



Swing-U

Developer Manual

Model Number: NTRM-U-2

1. Contents

1.	Contents.....	2
2.	Revision History.....	6
3.	Firmware upgrade	8
3.1.	The prepare items for firmware update	8
3.2.	Installing USB driver for Swing-U.....	9
3.3.	Installing firmware update program	10
3.4.	Booting up F/W update mode	11
3.5.	Run F/W update program	12
3.6.	F/W upgrade	13
4.	Communication protocol.....	18
4.1.	Communication Interface.....	18
4.1.1.	USB Interface.....	18
4.1.2.	Bluetooth Interface.....	19
4.2.	Packet Format.....	20
4.3.	Packet Data.....	21
4.4.	Inventory and search mode.....	22
4.4.1.	Inventory start	23
4.4.2.	Inventory Stop	23
4.4.3.	Inventory Notification	24
4.4.4.	Search target tag list and parameters.....	26
4.4.5.	Finding RF threshold	27
4.4.6.	Finding RF decrement step unit.....	28
4.4.7.	Exit to Finding mode.....	28
4.4.8.	Access Password Set.....	29
4.4.9.	Session	29
4.4.10.	Flag	30
4.5.	RFID tag memory access.....	31
4.5.1.	Tag memory read.....	31
4.5.2.	Tag memory write	31
4.5.3.	Tag memory select	32
4.5.4.	Tag memory lock.....	33
4.5.5.	Error Code	35
4.5.6.	Example.....	36

4.6.	Swing-U status control	38
4.6.1.	Version.....	38
4.6.2.	RF output power of reader.....	38
4.6.3.	Buzzer volume	39
4.6.4.	Continuous read mode	40
4.6.5.	Tag report mode	41
4.6.6.	Battery level	42
4.6.7.	Battery Threshold.....	43
4.6.8.	Battery charging Status.....	43
4.6.9.	Swing-U filter size setting.....	44
4.6.10.	Swing-U Read EPC max Length set	45
4.6.11.	Power-off notification.....	45
4.6.12.	LCD control mode.....	46
4.6.13.	Menu enable	47
4.6.14.	Inventory timeout	47
4.6.15.	Inventory mode get	48
4.6.16.	Read type	49
4.6.17.	RF output power of reader.....	50
4.6.18.	Device name.....	51
4.6.19.	Serial number.....	52
4.6.20.	Language.....	52
4.7.	User button notification.....	54
4.7.1.	User button click event	54
4.7.2.	User button long-click event.....	54
4.8.	Tag list synchronization.....	54
4.8.1.	Count of stored tag list.....	55
4.8.2.	Host tag count send Swing-U	56
4.8.3.	Stored tag list.....	56
5.	.NET C# Library API	58
5.1.	Overview	58
5.2.	Revision History	58
5.3.	Swing-U Library contents.....	58
5.4.	Library Event.....	59
5.4.1.	NotifyParameterChanged	59

5.4.2.	NotifyButtonEvent.....	60
5.4.3.	NotifyInventory	61
5.4.4.	NotifyInventoryBCD	61
5.4.5.	NotifyTagFound.....	62
5.4.6.	NotifyReadEvent	62
5.4.7.	NotifyStatusCheck.....	63
5.4.8.	NotifyError	63
5.5.	Library Parameters.....	65
5.5.1.	IsOpen	65
5.5.2.	PortName	65
5.6.	Functions.....	66
5.6.1.	Connection control.....	68
5.6.2.	Hardware parameter control.....	70
5.7.	UHF RFID control	94
5.7.1.	InventoryStart.....	94
5.7.2.	InventoryStop.....	94
5.7.3.	MemoryRead	95
5.7.4.	MemoryWrite	96
5.7.5.	MemorySelect	97
5.7.6.	MemoryLock	98
5.8.	Tag List control.....	100
5.8.1.	TagListAdd.....	100
5.8.2.	Access Password Set.....	100
5.8.3.	TagListClear.....	101
6.	Active-X Library API.....	102
6.1.	Overview	102
6.2.	Revision History	102
6.3.	Swing-U Library contents.....	102
6.4.	Installation of Active-X	103
6.4.1.	Run internet explorer	103
6.4.2.	Internet explorer setting	103
6.4.3.	Install from web page	105
6.5.	Library Event.....	107
6.5.1.	NotifyParameterChanged	107

6.5.2.	NotifyButtonEvent.....	109
6.5.3.	NotifyInventory	110
6.5.4.	NotifyError	111
6.6.	Library Functions.....	112
6.6.1.	Connection control.....	113
6.6.2.	Hardware parameter control.....	114
6.6.3.	UHF RFID control	121
6.6.4.	Tag List control.....	122
7.	Android Library API	123
7.1.	Overview	123
7.2.	Revision History	123
7.3.	Swing-U Library contents.....	123
7.4.	Library Thread	124
7.4.1.	AcceptThread	124
7.4.2.	ConnectThread	125
7.4.3.	ConnectedThread	126
7.4.4.	ConnectedThread.....	126
7.5.	Functions.....	127
7.5.1.	Connection control.....	127
7.5.2.	Hardware parameter control.....	128
7.6.	UHF RFID control	138
7.6.1.	InventoryStart.....	138
7.6.2.	InventoryStop.....	138
7.6.3.	InventoryClear.....	138
7.6.4.	MemoryRead	139
7.6.5.	MemoryReadsync.....	139
7.6.6.	MemoryWrite	140
7.6.7.	MemoryWritesync	140
7.6.8.	MemorySelect	141
7.6.9.	MemorySelectSync	141
7.6.10.	MemoryLock	142

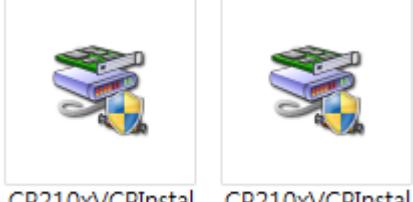
2. Revision History

Version	Date	Description
1.1.5	2016-08-26	<p>Following chapters are changed</p> <ul style="list-style-type: none"> 4.4.1. Inventory start 4.4.4. Search target tag list and parameters <p>Following chapters are added.</p> <ul style="list-style-type: none"> 4.4.10. Flag 4.5.2. Tag memory write <ul style="list-style-type: none"> - Data Response 4.6.7. Battery Threshold 4.8.3. Stored tag list <ul style="list-style-type: none"> - Response End 5.6.2. Hardware parameter control <ul style="list-style-type: none"> - Flag <p>F/W 1.2.4.9 Ver.</p>
1.1.4	2016-07-15	<p>Following chapters are added.</p> <ul style="list-style-type: none"> 4.4.9 Session 4.6.17. Device name 4.6.19. Language 5.6.2. Hardware parameter control <ul style="list-style-type: none"> - Language - Session 7.5.2. Hardware parameter control <ul style="list-style-type: none"> - Language - Session <p>F/W 1.2.4.8 Ver.</p>
1.1.3	2016-06-16	<p>Following chapters are added.</p> <ul style="list-style-type: none"> 4.4. Inventory and search mode
1.1.2	2016-06-14	<p>Following chapters are added.</p> <ul style="list-style-type: none"> 4.5.3. Tag memory select 5.7.5. MemorySelect 7.6.8. MemorySelect

		7.6.9. MemorySelectSync
1.1.1	2016-06-09	<p>Following chapters are added.</p> <p>4.6.19. Language</p> <p>5.6.2. Hardware parameter control</p> <p>- Language</p> <p>7.5.2. Hardware parameter control</p> <p>- Language</p> <p>F/W 1.2.4.6 Ver.</p>
1.1	2016-05-30	<p>Following chapters are added.</p> <p>4.4.8 Access password</p> <p>4.5.3 Tag memory lock</p> <p>4.6.8 Filter size set</p> <p>4.6.9 EPC max length set</p> <p>4.6.12 Menu enable</p> <p>4.6.13 Inventory timeout</p> <p>4.6.14 Read inventory mode</p> <p>4.6.15 Read type</p> <p>4.6.17 Device name</p> <p>4.6.18 Serial number</p>
1.0.1	2015-12-27	First release

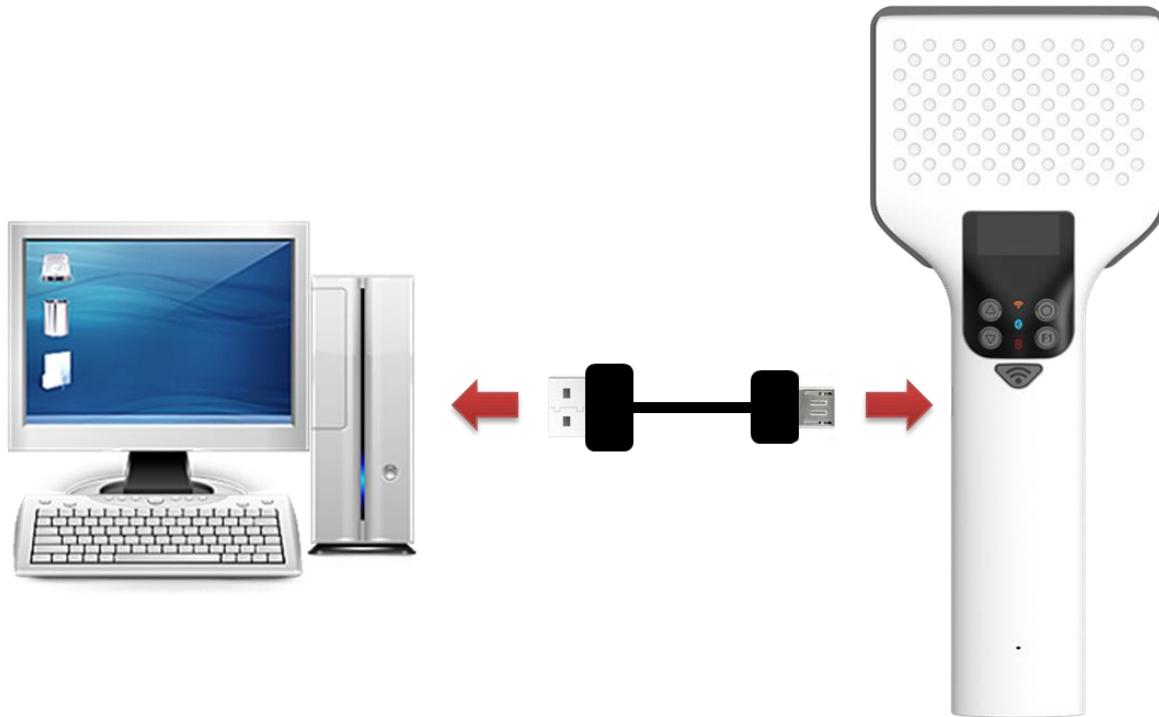
3. Firmware upgrade

3.1. The prepare items for firmware update

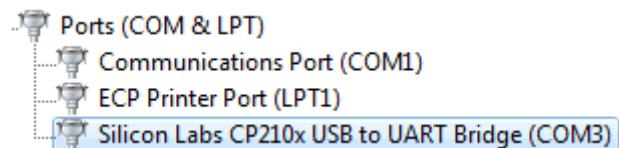
Item	Image
Micro 5pin USB cable	
Swing-U for update	
Installing USB driver for Swing-U	 CP210xVCPInstaller_x64.exe CP210xVCPInstaller_x86.exe
Installing firmware update program	 Flash_Loader_Demonstrator_V2.1.0_Setup.exe
The firmware file for Swing-U	 SwingH-REV1_0_0.hex

3.2. Installing USB driver for Swing-U

Using USB data cable, connect Swing-U to Host PC



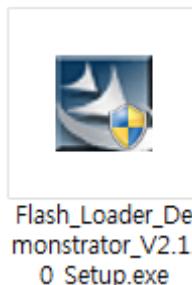
If USB device driver is already installed, then you can find COM port number on device manager. Please memo the COM port number of Swing-U for f/w update.



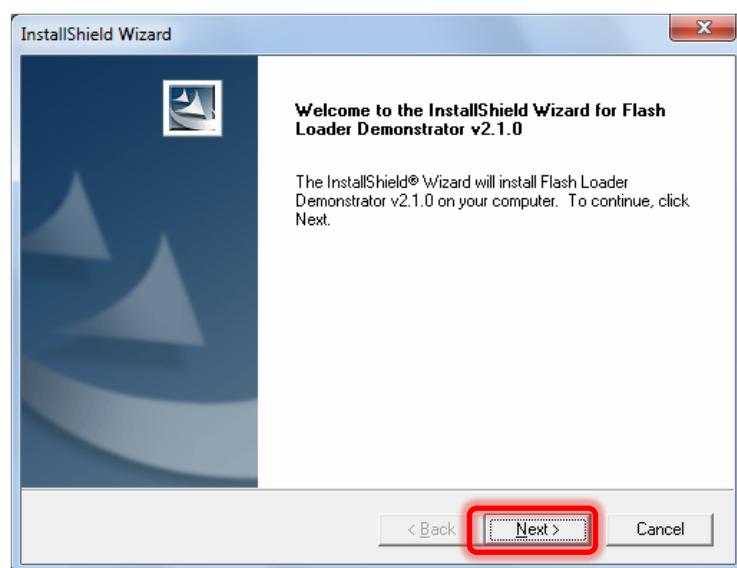
※ Notice

USB is normally detected on PC when Swing-U is powered-off.

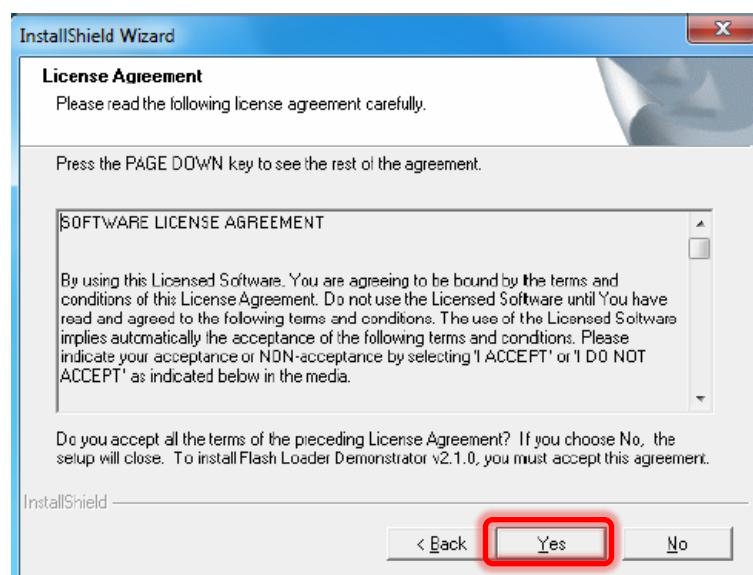
3.3. Installing firmware update program



For installation, start the setup program.

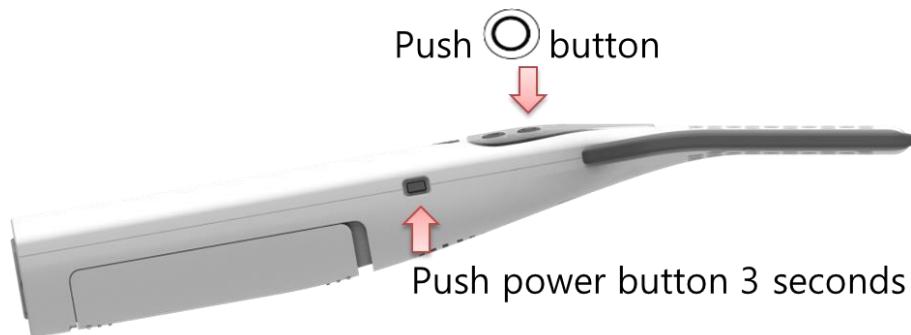


Click "Next" button.



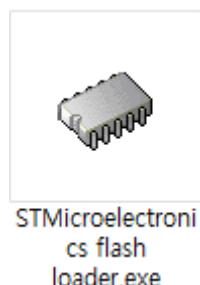
Click "Yes" button.

3.4. Booting up F/W update mode

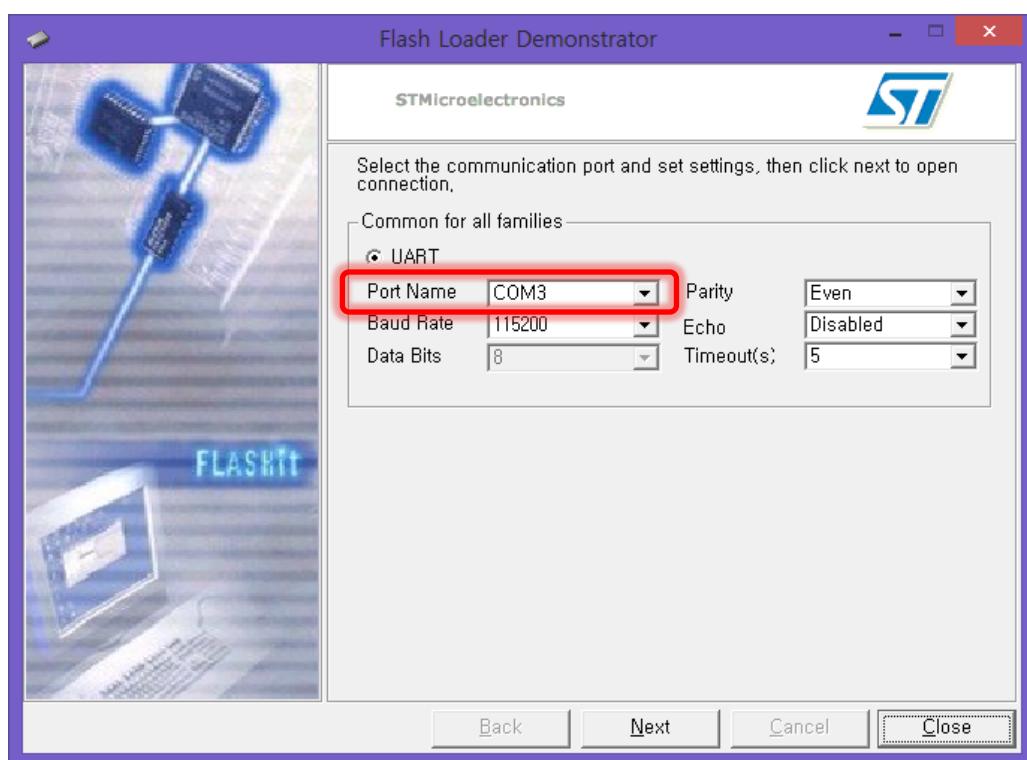


1. Turn off the power of Swing-U.
2. Push and holding button.
3. Turn on the power with 3 seconds long click of power button
4. There are no indications for F/W update mode.

3.5. Run F/W update program

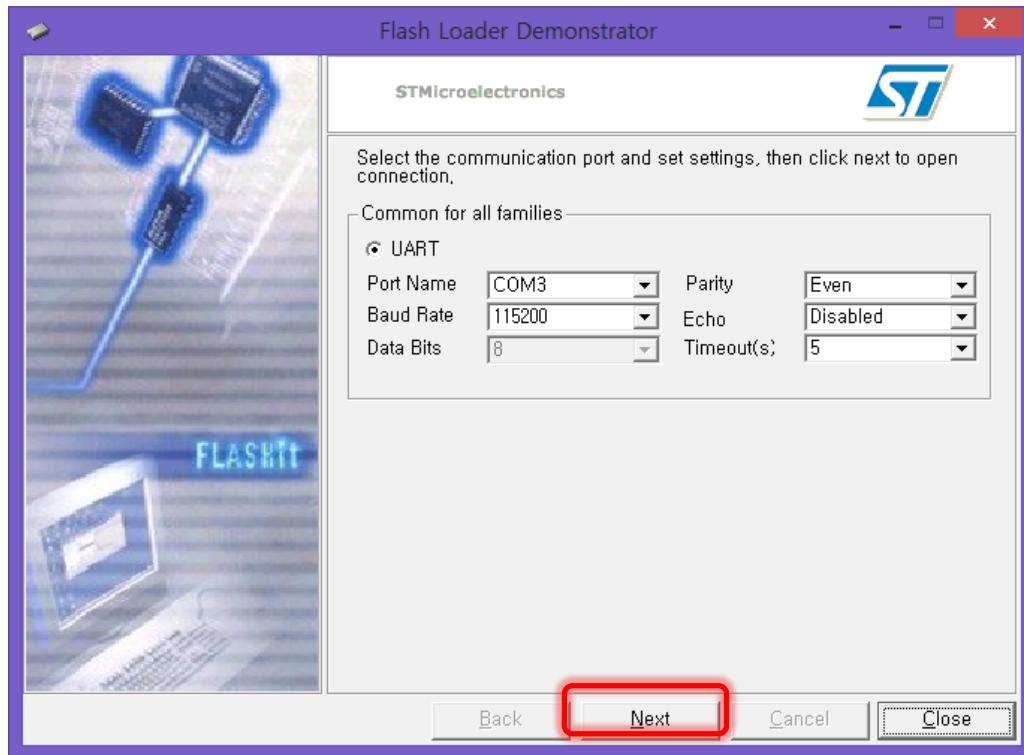


Run "STMicroelectronics flash loader.exe" program.

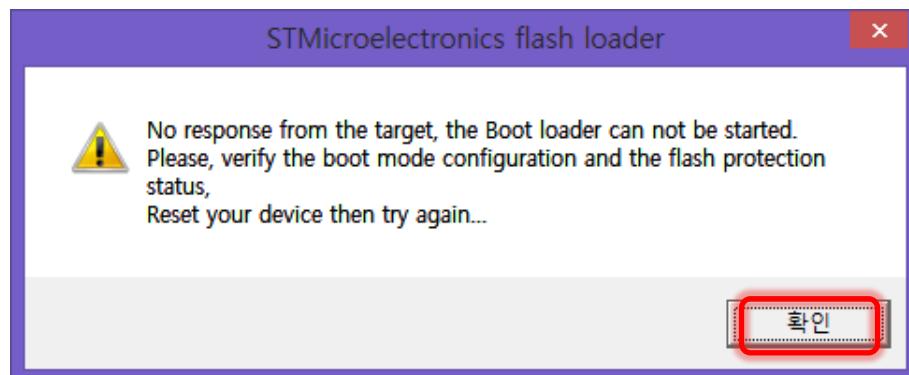


Then, select your COM-port for connection.

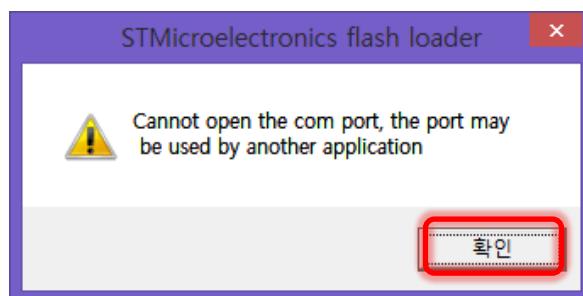
3.6. F/W upgrade



Click "Next" button.

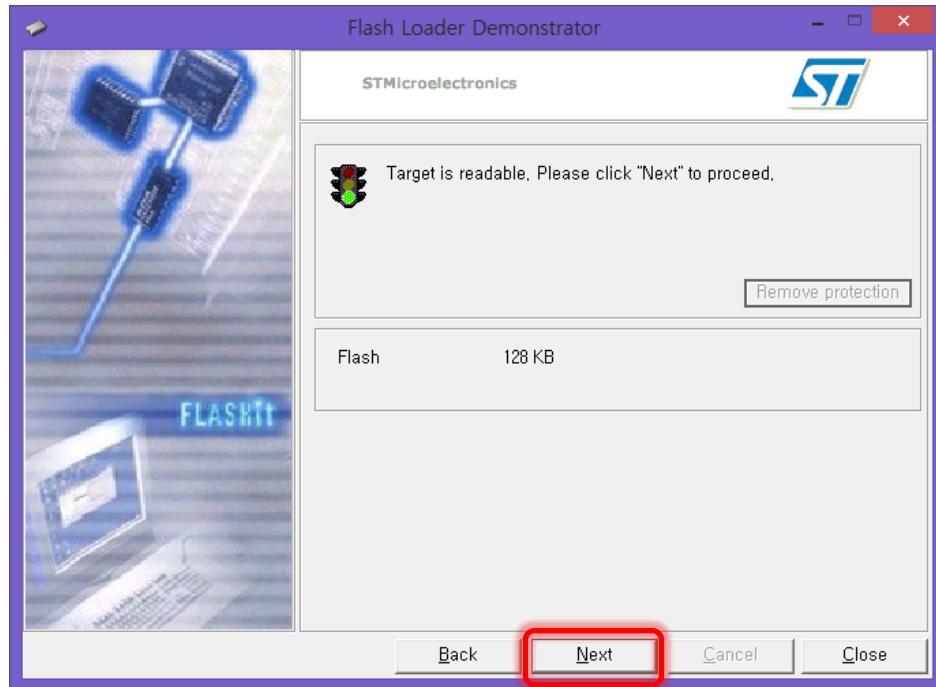


If above window is display, Swing-U is not F/W update mode. Retry chapter 4.1.

