Data file structure

Data of the Photon Distribution Modes

The 'Save' functions of the SPCM software generate both setup and data files. Setup files have the extension .set and contain the system and software setup parameters. Data files have the extension .sdt and contain the parameters and the measurement data. Both file types have the same general structure. They only differ in that the SET-files do not contain any measurement data.

Files generates by the 'Autosave' functions of the measurement routines are .sdt files. The structure is identical with that of the .sdt files generates by the 'save' routine.

Photon files generated by the FIFO modes have a different file format. For these files, please see 'FIFO Files', page 335.

With the introduction of the SPC-134 and the combined scanning and routing in the SPC-700/730 a modification of the file structure became necessary. The changes were required to identify the individual modules of a multi-SPC system and to save more than 65565 curves of a scan measurement. Older files of the SPC Standard Software for Windows, versions 2.0 to 6.9, are generally compatible with the new structure. However, loading files of version 7.0 or later into old software versions can (but need not) cause problems.

The data files consist of

- a *file header* containing structural data which are used to find the other parts of the file
- the *file information* which was typed in when the file was saved
- the *system setup* data for hardware and software
- one or more *measurement description blocks* which contain the system parameters corresponding to the particular data blocks
- *data blocks* containing sets of curves generated by one measurement.

File Header

Data recorded by the bh SPC modules may contain a large number of data blocks for different measurement steps, pixels of a scan, and detector channels. A SPCM data file may even contain several measurements, with possibly different system parameters. Therefore the SPCM data files start with a binary file header which contains general information about the location of the setup and measurement data within the file. An example of a file header is given below.

short	revision	software revision number (lower 4 bits = 11(decimal))
long	info offset	offset of the info part which contains general information (Title, date, time, contents etc.)
short	info length	length of the info part
long	setup_offs	offset of the setup data (system parameters, display parameters, trace parameters etc.)
short	setup_length	length of the setup data
long	data_block_offset	offset of the first data block
short	no_of_data_blocks	no_of_data_blocks valid only when in 0 .. 0x7ffe range, if equal to 0x7fff the field 'reserved1' contains valid no_of_data_blocks
long	data_block_length	length of the longest data block in the file
long	meas_desc_block_offset	offset to 1st. measurement description block (system parameters connected to data blocks)
short	no_of_meas_desc_blocks	number of measurement description blocks
short	meas_desc_block_length	length of the measurement description blocks
unsigned short	header_valid	valid: 0x5555, not valid: 0x1111
unsigned long	reserved1	reserved1 now contains no_of_data_blocks
unsigned short	reserved2	
unsigned short	chksum	checksum of file header

The exact definition of the header is available from the file 'spc_data_file_structure.h'. The installation procedure writes an spc_data_file_structure.h file into the same directory as the SPCM application.

File Information

The File Information part contains the general information written into the file info field during the 'Save' procedure (see 'Save Panel', page 243). The info part is stored in ASCII format. An example is given below.

```
*IDENTIFICATION
    ID          : _SPC Setup & Data File_
    Title       : startup
    Version     : 007
    Revision    : 1
    Date        : 10-10-2004
    Time        : 12:29:01
    Author      : Bond, James
    Company     : Unknown
    Contents    : Tissue sample from Dr. No, 2p excitation at 750 nm
*END
```

