NTAG I²C

Technical Product Presentation

July 2014

Michael Salfer
CAS

Agenda

- General
  - NFC Forum tag – NDEF
  - Product Portfolio – Differences
- **NTAG I²C**
  - Overview
  - Delivery Types
  - Memory Structure
  - RF Interface
  - I²C Interface
  - Features:
    - Field Detection
    - SRAM / SRAM Mirror
    - Passthrough Mode
    - Energy Harvesting
  - Memory Arbitration
  - Product Support Package
  - Known Software Issues

What is a NFC Tag?

- A **NFC tag** is a contactless tag capable of storing NDEF formatted data, which interoperates with ISO14443 infrastructure and NFC devices as defined by the NFC Forum
- A **NFC Forum tag** is compatible to one of four NFC Forum Tag platforms capable of storing NDEF formatted data
- ICODE, MIFARE Classic can be NFC tags, but NOT NFC Forum tags (ICODE is becoming a NFC Forum tag)
- UL, ULC, DESFire and SmartMX can be NFC Forum tags
- NTAGs are NFC Forum tags (already formatted)

NFC Forum Type Tag Platform – Product Overview

- The NFC Forum has standardized 4 Type Tags Platform.

#	NFC Forum Platform	Compatible Products
1	NFC Forum Type 1 Tag	Innovision Topaz
3	NFC Forum Type 3 Tag	Sony Felica
4	NFC Forum Type 4 Tag	NXP DESFire EV1 / NXP SmartMX with JCOP or other compatible contactless products

- Similarly NXP has also specified

NXP Specific Platform	Compatible Products
NFC Type MIFARE Classic Tag	MIFARE Classic 1k / MIFARE Classic 4k / MIFARE Plus X/S
NFC Type ICODE tag	ICODE SLI/SLIX / ICODE SLI-S/SLIX-S / ICODE SLI-L/SLIX-L

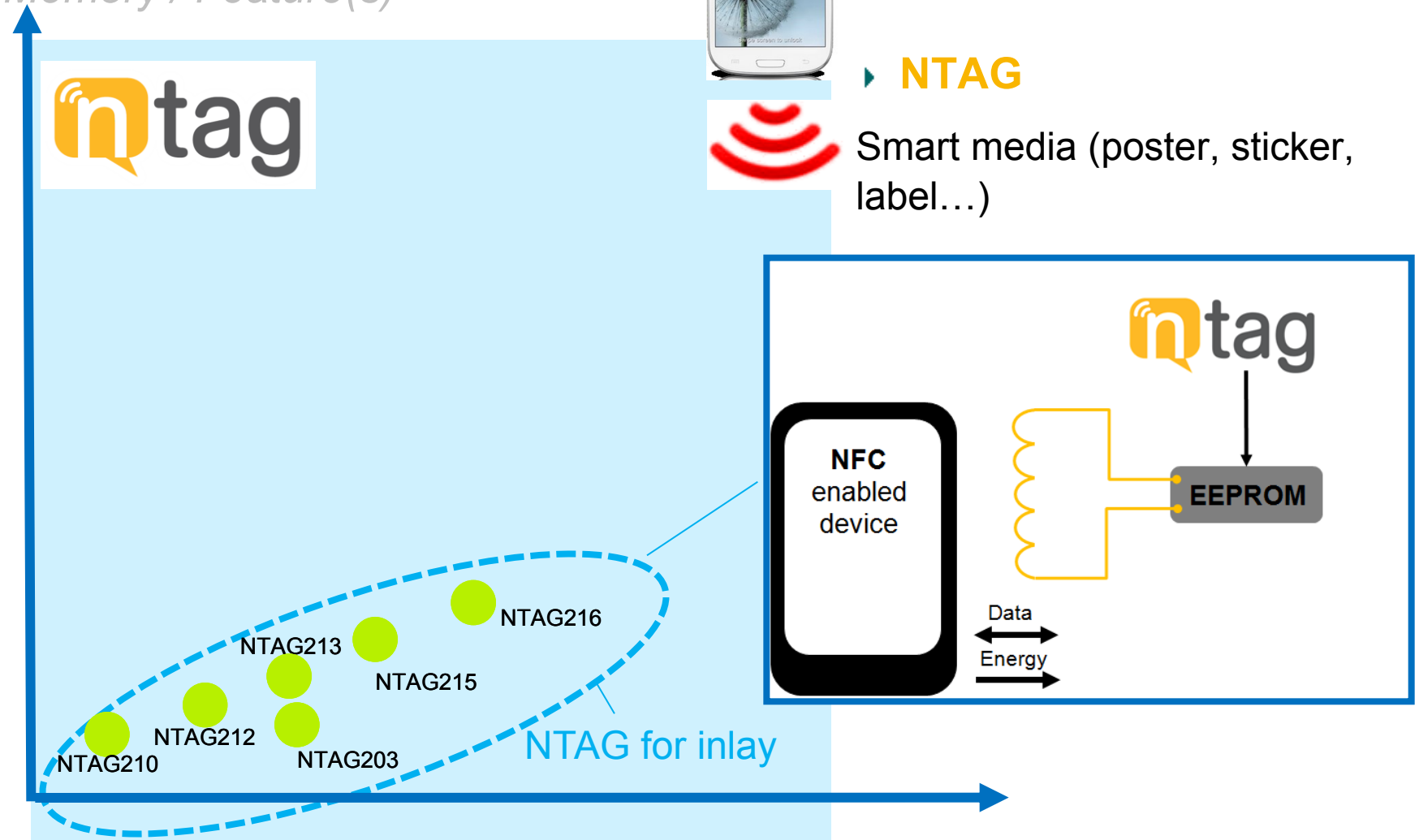
Product portfolio

A complete portfolio

Memory / Feature(s)

- **NTAG**

Smart media (poster, sticker, label…)



A complete portfolio

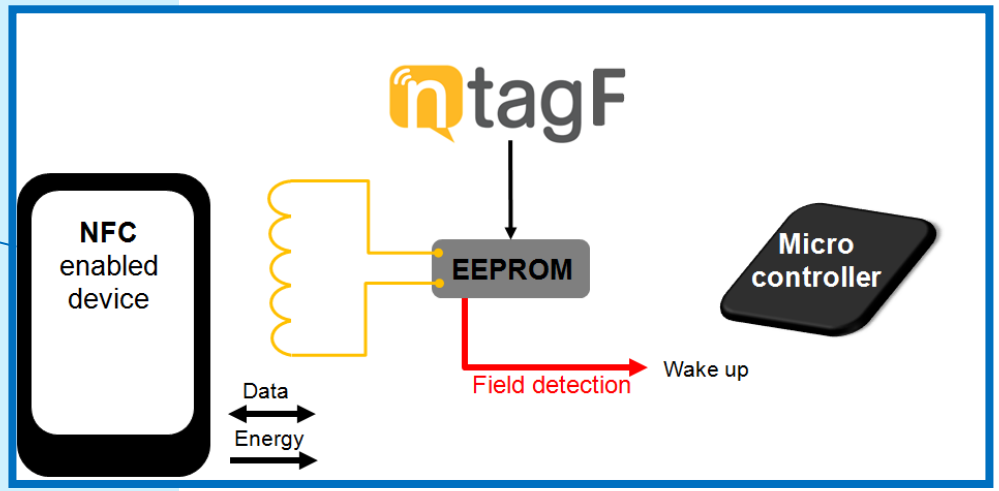
NTAG F

Simple WiFi / Bluetooth pairing with Field detection for waking up Electronics devices

Memory / Feature(s)

NTAG F with field detection

NTAG216F
NTAG213F
NTAG203F
NTAG216
NTAG213
NTAG215
NTAG212
NTAG210
NTAG203
NTAG for inlay



A complete portfolio

NTAG I²C

Full interaction NFC device/Electronics device for dynamic NDEF message update and direct data transfer

Memory / Feature(s)

NTAG I²C for contact and RF interface

NTAG I²C 2k memory

NTAG I²C 1k memory

NTAG F with field detection

NTAG216F

NTAG213F

NTAG203F

NTAG216

NTAG213

NTAG215

NTAG212

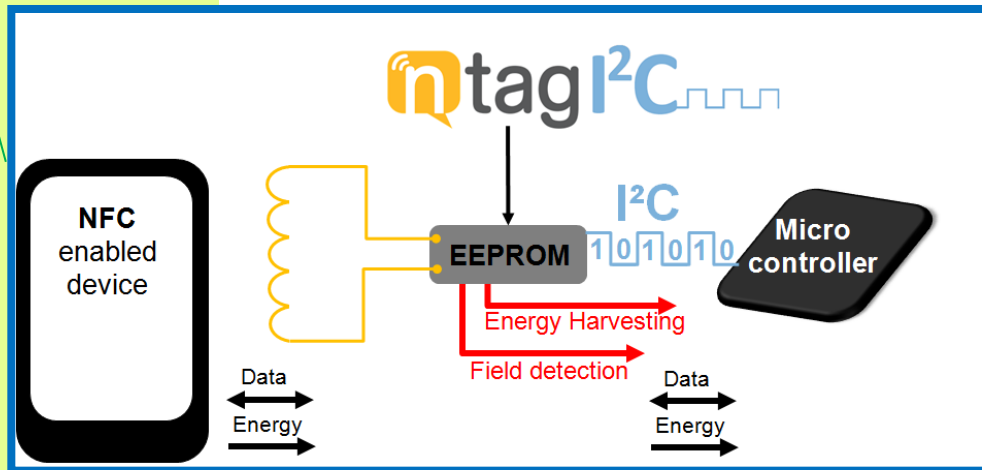
NTAG210

NTAG203

NTAG for inlay

Single interface

Dual-Interface



Product differences

NTAG Product Overview

	NTAG 203	NTAG 203F	NTAG 210 / 212	NTAG 213 / 215 / 216	NTAG 213F / 216F	NTAG I2C 1k / 2k
User memory in byte	144	144	48 / 128	144 / 504 / 888	144 / 888	888 / 1904
Pwd auth	-	-	r/w	r/w	r/w	-
Capacitance	50 pF	50 pF	17 pF	50 pF	50 pF	50 pF
UID mirror	-	-	●	●	●	-
NFC counter	-	-	-	●	●	-
One way counter	●	●	-	-	-	-
FD pin	-	●	-	-	●	●
I2C	-	-	-	-	-	●
READ0 to ACTIVE	●	●	●	●	●	-

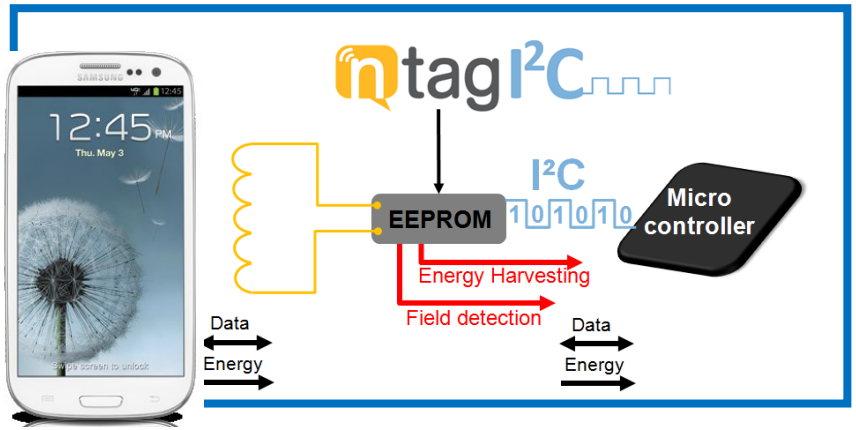
● = yes / supported

RF Command Code Overview

Command	ISO/IEC 14443	NFC Forum	Code	203(F)	210/ 212	213(F)/ 215/ 216(F)	I2C 1k / 2k
Request	REQA	SENS_REQ	26h (7 bit)	•	•	•	•
Wake-Up	WUPA	ALL_REQ	52h (7bit)	•	•	•	•
Anticollision CL1	Anticollision CL1	SDD_REQ CL1	93h 20h	•	•	•	•
Select CL1	Select CL1	SEL_REQ CL1	93h 70h	•	•	•	•
Anticollision CL2	Anticollision CL2	SDD_REQ CL2	95h 20h	•	•	•	•
Select CL2	Select CL2	SEL_REQ CL2	95h 70h	•	•	•	•
Halt	HLTA	SLP_REQ	50h 00h	•	•	•	•
Get_Version			60h	-	•	•	•
Read		READ	30h	•	•	•	•
Fast_Read			3Ah	-	•	•	•
Write		WRITE	A2h	•	•	•	•
Comp_Write			A0h	•	•	•	-
Read_Cnt			39h	-	-	•	-
Pwd_Auth			1Bh	-	•	•	-
Read_Sig			3Ch	-	•	•	-
Sector_Select (two part command)		SECTOR_SELECT	C2h	-	-	-	•

NTAG I²C

NTAG I²C Features



- NFC Forum Type 2 Tag compliant
- Dual interface
  - NFC & I²C interface
  - 50 pF input capacitance
- Either 888 or 1904 Byte of Non-Volatile User Memory
- PASS THROUGH mode for fast data transfer
  - 64 bytes SRAM Buffer for RF→I²C and I²C→RF data transfer
  - Mapped at the end of the memory sector
- Field detection feature
  - Open drain implementation
  - Configurable reaction on
    - RF field presence
    - First valid command (State of Frame) / NDEF message read
    - Device selection / Deselection
    - Data read/written in the SRAM buffer
- Clear arbitration / customized access between the 2 interfaces
  - Memory locked to one interface upon the other interface accessing the memory
  - Status flag to check access status
  - Flexible WRITE Access settings
- Energy harvesting functionality to power external devices (e.g. µC)
- XQFN8 package (1,6*1,6*0,5mm)

NTAG I²C for a complete NFC Eco System

Home automation
Tap your phone to adjust settings or for commissioning
Video (Youtube)

Smart Meters
Tap your phone to read out consumption data

NFC accessories
Tap your phone to configure or retrieve personal data

Consumer Electronics
Tap your phone to stream media or set the programming

Appliances
Tap your phone for service diagnostics or settings
Video (Youtube)

Healthcare
Tap your phone to read and upload vital measurements

Some features in more detail on the following slides

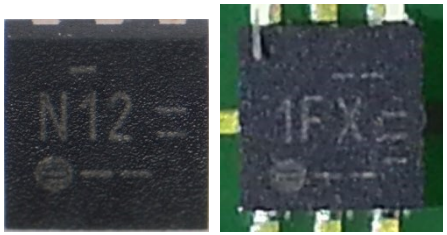
- Passive Tag
  - NTAG I²C can execute all RF operations in passive tag mode (powered by RF field)
- I²C Interface
  - Easy I²C commands for EEPROM and SRAM access
- Field detection
  - NTAG I²C can drive the open drain FD signal in passive tag mode
- Embedded SRAM
  - can be used to transfer data (pass-through-mode) or be mirrored on the EEPROM
- Energy Harvesting
  - can supply the I²C-part, the SRAM and external circuitry

