**Code Walkthrough**

Within the source file, there are a few global variables that are used:

* char cmd[12] – Holds the command string.
* char param[50] – Holds any parameters that need to be sent.
* char msgUART[DATATXRX] – Holds the data to be sent to the Nimbelink.
* char recvdMsg[DATATXRX] – Holds the data received from the Nimbelink.
* char msgParse[6][60] – Holds strings that were parsed from recvdMsg.
* char infoParse[6][20] – Holds strings that were parsed from msgParse.
* char \* nltoken – New line token.
* char \* spctoken – Space token.
* char \* cmatoken – Comma token.
* char reply\_num[13] – Holds the reply phone number from the SMS mailbox.
* char sckt[2] – Holds the socket number.
* hostData GS – Declares a structure to hold host IP address and socket number.

There are only 3 function calls in the main.c file:

* First function call is Nimbelink\_Start(). This function does three things:
  + Typed commands on the Nimbelink UART TX line will echo back on the UART RX line (this is also seen when typing commands to the device through PuTTy). This setting is a default setting which can be changed using “ATE0”. This disables the echo to the UART so that we only receive the response to the command.
  + Checks the network registration to make sure that the module is registered on the network and able to interact with the cellular network. This is done with the “AT+CREG?” command. The response to the command is “CREG: \*\*\*”, where “\*\*\*” will be the response code. If the code is either “0,1” or “0,5”, it means that there was successful registration.
  + Changes the message format for SMS messaging using the “AT+CMGF=” command, with an additional parameter of 1.
* Second function call is Nimbelink\_ConnectGS( char \* port , char \* IP )
  + Checks the status of all sockets on the Nimbelink. If any socket has an active connection, it is shut down and closed.
  + Checks to see if context has been activated to allow a data connection via TCP/IP or UDP. If it has not been activated, it will activate it and return a confirmation along with the IP address of the Nimbelink.
  + Sends a command to the Nimbelink to start a TCP/IP connection to the IP address and port provided in the function argument. It will send a confirmation on whether or not it succeeded.
* Third function call is Nimbelink\_GSDataXfer( uint8 sckt , char \* data )
  + Checks if there are any SMS messages to be read.
  + Checks if there is any data to be read from the socket.
  + Sends data through a socket as specified in the function arguments.