# Appendix B

## LabWindow Code for Memory Measurement

```
#include <gpib.h>
//#include <windows.h>
#include <utility.h>
//#include "decl-32.h"
#include <stdio.h>
#include <string.h>
#include <userint.h>
#include <dataacq.h>
#include <ansi_c.h>
#include "MUX_AC.h"

static int daq, daq1;
FILE *fp_out;
int Device1;
int cross_point[9][9],set_bit[9][9];
int num_read, all_switch,all_control=-1, ramp, ramp_num=20;
double time_write, time_read, volt_write_on,volt_write_off, volt_read, volt_hold, threshold_high,threshold_low;
double adch0,adch1, volt_ramp0, volt_ramp1, ramp_rate;
const char tmp_file[10]="tmp.dat";

void main(){
        int i;
        Device1=ibdev(0,18,0,10,1,0);                   /* initiate 707A  */
        ibwrt(Device1,"REMOTE",6);                              /* enable remote mode   */
        ibwrt(Device1,"E0X",3);                                         /* Point to present relays  */

        daq = LoadPanel (0, "MUX_AC.uir", MUX);
        DisplayPanel (daq);
        i=AI_Clear (1);
        RunUserInterface ();
}
int select_ind (int panel, int control, int event,
                void *callbackData, int eventData1, int eventData2)
        {
        daq1 = LoadPanel (1, "MUX_AC.uir", MUX1);
        DisplayPanel (daq1);
        return 1;
        }
int close_selection(int panel, int control, int event,
                void *callbackData, int eventData1, int eventData2)
        {
        int i,m;
        i=HidePanel(daq1);
        return 0;
        }
int switch_control(int panel, int control, int event,
                void *callbackData, int eventData1, int eventData2)
        {
                int m;
                if(all_control==-1){
                        m=SetCtrlAttribute(daq,MUX_ALL_SWITCHES, ATTR_DIMMED, 0);
                }
                else{
                        m=SetCtrlAttribute(daq,MUX_ALL_SWITCHES, ATTR_DIMMED, 1);
                }
                all_control=all_control*(-1);
                return 1;
        }
int configure_ind (int panel, int control, int event,
                void *callbackData, int eventData1, int eventData2)
        {
```

```
int i,j,k,m,i_ramp;
char c[5],d[6];
if (all_control!=1){
m = GetCtrlVal (daq, MUX_Switch1_1, &cross_point[1][1]);
m = GetCtrlVal (daq, MUX_Switch1_2, &cross_point[1][2]);
m = GetCtrlVal (daq, MUX_Switch1_3, &cross_point[1][3]);
m = GetCtrlVal (daq, MUX_Switch1_4, &cross_point[1][4]);
m = GetCtrlVal (daq, MUX_Switch1_5, &cross_point[1][5]);
m = GetCtrlVal (daq, MUX_Switch1_6, &cross_point[1][6]);
m = GetCtrlVal (daq, MUX_Switch1_7, &cross_point[1][7]);
m = GetCtrlVal (daq, MUX_Switch1_8, &cross_point[1][8]);
m = GetCtrlVal (daq, MUX_Switch2_1, &cross_point[2][1]);
m = GetCtrlVal (daq, MUX_Switch2_2, &cross_point[2][2]);
m = GetCtrlVal (daq, MUX_Switch2_3, &cross_point[2][3]);
m = GetCtrlVal (daq, MUX_Switch2_4, &cross_point[2][4]);
m = GetCtrlVal (daq, MUX_Switch2_5, &cross_point[2][5]);
m = GetCtrlVal (daq, MUX_Switch2_6, &cross_point[2][6]);
m = GetCtrlVal (daq, MUX_Switch2_7, &cross_point[2][7]);
m = GetCtrlVal (daq, MUX_Switch2_8, &cross_point[2][8]);
m = GetCtrlVal (daq, MUX_Switch3_1, &cross_point[3][1]);
m = GetCtrlVal (daq, MUX_Switch3_2, &cross_point[3][2]);
m = GetCtrlVal (daq, MUX_Switch3_3, &cross_point[3][3]);
m = GetCtrlVal (daq, MUX_Switch3_4, &cross_point[3][4]);
m = GetCtrlVal (daq, MUX_Switch3_5, &cross_point[3][5]);
m = GetCtrlVal (daq, MUX_Switch3_6, &cross_point[3][6]);
m = GetCtrlVal (daq, MUX_Switch3_7, &cross_point[3][7]);
m = GetCtrlVal (daq, MUX_Switch3_8, &cross_point[3][8]);
m = GetCtrlVal (daq, MUX_Switch4_1, &cross_point[4][1]);
m = GetCtrlVal (daq, MUX_Switch4_2, &cross_point[4][2]);
m = GetCtrlVal (daq, MUX_Switch4_3, &cross_point[4][3]);
m = GetCtrlVal (daq, MUX_Switch4_4, &cross_point[4][4]);
m = GetCtrlVal (daq, MUX_Switch4_5, &cross_point[4][5]);
m = GetCtrlVal (daq, MUX_Switch4_6, &cross_point[4][6]);
m = GetCtrlVal (daq, MUX_Switch4_7, &cross_point[4][7]);
m = GetCtrlVal (daq, MUX_Switch4_8, &cross_point[4][8]);
m = GetCtrlVal (daq, MUX_Switch5_1, &cross_point[5][1]);
m = GetCtrlVal (daq, MUX_Switch5_2, &cross_point[5][2]);
m = GetCtrlVal (daq, MUX_Switch5_3, &cross_point[5][3]);
m = GetCtrlVal (daq, MUX_Switch5_4, &cross_point[5][4]);
m = GetCtrlVal (daq, MUX_Switch5_5, &cross_point[5][5]);
m = GetCtrlVal (daq, MUX_Switch5_6, &cross_point[5][6]);
m = GetCtrlVal (daq, MUX_Switch5_7, &cross_point[5][7]);
m = GetCtrlVal (daq, MUX_Switch5_8, &cross_point[5][8]);
m = GetCtrlVal (daq, MUX_Switch6_1, &cross_point[6][1]);
m = GetCtrlVal (daq, MUX_Switch6_2, &cross_point[6][2]);
m = GetCtrlVal (daq, MUX_Switch6_3, &cross_point[6][3]);
m = GetCtrlVal (daq, MUX_Switch6_4, &cross_point[6][4]);
m = GetCtrlVal (daq, MUX_Switch6_5, &cross_point[6][5]);
m = GetCtrlVal (daq, MUX_Switch6_6, &cross_point[6][6]);
m = GetCtrlVal (daq, MUX_Switch6_7, &cross_point[6][7]);
m = GetCtrlVal (daq, MUX_Switch6_8, &cross_point[6][8]);
m = GetCtrlVal (daq, MUX_Switch7_1, &cross_point[7][1]);
m = GetCtrlVal (daq, MUX_Switch7_2, &cross_point[7][2]);
m = GetCtrlVal (daq, MUX_Switch7_3, &cross_point[7][3]);
m = GetCtrlVal (daq, MUX_Switch7_4, &cross_point[7][4]);
m = GetCtrlVal (daq, MUX_Switch7_5, &cross_point[7][5]);
m = GetCtrlVal (daq, MUX_Switch7_6, &cross_point[7][6]);
m = GetCtrlVal (daq, MUX_Switch7_7, &cross_point[7][7]);
m = GetCtrlVal (daq, MUX_Switch7_8, &cross_point[7][8]);
m = GetCtrlVal (daq, MUX_Switch8_1, &cross_point[8][1]);
m = GetCtrlVal (daq, MUX_Switch8_2, &cross_point[8][2]);
m = GetCtrlVal (daq, MUX_Switch8_3, &cross_point[8][3]);
m = GetCtrlVal (daq, MUX_Switch8_4, &cross_point[8][4]);
m = GetCtrlVal (daq, MUX_Switch8_5, &cross_point[8][5]);
m = GetCtrlVal (daq, MUX_Switch8_6, &cross_point[8][6]);
m = GetCtrlVal (daq, MUX_Switch8_7, &cross_point[8][7]);
m = GetCtrlVal (daq, MUX_Switch8_8, &cross_point[8][8]);
}
else{
        m = GetCtrlVal (daq, MUX_ALL_SWITCHES, &all_switch);
```

```
		for(i=1;i<=8;i++){
			for(j=1;j<=8;j++){
			cross_point[i][j]=all_switch;
			}
		}
}
m = GetCtrlVal (daq1, MUX1_Switch1_1, &set_bit[1][1]);
m = GetCtrlVal (daq1, MUX1_Switch1_2, &set_bit[1][2]);
m = GetCtrlVal (daq1, MUX1_Switch1_3, &set_bit[1][3]);
m = GetCtrlVal (daq1, MUX1_Switch1_4, &set_bit[1][4]);
m = GetCtrlVal (daq1, MUX1_Switch1_5, &set_bit[1][5]);
m = GetCtrlVal (daq1, MUX1_Switch1_6, &set_bit[1][6]);
m = GetCtrlVal (daq1, MUX1_Switch1_7, &set_bit[1][7]);
m = GetCtrlVal (daq1, MUX1_Switch1_8, &set_bit[1][8]);
m = GetCtrlVal (daq1, MUX1_Switch2_1, &set_bit[2][1]);
m = GetCtrlVal (daq1, MUX1_Switch2_2, &set_bit[2][2]);
m = GetCtrlVal (daq1, MUX1_Switch2_3, &set_bit[2][3]);
m = GetCtrlVal (daq1, MUX1_Switch2_4, &set_bit[2][4]);
m = GetCtrlVal (daq1, MUX1_Switch2_5, &set_bit[2][5]);
m = GetCtrlVal (daq1, MUX1_Switch2_6, &set_bit[2][6]);
m = GetCtrlVal (daq1, MUX1_Switch2_7, &set_bit[2][7]);
m = GetCtrlVal (daq1, MUX1_Switch2_8, &set_bit[2][8]);
