875-0073-03 RevF

ThingMagic Nano Firmware v1.9.0 Release Notes

This release note describes the features and characteristics of ThingMagic Nano module firmware (FW) version 1.9.0. This firmware is not supported on any other ThingMagic module (each model line has its own version). This firmware will be fully supported in API version 1.31.1.

The release notes for firmware version 1.7.3, 1.7.2, 1.7.1 and 1.5.0 are appended to these release notes for reference.

Topics covered in this release notes are:

For full information about the ThingMagic Nano product, please consult the Nano Design Guide, which can be found at:

https://www.jadaktech.com/documentation/rfid/Nano-module/

JADAK
A Novanta Company

Table of Contents

New Features

New features of version 1.9.0 include the following:

Option to send Select with every Query

By default, we send a Select command to a population of tags when reading starts and each time an antenna is switched. This is to minimize the overhead of sending this relatively long message. Once a tag is selected for an inventory round, it is designed to remember this until it is successfully inventoried during that or any successive inventory round. This works under normal circumstances, but not when tags frequently enter and leave the read field. A tag that is not present to hear the Select command will respond during the first opportunity, not knowing that it is not “qualified” to do so based on the Select criteria.

Also, tags at the very limit of their receive sensitivity may not hear the Select, but correctly interpret a Query of a subsequent inventory round and respond inappropriately.

To accommodate applications requiring more frequent Selects, we have added a setting to send the Select command with every Query (which begins an inventory round).

In API, the parameter to enable sending Select with every Query (start of inventory round) is: **TMR_PARAM_GEN2_SEND_SELECT (or) "/reader/gen2/sendSelect"**

By default, the value of this flag is 0x00 which means Select will not be sent with every Query. A value of 0x01 will enable Select-with-every-Query.

Note that sending Select with every Query is the default behavior when Fast Search mode is enabled, but this mode also makes other changes to the search algorithm, which may not be desirable for all applications.

Support for Complex Selects

Traditionally, Select has been used for one purpose: to select among a population of tags (or select all but a population of tags) based on a common value in memory. As tags gain more custom features, tag IC designers have found it convenient to use unusual, but legal forms of Select to alert all their tags to perform a proprietary function. To accommodate these features, we now support the full range of Select options that are mandated by the RAIN Gen2 specification.

JADAK
A Novanta Company

Table 6.29: *Select* command

	Command	Target	Action	MemBank	Pointer	Length	Mask	Truncate	CRC
# of bits	4	3	3	2	EBV	8	Variable	1	16
description	1010	000: **Inventoried** (S0) 001: **Inventoried** (S1) 010: **Inventoried** (S2) 011: **Inventoried** (S3) 100: **SL** 101: RFU 110: RFU 111: RFU	See Table 6.30	00: FileType 01: EPC 10: TID 11: File_0	Starting Mask address	Mask length (bits)	Mask value	0: Disable truncation 1: Enable truncation	CRC-16

Table 6.30: Tag response to Action parameter

Action	Tag Matching	Tag Not-Matching
000	assert **SL** or **inventoried** → *A*	deassert **SL** or **inventoried** → *B*
001	assert **SL** or **inventoried** → *A*	do nothing
010	do nothing	deassert **SL** or **inventoried** → *B*
011	negate **SL** or (*A* → *B*, *B* → *A*)	do nothing
100	deassert **SL** or **inventoried** → *B*	assert **SL** or **inventoried** → *A*
101	deassert **SL** or **inventoried** → *B*	do nothing
110	do nothing	assert **SL** or **inventoried** → *A*
111	do nothing	negate **SL** or (*A* → *B*, *B* → *A*)

The two fields that represent this additional functionality are Target (which flag to change) and Action (whether to set, reset, flip, or leave alone the flag). The choices combine the action for when the tag memory value matches the criteria and for when it doesn't.

Here is the list of Action values with respect to matching and non-matching tags: -

Action	Matching Tags	Non-Matching Tags	API Enumeration	Behavior if Flag is "SL" (Selected)
0	Assert Target	De-Assert Target	ON_N_OFF	Matching tags will respond, and Non-Matching tags will NOT respond
1	Assert Target	Do Nothing	ON_N_NOP	Matching tags will respond, and Non-Matching tags will respond based on previous SL flag status from last Action
2	Do Nothing	De-Assert Target	NOP_N_OFF	Matching tags will respond based on previous SL flag status from last Action and Non-Matching tags will NOT respond
3	Negate Target	Do Nothing	NEG_N_NOP	Previous SL flag will be nullified for Matching tags and Non-Matching tags will respond based on previous SL flag status from last Action

4	De-Assert Target	Assert Target	OFF_N_ON	Matching tags will NOT respond, and Non-Matching tags will respond
5	De-Assert SL	Do Nothing	OFF_N_NOP	Matching tags will NOT respond, and Non-Matching tags will respond based on previous SL flag status from last Action
6	Do Nothing	Assert Target	NOP_N_ON	Matching tags will respond based on previous SL flag status from last Action and Non-Matching tags will respond
7	Do Nothing	Negate Target	NOP_N_NEG	Matching tags will respond based on previous SL flag status from last Action and previous SL flag will be nullified for Non-Matching tags

In the API, the Gen2.Select command syntax has not changed. For example, in Java:
Gen2.Select(Boolean invert, Gen2.Bank bank, int bitPointer, int bitLength, byte[] mask)

What has changed in the API is the optional inclusion of target and action settings:

- Gen2.Select.Target
- Gen2.Select.Action

Actions are specified using the shorthand enumerations in the table above. (For example, "NOP_N_ON")
Target flags have the following enumerations:

- Inventoried_S0
- Inventoried_S1
- Inventoried_S2
- Inventoried_S3
- Select (SL Flag – the default)

For backward compatibility, we need to support a parameter we introduced to the Select definition, called “Invert”, which allowed a Select criteria to define which tags should not respond rather than those that should. Each one of the Actions has an opposite action as shown in this table:

Action set by user	**Action set in FW if Invert flag is enabled**
0	4
1	6
2	5
3	7
4	0
5	2
6	1

7	3

If the optional Target and Action parameters are not set, the behavior is as before:

- When "Invert" flag is not set, default value of Target = 4 and Action = 0.
- When "Invert" flag is set, default value of Target = 4 and Action = 4.

Support for Multiple Selects

RAIN tags support the ability to process multiple Select commands to combine individual populations of tags (based on common data fields) into a single group for purposes of responding to subsequent inventory operations. Until recently, there was little need for this feature, but emerging sensor tags now use an initial Select to enable sensor reading as well as an additional Select (or more) to restrict which sensor tags respond..

Sensor tag chips may have a temperature sensor, humidity sensor, gas sensor, resistance sensor, capacitance sensor, etc. They are differentiated by address pointers. For these tags, the first Select is used to identify and “wake up” the sensor. If there is only one tag in the field for the reader to poll, one Select is all that is needed. However, if it is necessary to select a tag or tags from a group of tags, as well as performing a sendor read, an additional Select (or Selects) are needed.

The maximum number of selects that will be allowed are 3. A typical use for 3 Selects would be to simultaneously:

1. Determine via the TID field that the tag is of the desired sensor tag make and model
2. Enable tag sensing
3. Isolate only a portion of the sensor tags that are in the read field at that time

A new option has been added in module FW to enable multiple selects.
GEN2_SELECT_OPTION_MULTIPLE_SELECTS = (1 << 10)

Multiple selects are expressed as a MultiFilter (a new subclass of TagFilter).These can be used in all the same places as an existing TagFilter, so it is a simple addition to the user-visible API.

