NUR APPLICATION NOTE 2 (NUR AN002)

GETTING STARTED: MODULE SETUP

Contents

Nordic ID Oy | Myllyojankatu 2 A | FI-24100 Salo | Finland
Office +358 2 727 7700 | Fax + 358 2 727 7720 | info@nordicid.com

www.nordicid.com | www.rfidarena.com | Facebook | Twitter | YouTube

SCOPE

This application note extends the NUR protocol documentation by showing few different module setup scenarios. For general command structure see the NUR application note 001 where the basic commands are explained (NUR_AN001_basic_commands.pdf).

These scenarios apply to module FW version 4.7-A and on; module is either NUR05WL2 or NUR10W.

Scenario	Description
Get all setup fields	Simple command that shows the setup reading using all the setup flags. The setup flags apply to module FW
Set basic setup	A very basic setup setting: • TX level • RX decoding • Link frequency • TX modulation • Antenna enabling • Antenna selection
Default inventory settings	Getting and setting the • Q value • session • inventory rounds
Setup store	Simple command to store the current settings into the module's non-volatile memory.
Resetting to factory defaults	Command to use in order to set the module to the factory defaults.

GET ALL SETUP FIELDS

GET ALL SETUP FIELDS PACKET

A5 07 00 00 00 5D 22 FF FF FF 01 25 CA

THE PACKET IN C

```c
const uint8_t getSetupCmd [] =
{ 0xA5,0x07,0x00,0x00,0x00,0x5D,0x22,0xFF,0xFF,0xFF,0x01,0x25,0xCA };

struct __packed NURCMDHDR
{
  uint8_t start;    /* 0x0A5 */
  uint16_t payLen;  /* Following payload length including CRC-16.*/
  uint16_t flags;   /* Protocol/command flags. */
  uint8_t char cs;  /* Header checksum. */
};

struct __packed GETSETUPCMD
{
  struct NURCMDHDR hdr;
  uint8_t setupCmd; /* 0x22 */
  uint32_t flags;   /* 0x01FFFFFF for all. */
  uint16_t crc;
};
```

GET SETUP COMMAND CONTENTS

Field	Value	Description	
Header	A5070000005D (6 bytes)	Header consisting of:	
		A5	
		0700 = 0x0007	Payload + CRC length
		0x0000	Command flags
		0x5D	Header check sum
Command	0x22	Load module setup.	
Flags	FF FF FF 01	Value is 0x01FFFFFF. See all setup flags.	
Payload CRC	0x25 0xCA	Little endian; value is **0xCA25**.	

Nordic ID Oy | Myllyojankatu 2 A | FI-24100 Salo | Finland
Office +358 2 727 7700 | Fax + 358 2 727 7720 | info@nordicid.com
www.nordicid.com | www.rfidarena.com | Facebook | Twitter | YouTube

EXAMPLE OF MODULE SETUP RESPONSE

This is an example response from a Stix module.

THE GET SETUP RESPONSE PACKET IN HEXADECIMAL

The response consists of 61 bytes of which 6 bytes are header and the payload and CRC length is 55.

Bytes	Are
0...15	A5 37 00 00 00 6D 22 00 FF FF FF 01 00 E8 03 00
16...31	02 0E 00 00 06 00 01 01 F4 01 E8 03 00 00 00 00
32...47	00 00 FF 00 00 00 00 00 00 F4 01 F4 01 F4 01 F4
47...60	01 03 00 00 00 00 00 FF FF FF FF 8A CC

