ThingMagic M6e Firmware v1.23.2 Release Notes

This release note describes the features and characteristics of ThingMagic M6e module firmware (FW) version 1.23.2. 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 1.31.4.

The release notes for firmware version 1.23.1, 1.23.0, 1.21.2, 1.21.1 and 1.19.0 are appended to these release notes for reference.

Topics covered in this release notes are:

For full information about the ThingMagic M6e product, please consult the User Manual, which can be found at:

https://www.jadaktech.com/documentation/rfid/mercury6e-modules/

M6e module has been modified with hardware changes due to component obsolescence as well as improvements for ESD immunity enhancements and better protection for antenna connectors to prevent the connectors from being detached from the PCBA.

ALL features implemented in this release support the modified M6e module. This firmware version is backwards compatible with previous revisions of M6e hardware; however, the following features are NOT supported on previous revisions:

- ***Support for Japan and Israel region in standard M6e-A (restricted to –JIC SKU in previous version)***
- ***Support for OPEN Extended region***

JADAK
A Novanta Company

Contents

New Features

This version 1.23.2 doesn't have any new features.

Resolved Issues

The following M6e bugs were fixed in this release version 1.23.2:

- Power accuracy and tag read issues at low power levels(<15dBm) on M6ePlus have been resolved in this firmware release.
- Tag read performance has improved and minor signal distortion was eliminated at higher power levels (>=30dBm) on M6ePlus in this firmware release. As a result, there could be a slight increase in power consumption if the module operated into a load that is not well matched to 50 Ohms.
- RSSI level has been re-calibrated for M6ePlus in order to present the value as accurate as possible to the user.

Operational notes

The following restrictions and caveats apply to the features and functionality of firmware version 1.23.2:

- User may observe slight variance in tag read rate (400+ Tag Population) with MultiReadPlan Async read compared to earlier release 1.23.0. (#Ref 6628)

ThingMagic M6e Firmware v1.23.1 Release Notes

This release note describes the features and characteristics of ThingMagic M6e module firmware (FW) version 1.23.1. 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 1.31.3.

The release notes for firmware version 1.23.0, 1.21.2, 1.21.1 and 1.19.0 are appended to these release notes for reference.

Topics covered in this release notes are:

For full information about the ThingMagic M6e product, please consult the User Manual, which can be found at:

https://www.jadaktech.com/documentation/rfid/mercury6e-modules/

M6e module has been modified with hardware changes due to component obsolescence as well as improvements for ESD immunity enhancements and better protection for antenna connectors to prevent the connectors from being detached from the PCBA.

ALL features implemented in this release support the modified M6e module. This firmware version is backwards compatible with previous revisions of M6e hardware; however, the following features are NOT supported on previous revisions:

- ***Support for Japan and Israel region in standard M6e-A (restricted to –JIC SKU in previous version)***
- ***Support for OPEN Extended region***

New Features

New features of version 1.23.1 include the following:

Support for EU4 Region

In addition to 800MHz lower band channels, the ETSI standard EN 302 208 v3.1.1(2016-11) now includes 900MHz upper band channels also. So a new region EU4 is added to support this upper band with 3 channels (916.3MHz, 917.5MHz and 918.7MHz).

This Region has following characteristics

Region name	EU4
Serial Interface code	0x1E (decimal 30)
Supported modules	M6e, M6e-Plus
Lowest Freq permitted	916300 KHz
Highest Freq permitted	918700 KHz
Smallest step size	200KHz
Highest dwell time	4 sec
Max RF power	31.5 dBm
LBT level	None
Lowest channel in hop table	916300 KHz
Highest channel in hop table	918700 KHz
Number of channels	3
Chan 1	916300 KHz
Chan 2	917500 KHz
Chan 3	918700 KHz

Return GPI status in Tag Metadata

In previous firmware releases, module had the capability to return GPO status in tag metadata. Now with this current firmware release end users can retrieve GPI status also along with GPO status in tag metadata. The Read code sample in the MercuryAPI SDK shows how to activate this functionality.

Support for Setting Custom Open Region (Open Region Enhancement)

In this firmware release, we have introduced additional settings to allow the Open region to be used to create new regions. The region must be set first because not all the settings described here can be applied to other regions.

Here are the relevant settings. The first 4 steps must be done in the order shown.

1. Set the region to “Open”.
2. Set “Quantization Step”: Defines the set of permissible channels.
3. Set “Minimum Frequency”: Defines the lower end of the desired band.
4. Set “Hop Table”: Defines the desired channels, in the order they will be used.
5. Set “Hop Time”: Defines the maximum length of time the reader can occupy a channel.

6. Set “Dwell Time”: Defines the minimum length of time a channel must be avoided after having been used. By default, channels in the Open Region can be re-used at will, so there is a “Dwell Time Enable” setting also that informs the reader that you would like to set a Dwell Time limit.
7. Set “LBT Threshold”: LBT stands for “Listen Before Talk”. It instructs the reader to listen on a channel before using it and to avoid the channel if a signal is already present at a value above a threshold limit. By default, the reader will simply use a channel, so there is an “LBT Enable” setting that informs the reader that you would like to activate LBT functionality.

Note that:

- “Dwell Time Enable”, “Dwell Time”, “Quantization Step” and “Minimum Frequency” can be set for OPEN region only. Error code 0x105 (Invalid parameter value) will be returned if user tries to set these parameters for other regions.
- “Hop Time” and “Hop Table” can be set for all regions.
- “LBT enable” and “LBT Threshold” can be set for OPEN, EU3, JP, JP2, JP3 regions; for other regions, error code 0x105 will be returned.

Details on the use of these settings are provided in next sections. Code samples showing how to use these settings are available in the latest MercuryAPI SDK. All these settings can be saved to module memory, and restored, with a single command, so it is not necessary to configure the module each time it is rebooted.

Setting the Open Region

Depending on the API used (C, C# or Java), set **"/reader/region/id"** to **Reader.Region.OPEN** or set **TMR_PARAM_REGION_ID** to **TMR_REGION_OPEN**.

Setting the Quantization Step

The Quantization Step setting defines all permissible channels, although the channels defined in the Hop Table are often a subset of these. The Quantization Step value affects both the minimum spacing between channels and the actual values of the channels themselves.

The selected value must adhere to certain constraints imposed by our reader design:

- The value must be evenly divisible into 6 MHz (6000 kHz).
- The values of the channels in the hop table must be an integer multiple of the Quantization Value.
- The lowest permitted value is 15 kHz, but Quantization Values below 100 kHz may create conditions where channels do not lock onto frequency properly, so are not recommended. Always use the largest Quantization Value consistent with your channel plan for best performance.

Example:
If you desire a channel spacing of 500 kHz (6 MHz/500 kHz=12000, which is a legal value), then the channels in the hop table would have to be on half-MHz boundaries (i.e. 902.0 kHz. 902.5 kHz, etc). If you wanted the channels to start at 902.75 kHz, you could not achieve this with a 500 kHz Quantization Value. You would have to reduce the Quantization Value to 250 kHz.

