

Университет ИТМО
Факультет программной инженерии и компьютерной техники

Лабораторная работа №3
«Изучение протоколов передачи данных между устройствами»
по дисциплине «Системы Ввода-Вывода»

Выполнили студенты (поток 1.5):
Ефремов Марк Андреевич Р3334
Строзенко Константин Викторович Р3331
Нодири Хисравхон Р3331

Вариант: 1

Преподаватель:
Быковский Сергей Вячеславович

г. Санкт-Петербург

2025 г.

Задание

Цель:

Познакомится с принципами обмена данными между устройствами, алгоритмами обмена и форматами передачи данных на примере интерфейсов I2C, 1-Wire.

Задачи:

Для датчика BMP280 с интерфейсом I2C и датчика DHT-11 с интерфейсом 1-Wire выполнить следующие шаги:

1. Подключить комплект с контроллер с датчиком и логическим анализатором к компьютеру
2. С помощью логического анализатора записать временную диаграмму обмена данными по сигнальным линиям в течении трех транзакций обмена.
3. Расшифровать протокол обмена данными.
4. Перевести значение физической величины, заданной в варианте задания, в человекочитаемый формат.
5. Нарисовать временную диаграмму передачи другого, отличного от полученных, значения физической величины.
6. Определить скорость интерфейса.

Оформить отчет по работе в электронном формате

Расшифровка протокола обмена I2C:

Температура рассчитывается по следующему алгоритму:

```
// Returns temperature in DegC, resolution is 0.01 DegC. Output value of "5123" equals 51.23 DegC.
// t_fine carries fine temperature as global value
BMP280_S32_t t_fine;
BMP280_S32_t bmp280_compensate_T_int32(BMP280_S32_t adc_T)
{
    BMP280_S32_t var1, var2, T;
    var1 = (((((adc_T >> 3) - ((BMP280_S32_t)dig_T1 << 1))) * ((BMP280_S32_t)dig_T2)) >> 11;
    var2 = (((((adc_T >> 4) - ((BMP280_S32_t)dig_T1)) * ((adc_T >> 4) - ((BMP280_S32_t)dig_T1))) >> 12) *
        ((BMP280_S32_t)dig_T3)) >> 14;
    t_fine = var1 + var2;
    T = (t_fine * 5 + 128) >> 8;
    return T;
}
..
```

Результат необходимо поделить на 100 для перевода в градусы.

Для расчета температуры необходимо найти значения:

- dig_T1 = 0x6DB9 (28089)
- dig_T2 = 0x67D5 (26581)
- dig_T3 = 0xFC18 (64536)
- temp_xslb = 0x8
- temp_lsb = 0x32
- temp_msb = 0x85

Register Address LSB / MSB	Register content	Data type
0x88 / 0x89	dig_T1	unsigned short
0x8A / 0x8B	dig_T2	signed short
0x8C / 0x8D	dig_T3	signed short

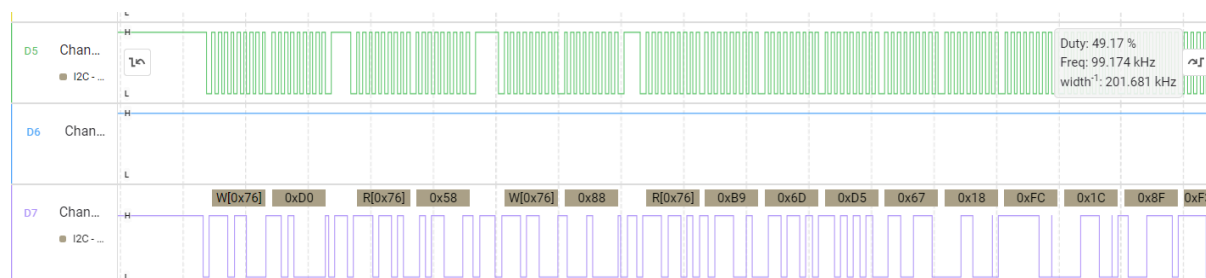
0x[0x89,0x88] - порядок байтов

Table 18: Memory map

Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state
temp_xlsb	0xFC	temp_xlsb<7:4>				0	0	0	0	0x00
temp_lsb	0xFB	temp_lsb<7:0>								0x00
temp_msb	0xFA	temp_msb<7:0>								0x80

Сначала передаются значения dig_T1, dig_T2, dig_T3.

