

Лабораторная работа 1. Исследование АЦП с использованием Arduino

Цель работы

1. Изучить принципы аналого-цифрового преобразования.
2. Освоить работу с АЦП на платформе Arduino.
3. Исследовать влияние частоты дискретизации и разрядности АЦП на качество оцифрованного сигнала.

Ход работы:

Часть 1: Подготовка и настройка

1. Соберите схему:
 - Подключите потенциометр к Arduino:
 - Один крайний вывод потенциометра подключите к **5V** на Arduino.
 - Другой крайний вывод подключите к **GND**.
 - Средний вывод подключите к аналоговому входу **A0**.
 - Подключите Arduino к компьютеру.
2. Используя Arduino IDE загрузите в Arduino следующий код:

```
void setup() {  
    Serial.begin(9600); // Инициализация последовательного порта  
}  
  
void loop() {  
    int sensorValue = analogRead(A0); // Чтение значения с аналогового  
    входа A0  
    float voltage = sensorValue * (5.0 / 1023.0); // Преобразование в  
    напряжение  
    Serial.println(voltage); // Вывод напряжения в монитор порта  
    delay(100); // Задержка для удобства чтения  
}
```

3. Откройте **Монитор порта** в Arduino IDE и убедитесь, что данные поступают. Поворачивайте потенциометр и наблюдайте за изменением значений.

Часть 2: Исследование АЦП

1. Исследование разрядности АЦП:

- Arduino Uno имеет 10-битный АЦП (диапазон значений от 0 до 1023).
- Рассчитайте шаг по напряжению, минимально и максимальное напряжение, которое можно измерить.
- загрузите в Arduino измененный код, который так же будет выводить измеренное значение в двоичном виде
- Подключите между выводами A0 и GND вольтметр и заполните таблицу 1, где U показания вольтметра, а U' показания Arduino.

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int max_voltage = 5; // Максимальное напряжение АЦП Arduino UNO  
  int bit_depth = 10; // Разрядность АЦП  
  
  int sensorValue = analogRead(A0); // Чтение значения АЦП  
  float voltage = sensorValue * (max_voltage / (pow(2.0,bit_depth) - 1)); //  
  Преобразование в напряжение  
  Serial.print("двоичный код: ");  
  Serial.print(sensorValue, BIN);  
  Serial.print(" | напряжение: ");  
  Serial.print(voltage, 2); // Вывод с точностью до 2 знаков  
  Serial.println(" B");  
  delay(1000);  
}
```

Таблица 1.

| U, В | 0,5 | 1 | 1,5 | 2 | 2,5 | 3 | 3,5 | 4 | 4,5 | 5 |
|--------------|-----|---|-----|---|-----|---|-----|---|-----|---|
| U', В | | | | | | | | | | |
| Двоичный код | | | | | | | | | | |

2. Исследование частоты дискретизации:

- Измените код загруженный в Ардуино, для измерения частоты дискретизации

```

unsigned long startTime; // Время начала измерения
unsigned long endTime;   // Время окончания измерения
unsigned long sampleCount = 0; // Счетчик измерений

void setup() {
  Serial.begin(9600); // Инициализация последовательного порта
}

void loop() {
  startTime = micros(); // Записываем начальное время

  // Выполняем измерения в течение 1 секунды
  while (micros() - startTime < 1000000) {
    analogRead(A0); // Чтение аналогового входа
    sampleCount++;  // Увеличиваем счетчик измерений
  }

  endTime = micros(); // Записываем конечное время

  // Вычисляем частоту дискретизации
  float samplingRate = sampleCount / ((endTime - startTime) / 1000000.0);

  // Выводим результат
  Serial.print("Частота дискретизации: ");
  Serial.print(samplingRate);
  Serial.println(" Hz");

  // Сбрасываем счетчик для следующего измерения
  sampleCount = 0;

  delay(2000); // Пауза перед следующим измерением
}

```

- выполните 10 измерений, рассчитайте среднюю частоту дискретизации АЦП Arduino

Контрольные вопросы

1. Основные этапы аналого-цифрового преобразования.
2. Какие процессы происходят на этапах дискретизации, квантования и кодирования?
3. Как разрядность АЦП влияет на точность измерений?
4. От чего зависит частота дискретизации?
5. Какие факторы влияют на точность АЦП?

Отчет по лабораторной работе

Отчет должен включать:

1. Цель работы.
2. Описание схемы и оборудования.
3. Исходный код программы.
4. Результаты измерений.
5. Ответы на вопросы
6. Выводы по работе.

#лр

#технологии_физического_уровня_передачи_данных