ClarIDy UHF USB Reader Demo Program
User's Manual for Linux

Product Name: UHF USB Reader Module

Model No.: UEE006

Applicant: ClarIDy Solutions, Inc.

Version: A.1-01

2008-12-09

ClarIDy Solutions, Inc.

Copyright Notice

Disclaimer

The information of this document is subject to change without notice and does not represent a commitment on any part of ClarIDy Solutions, Inc.

Trademarks

Intel is a registered trademark of Intel Corporation.

AMD is a registered trademark of Advanced Micro Devices, Inc.

Linux is a registered trademark of Linus Torvalds.

RED HAT is a registered trademark of Red Hat, Inc.

All other product names are trademarks or registered trademarks of their respective owners.

Change Log

Revision	Date	Author	Description
A.1	2008.11.07.	Jun-Rong Chang	Create the ClarIDy UHF SDK Demo Program User's Manual for Linux
A.1-01	2008.12.09	Jun-Rong Chang	Added the application of Lock Tag and Kill Tag

Table of Contents

1. Introduction

This document describes the demo program for ClarIDy UHF RFID Reader. The demo program provides “Setting”, “Algorithm”, “Inventory”, “Read/Write”, “Filter Inventory”, “Filter Read/Write” and “Lock/Kill” functions. Users can use those applications to control the ClarIDy UHF RFID Reader to communicate with EPC RFID tags. The operational procedures are described as the following chapters.

2. Installation

Before Installing the USB Driver and ClarIDy UHF Demo program, please see prerequisites first.

2.0 Prerequisites to Installation

In order to run ClarIDy UHF Demo program as smoothly as possible, we recommend some conditions as following.

2.0.1 Hardware Platform Requirements

The following hardware Platforms are supported by ClarIDy UHF RFID Reader

- CPU: Intel® Pentium® 4 or AMD® Athlon™ processor, 1.4 GHz or above
- RAM: 512MB or above

2.0.2 Operating System Requirements

The following host operating systems are supported by ClarIDy UHF RFID SDK:

- Red Hat* Enterprise Linux*4 or other binary-compatible Linux kernel 2.6 distribution for the IA-32 platform.

2.0.3 Other Software Requirements

The following other systems are supported by ClarIDy UHF RFID Reader:

- GCC 3.4.6 and above.

2.1 Setup the USB Driver

The ClarIDy UHF Reader is support on Linux using user-level libraries that communicate with the standard Linux USB drivers. No driver installation is necessary.

2.3 Install ClarIDy UHF Demo

We only provide a command line application “ClarIDy_UHF_Demo” for demonstration.

1. Copy “ClarIDy_UHF_Demo_Linux.tar.gz” to Linux PC, as figure 1.
2. Extract the “ClarIDy_UHF_Demo_Linux.tar.gz” from tarball, as figure 2.
3. Run the “source setup.sh”, as Figure 3.

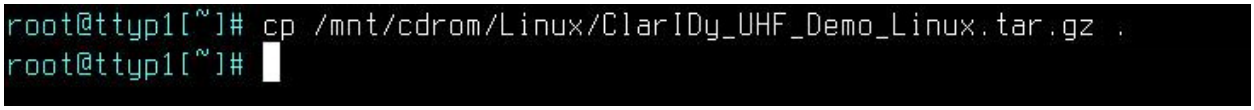

Figure 1 Copy “ClarIDy_UHF_SDK_Linux.tar.gz” to Linux PC

```
root@ttyp1[~]# tar zxvf ClarIDy_UHF_Demo_Linux.tar.gz
ClarIDy_UHF_Demo_Linux/
ClarIDy_UHF_Demo_Linux/ClarIDy_UHF_Demo/
ClarIDy_UHF_Demo_Linux/ClarIDy_UHF_Demo/ClarIDy_UHF_Demo
ClarIDy_UHF_Demo_Linux/lib/
ClarIDy_UHF_Demo_Linux/lib/libcpl.so
ClarIDy_UHF_Demo_Linux/lib/librfid.so
ClarIDy_UHF_Demo_Linux/lib/librfidtx.so
ClarIDy_UHF_Demo_Linux/lib/libRFID_UHF_SDK.so
ClarIDy_UHF_Demo_Linux/lib/libstdc++.so.5
ClarIDy_UHF_Demo_Linux/doc/
ClarIDy_UHF_Demo_Linux/doc/ClarIDy_UHF_Reader_Demo_Program_Linux.pdf
root@ttyp1[~]#
```

Figure 2 extract the “ClarIDy_UHF_SDK_Linux.tar.gz” from tarball

```
root@ttyp1[ClarIDy_UHF_Demo_Linux]# source setup.sh
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/root/ClarIDy_UHF_Demo_Linux/lib/
root@ttyp1[ClarIDy_UHF_Demo_Linux]#
```

Figure 3 Run “source setup.sh”

3. Demo Program Operation Guide

The following sections will describe the usage of the ClarIDy UHF Demo application.First run the "ClarIDy_UHF_Demo". The application is include two Controls, “Setting” and “Operation”, as Figure 4.

```
root@ttyp1[ClarIDy_UHF_Demo]# ./ClarIDy_UHF_Demo
Initialialize the RFID library....[OK]
Searching Reader ....[OK]
Reader0 Connect ...[OK]
```

```
1: Setting.
2: Operation.
Q: Quit.

input =>
```

Figure 4 Run “ClarIDy_UHF_Demo”

3.1 Setting

The ClarIDy UHF Demo Program support two configure (as Figure 5) as follows:

1. Set Reader Current Configure
2. Set Algorithm Configure

```
1: Set Reader Current Configure.
2: Set Algorithm Config.
3: Get Antenna Config.

B: Back.
Q: Quit.

input =>
```

Figure 5 Setting

2.2.1 Reader Current Configure

When the user select "1: Set Reader Current Configure" then show Configure page. This page allows the user to configure the following items:

1. Link Profile: Sets the current link profile for the reader module, as figure 6. The option is as follows:

   0: DSB ASK / MO / 40 khz

   1: DSB_ASK / M1 / 160 khz

   2: PR_ASK / M2 / 25U khz

   3: PR_ASK / M2 / 300 khz

   4: DSB_ASK / MO / 400 khz

   5: PR_ASK / M1 / 250 khz

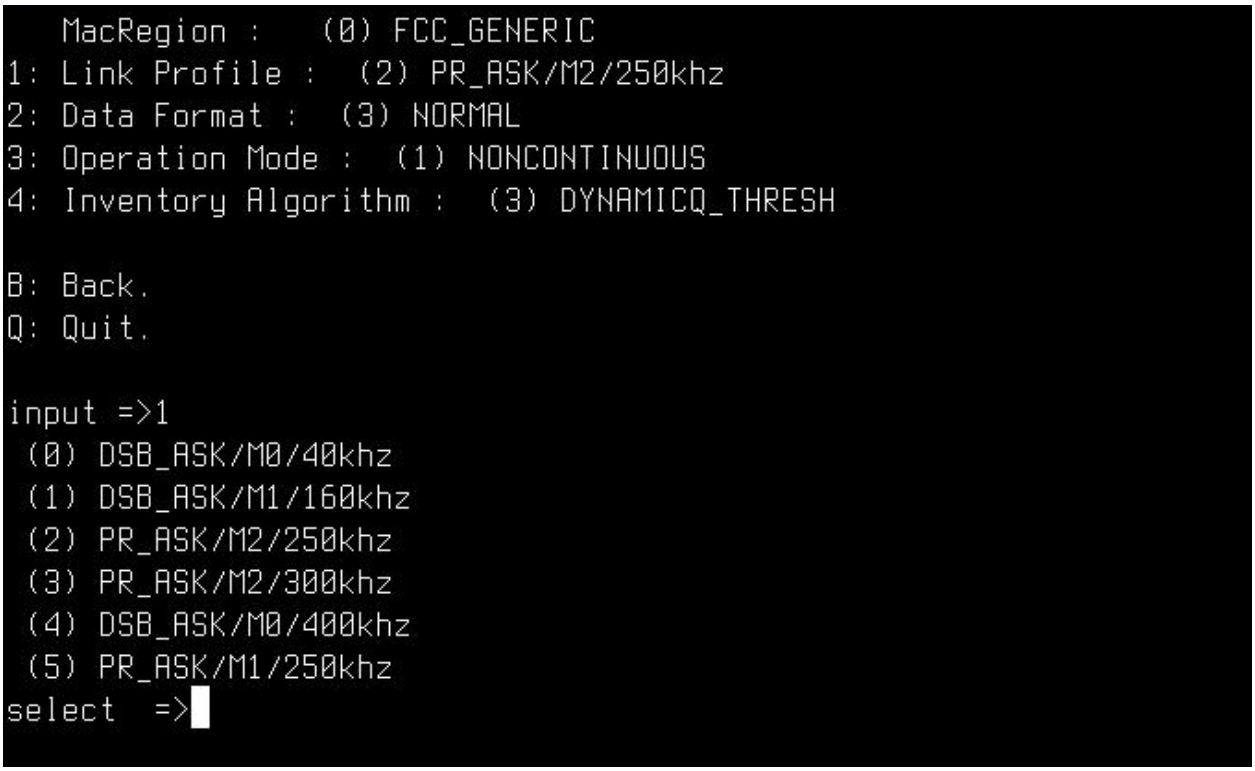

Figure 6 Set to Link Profile

2. Data Format: Sets the operation response data reporting mode for tag-protocol operations, as figure 7. The option is as follows:

   1: COMPACT: The response data is limited to provide the application with the pertinent tag-access operation data, but minimize the amount of MAC-to-host communication overhead.

   3: NORMAL: The response data builds on the compact mode to provide the application with status and contextual information to give additional finer-grained feedback such as the beginning of inventory cycles, etc.

   7: EXTENDED: The response data builds on the normal mode by providing additional diagnostics and statistical information.

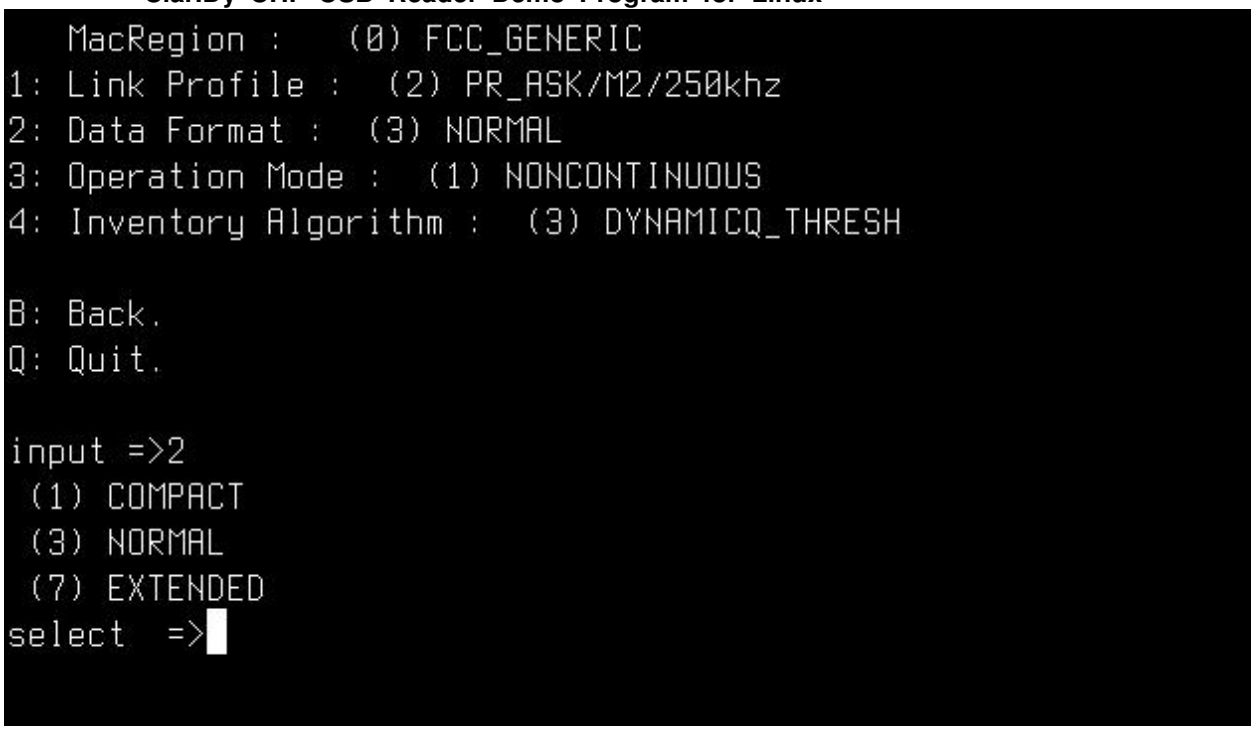

Figure 7 Set to Data Format

3. Operation Mode: Sets the reader's operation mode, as figure 8. The option is as follows:

   0: CONTINUOUS: In continuous mode, when a tag-protocol-operation cycle (i.e., one iteration through all enabled antenna ports) has completed, the reader module will begin a new tag-protocol-operation cycle with the first enabled antenna port and will continue to do so until the operation is explicitly cancelled by the application.

   1: NONCONTINUOUS: In non-continuous mode, only a single tag-protocol-operation cycle is executed upon the reader module.

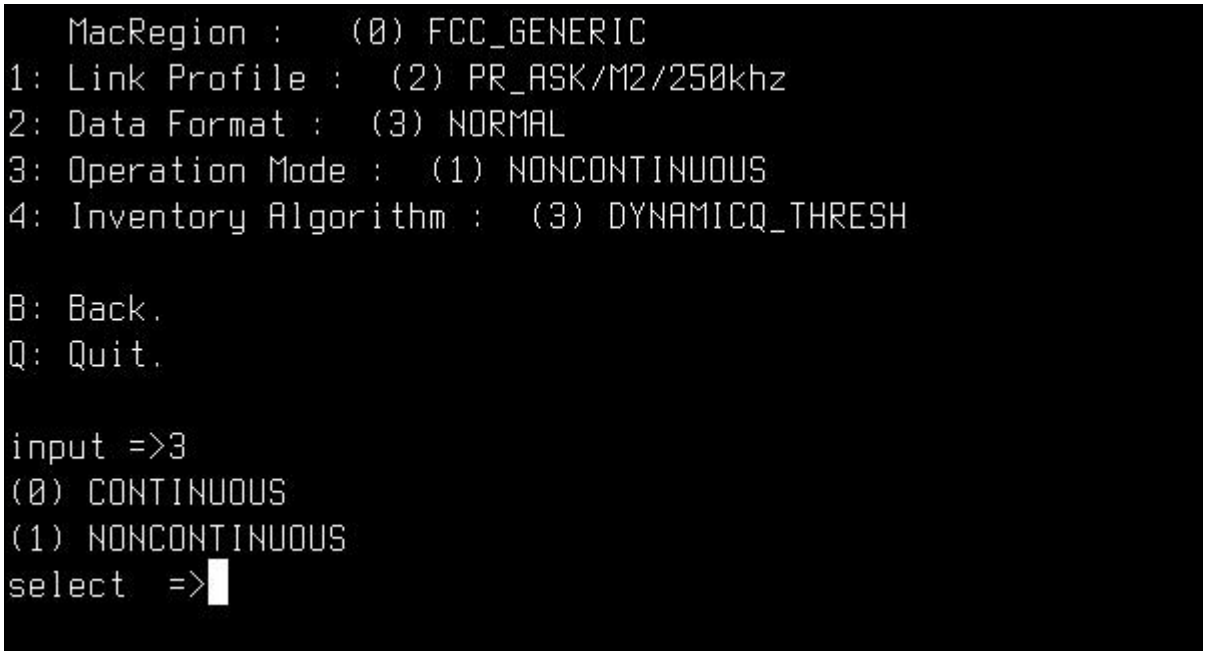

Figure 8 Operation Mode

4. Inventory Algorithm: Allows the application to set the currently-active singulation algorithm, as figure 9. The option is as follows:

0: ALGORITHM_FIXEDQ

1: DYNAMICQ

2: DYNAMICQ_ADJUST

3: DYNAMICQ_THRESHOLD

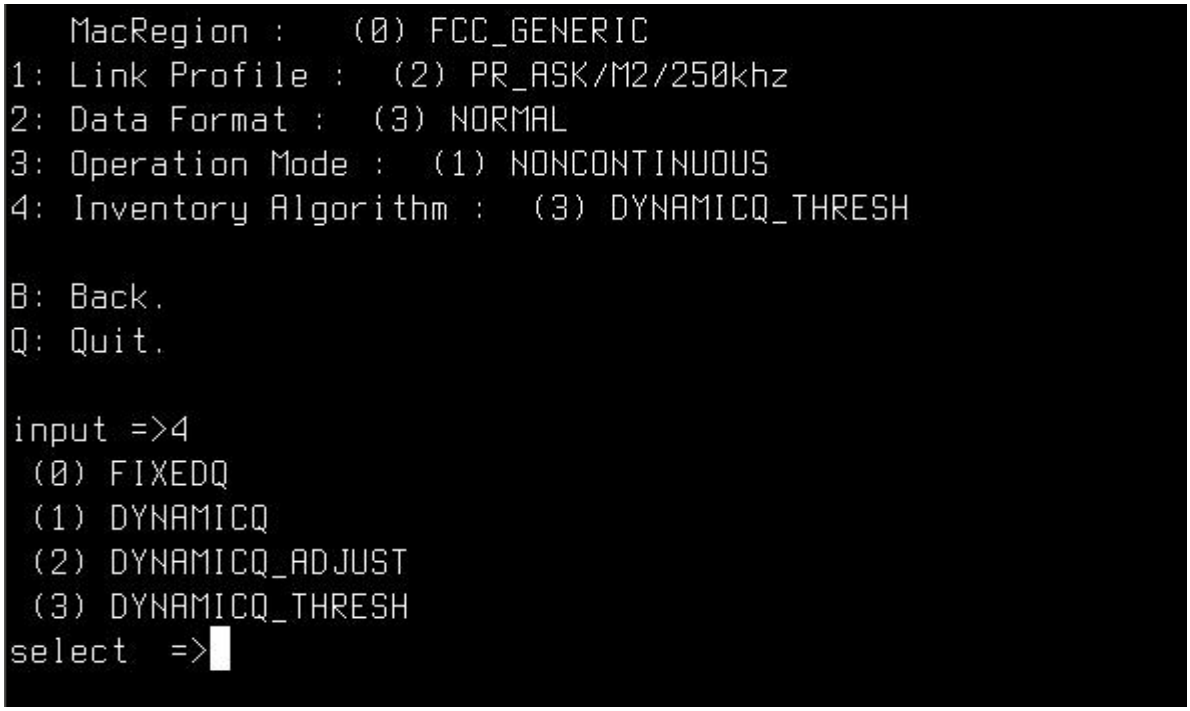

Figure 9 Set to Inventory Algorithm

2.2.2 Algorithm Configure

The Algorithm Configure page allows the user to configure the following items (as figure 10):

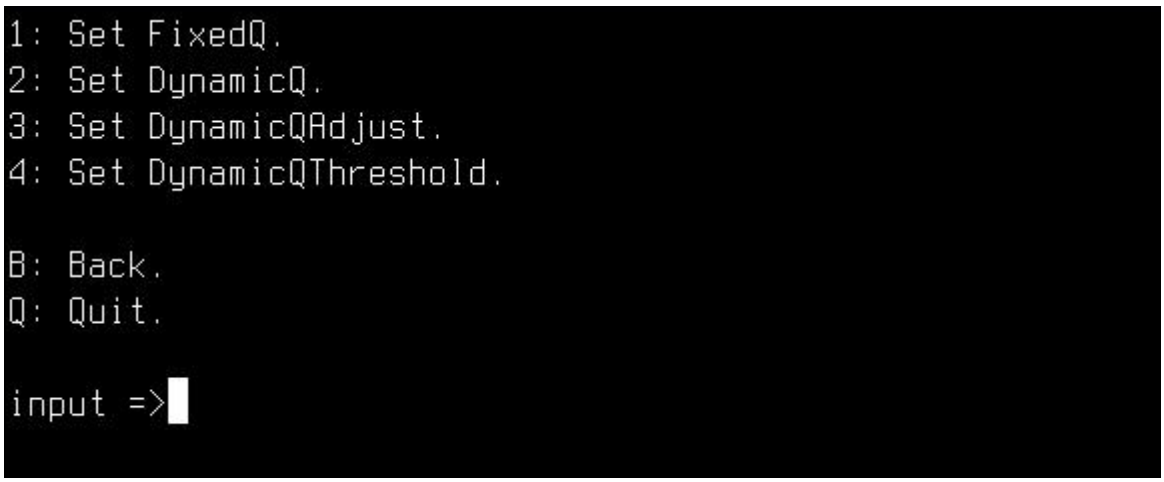

Figure 10 Algorithm Configure

1. FixedQ (as figure 11): Fixed Q algorithm. The items as follows:

   1: qValue: The Q value to use. Valid values are 0 to 15, inclusive.

   2: retryCount: Specifies the number of times to try another execution of the singulation algorithm for the specified session/target before either toggling the target (if toggleTarget is non-zero) or terminating the inventory/tag access operation. Valid values are 0-255, inclusive.

   3: toggleTarget: A flag that indicates if, after performing the inventory cycle for the specified target (i.e., A or B), if the target should be toggled (i.e., A to B or B to A) and another inventory cycle run. A non-zero value indicates that the target should be toggled. A zero value indicates that the target should not be toggled. Note that if the target is toggled, retryCount and repeatUntilNoTags will also apply to the new target.

   4: repeatUntilNoTags: A flag that indicates whether or not the singulation algorithm should continue performing inventory rounds until no tags are singulated. A non-zero value indicates that, for each execution of the singulation algorithm, inventory rounds should be performed until no tags are singulated. A zero value indicates that a single inventory round should be performed for each execution of the singulation algorithm.

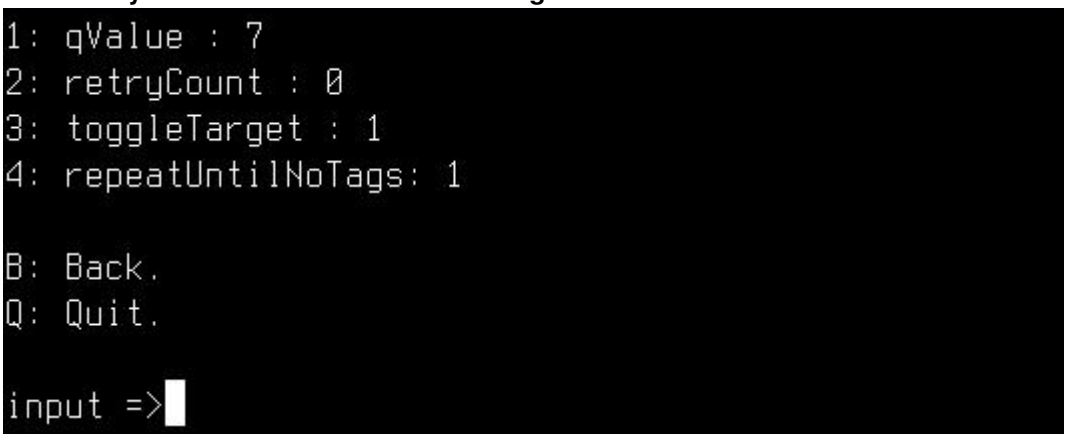

Figure 11 Set to FixedQ

2. DynamicQ (as figure 12): Adjusts the Q value based on the presence or absence of tags. The items as follows:

1: startQValue: The starting Q value to use. Valid values are 0 to 15, inclusive. minQValue <= startQValue <= maxQValue

2: minQValue: The minimum Q value to use. Valid values are 0 to 15, inclusive. minQValue <= startQValue <= maxQValue

3: maxQValue: The maximum Q value to use. Valid values are 0 to 15, inclusive. minQValue <= startQValue <= maxQValue

4: retryCount: Specifies the number of times to try another execution of the singulation algorithm for the specified session/target before either toggling the target (if toggleTarget is non-zero) or terminating the inventory/tag access operation. Valid values are 0-255, inclusive.

