

测试例程：读取无线网络注册节点列表

```
#include <stdio.h>
#include <Winsock2.h>
#include "API-WiMinet.h"
// *****
// Design Notes:
// -----
char Print_WiMode_Address( void )
{
    char iRetVal;
    unsigned char iMode;
    // Get the network mode
    iRetVal = GetNetworkWiMode( 0X00, 0X00, &iMode );

    // Validate the shell operation status
    if ( !iRetVal )
    {
        printf( "GetNetworkWiMode failed!\r\n" );
        return 0X00;
    }
    // The node network mode
    printf( "[0] The Node Mode is:" );

    // The WiMode status
    switch ( iMode )
    {
    case WIMODE_NETCLIENT:
        {
            printf( "Client\r\n" );
        }
        break;

    case WIMODE_NETSERVER:
        {
            printf( "Server\r\n" );
        }
        break;

    case WIMODE_AUTO_TREE:
        {
            printf( "Router\r\n" );
        }
        break;

    default:
        {
        }
        break;
    }
}
```

```

    return 0X01;
}
// *****
// Design Notes:
// -----
char Print_X16NET_Address( void )
{
    char iRetVal;
    unsigned short iAddr;

    // Get the X16 NET address
    iRetVal = GetX16NETAddress( 0X00, 0X00, &iAddr );

    // Validate the shell operation status
    if ( !iRetVal )
    {
        printf( "GetX16NETAddress failed!\r\n" );
        return 0X00;
    }

    // The X16NET address
    printf( "[1] X16NET Address=0X%04X\r\n", iAddr );
    return 0X01;
}
// *****
// Design Notes:
// -----
char Print_X64MAC_Address( void )
{
    char iRetVal;
    char pX64MAC[0X08];
    unsigned char index;
    // Get the X64MAC address
    iRetVal = GetX64MACAddress( 0X00, 0X00, pX64MAC, sizeof( pX64MAC ) );

    // Validate the shell operation status
    if ( !iRetVal )
    {
        printf( "GetX64MACAddress failed!\r\n" );
        return 0X00;
    }

    // The X64MAC header
    printf( "[2] X64MAC Address=" );

    // The X64MAC address
    for ( index = 0X00; index < 0X08; index++ )
    {
        printf( "%02X", ( unsigned char )pX64MAC[index] );
    }
}

```

```

}
printf( "\r\n" );
return 0X01;
}
// *****
// Design Notes:
// -----
char Print_Register_Table( void )
{
    unsigned char  iRetVal;
    unsigned char  iStep;
    unsigned short index;
    unsigned short iCount;
    WiMinet_MeshItem item;

    // The max registry count
    iRetVal = GetRegistryCount( 0X00, &iCount );
    // Validate the shell operation status
    if ( !iRetVal )
    {
        printf( "GetRegistryCount failed!\r\n" );
        return 0X00;
    }

    // The max registry count
    printf( "[3] Max Registry Count is %u\r\n", iCount );
    // The registry header
    printf( "    NO. Address Mode Tree Parent Device WakeUp X64MAC\r\n" );
    // Get all the item in this table
    for ( index = 0X00; index < iCount; index++ )
    {
        // Get the mesh item from the registry table
        GetRegistryTable( 0X00, index, &item );
        // Validate the node address
        if ( !item.m_Address.m_iTxAddr )
        {
            break;
        }
        // The device index
        printf( "    [%d] ", index );
        // The node information
        printf( " 0X%04X", item.m_Address.m_iTxAddr );

        // The node mode
        printf( " 0X%02X", item.m_Address.m_iTxMode );

        // The tree level
        printf( " 0X%02X", item.m_Address.m_iTxTree );

        // The parent information

```

```

printf( " 0X%04X", item.m_RunTime.m_iParent );

// The class information
printf( "    %u    ", item.m_Address.m_iDevice );
// The eW0R device status
printf( "    %u    ", item.m_Address.m_iWakeUp );
// The X64MAC address
for ( iStep = 0X00; iStep < 0X08; iStep++ )
{
    printf( "%02X", ( unsigned char )item.m_Address.m_pX64MAC[iStep] );
}
// The __I64 format has a revered byte order
// printf( "    %I64X", *( ( unsigned __int64 * )item.m_Address.m_pX64MAC ) );
// End of this item
printf( "\r\n" );
}
return 0X01;
}
// *****
// Design Notes:
// -----
int main( int argc, char* argv[] )
{
    char iRetVal;

    // COM port interface
    //iRetVal = OpenWiMinetShell( "COM6",115200, 0X01 );
    // Ethernet interface
    iRetVal = OpenWiMinetShell( "192.168.0.240",12580, 0X01 );
    // Validate the shell open interface
    if ( !iRetVal )
    {
        printf( "Open shell failed!\r\n" );
        return 0X00;
    }
    // The WiMode status
    Print_WiMode_Address();
    // The X16NET address
    Print_X16NET_Address();
    // The X64MAC address
    Print_X64MAC_Address();
    // The Register Table
    Print_Register_Table();

    // Stop the shell
    StopWiMinetShell( 0X00 );

    // Exit this main program
    return 0X01;
}

```