The API Parameter to set the Quantization Step is:

TMR_PARAM_REGION_QUANTIZATION_STEP
or
"/reader/region/quantizationStep" (depending on the API version in use).

Minimum Frequency Setting

Because we always round down to the next valid channel based on the Quantization step, we felt it was necessary to know the lowest frequency permitted so that if the hop table defined a channel frequency that was lower than desired, we would return an error rather than silently rounding down to a channel outside the desired band (There is no such check for the highest channel).

The Lowest Channel and all channels in the hop table must be within the permissible range for the module. These ranges are not contiguous. Here are the ranges for the module families (the limit frequencies given are valid, too):

For the Micro and M6e modules:
840000 KHz to 869000 KHz
or
902000 KHz to 928000 KHz

For the Nano module:
859000 KHz to 873000 KHz
or
915000 KHz to 930000 KHz

The API Parameter to set the Minimum Frequency is:
TMR_PARAM_REGION_MINIMUM_FREQUENCY
or
"/reader/region/minimumFrequency" (depending on the API version in use)

Hop Table

The “Hop Table” is an ordered list of channel frequencies (in kHz units). Up to 62 frequencies are supported (the North American region requires at least 50 channels). The only check on the value of entries in the hop table is whether the channel is within the range for that region (including the Minimum Frequency for the Open region).

Only frequencies that are an integer multiple of the Quantization Step are valid, but there is no check on this when the table is entered. Instead, any assigned channel which is not an integer multiple of the Quantization Step will be rounded down to the nearest permitted frequency when that channel is used. No notification of this will be provided by the reader. The frequency at which the tag was read (optionally reported with each tag read) will be the frequency you assigned – not the one actually used, so the only way to know that this rounding has occurred is to actually measure the carrier frequency.

The API parameter to set the Hop Table is:
TMR_PARAM_REGION_HOPTABLE
or
"/reader/region/hopTable" (depending on the API version in use)

Hop Time

Hop time is the maximum time a channel can be occupied. Typically, this time will only be fully utilized when the reader is continuously reading. If executing individual commands, the reader will hop to the next channel before each command.

Its valid range is 1 to 65535 msec. The default value is 400 (msec).

The API parameter to set the Hop Table is:
TMR_PARAM_REGION_HOPTIME
or
"/reader/region/hopTime" (depending on the API version in use)

Dwell Time

The "Dwell Time" is the minimum time a channel must remain unused once it has been occupied. It is enforced even if the channel was not previously occupied for the entire Hop Time).

By default, there is no limit on how long the reader will wait before returning to a channel in case the user wants to have a single channel be used repeatedly for testing. If a Dwell Time is to be enforced, this function must be activated by a "Dwell Time Enable" flag.

The API parameter which controls this is:
TMR_PARAM_REGION_DWELL_TIME_ENABLE
or
"/reader/region/dwellTime/enable" (depending on the API version)

The default value for Dwell Time is 100 milliseconds, if enabled. Its valid range is 1 to 65535 milliseconds.

The API parameter which controls this is:
TMR_PARAM_REGION_DWELL_TIME
or
"/reader/region/dwellTime" (depending on the API version).

Listen Before Talk (LBT)

If Listen-Before-Talk (LBT) is enabled, a channel will be skipped in the hopping sequence if the channel is already occupied by a signal at or above this threshold.

Valid values are from -128 to 0 dBm and the API parameter that controls this is:
TMR_PARAM_REGION_LBT_THRESHOLD
or
"/reader/region/lbtThreshold"

There is no check for interfering signals unless LBT is activated using the "Enable LBT" flag. Its default is "false" (disabled). The API parameter which controls this is:
TMR_PARAM_REGION_LBT_ENABLE
or

"/reader/region/lbt/enable"

Support for Brazil region via custom Open region

Now we can set Brazil region via Open region by setting following parameters

To create Brazil region:
1) Set OPEN region
2) Set hop table (902-907, 915-928)
(902250, 902750, 903250, 903750, 904250, 904750, 905250, 905750, 906250, 906750, 915250, 915750, 916250, 916750, 917250, 917750, 918250, 918750, 919250, 919750, 920250, 920750, 921250, 921750, 922250, 922750, 923250, 923750, 924250, 924750,
925250, 925750, 926250, 926750, 927250, 927750)
3) Disable LBTenable
4) Disable dwellTimeEnable
5) Set hop time to 375ms
6) Set quantization step to 250KHz
7) Set minimum frequency to 902MHz

Support for Johar Sensor tag

With this firmware version, M6e supports Johar sensor tag. This is a passive wireless temperature (humidity) sensor tag. The tag sensor data is acquired by the module as part of ACK response along with EPC. To activate the sensor, slightly modified SELECT command called SELSENSE must be sent i.e., select command with Target value = 7.

It must be noted that the time interval between successive selects (SELECTSENSE and normal SELECT) must be at least 15ms. So to get sensor data, T4 must be set to 15ms and select must be sent with every query followed by read tag ID multiple command with metadata flags.

Steps to get Sensor Data:

1. Create multifilter where
   Select1 Target = 7, Action = 4
   and
   Select2 Target = 2, Action = 4
2. Enable send select with every query
3. Set T4 value to 15ms
4. Perform Sync read
5. Sensor data can be received along with EPC (Last 4 bytes are sensor data)

Support for Fudan Sensor tag

With this firmware version, M6e supports Fudan sensor tag which has a sensor IC optimized for temperature measurement and logging. Users can read sensor tag data with the help of the following 9 custom commands that are added:

1. GEN2_FUDAN_READ_MEM

Used to read data (in bytes) from tag memory.

2. GEN2_FUDAN_WRITE_MEM
   Used to write data (in bytes) to tag memory.

3. GEN2_FUDAN_AUTH
   Used to verify the password (wherever applicable).

4. GEN2_FUDAN_MEASURE
   Used to get temperature. A single measurement process consists of two commands, first command to start the measuring process and the second command to send back the measured temperature (result).

5. GEN2_FUDAN_START_STOP_LOGGING
   Used to start/stop logging process. The tag controls the internal RTC upon receiving this command.

6. GEN2_FUDAN_WRITE_REG
   Used to write data (in bytes) to the tag's RTC register.

7. GEN2_FUDAN_READ_REG
   Used to read data (in bytes) from the tag's RTC register.

8. GEN2_FUDAN_STATE_CHECK
   Used to check the current operational mode (state) of the tag or to refresh the temperature logging process's configuration.

9. GEN2_FUDAN_LOAD_REG
   Used to initialize the value of register when batter power is used. It should be executed first every time whenever the battery is fixed.

