

METTLER TOLEDO

Торговые весы Тайгер-П

Система команд пересылки данных



METTLER TOLEDO

© METTLER-TOLEDO Inc.

Содержание

1. Подготовка конфигурационных файлов	3
2. Загрузка данных в весы	4
2.1 Загрузка PLU	4
2.2 Удаление PLU	6
2.3 Загрузка названия фирмы	6
2.4 Загрузка рекламного текста	6
2.5 Загрузка текста ингредиентов	7
2.6 Загрузка формата свободно программируемой этикетки	7
2.7 Другие часто используемые команды загрузки	8
3. Чтение PLU	9
4. Получение отчетов с весов	9
5. Статус весов	11
6. Протокол передачи данных	12

1. Подготовка конфигурационных файлов.

Для передачи информации на весы Тайгер-П используются две функции из динамической библиотеки TransferEth.dll, поставляемой вместе с программой управления весами SPCT.

Одна из этих функций с вызовом `ireturn=Transfer_Ethernet("Transscale.ini")` используется для загрузки PLU на весы из текстового файла, а другая с вызовом: `ireturn =Transfer_Ethernet_EX("Transscale.ini",251245664)`

позволяет загружать на весы и получать с весов любую допустимую информацию в так называемом "TxtCommand" формате, включая пересылку PLU в бинарном формате и формате со сжатием.

Указанные две функции, составляющие драйвер Ethernet Тайгер-П в процессе работы обращаются к файлу SCALEADDRESS.INI со списком IP адресов подключенных весов.

Этот файл выглядит следующим образом:

```
[CONFIG]
MEDIA=1
COMPORT=2
THREADNUM=4
[2]
NAME=
IP=172.21.108.120
PORT=3001
[3]
NAME= // Номер прибора
IP=172.21.108.121 // Название весов вводить не обязательно
PORT=3001 // IP адрес
// Номер порта всегда один и тот же =3001
[5]
NAME=
IP=172.21.108.126
PORT=3001
```

и может быть сформирован как вручную, так и с использованием программы SPCT.

Следующий конфигурационный файл "Transscale.ini" используется для указания файла, в котором находится пересылаемая команда (или несколько команд) и номеров весов – получателей команды. В варианте "SingleTrfOut" он выглядит следующим образом:

```
trf.out //Файл пересылаемой команды
2
3 //Номер весов
5
```

и также формируется в процессе работы программы SPCT. Для пересылки на разные весы различных наборов PLU возможно использование варианта рассылки "MultiTrfOut", при котором файл "Transscale.ini" имеет вид:

```
2: trf.out
3: trf1.out
5: trf2.out
```

Имя и расширение файла, в котором содержится строка команды может быть любым – использование в качестве примера "trf.out" обусловлено тем, что в одноименный файл помещается последняя по времени исполняемая команда в процессе работы SPCT. Она

(команда), сформированная с помощью SPCT, может быть использована как для непосредственной пересылки информации на весы с помощью драйвера, так и в качестве шаблона для формирования команды пользователя, например путем редактирования информации в текстовых полях. При этом длина поля должна оставаться неизменной, а символы вводиться с использованием DOS кодировки.

Исключение составляет команда 260 – формата этикетки, которая формируется программой "Visedit.exe" и помещается в файл "trf.out", а также команда пересылки PLU в сжатом формате, имеющая расширение .lz - последняя формируется с помощью "CompressLZ77.dll".

В процессе работы SPCT формируется множество файлов, в которых протоколируются выполняемые действия. В частности, при необходимости получения информации о результатах прохождения команды, заголовках передаваемых пакетов и т.п. могут быть использованы файлы "trfNr_scale.in.log" и "log.hex".

2. Загрузка данных в весы.

1) Загрузка PLU

А) в формате "txt"

Наиболее удобный способ загрузки PLU – с использованием функции `ireturn=Transfer_Ethernet("Transscale.ini")` и текстового формата хранения PLU.

