ООО «НПП Электромеханика»

Модуль ввода дискретного сигнала МС1202И. Описание протокола обмена данными стандарта МЭК-870-5-1-95 формата FT3

Данное описание применимо к МС1202И с программной версией, начиная с 246.

ОГЛАВЛЕНИЕ

ОБЩИЕ ПРИНЦИПЫ ПЕРЕДАЧИ ДАННЫХ ПО СТАНДАРТУ МЭК-870-5-1-95 ФОРМАТ КАДРА FT3

ПЕРЕДАЧА В СЕТИ

Устройства в сети отвечают на запросы главного контроллера. Байты идут непрерывным потоком. Запрос – ответ. Начало кадра запроса и ответа идентифицируется маркером (двумя специальными байтами). МС1202И начинает отвечать через 2 мс после получения последнего байта запроса.

ФРЕЙМ

Назначение битов: 1 стартовый бит; 8 бит данных, младшим значащим разрядом вперед, паритет отсутствует; 1 стоповый бит.

старт	1	2	3	4	5	6	7	8	стоп

ФОРМАТ КАДРА ЗАПРОСА

Кадр запроса состоит из стартовой последовательности длиной 2 байта, одного блока данных длиной 14 байт и двух байт CRC в конце. CRC рассчитывается для 14 байт, начиная с длины.

Кадр запроса содержит следующие поля:

Наименование	Описание	Размер	Значение
Head	Стартовая последовательность	2 байта	0x05 0x64
DataLen	Длина данных	1 байт	0x00
ControlByte	Контрольный байт	1 байт	0x00
Address	Адрес	2 байта, младший байт передается первым	0x0000– 0xFFFF*
Command	Команда для устройства	1 байт	
Parameters	Параметры команды	9 байт	
CRC	Контрольная сумма	2 байта, старший байт передается первым	

*Address = 0x00FF - широковещательный адрес. При совместном использовании с протоколом MODBUS помнить, что в MODBUS допустимый адрес устройства ограничен значением 0x01 – 0xF7.

См. структуру PKTSEND.

ФОРМАТ КАДРА ОТВЕТА

Кадр ответа состоит из стартовой последовательности длиной 2 байта и одного или нескольких блоков данных.

Кадр ответа с одним блоком данных имеет вид:

Наименование	Описание	Размер	Значение
Head	Стартовая последовательность	2 байта	0x05 0x64
DataLen	Длина данных	1 байт	0x0E
ControlByte	Контрольный байт	1 байт	0x00
Address	Адрес	2 байта, младший байт передается первым	0x0000– 0xFFFF
Data	Данные	10 байт, младший байт передается первым	
CRC	Контрольная сумма	2 байта, старший байт передается первым	

Если число передаваемых данных не более 10 байт, то кадр ответа содержит 1 блок данных, фиксированной длины - 16 байт (из них 4 байта – заголовочная часть, 2 байта - CRC). В поле длины DataLen, независимо от количества байт данных в блоке, передается 14. Содержимое незадействованных байт данных может быть произвольным, CRC считается для всех 14 байт, начиная с поля длины.

Если число передаваемых данных более 10 байт, то кадр ответа содержит несколько блоков данных. Каждый блок данных заканчивается двумя байтами CRC. Первый блок данных также имеет заголовочную часть (4 байта), которая является заголовочной частью для всего кадра (последующие блоки не содержат заголовочной части). В поле длины DataLen указывается количество байт данных в кадре (без стартовой последовательности и CRC).

Длина первого блока всегда 16 байт (с учетом заголовочной части и 2 байт CRC), длина последнего блока определяется количеством байт данных в нем и может находиться в пределах от 3 (1 байт данных, 2 байта CRC) до 16, все промежуточные блоки имеют длину 16 байт (14 байт данных, 2 байта CRC).

