

Delta Electronics, Inc[®]

Д о п о л н е н и е
к
РУКОВОДСТВУ ПО ЭКСПЛУАТАЦИИ
преобразователей частоты серии
VFD-L
(220 В 0.2 – 0.75 кВт)

Настоящее дополнение к руководству по эксплуатации (далее по тексту дополнение) распространяется на преобразователи частоты (ПЧ) серии VFD-L предназначенные для управления частотой вращения трехфазных асинхронных двигателей максимальной мощностью соответственно 0,2; 0,4; 0,75 кВт питанием от сети переменного тока напряжением 220/230В частотой 50/60 Гц.

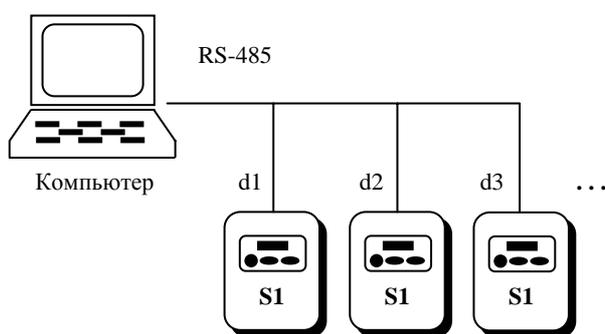
Дополнение описывает параметры коммуникации по последовательному интерфейсу RS-485.

Группа 9: Параметры коммуникации

Пользователи могут устанавливать параметры и управлять работой преобразователя частоты через последовательный интерфейс RS-485 с помощью промышленного контроллера или компьютера (далее по тексту, компьютер).

9-00	Коммуникационный адрес	Заводская уставка: d1
	Диапазон установки: 1 ... 247	Дискретность: 1
	Этот параметр может устанавливаться во время работы привода	

Возможно одновременное управление до 247 преобразователями от одного контроллера. Для идентификации конкретного преобразователя при коммуникации каждому ПЧ параметром Pr.9-00 устанавливается индивидуальный адрес.



9-01	Скорость передачи данных	Заводская уставка: d1
	Возможные значения: d0: 4800 бод (бит/сек); d1: 9600 бод; d2: 19200 бод.	
	Этот параметр может устанавливаться во время работы привода	

Этот параметр используется для установки скорости передачи между преобразователем и компьютером.

9-02	Реакция на ошибки в передаче	Заводская уставка: d0
	Возможные значения: d0: предупредить и продолжить работу; d1: предупредить и остановить привод с замедлением; d2: предупредить и остановить привод, - моментально обесточив двигатель. d3: нет реакции.	
	Этот параметр может устанавливаться во время работы привода	

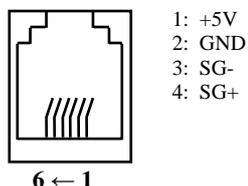
9-03	Modbus сторожевой таймер	Заводская уставка: d0
	Возможные значения: d0: не доступен; d1: 1 сек; d20: 20 сек.	
	Этот параметр может устанавливаться во время работы привода	

Если сторожевая функция таймера активизирована, таймер запустится на отсчет времени, как только первый достоверный Modbus сигнал связи будет получен после включения питания или сброса. Таймер сбросит значение счетчика времени при получении каждого достоверного Modbus сообщения. Если значение счетчика достигнет значения параметра Pr. 9-03, привод остановится и выведет на дисплей сообщение «CE10». Эту блокировку можно сбросить с внешнего терминала, клавишей RESET с цифрового пульта управления или Modbus командой сброса по RS-485.

9-04	Протокол коммуникации	Заводская уставка: d0
	Диапазон установки: d0: Modbus ASCII режим, протокол <7, N, 2>; d1: Modbus ASCII режим, протокол <7, E, 1>; d2: Modbus ASCII режим, протокол <7, 0, 1>; d3: Modbus ASCII режим, протокол <8, N, 2>; d4: Modbus ASCII режим, протокол <8, E, 1>; d5: Modbus ASCII режим, протокол <8, 0, 1>; d6: Modbus RTU режим, протокол <8, N, 2>; d7: Modbus RTU режим, протокол <8, E, 1>; d8: Modbus RTU режим, протокол <8, 0, 1>.	Дискретность: 1 сек
С помощью этого параметра выбирается протокол коммуникации. Параметр можно устанавливать во время работы привода.		

