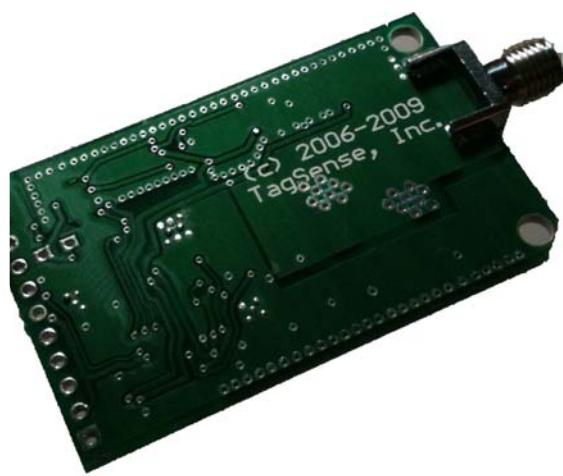




tagsense



TagSense Nano-UHF RFID Reader

Data Sheet v2.1

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

TagSense, Inc.

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



TABLE OF CONTENTS

NANO-UHF RFID READER	4
DOCUMENT VERSION HISTORY	5
NANO-UHF SPECIFICATIONS	6
NANO-UHF CERTIFICATIONS AND SAFETY	7
CERTIFICATIONS:	7
SAFETY:	7
NANO-UHF OPERATION.....	8
INTERFACING WITH THE NANO-UHF.....	9
STARTING WINDOWS DEMO PROGRAM	10
HARDWARE INTERFACE.....	11
NANO-UHF API	12
READER SETTINGS: BINARY STATE VARIABLES OR FLAGS:	12
READER SETTINGS: CONTINUOUS VARIABLES:.....	14
CONFIGURING THE NANO-UHF	15
PROTOCOL IDENTIFIER:.....	15
HEXADECIMAL NOTATION:	16
READER ID CODE:	17
DATA STREAMING:	17
POLLING MODE:.....	17
NOTE ON CHANGING ANTENNAS:	18
VIEWING OR SAVING THE STATE OF THE READER.....	19
SAVING THE CURRENT READER SETTINGS:	19
SAVING THE CURRENT READER SETTINGS:	19
WRITING A NEW ID TO EPC GEN 2 TAG.....	20
WRITING TO A SINGLE TAG:	20



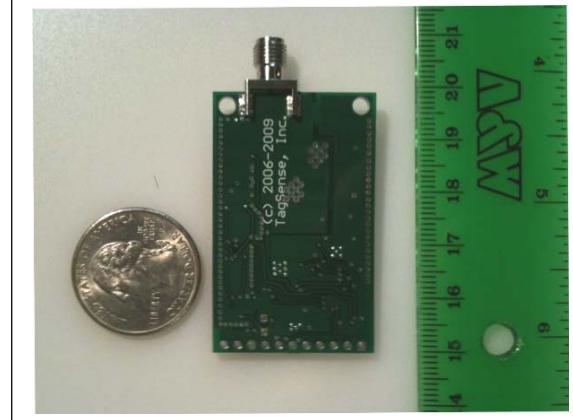
WRITING TO A SINGLE TAG WHEN MANY TAGS ARE PRESENT:	21
READER ACKNOWLEDGEMENT AND ERROR CODES:	21
EPC GEN 2 TAG MEMORY STRUCTURE	22
MEMORY BANK 0: RESERVED MEMORY:	22
MEMORY BANK 1: EPC MEMORY:	22
MEMORY BANK 2: PROTOCOL CONTROL TID:	22
MEMORY BANK 3: USER MEMORY:	23
READING AND WRITING TAG MEMORY.....	24
READ USER MEMORY:	24
WRITING TO USER MEMORY:	25
USING EPC GEN 2 SECURITY FEATURES	26
USING PASSWORDS	26
SETTING THE PASSWORDS IN THE TAG MEMORY	26
<i>Access password:</i>	26
<i>Kill password:</i>	27
<i>Setting and Using the Password Buffer in the Reader Memory</i>	28
LOCKING PARTS OF TAG MEMORY.....	30
USING THE LOCK COMMAND	30
<i>Argument #1: the Lock Mask</i>	31
<i>Argument #2: the Lock Action</i>	32
READER SESSIONS	34
PERSISTENCE	34
USING THE SESSION COMMAND	36
NANO-UHF READER BASIC COMMANDS	37
NANO-UHF READER ADVANCED COMMANDS.....	38



tagsense

NANO-UHF RFID READER

The Nano-UHF reader is a small, low-power, low-cost RFID reader module that is designed for short-range battery-powered mobile embedded applications, such as cell phones, handheld readers, printers, consumer electronics, or smart shelves. An external antenna can be connected via an SMA connector or MMCX connector. A small yet powerful ASCII command set makes this reader useful for many applications.



Unique Features:

- Very low power consumption (<3 mA avg.) at 5V during standby
- Software controlled power level
- Very easy to use command set
- Multi-protocol: EPC Class 1 Gen 1/Gen 2, Read/Write
- Operates at both the European UHF frequency band (865-868 MHz) and North American UHF frequency band (902-928 MHz)
- Programmable Reader ID for networking many readers together
- Onboard EEPROM memory allows custom user configuration to be saved in memory so reader will automatically boot-up in a user-defined mode.
- Programmable number of time slots for anti-collision
- Support for external sensors (door lock, motion sensor, etc.)
- Full support of all EPC Gen 2 data fields and security features
- Support of user memory for multiple tag vendors (NXP, Impinj, STM)

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

TagSense, Inc.

