

Протокол обмена датчика СЕНСОР-М с интерфейсом 1-Wire®

В.1 Описание протокола 1-Wire®

Протокол 1-Wire разработан фирмой Dallas Semiconductor (с 2001 года Maxim Integrated) для обмена данными с устройствами используя один провод для передачи данных (далее шина). Скорость передачи 15,4 кбит/с. На шине могут быть параллельно подключены несколько ведомых устройств «slave» и только одно ведущее «master». Сеть устройств с протоколом 1-Wire имеет название «MicroLan». Датчики СЕНСОР-М с выходом 1-Wire являются slave-устройствами, совместимы с другими устройствами протокола 1-Wire и могут работать совместно в единой сети MicroLan. Длительность импульсов RESET, PRESENCE, слоты времени передачи 1 и 0, а также порядок передачи байт и бит для датчиков СЕНСОР-М соответствуют спецификации 1-Wire (см. Приложение А). Требуемая пауза между байтами при записи в slave **не менее 65мкс**(1 тайм- слот) при чтении из slave **не менее 130мкс**(2 тайм-слота).

Master в любой момент может прервать сессию обмена данными передачей импульса RESET. Функция «паразитного питания» от шины в датчиках СЕНСОР-М не применяется.

В.2 Уникальный 64-битный код 1-Wire® в датчиках СЕНСОР-М

Каждое 1-Wire устройство имеет уникальный идентификационный 64-битный код (8 байт), который записан при выпуске в регистре ROM устройства и доступен только для чтения, он же является адресом устройства в сети MicroLan. Младший байт ROM – это код семейства, для датчиков СЕНСОР-М – код 0xC1. Следующие 6 байт – уникальный в семействе серийный номер, модель и версия. Последний байт – контрольная сумма CRC8, которая служит для проверки правильности приема всего кода. Состав регистра ROM в датчиках СЕНСОР-М представлен в табл. В.2. ROM-код для доступа к датчику можно считать командой READ ROM, получить алгоритмом поиска в сети командой SEARCH ROM, считать с маркировочной таблички датчика или паспорта, составить самостоятельно по данным маркировки датчика.

В.3 Команды протокола 1-Wire® в датчиках СЕНСОР-М

В.3.1 Стандартные ROM-команды

Команды ROM служат для доступа к устройствам 1-Wire и поиска устройств, подключенных к сети MicroLan. Команды ROM, которые поддерживают все 1-Wire устройства, в том числе датчики СЕНСОР-М, см. в табл. В.1.

Команды ROM 1-Wire устройств

Таблица В.1

Команда	Код HEX	Описание
SKIP ROM	0xCC	доступ к функциональным командам (CMD) устройства без сверки ROM
MATCH ROM	0x55	сверить 8 байт, передаваемых после команды с ROM устройства, доступ к CMD при совпадении
READ ROM	0x33	прочитать 8 байт ROM(корректно если в сети только 1 slave)
SEARCH ROM	0xF0	выполнить передачу бит ROM по алгоритму поиска, как указано в спецификации 1-Wire.

В.3.2 Функциональные команды, CMD – команды

Функциональные команды предназначены для чтения/записи регистров и сервисных функций датчика. Условные обозначения в описании команд:

RST – импульс сброса шины , RESET

PD – импульс присутствия slave на шине (PRESENCE detect)

ROMcmd – команда ROM, 1байт или 1байт+ROM-код(8байт) для MATCH ROM

ADDR – адрес чтения/записи памяти, см. табл. В.6-В.9

SP0..SP6 - байты регистра ScratchPad

MO..M15 – байты регистра MESS

DO..D15 – байты данных

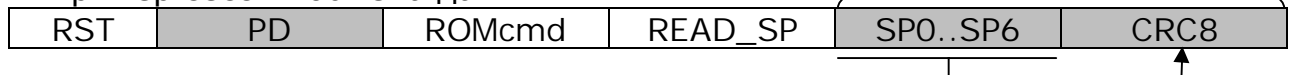
subCmd – байт субкоманды в команде SERV, см табл. В.9