Текстовый формат PLU для однострочного варианта прошивки выглядит следующим образом :

PLU No, ArticleNo, GroupNo, Unit Price, Tare, ExtraTxtNumber, TaxRate, SellByDateOffset, BestByDateOffset, FixWeight, pricemethod(weight=0/count=1), priceoverwrite(yes=1/no=0), discount(yes=1/no=0), PLU Name

(пример: 1, 33, 1, 123.45, 0, 0,0, 0, 0, 0,0,0,0,АБВГДЕ)

а для двухстрочного, соответственно:

PLU No, ArticleNo, GroupNo, Unit Price, Tare, ExtraTxtNumber, TaxRate, SellByDateOffset, BestByDateOffset, FixWeight, pricemethod(weight=0/count=1), priceoverwrite(yes=1/no=0), discount(yes=1/no=0), PLU Name1, PLU Name2

(пример: 1, 33, 1, 123.45, 0, 0,0, 0, 0, 0,0,0,0,АБВГДЕ, ДЕЖЗ)

Помимо "ручного" (имеется в виду из пользовательской программы) формирования записей PLU, возможна конвертация в рассматриваемый формат *.txt из внутреннего формата хранения PLU в SPCT (файлы *.plu) . Для этого может быть использована, входящая в состав SPCT программа "convert.exe" в режиме PLU->TXT. При наличии в пользовательской системе выхода информации в формате EXCEL возможно двухшаговое преобразование EXCEL->PLU, PLU->TXT и использование функции Transfer_Ethernet для пересылки данных в весы.

В процессе работы рассматриваемой функции передаются все записи PLU из файла *.txt. При необходимости загрузки одного PLU файл должен содержать одну соответствующую запись.

Как дополнительный результат работы драйвера (после выполнения PLUFmtConvert.dll и CompressLZ77.dll) формируются файлы &&trf!.out и &&trf!.lz PLU в "бинарном" и сжатом формате.

Б) в бинарном формате

Загрузка PLU в весы осуществляется функцией
`ireturn =Transfer_Ethernet_EX("Transscale.ini",251245664)`
из предварительно сформированного при помощи PLUFmtConvert.dll файла `&&trf!.out`.

В) в компрессированном формате

Загрузка файла `&&trf!.lz` осуществляется функцией
`ireturn =Transfer_Ethernet_EX("Transscale.ini",251245664)`.
Формирование `&&trf!.lz` осуществляется на основе `&&trf!.out` путем использования
функции из динамической библиотеки `CompressLZ77.dll`.

Г) в "TxtCommand" формате

Этот формат передачи является наиболее универсальным способом обмена РС с
весами и пригоден для передачи любых команд, сохраняемых в файлах `"trf.out"` и
`"trf.txt"`. Более того, система команд для весов Тайгер-П аналогична (с небольшими
отличиями) системе команд `TransL2` для торговых весов Мира и `L2` производства
Меттлер Толодо. Для передачи этих команд в весы Тайгер-П также используется вызов
функции `Transfer_Ethernet_EX("Transscale.ini",251245664)`.

Формат команды 207 передачи PLU для однострочной версии ПО выглядит следующим
образом:

`CMDHEADER"L06C13C28C01L08U01U02S04L11S04F04S03S03S03"`,

где заголовок `CMDHEADER` выглядит одинаково для всех команд:

`"U01S05S04S04U02"`

U01 = 0 - Передача , 1 or 3 - Отклик

S05 - Код команды (00207, ...)

S04 - Управл. поле(0000 - Write, 0003-RDGE)

S04 = 0001 (Номер отдела)

U02 = Номер весов (в ПО весов не используется)

Аббревиатуры U, S, C, B, L, F описания полей означают, соответственно, unsigned byte,
short int, char, byte, long, flag и используются в основном в целях информирования о
формате хранения данных в ПО весов. Двухзначное число после описания поля означает
количество знаков в данном поле при символьном (char) представлении команды.
Комбинация Rnn используется для сокращенной записи команды и означает повтор
следующего поля nn раз.

Назначение полей команды 207:

L06 - PLU No.

C13 - Article No.