1. Управление преобразователем от компьютера:

- связь компьютера с ПЧ осуществляется по последовательному интерфейсу через разъем RJ-11, расположенный планке управляющих терминалов. Назначение контактов разъема приведено ниже:



Каждый ПЧ имеет индивидуальный коммуникационный адрес, устанавливаемый с помощью параметра Pr.9-00. Компьютер управляет каждым ПЧ, различая их по адресу.

- преобразователь FDD-L может быть настроен для связи в Modbus сетях, использующих один из следующих режимов: ASCII (Американский Стандартный Код для Информационного Обмена) или RTU (Периферийное устройство). Пользователи могут выбирать режим наряду с протоколом связи последовательного порта, используя параметр Pr.9-04.

Режим ASCII:

Каждый 8-bit блок данных есть комбинация двух ASCII символов. Для примера, 1-байт данных: 64 Hex, показан как '64' в ASCII, состоит из '6' (36 Hex) и '4' (34Hex).

Символ	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'
ASCII код	30H	31H	32H	33H	34H	35H	36H	37H
Символ	'8'	'9'	'A'	'B'	'C'	'D'	'E'	'F'
ASCII код	38H	39H	41H	42H	43H	44H	45H	46H

Режим RTU:

Каждый 8-bit блок данных - комбинация двух 4-битных шестнадцатиричных символов. Для примера, 64 Hex.

2. Формат данных:

2.1. 10-bit кадр передачи (для 7-битного блока данных)

(7, N, 2: Pr.9-04=0)



(7, E, 1 : Pr.9-04=1) с проверкой на четность (even parity)

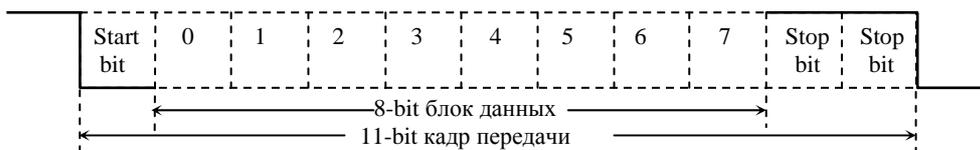


(7, 0, 1 : Pr.9-04=2) с проверкой на нечетность (odd parity)

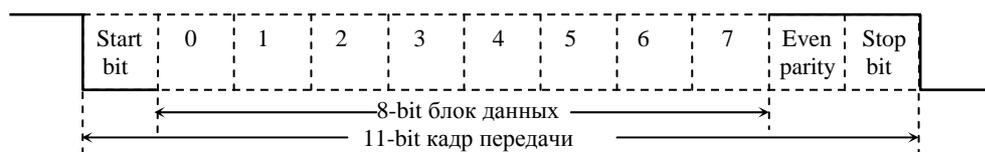


2.2. 11-bit кадр передачи (для 8-bit блока данных):

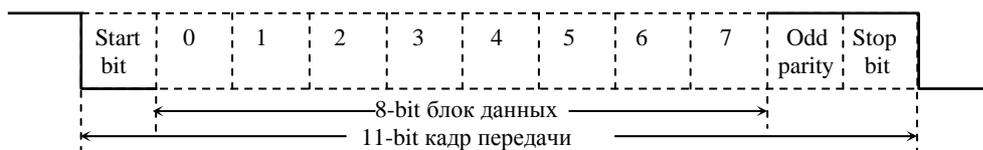
(8, N ,2 : Pr.9-04=3 или 6)



(8, E, 1: Pr.9-04 = 4 или 7) с проверкой на четность (even parity)



(8, 0, 1: Pr.9-04 = 5 или 8) с проверкой на нечетность (odd parity)