CRC8 – контрольная сумма (1 байт)

 - master передает для slave  - slave передает для master

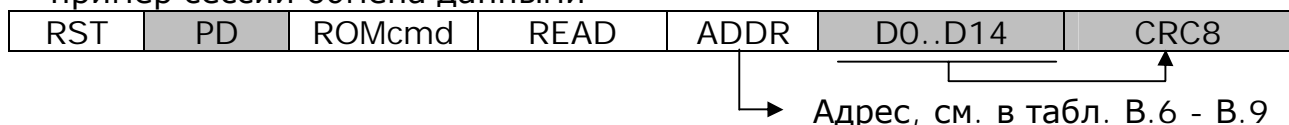
READ_SP (0xBE) - прочитать 8 байт регистра ScratchPad датчика
(состав регистра ScratchPad см. в табл. В.2)

- пример сессии обмена данными



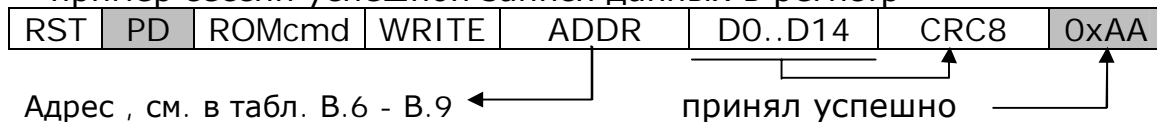
READ (0x45) - прочитать 16 байт регистра датчика с адреса ADDR

- пример сессии обмена данными

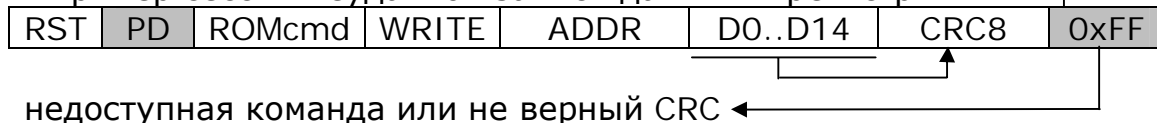


WRITE (0x65) - записать 16 байт в регистр датчика

- пример сессии успешной записи данных в регистр



- пример сессии неудачной записи данных в регистр

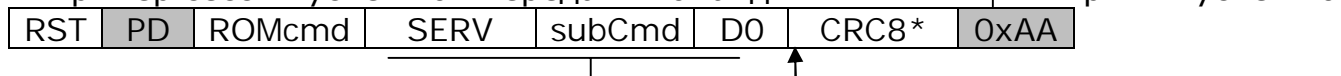


* - после сессии записи данных до начала следующей сессии обмена датчику требуется пауза 100 мс для записи принятых данных в регистр EEPROM

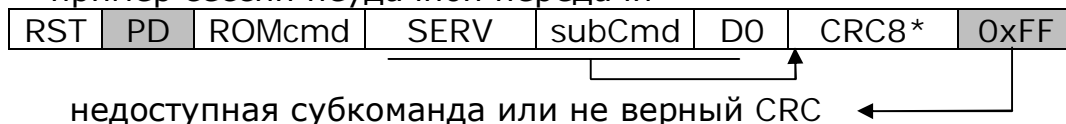
SERV (0x08) – сервис параметров датчика

Команда используется для управления настройками датчика в зависимости от субкоманды и передаваемого байта данных D0 см. Табл. В.9

- пример сессии успешной передачи команды



- пример сессии неудачной передачи



*- для передачи master вычисляет CRC8 по 3-м байтам (SERV, subCmd, D0)

WAKEUP (0x40) – вывести датчик из спящего режима и измерять на 100мс. При записи командой SERV (0x08) с субкомандой setSleep (0xB0) байта D0=0x01 датчик переходит в режим sleep с низким энергопотреблением. Ток питания см. на странице 5. Команда WAKEUP используется для запуска измерений и обновления данных в регистре ScratchPad при работе датчика в режиме sleep. Для активного режима датчика (командой SERV (0x08) с субкомандой в setSleep (0xB0) , записан D0=0) команда WAKEUP не используется. При выпуске датчика установлен активный режим.