Delivery types

NTAG I²C

Delivery Types Package and Pinning

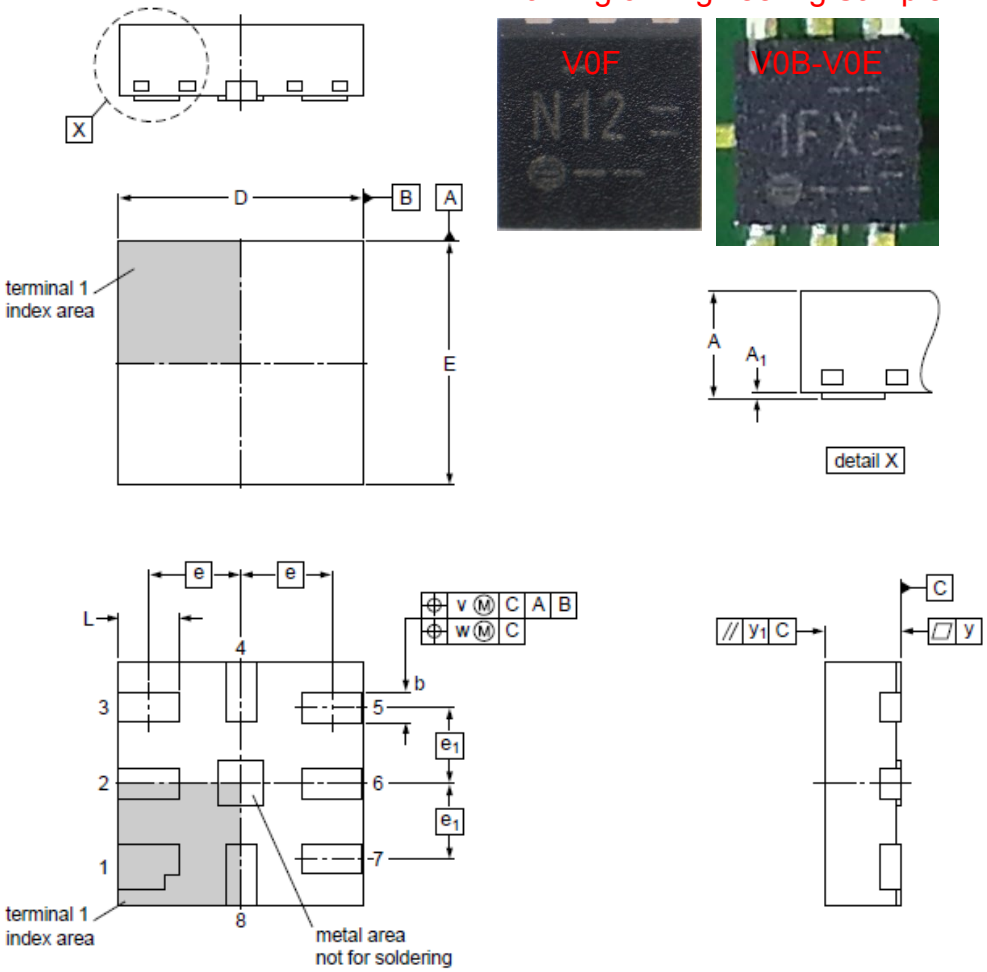
Model	Package	Package Version	Marking	Cap	Type	User Memory
NTAG I²C 1k	XQFN8	SOT902-3	N11 / yww	50 pF	NT3H1101W0FHK	888 bytes
NTAG I²C 2k	XQFN8	SOT902-3	N12 / yww	50 pF	NT3H1201W0FHK	1904 bytes



V0F V0B-V0E

Marking of Engineering Sample

Package SOT902-3 (XQFN8)



Marking of Engineering Sample

Dimensions — scale

Unit		A	A_1	b	D	E	e	e_1	L	v	w	y	y_1
mm	max	0.5	0.05	0.25	1.65	1.65			0.45				
	nom			0.20	1.60	1.60	0.6	0.5	0.40	0.1	0.05	0.05	0.05
	min		0.00	0.15	1.55	1.55			0.35				

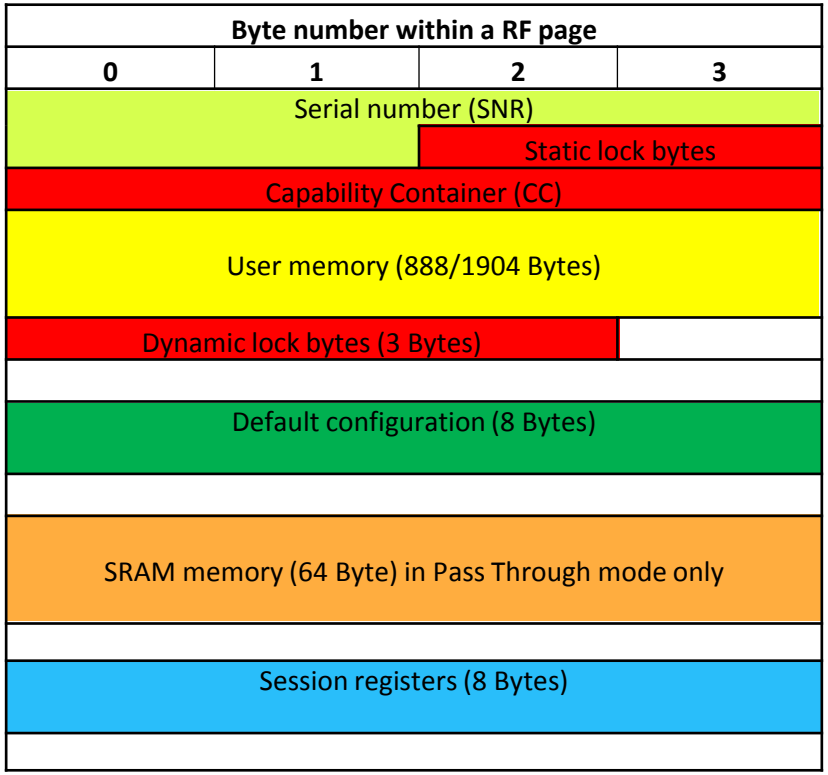
Note
1. Plastic or metal protrusions of 0.075 mm maximum per side are not included.

Pin	Symbol	Description
1	LA	Antenna coil connection LA
2	VSS	GND connection
3	SCL	Serial Clock I²C connection
4	FD	RF field detection connection
5	SDA	Serial Data I²C connection
6	VCC	VCC in connection
7	Vout	Voltage out (energy harvesting)
8	LB	Antenna coil connection LB

Memory structure

NTAG I²C Memory Structure

- 888/1904 Byte User Memory
- Organization RF:
  - 4 byte per page
  - 256 pages per sector
    - NTAG I²C 1k: 1 sector data
    - NTAG I²C 2k: 2 sectors of data
- Organization I²C:
  - 16 byte per block
  - No sectors



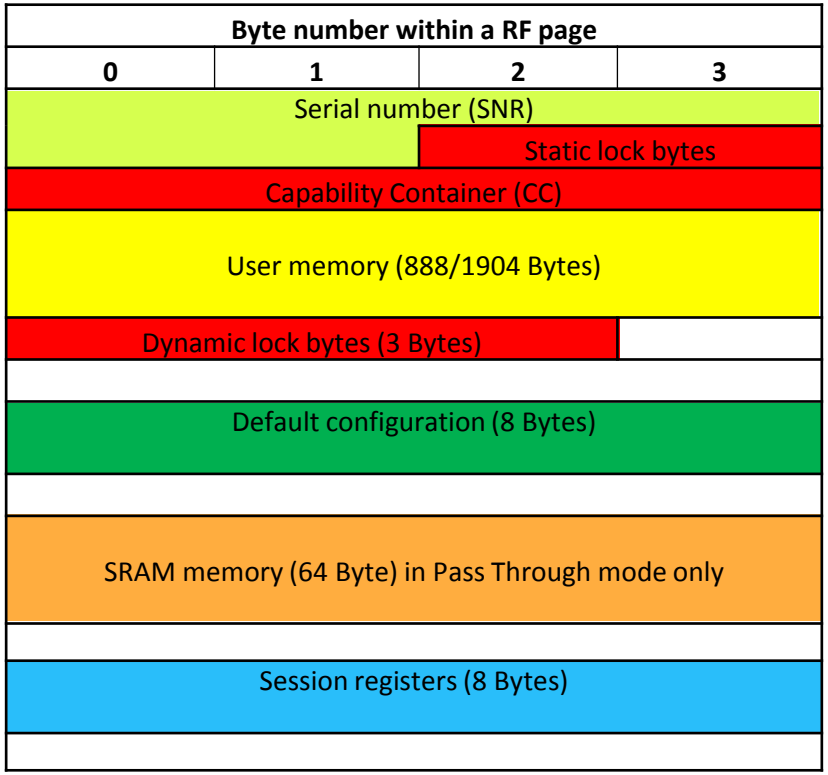
Detail 1k RF | Detail 1k I²C

Detail 2k RF | Detail 2k I²C

NTAG I²C Memory Access Rights

- 888/1904 Byte User Memory

Memory	Read		Write	
	RF	I²C	RF	I²C
Serial Number	ok	ok	-	I2C Address *
Lock Bytes / CC	ok	ok	OTP, lockable	ok
User Memory	ok	ok	ok, lockable	ok
Configuration	ok	ok	ok, lockable	ok, lockable
SRAM	ok	ok	ok	ok
Session Register	ok	ok	-	ok



*: I²C Address only writeable, but not readable

Special memory Block: Lock bits

- Static Lock Bits (static position)

Static Lock Bits			Address Byte / Bit															
	Page		2								3							
Model	Dec	hex	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
NTAG I2C	2	02h	7	6	5	4	CC	10-15	4-9	CC	15	14	13	12	11	10	9	8

- Dynamic Lock Bits (position depends on memory size)

Dynamic Lock Bits				Address Byte / Bit																							
	Page Address			0								1								2							
Model	Sector	Dec	Hex	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
NTAG I²C 1k	0	226	E2h	128-143	112-127	96-111	80-95	64-79	48-63	32-47	16-31	RFUI		224-225	208-223	192-207	176-191	160-175	144-159	RFUI	208-225	176-207	144-175	112-143	80-111	48-79	16-47
NTAG I²C 2k	1	224	E0h	240-271	208-239	176-207	144-175	112-143	80-111	48-79	16-47	RFUI	464-479	432-463	400-431	368-399	336-367	304-335	272-303	464-479	400-463	336-399	272-335	208-271	144-207	80-143	16-79

Lock bit: to lock corresponding page(s) to read only

Block lock bit: to lock corresponding lock bits (disable changing of lock bits)

Note: after writing to lock area the changes are immediately active

Note: Access from RF = OTP, Access from I2C = read/write

Note: for setting the lockbits according to NFC Forum rules, look at AN11456 “NTAG215_216(F)_I2C Using the dynamic lock bits to lock the tag_2769xx”

AN In publish process

NTAG I²C
Memory Initialization - NDEF

- Memory is already initialized with CC and TLVs to be used for NDEF-storage
- TLV = Type, Length, Value (generic container for data)

Memory Initialization		Page / Byte											
		Page 03				Page 04				Page 05			
Model	Mem Bytes	0	1	2	3	0	1	2	3	0	1	2	3
NTAG I²C 1k*	888	E1	10	6D	00	03	00	FE	00	00	00	00	00
NTAG I²C 2k*	1904	E1	10	EA	00	03	00	FE	00	00	00	00	00
Description		NDEF-Indicator	Version	Memory Size	Access Indicator	Lock Control TLV (Specify Position of Dynamic Lock Bits)				Zero Length NDEF TLV			

Note: for setting the lockbits according to NFC Forum rules, look at AN11456 "NTAG215_216(F)_I2C Using the dynamic lock bits to lock the tag_2769xx"

AN In publish process

* Special Memory Initialization

Special Memory Block: Configuration and Session Registers

- **Configuration** bytes are stored in EEPROM and are loaded in the session registers of NTAG I²C on power up
  - Define power-up behavior
  - Changes of the configuration bytes are only affective after Power on reset
  - Configuration is readable and writable from both interfaces
  - Write protection possible (REG_LOCK)
- **Session** registers contain current configuration and status of the tag
  - Used for runtime configuration and data transfer synchronization (Passthrough mode)
  - readonly via RF (prevent unwanted configuration from RF)
  - writeable for I²C (the I²C host has always full control over the tag)

RF interface

NTAG Read (30h)

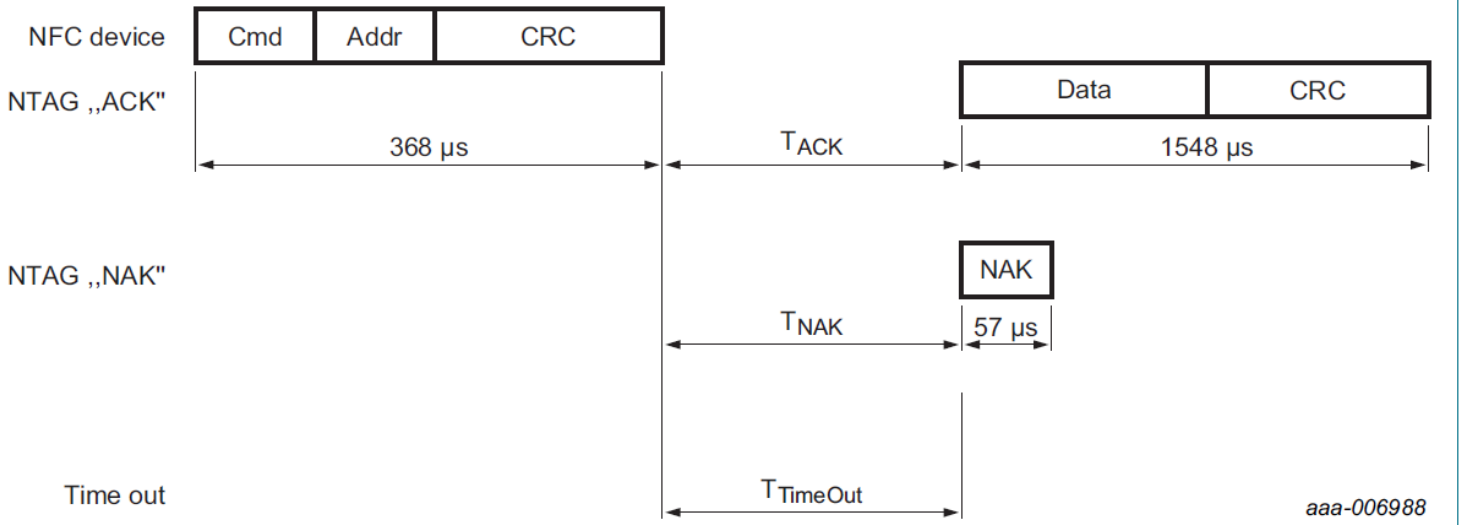


Fig 29. READ command

- READ command needs page address as parameter
  Tag returns 16 bytes (4 pages), starting from provided page address
- Note Ultralight: Reading over the end of the tag memory wraps over to the first page
  Note NTAG I²C: Reading over the end of the memory returns zero filled bytes