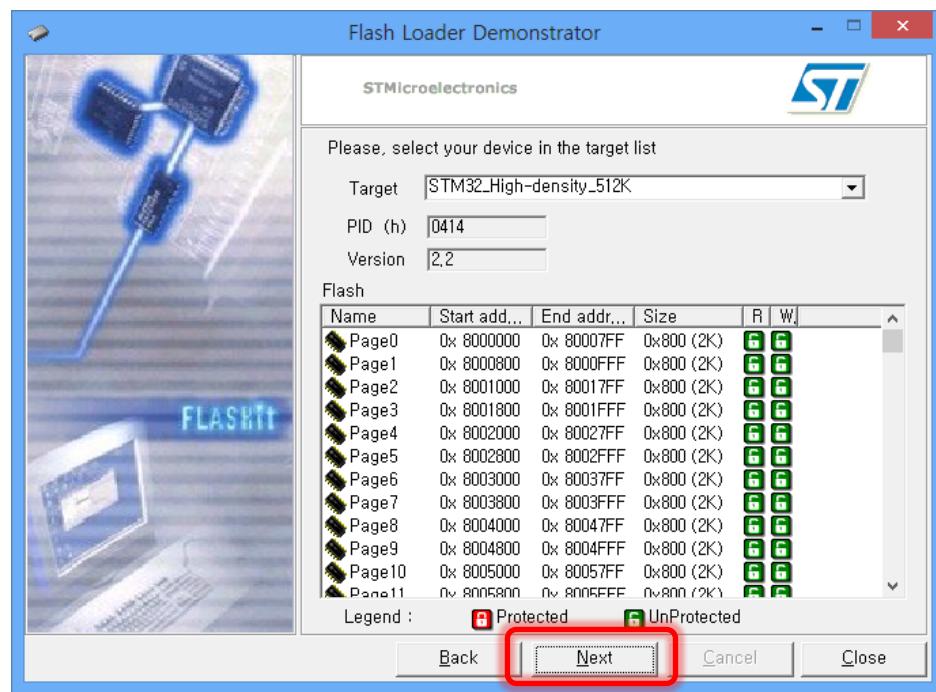


If COM-port of Swing-U is using in another program, above window will be display. Close all other serial communication program.

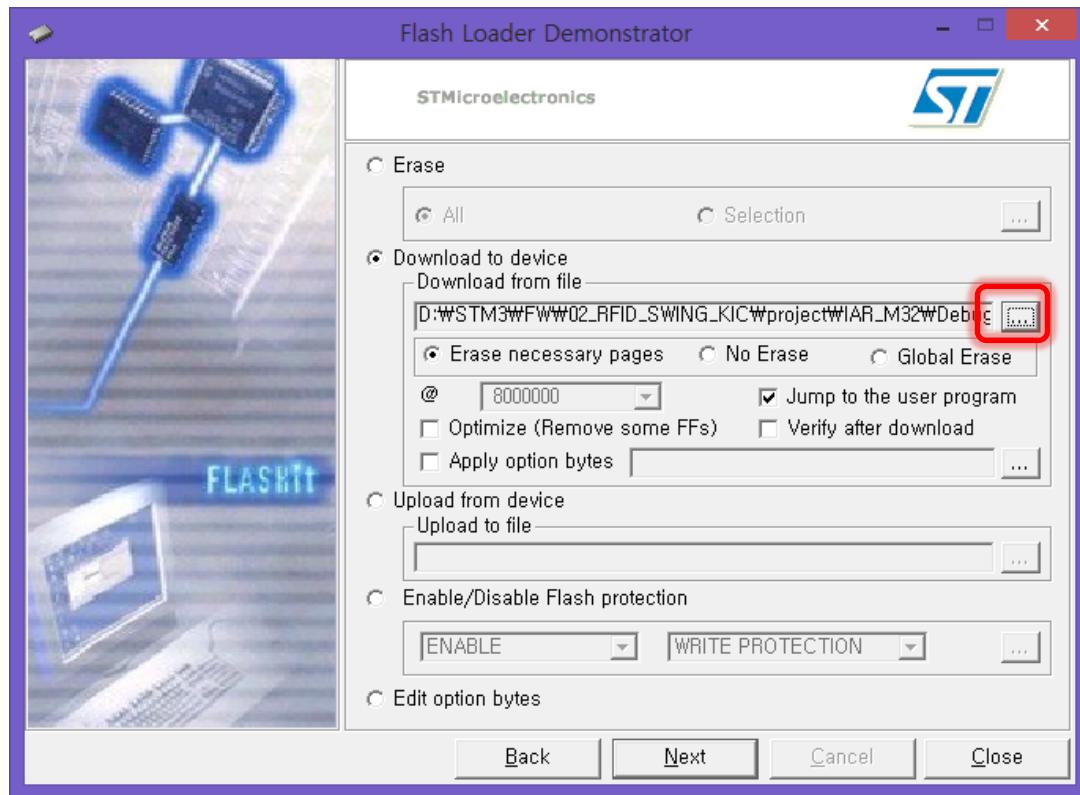
If Swing-U successfully boot up to F/W update mode, next window will be display.



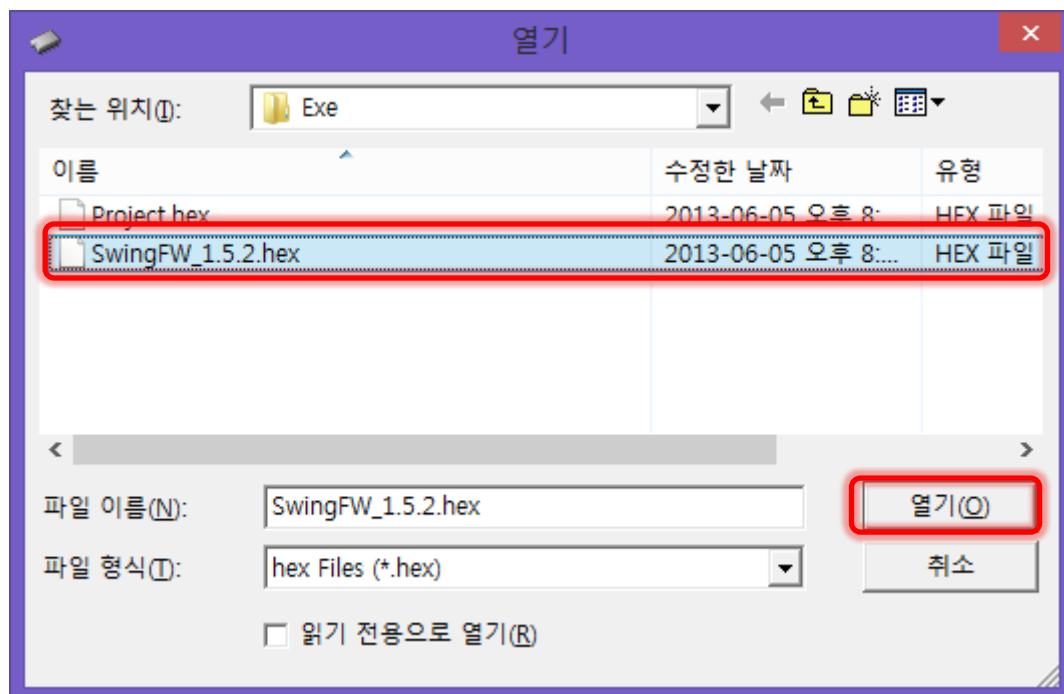
Click "Next" button.



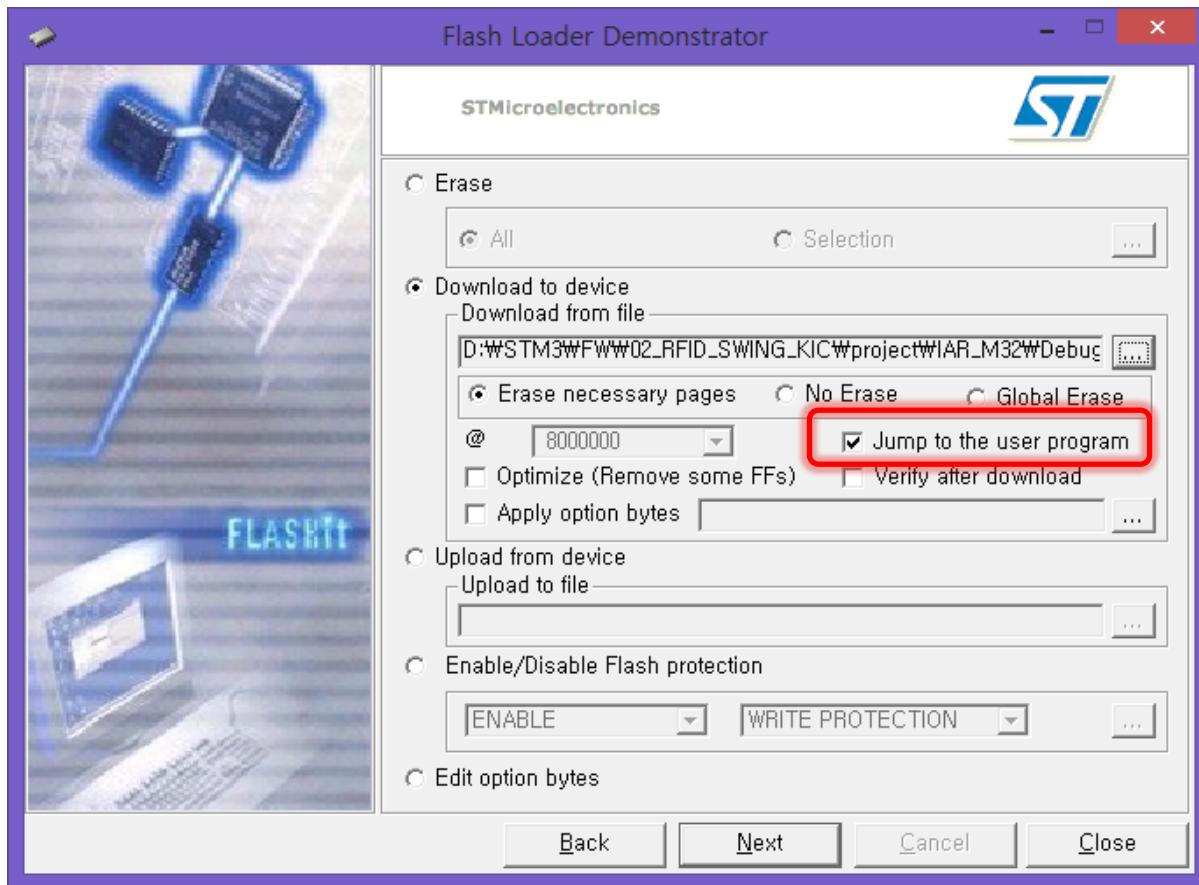
Click "Next" button.



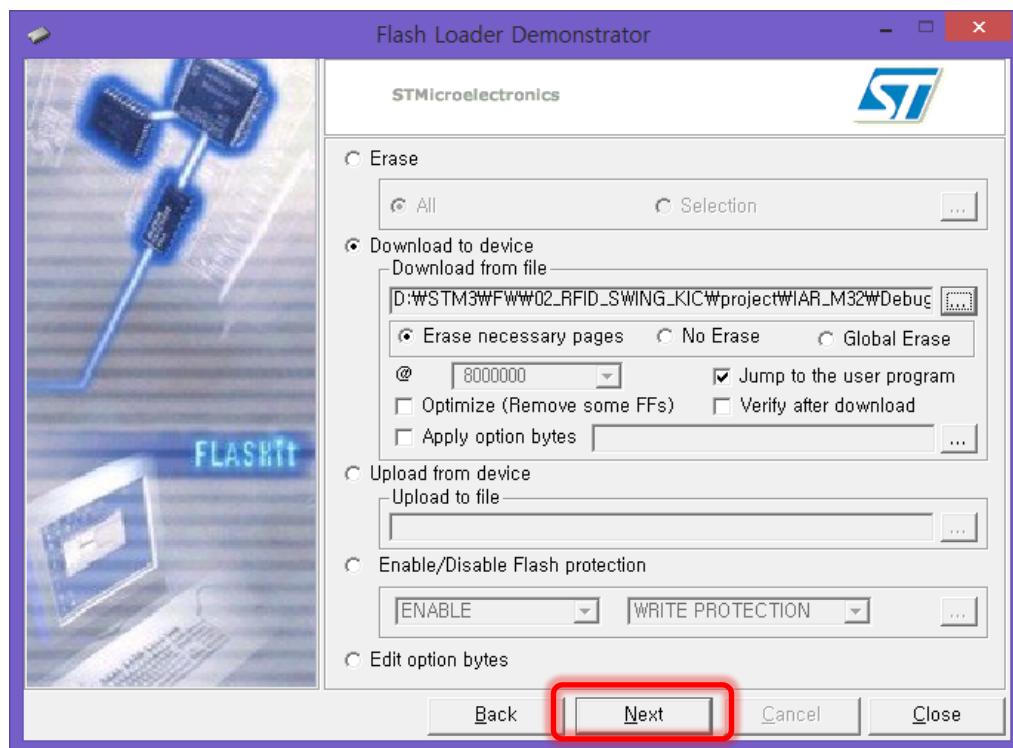
Click "... button for F/W file selection.



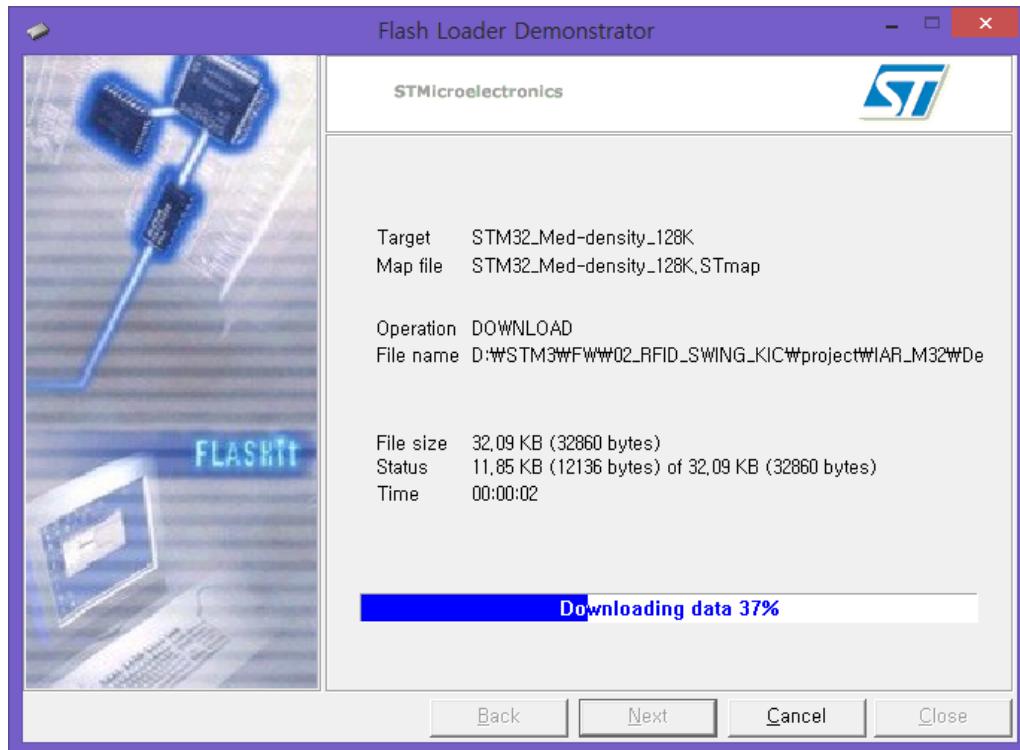
After select valid F/W binary file, then click "Open" button.



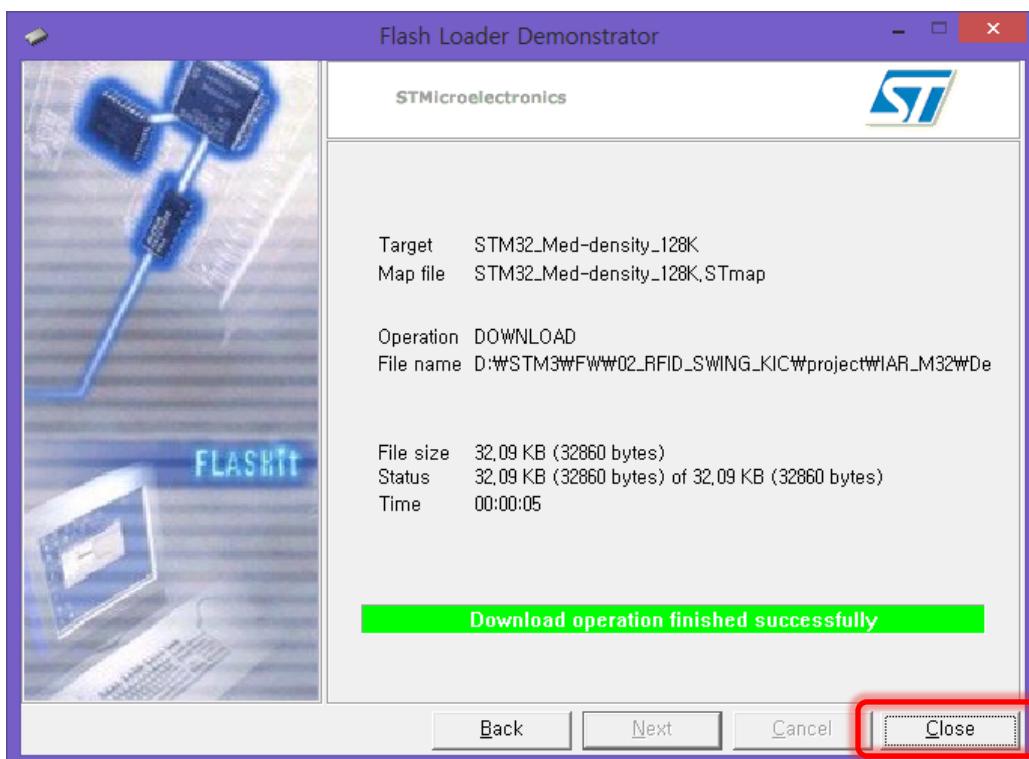
Check "Jump to the user program"



Click "Next" button.



Wait until progress bar to 100%.



Click "Close" button for finish the F/W update

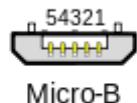
4. Communication protocol

4.1. Communication Interface

4.1.1. USB Interface

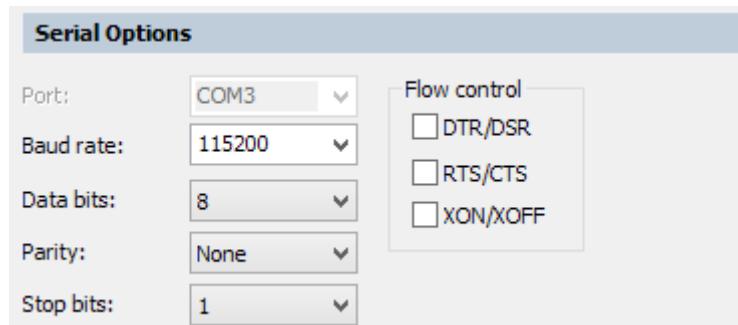
- ✓ USB port type

Swing-U has standard Micro-B type USB port like android smartphone



- ✓ Virtual Com port specification

Swing-U uses virtual com port for serial communication. Detail serial port options are as follows:



※ Port can have different value by PC environment.

- ✓ Device driver

You can obtain the newest version of Swing-U USB driver using follow link.

Web link

<http://www.silabs.com/products/mcu/pages/usbtouartbridgevcpdrivers.aspx>

4.1.2. Bluetooth Interface

Swing-U can communicate with host using SPP (Serial Port Profile) of Bluetooth.

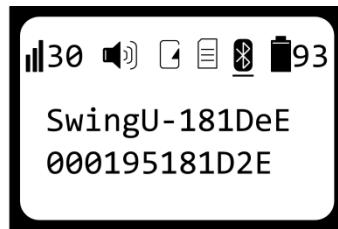
- ✓ Bluetooth MAC address / Device serial number

MAC address of Bluetooth has 12 characters. First 6 characters are same for every Swing-U. So, last 6 characters are used for serial number of Swing-U.

The rule of serial number is as follows:

MAC address Rule	0001951EEAB5		Serial Number Rule	SwingU-1EEAB6	
	Constant	Variable		Prefix	Variable
	000195	1EEAB5		SwingU-	1EEAB5

You can check the serial number of Swing-U by LCD menu.



- ✓ PIN code

When proceeding first pairing of Bluetooth, your host will ask the pin code for security.

The pin code is as follows:

PIN code	1234
----------	------

- ✓ Com port specification

Serial communication options are same with the options of USB.

4.2. Packet Format

The packet has three fields. STX, payload, and ETX each fields using ASCII code.

STX	Payload	ETX
>	...	\r\n

- ✓ STX and ETX

These are the delimiter for start of packet and end of packet.

Delimiter	Byte length	Byte	ASCII
STX	1	0x3E	>
ETX	2	0x0D0A	\r\n

after this line, ETX will be displayed with "<".

- ✓ Payload

Real data of packet

	Command	Separator	Parameters
Byte length	1	1	Variable
Data	...	' '(0x20)	...