GET SETUP RESPONSE'S CONTENTS

Field	Value	Description	
Header	A5370000006D (6 bytes)	Header consisting of:	
		A5	
		3700 = 0x0037	Payload + CRC length (55)
		0x0000	Command flags
		0x6D	Header check sum
Command	0x22	Load module setup echo.	
Status	0x00	No error (0).	
Flags	FF FF FF 01	Value is 0x01FFFFFF. See all setup flags.	
Link frequency	00 E8 03 00	0x0003E800. The Link frequency in Hz (256000).	
RX decoding	0x02	0 = FM0; 1/2/3 = Miller 2/4/8 = $2^{<value>}$.	
TX level	0x0E	0x0E = 14. Value that is extracted from top level dBm (27 or 30). This case = 27 - 14 = 13dBm.	
TX modulation	0x00		
Region	0x00	Region ID = 0x00, ETSI/EU. See region identifiers.	
Q	0x06	Default inventory Q value. 0 means automatic.	
Session	0x00	Default inventory session value.	
Rounds	0x01	Default number of inventory rounds per call; 0 means automatic.	
Antenna mask	0x01	Antenna mask: bits 0…3 (values 1, 2, 4, 8) are used. Each bit corresponds to an antenna.	
Timeout 1	0xF4 0x01	0x01F4 = 500. Default "scan single" timeout in ms.	
Timeout 2	0xE8 0x03	0x03E8 = 1000. Default triggered (GPIO, light, tap) inventory timeout in ms.	
Antenna	0x00	Selected antenna is 0 (range: 0…3).	
Flags	00 00 00 00	Operation flags (none set, see operation flag description).	
Target	0x00	Inventory target (A=0, B=1, toggle A-B = 2)	
EPC length	0xFF	0xFF = accept all. Usable values are 2…62.	
Read RSSI	0x00 0x00	0x00 = disabled. The order is min, max. Type is signed char, unit is dBm.	
Write RSSI	0x00 0x00	Same as above for write. Disabled.	
Inventory RSSI	0x00 0x00	Same as above for inventory. Disabled.	
Read timeout	0xF4 0x01	Default tag read timeout in ms (0x1F4 = 500).	
Write timeout	0xF4 0x01	Default tag write timeout in ms (0x1F4= 500).	
Lock timeout	0xF4 0x01	Default tag lock timeout in ms (0x1F4= 500).	
Kill timeout	0xF4 0x01	Default tag kill timeout in ms (0x1F4= 500).	
Period setup	0x03	Internal inventory stream duty cycle setup.	
Antenna power	00 00 00 00	Per antenna power. 0 = full, 0xFF = not used (overridden by the TX level field).	
Power offset	FF FF FF FF	Per antenna power offset (1…1). Only the first value is used rest 3 are currently ignored so the first value is applied to all used antennas.	
Packet CRC	0x8A 0xCC	Little-endian; actual value is 0xCC8A.	

SET BASIC SETUP

This simple packet set the following default values:

- TX level (set to 3 = maximum – 3)
- RX decoding (set to 2 i.e. $2^2 \rightarrow$ Miller-4)
- Link frequency (set to 256kHz)
- TX modulation (set to PR-ASK)
- Antenna enabling (enable antennas 0 and 1, mask = 0x01 | 0x02 = 0x03)
- Antenna selection (set to automatic selection 0xFF)

BASIC SETUP PACKET

A5 10 00 00 00 4A 22 0F 09 00 00 00 E8 03 00 02 03 01 03 FF 45 0E

THE BASIC SETUP PACKET IN C

```c
const unsigned char basicSetupCmd[] =
{
  0xA5, 0x10, 0x00, 0x00, 0x00, 0x4A, 0x22, 0x0F,
  0x09, 0x00, 0x00, 0x00, 0xE8, 0x03, 0x00, 0x02,
  0x03, 0x01, 0x03, 0xFF, 0x45, 0x0E
};
```

Nordic ID Oy | Myllyojankatu 2 A | FI-24100 Salo | Finland
Office +358 2 727 7700 | Fax + 358 2 727 7720 | info@nordicid.com
www.nordicid.com | www.rfidarena.com | Facebook | Twitter | YouTube