Setup

The setup block contains all system parameters, display parameters, trace parameters, etc. used to set the SPC system (hardware and software) into the state in which it was when the data file was stored. All parameter values are stored together with an identifier. This method allows to maintain compatibility between different SPC versions. If a parameter is missing in the setup part, i.e. if a file from an older software version is loaded, a default value is used when the file is loaded. A typical setup part is shown in Fig. 348. The list is for information only; new parameters may be added in future software versions. For Multi-SPC Systems the system parameters section contains separate subsections for module parameters of the individual modules.

```
SYS_PARA_BEGIN
#PR [PR_PDEV,I,0]
#PR [PR_PPORT,I,2]
#PR [PR_PWHAT,I,0]
#PR [PR_PF,B,0]
#PR [PR_PFNAME,S,IMAGE.PRT]
#PR [PR_PORIEN,I,1]
#PR [PR_PEJECT,B,1]
#PR [PR_PWIDTH,F,100]
#PR [PR_PHEIGH,F,100]
#PR [PR_PFULL,B,1]
#PR [PR_PAUTO,B,1]
#PR [PR_STP_FN,S,STP.CFG]
#PR [PR_SAVE_T,I,2]
#SP [SP_MODE,I,0]
#SP [SP_CFD_LL,F,-20]
#SP [SP_CFD_LH,F,80]
#SP [SP_CFD_ZC,F,0]
#SP [SP_CFD_HF,F,5]
#SP [SP_SYN_ZC,F,-9.8267717]
#SP [SP_SYN_FD,I,4]
#SP [SP_SYN_FQ,F,-20]
#SP [SP_SYN_HF,F,4]
#SP [SP_TAC_R,F,5.0000001e-08]
#SP [SP_TAC_G,,1]
#SP [SP_TAC_OF,F,9.4993896]
#SP [SP_TAC_LL,F,14.90196]
#SP [SP_TAC_LH,F,84.705879]
#SP [SP_TAC_TC,F,4.8828126e-11]
#SP [SP_TAC_TD,F,6.2500001e-09]
#SP [SP_ADC_RE,I,1024]
#SP [SP_EAL_DE,I,30]
#SP [SP_NCX,I,1]
#SP [SP_NCY,I,1]
#SP [SP_PAGE,I,1]
#SP [SP_COL_T,F,100.01]
#SP [SP_REP_T,F,100.01]
#SP [SP_DIS_T,F,0.99899995]
#SP [SP_REPEAT,B,0]
#SP [SP_STOPT,B,1]
#SP [SP_OVERFL,C,S]
#SP [SP_WL_STA,F,300]
#SP [SP_WL_STO,F,362]
#SP [SP_WL_STE,F,2]
#SP [SP_EXTST,B,0]
#SP [SP_STEPS,I,32]
#SP [SP_OFFSET,F,0]
#SP [SP_YWIN_N,I,8]
#SP [SP_XWIN_N,I,8]
#SP [SP_TWIN_N,I,8]
#SP [SP_X_EQU,B,1]
#SP [SP_Y_EQU,B,1]
#SP [SP_T_EQU,B,1]
#SP [SP_DITH,I,64]
#SP [SP_EN_INT,B,0]
#SP [SP_INCR,I,64]
#SP [SP_DAES,B,1]
#SP [SP_SPE_FN,S,SPEC1.SDT]
#SP [SP_CYCLES,U,1]
#SP [SP_DAEC,B,0]