NTAG Fast_Read (3Ah)

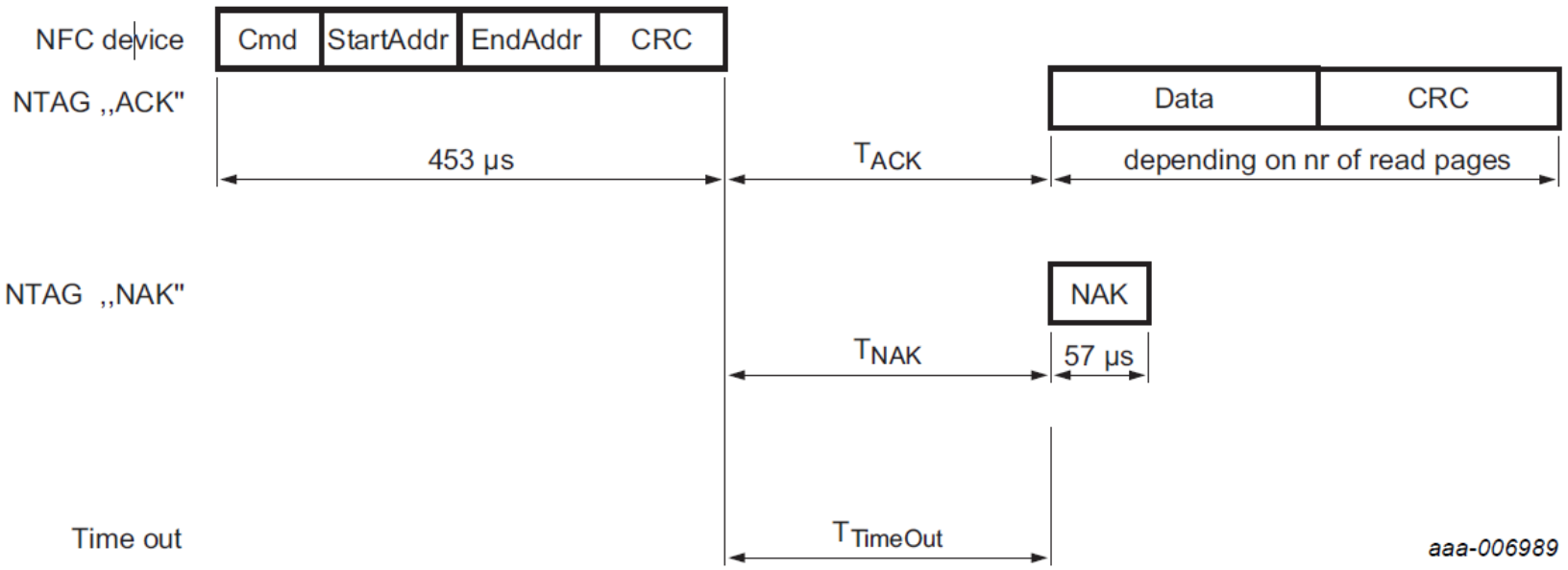


Fig 16. FAST_READ command

- With FAST_READ one entire memory sector can be read with one command

Note Ultralight: Reading over the end of the tag memory wraps over to the first page

Note NTAG I²C: Reading over the end of the memory returns zero filled bytes

NTAG Write (A2h)

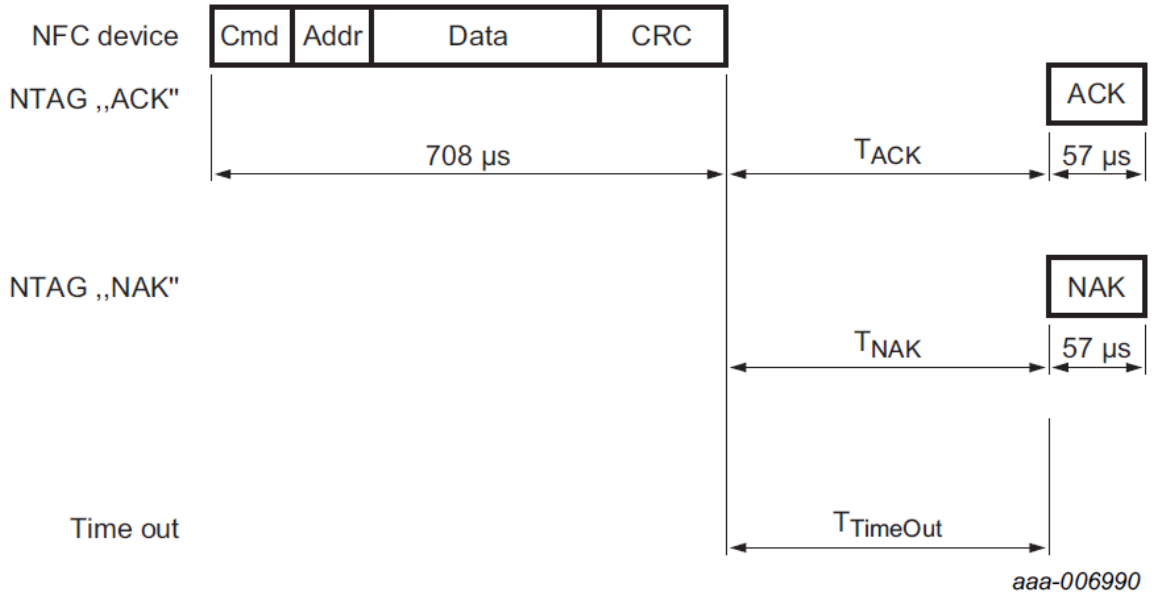


Fig 31. WRITE command

WRITE command needs page address and 4 bytes of data as parameters

NTAG Get_Version (60h)

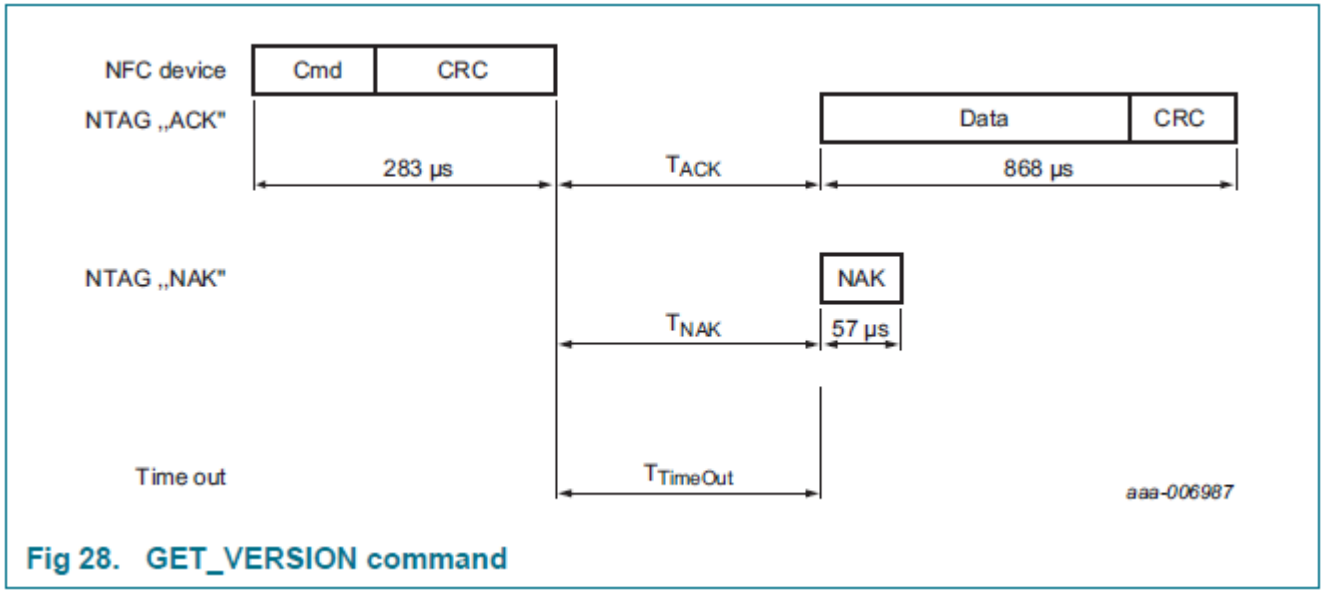


Fig 28. GET_VERSION command

- Provides 8 byte of identification data to exactly identify which NXP-IC it is

NTAG Sector_Select (C2h)

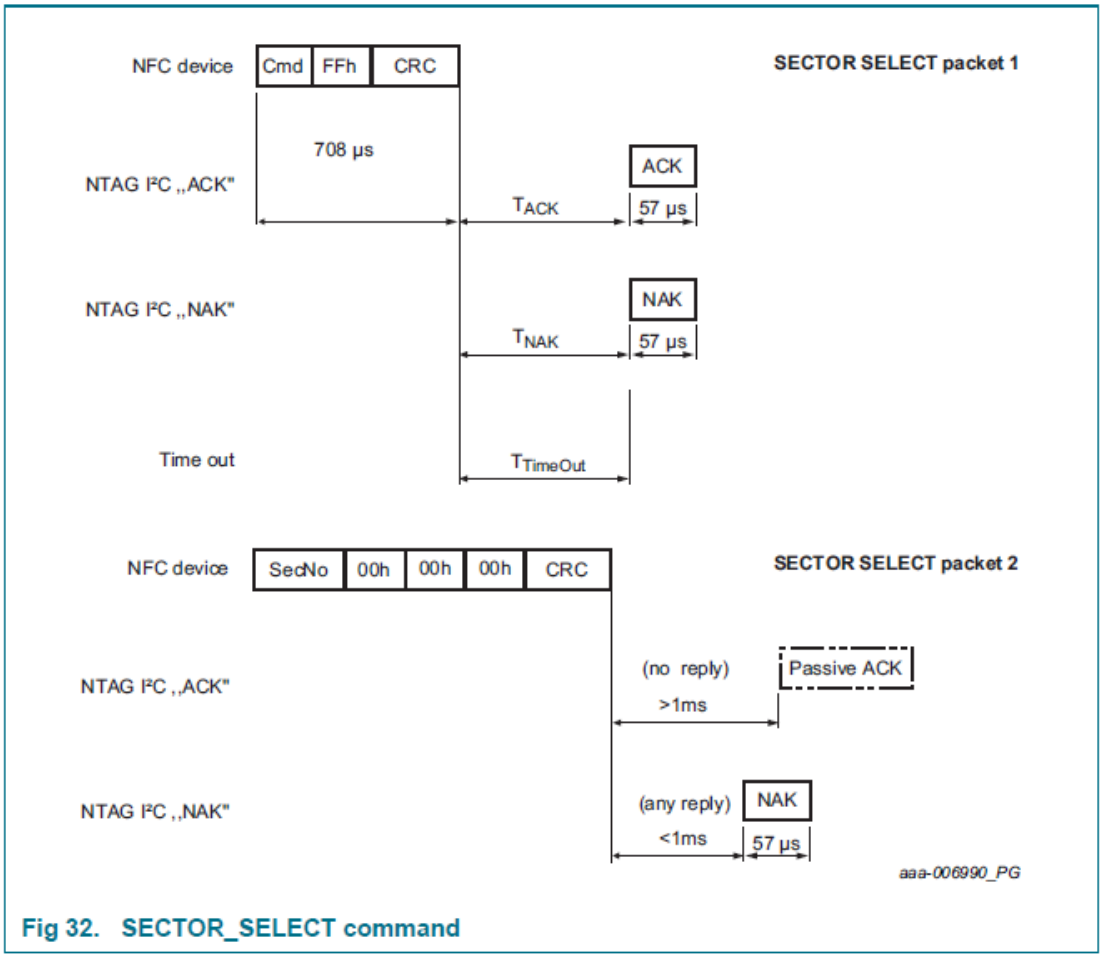
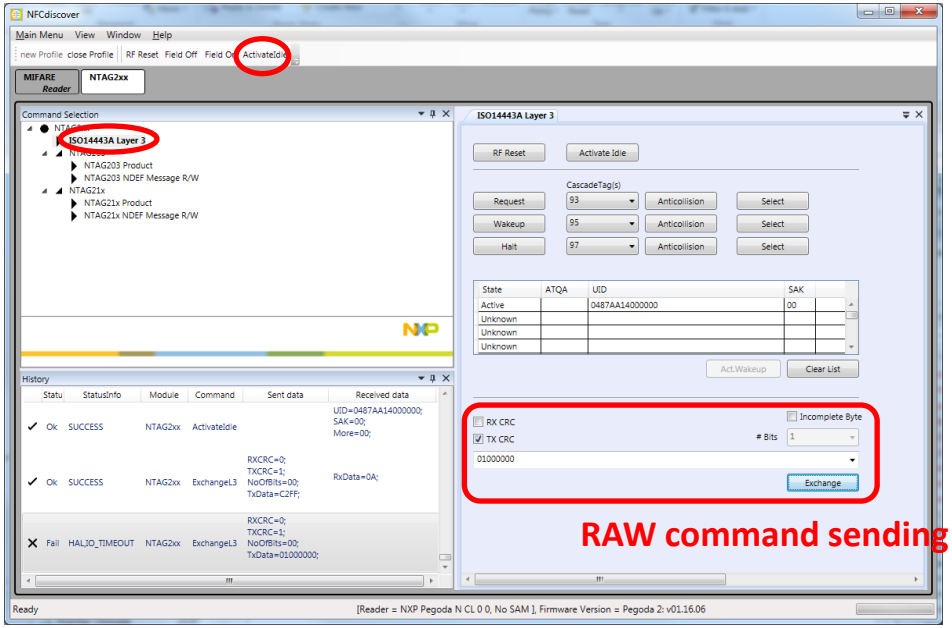


Fig 32. SECTOR_SELECT command

- Changes active memory sector which is accessed by read/write
  Note: second part of command has a passive ACK – no response is ACK!