После байта адреса (0x88): идут значения хранящиеся в 0x88, 0x89, 0x8A, 0x8B, 0x8C, 0x8D...:



0x88: 0xB9

0x89: 0x6D

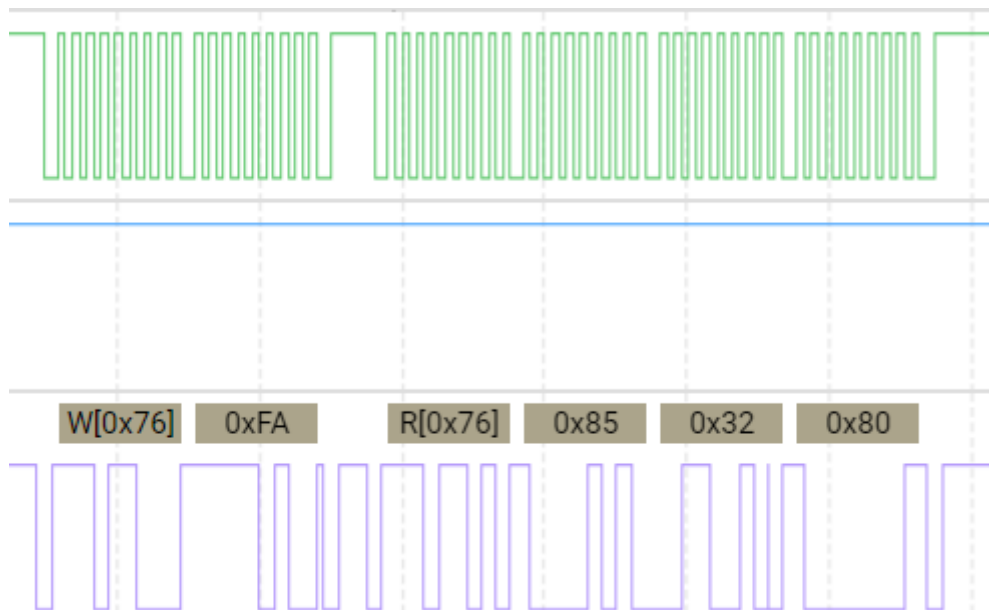
0x8A: 0xD5

0x8B: 0x67

0x8C: 0x18

0x8D: 0xFC

Затем передаются значения temp_xslb, temp_lsb, temp_msb:
Аналогично, они идут после байта адреса (0xFA).



0xFA: 0x85

0xFB: 0x32

0xFC: 0x80

adc_T = {temp_msb, temp_lsb, temp_xslb} = 0x85328 = 545576

Если представить алгоритм расчета в виде формулы, то получится следующее:

var1 = 128793,1787	var1 = (((double)adc_T)/16384.0 - ((double)dig_T1)/1024.0) * ((double)dig_T2);
var2 = -370,8917052	var2 = (((double)adc_T)/131072.0 - ((double)dig_T1)/8192.0) * (((double)adc_T)/131072.0 - ((double)dig_T1)/8192.0) * ((double)dig_T3);
tfine = 128422	t_fine = (BMP280_S32_t)(var1 + var2);
T = 25,08	T = (var1 + var2) / 5120.0;
integer result (**): 2508	Temperature [1/100 °C]

$$T = ((545576 / 16384 - 28089/1024) * 26581 + (545576 / 131072 - 28089 / 8192) * 64536) / 5120 = 39.714 \text{ } ^\circ\text{C}$$

Формирование временной диаграммы передачи других значений через I2C:

$$25^\circ\text{C} = ((x / 16384 - 27.43) * 26581 + (x / 131072 - 3.429) * 64536) / 5120$$

$$128000 = 1.622x + 0.49x - 729116 - 221293$$

$2.112x = 1078509$

$x = 510657 = 0x7CAC1 = \{0x7C, 0xAC, 0x1\}$

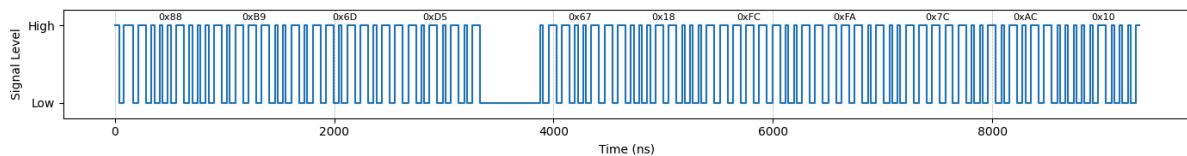
temp_msb = 0x7C

temp_lsb = 0xAC

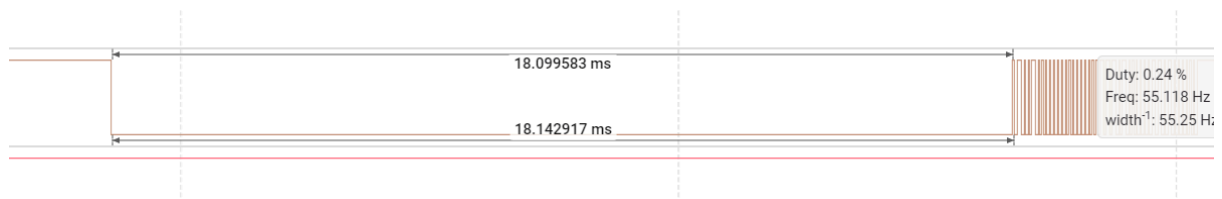
temp_xlsb = 0x1

Значения dig_T1, dig_T2, dig_T3 используются те же, что и в записанной временной диаграмме.

Байты dig_Px, используемые для кодирования давления, опущены за ненадобностью, чтобы не перегружать диаграмму нерелевантной информацией.



