

	DOTR-900	Date	2013-06-29
		Rev	1.0
		Page	1


# DOTR-900

안드로이드

매뉴얼


D.O.Tel




	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	3

## 목차


1. 소개 .....	7
2. 개발도구 .....	7
3. 제공 패키지 .....	7
4. 안드로이드 R900 함수 정의 및 예제 .....	7
4.1 BLUETOOTHACTIVITY.JAVA .....	8
4.1.1 SETONBTEVENTLISTENER .....	8
4.1.2 SCANBLUETOOTHDEVICE .....	9
4.1.3 BYEBLUETOOTHDEVICE .....	9
4.1.4 CONNECTTOBLUETOOTHDEVICE .....	10
4.1.5 SENDCMDOPENINTERFACE1 .....	11
4.1.6 SENDCMDINVENTORY .....	12
4.1.7 SENDCMDINVENTORY .....	13
4.1.8 SENDCMDSTOP .....	14
4.1.9 SENDHEARTBEAT .....	15
4.1.10 SENDCMDSELECTMASK .....	16
4.1.11 SENDSETSESSION .....	18
4.1.12 SENDSETQVALUE .....	19
4.1.13 SENDSETINVENTORYTARGET .....	20
4.1.14 SENDINVENTPARAM .....	21
4.1.15 SENDSETSELECTACTION .....	22
4.1.16 SETOPMODE .....	23

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	4

4.1.17 SENDREADTAG .....	24
4.1.18 SENDWRITETAG.....	26
4.1.19 CONVERTLOCKINDEX .....	27
4.1.20 SENDLOCKTAG.....	28
4.1.21 SENDLOCKTAG.....	29
4.1.22 SENDKILLTAG .....	31
4.1.23 SENDGETVERSION.....	32
4.1.24 SENDSETDEFAULTPARAMETER.....	33
4.1.25 SENDGETTINGPARAMETER .....	34
4.1.26 SENDSETTINGTXPOWER .....	35
4.1.27 SENDGETMAXPOWER.....	36
4.1.28 SENDSETTINGTXCYCLE.....	37
4.1.29 SENDCHANGECHANNELSTATE .....	38
4.1.30 SENDSETTINGCOUNTRY .....	39
4.1.31 SENDGETTINGCOUNTRY .....	40
4.1.32 SENDSETLOCKTAGMEMSTATEPERM .....	40
4.1.33 SENDPAUSETX .....	42
4.1.34 SENDSTATUSREPORTING .....	43
4.1.35 SENDINVENTORYREPORTINGFORMAT .....	44
4.1.36 SENDDISLINK .....	45
4.1.37 SENDUPLOADINGTAGDATA.....	45
4.1.38 SENDCLEARINGTAGDATA.....	46
4.1.39 SENDALERTREADERSTATUS .....	47
4.1.40 SENDGETTINGSTATUSWORD.....	49
4.1.41 SENDSETTINGBUZZERVOLUME .....	50
4.1.42 SENDBEEP .....	51

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	5

4.1.43 SENDSETTINGAUTOPOWEROFFDELAY.....	52
4.1.44 SENDGETTINGBATTERYLEVEL .....	53
4.1.45 SENDREPORTINGBATTERYSTATE .....	54
4.1.46 SENDTURNINGREADEROFF .....	55
<b>4.2 ONBTEVENTLISTENER.JAVA.....</b>	<b>55</b>
4.2.1 ONBTFOUNDNEWDEVICE .....	56
4.2.2 ONBTSCANCOMPLETED .....	56
4.2.3 ONBTCONNECTED.....	56
4.2.4 ONBTDISCONNECTED.....	57
4.2.5 ONBTCONNECTFAIL.....	57
4.2.6 ONBTDATASENT .....	58
4.2.7 ONBTDATATRANSEXCEPTION .....	58
4.2.8 ONNOTIFYBTDATAARECV .....	59
<b>4.3 R900MANAGER.JAVA.....</b>	<b>59</b>
4.3.1 ISBLUETOOTHENABLED.....	59
4.3.2 ENABLEBLUETOOTH .....	60
4.3.3 QUERYPAIREDDEVICES .....	61
4.3.4 STARTDISCOVERY .....	62
4.3.5 STOPDISCOVERY .....	63
4.3.6 GETBLUETOOTHDEVICE.....	63
4.3.7 SENDDATA.....	64
4.3.8 ISTRYINGCONNECT.....	65
<b>4.4 R900PROTOCOL.JAVA.....</b>	<b>66</b>
4.4.1 MAKEPROTOCOL .....	66

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	6

4.4.2 MAKEPROTOCOL .....	67
4.4.3 MAKEPROTOCOL .....	68
4.4.4 MAKEPROTOCOL .....	69
4.4.6 MAKEPROTOCOL .....	71
4.4.7 MAKEPROTOCOL .....	73
4.4.8 STRING2BYTES .....	74
4.5 R900RECVPACKETPARSER.JAVA .....	75
4.5.1 PUSHPACKET .....	75
4.5.2 POPPACKET .....	76
4.5.3 POPPACKET .....	77
5. DEMO 설치방법 .....	79
6. DEMO 사용방법 .....	80

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	7

## 1. 소개

이 문서는 DOTR-900의 API 함수 사용설명 및 사용방법에 대해서 설명한 문서입니다. UHF 900MHz 리더기이며, 다양한 모바일 디바이스와 블루투스 연결을 하여 사용을 합니다. 본 문서에서는 안드로이드와 R900을 연동하여 사용하는 방법을 다루었습니다. 주 사용되는 API 함수설명과 예제를 포함하여 사용자가 보다 쉽게 접근하도록 하였습니다. 또한 Demo 프로그램의 사용방법을 포함하고 있습니다.

## 2. 개발도구

### 개발 플랫폼

DOTR-900개발을 위해서 자바 플랫폼을 추가 합니다. 하기 다운로드를 통해서 인스톨 진행을 합니다.

- Developer.android.com 사이트에서 "android\_sdk\_r09\_windows.zip"을 다운로드 합니다.
- 이클립스 : Indigo version  
[이클립스 다운로드](#)
- 자바 : jdk1.6.0\_23  
[자바 다운로드](#) (JDK 자바 다운로드 버튼을 눌러서 다운로드 하여 인스톨합니다.)
- ADT12.0.0  
Developer.android.com 사이트에서 SDK를 누릅니다. ADT 12.0.0을 찾아서 다운로드 합니다.  
[ADT 다운로드](#)
- Android SDK and AVD Manager  
"SDK Platform Android 2.2, API 8, revision 2" 만약 인스톨이 되지 않으면, "SDK Platform Android 2.2, revision 2" 를 사용합니다.

## 3. 제공 패키지

프로그램 개발 시에 제공되는 패키지는 하기 테이블과 같이 제공됩니다.

Device	제공 문서
R900	R900 매뉴얼

## 4. 안드로이드 R900 함수 정의 및 예제

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	8

## 4.1 BluetoothActivity.java

RFID가 활성화 될 경우, 블루투스 기반으로 동작을 합니다.

### 4.1.1 setOnBtEventListener

Application로부터 이벤트를 등록하기 위한 함수 입니다.

---

**public void setOnBtEventListener( OnBtEventListener listener )**

---

#### Parameters

OnBtEventListener listener : 현재 생성된 객체입니다.

#### Return Values

None

#### Remarks

블루투스 장치 검색 및 추가 및 데이터 송/수신을 하기 위해 제공되는 함수입니다.

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
private OnBtEventListener mBtEventListener;
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

mBtEventListener = listener;

public void setOnBtEventListener( OnBtEventListener listener )
{
    mBtEventListener = listener;
    if( mR900Manager != null )
        mR900Manager.setOnBtEventListener(listener);
}
```

---



	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	9

#### 4.1.2 scanBluetoothDevice

ScanBluetoothDevice는 주변에 Bluetooth 장비를 검색하기 위한 함수입니다.

**public void scanBluetoothDevice()**

##### Parameters

None

##### Return Values

None

##### Remarks

함수를 사용하기 전에 SetOnBtEventListener() 함수를 사용하여 이벤트를 등록하고 사용합니다.

##### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void scanBluetoothDevice()
{
    if( mR900Manager != null )
        mR900Manager.startDiscovery();
}
```

#### 4.1.3 byeBluetoothDevice

R900 장비로부터 연결을 해제하기 위한 함수입니다.

**public void byeBluetoothDevice()**

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	10

#### Parameters

None

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void byeBluetoothDevice()
{
    if( mR900Manager != null )
        mR900Manager.sendData(R900Protocol.BYE);
}
```

#### 4.1.4 connectToBluetoothDevice

해당 블루투스 장비 ID와 주소에 연결하기 위한 함수입니다.

```
public void connectToBluetoothDevice( String address, UUID uuid )
```

#### Parameters

address

연결하기 위한 블루투스 장비 Mac 주소입니다.

uuid

연결하기 위한 블루투스 장비 ID입니다.

#### Return Values

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	11

None

#### Remarks

처음 연결을 시도한 경우에는 페어링을 진행합니다.

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

public void connectToBluetoothDevice( String address, UUID uuid )
{
    if( mR900Manager != null && address != null )
        mR900Manager.connectToBluetoothDevice(address, uuid);
}
```

#### 4.1.5 sendCmdOpenInterface1

R900을 Open하기 위한 함수입니다.

```
public void sendCmdOpenInterface1()
```

#### Parameters

None

#### Return Values

None

#### Remarks

위 함수는 connectToBluetoothDevice()을 통해 연결된 상태에서 사용해야 합니다.

#### 사용예제

##### JAVA

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	12

## BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

public void sendCmdOpenInterface1()
{
    if( mR900Manager != null )
        mR900Manager.sendData(R900Protocol.OPEN_INTERFACE_1);
}
```

### 4.1.6 sendCmdInventory

Tag를 Inventory 하기 위한 함수입니다.

```
public void sendCmdInventory()
```

#### Parameters

None

#### Return Values

None

#### Remarks

4.1.7 참조

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected boolean mSingleTag;
protected boolean mUseMask;
protected int mTimeout;
protected boolean mQuerySelected;
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	13

protected String mLastCmd;

import com.dotel.rfid.R900Status;

protected R900Manager mR900Manager;  
mR900Manager = new R900Manager( mHandler );

```
public void sendCmdInventory()
{
    sendCmdInventory( mSingleTag ? 1 : 0, mUseMask ? ( mQuerySelected ? 3 : 2 ) : 0, mTimeout );
}
```

#### 4.1.7 sendCmdInventory

Tag를 Inventory 하기 위한 함수입니다.

**public void sendCmdInventory( int f\_s, int f\_m, int to )**

##### Parameters

f\_s

연속 또는 한 개의 태그 판독 여부를 설정합니다.

0 : 연속적으로 태그를 판독합니다.

1: 한 개의 태그를 판독합니다.

f\_m

select mask 여부를 설정합니다.

0 또는 1 : select mask를 적용하지 않습니다.

2 : select mask 설정이 되지 않는 태그만 판독합니다.

3 : select mask 설정된 태그만 판독합니다.

to

Inventory 동작 시간을 설정합니다. 단위는 msec이며, 0인 경우 연속 동작을 합니다.

##### Return Values

None

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	14

#### Remarks

해당 태그만 판독하기 위해 sendCmdSelectMask() 통해 태그를 설정할 수 있습니다.

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected boolean mSingleTag;
protected boolean mUseMask;
protected int mTimeout;
protected boolean mQuerySelected;

protected String mLastCmd;

import com.dotel.rfid.R900Status;

protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

public void sendCmdInventory( int f_s, int f_m, int to )
{
    if( mR900Manager != null )
    {
        R900Status.setOperationMode(1);
        mLastCmd = R900Protocol.CMD_INVENT;
        mR900Manager.sendData(R900Protocol.makeProtocol(mLastCmd, new int[]{ f_s, f_m, to }));
    }
}
```

#### 4.1.8 sendCmdStop

진행중인 동작을 중단하기 위한 함수입니다.

```
public void sendCmdStop()
```

#### Parameters

**CONFIDENTIAL** Document D.O.Tel. All rights of reproduction and disclosure reserved.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	15

None

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
import com.dotel.rfid.R900Status;
```

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void sendCmdStop()
{
    if( mR900Manager != null )
    {
        R900Status.setOperationMode(0);
        mLastCmd = R900Protocol.CMD_STOP;
        mR900Manager.sendData(R900Protocol.makeProtocol(mLastCmd, (int[])null));
    }
}
```

#### 4.1.9 sendHeartBeat

호스트와 연결 상태 체크를 설정하는 함수입니다.

```
public void sendHeartBeat( int value )
```

#### Parameters

value

연결 상태를 체크하기 위한 시간을 설정합니다. 단위는 msec이며, 0인 경우 연결 상태 체크를 하지 않

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	16

습니다.