Java API:

```
// create and initialize tid filter
    Gen2.Select tidFilter = new Gen2.Select(false, Gen2.Bank.TID, 32, 16, new byte[]
{(byte)0x01,(byte)0x2F});
    tidFilter.target = Gen2.Select.Target.Select;
    tidFilter.action = Gen2.Select.Action.ON_N_OFF;

// create and initialize epc filter
   Gen2.Select epcFilter = new Gen2.Select(false, Gen2.Bank.EPC, 32, 16, new byte[]
{(byte)0xAA,(byte)0xAA});
   epcFilter.target = Gen2.Select.Target.Select;
```

Commented [PC1]: We did not mention C# API and CAPI part here.

```
    epcFilter.action = Gen2.Select.Action.ON_N_OFF;

// Initialize multiFilter by passing tidFilter and epcFilter as elements to TagFilter array
    MultiFilter multiFilter = new MultiFilter(new TagFilter[]{tidFilter, epcFilter});

// Pass this multiFilter in simple readplan
  SimpleReadPlan plan = new SimpleReadPlan(antennaList, TagProtocol.GEN2, multiFilter, null, 1000);
```

Generate GEN2V2 iChallenge in FW

In previous firmware releases, it was the host's responsibility to input an iChallenge value to perform GEN2V2 tag authentication operations. The tag encodes the iChallenge using its key and sends it back to the reader. If the reader can decode it, and the result is the value originally sent, then the reader and tag know the same secret key and the tag will be considered "authenticated".

In this firmware release, iChallenge will be generated in module FW randomly without having the need for the host to provide it as an input. The change to the API is that iChallenge is no longer needed for both TAM1 and TAM2 authentication methods.

C API Changes:

TAM1 Method:

```
TMR_Status
TMR_TagOp_init_GEN2_NXP_AES_Tam1authentication(TMR_TagOp_GEN2_NXP_Tam1Authenticat
ion *auth, TMR_NXP_KeyId keyid, TMR_uint8List *key, bool sendRawData )
```

TAM2 Method:

```
TMR_Status
TMR_TagOp_init_GEN2_NXP_AES_Tam2authentication(TMR_TagOp_GEN2_NXP_Tam2Authenticat
ion *auth, TMR_NXP_KeyId keyid, TMR_uint8List *key, TMR_NXP_Profile profile, uint16_t
Offset,uint8_t blockCount, int protMode, bool sendRawData)
```

Java API Changes:

TAM1 Method:

```
public Tam1Authentication(AES.KeyId keyId ,byte[] key, boolean enableRawData)
```

TAM2 Method:

```
public Tam2Authentication(AES.KeyId keyId ,byte[] key, AES.Profile profile,
                int offset, int blockCount, int protMode, boolean enableRawData)
```

C# API Changes:

TAM1 Method:

```
public Tam1Authentication(KeyId keyid, ushort[] key, bool sendRawData)
```

TAM2 Method:

```
public Tam2Authentication(KeyId keyid, ushort[] key, Profile profile,
                ushort offset, ushort blockCount, ushort protMode, bool sendRawData)
```

: base(keyid, key, sendRawData)

Japan Region Changes

The following changes have been made to the Japanese regional settings:

- A new JP3 region support has been added in Module FW with 6 channels and power limited to +24 dBm.
- The number of channels in the JP2 region channels has been reduced from 19 to 13. (The highest 6 channels were removed due to their recent allocation to a different class of RFID service, according to Japanese regulations.)
- The Max power for JP region is now limited to +30 dBm (instead of +31.5 dBm) per Japanese regulations.

The comparison table summarizes all details: -

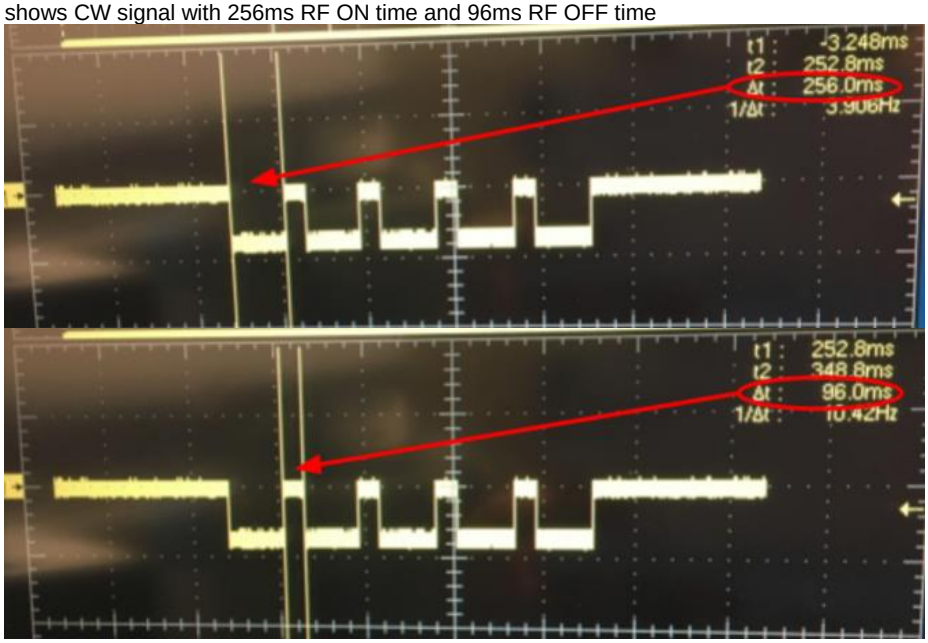
Region Name	**JP**	**JP2**	**JP3**
Country or Region Covered	Japan	Japan	Japan
Modules Supported	Micro Nano	Micro Nano	Micro Nano
Region Code	0x05	0x17	0x18
Lowest Freq Permitted in Band	916800	916800	916800
Highest Freq Permitted in Band	920800(Micro) 923400(Nano)	923400	923400
Smallest Step Size between Channels	100	100	100
Maximum Dwell Time On any Channel	4 sec	4 sec	4 sec
Max RF Power	+30 dBm	+24 dBm	+24 dBm
Carrier Sense Level	-74 dBm	-74 dBm	-74 dBm
Lowest Chan in Hop Table	916800	916800	916800
Highest Chan in Hop Table	920800	922200	920800
Hop Table			
Chan 1	920800	922200	920800
Chan 2	919200	921400	919200
Chan 3	920600	922000	920600
Chan 4	916800	921200	916800
Chan 5	918000	921800	918000
Chan 6	920400	921000	920400
Chan 7		921600	

Chan 8		920800	
Chan 9		920600	
Chan 10		920400	
Chan 11		916800	
Chan 12		919200	
Chan 13		918000	

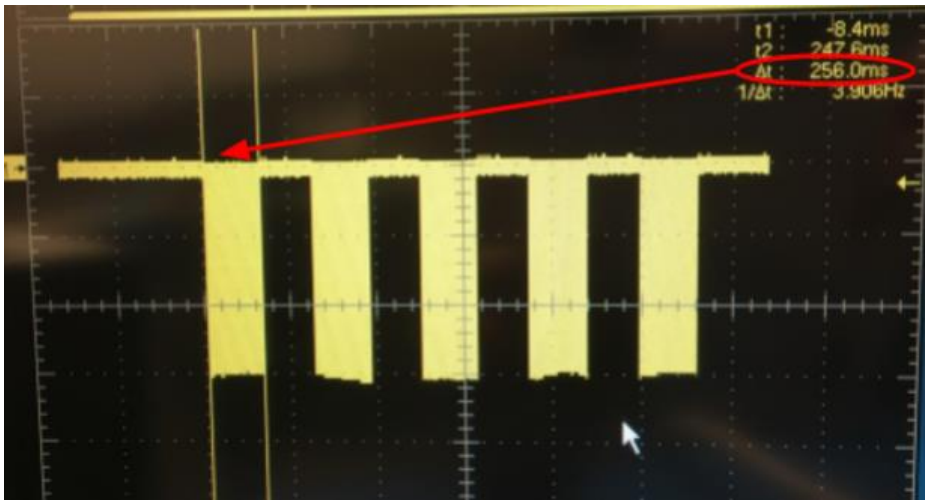
Support for Configurable CW ON/OFF Time (Regulatory Test Features)

