

Although this document is based on information made available by [Schoberer Radmeßsysteme / SRM Germany](#), it is . Therefore, please do not contact SRM Germany about it - any errors or inconsistencies are solely my own fault. Please send corrections and other feedback to step@stephanmantler.com .

The SRM file format is a binary file format created and used primarily by the SRM power analysis toolkit. It stores calibration data, power output, heart rate, cadence and speed information at various intervals. Marker intervals are also supported, and may overlap arbitrarily. Data is stored as a number of variable length data blocks, possibly as a function of the Powercontrol's power save mode.

In sequence, an SRM file contains:

- a magic tag & version identifier
- an overall header
- one or more markers (the first one always describes the entire file),
- one or more data block structures (for intermittent storage)
- calibration information
- and the actual data chunks.

SRM Files always begin with a 4-byte block of ASCII characters identifying the file as an SRM data file. Valid identifiers are:

OK19	?	Marker name length 3 characters. No apparent differences between these files.
SRM2	?	
SRM3	?	
SRM4	?	
SRM5	<= 1993	
SRM6	<= 2003	Marker name length 255 characters.

The overall header contains the following information:

2	Word	Wheel Circumference (mm)
2	Word	Training Date (days since Jan 1, 1880)
2	Byte x2	Recording Interval (Byte 1 / Byte 2; seconds)
2	Word	Block Count
2	Word	Marker Count
1	Byte	Padding (ignore)
1	Byte	Comment Length
70	String	Comment, zero padded to 70 Bytes

SRM Files always contain at least one marker that encompasses the entire file. Comment length depends on the file version and is either 3 bytes (up to and including SRM5), or 255 bytes (SRM6). Averaged data is multiplied by various factors and stored as integer values.

3 (up to SRM5) 255 (SRM6)	String	Comment (name for first marker, otherwise marker label).	
1	Byte	Marker Active Flag	
2	Word	Start Offset	
2	Word	End Offset.	
2	Word	Average Power (Watts)	8
2	Word	Average Heart Rate (1/min)	64
2	Word	Average Cadence (1/min)	32
2	Word	Average Speed (kph)	2500/9
2	Word	PWC150	

Since the Powercontrol unit may go into power save mode when idle, the data may represent multiple discontinuous blocks. Each data block header contains the time stamp (msec since midnight) that the block was begun, and the number of data chunks contained in the block.

4	Long	Timestamp
2	Word	Chunk count for this block

This structure contains power meter calibration data; adjusting these values in the SRM toolkit also changes the actual power data.

2	Word	Zero position (Hz)
2	Word	Gradient (internal format)
2	Word	Chunk count
1	Byte	Padding (ignore)

Data chunks contain seed and power in a combined 3-byte format (12 bytes each), heart rate and cadence. Speed is stored as km/h, power as Watts, and heart rate and cadence as 1/min.

1	Byte	SP1
1	Byte	SP2
1	Byte	SP3
1	Byte	Heart rate
1	Byte	Cadence

Speed = (SP2 & 0xF0) << 3 | (SP1 & 0x7F) * 3.0/26
Power = (SP2 & 0x0F) | (SP3 << 4);

The following data types may be helpful with implementing support for the SRM file format.

```
#pragma pack(1)
struct SRMHeader
{
    unsigned short date;           // date of this recording
    unsigned short circumference;  // wheel circumference in mm
    unsigned char interval1;       //
    unsigned char interval2;       // recording interval = interval1/interval2 sec
    unsigned short blockcount;     // number of data blocks
    unsigned short markercount;    // number of additional (!) markers
    unsigned char pad;             // ignore this
    unsigned char commentlen;      // number of characters used in comment
    unsigned char comment[70];     // comment string
};

struct SRMMarkerV5
{
    unsigned char comment[3];      // comment or rider name (in first marker)
    unsigned char active;          // marker is active?
    unsigned short start;          // first data chunk
    unsigned short end;            // last data chunk
    unsigned short avgPower;       // average power * 8
    unsigned short avgHR;          // average heart rate * 64
    unsigned short avgCadence;     // average cadence * 32
    unsigned short avgSpeed;       // average speed * 2500 / 9
    unsigned short pwc150;         // heart rate at 150 watts
};

struct SRMMarkerV6
{
    unsigned char comment[255];    // see SRMMarkerV5
    unsigned char active;
    unsigned short start;
    unsigned short end;
    unsigned short avgPower;
    unsigned short avgHR;
    unsigned short avgCadence;
    unsigned short avgSpeed;
    unsigned short pwc150;
};

struct SRMBlockHeader
{
    unsigned long timestamp;       // msec since midnight this block was started
    unsigned short count;         // number of data chunks for this block
};

struct SRMDataHeader
{
    unsigned short zeroPosition;   // power meter zero position (Hz)
    unsigned short slope;          // power meter slope (internal format)
    unsigned short dataCount;      // total number of data chunks
    unsigned char pad;
};

struct SRMDataChunk
{
    unsigned char ps1;             // combined speed / power
    unsigned char ps2;             // combined speed / power
    unsigned char ps3;             // combined speed / power
    unsigned char cadence;         // cadence (rpm)
    unsigned char hr;              // heart rate (bpm)
};
```

Stephan Mantler
step@stephanmantler.com
Payergasse 1/17
2340 Mödling
Austria

An up-to-date version of this document can be found at <http://www.stephanmantler.com/>

This work is licensed under the Creative Commons Attribution-NoDerivs 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

- 1.1 (2006-01-12) Document restructured, creative commons license added, C++ structure templates added.
- 1.0 (2006-01-10) Initial revision.