m = GetCtrlVal (daq1, MUX1_Switch3_1, &set_bit[3][1]);
m = GetCtrlVal (daq1, MUX1_Switch3_2, &set_bit[3][2]);
m = GetCtrlVal (daq1, MUX1_Switch3_3, &set_bit[3][3]);
m = GetCtrlVal (daq1, MUX1_Switch3_4, &set_bit[3][4]);
m = GetCtrlVal (daq1, MUX1_Switch3_5, &set_bit[3][5]);
m = GetCtrlVal (daq1, MUX1_Switch3_6, &set_bit[3][6]);
m = GetCtrlVal (daq1, MUX1_Switch3_7, &set_bit[3][7]);
m = GetCtrlVal (daq1, MUX1_Switch3_8, &set_bit[3][8]);
m = GetCtrlVal (daq1, MUX1_Switch4_1, &set_bit[4][1]);
m = GetCtrlVal (daq1, MUX1_Switch4_2, &set_bit[4][2]);
m = GetCtrlVal (daq1, MUX1_Switch4_3, &set_bit[4][3]);
m = GetCtrlVal (daq1, MUX1_Switch4_4, &set_bit[4][4]);
m = GetCtrlVal (daq1, MUX1_Switch4_5, &set_bit[4][5]);
m = GetCtrlVal (daq1, MUX1_Switch4_6, &set_bit[4][6]);
m = GetCtrlVal (daq1, MUX1_Switch4_7, &set_bit[4][7]);
m = GetCtrlVal (daq1, MUX1_Switch4_8, &set_bit[4][8]);
m = GetCtrlVal (daq1, MUX1_Switch5_1, &set_bit[5][1]);
m = GetCtrlVal (daq1, MUX1_Switch5_2, &set_bit[5][2]);
m = GetCtrlVal (daq1, MUX1_Switch5_3, &set_bit[5][3]);
m = GetCtrlVal (daq1, MUX1_Switch5_4, &set_bit[5][4]);
m = GetCtrlVal (daq1, MUX1_Switch5_5, &set_bit[5][5]);
m = GetCtrlVal (daq1, MUX1_Switch5_6, &set_bit[5][6]);
m = GetCtrlVal (daq1, MUX1_Switch5_7, &set_bit[5][7]);
m = GetCtrlVal (daq1, MUX1_Switch5_8, &set_bit[5][8]);
m = GetCtrlVal (daq1, MUX1_Switch6_1, &set_bit[6][1]);
m = GetCtrlVal (daq1, MUX1_Switch6_2, &set_bit[6][2]);
m = GetCtrlVal (daq1, MUX1_Switch6_3, &set_bit[6][3]);
m = GetCtrlVal (daq1, MUX1_Switch6_4, &set_bit[6][4]);
m = GetCtrlVal (daq1, MUX1_Switch6_5, &set_bit[6][5]);
m = GetCtrlVal (daq1, MUX1_Switch6_6, &set_bit[6][6]);
m = GetCtrlVal (daq1, MUX1_Switch6_7, &set_bit[6][7]);
m = GetCtrlVal (daq1, MUX1_Switch6_8, &set_bit[6][8]);
m = GetCtrlVal (daq1, MUX1_Switch7_1, &set_bit[7][1]);
m = GetCtrlVal (daq1, MUX1_Switch7_2, &set_bit[7][2]);
m = GetCtrlVal (daq1, MUX1_Switch7_3, &set_bit[7][3]);
m = GetCtrlVal (daq1, MUX1_Switch7_4, &set_bit[7][4]);
m = GetCtrlVal (daq1, MUX1_Switch7_5, &set_bit[7][5]);
m = GetCtrlVal (daq1, MUX1_Switch7_6, &set_bit[7][6]);
m = GetCtrlVal (daq1, MUX1_Switch7_7, &set_bit[7][7]);
m = GetCtrlVal (daq1, MUX1_Switch7_8, &set_bit[7][8]);
m = GetCtrlVal (daq1, MUX1_Switch8_1, &set_bit[8][1]);
m = GetCtrlVal (daq1, MUX1_Switch8_2, &set_bit[8][2]);
m = GetCtrlVal (daq1, MUX1_Switch8_3, &set_bit[8][3]);
m = GetCtrlVal (daq1, MUX1_Switch8_4, &set_bit[8][4]);
m = GetCtrlVal (daq1, MUX1_Switch8_5, &set_bit[8][5]);
m = GetCtrlVal (daq1, MUX1_Switch8_6, &set_bit[8][6]);
m = GetCtrlVal (daq1, MUX1_Switch8_7, &set_bit[8][7]);
m = GetCtrlVal (daq1, MUX1_Switch8_8, &set_bit[8][8]);
```

```
        m = GetCtrlVal (daq, MUX_TIME_WRITE, &time_write);
        m = GetCtrlVal (daq, MUX_VOLT_WRITE_ON, &volt_write_on);
        m = GetCtrlVal (daq, MUX_VOLT_WRITE_OFF, &volt_write_off);
        m = GetCtrlVal (daq, MUX_VOLT_HOLD, &volt_hold);
        m = GetCtrlVal (daq, MUX_Ramp, &ramp);
        m = GetCtrlVal (daq, MUX_Ramp_Rate, &ramp_rate);

/********   starting the loop of configuring   *******/

    /********  test ********/
        m=SetCtrlVal(daq,MUX_STOP_SCAN,1);
        m=SetCtrlVal(daq,MUX_Config_complete,0);
        m=SetCtrlVal(daq,MUX_Memory_Check_Done,0);
        ibwrt(Device1,"CA72X",5);                                   /* dummy line */
        ibwrt(Device1,"NA72X",5);
        for(i=1;i<=8;i++){
            for(j=1;j<=8;j++){
            if (set_bit[i][j]==1){                                  /* check if the bit is selected   */
                c[0]='C';
                c[1]='B';
                c[2]=(char)(48+i);
                c[3]='X';
                c[4]='\0';
                ibwrt(Device1,c,4);
                c[0]='N';
                c[1]='A';
                c[2]=(char)(48+i);
                c[3]='X';
                c[4]='\0';
                ibwrt(Device1,c,4);
                if (j<2){
                c[0]='C';
                c[1]='C';
                c[2]=(char)(48+j+8);
                c[3]='X';
                c[4]='\0';
                ibwrt(Device1,c,4);
                c[0]='N';
                c[1]='H';
                c[2]=(char)(48+j+8);
                c[3]='X';
                c[4]='\0';
                ibwrt(Device1,c,4);
                }
                else{
                d[0]='C';
                d[1]='C';
                d[2]='1';                                           /* two-digit  */
                d[3]=(char)(48+j-2);
                d[4]='X';
                d[5]='\0';
                ibwrt(Device1,d,5);
                d[0]='N';
                d[1]='H';
                d[2]='1';                                           /* two-digit  */
                d[3]=(char)(48+j-2);
                d[4]='X';
                d[5]='\0';
                ibwrt(Device1,d,5);
                }
            for(k=1;k<=16;k++){
                if((k!=i)&&(k!=j+8)){
                    if (k<10){
                        if (k<=8){
                        c[0]='C';
                        c[1]='A';       /* apply -1.0 volt to rows from Keithley 5-25-01  */
                        c[2]=(char)(48+k);
                        c[3]='X';
                        c[4]='\0';
                        ibwrt(Device1,c,4);
```

```
                                        }
                                        else{
                                        c[0]='C';
                                        c[1]='H';
        /*  Ground the columns */
                                        c[2]=(char)(48+k);
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        }
                                }
                                else{
                                d[0]='C';
                                d[1]='H';
                                d[2]='1';
                                d[3]=(char)(48+k-10);
                                d[4]='X';
                                d[5]='\0';
                                ibwrt(Device1,d,5);
                                }
                        }
                }
                /*  set write voltage */
                Delay(0.1);
                printf("\a");
                if(ramp==1){
                        if(cross_point[i][j]==1){
                                volt_ramp0=volt_hold;
                                volt_ramp1=0.0;
                                for(i_ramp=1; i_ramp<=ramp_num; i_ramp++){
                /* ramp-up */
                                        volt_ramp0=volt_ramp0 + (volt_write_on/2-volt_hold)/ramp_num;
                                        volt_ramp1=volt_ramp1 + (volt_write_on/2)/ramp_num;
                                        m = AO_VWrite (1, 0, volt_ramp0);
                                        m = AO_VWrite (1, 1, (-volt_ramp1-0.06225)/0.9938);
