

Описание предназначено для работы с весами, имеющими версию прошивки 3.63.

Запросы и ответы представлены в 16-ти разрядном дампе в виде байт и ASCII кодировке (при допустимости). В ответе желательно контролировать первые 12 байт – они выделены курсивом и должны однозначно соответствовать запросу. В ответе, по смещению 10 содержится информация о факте успешного выполнения команды, в зависимости от типа команды - 45 58 == EX (успешное выполнение) или 52 44 == RD (успешное чтение).

Получение информации от весов.

- **Чтение информации о весах.**

На устройство отправляется запрос:

43 4D 44 3E 06 00 00 12 20 00 52 44 5A 8E | CMD>....RDZ...

Ответ:

43 4D 44 3C 26 00 00 12 20 00 52 44 C2 CB D2 DD | CMD<&...RDВЛТЭ
00 00 00 00 00 00 00 00 00 00 00 00 31 35 30 00 |150.
00 00 00 00 00 00 00 00 00 00 00 00 46 2E |F...

В ответе возвращаются тип – ВЛТЭ и номинальная масса 150.

- **Чтение номера и модификации весов.**

На устройство отправляется запрос:

43 4D 44 3E 06 00 20 12 20 00 52 44 5D EE 00 00 | CMD>...RD].

Ответ:

43 4D 44 3C 26 00 20 12 20 00 52 44 4C 30 30 31 | CMD<&...RDL001
30 30 30 00 00 00 00 00 00 00 00 00 2D 33 37 37 | 000.....377
2D 81 00 00 00 00 00 00 00 00 00 5D 7E |/~

В ответе возвращаются тип – № L001000 Модификация 377.

- **Чтение текущих данных.**

На устройство отправляется запрос:

43 4D 44 3E 06 00 00 14 20 00 52 44 D2 8E | CMD>....RD..

Ответ:

43 4D 44 3C 26 00 00 14 20 00 52 44 00 35 7E A1 | CMD<&...RD.5~. |
94 17 7D A1 B6 D8 7D A1 2A C6 73 90 9C 86 01 00 | ..}...}*s..... |
00 00 00 00 03 00 DF 19 97 19 00 00 4D 75 |Mu |

В ответе возвращаются данные, содержащие текущий вес в выделенном подчеркиванием поле:

011869C == 99996

Месторасположение десятичной точки определяется выбранными единицами и диапазоном изменения.

- **Чтение текущего состояние весов.**

На устройство отправляется запрос:

43 4d 44 3e 06 00 20 14 20 00 52 44 d5 ee 00 00 |CMD>.. .RDXo..

Ответ:

43 4d 44 3c 26 00 20 14 20 00 52 44 00 00 00 00 | CMD<&. .RD....
b1 52 7a 90 6d 67 7a 90 03 81 00 08 00 00 00 00 |±Rzĥmgzĥ.Ġ.....
00 00 00 00 00 00 00 00 00 00 00 00 00 7d e3 |.....}Г

В ответ возвращаются текущее состояние весов в выделенном подчеркиванием поле:

03 81 00 08 == 0x08008103, где биты имеют следующее значение:

- d00 знак выводимой величины, 1 == отрицательные значения
- d01 признак включения, 1 == выключено
- d06 1 == запрос калибровки
- d08 1 == стабильные показания
- d09 1 == захват нуля
- d10 1 == вывод альтернативного значения
- d11 1 == достигнут максимальный вес
- d12 1 == достигнут минимальный вес
- d15 1 == выводимая величина *10
- d21 1 == режим калибровка (при 1)
- d22 диапазон 0/1

00 00 00 00 == 0x00000000, режим работы весов:

- 0 - взвешивание
- 1 - подсчет числа
- 2 - процентное взвешивание
- 3 - рецептурное взвешивание
- 4 - взвешивание нестабильных образцов

Управление весами.

- **Включение / выключение весов.**

На устройство отправляется запрос:

43 4d 44 3e 0e 00 48 14 08 00 57 52 02 00 00 00 00 00 00 00 ba ca

Ответ:

43 4d 44 3c 06 00 48 14 00 00 45 58 c1 9f

- **Перезагрузка весов.**

На устройство отправляется запрос:

43 4d 44 3e 0a 00 f8 a1 04 00 57 52 52 45 53 54
b1 e2

|CMD>..шЎ..WRREST
|±В.....

Ответ:

43 4d 44 3c 06 00 f8 a1 00 00 **45 58** 56 75 01

|CMD<..шЎ..EXVu.