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



DOCUMENT VERSION HISTORY

Version	Date	By	Changes
2.1	4/20/10	RRF	Added certification and safety section
2.02	2/12/10	RRF	Corrected some specs
2.01	2/11/10	JAR	Updated Images and Specifications
1.9	4/23/07	RMR	Added new Protocol Control printing command
1.8	4/5/07	RRF	Incorporated minor corrections to firmware
1.7	4/3/07	RMR	Updated Security & session commands
1.6	3/5/07	RMR	Updated general memory read/write commands
1.5	11/09/06	RRF	Added user memory, modified write command
1.4	9/20/06	RMR	Added power, slots commands, lock-write
1.3	7/25/06	RRF	updated physical specs
1.2	5/23/06	RRF	updated software instructions
1.1	5/09/06	RRF	added EPC write, error codes
1.0	3/31/06	RMR	Initial Draft



NANO-UHF SPECIFICATIONS

Electrical Specifications

Operating Voltage:

- 5.3 – 9 V (unregulated input)
- 3V and 5 V (regulated)

Current consumption

- 100 mA when transmitting
- <3 mA when idle

Antenna Connection:

- RP-SMA Jack (MMCX optional)
- 50 Ohms

RF Power Output:

- 50 milliwatts (+17dBm) at connector

Physical Specifications

Board Size:

- Width = 30mm (1.375 inch)
- Length = 48mm (1.875 inch)
- Height = 7mm (0.27 inch)

Environment (storage):

- 0C – 85C (32F – 185F)

Performance Specifications

Supported Protocols:

- EPC Class 1 Gen1
- EPC Class 1 Gen 2
- Read and Write

Reading range (avg):

- 80 cm (30 inches) using standard 6dbi antenna
- 40 cm (15 inches) using small patch antenna (ANT-LP1)

Interface Specifications

Host Interface:

- 4-pin header
 - (PWR, GND, RX,TX)
- RS-232 TTL-level

Sensor Input:

- 3-pin header (V+, GND, Vin)
- 8-bit A/D



tagsense

NANO-UHF CERTIFICATIONS AND SAFETY

Certifications:

FCC full modular certification
FCC ID = XQ7TAGSENSE-NANO

The Nano-UHF module can be embedded in another product (host) without requiring additional testing and certification, provided that the host is used in a *mobile configuration* as defined by the FCC and the proper safety rules (see below) are followed.



FCC ID:
XQ7TAGSENSE-NANO

Safety:

When the Nano-UHF is being used with the 8dBi antenna, the radiating side of the antenna must be kept at least 20 centimeters away from the human body (e.g. head, hands, feet).

TagSense, Inc.

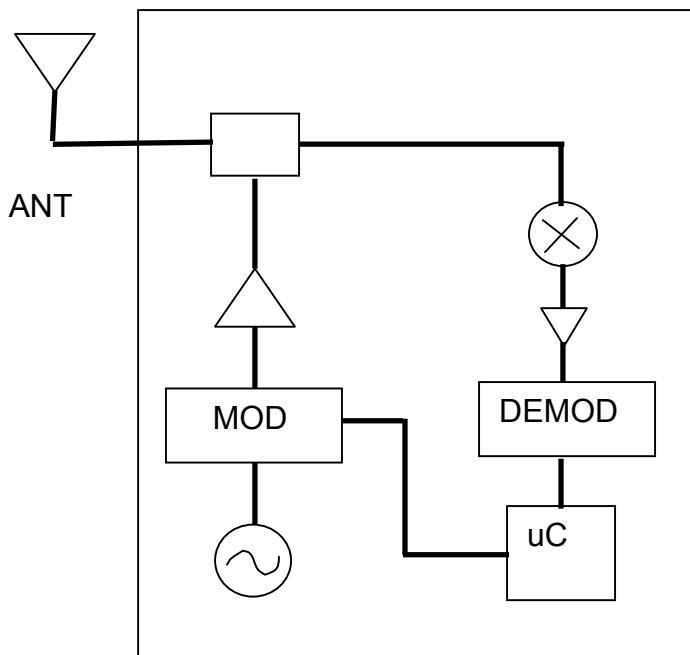
1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



tagsense

NANO-UHF OPERATION



The Nano-UHF is a single-antenna reader module. The backscatter signal is demodulated to baseband through a standard I/Q mixer stage and is filtered and amplified. The baseband signal is then sampled and processed by a nanocontroller.

The transmit RF power of the Nano-UHF is 50 mW, which is sufficient for low-power short range applications.

The Nano-UHF hardware supports software control for output power level.



tagsense

INTERFACING WITH THE NANO-UHF

The Nano-UHF is designed to interface with another circuit board and communicate using a simple 3-wire serial interface. However, for testing purposes, it is possible to connect the Nano-UHF to a computer using a serial RS-232 adapter (available separately from TagSense upon request). In this case, the TagSense Reader Control Panel software can be used.



Windows Interface

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

TagSense, Inc.

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.