- пример сессии успешного выполнения команды

RST	PD	ROMcmd	WAKEUP	0xAA	- перешел в активный режим на 100 мс
-----	----	--------	--------	------	--------------------------------------

- пример сессии неудачного выполнения команды

RST	PD	ROMcmd	WAKEUP	0xFF	- команда не принята, не выполнена
-----	----	--------	--------	------	------------------------------------

* - достоверные данные в ScratchPad ,будут ч-з 100мс после WAKEUP и датчик опять переходит в спящий режим

В.4 Организация и доступ к памяти в датчике СЕНСОР-М

Доступные по протоколу 1-Wire данные расположены в регистрах оперативной памяти RAM – регистр ScratchPad и энергонезависимой памяти EEPROM – все другие данные. Адресация, тип данных и доступность для чтения/записи (далее R/W) указаны в таблицах В.2 – В.9 . Младший байт регистров указан в верхней строке таблиц.

Регистр ROM

Таблица В.2

Тип	Байт	Обозначение	Описание (значения)	Доступ
u_char	0	FamCode	код семейства СЕНСОР-М= 0xC1	R
u_char	1	Mcode	код модели = модель-100	R
u_char	2	VerApr	Вер. аппарат. обеспечения Табл.В.3	R
u_char	3	VerPrg	версия ПО (v1.1.1->число 111)	R
u_char	4	SNO	серийный номер датчика 0..32630	R
u_char	5	SN1		R
u_char	6	reserved	пока 0x00	R
u_char	7	CRC8	CRC8 по 7 предыдущим байтам	R

Байт VerApr

Таблица В.3

бит7	бит6	бит5	бит4	бит3	бит2	бит1	бит0
Осн. погрешность (Acc)			Термокомпенсация (ТК)		Исполнение (Option)		
000 – 1%			00 – t1 (5..50°C)		000 – «-»		
001 – 0,5%			01 – t2 (-30..80°C)		001 - И		
010 – 0,25%			10 – t3 (-40..80°C)		010 – И1		
011 – 0,15%			11 – «-»		011 - Ex		
100 – 0,1%					100 - H		
					101 - H1		
					110 - Г		

Пример содержимого регистра ROM в HEX-формате:

25 00 1F 42 6F 4D 19 C1	-> СЕНСОР-М-125-Н1-t2-0.25 v1.1.1 зав.№8002
MSB	LSB

Регистр ScratchPad (чтение командой *READ_SP (0xBE)*)**Таблица В.4**

Тип	Байт	Обозначение	Описание (значения)	Доступ
u_char	0	UC	тек. Ед. измерения Табл. В.5	R
float*	1-4	P	текущее давление	R
u_char	5	t	тек. Температура, °C (-127..127)	R
char	6	Status1	байт тек. статуса датчика, Табл Г.3	R
u_char	7	CRC8	CRC8 по 7 предыдущим байтам	R

*- число с плавающей точкой (4 байта формата IEEE 754)

Байт Status1**Таблица Г.3**

Бит	Значение бита при установке в 1
7	неисправность датчика: не найден АЦП или залипла кнопка
6	конфигурация настроек изменялась командами или меню
5	холодный старт, это первый обмен данными после сброса
4	доступно больше статус-данных по ком. 48
3	Аналоговый выход в зафиксированном положении
2	Аналоговый вых. сигнал в насыщении (3,8мА/0,38В или 22,5мА/2,25В)
1	температура за пределами диапазона эксплуатации
0	давление за пределами диапазона измерения

- при нормальном режиме работы Status1=0x00

Коды единиц измерения***Таблица В.5**

Код ед. изм. (UC)	Обозначение	Коэф. Пересчета из kPa
4	mmH2O	101,972
6	psi	0,14504
7	bar	0,01
8	mbar	10
10	kg/cm2	0,0102
11	Pa	1000
12	kPa	1
14	atm	0,00987
237	MPa	0,001

*- независимо от единиц калибровки сенсора (UCS), пользователь может установить желаемые ед. изм. датчика, записав командой *SERV(0x08)* с субкомандой *setUC(0x7B)* соответствующий код UC. При записи некорректного кода UC датчик установит UC=UCS.

