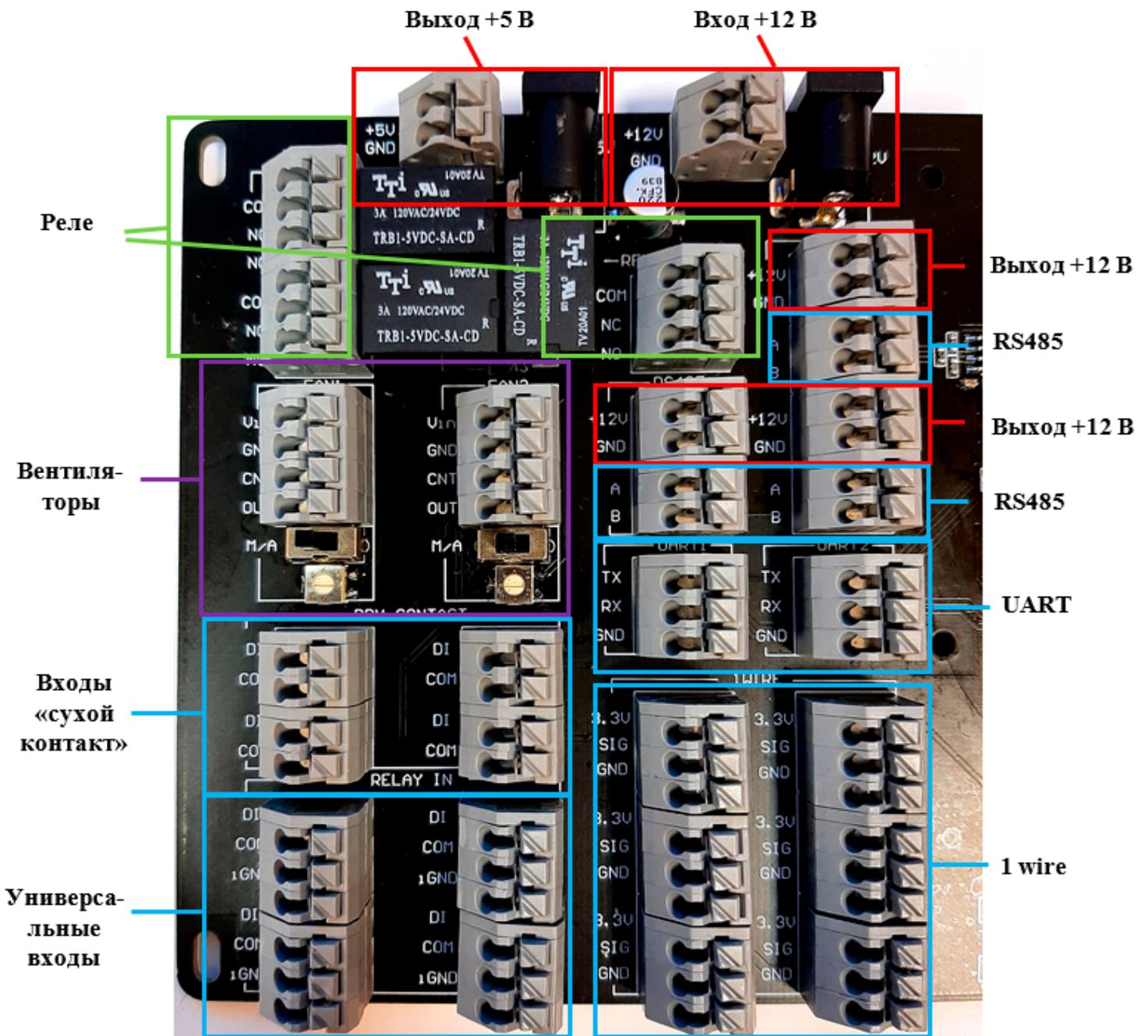


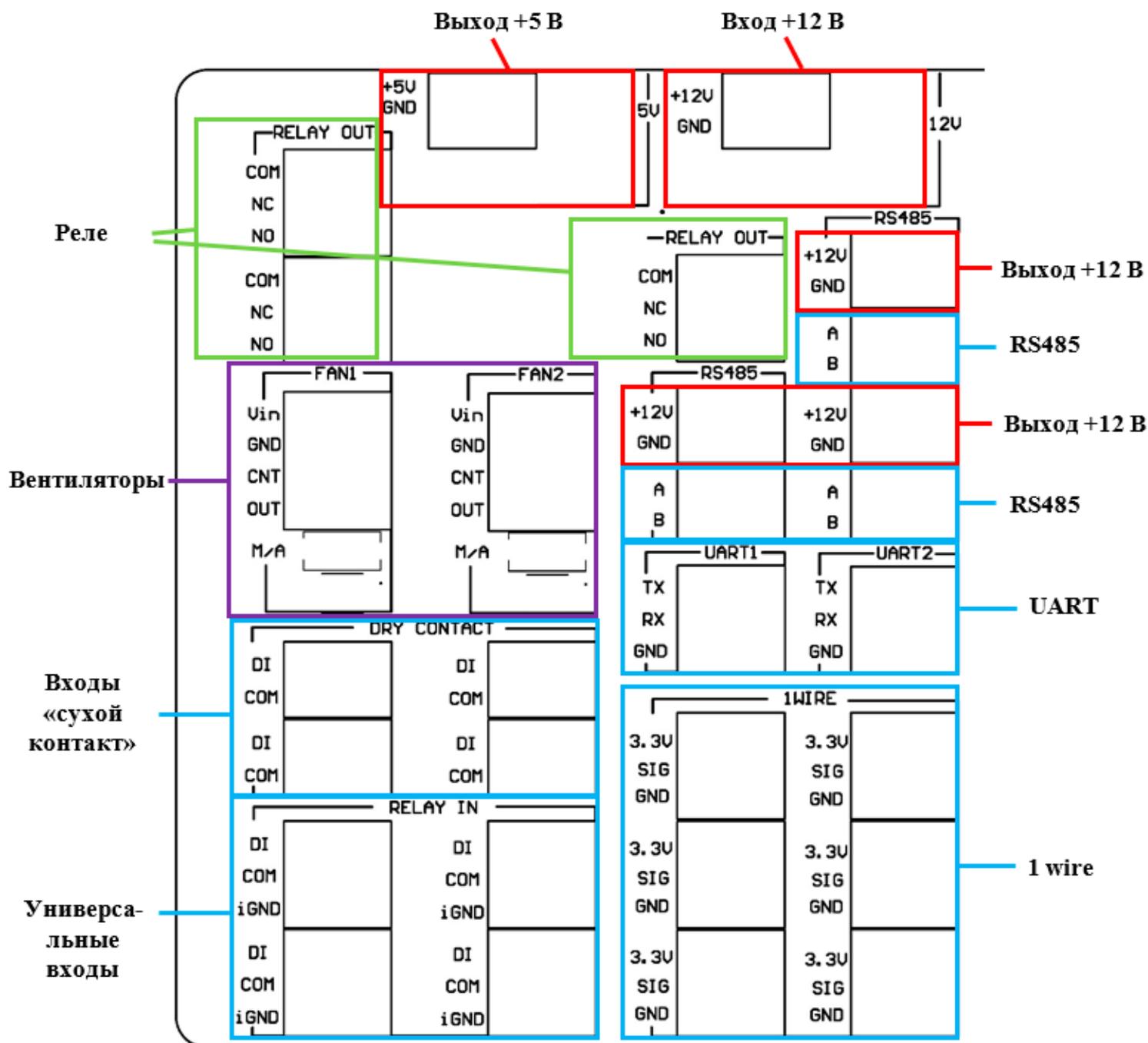
mTASK

Руководство по эксплуатации



Инструкция по подключению к контроллеру SHM-2.0





Питание контроллера осуществляется от постоянного напряжения 12 Вольт через dc jack или через клеммник, которые обозначены на схеме «Вход +12 В».

На плате есть клеммники с выводом входного постоянного напряжения 12 Вольт («Выход +12 В») для питания различных устройств.

«Выход + 5 В» – вывод постоянного напряжения +5 Вольт для подключения внешних устройств через клеммник или dc jack. Ограничение по току 2 А.

«Реле» – 3 канала реле. Каждое реле имеет общий контакт («COM»), нормально закрытый («NC») и нормально открытый («NO») контакты. Реле позволяет управлять нагрузкой до 3А 120 В АС / 24 В DC.

«Вентиляторы» – позволяет управлять двумя вентиляторами как в автоматическом, так и в ручном режимах. Для перехода в ручной режим управления переключатель должен быть сдвинуть в положение «М». При ручном режиме управления – задание скорости вращения вентилятора осуществляется путем вращения потенциометра. Для перехода в автоматический режим управления переключатель должен быть сдвинуть в положение «А». При автоматическом режиме управления – задание скорости вращения вентилятора осуществляется программно.