5: maxQueryRepCount: The maximum number of ISO 18000-6C QueryRep commands that will follow the ISO 18000-6C Query command during a single inventory round. Valid values are 0-255, inclusive.

6: toggleTarget: A flag that indicates if, after performing the inventory cycle for the specified target (i.e., A or B), if the target should be toggled (i.e., A to B or B to A) and another inventory cycle run. A non-zero value indicates that the target should be toggled. A zero value indicates that the target should not be toggled. Note that if the target is toggled, retryCount and repeatUntilNoTags will also apply to the new target.

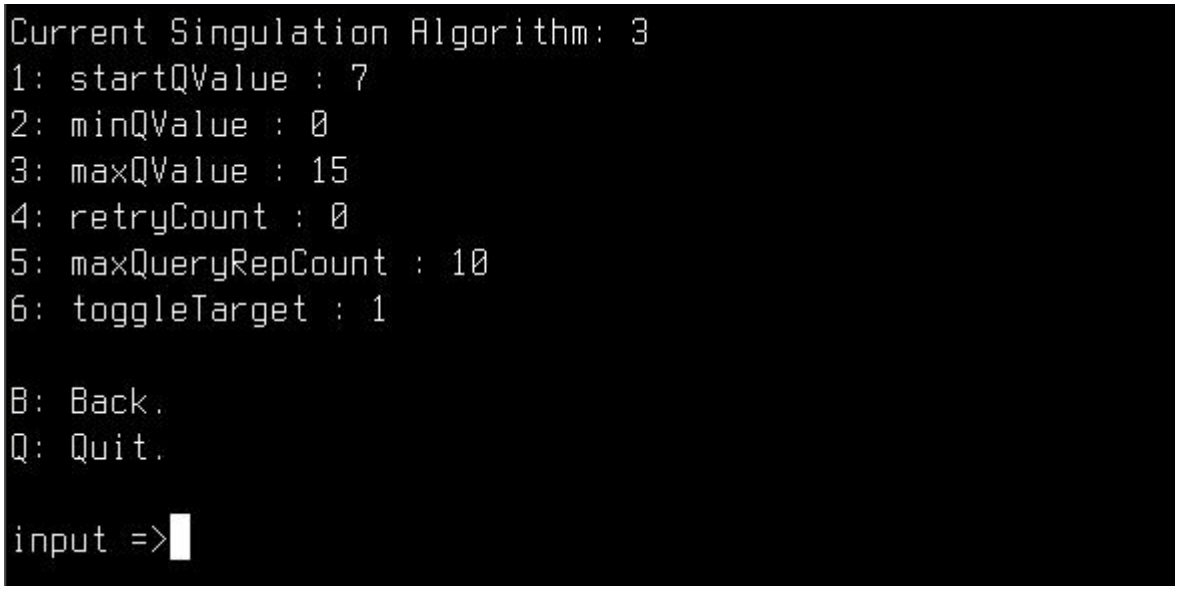

Figure 12 Set to DynamicQ

3. DynamicQAdjust (as figure 13): This algorithm modifies the previous dynamic Q algorithm by issuing ISO 18000-6C Query Adjust commands instead of ISO 18000-6C Query commands when adjusting the Q value. The items as follows:

   1: startQValue: The starting Q value to use. Valid values are 0 to 15, inclusive.
   minQValue <= startQValue <= maxQValue

   2: minQValue: The minimum Q value to use. Valid values are 0 to 15, inclusive.
   minQValue <= startQValue <= maxQValue

   3: maxQValue: The maximum Q value to use. Valid values are 0 to 15, inclusive.
   minQValue <= startQValue <= maxQValue

   4: retryCount: Specifies the number of times to try another execution of the singulation algorithm for the specified session/target before either toggling the target (if toggleTarget is non-zero) or terminating the inventory/tag access operation. Valid values are 0-255, inclusive.

   5: maxQueryRepCount: The maximum number of ISO 18000-6C QueryRep commands that will follow the ISO 18000-6C Query command during a single inventory round. Valid values are 0-255, inclusive.

   6: toggleTarget: A flag that indicates if, after performing the inventory cycle for the specified target (i.e., A or B), if the target should be toggled (i.e., A to B or B to A) and another inventory cycle run. A non-zero value indicates that the target should be toggled. A zero value indicates that the target should not be toggled. Note that if the target is toggled, retryCount and repeatUntilNoTags will also apply to the new target.

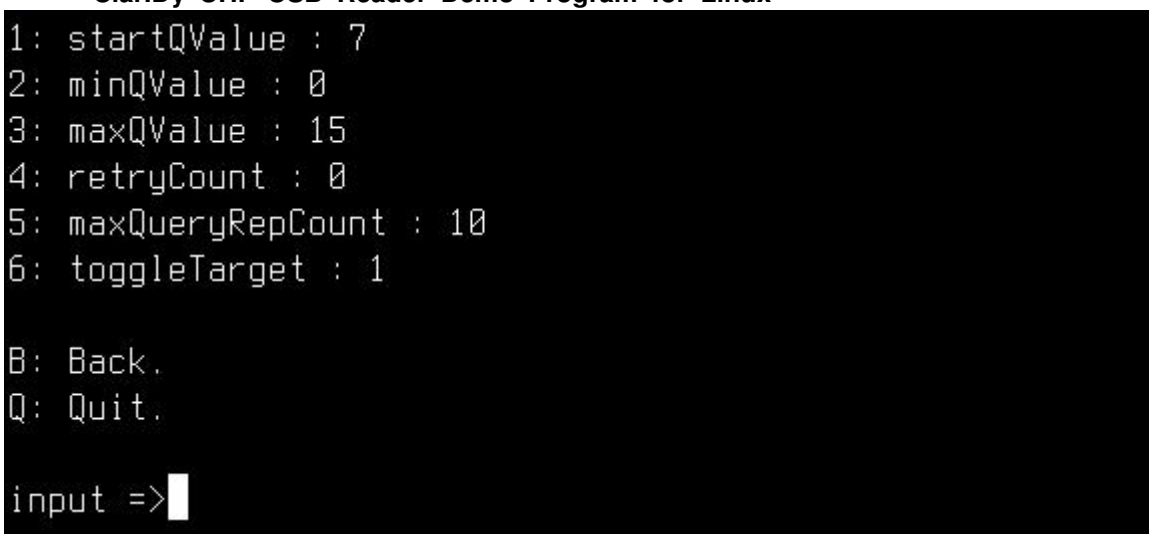

Figure 13 Set to DynamicQAdjust

4. DynamicQThreshold (as figure 14): This algorithm uses a Q-modification algorithm that allows the application to control the change of the Q-adjustment-threshold value. The items as follows:

   1: startQValue: The starting Q value to use. Valid values are 0 to 15, inclusive. minQValue <= startQValue <= maxQValue

   2: minQValue: The minimum Q value to use. Valid values are 0 to 15, inclusive. minQValue <= startQValue <= maxQValue

   3: maxQValue: The maximum Q value to use. Valid values are 0 to 15, inclusive. minQValue <= startQValue <= maxQValue

   4:retryCount: Specifies the number of times to try another execution of the singulation algorithm for the specified session/target before either toggling the target (if toggleTarget is non-zero) or terminating the inventory/tag access operation. Valid values are 0-255, inclusive.

   5: toggleTarget: A flag that indicates if, after performing the inventory cycle for the specified target (i.e., A or B), if the target should be toggled (i.e., A to B or B to A) and another inventory cycle run. A non-zero value indicates that the target should be toggled. A zero value indicates that the target should not be toggled. Note that if the target is toggled, retryCount and repeatUntilNoTags will also apply to the new target.

   6: thresholdMultiplier: The multiplier, specified in units of fourths (i.e., 0.25), that will be applied to the Q-adjustment threshold as part of the dynamic-Q algorithm. Valid values are 0-255, inclusive.

