

```

1  /*
2  *  main.c
3  *
4  *  =====
5  *  Copyright (c) Texas Instruments Inc 2005
6  *
7  *  Use of this software is controlled by the terms and conditions found in the
8  *  license agreement under which this software has been supplied or provided.
9  *  =====
10 */
11
12 // Standard Linux headers
13 #include    <stdio.h>                // Always include this header
14 #include    <stdlib.h>               // Always include this header
15 #include    <signal.h>               // Defines signal-handling functions (i.e. trap
Ctrl-C)
16
17
18 // Application headers
19 #include    "debug.h"
20 #include    "audio_thread.h"
21
22 // Global audio thread environment
23 audio_thread_env audio_env = {0};
24
25 /* Store previous signal handler and call it */
26 void (*pSigPrev)(int sig);
27
28 // Callback called when SIGINT is sent to the process (Ctrl-C)
29 void signal_handler(int sig)
30 {
31     DBG( "Ctrl-C pressed, cleaning up and exiting..\n" );
32     audio_env.quit = 1;
33
34     if( pSigPrev != NULL )
35         (*pSigPrev)( sig );
36 }
37
38
39 //*****
40 /*  main
41 //*****
42 int main( int argc, char *argv[] )
43 {
44     int    status = EXIT_SUCCESS;
45
46     void *audioThreadReturn;
47
48
49     // Set the signal callback for Ctrl-C
50     pSigPrev = signal(SIGINT, signal_handler);
51
52     // Call audio thread function
53     audioThreadReturn = audio_thread_fxn( (void *) &audio_env );

```

```
54
55     if( audioThreadReturn == AUDIO_THREAD_FAILURE )
56     {
57         DBG( "Audio thread exited with FAILURE status\n" );
58         status = EXIT_FAILURE;
59     }
60     else
61         DBG( "Audio thread exited with SUCCESS status\n" );
62
63     exit( status );
64 }
65
66
```