

# NHS3XXX

## DATA STORAGE

OVERVIEW OF COMPLEXITY AND  
RECOMMENDATIONS



EXTERNAL USE



SECURE CONNECTIONS  
FOR A SMARTER WORLD

# Parameters

Data storage is influenced by the setup of and the decisions made in a number of topics:

- EEPROM
- FLASH
- Security
- Retrieval
- Data type
- Timestamp
- Storage scheme
- Compression
- SDK support



# Storage

## EEPROM

- ~3.6 kB available (58 rows)
- Not active by default
- Endurance >10k write cycles
- Cannot be locked
- Writing takes 3 ms / page (64 bytes)
- Write cost of 300  $\mu$ A / MHz
- Abstraction using `EEPROM driver`

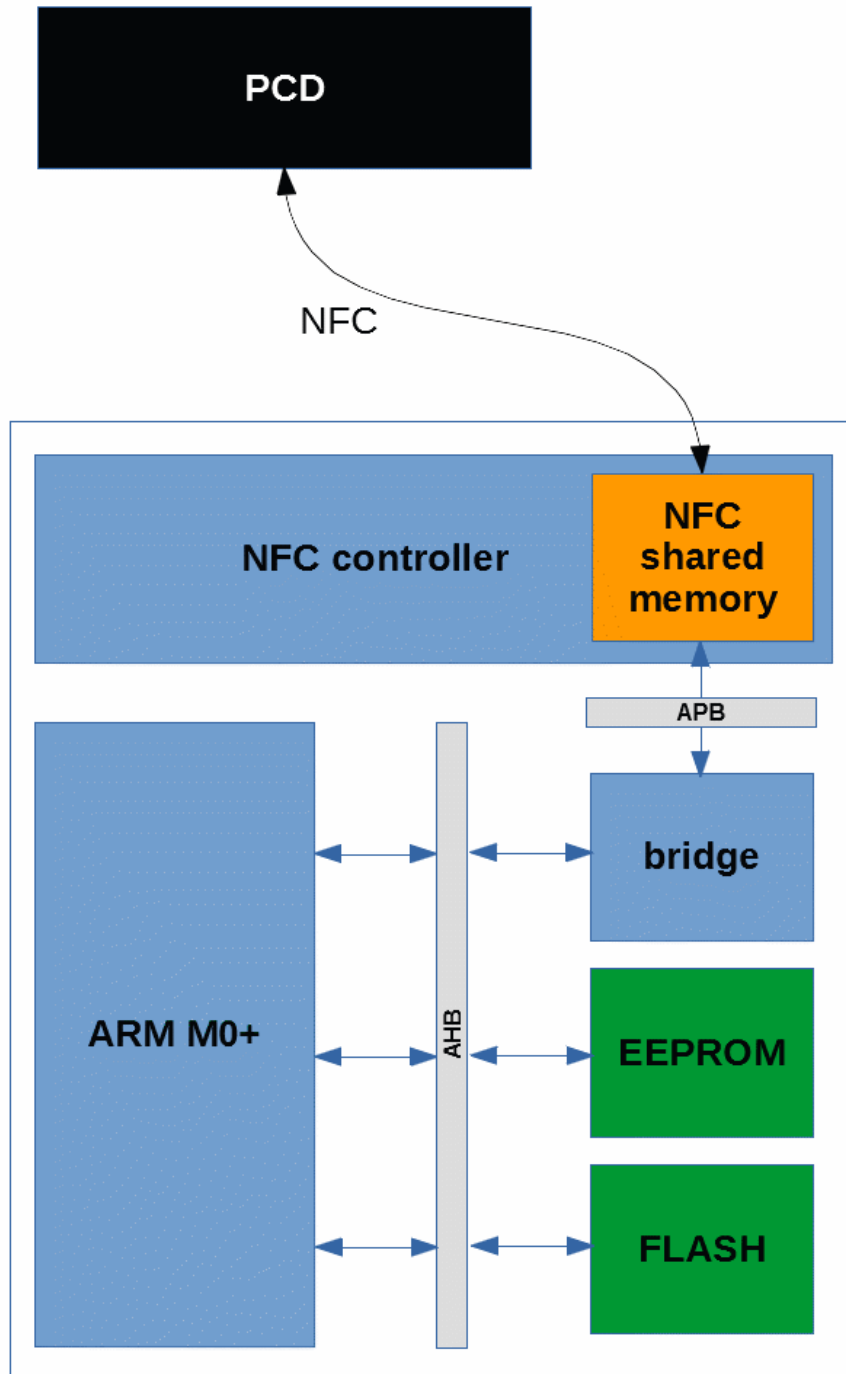
## FLASH

- (30 kB - program size) available
- Active by default
- Endurance >10k program/erase cycles
- Can be locked per sector (1 kB)
- Erase takes 100 ms  
Program takes 1 ms / page (64 bytes)
- Program cost of 500  $\mu$ A / MHz
- Abstraction using `IAP driver`

# Retrieval – built-in functional security

## No direct access to EEPROM and FLASH

1. PCD (e.g. an NFC enabled phone) writes command for data retrieval to NFC shared memory
2. ARM Cortex M0+ processes command
3. ARM Cortex M0+ retrieves data from EEPROM and/or FLASH
4. ARM Cortex M0+ writes response to NFC shared memory
5. PCD extracts requested data from response



# Compression of temperature values – 1

## Temperature sensor

- 0.3°C absolute temperature accuracy between 0°C and 40°C
- 0.5°C absolute temperature accuracy between -40°C and 85°C

## Size of one temperature value

- Depends on the required resolution of the measurement when stored.
