

```

1  /*-----*/
2  /* Company : ADLINK */
3  /* Date : 2008/07/30 */
4  /* */
5  /* This sample performs infinite AI acquisition by double buffer mode. */
6  /* There are 4(Differetial Mode) AI channels in USB-1210. */
7  /* You can use polling mode or DMA to acquired data from specified */
8  /* channels. */
9  /*-----*/
10 #include <windows.h>
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <conio.h>
14 #include <time.h>
15
16 #include "UsbDask.h"
17
18 #define AI_COUNT 204800
19 #define CHANNELCOUNT 1
20
21 #define INVALID_CARD_ID 0xFFFF
22
23 int main(int argc, char **argv)
24 {
25     I16 card, err;
26     U16 card_num;
27     U16 wModuleNum;
28     USBDAQ_DEVICE AvailModules[MAX_USB_DEVICE];
29
30     U16 ConfigCtrl = UD_AI_Differential;
31     U32 TriggerLevel = 0;
32     U32 ReTriggerCount = 0;
33     U32 PreTrigCount = 0;
34     U32 DelayCount = 0;
35     U32 AI_ReadCount = AI_COUNT; /*AI read count per one buffer*/
36     F64 fSampleRate = 200000.0f;
37     BOOLEAN Stopped;
38     BOOLEAN HalfReady;
39     U32 AccessCnt = 0;
40     U16 BufIdx = 0;
41     U8 FileName[] = "ai_data";
42     U16 i;
43     /*-----*/
44
45     printf("This sample performs Oneshot AI acquisition from AI0\n");
46     printf("at %6.3lf Hz scan rate.\n\n", fSampleRate);
47
48     err = UD_Device_Scan(&wModuleNum, AvailModules );
49     if(err<0)
50     {
51         printf("UD_Device_Scan Error: %d\n", err);
52         exit(1);
53     }
54
55     card_num = INVALID_CARD_ID;
56
57     for( i = 0; i < wModuleNum; i++ )
58     {
59         if( AvailModules[i].wModuleType == USB_1210 )
60         {
61             card_num = AvailModules[i].wCardID;
62             break;
63         }
64     }
65
66     if( card_num == INVALID_CARD_ID )
67     {
68         printf("No active USB_1210 USB module\n");
69         exit(2);
70     }
71
72     card = UD_Register_Card(USB_1210, card_num);
73     if(card<0)

```

```

74     {
75         printf("UD_Register_Card Error: %d\n", card);
76         exit(3);
77     }
78
79     /*Configure AI*/
80     err = UD_AI_Channel_Config( card, UD_AI_Differential, UD_AI_Differential,
81     UD_AI_Differential, UD_AI_Differential );
82     if (err != NoError)
83     {
84         printf("UD_AI_Channel_Config Error: %d\n", err);
85         exit(1);
86     }
87
88     err = UD_AI_Trigger_Config( card, UD_AI_CONVSRC_INT, UD_AI_TRGMOD_POST,
89     UD_AI_TRGSRC_SOFT, ReTriggerCount, PreTrigCount, DelayCount, TriggerLevel );
90     if (err != NoError)
91     {
92         printf("UD_AI_Trigger_Config Error: %d\n", err);
93         exit(1);
94     }
95
96     /*Disable Double Buffer Mode*/
97     err = UD_AI_AsyncDblBufferMode(card, 0); // double-buffer mode
98     if (err != NoError)
99     {
100         printf("UD_AI_AsyncDblBufferMode Error: %d\n", err);
101         exit(1);
102     }
103
104     /*AI Acquisition Start*/
105     err = UD_AI_ContReadChannelToFile(card, 0, AD_B_10_V, FileName, AI_ReadCount,
106     fSampleRate, ASYNCH_OP);
107     if (err != NoError)
108     {
109         printf("AI_ContReadMultiChannels Error: %d\n", err);
110         UD_Release_Card(card);
111         exit(1);
112     }
113
114     do{
115         /*Check Buffer Ready*/
116         err = UD_AI_AsyncCheck(card, &Stopped, &AccessCnt);
117         if(err<0)
118         {
119             printf("UD_AI_AsyncCheck Error: %d\n", err);
120             UD_AI_AsyncClear(card, &AccessCnt);
121             UD_Release_Card(card);
122             exit(1);
123         }
124         else
125         {
126             if( Stopped == TRUE )
127             {
128                 printf("\nLast %d samples had been written to %s.dat file...\n",
129                 AccessCnt, FileName);
130                 break;
131             }
132         }
133     }while( !kbhit() );
134
135     /*Clear AI Setting and Get Remaining data*/
136     err = UD_AI_AsyncClear(card, &AccessCnt);
137     if (err != NoError)
138     {
139         printf("AI_AsyncClear Error: %d\n", err);
140         UD_Release_Card(card);
141         exit(1);
142     }
143
144     UD_Release_Card(card);

```

```
143
144     printf("                Press any key to exit...\n");
145     getch();
146     return 0;
147 }
148
```