                                        Delay(volt_write_on/ramp_num/ramp_rate);
                                }
                                Delay(time_write);                                      /* hold */
                                for(i_ramp=1; i_ramp<=ramp_num; i_ramp++){
                /* ramp-down */
                                        volt_ramp0=volt_ramp0 - (volt_write_on/2-volt_hold)/ramp_num;
                                        volt_ramp1=volt_ramp1 - (volt_write_on/2)/ramp_num;
                                        m = AO_VWrite (1, 0, volt_ramp0);
                                        m = AO_VWrite (1, 1, (-volt_ramp1-0.06225)/0.9938);
                                        Delay(volt_write_on/ramp_num/ramp_rate);
                                }
                                                                                }
                        else{
                                volt_ramp0=volt_hold;
                                volt_ramp1=0.0;
                                for(i_ramp=1; i_ramp<=ramp_num; i_ramp++){
                /* ramp-up */
                                        volt_ramp0=volt_ramp0 + (volt_write_off/2-
volt_hold)/ramp_num;
                                        volt_ramp1=volt_ramp1 + (volt_write_off/2)/ramp_num;
                                        m = AO_VWrite (1, 0, volt_ramp0);
                                        m = AO_VWrite (1, 1, (-volt_ramp1-0.06225)/0.9938);
                                        Delay(volt_write_off/ramp_num/ramp_rate);
                                }
                                Delay(time_write);                                      /* hold */
                                for(i_ramp=1; i_ramp<=ramp_num; i_ramp++){
                /* ramp-down */
                                        volt_ramp0=volt_ramp0 - (volt_write_off/2-volt_hold)/ramp_num;
                                        volt_ramp1=volt_ramp1 - (volt_write_off/2)/ramp_num;
                                        m = AO_VWrite (1, 0, volt_ramp0);
                                        m = AO_VWrite (1, 1, (-volt_ramp1-0.06225)/0.9938);
                                        Delay(volt_write_off/ramp_num/ramp_rate);
                                }
                        }
                }               /* with ramp */
```

```
                    else{
                        if(cross_point[i][j]==1){
                            m = AO_VWrite (1, 0, (volt_write_on/2));
                            m = AO_VWrite (1, 1, (-volt_write_on/2-0.06225)/0.9938);
                        }
                        else{
                            m = AO_VWrite (1, 0, (volt_write_off/2));
                            m = AO_VWrite (1, 1, (-volt_write_off/2-0.06225)/0.9938);
                        }
                        Delay(time_write);
                        m = AO_VWrite (1, 0, volt_hold);
                        m = AO_VWrite (1, 1, -0.06225/0.9938);
                    }                   /* no ramp */
                /*****  set holding voltage to the row, and Ground to the column  *****/
                        c[0]='C';
                        c[1]='A';
                        c[2]=(char)(48+i);
                        c[3]='X';
                        c[4]='\0';
                        ibwrt(Device1,c,4);
                        c[0]='N';
                        c[1]='B';
                        c[2]=(char)(48+i);
                        c[3]='X';
                        c[4]='\0';
                        ibwrt(Device1,c,4);
                        if (j<2){
                        c[0]='C';
                        c[1]='H';
                        c[2]=(char)(48+j+8);
                        c[3]='X';
                        c[4]='\0';
                        ibwrt(Device1,c,4);
                        c[0]='N';
                        c[1]='C';
                        c[2]=(char)(48+j+8);
                        c[3]='X';
                        c[4]='\0';
                        ibwrt(Device1,c,4);
                        }
                        else{
                        d[0]='C';
                        d[1]='H';
                        d[2]='1';                                       /* two-digit  */
                        d[3]=(char)(48+j-2);
                        d[4]='X';
                        d[5]='\0';
                        ibwrt(Device1,d,5);
                        d[0]='N';
                        d[1]='C';
                        d[2]='1';                                       /* two-digit  */
                        d[3]=(char)(48+j-2);
                        d[4]='X';
                        d[5]='\0';
                        ibwrt(Device1,d,5);
                        }
                    /*ibwrt(Device1,"P0X",3);                   open all relays  5-21-01 */
                }   /*   finish setting one selected bit  */
                }   /* j */
                }
        /* close i loop  */
        m=SetCtrlVal(daq,MUX_Config_complete,1);
        return 1;
        }
int configure (int panel, int control, int event,
                void *callbackData, int eventData1, int eventData2)
        {
        int i,j,k,m,i_ramp;
        char c[5],d[6];
        if (all_control !=1){
```

```
m = GetCtrlVal (daq, MUX_Switch1_1, &cross_point[1][1]);
m = GetCtrlVal (daq, MUX_Switch1_2, &cross_point[1][2]);
m = GetCtrlVal (daq, MUX_Switch1_3, &cross_point[1][3]);
m = GetCtrlVal (daq, MUX_Switch1_4, &cross_point[1][4]);
m = GetCtrlVal (daq, MUX_Switch1_5, &cross_point[1][5]);
m = GetCtrlVal (daq, MUX_Switch1_6, &cross_point[1][6]);
m = GetCtrlVal (daq, MUX_Switch1_7, &cross_point[1][7]);
m = GetCtrlVal (daq, MUX_Switch1_8, &cross_point[1][8]);
m = GetCtrlVal (daq, MUX_Switch2_1, &cross_point[2][1]);
m = GetCtrlVal (daq, MUX_Switch2_2, &cross_point[2][2]);
m = GetCtrlVal (daq, MUX_Switch2_3, &cross_point[2][3]);
m = GetCtrlVal (daq, MUX_Switch2_4, &cross_point[2][4]);
m = GetCtrlVal (daq, MUX_Switch2_5, &cross_point[2][5]);
m = GetCtrlVal (daq, MUX_Switch2_6, &cross_point[2][6]);
m = GetCtrlVal (daq, MUX_Switch2_7, &cross_point[2][7]);
m = GetCtrlVal (daq, MUX_Switch2_8, &cross_point[2][8]);
m = GetCtrlVal (daq, MUX_Switch3_1, &cross_point[3][1]);
m = GetCtrlVal (daq, MUX_Switch3_2, &cross_point[3][2]);
m = GetCtrlVal (daq, MUX_Switch3_3, &cross_point[3][3]);
m = GetCtrlVal (daq, MUX_Switch3_4, &cross_point[3][4]);
m = GetCtrlVal (daq, MUX_Switch3_5, &cross_point[3][5]);
m = GetCtrlVal (daq, MUX_Switch3_6, &cross_point[3][6]);
m = GetCtrlVal (daq, MUX_Switch3_7, &cross_point[3][7]);
m = GetCtrlVal (daq, MUX_Switch3_8, &cross_point[3][8]);
m = GetCtrlVal (daq, MUX_Switch4_1, &cross_point[4][1]);
m = GetCtrlVal (daq, MUX_Switch4_2, &cross_point[4][2]);
m = GetCtrlVal (daq, MUX_Switch4_3, &cross_point[4][3]);