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void sendHeartBeat( int value )
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol(R900Protocol.CMD_HEART_BEAT,
            new int[]{ value }));
    }
}
```

#### 4.1.10 sendCmdSelectMask

선택한 태그를 판독하기 위해 설정하는 함수 입니다.

```
public void sendCmdSelectMask( int n, int bits, int mem, int b_offset, String pattern, int target, int action )
```

#### Parameters

n

마스크 테이블 인덱스를 설정합니다. 0에서 7까지 설정이 가능합니다.

bits



	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	17

마스크 패턴의 비트수를 설정합니다. 0인 경우 마스크를 해제합니다.

mem

태그의 메모리 영역을 설정합니다.

0 : Reserved 영역

1 : EPC 영역

2 : TID 영역

3 : User 영역

b\_offset

마스크 패턴을 적용할 태그의 메모리 시작 비트를 설정합니다.

EPC인 경우 메모리 시작 비트는 16입니다.

pattern

적용할 마스크 패턴의 Hexa String을 설정합니다.

target

Acion에 반영될 Flag를 설정합니다. 해당 태그를 판독하려면 4로 설정합니다.

action

선택된 태그의 flag 변경을 설정합니다. Flag의 의미는 Appendix I 을 참조합니다.

## Return Values

None

## Remarks

None

## 사용예제

### JAVA

#### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendCmdSelectMask( int n, int bits, int mem, int b_offset, String pattern, int target, int action )
```

```
{
```

```
    if( mR900Manager != null )
```

```
{
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	18

```

mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_SEL_MASK,
new int[]{ n, bits, mem, b_offset}, pattern, new int[]{ target, action } ));
}
}

```

#### 4.1.11 sendSetSession

Inventory에 적용할 Session을 설정하기 위한 함수입니다.

```
public void sendSetSession( int session )
```

##### Parameters

session

태그 query시 적용할 session을 지정합니다. session값은 0에서 3까지 설정할 수 있습니다. 기본 설정값은 0입니다.

- 0 : 판독한 태그 flag는 2초 이내로 판독 전 flag로 변경됩니다. (Session 0)
- 1 : 판독한 태그 flag는 2~5초 이내로 판독 전 flag로 변경됩니다. (Session 1)
- 2 : 판독한 태그 flag는 5초 이상으로 판독 전 flag로 변경됩니다. (Session 2)
- 3 : 판독한 태그 flag는 판독 전 flag로 변경되지 않습니다. (Session 3)

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```

protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

public void sendSetSession( int session )
{

```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	19

```

if( mR900Manager != null )
{
    mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_INVENT_PARAM,
    new int[]{ session, R900Protocol.SKIP_PARAM, R900Protocol.SKIP_PARAM } ) );
}
}

```

#### 4.1.12 sendSetQValue

Invenory에 적용할 Q값을 설정하기 위한 함수입니다.

```
public void sendSetQValue( int q )
```

##### Parameters

q

태그 query시 적용할 Q 값을 설정합니다. Q값은 0에서 15까지 설정할 수 있습니다. 기본 설정 값은 5 입니다.

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```

protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

```

```

public void sendSetQValue( int q )
{
    if( mR900Manager != null )
    {

```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	20

```

mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_INVENT_PARAM,
new int[]{ R900Protocol.SKIP_PARAM, q, R900Protocol.SKIP_PARAM } ) );
}
}

```

#### 4.1.13 sendSetInventoryTarget

Inventory에 적용할 flag상태를 지정하기 위한 함수입니다.

```
public void sendSetInventoryTarget( int m_ab )
```

##### Parameters

m\_ab

태그 query시 적용할 inventory flag 상태를 지정합니다. Target값은 0에서 2까지 설정할 수 있습니다.  
기본 설정 값은 2입니다.

0 : Inventory flag A

1 : Inventory flag B

2 : Inventory flag A or B

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```

protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

```

```
public void sendSetInventoryTarget( int m_ab )
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	21

```

{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_INVENT_PARAM,
        new int[]{ R900Protocol.SKIP_PARAM, R900Protocol.SKIP_PARAM, m_ab } ) );
    }
}

```

#### 4.1.14 sendInventParam

Inventory에 적용할 parameter를 설정하기 위한 함수입니다.

```
public void sendInventParam( int session, int q, int m_ab )
```

##### Parameters

session

태그 query시 적용할 session을 지정합니다. session값은 0에서 3까지 설정할 수 있습니다. 기본 설정값은 0입니다.

q

태그 query시 적용할 Q값을 지정합니다. Q값은 0에서 15까지 설정할 수 있습니다. 기본 설정 값은 5입니다.

m\_ab

태그 query시 적용할 태그 inventory flag 상태를 지정합니다. Target값은 0에서 2까지 설정할 수 있습니다. 기본 설정 값은 2입니다.

##### Return Values

None

##### Remarks

None

##### 사용예제

**JAVA**

**BluetoothActivity.java**

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	22

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void sendInventParam( int session, int q, int m_ab )
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_INVENT_PARAM,
            new int[]{ session, q, m_ab } ) );
    }
}
```

#### 4.1.15 sendSetSelectAction

Invenory시 사용할 select mask를 설정하기 위한 함수입니다.

```
public void sendSetSelectAction( int bits, int mem, int b_offset, String pattern, int action )
```

##### Parameters

bits

마스크 패턴의 비트수를 설정합니다. 0인경우 마스크를 해제합니다.

mem

태그의 메모리 영역을 설정합니다.

0 : Reserved 영역

1 : EPC 영역

2 : TID 영역

3 : User 영역

b\_offset

마스크 패턴을 적용할 태그의 메모리 시작 비트를 설정합니다.

EPC인 경우 메모리 시작 비트는 16입니다.

pattern

적용할 마스크 패턴의 Hexa String을 설정합니다.

action

선택된 태그의 flag 변경을 설정합니다. Flag의 의미는 Appendix I 을 참조합니다.

##### Return Values

**CONFIDENTIAL** Document D.O.Tel. All rights of reproduction and disclosure reserved.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	23

None

#### Remarks

None

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void sendSetSelectAction( int bits, int mem, int b_offset, String pattern, int action )
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_SEL_MASK,
        new int[]{ 0, bits, mem, b_offset}, pattern, new int[]{ R900Protocol.SKIP_PARAM, action } ));
    }
}
```

#### 4.1.16 setOpMode

Inventory 동작에 적용할 parameter를 기록하기 위한 함수입니다.

```
public void setOpMode( boolean singleTag, boolean useMask, int timeout, boolean querySelected )
```

#### Parameters

singleTag

태그 판독 횟수를 설정합니다.

True ; 연속적으로 태그를 판독합니다.

False : 한 개 태그만 판독합니다.

useMask

Select mask 선택 여부를 설정합니다.

True : 선택한 태그만 판독합니다.

False : 랜덤하게 태그를 판독합니다.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	24

timeout

태그 판독 시간을 설정합니다. 단위는 msec이며, 0인 경우 연속으로 판독합니다.

querySelected

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected boolean mSingleTag;
protected boolean mUseMask;
protected int mTimeout;
protected boolean mQuerySelected;
```

```
public void setOpMode( boolean singleTag, boolean useMask, int timeout, boolean querySelected )
{
    mSingleTag = singleTag;
    mUseMask = useMask;
    mTimeout = timeout;
    mQuerySelected = querySelected;
}
```

#### 4.1.17 sendReadTag

태그로부터 데이터를 출력하기 위한 함수입니다.

```
public void sendReadTag( int w_count, int mem, int w_offset, String ACS_PWD )
```

#### Parameters

w\_count



	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	25

태그로부터 읽어 들일 데이터 길이를 지정합니다. 단위는 word(16bits) 이며, 기본 설정값은 1 입니다.

mem

태그로부터 읽어 들일 메모리 위치를 지정합니다.

0 : RESERVED 영역

1 : EPC 영역

2 : TID 영역

3 : USER 영역

w\_offset

태그로부터 읽어 들일 메모리 시작 위치를 지정합니다. 단위는 word(16bits) 입니다.

ACS\_PWD

태그에 저장되어 있는 access password를 지정합니다. 기본 설정값은 0 입니다.

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendReadTag( int w_count, int mem, int w_offset, String ACS_PWD )
```

```
{
```

```
    if( mR900Manager != null )
```

```
    {
```

```
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_READ_TAG_MEM,
```

```
        new int[]{ w_count, mem, w_offset }, ACS_PWD,
```

```
        new int[]{ mSingleTag ? 1 : 0, mUseMask ? ( mQuerySelected ? 3 : 2 ) : 0, mTimeout } ) );
```

```
    }
```

```
}
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	26

#### 4.1.18 sendWriteTag

태그에 데이터를 기록하기 위한 함수입니다.

---

**public void sendWriteTag( int w\_count, int mem, int w\_offset, String ACS\_PWD, String wordPattern )**

---

##### Parameters

w\_count

태그에 기록하기 위한 데이터 길이를 지정합니다. 단위는 word(16bits) 입니다.

mem

태그에 기록하기 위한 메모리 위치를 지정합니다.

0 : RESERVED 영역

1 : EPC 영역

2 : TID 영역

3 : USER 영역

w\_offset

태그에 기록하기 위한 메모리 시작위치를 지정합니다. 단위는 word(16bits) 입니다.

ACS\_PWD

태그에 저장되어 있는 access password를 지정합니다.

wordPattern

태그에 기록하기 위한 데이터를 지정합니다. 데이터는 Hexa String으로 지정합니다.

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

protected boolean mSingleTag;

---

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	27

```
protected boolean mUseMask;
protected int mTimeout;
protected boolean mQuerySelected;
```

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void sendWriteTag( int w_count, int mem, int w_offset, String ACS_PWD, String wordPattern )
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_WRITE_TAG_MEM,
        new int[]{ w_count, mem, w_offset }, new String[]{ wordPattern, ACS_PWD },
        new int[]{ mSingleTag ? 1 : 0, mUseMask ? ( mQuerySelected ? 3 : 2 ) : 0, mTimeout } ) );
    }
}
```

#### 4.1.19 convertLockIndex

Lock 인덱스를 변환하기 위한 함수입니다.