Выполнение команды приводит к «зависанию» весов на время 30...60 секунд, с последующей перезагрузкой и переходом в состояние выключено.

- **Сброс ограничения калибровок.**

На устройство отправляется запрос:

00 43 4d 44 3e 0a 00 54 10 04 00 57 52

|CMD>..T ...WR..

Ответ:

43 4d 44 3c 06 00 54 10 00 00 45 58 32 03

|CMD<..T ...EX2..

- **Обнуление веса (функция тара).**

На устройство отправляется запрос:

43 4d 44 3e 0e 00 48 14 08 00 57 52 00 02 00 00 00 00 00 00 18 d3

Ответ:

43 4d 44 3c 06 00 14 00 00 45 58 c1 9f
--

- **Отправка команды ПЕЧАТЬ.**

На устройство отправляется запрос:

43 4d 44 3e 0e 00 48 14 08 00 57 52 00 00 02 00 00 00 00 00 3a f1

Ответ:

43 4d 44 3c 06 00 48 14 00 00 45 58 c1 9f

Чтение состояния кнопок.

- **Чтение данных.**

Доступно в версии протокола обмена 63220330...63391231.

На устройство отправляется запрос:

43 4D 44 3E 06 00 80 14 08 0X00 52 44 C4 EE

Ответ:

43 4D 44 3C 0E 00 80 14 08 00 52 44 **7D 7B 40 40 40 40 40** 00 82 6A

В ответ возвращаются данные :

Первые два байте = “{”

Далее, состояние кнопок **включение/меню/выбор/печать/тара**

В каждом байте

бит d0 - состояние кнопки "нажата" при 1;

бит d1 - событие "нажатие кнопки" при 1;

бит d2 - событие "короткое нажатие кнопки" при 1;

бит d3 - событие "длинное нажатие кнопки" при 1;

бит d4 - событие "двойное нажатие кнопки" при 1;

биты d7,d6,d5 равны '010'

После считывания информация о нажатии стирается. До считывания, информация о нажатии накапливается, не сбрасывается. Состояние после сброса/включения – не определено – требуется выполнить одно «холостое» считывание, игнорировать его значение.

Для тестирования- есть консольная программа Get1480.exe.

Get1480.exe Version 30.03.2022 4.63

Вкл	меню	выбор	печать	тара
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	11
0	0	0	0	101
0	0	0	111	0
0	0	111	0	0
0	111	0	0	0
0	10101	0	0	0
111	0	0	0	0
10101	0	0	0	0
10101	0	0	0	0
101	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Последние значения «ТАРА» И «ПЕЧТЬ».

Доступно в версии протокола обмена, начиная HW: 65 SW: 240301.

- **Чтение данных.**

На устройство отправляется запрос:

43 4D 44 3E 06 00 80 13 08 00 52 44 71 2E

Ответ:

43 4D 44 3C 0E 00 80 13 08 00 52 44 21 2C 12 00 B1 1E 12 00 C8 B2

В ответ возвращаются данные:

21 2C 12 00 – последняя сохраненное значение “ПЕЧАТЬ”

B1 1E 12 00 – последнее значение пред нажатием “ТАРА”

Формат, расположение десятичной точки – такое-же, как и при «чтении информации»

Блокировка клавиатуры и вывода на индикатор.

Доступно в версии протокола обмена, начиная HW: 65 SW: 240301.

- **Чтение состояния.**

На устройство отправляется запрос:

43 4D 44 3E 06 00 80 13 08 00 52 44 71 2E

Ответы:

- Или 43 4D 44 3C 06 00 80 13 08 00 52 44 00 10 00 00 EC 1D
- Или 43 4D 44 3C 06 00 80 13 08 00 52 44 00 00 00 00 ED D8

Если 12 бит = 0, то клавиатура и ввод на индикатор заблокированы

Если 12 бит = 1 – обычное состояние

- **Перевод в режим блокировки**

На устройство отправляется команда:

43 4D 44 3E 0A 00 88 13 04 00 57 52 00 00 00 00 A3 0C

Ответ:

43 4D 44 3C 06 00 88 13 00 00 45 58 –подтверждение выполнения команды

- **Отключение режима блокировки**

На устройство отправляется команда:

43 4D 44 3E 0A 00 88 13 04 00 57 52 FF FF FF FF A2 98

Ответ аналогичен переводу в режим блокировки

- **Программный опроса весов.**

В ОС Linux, Windows XP/2000/7/8/10/11, Android, FreeBSD, macOS и IOS работа с устройством обеспечивается без установки дополнительно устанавливаемых драйверов. Используются драйвера, как правило, обязательно входящие в ОС, и обслуживающие такие устройства, как например USB клавиатура, мышь. Обмен производится с использованием операторов чтения/записи системы.