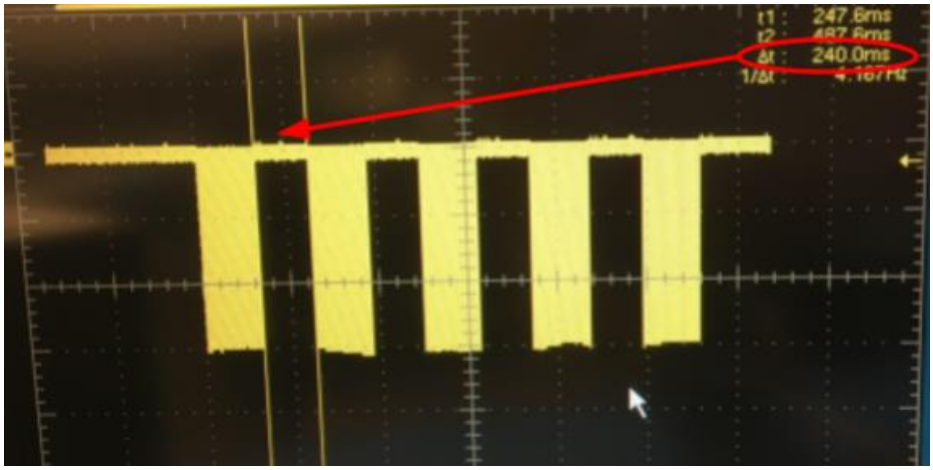
Previously, at the module FW level, a CW (continuous wave) signal could be turned on and off, with no time limit. A PRBS (randomly modulated) signal could be turned on, but only for a configured time limit. (Universal Reader Assistant artificially imposed a time limit setting on CW, which is not in the firmware or API.)

Now support has been added for much more flexibility for both CW and PRBS signal generation. Either can be enabled or disabled for an arbitrary amount of time and, while enabled, they can operate on an on-off duty cycle. This scope capture of signal amplitude shows CW signal with 256ms RF ON time and 96ms RF OFF time



This scope capture of signal amplitude show PRBS signal with 256ms RF ON time and 240ms RF OFF time





The module needs 2ms RF settling time after ramp up. So, if RF ON time is specified as 2ms or less, RF will not be turned on. 3ms is the least time where RF ON pulse can be seen. For RF ON time>2ms, a compensation of 2ms for settling time is included in FW for accurate timing.

RF off time is also compensated 1ms for power supply settling. So, RF is turned off only if RF off time > 1ms.

If RF on time is > 2ms and off time is 1ms or less, then RF will remain on continuously for the specified number of on/off cycles times on time.

Valid min and max limits of RF ON and OFF time are 2 to 65535 (0xFFFF) milliseconds. (0 and 1 are permitted, but not recommended.)

Valid min and max limits of number of ON/OFF cycles are 1 to 65535 (0xFFFF) milliseconds. (0 is permitted, but not recommended.)

The FW will monitor module temperature when CW/PRBS is running continuously to avoid the module heating up. If the limit of +85 C internal temperature is exceeded, a 0x504 error code will be returned in response to the CmdGetTemperature Method. (Periodic temperature status messages are not send when the module is in this mode, so it must be polled.)

The RF channel used will be value set with the CmdTestSetFrequency method. By default, this frequency will be the lowest one in the hop table so the hop table can be re-defined as an alternative to setting this frequency explicitly.

To add support for CW/PRBS feature, five new parameters has been added in API. These parameters are set to turn on the CW/PRBS in CONTINUOUS or TIMED mode.

a) TMR_PARAM_REGULATORY_MODE (Continuous or Timed)
b) TMR_PARAM_REGULATORY_MODULATION (CW or PRBS)
c) TMR_PARAM_REGULATORY_ONTIME (milliseconds)
d) TMR_PARAM_REGULATORY_OFFTIME (milliseconds)
e) TMR_PARAM_REGULATORY_ENABLE

Support for Per-Antenna On-Time Settings

With this FW version, it is possible to schedule per antenna ON times in continuous read mode. Previously, this was only possible through API control, resulting in significant delay between one antenna turning off and the next turning on. Now the API can set “RF ON time” directly for each antenna at the module level, delegating all timing and synchronization issues to the module. As before, the antenna switching order may be defined, along with the new read time setting.

The default behavior is to have no per-antenna on-time settings; i.e., use the dynamic search timeout behavior (switch when 5 inventory rounds result in no tags found). If any per-antenna on-times are set, then these override the default behavior.

There are no changes to the user-visible interface as the API already provided this feature through a weighted MultiReadPlan, but now the information is transferred to the module for it to execute rather than being executed in the API itself. Weighted per-antenna reading may now

executed as an asynchronous (continuous) read rather than a series of timed, synchronous, reads.

Support for Initial Q Setting

One of the assumptions that the writers of the RAIN gen2 protocol made is that if multiple tags responded in the same inventory response slot, each one was extremely unlikely to pick the same random number between 0 and 65535, so that when the reader acknowledges one tag's response with that number, the "losing" tags would realize their failure and try again during the next inventory round.

We have discovered that, in practice, this algorithm is not fool-proof. By default, our module always offers the same number of inventory slots as the previous inventory round, and adjusts them in subsequent inventory rounds as more or fewer tags respond. However, when conditions cause hundreds of tags to suddenly respond, many tags respond in the same few slots and the probability that no two tags will pick the same random number is surprisingly low. (There is a 75% chance that two of 430 tags will pick the same random number between 0 and 65535 if only one slot is available). After a long period of no tags responding, we offer as few as 4 slots, which reduces this probability by 25% (to 19%) of two tags selecting the same random number. This means that after many read cycles, it is highly likely that a tag will erroneously believe that it responded correctly to an inventory round (although it was another tag whose information was gathered) and then not respond again until its session timer expires (which can be several minutes under Session 2 or Session 3 rules). For some applications, this is has a significant negative impact.

We have made the following changes to reduce, but not entirely eliminate, this issue.

1. Extended the range of dynamic "Q" values from 2 (4 slots) to 10 (1024 slots) under normal operation.
2. Added an option to set "Initial Q" whenever we believe that it is likely a large number of tags will respond.

"Initial Q" would be imposed under the following circumstances:

1. Use whenever there has been a reboot; so, no previous "Q" is available (default is "3" if no "Initial Q" is set).
2. When in the middle of a read cycle, would be used instead of the previous "Q" whenever the target has changed from B to A or A to B.
3. Use at the start of a continuous read cycle or timed read.

"Initial Q" would not be imposed in the following cases:

1. At each asynchronous cycle when continuously reading.
2. Whenever there is a change of antenna. (It is likely all antennas are pointed at the same tag population, so a sudden increase in tag responses is not expected.)

Min and Max values of "Initial Q" are 2 to 10.

The new parameter in the API to set the “Initial Q” functionality is:

TMR_PARAM_GEN2_INITIAL_Q (or) "/reader/gen2/initQ"

Support for 5 new regions

Support for Vietnam, Thailand, Argentina, Hong Kong and Bangladesh regions has been added on Nano.