```
private int convertLockIndex( boolean enable, boolean index )
```

##### Parameters

enable

Lock 인덱스 활성화 여부를 지정합니다.

0 : 인덱스를 활성화 하지 않습니다.

1 : 인덱스를 활성화 합니다.

index

Lock 인덱스 위치를 지정합니다.

##### Return Values

None

##### Remarks

None

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	28

## 사용예제

### JAVA

#### BluetoothActivity.java

```
private int convertLockIndex( boolean enable, boolean index )
{
    return enable ? ( index ? 1 : 0 ) : -1;
}
```

#### 4.1.20 sendLockTag

태그 특정 메모리에 잠금 기능을 하기 위한 함수입니다.

**public void sendLockTag( LockPattern lockPattern, String ACS\_PWD )**

#### Parameters

lockPattern

해당 메모리 영역에 잠금 기능을 지정하기 위한 함수입니다. 해당 값이 0인 경우 잠금을 해제하고, 해당값이 1인 경우 잠금 기능을 설정합니다.

kill\_pwd : Kill password 메모리의 잠금 상태를 지정합니다.

acs\_pwd : Access password 메모리의 잠금 상태를 지정합니다.

epc : EPC 메모리의 잠금 상태를 지정합니다.

tid : TID 메모리의 잠금 상태를 지정합니다.

user : user 메모리의 잠금 상태를 지정합니다.

ACS\_PWD

태그에 저장되어 있는 access password를 지정합니다.

#### Return Values

None

#### Remarks

None

## 사용예제

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	29

## JAVA

### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

public void sendLockTag( LockPattern lockPattern, String ACS_PWD )
{
    if( mR900Manager != null )
    {
        final int user = convertLockIndex( lockPattern.enableUser,
        lockPattern.indexUser );//( ( lockPattern.indexUser == false ) ? ( R900Protocol.SKIP_PARAM ) :
        ( lockPattern.enableUser ? 1 : 0 ) );
        final int tid = convertLockIndex( lockPattern.enableTid, lockPattern.indexTid );//( ( lockPattern.indexTid
        == false ) ? ( R900Protocol.SKIP_PARAM ) : ( lockPattern.enableTid ? 1 : 0 ) );
        final int epc = convertLockIndex( lockPattern.enableUii, lockPattern.indexUii );//( ( lockPattern.indexUii
        == false ) ? ( R900Protocol.SKIP_PARAM ) : ( lockPattern.enableUii ? 1 : 0 ) );
        final int acs_pwd = convertLockIndex( lockPattern.enableAcsPwd,
        lockPattern.indexAcsPwd );//( ( lockPattern.indexAcsPwd == false ) ? ( R900Protocol.SKIP_PARAM ) :
        ( lockPattern.enableAcsPwd ? 1 : 0 ) );
        final int kill_pwd = convertLockIndex( lockPattern.enableKillPwd,
        lockPattern.indexKillPwd );//( ( lockPattern.indexKillPwd == false ) ? ( R900Protocol.SKIP_PARAM ) :
        ( lockPattern.enableKillPwd ? 1 : 0 ) );

        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_LOCK_TAG_MEM,
        new int[]{ user, tid, epc, acs_pwd, kill_pwd }, ACS_PWD,
        new int[]{ mSingleTag ? 1 : 0, mUseMask ? ( mQuerySelected ? 3 : 2 ) : 0, mTimeout } ) );
    }
}
```

#### 4.1.21 sendLockTag

태그 특정 메모리에 잠금 기능을 하기 위한 함수입니다.

```
public void sendLockTag( int lockMask, int lockEnable, String ACS_PWD )
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	30

#### Parameters

lockMask

태그 특정 메모리에 잠금 기능을 지정하기 위한 마스크 값을 지정합니다.

lockEnable

태그 특정 메모리에 잠금 기능을 지정하기 위한 마스크 필드 값을 지정합니다.

ACS\_PWD

태그에 저장되어 있는 access password를 지정합니다.

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendLockTag( int lockMask, int lockEnable, String ACS_PWD )
```

```
{
```

```
    if( mR900Manager != null )
```

```
    {
```

```
        int bitFlag;
```

```
        boolean mask;
```

```
        boolean enable;
```

```
//---
```

```
        bitFlag = 0x200;//0x02;
```

```
        mask = ( lockMask & bitFlag ) == bitFlag;
```

```
        enable = ( lockEnable & bitFlag ) == bitFlag;
```

```
        final int user = ( ( mask == false ) ? ( R900Protocol.SKIP_PARAM ) : ( enable ? 1 : 0 ) );
```

```
//---
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	31

```

bitFlag = 0x80;//0x08;
mask = ( lockMask & bitFlag ) == bitFlag;
enable = ( lockEnable & bitFlag ) == bitFlag;
final int tid = ( ( mask == false ) ? ( R900Protocol.SKIP_PARAM ) : ( enable ? 1 : 0 ) );
//---
bitFlag = 0x20;
mask = ( lockMask & bitFlag ) == bitFlag;
enable = ( lockEnable & bitFlag ) == bitFlag;
final int epc = ( ( mask == false ) ? ( R900Protocol.SKIP_PARAM ) : ( enable ? 1 : 0 ) );
//---
bitFlag = 0x08;//0x80;
mask = ( lockMask & bitFlag ) == bitFlag;
enable = ( lockEnable & bitFlag ) == bitFlag;
final int acs_pwd = ( ( mask == false ) ? ( R900Protocol.SKIP_PARAM ) : ( enable ? 1 : 0 ) );
//---
bitFlag = 0x02;//0x200;
mask = ( lockMask & bitFlag ) == bitFlag;
enable = ( lockEnable & bitFlag ) == bitFlag;
final int kill_pwd = ( ( mask == false ) ? ( R900Protocol.SKIP_PARAM ) : ( enable ? 1 : 0 ) );
//---
mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_LOCK_TAG_MEM,
new int[]{ user, tid, epc, acs_pwd, kill_pwd }, ACS_PWD,
new int[]{ mSingleTag ? 1 : 0, mUseMask ? ( mQuerySelected ? 3 : 2 ) : 0, mTimeout } ) );
}
}

```

#### 4.1.22 sendKillTag


태그를 사용할 수 없도록 하기 위한 함수입니다.

```
public void sendKillTag( String killPwd )
```

##### Parameters

killPwd

태그에 저장되어 있는 kill password를 지정합니다. 0인 경우 동작하지 않습니다.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	32

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

public void sendKillTag( String killPwd )
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_KILL_TAG,
            killPwd, new int[]{ mSingleTag ? 1 : 0, mUseMask ? ( mQuerySelected ? 3 : 2 ) : 0, mTimeout } ) );
    }
}
```

#### 4.1.23 sendGetVersion

R900 펌웨어 버전을 확인하기 위한 함수입니다.

**public void sendGetVersion()**

#### Parameters

None

#### Return Values

None

#### Remarks

함수를 사용하기 전에 SetOnBtEventListener() 함수를 사용하여 이벤트를 등록하고 사용합니다.



	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	33

## 사용예제

### JAVA

#### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

public void sendGetVersion()
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_GET_VERSION ) );
    }
}
```

#### 4.1.24 sendSetDefaultParameter

R900 리셋 이후 변경된 파라미터를 기본값으로 환원하기 위한 함수입니다.

```
public void sendSetDefaultParameter()
```

#### Parameters

None

#### Return Values

None

#### Remarks

None

## 사용예제

### JAVA

#### BluetoothActivity.java

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	34

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void sendSetDefaultParameter()
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_SET_DEF_PARAM ) );
    }
}
```

#### 4.1.25 sendGettingParameter

Invenory에 적용된 parameter를 확인하기 위한 함수입니다.

```
public void sendGettingParameter( String cmd, String p )
```

##### Parameters

None

##### Return Values

None

##### Remarks

None

##### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void sendGettingParameter( String cmd, String p )
{
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	35

```

if( mR900Manager != null )
{
    mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_GET_PARAM, new
    String[]{ cmd, p } ) );
}
}

```

#### 4.1.26 sendSettingTxPower

출력 레벨을 설정하기 위한 함수 입니다.

```
public void sendSettingTxPower( int a )
```

##### Parameters

a

출력 감쇄량이며 단위는 1dB 입니다. 0 또는 음의값을 설정해야 합니다.

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```

protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

```

```

public void sendSettingTxPower( int a )
{
    if( mR900Manager != null )
    {

```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	36

```

mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_SET_TX_POWER, new
int[] { a } ) );
}
}

```

#### 4.1.27 sendGetMaxPower

R900의 최대 출력 전력을 확인하기 위한 함수입니다.

```
public void sendGetMaxPower()
```

##### Parameters

None

##### Return Values

None

##### Remarks

None

##### 사용예제

##### JAVA

##### BluetoothActivity.java

```

protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

public void sendGetMaxPower()
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_GET_MAX_POWER ) );
    }
}

```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	37

#### 4.1.28 sendSettingTxCycle

한 채널에서 송신시간과 휴지시간을 지정합니다.

**public void sendSettingTxCycle( int on, int off)**

##### Parameters

on

최대 송출시간을 지정합니다. 단위는 msec입니다.

off

최대 휴지시간을 지정합니다. 단위는 msec입니다.

##### Return Values

None

##### Remarks

나라에 따라 관련 규정이 다르므로 규정에 맞도록 설정해야 합니다.

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendSettingTxCycle( int on, int off)
```

```
{
```

```
    if( mR900Manager != null )
```

```
    {
```

```
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_SET_TX_CYCLE, new int[]{ on, off } ) );
```

```
    }
```

```
}
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	38

#### 4.1.29 sendChangeChannelState

채널의 사용 여부를 설정하기 위한 함수입니다.

**public void sendChangeChannelState( int n, int f\_e )**

##### Parameters

n

채널 번호를 설정합니다.

f\_e

채널 사용 여부를 설정합니다.

0 : 채널 사용을 중단합니다.

1 : 채널 사용을 합니다.

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendChangeChannelState( int n, int f_e )
```

```
{
```

```
if( mR900Manager != null )
```

```
{
```

```
    mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_CHANGE_CH_STATE, new
    int[]{ n, f_e } ) );
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	39

```
}
}
```

#### 4.1.30 sendSettingCountry

모듈이 동작할 지역을 설정하기 위한 함수입니다.

```
public void sendSettingCountry( int code )
```

##### Parameters

code

사용할 국가 코드를 설정합니다.

##### Return Values

None

##### Remarks

None

##### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendSettingCountry( int code )
```

```
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_CHANGE_CH_STATE, new
        int[]{ code } ) );
    }
}
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	40

#### 4.1.31 sendGettingCountry

모듈이 동작할 지역을 확인하기 위한 함수입니다.

**public void sendGettingCountry()**

##### Parameters

None

##### Return Values

None

##### Remarks

None

##### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void sendGettingCountry()
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_GET_COUNTRY_CAP ) );
    }
}
```

#### 4.1.32 sendSetLockTagMemStatePerm

태그 특정 메모리를 영구 변경하는 기능을 제공하는 함수입니다.



	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	41

---

**public void sendSetLockTagMemStatePerm( int mem\_id, int f\_l, String ACS\_PWD )**

---

#### Parameters

mem\_id

해당 메모리를 지정합니다.