m = GetCtrlVal (daq, MUX_Switch4_4, &cross_point[4][4]);
m = GetCtrlVal (daq, MUX_Switch4_5, &cross_point[4][5]);
m = GetCtrlVal (daq, MUX_Switch4_6, &cross_point[4][6]);
m = GetCtrlVal (daq, MUX_Switch4_7, &cross_point[4][7]);
m = GetCtrlVal (daq, MUX_Switch4_8, &cross_point[4][8]);
m = GetCtrlVal (daq, MUX_Switch5_1, &cross_point[5][1]);
m = GetCtrlVal (daq, MUX_Switch5_2, &cross_point[5][2]);
m = GetCtrlVal (daq, MUX_Switch5_3, &cross_point[5][3]);
m = GetCtrlVal (daq, MUX_Switch5_4, &cross_point[5][4]);
m = GetCtrlVal (daq, MUX_Switch5_5, &cross_point[5][5]);
m = GetCtrlVal (daq, MUX_Switch5_6, &cross_point[5][6]);
m = GetCtrlVal (daq, MUX_Switch5_7, &cross_point[5][7]);
m = GetCtrlVal (daq, MUX_Switch5_8, &cross_point[5][8]);
m = GetCtrlVal (daq, MUX_Switch6_1, &cross_point[6][1]);
m = GetCtrlVal (daq, MUX_Switch6_2, &cross_point[6][2]);
m = GetCtrlVal (daq, MUX_Switch6_3, &cross_point[6][3]);
m = GetCtrlVal (daq, MUX_Switch6_4, &cross_point[6][4]);
m = GetCtrlVal (daq, MUX_Switch6_5, &cross_point[6][5]);
m = GetCtrlVal (daq, MUX_Switch6_6, &cross_point[6][6]);
m = GetCtrlVal (daq, MUX_Switch6_7, &cross_point[6][7]);
m = GetCtrlVal (daq, MUX_Switch6_8, &cross_point[6][8]);
m = GetCtrlVal (daq, MUX_Switch7_1, &cross_point[7][1]);
m = GetCtrlVal (daq, MUX_Switch7_2, &cross_point[7][2]);
m = GetCtrlVal (daq, MUX_Switch7_3, &cross_point[7][3]);
m = GetCtrlVal (daq, MUX_Switch7_4, &cross_point[7][4]);
m = GetCtrlVal (daq, MUX_Switch7_5, &cross_point[7][5]);
m = GetCtrlVal (daq, MUX_Switch7_6, &cross_point[7][6]);
m = GetCtrlVal (daq, MUX_Switch7_7, &cross_point[7][7]);
m = GetCtrlVal (daq, MUX_Switch7_8, &cross_point[7][8]);
m = GetCtrlVal (daq, MUX_Switch8_1, &cross_point[8][1]);
m = GetCtrlVal (daq, MUX_Switch8_2, &cross_point[8][2]);
m = GetCtrlVal (daq, MUX_Switch8_3, &cross_point[8][3]);
m = GetCtrlVal (daq, MUX_Switch8_4, &cross_point[8][4]);
m = GetCtrlVal (daq, MUX_Switch8_5, &cross_point[8][5]);
m = GetCtrlVal (daq, MUX_Switch8_6, &cross_point[8][6]);
m = GetCtrlVal (daq, MUX_Switch8_7, &cross_point[8][7]);
m = GetCtrlVal (daq, MUX_Switch8_8, &cross_point[8][8]);
}
else{
        m = GetCtrlVal (daq, MUX_ALL_SWITCHES, &all_switch);
        for(i=1;i<=8;i++){
                for(j=1;j<=8;j++){
```

```
                                cross_point[i][j]=all_switch;
                                }
                        }
                }
                m = GetCtrlVal (daq, MUX_TIME_WRITE, &time_write);
                m = GetCtrlVal (daq, MUX_VOLT_WRITE_ON, &volt_write_on);
                m = GetCtrlVal (daq, MUX_VOLT_WRITE_OFF, &volt_write_off);
                m = GetCtrlVal (daq, MUX_VOLT_HOLD, &volt_hold);
                m = GetCtrlVal (daq, MUX_Ramp, &ramp);
                m = GetCtrlVal (daq, MUX_Ramp_Rate, &ramp_rate);

/********   starting the loop of configuring   *******/

        /********  test ********/
                m=SetCtrlVal(daq,MUX_STOP_SCAN,1);
                m=SetCtrlVal(daq,MUX_Config_complete,0);
                m=SetCtrlVal(daq,MUX_Memory_Check_Done,0);
                ibwrt(Device1,"CA25X",5);                                       /* dummy line */
                ibwrt(Device1,"NA25X",5);
                for(i=1;i<=8;i++){
                        for(j=1;j<=8;j++){
                        /*              if(cross_point[i][j]==1){                                       2-17-01   */
                                        c[0]='C';
                                        c[1]='B';
                                        c[2]=(char)(48+i);
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        c[0]='N';
                                        c[1]='A';
                                        c[2]=(char)(48+i);
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        if (j<2){
                                        c[0]='C';
                                        c[1]='C';
                                        c[2]=(char)(48+j+8);
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        c[0]='N';
                                        c[1]='H';
                                        c[2]=(char)(48+j+8);
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        }
                                        else{
                                        d[0]='C';
                                        d[1]='C';
                                        d[2]='1';                                               /* two-digit  */
                                        d[3]=(char)(48+j-2);
                                        d[4]='X';
                                        d[5]='\0';
                                        ibwrt(Device1,d,5);
                                        d[0]='N';
                                        d[1]='H';
                                        d[2]='1';                                               /* two-digit  */
                                        d[3]=(char)(48+j-2);
                                        d[4]='X';
                                        d[5]='\0';
                                        ibwrt(Device1,d,5);
                                        }
                        /*              }
                                        else{
                                        c[0]='C';
                                        c[1]='B';
                                        c[2]=(char)(48+i);
                                        c[3]='X';
```

```
                    c[4]='\0';
                    ibwrt(Device1,c,4);
                    if (j<4){
                    c[0]='C';
                    c[1]='A';
                    c[2]=(char)(48+j+6);
                    c[3]='X';
                    c[4]='\0';
                    ibwrt(Device1,c,4);
                    }
                    else{
                    d[0]='C';
                    d[1]='A';
                    d[2]='1';
                    d[3]=(char)(48+j-4);
                    d[4]='X';
                    d[5]='\0';
                    ibwrt(Device1,d,5);
                    }
          }
2-17-01   */
          for(k=1;k<=16;k++){
                    if((k!=i)&&(k!=j+8)){
                              if (k<10){
                                        if (k<=8){
                                        c[0]='C';
                                        c[1]='A';      /* apply -1.0 volt to rows from Keithley 5-25-01  */
                                        c[2]=(char)(48+k);
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        }
                                        else{
                                        c[0]='C';
                                        c[1]='H';
/*  Ground the columns */
                                        c[2]=(char)(48+k);
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        }
                              }