NFC Discover: sending the sector_select

- Press Activate Idle if not done yet
- In 14443A-Layer 3 Window:
  - Deselect RX-CRC
  - Enter C2FF in command field
  - Click Exchange – tag sends ACK
  - Enter sector number + 3x „00“, e.g. 01000000 for sector 1
  - Click Exchange – tag sends nothing! Ignore error in NFC Discover

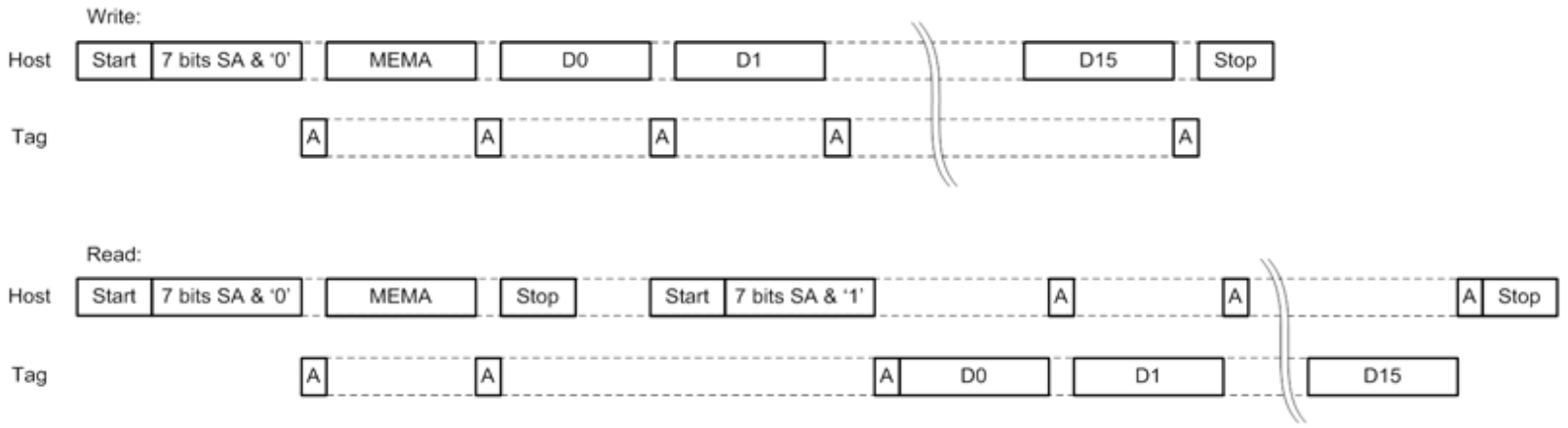


- You can now read/write in sector 1 until the next tag activation

I²C interface

I²C Communication on NTAG I²C Memory Access

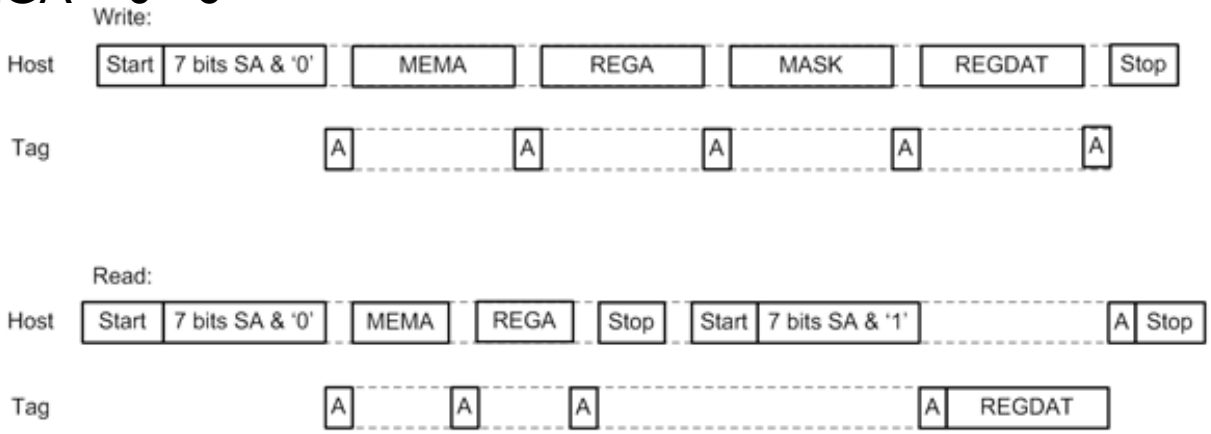
- For accessing memory and registers there are two different commands
- I²C Memory Access
  - access the EEPROM and SRAM
  - read/write always 16 byte



Note: If I2C_RESET_ON_REPEATED_START is set, don‘t forget the STOP after the first part of the read command (I²C Bird does not send STOP with GUI)

I²C Communication on NTAG I²C Register Access

- I²C Register Access:
  - access the Session registers located in 0xFE (I²C block address) read/write always only one byte
  - MEMA = 0xFE
  - REGA = 0 - 6

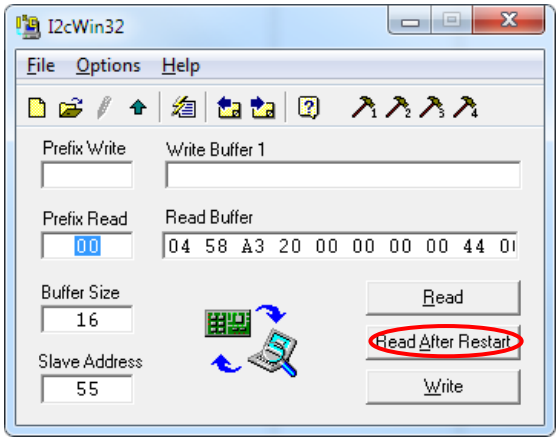


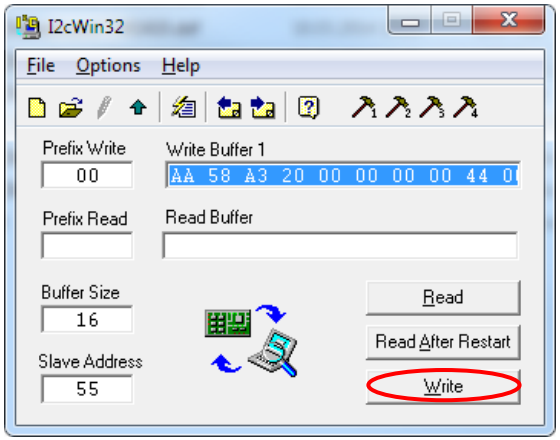
Note: If I2C_RESET_ON_REPEATED_START is set, don‘t forget the STOP after the first part of the read command (I²C Bird does not send STOP with GUI)

NTAG I²C Communication with I²C Bird

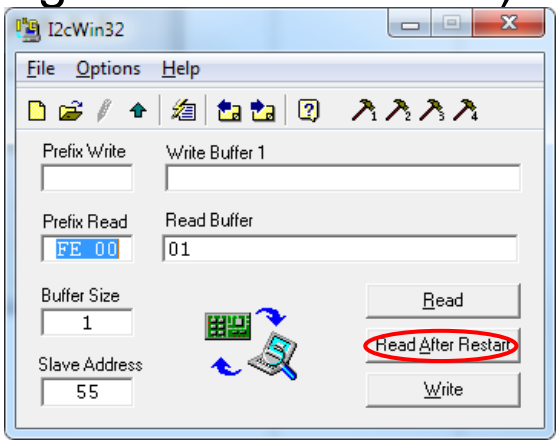
- Connect the Bird with 3V levelshifter to the tag (e.g. from UCODE I²C)

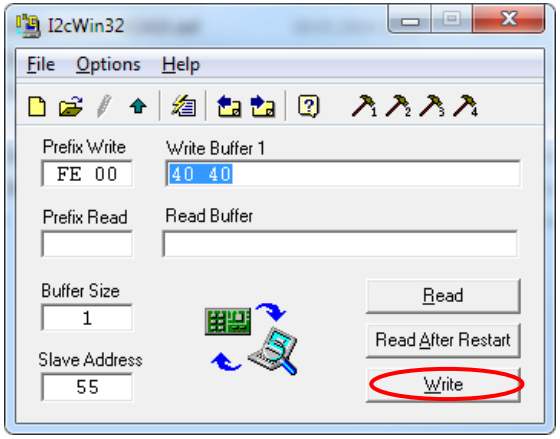
Read/Write Memory (always 16 byte blocks)





Read/Write Register (always one byte)



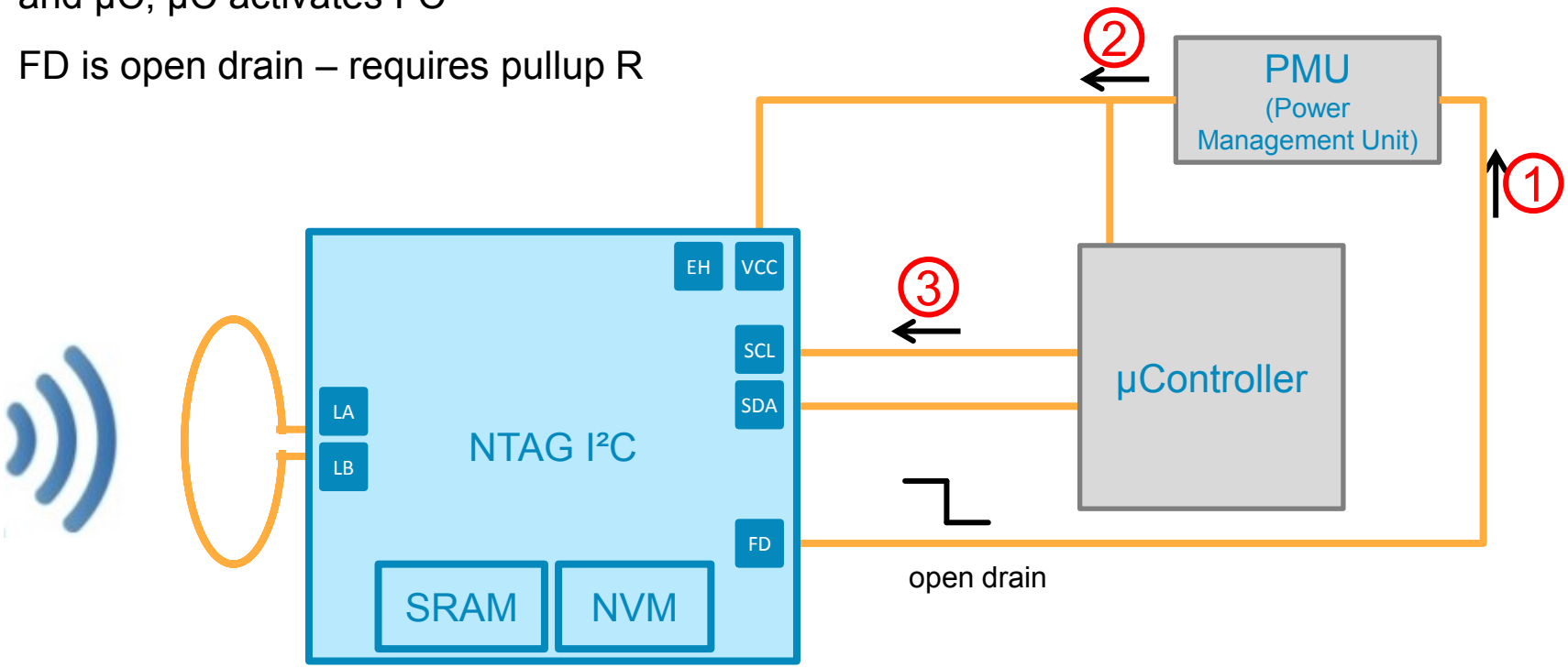


Note: If I2C_RESET_ON_REPEATED_START is set, don‘t forget the STOP after the first part of the read command (I²C Bird does not send STOP with GUI)

Special feature (1): Field Detection

Wake up on field detect - Passive tag mode

- For battery saving, the whole system can be powered up upon Field Detection
- FD goes low when field is detected; PMU receives interrupt, switches on VCC on NTAG and µC; µC activates I²C
- FD is open drain – requires pullup R

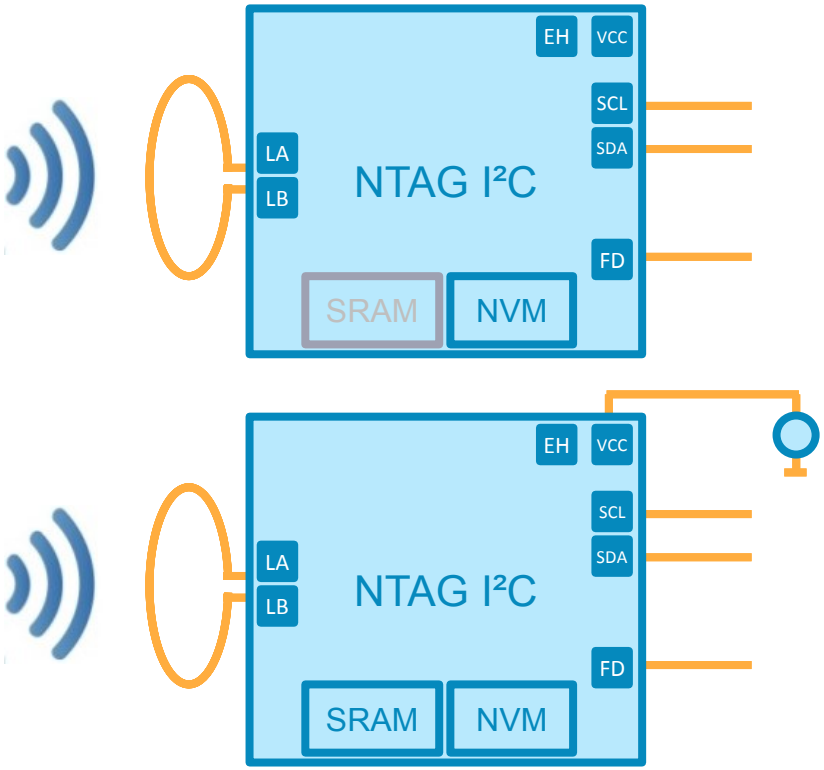


Field Detection Pin Configuration

- The ON and OFF-conditions for the FD pin can be configured:
  - FD_ON (FD goes low):
    - field is switched on
    - first valid SOF (retrigger only possible after RF-reset)
    - tag reaches state „ACTIVE“ (anti-collision done, retriggerable without RF-reset)
    - In pass-through-mode:
      - RF→I2C: data is ready for I2C to be read
      - I2C→RF: data is ready for RF to be read
  - FD_OFF (FD goes high):
    - field is switched off
    - field off or tag in HALT state
    - field switched off or NDEF message got read based on register LAST_NDEF_BLOCK
    - pass-through-mode:
      - If FD_ON=0x3:
        - RF→I2C: last data read by I2C
        - I2C→RF: last data written by I2C
      - If FD_OFF!=0x3:
        - If the field is switched OFF

Special feature (2): SRAM

Embedded 64-byte SRAM



- Passive tag mode
  - Tag works like a normal Type 2 tag with EEPROM
  - SRAM not useable without supply on VCC
- VCC powered
  - SRAM accessible from I²C, from RF when enabled (SRAM Mirror or Pass-through)
  - SRAM available for
    - Fast data transfer
    - Unlimited endurance

Special feature (3): SRAM Mirror

SRAM Mirroring

- The SRAM can be mirrored into the user memory
  - Behaves like a overlay – each read and write from RF to that addresses is not executed on the underlying EEPROM, but on the SRAM
  - EEPROM content is not influenced
- Use-Case:
  - Dynamic Update of e.g. pairing information (write every some seconds a new key)
- Address is given by register SRAM_MIRROR_BLOCK
  - Use the address like seen from I²C
  - Only addresses with valid user memory are usable
- Hints:
  - Not combinable with pass-through-mode
  - The mirroring is only effective for the RF side, I²C has to use the fixed SRAM address