0 : USER 영역

1 : TID 영역

2 : EPC 영역

3 : Access password 영역

4 : Kill password 영역

f\_l

영구히 변경되도록 지정합니다.

0 : 영구 변경되지 않습니다.

1 : 영구 변경됩니다.

ACS\_PWD

태그 Access password를 지정합니다.

#### Return Values

None

#### Remarks

영구 변경 되면, 다시 변경이 안되므로 주의해야 합니다. 영구 변경한 메모리 영역은 write가 불가능하며, access password를 사용해야만 write가 가능합니다.

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendSetLockTagMemStatePerm( int mem_id, int f_l, String ACS_PWD )
```

```
{
```

```
    if( mR900Manager != null )
```

---

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	42

```

{
    mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_SET_LOCK_TAG_MEM,
    new int[]{ mem_id, f_l }, ACS_PWD,
    new int[]{ mSingleTag ? 1 : 0, mUseMask ? ( mQuerySelected ? 3 : 2 ) : 0, mTimeout } ) );
}
}

```

#### 4.1.33 sendPauseTx

R900의 전송을 일시 중단시키는 함수입니다.

```
public void sendPauseTx()
```

##### Parameters

None

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```


protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

```

```

public void sendPauseTx()
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_PAUSE_TX ) );
    }
}

```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	43

}

#### 4.1.34 sendStatusReporting

R900의 상태 변화를 report 하기위한 함수입니다.

**public void sendStatusReporting( int f\_link )**

##### Parameters

f\_link

상태 변화에 대한 리포트 여부를 설정합니다.

0 : 리포트를 하지 않습니다.

1 : 리포트를 합니다.

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendStatusReporting( int f_link )
```

```
{
```

```
    if( mR900Manager != null )
```

```
    {
```

```
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_STATUS_REPORT,
        new int[]{ f_link } ) );
```

```
    }
```

```
}
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	44

#### 4.1.35 sendInventoryReportingFormat

Inventory report format을 지정합니다.

---

**public void sendInventoryReportingFormat( int f\_time, int f\_rssi )**

---

##### Parameters

f\_time

Inventory 한 시간을 출력합니다.

f\_rssi

Inventory 시의 rssi 값을 출력합니다.

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendInventoryReportingFormat( int f_time, int f_rssi )
```

```
{
```

```
    if( mR900Manager != null )
```

```
    {
```

```
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_INVENT_REPORT_FORMAT,
        new int[]{ f_time, f_rssi } ) );
```

```
    }
```

```
}
```

---

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	45

#### 4.1.36 sendDislink

Open 상태를 해지하기 위한 함수입니다.

**public void sendDislink()**

##### Parameters

None

##### Return Values

None

##### Remarks

None

##### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void sendDislink()
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_DISLINK ) );
    }
}
```

#### 4.1.37 sendUploadingTagData

R900에서 Local 모드에서 읽은 태그 정보를 upload하기 위한 함수입니다.

**public void sendUploadingTagData( int index, int count )**

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	46

#### Parameters

index

데이터 리스트를 지정합니다. 기본값은 0입니다.

count

태그 데이터 수를 지정합니다.

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

public void sendUploadingTagData( int index, int count )
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_UPLOAD_TAG_DATA,
            new int[]{ index, count } ) );
    }
}
```

#### 4.1.38 sendClearingTagData

R900에 저장된 태그 정보를 clear하기 위한 함수입니다.

```
public void sendClearingTagData()
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	47

#### Parameters

None

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void sendClearingTagData()
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_CLEAR_TAG_DATA ) );
    }
}
```

#### 4.1.39 sendAlertReaderStatus

R900 상태 변화시 report 여부를 설정하기 위한 함수입니다.

```
public void sendAlertReaderStatus( int f_link, int f_trigger, int f_lowbat, int f_autooff, int f_pwr )
```

#### Parameters

f\_link

링크 상태에 따른 리포트 여부를 지정합니다.

0 : 링크 상태 변화 시, 리포트를 하지 않습니다.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	48

1 : 링크 상태 변화 시, 리포트를 합니다.

f\_trigger

트리거 버튼 상태에 따른 리포트 여부를 지정합니다.

0 : 트리거 버튼 상태 변화 시, 리포트를 하지 않습니다.

1 : 트리거 버튼 상태 변화 시, 리포트를 합니다.

f\_lowbat

배터리 상태에 따른 리포트 여부를 지정합니다.

0 : 배터리 상태 변화 시, 리포트를 하지 않습니다.

1 : 배터리 상태 변화 시, 리포트를 합니다.

f\_autooff

R900 auto off 상태에 따른 리포트 여부를 지정합니다.

0 : auto off 상태 변화 시, 리포트를 하지 않습니다.

1 : auto off 상태 변화 시, 리포트를 합니다.

f\_pwr

R900이 종료 상태에 따른 리포트 여부를 지정합니다.

0 : R900 종료 시, 리포트를 하지 않습니다.

1 : R900 종료 시, 리포트를 합니다.

## Return Values

None

## Remarks

None

## 사용예제

### JAVA

#### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendAlertReaderStatus( int f_link, int f_trigger, int f_lowbat, int f_autooff, int f_pwr )
```

```
{
```

```
    if( mR900Manager != null )
```

```
{
```



	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	49

```

mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_ALERT_READER_STATUS,
new int[]{ f_link, f_trigger, f_lowbat, f_autooff, f_pwr } ) );
}
}

```

---

#### 4.1.40 sendGettingStatusWord

R900 상태값을 출력하기 위한 함수입니다.

---

**public void sendGettingStatusWord()**

---

##### Parameters

None

##### Return Values

None

##### Remarks

None

##### 사용예제

##### JAVA

##### BluetoothActivity.java

```

protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );

```

```

public void sendGettingStatusWord()
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_GET_STATUS_WORD ) );
    }
}

```

---

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	50

#### 4.1.41 sendSettingBuzzerVolume

R900 부저 볼륨값을 변경하기 위한 함수입니다.

---

**public void sendSettingBuzzerVolume( int volume, int f\_nv )**

---

##### Parameters

volume

변경하기 위한 볼륨값을 지정합니다.

0 : Mute

1 : middle

2 : high

f\_nv

EEProm 저장 여부를 지정합니다.

0 : 전원을 끄면, 마지막 상태로 환원합니다.

1 : 전원을 꺼도 현재 지정된 상태를 기억합니다.

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendSettingBuzzerVolume( int volume, int f_nv )
```

```
{
```

```
    if( mR900Manager != null )
```

```
    {
```

```
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_SET_BUZZER_VOL,
```

```
        new int[]{ volume, f_nv } ) );
```

```
    }
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	51

}

#### 4.1.42 sendBeep

R900 부저로 비프음을 발생시키기 위한 함수입니다.

**public void sendBeep( int f\_on )**

##### Parameters

f\_on

비프음을 발생 여부를 지정합니다.

0 : 비프음을 발생하지 않습니다.

1 : 비프음을 발생합니다.

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendBeep( int f_on )
```

```
{
```

```
    if( mR900Manager != null )
```

```
    {
```

```
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_BEEP,
        new int[]{ f_on } ) );
```

```
    }
```

```
}
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	52

#### 4.1.43 sendSettingAutoPowerOffDelay

리더의 자동 종료 시간 값을 변경하기 위한 함수입니다.

```
public void sendSettingAutoPowerOffDelay( int delay, int f_nv )
```

##### Parameters

delay

자동 종료 시간을 지정합니다. 단위는 msec입니다.

f\_nv

EEProm에 기록 여부를 지정합니다.

0 : 전원을 끄면 마지막 상태로 환원합니다.

1 : 전원을 꺼도 현재 지정된 설정 시간을 기억합니다.

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendSettingAutoPowerOffDelay( int delay, int f_nv )
```

```
{
```

```
    if( mR900Manager != null )
```

```
    {
```

```
        mR900Manager.sendData(R900Protocol.makeProtocol(
```

```
        R900Protocol.CMD_SET_AUTO_POWER_OFF_DELAY, new int[]{ delay, f_nv } ) );
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	53

```
}
}
```

#### 4.1.44 sendGettingBatteryLevel

R900 배터리 잔량을 체크하기 위한 함수입니다.

```
public void sendGettingBatteryLevel( int f_ext )
```

##### Parameters

f\_ext

테스트 정보를 포함합니다. 기본 설정값은 0 입니다.

##### Return Values

None

##### Remarks

본 함수는 Engineering 함수입니다.

##### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendGettingBatteryLevel( int f_ext )
```

```
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_GET_BATT_LEVEL,
            new int[]{ f_ext } ) );
    }
}
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	54

#### 4.1.45 sendReportingBatteryState

R900 배터리 상태를 리포트하기 위한 함수입니다.

---

**public void sendReportingBatteryState( int f\_report )**

---

##### Parameters

f\_report

리포트 여부를 설정합니다.

0 : 리포트를 하지 않습니다.

1 : 리포트를 합니다.

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### BluetoothActivity.java

```
protected R900Manager mR900Manager;
```

```
mR900Manager = new R900Manager( mHandler );
```

```
public void sendReportingBatteryState( int f_report )
```

```
{
```

```
    if( mR900Manager != null )
```

```
    {
```

```
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_REPORT_BATT_STATE,
        new int[]{ f_report } ) );
```

```
    }
```

```
}
```

---

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	55

#### 4.1.46 sendTurningReaderOff

R900을 종료하기 위한 함수입니다.

**public void sendTurningReaderOff()**

##### Parameters

None

##### Return Values

None

##### Remarks

None

##### 사용예제

##### JAVA

##### BluetoothActivity.java

```
protected R900Manager mR900Manager;
mR900Manager = new R900Manager( mHandler );
```

```
public void sendTurningReaderOff()
{
    if( mR900Manager != null )
    {
        mR900Manager.sendData(R900Protocol.makeProtocol( R900Protocol.CMD_TURN_READER_OFF ) );
    }
}
```

#### 4.2 OnBtEventListener.java

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	56

블루투스 이벤트 연결을 정의합니다.

#### 4.2.1 onBtFoundNewDevice

블루투스 스캔 시, 새 장비 등록을 전달하기 위한 이벤트입니다.

`void onBtFoundNewDevice( BluetoothDevice device )`

##### Parameters

device

블루투스 장비 정보입니다.

##### Return Values

None

##### Remarks

None

#### 4.2.2 onBtScanCompleted

블루투스 스캔 완료를 전달하기 위한 이벤트입니다.

`void onBtScanCompleted()`

##### Parameters

None

##### Return Values

None


##### Remarks

None

#### 4.2.3 onBtConnected

블루투스 연결하기 위한 이벤트입니다.



	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	57

---

**void onBtConnected( BluetoothDevice device )**

---

**Parameters**

device

블루투스 장비 정보입니다.

**Return Values**

None

**Remarks**

None

---

**4.2.4 onBtDisconnected**

블루투스 해지를 전달하기 위한 이벤트입니다.

---

**void onBtDisconnected( BluetoothDevice device )**

---

**Parameters**

device

블루투스 장비 정보입니다.

**Return Values**

None

**Remarks**

None

---

**4.2.5 onBtConnectFail**

블루투스 연결 실패를 전달하기 위한 이벤트입니다.

---

**void onBtConnectFail( BluetoothDevice device, String msg )**

---

**Parameters**

**CONFIDENTIAL** Document D.O.Tel. All rights of reproduction and disclosure reserved.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	58

device

블루투스 장비 정보입니다.

msg

해당 메시지를 전달합니다.