C28 - PLU Name

C01 - ''

L08 - Unit Price

U01 - Tax Rate

U02 - Tare

S04 - nothing (0000)

L11 - Fix Weight

S04 - GroupNo

F04 - pricemethod:1;//0 bit

priceoverwrite:1;//1 bit

Пример использования:
000220000000000005Благодарим за покупку!

5) Загрузка текста ингредиентов

Выполняется с помощью функции
[Transfer_Ethernet_EX\("Transscale.ini",251245664\)](#)
командой 209, имеющей формат
CMDHEADER"S03C200",
где S03 – номер текста ингредиентов (1...999);
(Для макс. кол-ва ингредиентов от 1000 до 9999 используется формат S04 !!!)
C200 – текстовое поле ингредиентов.

Пример использования:
0002090000000100001Состав соль, сахар, крахмал, свинина,
0002090000000100002Условия хранения:

6) Загрузка формата свободно программируемой этикетки

Выполняется с помощью функции
[Transfer_Ethernet_EX\("Transscale.ini",251245664\)](#)
командой 260, имеющей формат
CMDHEADER"U02U02R02S03U01" //Label information
"R05U01R05S03R05S03R05U01B01R05S03R05S03R05U01" //Graf.objects
"U01U02U01R08S03" //Article / Extra Text
"U01B01R04S03" //Store Name
"U01B01R04S03" //Advertise Text /Special offer text
"R12U02R12U01R12S03R12S03R12C20" //special text
"R03U02R03U01R03S03R03S03" //date
"R03U01B01R12S03" //date text
"R03U01B01R12S03" //date cross
"U02B01R02S03U01B01R02S03" //EAN Code
"U02U01U02B01R02S03" //Weight
"U02U01R02S03C05" //Weight Unit
"U02U01U02R02S03" //Price
"U02U01R02S03C08" //Price Unit
"U02U01U02B01R02S03" //Amount
"U02U01R02S03C05" //Amount Unit
"U02U01U02R02S03" //FixWeight
"U02U01R02S03C08" //Nutrition value unit
"U02U01U02B01R02S03" //Plu
"U02U01U02B01R02S03" //Device number
"U02U01U02B01R02S03" //Ticket number
"R12U02R12U01R24S03R12C02" //Speical value

Пример использования получен на основе загрузки этикетки 60x40 prepack из
имеющейся библиотеки "labels.lib":

0002600000000100010046432001122001301431015200006512806806600033330434432
310152000065128059059000000001000141031030190000000000000074261046281000000

3. Чтение PLU.

Процедура чтения всех PLU реализуется передачей в весы команды
0002070003000100000001

(формат CMDHEADER"L06C23" , где L06-поле номера PLU
C23 – поле из 23-х ' ')

с помощью функции

[Transfer_Ethernet_EX\("Transscale.ini",251245664\)](#)

Информация с весов помещается в бинарном формате в файл trf"Nr.Scale".in.

Поскольку структура полей команды 207 известна, чтение записи, относящейся к конкретному PLU не является серьезной проблемой. Однако, задачу чтения одного определенного PLU можно упростить, если в поле L06 команды чтения задать номер интересующего нас PLU. В соответствии со значением управляющего поля 0003=ReadGreaterEqual, записи во входном файле будут начинаться с заданного номера PLU и продолжаться до максимального, имеющегося в весах. В этом случае достаточно ограничиться чтением первой записи из файла с расширением *.in.

Для весов Тайгер-П серии 3300 отличия в процедуре чтения PLU заключаются лишь в том, что значением управляющего поля может быть 0007=ReadAll, допускающее чтение только всех PLU.

4. Получение отчетов с весов

Чтение отчетов реализуется передачей в весы команды

000909000200010100000001999999

(формат CMDHEADER"U02L06L06" , где U02-поле выбора вида отчета,
L06 L06– значения начального и конечного номера PLU)

с помощью функции

[Transfer_Ethernet_EX\("Transscale.ini",251245664\)](#)

Информация с весов размещается в бинарном формате в файле trf"Nr_Scale".in.
Управляющее поле U02 может принимать 4 значения: 00, 01, 02, 99.

