

# IoT 플랫폼 환경의 BLE 디바이스에서 6LoWPAN을 활용한 IP 통신 방법

2016년 3월

경북대학교 통신프로토콜연구실

강형우, 김철민

[hwkang0621@gmail.com](mailto:hwkang0621@gmail.com), [cheolminkim@vanilet.pe.kr](mailto:cheolminkim@vanilet.pe.kr)

## 요 약

사물인터넷, 저전력 통신 등이 네트워크 통신의 키워드가 되면서 Bluetooth, Zigbee 등의 M2M 통신에서도 IP통신을 활용한 저전력 통신방법에 대하여 연구가 활발히 진행되고 있다. 본 문서에서는 사물인터넷 플랫폼인 Open IoT 플랫폼 환경에서의 Bluetooth LE 통신에서 6LoWPAN을 활용한 IP 통신기법을 설명하도록 하겠다.

## 목 차

1. 서론 .....	2
2. 6LoWPAN을 활용한 BLUETOOTH LE 통신 기법 .....	2
2.1 디바이스에서의 6LoWPAN 네트워크 환경 구성방법 및 테스트 .....	2
2.2 6LoWPAN을 활용한 OPEN IOT PLATFORM 서버와 통신환경 구축 .....	4
2.3 &CUBE 디바이스 플랫폼 및 TAS 설정 .....	6
3. 결론 .....	8
참고 문헌 .....	9

## 1. 서론

사물인터넷, 저전력 통신이란 키워드가 네트워크의 핫이슈가 되면서 6LoWPAN이나 Bluetooth LE 등 저전력 통신을 위한 기술들이 많이 연구가 되고 있다. 특히 사물인터넷 기반의 통신 환경을 구축하기 위해서는 저전력 통신이 필요한 센서 디바이스에서도 인터넷 프로토콜을 활용한 통신이 필요하게 된다. 무선으로 인터넷에 연결되는 장비들은 2018년 1인당 1.5개까지 증가할 것으로 전망되며, 단일 네트워크 수준을 포함하면 실제로 인터넷에 연결되는 객체의 수는 기하급수적으로 증가할 것으로 보인다. 이러한 상황에서 객체의 식별정보로서 IPv6는 기존의 검증된 네트워크 통신 구조를 그대로 활용하면서 무제한에 가까운 식별정보를 생성할 수 있기 때문에 사물인터넷 시대의 필수조건이라 할 수 있겠다.

본 문서에서는 사물인터넷 플랫폼인 Open IoT 플랫폼 환경에서의 Bluetooth LE 통신 기반의 6LoWPAN을 활용한 IP 통신 방법에 대하여 설명을 하도록 하겠다.

## 2. 사물인터넷 환경에서의 6LoWPAN을 활용한 Bluetooth LE 통신기법

### 2.1 디바이스에서의 6LoWPAN 네트워크 환경 구성방법 및 테스트

본 문서에서는 Linux Kernel기반의 embedded system과 유사한 Raspberry PI환경에서 6LoWPAN을 활용한 Bluetooth LE 통신 기법에 대하여 설명하도록 하겠다. Linux Kernel 3.18버전부터 6LoWPAN 표준을 구현하므로, Raspberry PI에서 이를 이용할 수 있도록 최신 커널을 컴파일하고 설치한다.

```
root@raspberrypi:/home/pi# uname -a
Linux raspberrypi 3.18.0-trunk-rpi2 #1 SMP PREEMPT Debian 3.18.5-1-exp1+rpi19 (2015-08-08) armv7l GNU/Linux
```