EXAMPLE C STRUCTURE AND FLAG DEFINITION FOR BASIC SETUP

```c
#define BASICFLAGS \
(1 << 11 | 1 << 8 | 1 << 3 | 1 << 2 | 1 << 1 | 1)

struct __packed NURCMDHDR
{
  uint8_t start;     /* 0x0A5 */
  uint16_t payLen;  /* Following payload length including CRC-16.*/
  uint16_t flags;   /* Protocol/command flags. */
  uint8_t char cs;  /* Header checksum. */
};

// Note the order.
struct __packed BASICSETUP
{
  uint32_t flags;  /* Included setup flags */
  uint32_t lf;     /* Link frequency */
  uint8_t rxDec;   /* Receiver decoding */
  uint8_t txLevel; /* Transmission level */
  uint8_t txMod;   /* 0 =* ASK, 1 = PR-ASK */
  uint8_t antMask; /* *Antenna enable mask /
  uint8_t antenna; /* Selected antenna */
};

struct __packed BASICSETUPCMD
{
  struct NURCMDHDR hdr;
  uint8_t setupCmd;   /* 0x22 */
  struct BASICSETUP basic;
  uint16_t crc;  /* Calculate starting from command */
};
```

SET BASIC SETUP PACKET CONTENTS

Field	Value	Description		
Header	A5100000004A (6 bytes)	Header consisting of:		
		A5		Start byte
		1000 = 0x0010		Payload + CRC length (16)
		0x0000		Command flags
		0x4A		Header check sum
Command	0x22	Load module setup.		
Flags	0F 09 00 00	Value is 0x000009FF. Setup flags are:		
		Bit	(Value) Is	
		0	(1) Command includes link frequency	
		1	(2) Command includes RX decoding.	
		2	(4) Command includes TX level.	
		3	(8) Command includes TX modulation.	
		8	(0x100) Command includes antenna mask.	
		11	(0x800) Command includes selected antenna.	
Link frequency	00E80300	Value = 0x0003E800 = 256000Hz. Supported: 160000, 256000 and 320000.		
Receiver modulation	0x02	2 means Miller-4 (2^2).		
TX level	0x03	Value to be subtracted from the top level; in this case (27 − 3)dBm = 24 dBm (~250mW).		
TX modulation	0x01	1 = PR-ASK (0 = ASK).		
Enabled antennas	0x03	Antennas 0 (1) and 1 (2) are enabled: 0x01 | 0x02 = 0x03.		
Selected antenna	0xFF.	Value means automatic switching within the module. Range otherwise is 0…3.		
Packet CRC	0x45 0x0E	Little endian; actual is **0x0E45**.		