                              else{
                              d[0]='C';
                              d[1]='H';
                              d[2]='1';
                              d[3]=(char)(48+k-10);
                              d[4]='X';
                              d[5]='\0';
                              ibwrt(Device1,d,5);
                              }
                    }
          }
          /*  set write voltage */
          Delay(0.1);
          printf("\a");
          if(ramp==1){
                    if(cross_point[i][j]==1){
                              volt_ramp0=volt_hold;
                              volt_ramp1=0.0;
                              for(i_ramp=1; i_ramp<=ramp_num; i_ramp++){
          /* ramp-up */
                                        volt_ramp0=volt_ramp0 + (volt_write_on/2-volt_hold)/ramp_num;
                                        volt_ramp1=volt_ramp1 + (volt_write_on/2)/ramp_num;
                                        m = AO_VWrite (1, 0, volt_ramp0);
                                        m = AO_VWrite (1, 1, (-volt_ramp1-0.06225)/0.9938);
                                        Delay(volt_write_on/ramp_num/ramp_rate);
                                        Delay(-volt_write_on/ramp_num/ramp_rate);
                              }
```

```
                                        Delay(time_write);                                      /* hold */
                                        for(i_ramp=1; i_ramp<=ramp_num; i_ramp++){
                                /* ramp-down */
                                                volt_ramp0=volt_ramp0 - (volt_write_on/2-volt_hold)/ramp_num;
                                                volt_ramp1=volt_ramp1 - (volt_write_on/2)/ramp_num;
                                                m = AO_VWrite (1, 0, volt_ramp0);
                                                m = AO_VWrite (1, 1, (-volt_ramp1-0.06225)/0.9938);
                                                Delay(volt_write_on/ramp_num/ramp_rate);
                                                Delay(-volt_write_on/ramp_num/ramp_rate);
                                        }
                                }
                                else{
                                        volt_ramp0=volt_hold;
                                        volt_ramp1=0.0;
                                        for(i_ramp=1; i_ramp<=ramp_num; i_ramp++){
                                /* ramp-up */
                                                volt_ramp0=volt_ramp0 + (volt_write_off/2-
volt_hold)/ramp_num;
                                                volt_ramp1=volt_ramp1 + (volt_write_off/2)/ramp_num;
                                                m = AO_VWrite (1, 0, volt_ramp0);
                                                m = AO_VWrite (1, 1, (-volt_ramp1-0.06225)/0.9938);
                                                Delay(volt_write_off/ramp_num/ramp_rate);
                                                Delay(-volt_write_off/ramp_num/ramp_rate);
                                        }
                                        Delay(time_write);                                      /* hold */
                                        for(i_ramp=1; i_ramp<=ramp_num; i_ramp++){
                                /* ramp-down */
                                                volt_ramp0=volt_ramp0 - (volt_write_off/2-volt_hold)/ramp_num;
                                                volt_ramp1=volt_ramp1 - (volt_write_off/2)/ramp_num;
                                                m = AO_VWrite (1, 0, volt_ramp0);
                                                m = AO_VWrite (1, 1, (-volt_ramp1-0.06225)/0.9938);
                                                Delay(volt_write_off/ramp_num/ramp_rate);
                                                Delay(-volt_write_off/ramp_num/ramp_rate);
                                        }
                                }
                        }               /* with ramp */
                        else{
                                if(cross_point[i][j]==1){
                                        m = AO_VWrite (1, 0, (volt_write_on/2));
                                        m = AO_VWrite (1, 1, (-volt_write_on/2-0.06225)/0.9938);
                                }
                                else{
                                        m = AO_VWrite (1, 0, (volt_write_off/2));
                                        m = AO_VWrite (1, 1, (-volt_write_off/2-0.06225)/0.9938);
                                }
                                Delay(time_write);
                                m = AO_VWrite (1, 0, volt_hold);
                                m = AO_VWrite (1, 1, -0.06225/0.9938);
                        }                       /* no ramp */
                /*****  set holding voltage to the row, and Ground to the column  *****/
                                c[0]='C';
                                c[1]='A';
                                c[2]=(char)(48+i);
                                c[3]='X';
                                c[4]='\0';
                                ibwrt(Device1,c,4);
                                c[0]='N';
                                c[1]='B';
                                c[2]=(char)(48+i);
                                c[3]='X';
                                c[4]='\0';
                                ibwrt(Device1,c,4);
                                if (j<2){
                                c[0]='C';
                                c[1]='H';
                                c[2]=(char)(48+j+8);
                                c[3]='X';
                                c[4]='\0';
                                ibwrt(Device1,c,4);
                                c[0]='N';
```

```
                                        c[1]='C';
                                        c[2]=(char)(48+j+8);
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        }
                                        else{
                                        d[0]='C';
                                        d[1]='H';
                                        d[2]='1';                                       /* two-digit  */
                                        d[3]=(char)(48+j-2);
                                        d[4]='X';
                                        d[5]='\0';
                                        ibwrt(Device1,d,5);
                                        d[0]='N';
                                        d[1]='C';
                                        d[2]='1';                                       /* two-digit  */
                                        d[3]=(char)(48+j-2);
                                        d[4]='X';
                                        d[5]='\0';