Кадр ответа из нескольких блоков содержит следующие поля:

Наименование	Описание	Размер	Значение
Head	Стартовая последовательность	2 байта	0x05 0x64
DataLen	Длина данных в кадре	1 байт	0x0F – 0xFF
ControlByte	Контрольный байт	1 байт	0x00
Address	Адрес	2 байта, младший байт передается первым	0x0000– 0xFFFF
Data	Данные	10 байт, младший байт передается первым	
CRC	Контрольная сумма	2 байта, старший байт передается первым	
Data	Данные	14 байт, младший байт передается первым	
CRC	Контрольная сумма	2 байта, старший байт передается первым	
…	…	…	…
Data	Данные	1-14 байт, младший байт передается первым	
CRC	Контрольная сумма	2 байта, старший байт передается первым	

В поле DataLen указывается длина данных Data плюс 4 байта, учитывающие размер полей DataLen, ControlByte и Address. Длина кадра ответа (исключая поле Head и поля CRC) не должна превышать 255 байт.

См. структуры PKTHEAD, PKTREADHEAD, PKTREADDATA.

СИСТЕМА КОМАНД МОДУЛЯ ДИСКРЕТНОГО ВВОДА МС1202И

СПИСОК КОМАНД МС1202И

Код команды	Наименование
0x01	Подготовка к записи данных во флеш-память устройства
0x02	Изменение адреса устройства
0x03	Чтение адреса устройства
0x08	Получить типизацию устройства
0x15	Установка скорости обмена данными
0x16	Фиксировать данные
0x1B	Синхронизация времени
0x58	Получить байт состояния устройства
0x59	Сбросить байт состояния устройства
0x60	Получить состояние счетчиков импульсов
0x61	Сбросить счетчики импульсов в "0"
0x62	Получить состояние дискретного входа
0x63	Установить интервал фильтрации дребезга контактов
0x64	Получить интервал фильтрации дребезга контактов
0x65	Установить режим чтения входа
0x66	Получить интервал длительности дребезга контактов
0x67	Установить дату и время
0x68	Получить дату и время
0x69	Получить время последнего включения/выключения
0x71	Получить фиксированные данные
0x73	Получить количество записей в журнале
0x74	Получить запись журнала
0x75	Установить маску записи данных в журнал
0x76	Получить маску записи данных в журнал
0xFF	Задать протокол обмена данными

0X01 ПОДГОТОВКА К ЗАПИСИ ДАННЫХ ВО ФЛЕШ-ПАМЯТЬ УСТРОЙСТВА

Параметры	Байты структуры PARAMETRS
Признак команды	P1=0xA5:

Возвращаемые данные: нет

Команда 0x01 «Подготовка к записи данных» является предварительной для любой команды, изменяющей внутренние данные ПЗУ МС1202И.

0X02 ИЗМЕНЕНИЕ АДРЕСА УСТРОЙСТВА

Установить адрес в сети RS-485.

Предварительная команда: 0x01: Подготовка к записи данных во флеш-память устройства

Параметры	Байты структуры **PARAMETRS**
Старый адрес	P1-P2, младший байт передается первым
Новый адрес	P3-P4, младший байт передается первым

Возвращаемые данные: нет

0X03 ЧТЕНИЕ АДРЕСА УСТРОЙСТВА

Параметры: нет

Возвращаемые данные: (поле Data в разделе Формат кадра ответа)

- 0-1 байты Data: адрес; младший байт передается первым.

0X08 ПОЛУЧИТЬ ТИПИЗАЦИЮ УСТРОЙСТВА

Параметры: нет

Возвращаемые данные: структура TUsoType

Значение поля Model передается в шестнадцатеричном виде. Для правильной интерпретации модели устройства необходимо поменять байты местами.

Для устройства МС1202И значение поля Model 0x0212.

0X15 УСТАНОВКА СКОРОСТИ ОБМЕНА ДАННЫМИ

Установить скорость RS-485.

Предварительная команда: 0x01: “Подготовка к записи данных во флеш-память устройства”

Параметры	Байты структуры PARAMETRS
Код скорости	P1

Константа скорости	Скорость RS-485, бит/с
0x01	19200
0x02	9600 при первом включении
0x03	4800
0x04	2400
0x05	1200
0x11	38400
0x12	57600
0x13	115200

Возвращаемые данные: нет

0X16 ФИКСИРОВАТЬ ДАННЫЕ

Параметры	Байты структуры PARAMETRS
Таймерная метка	P1-P4
Параметр фиксации таймерной метки	P5

Если P5=1, в качестве таймерной метки будут взяты время и дата от внутренних часов устройства на момент прихода команды в формате – секунды, прошедшие с 2000 г. Иначе в качестве таймерной метки будет взята четырехбайтовая метка времени, которая в произвольном формате может быть передана для датчика контроллером верхнего уровня.