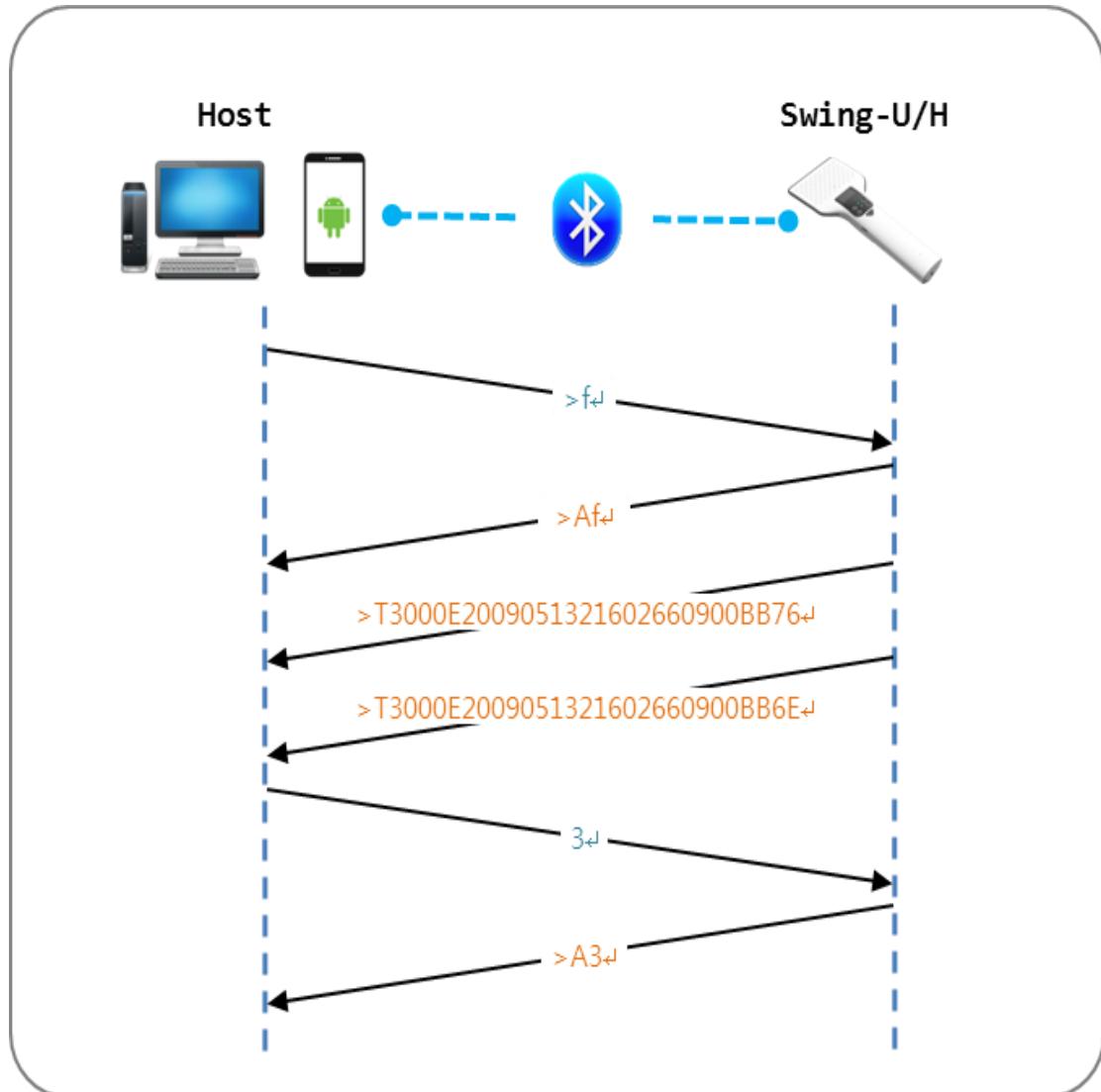
- ✓ Commands

Supported commands are as follows:

Command	Detail	Remarks
f	RFID inventory start	
3	RFID inventory stop	STX excluding
r	RFID memory read	
w	RFID memory write	
s	RFID memory select	
l	RFID memory lock	
c	Tag count initialization	
x	Set the status parameters of Swing-U	
y	Request the status parameters of Swing-U	
i	Request the all status parameters of Swing-U	
m	EPC length setting	
t	Same tag filtering size setting	
n	total tag's count from host	

4.3. Packet Data

- ✓ RFID inventory and searching tags



4.4. Inventory and search mode

Host can set and get the mode of inventory and search.

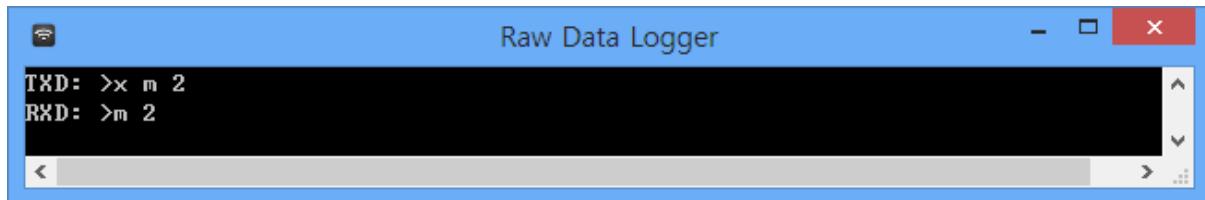
	STX	Command	SP	Parameter 1	SP	Parameter 2	ETX
Set	>	x		m		0	\r\n
Get	>	y		m			\r\n
Response	>	m		0			\r\n

- Parameters

Parameter		Values			Remarks	
1	Control parameter	m			Inventory mode	
2	Inventory and search mode	0	Normal inventory, read EPC ID		Default: 0	
		1	EPC+Data, read EPC ID with memory			
		2	Single tag searching			
		3	Multiple tag searching by lists			
		4	Multiple tag searching by wildcard character ('X' or 'x')			

- Example:

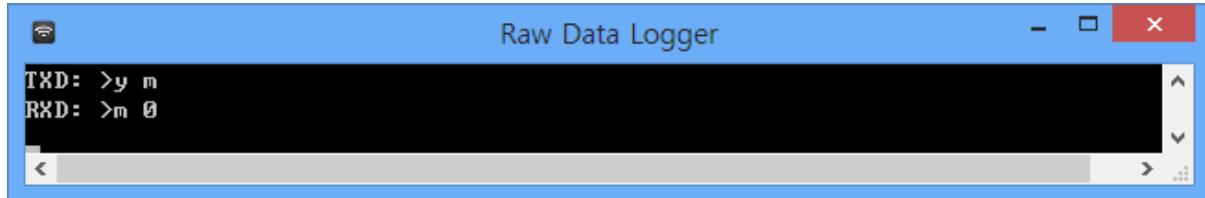
Set inventory and search mode to 2 for single tag searching mode



- Set inventory and search mode to 1 for memory inventory mode(Bank : EPC, offset: 2 Length : 6)



Get inventory mode: 0 = normal inventory mode



4.4.1. Inventory start

The inventory defined on ISO18000-6C/EPC global class1 gen2 will be started. Stated inventory will be stop when Swing-U requested inventory stop command by host or button

Sequence	STX	Command	Parameter	ETX
Request	>	f or fr		\r\n
Response	>	Af		\r\n

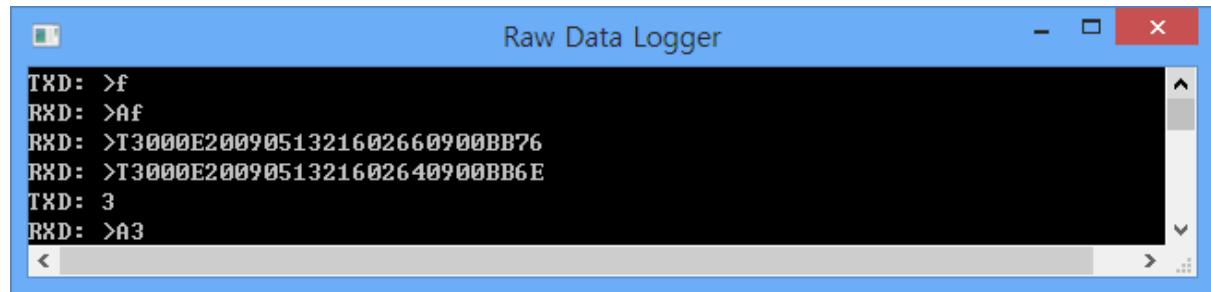
4.4.2. Inventory Stop

Swing-U stops the proceeding inventory.

Specifically, the inventory stop request needs no STX.

Sequence	STX	Command	Parameter	ETX
Request		3		\r\n
Response	>	A3		\r\n

- Example



4.4.3. Inventory Notification

The notification response is different for each inventory and search mode.

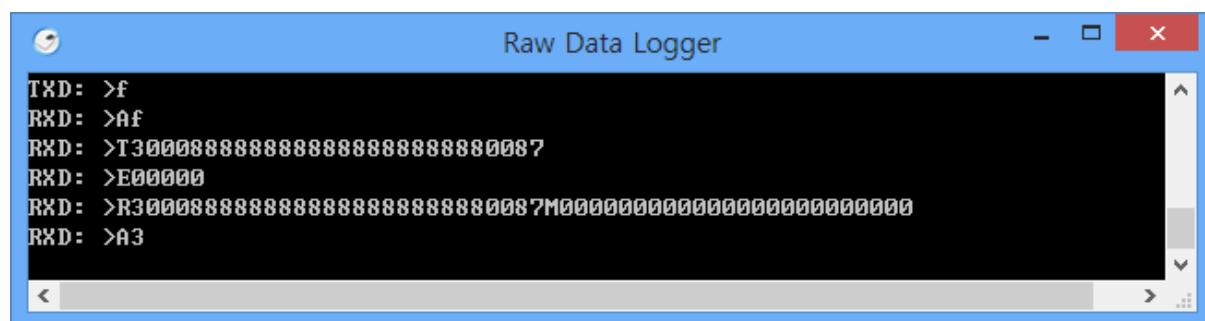
- **Normal inventory mode**

Sequence	STX	Command	Parameter	ETX
Notification	>	T	UID	\r\n



- EPC + User memory inventory mode

Sequence	STX	Command	Parameter1	SP	Parameter2	ETX
Notification	>	R	UID	M	User Data	\r\n



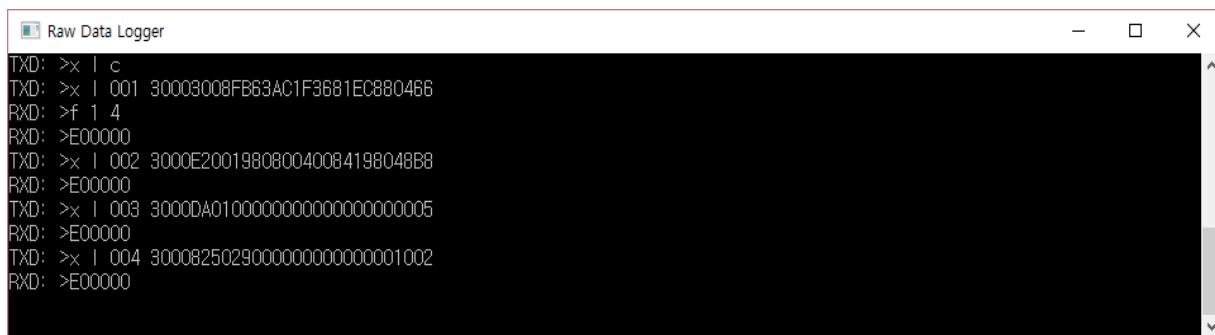
- Single tag searching mode

Sequence	STX	Command	Parameter1	SP	Parameter2	ETX
Notification	>	G	Target tag list Index(001)	,	UID	\r\n



- **Multiple tag searching mode by target tag lists**

Sequence	STX	Command	Parameter1	SP	Parameter2	ETX
Notification	>	G	Target tag list Index	,	UID	\r\n



The screenshot shows a window titled "Raw Data Logger". The log area contains the following text:

```

Raw Data Logger
TXD: >x | c
TXD: >x | 001 30003008FB63AC1F3681EC880466
RXD: >f 1 4
RXD: >E00000
TXD: >x | 002 3000E20019808004008419804BB8
RXD: >E00000
TXD: >x | 003 3000DA01000000000000000000000005
RXD: >E00000
TXD: >x | 004 30008250290000000000000000000002
RXD: >E00000

```

- **Multiple tag searching mode by wildcard character**

Sequence	STX	Command	Parameter1	SP	Parameter2	ETX
Notification	>	G	Target tag list Index	,	UID	\r\n

When wildcard is "30003008FB63AC1F3681EC8804xx",



The screenshot shows a window titled "Raw Data Logger". The log area contains the following text:

```

Raw Data Logger
RXD: >Af
RXD: >G99999,30003008FB63AC1F3681EC880466
RXD: >p 000001
RXD: >E00000

```

4.4.4. Search target tag list and parameters

On this chapter, finding tag mode will be explained. The finding tag mode is can set by above chapter.

- ✓ Add target ID to search target tag list and clear list

- Host set the target tag UID for Search mode.

	STX	Command	SP	Parameter	SP	Parameter	SP	Parameter	ETX
				1		2		3	
Set	>	x				1		UID	\r\n
Error Response	>	E00000 E00004							\r\n

- Parameters

Parameter		Values				Remarks	
1	Control parameter					add tag UID	
2	Table index	range: 1 ~ 200, 901~999, c					
		1	Single tag searching mode uses this index only.				
		1~200	Single tag searching mode uses this index only.				
		901~999	Wildcard searching mode uses this index only.				
		c	clear tag lists				
3	UID	UID of target tag. especially, when inventory mode is WILDCARD, UID can contains 'X' or 'x' for wildcard character.					

- Example:

Set target tag ID to "3000300833B2DDD9014000000804" and response is no error.



4.4.5. Finding RF threshold

The threshold of RF output power which is Swing-U will notify the finding of tag.

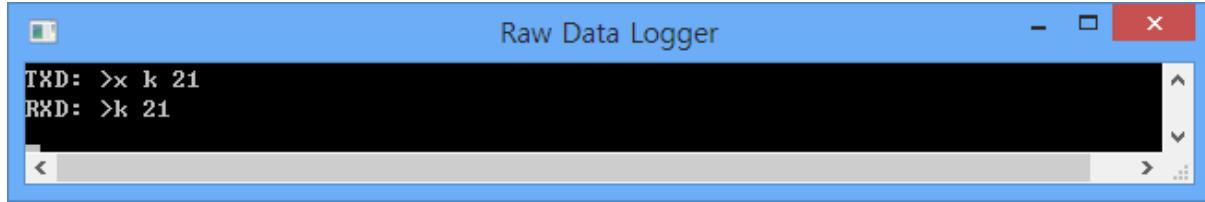
	STX	Command	Parameter	ETX
Set	>	x	k 8	\r\n
Get	>	y	k	\r\n
Response	>	k	8	\r\n

- Parameters

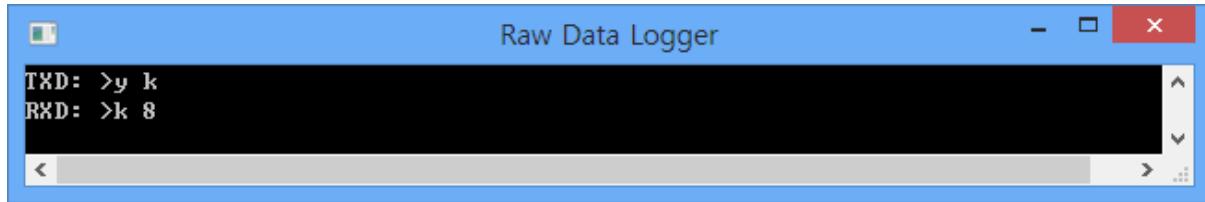
	Parameter	Values		Remarks	
1	Control parameter	k		Finding threshold	
2	Finding threshold	3	Threshold is 3 dBm	Default: 8	
		...			
		29	Threshold is 29 dBm		

- Example:

Set finding RF threshold to 21 [dBm]



Get finding RF threshold: 8 [dBm]



4.4.6. Finding RF decrement step unit

The decrement unit of RF output power when Swing-U operated in the single searching mode.

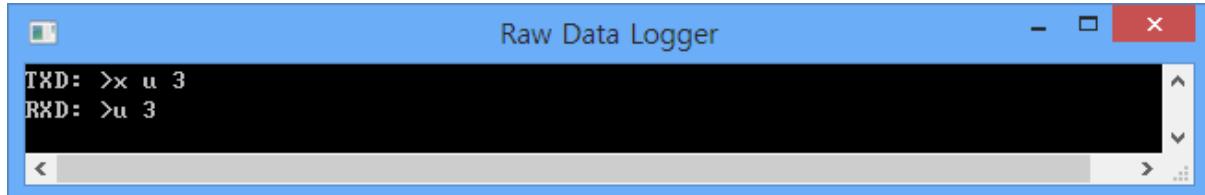
	STX	Command	Parameter	ETX
Set	>	x	u 3	\r\n
Get	>	y	u	\r\n
Response	>	u	3	\r\n

- Parameters

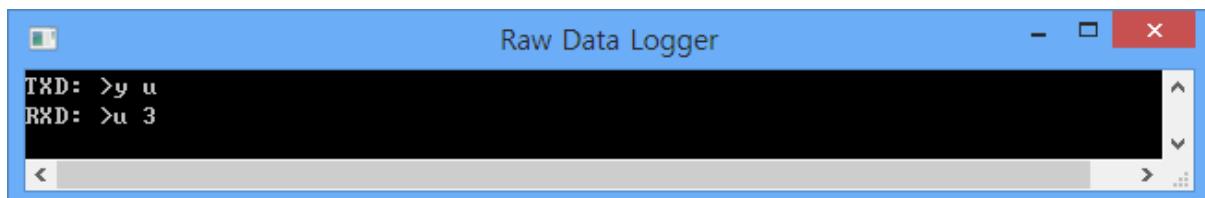
Parameter		Values		Remarks	
1	Control parameter	u		Decrement step	
2	Decrement step	3	Step unit is 3 dBm	Default: 3	
		...			
		6	Step unit is 6 dBm		

- Example:

Set decrement step to 3 [dBm]

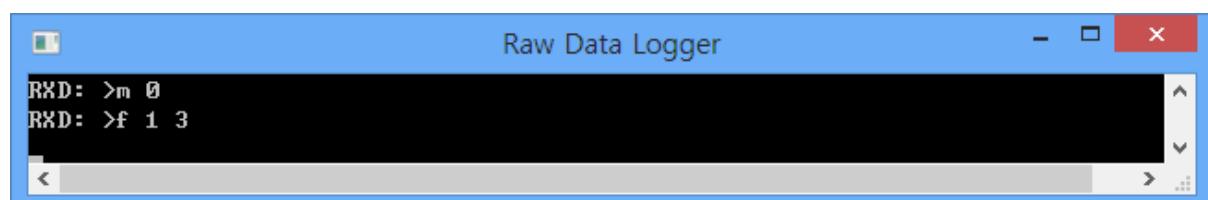


Get Decrement step: 3 [dBm]



4.4.7. Exit to Finding mode

Swing-U returns to normal inventory mode from single searching mode when user button F1 is pressed during three seconds.



4.4.8. Access Password Set

Swing-U access password setting

	STX	Command	Parameter	ETX
Set	>	a	11111111	\r\n
Error Response	>	E	00000~ 2000F	\r\n

- Parameters

Parameter		Values	Remarks
1	Password	Access password	Word (8 Byte) unit

- Example:

Set Access Password as 11111111

```
>a 11111111
>E00000
```

4.4.9. Session

Swing-U tag select session setting

	STX	Command	Parameter	ETX
Set	>	x	4 2	\r\n
Get	>	y	4	\r\n
Response	>	4	0~2	\r\n

- Parameters

Parameter		Values		Remarks
1	Control parameter	4		Session number
2	Session number	0	Session 0 select	Default: 2
		1	Session 1 select	
		2	Session 2 select	

- Example:

Set Session 2

```
>x 4 2
>4 2
```

4.4.10. Flag

Swing-U tag select Flag setting

	STX	Command	Parameter	ETX
Set	>	x	0 0	\r\n
Get	>	y	0	\r\n
Response	>	4	0~1	\r\n

- Parameters

	Parameter	Values		Remarks
1	Control parameter	4		Flag number
2	Flag number	0	Flag A select	
		1	Flag B select	

- Example:

Set Flag A

```
>x 0 0
>0 0
```

4.5. RFID tag memory access

4.5.1. Tag memory read

Swing-U can read the memory data of RFID tag.

Sequence	STX	Command	Parameter	ETX
Request	>	r	01 02 06	\r\n
Error Response	>	E	00000 ~ 2000F	\r\n
Data Response	>	R	UID+'M'+ Data	\r\n

- Parameters

Parameter		Values		Remarks
1	Memory bank	00	Reserved	
		01	EPC	
		02	TID	
		03	User	
2	Word offset	Memory offset address to read		Word (2 Byte) unit
3	Word count	Memory length to read		Word (2 Byte) unit

4.5.2. Tag memory write

Swing-U can write the data to memory of RFID tag.

	STX	Command	Parameter	ETX
Request	>	w	01 02 06 ...	\r\n
Error Response	>	E	00000 ~ 2000F	\r\n

- Parameters

Parameter		Values		Remarks
1	Memory bank	00	Reserved	
		01	EPC	
		02	TID	
		03	User	
2	Word offset	Memory offset address to write		Word (2 Byte) unit
3	Word count	Memory length to write		Word (2 Byte) unit
4	Data	Data for write		Minimum 2 bytes

4.5.3. Tag memory select

Swing-U can write the data to memory of RFID tag.

	STX	Command	Parameter	ETX
Request	>	S	01 02 06 ...	\r\n
Error Response	>	E	00000 ~ 2000F	\r\n
Data Response	>	T	UID	\r\n

- Parameters

Parameter		Values		Remarks
1	Memory bank	00	Reserved	
		01	EPC	
		02	TID	
		03	User	
2	Word offset	Memory offset address to write		Word (2 Byte) unit
3	Word count	Memory length to write		Word (2 Byte) unit
4	Data	Data for select		Minimum 2 bytes

4.5.4. Tag memory lock

Swing-U can lock the memory data of RFID tag.

	STX	Command	Parameter	ETX
Request	>	w	4 4 4 4 4 11111111	\r\n
Error Response	>	E	00000 ~ 2000F	\r\n

- Parameters

Parameter		Values		Remarks
1	Kill password Action	0	Accessible	Lock action for Kill password memory
		1	Always accessible	
		2	Secured accessible	
		3	Always not accessible	
		4	No change	
2	Access password action	0	Accessible	Lock action for Access password memory
		1	Always accessible	
		2	Secured accessible	
		3	Always not accessible	
		4	No change	
3	EPC memory action	0	Accessible	Lock action for EPC memory
		1	Always accessible	
		2	Secured accessible	
		3	Always not accessible	
		4	No change	
4	TID memory action	0	Accessible	Lock action for TID memory
		1	Always accessible	
		2	Secured accessible	
		3	Always not accessible	
		4	No change	
5	User memory action	0	Accessible	Lock action for USER memory
		1	Always accessible	

		2	Secured accessible	
		3	Always not accessible	
		4	No change	
6	Access Password	Tag Access Password		Always 8 bytes

4.5.5. Error Code

Category	Code	Separator	Remarks
Protocol Error	E00000	No Error	Success
	E00001	Invalid protocol	
	E00002	Invalid parameter	
	E00003	Unknown command	
	E00004	Operation failed	
Mac Error	E10001	Handle mismatch	
	E10002	CRC error	
	E10003	No tag reply	
	E10004	Invalid password	
	E10005	Zero kill password	
	E10006	Tag lost	
	E10007	Command format error	
	E10008	Read count invalid	
	E10009	Out of retries	
	E1FFFF	Operation failed	
Backscatter Error	E20000	General error	
	E20003	No memory	
	E20004	Memory locked	
	E2000B	Insufficient power	
	E2000F	Unknown error	

4.5.6. Example

- #### - Data reading:

Bank: EPC memory, offset address: 2, address count: 6 [words]

```
Raw Data Logger - □ X  
TXD: >r 01 02 06  
RXD: >T3400F20190511234989899998888  
RXD: >E00000  
RXD: >R3400F20190511234989899998888MF20190511234989899998888  
RXD: >A3
```

Bank: User memory, offset address: 0, address count: 10 [words]

```
Raw Data Logger - □ X  
TXD: >r 03 00 0A  
RXD: >E00000  
RXD: >R3400F2019051123498989998888M0102030405060708090A0A0B0C0D0E0F00000000  
RXD: >A3
```

- ### - Data writing

Bank: User memory [03], offset address: 0 [00], address count: 5 [05] [words]

A screenshot of a Windows application window titled "Raw Data Logger". The window contains a text log area with the following content:

```
TXD: >w 03 00 05 0102030405060708090A  
RXD: >E00000  
RXD: >A3
```

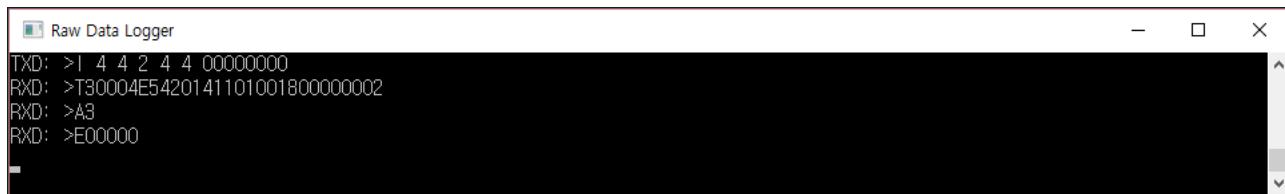
The window has standard blue title bar and window controls.