SETUP FLAG BITS AND CORRESPONDING VALUE DEFINITIONS

Bit	Mask value	Corresponding setup field and size
0	0x00000001	Link frequency in Hz, 32-bit unsigned value: 160000, 256000 or 320000.
1	0x00000002	Transmission level, one byte: maximum_level-value = TX level (in dBm). Range is from 0 (max) to 19 (min).
2	0x00000004	Receiver decoding, one byte: 0…3 = FM0, M-2, M-4, M-8 (=2^{value}).
3	0x00000008	Transmitter modulation, one byte: 0 = ASK, 1 = PR-ASK.
4	0x00000010	Region ID, one byte: 0…18, 0xFE = custom hoptable. (0=EU, 1 = FCC).
5	0x00000020	Default inventory Q, one byte: 0…15 where 0 is "automatic".
6	0x00000040	Default inventory session, one byte: 0…3 as specified by the Gen2 protocol.
7	0x00000080	Default number of inventory rounds per inventory call, one byte: 0…10 (0= automatic).
8	0x00000100	Antenna mask, one byte: (obsolete, use antenna mask extended wherever possible). Range is 0…0x0F each bit (0…3) corresponding to an antenna i.e. "antenna enable".
9	0x00000200	Single scan default timeout in milliseconds, 16-bit unsigned: 50…500.
10	0x00000400	Inventory timeout in milliseconds, 16-bit unsigned: 0…60000.
11	0x00000800	Selected antenna, one byte: 0…max_antenna-1.
12	0x00001000	Operation flags, 32-bit unsigned: bit0 = frequency hop event enabled, bit1 = inventory stream zeros enabled, bit12: autotune events enabled.
13	0x00002000	Default inventory target, one byte: 0 = A, 1 = B, 2 = auto-toggle A-B.
14	0x00004000	Default inventory EPC length in bytes, one byte: 0xFF = all, other = length in bytes, greater than 0 and aligned by 2 (2, 4, 6, …).
15	0x00008000	Read RSSI filter in dBm, 2 x signed 8-bit: e.g. -60, -50 performs read only when RSSI in range. 0 = unused.
16	0x00010000	Write RSSI filter in dBm, 2 x signed 8-bit: e.g. -60, -50 performs write only when RSSI in range. 0 = unused.
17	0x00020000	Inventory RSSI filter in dBm, 2 x signed 8-bit: e.g. -60, -50 filters tags that are out of range. 0 = unused.
18	0x00040000	Read timeout in ms, 16-it unsigned: 20…1000.
19	0x00080000	Write timeout in ms, 16-it unsigned: 20…2000.
20	0x00100000	Lock timeout in ms, 16-it unsigned: 20…2000.
21	0x00200000	Kill timeout in ms, 16-it unsigned: 20…2000.
22	0x00400000	Module's autoperiod setting, 8-bit unsigned: controls reader's "off time" when no tags in field (streaming). 0 = not in use, 1 = 1000ms, 2 = 500ms and 3 = 100ms "guard time".
23	0x00800000	Per antenna power, 4 bytes: overrides main TX level in per antenna basis, range is 0…19. if 0xFF then not used. Note: extended setting should be used.
24	0x01000000	Per antenna power offset, 4 x 8 bits signed, first one is used: added to per antenna level -1, 0, or 1 when applicable.
25	0x02000000	Extended antenna mask, 32-bit unsigned: mask can be for antennas 0…31.
26	0x04000000	Autotune setup, 8-bit unsigned + 8-bit signed: 8-bit unsigned defines mode (bit0 = on/off, bit1 = use/don't use threshold) and following 8-bit signed is threshold in dBm if used.
27	0x08000000	Per antenna power: 32 bytes in range 0…19; overrides main TX level per antenna basis.
28	0x10000000	RX sensitivity, unsigned 8-bit: 0 = "nominal", 1 = low, 2 = high.

SET BASIC SETUP RESPONSE PACKET

```
A5 11 00 00 00 4B 22 00 0F 09 00 00 00 E8 03 00 02 03 01 03 FF 2C 52
```

SET BASIC SETUP RESPONSE CONTENTS

The response consists of the command echo, status byte and echo of the setup flags and the parameters that were set.

Field	Value	Description		
Header	A5110000004B (6 bytes)	Header consisting of:		
		A5	Start byte	
		1100 = 0x0011	Payload + CRC length (17)	
		0x0000	Command flags	
		0x4B	Header check sum	
Command	0x22	Load module setup echo.		
Status	0x00	No error (0).		
Flags	0F 09 00 00	Value is 0x000009FF. Setup flags are:		
		Bit	(Value) Is	
		0	(1) Response includes link frequency	
		1	(2) Response includes RX decoding.	
		2	(4) Response includes TX level.	
		3	(8) Response includes TX modulation.	
		8	(0x100) Response includes antenna mask.	
		11	(0x800) Response includes selected antenna.	
Link frequency	00E80300	Value = 0x0003E800 = 256000Hz. Supported: 160000, 256000 and 320000.		
Receiver modulation	0x02	2 means Miller-4 (2^2).		
TX level	0x03	Value to be subtracted from the top level; in this case (27 − 3) dBm = 24 dBm (~250mW).		
TX modulation	0x01	1 = PR-ASK (0 = ASK).		
Enabled antennas	0x03	Antennas 0 (1) and 1 (2) are enabled: 0x01 | 0x02 = 0x03.		
Selected antenna	0xFF.	Value means automatic switching within the module. Range otherwise is 0…3.		
Packet CRC	0x2C 0x52	Little endian; actual is **0x522C**.		