Возвращаемые данные: нет

0X58 ПОЛУЧИТЬ БАЙТ СОСТОЯНИЯ УСТРОЙСТВА

Параметры	Байты структуры PARAMETRS
Флаг сброса	P1

Возвращаемые данные (поле Data в разделе “Формат кадра ответа”):

- 0 байт Data - байт состояния.

Если флаг сброса равен 1, то байт состояния сбрасывается после передачи (обнуляются все его биты).

Формат байта состояния см. “Байт состояния устройства МС1202И”

0X59 СБРОСИТЬ БАЙТ СОСТОЯНИЯ УСТРОЙСТВА

По этой команде устройство сбрасывает байт состояния.

Параметры: нет

Возвращаемые данные: нет

0X60 ПОЛУЧИТЬ СОСТОЯНИЕ СЧЕТЧИКОВ ИМПУЛЬСОВ

Устройство поддерживает 8 дискретных входов, 4 из которых работают в режиме счетчиков: 1 – с частотой 10 кГц (высокочастотный счетчик) и 3 – с частотой 1 кГц (низкочастотные счетчики импульсов).

Все счетчики инкрементируются переходом сигнала из логической 1 в 0.

Наименьшая длительность импульса (полупериод), при которой изменение высокочастотного счетчика будет зафиксировано, – 50 мкс. Наименьшая длительность импульса, при которой изменение низкочастотного счетчика будет зафиксировано, – 500 мкс.

Наименьшая длительность сигнала (полупериод), при которой изменение счетчика с фильтрацией дребезга контактов будет зафиксировано, - 1 миллисекунда (при интервале фильтрации дребезга контактов = 1).

Погрешность алгоритма фильтрации дребезга контактов не более 0,5 миллисекунды.

Параметры: нет

Возвращаемые данные: структура TCount

В поле Data придут значения счетчиков. Длина данных у этой команды 16. Чтобы обработать эту команду корректно, Ваша функция приема должна поддерживать прием данных длиной больше 14.

0X61 СБРОСИТЬ СЧЕТЧИКИ ИМПУЛЬСОВ В "0"

Параметры	Байты структуры PARAMETRS
Битовая маска сброса счетчиков (TClearCount).	P1

Возвращаемые данные: нет.

0X62 ПОЛУЧИТЬ СОСТОЯНИЕ ДИСКРЕТНОГО ВХОДА

Параметры	Байты структуры PARAMETRS
Если этот байт равен 1, то байт предыдущего изменения входов будет сброшен в 0.	P8
Если этот байт равен 1, то слово состояния устройства будет сброшено в 0.	P9

Возвращаемые данные: структура TUSOInput

В 1 устанавливаются биты, которые изменились со времени последней выборки пользователем. Младшему биту соответствует младший пин входа.

0X63 УСТАНОВИТЬ ИНТЕРВАЛ ФИЛЬТРАЦИИ ДРЕБЕЗГА КОНТАКТОВ

Предварительная команда: 0x01 “Подготовка к записи данных во флеш-память устройства”

Параметры	Байты структуры PARAMETRS
Значение интервала (миллисекунды) для соответствующего 0 - 7 пина входа.	P1-P8

По умолчанию интервал равен 20, он не может быть равен 0. (Если ноль, то сбрасывается в значение по умолчанию).

Возвращаемые данные: нет

Примечание: этот интервал используется в алгоритме фильтрации дребезга контактов. Импульсы длительностью меньше этого интервала считаются дребезгом.

Погрешность измерения не более 0.5 миллисекунд.

0X64 ПОЛУЧИТЬ ИНТЕРВАЛ ФИЛЬТРАЦИИ ДРЕБЕЗГА КОНТАКТОВ

Параметры: нет

Возвращаемые данные: (поле Data в разделе “Формат кадра ответа”):

- 0 - 7 байты Data - значение интервала (миллисекунды) для соответствующего 0 - 7 пина входа.