- ## - Tag Selecting

Bank: EPC memory [01], offset address: 2 [02], address count: 6 [06] [words]

TXD: >s 01 02 06 000000000000000000000000000000001793
RXD: >E00000

- Data lock
- Kill password: No change,
Access password: No change,
EPC: No change,
TID: No change,
USER: No change,
Access Password: 00000000



4.6. Swing-U status control

4.6.1. Version

Swing-U reports the version of H/W and F/W.

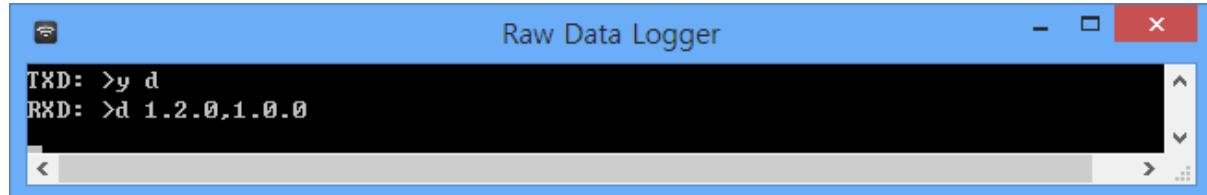
	STX	Command	Parameter	ETX
Get	>	y	d	\r\n
Response	>	d	1.2.0,1.0.0	\r\n

- Parameters

Parameter		Values		Remarks
1	Control parameter	d		Version
2	Version	X,Y	X: version of hardware Y: version of firmware	Initial Release H/W: 1.2.0 F/W: 1.0.0

- Example:

Get version of Swing-U: (H/W)1.2.0, (F/W)1.0.0



4.6.2. RF output power of reader

Host can set or get the RF output power of Swing-U.

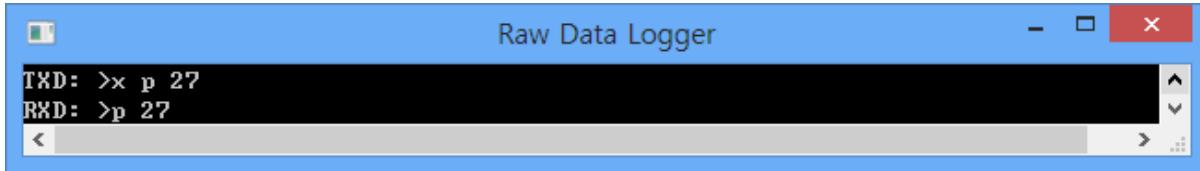
	STX	Command	Parameter	ETX
Set	>	x	p 0	\r\n
Get	>	y	p	\r\n
Response	>	p	0	\r\n

- Parameters

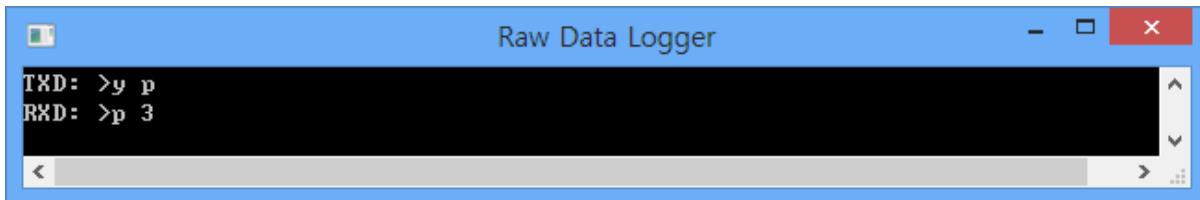
Parameter		Values		Remarks
1	Control parameter	p		RF attenuation
2	RF attenuation	0	attenuation is 0, RF output is 30dBm	

		
	27		attenuation is 27, RF output is 3dBm	

- Example:
Set RF output power to 3 [dBm]



Get RF output power: $30 - 3 = 27$ [dBm]



4.6.3. Buzzer volume

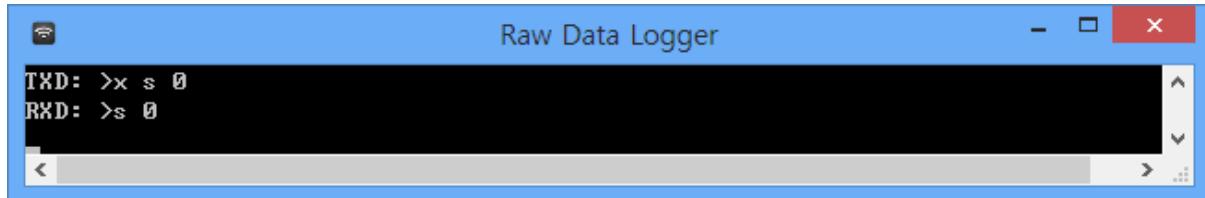
Host can set or get the buzzer volume of Swing-U.

	STX	Command	Parameter	ETX
Set	>	x	s 0	\r\n
Get	>	y	s	\r\n
Response	>	s	0	\r\n

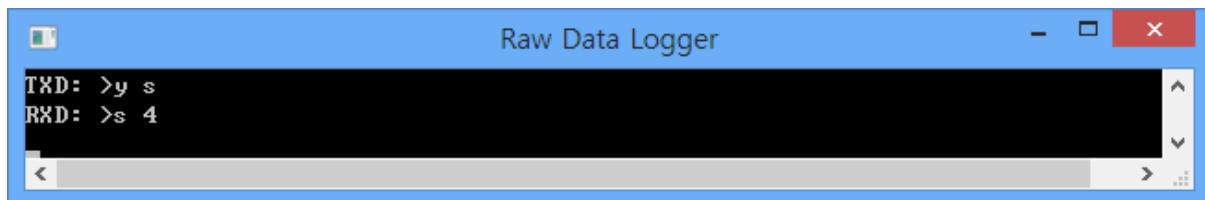
- Parameters

Parameter		Values		Remarks
1	Control parameter	s		Buzzer volume
2	Buzzer volume	0	Vibration	
		1	Mute	
		2	Minimum volume	
		3	Normal volume	
		4	Maximum volume	

- Example:
Set buzzer volume to 0 for vibration



Get buzzer volume: 4 = Maximum volume



4.6.4. Continuous read mode

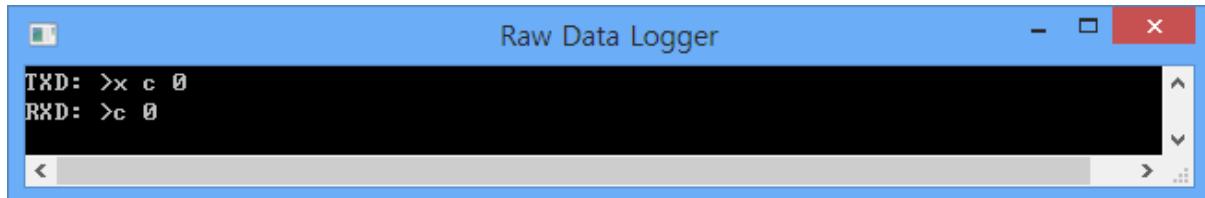
Host can set or get the continuous mode of RFID action like inventory and memory read.

	STX	Command	Parameter	ETX
Set	>	x	c 0	\r\n
Get	>	y	c	\r\n
Response	>	c	0	\r\n

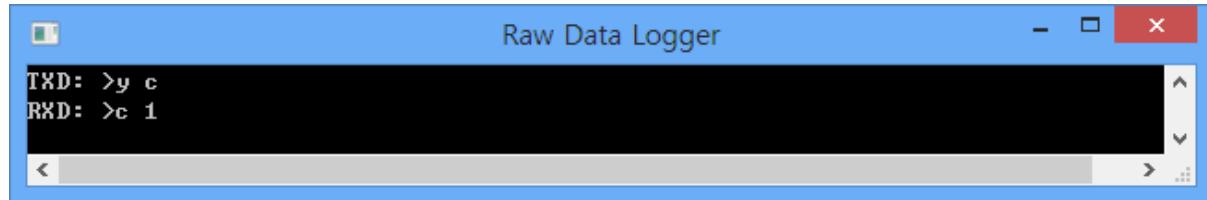
- Parameters

Parameter		Values		Remarks
1	Control parameter	c		Continuous Mode
2	Continuous Mode	0	Single read	
		1	Continuous read	

- Example:
Set continuous mode to 0 for single mode



Get continuous mode: 1 = continuous mode



4.6.5. Tag report mode

Host can set or get the UID report mode of Swing-U.

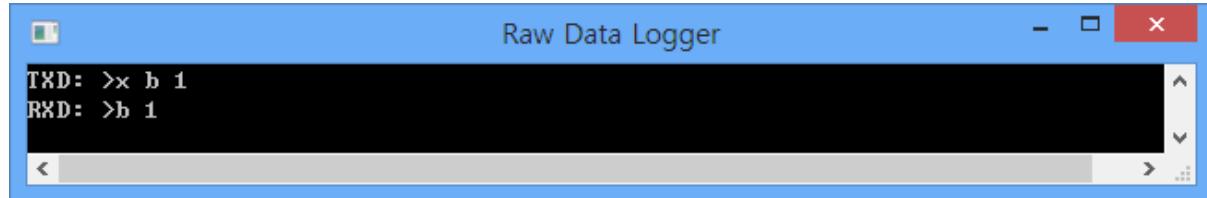
	STX	Command	Parameter	ETX
Set	>	x	b 0	\r\n
Get	>	y	b	\r\n
Response	>	b	0	\r\n

- Parameters

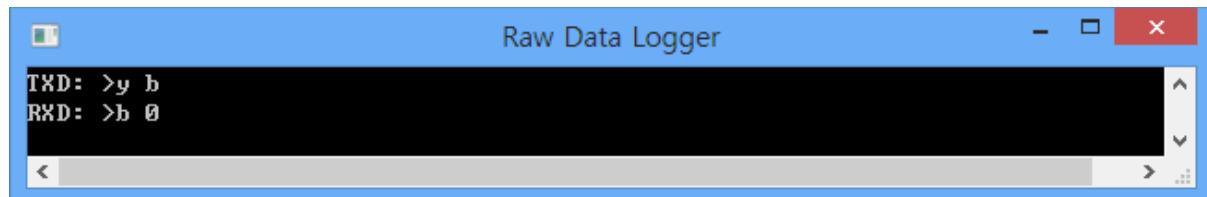
	Parameter	Values		Remarks
1	Control parameter	c		Tag report Mode
2	Tag report Mode	0	Trigger, Swing-U report new tag only	Default: 0
		1	Always, Swing-U report every tag reading	

- Example:

Set tag report mode to 1 for always mode



Get tag report mode: 0 = trigger mode



4.6.6. Battery level

Host can get the battery level of Swing-U.

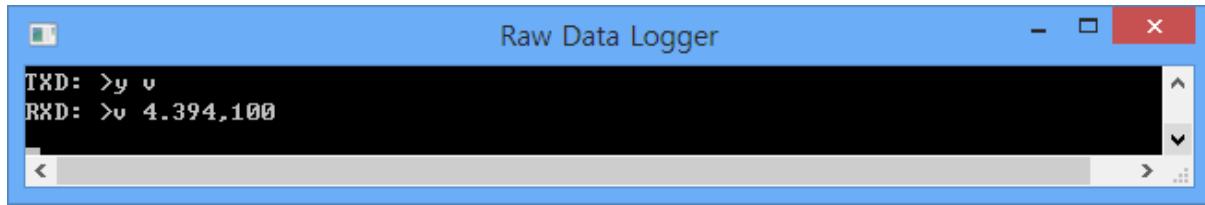
	STX	Command	Parameter	ETX
Get	>	y	v	\r\n
Response	>	v	0	\r\n

- Parameters

	Parameter	Values		Remarks
1	Control parameter	v		Battery level
2	Battery level	X,Y	X: volts of battery Y: percent of battery	

- Example:

Get battery level: battery level is 4.394 [volt], 100[%]



4.6.7. Battery Threshold

Swing-U Battery threshold setting

	STX	Command	Parameter	ETX
Set	>	x	h 0	\r\n
Response	>	h	0~50	\r\n

- Parameters

Parameter		Values		Remarks
1	Control parameter	4		Battery threshold
2	Battery threshold	0~50	Battery threshold	

- Example:

Set Battery threshold 25

```
>x h 25
>h 25
```

4.6.8. Battery charging Status

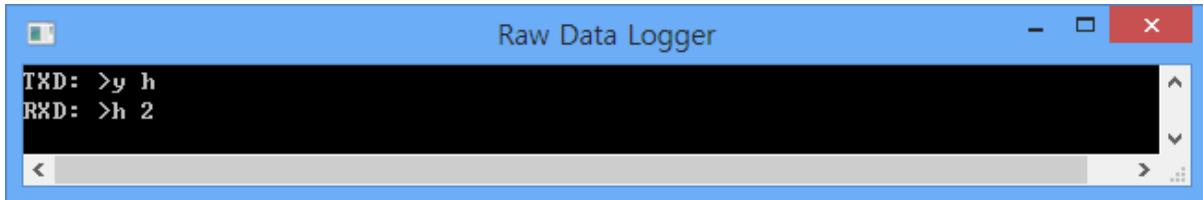
Host can get the battery charging status of Swing-U.

	STX	Command	Parameter	ETX
Get	>	y	h	\r\n
Response	>	h	0	\r\n

- Parameters

Parameter		Values		Remarks
1	Control parameter	h		Charging status
2	Charging status	0	Battery error	
		1	Charging	
		2	Full charged	
		3	Discharging	

- Example:
Get battery charging status: batter is full charged



4.6.9. Swing-U filter size setting

Host can set filter size to Swing-U.

	STX	Command	Parameter	ETX
Set	>	t	00000~99999	\r\n
Response	>	E	00000 ~ 2000F	\r\n

- Parameters

Parameter		Values	Remarks
1	Control parameter	t	Filter size
2	Filter size	00000~99999	Default: 0

- Example:
Swing-U filter size 1024 set



- Example:
Swing-U filter size max set



4.6.10. Swing-U Read EPC max Length set

Host send EPC max Length set to Swing-U

	STX	Command	Parameter	ETX
Set	>	t	96~512	\r\n
Response	>	E	00000 ~ 2000F	\r\n

- Parameters:

Parameter	Values	Remarks
1 Control parameter	000~512	EPC Length

- Example:

Swing-U EPC Length 512 set



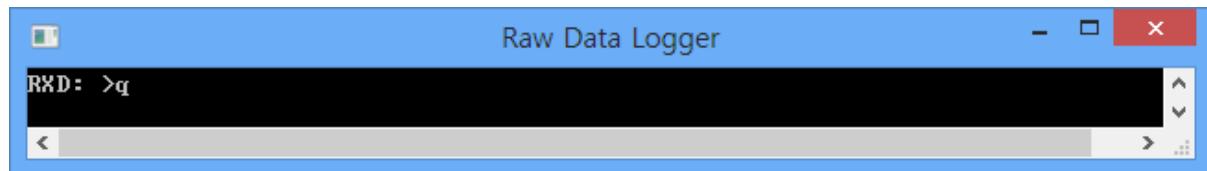
4.6.11. Power-off notification

Swing-U sends power-off notification when turned-off by power button.

	STX	Command	Parameter	ETX
Notification	>	q		\r\n

- Parameters: None

- Example:



4.6.12. LCD control mode

Host can set or get the UID report mode of Swing-U.

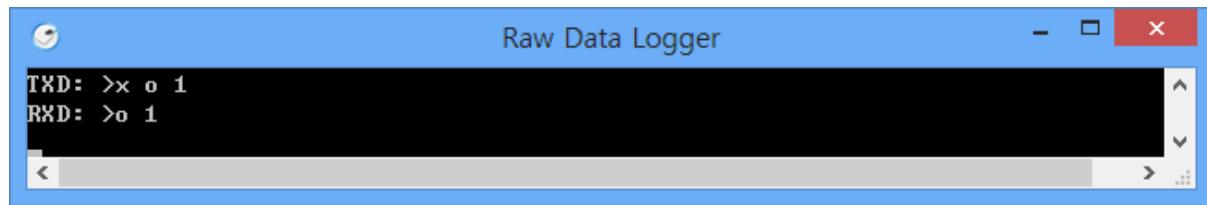
	STX	Command	Parameter	ETX
Set	>	x	o 0	\r\n
Get	>	y	o	\r\n
Response	>	o	0	\r\n

- Parameters

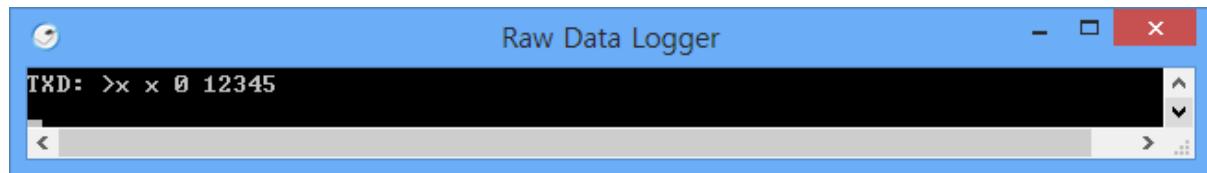
	Parameter	Values		Remarks	
1	Control parameter	o		LCD control mode	
2	LCD control mode	0	LCD controlled by Swing-U	Default: 0	
		1	LCD display large count set by user		
		2	LCD display text set by user		

- Example:

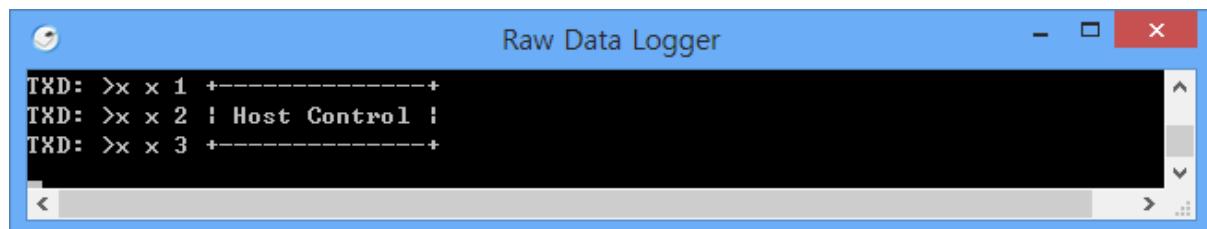
Set LCD control mode to 1 for large count set by user



Host set the large count as 12345



Host set the user text from line1 to line3



4.6.13. Menu enable

Host can set or get menu enable of Swing-U. (only for MK trend)

	STX	Command	Parameter	ETX
Set	>	x	a 0	\r\n
Get	>	y	A	\r\n
Response	>	a	0	\r\n

- Parameters

	Parameter	Values		Remarks
1	Control parameter	a		Menu enable
2	Menu enable	0	Menu can't control by buttons	
		1	Menu can control by buttons	

- Example:

Set Swing-U menu control



```

Raw Data Logger
TXD: >x a 1
RXD: >a 1
TXD: >x a 0
RXD: >a 0

```

4.6.14. Inventory timeout

Host can set or get Inventory mode read stop time of Swing-U.

	STX	Command	Parameter	ETX
Set	>	x	i 600	\r\n
Get	>	Y	i	\r\n
Response	>	y	600	\r\n

- Parameters

	Parameter	Values		Remarks
1	Control parameter	i		Inventory timeout
2	Time	0~32767	Set inventory mode read time by Swing-U 0: not stop	

- Example:

Set Inventory mode read 600 second stop



The screenshot shows a window titled "Raw Data Logger". The TXD log contains the command: >x i 600. The RXD log contains the response: >i 600.

4.6.15. Inventory mode get

Host can get Inventory mode read stop time of Swing-U.

	STX	Command	Parameter	ETX
Get	>	y	w	\r\n
Response	>	w	0 or 1	\r\n

- Parameters

	Parameter	Values		Remarks
1	Control parameter	w		Inventory mode
2	Inventory mode	0	Inventory read stop	
		1	Inventory reading	

- Example:

Set Swing-U menu control

```
>y w
>w 0
```

4.6.16. Read type

Host can set or get current read type of Swing-U.

	STX	Command	Parameter	ETX
Set	>	x	6 0	\r\n
Get	>	y	6	\r\n
Response	>	6	0	\r\n

- Parameters

	Parameter	Values		Remarks
1	Control parameter	6		Read type
2	type	0	RFID read mode	
		1	RFID temperature read mode [optional]	
		2	Barcode read mode	
		3	Encoding mode	

- Example:

Set RFID read mode



```
Raw Data Logger
TXD: >x 6 0
RXD: >6 0
```

- Example:

Set Barcode read mode



```
Raw Data Logger
TXD: >x 6 2
RXD: >6 2
```

- Example:

Set Encoding mode



```
Raw Data Logger
TXD: >x 6 3
RXD: >6 3
```

4.6.17. RF output power of reader

Host can set or get the Write output power of Swing-U.

	STX	Command	Parameter	ETX
Set	>	x	5 0	\r\n
Get	>	y	5	\r\n
Response	>	5	0	\r\n

- Parameters

	Parameter	Values		Remarks
1	Control parameter	5		Write attenuation
2	Write attenuation	0	attenuation is 0, Write output is 30dBm	Default: 0
		
		27	attenuation is 27, Write output is 3dBm	

- Example:

Set Write output power to 3 [dBm]



```
Raw Data Logger
TXD: >x 5 0
RXD: >5 0
```

Set Write output power: $30 - 3 = 27$ [dBm]



```
Raw Data Logger
TXD: >x 5 27
RXD: >5 27
```

4.6.18. Device name

Host can get Inventory mode read stop time of Swing-U.

	STX	Command	Parameter	ETX
Get	>	y	z	\r\n
Set	>	x	z xxxxxxxx	\r\n
Response	>	z	SwingU-xxxxxx	\r\n

- Parameters

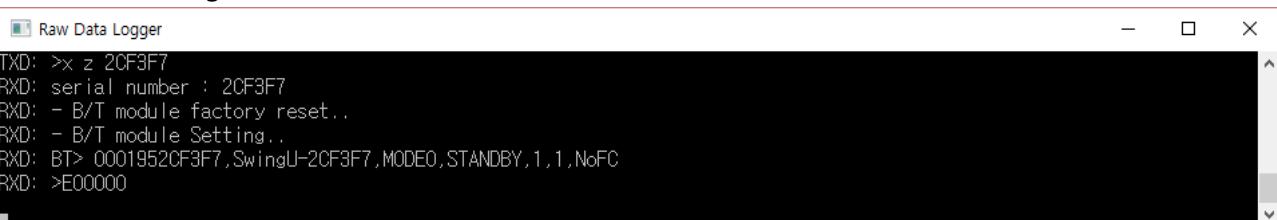
	Parameter	Values		Remarks
1	Control parameter	z		Device name
2	Swing-U Device name	SwingU-xxxxxx	Full Device name	Default:
		xxxxxxxx	Device name	Mac address

- Example:

Get Swing-U device name

```
>y z
>z SwingU-FFFFFF
```

Set Swing-U device name



```
Raw Data Logger
TXD: >x z 20F3F7
RXD: serial number : 20F3F7
RXD: - B/T module factory reset..
RXD: - B/T module Setting..
RXD: BT> 00019520F3F7,SwingU-20F3F7,MODE0,STANDBY,1,1,NoFC
RXD: >E00000
-
```

4.6.19. Serial number

Host can get the serial number and Bluetooth of Swing-U.

	STX	Command	Parameter	ETX
Get	>	y	8	\r\n
Response	>	8	NTU2-KCB0000;0001952CF3F7	\r\n

- Parameters

Parameter		Values		Remarks
1	Control parameter	8		Serial number
2	Serial number	NTU2-KCB0000	Serial number	Default: NTU2-KCB0000
		0001952CF3F7	Bluetooth MAC address	

Example:

Get Serial number of Swing-U



4.6.20. Language

Host can set or get the language of Swing-U.