Here is a table summarizing details of 5 new regions: -

Country	Vietnam (VN)	Thailand (TH)	Argentina (AR)	Hong Kong (HK)	Bangladesh (BD)
Region code	0x19	0x1A	0x1B	0x1C	0x1D
Lowest freq permitted	866000	920000	915000	865000	925000
Highest freq permitted	869000	925000	928000	868000	927000
Smallest step size	50	250	250	100	100
Maximum dwell time	0.4 sec	0.4 sec	0.4 sec	0.4 sec	0.4 sec
Max RF power	27 dBm	27 dBm	27 dBm	27 dBm	27 dBm
LBT level	None	None	None	None	None
Lowest channel in hop table	866450	920750	915250	865100	925400
Highest channel in hop table	867250	924250	927250	867900	926600
Hop table	866550, 866850, 867150, 866450, 866750, 867050, 866650, 866950,	921250, 923250, 922250, 924250, 921750, 923750, 920750, 922750	921250, 923750, 919250, 916750, 926250, 922750, 918250,	866500, 867100, 865900, 866300, 867700, 865500, 866100,	926000, 926200, 925800, 926600, 925400, 926400, 925600

	867250		915750, 925250, 924750, 920250, 917750, 927250, 921750, 923250, 919750, 916250, 926750, 922250, 918750, 915250, 925750, 924250, 920750, 917250	867900, 865300, 866700, 867300, 865100, 866900, 867500, 865700	

5 new enum numbers have been added to “TMR_Region” enum to support newly added regions through API.

/ Vietnam **/ TMR_REGION_VN = 25,**
/ Thailand **/ TMR_REGION_TH = 26,**
/ Argentina **/ TMR_REGION_AR = 27,**
/ Hong Kong **/ TMR_REGION_HK = 28,**
/ Bangladesh **/ TMR_REGION_BD = 29,**

Resolved Issues

The following Nano bugs fixed and verified during this release version 1.9.0:

- Reading tags on Nano with JP region, being in sleep power mode now works fine. With the previous release FW version, module is returning assert error. (Ref #5846).

Operational Notes

The following restrictions and caveats apply to the features and functionality of firmware version 1.9.0

- CW/PRBS pulse is visible only when RF ON time is >2msec as module needs 2ms RF settling time after ramp up. 3ms is the least time where RF ON pulse can be seen. (Ref #5572).

- User may observe difference in tag read timestamp with system time when module temperature exceeds 85c. Usually it goes behind the system time. (Ref #5958).

- User may observe difference in tag read timestamp with system time during long run. Usually it goes ahead of system time. (Ref #5925).

ThingMagic Nano Firmware v1.7.3 Release Notes

These Firmware v1.7.3 release notes mainly describe the feature modifications and bug fixes from previous release v1.7.2. This firmware is not supported on any other ThingMagic module. Release notes for versions 1.7.2, 1.7.1 and 1.5.0 are included for reference.

Nano firmware version 1.7.3 has been developed in conjunction with version 1.29.4 of the MercuryAPI and should be used with that version (or higher) to achieve best results. Previous versions of the API will not support all the features of this firmware release. See the API release notes and MercuryAPI Programmer's Guide for further information on its features and functions.

Topics covered in these release notes are:

- Resolved Issues

For full information about the ThingMagic Nano product, please consult the Nano Design Guide, which can be here:

http://www.thingmagic.com/manuals-firmware

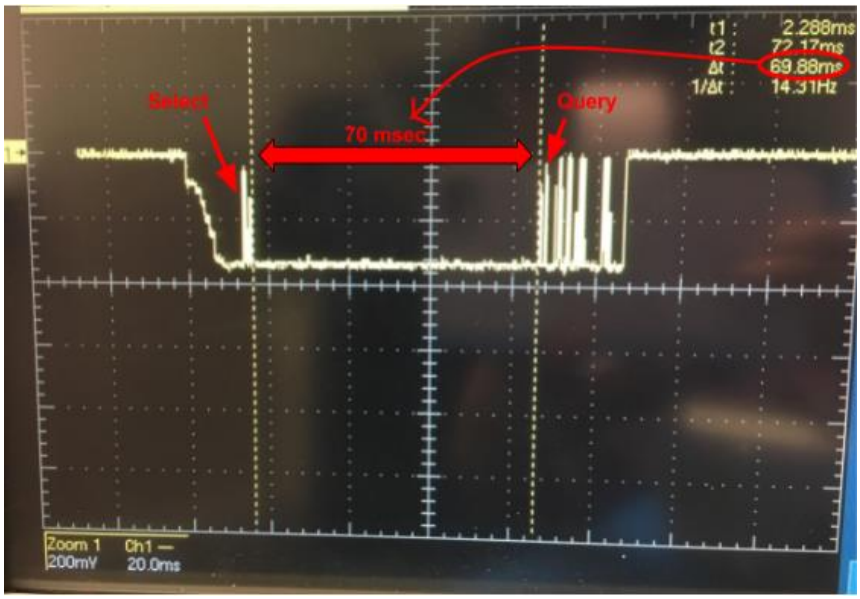
Resolved Issues

Configurable T4 for Gen2 protocol

Nano-1.7.2 FW build was released and then a bug in T4 timing was discovered causing a dropped read rate (bug-5551) and it is fixed. Minimum value of T4 was 64us earlier. Now it has been adjusted to accommodate R2000/R500 setup time; thus, more accuracy in timing.

Here is the updated description:

T4 value is a 4-byte value specified in microseconds. Minimum value of T4 allowed for 250 kHz (25us, 12.5us, 62.5us Tari) is 440usec (0x1B8) and for 640 kHz (6.25us Tari) is 220usec (0xDC). Max value allowed is 1sec (0xF4240). Here is an oscilloscope trace of the reader output signal showing the effect of changing this setting.



The following additional Nano bugs, identified in the previous release version 1.7.2, have been addressed in version 1.7.3:

- Changes made between firmware versions 1.7.3 and 1.7.2 to enable duty cycle control were found to impact the accuracy of the read time relative to that requested in a timed read request. The read time was found to be significantly longer than the time requested. Version 1.7.3 corrects this and the actual read times are now within 4 msec of that requested. (Ref #5551)

- Gen2V2 (NXP UCODE AES and NMV2D) tag operations now works fine on Nano (In previous release v1.7.2, Gen2V2 standalone and embedded tag operations were broken). (Ref #5549).

Hop time, hop table, GEN2 Q and Tari parameters are now restored when read plan configuration is saved and the module is rebooted. Previously, the module returned default values after boot instead of saved values. (Ref# 5515).

ThingMagic Nano Firmware v1.7.2 Release Notes

This release note describes the features and characteristics of ThingMagic Nano module firmware (FW) version 1.7.2 relative to version 1.7.1. Release notes for 1.71 and 1.5.0 have been included since they contain information not yet found in the Nano User Guide. Nano firmware is not supported on any other ThingMagic module.

Nano firmware version 1.7.2 has been developed in conjunction with version 1.29.4 of the MercuryAPI and should be used with that version (or higher) to achieve best results. Previous versions of the API will not support all the features of this firmware release. See the API release notes and MercuryAPI Programmer's Guide for further information on its features and functions.

For full information about the ThingMagic Nano product, please consult the Nano Design Guide, which can be found here:

http://www.thingmagic.com/manuals-firmware

New Features

New features of version 1.7.2 include the following:

Margin read support for Monza6 tags

MarginRead is an EPC Gen2 compliant custom command supported by tag chips with the “Integra” feature. This command allows a reader to explicitly verify that the non-volatile memory (NVM) in the tag chip is not weakly written, guaranteeing a minimum margin on NVM. It is used for quality control to ensure data integrity and for failure analysis.

There are several ways that the MarginRead command could be used with Monza 6. A recommended use of MarginRead is independent verification of the encoding quality, either on a sample basis or for diagnosis during failure analysis.

