```c
#include <stdlib.h>
#include <stdio.h>

#include "thinkgear.h"

/**
 * Prompts and waits for the user to press ENTER.
 */
void
wait() {
    printf( "\n" );
    printf( "Press the ENTER key...\n" );
    fflush( stdout );
    getc( stdin );
}

/**
 * Program which prints ThinkGear EEG_POWERS values to stdout.
 */
int
main( void ) {

    char *comPortName = NULL;
    int   dllVersion = 0;
    int   connectionId = 0;
    int   packetsRead = 0;
    int   errCode = 0;

    /* Print driver version number */
    dllVersion = TG_GetDriverVersion();
    printf( "ThinkGear DLL version: %d\n", dllVersion );

    /* Get a connection ID handle to ThinkGear */
    connectionId = TG_GetNewConnectionId();
    if( connectionId < 0 ) {
        fprintf( stderr, "ERROR: TG_GetNewConnectionId() returned %d.\n",
                 connectionId );
        wait();
        exit( EXIT_FAILURE );
    }

    /* Set/open stream (raw bytes) log file for connection */
    errCode = TG_SetStreamLog( connectionId, "streamLog.txt" );
    if( errCode < 0 ) {
        fprintf( stderr, "ERROR: TG_SetStreamLog() returned %d.\n", errCode );
        wait();
        exit( EXIT_FAILURE );
    }

    /* Set/open data (ThinkGear values) log file for connection */
    errCode = TG_SetDataLog( connectionId, "dataLog.txt" );
    if( errCode < 0 ) {
        fprintf( stderr, "ERROR: TG_SetDataLog() returned %d.\n", errCode );
        wait();
        exit( EXIT_FAILURE );
    }

    /* Attempt to connect the connection ID handle to serial port "COM5" */
    comPortName = "\\\\.\\COM6";
    errCode = TG_Connect( connectionId,
                          comPortName,
                          TG_BAUD_9600,
                          TG_STREAM_PACKETS );
    if( errCode < 0 ) {
        fprintf( stderr, "ERROR: TG_Connect() returned %d.\n", errCode );
        wait();
        exit( EXIT_FAILURE );
```

```
68      }
69
70      /* Read 10 ThinkGear Packets from the connection, 1 Packet at a time */
71      packetsRead = 0;
72      while( packetsRead < 10 ) {
73
74          /* Attempt to read a Packet of data from the connection */
75          errCode = TG_ReadPackets( connectionId, -1 );
76
77          /* If TG_ReadPackets() was able to read a complete Packet of data... */
78          if( errCode == 1 ) {
79              packetsRead++;
80
81              /* If attention value has been updated by TG_ReadPackets()... */
82              //if( TG_GetValueStatus(connectionId, TG_DATA_ATTENTION) != 0 ) {
83              //    /* Get and print out the updated attention value */
84              //    fprintf( stdout, "New attention value: %d\n",
85              //             TG_GetValue(connectionId, TG_DATA_ATTENTION) );
86
87              //    fflush( stdout );
88
89              //} /* end "If attention value has been updated..." */
90              /*if( TG_GetValueStatus(connectionId, TG_DATA_ATTENTION) != 0 ) {
91                  fprintf( stdout, "New TG_DATA_BATTERY value: %f\n",
92                               TG_GetValue(connectionId, TG_DATA_BATTERY) );
93                  fflush( stdout );
94              }
95
96              if( TG_GetValueStatus(connectionId, TG_DATA_POOR_SIGNAL) != 0 ) {
97                  fprintf( stdout, "New TG_DATA_POOR_SIGNAL value: %f\n",
98                               TG_GetValue(connectionId, TG_DATA_POOR_SIGNAL) );
99                  fflush( stdout );
100             }*/
101
102             if( TG_GetValueStatus(connectionId, TG_DATA_ATTENTION) != 0 ) {
103                 fprintf( stdout, "New attention value: %f\n",
104                              TG_GetValue(connectionId, TG_DATA_ATTENTION) );
105                 fflush( stdout );
106             }
107
108             if( TG_GetValueStatus(connectionId, TG_DATA_MEDITATION) != 0 ) {
109                 fprintf( stdout, "New TG_DATA_MEDITATION value: %f\n",
110                              TG_GetValue(connectionId, TG_DATA_MEDITATION) );
111                 fflush( stdout );
112             }
113
114             if( TG_GetValueStatus(connectionId, TG_DATA_RAW) != 0 ) {
115                 fprintf( stdout, "New TG_DATA_RAW value: %f\n",
116                              TG_GetValue(connectionId, TG_DATA_RAW) );
117                 fflush( stdout );
118             }
119
120             if( TG_GetValueStatus(connectionId, TG_DATA_DELTA) != 0 ) {
121                 fprintf( stdout, "New TG_DATA_DELTA value: %f\n",
122                              TG_GetValue(connectionId, TG_DATA_DELTA) );
123                 fflush( stdout );
124             }
125
126             if( TG_GetValueStatus(connectionId, TG_DATA_THETA) != 0 ) {
127                 fprintf( stdout, "New TG_DATA_THETA value: %f\n",
128                              TG_GetValue(connectionId, TG_DATA_THETA) );
129                 fflush( stdout );
130             }
131
132             if( TG_GetValueStatus(connectionId, TG_DATA_ALPHA1) != 0 ) {
133                 fprintf( stdout, "New TG_DATA_ALPHA1 value: %f\n",
134                              TG_GetValue(connectionId, TG_DATA_ALPHA1) );
```

```
135                 fflush( stdout );
136             }
137 
138             if( TG_GetValueStatus(connectionId, TG_DATA_ALPHA2) != 0 ) {
139                 fprintf( stdout, "New TG_DATA_ALPHA2 value: %f\n",
140                             TG_GetValue(connectionId, TG_DATA_ALPHA2) );
141                 fflush( stdout );
142             }
143 
144             if( TG_GetValueStatus(connectionId, TG_DATA_BETA1) != 0 ) {
145                 fprintf( stdout, "New TG_DATA_BETA1 value: %f\n",
146                             TG_GetValue(connectionId, TG_DATA_BETA1) );
147                 fflush( stdout );
148             }
149 
150             if( TG_GetValueStatus(connectionId, TG_DATA_BETA2) != 0 ) {
151                 fprintf( stdout, "New TG_DATA_BETA2 value: %f\n",
152                             TG_GetValue(connectionId, TG_DATA_BETA2) );
153                 fflush( stdout );
154             }
155 
156             if( TG_GetValueStatus(connectionId, TG_DATA_GAMMA1) != 0 ) {
157                 fprintf( stdout, "New TG_DATA_GAMMA1 value: %f\n",
158                             TG_GetValue(connectionId, TG_DATA_ALPHA2) );
159                 fflush( stdout );
160             }
161 
162             if( TG_GetValueStatus(connectionId, TG_DATA_GAMMA2) != 0 ) {
163                 fprintf( stdout, "New TG_DATA_GAMMA2 value: %f\n",
164                             TG_GetValue(connectionId, TG_DATA_GAMMA2) );
165                 fflush( stdout );
166             }
167 
168 
169         } /* end "If a Packet of data was read..." */
170 
171     } /* end "Read 10 Packets of data from connection..." */
172 
173     /* Clean up */
174     TG_FreeConnection( connectionId );
175 
176     /* End program */
177     wait();
178     return( EXIT_SUCCESS );
179 }
```