

NUR2-0W1 MIGRATION DOCUMENT

VERSION HISTORY

VERSION	DATE	AUTHOR	CHANGES
0.1	2019-Dec-02	JRaut	Initial draft

TABLE OF CONTENTS

VERSION HISTORY	2
TABLE OF CONTENTS	3
1. SCOPE 4	
2. NEW IN NUR2-0W1	5
2.1. MODULE TYPE DETECTION	5
2.2. RF PROFILE	6
2.2.1. SETTING RF PROFILE	6
2.2.2. BACKWARDS COMPATIBILITY	6
2.2.3. DETECT RF PROFILE SUPPORT	6
2.3. DIAGNOSTICS	7
2.3.1. DIAGNOSTICS DATA	7
2.3.2. GETTING DATA	7
2.4. LOW POWER SUPPORT	8
3. NOT SUPPORTED IN NUR2-0W1	9
3.1. MODULE SETUP FIELDS	9
3.1.1. LINKFREQ BACKWARDS COMPATIBILITY	9
3.2. XTID INVENTORY	10
3.3. CUSTOM EXCHANGE – SUPPORTED, LEGACY FUNCTIONS	11
3.4. GEN2V2 FUNCTIONALITY	12
3.5. LISTEN BEFORE TALK	13
3.6. TAG PHASE INFO	14

1. SCOPE

This document provides information how to migrate application from NUR based devices to NUR2-0W1 devices.

If you have any questions regarding to migration or NUR2-0W1 features, please contact support@nordicid.com

This document applies to NUR2-0W1 firmware version 10.2-A

2. SUPPORTED IN NUR2-0W1

2.1. MODULE TYPE DETECTION

NUR2-0W1 module can be detected using NURAPI device capabilities.

Used value in `NUR_DEVICECAPS.chipVersion`:

```
/** Chip version AS3993 */  
NUR_CHIPVER_AS3993 = 2
```

New value in `NUR_DEVICECAPS.moduleType`:

```
/** Module type NUR2-0W1 */  
NUR_MODULETYPE_NUR2_0W1 = 6
```

2.2. RF PROFILE

NUR2-OW1 RFID low level settings, such as link frequency, miller, modulation are controlled via new rfProfile member in NURAPI module setup.

2.2.1. SETTING RF PROFILE

NUR_MODULESETUP.rfProfile accepted values:

```
/** Robust RF profile. This profile is recommended to use in noisy RF environments. */
#define NUR_RFPROFILE_ROBUST 0
/** Nominal RF profile. This profile works good in most environments. */
#define NUR_RFPROFILE_NOMINAL 1
/** High speed RF profile. This profile provides best throughput, but is prone to RF
interference. */
#define NUR_RFPROFILE_HIGHSPEED 2
```

For example:

```
moduleSetup.rfProfile = NUR_RFPROFILE_HIGHSPEED;

NurApiSetModuleSetup(hApi, NUR_SETUP_RFPROFILE, &moduleSetup, sizeof(moduleSetup));
```

2.2.2. BACKWARDS COMPATIBILITY

For backwards compatibility with old code, it is still allowed to set any existing module setup members, such as link frequency, miller, modulation.

However, they do not change RFID behavior in any way.

You can still use old API to change rfProfile via linkFreq backward compatibility mode,

see section 3.1.1. LINKFREQ BACKWARDS COMPATIBILITY.

2.2.3. DETECT RF PROFILE SUPPORT

RF Profile support can be detected using NURAPI device capabilities.

New value in NUR_DEVICECAPS.flagSet1:

```
/* The module FW supports RF profile setting. */
NUR_DC_RFPROFILE = (1<<24)
```

2.3. DIAGNOSTICS

NUR2-0W1 can report health diagnostics information to host. This is useful especially for debugging purposes.

