

```
1 #include "tgLink.h"
2 #include "thinkgear.h"
3 #include "console/simBase.h"
4 #include "console/console.h"
5 #include "console/consoleInternal.h"
6
7 IMPLEMENT_CONOBJECT(TGLink);
8
9 void TGLink::initPersistFields()
10 {
11     Parent::initPersistFields();
12 };
13
14 int TGLink::driverVersion()
15 {
16     return TG_GetDriverVersion();
17 };
18
19 // just direct transfer of test app c++ program to torque3d engine
20 ConsoleMethod( TGLink, connect, void, 2, 2, "Creates a connection." )
21 {
22     char *comPortName = NULL;
23     int connectionId = 0;
24     int errCode = 0;
25
26     /* Get a connection ID handle to ThinkGear */
27     connectionId = TG_GetNewConnectionId();
28     if( connectionId < 0 )
29     {
30         Con::errorf( "ERROR: TG_GetNewConnectionId() returned %d.\n", connectionId );
31         exit( 1 );
32     }
33     /* Attempt to connect the connection ID handle to serial port "COM5" */
34     comPortName = "\\\\.\\COM6";
35     errCode = TG_Connect( connectionId,
36                          comPortName,
37                          TG_BAUD_9600,
38                          TG_STREAM_PACKETS );
39
40     if( errCode < 0 )
41     {
42         Con::errorf( "ERROR: TG_Connect() returned %d.\n", errCode );
43         exit( 1 );
44     }
45     else Con::errorf( "Connection successful returned %d.\n", connectionId );
46
47     int packetsRead = 0;
48     while( packetsRead < 10 ) {
49         errCode = TG_ReadPackets( connectionId, -1 );
50
51         if( errCode == 1 ) {
52             Con::errorf( "%d", ++packetsRead );
53             if( TG_GetValueStatus( connectionId, TG_DATA_MEDITATION ) != 0 ) {
54                 Con::errorf( "New meditation value: %f\n", TG_GetValue( connectionId,
55                                     TG_DATA_MEDITATION ) );
56             }
57             if( TG_GetValueStatus( connectionId, TG_DATA_ATTENTION ) != 0 ) {
58                 Con::errorf( "New attention value: %f\n", TG_GetValue( connectionId,
59                                     TG_DATA_ATTENTION ) );
60             }
61             if( TG_GetValueStatus( connectionId, TG_DATA_RAW ) != 0 ) {
62                 Con::errorf( "New TG_DATA_RAW value: %f\n", TG_GetValue( connectionId,
63                                     TG_DATA_RAW ) );
64             }
65             if( TG_GetValueStatus( connectionId, TG_DATA_DELTA ) != 0 ) {
66                 Con::errorf( "New TG_DATA_DELTA value: %f\n",
67 
```

```
64         TG_GetValue(connectionId, TG_DATA_DELTA) );
65
66     }
67
68     if( TG_GetValueStatus(connectionId, TG_DATA_THETA) != 0 ) {
69         Con::errorf( "New TG_DATA_THETA value: %f\n",
70                     TG_GetValue(connectionId, TG_DATA_THETA) );
71
72     }
73
74     if( TG_GetValueStatus(connectionId, TG_DATA_ALPHA1) != 0 ) {
75         Con::errorf( "New TG_DATA_ALPHA1 value: %f\n",
76                     TG_GetValue(connectionId, TG_DATA_ALPHA1) );
77
78     }
79
80     if( TG_GetValueStatus(connectionId, TG_DATA_ALPHA2) != 0 ) {
81         Con::errorf( "New TG_DATA_ALPHA2 value: %f\n",
82                     TG_GetValue(connectionId, TG_DATA_ALPHA2) );
83
84     }
85
86     if( TG_GetValueStatus(connectionId, TG_DATA_BETA1) != 0 ) {
87         Con::errorf( "New TG_DATA_BETA1 value: %f\n",
88                     TG_GetValue(connectionId, TG_DATA_BETA1) );
89
90     }
91
92     if( TG_GetValueStatus(connectionId, TG_DATA_BETA2) != 0 ) {
93         Con::errorf( "New TG_DATA_BETA2 value: %f\n",
94                     TG_GetValue(connectionId, TG_DATA_BETA2) );
95
96     }
97
98     if( TG_GetValueStatus(connectionId, TG_DATA_GAMMA1) != 0 ) {
99         Con::errorf( "New TG_DATA_GAMMA1 value: %f\n",
100                    TG_GetValue(connectionId, TG_DATA_ALPHA2) );
101
102     }
103
104     if( TG_GetValueStatus(connectionId, TG_DATA_GAMMA2) != 0 ) {
105         Con::errorf( "New TG_DATA_GAMMA2 value: %f\n",
106                     TG_GetValue(connectionId, TG_DATA_GAMMA2) );
107     }
108 }
109
110 }
111
112 TG_FreeConnection( connectionId );
113 };
114
115
116
117 ConsoleFunction( tg, void, 1, 1, "returns driver number for ThinkGear." )
118 {
119     TGLink* tgObject = new TGLink;
120     tgObject->registerObject( "thinkgear" );
121     Con::errorf( tgObject->getName() );
122 };
123
124 TGLink::~TGLink()
125 {
126     disconnectHeadset();
127 };
```