0X65 УСТАНОВИТЬ РЕЖИМ ЧТЕНИЯ ВХОДА

Предварительная команда: “Подготовка к записи данных во флеш-память устройства”

Параметры	Байты структуры PARAMETRS
Код режима работы чтения входа	P1

Код	Режим работы чтения выхода
1	Чтение с фильтрацией дребезга
0	Прямое чтение

Возвращаемые данные: нет

0X66 ПОЛУЧИТЬ ИНТЕРВАЛ ДЛИТЕЛЬНОСТИ ДРЕБЕЗГА КОНТАКТОВ

Параметры: нет

Возвращаемые данные: структура TAdoTime.

Формула пересчета для AdoTime0-AdoTime7:

$y=x\cdot 0{,}5;$

где x – значение в регистре; y – результат в мс.

Младший бит в байте endcalc соответствует 1 входу и т. д. Установленный бит в байте endcalc говорит о том, что анализ дребезга и подсчет времени дребезга завершены – получен стабильный импульс.

0X67 УСТАНОВИТЬ ДАТУ И ВРЕМЯ

Параметры	Байты структуры PARAMETRS
Количество секунд, прошедших с 2000 года (unsigned long sec2000)	P1-P4

Возвращаемые данные: нет

0X68 ПОЛУЧИТЬ ДАТУ И ВРЕМЯ

Параметры: нет

Возвращаемые данные (поле Data в разделе “Формат кадра ответа”):

- unsigned long sec2000 // Количество секунд, прошедших с 2000 года
- unsigned char ms256 // 1/256 секунды

0X69 ПОЛУЧИТЬ ВРЕМЯ ПОСЛЕДНЕГО ВКЛЮЧЕНИЯ/ВЫКЛЮЧЕНИЯ

Параметры: нет

Возвращаемые данные (поле Data в разделе “Формат кадра ответа”):

- unsigned long onsec2000; //время последнего включения: кол-во секунд, прошедших с 2000 г.
- unsigned char onms256; // время последнего включения: 1/256 секунды
- unsigned long offsec2000; //время последнего выключения: кол-во секунд, прошедших с 2000 г.
- unsigned char offms256; // время последнего выключения: 1/256 секунды

0X71 ПОЛУЧИТЬ ФИКСИРОВАННЫЕ ДАННЫЕ

Параметры: нет

Возвращаемые данные (поле Data в разделе “Формат кадра ответа”):

- unsigned long sec2000; //Временная метка, формат которой зависит от параметра в // команде 0x16 ”Фиксировать данные”
- unsigned long fastcount7; // Высокочастотный счетчик
- unsigned long count6; //
- unsigned long count5; //
- unsigned long count4; //
- unsigned char input; // Состояние входов

Все данные на момент фиксации.

0X73 ПОЛУЧИТЬ КОЛИЧЕСТВО ЗАПИСЕЙ В ЖУРНАЛЕ

Параметры: нет

Возвращаемые данные (поле Data в разделе “Формат кадра ответа”):

- unsigned char count; //Количество записей сделанных в журнале
- unsigned char maxcount; //Максимально возможное количество записей

0X74 ПОЛУЧИТЬ ЗАПИСЬ ЖУРНАЛА

Параметры	Байты структуры PARAMETRS
Номер записи	P1 (0 – последняя запись, 1 – предпоследняя и т. д.)

Возвращаемые данные (поле Data в разделе “Формат кадра ответа”):

- unsigned char input; // Состояние входов, младшему биту соотв. младший пин входа
- unsigned long sec2000;// Количество секунд, прошедших с 2000 года

- unsigned char ms256; // 1/256 секунды

Примечание: при возникновении ситуации, требующей записи в журнал выполняются следующие операции:

- Формирование записи.
- Занесение записи в буфер.
- Перенос записи из буфера в энергонезависимую память устройства.

При попадании записи в буфер инициализируется процесс записи в журнал. Место, занимаемое записью в буфере, освобождается после того, как запись полностью запишется в журнал. Максимальное время занесения одной записи в журнал - 50 мс. Размер буфера 13 записей.

Если буфер заполнен и возникла ситуация, требующая записи в журнал, то эта запись будет потеряна и в слове состояния МС1202И будет выставлен флаг RecMis (запись пропущена).