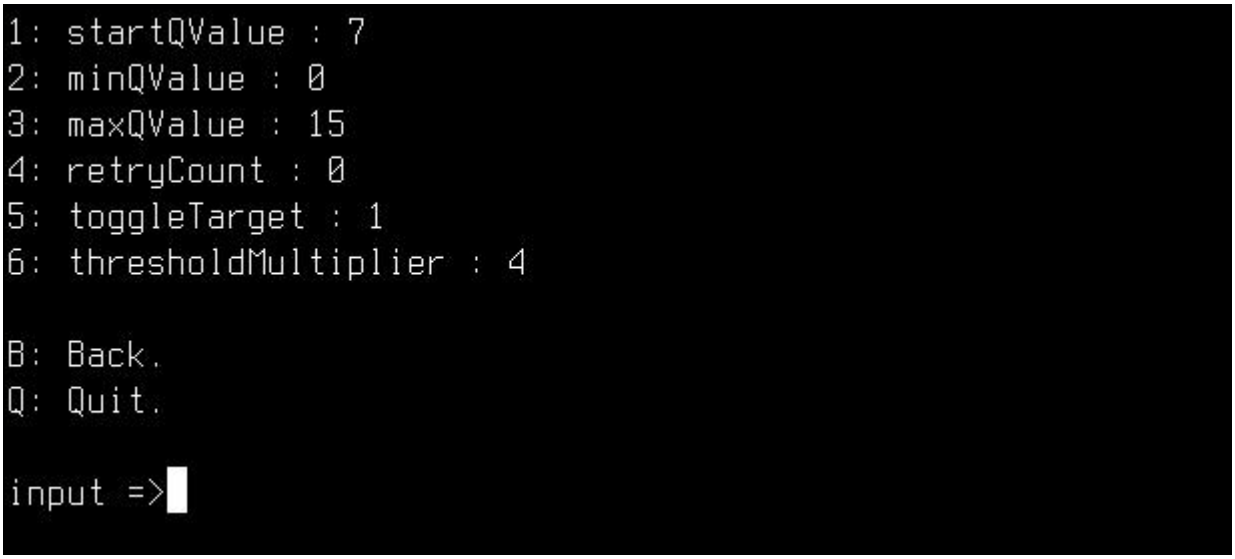

Figure 14 Set to DynamicQThreshold

3.2 Operation

The ClarIDy UHF Demo Program supports six Operations as follows (as figure 15):

1. Inventory
2. Read
3. Write
4. Filter Inventory
5. Filter Read
6. Filter Write

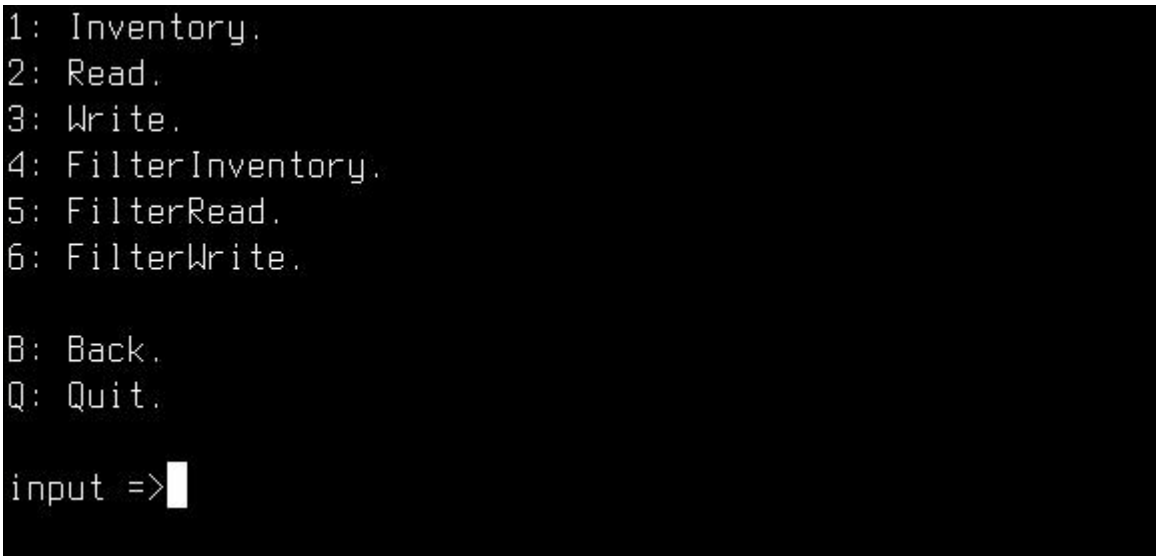

Figure 15 Operation

3.2.1 Inventory

Inventory step by step as the following instructions:

1. When the user select "1: Inventory" then showed the "Input interval time (second)", as figure 16.
2. Input interval time.
3. Place the RFID tag in the RF field of the ClarIDy UHF RFID Reader.
4. The PC (“PC”), EPC (EPC), CRC16 (CRC), number of reads (“Count”), and Receive Signal Strength Indicator (“RSSI”) will be shown on the table, as figure 16.

Note : interval time: Runs out the Inventory total time.

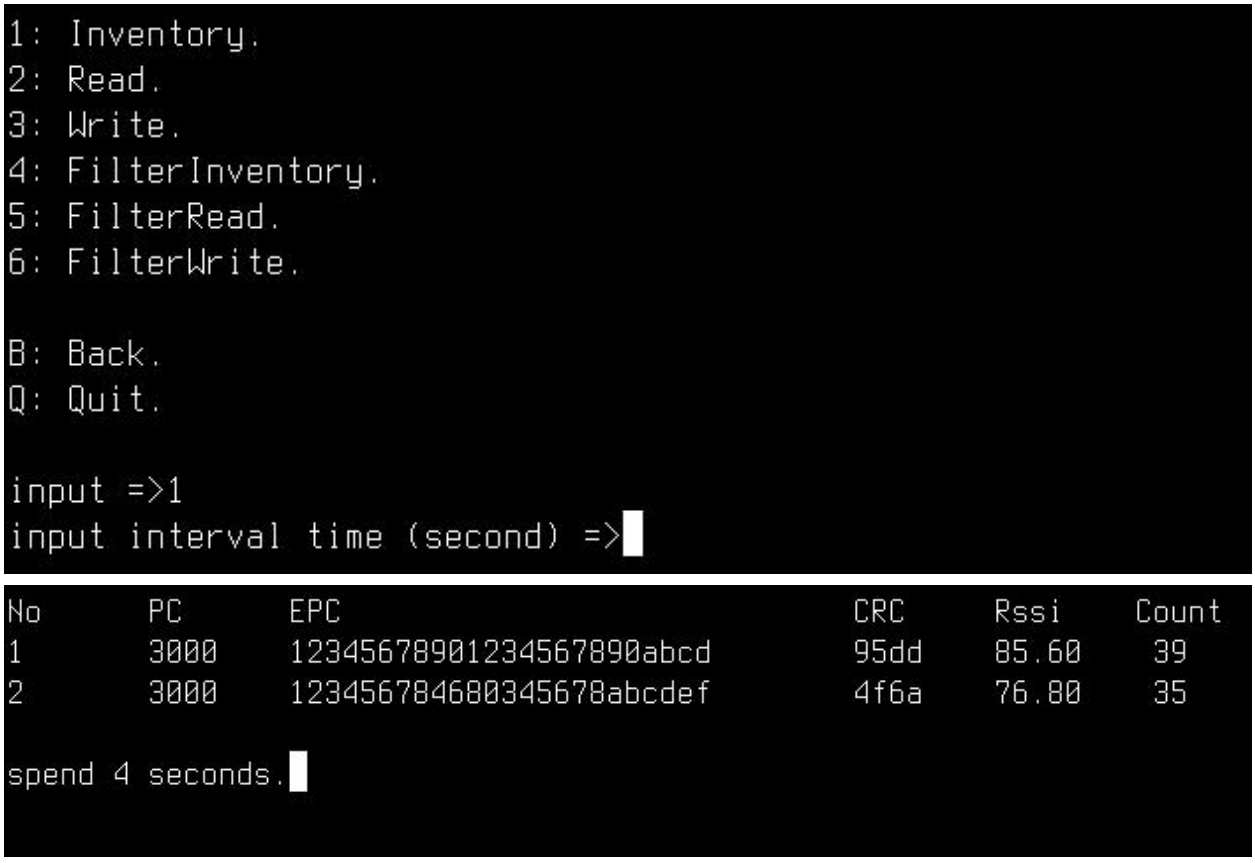

Figure 16 Inventory

3.2.2 Read

The Read page allows the user to configure the following items:

1. Read EPC (as figure 17): Read Tag EPC Data.

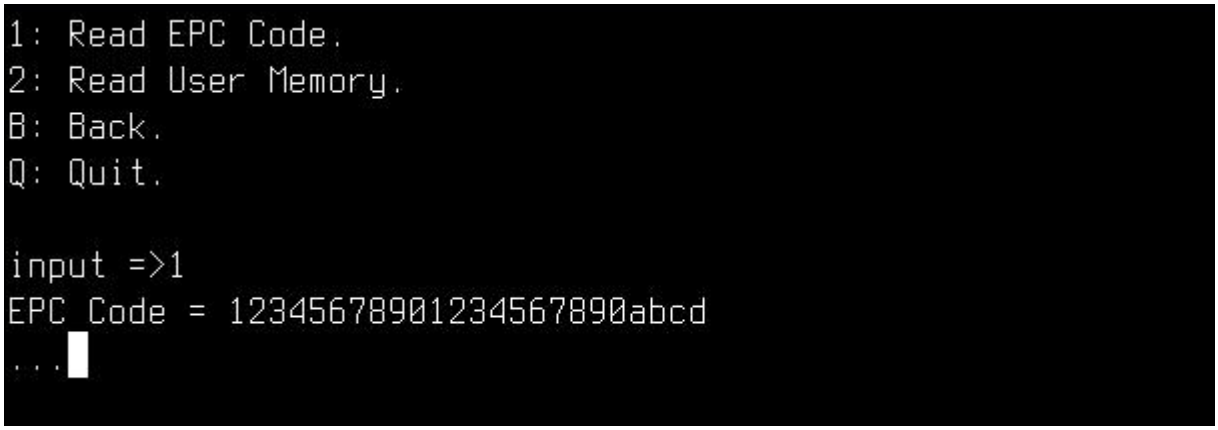

Figure 17 Read EPC

2. Read User Memory (as figure 18): Input “StartOffset” and “Count” then showed the user memory data.

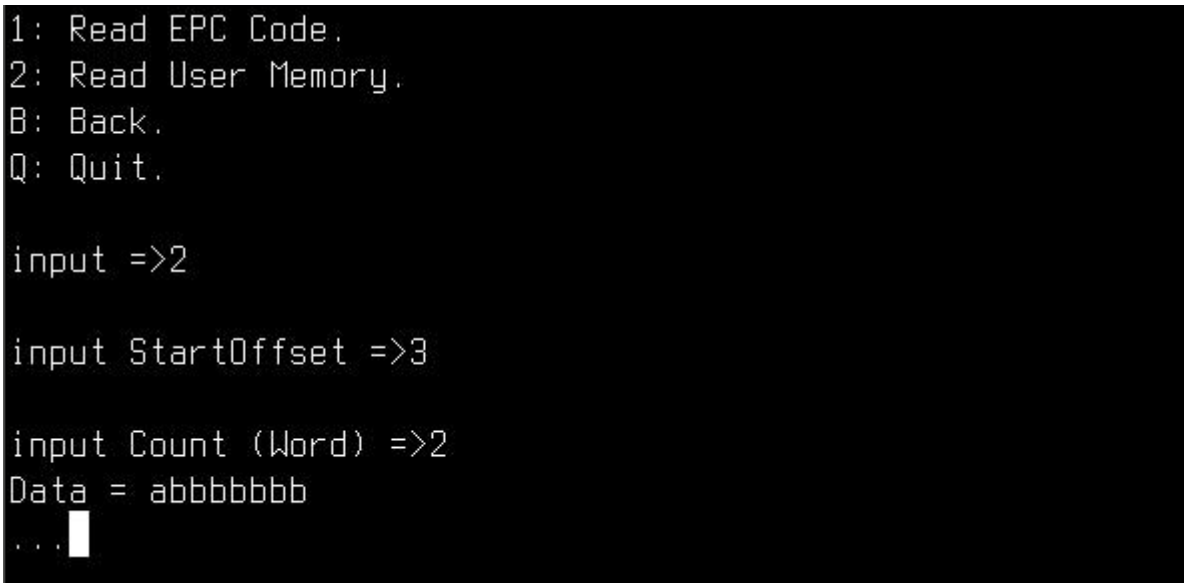

Figure 18 Read User Memory

Note : StartOffset: The offset of the first 16-bit word to read.

Count: The nubmer of 16-bit words to read.

3.2.3 Write

The Write page allows the user to configure the following items:

1. Write EPC (as figure 19): input 24 nibble to Write Tag EPC Data.

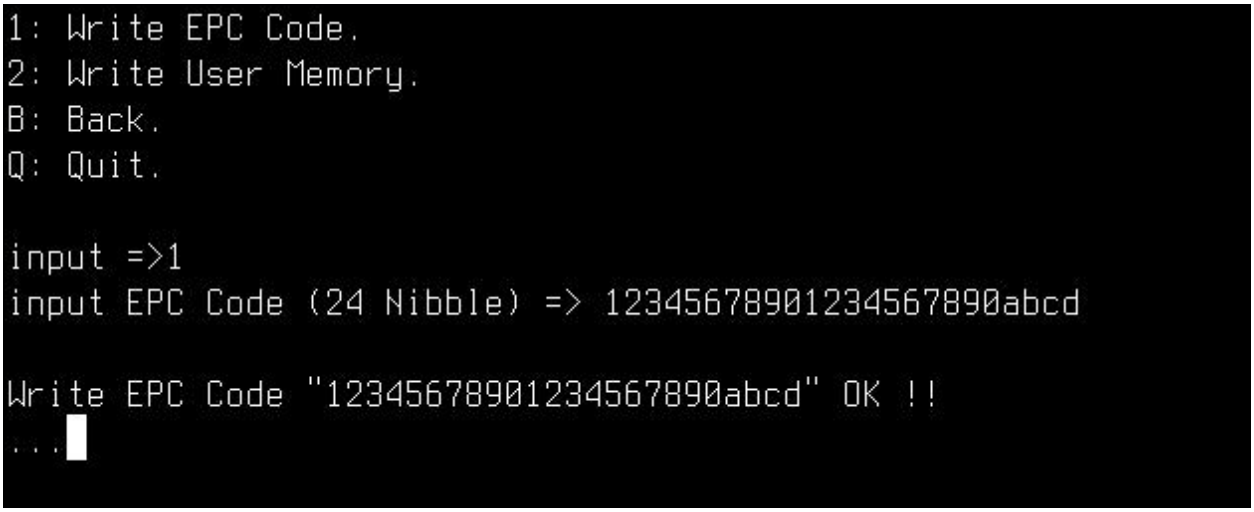

Figure 19 Write EPC

2. Write User Memory (as figure 20): Input “StartOffset”, “Count” and “WriteData” then write the data to user memory.

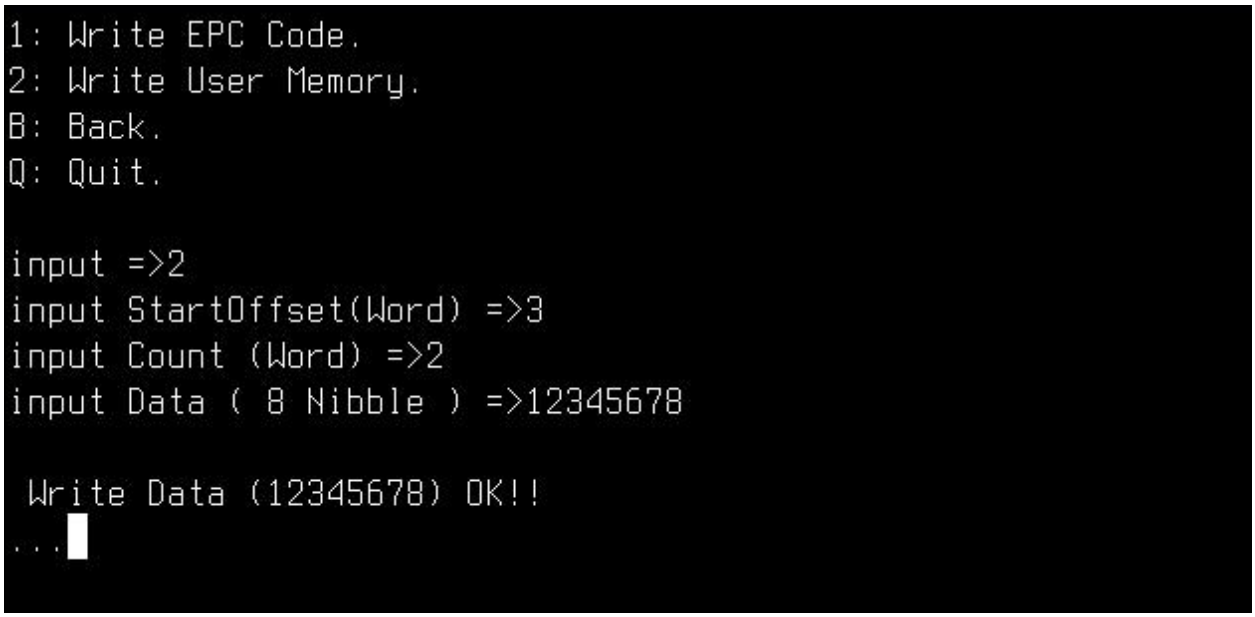

Figure 20 Write User Memory

Note : StartOffset: The offset of the first 16-bit word to write.

Count: The nubmer of 16-bit words to write.

WriteData: The Write data length is Count * 4 nibble.

3.2.4 Filter Inventory

Follow the instructions step by step as following:

1. When the user select "4: Filter Inventory", the system shows the input dialogs: “Mask bank”, “Mask start”, “Mask Length”, “Mask data”, “match or unmatched” and “interval time”, as figure 21.
2. Place the RFID tag in the RF field of the ClarIDy UHF RFID Reader.
3. The PC (“PC”), EPC (EPC), CRC16 (CRC), number of reads (“Count”), and Receive Signal Strength Indicator (“RSSI”) will be shown on the table, as Figure 21.

```
1: Inventory.
2: Read.
3: Write.
4: FilterInventory.
5: FilterRead.
6: FilterWrite.

B: Back.
Q: Quit.

input =>4
input Mask bank =>1
input Mask start offset (byte) =>4
input Mask Length (byte) =>2
input Mask Data ( 4 Nibble ) =>1234
input match or unmatched ( 2: unmatched  3: match) =>3
input interval time (second) =>20
```

```
No      PC      EPC                                 CRC     Rssi    Count
1       3000    123456789012345678900abcd           95dd    88.00    74
2       3000    123400000000000000000000            9217    77.60    42

spend 7 seconds.
```

Figure 21 Filter Inventory

Note : Mask bank: The memory bank to match against (0: Reserved, 1: EPC, 2: TID, 3: USER Memory).

Mask start offset: The offset of the first byte to match.

Mask Length: The number of bits in the mask.

Mask data: The byte pattern to match.

Match or unmatched: selected to match or unmatched (0: none 2: unmatched, 3: match).

Interval time: Runs out the FlterInventory total time

3.2.5 Filter Read

Follow the instructions step by step as following:

1. Place the RFID tag in the RF field of the ClarIDy UHF RFID Reader.
2. Selects "5: Filter Read" , the system shows the input dialogs:
   "Mask bank", "Mask start", "Mask Length", "Mask data", "match or unmatched", "Start offset" and "Count", as figure 22.
3. The system will show the filtered tag's memory data, as figure 20.

Note : Mask bank: The memory bank to match against (0: Reserved, 1: EPC, 2: TID, 3: USER Memory).

Mask start offset: The offset of the first byte to match.

Mask Length: The number of bits in the mask.

Mask data: The byte pattern to match.

Match or unmatched: selected to match or unmatched (0: none 2: unmatched, 3: match).

Memory bank: The RFID tag's memory bank (0: Reserved, 1: EPC, 2: TID, 3: USER Memory).

Start offset: The offset of the first 16-bit word to read.

Count: The nubmer of 16-bit words to read.

```
1: Inventory.
2: Read.
3: Write.
4: FilterInventory.
5: FilterRead.
6: FilterWrite.

B: Back.
Q: Quit.

input =>5
input Mask bank =>1
input Mask start offset (byte) =>4
input Mask Length (byte) =>3
input Mask Data ( 6 Nibble ) =>123400
input match or unmatched ( 2: unmatched  3: match ) =>3
input Memory Bank =>3

input StartOffset (word)=>2

input Count (Word) =>2
Data = 00000000
...
```

Figure 22 Filter Read

3.2.6 Filter Write

Follow the instructions step by step as following:

1. Place the RFID tag in the RF field of the ClarIDy UHF RFID Reader.
2. Selects "6: Filter Write" , the system shows the input dialogs:
   "Mask bank", "Mask start", "Mask Length", "Mask data", "match or unmatched", "Start offset", "Count" and "WriteData", as figure 23.
3. The system will show the write OK or failed, as figure 23.

Note : Mask bank: The memory bank to match against (0: Reserved, 1: EPC, 2: TID, 3: USER Memory).

Mask start offset: The offset of the first byte to match.

Mask Length: The number of bits in the mask.

Mask data: The byte pattern to match.

Match or unmatched: selected to match or unmatched (0: none 2: unmatched, 3: match).

Memory bank: The RFID tag's memory bank (0: Reserved, 1: EPC, 2: TID, 3: USER Memory).

Start offset: The offset of the first 16-bit word to Write.

Count: The number of 16-bit words to write.

WriteData: The Write data length is Count * 4 nibble.

```
1: Inventory.
2: Read.
3: Write.
4: FilterInventory.
5: FilterRead.
6: FilterWrite.

B: Back.
Q: Quit.

input =>6
input Mask bank =>1
input Mask start offset (byte) =>4
input Mask Length (byte) =>3
input Mask Data ( 6 Nibble ) =>123400
input match or unmatched ( 2: unmatched  3: match ) =>3
input Memory Bank =>3
input StartOffset (Word)=>2
input Count (Word) =>2
input Data ( 8 Nibble ) =>1234abcd

 Write Data (1234abcd) OK!!
...
```

Figure 23 Filter Write

3.2.7 Lock Tag

Follow the instructions step by step as following:

1. Place the RFID tag in the RF field of the ClarIDy UHF RFID Reader.
2. Selects "7: Lock Tag" , the system shows the input dialogs:
   “Mask bank”, “Mask start”, “Mask Length”, “Mask data” and “match or unmatched”.
3. Select “KillPassword”,” AccessPassword”, “EPC Memory Bank”, “TID Memory Bank” and “User Memory Bank” state.
4. Input the AccessPassword.
5. The system will show the success or failed.

```
input Mask bank =>1
input Mask start offset (byte) =>4
input Mask Length (byte) =>12
input Mask Data ( 24 Nibble ) =>1234567890abcdef12345678
input match or unmatched ( 0: none 2: unmatched  3: match ) =>3
```

Figure 24 Set Filter Mask

```
(0): Accessible
(1): Accessible Permanently
(2): Secured Accessible
(3): Not Accessible Permanently
(4): No Change
Select killPassword state=>4
(0): Accessible
(1): Accessible Permanently
(2): Secured Accessible
(3): Not Accessible Permanently
(4): No Change
Select AccessPassword state=>4
(0): Writeable
(1): Writeable Permanently
(2): Secured Writeable
(3): Not Writeable Permanently
(4): No Change
Select EPC Memory Bank state=>4
(0): Writeable
(1): Writeable Permanently
(2): Secured Writeable
(3): Not Writeable Permanently
(4): No Change
Select TID Memory Bank state=>4
(0): Writeable
(1): Writeable Permanently
(2): Secured Writeable
(3): Not Writeable Permanently
(4): No Change
Select User Memory Bank state=>3
Input the AccessPassword=>0
Lock Tag Success!!...
```

Figure 25 Set Lock Tag State

Note : Mask bank: The memory bank to match against (0: Reserved, 1: EPC, 2: TID, 3: USER Memory).

