

3. Цифровой термометр с использованием термистора.

Цель: измерить температуру с помощью термистора и ADC0804, а затем отобразить результат на Raspberry Pi.

Компоненты Proteus:

- ADC0804
- Raspberry Pi
- Thermistor (NTC) (термистор для измерения температуры)
- Resistor (резистор для делителя напряжения)
- LED (светодиод для индикации высокой температуры)

Описание:

Создайте делитель напряжения с термистором и резистором. Выведите среднюю точку делителя на вход ADC0804.

Считайте аналоговый сигнал с ADC0804 на Raspberry Pi.

Если температура превышает заданное значение, включите светодиод.