- ! This is *not* the sensor accuracy!
  - Example: temperature sensor returns 5.24°C. The real temperature is 5.24°C +/- 0.3°C. This can be stored using a 0.1°C resolution as 5.2°C

# Compression of temperature values – 2

Temperature range

Limited by

- IC specification:  $-40^{\circ}\text{C} \dots 85^{\circ}\text{C}$
- battery characteristics: *e.g.*  $-20^{\circ}\text{C} \dots 65^{\circ}\text{C}$
- Specific use case: *e.g.*  $-2^{\circ}\text{C} \dots 12^{\circ}\text{C}$  for a required cold chain environment between  $2^{\circ}\text{C}$  and  $8^{\circ}\text{C}$

The combination of range and resolution results in a limited number of possible temperature values.

Example: With a range between  $0^{\circ}\text{C}$  and  $10^{\circ}\text{C}$  and a resolution of  $0.1^{\circ}\text{C}$ , there are 101 possible values.

A mapping between a temperature value and a number then reduces the required storage space for one temperature value.

Example: for 101 possible values, 7 bits are required per value.



# Compression of temperature values – 3

Use case range	Storage resolution	#distinct values	Required bit size per value	Total storage capacity	
				Fits in 24 KB:	Fits in 20 KB:
-40°C ... 85°C	0.1°C	1251	11	17455	14545
	0.2°C	626	10	19200	16000
	0.5°C	251	8	24000	20000
-20°C ... 65°C	0.1°C	851	10	19200	16000
	0.2°C	526	10	19200	16000
	0.5°C	171	8	24000	20000
0°C ... 10°C	0.1°C	101	7	27429	22857
	0.2°C	51	6	32000	26667
2°C ... 8°C	0.1°C	61	6	32000	26667
	0.2°C	31	5	38400	32000

