Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа №4**

“Изучение работы контроллеров ввода/вывода”

по дисциплине “Системы ввода-вывода”

**Вариант: 4**

**Выполнили:**

Барсуков Максим Андреевич

Аскаров Эмиль Рамилевич

Мальков Павел Александрович

Поток **1.3**

**Преподаватель:**

Табунщик Сергей Михайлович

г. Санкт-Петербург

2025

**Цель работы:** познакомится с принципами работы с контроллерами ввода

вывода на примере контроллера UART.

**Задание к лабораторной работе**

1. Написать программу для микроконтроллера Atmega328, принимающую и отправляющую пакеты по интерфейсу UART в соответствии с обозначенным форматом пакета. Драйвер UART должен быть реализован с использованием операций ввода/вывода в регистры аппаратного контроллера UART.
2. Контроллер должен принимать данные с ПК, проверять их на корректность и отправлять обратно корректные пакеты. Если пакет пришел с ошибкой, то он отбрасывается.
3. Контроллер должен раз в секунду передавать данные с датчика, указанного в варианте задания.
4. Написать клиентскую программу на ПК для приема и отправки пакетов к микроконтроллеру по интерфейсу UART, моделирующей как корректную отправку пакетов, так и случаи с ошибками: неправильная длина, отсутствие синхробайта, недостаточное количество данных.
5. Подключить микроконтроллер к ПК и протестировать работоспособность написанных программ
6. Снять осциллограмму передачи любого пакета по интерфейсу UART
7. Оформить отчет по работе в электронном формате

**Выполнение**

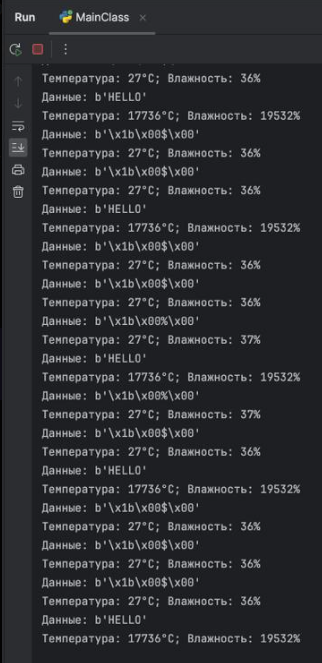
**Описание кода Arduino**

| #include <DHT11.h>  #include "CRC8.h"  #include "CRC.h"  #define SetBit(reg, bita) reg |= (1<<bita)  volatile char b = 0;  typedef unsigned char uint8\_t;  DHT11 dht11(2);  CRC8 crc;  void USART\_Transmit(uint8\_t data)  {  /\* Wait for empty transmit buffer \*/  while (!(UCSR0A & (1<<UDRE0)));  /\* Put data into buffer, sends the data \*/  UDR0 = data;  }  void USART\_Transmit(uint8\_t data[], int len) {  for (int i=0; i<len; i++) {  USART\_Transmit(data[i]);  }  }  // Отправка пакета UART клиенту. Содержит синхробайт, длину данных, сами данные и контрольная сумма  void USART\_SendPacket(uint8\_t data[], int len) {  USART\_Transmit(0x5A);  USART\_Transmit((uint8\_t)len);  USART\_Transmit(data, len);  uint8\_t crc = calcCRC8(data, len);  USART\_Transmit(crc);  }  void setup() {  // Initialize serial communication to allow debugging and data readout.  // Using a baud rate of 9600 bps.  dht11.setDelay(2000); // Set this to the desired delay. Default is 500ms.  uint32\_t baudRate = 115200;  uint32\_t ubrr = 16000000 / 16 / baudRate;  UBRR0H = (unsigned char) (ubrr >> 8);  UBRR0L = (unsigned char) ubrr;  SetBit(UCSR0B, TXEN0);  SetBit(UCSR0B, RXEN0);  SetBit(UCSR0B, RXCIE0);  // Настройка по условиям варианта  SetBit(UCSR0C, 1);  SetBit(UCSR0C, 2);  SetBit(UCSR0C, 5);  SetBit(UCSR0C, 4);  SetBit(UCSR0C, 3);  pinMode(13, OUTPUT);  }  uint8\_t packet[256];  int state = 0, len = 0, pos = 0;  /\*  ISR --- стандартный встроенный метод в C для вызова векторов прерывания в Atmega.  Здесь происходит обработка данных, принятые от клиента:  - 0: Ожидание начала передачи.  - 1: Ожидание получения длины пакета.  - 2: Получение данных пакета.  - 3: Проверка контрольной суммы, здесь сбрасывается  pos для принятия нового пакета.  \*/  ISR(USART\_RX\_vect) {  b = UDR0;  if(b == 'A') digitalWrite(13, HIGH);  if(b == 'B') digitalWrite(13, LOW);  switch(state) {  case 0:  if (b == 0x5A) state = 1;  break;  case 1:  len = b;  state = 2;  break;  case 2:  packet[pos++] = b;  if (pos >= len) state = 3;  break;  case 3:  uint8\_t crc = calcCRC8(packet, len);  if (b == crc) USART\_SendPacket(packet, len);  state = 0;  pos = 0;  }  }  void loop() {  int temperature = 0;  int  = 0;  // Attempt to read the temperature and humidity values from the DHT11 sensor.  int result = dht11.readTemperatureHumidity(temperature, humidity);  // Check the results of the readings.  // If the reading is successful, print the temperature and humidity values.  // If there are errors, print the appropriate error messages.  if (result == 0) {  int res[2];  res[0] = temperature;  res[1] = humidity;  // Отправка пакета клиенту вместо вывода в Serial Monitor.  USART\_SendPacket((uint8\_t\*)res, 4);  }  } |
| --- |