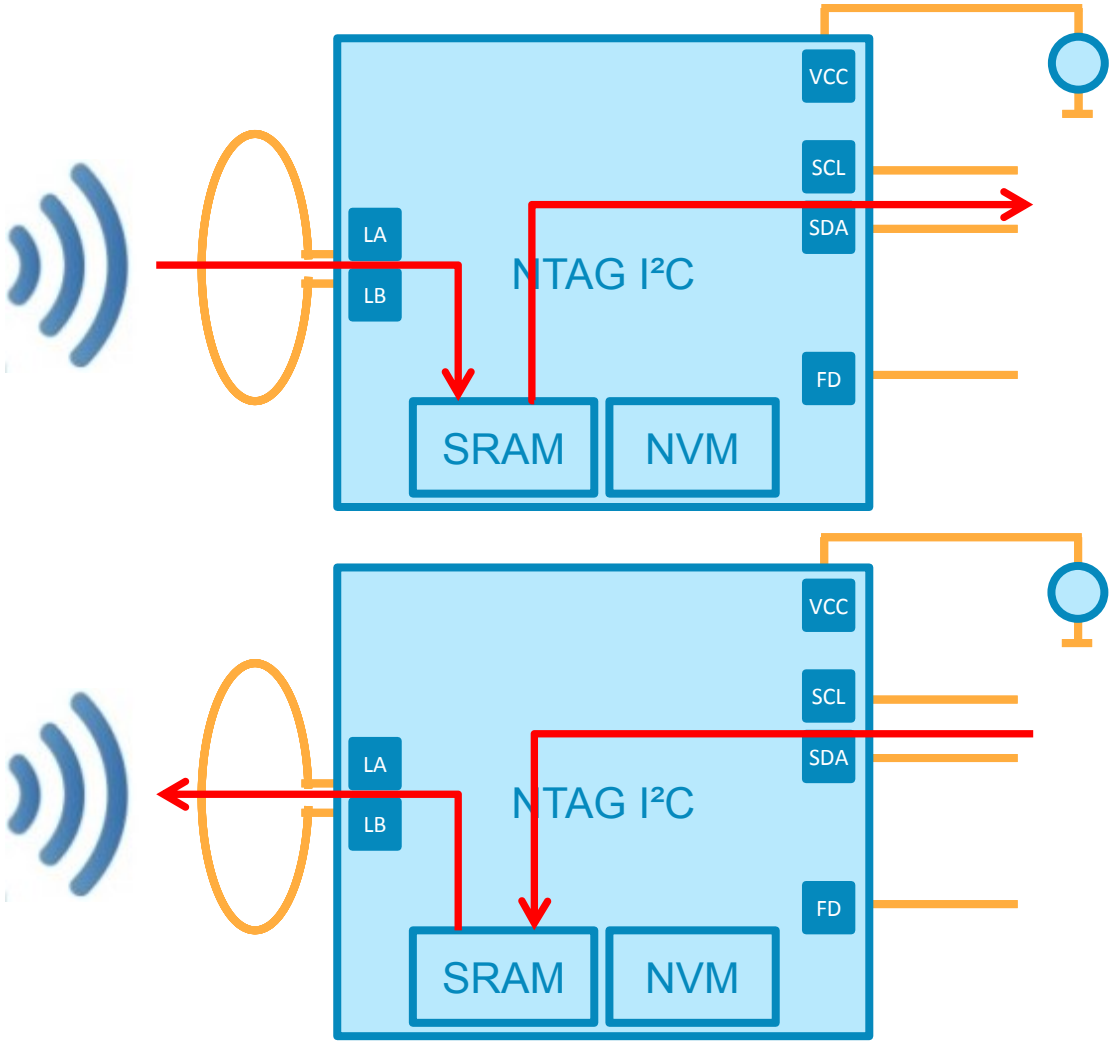
SRAM Mirroring – memory map illustration

- Example:
  - NTAG I²C 1k
  - SRAM_MIRROR enabled
  - SRAM_MIRROR_BLOCK = 0x01
- →SRAM Mirror starts on RF page 0x04 (Block address multiplied by 4)

Sector adr	Page address		Byte number within a page				Access
Hex	Dec.	Hex.	0	1	2	3	conditions
0h	0	0h	Serial number				READ
	1	1h	Serial number			Internal data	READ
	2	2h	Internal data		lock bytes		READ/R&W
	3	3h	Capability Container (CC)				READ
	4	4h	SRAM memory (16 pages) with memory mirror mode enabled only with SRAM_MIRROR_ADDR set to 1h				READ & WRITE
		...					
	19	13h					
	....	....	User memory				
	225	E1h					
	226	E2h	Dynamic lock bytes			00h	R&W / R
	227	E3h	Invalid access - returns NAK				n.a.
	228	E4h					
	229	E5h					
	230	E6h					
	231	E7h					
	232	E8h	Configuration				See section 8.5.9
	233	E9h					
	234	EAh	Invalid access - returns NAK				n.a.
	...	...					
	240	F0h	SRAM memory (16 pages)				READ & WRITE
	....	....					
	255	FFh					
1h	....	....	Invalid access - returns NAK				n.a.
2h	...	...	Invalid access - returns NAK				n.a.
3h	0	0h	Invalid access - returns NAK				n.a.
	....	....					
	248	F8h	Session registers				See section 8.5.9
	249	F9h					
	...	....	Invalid access - returns NAK				n.a.
	255	FFh					

Special feature (4): Pass Through

Pass-through functionality



- Use case:
  - Phone writes data (e.g. configuration) into µC
  - µC-Firmware update
- Use case:
  - Download of logging-Data
  - Service Information
  - Error descriptions

Pass-Trough mode features

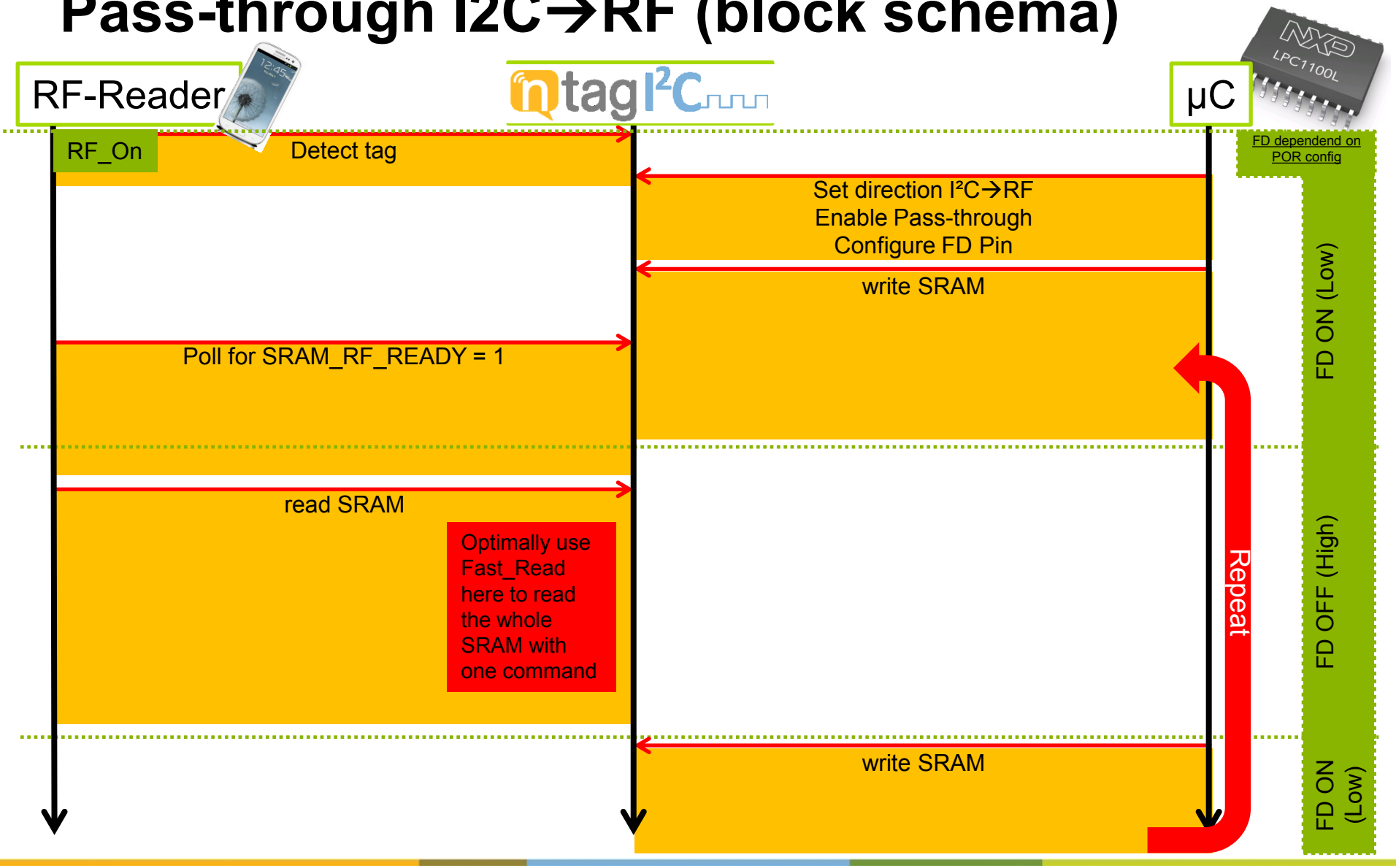
- Passthrough automates signalling for the arbitration of RF and I2C side. It provides help to speed up the communication.
  - A read/write of the last SRAM page/block triggers the side switch

- Synchronization signalling on I²C-side can be done either by:
  - Field detect pin
  - Polling on register NS_REG (bit SRAM_I2C_READY and SRAM_RF_READY)
- Synchronization signalling on RF side can be done by
  - Polling on register NS_REG (bit SRAM_I2C_READY and SRAM_RF_READY)
  - Timing, when µC-timing is reproducible

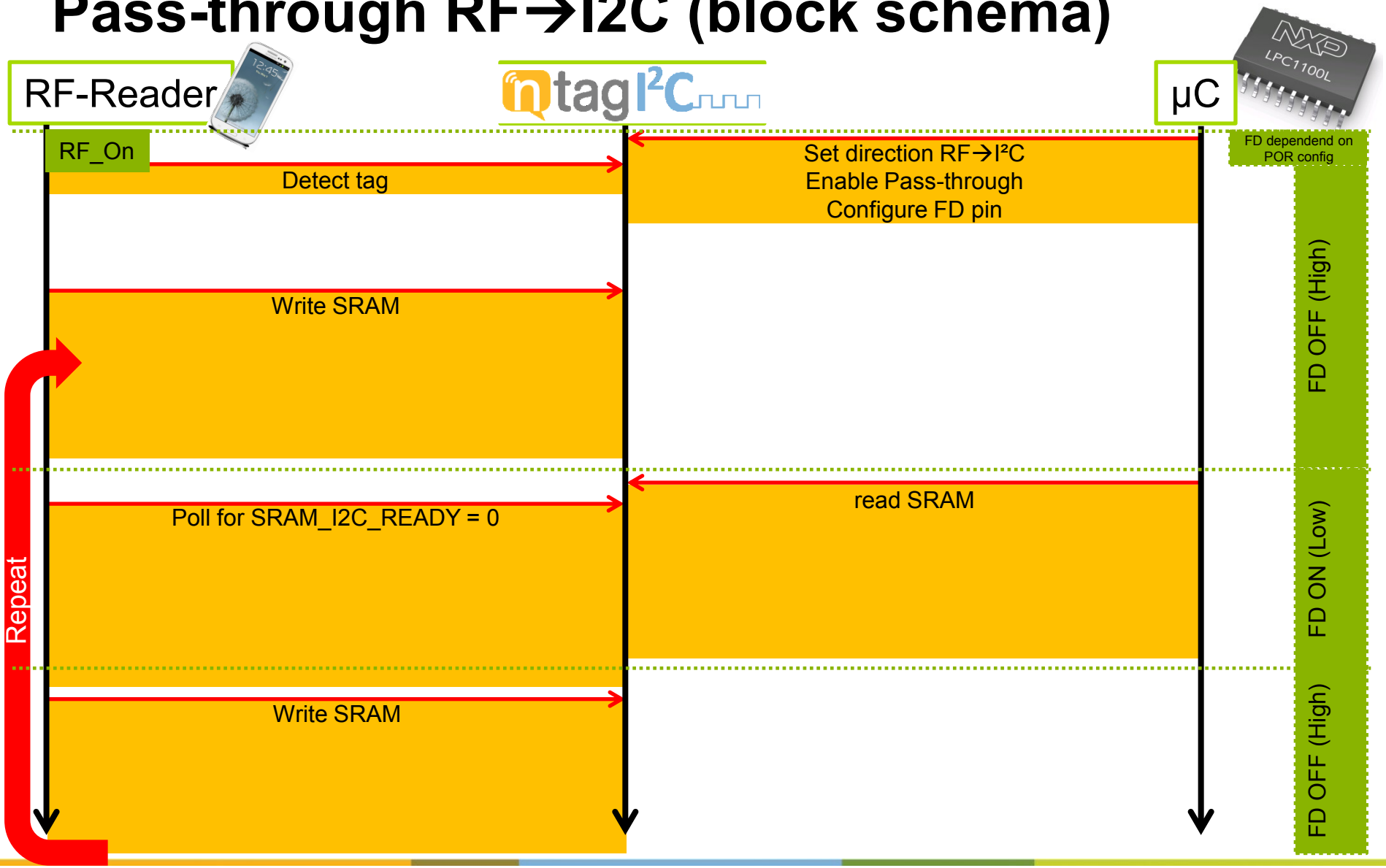
Pass Through Mode
Command flow

- The following slides illustrate the communication in pass trough mode
- shown is the mode with:
  - synchronization RF-side: polling
    - Wait for SRAM_RF_READY = 1 → Data ready in SRAM for RF to be fetched
    - Wait for SRAM_I2C_READY = 0 → Data was fetched from SRAM by I²C
  - synchronization I²C-side: Field detect pin
- Slides with title „block schema“ shows the basic communication blocks
- Slides with title „details“ shows commands in more depth, but not all

Pass-through I2C→RF (block schema)



Pass-through RF→I2C (block schema)



