



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
“Национальный исследовательский университет ИТМО”

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ
И КОМПЬЮТЕРНОЙ ТЕХНИКИ**



I/O.

0000—0?
0000 0000 0000
0000 0000 0000
0000 0000 0000
0000 0000 0000

ULTRAVIOLENCE.

ЛАБОРАТОРНАЯ РАБОТА №4

“Изучение работы контроллеров ввода/вывода”

по дисциплине

“Системы ввода-вывода”

Вариант: 4.

Работу выполнили:

Студенты потока 1.1

Болорболд Аригуун, группа Р3311

Черноморов Кирилл Александрович, группа Р3309

Щербинин Эдуард Павлович, группа Р3314

Лектор:

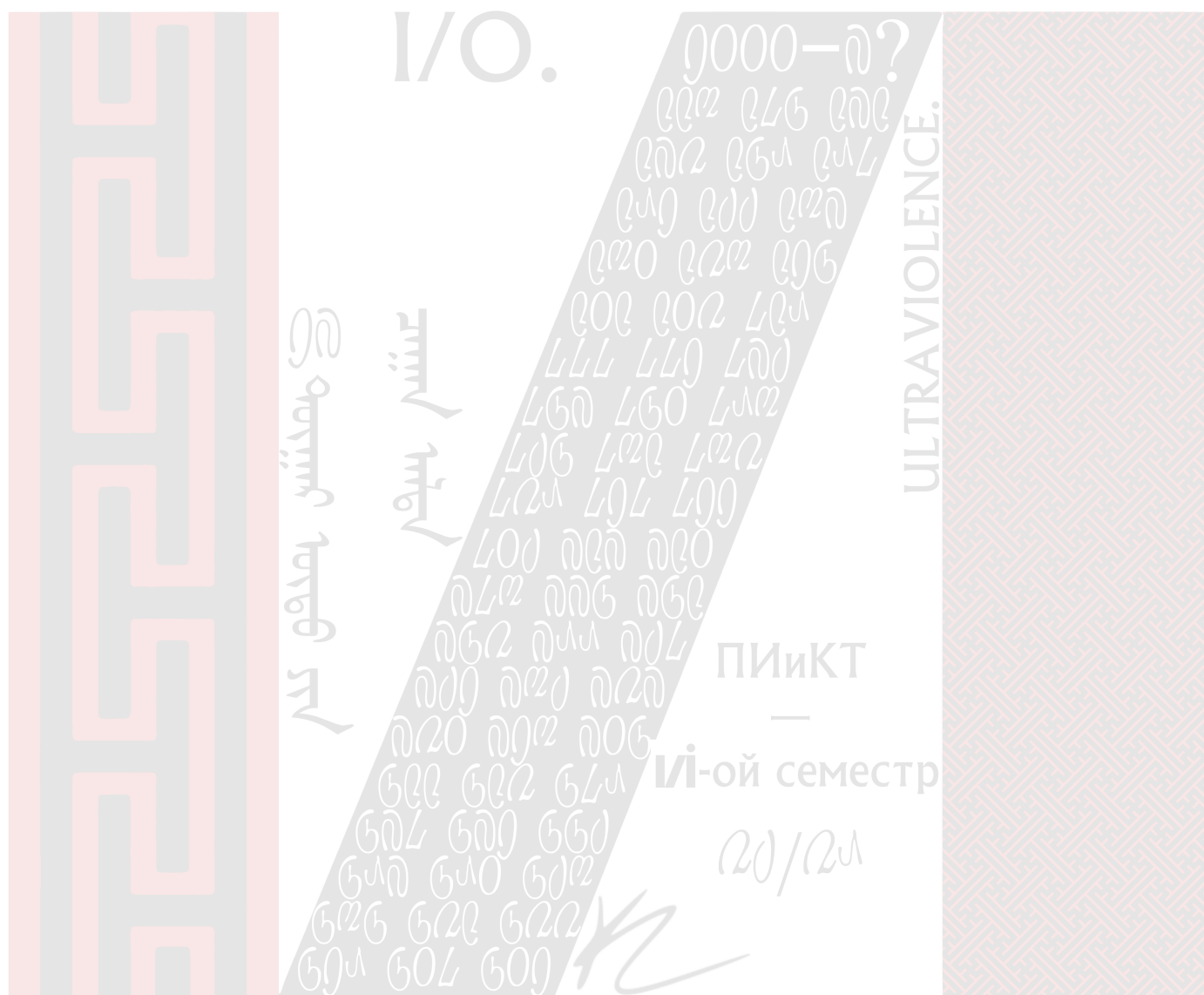
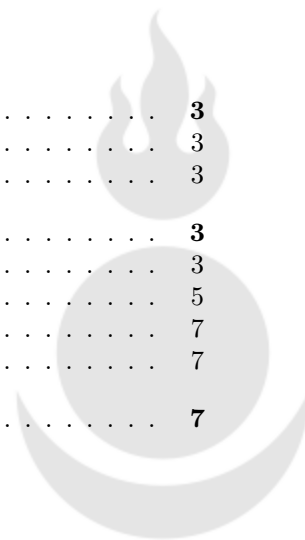
Быковский Сергей Вячеславович

Практик:

Табунщик Сергей Михайлович

Содержимое

1 Текст задания	3
1.1 Цель задания	3
1.2 Задачи	3
2 Выполнение	3
2.1 Описание кода Arduino (комментарии в самом коде)	3
2.2 Описание кода клиента	5
2.3 Демонстрация	7
2.4 Временная диаграмма	7
3 Вывод	7



1 Текст задания

1.1 Цель задания

Познакомиться с принципами работы с контроллерами ввода вывода на примере контроллера UART.