                                        ibwrt(Device1,d,5);
                                        }
                                /*   ibwrt(Device1,"P0X",3);                    /*  open all relays (skipped  5-25-01) */
                        }
                }
/* close the loop  */
m=SetCtrlVal(daq,MUX_Config_complete,1);
return 1;
}
int logic_check(int panel, int control, int event,
                        void *callbackData, int eventData1, int eventData2){
                /*SetCtrlVal(daq,MUX_STOP_SCAN,1);      */
                return 1;
}
int memory_check(int panel, int control, int event,
                        void *callbackData, int eventData1, int eventData2){
                /*SetCtrlVal(daq,MUX_STOP_SCAN,1);      */
int i,j,k,ii,m;
double r_dummy;
int fail[9][9];
double AD0[9][9][100],AD1[9][9][100];
char c[5],d[6];
                DeleteGraphPlot (daq, MUX_GRAPH, -1, VAL_IMMEDIATE_DRAW);
                m=SetCtrlVal(daq,MUX_Memory_Check_Done,0);
                m=SetCtrlVal(daq,MUX_Set_phase,0);
/*
                m=SetCtrlAttribute(daq,MUX_switch1_1r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch1_2r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch1_3r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch1_4r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch1_5r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch1_6r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch2_1r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch2_2r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch2_3r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch2_4r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch2_5r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch2_6r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch3_1r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch3_2r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch3_3r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch3_4r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch3_5r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch3_6r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch4_1r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch4_2r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch4_3r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch4_4r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch4_5r, ATTR_DIMMED, TRUE);
                m=SetCtrlAttribute(daq,MUX_switch4_6r, ATTR_DIMMED, TRUE);
```

```
m=SetCtrlAttribute(daq,MUX_switch5_1r, ATTR_DIMMED, TRUE);
m=SetCtrlAttribute(daq,MUX_switch5_2r, ATTR_DIMMED, TRUE);
m=SetCtrlAttribute(daq,MUX_switch5_3r, ATTR_DIMMED, TRUE);
m=SetCtrlAttribute(daq,MUX_switch5_4r, ATTR_DIMMED, TRUE);
m=SetCtrlAttribute(daq,MUX_switch5_5r, ATTR_DIMMED, TRUE);
m=SetCtrlAttribute(daq,MUX_switch5_6r, ATTR_DIMMED, TRUE);
m=SetCtrlAttribute(daq,MUX_switch6_1r, ATTR_DIMMED, TRUE);
m=SetCtrlAttribute(daq,MUX_switch6_2r, ATTR_DIMMED, TRUE);
m=SetCtrlAttribute(daq,MUX_switch6_3r, ATTR_DIMMED, TRUE);
m=SetCtrlAttribute(daq,MUX_switch6_4r, ATTR_DIMMED, TRUE);
m=SetCtrlAttribute(daq,MUX_switch6_5r, ATTR_DIMMED, TRUE);
m=SetCtrlAttribute(daq,MUX_switch6_6r, ATTR_DIMMED, TRUE);
12-12-01  LED's removed and kept in an untitled panel    */
        for(i=1;i<=8;i++){
                for(j=1;j<=8;j++){
                PlotLine(daq, MUX_GRAPH, (i-1)*8+(j-1), cross_point[i][j], (i-1)*8+j, cross_point[i][j],VAL_BLUE);
                }
        }
        m = GetCtrlVal (daq, MUX_TIME_READ, &time_read);
        m = GetCtrlVal (daq, MUX_VOLT_READ, &volt_read);
        m = GetCtrlVal (daq, MUX_Threshold_High, &threshold_high);
        m = GetCtrlVal (daq, MUX_Threshold_Low, &threshold_low);
        m = GetCtrlVal (daq, MUX_NUM_READ, &num_read);
        m = GetCtrlVal (daq, MUX_VOLT_HOLD, &volt_hold);
        fp_out=fopen(tmp_file,"w");
/*      Device1=ibdev(0,18,0,10,1,0);                   /* initiate 707A  */
/*      ibwrt(Device1,"E0X",3);                                                 /* Point to present relays  */
        for(i=1;i<=8;i++){
                for(j=1;j<=8;j++){
                                c[0]='C';
                                c[1]='D';               /* use relay row D to read (Vread+AC from function generater)  */
                                2]=(char)(48+i);
                                c[3]='X';
                                c[4]='\0';
                                ibwrt(Device1,c,4);
                                c[0]='N';
                                c[1]='A';
                                c[2]=(char)(48+i);
                                c[3]='X';
                                c[4]='\0';
                                ibwrt(Device1,c,4);
                                if (j<2){
                                c[0]='C';
                                c[1]='G';                                       /* amp-meter  */
                                c[2]=(char)(48+j+8);
                                c[3]='X';
                                c[4]='\0';
                                ibwrt(Device1,c,4);
                                c[0]='N';
                                c[1]='H';                                       /*  GND  */
                                c[2]=(char)(48+j+8);
                                c[3]='X';
                                c[4]='\0';
                                ibwrt(Device1,c,4);
                                }
                                else{
                                d[0]='C';
                                d[1]='G';                                       /* amp-meter  */
                                d[2]='1';
                                d[3]=(char)(48+j-2);
                                d[4]='X';
                                d[5]='\0';
                                ibwrt(Device1,d,5);
                                d[0]='N';
                                d[1]='H';                                       /*  GND  */
                                d[2]='1';
                                d[3]=(char)(48+j-2);
                                d[4]='X';
                                d[5]='\0';
                                ibwrt(Device1,d,5);
```

```
                    }
                for(k=1;k<=16;k++){
                    if((k!=i)&&(k!=j+8)){
                        if (k<10){
                            if (k<=8){
                            c[0]='C';
                            c[1]='A';       /* apply -1.0 volt to rows from Keithley 5-25-01  */
                            c[2]=(char)(48+k);
                            c[3]='X';
                            c[4]='\0';
                            ibwrt(Device1,c,4);
                            }
                            else{
                            c[0]='C';
                            c[1]='H';