A basic description of MarginRead:

> When data is written to a tag using the Gen2 protocol, charge is built up in the memory cells until they reach the appropriate level. Once that happens, the tag returns a "done" signal telling the interrogator (reader) or encoding system that the write operation has completed successfully.
>
> It is a known field issue that not all encoding systems properly wait for the "done" signal and instead issue a read operation to check if the data is correct. A read operation may return correct data even if the write operation did not complete successfully.
>
> A partially charged memory cell might retain data for a limited time but then it will lose data integrity over time. Data retention could be for an unpredictable amount of time from a few minutes to several years.
>
> A fully charged memory cell will retain data for a long period of time. Specifically, the Monza 6 tag is expected to retain data for up to 50 years.
>
> The MarginRead command allows customers to check if Monza 6 tag chip memory cells are fully charged.
>
> If a customer encounters a data integrity issue in the field, MarginRead may be used to diagnose the problem. If MarginRead indicates an issue, then the encoding method should be investigated.

Refer Mercury API v1.29.4 release notes for API commands to work with this functionality.

Gen2 parameters in metadata

Now that we allow modification of the Gen2 parameters on the fly, it is desirable to include current Gen2 settings as metadata in tag reads so that the current settings under which the tag was read are reported.

For example, the fact that Gen2 Q value can change dynamically and if the customer is trying to determine the best static value, it would be good to know the value that our automated algorithm has selected. Gen2 parameters that are now included in metadata are:

- Gen2 Q
- Gen2 Link Frequency
- Gen2 Target

Gen2 Q, BLF and Target parameters have been added to the TagReadData.TagMetadata method. The Read code sample in the MercuryAPI SDK shows how to activate this functionality.

Support for Acura Gen2V2 tags

NMV2D tag support has been added in Nano FW, which returns 352(256+96) bits in TAM2 reply for ProtModes 0x02 and 0x03 and 256 bits for ProtModes 0x00 and 0x01. Previous release version of FW v1.7.1 replies with 256 bits irrespective of any ProtMode.

The NMV2D tag supports the same set of commands as NXP UCODE AES tag except following: -

1. NXP UCODE AES tag chip only supports ProtMode=1 while NMV2D tag supports ProtModes=0,1,2,3.
2. Untrace-Access and Untrace-Authen commands don't work for NMV2D tag as they do for UCODE AES tag.

Refer to the Authenticate, ReadBuffer and Untraceable code samples in the MercuryAPI SDK to test this functionality.

Support for GEN2V2 embedded tag ops

In previous firmware releases, GEN2V2 operations that supported the NXP UCODE DNA tag were only available as stand-alone, single tag functions. In this firmware release, support for

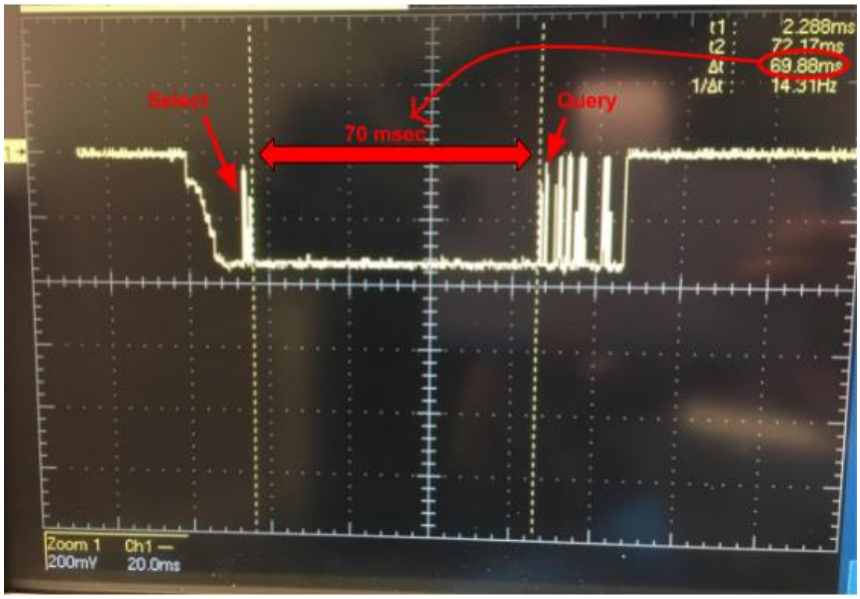
embedded tag operations has been added for both NXP UCODE DNAtag and the NMV2D tag. This allows for high speed secure reading in Asynchronous modes.

Refer to the Authenticate, ReadBuffer and Untraceable code samples in the MercuryAPI SDK to test this functionality.

Configurable T4 for Gen2 protocol

Some sensor tags use a Select command to trigger reading of their sensor. We found that the time our reader was waiting between the Select command and start of inventory (when the reader sends the Query command) was insufficient for the sensor tag to obtain its reading before having to report it to the reader. This delay time is controlled by a Gen2 parameter called the “T4 timer”. We have now added the ability to set the T4 timer to a larger value.
The parameter ‘PROTOCOL_PARAM_GEN2_T4’ has been added to set/get the Gen2 T4 parameter with sub command for 0x9b and allow T4 to be set in microseconds.

T4 value is a 4-byte in length and specified in microseconds. Minimum value of T4 allowed is 64 microseconds (0x40). Max value allowed is 1 second (0xF4240). Here is an oscilloscope trace of the reader output signal showing the effect of changing this setting.



Refer Mercury API v1.29.4 release notes for API commands to work with this functionality.

Ability to “Read Data” immediately after sending a “Write EPC” or “Write Data” command

Some sensor tags require the module to write to a memory bank to trigger the sensor measurement, then read the sensor data field without dropping power between (as happens if the two operations are done as separate commands). This functionality could be useful for streamlining read-then-write operations for other applications as well.

To support this requirement, Read Data support has been added as an option for the Write EPC and Write Bank Data commands. This allows the module to read the data from any of the memory banks following a successful write operation of data to any memory bank (or write EPC) through a single command. The standard commands to Write Tag Data and Write Tag EPC will now optionally include the read memory bank, read word address, and read count to implement this feature.

For more details on the application interface, refer to WriteTag code sample in the MercuryAPI SDK.

Decoupling antenna selection from AsynchOnTime

In previous releases, when reading continuously, the reader would always return to antenna 1 (or the first antenna in the configured list) at the beginning of each AsynchOnTime cycle. This encouraged users to configure a high value for AsyncOnTime to ensure that all antennas would be activated each read cycle. However, some of the settings that can now be changed without interrupting reading will take effect only at the beginning of the next AsyncOnTime cycle, so users now have a contrary reason to set this value as small as possible.

To eliminate this contradiction, the antenna selection algorithm has been changed to remember the last antenna that was active in the previous read cycle and start with that antenna for the next AsyncOnTime cycle. This way, the active antenna cycles through the list with regularity and the AsynchOnTime can be optimized to make the time for on-the-fly settings take effect as quickly as possible, or for other reasons.

The ReadAsync code sample can be run to see the effect of this change.

Support for additional Asian regions

One of our customers calculated the optimum channel frequencies to permit the greatest number of channels while still meeting out-of-band emissions standards for Asian regions and Russia. Although the frequency range of these new regions was within the existing regions, the exact desired channels could not be realized with our quantization rules (all channels must be an even multiple of the quantization value above the low channel boundary for that region). To achieve the desired channels, the following additional regions have been added.

