According to the [CC3000](http://www.ti.com/product/CC3000) Host Programming Guide, "The SPI protocol is used to communicate with the [CC3000](http://www.ti.com/product/CC3000) device from the host MCU that is acting as the SPI master while the [CC3000](http://www.ti.com/product/CC3000) device is the SPI slave. The protocol is an extension of the existing standard SPI. The endianness on transport is assumed to be most-significant bit (MSB) first."  That is the bit-endianness of the SPI implementation.  However, no mention is made of the byte-endianness.

The documentation should be updated to reflect that the protocol is implemented as Least Significant Byte First, when it comes to sending commands and arguments greater than 8 bits.  So, yes, each byte is sent as a series of 8 bits, with the most significant bit being sent first, but one should also realize that when a command or command argument is sent that is comprised of more than one byte, the least significant byte will be sent first (starting, of course, with that byte's most significant bit according to the bit-endianness).

Note the call, for example, to UINT16\_TO\_STREAM

unsigned short   
hci\_command\_send(unsigned short usOpcode, unsigned char \*pucBuff,  
                     unsigned char ucArgsLength)  
{   
    unsigned char \*stream;  
   
    stream = (pucBuff + SPI\_HEADER\_SIZE);  
      
    UINT8\_TO\_STREAM(stream, HCI\_TYPE\_CMND);  
    stream = **UINT16\_TO\_STREAM**(stream, usOpcode);  
    UINT8\_TO\_STREAM(stream, ucArgsLength);  
  
    //  
    // Update the opcode of the event we will be waiting for  
    //  
    SpiWrite(pucBuff, ucArgsLength + SIMPLE\_LINK\_HCI\_CMND\_HEADER\_SIZE);  
  
  
    return(0);  
}:

UNIT16\_TO\_STREAM is defined as:

extern unsigned char\* UINT16\_TO\_STREAM\_f (unsigned char \*p, unsigned short u16);

And UINT16\_TO\_STREAM\_f is defined as:

//This function is used for copying 16 bit to stream while converting to little endian format.  
unsigned char\* UINT16\_TO\_STREAM\_f (unsigned char \*p, unsigned short u16)  
{  
    \*(p)++ = (unsigned char)(u16);  
    \*(p)++ = (unsigned char)((u16) >> 8);  
    return p;  
}