

## Datalog format of devices HWg-PWR and HWg-Ares

The data is stored in a simple binary format:

<record1>< record2>< record3><record4><record5>...<recordN>

The record format is following:

- Sensor ID (2 bytes)
- TimeStamp (4 bytes)
- Value (4 bytes)

The TimeStamp is stored in Unix time format (time\_t). More information about the format you can find for example here [http://en.wikipedia.org/wiki/Unix\\_time](http://en.wikipedia.org/wiki/Unix_time). The Value is stored without decimal point. An exponent tells you were you have to move decimal point to get real value (mathematical expression: right\_value = value \* 10<sup>EXP</sup>). On screen shot you can see an example of sensor with ID = 1012 and exponent -3.

The screenshot shows the HWg-PWR25 M-Bus Meter software interface. On the left is a tree view of the device structure. In the center, under the 'Device' tab, there are three sections: 'State', 'Logging', and 'Value'. The 'State' section shows 'Name' (Enable) with a checked 'Enable' radio button. The 'Logging' section shows 'Logging Enable' with a checked 'Enable' radio button. The 'Value' section shows details for sensor ID 1012: Name (Spotřeba), Unit (kWh), Exponent (-3), Tarif (0), Zero Offset (0.000), and Final Value (14456.942 kWh).

The exponent you can get from values.xml too. On following screen shot there is the same information about the the same sensor you can see in values.xml file.

But there is a few little differences between these devices. Some of them store data in little endian format and others in big endian. More information about endianness you can see for example here <http://en.wikipedia.org/wiki/Endianness>. And some of them store timestamp in local and others in universal time.

The screenshot shows a web browser displaying the XML configuration for sensor ID 1012. The URL is http://192.168.1.74/values.xml. The XML content includes configuration for sensor ID 1012, including its name (M-Count), address (0), security address (13100166), raw value (17725.253), and various timestamp and state parameters.

Endianess	TimeStamp
big	devices's local time
little	devices's local time
little	universal time (UTC)

```
/*
 * hwg_pwr_datalog.c
 *
 * Default is datalog written on standard output.
 * Command "hwg_pwr_datalog.exe > out.txt" writes the records into file.
 */

#include <stdio.h>
#include <time.h>
#include <winsock.h>

/* For device HWg-PWR 25 and HWg-Ares uncomment next line */
/* #define ARES */
```

```
#ifdef ARES
#define _HTONS_(VAL) ( VAL )
#define _HTONL_(VAL) ( VAL )
#else
#define _HTONS_(VAL) ( htons(VAL) )
#define _HTONL_(VAL) ( htonl(VAL) )
#endif

/* Special pragma for Borland C++ Builder - other compilers probably use a different way
 * how to say to compiler an information about structures packing
 */
#pragma pack(push)
#pragma pack(1)
typedef struct {
    unsigned __int16 val_id; /* Value ID */
    unsigned __int32 time; /* Unix time format time_t */
    __int32 value; /* Value = value * 10^EXP, where EXP is exponent */
} HWG_PWR_LOG_ENTRY;
#pragma pack(pop)

int main(int argc, char* argv[])
{
    FILE *InFile;
    HWG_PWR_LOG_ENTRY Entry;
    time_t t;

    InFile = fopen("datalog.bin", "rb");
    if (!InFile) {
        fprintf(stderr, "File 'datalog.bin' could not open!");
        return 1;
    }

    printf("-----\n");
    printf(" ID      | VALUE      | TIME\n");
    printf("-----\n");
    while (fread(&Entry, sizeof(HWG PWR LOG ENTRY), 1, InFile)) {
        t = _HTONL_(Entry.time);
        printf(" %5d | %10d | %s",
               _HTONS_(Entry.val_id),
               _HTONL_(Entry.value),
               asctime(gmtime(&t))
        );
    }
    printf("-----\n");
    printf(" ID      | VALUE      | TIME\n");
    printf("-----\n");

    fclose(InFile);

    return 0;
}
```



HW group s.r.o. Rumunská 26/12, Praha 2, 120 00, Czech Republic  
Phone: +420 222 511 918, E-mail: sales@hwg.cz, support@hwg.cz  
[www.HW-group.com](http://www.HW-group.com)

---