2.3.1. DIAGNOSTICS DATA

NUR2 family diagnostics data structure.

```
struct NUR_DIAG_REPORT
{
    DWORD flags;           /**< Report flags. see enum NUR_DIAG_REPORT_FLAGS */
    DWORD uptime;          /**< Uptime in milliseconds */
    DWORD rfActiveTime;    /**< RF on time in milliseconds */
    int temperature;       /**< Temperature in celcius. 1000 if not supported */
    DWORD bytesIn;         /**< Number of bytes in to module */
    DWORD bytesOut;        /**< Number of bytes out from module */
    DWORD bytesIgnored;    /**< Number of ignored (invalid) bytes */
    DWORD antennaErrors;   /**< Number of bad antenna errors */
    DWORD hwErrors;        /**< Number of automatically recovered internal HW failures */
    DWORD invTags;         /**< Number of successfully inventoried tags */
    DWORD invColl;         /**< Number of collisions during inventory */
    DWORD readTags;        /**< Number of successfully read tag commands */
    DWORD readErrors;      /**< Number of failed read tag commands */
    DWORD writeTags;       /**< Number of successfully write tag commands */
    DWORD writeErrors;     /**< Number of failed write tag commands */
    DWORD errorConds;      /**< Number of temporary error conditions (over temp, low voltage) occurred */
    DWORD setupErrs;       /**< Number of invalid setup errors */
    DWORD invalidCmds;     /**< Number of invalid (not supported) commands received */
};
```

2.3.2. GETTING DATA

NUR2 diagnostics data can be fetched from module with NurApiDiagGetReport() function.

For example:

```
struct NUR_DIAG_REPORT report;
NurApiDiagGetReport(hApi, 0, &report, sizeof(report));
```

See also in documentation:

```
int NurApiDiagSetConfig(HANDLE hApi, DWORD flags, DWORD interval);
int NurApiDiagGetConfig(HANDLE hApi, DWORD *flags, DWORD *interval);
int NurApiDiagGetReport(HANDLE hApi, DWORD flags, struct NUR_DIAG_REPORT
*report, DWORD reportSize);
```

2.4. LOW POWER SUPPORT

NUR2-0W1 aims to minimize power consumption and thus maximize battery life.

For this reason, the maximum transmit power is limited to 100mW.

Also, the module will enter low power state after a period of inactivity. The default timeout to enter low power state is five seconds.

3. NOT SUPPORTED IN NUR2-0W1

Some of the legacy functionality is not supported in NUR2-01W based modules.

3.1. MODULE SETUP FIELDS

Following fields in module setup (`struct NUR_MODULESETUP`) are not supported, setting them does not affect in any way

- rxDecoding
 - Automatically controlled by rfProfile
- txModulation
 - Automatically controlled by rfProfile

3.1.1. LINKFREQ BACKWARDS COMPATIBILITY

NUR2-0W1 module has backwards compatibility enabled with linkFreq field. You can use linkFreq field to change rfProfile.

- linkFreq 160000 = rfProfile 0 (robust)
- linkFreq 256000 = rfProfile 1 (nominal)
- linkFreq 320000 = rfProfile 2 (high speed)

3.2. XTID INVENTORY

Reading XTID memory during inventory automatically is not supported in NUR2-0W1.

Following functions are not supported

- `int NurApiConfigXTIDInventory(HANDLE hApi, BOOL dataOnly, BOOL includeHeader);`

3.3. CUSTOM EXCHANGE – SUPPORTED, LEGACY FUNCTIONS DROPPED

NUR2-OW1 does support 'Custom Exchange' command.

Following legacy functions are not supported. Note that this functionality can be done using 'Custom Exchange'.

- `int NurApiCustomReadSingulatedTag32(HANDLE hApi, DWORD rdCmd, BYTE cmdBits, DWORD rdBank, BYTE bankBits, DWORD passwd, BOOL secured, BYTE sBank, DWORD sAddress, int sMaskBitLength, BYTE *sMask, DWORD rdAddress, int rdByteCount, BYTE *rdBuffer);`
- `int NurApiCustomReadTagByEPC(HANDLE hApi, DWORD rdCmd, BYTE cmdBits, DWORD rdBank, BYTE bankBits, DWORD passwd, BOOL secured, BYTE *epcBuffer, DWORD epcBufferLen, DWORD rdAddress, int rdByteCount, BYTE *rdBuffer);`
- `int NurApiCustomReadTag32(HANDLE hApi, DWORD rdCmd, BYTE cmdBits, DWORD rdBank, BYTE bankBits, DWORD passwd, BOOL secured, DWORD rdAddress, int rdByteCount, BYTE *rdBuffer);`
- `int NurApiDisableCustomReselect(HANDLE hApi);`
- `int NurApiCustomWriteSingulatedTag32(HANDLE hApi, DWORD wrCmd, BYTE cmdBits, DWORD wrBank, BYTE bankBits, DWORD passwd, BOOL secured, BYTE sBank, DWORD sAddress, int sMaskBitLength, BYTE *sMask, DWORD wrAddress, int wrByteCount, BYTE *wrBuffer);`
- `int NurApiCustomWriteTagByEPC(HANDLE hApi, DWORD wrCmd, BYTE cmdBits, DWORD wrBank, BYTE bankBits, DWORD passwd, BOOL secured, BYTE *epcBuffer, DWORD epcBufferLen, DWORD wrAddress, int wrByteCount, BYTE *wrBuffer);`

3.4. GEN2V2 FUNCTIONALITY

NUR2-OW1 does not support Gen2V2 commands. These features are not implemented yet.