그림 1 최신 Kernel이 설치된 Device

이후, Bluetooth Stack을 디바이스에 설치한다. 그리고 Kernel에서 이를 이용할 수 있도록 6LoWPAN Module을 활성화시키고, Bluetooth에서도 6LoWPAN을 사용할 수 있도록 Module을 활성화 시킨다.

```
root@raspberrypi:/home/pi# lsmod
Module                Size  Used by
ntfs                   221038  0
sg                     22488  0
sd_mod                36763  0
usb_storage            42573  0
scsi_mod              184774  3 sg,usb_storage,sd_mod
rfcomm                 51947  10
bnep                   12598  2
cfg80211              401220  0
bluetooth_6lowpan     14875  1
6lowpan               10256  1 bluetooth_6lowpan
snd_bcm2835           18872  0
snd_pcm                75718  1 snd_bcm2835
snd_seq                55632  0
snd_seq_device         5469  1 snd_seq
snd_timer              17746  2 snd_pcm,snd_seq
snd                    52451  5 snd_bcm2835,snd_timer,snd_pcm,snd_seq,snd_seq_device
soundcore              5387  1 snd
ecb                    2025  1
btusb                  23193  0
bluetooth              375475  25 bnep,btusb,rfcomm,bluetooth_6lowpan
rfkill                 16473  3 cfg80211,bluetooth
cdc_acm                21181  0
evdev                  9834  3
joydev                 8752  0
```

그림 2 Module 활성화

Gateway 디바이스에서 Bluetooth Advertisement를 활성화 시켜, 다른 6LoWPAN 디바이스가 Gateway에 연결될 수 있도록 한다.

```
root@raspberrypi:/home/pi# hciconfig hci0 leadv
LE set advertise enable on hci0 returned status 12
```

그림 3 Bluetooth Advertisement를 활성화

6LoWPAN 디바이스에서 Bluetooth scan을 하고, 연결가능한 한 Gateway를 찾아 연결을 한다. 연결이 성공되면 6LoWPAN이 활성화 된 Bluetooth 디바이스가 생성된다.

```
root@raspberrypi:/home/pi# ifconfig
bt0    Link encap:UNSPEC  HWaddr 00-1A-7D-FF-FE-DA-71-0C-00-00-00-00-00-00-00-00
       inet6 addr: 2001:db8:abcd:2::2/64 Scope:Global
       inet6 addr: fe80::21a:7dff:feda:710c/64 Scope:Link
       UP POINTOPOINT RUNNING MULTICAST  MTU:1280  Metric:1
       RX packets:28659 errors:0 dropped:0 overruns:0 frame:0
       TX packets:52132 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:0
       RX bytes:2382746 (2.2 MiB)  TX bytes:3669905 (3.4 MiB)
```

그림 4 Gateway와 연결되고 6LoWPAN이 활성화 된 디바이스

6LoWPAN Gateway 디바이스와 통신이 정상적으로 이루어지는지 확인하기 위해, Gateway의 Link-local address를 이용하여 ping 테스트를 실행한다.

```
root@raspberrypi:/home/pi# ping6 fe80::21a:7dff:feda:7104%bt0
PING fe80::21a:7dff:feda:7104%bt0(fe80::21a:7dff:feda:7104) 56 data bytes
64 bytes from fe80::21a:7dff:feda:7104: icmp_seq=1 ttl=64 time=143 ms
64 bytes from fe80::21a:7dff:feda:7104: icmp_seq=2 ttl=64 time=186 ms
64 bytes from fe80::21a:7dff:feda:7104: icmp_seq=3 ttl=64 time=102 ms
^C
--- fe80::21a:7dff:feda:7104%bt0 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 102.395/144.266/186.664/34.407 ms
```

그림 5 Ping 테스트를 통한 정상동작 확인

## 2.2 6LoWPAN을 활용한 Open IoT 플랫폼 서버와 통신환경 구축

6LoWPAN 디바이스와 Open IoT 플랫폼 서버간 통신이 가능하도록 IPv6 주소를 할당하고 Routing을 설정한다.

```
root@mobiuss:/home/mobius# ifconfig
eth0   Link encap:Ethernet  HWaddr 1c:b7:2c:ae:d8:7a
       inet addr:155.230.105.165  Bcast:155.230.105.255  Mask:255.255.255.0
       inet6 addr: 2001:db8:abcd:1::2/64 Scope:Global
       inet6 addr: fe80::1eb7:2cff:feae:d87a/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:107332965 errors:0 dropped:1338142 overruns:0 frame:0
       TX packets:1335342 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:10940625402 (10.9 GB)  TX bytes:332808770 (332.8 MB)
       Interrupt:20 Memory:dfc00000-dfc20000
```