3. Протокол коммуникации

3.1 Коммуникационный блок данных:

ASCII режим

STX	Стартовый символ ':' (3AH)
ADR1	Коммуникационный адрес: 8-bit адрес, состоящий из 2 ASCII кодов
ADR0	
CMD1	Командный код: 8-bit адрес, состоящий из 2 ASCII кодов
CMD0	
DATA(n-1)	Содержание данных: n x 8-bit данных, состоящих из 2-х ASCII кодов n<=25, максимум 50 ASCII кодов
...	
DATA0	
LRC CHK 1	LRC контрольная сумма: 8-bit контрольная сумма, состоящая из 2 ASCII кодов
LRC CHK 0	
END1	Конец символов: END1= CR (ODH), ENDO= LF(OAH)
END0	

RTU режим

START	интервал молчания - более 10 мс
ADR	Адрес коммуникации: 8-bit адрес
CMD	Код команды: 8-bit команда
DATA (n-1)	Содержание данных: n x 8-bit данных. n<=25
...	
DATA0	
CRC CHK Low	CRC контрольная сумма: 16-bit контрольная сумма из 2-ух 8-bit символов
CRC CHK High	
END	интервал молчания - более 10 мс

3.2. ADR (Коммуникационный адрес):

Допустимый коммуникационный адрес должен быть выбран из диапазона 0 ... 247. Коммуникационный адрес равный 0 – средство трансляции всем ПЧ (VFD) одновременно, в этом случае, ПЧ не будут отвечать ни на какое сообщение ведущему устройству.

Для примера, связь VFD с адресом 16 decimal:

ASCII режим: (ADR 1, ADR 0)='1','0' => '1'=31H, '0'=30H

RTU режим: (ADR)=10H

3.3. CMD (код команды) и DATA (символы данных):

Формат символов данных зависит от командных кодов. Доступные командные коды - 03H, чтение N слов. Максимальное значение N это 12. Для примера, чтение непрерывных 2 слов от начального адреса 2102H VFD с адресом 01H.

ASCII режим:

Командное сообщение:	
STX	‘.’
ADR 1	‘0’
ADR 0	‘1’
CMD 1	‘0’
CMD 0	‘3’
Стартовый адрес данных	‘2’ ‘1’ ‘0’ ‘2’
Число данных (в словах)	‘0’ ‘0’ ‘0’ ‘2’
LRC CHK 1	‘D’
LRC CHK 0	‘7’
END 1	CR
END 0	LF

Ответное сообщение:	
STX	‘.’
ADR 1	‘0’
ADR 0	‘1’
CMD 1	‘0’
CMD 0	‘3’
Число данных (в байтах)	‘0’ ‘4’
Содержание данных по стартовому адресу 2102H	‘1’ ‘7’ ‘7’ ‘0’
Содержание данных по адресу 2103H	‘0’ ‘0’ ‘0’ ‘0’
LRC CHK 1	‘7’
LRC CHK 0	‘1’
END 1	CR
END 0	LF

RTU режим

Командное сообщение:	
ADR	01H
CMD	03H
Стартовый адрес данных	21H 02H
Число данных в словах	00H 02H
CRC CHK Low	6FH
CRC CHK High	F7H

Ответное сообщение:	
ADR	01H
CMD	03H
Число данных в байтах	04H
Содержание данных по адресу 2102H	17H 70H
Content of data address 2103H	00H 00H
CRC CHK Low	FEH
CRC CHK High	5CH

Код команды: 06H, запись 1 слово.

Для примера, запись 6000(1770H) в адрес 0100H VFD с адреса 01H.