Resolved Issues

The following M6e bugs fixed and verified during this release version 1.23.1:

- M6e now starts its internal Read Tag Multiple timer when it starts sending an RF signal rather than when it activates the transmit algorithm. This accommodates unanticipated delays, such as for a channel Dwell Time time-out, which could cause the Read tag multiple command operation to terminate prematurely. (Ref #6356)
- M6e was returning error code(0x100) when extended data length select option (0x20) was used for tag operations. This issue is now fixed and gives success response. (Ref #6363)
- M6e was returning incorrect BrandID at times. This issue is now fixed. (Ref #6115)
- M6e hops to next frequency after 5 inventory rounds (5ms per round i.e., 20ms for 5 rounds) when there are no tags placed in the field. Previously it was hopping to next frequency after 375ms due to which average channel occupancy is going over 400ms which is a violation of FCC standard. (Ref #6410)

Operational Notes

The following restrictions and caveats apply to the features and functionality of firmware version 1.23.1:

- M6e gives “No tag found” error sometimes when user first tries to write EPC > 448 bits and then tries to write EPC < 448 bits. (Ref #6361)
- In order to improve sensitivity, few changes were made in module FW as a result of which the power consumption of M6e module has increased (typically ranges from 7.5W to 9.5W at 31.5 dBm RF output levels, depending on the region of operation and GEN2 settings).

ThingMagic M6e Firmware v1.23.0 Release Notes

This release note describes the features and characteristics of ThingMagic M6e module firmware (FW) version 1.23.0. 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 1.31.2.

The release notes for firmware version 1.21.2, 1.21.1 and 1.19.0 are appended to these release notes for reference.

Topics covered in this release notes are:

For full information about the ThingMagic M6e product, please consult the "M6e User Manual", which can be found at:

https://www.jadaktech.com/documentation/rfid/mercury6e-modules/

M6e module is modified with few hardware changes due to component obsolescence as well as improvements for ESD immunity enhancements and better protection for antenna connectors to prevent the connectors from being detached from the PCBA.

All features implemented in this release support the modified M6e module. Following features are not supported on previous revisions of M6e module, but the current M6e FW v1.23.0 is fully backwards compatible with previous versions of the module hardware.

- ***Japan and Israel regions Support (restricted to –JIC SKU in previous version)***
- ***OPEN Extended region support***

New Features of v1.23.0

New features of version 1.23.0 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 the 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.

Refer Mercury API v1.31.2 release notes for API commands to work with this functionality.

Support for Initial Q Setting

As per the RAIN Gen2 protocol, one of the implications suggests that if multiple tags responded in the same inventory response slot, each one was extremely unlikely to pick the same number from 0 to 65535 so that when the reader acknowledges one tag’s response with that number, the “losing” tags would realize their failure and try again the very 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 0.19%) of two tags selecting the same random number. This means that it is highly likely that a tag will erroneously believe that it responded correctly to an inventory round 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 cannot be tolerated. Min and Max values of "Initial Q" are 2 to 10.

We have made the following changes to reduce, but not 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 “2” 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.)

Refer Mercury API v1.31.2 release notes for API commands to work with this functionality.

Support for Complex Selects

Traditionally, Select has been used for one purpose: to select among a population of tag or select all but a population of tags based on a desired (or undesired) 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.

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 (how to set, reset, flip, or leave alone flags) when the tag memory value either matches the criteria or do not match the criteria.

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

Optional inclusion of target and action settings are:

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

Actions are specified using the shorthand enumerations in the table above.
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 criterion to define which tags should not respond rather than those that would. 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 (Ref #5394).

Sensor tag chips may have temperature sensor, humidity sensor, gas sensor, resistance sensor, capacitance sensor, etc. They are differentiated by address pointers. So, the first Select is used to identify and “wake up” the sensor. If there is only one tag in 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 to read sensor data from, an additional Select (or Selects) are needed.

The maximum number of selects that will be allowed are 3. This number of Selects would be required 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.

Refer Mercury API v1.31.2 release notes for API commands to work with this functionality.

Support for NXP’s UCODE8 Brand Identifier

This feature will allow brand owners to implement a product originality check for products tagged with NXP UCODE8-based Tags. Customers of NXP are granted a dedicated unique 16-bit brand identifier which is programmed during the manufacturing process by NXP and is unalterable in the field (Ref #5658).

A Select command on address bit 204h (with mask length = 1 and mask = 0x80) in the EPC memory bank will trigger the UCODE8/8m tag to respond in the inventory round with PC + EPC + Brand Identifier + CRC as if it were one long EPC. To prevent cloning, the Brand Identifier is scrambled. The module FW automatically descrambles the Brand Identifier. The resultant

decoded Brand Identifier will be sent as part of tag metadata if Brand Identifier metadata flag is enabled. At the host, the Brand identifier can be received, and the originality check completed.

Default value of the UCODE8/8m tag's Brand Identifier is "AAAAh". Customer specific Brand Identifiers can be requested from NXP by ordering a dedicated product code.

If Select is not used, then this field will be skipped, and no Brand Identifier information will be retuned even if the Brand Identifier metadata flag is enabled.

In the API, new metadata flag named "**BRAND_IDENTIFIER(0x1000)**" has been added in TagMetadata enumeration to support the functionality of Brand identifier feature. Refer "read" sample codelet in the MercuryAPI SDK to test this functionality.

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. Refer Mercury API v1.31.2 release notes for API commands or refer "authenticate" sample codelet in the MercuryAPI SDK to test this functionality.

Embedded TagOp Support for Read-After-Write Option

Embedded tag operation support has been added in this FW release for read-after-write option in both EPC write and Data write operations. Earlier firmware releases only supported this function for standalone tag operations. (Both functions are performed without dropping the RF signal, as is required for some sensor tags which use the write operation to trigger sensor value updating, which is subsequently read.)

User can achieve this functionality through two operations:
1) Gen2.WriteData + Gen2.ReadData
2) Gen2.WriteTag + Gen2.ReadData

For executing embedded operation:
1) Create a TagOpList.

2) Add either of the above-mentioned operations to the list
3) Set a simple readplan by pass the tagoplist as the tagop
4) Execute r.read(5000) to execute the embedded operation.

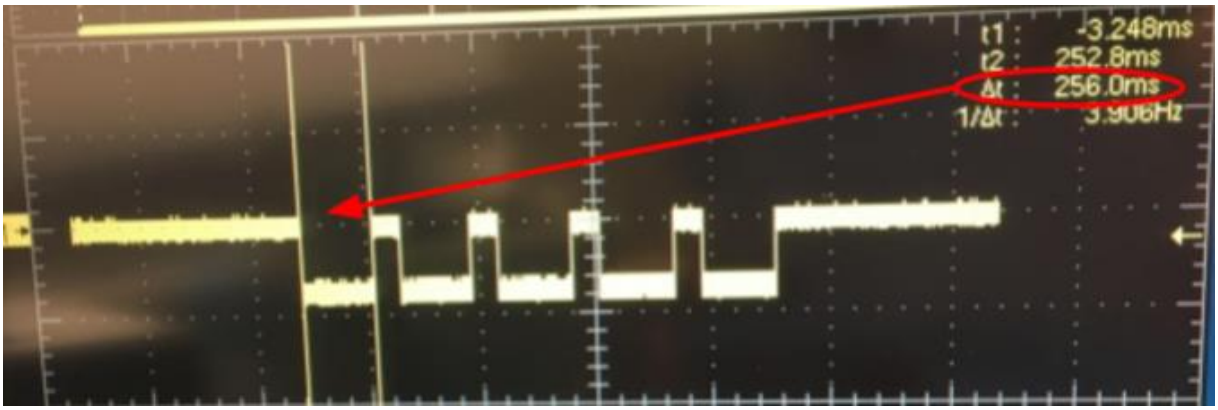
Refer Mercury API v1.31.2 release notes for API commands or refer “writeTag” sample codelet in the MercuryAPI SDK to test this functionality.