Регистр INFO (адрес ADDR=0x08)**Таблица В.6**

Тип	Байт	Обознач.	Описание (значения)	Доступ
u_char	0	UCS	Код Ед.изм. калибровки Табл. В.5	R
float*	1-4	Pmin	Нижний предел измерения давления	R
float*	5-8	Pmax	Верхний предел измерения давления	R
u_char	9-11	Data[3]	дата выпуска : день, месяц, год-1900	R
u_char	12	UC	Код текущих Ед. Изм. Табл. В.5	R
u_char	13	Sleep	режим sleep (1-sleep, 0-active)	R
u_char	14	Damp	пост. времени фильтра [сек.*10]	R
u_char	15	CRC8	CRC8 по 15 предыдущим байтам	R

*- число с плавающей точкой (4 байта формата IEEE 754)

Регистр MESS (адрес ADDR=0x96)**Таблица В.7**

Тип	Байт	Обозначение	Описание (значения)	Доступ
char	0-14	Mess[15]	Сообщение пользователя 15 симв.	R/W
u_char	15	CRC8	CRC8 по 15 предыдущим байтам	R/W

Регистр DESCR (адрес ADDR=0x87)**Таблица В.8**

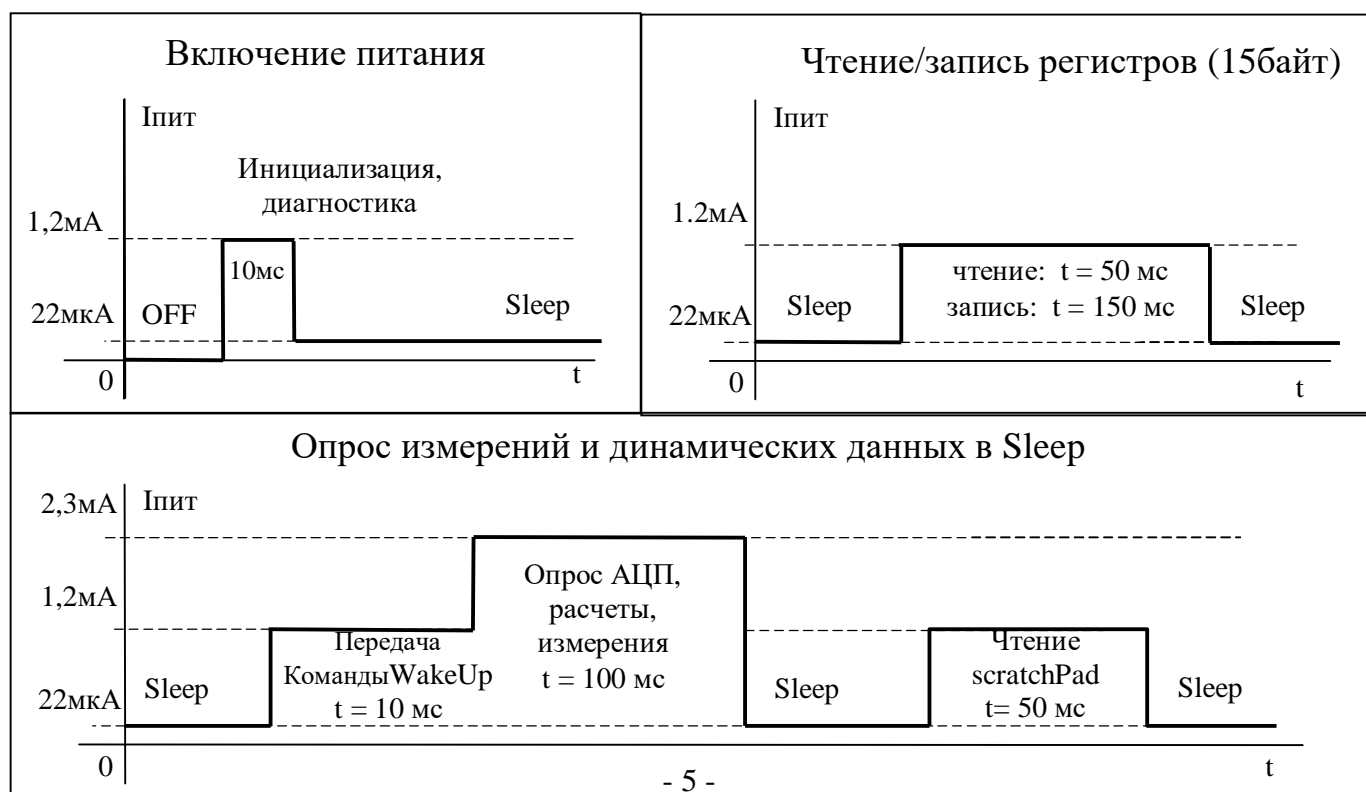
Тип	Байт	Обознач.	Описание (значения)	Доступ
char	0-11	Descr[12]	Описатель датчика, 12 символов	R/W
u_char	12-14	DataU[3]	Дата пользователя: день, месяц, год-1900	R/W
u_char	15	CRC8	CRC8 по 15 предыдущим байтам	R/W