ASCII режим

Сообщение команды:	
STX	‘.’
ADR1	‘0’
ADR0	‘1’
CMD1	‘0’
CMD0	‘6’
Адрес данных	‘0’
	‘1’
	‘0’
	‘0’
Содержание данных	‘1’
	‘7’
	‘7’
	‘0’
LRC CHK 1	‘7’
LRC CHK 0	‘1’
END1	CR
END0	LF

Ответное сообщение:	
STX	‘.’
ADR1	‘0’
ADR0	‘1’
CMD1	‘0’
CMD0	‘6’
Адрес данных	‘0’
	‘1’
	‘0’
	‘0’
Содержание данных	‘1’
	‘7’
	‘7’
	‘0’
LRC CHK 1	‘7’
LRC CHK 0	‘1’
END1	CR
END0	LF

RTU режим

Сообщение команды:	
ADR	01H
CMD	06H
Адрес данных	01H
	00H
Содержание команды	17H
	70H
CRC CHK Low	86H
CRC CHK High	22H

Ответное сообщение:	
ADR	01H
CMD	06H
Адрес данных	01H
	00H
Содержание команды	17H
	70H
CRC CHK Low	86H
CRC CHK High	22H

3.4. CHK (проверка суммы)**ASCII режим:**

LRC (продольная проверка избыточности) рассчитана в итоге, модуль 256, значение байтов от ADR1 до последнего символа данных, тогда вычисление шестнадцатичного представления 2-ух дополнений отрицание суммы. Для примера, читая 1 слово с адреса 0401H преобразователя с адресом 01H.

STX	‘.’
ADR1	‘0’
ADRO	‘1’
CMD1	‘0’
CMDO	‘3’
Стартовый адрес данных	‘0’
	‘4’
	‘0’
	‘1’
Число данных	‘0’
	‘0’
	‘0’
	‘1’
LRC CHK 1	‘F’
LRC CHK 0	‘6’
END1	CR
END0	LF

01H+03H+04H+01H+00H+01H=0AH,

2-ух дополнений отрицание 0AH - **F6H**.

RTU Режим:

ADR	01H
CMD	03H
Начальный адрес	21H
	02H
Число данных (Индекс слова)	00H
	02H
CRC CHK Low	6FH
CRC CHK High	F7H

CRC (циклическая проверка по избыточности) рассчитанная следующими шагами:

Шаг 1 : Загрузка 16-bit регистра (называемого CRC регистром) с FFFFH;

Шаг 2: Исключающее ИЛИ первому 8-bit байту из командного сообщения с байтом младшего порядка из 16-bit регистра CRC, помещение результата в CRC регистр.

Шаг 3: Сдвиг одного бита регистра CRC вправо с MSB нулевым заполнением. Извлечение и исследование LSB.

Шаг 4: Если LSB CRC регистра равно 0, повторите шаг 3, в противном случае исключайте ИЛИ CRC регистра с полиномиальным значением A001H.

Шаг 5: Повторяйте шаг 3 и 4, до тех пор, пока восемь сдвигов не будут выполнены. Затем, полный 8-bit байт будет обработан.

Шаг 6: Повторите шаг со 2 по 5 для следующих 8-bit байтов из командного сообщения.

Продолжайте пока все байты не будут обработаны. Конечное содержание CRC регистра CRC значение. При передачи значения CRC в сообщении, старшие и младшие байты значения CRC должны меняться, то есть сначала будет передан младший байт.

На следующем примере приведена CRC генерация с использованием языка C. Функция берет два аргумента:

Unsigned char* data ← a pointer to the message buffer
Unsigned char length ← the quantity of bytes in the message buffer

```
The function returns the CRC value as a type of unsigned integer.
Unsigned int crc_chk(unsigned char* data, unsigned char length){
intj;
unsigned int reg cre=0xFFFF;
while(length-){
reg_crc ^= *data++;
for(j=0;j<8;j++){
if(reg_crc & 0x01){ /*LSB(b0)=1 */
reg_crc=(reg_crc>>1) ^ 0xA001;
}else{
reg_crc=reg_crc>>1;
}
}
}
return reg_crc;
}
```

3.5. Адресный список:

Содержание доступных адресов показано ниже :