BASIC SETUP RESPONSE'S EXAMPLE C STRUCTURE

```c
/* Like the command version but added with 'status' byte. */
struct __packed BASICSETUPRESP
{
  struct NURCMDHDR hdr;
  uint8_t setupCmd;
  uint8_t status;
  struct BASICSETUP basic;
  uint16_t crc;
};
```

GET BASIC INVENTORY SETUP

This section focuses only on the basic inventory settings: Q, session and number of inventory rounds per command.

GET BASIC INVENTORY SETUP PACKET

A5 07 00 00 00 5D 22 E0 00 00 00 D1 D9

GET BASIC INVENTORY SETUP PACKET IN C

```c
const unsigned char inventoryGetCmd[] =  {
  0xA5, 0x07, 0x00, 0x00, 0x00, 0x5D, 0x22,
  0xE0, 0x00, 0x00, 0x00, 0xD1, 0xD9
};

#define INVENTORYFLAGS \
(1 << 7 | 1 << 6 | 1 << 5)

struct __packed NURCMDHDR
{
  uint8_t start;    /* 0x0A5 */
  uint16_t payLen;  /* Following payload length including CRC-16.*/
  uint16_t flags;   /* Protocol/command flags. */
  uint8_t char cs;  /* Header checksum. */
};

struct __packed GETBASICINVENTORY
{
  struct NURCMDHDR hdr;
  unsigned char setupCmd;
  uint32_t flags;  /* Set to INVENTORYFLAGS */
  uint16_t crc;
};
```

GET BASIC INVENTORY SETUP CONTENTS

Field	Value	Description		
Header	A5070000005D (6 bytes)	Header consisting of:		
		A5	Start byte	
		0700 = 0x0007	Payload + CRC length (7)	
		0x0000	Command flags	
		0x5D	Header check sum	
Command	0x22	Load module setup.		
Flags	E0 00 00 00	Value is 0x000000E0. Setup flags are:		
		Bit	(Value) Is	
		5	(0x20) Command requests inventory Q.	
		6	(0x40) Command requests inventory session.	
		7	(0x80) Command requests inventory rounds.	
Packet CRC	0xD1 0xD9	Little-endian actual is **0xD9D1**.		

GET BASIC INVENTORY SETUP RESPONSE

```
A5 0B 00 00 00 51 22 00 E0 00 00 00 05 00 01 DB 9A
```

GET BASIC INVENTORY SETUP RESPONSE'S CONTENTS

Field	Value	Description		
Header	A50B00000051 (6 bytes)	Header consisting of:		
		A5	Start byte	
		0B00 = 0x000B	Payload + CRC length (11)	
		0x0000	Command flags	
		0x51	Header check sum	
Command	0x22	Load module setup.		
Status	0x00	No error (0).		
Flags	E0 00 00 00	Value is 0x000000E0. Setup flags are:		
		Bit	(Value) Is	
		5	(0x20) Response includes inventory Q.	
		6	(0x40) Response includes inventory session.	
		7	(0x80) Response includes inventory rounds.	
Inventory Q	0x05	Q value is 5.		
Inventory session	0x00	Current default session is 0.		
Inventory rounds	0x01	Number inventories per call is 1.		
Packet CRC	0xD1 0xD9	Little-endian actual is **0xD9D1**.		

Nordic ID Oy | Myllyojankatu 2 A | FI-24100 Salo | Finland
Office +358 2 727 7700 | Fax + 358 2 727 7720 | info@nordicid.com
www.nordicid.com | www.rfidarena.com | Facebook | Twitter | YouTube

GET BASIC INVENTORY SETUP RESPONSE'S C STRUCTURE

```c
/* Like the command but added with status & parameters. */

struct __packed GETBASICINVENTORY
{
  struct NURCMDHDR hdr;
  uint8_t setupCmd;
  uint8_t status;
  uint32_t flags;  /* Set to INVENTORYFLAGS */
  uint8_t Q;
  uint8_t session;
  uint8_t rounds;
  uint16_t crc;
};
```

SET BASIC INVENTORY SETUP

The command set the inventory parameters: Q = 6, session = 1 and rounds = 2.