#### Return Values

None

#### Remarks

None

#### 4.2.6 onBtDataSent

블루투스 장비로 데이터를 전달하기 위한 이벤트입니다.

**void onBtDataSent( byte[] data )**

#### Parameters

data

블루투스 장비로부터 수신된 데이터를 전달합니다.

#### Return Values

None

#### Remarks

None


#### 4.2.7 onBtDataTransException

블루투스 예외발생에 대한 데이터를 전달하기 위한 이벤트입니다.

**void onBtDataTransException( BluetoothDevice device, String msg )**

#### Parameters

device

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	59

블루투스 장비 정보입니다.

msg

예외 발생에 대한 정보를 전달하기 위한 메시지입니다.

#### Return Values

None

#### Remarks

None

#### 4.2.8 onNotifyBtDataRecv

블루투스 장비로부터 받은 데이터를 전달하기 위한 이벤트입니다.

**void onNotifyBtDataRecv()**

#### Parameters

None

#### Return Values

None

#### Remarks

None

### 4.3 R900Manager.java

R900 연결을 제어합니다.

#### 4.3.1 isBluetoothEnabled

블루투스 활성화를 요청하기 위한 함수입니다.

**public boolean isBluetoothEnabled()**

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	60

#### Parameters

None

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### R900Manager.java

```
private BluetoothAdapter mBluetoothAdapter;
mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
```

```
public boolean isBluetoothEnabled()
{
    if( mBluetoothAdapter == null )
        return false;
    return mBluetoothAdapter.isEnabled();
}
```

#### 4.3.2 enableBluetooth

블루투스 활성화가 안된 경우 활성화를 요청하기 위한 함수입니다.

```
public void enableBluetooth( Activity host )
```

#### Parameters

host

#### Return Values

None

#### Remarks

**CONFIDENTIAL** Document D.O.Tel. All rights of reproduction and disclosure reserved.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	61

None

#### 사용예제

##### JAVA

##### R900Manager.java

```
private BluetoothAdapter mBluetoothAdapter;
mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

public void enableBluetooth( Activity host )
{
    if( mBluetoothAdapter == null || mBluetoothAdapter.isEnabled() == false )
    {
        Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        host.startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
    }
}
```

#### 4.3.3 queryPairedDevices

다른 블루투스 기기와 연결 여부를 확인하기 위한 함수입니다.

```
public Set<BluetoothDevice> queryPairedDevices()
```

##### Parameters

None

##### Return Values

None


##### Remarks

None

#### 사용예제

##### JAVA

**CONFIDENTIAL** Document D.O.Tel. All rights of reproduction and disclosure reserved.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	62

## R900Manager.java

```
private BluetoothAdapter mBluetoothAdapter;
mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
```

```
public Set<BluetoothDevice> queryPairedDevices()
{
    if( mBluetoothAdapter != null )
        return mBluetoothAdapter.getBondedDevices();
    return null;
}
```

### 4.3.4 startDiscovery

주변 블루투스 장비를 검색하는 함수입니다.

```
public void startDiscovery()
```

#### Parameters

None

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### R900Manager.java

```
public void startDiscovery()
{
    stopDiscovery();
    mBluetoothAdapter.startDiscovery();
}
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	63

#### 4.3.5 stopDiscovery

블루투스 검색중인 경우 검색을 종료하기 위한 함수입니다.

**public void stopDiscovery()**

##### Parameters

None

##### Return Values

None

##### Remarks

None

##### 사용예제

###### JAVA

###### R900Manager.java

```
private BluetoothAdapter mBluetoothAdapter;
mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

public void stopDiscovery()
{
    if( mBluetoothAdapter != null && mBluetoothAdapter.isDiscovering() )
        mBluetoothAdapter.cancelDiscovery();
}
```

#### 4.3.6 getBluetoothDevice

블루투스 mac 주소를 블루투스 장비 객체를 반환하는 함수입니다.

**public BluetoothDevice getBluetoothDevice( String address )**

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	64

#### Parameters

address

블루투스 mac 주소

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### R900Manager.java

```
private BluetoothAdapter mBluetoothAdapter;
mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

public BluetoothDevice getBluetoothDevice( String address )
{
    if( mBluetoothAdapter != null )
        return mBluetoothAdapter.getRemoteDevice(address);
    return null;
}
```

#### 4.3.7 sendData

블루투스 장비로 데이터를 전송하기 위한 함수입니다.

```
public void sendData( byte[] bytes )
```

#### Parameters

bytes

전송하기 위한 데이터를 지정합니다.



	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	65

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### R900Manager.java

```

private BluetoothSocket mBluetoothSocket;
private ConnectedThread mConnectedThread;
mConnectedThread = new ConnectedThread(mBluetoothSocket);

public void sendData( byte[] bytes )
{
    if( mConnectedThread != null )
    {
        Log.d(BluetoothActivity.TAG, "Send : " + new String(bytes));
        mConnectedThread.write(bytes);
    }
    else
    {
        if( mBtEventListener != null )
            mBtEventListener.onBtDataTransException(mBluetoothDevice,
                "[Bluetooth Socket] Write Fail : Bluetooth Thread is not running.");
    }
}

```

#### 4.3.8 isTryingConnect

블루투스 연결 여부를 확인하기 위한 함수입니다.

##### public boolean isTryingConnect()

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	66

#### Parameters

None

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### R900Manager.java

```
public boolean isTryingConnect()
{
    return mConnectThread != null;
}
```

## 4.4 R900Protocol.java.

R900 프로토콜을 제어합니다.

### 4.4.1 makeProtocol

R900과 통신하기 위한 프로토콜을 변환하는 함수입니다.

```
public static final byte[] makeProtocol( String cmd, int[] param )
```

#### Parameters

cmd

명령어를 지정합니다.

param

명령어에 해당하는 parameter를 지정합니다.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	67

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### R900Protocol.java

```

public static final byte[] makeProtocol( String cmd, int[] param )
{
    StringBuilder protocol = new StringBuilder();
    protocol.append(cmd);

    if( param != null && param.length > 0 )
    {
        for( int i = 0; i < param.length; ++i )
        {
            protocol.append(',');
            if( param[ i ] != SKIP_PARAM )
                protocol.append(param[ i ]);
        }
    }
    return string2bytes( protocol.toString() );
}

```

#### 4.4.2 makeProtocol

R900과 통신하기 위한 프로토콜을 변환하는 함수입니다.

```
public static final byte[] makeProtocol( String cmd )
```

#### Parameters

cmd

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	68

명령어를 지정합니다.

#### Return Values

None

#### Remarks

None

#### 사용예제

##### JAVA

##### R900Protocol.java

```
public static final byte[] makeProtocol( String cmd )
{
    StringBuilder protocol = new StringBuilder();
    protocol.append(cmd);
    return string2bytes( protocol.toString() );
}
```

#### 4.4.3 makeProtocol

R900과 통신하기 위한 프로토콜을 변환하는 함수입니다.

```
public static final byte[] makeProtocol( String cmd, String[] options )
```

#### Parameters

cmd

명령어를 지정합니다.

options

명령어에 해당하는 options을 지정합니다.

#### Return Values

None

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	69

#### Remarks

None

#### 사용예제

##### JAVA

##### R900Protocol.java

```

public static final byte[] makeProtocol( String cmd, String[] options )
{
    StringBuilder protocol = new StringBuilder();
    protocol.append(cmd);

    if( options != null && options.length > 0)
    {
        for( int i = 0; i < options.length; ++i )
        {
            protocol.append( "," );
            if( options[ i ] != null )
                protocol.append( options[ i ] );
        }
    }
    return string2bytes( protocol.toString() );
}

```

#### 4.4.4 makeProtocol

R900과 통신하기 위한 프로토콜을 변환하는 함수입니다.

```
public static final byte[] makeProtocol( String cmd, int[] param, String[] options, int[] param2 )
```

#### Parameters

cmd

명령어를 지정합니다.

param

명령어에 대한 parameter를 지정합니다.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	70

options

명령어에 대한 option을 지정합니다.

param2

명령어에 대한 parameter를 지정합니다.

#### Return Values

None

#### Remarks

None

#### JAVA

##### R900Protocol.java

```

public static final byte[] makeProtocol( String cmd, int[] param, String[] options, int[] param2 )
{
    StringBuilder protocol = new StringBuilder();
    protocol.append(cmd);

    if( param != null && param.length > 0 )
    {
        for( int i = 0; i < param.length; ++i )
        {
            protocol.append(',');
            if( param[ i ] != SKIP_PARAM )
                protocol.append(param[ i ]);
        }
    }

    if( options != null )
    {
        for( int i = 0; i < options.length; ++i )
        {
            protocol.append( "," );
            if( options[ i ] != null )

```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	71

```

        protocol.append( options[ i ] );
    }
}

if( param2 != null && param2.length > 0 )
{
    for( int i = 0; i < param2.length; ++i )
    {
        protocol.append(',');

        if( param2[ i ] != SKIP_PARAM )
            protocol.append(param2[ i ]);
    }
}

return string2bytes( protocol.toString() );
}

```

#### 4.4.6 makeProtocol

R900과 통신하기 위한 프로토콜을 변환하는 함수입니다.

```
public static final byte[] makeProtocol( String cmd, int[] param, String option, int[] param2 )
```

##### Parameters

cmd

명령어를 지정합니다.

param

명령어에 대한 parameter를 설정합니다.

option

명령어에 대한 option을 설정합니다.

param2

명령어에 대한 parameter를 설정합니다.

##### Return Values

**CONFIDENTIAL** Document D.O.Tel. All rights of reproduction and disclosure reserved.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	72

None

## Remarks

None

## JAVA

### R900Protocol.java

```

public static final byte[] makeProtocol( String cmd, int[] param, String option, int[] param2 )
{
    StringBuilder protocol = new StringBuilder();
    protocol.append(cmd);

    if( param != null && param.length > 0 )
    {
        for( int i = 0; i < param.length; ++i )
        {
            protocol.append(',');
            if( param[ i ] != SKIP_PARAM )
                protocol.append(param[ i ]);
        }
    }

    protocol.append( "," );
    if( option != null )
        protocol.append( option );

    if( param2 != null && param2.length > 0 )
    {
        for( int i = 0; i < param2.length; ++i )
        {
            protocol.append(',');
            if( param2[ i ] != SKIP_PARAM )
                protocol.append(param2[ i ]);
        }
    }
}

```



	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	73

```

return string2bytes( protocol.toString() );
}

```

#### 4.4.7 makeProtocol

R900과 통신하기 위한 프로토콜을 변환하는 함수입니다.

```
public static final byte[] makeProtocol( String cmd, String option, int[] param2 )
```

##### Parameters

cmd

명령어를 지정합니다.

options

명령어에 대한 option을 지정합니다.

param2

명령어에 대한 parameter를 지정합니다.

##### Return Values

None

##### Remarks

None

##### JAVA

##### R900Protocol.java

```

public static final byte[] makeProtocol( String cmd, String option, int[] param2 )
{
    StringBuilder protocol = new StringBuilder();
    protocol.append(cmd);

    protocol.append( "," );
    if( option != null )
        protocol.append( option );
}

```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	74

```

if( param2 != null && param2.length > 0 )
{
    for( int i = 0; i < param2.length; ++i )
    {
        protocol.append(',');

        if( param2[ i ] != SKIP_PARAM )
            protocol.append(param2[ i ]);
    }
}
return string2bytes( protocol.toString() );
}

```

#### 4.4.8 string2bytes

String 데이터를 byte로 변환하는 함수입니다.