Regions that are added in current firmware version have the following characteristics

Region	Region Number	Region Number	Low Channel Boundary	High Channel Boundary	Min Step Size (Quantiztion)	Hop Table	Max RF Power Allowed
Malaysia	MY	0x10	919 MHz	923 MHz	250 kHz	921750, 919250, 920750, 922250, 919750, 921250, 920250, 922750	31.5 dBm
Indonesia	ID	0x11	923 MHz	925 MHz	125 kHz	924625, 923375, 924125, 923875, 924375, 923625, 924875, 923125	31.5 dBm
Philippines	PH	0x12	918 MHz	920 MHz	250 kHz	919250, 918750, 919750, 918250	31.5 dBm
Taiwan	TW	0x13	922 MHz	928 MHz	250 KHz	926250, 924750, 922250, 925750, 923250, 927750, 926750, 924250, 922750, 925250, 923750, 927250	30 dBm

Region	Region Number	Region Number	Low Channel Boundary	High Channel Boundary	Min Step Size (Quantiztion)	Hop Table	Max RF Power Allowed
Macao	MO	0x14	920 MHz	925 MHz	250 kHz	923250, 921750, 924250, 922750, 920250, 923750, 921250, 924750, 922250, 920750	31.5 dBm
Russia	RU	0x15	866 MHz	868 MHz	200 kHz	866600, 867800, 866200, 867000, 866400, 867600, 866800, 867200	31.5 dBm
Singapore	SG	0x16	920 MHz	925 MHz	100 kHz	923100, 921900, 924300, 920700, 922500, 923700, 921300, 924900, 920100	31.5 dBm

Notes:

(1) Maximum Dwell Time 0.4 sec for all these regions (same as North American region)
(2) Max RF power limit is that given in table or whatever the module is capable of, whichever is lower.
(3) Any channel frequency can be requested that is between the upper and lower bounds, but the module will silently round down to the nearest channel that is the lower bound plus an integer multiple of quantization steps.

The new Asian regions have been added to Reader. Region method. Refer Mercury API v1.29.4 release notes for more information.

Support for set/get quantization value and min frequency

The Open region as defined in previous releases was intended for testing only. In order to permit the most flexibility in defining channels, it allowed a minimum channel step size

(quantization) of 25 kHz. We did not recommend the use of the Open region to support channel plans which could not be easily accommodated by changing the hop table of an existing region because such a small step size will result in lower channel frequency stability. (This frequency not only defines the minimum step size that can be set, but also represents how often the channel is nudged back to its desired value - more frequent nudges create a more stable channel.)

To allow the Open region to be used more flexibility, we now permit the setting of the quantization value. It may be any value between 15 kHz and 6 MHz, but must divide evenly into 6 MHz (6000 kHz). If not, an error will be returned (error code number 105).

To permit the largest quantization value possible, we also allow setting the minimum frequency value for the Open region. (Smaller quantization values are often driven by the rule that all channels must be an integral multiple of the quantization value above the minimum frequency value).

Only the Open region supports changing of the quantization value. We do not recommend that quantization values less than 100 kHz be used except for laboratory testing to maintain a high degree of channel frequency stability and prevent interference with other readers or RF services.

Refer Mercury API v1.29.4 release notes to activate this functionality.

Resolved Issues

The following Nano bugs, identified in the release notes for version 1.7.1, have been addressed in version 1.7.2:

- The channel frequency jitter issue is resolved after creating dedicated regions for 7 Asian regions (Malaysia, Indonesia, Singapore, Philippines, Macau, Russia and Taiwan) and allowing configuration of the quantization value in the Open regioon. (With previous release firmware, high Jitter had been observed while testing the Nano module with the lowest frequency using Open region). (Ref# 5347).

- The Gen2 tag kill function is now more reliable. In the previous firmware version, we found that some tags which had been reported as being killed were able to be read. (Ref# 5427).

- Embedded async read error reporting is more accurate now. (With previous firmware releases, the module was returning the 'No Tags Found(400h)' responses along with data when we perform embedded async read operations.) (Ref# 5504).

Operational Notes

The following restrictions and caveats apply to the features and functionality of firmware version 1.7.2

- Customers have reported getting a timeout error when continuous read(async) is run for several days, controlled by Universal Reader Assistant. The module must be reset to clear this condition. It has not been observed for continuous autonomous reading and the failure mechanism has not been identified yet. (Ref# 5323).

Per the Gen2 specification, the Nano module tells the tag how fast to modulate the data it returns. Tags attempt to transmit at that "Backscatter Link Frequency", but some variation is permitted. The Nano module is expected to read tags in between 225 and 275 kHz BLF (250 kHz +/-10%). However, the Nano currently reads tags in the range of 235 to 340 kHz – a wider range than required, but not completely covering the expected range. Though it covers a wider range in the higher limit (reads up to 340 kHz BLF), lower limit is not reached. (Ref# 4402).

ThingMagic Nano Firmware v1.7.1 and 1.5.0 Release Notes

These release notes describe features and characteristics of ThingMagic Nano module firmware (FW) version 1.7.1. This firmware is not supported on any other ThingMagic module. For reference, the release notes for version 1.5.0 of firmware are included as well.

WARNING!

This version of firmware contains a fix for a very serious issue that could cause Nano modules to become inoperative. All Nano modules must be upgraded to this version. Previous versions of firmware are no longer supported.

Nano firmware version 1.7.1 has been developed in conjunction with version 1.29.2 of the MercuryAPI. Previous versions of the API will not support all the features of this firmware release.

New Features

v1.7.1

Save/Restore of Hop Table and Hop Time

For use with Autonomous Operation, or if you simply wish to restore settings on reboot without needing to send many configuration commands, the hop table and hop time may be stored and retrieved manually or automatically.

Hop Table and Hop Time (maximum dwell time on any channel) are necessary to use in countries that do not permit all the channels we provide in default region settings. For example, many countries have regulatory requirements that are identical to that of our North American (“NA”) region, but do not permit the full range of channels from 902 to 928 MHz.

There are no new settings associated with this, it is simply now included when the Save, Restore, Verify, and Clear utilities are used.

Ability to Change Settings During Continuous Reading

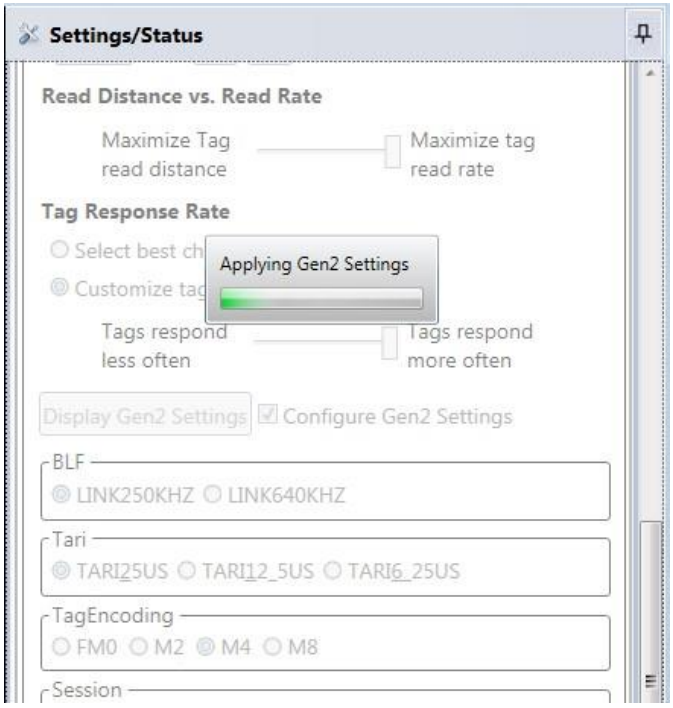
In previous firmware releases, continuous reading had to be halted in order to change any settings. Now, a subset of available settings can be changed while the reader is actively reading. This allows the host to optimize settings on-the-fly. Settings that are supported during continuous reading are:

- Global TX Read Power
- Global TX Write Power
- Gen2 BLF
- Gen2 TARI
- Gen2 Encoding (“M” value)
- Gen2 Q
- Gen2 Session
- Gen2 Target
- GPO line state (and learn the value of GPI lines). Note: You cannot change the sense of a line (i.e. input to output) during continuous reading.

