The Micro-UHF reader is a small profile, low power, low-cost RFID reader with a USB interface suited for embedded applications, such as handheld readers, printers, laptops, or smart shelves. An external antenna can be connected via an SMA connector. A small yet powerful ASCII command set makes this reader useful for many applications.

**Unique Features:**
- Very easy to use command set
- Multi-protocol: EPC Class 1 Gen 1, EPC Class 1 Gen 2 (Read/Write)
- Software-controlled power level
- Programmable number of time slots to allow for small and large tag populations
- Operates at both the European UHF frequencies (865-868 MHz) and North American UHF frequencies (902-928 MHz)
- Reader can automatically identify the protocol of the tag being read
- Programmable Reader ID code 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.
### DOCUMENT VERSION HISTORY

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>By</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>11/09/06</td>
<td>RRF</td>
<td>Added user memory, modified write command</td>
</tr>
<tr>
<td>1.4</td>
<td>9/20/06</td>
<td>RMR</td>
<td>Added power, slots commands, lock-write</td>
</tr>
<tr>
<td>1.3</td>
<td>7/25/06</td>
<td>RRF</td>
<td>updated physical specs</td>
</tr>
<tr>
<td>1.2</td>
<td>5/23/06</td>
<td>RRF</td>
<td>updated software instructions</td>
</tr>
<tr>
<td>1.1</td>
<td>5/09/06</td>
<td>RRF</td>
<td>added EPC write, error codes</td>
</tr>
<tr>
<td>1.0</td>
<td>3/31/06</td>
<td>RMR</td>
<td>Initial Draft</td>
</tr>
</tbody>
</table>
MICRO-UHF SPECIFICATIONS

Electrical Specifications
- Operating Voltage:
  - 5 VDC nominal (supplied by USB port)
  - 4.3 Volts Min
  - 6 Volts Max.
- Current consumption (not including USB):
  - 120mA when transmitting
  - <2 mA when idle
- Antenna:
  - 50 Ohms
  - SMA female jack
  - Can be customized

Interface Specifications
- Host Interface:
  - 3-pin header (V+, GND, Data)
  - RS-232 TTL-level
- Sensor Input:
  - 3-pin header (V+, GND, Vin)
  - 8-bit A/D

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)

Physical Specifications
- Board Size:
  - Width = 38mm (1.5 inch)
  - Length = 46mm (1.8 inch)
  - Height = 7mm (0.27 inch)
- Environment:
  - 0C – 85C (32F – 185F)
The Micro-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 microcontroller.

The transmit RF power of the Micro-UHF is 50 mW, which is sufficient for low-power short range applications. The Micro-UHF hardware supports software control for output power level.
There are two ways to interface with the Micro-UHF. The simplest way is to use the TagSense Reader Control Panel software, which comes with the Micro-UHF Evaluation Kit. To install this software, follow the instructions that come with the CD. Additionally, drivers for the USB Virtual Com Port must be installed (instructions for installing these can also be found on the CD). Linux drivers are also available, although TagSense does not provide any sample code.
After installing the USB drivers and the demo program, you can connect the reader to the computer using a USB cable. 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 “MICRO-UHF”. Then go to the ComPort menu and select the proper COM port for your USB port. If you are not sure, then you can use the USB COM Port locator utility, which is under the Help menu. You should then verify that the BaudRate setting is correct (9600), 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.
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 Micro-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 Micro-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 Micro-UHF. However, the Micro-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 Micro-UHF will output a line feed LF = CHR$(10), and then begin processing the command. If the command is valid and well formed, the Micro-UHF will output an acknowledgement (with two exceptions, noted later). The acknowledgement consists of the command character (without the argument), followed by a carriage return CHR$(13) and line feed CHR$(10).
CONFIGURING THE MICRO-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:

\[ p2 \] put the reader in EPC C1G2 ASK mode
\[ k \] set the reader to continually scan for tags
\[ m \] 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.
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.
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.
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 power levels above 100 mW. 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.

Saving the current reader settings:

At any time, the setting of all the reader parameters can be displayed using the “s” command. Below is some sample data output that would be returned after sending the (lower-case) “s” command:

```
p=0 protocol mode
L=1 continuous autoscan – 0: disabled, 1: enabled
I=0 printing of reader ID
M=1 printing of protocol type
```

After changing the reader parameters, the power-up configuration of the reader can be set to the current state by using the (upper-case) “S” command. By sending the “S” command, the current reader state is saved and the reader will keep this state upon power reset.

```
S save the current configuration in the EEPROM
```
The Micro-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 and is illustrated below. This command has 2 arguments, separated by a comma.