Following functions are not supported

- `int NurApiGen2v2Authenticate (HANDLE hApi, BOOL secured, DWORD passwd, struct NUR_AUTHENTICATE_PARAM *pAuth, DWORD szParam, struct NUR_AUTHENTICATE_RESP *pResp);`
- `int NurApiGen2v2Authenticate32 (HANDLE hApi, BOOL secured, DWORD passwd, BYTE sBank, DWORD sAddress, int sMaskBitLength, BYTE *sMask, struct NUR_AUTHENTICATE_PARAM *pAuth, DWORD szParam, struct NUR_AUTHENTICATE_RESP *pResp);`
- `int NurApiGen2v2AuthenticateByEPC (HANDLE hApi, BOOL secured, DWORD passwd, BYTE *epcBuffer, DWORD epcBufferLen, struct NUR_AUTHENTICATE_PARAM *pAuth, DWORD szParam, struct NUR_AUTHENTICATE_RESP *pResp);`
- `int NurApiGen2v2Untraceable (HANDLE hApi, DWORD passwd, struct NUR_UNTRACEABLE_PARAM *pUtrace, DWORD szParam);`
- `int NurApiGen2v2Untraceable32 (HANDLE hApi, DWORD passwd, BYTE sBank, DWORD sAddress, int sMaskBitLength, BYTE *sMask, struct NUR_UNTRACEABLE_PARAM *pUtrace, DWORD szParam);`
- `int NurApiGen2v2UntraceableByEPC (HANDLE hApi, DWORD passwd, BYTE *epcBuffer, DWORD epcBufferLen, struct NUR_UNTRACEABLE_PARAM *pUtrace, DWORD szParam);`
- `int NurApiReadPermalock (HANDLE hApi, DWORD password, BOOL secured, struct NUR_PERMALOCK_RD_PARAM *pLock, struct NUR_PERMALOCK_RD_RESP *pResp);`
- `int NurApiReadPermalockSingulated32 (HANDLE hApi, DWORD password, BOOL secured, BYTE sBank, DWORD sAddress, int sMaskBitLength, BYTE *sMask, struct NUR_PERMALOCK_RD_PARAM *pLock, struct NUR_PERMALOCK_RD_RESP *pResp);`
- `int NurApiReadPermalockByEPC (HANDLE hApi, DWORD password, BOOL secured, BYTE *epc, DWORD epcLen, struct NUR_PERMALOCK_RD_PARAM *pLock, struct NUR_PERMALOCK_RD_RESP *pResp);`
- `int NurApiPermalock (HANDLE hApi, DWORD password, BOOL secured, struct NUR_PERMALOCK_PARAM *pLock);`
- `int NurApiPermalockSingulated32 (HANDLE hApi, DWORD password, BOOL secured, BYTE sBank, DWORD sAddress, int sMaskBitLength, BYTE *sMask, struct NUR_PERMALOCK_PARAM *pLock);`
- `int NurApiPermalockByEPC (HANDLE hApi, DWORD password, BOOL secured, BYTE *epc, DWORD epcLen, struct NUR_PERMALOCK_PARAM *pLock);`
- `int NurApiGen2v2ReadBuffer (HANDLE hApi, BOOL secured, DWORD passwd, WORD bitAddress, WORD bitCount, BYTE *buffer, int *actualBits);`
- `int NurApiGen2v2ReadBuffer32 (HANDLE hApi, BOOL secured, DWORD passwd, BYTE sBank, DWORD sAddress, DWORD sMaskBitLength, BYTE *sMask, WORD bitAddress, WORD bitCount, BYTE *buffer, int *actualBits);`
- `int NurApiGen2v2ReadBufferByEPC (HANDLE hApi, BOOL secured, DWORD passwd, BYTE *epcBuffer, DWORD epcBufferLen, WORD bitAddress, WORD bitCount, BYTE *buffer, int *actualBits);`

3.5. LISTEN BEFORE TALK

NUR2-0W1 does not support LBT, listen before talk. This feature is not implemented yet.

3.6. TAG PHASE INFO

NUR2-0W1 does not support tag phase angle. Due to technical limitations tag phase info cannot be supported.