1.2 Задачи

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

№ варианта	Датчик	Скорость UART	Чётность	Кол-во стоповых бит
4	DHT11 температура и влажность	115200	odd parity	2

2 Выполнение

2.1 Описание кода Arduino (комментарии в самом коде)

```
#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 (!(UCSROA & (1<<UDRE0)));
    /* Put data into buffer, sends the data */
    UDRO = 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.
    //Serial.begin(9600);

    // Uncomment the line below to set a custom delay between sensor readings (in milliseconds).
    dht11.setDelay(2000); // Set this to the desired delay. Default is 500ms.

    uint16_t baudRate = 9600;
    uint16_t ubrr = 16000000 / 16 / baudRate - 1;

    UBRR0H = (unsigned char) (ubrr >> 8);
    UBRR0L = (unsigned char) ubrr;

    SetBit(UCSR0B, TXEN0);
    SetBit(UCSR0B, RXEN0);
    SetBit(UCSR0B, RXCIE0);

    // Обращаю внимание сюда: здесь происходит 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 humidity = 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);
  }
}

```

2.2 Описание кода клиента

```

import serial
import crc8
import time
import threading

```

```

# Настройка по варианту
ser = serial.Serial("COM5", 9600, 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")
    pres = int.from_bytes(data[2:4], "little")
    print(f'Data: {data}\nTemp: {temp}°C; Humidity: {pres}%')

```



ПИиКТ

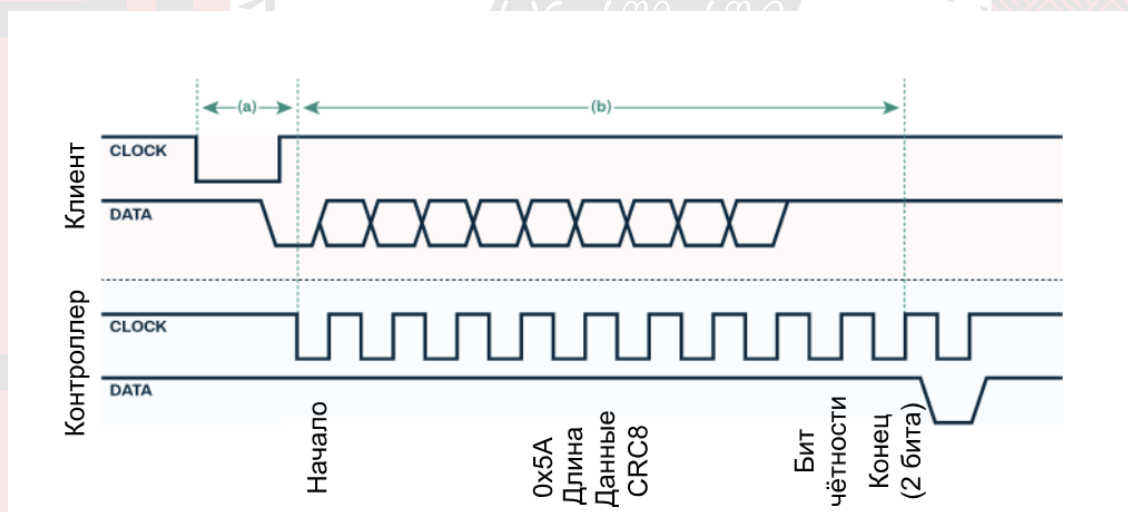
—
II-ой семестр

20/21

2.3 Демонстрация

```
Temp: 26°C; Humidity: 19%  
Data: b'\x1a\x00\x13\x00'  
Temp: 26°C; Humidity: 19%  
Data: b'\x1a\x00\x13\x00'  
Temp: 26°C; Humidity: 19%  
Data: b'HELLO'  
Temp: 17736°C; Humidity: 19532%  
Data: b'\x1a\x00\x13\x00'  
Temp: 26°C; Humidity: 19%  
Data: b'\x1a\x00\x13\x00'  
Temp: 26°C; Humidity: 19%  
Data: b'HELLO'  
Temp: 17736°C; Humidity: 19532%  
Data: b'\x1a\x00\x13\x00'  
Temp: 26°C; Humidity: 19%
```

2.4 Временная диаграмма



3 Вывод

В рамках этой лабораторной работы мы ознакомились с работой микроконтроллера Atmega328 на основе DHT11 через протокол передачи UART.