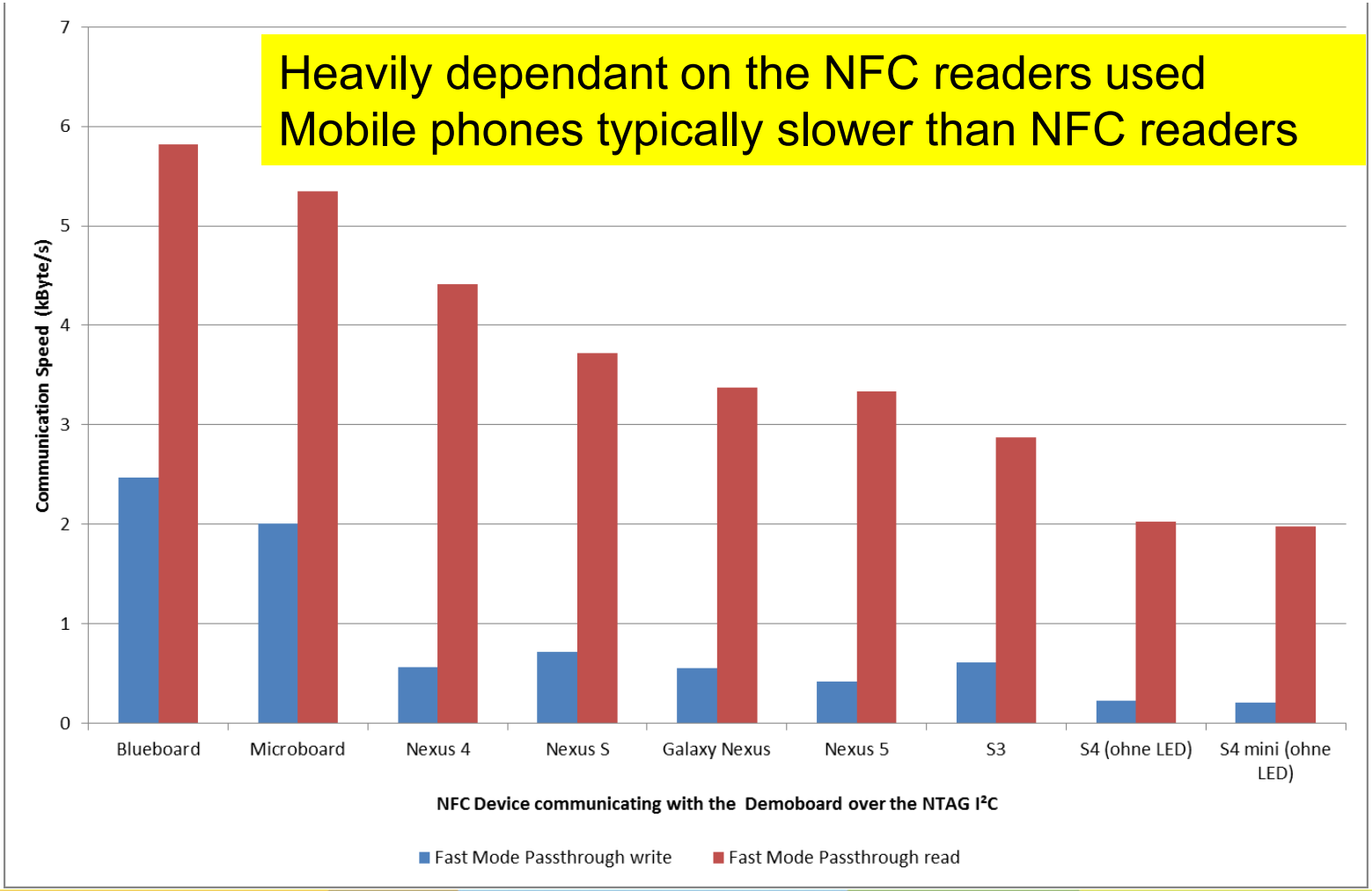
Pass through implementation hints

- **Pass through and SRAM Mirror activation**
  It needs to be (re-)enabled in the session register after every power down of RF or I2C, it cannot effectively enabled in the configuration to be on automatically on every power up
- **Memory locked to one side in pass through**
  When the pass-through mode is enabled, the arbiter locks the memory access either to RF or I2C – the other side can the access the memory only, when the active side was changed by the arbiter (due to a read/write to the terminator page in SRAM)
  - In case of communication with mobile phones the time where I2C accesses the memory has to be kept as short as possible, because some phones need regular memory access to keep RF active – if the memory access fails, RF is switched off and pass-through is automatically disabled
- **Use FAST_READ:** Readout from SRAM with RF can be speed up by using FAST_READ to read the whole SRAM with one command. FAST_READ can **not** be used to read the registers.

Pass through implementation hints

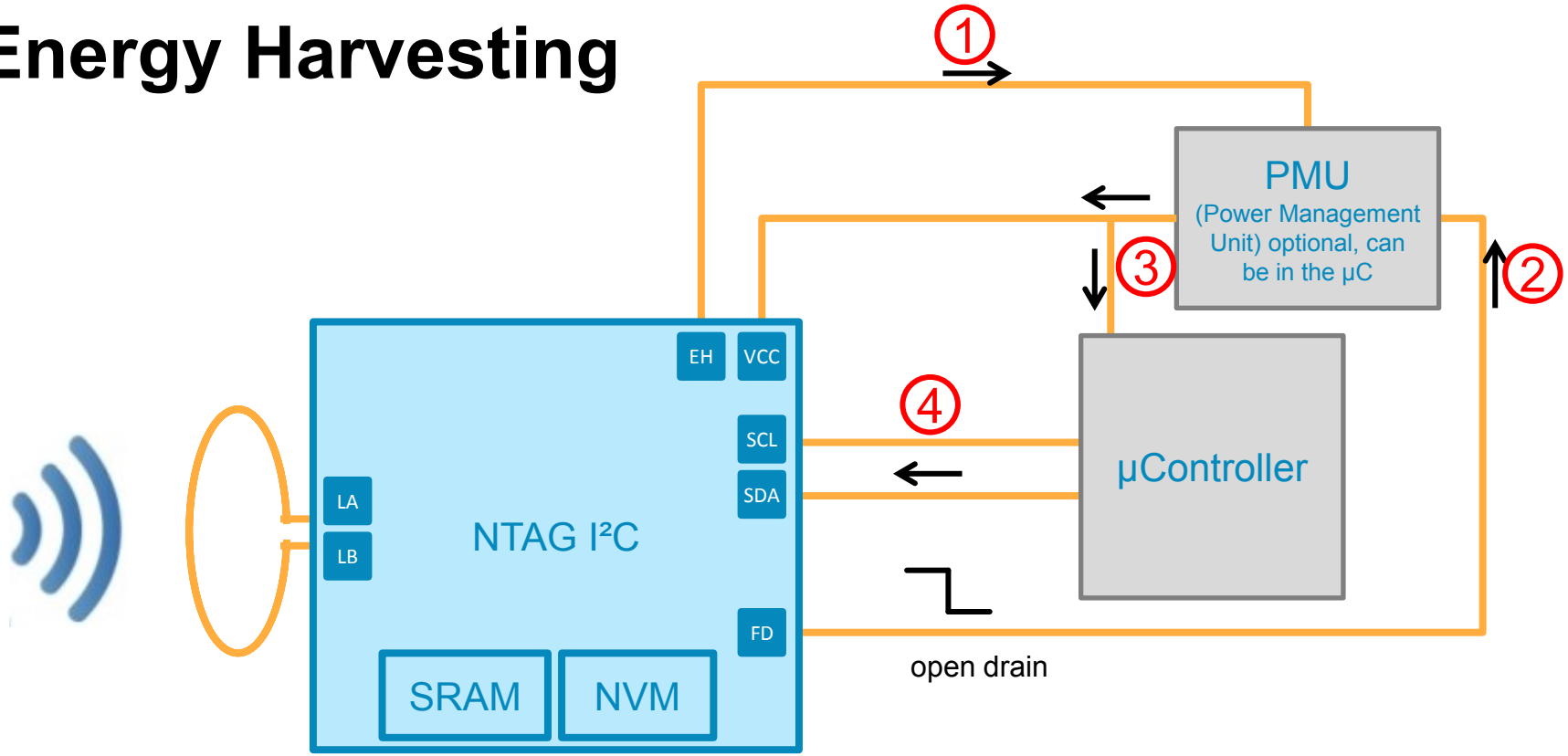
- **Bit PTHRU_DIR in NC_REG:** Double function, is also write enable in non-passthrough-mode. If set to 0, memory is readonly from RF
- **Bidirectional communication:** before changing the passthrough direction with PTHRU_DIR, always disable PTHRU_ON_OFF first
- **Fastest combination:**
  - Use the FD pin on the I²C side
  - On RF use polling only to detect the start of the transfer and do the following transfers based on timing

Pass-through – Speed Test



Special feature (5): Energy harvesting

Energy Harvesting

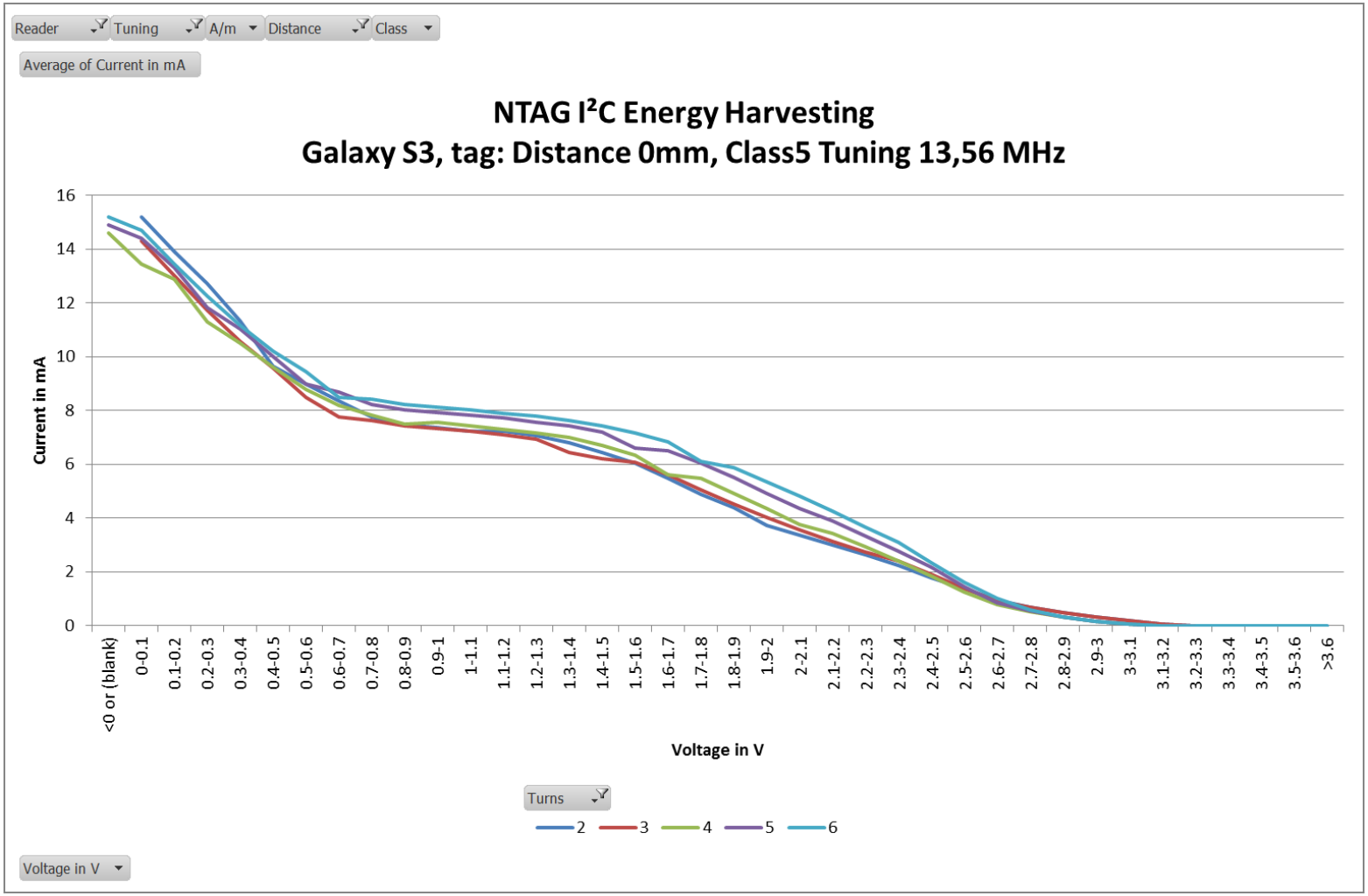


- With energy harvesting a microcontroller can be supplied with power
  - Typically >5mA @2V
  - Up to 13mA in case of strong NFC readers (not NFC phones)
- Power up can be controlled by Field Detection for further power management optimization
  - FD goes low when field is detected; PMU receives interrupt, switches on VCC on NTAG and µC; µC starts communication on I²C

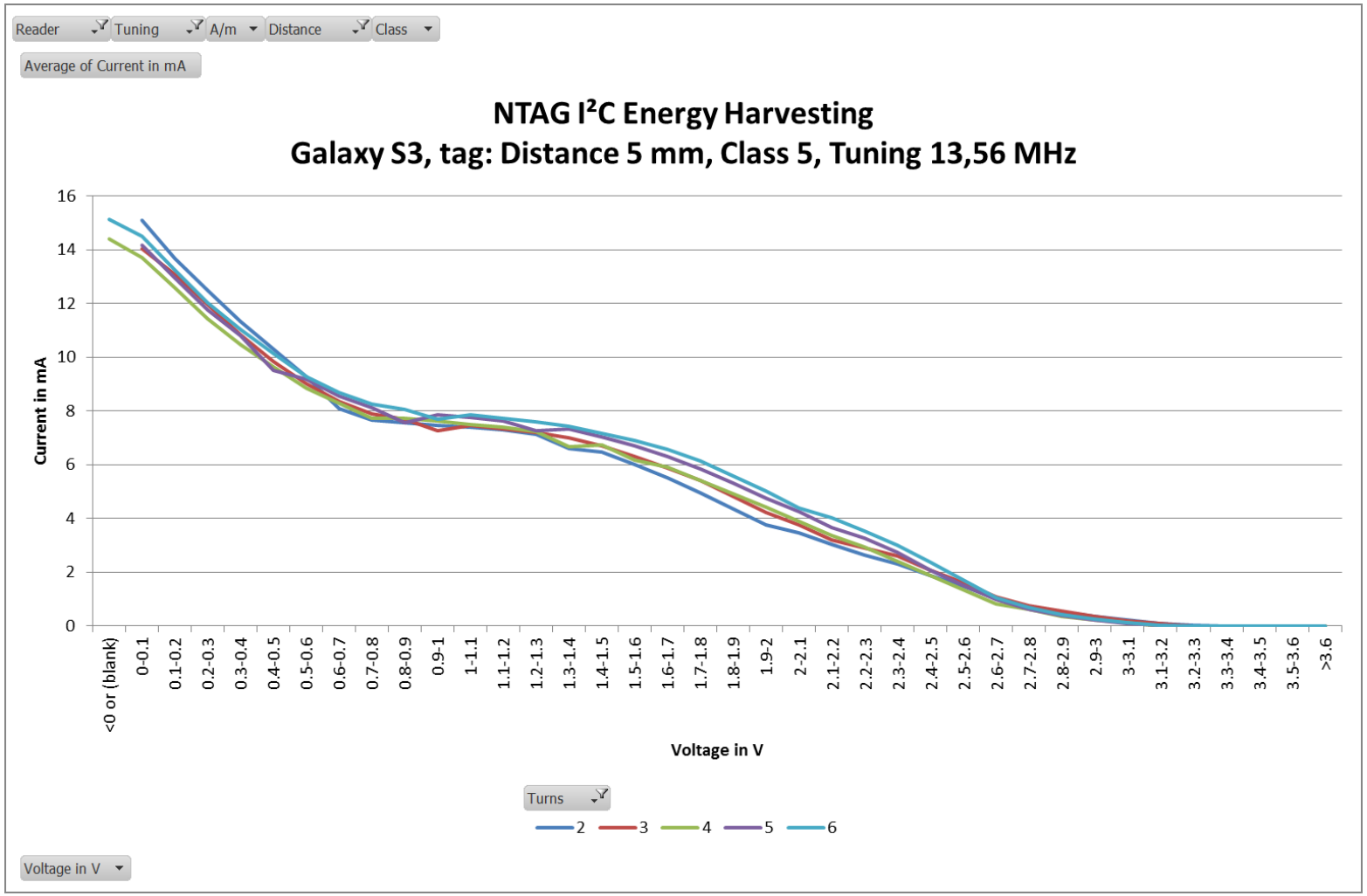
Influence factors on Energy Harvesting

- Reader:
  - Field strength
  - „Stiffness“ of the reader
- Antenna Geometry
  - Size
  - Antenna Turn Count
  - Tuning (low influence)
- A higher turncount improves Energy Harvesting output power on mobile phones in close coupling
- Reducing the turncount improves the stability of energy harvesting in the distance
- Application Note in preparation, ready: R-Gate, CW1431