	STX	Command	Parameter	ETX
Set	>	x	1 0	\r\n
Get	>	y	1	\r\n
Response	>	1	0	\r\n

- Parameters

Parameter		Values		Remarks
1	Control parameter	1		Language
2	Language	0	English	Default: English
		1	Chinese	
		2	Japanese	

- Example:

Set Swing-U language : English



4.7. User button notification

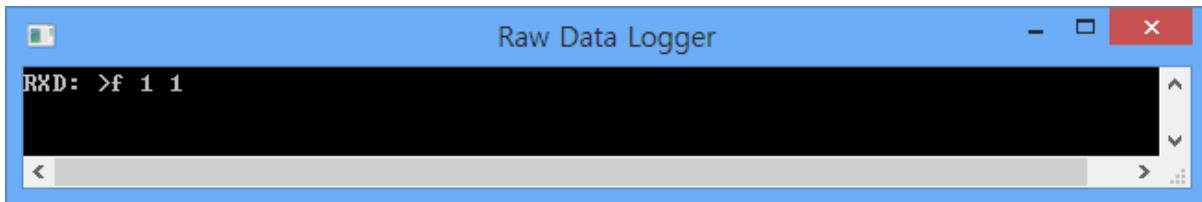
Swing-U notify the event of user button, **F1**.

4.7.1. User button click event

Applications can use this notification to operate some function.

	STX	Command	Parameter	ETX
Response	>	f	1 1	\r\n

- Example



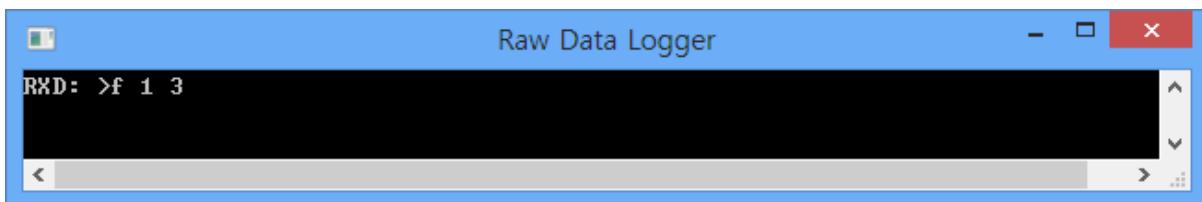
4.7.2. User button long-click event

Swing-U notifies the clear of tag list in memory.

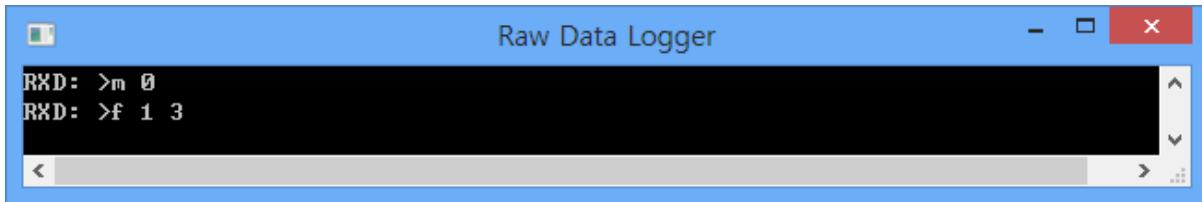
If the inventory mode of swing-U is finding mode, will be return to normal inventory mode.

	STX	Command	Parameter	ETX
Response	>	f	1 3	\r\n

- Example: when normal inventory mode,



- Example: when find mode,



4.8. Tag list synchronization

Swing-U can synchronize the list of tag with host.

4.8.1. Count of stored tag list

Host can get the count of tag list to Swing-U.

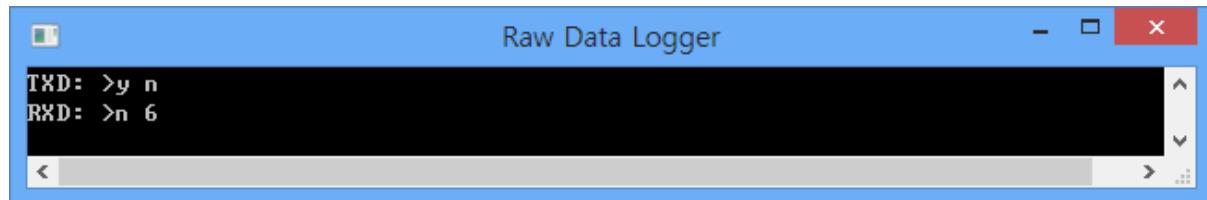
	STX	Command	Parameter	ETX
Get	>	y	n	\r\n
Response	>	n	5	\r\n

- Parameters

Parameter		Values		Remarks	
1	Control parameter	n		Tag list count	
2	Tag list count	0	No tag in list		
		...			
		29,000	Maximum value Swing-U can store		

- Example:

Get the stored list count of Swing-U: 6 tag ID in Swing-U



4.8.2. Host tag count send Swing-U

Host can set the tag count to Swing-U.

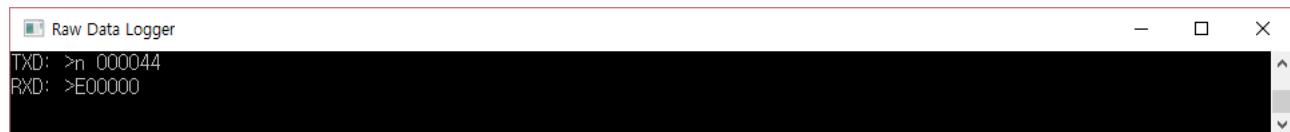
	STX	Command	Parameter	ETX
Get	>	n	000000~999999	\r\n
Response	>	E	00000 ~ 2000F	\r\n

- Parameters

Parameter	Values	Remarks
1 Control parameter	n	Tag count set

- Example:

Set the stored list count of Swing-U: 6 tag ID in Swing-U



4.8.3. Stored tag list

Host can add the specific tag UID to list of Swing-U or request the entire list of tag.

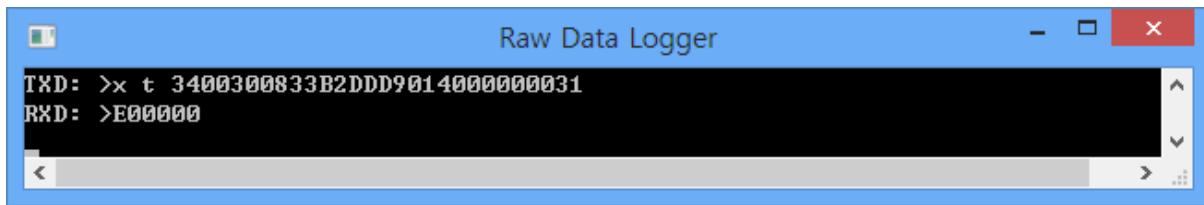
	STX	Command	Parameter	ETX
Set	>	x	t 3000...	\r\n
Get	>	y	t	\r\n
Response	>	T	3000...	\r\n
End Response	>	n	Tag count	\r\n

- Parameters

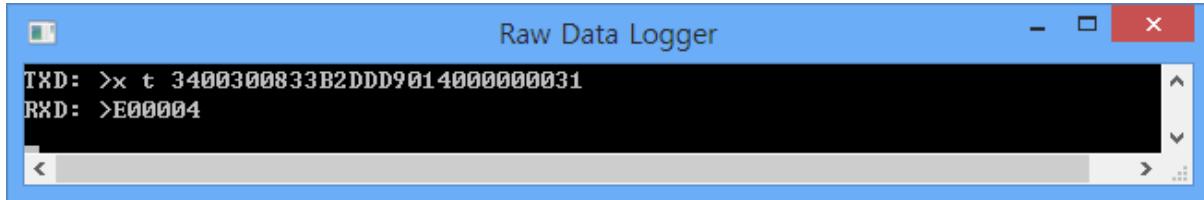
Parameter	Values	Remarks
1 Control parameter	t	Tag ID
2 Tag ID	3000...	

- Example:

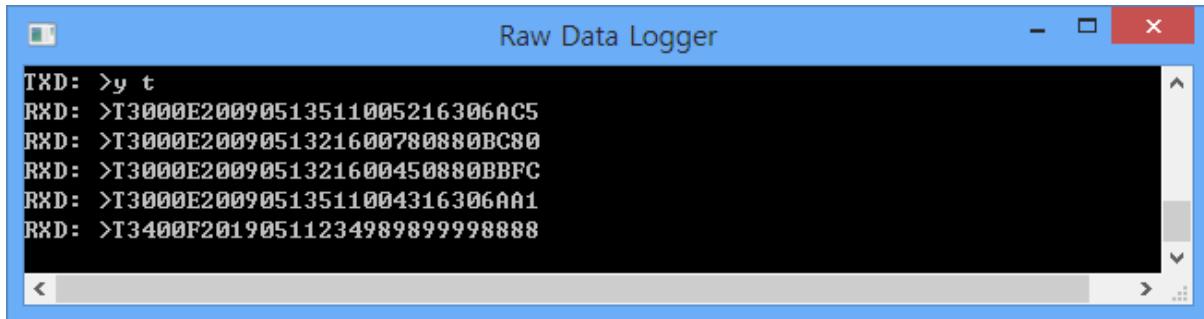
Add tag ID "3400300833B2DDD9014000000031" to list of Swing-U



Tag ID was already in list, "E00004(operation failed)" error returned



Get tag list of Swing-U: 5 tag ID in memory of Swing-U



5. .NET C# Library API

5.1. Overview

This document is library reference of Swing-U(NTRM-U-2) for .NET C# library on windows operation system.

5.2. Revision History

Version	Date	Description
1.0.0	2014-05-30	First Release
1.0.1	2014-06-24	<ul style="list-style-type: none">- Add notification when Swing-U is turned off.- Add notification for finding tag is completed.- Add new sample project using InTheHand.dll for Bluetooth connection without serial COM port adding.-
1.0.2	2014-06-30	<ul style="list-style-type: none">- inventory mode control modified- LCD mode and contents control added
1.0.3	2016-04-27	<ul style="list-style-type: none">- Memory lock function added- EPC length setting function added- Same tag filtering size setting function added

5.3. Swing-U Library contents

Swing-U Library contains as follows:

- ✓ SwingLibrary.dll
 - : Dynamic Link Library for functions and event to control Swing-U, UHF RFID reader
- ✓ SwingU_Demo.exe
 - : Demo application using SwingLibrary.dll

※ Notice

1. This API is developed on Microsoft Visual Studio 2008 platform.
2. This API can operate on .NET Framework 2.0 or above.
3. Sample Project is also developed on Visual Studio 2008, using C#.

5.4. Library Event

API supports four events for Swing-U

-  NotifyButtonEvent
-  NotifyError
-  NotifyInventory
-  NotifyInventoryBCD
-  NotifyParameterChanged
-  NotifyReadEvent
-  NotifyStatusCheck
-  NotifyTagFound

5.4.1. NotifyParameterChanged

Name	ParameterDelegate NotifyParameterChanged
Delegate	void ParameterDelegate(SwingParameter parameterType)
Return	void
Argument	<pre>enum SwingParameter { ContinuousMode, InventoryMode, TagReportMode, BuzzerVolume, ChargingStatus, BatteryStatus, RFPower, FindStepUnit, // Step down unit of RF output power for find mode FindThreshold, // Threshold of RF output power for find mode Version, TagCount, PowerOff, // Turned-off by power button, add in 1.0.1 LCDControlMode, InventoryTimeout, DeviceName, ModelNumber, PowerDown, InventoryStatus, SwingMode, BarcodeType, SerialNumber, CustomerCode, FlashLock, MenuEnable, };</pre>
Operation	Event for parameter changing notification by host request or button from Swing-U to host
Example	<pre>Swing.NotifyParameterChanged += new SwingLibrary.ParameterDelegate(processor); ... void processor(SwingLibrary.SwingParameter parameterType) { switch (parameterType) {</pre>

```

case SwingLibrary.SwingParameter.BatteryStatus:
    ...
    break;
case SwingLibrary.SwingParameter.BuzzerVolume:
    ...
    break;
...
case SwingLibrary.SwingParameter.PowerOff: // add in 1.0.1
    MessageBox.Show("Power off");
    break;
}

```

5.4.2. NotifyButtonEvent

Name	ButtonDelegate NotifyButtonEvent
Delegate	void ButtonDelegate(ButtonEvent buttonType)
Return	void
Argument	<pre> enum ButtonEvent { READSTART, // Read starts on Swing-U by button or request READSTOP, // Read stops on Swing-U by button or request TAGLISTCLEAR, // Tag list cleared by three seconds long-click FN button of Swing-U FN // short click of FN button on Swing-U }; </pre>
Operation	Event for Read button and FN button click notification from Swing-U to host
Example	<pre> Swing.NotifyButtonEvent += new SwingLibrary.ButtonDelegate(NotifyButtonEvent); ... void NotifyButtonEvent(SwingLibrary.ButtonEvent buttonType) { switch (buttonType) { case SwingLibrary.ButtonEvent.FN: ... break; case SwingLibrary.ButtonEvent.READSTART: ... break; case SwingLibrary.ButtonEvent.READSTOP: ... break; case SwingLibrary.ButtonEvent.TAGLISTCLEAR: ... break; default: break; } } </pre>

5.4.3. NotifyInventory

Name	DataDelegate NotifyInventory
Delegate	void DataDelegate (string data)
Return	void
Argument	string data;
Operation	Event for read tag ID and memory information from Swing-U to host
Example	<pre>Swing.NotifyInventory += new SwingLibrary.DataDelegate(Swing_NotifyInventory); ... void Swing_NotifyInventory(string data) { if(data.Contains("M")) // Memory Read Swing_ParseMemoryReadReport(data); else // EPC ID read Swing_ParseTagReport(data); }</pre>

5.4.4. NotifyInventoryBCD

Name	DataDelegate NotifyInventory
Delegate	void DataDelegate (string data)
Return	void
Argument	string data;
Operation	Event for read Barcode Data and memory information from Swing-U to host
Example	<pre>Swing.NotifyInventoryBCD+=new SwingLibrary.DataDelegate(Swing_NotifyInventoryBCD); ... void Swing_NotifyInventoryBCD(string data) { if(data.Contains("M")) // Memory Read Swing_ParseMemoryReadReport(data); else // Barcode data read Swing_ParseTagReport(data, "B"); }</pre>

5.4.5. NotifyTagFound

Name	DataDelegate NotifyTagFound
Delegate	void DataDelegate (string data)
Return	void
Argument	string data;
Operation	Event is thrown when searching tag is found.
Example	<pre>Swing.NotifyTagFound += new SwingLibrary.DataDelegate(Swing_NotifyTagFound); ... void Swing_NotifyTagFound(string data) // add in 1.0.1 { MessageBox.Show(string.Format("{0} is found!!", data), "Search information", MessageBoxButtons.OK, MessageBoxIcon.Information); }</pre>

5.4.6. NotifyReadEvent

Name	ReadDelegate NotifyButtonEvent
Delegate	void ReadDelegate(ReadEvent readType)
Return	void
Argument	<pre>enum ReadEvent { START, // Read starts Swing-U STOP, // Read stops Swing-U };</pre>
Operation	Event for Start, Stop from Swing-U to host
Example	<pre>Swing.NotifyReadEvent += new SwingLibrary.ReadDelegate(Swing_notifyReadEvent); ... void Swing_NotifyReadEvent(SwingLibrary.ReadEvent readType) { switch (readType) { case SwingLibrary.ReadEvent.START: ... break; case SwingLibrary.ReadEvent.STOP: ... break; default: break; } }</pre>

5.4.7. NotifyStatusCheck

Name	StatusDelegate NotifyStatusCheck
Delegate	void StatusDelegate(SwingStatusParameter parameterType)
Return	void
Argument	enum SwingStatusParameter { BatterWarning, // Battery Warning Swing-U InitLoadingEnd, // Init Loading End Swing-U };
Operation	Event for Batterwarning, InitLoadingEnd from Swing-U to host
Example	<pre>Swing.NotifyStatusCheck += new SwingLibrary.StatusDelegate(Swing_notifyStatus); ... void Swing_NotifyStatus(SwingLibrary.SwingStatusParameter parameterType) { switch (parameterType) { case SwingLibrary.SwingStatusParameter.BatteryWarning: ... break; case SwingLibrary.SwingStatusParameter.InitLoadingEnd: ... break; default: break; } }</pre>

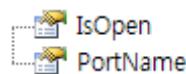
5.4.8. NotifyError

Name	DataDelegate NotifyInventory
Delegate	void DataDelegate (string data)
Return	void
Argument	string data;
Operation	Event for error notification from Swing-U to host
Example	<pre>Swing.NotifyError += new SwingLibrary.DataDelegate(Swing_NotifyError); ... void Swing_NotifyError(string data) { rTB_error.Clear(); if (data.Equals("No Error")) { Console.WriteLine("success"); } }</pre>

		<pre> else { Console.WriteLine(string.Format("Error: {0}", data)); } }</pre>
Error List	Common	"No Error"
	Protocol Error	"Invalid protocol" "Invalid parameter" "Unknown command" "Operation failed"
	Mac Error	"Handle mismatch" "CRC error" "No tag reply" "Invalid password" "Zero kill password" "Tag lost" "Command format error" "Read count invalid" "Out of retries" "Operation failed"
	Backscatter Error	"General error" "No memory" "Memory locked" "Insufficient power" "Unknown error"

5.5. Library Parameters

API support two parameters



5.5.1. IsOpen

Parameter	<code>bool IsOpen</code>
Operation	Host can optain connection state of serial com port for Swing-U.
Example	<pre>private void button_com_open_Click(object sender, EventArgs e) { Swing.Open("COM3"); if (Swing.IsOpen) { Console.WriteLine("{0} is open successfully", Swing.PortName); } else { Console.WriteLine("Failed to open {0}", Swing.PortName); } }</pre>

5.5.2. PortName

Parameter	<code>string PortName</code>
Operation	Host can optain serial com port name for Swing-U.
Example	<pre>private void button_com_open_Click(object sender, EventArgs e) { Swing.Open("COM3"); if (Swing.IsOpen) { Console.WriteLine("{0} is open successfully", Swing.PortName); } else { Console.WriteLine("Failed to open {0}", Swing.PortName); } }</pre>

5.6. Functions

Category	Function	Action
Connection Control	<code>≡ ConnectionClose()</code> <code>≡ ConnectionOpen(string)</code> <code>≡ ConnectionOpen(string, int)</code>	Control connection of Swing-U
Hardware Parameters Control	<code>≡ ReportAllInformation()</code> <code>≡ ReportBatteryStatus()</code> <code>≡ ReportBuzzerVolume()</code> <code>≡ ReportChargeMode()</code> <code>≡ ReportContinuous()</code> <code>≡ ReportDeviceName()</code> <code>≡ ReportFindStepUnit()</code> <code>≡ ReportFindThreshold()</code> <code>≡ ReportFlashLock()</code> <code>≡ ReportInventoryMode()</code> <code>≡ ReportInventoryTimeout()</code> <code>≡ ReportLCDControlMode()</code> <code>≡ ReportMenuEnable()</code> <code>≡ ReportModelNumber()</code> <code>≡ ReportRFPower()</code> <code>≡ ReportSerialNumber()</code> <code>≡ ReportSwingMode()</code> <code>≡ ReportTagCount()</code> <code>≡ ReportTagList()</code> <code>≡ ReportTagReportMode()</code> <code>≡ ReportVersion()</code>	Host can request current parameter state to Swing After response of Swing, library generate event: NotifyParameterChanged

	<pre>=♀ GetBarcodeType() =♀ GetBatteryRate() =♀ GetBatteryVolt() =♀ GetBuzzerVolume() =♀ GetChargeMode() =♀ GetContinuous() =♀ GetContinuousSync() =♀ GetCustomerCode() =♀ GetDeviceName() =♀ GetFindStepUnit() =♀ GetFindThreshold() =♀ GetFlashLock() =♀ GetInventoryMode() =♀ GetInventoryStatus() =♀ GetInventoryTimeout() =♀ GetLCDControlMode() =♀ GetMenuEnable() =♀ GetModelNumber() =♀ GetRFPower() =♀ GetSerialNumber() =♀ GetSwingMode() =♀ GetSwingOnLineCheck() =♀ GetTagCount() =♀ GetTagReportMode() =♀ GetVersionFW() =♀ GetVersionHW()</pre>	After event receiving: NotifyParameterChanged, then application can obtain new state of parameters
	<pre>=♀ SetBuzzerVolume(SwingLibrary.SwingAPI.BuzzerVolume) =♀ SetContinuous(SwingLibrary.SwingAPI.ContinuousMode) =♀ SetCustomerCode(int) =♀ SetFindStepUnit(int) =♀ SetFindThreshold(int) =♀ SetFlashLock(bool) =♀ SetInventoryMode(SwingLibrary.SwingAPI.InventoryMode) =♀ SetInventoryTimeout(int) =♀ SetLCDControlMode(SwingLibrary.SwingAPI.LCDControlMode) =♀ SetLCDCount(int) =♀ SetLCDText(int, string) =♀ SetMenuEnable(bool) =♀ SetRFPower(int) =♀ SetSerialNumber(string) =♀ SetSwingOnLineCheck(bool) =♀ SetTagReportMode(SwingLibrary.SwingAPI.TagReportMode)</pre>	Host can request to change state of Swing-U parameters
UHF RFID Control	<pre>=♀ InventoryStart() =♀ InventoryStatus() =♀ InventoryStop() =♀ LockTag(SwingLibrary.SwingAPI.AccessPermissionType, SwingLibrary.Sw =♀ MemoryRead(SwingLibrary.SwingAPI.MemoryBank, int, int) =♀ MemoryWrite(SwingLibrary.SwingAPI.MemoryBank, int, int, string)</pre>	EPC C1G2 operation

Tag List Control	 TagListAdd(string)  TagListClear()	Host can control tag list of Swing-U
-------------------------	--	--------------------------------------

5.6.1. Connection control

✓ ConnectionOpen

Function	<code>bool ConnectionOpen(string portName)</code>	
Return	<code>bool</code>	<code>true</code> when success, <code>false</code> when fail
Argument	<code>string portName</code>	Serial port name connected to Swing-U
Operation	Request communication start to Swing-U	
Example	<pre>private void button_com_open_Click(object sender, EventArgs e) { if (Swing.ConnectionOpen("COM3")) { Console.WriteLine("{0} is open successfully", Swing.PortName); } else { Console.WriteLine("Failed to open {0}", Swing.PortName); } }</pre>	

Function	<code>bool ConnectionOpen(string portName, int batteryWarningRate)</code>	
Return	<code>bool</code>	<code>true</code> when success, <code>false</code> when fail
Argument	<code>string portName</code>	Serial port name connected to Swing-U
	<code>int batteryWarningRate</code>	Battery Warning Rate to Swing-U.(default : 25)
Operation	Request communication start to Swing-U	
Example	<pre>private void button_com_open_Click(object sender, EventArgs e) { if (Swing.ConnectionOpen("COM3", 25)) { Console.WriteLine("{0} is open successfully", Swing.PortName); } else { Console.WriteLine("Failed to open {0}", Swing.PortName); } }</pre>	

✓ ConnectionClose

Function	bool ConnectionClose()	
Return	bool	true when success, false when fail
Argument	void	None
Operation	Request communication stop to Swing-U	
Example	<pre>private void button_com_close_Click(object sender, EventArgs e) { if (Swing.ConnectionClose()) { Console.WriteLine("{0} is closed successfully", Swing.PortName); } else { MessageBox.Show("Error on closing"); } }</pre>	

5.6.2. Hardware parameter control

- ✓ Report Hardware All Parameter Information

Function	<code>void ReportAllInformation()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the All information. After receiving NotifyParameterChanged event, application cant get the infromation.	
Example	... <code>Swing.ReportAllInformation();</code> ...	

- ✓ Remaining Battery Level

Function	<code>void ReportBattertStatus()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the information of battery level. After receiving NotifyParameterChanged event, application cant get the infromation.	
Example	... <code>Swing.ReportBattertStatus();</code> ...	

Function	<code>int GetBatteryRate()</code>	
Return	<code>int</code>	Percent of remaining battery
Argument	<code>void</code>	None
Example	... <code>label_percent_Text = string.Format("{0:000 [%%]}", Swing.GetBatteryRate());</code> ...	

Function	<code>double GetBatteryVolt()</code>	
Return	<code>double</code>	Volt of remaining battery
Argument	<code>void</code>	None
Example	... <code>label_volt.Text = string.Format("Volts: {0:F3} [V]", Swing.GetBatteryVolt());</code> ...	