содержание	адрес	функция	
Параметры ПЧ	ggnnH	gg – группа параметра, nn – параметр. Номер параметра, для примера, адрес 0401H параметра Pr.4-01. См. раздел 5 по функциям каждого параметра. При чтении параметра командным кодом 03H, только один параметр может читаться в одно и тоже время.	
Команда Только запись	2000H	Bit 0-1	00: Никакая функция 01: Stop 10: Run 11: Jog+Run
		Bit 2-3	Не используется
		Bit 4-5	00: Никакая функция 01: FWD 10: REV 11: Изменение направления
		Bit 6-15	Не используется
	2001H	Команда задания частоты	
	2002H	Bit 0	1: EF (внешняя ошибка) on;
		Bit 1	1: Сброс;
		Bit 2-15	Не используется
Монитор состояния. Только чтение	2100H	Код ошибки: 0: Ошибок не было; 1: Превышение тока (oc) 2: Перенапряжение (ov) 3: Перегрев ПЧ (oH) 4: Перегрузка по току (oL)	

содержание	адрес	функция	
		5: Overload 1 (oL1) 6: Внешняя ошибка (EF) 7: Ошибка CPU (cF3) 8: Ошибка аппаратной защиты (HPF) 9: Двухкратное превышение номинального тока при разгоне (ocA) 10: Двухкратное превышение номинального тока при замедлении (ocd) 11: Двухкратное превышение номинального тока в установившемся режиме (осп) 12: Замыкание на землю (GF) 13: Зарезервирован 14: Низкое напряжение (Lv) 15: Ошибка CPU 1 (cF1) 16: CPU failure 2 (cF2)	
Монитор состояния. Только чтение		17: Внешний блок управления 18: Перегрузка (oL2) 19: Авария автоматического разгона/замедления (cFA) 20: Разрешение программной защиты (codE)	
	2101H	Статус VFD	
		Bit 0-1	00: RUN LED не горит, STOP LED горит. 01: RUN LED мигает, STOP LED горит. 10: RUN LED горит, STOP LED мигает. 11: RUN LED горит, STOP LED не горит.
		Bit 2	01: Jog активен.
		Bit 3-4	00: REV LED не горит, FWD LED горит. 01: REV LED мигает, FWD LED горит. 10: REV LED горит, FWD LED мигает. 11: REV LED горит, FRD LED не горит.
		Bit 5-7	не используется
		Bit 8	1: Главная частота, управляемая по интерфейсу.
		Bit 9	1: Главная частота, управляемая по внешним терминалам.
		Bit 10	1: Команда работы управляемая по RS-485.
		Bit 11	1: Параметры заблокированы.
		Bit 12-15	не используется
		2102H	Ведущая частота F (XXX.XX)
		2103H	Выходная фактическая частота H (XXX.XX)
		2104H	Выходной ток фазы (XXX.XX)
		2105H	Напряжение на шине DC U (XXX.XX)
		2106H	выходное напряжение E (XXX.XX)
		2107H	Номер шага пошагового управления скоростью
		2108H	Номер шага PLC программы
		2109H	Время действия PLC
		210AH	Значение счетчика

3.6. Ответ исключение:

Ниже приводятся ситуации в которых преобразователь не дает нормального ответа управляющему устройству, например, компьютеру.

- Преобразователь не принимает сообщения из-за ошибки связи и не отвечает компьютеру. В этом случае компьютер исчерпает лимит ожидания.
- ПЧ принимает сообщение без ошибки, но не может его обработать, ответ исключения возвратится ведущему устройству, и сообщение об ошибке "CExx" будет выведено на цифровой панели преобразователя. "xx" в сообщении "CExx" есть десятичный код равный коду исключения, который описан ниже. В ответе исключения, старший значащий бит первоначального кода команды установлен в 1, и код исключения объясняет условие, которое вызвало исключение.