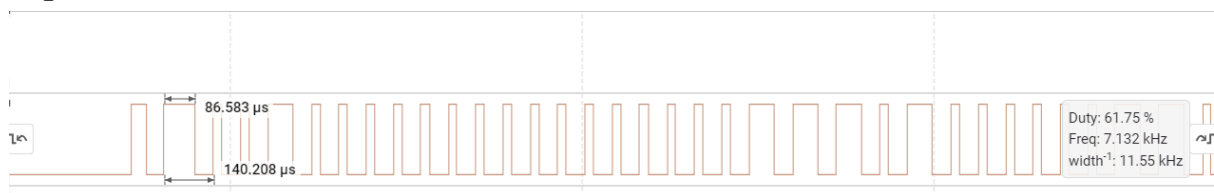
Расшифровка протокола обмена 1-Wire:

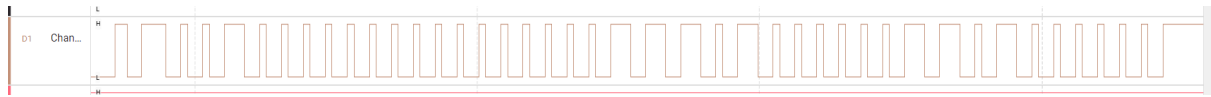


Сначала хост отправляет стартовый сигнал низкого уровня длиной не менее 18 мс.

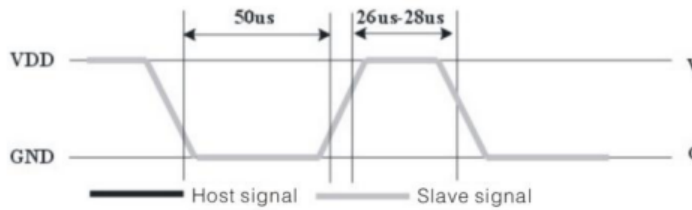
Далее передается короткий сигнал высокого уровня (перед сигналом 80 мкс). (в документации нет про него особо никакой информации)

После сигнала высокого уровня длиной около 80 мкс, начинается передача 40 битов данных:



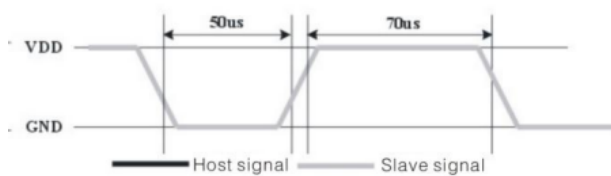


Передача бита 0 выглядит следующий образом:

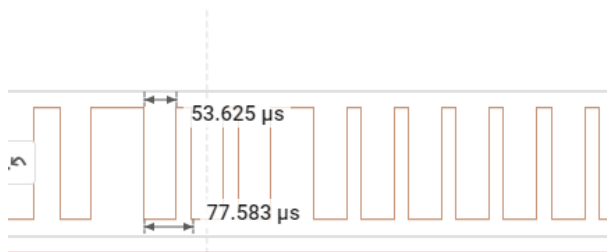


Bit data "0" bit format

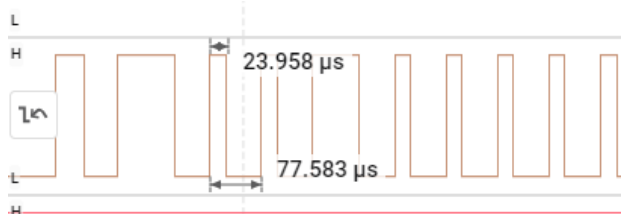
Бита 1:



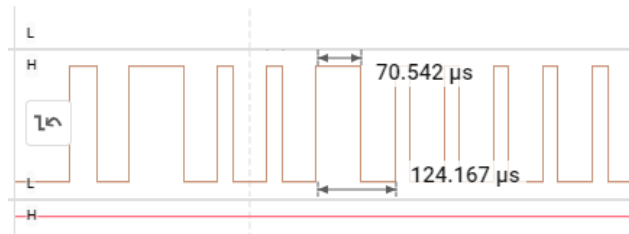
То есть сначала идет низкий уровень в примерно 50 мкс



а далее высокий около 25 мкс для нуля



и около 70 мкс для единицы



Таким образом, последовательность 40 бит данных:

00100000000000000000111010000001101000000:

Первые 8 бит - целая часть влажности, вторые - десятичная часть влажности, третьи - целая часть температуры, четвертые - дробная часть температуры, пятые - контрольная сумма (должна быть равна сумме предыдущих байт):

00100000	00000000	00011101	00000011	01000000
32	0	29	3	64

Таким образом:

Влажность: 32%

Температура: 29.3 C

Контрольная сумма совпадает: $64 = 32 + 29 + 3 = 64$

Формирование временной диаграммы передачи других значений через 1-Wire:

Температура 25,7 C, влажность 50,3%.

40 бит данных:

00110010	00000011	00011001	00000111	01010101
50	3	25	7	85

Временная диаграмма:

