

## Raspberry Pi 3와 Arduino 101간에 Arduino 101 BLE (Bluetooth Low Energy) 제어 및 정보 가져오기 ( python 라이브러리 사용 )

2016.12.06 07:36

#Arduino 101 #BLE #bluepy #bluetooth #pybluez #Python #raspberry pi 3

이번 포스팅에서는 파이썬 라이브러리인 pyBluez를 Raspberry Pi 3에 설치하여 Arduino 101 BLE (Bluetooth Low Energy) 제어 및 정보 가져오는 과정을 설명합니다.

C언어하는 과정을 다음 포스팅에 소개하고 있습니다.

[Arduino/Arduino 101 ( Genuino 101 )] - Raspberry Pi 3에서 Arduino 101 BLE (Bluetooth Low Energy) 제어 및 정보 가져오기 ( C언어 라이브러리 사용 )

우선 Arduino 101에 블루투스를 이용하여 LED를 제어하는 예제 프로그램이 업로드 되어 있어야 하는데 그 과정은 다음 두 개의 포스팅에 소개하고 있습니다.

먼저 아래 포스팅은 아두이노 IDE에서 Arduino 101를 대상으로 프로그래밍 및 업로드를 하기 위한 준비과정을 설명하고 있습니다.

[Arduino & AVR/Android 101 ( Genuino 101 )] - Genuino 101 ( Arduino 101) 처음 사용해보기

그리고 다음 포스팅은 블루투스 예제를 Arduino 101에 업로드하고 안드로이드에서 Arduino 101에 연결된 LED를 블루투스 통신을 통해 제어하는 방법을 소개하고 있습니다.

[Arduino & AVR/Android 101 ( Genuino 101 )] - Genuino 101 ( Arduino 101) 블루투스 예제 실행시켜보기

파이썬 라이브러리를 사용하여 진행하기 전에 터미널에서 hcitool과 gatttool을 이용하여 통신 접속 및 데이터 전송하는 과정을 다시 한번 확인해보았습니다.

라즈베리파이3에 내장된 블루투스 디바이스를 확인합니다.

```
pi@pi-desktop:~$ hcitool dev
Devices:
    hci0 B8:27:EB:DA:FA:66
```

arduino 101의 블루투스 장치의 맥주소를 검색을 통해 알아냅니다.

```
pi@pi-desktop:~$ sudo hcitool lescan
LE Scan ...
FC:F1:36:2B:34:32 (unknown)
98:4F:EE:0F:4E:1E LED
98:4F:EE:0F:4E:1E (unknown)
^Cpi@pi-desktop:~$
```

그 다음 gatttool을 사용하여 arduino 101의 블루투스에 연결하여 아두이노 101에서 제공하는 블루투스 서비스 목록과 characteristics를 확인해봤습니다.

```
pi@pi-desktop:~$ sudo gatttool -i hci0 -b 98:4F:EE:0F:4E:1E -I
[98:4F:EE:0F:4E:1E][LE]> connect
Attempting to connect to 98:4F:EE:0F:4E:1E
Connection successful
[98:4F:EE:0F:4E:1E][LE]> primary
att handle: 0x0001, end grp handle: 0x0007 uuid: 00001800-0000-1000-8000-00805f9b3
```



```
attr handle: 0x0008, end grp handle: 0x0008 uuid: 00001801-0000-1000-8000-00805f9b34fb
```

```
attr handle: 0x0009, end grp handle: 0xffff uuid: 19b10000-e8f2-537e-4f6c-d104768a1214
```

```
[98:4F:EE:0F:4E:1E][LE]> char-desc
```

```
handle: 0x0001, uuid: 00002800-0000-1000-8000-00805f9b34fb
```

```
handle: 0x0002, uuid: 00002803-0000-1000-8000-00805f9b34fb
```

```
handle: 0x0003, uuid: 00002a00-0000-1000-8000-00805f9b34fb
```

```
handle: 0x0004, uuid: 00002803-0000-1000-8000-00805f9b34fb
```

```
handle: 0x0005, uuid: 00002a01-0000-1000-8000-00805f9b34fb
```

```
handle: 0x0006, uuid: 00002803-0000-1000-8000-00805f9b34fb
```

```
handle: 0x0007, uuid: 00002a04-0000-1000-8000-00805f9b34fb
```

```
handle: 0x0008, uuid: 00002800-0000-1000-8000-00805f9b34fb
```

```
handle: 0x0009, uuid: 00002800-0000-1000-8000-00805f9b34fb
```

```
handle: 0x000a, uuid: 00002803-0000-1000-8000-00805f9b34fb
```

```
handle: 0x000b, uuid: 19b10001-e8f2-537e-4f6c-d104768a1214
```

arduino 101에 업로드 시켜둔 LED 예제코드를 열어보면 아래와 같이 선언되어 있는 것이 보입니다.

```
BLEService ledService("19B10000-E8F2-537E-4F6C-D104768A1214"); // BLE LED Service
```

```
// BLE LED Switch Characteristic - custom 128-bit UUID, read and writable by central
BLEUnsignedCharCharacteristic switchCharacteristic("19B10001-E8F2-537E-4F6C-D104768A1214", BLERead | BLEWrite);
```

이제 확인한 characteristics의 핸들을 가지고 arduino 101 의 LED를 제어해봅니다.

첫번째 줄을 실행시키면 LED가 켜지고, 두번째 줄을 실행시키면 LED가 꺼지게 됩니다.

```
[98:4F:EE:0F:4E:1E][LE]> char-write-req b 01
```

```
Characteristic value was written successfully
```

```
[98:4F:EE:0F:4E:1E][LE]> char-write-req b 00
```

```
Characteristic value was written successfully
```



## pybluez 라이브러 테스트

이제 Raspberry Pi 3에 pybluez 라이브러리를 다음 과정을 통해 설치했습니다.

```
pi@pi-desktop:~$ sudo apt-get install bluetooth libbluetooth-dev
```

```
pi@pi-desktop:~$ sudo pip install pybluez
```

<https://github.com/karulis/pybluez/tree/master/examples/ble> 에 존재하는 BLE를 위한 pybluez예제를 테스트 해보았습니다.

우선 scan.py를 실행시켜 보니 다음과 같은 에러가 나서

```
pi@pi-desktop:~/iBeacon-Scanner$ python scan.py
Traceback (most recent call last):
  File "scan.py", line 1, in <module>
    from bluetooth.ble import DiscoveryService
  File "/usr/local/lib/python2.7/dist-packages/bluetooth/ble.py", line 1, in <module>
    from gattlib import *
ImportError: No module named gattlib
```

gattlib를 설치하려고 하니 boost 관련 헤더파일이 없다는 에러가 나서 먼저 libboost-all-dev를 설치한 후 gattlib를 설치했습니다.

```
pi@pi-desktop:~/iBeacon-Scanner$ sudo apt-get install libboost-all-dev
```

```
pi@pi-desktop:~/iBeacon-Scanner$ sudo pip install gattlib
```

이제 다시 실행을 시켜본 결과 입니다.

```
pi@pi-desktop:~/iBeacon-Scanner$ sudo python scan.py
```

```
name: , address: 98:4F:EE:0F:4E:1E
```



hcitool 명령 결과와 비교해보니 98:4F:EE:0F:4E:1E이 arduino 101의 블루투스 맥어드레스이군요..

```
pi@pi-desktop:~/iBeacon-Scanner-$ sudo hcitool -i hci0 lescan
Set scan parameters failed: Input/output error
pi@pi-desktop:~/iBeacon-Scanner-$ sudo hciconfig hci0 reset
pi@pi-desktop:~/iBeacon-Scanner-$ sudo hcitool -i hci0 lescan
LE Scan ...
FC:F1:36:2B:34:32 (unknown)
98:4F:EE:0F:4E:1E LED
98:4F:EE:0F:4E:1E (unknown)
^Cpi@pi-desktop:~/iBeacon-Scanner-$
```

scan.py의 코드는 아래와 같습니다.

```
1 from bluetooth.ble import DiscoveryService
2
3 service = DiscoveryService()
4 devices = service.discover(2)
5
6 for address, name in devices.items():
7     print("name: {}, address: {}".format(name, address))
```

cs

패키지로 설치하는 경우 위 코드에서 첫번째 줄 에러가 나는군요.. 그럴 경우 아래 처럼 따라하시면 됩니다.

```
문제가 있다고 댓글에 알려주셔서 확인해보니 에러가 나는군요..
webnautes@webnautes-W54-55SU1-SUW ~ $ python
Python 2.7.12 (default, Jul 1 2016, 15:12:24)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from bluetooth.ble import DiscoveryService
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named ble
```



github에는 ble.py가 존재하는데 패키지로 설치시 없는 것으로 미루어 소스로 설치하면 문제가 해결될 거 같았습니다.

우선 설치했던 패키지를 지우고

```
$ sudo pip uninstall pybluez
```

github에서 소스코드를 다운로드 받아 설치를 합니다.

```
$ git clone https://github.com/karulis/pybluez.git pybluez
```

```
$ cd pybluez
```

```
$ sudo python setup.py install
```

```
$ sudo pip install -e .[ble]
```

문제가 해결되었습니다..

```
webnautes@webnautes-W54-55SU1-SUW ~/pybluez $ python
```

```
Python 2.7.12 (default, Jul 1 2016, 15:12:24)
```

```
[GCC 5.4.0 20160609] on linux2
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> from bluetooth.ble import DiscoveryService
```

이제 pybluez 라이브러리를 이용하여 arduino 101에 연결된 LED를 제어해보려했지만 ble 관련 예제를 모아둔 폴더에 예제코드도 없었고 따로 검색해봤지만 마땅한 코드를 찾지 못했습니다.

pybluez에서 아직 BLE를 완벽하게 지원하지 않으니 대안으로 사용할 라이브러리를 수소문한 결과 bluepy를 찾을 수 있었습니다...

## Bluepy 라이브러리 테스트



우선 라이브러리를 설치하고

```
pi@pi-desktop:~$ sudo pip install bluepy
```

그리고 나서 예제코드를 다운로드 받았습니다. bluepy 홈페이지에는 예제코드는 거의 없고 라이브러리 레퍼런스 메뉴얼만 있어서 따로 찾은 예제들입니다.

```
pi@pi-desktop:~$ git clone https://github.com/rlangoy/bluepy_examples_nRF51822_mbed.git
```

테스트 해보면서 일부 수정한 경우에는 코드도 같이 올려둡니다.

## blesca.py

블루투스 다바이스 이름과 RSSI, Flags, 서비스 uuid,, 이름을 출력해주는 예제입니다.

주소타입이 public으로 검색된 것을 확인할 수 있는데 이후 테스트한 예제들은 모두 random으로 코드에 적혀있어서 제대로 동작을 안했습니다. public로 변경해야 제대로 동작하더군요..

```
1 pi@pi-desktop:~/bluepy_examples_nRF51822_mbed$ sudo python blesc
2 [sudo] password for pi:
3 Device 98:4f:ee:0f:4e:1e (public), RSSI=-41 dB
4   Flags = 06
5   Incomplete 128b Services = 14128a7604d16c4f7e53f2e80000b119
6   Complete Local Name = LED
7 Device fc:f1:36:2b:34:32 (public), RSSI=-95 dB
8   Short Local Name = [TV] UN65J6350
9   Flags = 00
10  Manufacturer = 7500420401010efcf1362b34323a0195d03e2a01000000c
```

## getServices.py

지정한 블루투스 장치의 서비스를 출력해줍니다.

```
1 pi@pi-desktop:~/bluepy_examples_nRF51822_mbed$ sudo python getSer
2 EE:0F:4E:1E
3 [sudo] password for pi:
```

```

4 Service <uuid=Generic Attribute handleStart=8 handleEnd=8>
5 Service <uuid=Generic Access handleStart=1 handleEnd=7>
6 Service <uuid=19b10000-e8f2-537e-4f6c-d104768a1214 handleStart=9

```

```

1 import sys
2 from bluepy.btlib import UUID, Peripheral
3
4 if len(sys.argv) != 2:
5     print "Fatal, must pass device address:", sys.argv[0], "<device address>"
6     quit()
7
8 p = Peripheral(sys.argv[1], "public")
9
10 services=p.getServices()
11
12 #displays all services
13 for service in services:
14     print service

```

Co.

## getDeviceCharacteristics.py

지정한 블루투스 장치의 Characteristics를 출력해줍니다.

```

1 pi@pi-desktop:~/bluepy_examples_nRF51822_mbed$ sudo python getDev
2 Handle      UUID
3 -----
4 0x03        00002a00-0000-1000-8000-00805f9b34fb READ
5 0x05        00002a01-0000-1000-8000-00805f9b34fb READ
6 0x07        00002a04-0000-1000-8000-00805f9b34fb READ
7 0x0B        19b10001-e8f2-537e-4f6c-d104768a1214 READ WRITE

```

```

1 import sys
2 from bluepy.btlib import UUID, Peripheral
3
4 if len(sys.argv) != 2:
5     print "Fatal, must pass device address:", sys.argv[0], "<device address>"
6     quit()
7
8 p = Peripheral(sys.argv[1], "public")
9
10 chList = p.getCharacteristics()
11 print "Handle      UUID                               Properties"
12 print "-----"

```



```

13 for ch in chList:
14     print (" 0x"+ format(ch.getHandle(), '02X') + "    "+str(ch.uu

```

## getDeviceName.py

지정한 블루투스 장치의 이름을 출력해줍니다.

```

1 pi@pi-desktop:~/bluepy_examples_nRF51822_mbed$ sudo python getDev
2 ARDUINO 101-4E1E

```

```

1 import sys
2 from bluepy.btle import UUID, Peripheral
3
4 dev_name_uuid = UUID(0x2A00)
5
6 if len(sys.argv) != 2:
7     print "Fatal, must pass device address:", sys.argv[0], "<devic
8     quit()
9
10 p = Peripheral(sys.argv[1], "public")
11
12 try:
13     ch = p.getCharacteristics(uuid=dev_name_uuid)[0]
14     if (ch.supportsRead()):
15         print ch.read()
16
17 finally:
18     p.disconnect()

```

co.

## writeLed2.py

기존 코드를 수정하여 아두이노 101의 LED가 깜빡이는 것을 확인했습니다..

```

1 pi@pi-desktop:~/bluepy_examples_nRF51822_mbed$ sudo python writel
2 Service <uuid=19b10000-e8f2-537e-4f6c-d104768a1214 handleStart=9
3 Led on
4 Led off
5 Led on
6 Led off

```

```

7  Led on
8  Led off

1  import sys
2  import binascii
3  import struct
4  import time
5  from bluepy.btle import UUID, Peripheral
6
7  led_service_uuid = "19b10000-e8f2-537e-4f6c-d104768a1214"
8  led_char_uuid = "19b10001-e8f2-537e-4f6c-d104768a1214"
9
10 if len(sys.argv) != 2:
11     print "Fatal, must pass device address:", sys.argv[0], "<device"
12     quit()
13
14 p = Peripheral(sys.argv[1], "public")
15 LedService=p.getServiceByUUID(led_service_uuid)
16 print LedService
17
18 try:
19     ch = LedService.getCharacteristics(led_char_uuid)[0]
20
21     while 1:
22         ch.write(struct.pack('<B', 0x00));
23         print ("Led on")
24         time.sleep(1)
25         ch.write(struct.pack('<B', 0x01));
26         print ("Led off")
27         time.sleep(1)
28 finally:
29     p.disconnect()

```

co.

## readChar.py

LED 서비스가 제공하는 Characteristics의 값을 읽어오기 위해 작성했습니다.

```

1 pi@raspberrypi:~/bluepy_examples_nRF51822_mbed $ sudo python read
2 Service <uuid=19b10000-e8f2-537e-4f6c-d104768a1214 handleStart=9
3 0x00
4 0x00
5 0x00
6 0x00
7 0x00
8 0x00
9 0x00

```

```
1 import sys
2 import binascii
3 import struct
4 import time
5 from bluepy.btle import UUID, Peripheral
6
7 led_service_uuid = "19b10000-e8f2-537e-4f6c-d104768a1214"
8 led_char_uuid = "19b10001-e8f2-537e-4f6c-d104768a1214"
9
10 if len(sys.argv) != 2:
11     print "Fatal, must pass device address:", sys.argv[0], "<device address>"
12     quit()
13
14 p = Peripheral(sys.argv[1], "public")
15 led_service = p.getServiceByUUID(led_service_uuid)
16 print led_service
17
18 try:
19     ch = led_service.getCharacteristics(led_char_uuid)[0]
20     if ch.supportsRead():
21         while 1:
22             val = binascii.b2a_hex(ch.read())
23             print ("0x" + val)
24             time.sleep(1)
25 finally:
26     p.disconnect()
27
```

Co.

## readButton1Notify.py

아직 테스트 안함...

## readButton1.py

아직 테스트 안함...

bluepy\_examples\_nRF51822\_mbed-master.zip



## 공감

 신고

NAME

PASSWORD

HOMEPAGE

SECRET ☐

WRITE

rblblblbl

2016.06.08 14:24

감사합니다. ble 글 보고 많이 배워갑니다

[URL](#) [Delete](#) [Reply](#)

webnautes

2016.06.08 14:42

감사합니다..~

[URL](#) [Delete](#)

이성준

2016.11.11 22:08

저의 경우는

bluetooth.ble의 ble가 없다고 나온데요... 혹시 ble는 어디서 다운 받는지 아시나요?

[URL](#) [Delete](#) [Reply](#)

webnautes webnautes

2016.11.11 23:46

언제 그런 에러가 나는지 설명좀 해주세요.. 그리고 에러메시지를 댓글에 남겨주세요..

[URL](#) [Delete](#)

이성준

2016.11.14 13:18

pybluez 라이브러 테스트 이 항목 아래에 보면

from bluetooth.ble import DiscoveryService

부분있죠? 여기에서 ImportError: No module named ble라고 오류가 납니다.

[URL](#) [Delete](#)

webnautes webnautes

2016.11.14 13:58

scan.py코드 적어놓은 아래에 해결방법을 추가로 적어놓았습니다.

[URL](#) [Delete](#)

read-hnd

2016.12.06 06:14

내용이 참고가 많이 되었습니다. 감사합니다.

그런데 혹시 char-read-hnd 라던가 char-read-uuid등 읽은 명령어는 어떻게

해야하는지 알려 주실수 있나요 ?

[URL](#) [Delete](#) [Reply](#)

read-hnd

2016.12.06 06:15

파이썬으로서 구현을 묻는 겁니다 ^^

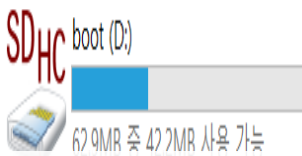
[URL](#) [Delete](#)

webnautes webnautesuuid로 읽어오는건 추가해놓은 readChar.py를 참고하 2016.12.06 07:35  
면 될듯합니다.

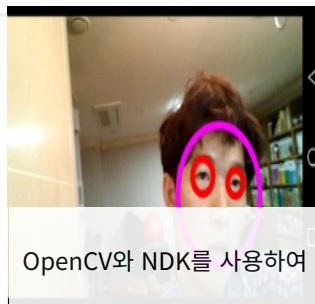
URL Delete

PREV 1 2 3 4 5 6 7 8 9 ... 603 NEXT

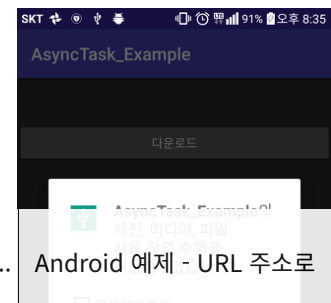
#### + Recent posts



Raspberry Pi 3 처음 사용해..



안드로이드 개념 및 예제 - As..



Rss Feed and Twitter, Facebook, Youtube, Google+

Powered by Tistory, Designed by wallel