/*  Ground the columns */
                            c[2]=(char)(48+k);
                            c[3]='X';
                            c[4]='\0';
                            ibwrt(Device1,c,4);
                            }
                        }
                        else{
                        d[0]='C';
                        d[1]='H';
                        d[2]='1';
                        d[3]=(char)(48+k-10);
                        d[4]='X';
                        d[5]='\0';
                        ibwrt(Device1,d,5);
                        }
                    }
                }

                /*  set read voltage and measure the current  */
                /* Delay (0.1);                                5-25-01  */
                printf("\a");
                m = AO_VWrite (1, 0, volt_read);              /* channel 0's output goes to relay row B directly
                                                              and goes to row D through function generater  */
                Delay (0.1);                                  /* delay after setting the read voltage */
                /*manually set phase on the lock-in  5_28_01  */
                /*
                m=SetCtrlVal(daq,MUX_Set_phase,1);
                scanf("%f",r_dummy);
                m=SetCtrlVal(daq,MUX_Set_phase,0);          taken out for non-volatile devices   6-5-01*/

                for (ii=0;ii<num_read;ii++){
                m = AI_VRead (1, 0, 1, &adch0);                     /* output from current amplifier  */
                m = AI_VRead (1, 1, 1, &adch1);                     /* output from lock-in amplifier   */

                AD0[i][j][ii]=-adch0;                         /****  Current Amplifier revise the polarity!!  *****/
                AD1[i][j][ii]=adch1;

                if(ii>0) m=PlotLine (daq, MUX_GRAPH, 8.0*(i-1)+j-1+(double)(ii-1)/(double)(num_read-1),
AD0[i][j][ii-1], 8.0*(i-1)+j-1+(double)(ii)/(double)(num_read-1), AD0[i][j][ii], VAL_RED);
                if(ii>0) m=PlotLine (daq, MUX_GRAPH, 8.0*(i-1)+j-1+(double)(ii-1)/(double)(num_read-1),
AD1[i][j][ii-1], 8.0*(i-1)+j-1+(double)(ii)/(double)(num_read-1), AD1[i][j][ii], VAL_GREEN);

                Delay (time_read/num_read);
                }
/*****  set holding voltage to the row, and Ground to the column  *****/
                c[0]='C';
                c[1]='A';
                c[2]=(char)(48+i);
                c[3]='X';
                c[4]='\0';
                ibwrt(Device1,c,4);
                c[0]='N';
                c[1]='D';
                c[2]=(char)(48+i);
```

```
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        if (j<2){
                                        c[0]='C';
                                        c[1]='H';
                                        c[2]=(char)(48+j+8);
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        c[0]='N';
                                        c[1]='G';
                                        c[2]=(char)(48+j+8);
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        }
                                        else{
                                        d[0]='C';
                                        d[1]='H';
                                        d[2]='1';                                       /* two-digit  */
                                        d[3]=(char)(48+j-2);
                                        d[4]='X';
                                        d[5]='\0';
                                        ibwrt(Device1,d,5);
                                        d[0]='N';
                                        d[1]='G';
                                        d[2]='1';                                       /* two-digit  */
                                        d[3]=(char)(48+j-2);
                                        d[4]='X';
                                        d[5]='\0';
                                        ibwrt(Device1,d,5);
                                        }
                                fail[i][j]=0;
                                if(cross_point[i][j]==0){
                                        for (ii=0;ii<num_read;ii++){
                                                if(AD0[i][j][ii]>threshold_low) {
                                                        fail[i][j]=1;
                                                        break;
                                                }
                                        }
                                }
                                if(cross_point[i][j]==1){
                                        for (ii=0;ii<num_read;ii++){
                                                if(AD0[i][j][ii]<threshold_high) {
                                                        fail[i][j]=1;
                                                        break;
                                                }
                                        }
                                }
                                /*  m = AO_VWrite (1, 0, 0.0);
                                ibwrt(Device1,"P0X",3);                                                 5-25-01  */
                        }
                }
/* close the loop  */

/*****  set holding voltage to the rows, and Ground to the columns  *****/
                for(k=1;k<=16;k++){
                        if (k<10){
                                if (k<=8){
                                        c[0]='C';
                                        c[1]='A';               /* apply -1.0 volt to rows from Keithley 5-25-01  */
                                        c[2]=(char)(48+k);
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        }
                                        else{
                                        c[0]='C';
                                        c[1]='H';                               /*  Ground the columns */
```

```
                                        c[2]=(char)(48+k);
                                        c[3]='X';
                                        c[4]='\0';
                                        ibwrt(Device1,c,4);
                                        }
                                }
                                else{
                                d[0]='C';
                                d[1]='H';
                                d[2]='1';
                                d[3]=(char)(48+k-10);
                                d[4]='X';
                                d[5]='\0';
                                ibwrt(Device1,d,5);
                                }
                        }
/*
if(fail[1][1]==0) {
        m=SetCtrlAttribute(daq,MUX_switch1_1r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch1_1r,cross_point[1][1]);}
if(fail[1][2]==0) {