# Compression of generic data (not limited to temperature)

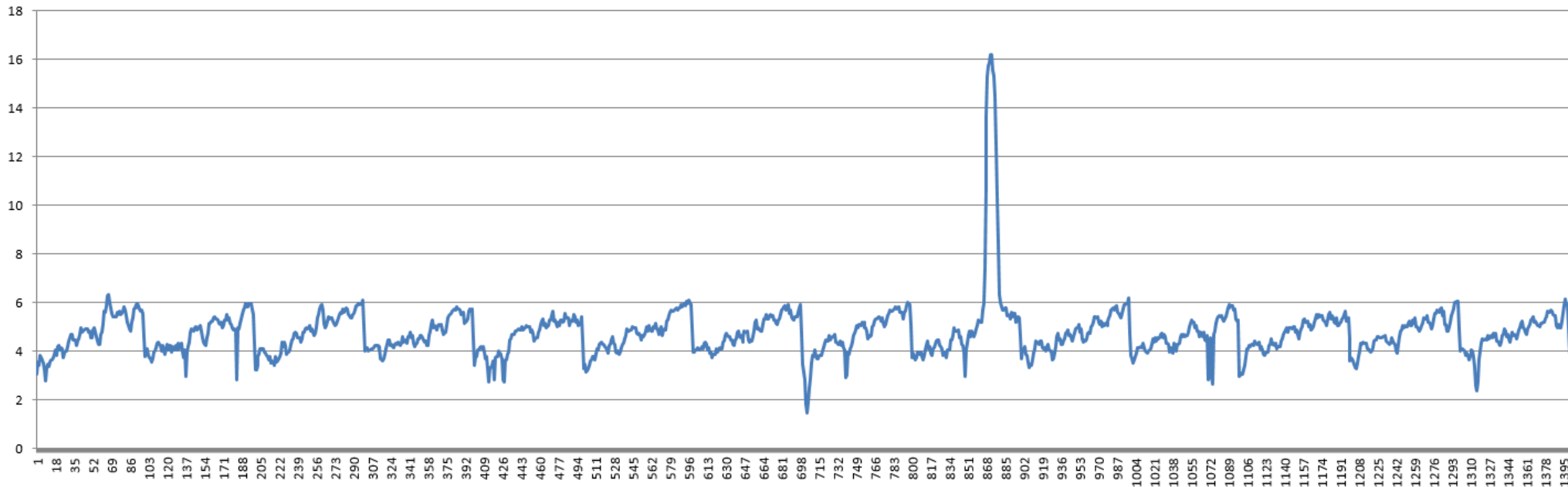
Other techniques to reduce required storage space

- Reconstruct timestamps using the start time and the sequence number. The sequence number can be recovered based on the physical location in the memory assigned for data storage.
- Use a lossless use case agnostic compression.
  - Huffman coding
  - Run-length encoding
  - Delta encoding
  - Specialized algorithms (e.g. [heatshrink](#))
- Use a lossy compression – use case specific
  - Zone-coding



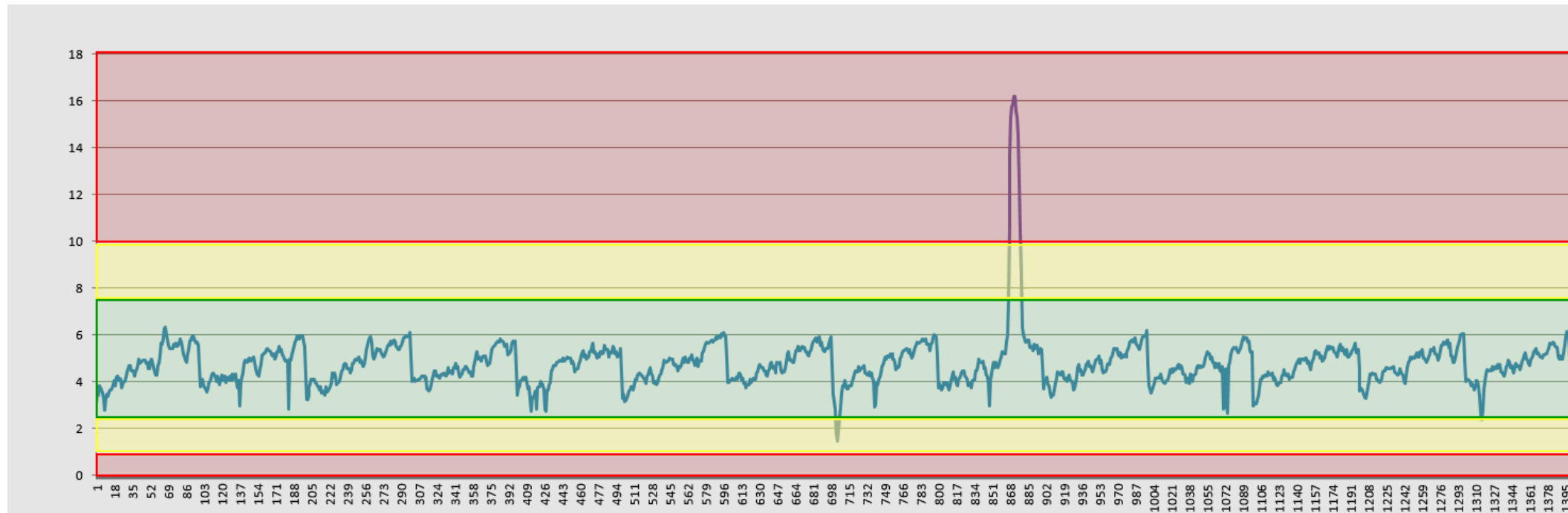
# Compression of generic data – zone-coding – 1