No special command is needed to set parameters during continuous reading. The API will automatically send the correct command to the module based on its knowledge of the state the module is in.

Universal Reader Assistant can now demonstrate the ability to change settings during continuous reading. Any settings in the “Display Gen2 Settings” category can be altered, as well as the global read and write power levels (although write power is of limited use since the “write” tag operation cannot be specified under continuous reading in this version of Universal Reader Assistant).

Changes to the power levels are applied silently. Changes to Gen2 parameters result in a pop-up progress bar which disables further changes until the one you made is applied, as shown below.



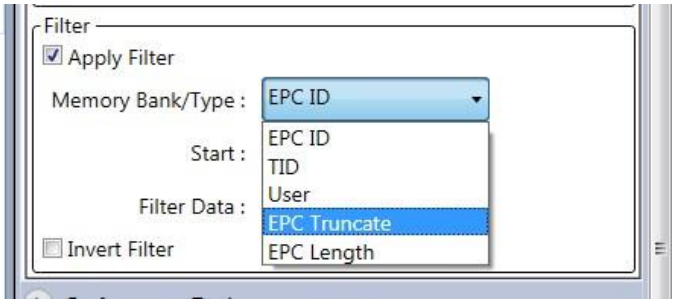
Meta-data Control at Module Level

In previous releases, all tag meta-data (read count, RSSI, timestamp, etc) was returned by the module and choices of what meta-data to report simply caused the API to discard unwanted values. Now, the meta-data selection information is transferred to the module and the module only reports desired values, resulting in a small increase in performance under some circumstances. No additional configuration parameters are necessary to take advantage of this feature.

Filtering on Tag Length and EPC Truncation Support

Two enhancements have been added to limit tags returned by the module. One only returns tags if their EPC is of the expected length, which is useful to weed out stray and phantom tags. The other instructs the module to announce that only tags whose EPCs contain a certain beginning value and length should respond. The desired EPC value includes both the PC word (which gives the EPC length) and the desired starting value for EPC. Not only do tags not respond if they do not have that start value and length, but they only respond with the unique portion of their EPC (not the shared prefix value) to increase performance.

Universal Reader Assistant can now demonstrate filtering based on tag length and EPC truncation, as shown below.

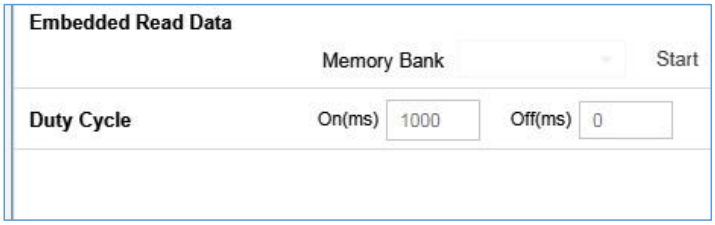


Note that “EPC Truncate” is difficult to distinguish between a normal filter on EPC ID because the part of the EPC that is not reported by the tag is appended to the EPC as reported in the tag results screen.

Ability to Set the Duty Cycle for Continuous Reading

In previous software releases, duty cycle was controlled by the host by starting and stopping timed reading. With this release, the module has the ability to control the duty cycle, allowing the host to not have to interact with the module as much and permitting greater control under Autonomous Operation, which previously only supported continuous reading at a 100% duty cycle.

The Autonomous Configuration Tool now supports duty cycle control, to complement support in emerging versions of module firmware. This will allow the module firmware to control duty cycle to save battery life and reduce temperature rise.



Save/Restore of additional Gen2 Settings

For use with Autonomous Operation, or if you simply wish to restore settings on reboot without needing to send many configuration commands, the Gen2 values for Q and TARI can now be saved and restored on the module. These settings are not yet supported in the Autonomous Configuration Tool.

ThingMagic offers an application note which explains the use and selection of these values if you do not know what they mean.

“Save”, “Restore”, “Verify” and “Clear” work as they did before, but encompass more settings.

v1.5.0

Gen2V2 Support

The ThingMagic Nano now supports the Gen2V2 features of the NXP DNA tags. These features include:

- “Untraceable”. Ability to limit reading of all or part of EPC, TID and User memory fields by unauthorized readers.
- Ability to download and activate security keys.
- Ability to authenticate tag using random challenge strings and AES encryption.
- Ability to obtain memory data in encrypted form, which can be successfully decoded if the host knows the key that has been activated on the tag.
- Ability to obtain authentication and encrypted memory data from a tag buffer rather than the tag backscattering that information to the reader immediately.

These capabilities are supported in the 1.27.2 version of the API and may be demonstrated using code samples and the version of Universal Reader Assistant which is distributed with the API.

Temperature limits extended to -40C

The Nano has been qualified to operate down to -40 C, so the low temperature limit imposed by the firmware have been lowered from -20 C to - 40 C.

Support for Return Loss Measurement

The firmware will now estimate the return loss of its port, based on multiple readings at multiple channels within the active region. (Note: For the North American region, with 50 channels, this measurement can take as long as 600 msec). The return loss value can be obtained though the API by getting the “/reader/antenna/returnloss” parameter value as well as by using the “CmdGetAntennaReturnLoss” method. The sample code “ReaderStats” has been enhanced to illustrate the recommended method for obtaining this information. The values returned will look like this:


```
Antenna Return Loss
 Antenna 1 | 30
```

Which indicates a return loss of 30 dB for the antenna port.

Note that this measurement loses accuracy as the numbers increase due to the impact of internal signal reflections that increasingly obscure the measurement of the small signal reflected only at the antenna.

The return loss is measured at an RF level of +15 dB in order to limit impact to other services that are running in the same region while the return loss measurement is being made.

Support for Antenna Detection

Antenna detection is now supported for the ThingMagic Nano, but uses the new return loss measurement as opposed to testing the port with a small amount of DC current as is done in the M6e module. A port that shows a return loss of less than 10 dB (for example the value “4”: above) is considered to be un-terminated. A port with a return loss greater than or equal to 10 dB is considered to be terminated (for example the value “30” above).

Antenna detection is only activated when /reader/antenna/checkport is enabled. Unlike the M6e module, this only enables detecting antennas with an explicit request, not checking each time the reader transmits. It is recommended that the host application periodically check for antenna status and adjust the antenna list based on which antennas are present.

Autonomous Operation

The ThingMagic Nano can now be configured to operate in autonomous reading mode. This provides the ability to setup the module to read and output tag results without a host MercuryAPI application by enabling hardware read triggers and savable configuration settings in the module.

In order to setup the module for autonomous operation the Autonomous Configuration Tool (ACT), distributed as version “autoread-1.27.2” or later must be used. (When installed, the home screen indicates that the software version is 1.0.0.0, built on Mercury API version 1.27.0.8”

Save/Restore Configuration Settings

Autonomous operation relies on the ability of the module to store configuration settings across reboots. This ability can be used even if autonomous operation is not enabled - it will essentially change the default values that the module uses whenever it is rebooted. The settings that can be manipulated in this way include:

- Serial interface baud rate

- Region of operation
- Default protocol (only “Gen2” is supported on Nano)
- Gen2 settings (“M” value, BLF, Session, Target)
- RF Power level

Support for narrow North American Regions (NA2, NA3)

In order to support narrow-band antennas, a new region has been introduced, “NA3”. Here are the characteristics of this new region, relative to the existing “NA2” region:

Paremeter	NA2 region	NA3 region
Enumerated Region Code	0x0D	0x0E
Lowest Permitted Frequency	917400 kHz	917500 kHz
Highest Permitted Frequency	927200 kHz	922500 kHz
Minimum Channel Spacing	200 kHz	100 kHz
Number of Channels in Default Hop Table	50	51

Both these channels are legal for use in regions that adhere to FCC regulatory requirements.

