```
void      main(int, char *[]);
void      txLia(char *);
void      initGpib(char *);
void      setupLia(void);
void      printOutBinaryResults(void);
void      printOutIEEEResults(void);
void      printOutLIAResults(void);

/* National Instruments Interface Function Prototypes (488.1 Calls - see the National software manual).
   These are declared in "decl.h"
int       ibfind(char*);
void      ibwrt(int,char *,int);
void      ibrd(int,char *,unsigned long);
void      ibrsp(int,char *);
void      ibeos(int,int);
void      ibtmo(int,int);
*/

/* global variables */

int  lia;                 /* SR830 handle */
int       rxBuf[660*2];   /* FAST mode data buffer */
float     rfBuf[1000];    /* Floating point data buffer */

void main(int argc, char *argv[])
{
    int       nPts,i;
    char      tstr[20];

    if (argc<2) {
        printf("\nUsage: liatest <devName>\n");
        exit(1);
    }
    else
        initGpib(SR830);

    txLia("OUTX1");      /* Set the SR830 to output responses to the GPIB port */
    setupLia();          /* Setup the SR830 */

    printf("\nAcquiring Data\n");
    ibtmo(lia,0);            /* turn off timeout for lia or set the timeout longer than the scan (10 seconds). The
                             timeout measures the time to transfer the FULL number of bytes, not the time since
                             the most recent byte is received.*/

    txLia("FAST2;STRD");        /* Turn FAST mode data transfer ON, then start scan using the STRD start
                                after delay command. The STRD command MUST be used if the scan is to
                                be started by this program! Do NOT use STRT. */

    /* take data for 10 seconds and then stop */
    ibrd(lia,(char *)rxBuf,2564L);         /* get FAST mode data for 10 seconds.
                                           10 seconds of data at 64 Hz sample rate has 64*10 + 1 points,
                                           each point consists of X (2 bytes) and Y (2 bytes) for a total of
                                           4*(64*10+1) = 2564 bytes. */
    i=(int)ibcnt;                          /* save total number of bytes read */
```