Субкоманды в команде SERV (0x08)**Таблица В.9**

ID	subCmd	DO	Описание субкоманды
setSleep	0xB0	0/1	Установить режим Sleep=DO (см. потребл. Тока)
setUC	0x7B	4..237	Установить ед.изм., код UC =DO (Табл. В.5)
setDamp	0xB1	2..40	Установить пост. фильтра Damp=DO/10 [с]
RESET*	0xFF	0	Выполнить рестарт датчика
setKey	0x65	KEY*	Записать ключ доступа в cmdEx
cmdEx	0x11	0xAA	Выполнить Авто0 (аналогично нажатию кнопки)
		0xBB	Восстановить заводские настройки
		0xDD	Запомнить текущее АЦП для Pmin
		0xEE	Запомнить текущее АЦП для Pmax
		0xDE*	запомнить нелинейность при P=(Pmax-Pmin)/2

* время инициализации после рестарта или подачи питания 200 мс.
 * cmdEx выполняются если записан KEY=SNO (табл.В.2), ключ действует до рестарта
 * реализовано в v1.0.1 и выше

Потребление тока питания в спящем режиме (Sleep)



В.5 Контрольная сумма CRC8 в конце сообщения



Для контроля целостности сообщения master отправляет последним байтом контрольную сумму (далее CRC8), рассчитанную в соответствии с алгоритмом 1-Wire. Slave рассчитывает CRC8 по принятым байтам и сравнивает с принятым CRC8. Если рассчитанное CRC8 равно принятому, то slave обрабатывает сообщение, иначе – игнорирует (ошибка обмена). При ошибке обмена master может повторять запросы к slave .

Пример функции для расчета CRC8 для сообщения 1-Wire на языке СИ :

```
unsigned char crc8(unsigned char *addr, unsigned char len) {
    unsigned char crc = 0;
    while (len--) { //цикл по байтам
        unsigned char inbyte = *addr++;
        for (unsigned char i = 8; i; i--) { //цикл по битам тек. байта
            unsigned char mix = (crc ^ inbyte) & 0x01;
            crc >>= 1;
            if (mix) crc ^= 0x8C;
            inbyte >>= 1;
        }
    }
    return crc; //результат расчета
}
```

В.6 Тестовое ПО и оборудование

Для работы с slave-устройствами 1-Wire пользователь может создать master на базе микроконтроллера и обработать сигналы 1-Wire на ножках микроконтроллера программно. Для опроса сети 1-Wire с ПК пользователь может использовать адаптеры USB/1-Wire (DS9490), RS232/1-Wire (DS9097) и приложения Master1W.exe , OneWireViever.exe или создать собственное приложение. Пример приложения Demo1W.exe, исходный код, библиотека функций, драйвера для адаптеров, инструкцию к ПО можно скачать с сайта www.belsensor.by. Адаптер СЕНСОР-USB/1W для подключения датчиков и других 1-Wire устройств к ПК а также ручной коммуникатор СЕНСОР-ПК (с автономным питанием) для опроса датчика и сервиса параметров потребитель может заказать в комплекте с датчиками или отдельно. Контакты для заказа на последней странице настоящего Руководства.