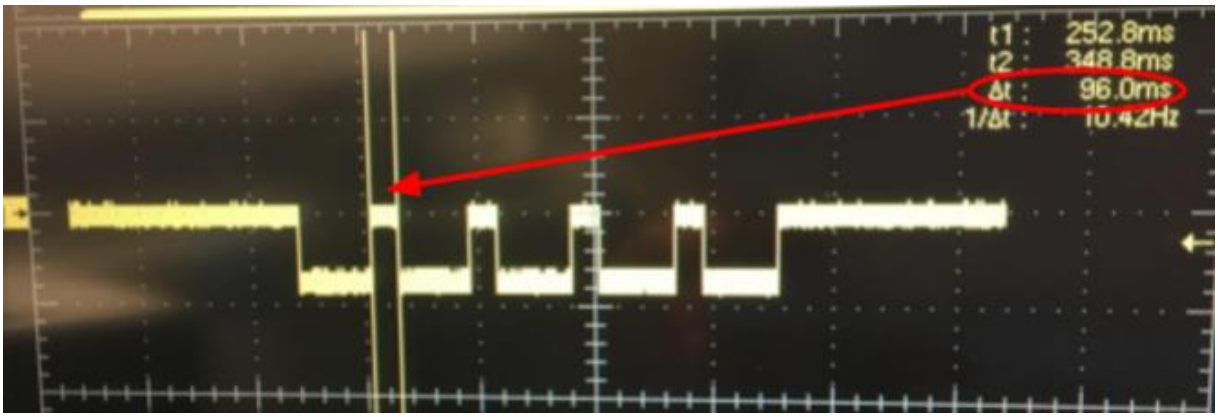
Support for Configurable CW/PRBS ON/OFF Time (Regulatory Test Features)

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.)

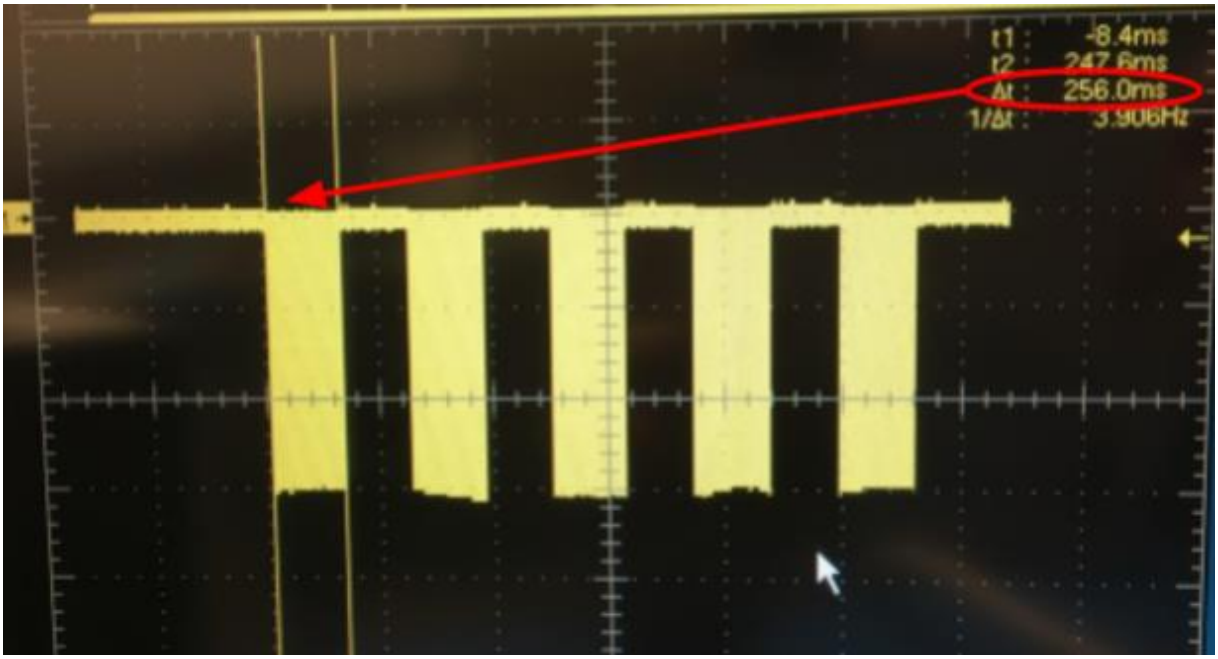
Now support has been added for much more flexibility for both CW and PRBS signal generation. Either can be enabled or disabled for an arbitrary amount of time and, while enabled, they can operate on an on-off duty cycle.

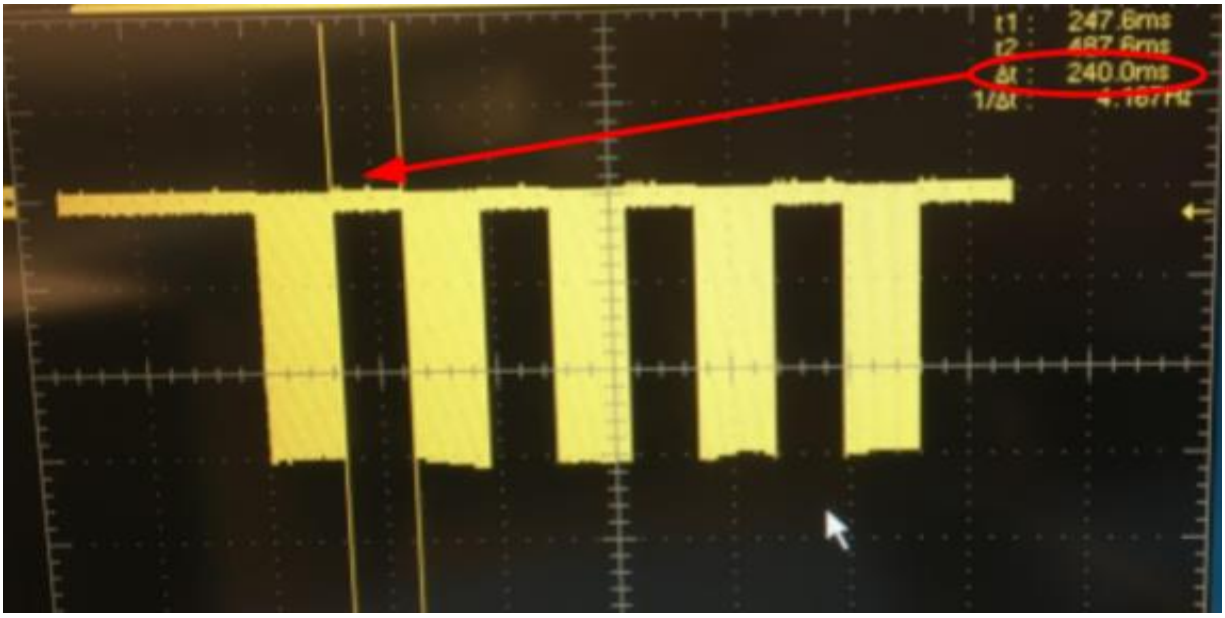
This scope capture of signal amplitude shows CW signal with 256ms RF ON time and 96ms RF OFF time





This scope capture of signal amplitude shows 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 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 sent when the module is in this mode, so it must be polled.)

The 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 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

Refer Mercury API v1.31.2 release notes for API commands or refer “RegulatoryTesting” sample codelet in the MercuryAPI SDK to test this functionality.

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 is defined, along with the read time.

The default 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. Weighted per-antenna reading is now executed as an asynchronous (continuous) read rather than a series of timed, synchronous, reads.

Refer Mercury API v1.31.2 release notes for API commands to work with this functionality.

Support for 5 new regions

Support for Vietnam, Thailand, Argentina, Hong Kong and Bangladesh regions has been added for M6e module. So, 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,**

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

Country	Vietnam	Thailand	Argentina	Hong Kong	Bangladesh

	(VN)	(TH)	(AR)	(HK)	(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 867250	921250 923250 922250 924250 921750 923750 920750 922750	921250 923750 919250 916750 926250 922750 918250 915750 925250 924750 920250 917750 927250 921750 923250 919750 916250 926750 922250 918750 915250 925750 924250 920750 917250	866500 867100 865900 866300 867700 865500 866100 867900 865300 866700 867300 865100 866900 867500 865700	926000 926200 925800 926600 925400 926400 925600

Refer Mercury API v1.31.2 release notes for API commands to work with this functionality.

Support for JP and IS regions for new M6e module

New M6e hardware has filters to support JP and IS regions. So, support is added in module FW to support these 2 regions.

Region Name	**JP**	**IS**
Country or Region	Japan	Israel
Serial Interface Region Code	0x05	0x0F
Lowest Freq Permitted	916800	915000
Highest Freq Permitted	920800	917000
Smallest Step Size	100	250
Maximum Dwell Time	4 sec	0.4 sec
Max RF Power	+30 dBm	+31.5 dBm
LBT Level	-74 dBm	None
Lowest Chan in Hop Table	916800	915500
Highest Chan in Hop Table	920800	915500
Chan 1	920800	915500
Chan 2	919200	915500
Chan 3	920600	915500
Chan 4	916800	915500
Chan 5	918000	
Chan 6	920400	

Refer Mercury API v1.31.2 release notes for API commands to work with this functionality.

Support for new extended OPEN region for new M6e module

New OPEN region support with extended frequency range 840-960MHz has been added in module FW for new M6e hardware. User needs to install license to work with the extended OPEN region. Contact ThingMagic support (RFID-support@jadaktech.com) for further details.

Region Name	**OPEN_EXTENDED**
Serial Interface Region Code	0xFE (Dec 254)

Supported Modules	New M6e Hardware (license key is needed)
Lowest Freq Permitted	840000
Highest Freq Permitted	960000
Smallest Step Size	25
Maximum Dwell Time	0.4 sec
Max RF Power	+31.5 dBm
LBT Level	Optional at -96dBm
Lowest Chan in Hop Table	840000
Highest Chan in Hop Table	960000
Number of Channels	25
Chan 1	840000
Chan 2	845000
Chan 3	850000
Chan 4	855000
Chan 5	860000
Chan 6	865000
Chan 7	870000
Chan 8	875000
Chan 9	880000
Chan 10	885000
Chan 11	890000
Chan 12	895000
Chan 13	900000
Chan 14	905000
Chan 15	910000
Chan 16	915000
Chan 17	920000
Chan 18	925000
Chan 19	930000
Chan 20	935000
Chan 21	940000
Chan 22	945000
Chan 23	950000
Chan 24	955000
Chan 25	960000

Refer Mercury API v1.31.2 release notes for API commands to work with this functionality.

Operational Notes of v1.23.0

The following restrictions and caveats apply to the features and functionality of firmware v1.23.0:

- 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 during long run. Usually it goes ahead of system time. (Ref #5925).

New Features of v1.21.2

New features of version 1.21.2 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 for diagnostics. 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.

NXP UCODE7 Configuration Support

Prior to UCODE 7, NXP supported a set of custom commands that could change the configuration word values. Unfortunately, these commands that were developed for the G2i line of tags do not work for the UCODE 7 tags. A new custom command has been implemented in M6e FW to change NXP UCODE7 configuration word for M6e and M6e modules (Nano does not support custom commands).

UCODE7 no longer support ChangeConfig commands. An alternative way to change the configuration word for UCODE7 tags has been developed.

UCODE7 configuration word contains 2 different type of bits:

- Action bits: meant to trigger a feature upon a SELECT command on the related bit:
  Parallel encoding (at address 0x202)
  Tag Power indicator (at address 0x204)

- Permanent bits: permanently stored bits in the memory
  Max. Backscatter Strength (at address 0x209)
  PSF Alarm bit (at address 0x20F)

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 when tags are read so that the active setting under which the tag was read is 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 M6e 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

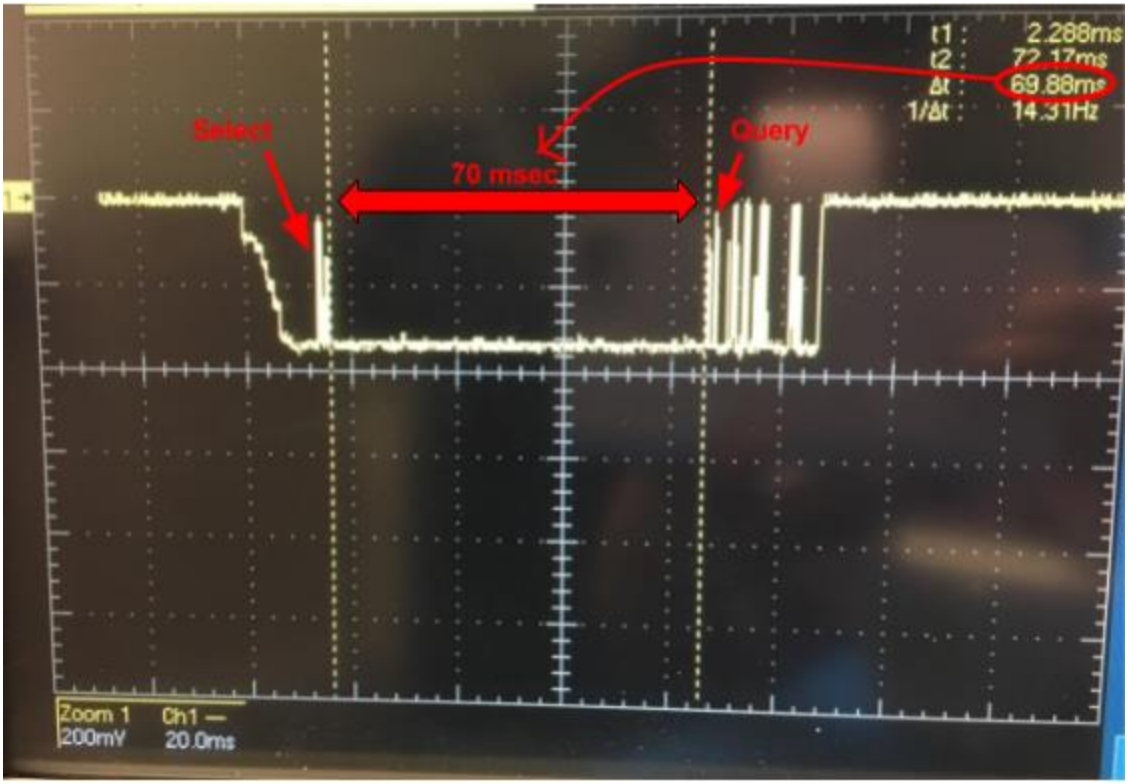
In previous firmware releases, GEN2V2 operations that supported the NXP UCODE DNA tag were only available as stand-alone, single tag functions. In this firmware release, support for embedded tag operations has been added for both NXP UCODE DNAtag and the NMV2D tag. This allows for high speed secure reading 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 M6eseconds.

T4 value is a 4-byte in length and specified in M6eseconds. 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.



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 AsyncOnTime

In previous releases, when reading continuously, the reader would always return to antenna one (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.

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

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 (Quantization)	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 (Quantization)	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 in Open region

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 existing regions because such a small step size will result in lower channel frequency stability. (This setting 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 of v1.21.2

The following M6e bugs fixed and verified during this release version 1.21.2:

- Changes made between firmware versions 1.19.0 and 1.21.1 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 1.21.2 corrects this and the actual read times are now within 4 msec of that requested. (Ref# 5551)

- Gen2.IDS.SL900A.GetLogState command now returning the proper (non-zero) last byte data of log status to the user. (In previous releases, the last byte of the flag response is always 0x00). (Ref# 5514)

- Now M6e module will validates “EPC length filter” when user performs tag operations(In previous releases, module wasn’t validating the filter when we perform gen2 tag operations, module was performing the tag operations irrespective of EPC length we specified in filter) (Ref# 5480)

- The multiprotocol (Authenticate_OBU + OBU_Auth_Full_Pass and Authenticate_OBU + OBU_Auth_Full_Pass1) operations now works fine(In previous releases, The multiprotocol changes was made only for "Authenticate_OBU (PA) + OBU_Auth_ID (G0)" and not for others options like "Authenticate_OBU + OBU_Auth_Full_Pass" and "Authenticate_OBU + OBU_Auth_Full_Pass1" are reporting only tags starting with 0x003). (Ref# 5120).

- SL900A tag “Get Battery Level” command now works correctly. In previous releases, the module returned the same response (0x00) for both “GetBatteryLevel” with and without retrigger commands to the tag. (Ref #5518).

- Trigger read with M6e-JIC module now works correctly. With previous releases, you might experience “connection lost” after turning on the specified GPI pin when we initiate trigger read on M6e-JIC module with the JP region. (Ref #5218)

- Gen2 specific metadata is no longer returned when non-Gen2 protocols are active (Ref# 5415)

- 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)

- GEN2 tag kill function is now more reliable. In the previous firmware version, we found that some tags that had been reported as being killed could 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 of v1.21.2

The following restrictions and caveats apply to the features and functionality of firmware version 1.21.2

- GEN2 Reconnection to the module may not possible if the host PC is restarted while auto read is in progress. To reconnect to the module, the user must either reboot the module or unplug and plug back in the USB cable. (Ref #5583).

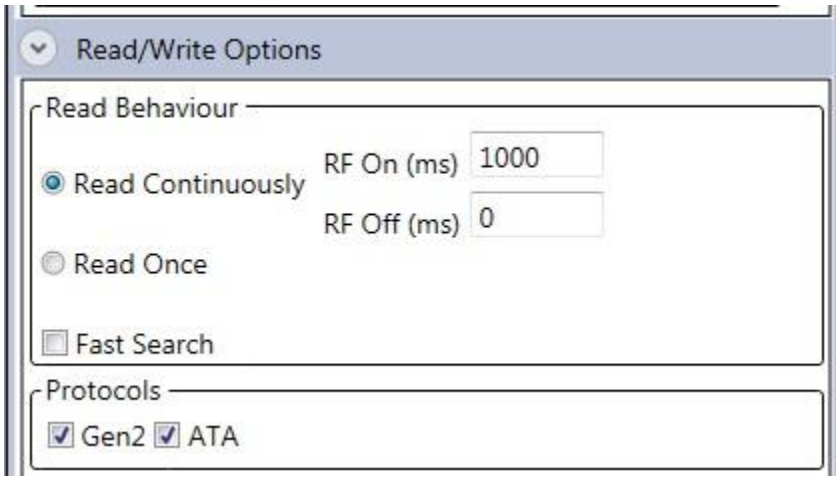
New Features of v1.21.1

New features of version 1.21.1 include the following:

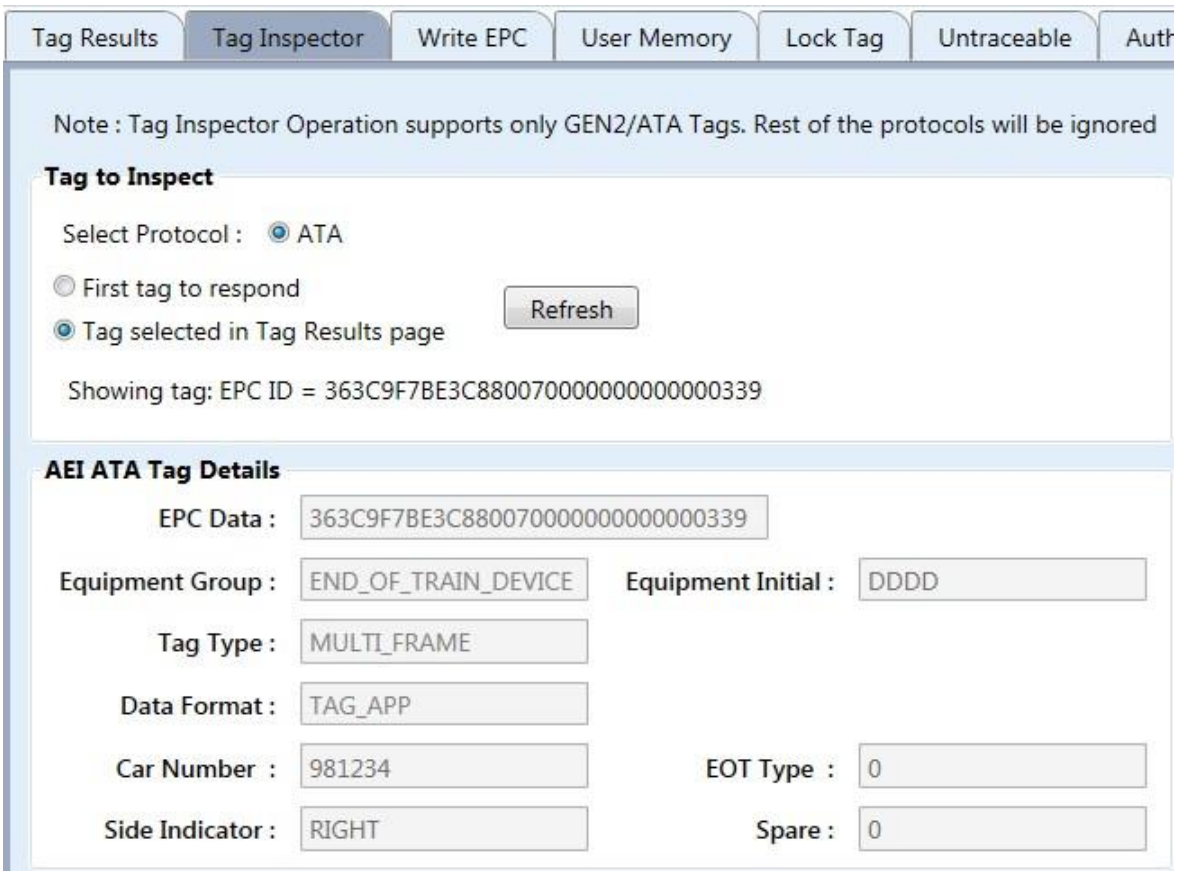
Support for AEI ATA protocol with Stop Trigger Read Plan

The AEI ATA protocol is supported on the M6e module with an optional license key.The ATA protocol is supported under a Stop Trigger Read Plan so that results can be provided continuously instead of at the end of a read cycle.

Universal Reader Assistant now supports readers which have been licensed to read the AEI ATA or IP-X protocols.



The “Tag Inspector” tab has been revised so it can interpret the information in AEI ATA tags per the AAR S-918 encoding standard.



1:8 Multiplexing Up To 32 Ports

The M6e module now has the ability to use 3 GPO lines to control an external multiplexer which expands one RF port to 8 RF ports.

The following table provides the list of all possible “logical” antenna ports and how the selection of that logical antenna port affects the GPO line state and which physical antenna is active. If you are using fewer than 3 GPO lines to control the module or using fewer than the 4 physical ports, do not include the logical ports in your port list that do not correspond to desired GPO and antenna configurations. When a port is not defined as a GPO control, you can assume it is low with respect to the chart.

Note

Use of fewer than the maximum number of ports and GPO lines will result in gaps in the logical antenna list. This is desirable because assigning additional GPO lines to multiplexer control will not change the port assignments already established with fewer lines. If the non-contiguous numbering is undesirable, you have the option to rename any logical port to a name or number you choose.

Mapping of Logical Antenna Numbers to GPO lines and RF ports

Logical Ant #	GPO 3	GPO 1	GPO 2	Physical Ant #
1	Low	Low	Low	1
2	Low	Low	Low	2
3	Low	Low	Low	3
4	Low	Low	Low	4
5	Low	Low	High	1
6	Low	Low	High	2
7	Low	Low	High	3
8	Low	Low	High	4
9	Low	High	Low	1
10	Low	High	Low	2
11	Low	High	Low	3
12	Low	High	Low	4
13	Low	High	High	1
14	Low	High	High	2
15	Low	High	High	3
16	Low	High	High	4
17	High	Low	Low	1
18	High	Low	Low	2
19	High	Low	Low	3
20	High	Low	Low	4
21	High	Low	High	1
22	High	Low	High	2
23	High	Low	High	3
24	High	Low	High	4
25	High	High	Low	1

26	High	High	Low	2
27	High	High	Low	3
28	High	High	Low	4
29	High	High	High	1
30	High	High	High	2
31	High	High	High	3
32	High	High	High	4

The additional GPO line is configured just like the first two, using the /reader/antenna/portSwitchGpos parameter. Once GPO lines are configured to act as multiplexer controls, you may use the virtual port numbers as if they were physical ports on the reader.

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.

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.

Enhanced Licensing Tool Support

The M6e module supports protocols and features that are activated by installation of a license key. Previous versions of firmware required that a software utility be generated for each module or license purchased, which would read the serial number of the module and only install the license if the module had the correct serial number. A new class of licenses has been defined that incorporates the serial number into the license, so that the module (or reader) firmware can do the checking. This allows us to create license installation utilities that can be used with any module or reader.

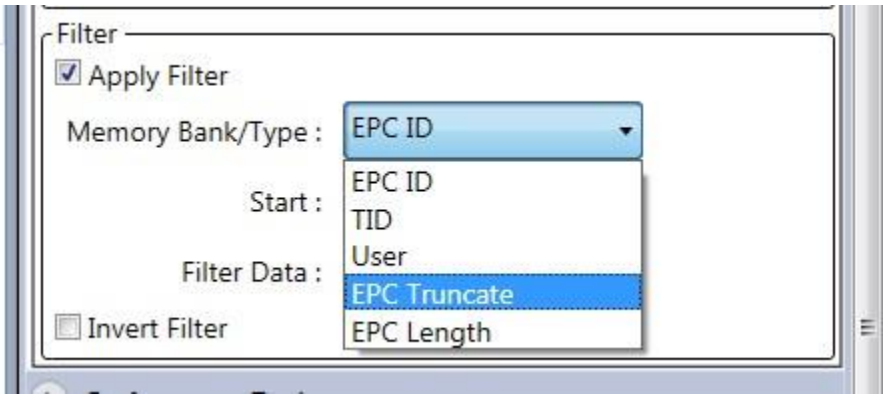
The Universal Reader Assistant Firmware Update panel has been enhanced to be able to install license keys.

The file is in CSV format and contains both the information necessary to generate a license (so you can make sure all the desired features are included) and the license itself. Minimally, it need contain only a list of serial number(s) and license(s).