그림 6 Open IoT 플랫폼 서버에 IPv6 주소 할당

6LoWPAN 디바이스에서 플랫폼 서버와 통신이 가능하도록 Routing을 설정한다.

```

root@raspberrypi:/home/pi# route -6
Kernel IPv6 routing table
Destination          Next Hop              Flag Met Ref Use If
2001:db8:abdc:2::/64  ::                    U    256 0   1 bt0
fe80::/64            ::                    U    256 0   0 bt0
fe80::/64            ::                    U    256 0   0 eth0
::/0                 2001:db8:abdc:2::1  UG   1024 0   0 bt0
::/0                 ::                    !n   -1  1 28608 lo
::1/128              ::                    Un   0   1   22 lo
2001:db8:abdc:2::2/128  ::                    Un   0   1 5114 lo
fe80::21a:7dff:feda:710c/128  ::                    Un   0   1   6 lo
fe80::6f6a:c9ce:d2ed:8c11/128  ::                    Un   0   1   0 lo
ff00::/8             ::                    U    256 0   0 eth0
ff00::/8             ::                    U    256 0   0 bt0
::/0                 ::                    !n   -1  1 28608 lo

```

그림 7 Open IoT Platform 서버와 통신을 위한 라우팅 설정 (Device - Gateway)

6LoWPAN 디바이스에서 전달된 패킷이 플랫폼 서버로 전달되도록 Routing 설정을 한다.

```

root@raspberrypi:/home/pi# route -6
Kernel IPv6 routing table
Destination          Next Hop              Flag Met Ref Use If
2001:db8:abdc:1::/64  ::                    U    256 1   0 eth0
2001:db8:abdc:2::/64  ::                    U    256 0   0 bt0
fe80::/64            ::                    U    256 0   0 eth0
fe80::/64            ::                    U    256 0   0 bt0
::/0                 ::                    !n   -1  1 98072 lo
::1/128              ::                    Un   0   1 56320 lo
2001:db8:abdc:1::/128  ::                    Un   0   1   0 lo
2001:db8:abdc:1::1/128  ::                    Un   0   2   115 lo
2001:db8:abdc:2::/128  ::                    Un   0   1   0 lo
2001:db8:abdc:2::1/128  ::                    Un   0   1   0 lo
fe80::/128           ::                    Un   0   1   0 lo
fe80::/128           ::                    Un   0   1   0 lo
fe80::21a:7dff:feda:7104/128  ::                    Un   0   1   3 lo
fe80::ba27:ebff:fe75:48b9/128  ::                    Un   0   1 35310 lo
ff00::/8             ::                    U    256 1   0 eth0
ff00::/8             ::                    U    256 0   0 bt0
::/0                 ::                    !n   -1  1 98072 lo

```

그림 8 Open IoT 플랫폼 서버와 통신을 위한 라우팅 설정 (Gateway - Server)

6LoWPAN 디바이스에서 Open IoT 플랫폼 서버와 ping 테스트를 하여, 통신이 정상적으로 잘 이루어지는지 확인한다.

```

root@raspberrypi:/home/pi# ping6 2001:db8:abdc:1::2
PING 2001:db8:abdc:1::2(2001:db8:abdc:1::2) 56 data bytes
64 bytes from 2001:db8:abdc:1::2: icmp_seq=1 ttl=63 time=84.4 ms
64 bytes from 2001:db8:abdc:1::2: icmp_seq=2 ttl=63 time=139 ms
64 bytes from 2001:db8:abdc:1::2: icmp_seq=3 ttl=63 time=111 ms
^C
--- 2001:db8:abdc:1::2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 84.465/111.740/139.355/22.413 ms

