

# SR850 File Format

## Application Note #14

### File Format for the SR850 Lock-In Amplifier

#### FAST command

##### Problem:

Some users want to read the settings and data from an SR850 file into their programs.

##### Solution:

The following C code shows the order that the data structures (below) are written to the data file. The data structures can be used directly in a C program—the code can not since it uses proprietary I/O routines.

```
{
k += pprintn(&str1,dataXfer,typStr[opType], (ulng) 8 );

k+=pprintn(&str1,dataXfer,aD->myT, (ulng) sizeof(liaTrace));
k+=pprintn(&str1,dataXfer,&presState, (ulng) sizeof(LiaState));
k+=pprintn(&str1,dataXfer,&nPts, (ulng) sizeof(long));
nPts<=<=2;
while (nPts)
{
dPtr = MK_FP( TraceSegs[(lp.segBase<<2)+lp.segIdx],lp.off );
il = min(0x10000-lp.off,nPts);
k+=pprintn(&str1,dataXfer,dPtr, (ulng) il);
nPts -= il;
lp.off=0; lp.segIdx = (lp.segIdx+1)%4;
}
k+=pprintn(&str1,dataXfer,&svMarkBin, (ulng) sizeof(long));
k+=pprintn(&str1,dataXfer,theMarks, (ulng) sizeof(userMark)*MAX_UMARK);
k+=pprintn(&str1,dataXfer,markTxts, (ulng) UM_TXT_LEN*MAX_UMARK);
}

typedef struct LIA_TRACE
{
int buffNo; /* row index of TraceSegs[][] */
int isStored; /* is this being stored in time series? */
int dataType; /* is this float? */
char gLabel[16]; /* Graph Label: used in plt_dsk.c only ??? */
int TraceQuantities[3]; /* list indexes for the menu */
int *underVal; /* pointer into rawData[] */
LiaState *myS;
} liaTrace;

/* menu variables only - hardware level stuff are separate
variables. e.g. latch1Data, filter integrate, leakVar etc. */
typedef struct LIA_STATE {
int gain; /* index into list */
int timeConst;
int reserveMode; /* Max, Man, Min */
int reserveMan;
int filterSlope;
int filterSynch;
long intFreq;
int refLevel; /* refLevel (in milliVolts) */
int refSource;
long refStart;
long refStop;
```

```

int refSweepType;
int refSweepBW;
int refSweepBWType;
long Phase;
int refRange;          /* (.5 to 100kHz or .001 to 500) */
int harmonic;
int sigSource;
int sigCouple;
int sigNotch;
int refSlope;
int sigGround;
int rOffset;
int rExpand;
int xOffset;
int xExpand;
int yOffset;
int yExpand;
int SampleRate;
int unityVal;
int leftBNCsrc;
int rightBNCsrc;
int Looping;           /* vs. single shot mode */
unsigned long rtuScanLen; /* scan time in rtu */
int displayType;      /* single (full) or up/down */
int fullPen;         /* 41st int (presState[80]) */
int upPen;
int downPen;
int fullGType;       /* polar, chart, 2axis, eventLog */
int upGType;        /* desired type: blank, bar, (scope,) strip */
int downGType;
int trigStart;      /* 47th int (presState[92]) */
int curGain;        /* // new 9/14/93 */
int topShowMode;    /* Parameters or Input/Output */
int AuxOutSweep[4]; /* True or Fixed */
int AuxOutV[4];
int AuxOutStart[4]; /* start mVolts */
int AuxOutStop[4]; /* stop mVolts */
int AuxOutOff[4];  /* offset mvolts */
} LiaState;

```