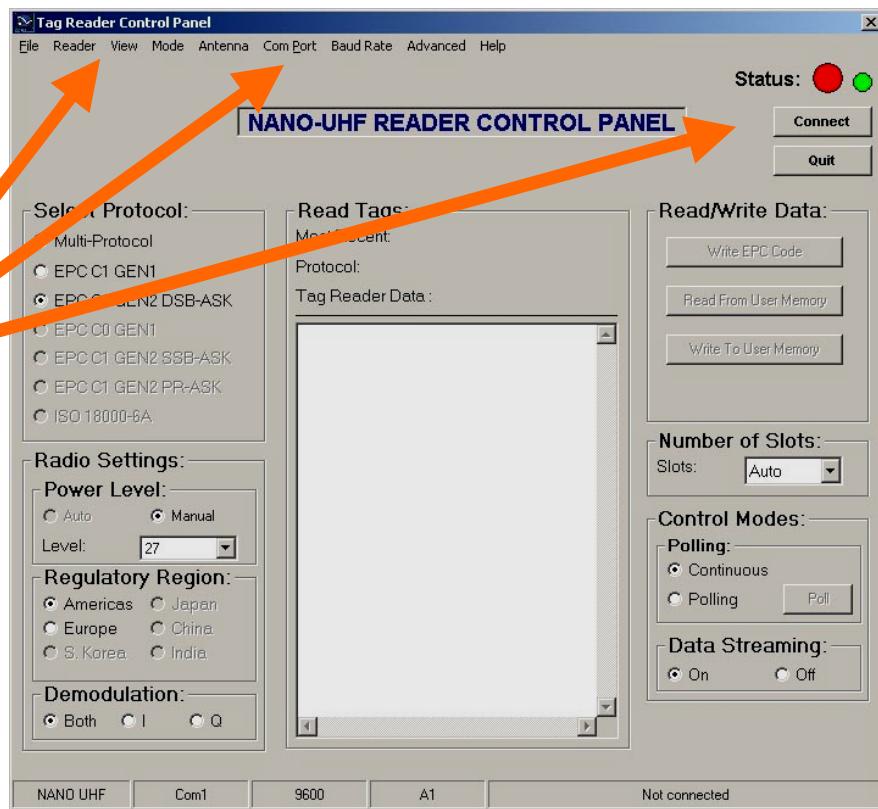


tagsense

STARTING WINDOWS DEMO PROGRAM

After installing the demo program, you can connect the reader to the computer using a serial cable (standard DB9 connectors). Please make sure that you have an antenna connected to the reader.

The program supplied with the kit is able to interface to all the RFID readers made by TagSense. You should open the program (TagSense reader control panel), then go to the **Reader** menu and select “**NANO-UHF**”. Then go to the **ComPort** menu and select the proper COM port where you connected the reader. You should then verify that the **BaudRate** setting is correct (115200), then you can click the **CONNECT** button. After you run the program for the first time, the software settings will be saved automatically so you should not need to reconfigure the control program unless you switch to another reader product.



TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

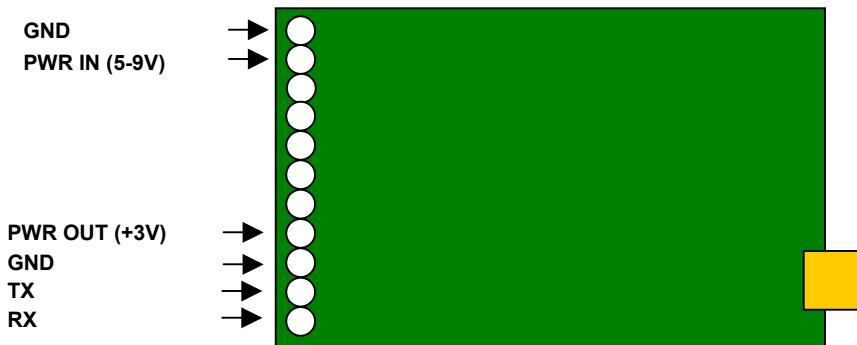
<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



tagsense

HARDWARE INTERFACE

The Nano-UHF connects to the host via a simple 3-pin header for data and 2-pin header for power. The 3 data lines are: Ground, Data Transmit (TX) and Data receive (RX). The data header also contains an additional pin which outputs 3 Volts. This pin can be used to provide power to another sensor or auxiliary board (such as the TagSense ZT-Link Radio module). The power header pins supplying power to the Nano-UHF is 5 Volts (and must be low-noise and regulated). A diagram of the pins is given below.



Pin-out for Nano-UHF



Photograph of Covered Nano-UHF module

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



tagsense

NANO-UHF API

For the purpose of this datasheet, it is assumed that the TagSense *reader* module is being controlled by some *host* device (such as a PC or external circuit board) and the reader is in turn communicating with one or more passive RFID *tags*.



The command set used by the TagSense RFID readers was designed to be simple to use while still preserving efficiency. The commands consist of a single human-readable character plus an optional argument. The command set was also designed to allow users to easily interface the TagSense reader module to other embedded electronics or computer software. The human-readable character commands allow a user to easily test the reader manually by connecting the reader to a PC computer, using any simple terminal program, such as HyperTerminal.

The following conventions are used throughout the protocol:

Reader Settings: Binary State Variables or Flags:

Certain settings on the reader are binary, which means they have only 2 values, ON or OFF. For any parameter that is ON/OFF, such as enabling or disabling the printing to the screen, we defined the single-character command so that the UPPER-CASE letter will turn OFF the parameter, and the LOWER-CASE character will turn ON the parameter. In order to read the value of these parameters or "program flags," the user can use the command "s" at any time and the reader will return a list showing the value of all parameters.

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



tagsense

NANO-UHF API

How to Communicate with the TagSense Reader Module:

If you want to communicate directly with our reader without using the Windows demo software, you can use any program that allows you to communicate with a COM port, such as a terminal program or Windows program.

Once connected to a host computer, the Nano-UHF appears as a COM port and can be easily integrated into any program that can read/write to a COM port (for example JAVA, Visual C++, Visual Basic, etc.). Examples of commonly used terminal programs are Hyperterminal, Secure CRT, or Putty. The Com port settings are: 9600 baud, 8 data bits, 1 stop bit, and no flow control.

The Nano-UHF is controlled by sending it command strings based on the API. A command string is defined as a single ASCII command character from the above table, followed by the argument (if any), and terminated by a carriage return <CR> = CHR\$(13). Several command string examples are:

P0<CR>

P

k<CR>

k

Note that as the characters of the command string are typed, they are immediately echoed back by the Nano-UHF. However, the Nano-UHF does not process the command string (and does not send any acknowledgement) until it receives a carriage return.

Upon receiving the carriage return, the Nano-UHF will output a line feed LF = CHR\$(10), and then begin processing the command. If the command is valid and well formed, the Nano-UHF will output an acknowledgement (with two exceptions, noted later). The acknowledgement consists of the command character (without the argument), followed by a carriage return CHR\$(13) and line feed CHR\$(10).

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



tagsense

NANO-UHF API

Reader Settings: Continuous Variables:

For any reader setting which can be a range of values, such as the power level, or the frequency channel, these parameters are set by using the UPPER-CASE letter command, followed by the argument. In order to query this value, the user can send the LOWER-CASE letter, and the reader will return the current value of that setting (variable). For example, to set the reader power level to 22, the user would send the command: "J22". Then later, if the user wants to know the power level used by the reader, the user can send the following command: "j", and the reader will reply with the answer "22".

By following these conventions, a user can easily configure an RFID reader and also to easily query the state of a reader.

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



tagsense

CONFIGURING THE NANO-UHF

Data output:

The data output from the reader has the following byte format:

CHR\$(42) + 12 byte EPC ID (hex characters) + CHR\$(13) + CHR\$(10)

When printed to the screen, the output appears as the following:

***aebf44f68702d72bc96a23e8**

Protocol Identifier:

The reader module can be configured to also include information about the tag. If the reader is able to support different protocols and modulation types, then the reader can also supply some information about the protocol and modulation type of the tag. The data format used by the reader is controlled by the commands listed previously.

If you want to configure the reader to automatically detect the type of tag and report it back, send the following commands to the reader:

<i>P2</i>	put the reader in EPC C1G2 ASK mode
<i>k</i>	set the reader to continually scan for tags
<i>m</i>	enable printing of tag protocol identifier

If you bring an EPC Gen2 tag into the reader's field the reader will print:

***2 33b2ddd901402805000060c6** (12 byte tag ID)

The number "2" is the protocol identifier. In this case, the reader prints a number "2" to indicate that the tag uses EPC Gen2. The remaining number is the Tag ID.

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



tagsense

CONFIGURING THE NANO-UHF

Hexadecimal Notation:

It should be noted that the 24 character tag ID is represented as standard hexadecimal byte notation, where each byte represented as 2 ASC characters. Since this is hexadecimal, the characters 'a' though 'f' are allowed in addition to the standard numbers 0-9. Therefore the following ID strings are also valid EPC ID numbers: **012abc67890abc4567890abc, 012abc67890abc45fdebb123.**

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

TagSense, Inc.

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



CONFIGURING THE NANO-UHF

Reader ID Code:

When many readers are networked together, via a wireless link or common data bus, the remote controller or data server will receive data from multiple readers and will not be able to know which reader detected the tag. For this case, TagSense provides a Reader ID code for all its reader. This ID code field can be turned on using the command "i". Once turned on, the reader will send its ID code along with the tag ID in the same data packet. This data field is disabled by sending the capital i ("I"). Remember that all commands must be terminated by a carriage return, CHR\$(13).

Data Streaming:

When the reader is not in polling mode, the default state is data streaming = ON. This means that the reader will output data continuously as long as tags are in range of the antenna. However, data streaming can be turned OFF by sending the reader a "D" command. When data streaming is OFF, the reader will output the tag ID only one time. If a tag is held in range of the reader antenna, the ID of the tag will be sent by the reader only one time; to see the tag ID again, the tag must be removed from the reader field long enough to let the tag discharge, and then brought back into the electromagnetic field. Note: EPC Gen 2 tags contain internal energy storage (capacitor) and thus can remain active for as much as 1-2 seconds after removing it from the RF field.

Polling Mode:

The default state of the reader is to transmit *continuously* and scan for tags. However, the reader can be operated under polling mode by issuing the reader a "K" command. *This command will cause the reader output power to turn OFF.*

In polling mode, the reader output power will remain OFF until the reader receives a polling command ("Z"). Upon receiving the polling command, the reader will turn ON briefly, scan for tags, output the inventory list of tags that are present, and then turn the RF field OFF once again.



tagSense

CONFIGURING THE NANO-UHF

Note on Changing Antennas:

Although the reader will not be damaged if the reader temporarily disconnects the antenna during operation, prolonged operation (hours) of the reader without an antenna is not recommended, especially for higher power levels. If the user wished to change antennas without turning off the reader, it is recommended to turn off the RF field while doing so. This can be accomplished using the polling command "K" described previously. The field can then be turned on again using the lower-case "k" command.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

TagSense, Inc.

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



tagsense

VIEWING OR SAVING THE STATE OF THE READER

Saving the current reader settings:

Since there are many settings on the reader that can be set by the user, TagSense provides a simple way to view all the settings of the reader in a single command. This command is the "s" command.

After sending an "s" command, the reader will return the firmware version number (such as "v2.12"), followed by a blank line, then followed by a list all of the settings in alphabetical order. The output of the state command is terminated by an exclamation point, "!".

Below is a shortened sample data output that would be returned after sending the (lower-case) "s" command:

```
v2.04      firmware version

k1      continuous autoscan – 0: disabled, 1: enabled
l0      printing of reader ID
m1      printing of protocol type
n4      number of slots is set to 2^4=16
!
```

Saving the current reader settings:

TagSense also provides a simple command for saving the reader settings. At any time, the "S" command (upper case S) can be sent to the reader. This will change the default power-up state of the reader to be the same as the current settings

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



tagsense

WRITING A NEW ID TO EPC GEN 2 TAG

The Nano-UHF also allows the user to easily write an EPC ID number to an EPC Class 1 Gen 2 Tag. The command for EPC write is simply the letter capital W, followed by the 24 ASCII characters representing the 96-bit EPC ID. The target command T is used in conjunction with the write command in order to address a specific tag, as described below.

Writing to a single tag:

If there is a single tag present in the reading zone of the reader antenna, then there is no need to specify a particular target tag to be written to; by sending the command T followed by a period and carriage return, the target is set to *none*, meaning that any subsequent write command will write to any tag in the field, as long as there is only one present. If more than one tag is present, an error will be returned.

The EPC ID of the tag can be written with the command W followed by the 24 characters which represent the 96-bit EPC ID, followed by a carriage return.

If the command completes successfully, then the reader will return a 'W' character. If the command is not successful, then an error code (number) will be returned.

The following is an example:

T.<CR>
<LF>**T<CR><LF>**

W012345678901234567890123<CR>
<LF>**W<CR><LF>**

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



WRITING A NEW ID TO AN EPC GEN2 TAG

Writing to a single tag when many tags are present:

In some cases, there may be many tags present in the reading zone, but it is desired to write to just one of the tags present. In this case, it is necessary to select which tag is to be written. Using the target command “T”, a specific EPC tag can be selected. After selecting the tag using the T command, the new 96-bit EPC ID can be written using the W command.

For example, if we have a tag with ID 222222222222222222222222 and wish to change it to ID 444444444444444444444444, then the following commands would be used:

```
T222222222222222222222222<CR>
<LF>T<CR><LF>
W444444444444444444444444<CR>
<LF>W<CR><LF>
```

Reader Acknowledgement and Error Codes:

The character “W” or “L” returned by the reader indicates that the write procedure was successful. If the writing fails the reader will return an error code.

ERROR CODE	MEANING
1	Invalid command
2	Missing argument
3	Incorrect argument
4	Operation failed because no tag was detected
5	Operation failed because more than one tag was detected
6	Tag was detected but operation did not complete
7	Lock failed (only on ‘L’ command)
8	Operation failed because tag is locked
9	Operation failed due to insufficient power
10	Operation failed because of incorrect access password



tagsense

EPC GEN 2 TAG MEMORY STRUCTURE

The memory structure for EPC Gen 2 tags contains 4 main memory *banks*. Each memory bank is comprised of several memory blocks or *words*, which are 16-bits long (2 bytes). Each of the memory banks has a different size, has a different behavior, and serves a different purpose. These are briefly described below:

Memory Bank 0: Reserved Memory:

Memory Bank 0 is named the “Reserved Memory Bank” and contains the main access password of the tag as well as the kill command password. Each of these passwords is 32 bits.

Memory Bank 1: EPC Memory:

The memory bank contains the EPC ID of the tag as well as several other fields:

CRC-16 bits – used for tag to reader communications

Protocol Control (PC) bits – contains the **NSI** (Numbering system Identifier) and optionally the **AFI** as well as bits that describe the version of EPC code that is being used by the tag

Memory Bank 2: Protocol Control TID:

This memory bank contains vendor-specific information that helps identify the vendor of the tag chip as well as specific product information, such as the user memory size and any custom commands or capabilities present in the specific tag.

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



EPC GEN 2 MEMORY STRUCTURE

Memory Bank 3: User Memory:

This memory bank is optional. Several tag chips are currently available with user memory. Some of these are listed below:

Supported Tag Vendors: (these are only the ones that are publically disclosed)

- Impinj Monaco – 8 bytes of user memory (= 4 words)
- NXP (Phillips) – 28 bytes of user memory (= 14 words)
- ST Microelectronics (XRAG2) – 16 bytes of user memory (= 8 words)
- EM microelectronic
- Alien Technology
- FEC (Japan, Malaysia)

Following the EPC protocol, the User memory is usually read and written in units of words, which is equivalent to 2 bytes at a time.

Remember that when using the **U** command, the argument is given in units of *words*, not *bytes*. (1 word = 2 Bytes = 16 bits)

For more information on the memory structure of the EPC Gen 2 Tag, we refer the user to the EPC Gen 2 Specification document, which can be obtained from the EPC Global web site (<http://www.epcglobal.org>).



READING AND WRITING TAG MEMORY

TagSense provides 2 simple commands to reader and write to any memory bank on the EPC tag. The usage of the user memory commands is similar to the EPC write command, including the use of the target command **T** to either specify a particular target tag, or to target any single tag in the field. The syntax for these commands is given below:

Read user memory:

u<Memory Bank>,<Starting Address>,<Length of Data Requested><CR>

This command has three arguments separated by commas.

The **Memory Bank** selects which bank to read from; valid values are 0,1,2 or 3. Each of these memory banks is described in another section of this datasheet.

It is important to note that each bank is a different size, and not all memory banks may be implemented on all tags.

The **Starting Address** is expressed in units of data words. One data word = 16 bits. For example, on the NXP chip, valid starting addresses for the user memory range from 0 to 13.

The **Length of Data** is also in units of words (2 bytes). The term “memory block” is sometimes used instead of the term “data word.”

Example #1:

We want to read the certain specific bits in the user memory of tag number 777777777777777777777777. We want to read user memory bits 32 through 80. Bit 32 is the start of the second memory block, so the starting address is 2. We want to read 48 bits, which is equivalent to 3 data words.



tagsense

READING AND WRITING TAG MEMORY

For example #1, the following commands would be issued:

```
T77777777777777777777777777<CR>
<LF>T<CR><LF>
u3,2,3<CR>
<LF>ubdbdbdbdbdbdbdbdbdbdbbd<CR><LF>
```

The characters “bdbd...” is returned by the reader and represents the 6-bytes (=48-bits) of user memory that was requested.

Writing to user memory:

```
U<Memory Block>,<Starting Address>,< Data ><CR>
```

This command also has three arguments separated by commas.

The Memory Bank selects which bank to write to; valid values are 0,1,2 or 3.

The Starting address is expressed in units of data words. One data word = 16 bits = 2 bytes. For the NXP chip, the starting address ranges from 0 to 13.

The Data field is the data to be written to the tag. Since reading/writing to a tag is done in units of words (2 bytes), the data should be a multiple of 2 bytes.

Example #2:

We want to write the user memory of tag number 77777777777777777777777777.

We want to write a value of 12345678 (in hexadecimal units) to the user memory, starting at memory location 0, which is the start of user memory.

The following commands would be issued:

```
T77777777777777777777777777<CR>
<LF>T<CR><LF>
U3,0,12345678<CR>
<LF>U<CR><LF>
```

Note that the data contains 8 hex characters, which is 4 bytes. Remember that the data field must be multiple of 2 bytes (= 1 data word).

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



USING EPC GEN 2 SECURITY FEATURES

The TagSense readers support all of the security features contained in the EPC Generation 2 protocol. Although the EPC Gen 2 tags were designed to be low-cost and relatively simple, the EPC Gen 2 protocol provides a useful amount of security for general purpose applications.

Using Passwords

There are 2 fundamental passwords that are used in an EPC Gen 2 tag: the Access Password and the Kill Password. The Access Password must be used to read or write any part of memory that has been locked. The Kill Password is used to permanently deactivate the tag.

Setting the Passwords in the Tag Memory

Access password:

The access password is a 32-bit value located at word memory address 2 of the reserved memory bank. Typically by default, this password is set to zero. This password can be read from or written to the tag by using the **u/U** commands. To set a tag's access password to 0x12345678, use the following command:

U0,2,12345678<CR>
<LF>U<CR><LF>

Subsequently, to read back a tag's access password, use the following command:

u0,2,2<CR>
<LF>u12345678<CR><LF>



USING EPC GEN2 SECURITY FEATURES

Kill password:

The kill password is a 32-bit value located at word memory address 0 of the reserved memory bank. Typically by default, this password is set to zero. This password can be read from or written to the tag by using the **u/U** commands. For example, to set a tag's kill password to 0x87654321, the following command would be used:

U0,0,87654321<CR>
<LF>U<CR><LF>

Subsequently, to read back a tag's kill password, use the following command:

u0,0,2<CR>
<LF>u87654321<CR><LF>

The kill password is needed in order to kill a tag. A killed tag will no longer respond to any further reader commands. Tags must have their kill passwords set to a value other than the default 0 in order to be killed; a tag with its kill password set to zero cannot be killed. The kill password can be supplied to the reader and read back from the reader using the **c/C** commands. This kill password will be applied to any kill commands until a new access password is set with the **C** command. The **X** command is used to kill a tag.

There is no real practical reason to set the internal kill password on the reader to all zeroes. To set the kill password used by the reader to another value, such as 0x87654321, the following command is used:

C12345678<CR>
<LF>C<CR><LF>

The kill password stored in the reader can also be read back using the **c** command:

c<CR>
<LF>c12345678<CR><LF>



USING EPC GEN 2 SECURITY FEATURES

Setting and Using the Password Buffer in the Reader Memory

Once the passwords have been set in the tag memory, the reader must use the correct password every time it needs to communicate with the tag to execute certain commands. The reader itself needs to know the access password of a particular tag in order to write to any of its locked memory banks. Additionally, the access password is needed in order to read back either the access password or the kill password if either of them is locked. Any other memory bank beside the reserved memory bank can always be read regardless of their lock status. The Reserved memory bank has the ability to be read-locked in order to prevent unauthorized used from reading the tag passwords.

In order to simplify the communication between the host and the reader, TagSense has created a *password buffer* in the reader memory. Instead of requiring the host to transmit the password for every relevant command, the host simply needs to write the password buffer once and the reader will use that access password as needed to perform its functions. The password buffer thus contains the password to be used in all communications between tag and reader.

The **B** command is used to supply the reader internally with an access password to be used when performing memory read (**u**) or write (**U** and **W**) commands. Additionally, this access password is used when changing the lock status of any memory bank (**L**) (see “*Locking the Tag*”). This access password will be applied to any subsequent memory accessing/locking commands until a new access password is set with the **B** command. To set the access password used by the reader to zero, either of the following commands can be used:

B.<CR>
<LF>**B<CR><LF>**

or

B00000000<CR>
<LF>**B<CR><LF>**

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



tagsense

USING EPC GEN 2 SECURITY FEATURES

To set the access password to any other arbitrary 32 bit value (e.g. 0x12345678), use:

```
B12345678<CR>  
<LF>B<CR><LF>
```

The access password stored in the reader can also be read back using the **b** command:

```
b<CR>  
<LF>b12345678<CR><LF>
```

There are several important things to note when implementing passwords:

- First, in general, if the reader password buffer (set by the **B** command) differs from the access password stored on the tag during a memory access operation, an error will be returned. An exception to this is in the case when the internal access password is set to all zeroes, and the particular memory location being accessed is not locked.
- Second, the EPC, TID, and User memory banks are all always readable; however, these fields can be write protected (locked).
- The Reserved memory bank where the access and kill passwords reside is both read and write lockable, so the correct access password must be used in order to read or to write these passwords.
- Obviously, if the reader is communicating with multiple tags, and each tag contains a different access password, then the host must remember to rewrite the reader password buffer as needed for each tag.

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



LOCKING PARTS OF TAG MEMORY

Each of the different parts of the tag memory can be locked, which restricts access to the tag in various ways. If a certain part of memory is locked, then it is not possible to access this memory without using a specific password. In addition, the EPC specification provides the ability to *permanently* lock or *permanently* unlock certain parts of memory. Permanently locking or permanently unlocking is known as "**permalock**".

The following options are available for locking:

- **Read Unlocked** – there is unrestricted access to read from this memory
- **Read Unlocked** – there is unrestricted access to read from this memory
- **Read Locked** – the memory cannot be accessed for reading without using a password
- **Write Locked** – the memory cannot be accessed for writing without using a password
- **Read Permanently Unlocked** – there is unrestricted access to read from this memory and this memory can never be locked
- **Write Permanently Unlocked** – this memory cannot be accessed for writing without a password and this memory can never be locked

There are five sections of memory that can be each individually locked: the kill password, the access password, the EPC memory, the TID memory, and the user memory. Additionally, each of these memory locations can be locked, unlocked, permanently locked, or permanently unlocked.

Using the Lock Command

Since there are multiple sections of lockable memory and there are six different possible settings for each one, the memory lock configuration of a tag can be quite complicated. However, TagSense has created a simple but powerful command that enables the user to configure the tag memory using just a single command.

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



LOCKING PARTS OF TAG MEMORY

The lock status of a tag can be set/changed using the **L** command in the following format:

L02AA,01BB

Where *02AA* and *01BB* are each 10 bit arguments.

The **L** command takes two hexadecimal arguments; the first argument is a 10-bit lock mask and the second argument is a 10-bit action parameter. Each of these is described below.

Argument #1: the Lock Mask

The lock mask argument provides a way to select which specific memory banks are to be modified. Each bit in the 10-bit mask argument corresponds to a particular memory bank. The user can designate which memory blocks to modify by setting the corresponding bits to 1. If a particular bit is set to 0, then the corresponding memory lock bit is left unchanged. The following table shows the meaning of each lock mask bit.

Bit 9	Bit 8	Bit 7	Bit 6	Bit 5
1: Lock/unlock kill password 0: Do nothing to kill password lock bit	1: Permalock kill password 0: Do nothing to kill password permalock bit	1: Lock/unlock access password 0: Do nothing to access password lock bit	1: Permalock access password 0: Do nothing to access password permalock bit	1: Lock/unlock EPC memory 0: Do nothing to EPC memory lock bit
Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1: Permalock EPC memory 0: Do nothing to EPC memory permalock bit	1: Lock/unlock TID memory 0: Do nothing to TID memory lock bit	1: Permalock TID memory 0: Do nothing to TID memory permalock bit	1: Lock/unlock User memory 0: Do nothing to User memory lock bit	1: Permalock User memory 0: Do nothing to User memory permalock bit

Table 1. Lock mask bits definition.



LOCKING PARTS OF TAG MEMORY

Argument #2: the Lock Action

The second argument in the Lock command defines the specific operation to be performed on the memory banks selected by the first argument. The lock action bits determine if the corresponding memory banks are locked, unlocked, permanently locked or permanently unlocked.

The following table shows the meaning of each lock action bit. Keep in mind that if the corresponding bit in the lock mask is set to 0, then no action is taken regardless of the value of that particular bit in the lock action.

Bit 9	Bit 8	Bit 7	Bit 6	Bit 5
1: Lock kill password 0: Unlock kill password	1: Permanently lock kill password 0: Permanently unlock kill password	1: Lock access password 0: Unlock access password	1: Permanently lock access password 0: Permanently unlock access password	1: Lock EPC memory 0: Unlock EPC memory
Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1: Permanently lock EPC memory 0: Permanently unlock EPC memory	1: Lock TID memory 0: Unlock TID memory	1: Permanently lock TID memory 0: Permanently unlock TID memory	1: Lock User memory 0: Unlock User memory	1: Permanently lock User memory 0: Permanently unlock User memory

Table 2. Action bits definition.

IMPORTANT: In order to use the lock command, the tag access password is required. Therefore, the password stored in the reader password buffer must match the access password stored in the tag's memory. For convenience, the TagSense reader will automatically supply the password as needed, but the user should be aware of this, and update the password buffer properly.



tagsense

LOCKING PARTS OF TAG MEMORY

The following examples will help illustrate the use of the **L** command.

To lock the EPC memory such that new IDs cannot be written, the following command would be used:

```
L0020,0020<CR>
<LF>L<CR><LF>
```

Note that if the access password on the tag has not been set to a non zero value, the EPC memory will still be writeable.

To unlock the EPC memory, the following command would be used:

```
L0020,0000<CR>
<LF>L<CR><LF>
```

Several memory locations can have their lock status changed in the same command. For example, to lock the user memory, permanently lock the access password, and permanently unlock the EPC memory, the following command would be used:

```
L0062,0042<CR>
<LF>L<CR><LF>
```

NOTE: Unfortunately, the EPC Gen2 protocol does not provide any direct way to read back the lock status of a tag; therefore the lock status of particular memory banks must be inferred through interactions with the tag.

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



tagsense

READER SESSIONS

There are many RFID applications where there are several RFID readers (or antennas) that are adjacent to each other and reading a common population of tags. For these applications, the EPC Gen 2 protocol contains a protocol parameter known as "Sessions." A session is essentially a communication channel between the reader and a tag.

If two adjacent readers are simultaneously sending commands to the same tag, then special care must be taken to ensure that the two readers will not confuse the tag and produce erratic behavior. The use of the Sessions parameter in the EPC Gen 2 protocol allows the user to choose if both readers will share the same communications channel with the tag or if each reader will have its own "private" communications channel with the tag. This provides the ability to create RFID applications where several readers (or reader antennas) are in close proximity to each other.

Persistence

The EPC Gen 2 protocol provides up to 4 separate sessions per tag. However, the 4 sessions have different physical behaviors based on the concept of *persistence*. The persistence refers to the amount of time that the tag will remain in a particular state without resetting.

In certain RFID applications, for example, it is desirable for a tag to have a low persistence so that the tag will quickly reset itself when it moves from one reader to another reader. In other applications, however, it is desirable to have a long persistence, so that the tag will tolerate momentary losses of electromagnetic power without resetting. For these reasons, the EPC Gen 2 protocol was designed to have sessions with different degrees of persistence. These are listed in Table 3.

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



tagsense

READER SESSIONS

	PERSISTENCE IN FIELD	PERSISTENCE OUT OF FIELD
SESSION 0	INDEFINITELY LONG	None
SESSION 1	.5 to 5 seconds	None
SESSION 2	INDEFINITELY LONG	.5 to 5 seconds
SESSION 3	INDEFINITELY LONG	.5 to 5 seconds

Table 3. Persistence value for different Reader Sessions inside and outside the electromagnetic field of the reader antenna.

Since Session 2 and Session 3 are both identical, these 2 sessions are commonly used for installations requiring adjacent readers. Session 0 and Session 1 have special applications such as for situations where tags must move quickly across several reading stations.

TagSense, Inc.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.