✓ Buzzer Volume

Function	<code>void ReportBuzzerVolume()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	<p>Request to Swing-U report the volume of buzzer.</p> <p>After receiving NotifyParameterChanged event, application cant get the information</p>	
Example	<pre>... Swing.ReportBuzzerVolume(); ...</pre>	

Function	<code>BuzzerVolume GetBuzzerVolume()</code>		
Return	<code>enum BuzzerVolume</code>	VIBRATION	No sound, vibration only
		MUTE	No sound, no vibration
		MIN	Minimum volume
		NORMAL	Normal volume
		MAX	Maximum volume
Argument	<code>void</code>	None	
Example	<pre>... switch (Swing.GetBuzzerVolume()) { case SwingLibrary.SwingAPI.BuzzerVolume.MAX: radioButton_vol_max.Checked = true; break; case SwingLibrary.SwingAPI.BuzzerVolume.MIN: radioButton_vol_min.Checked = true; break; case SwingLibrary.SwingAPI.BuzzerVolume.MUTE: radioButton_vol_mute.Checked = true; break; case SwingLibrary.SwingAPI.BuzzerVolume.NORMAL: radioButton_vol_normal.Checked = true; break; case SwingLibrary.SwingAPI.BuzzerVolume.VIBRATION: radioButton_vol_vib.Checked = true; break; ... }</pre>		

Function	<code>void SetBuzzerVolume(BuzzerVolume volume)</code>	
Return	<code>void</code>	None
Argument	<code>BuzzerVolume</code>	Default value is <code>BuzzerVolume.MAX</code>
Example	<pre>... if (radioButton_vol_min.Checked) Swing.SetBuzzerVolume(SwingLibrary.SwingAPI.BuzzerVolume.MIN); ...</pre>	

✓ Battery Charging Status

Function	<code>void ReportChargeMode()</code>		
Return	<code>void</code>	없음	
Argument	<code>void</code>	없음	
Operation	Request to Swing-U report the status of charging. After receiving NotifyParameterChanged event, application can't get the information		
Example	<pre>... Swing.ReportChargeMode(); ...</pre>		

Function	<code>ChargingMode GetChargeMode()</code>		
Return	<code>enum ChargingMode</code>	ERROR	Battery error
		CHARGING	On charging
		FULL	Full charged
		DISCHARGING	On using
Argument	<code>void</code>	None	
Example	<pre>... switch (Swing.GetChargeMode()) { case SwingLibrary.SwingAPI.ChargingMode.CHARGING: radioButton_btr_charging.Checked = true; break; case SwingLibrary.SwingAPI.ChargingMode.DISCHARGING: radioButton_btr_discharging.Checked = true; break; case SwingLibrary.SwingAPI.ChargingMode.FULL: radioButton_btr_full.Checked = true; break; default: // battery error break; } ...</pre>		

✓ Continuous Mode

Function	<code>void ReportContinuous()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the mode of continuous RFID read. After receiving NotifyParameterChanged event, application can't get the information	
Example	... <code>Swing.ReportContinuous();</code> ...	

Function	<code>ContinuousMode GetContinuous()</code>		
Return	<code>enum ContinuousMode</code>	SINGLE	Read single tag
		CONTINUOUS	Read tag continuously
Argument	<code>void</code>	None	
Example	... <code>switch (Swing.GetContinuous())</code> <code>{</code> <code> case SwingLibrary.SwingAPI.ContinuousMode.SINGLE:</code> <code> ...</code> <code> break;</code> <code> case SwingLibrary.SwingAPI.ContinuousMode.CONTINUOUS:</code> <code> ...</code> <code> break;</code> <code> default:</code> <code> break;</code> <code>}</code> <code>...</code>		

Function	<code>void SetContinuous(ContinuousMode mode)</code>	
Return	<code>void</code>	None
Argument	<code>ContinuousMode</code>	Default value is <code>ContinuousMode.CONTINUOUS</code>
Example	... <code>if (radioButton_ac_multi.Checked)</code> <code> Swing.SetContinuous(SwingLibrary.SwingAPI.ContinuousMode.CONTINUOUS);</code> ...	

✓ Device Name

Function	<code>void ReportDeviceName()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the Device name. After receiving NotifyParameterChanged event, application cant get the information	
Example	<pre>... Swing.ReportDeviceName(); ...</pre>	

Function	<code>int GetDeviceName()</code>	
Return	<code>string</code>	Swing-U Device Name
Argument	<code>void</code>	None
Example	<pre>... string deviceName = Swing.GetContinuous(); ...</pre>	

Function	<code>void SetDeviceName (string DeviceName)</code>	
Return	<code>void</code>	None
Argument	<code>string DeviceName</code>	SwingU device name
Operation	Set the device name to Swing-U	
Example	<pre>... string devicename = maskedTextBox_BT_Device_Name.Text; Swing.SetDeviceName(devicename); ...</pre>	

✓ Find Step Unit

Function	<code>void ReportFindStepUnit()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the RF output decrement unit[dBm] for find mode After receiving NotifyParameterChanged event, application can't get the information	
Example	<pre>Swing.ReportFindStepUnit(); ...</pre>	

Function	<code>int GetFindStepUnit()</code>	
Return	<code>int</code>	Step unit of RF power decresement in dBm
Argument	<code>void</code>	None
Example	<pre>... int find_step = Swing.GetFindStep unit(); ...</pre>	

Function	<code>void SetFindStepUnit (int step)</code>	
Return	<code>void</code>	None
Argument	<code>int step</code>	Default value: 3 [dBm]
Example	<pre>... Swing.SetFindStepUnit(6); // 6dBm step ...</pre>	

✓ Find Target ID adding

Function	<code>void SetFindTargetUID(int index, string UID)</code>	
Return	<code>void</code>	None
Argument	<code>int index</code>	index of finding target table
	<code>string UID</code>	UID for finding target, UID means PC+EPC ID
Operation	Set the finding target UID to Swing-U, works in InventoryMode . FIND mode	
Example	<pre>... string target_tag_id = textBox_find_id.Text.Trim(); Swing.SetFindTargetUID(1, target_tag_id); ...</pre>	

✓ Tag Find Threshold

Function	<code>void ReportFindThreshold()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the RF output threshold[dBm] level for find mode After receiving NotifyParameterChanged event, application cant get the information.	
Example	<pre>... Swing.ReportFindThreshold(); ...</pre>	

Function	<code>int GetFindThreshold()</code>	
Return	<code>int</code>	Minimum RF power of target tag finding
Argument	<code>void</code>	None
Example	<pre>... int find_threshold = Swing.GetFindThreshold(); ...</pre>	

Function	<code>void SetFindThreshold(int dBm)</code>	
Return	<code>void</code>	None
Argument	<code>int dBm</code>	Minimum RF power of target tag finding in dBm
Example	<pre>... Swing.SetFindThreshold(17); // set threshold to 17 dBm ...</pre>	

✓ Session

Function	<code>void ReportSession()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	<p>Request to Swing-U report the Session.</p> <p>After receiving NotifyParameterChanged event, application cant get the information</p>	
Example	<pre>... Swing.ReportSession(); ...</pre>	

Function	<code>Session GetSession()</code>	
Return	<code>Session</code>	Swing-U Session
Argument	<code>void</code>	None
Example	<pre>... Session Session = Swing.GetSession(); ...</pre>	

Function	<code>void SetSession (int session)</code>	
Return	<code>void</code>	None
Argument	<code>int session</code>	SwingU Session Number
Operation	Set the Session to Swing-U	
Example	<pre>... int session = comboBox_session.SelectedIndex; Swing. SetSession (session); ...</pre>	

✓ Flag

Function	<code>void ReportFlag()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the Flag. After receiving NotifyParameterChanged event, application cant get the information	
Example	... <code>Swing.ReportFlag();</code> ...	

Function	<code>Session GetFlag()</code>	
Return	<code>Flag</code>	Swing-U Flag
Argument	<code>void</code>	None
Example	... <code>Flag Flag = Swing.GetFlag();</code> ...	

Function	<code>void SetFlag (int Flag)</code>	
Return	<code>void</code>	None
Argument	<code>Flag Flag</code>	SwingU Flag Number
Operation	Set the Session to Swing-U	
Example	... <code>int flag = comboBox_Flag.SelectedIndex;</code> <code>Swing.SetFlag((SwingLibrary.SwingAPI.Flag)flag);</code> ...	

✓ Inventory Mode

Function	<code>void ReportInventoryMode()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	<p>Request to Swing-U report the mode of inventory.</p> <p>After receiving NotifyParameterChanged event, application cant get the information.</p>	
Example	<pre>... Swing.ReportInventoryMode(); ...</pre>	

Function	<code>InventoryMode GetInventoryMode()</code>		
Return	<code>enum InventoryMode</code>	<code>INVENTORY</code>	Normal inventory
		<code>INVENTORY_WITHUSER</code>	EPC + User memory data
		<code>SEARCH_SINGLE</code>	Seaching single tag
		<code>SEARCH_MULTI</code>	Seaching multiple tags with list
		<code>SEARCH_WILDCARD</code>	Seaching multiple tags with wildcard character
Argument	<code>void</code>	None	
Example	<pre>... switch (Swing.GetInventoryMode()) { case SwingLibrary.SwingAPI.InventoryMode.INVENTORY: ... break; ... case SwingLibrary.SwingAPI.InventoryMode.SEARCH_WILDCARD: ... break; default: break; } ...</pre>		

Function	<code>void SetInventoryMode(InventoryMode mode)</code>	
Return	<code>void</code>	None
Argument	<code>InventoryMode</code>	<code>enum InventoryMode {</code>

		INVENTORY_NORMAL, INVENTORY_WITHUSER, SEARCH_SINGLE, SEARCH_MULTI, SEARCH_WILDCARD }
Example		<pre>... if (chk_find) { Swing.SetInventoryMode(SwingLibrary.SwingAPI.InventoryMode.INVNETORY); Swing.InventoryStart(); } ...</pre>

✓ Inventory Timeout

Function	<code>void ReportInventoryTimeout()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the inventory timeout. After receiving NotifyParameterChanged event, application cant get the information.	
Example	<pre>... Swing.ReportInventoryTimeout(); ...</pre>	

Function	<code>int GetInventoryTimeout()</code>	
Return	<code>int</code>	Inventory Timeout
Argument	<code>void</code>	None
Example	<pre>... int inventoryTimeout = Swing.GetInventoryTimeout(); ...</pre>	

Function	<code>void SetInventoryTimeout(int timeout)</code>	
Return	<code>void</code>	None
Argument	<code>int</code>	Inventory Timeout of Swing-U(Less than 65000, default is 60.)
Example	<pre>... Swing.SetInventoryTimeout(180); ...</pre>	

✓ Inventory Status

Function	InventoryStatusMode GetInventoryStatus()		
Return	enum InventoryStatusMode	WAIT_STATUS	Not Reading
		ON_INVENTORY	Reading Swing-U
Argument	void	None	
Example	<pre>... switch (Swing.GetInventoryStatusMode()) { case SwingLibrary.SwingAPI.InventoryStatusMode. WAIT_STATUS: ... break; ... case SwingLibrary.SwingAPI.InventoryStatusMode. ON_INVENTORY: ... break; default: break; } ...</pre>		

✓ LCD control mode

Function	<code>void ReportLCDControlMode()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	<p>Request lcd control mode to Swing-U.</p> <p>After receiving NotifyParameterChanged event, application can get the information.</p>	
Example	<pre>... Swing. ReportLCDControlMode(); ...</pre>	

Function	<code>LCDControlMode GetLCDControlMode()</code>		
Return	<code>enum LCDControlMode</code>	<code>SWING</code>	LCD controled by Swing-U
		<code>HOST_COUNT</code>	LCD display large counter setted by host
		<code>HOST_TEXT</code>	LCD display user text setted by host
Argument	<code>void</code>	None	
Example	<pre>... switch (Swing.GetLCDControlMode()) { case SwingLibrary.SwingAPI.LCDControlMode.HOST_COUNT: ... break; case SwingLibrary.SwingAPI.LCDControlMode.HOST_TEXT: ... break; default: break; } ...</pre>		

Function	<code>void SetLCDControlMode(LCDControlMode mode)</code>	
Return	<code>void</code>	None
Argument	<code>LCDControlMode</code>	<code>enum LCDControlMode</code> <code>{ SWING, HOST_COUNT, HOST_TEXT }</code>
Example	<pre>... if (radioButton_lcd_swing.Checked) Swing. SetLCDControlMode(SwingLibrary.SwingAPI.LCDControlMode.SWING); ...</pre>	

✓ LCD contents setting

Function	<code>void SetLCDCount(int count)</code>	
Return	<code>void</code>	None
Argument	<code>int</code>	count
Operation	Setting the count will be displayed in LCDControlMode . HOST_COUNT mode.	
Example	<pre>... Swing.SetLCDCount(127); ...</pre>	

Function	<code>void SetLCDText(int line, string text)</code>	
Return	<code>void</code>	None
Argument	<code>int</code> <code>string</code>	line text
Operation	Setting the user text will be displayed in LCDControlMode . HOST_TEXT mode. LCD has lines and can displays 16 characters in each line	
Example	<pre>... Swing.SetLCDText(1, "Hello"); ...</pre>	

✓ Reader Menu setting

Function	<code>void ReportMenuEnable()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the Menu Enable of Swing-U After receiving NotifyParameterChanged event, application can't get the information.	
Example	<pre>... Swing.ReportMenuEnable(); ...</pre>	

Function	<code>bool GetMenuEnable()</code>	
Return	<code>bool</code>	Menu Enable of Swing-U
Argument	<code>void</code>	None
Example	<pre>... bool menuEnable = Swing.GetMenuEnable(); ...</pre>	

Function	<code>void SetMenuEnable(bool enable)</code>	
Return	<code>void</code>	None
Argument	<code>bool</code>	Menu Enable of Swing-U
Example	<pre>... Swing.SetMenuEnable(true); ...</pre>	

✓ Reader RF power

Function	<code>void ReportRFPower()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the RF ouput power of RFID reader After receiving NotifyParameterChanged event, application cant get the infromation.	
Example	<pre>... Swing.ReportRFPower(); ...</pre>	

Function	<code>int GetRFPower()</code>	
Return	<code>int</code>	RF attenuation of reader
Argument	<code>void</code>	None
Example	<pre>... int atten = Swing.GetRFPower(); int rf_power = 30 - atten; // real RF out power ...</pre>	

Function	<code>void SetRFPower(int attenuation)</code>	
Return	<code>void</code>	None
Argument	<code>int</code>	RF attenuation of reader
Example	<pre>... Swing.SetRFPower(3); // set RF powert to 27dbm ...</pre>	

✓ Serial Number

Function	<code>void ReportSerialNumber()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the Serial Number of Swing-U After receiving NotifyParameterChanged event, application cant get the information.	
Example	<pre>... Swing.ReportSerialNumber(); ...</pre>	

Function	<code>int GetSerialNumber()</code>	
Return	<code>string</code>	Serial Number of Swing-U
Argument	<code>void</code>	None
Example	<pre>... string serialNumber = Swing.GetSerialNumber(); ...</pre>	

✓ Tag Count

Function	<code>void ReportTagCount()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	<p>Request tag count to Swing-U.</p> <p>After receiving NotifyParameterChanged event, application can get the information.</p>	
Example	<pre>... Swing.ReportTagCount(); ...</pre>	

Function	<code>void Send_count()</code>	
Return	<code>void</code>	
Argument	<code>long</code>	Now tag count
Example	<pre>... Swing.Send_count(1); ...</pre>	

Function	<code>int Send_count()</code>	
Return	<code>int</code>	Tag count int Swing-U
Argument	<code>void</code>	None
Example	<pre>... int count = Swing.GetTagCount(); ...</pre>	

✓ Tag List

Function	<code>void ReportTagList()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	<p>Request report all tag in memory to Swing-U.</p> <p>with NotifyInventory event, application can get the information.</p>	
Example	<pre>... Swing.ReportTagList(); ...</pre>	

Function	<code>void Epc_BitLengthSet()</code>	
Return	<code>void</code>	None
Argument	<code>int</code>	Max EPC length(bit)
Operation	Set max EPC length Swing-U generate NotifyError event after operation.	
Example	<pre>... Int epc_length = 256; Swing.EpcBitLengthSet(epc_length); ... void Swing_NotifyError(string data) { rTB_error.Clear(); if (data.Equals("No Error")) { Console.WriteLine("success"); } else { Console.WriteLine(string.Format("Error: {0}", data)); } }</pre>	

Function	<code>void Max_count ()</code>	
Return	<code>void</code>	None
Argument	<code>int</code>	Same tag filtering size
Operation	Use only when host connected via BT If host is not connected then the filtering size dependant with Max EPC length Swing-U generate NotifyError event after operation.	
Example	<pre>... Int max_count = 1024; Swing.MaxCountSet(max_count); ... void Swing_NotifyError(string data) { rTB_error.Clear(); if (data.Equals("No Error")) { Console.WriteLine("success"); } else { Console.WriteLine(string.Format("Error: {0}", data)); } }</pre>	

✓ Version

Function	<code>void ReportVersion()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request H/W, F/W version to Swing-U. After receiving NotifyParameterChanged event, application cant get the information.	
Example	<pre>... Swing.ReportVersion(); ...</pre>	

Function	<code>string GetVersionHW()</code>	
Return	<code>string</code>	H/W version
Argument	<code>void</code>	None
Operation	Get version of Swing-U H/W	
Example	<pre>... string hw_ver = Swing.GetVersionHW(); ...</pre>	

Function	<code>string GetVersionFW()</code>	
Return	<code>string</code>	F/W version
Argument	<code>void</code>	None
Operation	Get version of Swing-U F/W	
Example	<pre>... string fw_ver = Swing.GetVersionFW(); ...</pre>	

✓ Language

Function	<code>string setLanguage()</code>	
Return	<code>void</code>	None
Argument	<code>int</code>	0 : English 1 : Chinese
Operation	Set language of Swing-U	
Example	<pre>... int language = comboBox_Language_list.SelectedIndex; Swing.setLanguage(language); ...</pre>	

5.7. UHF RFID control

5.7.1. InventoryStart

Function	<code>void InventoryStart()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request start of UHF RFID inventory to Swing-U. Library generate NotifyButtonEvent.	
Example	<pre>... Swing.InventoryStart(); ... void NotifyButtonEvent(SwingLibrary.ButtonEvent buttonType) { switch (buttonType) { ... case SwingLibrary.ButtonEvent.READSTART: // Inventory started break; ... } }</pre>	

5.7.2. InventoryStop

Function	<code>void InventoryStop()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request stop of UHF RFID inventory to Swing-U. Library generate NotifyButtonEvent.	
Example	<pre>... Swing.InventoryStop(); ... void NotifyButtonEvent(SwingLibrary.ButtonEvent buttonType) { switch (buttonType) { ... case SwingLibrary.ButtonEvent.READSTOP: // Inventory stop break; ... } }</pre>	

5.7.3. MemoryRead

Function	<code>void MemoryRead(MemoryBank bank, int offset, int count)</code>	
Return	<code>void</code>	None
Argument	<code>MemoryBank</code> bank <code>int</code> offset <code>int</code> count	RESERVED, EPC, TID, USER Start address of memory in word unit Address length of memory in word unit
Operation	Request reading memory data of tag. Swing-U generate NotifyError event after operation. Read data has format of <code>string</code> : "UID" + "M" + "DATA"	
Example	<pre> ... SwingLibrary.SwingAPI.MemoryBank bank = SwingLibrary.SwingAPI.MemoryBank.EPC int offset = 2; int count = 6; Swing. MemoryRead(bank, offset, count); void Swing_NotifyInventory(string data) { if(data.Contains("M")) // Memory Read string[] datas = data.Split('M'); string id = datas[0]; string memory_data = datas[1]; else // EPC ID read ... } </pre>	

5.7.4. MemoryWrite

Function	<code>void MemoryWrite(MemoryBank bank, int offset, int count, string data)</code>	
Return	<code>void</code>	None
Argument	<code>MemoryBank</code> bank <code>int</code> offset <code>int</code> count <code>string</code> data	RESERVED, EPC, TID, USER Start address of memory in word unit Address length of memory in word unit Data for write
Operation	Request writing data to memory of tag. Swing-U generate NotifyError event after operation.	
Example	<pre> ... SwingLibrary.SwingAPI.MemoryBank bank = SwingLibrary.SwingAPI.MemoryBank.EPC int offset = 5; int count = 3; Swing. MemoryWrite(bank, offset, count, "111122223333"); void Swing_NotifyError(string data) { rTB_error.Clear(); if (data.Equals("No Error")) { Console.WriteLine("success"); } else { Console.WriteLine(string.Format("Error: {0}", data)); } } </pre>	

5.7.5. MemorySelect

Function	<code>void MemorySelect(MemoryBank bank, int offset, int count, string data)</code>	
Return	<code>void</code>	None
Argument	<code>MemoryBank</code> bank <code>int</code> offset <code>int</code> count <code>string</code> data	RESERVED, EPC, TID, USER Start address of memory in word unit Address length of memory in word unit Data for Select
Operation	Request select data to memory of tag. Swing-U generate NotifyError event after operation.	
Example	<pre> ... SwingLibrary.SwingAPI.MemoryBank bank = SwingLibrary.SwingAPI.MemoryBank.EPC int offset = 2; int count = 6; Swing. MemorySelect(bank, offset, count, "123456789012345678901234 void Swing_NotifyError(string data) { rTB_error.Clear(); if (data.Equals("No Error")) { Console.WriteLine("success"); } else { Console.WriteLine(string.Format("Error: {0}", data)); } } </pre>	

5.7.6. MemoryLock

Function	<code>void MemoryWrite(int kill, int access, int epc, int tid, int user, string access_pwd)</code>			
Return	<code>void</code>	None		
Argument	<code>int kill</code>	Lock action for Kill password memory	0	Accessible
			1	Always accessible
			2	Secured accessible
			3	Always not accessible
			4	No change
	<code>int access</code>	Lock action for Access password memory	0	Accessible
			1	Always accessible
			2	Secured accessible
			3	Always not accessible
			4	No change
	<code>int epc</code>	Lock action for EPC memory	0	Accessible
			1	Always accessible
			2	Secured accessible
			3	Always not accessible
			4	No change
	<code>int tid</code>	Lock action for TID memory	0	Accessible
			1	Always accessible
			2	Secured accessible
			3	Always not accessible
			4	No change
	<code>int user</code>	Lock action for USER memory	0	Accessible
			1	Always accessible
			2	Secured accessible
			3	Always not accessible
			4	No change
	<code>string access_pwd</code>	Access password	Tag Access Password	

Operation	Request locked to memory of tag. Swing-U generate NotifyError event after operation
Example	<pre>... int killpwd = 0; //set the kill pwd memory accessible int accpwd = 0; //set the access pwd memory accessible int epc = 2; //set the EPC memory secured accessible int tid = 0; //set the TID memory accessible int user = 0; //set the USER memory accessible string pwd = "00000000"; //password of tag Swing.LockTag(killpwd, accpwd, epc, tid, user, pwd); void Swing_NotifyError(string data) { rTB_error.Clear(); if (data.Equals("No Error")) { Console.WriteLine("success"); } else { Console.WriteLine(string.Format("Error: {0}", data)); } }</pre>

5.8. Tag List control

5.8.1. TagListAdd

Function	<code>void TagListAdd(int index, string UID)</code>	
Return	<code>void</code>	None
Argument	<code>string</code> <code>UID</code>	EPC ID(PC including) for add to Swing-U
Operation	<p>Request add specific tag <code>UID</code> to list.</p> <p>Swing-U generate <code>NotifyError</code> event after operation.</p> <p>"No Error": Success</p> <p>"Operation failed": Fail</p>	
Example	<pre>... Swing.TagListAdd("3000111122223333444455556666"); ...</pre>	

5.8.2. Access Password Set

Function	<code>void AccessPwdSet (string access_pwd)</code>	
Return	<code>void</code>	None
Argument	<code>string</code> <code>access</code>	Access password
Operation	<p>Set access password set</p> <p>Swing-U generate <code>NotifyError</code> event after operation.</p>	
Example	<pre>... string access_pwd = "00000000"; Swing.AccessPwdSet (access_pwd); ... void Swing_NotifyError(string data) { rTB_error.Clear(); if (data.Equals("No Error")) { Console.WriteLine("success"); } else { Console.WriteLine(string.Format("Error: {0}", data)); } }</pre>	

5.8.3. TagListClear

Function	void TagListClear()	
Return	void	None
Argument	void	None
Operation	Request tag list clear to Swing-U	
Example	<pre>... Swing.TagListClear(); ...</pre>	

6. Active-X Library API

6.1. Overview

This document is library reference of Swing-U(NTRM-U-2) for .NET C# library on windows operation system.

