Sending a command to the Tag from a Host (Reader)

This example demonstrates the user setting the transmit interval to 5 seconds, displaying the current time, displaying the battery level and enabling IO pins 1 and 3 to analog inputs referenced at 3 volts. The following outlines how this would be accomplished.

Start Byte: **55**

"55" signals the reader a tag command is being sent, thus all packets being sent to the tag begin with byte "55". This is referred to as the start byte.

Packet Length: **E0**

The packet length tells the tag how many bytes are being sent (not including the start bytes or packet length bytes, i.e. starting with the packet identifier bytes).

Packet Identifer: **02**

The packet identifier can hold two values "02" or "04", "02" is used when sending a command to the tag. "04" is used when a data packet is being sent from the tag.

Command Handle: **05**

The command handle is used by the reader to identify the command, in the event that multiple commands are being sent to tags with different ID's, a queue is used to store the commands until the tag wakes from sleep. Therefore the command handle should never be the same as another command handle. A simple way to implement this is to increment the command handle each command.

Tag ID: **0003**
The ID of the tag that the command is being sent to.

Command Header: **80 =** *1000 0000*

The command header is asking the tag to send an acknowledge packet back to the reader and is indicating to the tag the parameters will apply to mode 0 (because bits 4,5,6 "000" in decimal is equal to 0).

Packet Control Header: **C0** = *1100 0000*

The packet control header is telling the tag that a network field (bit 7) and GPIO field (bit 6) are present.

Network Control Mask Header: **60** = *0110 0000*

This indicates to the tag the the Time mask bit and TX interval setting bit are enabled (Pins 5 and 6 are "11").

Network Control Action Header: **60** = *0110 000*

This indicates to the tag the the Time report is to be enabled and the TX interval is to be set to the TX interval value.

TX Interval Bit: **04**

This tells the tag to change the TX interval to the value associated with the byte 04, which is equal to 5 seconds. This is determined using a lookup table, which is preset in the firmware.

GPIO Mask Byte: ***0A*** *= 0000 1010*

This indicates that IO pins 1 and 3 are being configured.

GPIO A/D Configuration Byte: ***0A*** *= 0000 1010*

This indicates that IO pins 1 and 3 are being configured at analog inputs.

GPIO I/O Configuration Byte: **00** *= 0000 0000*

This indicates that IO pins 1 and 3 are being configured as inputs (bits 1 and 3 are "0" indicating inputs, where as a value of "1" indicates an output).

GPIO Digital Configuartion/Analog Reference Byte: ***0A*** *= 0000 1010*

This indicates that the Analog inputs on IO pins 1 and 3 are being referenced at 3 volts. A "1" indicates a 3 volt reference, while a "0" would represent a 1.1V reference. Assuming pins were configured as analog inputs in the previous Bytes.

GPIO Reporting Configuration Byte: ***0A*** *= 0000 1010*

This indicates that the data on IO inputs 1 and 3 (because bits 1 and 3 are set to "1") are to be transmitted in the beacon packet sent by the tag. A 10-bit Analog to Digital converter is used thus each analog input is reported as two bytes.

Finally, the required packet would be:

55E00205000380C06060040A0A000A0A

Sending this command produces the following reply from both the tag and reader:

```
+DEBUG [>>550E0208000380C06060040A0A000A0A]  <=== Host Command

+DEBUG [<<AA03020800]  <=== Reader Command Packet Response

+DEBUG [<<AA12040003FFFFEC30C010C888050501560000DE] <=== Tag transmitting old data

+DEBUG [<<AA09040003FFFFEC3100DF]  <=== Tag acknowledges command

+DEBUG [<<AA1A040003FFFFED30C09000004DA2C8880F0F0145011F00C00000DF] <=== Tag adjusts output
```