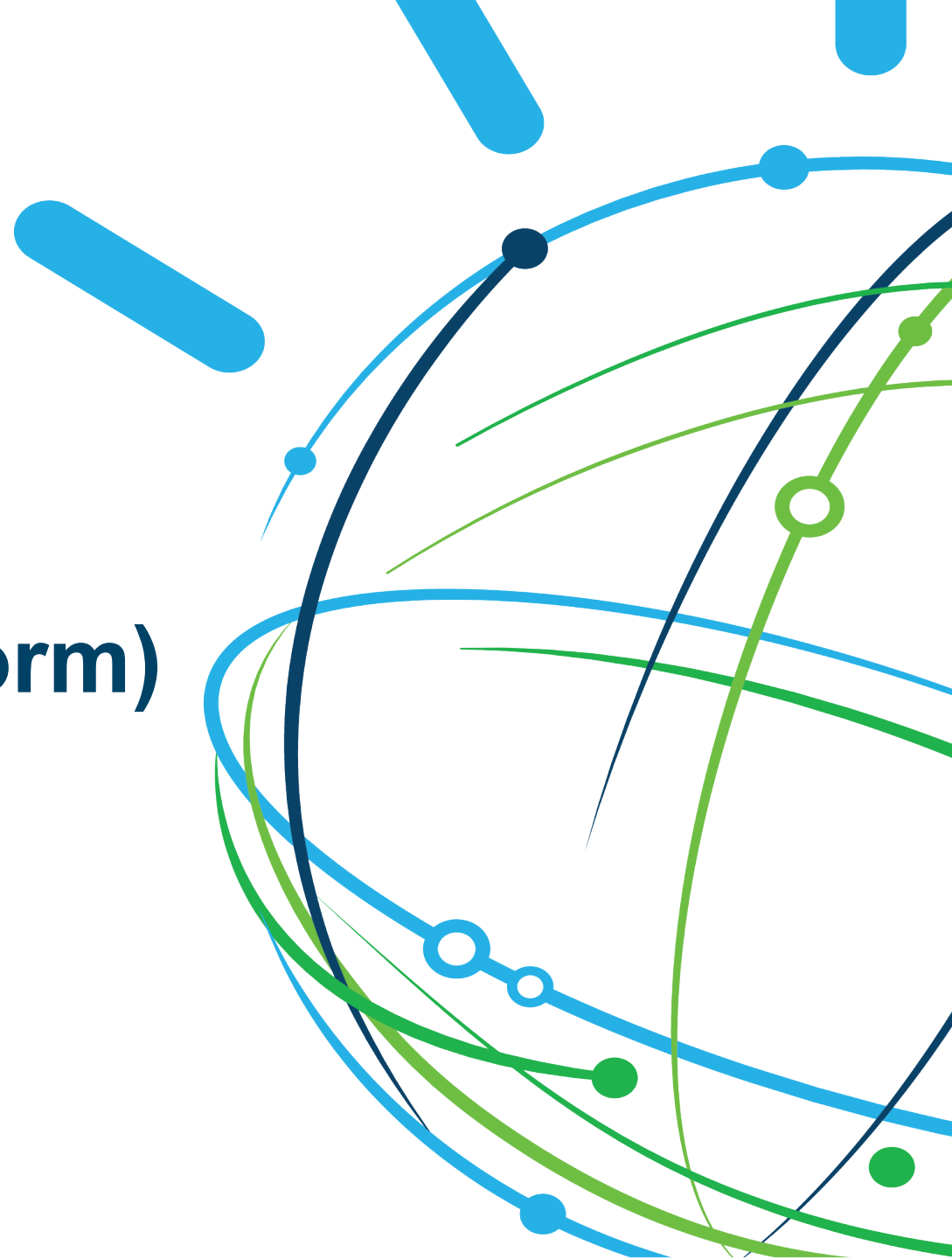


Embedded System

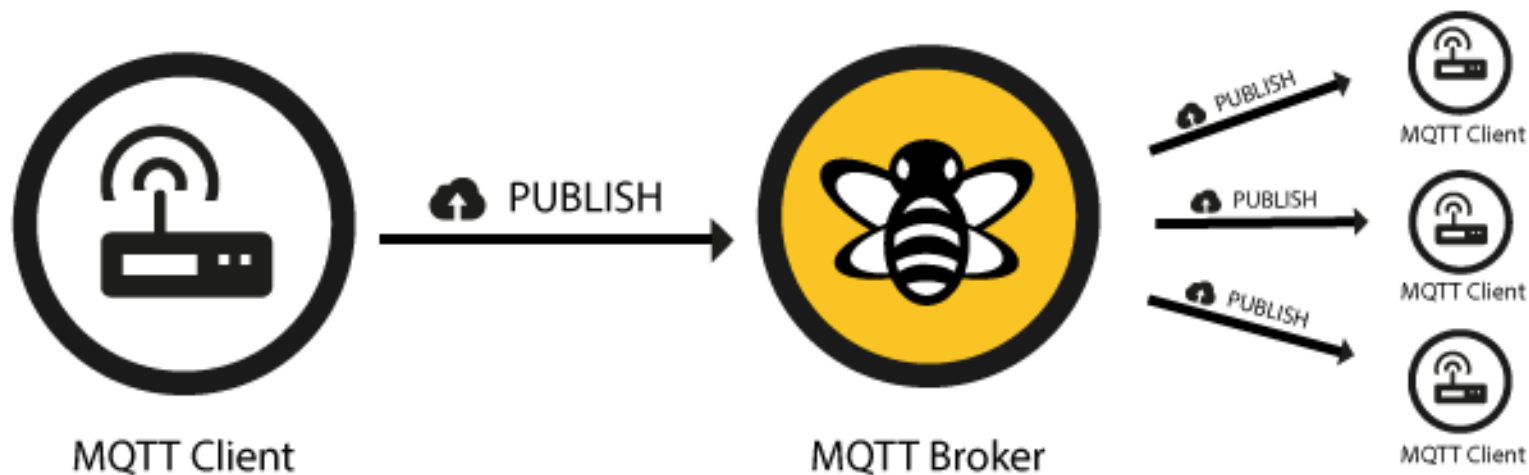
MQTT communication (IoT Device & Server Platform)

2019 2학기



MQTT 소개

- **MQTT (MQ Telemetry Transport)**는 센서장치나 라즈베리파이 같은 임베디드 장치, 모바일 장치 사이의 통신을 위한 가벼운 메시징 프로토콜입니다. TCP/IP 기반으로 대역폭이 작은 네트워크에서 동작할 수 있도록 설계된 프로토콜입니다. 임베디드 장치들을 위한 트위터라고 볼 수 있습니다.
- MQTT 자체는 메시지를 어떻게 보낼 것인지를 정의하는 규약일 뿐입니다. 따라서 실제 **“MQTT 트위터”**를 동작시키기 위해서는 서버 역할을 해주는 장치와 프로그램이 필요한데 이를 MQTT 브로커(Broker)라고 합니다. MQTT 브로커는 각종 장치들(MQTT Client)이 보내주는 메시지를 수집하고 이걸 다시 필요한 장치들에게 전송해주는 중계서버 역할을 합니다.



MQTT 동작 구조

- 트위터 서비스의 동작구조를 떠올려보세요. 트위터에서 사용자는 다른 사용자를 follow 할 수 있습니다. 그럼 다른 사용자가 생성하는 메시지(트윗)를 받아볼 수 있죠. 그리고 스스로 메시지를 생성할 수도 있습니다. 그럼 자신을 follow하는 사용자에게 메시지가 전달되죠.
- MQTT도 거의 유사한 동작구조를 갖습니다. MQTT 시스템에 참여하는 MQTT 클라이언트는 메시지 **발행(publish, 트윗에 해당)**, 메시지 **구독(subscribe, follow에 해당)** 두 가지 동작을 할 수 있습니다. MQTT 클라이언트가 메시지를 특정 **채널(Topic, 토픽)**에 발행하면 이 채널을 구독한 모든 클라이언트에게 메시지가 전달되는 겁니다. 중간에서 메시지를 수집, 재분해 하는 작업은 MQTT 브로커가 담당합니다.

MQTT 동작 구조

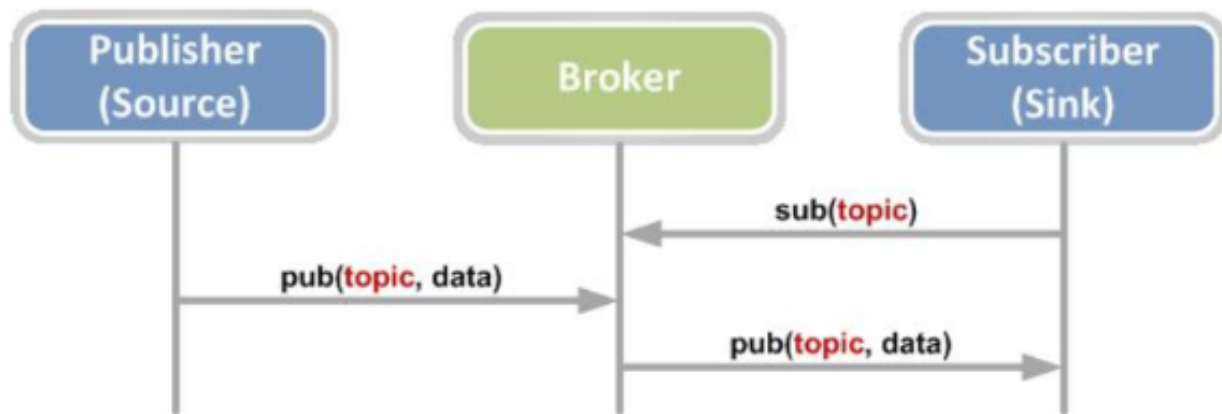


Figure 1: The publish/subscribe communication model

토픽 (Topic)

메시지를 발행/구독(pub/sub) 하는 행위는 채널 단위로 일어납니다. 이를 MQTT에서는 토픽(Topic)이라고 합니다. 토픽은 슬래시(/)로 구분된 계층구조를 갖습니다.

topic level separator
↓
myhome / groundfloor / livingroom / temperature
topic level topic level

개발 환경 구축(IDP - Raspberry pi)

개발 환경 구축

■ 운영체제 update / upgrade

▶ 패키지 업데이트 및 업그레이드

- pi@raspberrypi:~ \$ sudo apt-get update
- pi@raspberrypi:~ \$ sudo apt-get upgrade
- » Do you want to continue? [Y/n] y

% 위 과정을 처음 실행한다면 시간이 꽤 걸림.

개발 환경 구축(IDP - Raspberry pi)

MQTT

■ MQTT – Mosquitto Broker 설치

▶ Mosquitto 프로그램에대한 서명키(인증키) 다운로드

```
$ cd  
$ wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key  
$ sudo apt-key add mosquitto-repo.gpg.key  
OK
```

▶ Mosquitto Repository Package 등록

```
$ cd /etc/apt/sources.list.d/  
$ sudo wget http://repo.mosquitto.org/debian/mosquitto-stretch.list
```

▶ Mosquitto Broker 설치

```
$ sudo apt-get update  
$ sudo apt-cache search mosquitto  
$ sudo apt-get install mosquitto mosquitto-clients
```

▶ Mosquitto 실행

```
$ sudo /etc/init.d/mosquitto start
```

MQTT Broker Test

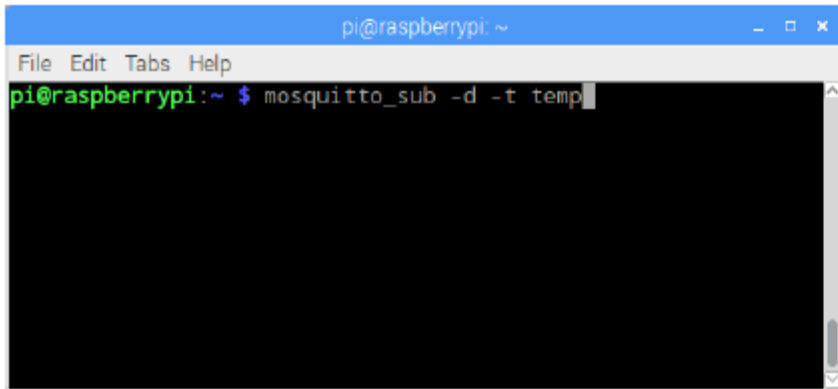
- -d 데몬으로 실행, -h 호스트ip, -t 토픽명, -m 메시지

Topic = temp
Message = 23.5

MQTT

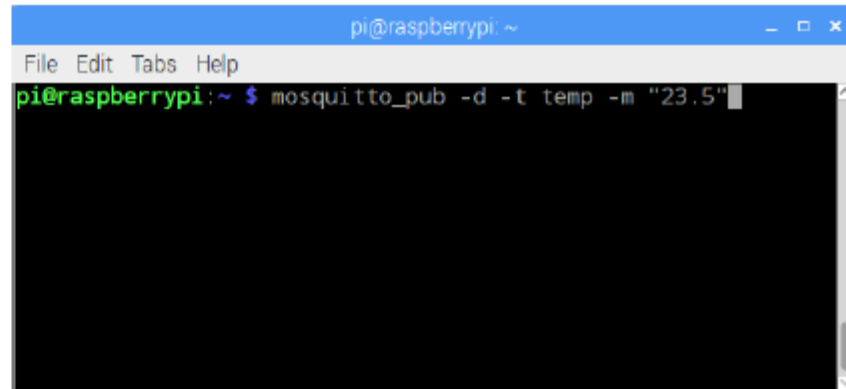
■ MQTT – Mosquitto Broker 동작 확인

```
$ mosquitto_sub -d -t temp
```



A terminal window titled 'pi@raspberrypi: ~' with a menu bar 'File Edit Tabs Help'. The command 'pi@raspberrypi:~ \$ mosquitto_sub -d -t temp' has been entered and executed. The terminal background is black, and the text is green.

```
$ mosquitto_pub -d -t temp -m "23.5"
```



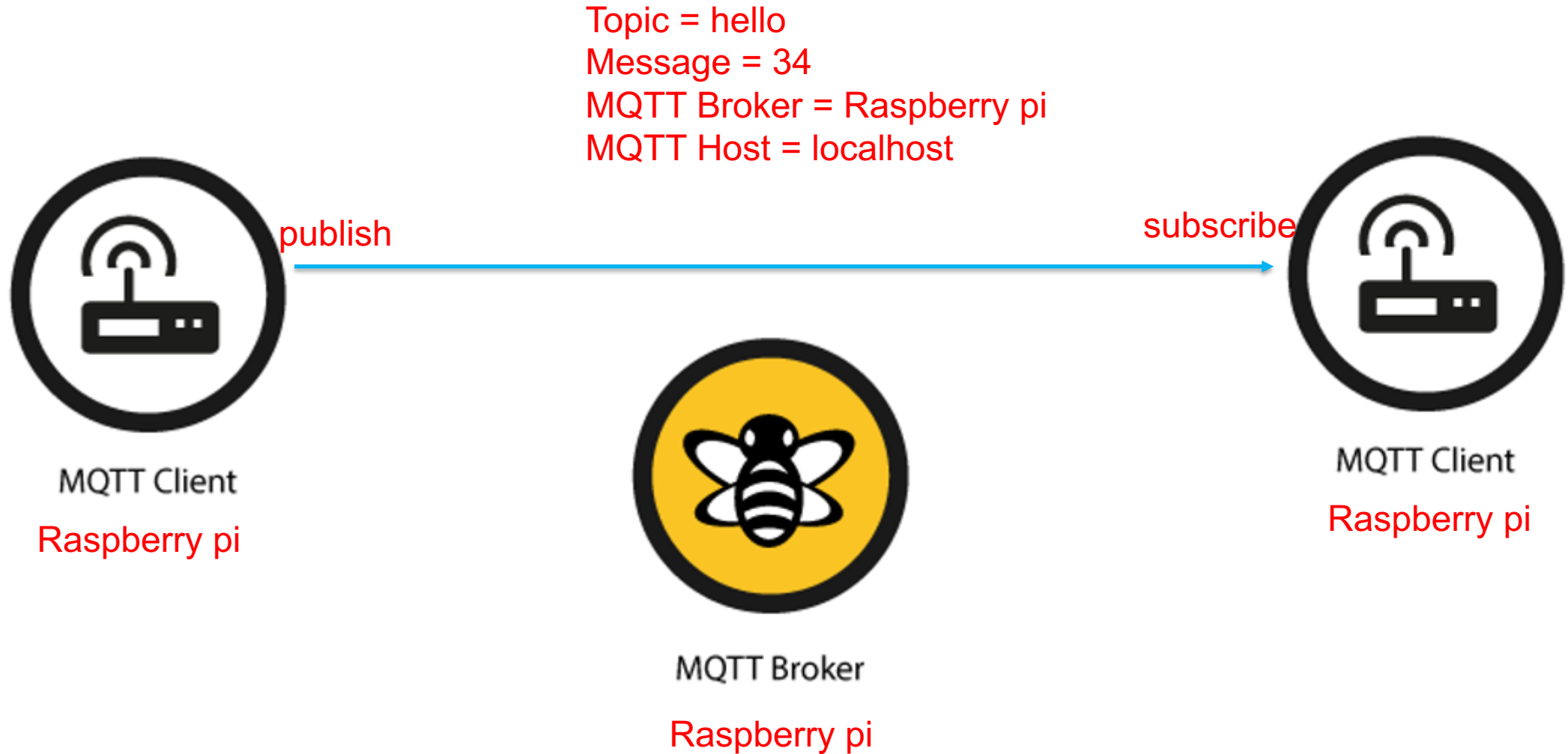
A terminal window titled 'pi@raspberrypi: ~' with a menu bar 'File Edit Tabs Help'. The command 'pi@raspberrypi:~ \$ mosquitto_pub -d -t temp -m "23.5"' has been entered and executed. The terminal background is black, and the text is green.

개발 환경 구축(IDP - Raspberry pi)

▶ MQTT – Python 라이브러리 설치

```
$ cd
$ sudo pip install --upgrade pip
$ sudo pip install paho-mqtt
$ sudo git clone https://github.com/eclipse/paho.mqtt.python
$ cd paho.mqtt.python
$ sudo python setup.py install
$ cd
```


MQTT Test



MQTT Test

MQTT

■ MQTT - Subscribe 프로그램 & Publisher 프로그램 작성

▶ MQTT - Subscribe 프로그램 작성

- 파일 명: subBasic.py

```
1  import paho.mqtt.client as mqtt
2
3  def on_connect ( client, userdata , flags, rc) :
4      print("Connect with result code " +str(rc) )
5      client.subscribe("hello")
6
7  def on_message ( client, userdata , msg ) :
8      print(msg.topic + " " + str(msg.payload))
9
10 client = mqtt.Client()
11 client.on_connect = on_connect
12 client.on_message = on_message
13 client.connect("localhost", 1883,60)
14 client.loop_forever()
```

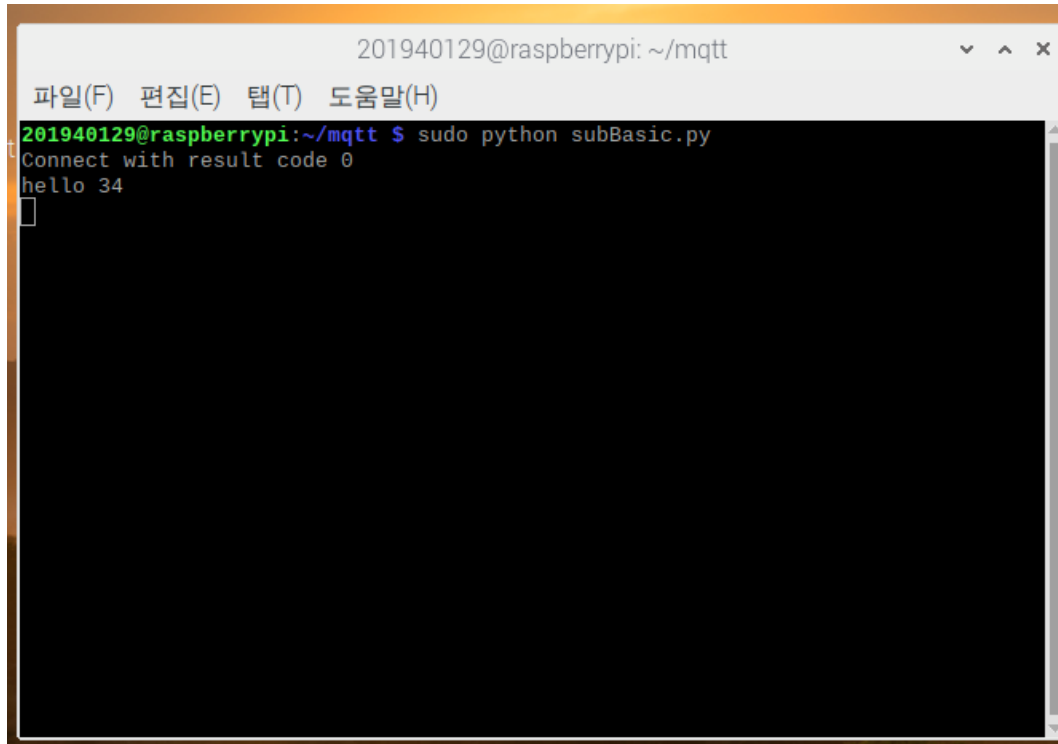
MQTT Test

MQTT

▶ MQTT – Subscribe 프로그램 동작 테스트

터미널 창 1

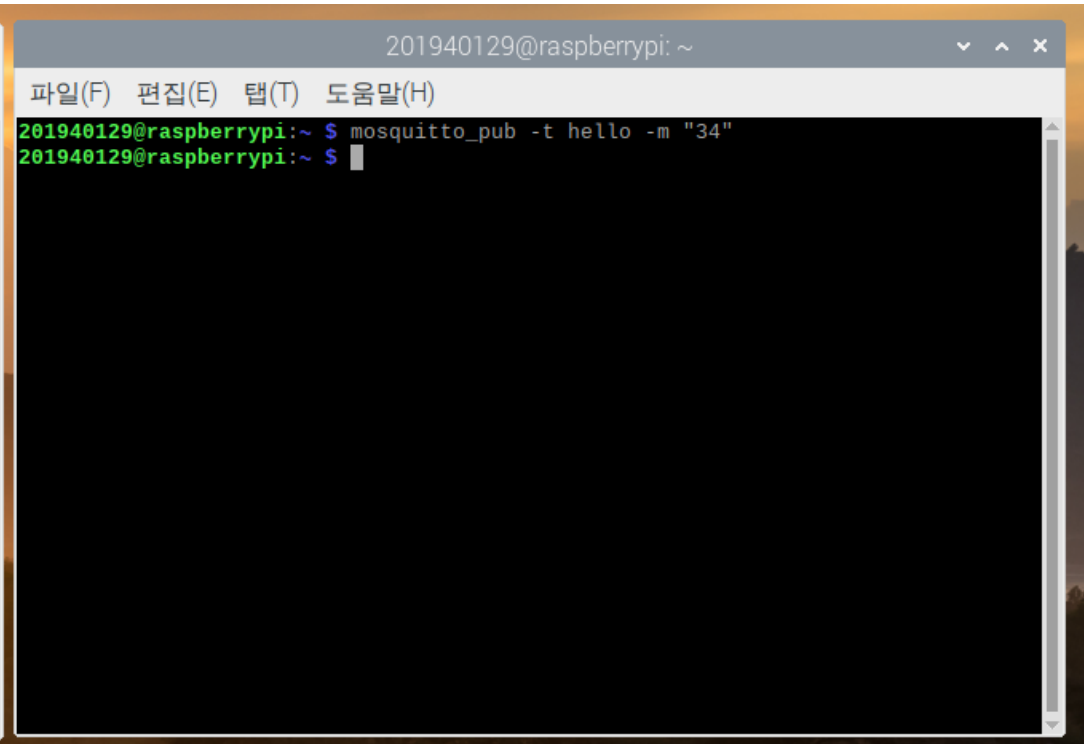
`$ sudo python subBasic.py`



```
201940129@raspberrypi: ~/mqtt
파일(F) 편집(E) 탭(T) 도움말(H)
201940129@raspberrypi:~/mqtt $ sudo python subBasic.py
Connect with result code 0
hello 34
█
```

터미널 창 2

`$mosquitto_pub -d -t hello -m "34"`
mosquitto_pub



```
201940129@raspberrypi: ~
파일(F) 편집(E) 탭(T) 도움말(H)
201940129@raspberrypi:~ $ mosquitto_pub -t hello -m "34"
201940129@raspberrypi:~ $ █
```

MQTT Test

MQTT

▶ MQTT – Publish 프로그램 작성

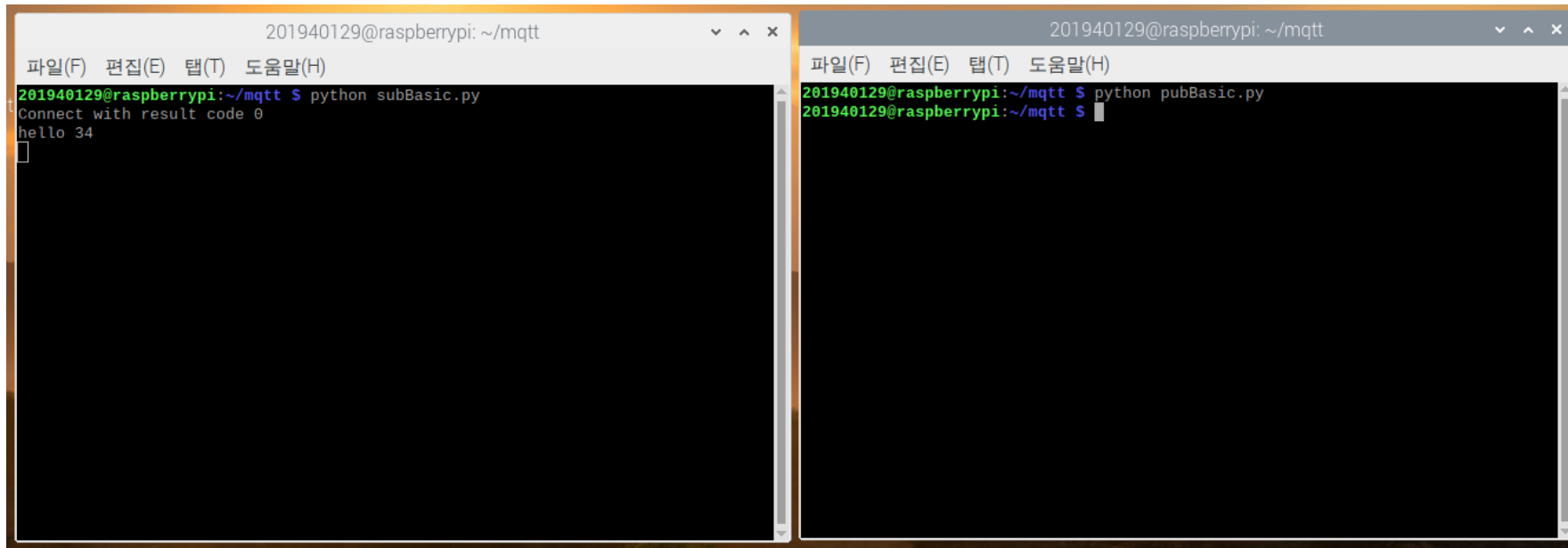
- 파일 명 : pubBasic.py

```
1 import paho.mqtt.publish as publish
2
3 publish.single("hello", "34", hostname="localhost")
```

▶ MQTT – Publish 프로그램 동작 테스트

\$ python subBasic.py

\$ python pubBasic.py



PC Server 환경 Setup (mosquitto 설치)

- <https://mosquitto.org/download/>
- 64비트, 32비트

Binary Installation

The binary packages listed below are supported by the Mosquitto project. In many cases Mosquitto is also available directly from official Linux/BSD distributions.

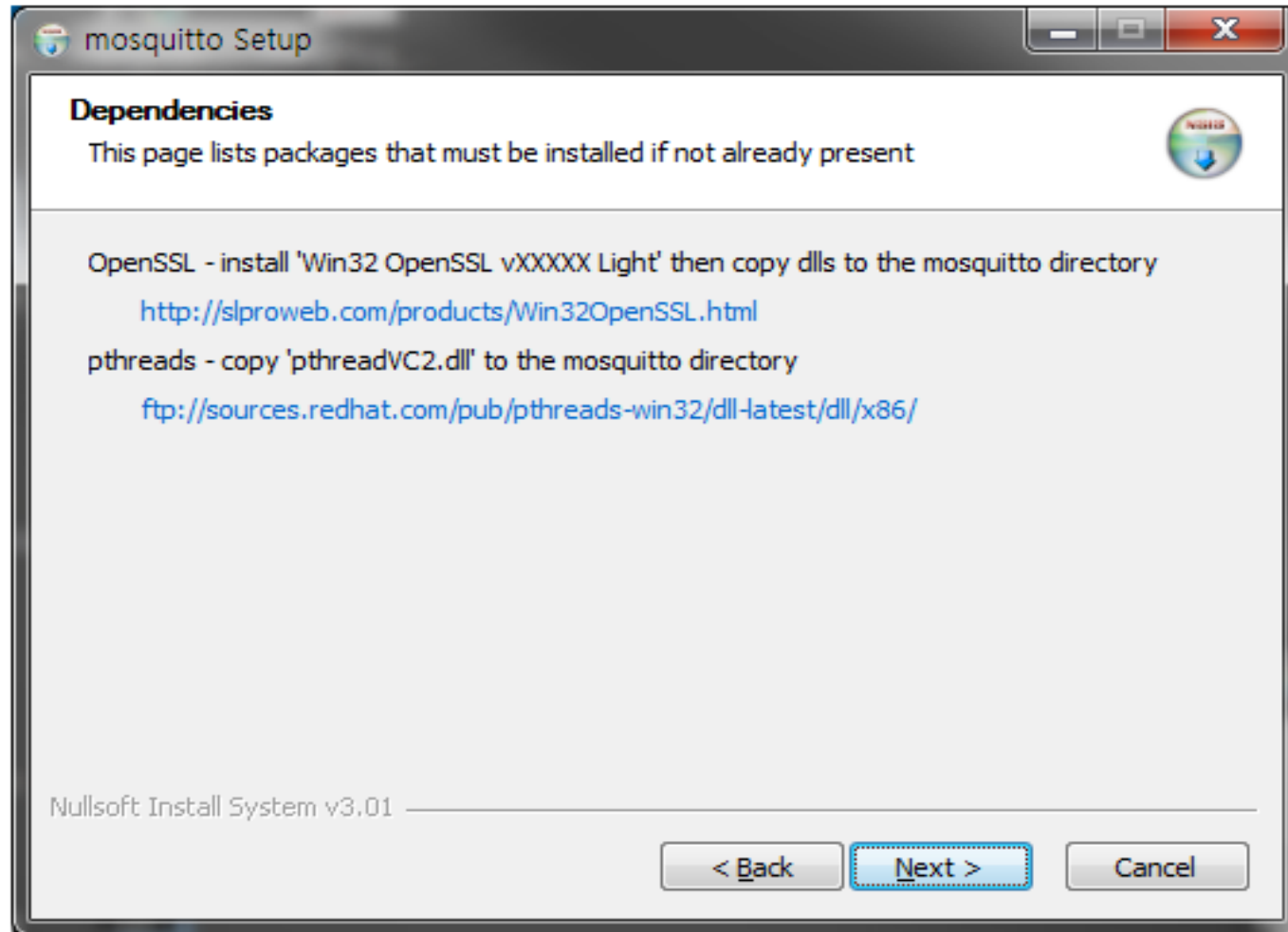
Windows

- [mosquitto-1.6.7-install-windows-x64.exe](#) (~1.4 MB) (64-bit build, Windows Vista and up, built with Visual Studio Community 2017)
- [mosquitto-1.6.7-install-windows-x32.exe](#) (~1.4 MB) (32-bit build, Windows Vista and up, built with Visual Studio Community 2017)

See also readme-windows.txt after installing.

PC Server 환경 Setup (mosquitto 설치)

- 설치 과정에서 아래와 같은 설치 조건을 확인할 수 있습니다.
- Next, Next를 눌러 설치를 진행합니다.



PC Server 환경 Setup (mosquitto 설치)

- 설치 조건 중 하나인 OpenSSL을 설치하기 위해 설치 파일을 다운로드합니다.
- 자신의 Windows 환경에 맞게 다운받습니다.
- Siproweb이란
- <http://siproweb.com/products/Win32OpenSSL.html>

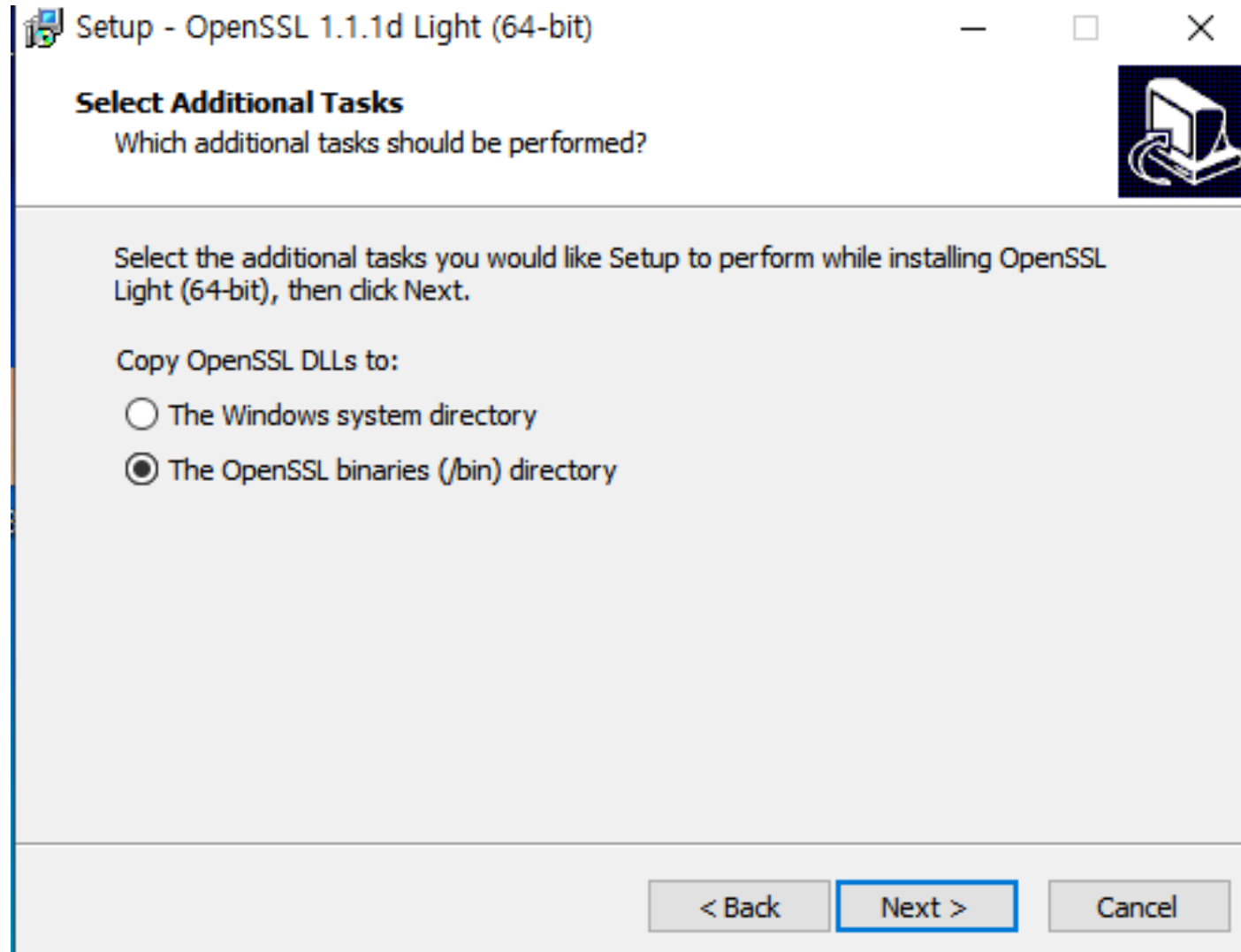
Download Win32/Win64 OpenSSL

Download Win32/Win64 OpenSSL today using the links below!

File	Type	Description
Win64 OpenSSL v1.1.1d Light EXE MSI (experimental)	3MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v1.1.1d (Recommended for users by the creators of OpenSSL). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v1.1.1d EXE MSI (experimental)	43MB Installer	Installs Win64 OpenSSL v1.1.1d (Recommended for software developers by the creators of OpenSSL). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v1.1.1d Light EXE MSI (experimental)	3MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v1.1.1d (Only install this if you need 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v1.1.1d EXE MSI (experimental)	30MB Installer	Installs Win32 OpenSSL v1.1.1d (Only install this if you need 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v1.1.0L Light	3MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v1.1.0L (Recommended for users by the creators of OpenSSL). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v1.1.0L	37MB Installer	Installs Win64 OpenSSL v1.1.0L (Recommended for software developers by the creators of OpenSSL). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v1.1.0L Light	3MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v1.1.0L (Only install this if you need 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v1.1.0L	30MB Installer	Installs Win32 OpenSSL v1.1.0L (Only install this if you are a software developer needing 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v1.0.2t Light	3MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v1.0.2t (Recommended for users by the creators of OpenSSL). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v1.0.2t	23MB Installer	Installs Win64 OpenSSL v1.0.2t (Recommended for software developers by the creators of OpenSSL). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v1.0.2t Light	2MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v1.0.2t (Only install this if you need 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v1.0.2t	20MB Installer	Installs Win32 OpenSSL v1.0.2t (Only install this if you are a software developer needing 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.

PC Server 환경 Setup (mosquitto 설치)

- Next, Next 하여 다운 받습니다.



PC Server 환경 Setup (mosquitto 설치)


- OpenSSL을 설치한 후, 설치 경로(C:\Program Files\OpenSSL-Win64)에서 DLL 파일을 복사하여 모스키토 설치 경로(C:\Program Files\mosquitto)에 복사합니다.

로컬 디스크 (C:) > Program Files > mosquitto				
이름	수정된 날짜	유형	크기	
devel	2019-08-14 오후...	파일 폴더		
aclfile.example	2019-08-02 오전...	EXAMPLE 파일	1KB	
ChangeLog	2019-08-02 오전...	텍스트 문서	93KB	
edl-v10	2019-08-02 오전...	파일	2KB	
epl-v10	2019-08-02 오전...	파일	12KB	
libcrypto-1_1-x64.dll	2019-05-29 오전...	응용 프로그램 확장	3,328KB	
libssl-1_1-x64.dll	2019-05-29 오전...	응용 프로그램 확장	666KB	
mosquitto.conf	2019-08-02 오전...	CONF 파일	44KB	
mosquitto.dll	2019-08-02 오전...	응용 프로그램 확장	80KB	
mosquitto	2019-08-02 오전...	응용 프로그램	295KB	
mosquitto_passwd	2019-08-02 오전...	응용 프로그램	20KB	
mosquitto_pub	2019-08-02 오전...	응용 프로그램	46KB	
mosquitto_sub	2019-08-02 오전...	응용 프로그램	48KB	
mosquittoptp.dll	2019-08-02 오전...	응용 프로그램 확장	18KB	
pthreadVC2.dll	2019-08-14 오후...	응용 프로그램 확장	55KB	
pwfile.example	2019-08-02 오전...	EXAMPLE 파일	1KB	
readme	2019-08-02 오전...	MD 파일	4KB	
readme-windows	2019-08-02 오전...	텍스트 문서	3KB	
Uninstall	2019-08-14 오후...	응용 프로그램	65KB	

PC Server 환경 Setup (mosquitto 설치)

- 또 다른 설치 조건인 pthreads를 해결하기 위해 FTP에서 파일을 다운로드하여 모스키토 설치 경로에 복사합니다.
- <ftp://sources.redhat.com/pub/pthreads-win32/dll-latest/dll/x86/>

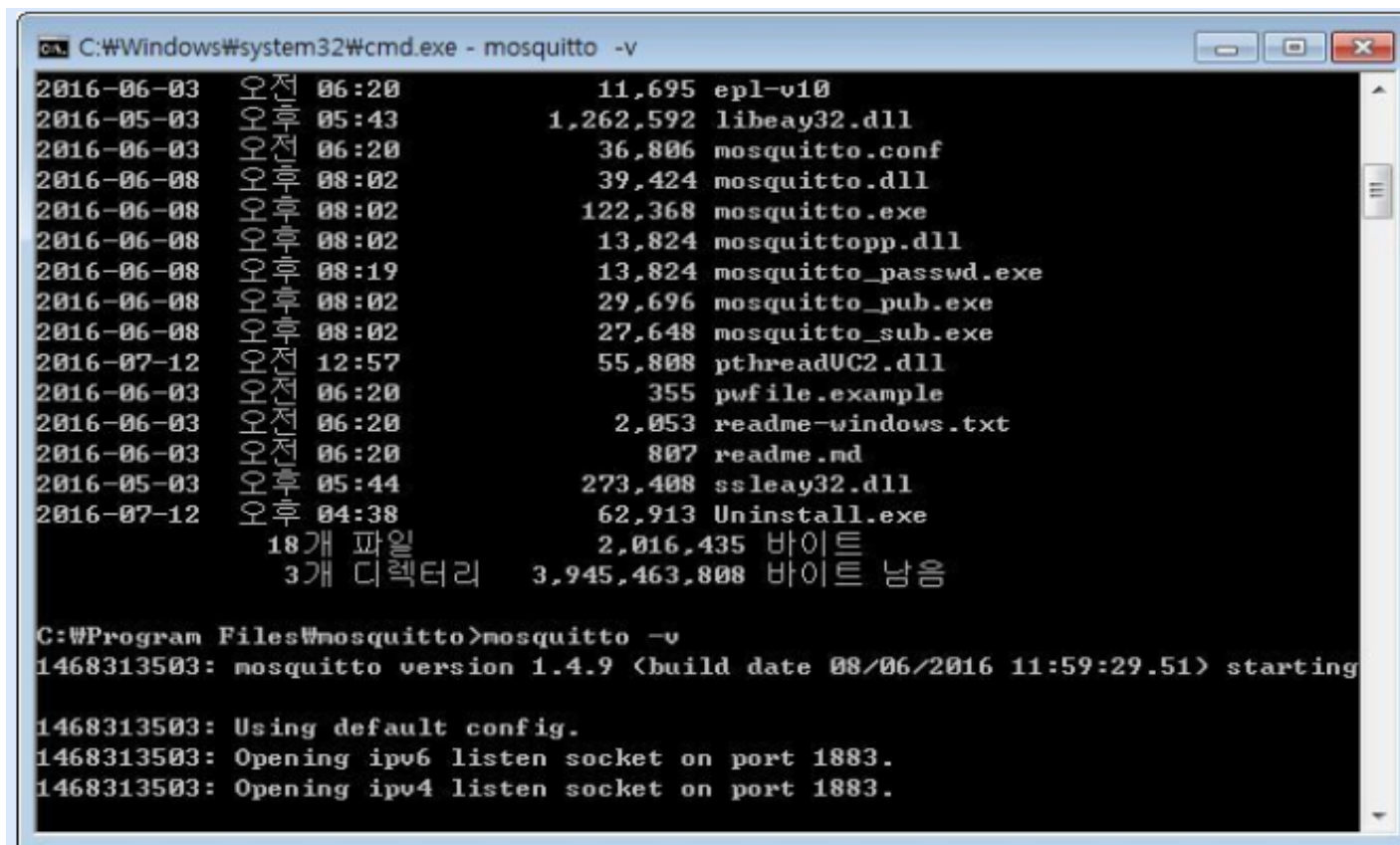
/pub/pthreads-win32/dll-latest/dll/x86/의 색인