6.2. Revision History

Version	Date	Description
1.0	2014-06-12	First Release

6.3. Swing-U Library contents

Swing-U Library contains as follows:

- ✓ SwingLibrary.dll
 - : Dynamic Link Library for functions and event to control Swing-U, UHF RFID reader
- ✓ SwingU_Demo.exe
 - : Demo application using SwingLibrary.dll

※ Notice

4. This API is developed on Microsoft Visual Studio 2008 platform.
5. This API can operate on .NET Framework 2.0 or above.
6. Sample Project is also developed on Visual Studio 2008, using C#.

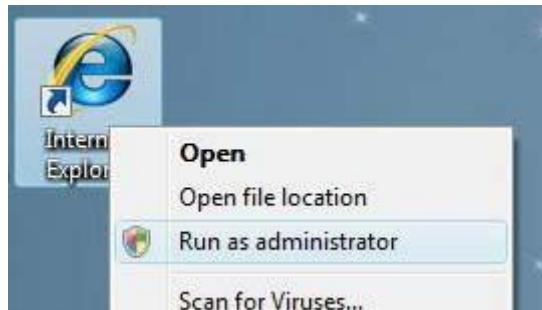
6.4. Installation of Active-X

※ Notice

1. Example environment is Windows 8 and internet explorer 10.
2. If host use Windows XP, host have already installed .NET Framework 2.0 or above.

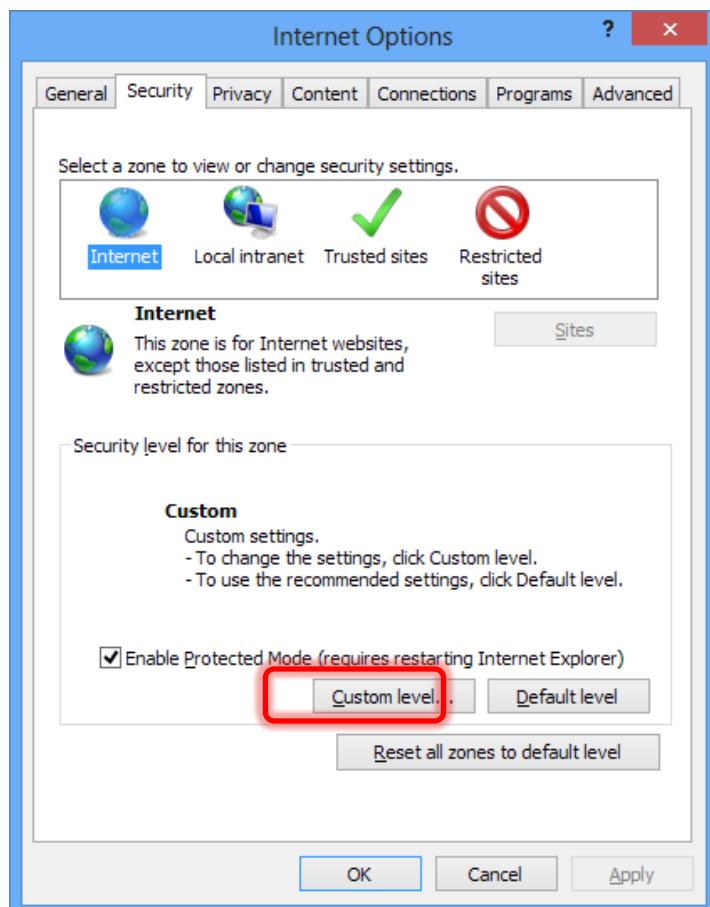
6.4.1. Run internet explorer

For the installation of Active-X, host must start internet explorer as administrator.



6.4.2. Internet explorer setting

Set the Security level of explorer to custom level.

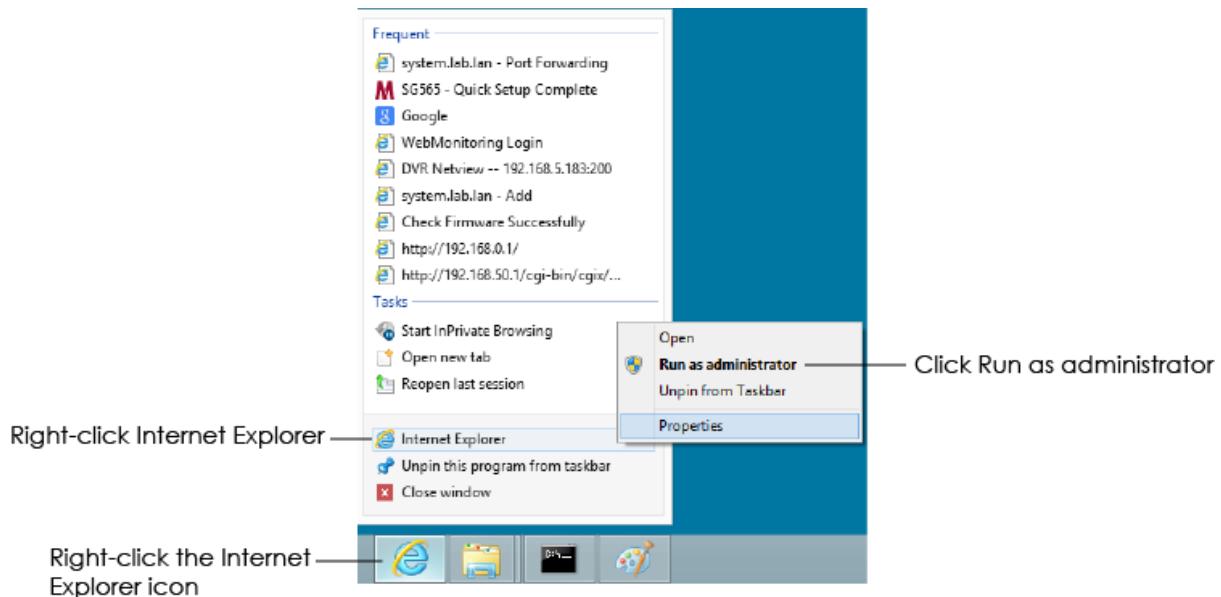


And then, set the as follows:

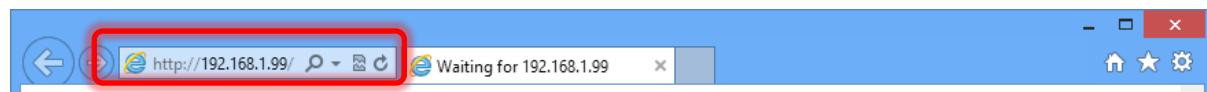
- ActiveX controls and plug-ins
 - Allow ActiveX Filtering
 - Disable
 - Enable
 - Allow previously unused ActiveX controls to run without prompt
 - Disable
 - Enable
 - Allow Scriptlets
 - Disable
 - Enable
 - Prompt
 - Automatic prompting for ActiveX controls
 - Disable
 - Enable
 - Binary and script behaviors
 - Administrator approved
 - Disable
 - Enable
 - Display video and animation on a webpage that does not use ActiveX
 - Disable
 - Enable
 - Download signed ActiveX controls
 - Disable
 - Enable (not secure)
 - Prompt (recommended)
 - Download unsigned ActiveX controls
 - Disable (recommended)
 - Enable (not secure)
 - Prompt
 - Initialize and script ActiveX controls not marked as safe for scripting
 - Disable (recommended)
 - Enable (not secure)
 - Prompt
 - Only allow approved domains to use ActiveX without prompt
 - Disable
 - Enable
 - Run ActiveX controls and plug-ins
 - Administrator approved
 - Disable
 - Enable
 - Prompt
 - Script ActiveX controls marked safe for scripting*

6.4.3. Install from web page

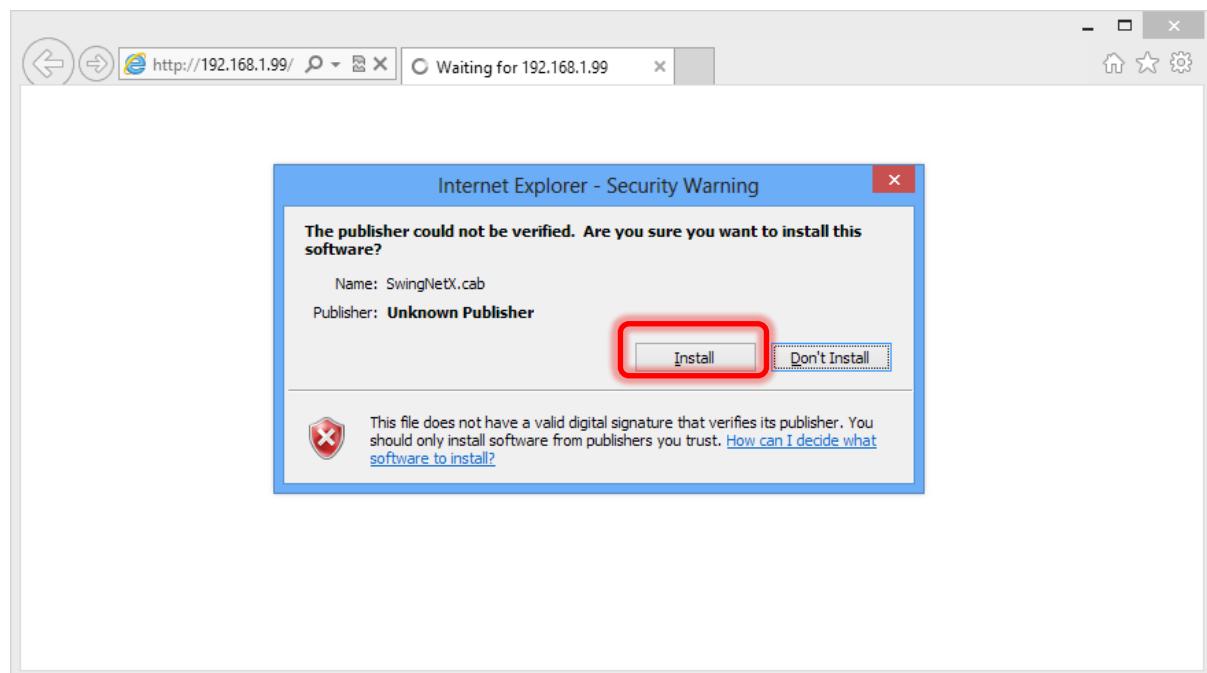
- ✓ Run internet explorer as administrator



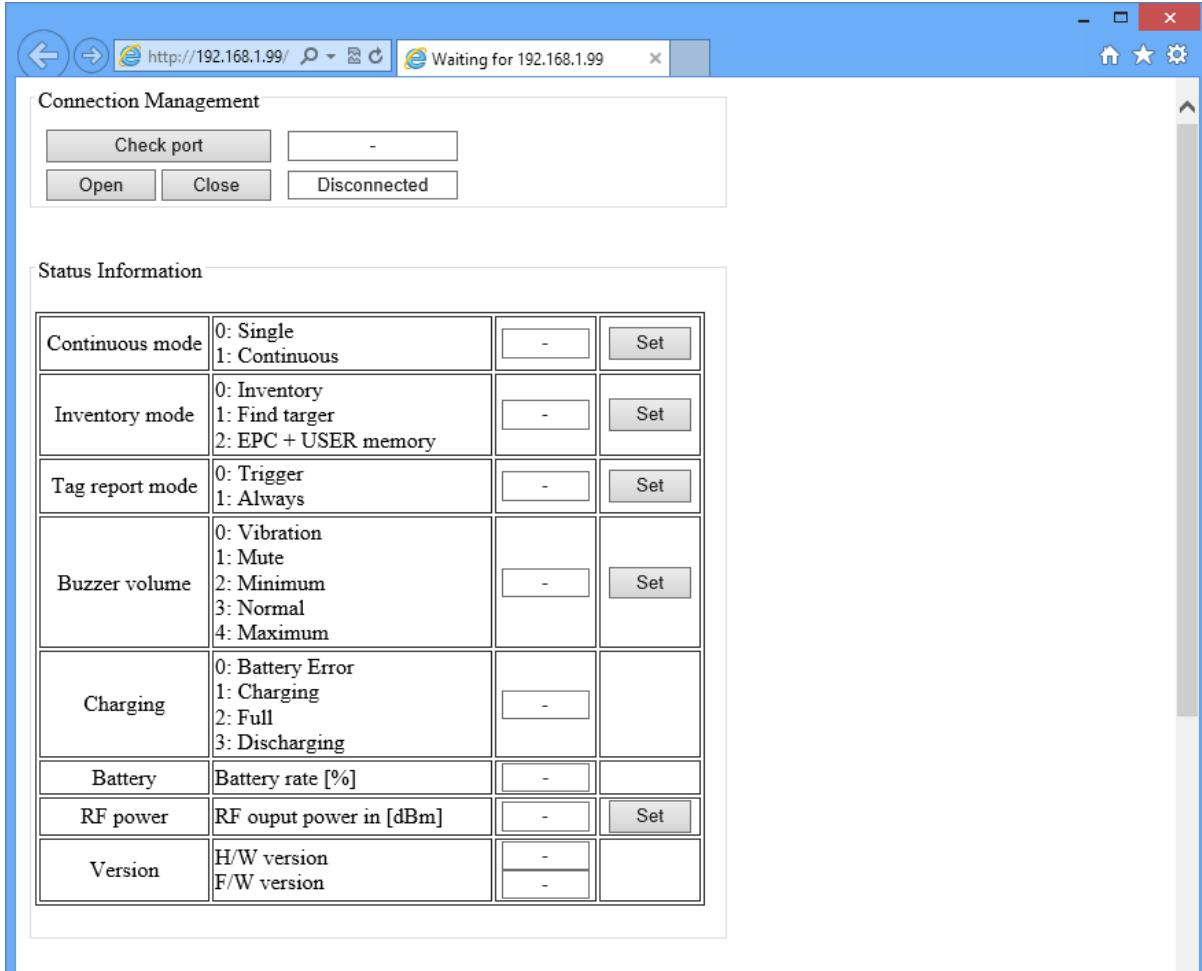
- ✓ Connect to web server
 - This example use apache server in local network.



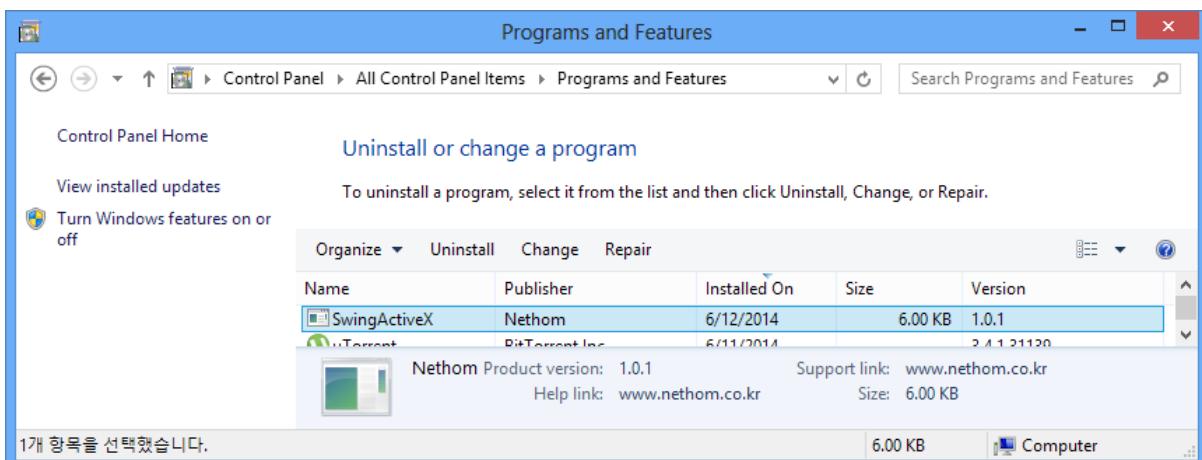
- Click Install button



- ✓ Installation complete
 - After few seconds for installation, following page will be display.

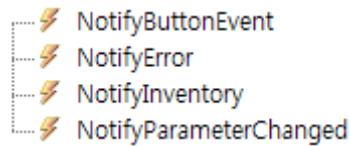


- User can see also the installed Active-X in
Control Panel>All Control Panel Items>Programs and Features



6.5. Library Event

API supports four events for Swing-U



6.5.1. NotifyParameterChanged

Name	NotifyParameterChanged
Event	<code>void NotifyParameterChanged(int type)</code>
Return	<code>void</code>
Argument	<p><code>int type</code></p> <p>0: ContinuousMode, 1: InventoryMode, 2: TagReportMode, 3: BuzzerVolume, 4: ChargingStatus, 5: BatteryStatus, 6: RFPower, 7: FindStepUnit, // Step down unit of RF output power for find mode 8: FindThreshold, // Threshold of RF output power for find mode 9: Version, 10: TagCount</p>
Operation	Event for parameter changing notification by host request or button from Swing-U to host
Example	<pre><script for="SwingNetX" event="NotifyParameterChanged(type)" language="javascript"> function SwingNetX::NotifyParameterChanged(type) { switch (type) { case 0: //ContinuousMode ... break; case 1: //InventoryMode ... break; case 2: //TagReportMode ... break; case 3: //BuzzerVolume ... break; case 4: //ChargingStatus</pre>

```
...
break;
case 5: //BatteryStatus
...
break;
case 6: //RFPower
...
break;
case 7: //FindStepUnit
break;
case 8: //FindThreshold
break;
case 9: //Version
    document.getElementById('text_ver_hw').innerText = SwingNetX.GetVersionHW();
    document.getElementById('text_ver_fw').innerText = SwingNetX.GetVersionFW();
    break;
case 10: //TagCount
break;
default:
    alert("Unkown Parameter Event");
    break;
}
}
</script>
```

6.5.2. NotifyButtonEvent

Name	NotifyButtonEvent
Event	<code>void NotifyButtonEvent(int type)</code>
Return	<code>void</code>
Argument	<code>int type</code> 0 Read starts on Swing-U by button or request 1 Read stops on Swing-U by button or request 2 Tag list cleared by three seconds long-click FN button of Swing-U 3 short click of FN button on Swing-U
Operation	Event for Read button and FN button click notification from Swing-U to host
Example	<pre><script for="SwingNetX" event="NotifyButtonEvent(type)" language="javascript"> function SwingNetX::NotifyButtonEvent(type) { switch (type) { case 0: document.getElementById('text5').innerText = "Reading"; break; case 1: document.getElementById('text5').innerText = "Ready"; break; case 2: document.getElementById('tag_list').value = ""; break; case 3: alert("User FN Button Event"); break; default: alert("Unkown Button Event"); break; } }</script></pre>

6.5.3. NotifyInventory

Name	NotifyInventory
Event	<code>void NotifyInventory(string data)</code>
Return	<code>void</code>
Argument	<code>string data</code>
Operation	Event for read tag ID and memory information from Swing-U to host
Example	<pre><script for="SwingNetX" event="NotifyInventory(data)" language="javascript"> function SwingNetX::NotifyInventory(data) { if(data.indexOf('M') != -1) { var datas = data.split('M'); add_tag(datas[0]); document.getElementById('mem_read').innerText = datas[1]; } else { add_tag(data); } } </script></pre>

6.5.4. NotifyError

Name	NotifyError		
event	<code>void NotifyError (string error)</code>		
Return	<code>void</code>		
Argument	<code>string error</code>		
Operation	Event for error notification from Swing-U to host		
Example	<pre><script for="SwingNetX" event="NotifyError(error)" language="javascript"> function SwingNetX::NotifyError(error) { document.getElementById('mem_error').innerText = error; } </script></pre>		
Error List	Common	<p>"No Error"</p> <p>"Invalid protocol"</p> <p>"Invalid parameter"</p> <p>"Unknown command"</p> <p>"Operation failed"</p>	
	Protocol Error	<p>"Handle mismatch"</p> <p>"CRC error"</p> <p>"No tag reply"</p> <p>"Invalid password"</p> <p>"Zero kill password"</p> <p>"Tag lost"</p> <p>"Command format error"</p> <p>"Read count invalid"</p> <p>"Out of retries"</p> <p>"Operation failed"</p>	
	Mac Error	<p>"General error"</p> <p>"No memory"</p> <p>"Memory locked"</p> <p>"Insufficient power"</p> <p>"Unkown error"</p>	
	Backscatter Error	<p>"General error"</p> <p>"No memory"</p> <p>"Memory locked"</p> <p>"Insufficient power"</p> <p>"Unkown error"</p>	
		<p>"General error"</p> <p>"No memory"</p> <p>"Memory locked"</p> <p>"Insufficient power"</p> <p>"Unkown error"</p>	

6.6. Library Functions

Category	Function	Action
Connection Control	<code>...≡ GetPort()</code> <code>≡ Connect()</code> <code>≡ Disconnect()</code>	Control connection of Swing-U
Hardware Parameters Control	<code>≡ ReportBatteryStatus()</code> <code>≡ ReportBuzzerVolume()</code> <code>≡ ReportChargeMode()</code> <code>≡ ReportContinuous()</code> <code>≡ ReportFindStepUnit()</code> <code>≡ ReportFindThreshold()</code> <code>≡ ReportInventoryMode()</code> <code>≡ ReportRFPower()</code> <code>≡ ReportTagCount()</code> <code>≡ ReportTagList()</code> <code>≡ ReportTagReportMode()</code> <code>≡ ReportVersion()</code>	Host can request current parameter state to Swing After response of Swing, library generate event: <code>NotifyParameterChanged</code>
	<code>≡ GetAllParameters()</code> <code>≡ GetBatteryRate()</code> <code>≡ GetBatteryVolt()</code> <code>≡ GetBuzzerVolume()</code> <code>≡ GetChargeMode()</code> <code>≡ GetContinuous()</code> <code>≡ GetFindStepUnit()</code> <code>≡ GetFindThreshold()</code> <code>≡ GetInventoryMode()</code> <code>≡ GetRFPower()</code> <code>≡ GetTagCount()</code> <code>≡ GetTagReportMode()</code> <code>≡ GetVersionFW()</code> <code>≡ GetVersionHW()</code>	After event receiving: <code>NotifyParameterChanged</code> , then application can obtain new state of parameters
	<code>≡ SetBuzzerVolume(SwingLibrary.SwingAPI.BuzzerVolume)</code> <code>≡ SetContinuous(SwingLibrary.SwingAPI.ContinuousMode)</code> <code>≡ SetFindStepUnit(int)</code> <code>≡ SetFindTargetUID(string)</code> <code>≡ SetFindThreshold(int)</code> <code>≡ SetInventoryMode(SwingLibrary.SwingAPI.InventoryMode)</code> <code>≡ SetRFPower(int)</code> <code>≡ SetTagReportMode(SwingLibrary.SwingAPI.TagReportMode)</code>	Host can request to change state of Swing-U parameters
UHF RFID Control	<code>≡ InventoryStart()</code> <code>≡ InventoryStop()</code> <code>≡ MemoryRead(SwingLibrary.SwingAPI.MemoryBank, int, int)</code> <code>≡ MemoryWrite(SwingLibrary.SwingAPI.MemoryBank, int, int,</code>	EPC C1G2 operation
Tag List Control	<code>≡ TagListAdd(string)</code> <code>≡ TagListClear()</code>	Host can control tag list of Swing-U

6.6.1. Connection control

✓ GetPort

Function	<code>string GetPort()</code>	
Return	<code>string</code>	port name for SwingU
Argument	<code>void</code>	None
Operation	Request communication serial port name for Swing-U	

✓ Connecte

Function	<code>bool Connect()</code>	
Return	<code>bool</code>	<code>true</code> when success, <code>false</code> when fail
Argument	<code>void</code>	None
Operation	Request communication start to Swing-U with port from GetPort()	

✓ Disconnect

Function	<code>bool Disconnect()</code>	
Return	<code>bool</code>	<code>true</code> when success, <code>false</code> when fail
Argument	<code>void</code>	None
Operation	Request communication stop to Swing-U	

6.6.2. Hardware parameter control

✓ Remaining Battery Level

Function	<code>void ReportBatterStatus()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the information of battery level. After receiving NotifyParameterChanged event, application cant get the infromation.	