Приоритет записи в журнал выше, чем чтения. Т. е., если в процессе занесения записи в журнал будет подана команда чтения из журнала, то она выполнится только после окончания записи. Чтение производится значительно быстрее, чем запись.

Если вы хотите прочитать несколько записей из журнала (это осуществляется выполнением в цикле команды «Получить запись журнала»), то настоятельно рекомендуется перед каждым новым циклом чтения выполнять команду 0x73 «Получить количество записей в журнале», т. к. количество записей в журнале может измениться за время выполнения цикла.

При выполнении команды 0x73 «Получить количество записей в журнале» в МС фиксируется количество записей и номер текущей записи.

0X75 УСТАНОВИТЬ МАСКУ ЗАПИСИ ДАННЫХ В ЖУРНАЛ

Параметры	Байты структуры PARAMETRS
Маска записи данных в журнал	P1

Возвращаемые данные: нет

Младший бит соответствует младшему пину входа.

Примечание: запись в журнал производится при изменении состояния на пине входа, для которого установлен соответствующий бит маски.

0X76 ПОЛУЧИТЬ МАСКУ ЗАПИСИ ДАННЫХ В ЖУРНАЛ

Параметры: нет

Возвращаемые данные (поле Data в разделе “Формат кадра ответа”):

- unsigned char mask; // Маска записи данных в журнал

0X1B СИНХРОНИЗАЦИЯ ВРЕМЕНИ

Параметры: нет

Возвращаемые данные: нет

По этой команде сбрасываются счетчики секунд и миллисекунд. Если в момент прихода команды значение счетчика секунд меньше 30, то происходит просто очищение, если больше или равно, то счетчик секунд очищается, а счетчик минут увеличивается на 1.

0XFF ЗАДАТЬ ПРОТОКОЛ ОБМЕНА ДАННЫМИ

Параметры	Байты структуры PARAMETRS
Константа протокола	P1
Контрольный байт	P2=0x22
Контрольный байт	P3=0xBA
Контрольный байт	P4=0x1E
Контрольный байт	P5=0x45

Предварительная команда: 0x01 Подготовка к записи данных во флеш-память устройства

Константы протоколов:

Константа протокола	Протокол
0x01	ГОСТ Р МЭК-870-5-1-95 формат FT3
0x02	MODBUS RTU

Возвращаемые данные: нет

Команда 0xFF “Задать протокол обмена данными” устанавливает канал связи в режим работы по протоколу обмена данными согласно приведенной выше таблице.

ПРИЛОЖЕНИЕ А. СТРУКТУРЫ ДАННЫХ

PARAMETRS

```
//Параметры пакета передачи
typedef struct _PARAMETRS
{
 unsigned  char  P1;     //Параметр N1
 unsigned  char  P2;     //Параметр N2
 unsigned  char  P3;     //Параметр N3
 unsigned  char  P4;     //Параметр N4
 unsigned  char  P5;     //Параметр N5
 unsigned  char  P6;     //Параметр N6
 unsigned  char  P7;     //Параметр N7
 unsigned  char  P8;     //Параметр N8
 unsigned  char  P9;     //Параметр N9
} PARAMETRS;
```

PKTHEAD

```
//Заголовок пакета.
typedef struct _PKTHEAD
{
 unsigned  char   HeadByte1;  //Сигнатура заголовка: байт N1 = 0x05
 unsigned  char   HeadByte2;  //Сигнатура заголовка: байт N2 = 0x64
}PKTHEAD;
```

PKTREADHEAD

```
//Стартовый пакет приема
typedef struct _PKTREADHEAD
{
 unsigned  char   DataLen;          //Длина данных
 unsigned  char   ControlByte;      //Контрольный байт
 unsigned  short  Address;          //Адрес устройства
 unsigned  char   Data[10];         //Данные
 unsigned  short  CRC;              //Контрольная сумма
} PKTREADHEAD;
```

PKTREADDATA

```
//Пакет приема данных
typedef struct _PKTREADDATA
{
 unsigned  char   Data[14];   //Данные
 unsigned  short  CRC;        //Контрольная сумма
} PKTREADDATA;
//Примечание: длина поля Data в зависимости от размера кадра может варьироваться
от 1 до 14.
```