```
public static final byte[] string2bytes( String str )
```

##### Parameters

str

byte로 변환하기 위한 데이터입니다.

##### Return Values

None

##### Remarks

None

##### JAVA

##### R900Protocol.java

```

public static final byte[] string2bytes( String str )
{
    char[] charProtocol = str.toCharArray();

```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	75

```

byte[] byteProtocol = new byte[charProtocol.length + getTypeSize()];
int index = 0;
for( int i = 0; i < charProtocol.length; ++i, ++index )
    byteProtocol[ index ] = (byte) ( charProtocol[ i ] & 0xff );

// ---
for( int i = 0; i < getTypeSize(); ++i, ++index )
    byteProtocol[ index ] = getType()[ i ];

return byteProtocol;
}

```

## 4.5 R900RecvPacketParser.java

데이터 패킷을 받아서 파싱을 합니다.

### 4.5.1 pushPacket

데이터를 송신하기 위한 함수입니다.

---

**synchronized public void pushPacket( byte[] buffer, int len )**

---

#### Parameters

buffer

송신하기 위한 데이터를 지정합니다.

len

송신하기 위한 데이터 길이를 지정합니다.

#### Return Values

None

#### Remarks

None

#### JAVA

**CONFIDENTIAL** Document D.O.Tel. All rights of reproduction and disclosure reserved.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	76

## R900RecvPacketParser.java

```
synchronized public void pushPacket( byte[] buffer, int len )
{
    if( mCharBuffSize < len )
    {
        mCharBuffSize = ( len << 1 );
        mCharBuff = new char[ mCharBuffSize ];
    }
    for( int i = 0; i < len; ++i )
        mCharBuff[ i ] = (char)( buffer[ i ] & 0xff );
    mPacket.append( mCharBuff, 0, len );
}
```

## 4.5.2 popPacket

데이터를 수신하기 위한 함수입니다.

```
synchronized public String popPacket( int offset, int len )
```

### Parameters

offset

데이터 시작 위치를 지정합니다.

len

데이터 길이를 지정합니다.

### Return Values

None

### Remarks

None

## JAVA

## R900RecvPacketParser.java

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	77

```
synchronized public String popPacket( int offset, int len )
{
    String pop = mPacket.substring(offset, offset + len);
    mPacket.delete(0, offset + len);
    return pop;
}
```

#### 4.5.3 popPacket

R900으로 송신하기 위한 함수입니다.

```
synchronized public String popPacket()
```

##### Parameters

None

##### Return Values

None

##### Remarks

None

##### JAVA

**R900RecvPacketParser.java**

```
synchronized public String popPacket()
{
    final String STR_PACKET = mPacket.toString();
    final String DELIMITER = R900Protocol.getDelimiter();

    int cmdIndex = STR_PACKET.indexOf("$>");
    if( cmdIndex >= 0 )
    {
        if( STR_PACKET.replace("\n", "").indexOf("$>") == 0 )
            return popPacket(cmdIndex, 2);
    }
}
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	78

}

```
StringTokenizer st = new StringTokenizer(STR_PACKET, DELIMITER);
if( st.hasMoreTokens() )
{
    final String str = st.nextToken();
    if( str != null && str.length() > 0 )
    {
        if( str.length() + DELIMITER.length() > mPacket.length() )
            return null;
        if( str.length() == 1 || str.length() > 40 )
            Log.d("kueen108", "Somethins is wrong!!!");
        mPacket.delete(0, Math.min(mPacket.length(), str.length() + DELIMITER.length()));
        return str;
    }
}
return null;
}
```

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	79

## 5. DEMO 설치방법

프로그램 설치를 위해 제공하는 “RFIDUsbHost.apk” 파일을 안드로이드 폰에 복사합니다.  
해당하는 \*.apk 파일을 복사 후, 클릭합니다.

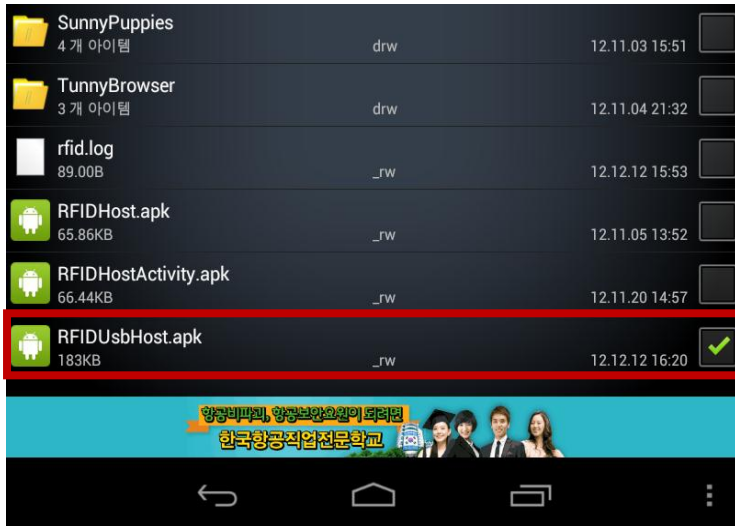


그림 1. apk 파일 복사

“설치” 버튼을 누르면 설치가 되며, “취소” 버튼을 누르면 설치를 취소합니다.

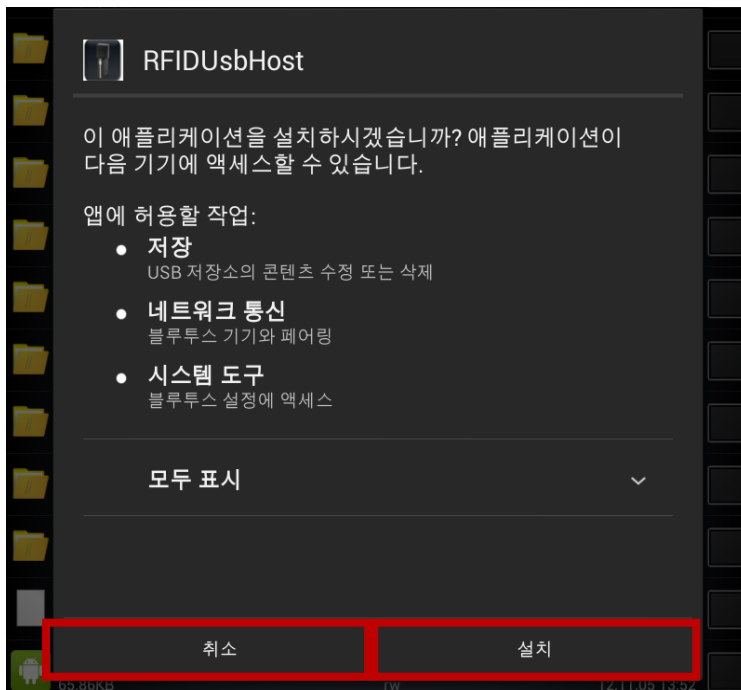


그림 2. RFIDUhfHost 프로그램 설치화면

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	80

“완료” 버튼을 누르면 설치를 종료하고, “열기” 버튼을 누르면 설치된 프로그램을 실행합니다.

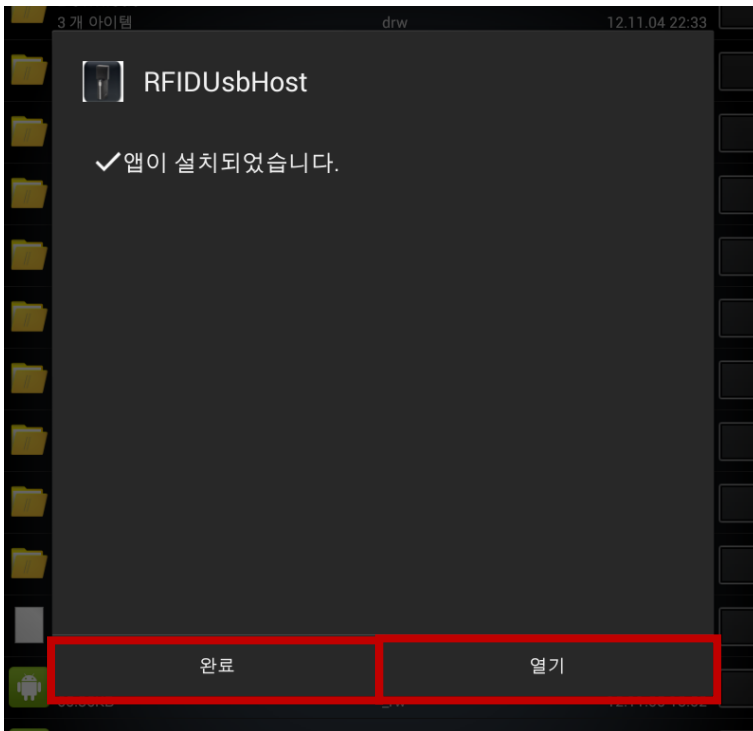


그림 3. RFIDUsbHost 프로그램 설치완료

RFIDUsbHost 프로그램 설치가 완료되면 Apps에 (그림4)와 같은 아이콘이 생성됩니다.

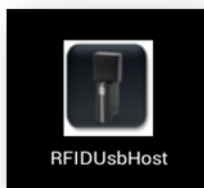


그림 4. RFIDUsbHost 아이콘

## 6. DEMO 사용방법

본 문서는 RFIDUsbHost 프로그램의 주요 기능을 설명한다. RFIDUsbHost 프로그램은 USB케이블로 연결하는 “USB 모드”와 “BT 모드”로 2가지 동작 모드를 지원합니다.



	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	81

RFIDUsbHost 프로그램을 실행하면 초기 화면은 다음과 같습니다.



그림 1. RFIDUsbHost 프로그램 메인 화면

"Bluetooth"를 선택하면 리더기와 Bluetooth 연결을 지원합니다.

"USB"를 선택하면 리더기와 USB 연결을 지원합니다.

주의사항 :

USB 모드 사용 시 : 리더기와 안드로이드 폰을 케이블로 연결해야 합니다.

BT 모드 사용 시 : 리더기와 안드로이드 폰에 연결된 케이블을 분리해야 합니다.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	82

## USB 모드

USB 모드를 사용하기 위해서는 리더기와 폰에 OTG케이블을 사용합니다.



그림 2. OTG 케이블


주의 : OTG 케이블은 안드로이드 폰이 Host 기능을 제공합니다.

OTG 케이블은 USB 호스트 기능을 지원하는 기기에서만 사용이 가능합니다. 반드시 지원 가능 여부를 확인하시고 사용해야 합니다.

안드로이드 폰에 OTG 케이블을 연결 후, 리더기에 USB 케이블을 연결한 후, OTG 케이블에 연결합니다.

<Table 1> 마이크로 5핀 OTG 지원 기종표

마이크로 5핀 OTG 지원 기종표	
삼성	갤럭시 S2(SHW-M250S/M250K/M250L), 갤럭시 S2 LTE(SHW-E110S)
	갤럭시 S2 HD LTE(SHV-E120S/E120L), 갤럭시 노트(SHV-E160L/E160K/E160S)
	갤럭시 S3(SHV-M440S), 갤럭시 S3 LTE(SHV-E210S/E210K/E210L)
	갤럭시 넥서스(SHW-M420S/M420K)
노키아	N810/N8
모토로라	Xoom
소니	엑스페리아 아크(Xperia Arc)(글로벌 283롬이 올라간 아트에서만 동작)
포유디지털	아이유즈 TX97, TX82S
도시바	Archos G9/TG01
Google	Nexus 7/10
누리안	M9
KT테크	테이크 HD(KM-S300)

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	83

RFIDUsbHost 프로그램 시작 화면에서 USB 버튼을 클릭합니다.