Адаптер СЕНСОР-USB/1W	Ручной коммуникатор СЕНСОР-ПК
	

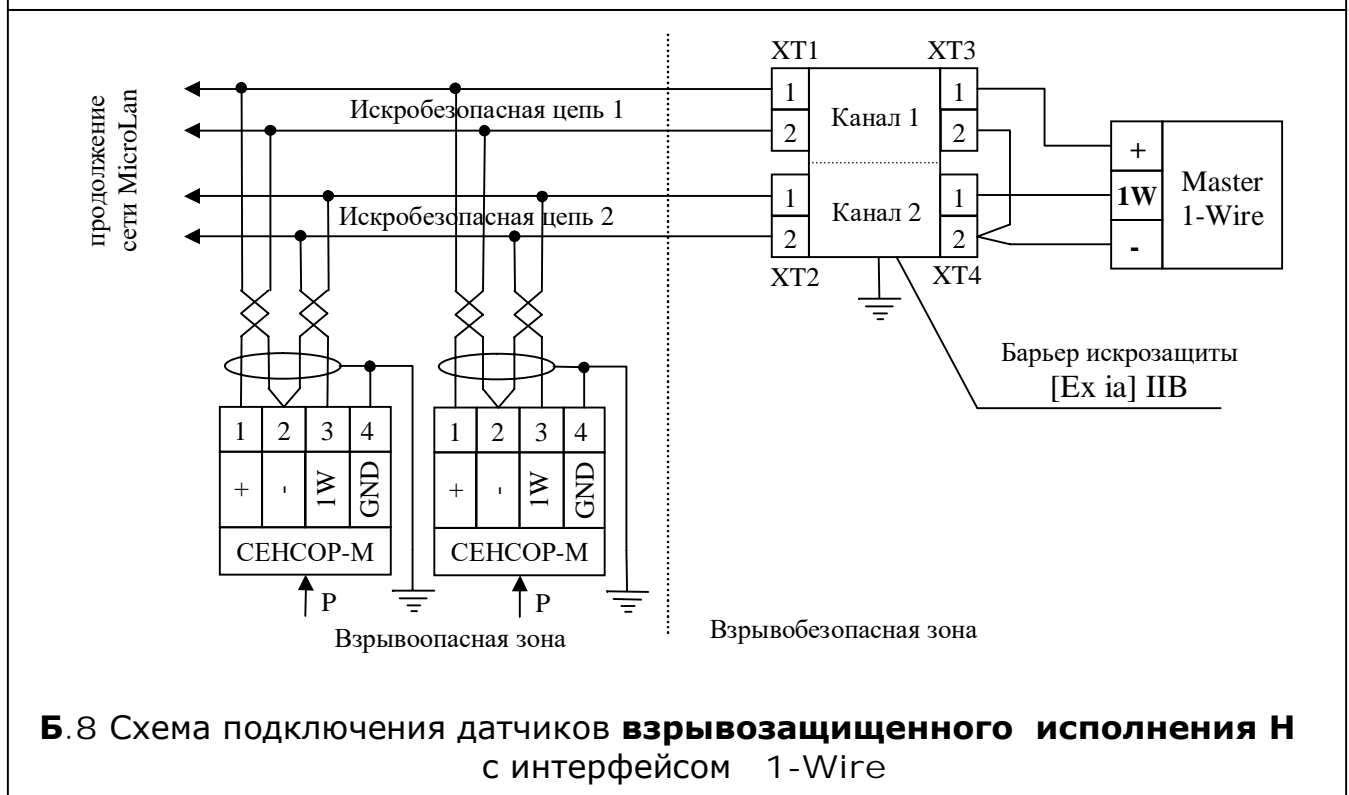
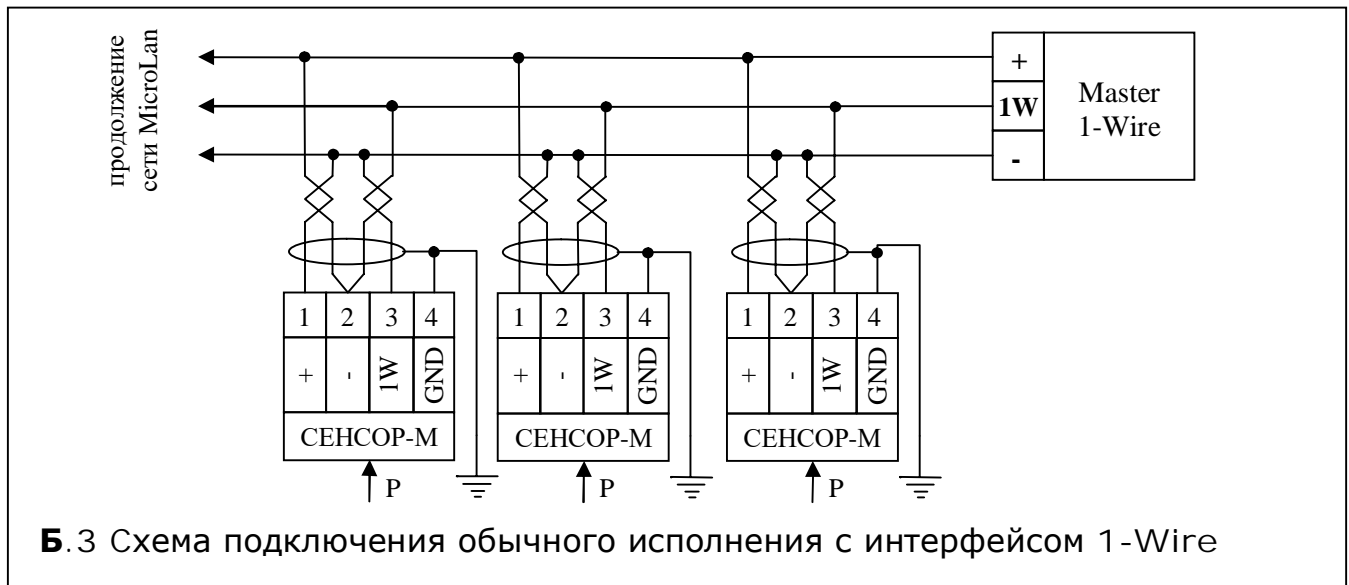
В.7 ЛИНИИ СВЯЗИ К ДАТЧИКАМ СЕНСОР-М