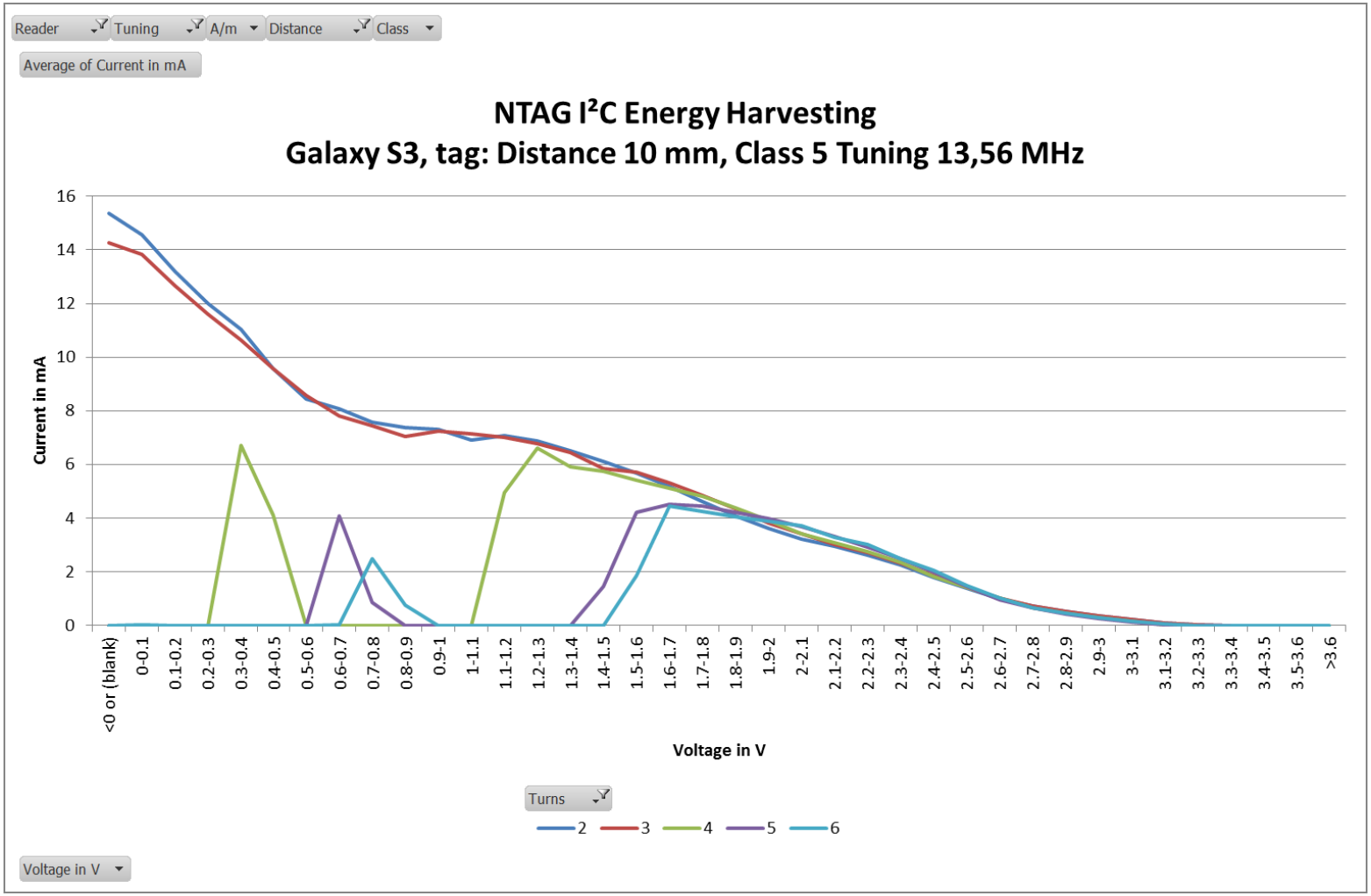
Energy Harvesting – Achievable Power



Energy Harvesting – Achievable Power



Energy Harvesting – Achievable Power



Memory Arbitration

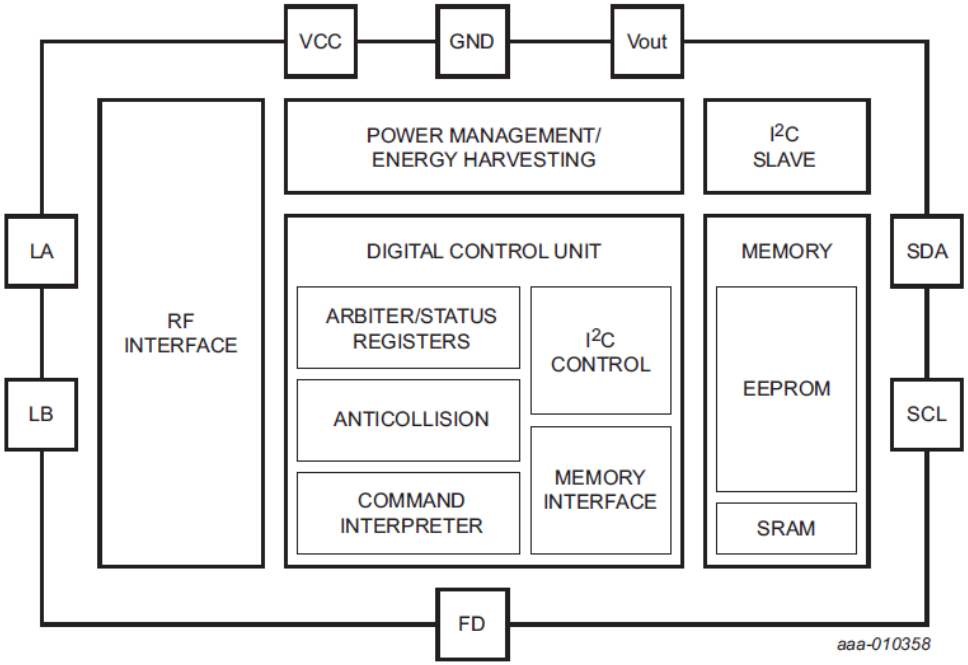
NTAG I²C Memory Arbitration

- Registers can always be read/written
- RF and I²C cannot access the EEPROM / SRAM at the same time

→Arbiter controls access to the memory

Arbiter status in NS_REG:

- RF_LOCKED
  →RF Access to Memory OK
  →I2C has no access
- I2C_LOCKED for I2C
  →I²C access to memory OK
  →RF has no access



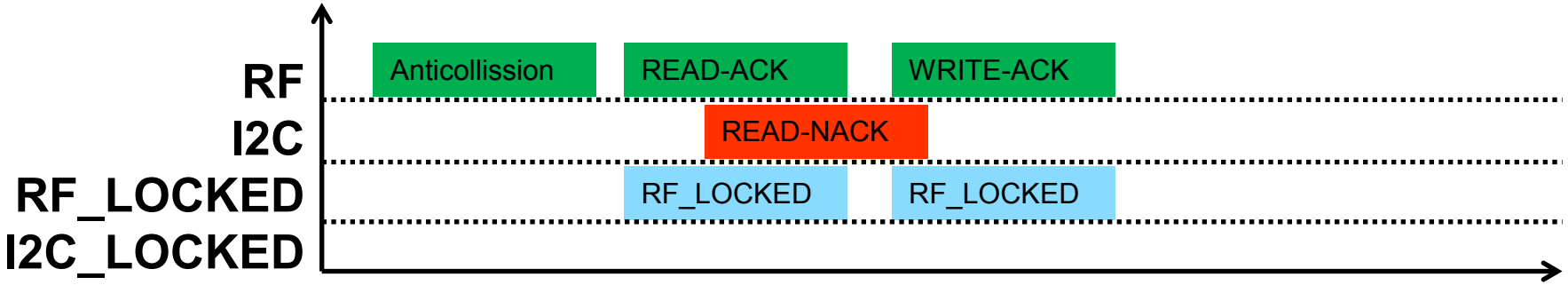
NTAG I²C Memory Arbitration
Normal mode (non-passthrough)

- Each RF read/write on memory sets RF_LOCKED during the handling of the RF command. It clears directly afterwards.
- Each I²C read/write on memory sets I2C_LOCKED. To prevent intermittent RF accesses it stays locked after the command. It either needs to be cleared manually by
  - Sending a different slave-address on I²C as the NTAG I²C has
  - Writing 0 to I2C_LOCKED over I²C
- Or it gets automatically unset by the Watchdog timer
  - Default timeout: 20 ms

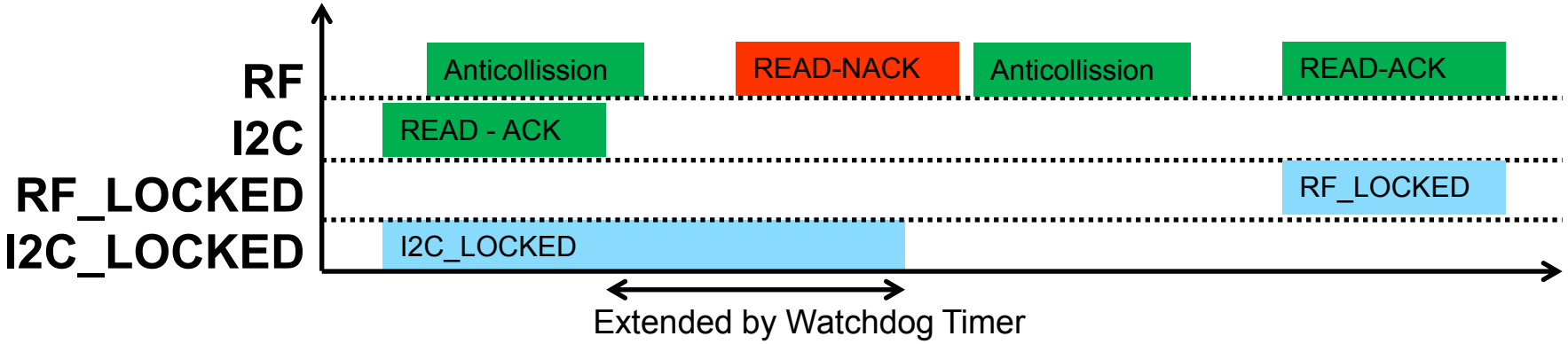
Reason: the delayed clearing of I2C_LOCKED is to prevent RF accesses on partially updated memory

NTAG I²C Memory Arbitration Normal mode (non-passthrough)

- RF-Access:



- I2C-Access:



NTAG I²C Memory Arbitration Pass Through Mode

- Access to EEPROM and SRAM has the same locking bit – when the SRAM is accessed by one interface, also the EEPROM is not accessible by the other interface
- the arbiter locks the memory access exclusively to the current active side.
- The other side can meanwhile only access the registers
- Two additional bits signal the availability of new data in the SRAM:
  - SRAM_I2C_READY: Data is ready in SRAM to be fetched by I2C
  - SRAM_RF_READY: Data is reay in SRAM to be fetched by RF
- The arbiter switches automatically the active side and sets/clears the SRAM_xx_READY flags, when the last page/block of the SRAM was read/written

Product support package

NTAG I²C
General Support Tools

- PC-Software
  - NFC Discover (usable with Pegoda)
    - http://www.nxp.com/documents/software/SW275410.zip
- Android general
  - NFC TagInfo by NXP (in Google PlayStore)
    - Identify tag and read memory
  - NXP TagWriter by NXP (in Google PlayStore)
    - Store NDEF messages
- Android: “NTAG I2C Demoboard”
  - Controlling LEDs and showing communication speed available on Google Play store „NTAG I2C Demoboard“

NTAG I²C Hardware

- Antenna board (also called Evalboard)
  - Only NTAG I²C connected to Antenna (Class 5)
  - Can be connected to demo board or any I²C interface (e.g. I2C Bird)
- Demoboards:
  - LPC812 Demoboard
  - NTAG I²C Explorer Kit Rev 1 and 2
- Development board:
  - Both demoboards have programming connectors
- Source code of demos available
  - For µC
  - For Android

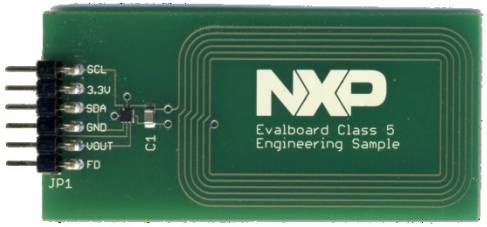
NTAG I²C Chip and Eval Board Generations

- V0B / Wave#1
  - UID „04 xx yy 13 00 00 00“
- V0C / Wave#2
  - UID “04 xx yy 14 00 00 00”
- V0D / Wave#3
  - UID “04 xx yy 08 00 00 00”
  - UID “04 xx yy 10 00 00 00”
  - UID “04 xx yy 30 00 00 00”
- V0E / Wave#4
  - UID „04 xx yy 18 00 00 00“
  - UID „04 xx yy 20 00 00 00“
  - UID „04 xx yy 28 00 00 00“
- V0F / Wave#5
  - UID “04 xx yy 3A 7C 3B 80”