«Входы «сухой контакт»».

«Универсальные входы» – могут работать в двух режимах работы: «сухой контакт» (рисунок 1 сверху) и режим наличия напряжения (рисунок 1 снизу).

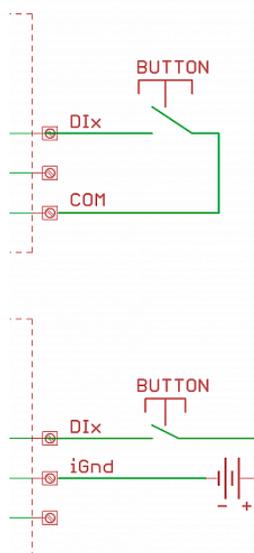


Рисунок 1 - Варианты подключения

Таблица 1 – Режим наличия напряжения

Режим наличия напряжения	
Напряжение срабатывания (обе полярности)	• Есть сигнал — >9 В AC/DC • Нет сигнала — <5 В AC/DC
Номинальное напряжение	12 — 24 В AC/DC
Максимальное допустимое напряжение	50 В AC/DC

«RS485» – интерфейс RS485 линия А и В соответственно шелкографии на плате.

«UART» – 2 отдельных UART интерфейса (лог. уровень 5 В).

«1 wire» – подключение датчиков по 1 wire интерфейсу (лог. уровень 3,3 В).

Инструкция программиста. Общие положения

Инструкция программиста содержит описание программных инструментов взаимодействия с компонентами контроллера.

Программное обеспечение по управлению многофункциональным контроллером располагается на компьютере Raspberry Pi, входящем в состав Изделия. Схема подключения функциональных блоков к GPIO компьютера представлена на рисунке 2.

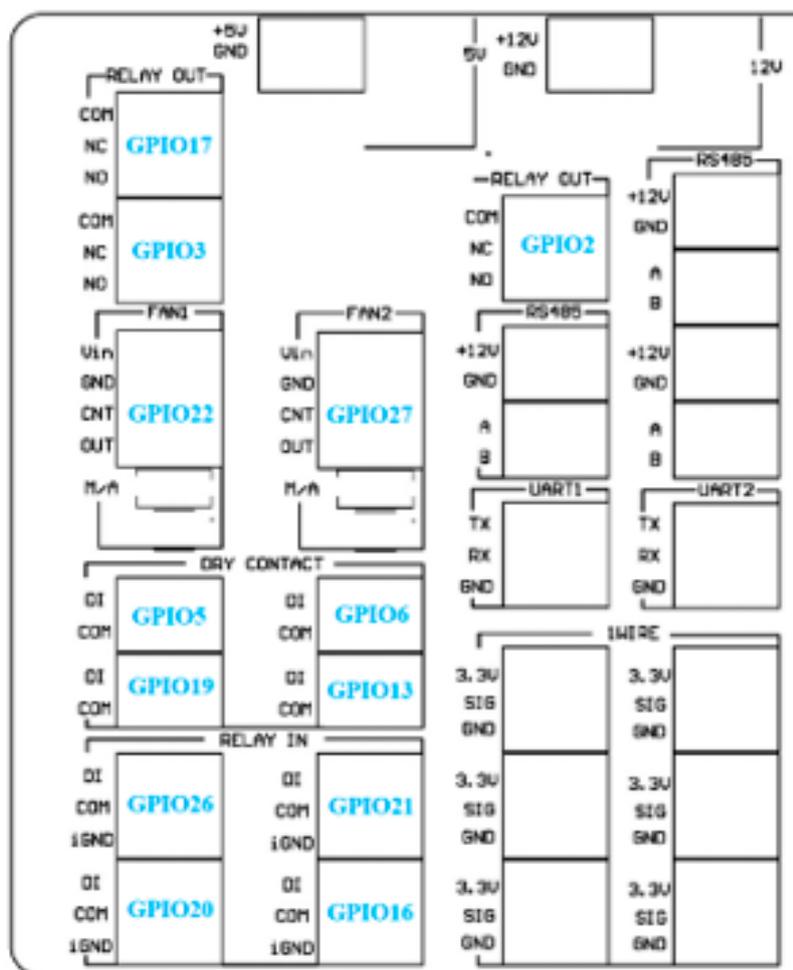


Рисунок 2 - Схема подключения к GPIO Raspberry Pi

Интерфейсы UART1, UART2 и RS485 подключены к Raspberry Pi через USB концентратор. Соответствие порта концентратора и интерфейса представлено в таблице 2.