        m=SetCtrlAttribute(daq,MUX_switch1_2r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch1_2r,cross_point[1][2]);}
if(fail[1][3]==0) {
        m=SetCtrlAttribute(daq,MUX_switch1_3r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch1_3r,cross_point[1][3]);}
if(fail[1][4]==0) {
        m=SetCtrlAttribute(daq,MUX_switch1_4r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch1_4r,cross_point[1][4]);}
if(fail[1][5]==0) {
        m=SetCtrlAttribute(daq,MUX_switch1_5r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch1_5r,cross_point[1][5]);}
if(fail[1][6]==0) {
        m=SetCtrlAttribute(daq,MUX_switch1_6r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch1_6r,cross_point[1][6]);}
if(fail[2][1]==0) {
        m=SetCtrlAttribute(daq,MUX_switch2_1r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch2_1r,cross_point[2][1]);}
if(fail[2][2]==0) {
        m=SetCtrlAttribute(daq,MUX_switch2_2r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch2_2r,cross_point[2][2]);}
if(fail[2][3]==0) {
        m=SetCtrlAttribute(daq,MUX_switch2_3r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch2_3r,cross_point[2][3]);}
if(fail[2][4]==0) {
        m=SetCtrlAttribute(daq,MUX_switch2_4r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch2_4r,cross_point[2][4]);}
if(fail[2][5]==0) {
        m=SetCtrlAttribute(daq,MUX_switch2_5r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch2_5r,cross_point[2][5]);}
if(fail[2][6]==0) {
        m=SetCtrlAttribute(daq,MUX_switch2_6r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch2_6r,cross_point[2][6]);}
if(fail[3][1]==0) {
        m=SetCtrlAttribute(daq,MUX_switch3_1r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch3_1r,cross_point[3][1]);}
if(fail[3][2]==0) {
        m=SetCtrlAttribute(daq,MUX_switch3_2r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch3_2r,cross_point[3][2]);}
if(fail[3][3]==0) {
        m=SetCtrlAttribute(daq,MUX_switch3_3r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch3_3r,cross_point[3][3]);}
if(fail[3][4]==0) {
        m=SetCtrlAttribute(daq,MUX_switch3_4r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch3_4r,cross_point[3][4]);}
if(fail[3][5]==0) {
        m=SetCtrlAttribute(daq,MUX_switch3_5r, ATTR_DIMMED, FALSE);
        m=SetCtrlVal(daq,MUX_switch3_5r,cross_point[3][5]);}
if(fail[3][6]==0) {
        m=SetCtrlAttribute(daq,MUX_switch3_6r, ATTR_DIMMED, FALSE);
```

```
		m=SetCtrlVal(daq,MUX_switch3_6r,cross_point[3][6]);}
if(fail[4][1]==0) {
		m=SetCtrlAttribute(daq,MUX_switch4_1r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch4_1r,cross_point[4][1]);}
if(fail[4][2]==0) {
		m=SetCtrlAttribute(daq,MUX_switch4_2r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch4_2r,cross_point[4][2]);}
if(fail[4][3]==0) {
		m=SetCtrlAttribute(daq,MUX_switch4_3r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch4_3r,cross_point[4][3]);}
if(fail[4][4]==0) {
		m=SetCtrlAttribute(daq,MUX_switch4_4r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch4_4r,cross_point[4][4]);}
if(fail[4][5]==0) {
		m=SetCtrlAttribute(daq,MUX_switch4_5r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch4_5r,cross_point[4][5]);}
if(fail[4][6]==0) {
		m=SetCtrlAttribute(daq,MUX_switch4_6r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch4_6r,cross_point[4][6]);}

if(fail[5][1]==0) {
		m=SetCtrlAttribute(daq,MUX_switch5_1r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch5_1r,cross_point[5][1]);}
if(fail[5][2]==0) {
		m=SetCtrlAttribute(daq,MUX_switch5_2r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch5_2r,cross_point[5][2]);}
if(fail[5][3]==0) {
		m=SetCtrlAttribute(daq,MUX_switch5_3r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch5_3r,cross_point[5][3]);}
if(fail[5][4]==0) {
		m=SetCtrlAttribute(daq,MUX_switch5_4r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch5_4r,cross_point[5][4]);}
if(fail[5][5]==0) {
		m=SetCtrlAttribute(daq,MUX_switch5_5r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch5_5r,cross_point[5][5]);}
if(fail[5][6]==0) {
		m=SetCtrlAttribute(daq,MUX_switch5_6r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch5_6r,cross_point[5][6]);}
if(fail[6][1]==0) {
		m=SetCtrlAttribute(daq,MUX_switch6_1r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch6_1r,cross_point[6][1]);}
if(fail[6][2]==0) {
		m=SetCtrlAttribute(daq,MUX_switch6_2r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch6_2r,cross_point[6][2]);}
if(fail[6][3]==0) {
		m=SetCtrlAttribute(daq,MUX_switch6_3r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch6_3r,cross_point[6][3]);}
if(fail[6][4]==0) {
		m=SetCtrlAttribute(daq,MUX_switch6_4r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch6_4r,cross_point[6][4]);}
if(fail[6][5]==0) {
		m=SetCtrlAttribute(daq,MUX_switch6_5r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch6_5r,cross_point[6][5]);}
if(fail[6][6]==0) {
		m=SetCtrlAttribute(daq,MUX_switch6_6r, ATTR_DIMMED, FALSE);
		m=SetCtrlVal(daq,MUX_switch6_6r,cross_point[6][6]);}

  12-12-01    taken out, because the LED's are removed*/

m=SetCtrlVal(daq,MUX_Memory_Check_Done,1);
		for(i=1;i<=8;i++){
			for(j=1;j<=8;j++){
				for (k=0;k<num_read;k++){
					fprintf(fp_out, "%d %d %d %f %f\n", i, j, cross_point[i][j], AD0[i][j][k], AD1[i][j][k]);
					}
				}
			}
		fclose(fp_out);
		return 1;
}
```

```
int stop(int panel, int control, int event,
		void *callbackData, int eventData1, int eventData2){
	return 1;
}
int save_file(int panel, int control, int event,
		void *callbackData, int eventData1, int eventData2){
	int i;
	int tmp1[6400],tmp2[6400],tmp3[6400];
	float tmp4[6400], tmp5[6400];
	char line[100];
	char name[30];
	fp_out=fopen (tmp_file,"r");
	for (i = 0; i < num_read*64; ++i)
		{
		fgets(line,sizeof(line),fp_out);
		sscanf(line,"%d %d %d %f %f", &tmp1[i], &tmp2[i], &tmp3[i], &tmp4[i], &tmp5[i]);
		}
	 fclose(fp_out);
	PromptPopup ("SAVE FILE", "Enter the file name (*.txt).", name, 20);
	fp_out=fopen(name,"w");
	for (i =0; i < num_read*64; ++i)
		fprintf(fp_out,"%d %f %f\n",tmp3[i], tmp4[i], tmp5[i]);
	fclose(fp_out);
	return 1;
}
int quit(int panel, int control, int event,
		void *callbackData, int eventData1, int eventData2)
{
	int i;
	switch (event) {
		case EVENT_COMMIT:
			i = AO_VWrite (1, 0, 0.0);
			i = AO_VWrite (1, 1, 0.0);
			ibwrt(Device1,"P0X",3);
			QuitUserInterface (0);
			break;
		case EVENT_RIGHT_CLICK:
			break;
	}
	return 0;
}
/*
int load_individual_panel (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
	daq1 = LoadPanel (0, "MUX.uir",SET_INDIVI);
	DisplayPanel (daq1);
	return 0;
}
*/
int clear (int panel, int control, int event,
		void *callbackData, int eventData1, int eventData2)
{
	int i;
	switch (event) {
		case EVENT_COMMIT:
			DeleteGraphPlot (daq, MUX_GRAPH, -1, VAL_IMMEDIATE_DRAW);
/*			DeleteGraphPlot (daq, DAQ_GRAPH_2, -1, VAL_IMMEDIATE_DRAW);		*/
			break;
	}
	return 0;
}
```