```

그림 9 6LoWPAN 디바이스와 Open IoT 플랫폼 서버간의 ping 테스트 수행

### 2.3 &Cube 디바이스 플랫폼 및 TAS 설정

Open IoT 플랫폼 서버와 연동하기 위해 &Cube 디바이스 플랫폼을 6LoWPAN 디바이스에서 동작하도록 수정한다.

```

root@raspberrypi:/nCube# cat reg.conf
CSEid=0.2.481.1.0001.001.7594
CSEpasswd=1234
CSEName=0.2.481.1.0001.001.7594
CSEPointOfAccess=MQTT|0.2.481.1.0001.001.7594
requestReachability=true
firmwareName=nCube_Lavender
firmwareDescription=nCube1.0_Test_version
firmwareVersion=1.0
firmwareURL=http://www.keti.re.kr
firmwareStatus=1
deviceName=KETI_ASN_Device
deviceLabel=KETI_ASN_Device
deviceDescription=KETI's ASN Device
deviceManufacturer=KETI
deviceModel=KETI001
deviceType=Sensor device
deviceFwVersion=1.0
deviceSwVersion=1.0
deviceHwVersion=1.0
INCSEAddress=[2001:db8:abdc:1::2]:9000
MQTTBrokerAddress=2001:db8:abdc:1::2
interopType=0
primitiveType=0
bodyType=1
protocolBinding=1
debugPrint=1

```

그림 10 &cube 플랫폼 동작에 필요한 설정파일

&Cube 플랫폼을 실행하고 Open IoT 플랫폼 서버에서 이를 인식하는지 확인한다.

```
[ResourceManager] Return to mgmtOnd Result
[ResourceManager] Receive mgmtOnd Registration request
[ResourceManager] CSE Profile get... OK
[DeviceManager] Receive from Resource Manager - MgmtOnd regist request
[ResourceManager] Send to Interaction Manager - MgmtOnd Registration

[DeviceManager] Received labels : deviceManagement
[InteractionManager] MgmtOnd Registration start...
[DeviceManager] Received body : MC00000000000000000011
HTTP Response Code : 403
HTTP Response String : <?xml version="1.0" encoding="UTF-8" standalone="yes"?><m2m:mgc xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><rtyp>12</rtyp><ri>MC00000000000000000012</ri><rm:firmwareUpgrade/><pi>RC0000000000000000000006</pi><ct>2015-12-09T15:53:59+09:00</ct><lt>2015-12-09T15:53:59+09:00</lt><lbl></lbl><cm>90000002</cm><exe>true</exe><ext>0.2.481.1.0001.001.7594</ext></m2m:mgc>
[InteractionManager] MgmtOnd Registration Success
[InteractionManager] Send to Resource Manager - mgmtOnd Registration Result

[ResourceManager] Receive from Interaction Manager - MgmtOnd Registration Result
[ResourceManager] MgmtOnd set... OK
[ResourceManager] mgmtOnd Registration... OK
[ResourceManager] Return to mgmtOnd Result

[ResourceManager] Receive mgmtOnd Registration request
[ResourceManager] CSE Profile get... OK
[DeviceManager] Receive from Resource Manager - MgmtOnd regist request
[DeviceManager] Received labels : firmwareUpgrade
[DeviceManager] Received body : MC00000000000000000012
[ResourceManager] Send to Interaction Manager - MgmtOnd Registration

[InteractionManager] MgmtOnd Registration start...
HTTP Response Code : 403
HTTP Response String : <?xml version="1.0" encoding="UTF-8" standalone="yes"?><m2m:mgc xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><rtyp>12</rtyp><ri>MC00000000000000000013</ri><rm:appInstall/><pi>RC00000000000000000006</pi><ct>2015-12-09T15:53:59+09:00</ct><lt>2015-12-09T15:53:59+09:00</lt><lbl></lbl><cm>90000003</cm><exe>true</exe><ext>0.2.481.1.0001.001.7594</ext></m2m:mgc>
[InteractionManager] MgmtOnd Registration Success
[InteractionManager] Send to Resource Manager - mgmtOnd Registration Result

[ResourceManager] Receive from Interaction Manager - MgmtOnd Registration Result
[ResourceManager] MgmtOnd set... OK
[ResourceManager] mgmtOnd Registration... OK
[ResourceManager] Return to mgmtOnd Result

[DeviceManager] Receive from Resource Manager - MgmtOnd regist request
[DeviceManager] Received labels : appInstall
[DeviceManager] Received body : MC00000000000000000013
```