В случае выбора 00 формируется отчет по PLU, причем информация размещается в файле в соответствии с форматом:

```

FileHeader;//40 bytes
BYTE STX;// '02'
WORD command;//2 bytes '8D' '03'
BYTE control1;// '00'
BYTE control2;// '00'
    struct PLUREPORT{
        long int plu_nr;//4 bytes
        char ArtNr[14];//14 bytes
        char Name[28];//28 bytes
        long int amount;//4 bytes
        long int weight;//4 bytes
        long int count;//4 bytes
    }PluReport[5];//290 bytes
BYTE crc1;
BYTE crc2;
BYTE ACK;// '06'

```

Наибольшее число PLU в каждом отчете не превышает 5, поэтому для получения полного отчета требуется многократная посылка команды 909 с последовательно возрастающими значениями стартового номера PLU (новый стартовый номер PLU должен на единицу превышать последний номер PLU из предыдущего отчета). Конечный номер PLU команды 909 в рассматриваемом виде отчета особого значения не имеет.

При задании управляющего параметра, равным 01, реализуется итоговый отчет по весам в формате:

```

FileHeader;//40 bytes
BYTE STX;// '02'
WORD command;//2 bytes '8D' '03'
BYTE control1;// '01'
BYTE control2;// '00'
    long int amount;//4 bytes
    long int weight;//4 bytes
    long int count;//4 bytes
BYTE crc1;
BYTE crc2;
BYTE ACK;// '06'

```

Полученные значения суммы, суммарного веса и кол-ва продаж в общем случае могут не совпадать с результатами суммирования значений, полученных в предыдущем отчете. Вызвано это тем, что итоговый отчет содержит информацию о транзакциях, реализованных также и без выбора какого-либо значения PLU. Начальное и конечное значения PLU команды 909 при формировании этого отчета не используются.

Выбор параметра 02 дает возможность получения информации об изменениях цен товаров. Формат представления информации следующий:

```

FileHeader;//40 bytes
BYTE STX;// '02'
WORD command;//2 bytes '8D' '03'
BYTE control1;// '02'
BYTE control2;// '00'
struct PRICE_CHANGE_REP{
    long int plu_nr;//4 bytes
    long int date;//4 bytes
    int time;//2 bytes
    long int oldprice;//4 bytes
    long int newprice;//4 bytes
}PRICE_CHANGE_rep[5];//90 bytes
BYTE crc1;
BYTE crc2;
BYTE ACK;// '06'

```

Также как и в первом отчете при однократном использовании команды 909, максимальное кол-во записей равно 5. Полный отчет требует повторного запуска команды с изменяемыми значениями начального и конечного PLU. Максимально возможное кол-во записей этого отчета равно 800. Необходимо отметить, что в отчет попадают только PLU с непосредственным изменением цены, а не те, в которых цена изменилась из-за скидки или спец.предложения.

Параметр 99 обеспечивает удаление отчетов в памяти весов. При этом управляющее поле в CMDHEADER вместо 0002 (READ) должно принимать значение 0000 (WRITE).

5. Статус весов

Чтение текущего статуса весов

```

00020400030000000000
( формат команды CMDHEADER"S03" )

```

может осуществляться с помощью вызова функции [Transfer_Ethernet_EX\("Transscale.ini",251245664\)](#)

Рассматриваемая команда на самом деле считывает список операторов из весов в файл trf"Nr.Sc.".in. При отсутствии записей в списке продавцов в файле тем не менее имеется служебная информация. Алгоритм определения статуса (Вкл/Выкл) состоит в циклическом запуске команды 204 и анализе содержимого файла отклика trf"Nr.Sc.".in.

6. Протокол передачи данных

Драйвер от производителя весов в виде набора dll файлов, входящих в состав SPCT (TransferEth.dll, PLUFmtConvert.dll, ArcnetTxt_Bin.dll, CompressLZ77.dll, CrcModule.dll), за предшествующие годы эксплуатации подтвердил высокую скорость и стабильность передачи данных в “ многопользовательском “ (несколько весов в списке рассылки) и “многозадачном” (различные файлы команд) режимах.