READER SESSIONS

Using the Session Command

When inventorying tags using either the continuous looping (**k**) or polling (**Z**) commands, the user has the option of selecting which one of four sessions is used. Tags have independent inventory flags for each session, allowing multiple readers to query a common population of tags without interfering each other assuming they use different sessions. The session can be set using the **Q** command, and read back using the **q** command. For example, to set the reader to use session 3 when inventorying tags, the following command would be used:

```
Q3<CR>
<LF>Q<CR><LF>
```

To read back which session is set on the reader, the following command would be used:

```
q<CR>
<LF>q3<CR><LF>
```

When data streaming is turned off (**D**), there are some differences in performance depending on which session is used by the reader. The cause of this is that the tag's inventory flag for each session has a different *persistence* value. The persistence value is the amount of time a tag is capable of retaining the state of its inventory flag both when its powered and unpowered.



NANO-UHF READER BASIC COMMANDS

The basic commands of the serial ASCII based API (Application Peripheral Interface) are summarized below.

Command	Arguments	Description
P	0	Enables multi-protocol mode.
P	1	Sets the reader to EPCG1
P	2	Sets the reader to EPCG2 ASK
P	3	Sets the reader to EPCG2 PSK (<i>not yet implemented</i>)
p	none	Returns the current protocol mode
k	none	Turns on transmit power and enables continuous scanning
K	none	Turns off transmit power and disables continuous scanning
Z	none	Triggers/polls the reader to do a single inventory scan
W	12-bytes	Writes 12 byte ID to EPC tag without locking the tag
d	none	Enables Tag ID streaming
D	none	Disables Tag ID streaming
m	none	Enables output of the protocol type in addition to the tag ID.
M	none	Disables including the protocol type in the reader output
i	none	Enable adding reader ID code to the output
I	none	Disable adding reader ID to the output
v	none	Enable adding frequency channel to the output
V	none	Disable adding frequency channel to the output
F	1	Sets frequency range to N. American limits (902-928 MHz)
F	2	Sets frequency range to European/India limits (865-868 MHz)
F	3	Sets frequency range to Japanese limits (~955)
F	4	Sets transmit frequency to Korean limits (905-912 MHz)
f	none	Returns the current frequency range
R	8-byte int	set the reader ID number
r	none	prompts the reader to return its reader ID number
s	none	Displays the current state of the reader.
S	none	Saves the current state.
T	12-bytes or ''	Sets the target to the 12 byte ID
t	none	Returns the current target

Note: All commands must be terminated by a carriage return CHR\$(13).



NANO-UHF READER ADVANCED COMMANDS

The following are advanced commands which can be used to fine-tune the performance of the reader or customize the reader for special applications:

Command	Argument	Description
h	None	Enables frequency hopping (Only available to research/university labs)
H	None	Disables frequency hopping (Only available to research/university labs)
G	1-50	Sets the frequency channel when frequency hopping is disabled. By default, reader uses mid-band frequency.
g	None	Returns current frequency channel
J	0-27	Sets the power level output
j	None	Returns current power level
N	0-9	Sets the number of slots used during the EPC C1G2 anti-collision round. The number 0-9 indicates the exponent (power of 2). For example, N=4 represents $2^4 = 16$ slots. For fastest performance, it is good to set the number of slots just slightly greater than the number of tags expected to be present. Setting too few or too many slots will waste time.
n	None	Returns current number of slots used for anticollision
Q	0-3	Sets the session number used by the reader.
q	None	Returns the current session number.
B	4-bytes or '.'	Sets the Access Password to be used by the reader
b	None	Returns the currently set Access Password
C	8-bytes or '.'	Sets the Kill Password to be used by the reader
c	None	Returns the currently set Kill Password
Y	0-9	Sets the number of anticollision rounds attempted per each poll command ('Z').
y	None	Returns number of anticollision rounds attempted per each poll
E	none	Enables echo of RS-232 command (user can view characters as the commands are being sent).
e	none	Disables echo of RS-232 command

Note: All commands must be terminated by a carriage return CHR\$(13).



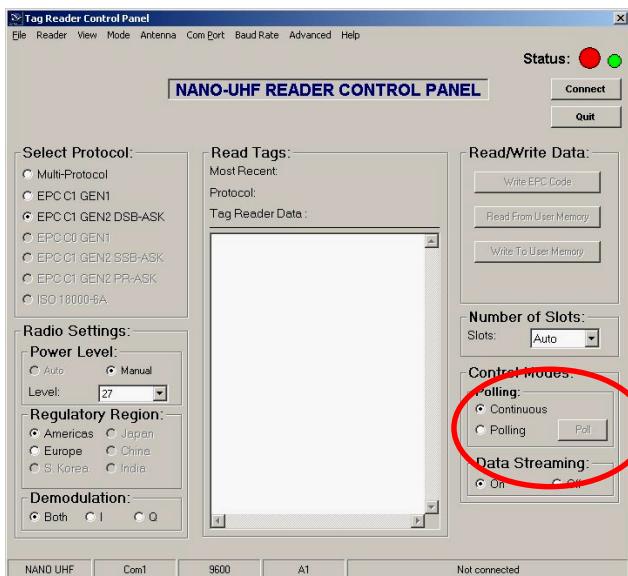
tagsense

POWER SAVING MODE

The Nano-UHF reader will automatically turn off the power to all the reader radio circuitry when the reader is not transmitting. This is called *power saving mode* or *stand-by mode*.

To enter the power saving mode, the host should send a "K" command. This will turn off the transmitter and also turn off the radio section. Tag reading is accomplished through polling. To read a tag, the host can simply send a polling command "Z". This will briefly turn ON the reader to read some tags and then return to standby mode. For handheld RFID reader applications, this is usually the best way to save power.

If you want the reader to remain on continuously, then the host should send the "k" command. This will turn OFF polling and cause the reader to transmit continuously. If you are using the Windows software supplied with the evaluation kit, then the "k" and "K" commands can be controlled using the polling mode control shown below.



Use these buttons
to control the
power saving mode
by enabling or
disabling polling.

1035 Cambridge St. #8
Cambridge, MA 02141
(617) 494-1001

TagSense, Inc.

<http://www.tagsense.com>
email: info@tagsense.com
©2000-2010 TagSense, Inc.