Mask start offset: The offset of the first byte to match.

Mask Length: The number of bits in the mask.

Mask data: The byte pattern to match.

Match or unmatched: selected to match or unmatched (0: none 2: unmatched, 3: match).

AccessPassword: The access password for the tags. A value of zero indicates no access password.

The state of “KillPassword” and “AccessPassword” as follows:

Accessible: The password may be read and written when the tag is in either the open or secured states.

Accessible Permanently: The password may be read and written when the tag is in either the open or secured states and this access permission should be set permanently.

Secured Accessible: The password may be read or written only when the tag is in the secured state.

Not Accessible Permanently: The password may not be read or written and this access permission should be set permanently.

No Change: The password’s access permission should remain unchanged.

The state of “EPC Memory Bank”, “TID Memory Bank” and “User Memory Bank”as follows:

Writeable: The memory bank is writeable when the tag is in either the open or secured states.

Writeable Permanently: The memory bank is writeable when the tag is in either the open or secured states and this access permission should be set permanently.

Secured Writeable: The memory bank is writeable only when the tag is in the secured state.

Not Writeable Permanently: The memory bank is not writeable and this access permission should be set permanently.

No Change: The memory bank’s access permission should remain unchanged.

3.2.8 Kill Tag

Follow the instructions step by step as following:

1. Place the RFID tag in the RF field of the ClarIDy UHF RFID Reader.
2. Selects "8: Kill Tag" , the system shows the input dialogs:
   “Mask bank”, “Mask start”, “Mask Length”, “Mask data” and “match or unmatched”.
3. Input the AccessPassword and KillPassword.
4. The system will show the success or failed.

Note : Mask bank: The memory bank to match against (0: Reserved, 1: EPC, 2: TID, 3: USER Memory).

Mask start offset: The offset of the first byte to match.

Mask Length: The number of bits in the mask.

Mask data: The byte pattern to match.

Match or unmatched: selected to match or unmatched (0: none 2: unmatched, 3: match).

AccessPassword: The access password for the tags. A value of zero indicates no access password.

KillPassword: The kill password for the tags.

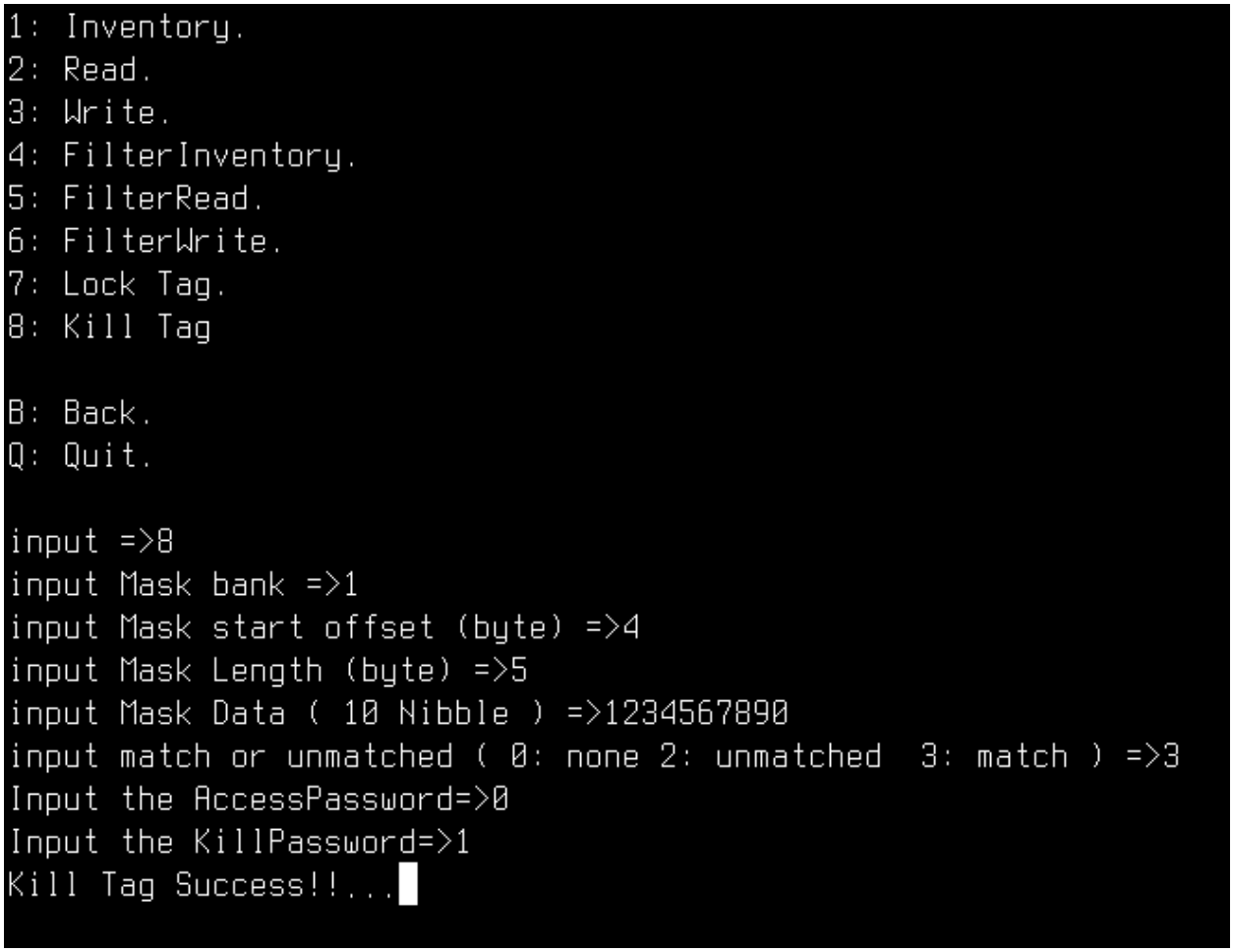

Figure 26 Kill Tag

4. Error Code

The following lists provide error codes of demo program. These values are defined in the description.

Code (dec)	Description
0	Success
-9999	Attempted to open a reader that is already open
-9998	Buffer supplied is too small
-9997	General failure
-9996	Failed to load reader bus driver
-9995	Library cannot use version of reader bus driver present on system
-9994	Operation cannot be performed while library is in emulation mode
-9993	Antenna number is invalid
-9992	Reader handle provided is invalid
-9991	One of the parameters to the function is invalid
-9990	Attempted to open a non-existent reader
-9989	Library has not been successfully initialized
-9988	Function not supported
-9987	Operation was cancelled by call to cancel operation, close reader, or shut down the library
-9986	Library encountered an error allocating memory
-9985	The operation cannot be performed because the reader is currently busy
-9984	The underlying reader module encountered an error
-9983	The reader has been detached from the system
-9982	The RFID library function is not allowed at this time.
-9981	The reader module's MAC firmware is not responding to requests.
-9980	The MAC firmware encountered an error while initiating the nonvolatile memory update. The MAC firmware will return to its normal idle state without resetting the reader module.
-9979	An attempt was made to write data to an address that is not in the valid range of reader module nonvolatile memory addresses.
-9978	The MAC firmware encountered an error while trying to write to the reader module's nonvolatile memory region.
-9977	The underlying transport layer detected that there was an overflow error resulting in one or more bytes of the incoming data being dropped. The operation was aborted and all data in the pipeline was flushed.

-7999	Fail to find reader
-7998	Fail to allocate memory
-7997	Write Data failure
-7996	Read Data failure
-7995	Lock Tag failure
-7994	Kill Tag failure