Пример ответа исключения с кодом команды 06H и кодом исключения 02H:

ASCII режим:	
STX	‘.’
ADR 1	‘0’
ADR 0	‘1’
CMD 1	‘8’
CMD 0	‘6’
Код исключения	‘0’
	‘2’
LRC CHK 1	‘7’
LRC CHK 0	‘7’
END 1	CR
END 0	LF

RTU режим:	
ADR	01H
CMD	86H
Код исключения	02H
CRC CHK Low	C3H
CRC CHK High	A1H

Значение кода исключения:

Код исключения	Значение
1	Код запрещенной команды: Код команды, полученный в командном сообщении, не доступен для понимания ПЧ.
2	Недоступный адрес данных: Адрес данных, полученный в командном сообщении, не доступен для понимания ПЧ.
3	Не допустимое значение данных: Значение данных, полученное в командном сообщении, не доступно для понимания ПЧ.
4	Ошибка в ведомом устройстве: ПЧ не может выполнить требуемое действие.

ПЧ принимает сообщение, но обнаруживает ошибку коммуникации, таким образом, никакого ответа не следует, но на дисплей цифровой панели ПЧ будет выведена код ошибки сообщения "CExx". Компьютер в конце концов исчерпает лимит ожидания ответа. "xx" в сообщении "CExx" есть десятичный код равный коду исключения, который описан ниже.

Сообщение об ошибке	Значение
5	Зарезервированный
6	ПЧ занят: Временной интервал между командами слишком короток. Сохраните интервал 10мс после возвращения из команды. Если ответ на команду не поступает, сохраните интервал 10мс по той же причине.
7 и 8	Зарезервированный
9	Ошибка контрольной суммы. Проверьте правильность контрольной суммы.
10	Сторожевой таймер. Таймер сбрасывается на 0 после приема каждого правильного Modbus сообщения.
11	Ошибка кадра: Проверьте, соответствует ли скорость передачи формату данных.
12	Сообщение команды слишком короткое.
13	Длина сообщения более допустимой.
14	Сообщения команды включают данные, не принадлежащие символам '0' ... '9', 'A' ... 'F' кроме символов старта и конца (только для Modbus режима ASCII).

3.7. Коммуникационная программа PC:

Ниже приведен пример написания программы коммуникации компьютера с ПЧ для Modbus режима ASCII на языке Си.

```
#include<stdio.h>
#include<dos.h>
#include<conio.h>
#include<process.h>

#define PORT 0x03F8 /* the address of COM1 */

/* the address offset value relative to COM1 */
#define THR 0x0000
#define RDR 0x0000
#define BRDL 0x0000
#define IER 0x0001
#define BRDH 0x0001
#define LCR 0x0003
#define MCR 0x0004
#define LSR 0x0005
#define MSR 0x0006

unsigned char rdat[60];
/* read 2 data from address 2102H of AC drive with address 1 */
unsigned char idat[60]={':', '0', '1', '0', '3', '2', '1', '0', '2', '0', '0', '2', 'D', '7', 'r', '\n'};

void main(){
    int i;
    outportb(PORT+MCR,0x08); /* interrupt enable */
    outportb(PORT+IER,0x01); /* interrupt as data in */
    outportb(PORT+LCR,(inportb(PORT+LCR) 10x80));
    /* the BRDL/BRDH can be access as LCR.b7==1 */
```

```
outportb(PORT+BRDL,12);      /* set baudrate=9600,12=115200/9600*/
outportb(PORT+BRDH,0x00);
outportb(PORT+LCR,0x06);     /* set protocol, <7,N,2>=06H
                               <7,E,1>=1AH,  <7,O,1>=0AH
                               <8,N,2>=07H,  <8,E,1>=1BH
                               <8,0,1>=0BH      */

for(i=0;i<=16;i++){
while(!(inportb(PORT+LSR) & 0x20)); /* wait until THR empty */
outportb(PORT*THR.tdat[i]);        /* send data to THR */
}

i=0;
while(!kbhit()){
  if(inportb(PORT+LSR) & 0x01){ /* b0==1. read data ready */
    rdat[i++]=inportb(PORT+RDR); /* read data form RDR */
  }
}
}
```

Конец дополнения к руководству по эксплуатации преобразователей частоты VFD-S.