**Writing to a single tag:**

If there is a single tag present in the reading zone of the reader antenna, then the EPC ID code of the tag can be written with the command W followed by a period, comma, and the 24 characters which represent the 96-bit EPC ID.

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:

```
W.,012345678901234567890123<CR>
<LF>
W<CR><LF>
```

**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 one of the tags present. In this case, it is necessary to select which tag is to be written. To write to a specific tag, the command W is issued, followed by the 96-bit ID of the tag to be written, a comma, and the 96-bit new ID.

For example, if the tag we wish to rewrite has the following ID:

```
222222222222222222222222
```

and we wish to change this to the following new ID:

```
444444444444444444444444
```

the following command would be used:

```
W222222222222222222222222,444444444444444444444444<CR>
```


**Locking the ID code:**

In many cases, it is desired to write a permanent ID code to the tag and prevent the tag ID from being reprogrammed at a later time. This is accomplished through a *lockable write* command, which permanently locks the tag. The command for lockable write is the “L” command. The usage of the “L” command is exactly the same as the standard write command “W”:

\[
\text{L.,012345678901234567890123<CR>}
\]
\[
\text{<LF>}
\]
\[
\text{L<CR><LF>}
\]

**Write Password:**

Not publicly available.
Contact TagSense.

**Kill Password:**

Not publicly available.
Contact TagSense.
WRITING TO A TAG

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. Since there are many reasons why the writing process can fail, we provide a few different error codes to help give the user more information. These error codes are given in the table below:

<table>
<thead>
<tr>
<th>ERROR CODE</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>No tag was detected</td>
</tr>
<tr>
<td>5</td>
<td>More than one tag was detected</td>
</tr>
<tr>
<td>6</td>
<td>Tag was detected but writing did not complete</td>
</tr>
<tr>
<td>7</td>
<td>Tag was written to, but lock failed (only on ‘L’ command)</td>
</tr>
<tr>
<td>8</td>
<td>Tag is locked (no write is possible)</td>
</tr>
<tr>
<td>9</td>
<td>Write failed due to insufficient power</td>
</tr>
</tbody>
</table>
Certain RFID Tags are available with additional memory in addition to the EPC code. The amount of user memory available depends on the specific chip vendor and chip product. At present, the TagSense reader modules support the user memory EPC tag from NXP (formerly Philips Semiconductors).

**NXP (formerly Philips) U-Code EPC:**

This chip contains a 512-bit memory:

- EPC ID: 96-bits
- User memory: 224-bits
- Factory-programmed serial number: 64-bits
- User password: 64-bits
- Kill password: 64-bits
TagSense provides 2 simple commands to reader and write to user memory. The usage of the user memory commands is similar to the EPC write command. The primary difference is that the user memory field has an arbitrary length. The syntax for these commands is given below:

**Read user memory:**

u<TagID>,<Starting Address>,<Length of Data Requested><CR>

This command has three arguments separated by commas. The Tag ID is the usual 96-bit tag EPC ID. The Starting Address is expressed in units of data words. One data word = 16 bits. For the NXP chip, the starting address ranges from 0 to 13. The Length of Data is also in units of words. The term “memory block” is sometimes used instead of the term “data word.”

**Example:**

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.

The following command would be issued:

u777777777777777777777777,2,3<CR>

bdbdbdbdbdbdbdbdbdbdb<CR>

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<TagID>,<Starting Address>,< Data ><CR>

This command also has three arguments separated by commas. The Tag ID is the usual 96-bit tag EPC ID. 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 that you want to write to the tag. Since reading and writing to a tag is done in units of words (2 bytes), the data should be a multiple of 2 bytes.

Example:

We want to write the certain specific bits in the user memory of tag number 777777777777777777777777.
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 command would be issued:

U777777777777777777777777,0,12345678<CR>
U<CR>

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).

Shortcut for reading and writing:

If there is only a single tag in the field, a period (".") can be used in place of the tag ID. This eliminates the need to know the tag ID.
The basic commands of the serial ASCII based API (Application Peripheral Interface) are summarized below.