ТАБЛИЦА Б.1

Исполнение	Выходной сигнал, Интерфейс	Рекомендуемый тип кабеля	Длина*, м не более	Схема
Обычное, Г, И1	1-Wire	2 витые пары в экране	100	Б.3
Н	1-Wire	2 витые пары в экране	40	Б.8

* - суммарная длина всех линий связи в схеме(в сети) подключения

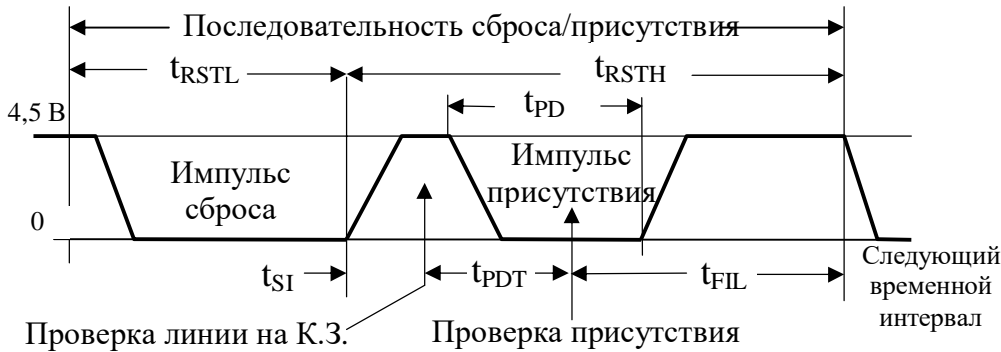
СХЕМЫ ВНЕШНИХ ЭЛЕКТРИЧЕСКИХ СОЕДИНЕНИЙ ДАТЧИКА.