THE BASIC INVENTORY SETTING PACKET

```
A5 0A 00 00 00 50 22 E0 00 00 00 06 01 02 C0 7C
```

THE BASIC SET INVENTORY SETUP PACKET IN C

```c
#define INVENTORYFLAGS \
(1 << 7 | 1 << 6 | 1 << 5)

struct __packed NURCMDHDR
{
  uint8_t start;     /* 0x0A5 */
  uint16_t payLen;   /* Following payload length including CRC-16.*/
  uint16_t flags;    /* Protocol/command flags. */
  uint8_t char cs;   /* Header checksum. */
};

struct __packed SETBASICINVENTORYCMD
{
  struct NURCMDHDR hdr;
  uint8_t setupCmd; /* 0x22 */
  uint32_t flags;
  uint8_t Q;
  uint8_t session;
  uint8_t rounds;
  uint16_t crc;  /* Calculate starting from command */
};
```

Nordic ID Oy | Myllyojankatu 2 A | FI-24100 Salo | Finland
Office +358 2 727 7700 | Fax + 358 2 727 7720 | info@nordicid.com
www.nordicid.com | www.rfidarena.com | Facebook | Twitter | YouTube

SET BASIC INVENTORY SETUP PACKET CONTENTS

Field	Value	Description	
Header	A50A00000050 (6 bytes)	Header consisting of:	
		A5	Start byte
		0A00 = 0x000A	Payload + CRC length (10)
		0x0000	Command flags
		0x50	Header check sum
Command	0x22	Load module setup.	
Flags	E0 00 00 00	Value is 0x000000E0. Setup flags are:	
		Bit	(Value) Is
		5	(0x20) Command includes inventory Q.
		6	(0x40) Command includes inventory session.
		7	(0x80) Command includes inventory rounds.
Inventory Q	0x06	Q value is 6.	
Inventory session	0x01	Use default session 1.	
Inventory rounds	0x02	Number of inventories per call is 2.	
Packet CRC	0xC0 0x7C	Little-endian actual is **0x7CC0**.	

THE RESPONSE TO THE BASIC INVENTORY SETUP CHANGE

Field	Value	Description	
Header	A50B00000051 (6 bytes)	Header consisting of:	
		A5	Start byte
		0B00 = 0x000A	Payload + CRC length (11)
		0x0000	Command flags
		0x51	Header check sum
Command	0x22	Load module setup.	
Status	0x00	No error (0)	
Flags	E0 00 00 00	Value is 0x000000E0. Setup flags are:	
		Bit	(Value) Is
		5	(0x20) Response includes inventory Q.
		6	(0x40) Response includes inventory session.
		7	(0x80) Response includes inventory rounds.
Inventory Q	0x06	Q value is 6.	
Inventory session	0x01	Use default session 1.	
Inventory rounds	0x02	Number of inventories per call is 2.	
Packet CRC	0xD9 0xC0	Little-endian actual is **0xC0D9**.	

MAKE CURRENT SETUP PERMANENT

When the current setup is made permanent, the currently saved is first compared to the one in use. So it is safe to issue this command even if there were no changes; the internal flash is not written if no changes were detected.

STORE CURRENT SETUP (ALL) PACKET

```
A5 04 00 00 00 5E 28 0F AF 63
```

STORE CURRENT SETUP (ALL) PACKET IN C

```c
#define STORE_RF     (1) /* RF, antenna, RSSI levels, tuning etc. */
#define STORE_GPIO   (2) /* Store sensor and GPIO. */
#define STORE_BDR    (4) /* Make baudrate permanent. */
#define STORE_OPF    (8) /* Operation flags. */

struct __packed NURCMDHDR
{
  uint8_t start;    /* 0x0A5 */
  uint16_t payLen;  /* Following payload length including CRC-16.*/
  uint16_t flags;   /* Protocol/command flags. */
  uint8_t char cs;  /* Header checksum. */
};

struct __packed STORESETUP
{
  struct NURCMDHDR hdr;
  uint8_t storeCmd; /* 0x28 */
  uint8_t flags;    /* Combination of STORE_<xx> */
  uint16_t crc;     /* Calculate starting from command */
};
```