그림 3. LINK 화면 (USB 모드)

"Connect" 버튼을 누르면 리더기와 USB 통신으로 연결을 시도합니다.

"Disconnect" 버튼을 누르면 기존에 연결된 USB 통신을 종료하며, RFIDUsbHost 프로그램 시작 화면으로 돌아갑니다.

연결이 되면 하단에 "Connect OK" 팝업창이 나타납니다.

"Disconnect" 되면 RFIDUsbHost 프로그램 시작 화면으로 돌아갑니다.

리더기와 연결이 완료되면, Inventory나 Access(Read/Write/Lock/kill), Config(환경설정) 메뉴를 통하여 사용자가 원하는 기능을 사용할 수 있습니다.

	<b>DOTR-900</b>	Date	2013-06-29
		Rev	1.0
		Page	84

## BT 모드

BT 모드 시, 리더기와 폰에 연결된 OTG케이블을 분리합니다.

RFIDUsbHost 프로그램 시작 화면에서 “Bluetooth”버튼을 클릭합니다.

폰에 블루투스 장치가 켜져 있지 않으면 (그림8)과 같이 블루투스 승인 요청 경고 창이 생기며, “예”를 선택하면 블루투스 장치가 켜지며, “아니오”를 선택하면, 블루투스 장치가 켜지지 않으며, 리더기와 연결하지 못합니다.

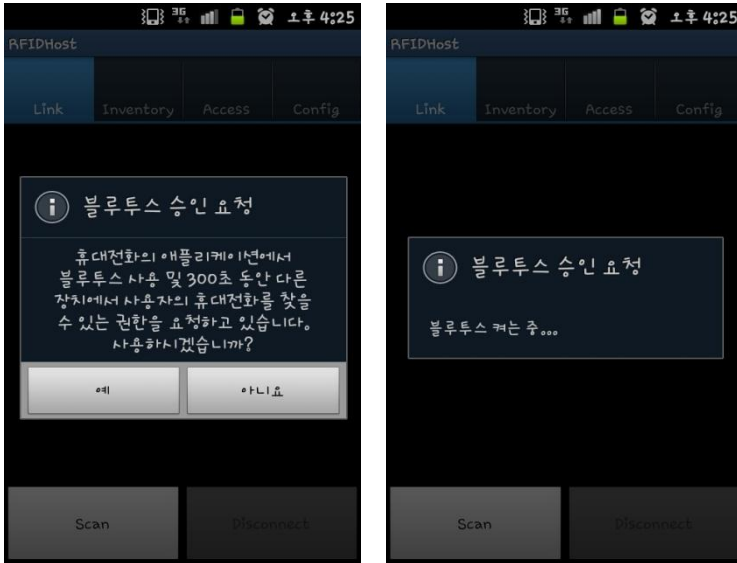


그림 4. 블루투스 승인 요청화면

블루투스 승인 요청이 끝나면, (그림 5)에서 “Scan”을 선택하여 블루투스 장비를 검색한다. 기존에 연결된 리더기가 있는 경우 자동으로 연결됩니다.

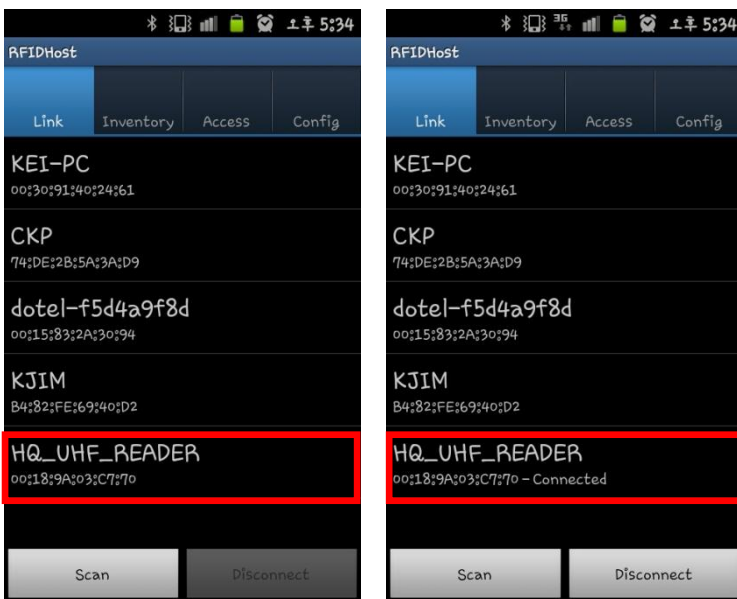


그림 5. 블루투스 연결 화면

	<h1 style="text-align: center;">DOTR-900</h1>	Date	2013-06-29
		Rev	1.0
		Page	85

검색 화면에 장치 명이 “HQ\_UHF\_READER”인 장치를 선택하면, 리더기와 연결을 시도하며, 성공하면, Mac address 옆에 “- Connected”라고 나타납니다.

리더기와 연결이 성공되면, Inventory나 Access(Read/Write/Lock/Kill), Config(환경설정) 메뉴를 통하여 사용자가 원하는 기능을 사용할 수 있습니다.

## Inventory 화면

다음은 Access 동작에 read 화면에 대한 설명이다.

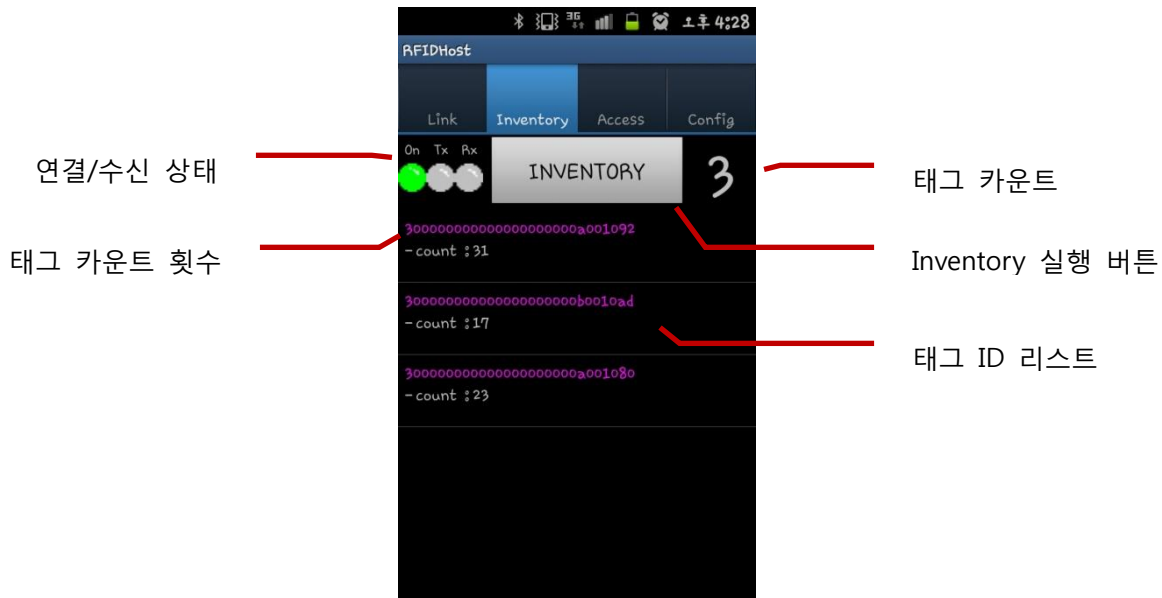


그림 6. Inventory 화면

Inventory 실행버튼을 누르면 inventory가 실행됩니다. Inventory가 실행되면 버튼의 제목이 “STOP” 으로 변경됩니다. “STOP” 버튼을 누르면 실행이 중단됩니다.

Inventory된 PC/EPC 값은 Tag ID 리스트에 표시됩니다.

연결/수신 상태에 따라 램프 색이 변경됩니다.

- On : 연결된 상태는 “녹색” 으로, 연결이 되지 않은 상태는 “빨강색” 으로 나타납니다.
- Tx : 출력하면 “빨강색” 으로, 출력하지 않으면 “회색” 으로 표시됩니다.
- Rx : 태그로부터 응답을 받은 상태는 “녹색” 으로, 태그로부터 응답을 받지 못한 상태는 “회색” 으로 표시됩니다.

태그 리스트 창을 클릭하면 리스트 메뉴(그림11)가 나타납니다. “Mask”는 태그 mask를 설정하고, “Clear”





	<h1>DOTR-900</h1>	Date	2013-06-29
		Rev	1.0
		Page	88

상태 입니다.

Secured state : 태그에 저장된 access PWD가 “00000000”인 태그가 inventory 되어 access 대기인 상태나 Open state에서 access 명령을 받아 전환된 상태 입니다.

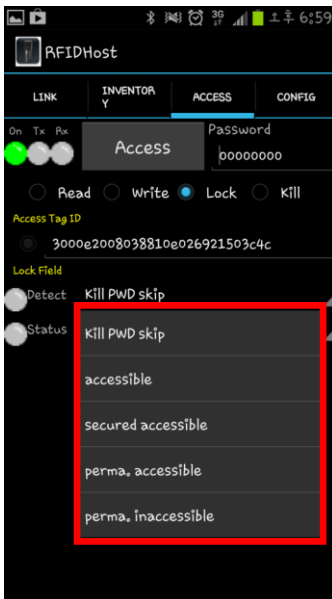


그림 10. Kill PWD Lock 화면(Access)

“accessible”은 “open” 상태나 “secured” 상태에서 kill PWD를 읽거나 변경할 수 있도록 선택합니다.

“secured accessible” 은 “secured” 상태에서 kill PWD를 읽거나 변경할 수 있도록 선택합니다.

“perma, accessible”은 “open” 상태나 “secured” 상태에서 kill PWD를 읽거나 변경할 수 있도록 선택합니다. 한번 선택하면 변경이 불가능 합니다.

“perma, accessible”은 어느 상태에서도 kill PWD를 읽거나 변경할 수 없도록 선택합니다. 한번 선택하게 되면, 변경이 불가능 합니다.



	<h1 style="text-align: center;">DOTR-900</h1>	Date	2013-06-29
		Rev	1.0
		Page	89

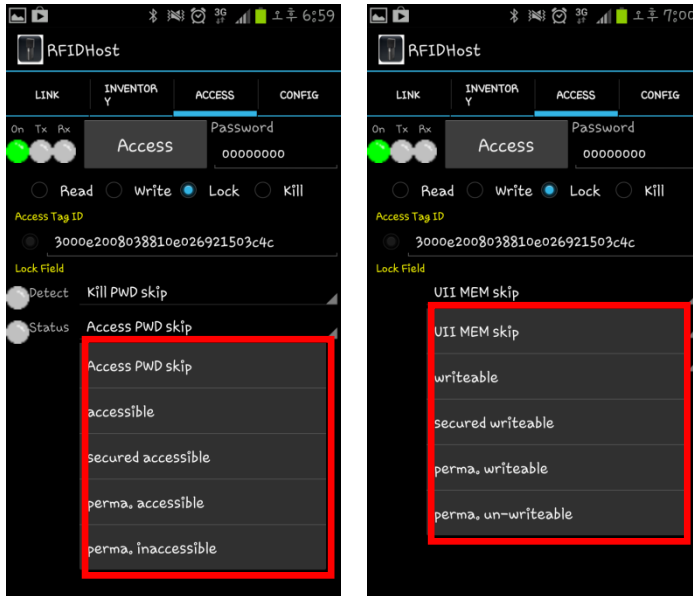


그림 11. Access PWD 및 UII MEM Lock 화면(Access)

#### <Access PWD MEM LOCK>

“accessible”은 “open” 상태나 “secured” 상태에서 kill PWD를 읽거나 변경할 수 있도록 선택합니다.