ПРИЛОЖЕНИЕ А

Временные диаграммы 1-Wire

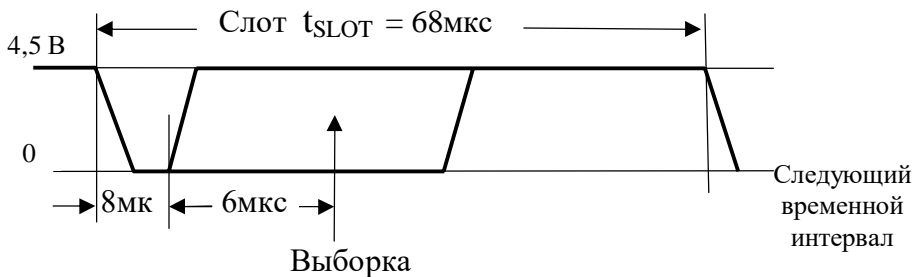
Каждый сеанс обмена данными мастер начинает с импульса сброса, затем отпускает шину 1-Wire в высокий уровень и проверяет шину на короткое замыкание. Если высокий уровень в момент проверки на К.З. не обнаружен то мастер регистрирует ошибку обмена 1. Датчик после обнаружения импульса сброса затягивает шину в низкий уровень и формирует импульс присутствия. Мастер регистрирует ошибку если не обнаруживает импульс присутствия датчика.



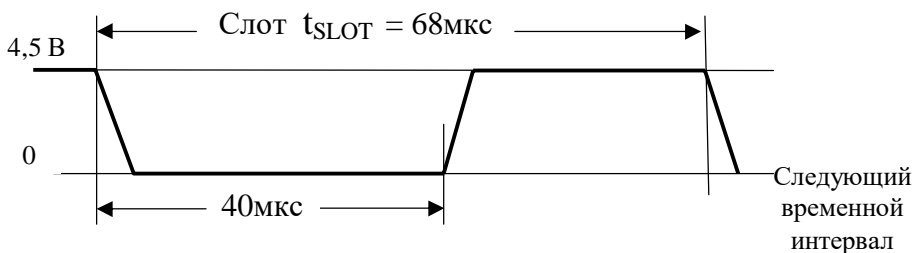
t_{RSTL}	t_{RSTH}	t_{SI}	t_{PDT}	t_{PD}	t_{FILL}
512 мкс	584 мкс	8 мкс	64 мкс	150 мкс	512 мкс

Для передачи логических "1" и "0" в запросе мастера и чтения в ответе датчика используются интервалы времени "слоты"

Передача от мастера к датчику "1" и чтение ответа датчика.



Передача от мастера к датчику "0".



После выполнения процедуры сброс/присутствие мастер передает байты запроса к датчику побитно. Для передачи логической "1" мастер формирует фронт вниз на шине 1-Wire, удерживает шину в течение 5-8 мкс и отпускает. Для передачи логического "0" мастер формирует фронт вниз, удерживает шину 40 мкс и отпускает. Датчик считывает состояние шины через 15 мкс после обнаружения фронта вниз на шине и определяет передаваемый ему бит данных (высокий уровень - 1, низкий уровень - 0).

После передачи запроса к датчику мастер начинает чтение ответа датчика. Для чтения мастер формирует фронт вниз на шине, удерживает шину 5-8 мкс, отпускает и считывает состояние шины через 14 мкс после фронта вниз. При передаче датчиком "1" в момент выборки мастер считывает на шине высокий уровень, при передаче датчиком "0" мастер считывает на шине низкий уровень.