Идентификация (поиск) данного устройства USB HID производится по строке (пробелы и регистр обязательно учитывать):

PSE WeightA Controller

Для упрощения работы с HID устройствами можно использовать сторонние библиотеки. Например, для языков C/C++ может использоваться библиотека - **HIDAPI library for Windows, Linux, FreeBSD and macOS**. Ее можно скачать по адресам

```
// https://qbicz.github.io/hidapi/#windows
// https://github.com/Qbicz/hidapi
```

Они также доступны с примером, архив с которым можно получить у производителя – если ссылки станут не доступны. Ниже приведен пример обмена с весами в виде одного файла на языке C. В архиве есть скомпилированные файлы под Win32 и Win64.

- **Полный исходный текст программы.**

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <windows.h>
#include "hidapi.h"

int main(int argc, char* argv[])
{
    int res;
    unsigned char buf[256];
    UINT16 Vid, Pid;
    hid_device* handle;
    int i;
    struct hid_device_info* devs, * cur_dev;
    if (hid_init())
        return -1;
    devs = hid_enumerate(0x0, 0x0);
    cur_dev = devs;
    Vid = Pid = 0;
    while (cur_dev)
    {
        if ((0x1111 != cur_dev->vendor_id)
            || (0x004 != cur_dev->product_id))
        {
            cur_dev = cur_dev->next;
            continue;
        }
        Vid = cur_dev->vendor_id;
        Pid = cur_dev->product_id;
        printf(" Vid & Pid: 0x%04hx 0x%04hx\n serial_number: %ls", cur_dev->vendor_id,
            cur_dev->product_id, cur_dev->serial_number);
        printf("\n");
        printf(" Product:      %ls\n", cur_dev->product_string);
        printf("\n");
        break;
    }
    hid_free_enumeration(devs);
    if (0 == (Vid + Pid))
        // Set up the command buffer.
```

```

{
    printf("The scale was not connected or buse\n");
    hid_exit();
    system("pause");
    return 1;
}
handle = hid_open(Vid, Pid, NULL);        // Open the device using the VID, PID,
if (!handle)
{
    printf("unable to open device\n");
    return 1;
}
hid_set_nonblocking(handle, 1);           // Set the hid_read() function to be non-blocking.
Sleep(10);
//    Данные взяты из описание Получение информации от
//    весов • Чтение информации о весах
//    Для чтения других парметров - установите требуемые //    данные
unsigned char Request[65] = { 0x00, 0x43, 0x4D, 0x44, 0x3E, 0x06, 0x00, 0x00, 0x12, 0x20, 0x00,
0x52, 0x44, 0x5A, 0x8E };
printf("Data write (bytes in hex):\n ");    // Print the wrote buffer.
for (i = 1; i < 15; i++)
    printf("%02hhx ", Request[i]);
res = hid_write(handle, Request, 65);
if (res < 0)
{
    printf("Unable to write()\n");
    printf("Error: %ls\n", hid_error(handle));
    hid_close(handle);                    // Free static HIDAPI objects.
    hid_exit();
    system("pause");
    return 1;
}
Sleep(10);
memset(buf, 0, sizeof(buf));
res = hid_read(handle, buf, 65);
if (res < 0)
{
    printf("\nUnable to read()\n");
    hid_close(handle);                    // Free static HIDAPI objects.
    hid_exit();
    system("pause");
    return 1;
}
printf("\nData read (bytes in hex):\n ");    // Print the returned buffer.
for (i = 1; i < 65; i++)
    printf("%02hhx ", buf[i]);
printf("\n");
hid_close(handle);                        // Free static HIDAPI objects.
hid_exit();
system("pause");
return 0;
}

```

• Пример вывода данной программы.

Vid & Pid: 0x1111 0x0004
 serial_number: 60220330 19:42:04
 Product: PSE WeightA Controller

Data write (bytes in hex):

43 4d 44 3e 06 00 00 12 20 00 52 44 5a 8e

Data read (bytes in hex):

4d 44 3c 26 00 00 12 20 00 52 44 c2 cb d2 c5 00 00 00 00
 00 00 00 00 00 00 00 00 33 31 30 30 54 00 00 00 00 00 00
 00 00 00 00 b3 a1 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00