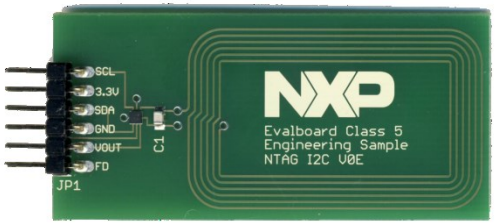
V0B

V0C

V0D



V0E

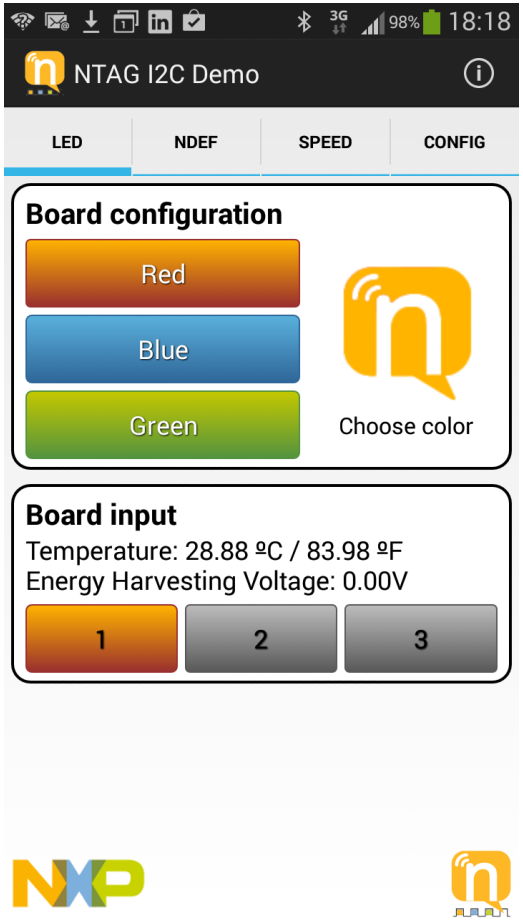


NTAG I²C Android Demo Application

- Available in Google Playstore
- Shows:
  - Passthrough Communication in both directions:
    - Set LED colour (interactive)
    - Read Buttons pressed and demoboard temperature
  - NDEF read/write
    - Depends on demoboard: show message content on LCD
  - Communication speed
    - Passthrough and EEPROM
  - NTAG I²C Configuration:
    - Memory Readout
    - Configuration Read/Write
    - Register Read
- Old version on Playstore to be replaced ~ next week
- Source Code: https://nxp.box.com/NTAGI2CDemoboardLPC812

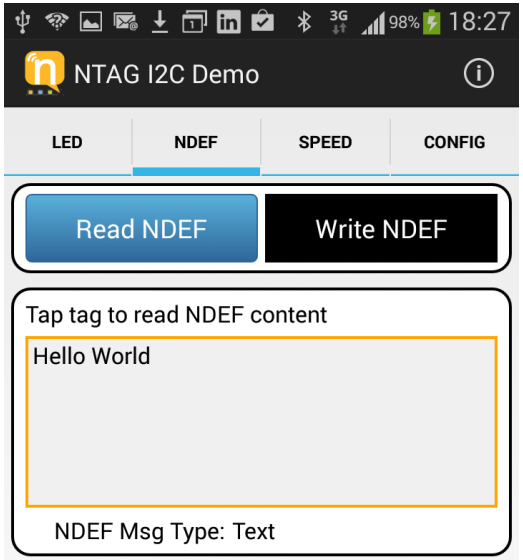
NTAG I²C Android Demo Application Screen: LEDDemo

- Select a LED colour which should light up on demoboard
- Reads back (if on demoboard):
  - Temperature
  - Energy Harvesting Voltage
- Communication is done in Passthrough Mode



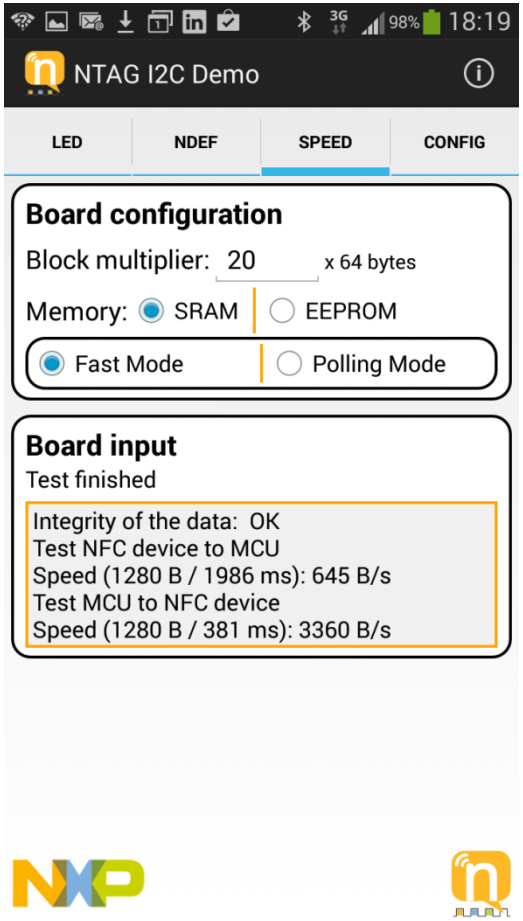
NTAG I²C Android Demo Application Screen: NDEF

- Allows to read NDEF-Messages from type text
- Write NDEF messages from type:
  - Text
  - URI
  - Bluetooth Pairing Information
- In case the demoboard has a display:
  - The LCD displays the written text



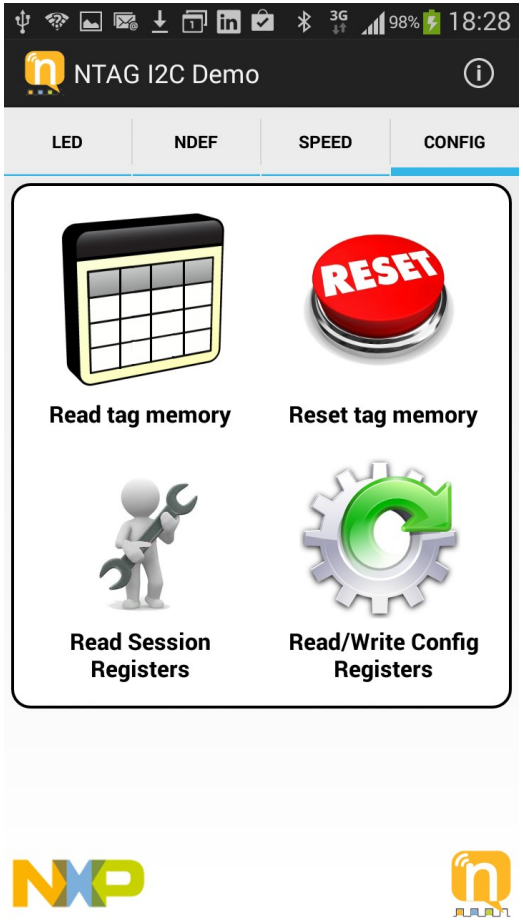
NTAG I²C Android Demo Application Screen: Speed

- Measures data troughput in the following modes:
  - SRAM fast: Data transfer Phone<->MCU without register polling (MCU timing needs to be known)
  - SRAM polling: Data transfer Phone<->MCU with register polling. Slower, but works independent of MCU timing
  - EEPROM: Read/Write to NTAG I²C EEPROM, MCU not involved (demo case: zero power configuration)

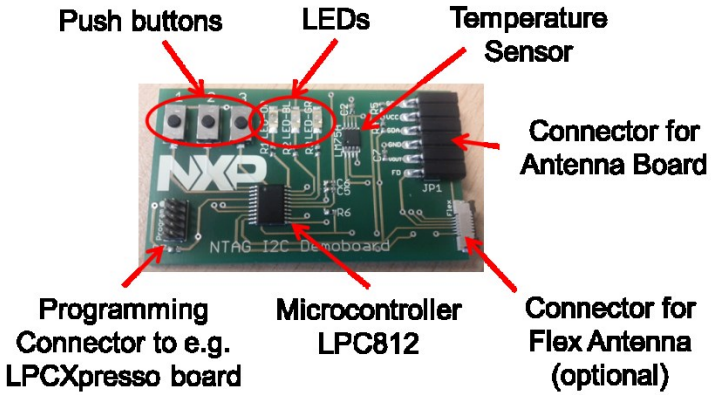


NTAG I²C Android Demo Application Screen: Config

- Allows to:
  - Dump Memory Content
  - Reset complete tag memory (if not locked)
  - Read Session Registers
  - Read/Write Configuration



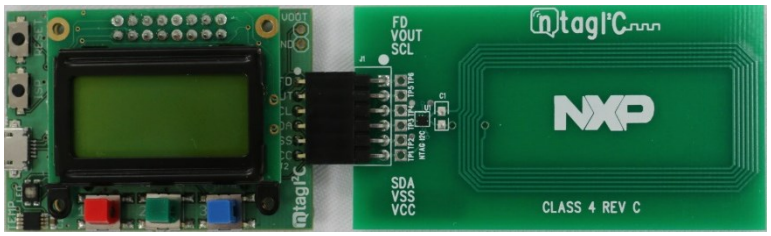
NTAG I²C LPC812 Demoboar



- To be connected to a NTAG I²C Antennaboa
- Shows:
  - Energy Harvesting
  - LED Demo with interactive LED control and Button + Temp feedback
  - Zero-Power Configuration (switch LED on depending on NDEF)
  - SRAM and EEPROM speed measurement
- Code for LPC and Android downloadable: https://nxp.box.com/NTAGI2CDemoboardLPC812
- Will be put on public Internet at release
- Availability: on eDemoboard ~CW36

NTAG I²C Explorer Board

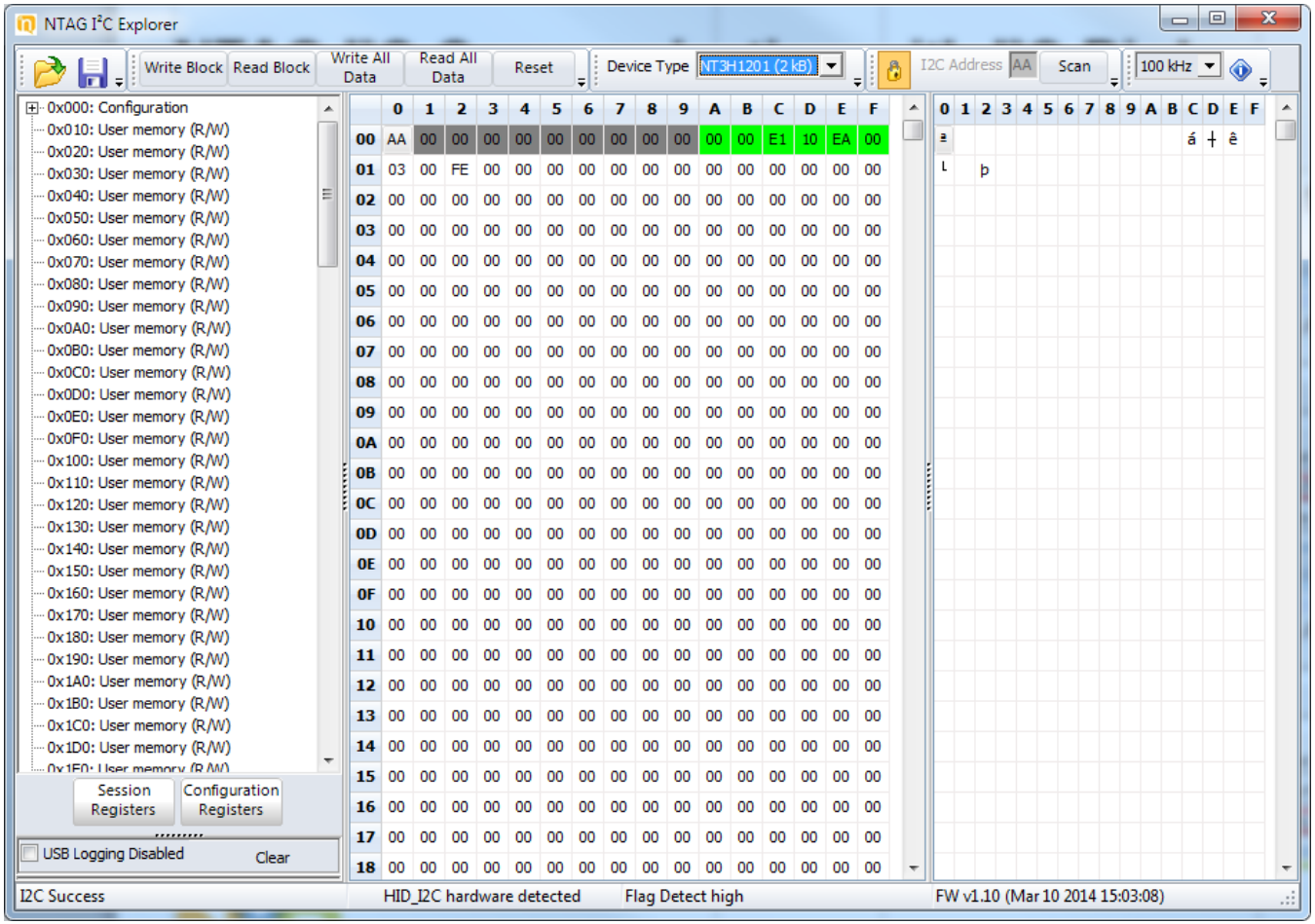
- Components:
  - LPC11U24
  - RGB-LED
  - Buttons
  - Temp-Sensor
  - LCD
  - micro-USB



- Shows:
  - Energy Harvesting
  - LED Demo with interactive LED control and Button + Temp feedback
  - Zero-Power Configuration (switch LED on depending on NDEF)
  - SRAM and EEPROM speed measurement
  - NDEF message read write with NDEF content display on LCD
  - I²C memory read/write with a windows GUI (I²C-USB-Bridge)

NTAG I²C Communciation with NTAG I²C Explorer PC GUI

- LPC µC used as bridge USB<->I²C
- Needed Firmware integrated in NTAG I²C Explorer Kit



NTAG I²C Application Notes

- How to lock the tag to readonly
- How to optimize Energy Harvesting
- How to use the Passthrough Mode
- Manual for the Android Demo Application
- Porting guide for general NTAG_UCODE-API

Software Issues

Known Software Issues with NFC Devices

- **Memory Initialization:**
  - To improve NFC Device compatibility the Memory is initialized without Lock Control TLV. Effect: Locking a tag with automatic lock functions defined by NFC Forum only locks a part of the tag.
    Details: AN11456: Using the dynamic lock bits to lock the tag
- **Sector Select:**
  - Automatic read of NDEF messages which are larger as ~1 kByte does not work on all NFC devices in the market. It is recommended to use the corresponding NDEF message read functions in the MIFARE SDK
- **Sector Select Timeout:**
  - When sending the sector select command with NFC devices it is necessary to reduce the timeout as the second part of the command has no answer. Otherwise the command execution gets very slow. On some NFC devices this timeout cannot be changed and a sector select can take up to 5 seconds time

Thank you!