- Identify zones, and treat data in each zone differently.
- As an example, consider a product that is kept in a refrigerator, where temperature values have been samples as per below:



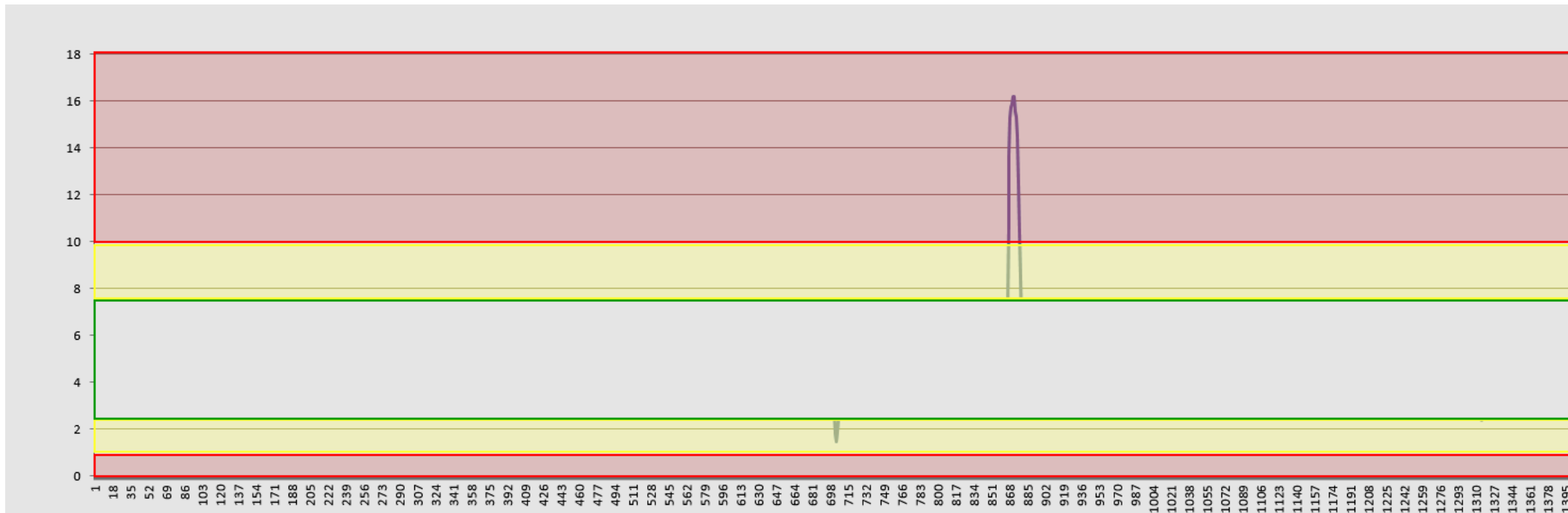
# Compression of generic data – zone-coding – 2

- Divide the full temperature range in different bands.
- This is specific per type of product the tag is attached to.
- At least a ‘safe’ zone and a ‘red’ zone – leading to three bands, but more zones can be defined. Below also a ‘danger’ zone is defined – leading to five bands.



# Compression of generic data – zone-coding – 3

- As an extreme: only store the outliers.
- Add an index per outlier for reconstruction of the timestamp.
- A middle ground can be to store data in the green zone as well, using a lower resolution.



# SDK support

The SDK provides the *storage* module that simplifies storage and retrieval of samples. It is used by the ARM temperature logger firmware demo application.

## Benefits:

- Simple API abstracting away all intricacies.
- EEPROM and FLASH characteristics are taken into account
  - Endurance limit will never be reached
  - FLASH Write/Program cost is minimized
- Data is stored packed – no unused bits
- Data can be compressed
  - Algorithm to be provided by the application
  - Can be linked in at compile time
- Direct access to any stored value



SECURE CONNECTIONS  
FOR A SMARTER WORLD