Тем не менее, некоторые пользователи предпочитают писать собственный драйвер, как правило, с целью передачи только команд загрузки PLU (команда 207) и ингредиентов (команда 209). “Открытый” вход для передачи любых из рассмотренных выше команд весов Тайгер-П обычно не реализуется из-за отсутствия синтаксического контроля команд.

Задача написания собственного драйвера для весов Тайгер-П не представляется очень сложной в силу того, что при пересылке любой информации с помощью программы SPCT в одном из файлов trf”Nr_scale”.in.log (передача по Ethernet) или log.hex (передача по COM) сохраняется протокол (**последовательность байт**) последней переданной команды. Эта подсказка оказывает существенную помощь при написании и отладке драйвера. Передаваемая последовательность в файле структурирована путем разбиения информации на 4 строки, при этом во второй и третьей строке содержатся описанные в предыдущих разделах заголовки и тело команды, соответственно, а первая и последняя строка содержат заголовок посылки и контрольную сумму. Для изучения удобнее использовать файл протокола передачи по Ethernet trf”Nr_scale”.in.log (где ”Nr_scale” – номер весов), поскольку в начале этого файла записан заголовок и тело команды в привычном символьном виде.

Первая строка (заголовок посылки) содержит семь байт и описывается структурой вида:

```
struct packetheader
{
    unsigned char start;           //start flag of one packet always 0x2
    unsigned short totallength;    //totallength = sizeof(cmdheader) + pagenumber * pagelength
    unsigned short pagenumber;    //the number of pages of itemdatas in this packet
    unsigned short pagelength;    //the length of each page of itemdatas
}
```

Передача всегда начинается с посылки стартового байта “02”, далее идут 2 байта общей длины посылки с учетом заголовка команды, потом кол-во страниц (фактически кол-во пересылаемых последовательно команд) и два байта длины тела команды.

Во второй строке содержатся 8 байт неоднократно описанного ранее заголовка CMDHEADER команды:

```
struct cmdheader
{
    unsigned char cmdrsp;         //command or response
    unsigned short command;      //command id such as 207,213 and so on
    unsigned short control;      //command type: RD(read) or WR(write) and so on
    unsigned short departno;     //department number always 1
    unsigned char deviceno;      //scale number
}
```

В третьей строке записано тело команды (различные типы команд рассмотрены в предыдущих разделах) с конвертацией символьного формата команды в байтовый на основе учета требуемого кол-ва байт для представления разных типов полей:

C	-	1 байт; //char or unsigned char, 1 byte for each char
U	-	1 байт; //BYTE //U01...U03
B	-	1 байт; //BYTE //B01
F04	-	2 байта; //4 Hex flags, each 0...0Fh
S	-	2 байта; //short int, int, unsigned int or WORD // S01...S05
L	-	4 байта; //long int or unsigned long int // L06, L08, L11.

Таким образом, при пересылке данных под поле вида B01 или Uxx отводится 1 байт, поля F04 и Sxx кодируются двумя байтами, а каждое поле типа Lxx передается с помощью четырех байт. При этом в случае многобайтового представления первым передается младший байт, а последним – старший. Описанное преобразование используется не только в теле команды, но и в заголовках команды и пакета. Что касается текстовых полей вида Cnn, то каждый символ представлен в DOS – кодировке (page 866) и передача текстовой строки занимает nn байт.

Четвертая (последняя) строка содержит два байта контрольной суммы, вычисляемой путем вызова функции:

`unsigned short int Crc = CalcCRC16 (char buf+1, unsigned int len-1),`

из динамической библиотеки, находящейся в файле CrcModule.dll.

В буфер "buf" записывается вся сформированная посылка длиной len, но в формировании контрольной суммы не участвует первый "02" стартовый байт (buf + 1, len - 1).

Полученное двухбайтовое целое Crc добавляется к посылке в следующем порядке: сначала старший байт, затем – младший.

METTLER TOLEDO Scales & Systems
1900 Polaris Parkway Columbus, Ohio 43240

METTLER TOLEDO® - зарегистрированный товарный знак Mettler-Toledo, Inc.
©2000 Mettler-Toledo, Inc.