그림 11 실행된 &cube 플랫폼

```
{
  "_id" : ObjectId("5667d006e4b0e2be541e49ad"),
  "_class" : "kr.re.keti.iot.domain.oneM2M.Node",
  "nodeID" : "0.2.481.1.0001.001.7594",
  "hostedCSELINK" : "RC00000000000000000006",
  "announceTo" : [],
  "announcedAttribute" : [],
  "accessControlPolicyIDs" : [
    "AP00000000000000000006"
  ],
  "resourceType" : "14",
  "resourceID" : "ND00000000000000000006",
  "resourceName" : "0.2.481.1.0001.001.7594",
  "parentID" : "Mobius",
  "creationTime" : "2015-12-09T15:53:58+09:00",
  "lastModifiedTime" : "2015-12-09T15:53:58+09:00",
  "labels" : [],
  "resourceRef" : {
    "resourceID" : "ND00000000000000000006",
    "value" : "http://155.230.105.165:9000/Mobius/node-0.2.481.1.0001.001.7594",
    "resourceName" : "0.2.481.1.0001.001.7594",
    "type" : "14"
  }
}
```

그림 12 Open IoT 플랫폼 서버에서 인식된 &cube 플랫폼

IoT 장비를 제어하고, 측정값을 받아오기 위한 Things Adaptation S/W (TAS)를 개발한다. TAS는 6LoWPAN 디바이스에서 실행된 &Cube와 호환이 가능하도록 개발한다.

```
root@raspberrypi:/nCube# java -jar godapu_TAS.jar
BlueCove version 2.1.0 on bluez
Sat Dec 12 21:24:31 KST 2015 => Info: A TAS server for 6CUBE is running now!
Dec 12, 2015 9:24:31 PM com.keti.tas.soft.ThingTasConnector start
INFO: A TAS server for Thing is running.
Sat Dec 12 21:24:31 KST 2015 => Client send (requestTASRegistration:<TAS><TASProfile><Pcc>S4961</Pcc><Name>TAS_SensorTest</Name><Description>Sample TAS for sensor test</Description></TASProfile></TAS>)
to 6CUBE!
Sat Dec 12 21:24:31 KST 2015 => Receive a message (registSuccess:ThingAdaptationSoftware) from 6CUBE
Dec 12, 2015 9:24:31 PM kr.pe.vanilet.DataAdapter main
INFO: The thing "HR" added.
Dec 12, 2015 9:24:31 PM kr.pe.vanilet.DataAdapter main
INFO: The thing "STRESS" added.
Sat Dec 12 21:24:33 KST 2015 => Client send (requestThingRegistration:<TAS><ThingProfile><name>#k</name><containerType>sensoractor</containerType><uploadCondition>nothings</uploadCondition><uploadConditionValue>nothings</uploadConditionValue></ThingProfile></TAS>) to 6CUBE!
Sat Dec 12 21:24:34 KST 2015 => Receive a message (registSuccess:#k,null) from 6CUBE
Sat Dec 12 21:24:34 KST 2015 => Client send (requestThingRegistration:<TAS><ThingProfile><name>STRESS</name><containerType>sensoractor</containerType><uploadCondition>nothings</uploadCondition><uploadConditionValue>nothings</uploadConditionValue></ThingProfile></TAS>) to 6CUBE!
Sat Dec 12 21:24:34 KST 2015 => Receive a message (registSuccess:STRESS,null) from 6CUBE
Device Inquiry completed!
wait for device inquiry to complete...
0 devices (s) found!!
Dec 12, 2015 9:24:42 PM com.keti.tas.soft.ThingTasConnector run
SEVERE: start scan!!
```