TUSOTYPE

```
typedef struct {
    unsigned  short  Model;         // Модель устройства
    unsigned  char  HardwareVersion;// Аппаратная версия
    unsigned  char  SoftwareVersion;// Программная версия
    unsigned  char  Reserv4;        // Не используется
    unsigned  char  Reserv5;        // Не используется
    unsigned  char  Reserv6;        // Не используется
    unsigned  char  SerialNumberHigh;// Серийный номер – старший байт
    unsigned  short  SerialNumber;  // Серийный номер
}TUsoType;
```

БАЙТ СОСТОЯНИЯ УСТРОЙСТВ МС1202И

```
//Формат байта состояния
typedef  struct{
 unsigned char SupOff:1;      // Отключение питания
 unsigned char ErrFlash :1;   // Ошибка: флеш-память не отвечает
 unsigned char ErrFlashCRC:1; // Ошибка CRC флеш
 unsigned char ErrPocketCRC:1;// Ошибка CRC пакета RS-485
 unsigned char ErrFrame :1;   // Ошибка канала RS-485
 unsigned char ErrOverflow:1; // МС не успевает считать данные из канала RS-485
 unsigned char RecMis:1;      // МС не успевает производить запись в журнал
 unsigned char ProcReset:1;   // Сброс процессора
}TUso1202State;
```

PKTSEND

```
//Пакет для передачи
typedef struct _PKTSEND
{
 PKTHEAD           Head;      //Заголовок пакета
 unsigned  char    DataLen;   //Длина данных
 unsigned  char    ControlByte;//Контрольный байт
 unsigned  short   Address;   //Адрес устройства
 unsigned  char    Command;   //Команда для устройства
 PARAMETRS         P1P9;      //Параметры
 unsigned  short   CRC;       //Контрольная сумма
} PKTSEND;
```

TCOUNT

```
typedef struct {
  unsigned  long FastCount7; //Высокочастотный счетчик
  unsigned  long Count6;
  unsigned  long Count5;
  unsigned  long Count4;
}TCount;
```

TCLEARCOUNT

```
typedef struct {
   unsigned  char  ClearCount7  :1;
   unsigned  char  ClearCount6  :1;
```

```
   unsigned  char  ClearCount5  :1;
   unsigned  char  ClearCount4  :1;
}TClearCount;
```

TUSOINPUT

```
typedef   struct{
   unsigned char  Input;                  //Состояние дискретного входа
   unsigned char  CurrentInputPinModify;  //Текущий байт изменений входов
   unsigned char  WorkMode;               /* Текущий режим чтения входов(1 – с
                                             фильтрацией дребезга, 0 – прямой*/
   unsigned char  PreviousInputPinModify;//Предыдущий байт изменений входов
   unsigned char  Dirt[4];
   unsigned char  USOState;               // Байт состояния устройства МС1202И
 }TUSOInput;
```

TADOTIME

```
typedef struct {
unsigned  short  AdoTime0;
unsigned  short  AdoTime1;
unsigned  short  AdoTime2;
unsigned  short  AdoTime3;
unsigned  short  AdoTime4;
unsigned  short  AdoTime5;
unsigned  short  AdoTime6;
unsigned  short  AdoTime7;
unsigned  char    endcalc;
}TAdoTime;
```