<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>0</td>
<td>Enables multi-protocol mode.</td>
</tr>
<tr>
<td>p</td>
<td>1</td>
<td>Sets the reader to EPCG1</td>
</tr>
<tr>
<td>p</td>
<td>2</td>
<td>Sets the reader to EPCG2 ASK</td>
</tr>
<tr>
<td>p</td>
<td>3</td>
<td>Sets the reader to EPCG2 PSK (<em>not yet implemented</em>)</td>
</tr>
<tr>
<td>k</td>
<td>none</td>
<td>Turns on transmit power and enables continuous scanning</td>
</tr>
<tr>
<td>K</td>
<td>none</td>
<td>Turns off transmit power and disables continuous scanning</td>
</tr>
<tr>
<td>Z</td>
<td>none</td>
<td>Triggers/polls the reader to do a single inventory scan</td>
</tr>
<tr>
<td>W</td>
<td>12-bytes</td>
<td>Writes 12 byte ID to EPC tag without locking the tag</td>
</tr>
<tr>
<td>L</td>
<td>12-bytes</td>
<td>Writes 12 byte ID to EPC tag and locks the tag</td>
</tr>
<tr>
<td>u</td>
<td>variable</td>
<td>Reads Data to specified user memory blocks</td>
</tr>
<tr>
<td>U</td>
<td>variable</td>
<td>Writes Data to specified user memory blocks</td>
</tr>
<tr>
<td>d</td>
<td>none</td>
<td>Enables Tag ID streaming</td>
</tr>
<tr>
<td>D</td>
<td>none</td>
<td>Disables Tag ID streaming</td>
</tr>
<tr>
<td>m</td>
<td>none</td>
<td>Enables output of the protocol type in addition to the tag ID.</td>
</tr>
<tr>
<td>M</td>
<td>none</td>
<td>Disables including the protocol type in the reader output.</td>
</tr>
<tr>
<td>i</td>
<td>none</td>
<td>Enable adding reader ID code to the output</td>
</tr>
<tr>
<td>I</td>
<td>none</td>
<td>Disable adding reader ID to the output</td>
</tr>
<tr>
<td>V</td>
<td>none</td>
<td>Enable adding frequency channel to the output</td>
</tr>
<tr>
<td>V</td>
<td>none</td>
<td>Disable adding frequency channel to the output</td>
</tr>
<tr>
<td>f</td>
<td>1</td>
<td>Sets transmit frequency to N. American limits (~915 MHz)</td>
</tr>
<tr>
<td>f</td>
<td>2</td>
<td>Sets transmit frequency to European limits (~868 MHz)</td>
</tr>
<tr>
<td>r</td>
<td>8-byte int</td>
<td>set the reader ID number</td>
</tr>
<tr>
<td>R</td>
<td>none</td>
<td>prompts the reader to return its reader ID number</td>
</tr>
<tr>
<td>s</td>
<td>none</td>
<td>Displays the current state of the reader.</td>
</tr>
<tr>
<td>S</td>
<td>none</td>
<td>Saves the current state.</td>
</tr>
</tbody>
</table>

**Note:** All commands must be terminated by a carriage return CHR$(13).
## MICRO-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:

<table>
<thead>
<tr>
<th>Command</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>None</td>
<td>Enables frequency hopping (Only available to research/university labs)</td>
</tr>
<tr>
<td>H</td>
<td>None</td>
<td>Disables frequency hopping (Only available to research/university labs)</td>
</tr>
<tr>
<td>G</td>
<td>1-50</td>
<td>Sets the frequency channel when frequency hopping is disabled. By default, reader uses mid-band frequency.</td>
</tr>
<tr>
<td>J</td>
<td>0-27</td>
<td>Sets the power level output</td>
</tr>
<tr>
<td>N</td>
<td>0-9</td>
<td>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.</td>
</tr>
<tr>
<td>Y</td>
<td>0-9</td>
<td>Sets the number of anticollision rounds attempted per each poll command ('Z'). Default is 1.</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>Both I/Q receive channels are used.</td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>Only I channel is used.</td>
</tr>
<tr>
<td>a</td>
<td>2</td>
<td>Only Q channel is used.</td>
</tr>
<tr>
<td>e</td>
<td>none</td>
<td>Enables echo of RS-232 command (user can view characters as the commands are being sent). This mode is turned off by default to increase speed of communications, but is useful for manual control by human (vs machine).</td>
</tr>
<tr>
<td>E</td>
<td>none</td>
<td>Disables echo of RS-232 command</td>
</tr>
</tbody>
</table>

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