Function	<code>int GetBatteryRate()</code>	
Return	<code>int</code>	Percent of remaining battery
Argument	<code>void</code>	None

Function	<code>double GetBatteryVolt()</code>	
Return	<code>double</code>	Volt of remaining battery
Argument	<code>void</code>	None

✓ Battery Charging Status

Function	<code>void ReportChargeMode()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the status of charging. After receiving NotifyParameterChanged event, application cant get the infromation	

Function	<code>int GetChargeMode()</code>		
Return	<code>int</code>	0	Battery error
		1	On charging
		2	Full charged
		3	On using
Argument	<code>void</code>	None	

✓ Buzzer Volume

Function	<code>void ReportBuzzerVolume()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the volume of buzzer. After receiving NotifyParameterChanged event, application cant get the information	

Function	<code>int GetBuzzerVolume()</code>		
Return	<code>int</code>	0	No sound, vibration only
		1	No sound, no vibration
		2	Minimum volume
		3	Normal volume
		4	Maximum volume
Argument	<code>void</code>	None	

Function	<code>void SetBuzzerVolume(int volume)</code>	
Return	<code>void</code>	None
Argument	<code>int</code>	Default value is MAX

✓ Continuous Mode

Function	<code>void ReportContinuous()</code>		
Return	<code>void</code>	None	
Argument	<code>void</code>	None	
Operation	Request to Swing-U report the mode of continuous RFID read. After receiving NotifyParameterChanged event, application can't get the information		

Function	<code>int GetContinuous()</code>		
Return	<code>int</code>	0	Read single tag
		1	Read tag continuously
Argument	<code>void</code>	None	

Function	<code>void SetContinuous(int mode)</code>		
Return	<code>void</code>	None	
Argument	<code>int</code>	Default value is 1(CONTINUOUS)	

✓ Find Step Unit

Function	<code>void ReportFindStepUnit()</code>		
Return	<code>void</code>	None	
Argument	<code>void</code>	None	
Operation	Request to Swing-U report the RF output decrement unit[dBm] for find mode After receiving NotifyParameterChanged event, application can't get the information		

Function	<code>int GetFindStepUnit()</code>		
Return	<code>int</code>	Step unit of RF power decresement in dBm	
Argument	<code>void</code>	None	

Function	<code>void SetFindStepUnit (int step)</code>		
Return	<code>void</code>	None	
Argument	<code>int step</code>	Default value: 3 [dBm]	

✓ Find Target ID

Function	<code>void SetFindTargetUID(string UID)</code>	
Return	<code>void</code>	None
Argument	<code>string UID</code>	UID for finding target, UID means PC+EPC ID
Operation	Set the finding target UID to Swing-U, works in InventoryMode 1, FIND mode	

✓ Tag Find Threshold

Function	<code>void ReportFindThreshold()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request to Swing-U report the RF output threshold[dBm] level for find mode After receiving NotifyParameterChanged event, application can get the information.	

Function	<code>int GetFindThreshold()</code>	
Return	<code>int</code>	Minimum RF power of target tag finding
Argument	<code>void</code>	None

Function	<code>void SetFindThreshold(int dBm)</code>	
Return	<code>void</code>	None
Argument	<code>int dBm</code>	Minimum RF power of target tag finding in dBm

✓ Inventory Mode

Function	<code>void ReportInventoryMode()</code>		
Return	<code>void</code>	None	
Argument	<code>void</code>	None	
Operation	Request to Swing-U report the mode of inventory. After receiving NotifyParameterChanged event, application can't get the information.		

Function	<code>int GetInventoryMode()</code>		
Return	<code>int</code>	0	Normal inventory
		1	Find target tag
		2	Read user memory with EPC
Argument	<code>void</code>	None	

Function	<code>void SetInventoryMode(int mode)</code>		
Return	<code>void</code>	None	
Argument	<code>int</code>	Default value is 0, normal inventory	

✓ Reader RF power

Function	<code>void ReportRFPower()</code>		
Return	<code>void</code>	None	
Argument	<code>void</code>	None	
Operation	Request to Swing-U report the RF output power of RFID reader After receiving NotifyParameterChanged event, application can't get the information.		

Function	<code>int GetRFPower()</code>		
Return	<code>int</code>	RF attenuation of reader	
Argument	<code>void</code>	None	

Function	<code>void SetRFPower(int attenuation)</code>		
Return	<code>void</code>	None	
Argument	<code>int</code>	RF attenuation of reader	

✓ Tag Count

Function	<code>void ReportTagCount()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request tag count to Swing-U. After receiving NotifyParameterChanged event, application cant get the information.	

Function	<code>int GetTagCount()</code>	
Return	<code>int</code>	Tag count int Swing-U
Argument	<code>void</code>	None

✓ Tag List

Function	<code>void ReportTagList()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request report all tag in memory to Swing-U. with NotifyInventory event, application cant get the information.	

✓ Tag report mode

Function	<code>void ReportTagReportMode()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request tag reporting mode to Swing-U. After receiving NotifyParameterChanged event, application cant get the information.	

Function	<code>int GetTagReportMode()</code>		
Return	<code>int</code>	0	Only new tag can be reported.
		1	Every tag can be reported.
Argument	<code>void</code>	None	

Function	<code>void SetTagReportMode(int mode)</code>		
Return	<code>void</code>	None	
Argument	<code>int</code>	Default value is 0, trigger mode	

✓ Version

Function	<code>void ReportVersion()</code>		
Return	<code>void</code>	None	
Argument	<code>void</code>	None	
Operation	Request H/W, F/W version to Swing-U. After receiving NotifyParameterChanged event, application cant get the information.		

Function	<code>string GetVersionHW()</code>		
Return	<code>string</code>	H/W version	
Argument	<code>void</code>	None	
Operation	Get version of Swing-U H/W		

Function	<code>string GetVersionFW()</code>		
Return	<code>string</code>	F/W version	
Argument	<code>void</code>	None	
Operation	Get version of Swing-U F/W		

6.6.3. UHF RFID control

✓ InventoryStart

Function	<code>void InventoryStart()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request start of UHF RFID inventory to Swing-U. Library generate NotifyButtonEvent.	

✓ InventoryStop

Function	<code>void InventoryStop()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request stop of UHF RFID inventory to Swing-U. Library generate NotifyButtonEvent.	

✓ MemoryRead

Function	<code>void MemoryRead(int bank, int offset, int count)</code>	
Return	<code>void</code>	None
Argument	<code>int</code> bank	0:RESERVED, 1:EPC, 2:TID, 3:USER
	<code>int</code> offset	Start address of memory in word unit
	<code>int</code> count	Address length of memory in word unit
Operation	Request reading memory data of tag. Swing-U generate NotifyError event after operation. Read data has format of <code>string</code> : "UID" + "M" + "DATA"	

✓ MemoryWrite

Function	<code>void MemoryWrite(int bank, int offset, int count, string data)</code>	
Return	<code>void</code>	None
Argument	<code>int</code> bank	0:RESERVED, 1:EPC, 2:TID, 3:USER
	<code>int</code> offset	Start address of memory in word unit
	<code>int</code> count	Address length of memory in word unit
	<code>string</code> data	Data for write
Operation	Request writing data to memory of tag. Swing-U generate NotifyError event after operation.	

6.6.4. Tag List control

✓ TagListAdd

Function	<code>void TagListAdd(string UID)</code>	
Return	<code>void</code>	None
Argument	<code>string</code> <code>UID</code>	EPC ID(PC including) for add to Swing-U
Operation	<p>Requset add specific tag UID to list. Swing-U generate NotifyError event after operation.</p> <p>"No Error": Success "Operation failed": Fail</p>	

✓ TagListClear

Function	<code>void TagListClear()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Requset tag list clear to Swing-U	

7. Android Library API

7.1. Overview

This document is library reference of Swing-U(NTRM-U-2) for Android library.

7.2. Revision History

Version	Date	Description
1.0.0	2015-03-31	First Release

7.3. Swing-U Library contents

Swing-U Library contains as follows:

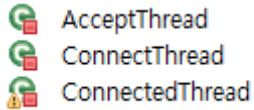
- ✓ SwingAPI.java
 - : Dynamic Link Library for functions and thread to control Swing-U, UHF RFID reader
- ✓ ItemManager
 - : Demo application using SwingAPI.java

※ Notice

1. This API is developed on Android platform.
2. Sample Project is also developed on Eclipse, using JAVA(Android).

7.4. Library Thread

API supports three threads for Swing-U



7.4.1. AcceptThread

Name	AcceptThread
Argument	<pre>public static final int STATE_NONE = 0; // we're doing nothing public static final int STATE_LISTEN = 1; // now listening for incoming connections public static final int STATE_CONNECTING = 2; // now initiating an outgoing connection public static final int STATE_CONNECTED = 3; // now connected to a remote device</pre>
Operation	This thread runs while listening for incoming connections. It behaves like a server-side client. It runs until a connection is accepted
Example	<pre>// If a connection was accepted if (socket != null) { synchronized (SwingAPI.this) { switch (mState) { case STATE_LISTEN: case STATE_CONNECTING: break; case STATE_NONE: case STATE_CONNECTED: break; } } }</pre>

7.4.2. ConnectThread

Name	ConnectThread
Argument	private final BluetoothSocket mmSocket; private final BluetoothDevice mmDevice;
Operation	This thread runs while attempting to make an outgoing connection with a device. It runs straight through; the connection either succeeds or fails.
Example	<pre>try { // Connect the device through the socket. This will block until it succeeds or throws an // exception mmSocket.connect(); } catch (IOException e) { Log.e(TAG, "connect socket failed", e); // Close the socket try { mmSocket.close(); } catch (IOException e2) { Log.e(TAG, "unable to close() socket during connection failure", e2); } connectionFailed(); return; } // Reset the ConnectThread because we're done synchronized (SwingAPI.this) { mConnectThread = null; } // Do work to manage the connection (in a separate thread) manageConnectedSocket(mmSocket, mmDevice);</pre>

7.4.3. ConnectedThread

Name	ConnectedThread
Argument	private final BluetoothSocket mmSocket; private final InputStream mmInStream; private final OutputStream mmOutStream;
Operation	This thread runs during a connection with a remote device. It handles all incoming and outgoing transmissions.
Example	<pre>// Keep listening to the InputStream while connected while (!stopThread) { try { // Read from the InputStream bytes = mmInStream.available(); if(bytes == 0) { sleep(1); continue; } offset += mmInStream.read(buffer, offset, 1); for(int i = 0; i < offset; i++) { if(buffer[i] == mEtx) { pktlength = i; isPkt = true; break; } } if(isPkt) { parse(buffer, pktlength); Arrays.fill(buffer, (byte)0); offset = 0; pktlength = 0; isPkt = false; } sleep(0); } catch (IOException e) { Log.e(TAG, "disconnected", e); connectionLost(); break; } catch (InterruptedException ie) { Log.e("dsm362", ie.getMessage()); } }</pre>

7.5. Functions

7.5.1. Connection control

- ✓ Connect

Function	<code>void connect(BluetoothDevice device)</code>	
Return	<code>void</code>	
Argument	BluetoothDevice device	android.bluetooth.BluetoothDevice
Operation	Request communication start to Swing-U	
Example	<pre>public void ConnectTo(BluetoothDevice scanner) { // TODO Auto-generated method stub if(mSwing != null) { if(mSwing.isConnected() == true) { mSwing.stop(); } mSwing.connect(scanner); } }</pre>	

7.5.2. Hardware parameter control

- ✓ Request all infomation

Function	<code>int swing_getAllInformation ()</code>	
Return	int	-2 when timeout, -1 when response error
Argument	void	None
Operation	<p>Request to Swing-U report the information. After receiving ConnectedThread, application cant get the infromation.</p>	
Example	<pre>... mSwing. swing_getAllInformation(); ...</pre>	

- ✓ Volume control of the device.

Function	int swing_getBuzzerVolume()	
Return	int	-2 when timeout, -1 when response error
		0 Vibration
		1 Mute
		2 Minimum volume
		3 Normal volume
		4 Maximum volume
Argument	void	None
Example	<pre>... mSwing.swing_getBuzzerVolume(); ...</pre>	

Function	void swing_setBuzzerVolume(int level)		
Return	void	None	
Argument	int	0	Vibration
		1	Mute
		2	Minimum volume
		3	Normal volume
		4	Maximum volume
Example	<pre>... mSwing. swing_setBuzzerVolume(4); ...</pre>		

- ✓ Continuous mode.

Function	<code>int swing_getContinuous()</code>		
Return	<code>int</code>	<code>-2</code> when timeout, <code>-1</code> when response error	
		<code>0</code>	Single read
		<code>1</code>	Continuous read
Argument	<code>void</code>	None	
Example	<pre>... Swing.swing_getContinuous(); ...</pre>		

Function	<code>void swing_setContinuous(boolean on_continuous)</code>		
Return	<code>void</code>	None	
Argument	<code>boolean</code>	<code>true</code>	Continuous read
		<code>false</code>	Single read
Example	<pre>... Swing.swing_setContinuous(true); ...</pre>		

- ✓ Tag Report mode.

Function	void swing_setAllTagReport (boolean on_all_tag_report)		
Return	void	None	
Argument	boolean	true	Always, Swing-U report every tag reading
		false	Trigger, Swing-U report new tag only
Example	<pre>... Swing. swing_setAllTagReport (true); ...</pre>		

✓ Menu enable

Function	<code>int swing_getMenuEnable()</code>		
Return	<code>int</code>	<code>-2</code> when timeout, <code>-1</code> when response error	
		<code>0</code>	Enable
		<code>1</code>	Disable
Argument	<code>void</code>	None	
Example	<pre>... Swing.swing_getMenuEnable(); ...</pre>		

Function	<code>void swing_setMenuEnable (boolean enabled)</code>		
Return	<code>void</code>		None
Argument	<code>boolean</code>	<code>true</code>	Enable
		<code>false</code>	Disable
Example	<pre>... Swing.swing_setMenuEnable(true); ...</pre>		

✓ RF Power

Function	<code>void swing_setPower(int attenuation)</code>		
Return	<code>void</code>		None
Argument	<code>int</code>	RF attenuation of reader	
Example	<pre>... Swing.swing_setPower(3) // set RF power to 27dbm ...</pre>		

✓ Find Step Unit

Function	<code>void swing_setThreshold(int th)</code>		
Return	<code>void</code>		None
Argument	<code>int</code>	<code>3</code>	Threshold is 3 dBm
		<code>...</code>	<code>...</code>
		<code>29</code>	Threshold is 29 dBm
Example	<pre>... Swing.swing_setThreshold(3); ...</pre>		

Function	void swing_setUnit(int step)	
Return	void	None
Argument	int step	Default value: 3 [dBm]
Example	<p>...</p> <pre>Swing.swing_setThreshold (6); // 6dBm step</pre> <p>...</p>	

✓ Max EPC length set

Function	void swing_EpcBitLengthSet(int unit)	
Return	void	None
Argument	int	Max epc length(bit)
Example	<p>...</p> <pre>Swing. swing_EpcBitLengthSet (256)</pre> <p>...</p>	

✓ Same tag filtering size set

Function	void swing_MaxCountSet(int unit) // use only when host connected via BT // if host is not connected then the filtering size dependant with Max EPC length.	
Return	void	None
Argument	int	Same tag filtering size
Example	<p>...</p> <pre>Swing. swing_EpcBitLengthSet (1024)</pre> <p>...</p>	

✓ Send Count

Function	void swing_sendcount(int count)	
Return	void	None
Argument	long	Now tag count
Example	<p>...</p> <pre>Swing. swing_sendcount (1)</pre> <p>...</p>	

- ✓ Find Target ID adding

Function	<code>void swing_set_add_search_target(int idx, String id)</code>	
Return	<code>void</code>	None
Argument	<code>int idx</code>	index of finding target table
	<code>String id</code>	UID for finding target, UID means PC+EPC ID
Operation	Set the finding target UID to Swing-U, works in InventoryMode . FIND mode	
Example	<pre>... String target_tag_id = "3000111222333444455556666"; Swing.swing_set_add_search_target(1, target_tag_id); ...</pre>	

Function	<code>void swing_clear_search_taget_list()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	clear finding target UID lists	
Example	<pre>... Swing.swing_clear_search_taget_list(); ...</pre>	

✓ Inventory Mode

Function	<code>void swing_set_inventory_mode (InventoryMode mode)</code>	
Return	<code>void</code>	None
Argument	<code>InventoryMode</code>	<code>enum InventoryMode { INVENTORY_NORMAL, INVENTORY_WITHUSER, SEARCH_SINGLE, SEARCH_MULTI, SEARCH_WILDCARD }</code>
Example	<pre>... Swing.swing_set_inventory_mode (InventoryMode.INVENTORY_NORMAL); Swing.swing_readStart (); ...</pre>	

✓ Language

Function	<code>void setLanguage (int language)</code>	
Return	<code>void</code>	None
Argument	<code>int</code>	0: English 1: Chinese 2: Japanese
Operation	Set Language of Swing-U	
Example	<pre>... int Language = 1//Chinese Swing.setLanguage (Language); ...</pre>	

Function	<code>void getLanguage()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Get Language of Swing-U	
Example	<pre>... Swing.getLanguage(); ...</pre>	

✓ Session

Function	<code>void setSession (int session)</code>	
Return	<code>void</code>	None
Argument	<code>int</code>	0: Session 0 1: Session 1 2: Session 2
Operation	Set Session of Swing-U	
Example	<pre>... int Session = 1//Session 1 Swing.setSession (Session); ...</pre>	

Function	<code>void ReportSession()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Report Session of Swing-U	
Example	<pre>... Swing. ReportSession(); ...</pre>	

7.6. UHF RFID control

7.6.1. InventoryStart

Function	<code>void swing_readStart()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request start of UHF RFID inventory to Swing-U.	
Example	<pre>... Swing.swing_readStart(); ...</pre>	

7.6.2. InventoryStop

Function	<code>void swing_readStop ()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Request stop of UHF RFID inventory to Swing-U.	
Example	<pre>... Swing.swing_readStop (); ...</pre>	

7.6.3. InventoryClear

Function	<code>void swing_clear_inventory ()</code>	
Return	<code>void</code>	None
Argument	<code>void</code>	None
Operation	Requset tag list clear to Swing-U.	
Example	<pre>... Swing. swing_clear_inventory(); ...</pre>	

7.6.4. MemoryRead

Function	<code>void swing_readMemory(int type, int offset, int wordLength)</code>	
Return	<code>void</code>	None
Argument	<code>int type</code> <code>int offset</code> <code>int wordLength</code>	0 : RESERVED, 1 : EPC, 2 : TID, 3 : USER Start address of memory in word unit Address length of memory in word unit
Operation	Request reading memory data of tag. Swing-U generate ConnectedThread after operation. Read data has format of String: "UID" + "M" + "DATA"	
Example	<pre>... int type = 3; //USER int offset = 2; int wordLength = 6; Swing.swing_readMemory(type, offset, wordLength); ...</pre>	

7.6.5. MemoryReadsync

Function	<code>boolean swing_readMemorysync(int type, int offset, int wordLength)</code>		
Return	<code>boolean</code>	<code>true</code>	Read success
		<code>false</code>	Not Read
Argument	<code>int type</code> <code>int offset</code> <code>int wordLength</code>		
Operation	Request reading memory data of tag. Swing-U generate ConnectedThread after operation. Read data has format of String: "UID" + "M" + "DATA"		
Example	<pre>... int type = 3; //USER int offset = 2; int wordLength = 6; Swing.swing_readMemorysync(type, offset, wordLength); ...</pre>		

7.6.6. MemoryWrite

Function	void swing_writeMemory (int bank, int offset, int count, String data)		
Return	void	None	
Argument	int bank int offset int count String data	0 : RESERVED, 1 : EPC, 2 : TID, 3 : USER Start address of memory in word unit Address length of memory in word unit Data for write	
Operation	Request writing data to memory of tag.		
Example	<pre>... int bank = 1; //epc int offset = Integer.parseInt(ed_offset.getText().toString()); int count = Integer.parseInt(ed_length.getText().toString()); String data = ed_data.getText().toString(); Swing.swing_writeMemory(bank, offset, count, data); ...</pre>		

7.6.7. MemoryWritsync

Function	boolean swing_writeMemorysync (int bank, int offset, int count, String data)		
Return	boolean	true	Write success
		false	Not Write
Argument	int bank int offset int count String data	0 : RESERVED, 1 : EPC, 2 : TID, 3 : USER Start address of memory in word unit Address length of memory in word unit Data for write	
Operation	Request writing data to memory of tag.		
Example	<pre>... int bank = 1; //epc int offset = Integer.parseInt(ed_offset.getText().toString()); int count = Integer.parseInt(ed_length.getText().toString()); String data = ed_data.getText().toString(); Swing.swing_writeMemorysync(bank, offset, count, data); ...</pre>		

7.6.8. MemorySelect

Function	boolean swing_SelectMemory (int bank, int offset, int count, String data)		
Return	void	None	
Argument	int bank int offset int count String data	0 : RESERVED, 1 : EPC, 2 : TID, 3 : USER Start address of memory in word unit Address length of memory in word unit Data for Select	
Operation	Request selecting data to memory of tag.		
Example	<pre>... int bank = 1; //epc int offset = Integer.parseInt(ed_offset.getText().toString()); int count = Integer.parseInt(ed_length.getText().toString()); String data = ed_data.getText().toString(); Swing.swing_SelectMemory(bank, offset, count, data); ...</pre>		

7.6.9. MemorySelectSync

Function	boolean swing_SelectMemorySync (int bank, int offset, int count, String data)		
Return	boolean	true	Select success
		false	Not Select
Argument	int bank int offset int count String data	0 : RESERVED, 1 : EPC, 2 : TID, 3 : USER Start address of memory in word unit Address length of memory in word unit Data for Select	
Operation	Request selecting data to memory of tag.		
Example	<pre>... int bank = 1; //epc int offset = Integer.parseInt(ed_offset.getText().toString()); int count = Integer.parseInt(ed_length.getText().toString()); String data = ed_data.getText().toString(); Swing.swing_SelectMemorySync (bank, offset, count, data); ...</pre>		

7.6.10. MemoryLock

Function	<code>void swing_LockMemory (int kill, int access, int epc, int tid, int user, String access_pwd)</code>				
Return	<code>void</code>	None			
Argument	<code>int kill</code>	Lock action for Kill password memory	0	Accessible	
			1	Always accessible	
			2	Secured accessible	
			3	Always not accessible	
			4	No change	
	<code>int access</code>	Lock action for Access password memory	0	Accessible	
			1	Always accessible	
			2	Secured accessible	
			3	Always not accessible	
			4	No change	
	<code>int epc</code>	Lock action for EPC memory	0	Accessible	
			1	Always accessible	
			2	Secured accessible	
			3	Always not accessible	
			4	No change	
	<code>int tid</code>	Lock action for TID memory	0	Accessible	
			1	Always accessible	
			2	Secured accessible	
			3	Always not accessible	
			4	No change	
	<code>int user</code>	Lock action for USER memory	0	Accessible	
			1	Always accessible	
			2	Secured accessible	
			3	Always not accessible	
			4	No change	
	<code>String access_pwd</code>	Access password	Tag Access Password		

Operation	Request locked to memory of tag.
Example	<pre>... int killpwd = 4; //there are no change on kill pwd memory int accpwd = 4; //there are no change on access pwd memory int epc = 2; //set the EPC memory secured accessible int tid = 4; //there are no change on TID memory int user = 4; //there are no change on User memory String access_pwd = "00000000"; //password of tag Swing.swing_writeMemorysync(killpwd, accpwd, epc, tid, user, access_pwd) ...</pre>