Таблица 2 – Соответствие порта концентратора и интерфейса

Порт	Интерфейс
Порт 1	UART1
Порт 2	UART2
Порт 3	RS485

Примеры работы с компонентами выполнены с использованием библиотеки wiringPi.

Инструкция программиста. Управление вентиляторами

Изделие позволяет управлять двумя вентиляторами как в автоматическом, так и в ручном режимах. Для перехода в ручной режим управления переключатель должен быть сдвинут в положение "М" ("manual"). При ручном режиме управления – задание скорости вращения вентилятора осуществляется путем вращения потенциометра. Для перехода в автоматический режим управления переключатель должен быть сдвинут в положение "А" ("automatic"). При автоматическом режиме управления – задание скорости вращения вентилятора осуществляется программными средствами. Также есть обратная связь с вентилятором – считывается его текущая скорость вращения (через "сухой" контакт).

Изменение скорости вращения вентилятора осуществляется путем изменения управляющего напряжения. На плате в качестве ЦАП применена резисторная матрица R-2R. С помощью этой матрицы задаются 16 уровней сигнала. Для передачи сигнала на матрицу используется сдвиговый регистр SN74HC164D. Отправка данных в сдвиговый регистр осуществляется с помощью SPI (SPI0). Изменение выходного напряжения выполняется путем отправки байта в сдвиговый регистр. 4 младших бита отвечают за 2-ой вентилятор, 4 старших – за 1-ый вентилятор. В таблице 3 приведена зависимость выходного напряжения от значения в сдвиговом регистре.

Таблица 3 –Зависимость выходного напряжения от значения в сдвиговом регистре

Старшие 4 бита	Напряжение на 2 вентиляторе (В ±0.1)	Младшие 4 бита	Напряжение на 1 вентиляторе (В ±0.1)
16 (0b00010000)	0,7	8 (0b00001000)	0,7
32 (0b00100000)	1,4	4 (0b00000100)	1,4
48 (0b00110000)	2,1	12 (0b00001100)	2,1
64 (0b01000000)	2,7	2 (0b00000010)	2,7
80 (0b01010000)	3,4	10 (0b00001010)	3,4
96 (0b01100000)	4,1	6 (0b00000110)	4,1
112 (0b01110000)	4,9	14 (0b00001110)	4,9
128 (0b10000000)	5,5	1 (0b00000001)	5,5
144 (0b10010000)	6,2	9 (0b00001001)	6,2
160 (0b10100000)	6,9	5 (0b00000101)	6,9
176 (0b10110000)	7,6	13 (0b00001101)	7,6
192 (0b11000000)	8,3	3 (0b00000011)	8,3
208 (0b11010000)	9,0	11 (0b00001011)	9,0
224 (0b11100000)	9,7	7 (0b00000111)	9,7
240 (0b11110000)	10	15 (0b00001111)	10
0 (0b00000000)	0	0 (0b00000000)	0

Работа с устройством mTASK на Raspberry Pi представляет собой работу с файлом. В листинге 1 показан пример отправки байта в сдвиговый регистр. В параметрах при запуске программы передается байт, который должен быть отправлен.

```
int main(int argc, char *argv){
    unsigned int fd;                // file handle
    uint8_t mode = 3;              // SPI mode 3
    // The following calls set up the SPI bus properties
    if ((fd = open(SPI_PATH, O_RDWR)<0) {
        perror("SPI Error: Can't open device.");
        return -1;
    }
    if (ioctl(fd, SPI_IOC_WR_MODE, &mode)==-1) {
        perror("SPI: Can't set SPI mode.");
        return -1;
    }
    if (ioctl(fd, SPI_IOC_RD_MODE, &mode)==-1) {
        perror("SPI: Can't get SPI mode.");
        return -1;
    }

    unsigned char n = *(argv[1]) - '0';
    if (argv[1][1] != '\0') {
        n = n * 10 + (*(argv[1] + 1) - '0');
        if (argv[1][2] != '\0') {
            n = n * 10 + (*(argv[1] + 2) - '0');
        }
    }

    if (transfer(fd, (unsigned char *) &n, 1)==-1){
        perror("Failed to send");
        return -1;
    }

    close(fd);                    // close the file
    return 0;
}
```