그림 13 TAS 실행

```
[InteractionManager] Thing Registration start...
HTTP Response Code : 403
HTTP Response String : <?xml version="1.0" encoding="UTF-8" standalone="yes"?><m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><rtty>3</rtty>
<rtt>CT00000000000000000001</rtt><rm>#R</rm><pi>RC00000000000000000000</pi><ct>2015-12-09T18:05:21+09:00</ct><lt>2015-12-10T12:13:40+09:00</lt><lbl></lbl><at></at><aa></aa><st>23</st><cr>RC00000000000000000000000000000000</cr><cn1>23</cn1><cb>6840</cb></m2m:cnt>
[InteractionManager] Container Registration Success
[InteractionManager] Send to Resource Manager - Container Registration Result

[ResourceManager] Receive from Interaction Manager - Thing Registration Result
[ResourceManager] Thing Profile set... OK
[ResourceManager] Thing Registration... OK
[ResourceManager] Return to Thing Registration Result

[ThingManager] Receive to Resource Manager - Response Thing Registration
[ThingManager] Send to Thing Adaptation Software - Thing Registration Result

[ThingManager] Receive to Thing Adaptation Software - Thing Registration request
[ThingManager] Send to Resource Manager - Request Thing Registration

[ResourceManager] Receive Thing Registration request
[ResourceManager] CSE Profile get... OK
[ResourceManager] Send to Interaction Manager - Thing Registration

[InteractionManager] Thing Registration start...
HTTP Response Code : 403
HTTP Response String : <?xml version="1.0" encoding="UTF-8" standalone="yes"?><m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><rtty>3</rtty>
<rtt>CT00000000000000000001</rtt><rm>#R</rm><pi>RC00000000000000000000</pi><ct>2015-12-09T18:05:22+09:00</ct><lt>2015-12-10T12:13:40+09:00</lt><lbl></lbl><at></at><aa></aa><st>23</st><cr>RC00000000000000000000000000000000</cr><cn1>23</cn1><cb>6898</cb></m2m:cnt>
[InteractionManager] Container Registration Success
[InteractionManager] Send to Resource Manager - Container Registration Result

[ResourceManager] Receive from Interaction Manager - Thing Registration Result
[ResourceManager] Thing Profile set... OK
[ResourceManager] Thing Registration... OK
[ResourceManager] Return to Thing Registration Result

[ThingManager] Receive to Resource Manager - Response Thing Registration
[ThingManager] Send to Thing Adaptation Software - Thing Registration Result
```

그림 14 &cube에서 TAS를 인식

### 3. 결론

지금까지 사물인터넷 플랫폼인 Open IoT 플랫폼과 &Cube 플랫폼 환경에서의 6LoWPAN을 활용한 Bluetooth LE 통신기법에 대하여 설명하였다. 본 문서에서는 사물인터넷 환경에서 기존의 M2M 통신 기술인 Bluetooth 통신 기법을 인터넷 프로토콜인 6LoWPAN 기술과 결합하여 새로운 사물인터넷 환경의 통신 테스트베드를 구축하여 기존의 M2M 통신 기술이 사물인터넷에 어떻게 활용이 될 수 있는지 실험해 보았다. Bluetooth LE의 저전력 통신과 6LoWPAN의 IP 통신기술을 활용하여 사물인터넷 환경에서 효율적인 통신 기술은 앞으로 다른 사물인터넷 연구에도 큰 도움이 될 것으로 생각한다.



## 참고 문헌

- [1] 사물인터넷 Alliance Ocean 홈페이지, <http://www.iotocean.org/main/>