AS321X Application Note – AN04-1
Access to sensor data

AS3213S.5 – AS3213T.5 – AS3213C.4 – AS3212.6 – AS3211.7

Revision History

Revision	Date	Comment
2.0	2023-02-10	First version with revision history

Table of Contents

1. Description

The purpose of this document is to help users of ASYGN AS321X integrated circuits to collect and interpret sensor data. It explains how data is organized and formatted inside the memory of the IC.

This document is valid for the following references:
AS3213S.5 – AS3213T.5 – AS3213C.4 – AS3212.6 – AS3211.7

2. How to obtain the sensor data

First of all, it is important to note that sensor data can be obtained using a simple standard EPC Gen2 *Read* command both in the USER bank at addresses 0x0 and 0x1 and in the EPC bank at addresses 0xE and 0xF. The memory maps of these banks are shown below.

Memory map of the USER bank

Label			LogAddr	Bank	NVM mapped
USER7	0x	7	7	USER	YES
USER6	0x	6	6		YES
USER5	0x	5	5		YES
USER4	0x	4	4		YES
USER3	0x	3	3		YES
USER2	0x	2	2		YES
ACQ_TEMP	0x	1	1		NO
ACQ_SENS	0x	0	0		NO

Memory map of the EPC bank

Label			LogAddr	Bank	NVM mapped
CONFIG2	0x	12	18	EPC	YES
CONFIG1	0x	11	17		YES
CONFIG0	0x	10	16		YES
ACQ_TEMP	0x	F	15		NO
ACQ_SENS	0x	E	14		NO
EPC0	0x	D	13		YES
EPC1	0x	C	12		YES
EPC2	0x	B	11		YES
EPC3	0x	A	10		YES
EPC4	0x	9	9		YES
EPC5	0x	8	8		YES
EPC6	0x	7	7		YES
EPC7	0x	6	6		YES
EPC8	0x	5	5		YES
EPC9	0x	4	4		YES
EPC10	0x	3	3		YES
EPC11	0x	2	2		YES
STORED_PC	0x	1	1		YES
STORED_CRC	0x	0	0		YES

Note 1 - Alternative option to obtain the sensor data, for applications requiring faster processing

By changing the value of bits <16:12> in the STORED_PC word (EPC bank, address 0x1), it is also possible to obtain sensor data directly during the Inventory round: use STORE_PC = 0x7400 (rather than default value = 0x6400).

In this case, the capture of sensor data is accelerated because the additional read command is not needed.

In addition, sensor data from multiple tags can be obtained in a single Inventory sequence with this option.

Note 2 – Sensor words are stored in a volatile register

Even though sensor data words are mapped into the USER and EPC banks, they are not written in the non-volatile memory (NVM) of the IC but in a volatile register. As a consequence, the sensor words are only available when the tag is powered by the RF field. Each interrogation round provides a new set of data.

3. Sensor data format

The resolution of the ADC embedded into the IC is 10 bits. Acquired sensor data is available as a 16-bit word as shown below. This format is the same for ACQ_SENS and ACQ_TEMP data.

	ADC_NSMPL	PWR_OK	ACQ_SENS / ACQ_TEMP DATA												
# of samples	15:14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	ADC_NSMPL	PWR_OK				D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
2	ADC_NSMPL	PWR_OK			D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	D-1
4	ADC_NSMPL	PWR_OK		D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	D-1	D-2
8	ADC_NSMPL	PWR_OK	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	D-1	D-2	D-3

PWR_OK (bit 13 of this sensor data word) is a status bit indicating that the power supply remained within the specified limits during the measurement, which guarantees the accuracy and precision of the measurement.

ADC_NSMPL (bits <15:14> of this sensor data word) gives the number of acquisitions performed by the ADC. This parameter is set in CONFIG0 register (CONFIG0<1:0>), available at address 0x10 in the EPC bank.

ADC_NSMPL is explained in the table below:

ADC_NSMPL	Number of samples acquired and converted in a row, for averaging. ‘00’: Only one acquisition is performed ‘01’: 2 acquisitions ‘10’: 4 acquisitions ‘11’: 8 acquisitions

Depending on the value of ADC_NSMPL, used for averaging, the samples taken are summed in a binary way and written in the register.

Specifically:

- if ADC_NSMPL = 0x0 (1 acquisition), the recorded data is 10-bit long, ACQ_XXX<9:0>, because only one sample is stored in the data register.
- if ADC_NSMPL = 0x1 (2 acquisitions), the recorded data is 11-bit long, ACQ_XXX<10:0>, because 2 successive samples are added in a binary sum and stored in the data register.

  Similarly:
- if ADC_NSMPL = 0x2 (4 acquisitions), the recorded data is 12-bit long, ACQ_XXX<11:0>,
- if ADC_NSMPL = 0x3 (8 acquisitions), the recorded data is 13-bit long, ACQ_XXX<12:0>.

Each time the ADC performs an acquisition the data is summed with the previous one.

It is therefore possible to reduce noise by averaging as explained below:

- For 1 acquisition, take ACQ_XXX<9:0> and divide it by 1 (2^ADC_NSMPL --> 2^0)
- For 2 acquisitions, take ACQ_XXX<10:0> and divide it by 2 (2^ADC_NSMPL --> 2^1)
- For 4 acquisitions, take ACQ_XXX<11:0> and divide it by 4 (2^ADC_NSMPL --> 2^2)
- For 8 acquisitions, take ACQ_XXX<12:0> and divide it by 8 (2^ADC_NSMPL --> 2^3)