ПРИЛОЖЕНИЕ Б. ПРИМЕР ПРОГРАММЫ РАСЧЕТА CRC

```
const unsigned short crctable_ft3[256] = {
0x0000, 0x9EB3, 0xA3D5, 0x3D66, 0xD919, 0x47AA, 0x7ACC, 0xE47F,
0x2C81, 0xB232, 0x8F54, 0x11E7, 0xF598, 0x6B2B, 0x564D, 0xC8FE,
0x5902, 0xC7B1, 0xFAD7, 0x6464, 0x801B, 0x1EA8, 0x23CE, 0xBD7D,
0x7583, 0xEB30, 0xD656, 0x48E5, 0xAC9A, 0x3229, 0x0F4F, 0x91FC,
0xB204, 0x2CB7, 0x11D1, 0x8F62, 0x6B1D, 0xF5AE, 0xC8C8, 0x567B,
0x9E85, 0x0036, 0x3D50, 0xA3E3, 0x479C, 0xD92F, 0xE449, 0x7AFA,
0xEB06, 0x75B5, 0x48D3, 0xD660, 0x321F, 0xACAC, 0x91CA, 0x0F79,
0xC787, 0x5934, 0x6452, 0xFAE1, 0x1E9E, 0x802D, 0xBD4B, 0x23F8,
0xFABB, 0x6408, 0x596E, 0xC7DD, 0x23A2, 0xBD11, 0x8077, 0x1EC4,
0xD63A, 0x4889, 0x75EF, 0xEB5C, 0x0F23, 0x9190, 0xACF6, 0x3245,
0xA3B9, 0x3D0A, 0x006C, 0x9EDF, 0x7AA0, 0xE413, 0xD975, 0x47C6,
0x8F38, 0x118B, 0x2CED, 0xB25E, 0x5621, 0xC892, 0xF5F4, 0x6B47,
0x48BF, 0xD60C, 0xEB6A, 0x75D9, 0x91A6, 0x0F15, 0x3273, 0xACC0,
0x643E, 0xFA8D, 0xC7EB, 0x5958, 0xBD27, 0x2394, 0x1EF2, 0x8041,
0x11BD, 0x8F0E, 0xB268, 0x2CDB, 0xC8A4, 0x5617, 0x6B71, 0xF5C2,
0x3D3C, 0xA38F, 0x9EE9, 0x005A, 0xE425, 0x7A96, 0x47F0, 0xD943,
0x6BC5, 0xF576, 0xC810, 0x56A3, 0xB2DC, 0x2C6F, 0x1109, 0x8FBA,
0x4744, 0xD9F7, 0xE491, 0x7A22, 0x9E5D, 0x00EE, 0x3D88, 0xA33B,
0x32C7, 0xAC74, 0x9112, 0x0FA1, 0xEBDE, 0x756D, 0x480B, 0xD6B8,
0x1E46, 0x80F5, 0xBD93, 0x2320, 0xC75F, 0x59EC, 0x648A, 0xFA39,
0xD9C1, 0x4772, 0x7A14, 0xE4A7, 0x00D8, 0x9E6B, 0xA30D, 0x3DBE,
0xF540, 0x6BF3, 0x5695, 0xC826, 0x2C59, 0xB2EA, 0x8F8C, 0x113F,
0x80C3, 0x1E70, 0x2316, 0xBDA5, 0x59DA, 0xC769, 0xFA0F, 0x64BC,
0xAC42, 0x32F1, 0x0F97, 0x9124, 0x755B, 0xEBE8, 0xD68E, 0x483D,
0x917E, 0x0FCD, 0x32AB, 0xAC18, 0x4867, 0xD6D4, 0xEBB2, 0x7501,
0xBDFF, 0x234C, 0x1E2A, 0x8099, 0x64E6, 0xFA55, 0xC733, 0x5980,
0xC87C, 0x56CF, 0x6BA9, 0xF51A, 0x1165, 0x8FD6, 0xB2B0, 0x2C03,
0xE4FD, 0x7A4E, 0x4728, 0xD99B, 0x3DE4, 0xA357, 0x9E31, 0x0082,
0x237A, 0xBDC9, 0x80AF, 0x1E1C, 0xFA63, 0x64D0, 0x59B6, 0xC705,
0x0FFB, 0x9148, 0xAC2E, 0x329D, 0xD6E2, 0x4851, 0x7537, 0xEB84,
0x7A78, 0xE4CB, 0xD9AD, 0x471E, 0xA361, 0x3DD2, 0x00B4, 0x9E07,
0x56F9, 0xC84A, 0xF52C, 0x6B9F, 0x8FE0, 0x1153, 0x2C35, 0xB286};
```

```
unsigned short crc_ft3(unsigned char *Data, unsigned char DataLen)
{
  unsigned short crc = 0;
  unsigned char uIndex;
  while (DataLen--)
  {
    uIndex= ((crc>>8) ^ *Data++);
    crc<<=8;
    crc ^= crctable_ft3[uIndex];
  }
return (crc>>8)|(crc<<8);
}
```