#SP [SP_MEM_BANK,I,0]
#SP [SP_DTCOMP,B,1]
#DI [DI_SCALE,I,0]
#DI [DI_MAXCNT,L,65535]
#DI [DI_LBLINE,L,100]
#DI [DI_BLINE,L,0]
#DI [DI_GRID,B,0]
#DI [DI_GCOL_F,I,8]
#DI [DI_GCOL_B,I,0]
#DI [DI_TRACE,I,0]
#DI [DI_BOD_C,I,3]
#DI [DI_2DDIS,I,0]
#DI [DI_2DTRNO,I,1]
#DI [DI_3DOFFX,I,4]
#DI [DI_3DOFFY,I,4]
#DI [DI_3DINCX,I,0]
#DI [DI_3DCOL,I,15]
#DI [DI_3DMODE,I,3]
#DI [DI_YWIN,I,2]
#DI [DI_XWIN,I,1]
#DI [DI_TWIN,I,1]
#DI [DI_PSTYLE,I,9]
#DI [DI_PFREQ,I,1]
#DI [DI_CUR,B,0]
#DI [DI_RATE,B,1]
#DI [DI_2DC1,B,1]
#DI [DI_2DC2,B,1]
#DI [DI_2DC1C,I,1]
#DI [DI_2DC2C,I,5]
#DI [DI_2DC1S,I,0]
#DI [DI_2DC2S,I,0]
#DI [DI_3DC1C,I,12]
#DI [DI_3DC2C,I,14]
#DI [DI_SIZE,I,1]
#MP0 [MP_CFD_LL,F,0]
#MP0 [MP_CFD_LH,F,39.843136]
#MP0 [MP_CFD_ZC,F,-7.5590553]
#MP0 [MP_CFD_HF,F,5]
#MP0 [MP_SYN_ZC,F,-4.5354333]
#MP0 [MP_SYN_FD,I,1]
#MP0 [MP_SYN_FQ,F,-19.607843]
#MP0 [MP_SYN_HF,F,4]
#MP0 [MP_TAC_LL,F,7.8431373]
#MP0 [MP_TAC_LH,F,90.588234]
#MP0 [MP_TRIGGER,I,0]
#MP0 [MP_TAC_OF,F,9.8039217]
#MP1 [MP_CFD_LL,F,0]
#MP1 [MP_CFD_LH,F,39.843136]
#MP1 [MP_CFD_ZC,F,-7.5590553]
#MP1 [MP_CFD_HF,F,5]
#MP1 [MP_SYN_ZC,F,-4.5354333]
#MP1 [MP_SYN_FD,I,1]
#MP1 [MP_SYN_FQ,F,-19.607843]
#MP1 [MP_SYN_HF,F,4]
#MP1 [MP_TAC_LL,F,7.8431373]
#MP1 [MP_TAC_LH,F,90.588234]
#MP1 [MP_TRIGGER,I,0]
#MP1 [MP_TAC_OF,F,9.8039217]
SYS_PARA_END
TRACE_PARA_BEGIN
#TR #0 [1,15,1,1,4,1,1]
#TR #1 [0,9,1,2,1,2,1]
#TR #2 [0,10,1,3,1,3,1]
#TR #3 [0,14,1,4,1,4,1]
#TR #4 [0,9,1,5,1,1,1]
#TR #5 [0,12,1,6,1,1,1]
#TR #6 [0,13,1,7,1,1,1]
#TR #7 [0,11,1,8,1,1,1]
TRACE_PARA_END
WIND_PARA_BEGIN
#WI #0 *NO *0 [0,0]
#WI #0 *NO *1 [0,0]
#WI #0 *NO *2 [0,0]
#WI #0 *NO *3 [0,0]
#WI #0 *NO *4 [0,0]
#WI #0 *NO *5 [0,0]
#WI #0 *NO *6 [0,0]
#WI #0 *NO *7 [0,0]
#WI #1 *NO *0 [0,0]
#WI #1 *NO *1 [0,0]
#WI #1 *NO *2 [0,0]
#WI #1 *NO *3 [0,0]
#WI #1 *NO *4 [0,0]
#WI #1 *NO *5 [0,0]
#WI #1 *NO *6 [0,0]
#WI #1 *NO *7 [0,0]
#WI #2 *NO *0 [0,127]
#WI #2 *NO *1 [128,255]
#WI #2 *NO *2 [256,383]
#WI #2 *NO *3 [384,511]
#WI #2 *NO *4 [512,639]
#WI #2 *NO *5 [640,767]
#WI #2 *NO *6 [768,895]
#WI #2 *NO *7 [896,1023]
WIND_PARA_END
*END
```