STORE CURRENT SETUP PACKET CONTENTS

Field	Value	Description		
Header	`A5040000005E` `(6 bytes)`	Header consisting of:		
		`A5`	Start byte	
		`0400 = 0x0004`	Payload + CRC length (4)	
		`0x0000`	Command flags	
		`0x5E`	Header check sum	
Command	`0x28`	Store current setup		
Flags	`0x0F`	Value is `0x0F`. Setup flags are:		
		Bit	(Value) Is	
		0	(0x01) RF settings (TX level, RX decoding, etc.). Also tuning if RAM values in use differ from the non-volatile user tune area.	
		1	(0x02) GPIO/sensors.	
		2	(0x03) Baudrate.	
		3	(0x03) operation flag field in the module setup.	
Packet CRC	`0xAF 0x63`	Little-endian actual is **`0x63AF`**.		

STORE CURRENT SETUP RESPONSE

Command is always successful with correct parameters i.e. the command is echoed with the status byte being 0.

STORE CURRENT SETUP RESPONSE PACKET

`A5 04 00 00 00 5E 28 00 40 92`

STORE CURRENT SETUP RESPONSE PACKET CONTENTS

Field	Value	Description	
Header	`A5040000005E` `(6 bytes)`	Header consisting of:	
		`A5`	Start byte
		`0400 = 0x0004`	Payload + CRC length (4)
		`0x0000`	Command flags
		`0x5E`	Header check sum
Command	`0x28`	Store current setup	
Status	`0x00`	No error (0).	
Packet CRC	`0x40 0x92`	Little-endian actual is **`0x9240`**.	

RESETTING TO FACTORY DEFAULTS

FACTORY DEFAULT PACKET

A5 07 00 00 00 5D 13 32 6B 4F 99 FD 5E

FACTORY DEFAULT PACKET IN C

```c
struct __packed NURCMDHDR
{
  uint8_t start;     /* 0x0A5 */
  uint16_t payLen;   /* Following payload length including CRC-16.*/
  uint16_t flags;    /* Protocol/command flags. */
  uint8_t char cs;   /* Header checksum. */
};

struct __packed FACTORYRESET
{
  struct NURCMDHDR hdr;
  uint8_t resetCmd; /* 0x13 */
  uint32_t magic;    /* 0x994F6B32 */
  uint16_t crc;  /* Calculate starting from command */
};
```

FACTORY DEFAULT PACKET CONTENTS

Field	Value	Description		
Header	A5070000005D (6 bytes)	Header consisting of:		
		A5		Start byte
		0700 = 0x0007		Payload + CRC length (7)
		0x0000		Command flags
		0x5D		Header check sum
Command	0x13	Store current setup		
Factory default magicode	32 6B 4F 99	0x994F6B32. Using incorrect code causes error.		
Packet CRC	0xFD 0x5E	Little-endian, actual is **0x5EFD**.		

Nordic ID Oy | Myllyojankatu 2 A | FI-24100 Salo | Finland
Office +358 2 727 7700 | Fax + 358 2 727 7720 | info@nordicid.com
www.nordicid.com | www.rfidarena.com | Facebook | Twitter | YouTube

FACTORY DEFAULT RESET RESPONSE

FACTORY DEFAULT RESET RESPONSE PACKET

```
04 00 00 00 5E 13 00 2F 4B
```

FACTORY DEFAULT RESET RESPONSE CONTENTS

Field	Value	Description		
Header	A5040000005E (6 bytes)	Header consisting of:		
		A5		Start byte
		0400 = 0x0004		Payload + CRC length (7)
		0x0000		Command flags
		0x5E		Header check sum
Command	0x13	Store current setup		
Status	0x00	No error (0).		
Packet CRC	0x2F 0x4B	Little-endian, actual is **0x4B2F**.		