Support for 25 kHz channel spacing in “Open” region

The minimum channel spacing for the “Open” region has been reduced from 100 kHz to 25 kHz in order to accommodate more flexible channel definitions. Here is a table that gives the information for all supported channels for the ThingMagic Nano module.

Region Name	NA2	NA3	IN	JP	PRC	EU3	KR2	AU	NZ	Open
Lowest Freq permitted	91740	917500	865000	916800	920125	865600	917000	920000	922000	859000 / 915000
Highest Freq Permitted	92720	922500	867000	923400	924875	867600	923500	926000	927500	873000 / 930000

Region Name	NA2	NA3	IN	JP	PRC	EU3	KR2	AU	NZ	Open
Smallest Step Size (1)	200	100	100	100	125	100	100	250	250	25
Maximum Dwell Time	0.4 sec	0.4 sec	4 sec	4 sec	2 sec	4 sec	0.4 sec	0.4 sec	0.4 sec	0.4 sec
LBT Level (4)	None	None	None	-74 dBm	None	Optional at -72 dBm	None	None	None	Optional at -96 dBm
Lowest Chan in Hop Table	91740	917500	865200	916800	920625	865700	917300	920750	922250	859000
Highest Chan in Hop Table	92720	922500	866800	920800	924375	867500	920300	925250	927250	930000
Hop Table (2)	92600	918800	866000	92080	92137	86570	91730	92375	92375	859000
	91960	919600	865600	91920	92262	86630	91910	92125	92625	860000
	92480	918400	866800	92060	92087	86690	91790	92425	92425	861000
	92260	920800	865200	91680	92362	86750	91850	92275	92675	862000
	92160	919500	866400	91800	921125		92030	92525	92325	863000
	92520	922500		920400	920625		91970	92175	92525	864000
	91840	921700			923125			92475	92275	865000
	91920	918000			921625			92225	92725	866000
	92360	917500			922125			92325	92575	867000
	92180	919700			923875			92075	92225	868000
	92320	919100			921875				92475	869000
	92020	917900			922875					870000
	91940	918200			924125					871000
	92120	918700			923375					872000
	92660	920500			924375					873000
	92680	922100			922375					915000
	92080	920100								916000

Region Name	NA2	NA3	IN	JP	PRC	EU3	KR2	AU	NZ	Open
	92440	918100								917000
	92380	922000								918000
	92100	919300								919000
	92460	921200								920000
	92200	921500								921000
	91780	917600								922000
	92280	919200								923000
	91900	921300								924000
	91740	918500								925000
	92640	921400								926000
	92400	922200								927000
	91980	918900								928000
	92620	921100								929000
	92500	920300								930000
	91860	919000								
	91880	917800								
	92300	920900								
	91820	922300								
	92700	921800								
	92540	917700								
	92220	920200								
	92240	922400								
	91760	920000								
	92040	921600								
	92580	919800								

Region Name	NA2	NA3	IN	JP	PRC	EU3	KR2	AU	NZ	Open
	92060	919400								
	92420	919900								
	91800	918300								
	92720	921000								
	92000	918600								
	92140	921900								
	92340	920700								
	92560	920600								
		920400								

Footnotes:

- Permissible channels are the lowest channel and the lowest channel plus multiples of the "smallest increment", up to the value of the highest channel. No errors will result if frequencies are selected between these values, but the module will automatically round up or down to the nearest permitted value.
- This is the exact order of channels in the default hop table
- LBT is "Listen Before Talk". If a carrier is detected above the threshold, the channel will not be used.

Default Output Power Level Changed to +23 dBm

The factory default RF Output Power level has been changed to +23 dBm, from +27 dBm. A different default level may be configured and stored in module memory if desired.

Antenna Multiplexer Support

The ThingMagic Nano now has the ability to control an antenna multiplexer through one or two of its GPO lines. This allows the antenna port to be split into 2 or 4 antenna ports. Instructions for how to configure this mode of operation are provided in the ThingMagic Nano Design Guide. One configured, the Nano behaves as if it had additional physical antenna ports for configuration and reporting purposes.

Tag Buffer Length Increased

The tag buffer length had been 38 tags. It is now 50 tags. This will allow information for up to 50 unique tags to be stored during a single synchronous read cycle or allow up to 50 tag read records to be stored during continuous read operations when they cannot be offloaded immediately to the host.

Reader Stats Reporting Now Supported

The API parameter "/reader/stats/enable" is now supported. This was not a bug in the firmware, but rather the Mercury API erroneously reporting the error: "reader received an invalid or unsupported protocol". This has been fixed in MercuryAPI version 1.27.2 and is now supported in all API languages.

Resolved Issues

v1.7.1

- Nano 1.5.0 firmware reported RSSI values that are too low; firmware releases prior to 1.5.0 were accurate; accuracy has been restored (Ref# 4976)

Flash Memory Locking Issue

We have discovered that under certain circumstances Nano firmware can be corrupted, resulting in a module that may or may not communicate with the host, or may not start active RFID operation. What the host sees is a "CRC" error message or no response from Nano module. This failure condition is most often caused by insufficient power being supplied to the module during high demand but may occur under other circumstances as well.

Typically these conditions are handled by securely locking firmware as per the microcontroller data-sheet and user guide recommendations. Secure locking was NOT correctly implemented in the Nano module firmware leading to above corruption.

To prevent this corruption, we have created this version of firmware that securely locks the firmware. To activate this fix, you need to update the firmware to version 1.7.1 or higher. Note that once the corruption occurs, this remedy does not work because securely locking the corrupted firmware only makes the flaw permanent. To prevent your operational Nano modules from becoming irreversibly corrupted, we recommend that all Nano modules currently in the field be upgraded immediately. There is no down-side to upgrading and much risk to using earlier versions of code over an extended period of time. If you currently have modules that exhibit the failures, please return them immediately for replacement under an RMA before their warranty expires.

Note that this issue is ONLY related to Nano module firmware. All other ThingMagic modules have securely locked firmware.

v1.5.0

- The receiver tolerance for Backscatter Link Frequency deviation has been improved to accommodate tags that have wider variation than normal. For example, tags that respond with a BLF of 225 kHz when a BLF of 250 kHz is requested by the reader will be accommodated. (Ref#4402)

- The “NA” region has been removed from the supported list because it contains too many channels that do not fall within the range of the ThingMagic Nano. (ref#4579) Resolved Issues.

- All 4 GPIO lines can now be configured and used. (Previous versions of firmware only supported 2 of the 4 lines). (Ref#4583)

- Reading and writing to large memory tags now works correctly (in previous firmware releases, addresses above a value of 128 bytes would not work correctly) (Ref#4702).

Operational Issues

v1.7.1

- When the Truncation filter is applied, the tags will return data even if the Access Password is not correct. (Ref# 5075)

No Ability to “Get” Saved Value of Settings

The module firmware is able to save many settings in flash memory. As of firmware version 1.7.1, the module has the ability to report the following values from flash memory, but until the next release of the API, this feature will not be available to users.

- Baud Rate
- Region
- Protocol
- Hop Table
- Hop Time
- Read Power
- Per-port Read Power

    - Antenna Configuration
    - Gen2 Session
    - Gen2 Target
    - Gen2 “M” value
    - Gen2 Backscatter Link Frequency
    - Gen2 TARI
    - Gen2 Q
    - Enable Filtering Value
- Trigger Read GPIO Value