Fig. 348: Setup parameters

Measurement Description Blocks

The SPCM software allows you to run several measurements in different memory pages and to store the results in a single data file. These measurements may be done with different system parameters. Therefore, each data block can (but need not) have its own set of system parameters. These parameters may differ from the general setup parameters.

The system parameters of the individual measurements are stored in the measurement description blocks. The number of measurement description blocks can vary from one (if all stored data blocks have the same system parameters) to the overall number of saved data blocks (if all blocks were measured with different hardware parameters).

In the block header of each data block a corresponding measurement description block is specified. The number of measurement description blocks, and the length and the location of the measurement description blocks are stored in the file header at the beginning of the .sdt file.

The information in the measurement description blocks is used for the 'Block Info' or 'Set Info' function of the Load, Save and Trace Parameter panels. If the button 'Use System Parameters from the Selected Block' is pressed, the system parameters are replaced with the data in the measurement description block.

The measurement description blocks are stored in a binary format. The structure is shown in Fig. 349. Fig. 349 should be considered an example; more parameters may be added for new software versions or new TCSPC modules. For details of the current structure definitions, please see SPC_data_file_structure.h file of the DLL library.

```
cchar time[9];        /* time of creation */
char date[11];        /* date of creation */
char mod_ser_no[16];  /* serial number */
short   meas_mode;
float   cfd_ll;
float   cfd_lh;
float   cfd_zc;
float   cfd_hf;
float   syn_zc;
short   syn_fd;
float   syn_hf;
float   tac_r;
short   tac_g;
float   tac_of;
float   tac_ll;
float   tac_lh;
short   adc_re;
short   eal_de;
short   ncx;
short   ncy;
unsigned short page;
float   col_t;
float   rep_t;
short   stopt;
char    overfl;
short   use_motor;
short   steps;
float   offset;
short   dither;
short   incr;
short   mem_bank;
char    mod_type[16];  /* module type */
float   syn_th;
short   dead_time_comp;
short   polarity_l;
short   polarity_f;
short   polarity_p;
short   linediv;
short   accumulate;
int     flbck_y;
int     flbck_x;
int     bord_u;
int     bord_l;
float   pix_time;
short   pix_clk;
short   trigger;
int     scan_x;
int     scan_y;
int     scan_rx;
int     scan_ry;
short   fifo_typ;
int     epx_div;
int     mod_type_code;
float   overflow_corr_factor;
int   adc_zoom;
int cycles, //cycles (accumulation cycles in
FLOW mode)
MeasStopInfo  StopInfo;
```

Fig. 349: Measurement description block

Data Blocks

Each data block starts with a data block header which describes the length of the data block, the type of the data, and the measurement description block related to the data block.

With the software version 7.0 the data block header was changed to make a higher number of data blocks and a variable block size possible. Each data block can now contain a 'Data Set' i.e. the data of several curves which were obtained in one measurement. The structure of the block header is shown in Fig. 350.

short	block_no	number of the block in the file, valid only when in 0 .. 0x7ffe range, if equal to 0x7fff block_no (old software version - reserved1) field contains valid number of the block in the file
long	data_offs	offset of the data block from the beginning of the file
long	next_block_offs	offset to the data block header of the next data block
unsigned short	block_type	0: unused 1: measured block 2: flow data 3: data block from file 4: calculated data [illegible] 5: simulated data block, 11(hex): measured data set 13(hex): data set from file 14(hex): calculated data set 15(hex): simulated data set,
short	meas_desc_block_no	Number of the measurement description block corresponding to this data block
unsigned long	lblock_no	reserved1 now contains number of the block in the file*
unsigned long	block_length	reserved2 now contains block(set) length in bytes

* The field 'lblock_no' contains the data block / data set number in the bits 0 to 23 and the module number (0 to 3) in the bits 24 to 25

Fig. 350: Structure of the data block header

The data of the set specified by the block header are stored as shown below. It follows dire[illegible] after the data block header:

unsigned short	curvepoint[0][0]
unsigned short	curvepoint[0][1]
.	.
.	curvepoint[0][adc_re -1]
unsigned short	curvepoint[1][0]
unsigned short	curvepoint[1][1]
.	.
.	curvepoint[1][adc_re -1]
.	
unsigned short	curvepoint[n][0]
unsigned short	curvepoint[n][1]
.	.
.	curvepoint[n][adc_re -1]

The photon numbers in each curve point are unsigned short integers, i.e. values from [illegible] 65,535. The number of curves in the data set depends on the measurement parameters, e[illegible] measurement mode, no of routing bits etc. The number of curves in the block is equa[illegible] 'block_length' (from the block header) divided by adc_resolution (from the correspond[illegible] measurement description block).