Листинг 1 – Отправка данных по mTASK на Raspberry Pi

В листинге 2 приведена функция отправки одного байта по интерфейсу SPI, в которой заполняется структура передачи данных по SPI с указанием массивов передаваемых и принимаемых данных, количества байт, скорости передачи, количества битов в кадре, задержки ожидания, управления линией «CS». После формирования структура передается в модуль SPI, где осуществляется передача данных.

```
int transfer(int fd, unsigned char * send, int len){
    struct spi_ioc_transfer transfer;          // transfer structure

    memset(&transfer, 0, sizeof(transfer));

    transfer.tx_buf = (unsigned long) send;   // buffer for sending data
    transfer.rx_buf = (unsigned long) NULL;   // buffer for receiving data
    transfer.len = len;                       // length of buffer
    transfer.speed_hz = 1000000;             // speed in Hz
    transfer.bits_per_word = 8;              // bits per word
    transfer.delay_usecs = 0;                // delay in us
    transfer.cs_change = 0;                  // affects chip select after transfer8
    // send the SPI message (all of the above fields, inc. buffers)
    int status = ioctl(fd, SPI_IOC_MESSAGE(1), &transfer);
    if (status < 0) {
        perror("SPI: SPI_IOC_MESSAGE Failed");
        return -1;
    }
    return status;
}
```

Листинг 2– Функция отправки одного байта по интерфейсу SPI

Для определения скорости вращения вентиляторов нужно посчитать количество импульсов за минуту. Сигналы с вентиляторов заведены на GPIO22 и GPIO27 (первый и второй вентилятор соответственно). Пример приведен в листинге 3.

```
int counter = 0;

void increase_counter(void) {
    counter++;
}

int main() {
    wiringPiSetupGpio();
    pinMode(22, INPUT);

    wiringPiISR(22, INT_EDGE_RISING, &increase_counter);

    while (1) {
        printf("Rpm %d\n", counter * 60);
        counter = 0;
        sleep(1);
    }
    return 0;
}
```

Листинг 3 – Определение скорости вращения вентилятора

В примере задается функция `increase_counter`, которая будет вызываться при возникновении события возрастающего фронта сигнала. В этой функции увеличивается счетчик импульсов.

В основном цикле раз в секунду выполняется вывод текущего значения импульсов за минуту (подсчет ведется за секунду, поэтому умножаем на 60) и обнуление счетчика.

Инструкция программиста. Считывание сигналов с универсальных входов

Подключены на GPIO16, GPIO20, GPIO21, GPIO26. Если контакты "DI" и "COM" не замкнуты или на выходах "DI" и "iGND" отсутствует необходимое напряжение, то на соответствующем пине считывается логический ноль, при замыкании "DI" и "COM" или при наличии напряжения на входах "DI" и "iGND" – логическая единица.

Пример считывания представлен в предыдущем пункте 3.

Инструкция программиста. Интерфейс 1-wire

Интерфейс подключен к GPIO4. Необходимо разрешить работу интерфейса через raspi-config. В каталоге /sys/bus/w1/devices будут отражены серийные номера устройств, подключенных по 1-wire, для которых есть драйвера в системе.

Для считывания показания температуры с датчика DS18B20 необходимо прочитать содержимое файла СЕРИЙНЫЙ_НОМЕР/w1_slave. Пример содержимого файла представлен на рисунке 3. В первой строке выводимых данных находится информация о достоверности считанной температуры («YES»/ «NO»). Во второй строке – текущая температура (температура указывается в 1000 раз больше).

```
a2 01 4b 46 7f ff 0e 10 d8 : crc=d8 YES  
a2 01 4b 46 7f ff 0e 10 d8 t=26125
```

Рисунок 3 – Пример содержимого файла датчика DS18B20

Инструкция программиста. Интерфейс RS485 и 2 интерфейса UART

В подключенных устройствах на Raspberry должны появиться 3 виртуальных COM порта (один для RS485 и два для UART). Работа с интерфейсами – прием и передача данных через эти COM порты.

Инструкция программиста. Управление реле

Выходы управления реле подключены на GPIO2, GPIO3, GPIO17. Для включения реле необходимо на соответствующий вход подать логическую единицу, для выключения – логический ноль.

Для конфигурации вывода в режим «OUT» необходимо выполнить в терминале – «gpio mode 8 out». Для отправки логической единицы – «gpio write 8 1» (замыкание реле). Для отправки логического нуля – «gpio write 8 0» (выключение реле).