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 Determine Module’s Ability to Support Custom Gen2 Capabilities

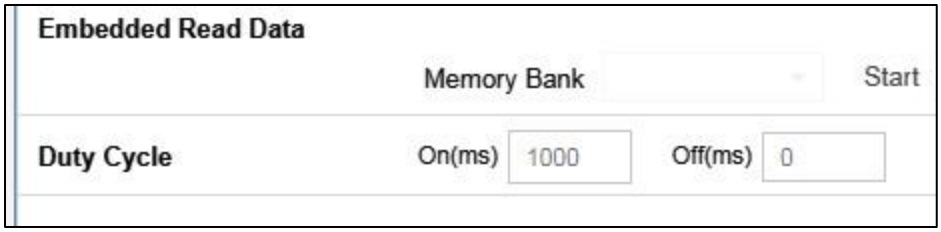
The host has the ability to “get” many settings when it first connects to a module in order to determine whether the settings are as desired or if they need to be changed. Now, an additional “get” is allowed to determine if any optional Gen2 extensions have been enabled for the module. (Currently there is only one custom extension offered - IAV Denatran support.)

The parameter to get protocol extensions is /reader/Gen2/ProtocolExtension.

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.



Ability to Change Settings During Continuous Reading

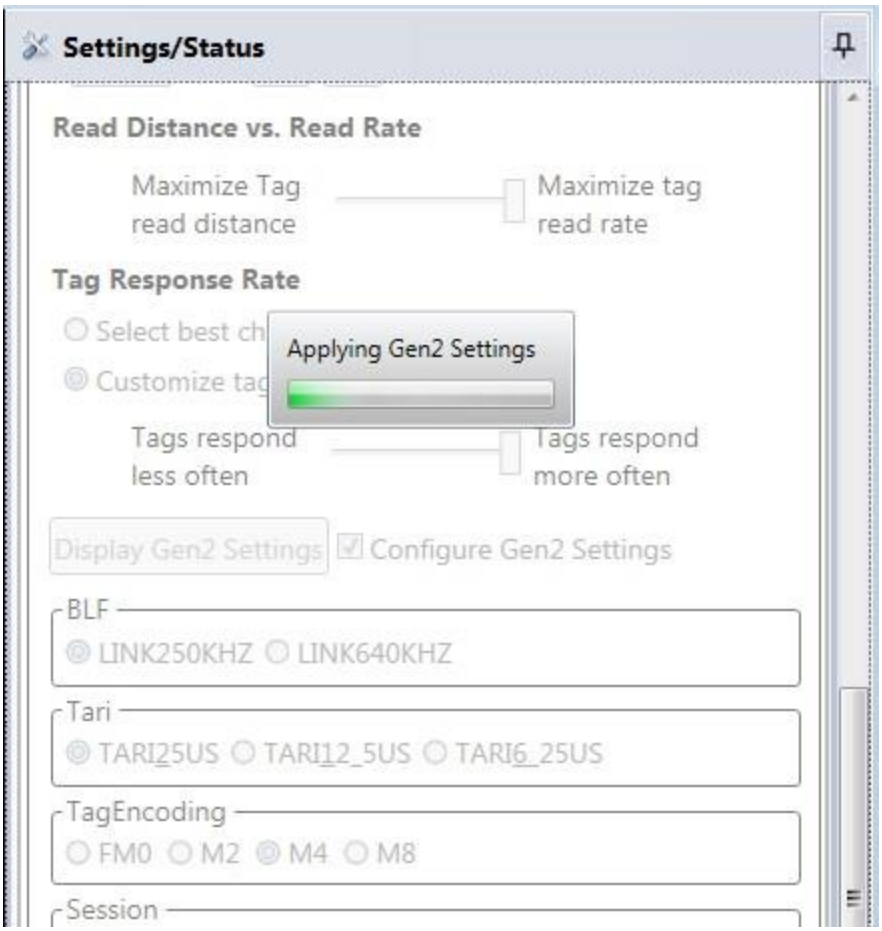
In previous firmware releases, continuous reading had to be halted in order to change any settings. Now, a subset of available settings can be changed while the reader is actively reading. This allows the host to optimize settings on-the-fly. Settings that are supported during continuous reading are:

- 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.

Serial Number Added to USB Device Descriptor

Adding a serial number to the USB device descriptor allows the host to assign a COM port number which follows the device regardless of which physical USB port it is plugged into.

Denatran Tag Support

The M6w module now supports the Denatran extension to the Gen2 protocol as a licensed feature. Contact ThingMagic support for further details.

Resolved Issues of v1.21.1

- The M6e-PRC module (recently replaced by the M6e-JIC module) had been found to not have the required 20 ppm channel frequency accuracy since firmware release 1.13.1. This has been corrected (Ref# 4756)

- M6e firmware version 1.19.0 did not calculate return loss correctly and displayed extremely high numbers that were clearly incorrect. Firmware releases prior to 1.19.0 were accurate and this accuracy has been restored (4838)

Operational Notes of v1.21.1

- When the Truncation filter is applied, the tags will return data even if the Access Password is not correct. (Ref# 5075)
- The implementation of the command and response logic for Denatran tags is incomplete. Please contact ThingMagic support for details. (Ref # 5078)

- **No Ability to "Get" Saved Value of Settings**
  The module firmware is able to save many settings in flash memory. As of firmware version 1.21.1, 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

New Features of v1.19.0

New features of version 1.19.0 include the following:

Load/Save Support

Settings can now be saved in the module, creating new default values that are restored when the module reboots. Although primarily intended to be used for autonomous operation, it can also be used to reduce the number of commands that must be sent to the module to restore operation after a reboot

Autonomous Operation Support

A read plan can be saved which allows the module to automatically begin continuously reading, and optionally return data memory values, whenever the module is powered or whenever one of its GPI lines is asserted. An Autonomous Configuration Tool is available to configure the settings and read plan necessary to implement this feature.

Gen2V2 Support

The M6e 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 version of the API and may be demonstrated using code samples and the version of Universal Reader Assistant which is distributed with the API.

Support for Return Loss Measurement

The firmware will now estimate the return loss of individual antenna ports, based on multiple readings at multiple channels within the active region. (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
```

```
Antenna 2 | 4
```

Which indicates a return loss of 30 dB for antenna 1, and 4 dB for antenna 2.

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.

Note:

Unlike the M6e and Nano, which use this function for antenna detection, the M6e still uses a small amount of DC current to detect antennas, as it always has. Use of this original method to determine if an antenna is present and of the return loss to determine if the antenna is tuned to the correct frequency, is the best way of ensuring maximum performance for the channel of operation.

Resolved Issues of v1.19.0

- A bug was fixed where the firmware could not read or write User Memory locations higher than 127 kB (Words). In previous versions of firmware, no error would result, but memory addresses of 128 kB and above would be interpreted as starting back at address “0”, repeating every 128 kB. (Ref# 4702).