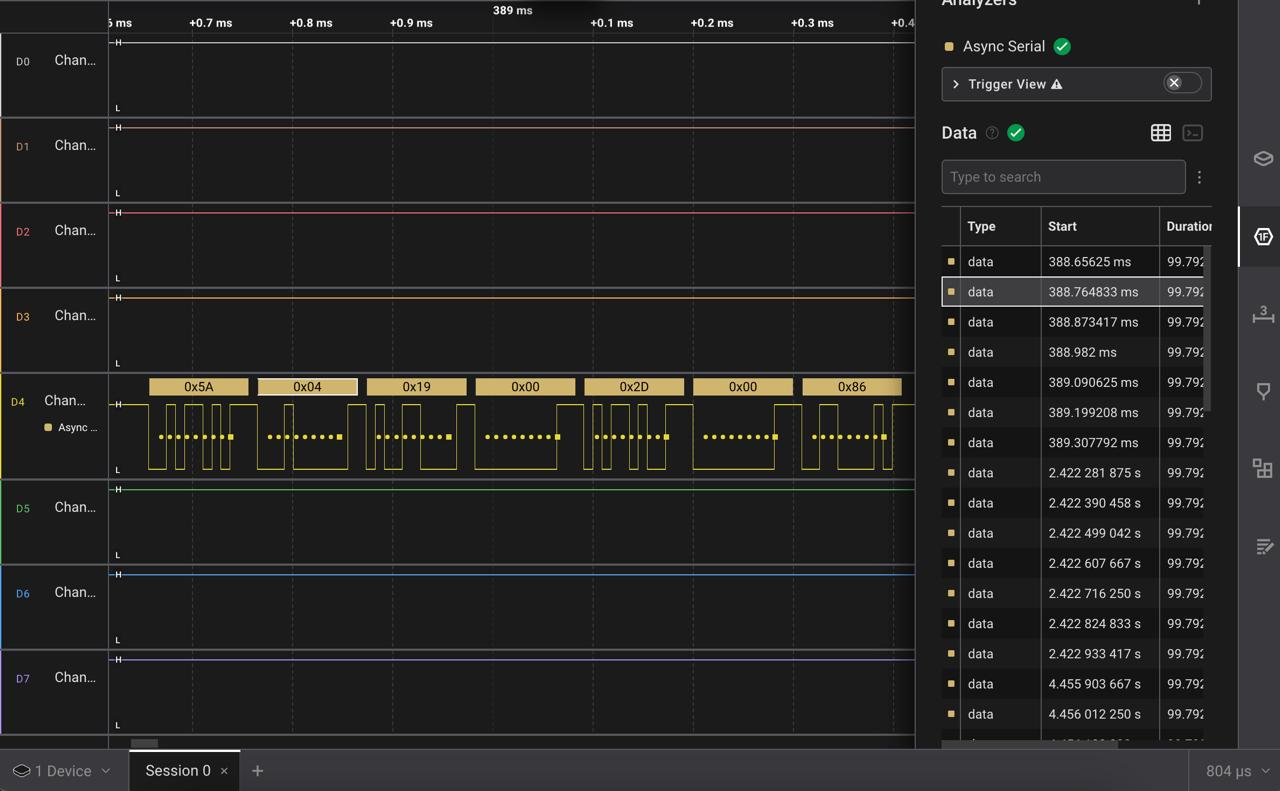
**Описание клиентского кода**

| import serial  import crc8  import time  import threading  # Настройка по варианту  ser = serial.Serial("/dev/cu.usbserial-110", 115200, parity=serial.PARITY\_ODD, stopbits=serial.STOPBITS\_TWO)  hash = crc8.crc8()  # Функция для получения хеш-суммы  def tocrc(data):  return hash.reset().update(data).digest()  # Функция для отправки пакета в контроллер  def sendPacket(data):  ser.write(b'\x5A')  ser.write(len(data).to\_bytes(1, "little"))  ser.write(data)  ser.write(tocrc(data))  # Эта функция периодически отправляет пакет в контроллер.  # Строка после функции как раз ответственен за её активацию  def interval():  while True:  sendPacket(b'HELLO')  time.sleep(5)  threading.Thread(target=interval).start()  # Бесконечный цикл с валидацией и обработкой принятых данных от контроллера  while True:  bs = ser.read()  if (bs != b'\x5A'):  continue  n = ser.read()  n = int.from\_bytes(n, "little")  data = ser.read(n)  crc = ser.read()  if tocrc(data) != crc:  print("Error in crc8")  continue    temp = int.from\_bytes(data[0:2], "little")  hum = int.from\_bytes(data[2:4], "little")  print(f'Данные: {data}\nТемпература: {temp}°C; Влажность: {hum}%') |
| --- |

**Демонстрация**



**Временная диаграмма**



**Вывод**

В ходе выполнения лабораторной работы мы познакомились с принципами работы с контроллерами ввода-вывода через протокол передачи UART для DHT11 на основе Atmega328.