“secured accessible”은 “secured” 상태에서 lock PWD를 변경할 수 있도록 선택합니다.

“perma, accessible”은 “open”상태나 “secured” 상태에서 lock PWD를 변경할 수 있도록 선택합니다. 한번 선택하면 변경이 불가능 합니다.

“perma, inaccessible”은 어느 상태에서도 lock PWD를 변경할 수 없도록 선택합니다. 한번 선택하면 변경이 불가능 합니다.

#### <UII MEM LOCK>

“writeable”은 “open” 상태나 “secured” 상태에서 EPC memory의 내용을 변경 할 수 있도록 선택합니다.

“secured writeable”은 “secured” 상태에서 EPC memory의 내용을 변경 할 수 있도록 선택합니다.

“perma, writeable”은 “open” 상태나 “secured” 상태에서 EPC memory의 내용을 변경할 수 있도록 선택합니다. 한번 선택하면 변경이 불가능 합니다.

“perma,un-writeable”은 “open” 상태나 “secured” 상태에서 EPC memory의 내용을 변경할 수 없도록 선택합니다. 한번 선택하면 변경이 불가능합니다.

	<h1 style="text-align: center;">DOTR-900</h1>	Date	2013-06-29
		Rev	1.0
		Page	90

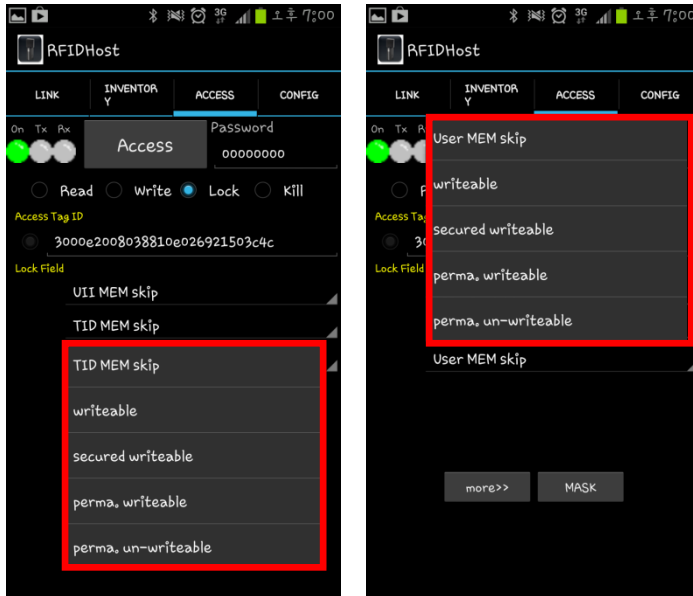


그림 12. TID MEM 및 User MEM Lock 화면(Access)

#### <TID MEM LOCK>

“writeable”은 “open” 상태나 “secured” 상태에서 TID memory의 내용을 변경 할 수 있도록 선택합니다.

“secured writeable”은 “secured” 상태에서 TID memory의 내용을 변경 할 수 있도록 선택합니다.

“perma, writeable”은 “open” 상태나 “secured” 상태에서 TID memory의 내용을 변경할 수 있도록 선택합니다. 한번 선택하면 변경이 불가능 합니다.

“perma,un-writeable”은 “open” 상태나 “secured” 상태에서 TID memory의 내용을 변경할 수 없도록 선택합니다. 한번 선택하면 변경이 불가능합니다.

#### <USER MEM LOCK>

“writeable”은 “open” 상태나 “secured” 상태에서 User memory의 내용을 변경 할 수 있도록 선택합니다.

“secured writeable”은 “secured” 상태에서 User memory의 내용을 변경 할 수 있도록 선택합니다.

“perma, writeable”은 “open” 상태나 “secured” 상태에서 User memory의 내용을 변경할 수 있도록 선택합니다. 한번 선택하면 변경이 불가능 합니다.

“perma,un-writeable”은 어느 상태에서도 User memory의 내용을 변경할 수 없도록 선택합니다. 한번 선택하면 변경이 불가능 합니다.



	<h1 style="text-align: center;">DOTR-900</h1>	Date	2013-06-29
		Rev	1.0
		Page	92

## Config 화면

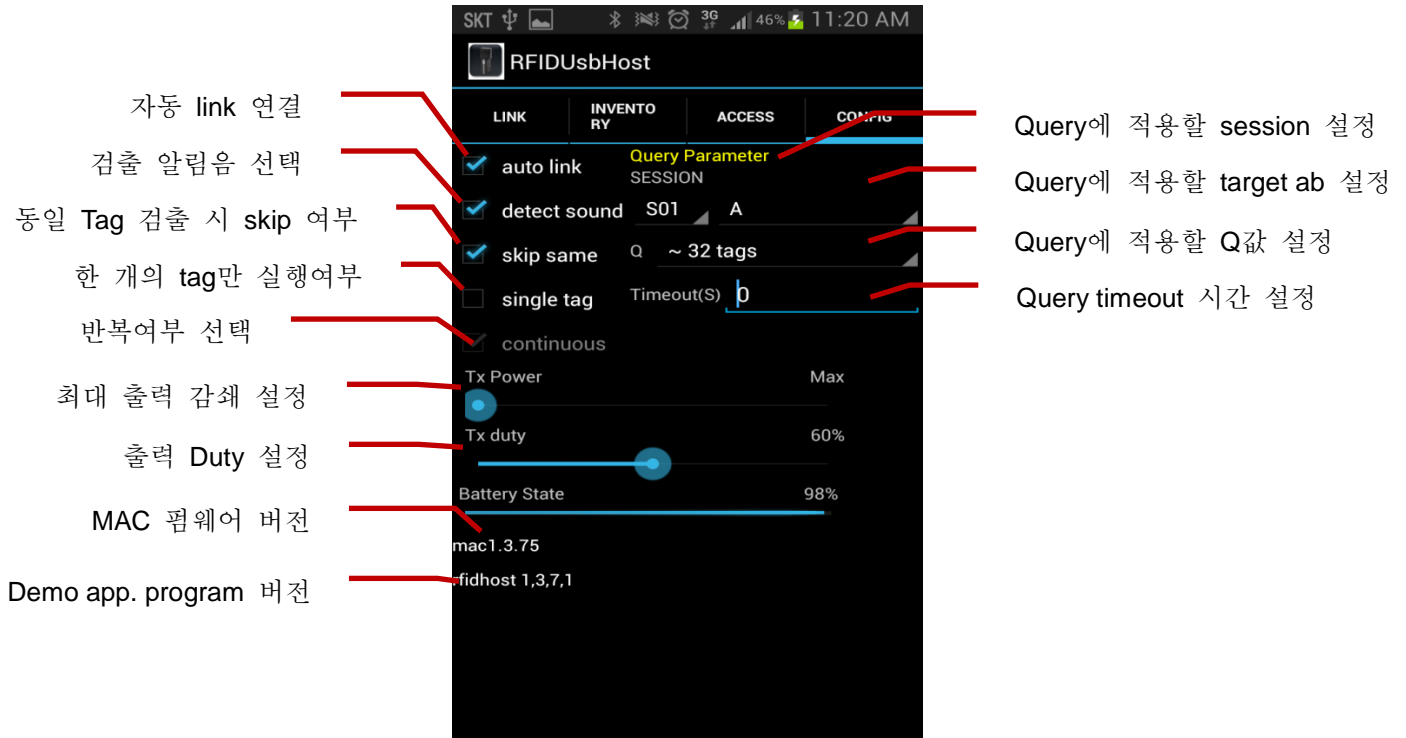


그림 14. Config 화면

Tag query시의 parameter를 설정한다. 이 설정 값은 inventory와access 실행에 모두 적용됩니다. **USB포트에서는 지원하지 않습니다.**

Tag query시 session은 S00, S01, S10, S11로 설정이 가능합니다.

Tag query시 Q값은 (1~32769tags) 범위에서 설정이 가능합니다.

Tag query시 target은 A, B, AB로 설정이 가능합니다.

Query timeout이 0이면, timeout이 없다. 설정 값의 단위는 초입니다.

최대 출력은 9dB까지 줄일 수 있습니다.

출력 Duty는 20~90%까지 조절할 수 있습니다.

Battery State는 현재 R900에 있는 배터리 충전 상태를 표시합니다.

Access 실행 시에는 single tag를 선택합니다.

Auto link는 연결이 끊어졌을 때, 자동으로 연결합니다.

MAC 펌웨어 버전을 제공합니다.

## Appendix I. Select Mask 사용방법

Select mask는 태그를 Query할 때(태그를 검출할 때) 일정 조건에 일치하는 태그만 응답을 받기 위하여 패턴을 지정해 주는 명령어 입니다. Select mask는 8개까지 지정이 가능하며, 지정된 mask는 Inventory나 access 명령 실행 시, 태그에 전송됩니다.

ISO 18000-6C/EPC global C1G2 호환 태그는 Query 동작과 관련하여 태그 안에 5개의 Flag를 가지고 있는데, 4개의 Session에 대하여 각각 Inventoried Flag를 가지고 있으며, 1 개의 Select Flag를 가지고 있습니다.

Mask command target code는 다음과 같습니다.

Code	대상 flag
0	Session 0 Inventoried flag
1	Session 1 Inventoried flag
2	Session 2 Inventoried flag
3	Session 3 Inventoried flag
4	Select flag

Mask command의 action code는 다음과 같습니다.

Code	Select flag		Inventoried flag	
	Matching tag	Non-matching tag	Matching tag	Non-matching tag
0	SET	RESET	Inventoried -> A	Inventoried -> B
1	SET	변동없음	Inventoried -> A	변동없음
2	변동없음	RESET	변동없음	Inventoried -> B
3	상태 반전	변동없음	A->B, B->A	변동없음
4	RESET	SET	Inventoried -> B	Inventoried -> A
5	RESET	변동없음	Inventoried -> B	변동없음
6	변동없음	SET	변동없음	Inventoried -> A
7	변동없음	상태반전	변동없음	A->B, B->A

Select flag를 사용하는 방법은 다음과 같습니다.

한종류의 tag 만 지정하고 싶을 때	Action 값을 0으로 한다.
한종류의 tag 만 배제하고 싶을 때	Action 값을 4로 한다.
여러 종류의 tag 를 지정하고 싶을 때	Action 값을 1로 한다.

※ 태그의 Flag는 일정 시간 상태를 유지할 수 있으므로 연속으로 Inventory나 Access를 할 경우 먼저 flag를 reset해야 합니다.

※ Inventory flag를 변경하는 mask 설정은 별도의 고려가 필요하므로 사용해서는 안됩니다.