# Memo

| | | |
|---|---|---|
| **Topic :** | **Example for DDC Application Note** | |
| | GPS.G5-X-08023-addendum | Confidential – For Internal Use |
| **Author :** | NV, TG | |
| **Date :** | 10/11/2008 | |
| **To :** | FAEs | |



# 1 Introduction

This document includes additional information and an example for the DDC Application Note. The purpose is to allow for feedback from FAEs to see if the following input can answer customer questions arising from the existing document.

# 2 Data Availability (Section 7)

Slave outputs valid messages (data different than 0xFF) once the following requirements are met:

1. The START CONDITION is present followed by the recognized receiver's address
2. The messages to be output are prepared at the end of the epoch.
    - Please note that the output buffer filling is synchronised with each epoch. Hence if the buffer is empty an attempt to read the data before the next epoch defaults to 0xFF. While continuously polling it is therefore normal to observe entire sections of data containing 0xFF. Necessity to poll the buffer until the data is different than 0xFF becomes also apparent (please see Figure 8).
    - *Slave* does not prepare messages in advance unless condition 1 is met. Consequently the number of available bytes is initially 0 and is to be polled continuously until different than 0. It is only then that reading can take place. If reading takes place prior to this indication slave outputs 0xFF by default.
    - As discussed earlier failure to provide regular read requests at the frequency lower than 1.5 sec results in the communication timeout and interface suspension (data unavailable).

# 3 Example: Transmission of UBX-CFG-RST command (MASTER to SLAVE) (Section 7)

1. Master generates the START condition.
2. Master initiates transmission by sending the WRITE request. The WRITE request is addressed to the GPS receiver (default addr 0x42) and the data field points to the address 0xFF (data buffer for incoming / out coming transmission). The clock line SCL is kept low.

   Please note that this step is optional. The register pointer might already be set to Data Stream (addr 0xFF) and there would be no need to set it again. This step is only to ensure that the current data exchange relates to the Data Stream register.
3. Master initiates transmission by sending another WRITE request. The WRITE request is addressed to the GPS receiver (default addr 0x42) and the data fields following the WRITE request contain a valid UBX message (in this case UBX-CFG-RST).
4. Master sends the STOP condition.

The following steps are optional and apply to all messages requiring acknowledgement in the form of ACK / NACK message. For other messages this would be the end of transmission.

5. Master generates the START condition.
6. Master initiates transmission by sending the WRITE request. The WRITE request is addressed to the GPS receiver (default addr 0x42) and the data fields following the WRITE command point to the address 0xFF (data buffer for incoming / out coming transmission).
7. Master initiates transmission by sending the READ request. The READ request is addressed to the GPS receiver (default addr 0x42) and the data field points to the address 0xFF (data buffer for incoming / out coming transmission).
8. Master reads as many bytes as needed, i.e. until it gets a valid ACK / NACK message.

   Please note that this method of reading is based on polling. Another approach for MASTER would be to scan the number of available bytes first by reading locations 0xFD and 0xFE. For as long as these fields indicate 0 available bytes the MASTER would wait in the loop before starting to read and parse the data.

   ☞ Please note that every successive read / write attempt automatically moves the register pointer forward until the last location is reached (Data Stream at 0xFF). Hence, if continuous reading of any other register is required, periodical register pointer presetting is required before reading. The register pointer does not wrap around 0xFF.
9. Master sends STOP condition.

End of transmission.