 [상위 디렉터리]

이름	크기	수정된 날짜
 md5.sum	293 B	15. 2. 5. 오전 9:00:00
 pthreadGC2.dll	117 kB	12. 5. 27. 오전 9:00:00
 pthreadGCE2.dll	119 kB	12. 5. 27. 오전 9:00:00
 pthreadVC2.dll	54.5 kB	12. 5. 27. 오전 9:00:00
 pthreadVCE2.dll	60.5 kB	12. 5. 27. 오전 9:00:00
 pthreadVSE2.dll	56.0 kB	12. 5. 27. 오전 9:00:00
 sha512.sum	866 B	15. 2. 5. 오전 9:00:00

PC - Mosquitto broker 실행

- mosquitto -v
- Mosquitto 폴더에서 실행
- -v 옵션은 모든 통신과정을 보여주는 옵션입니다.
- 기본 포트인 1883포트로 실행됩니다.



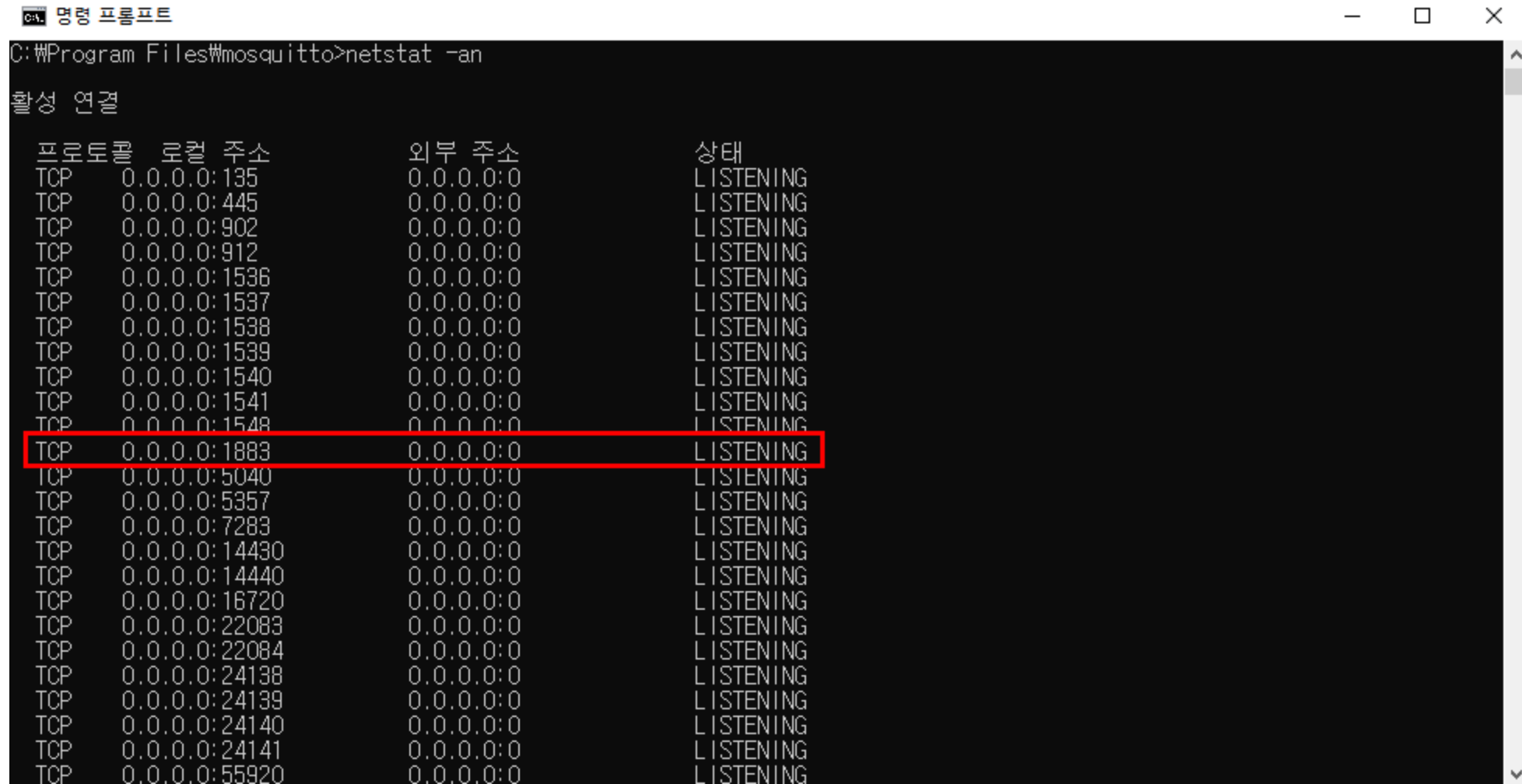
```
C:\Windows\system32\cmd.exe - mosquitto -v
2016-06-03 오전 06:20 11,695 epl-v10
2016-05-03 오후 05:43 1,262,592 libeay32.dll
2016-06-03 오전 06:20 36,806 mosquitto.conf
2016-06-08 오후 08:02 39,424 mosquitto.dll
2016-06-08 오후 08:02 122,368 mosquitto.exe
2016-06-08 오후 08:02 13,824 mosquitto_top.dll
2016-06-08 오후 08:19 13,824 mosquitto_passwd.exe
2016-06-08 오후 08:02 29,696 mosquitto_pub.exe
2016-06-08 오후 08:02 27,648 mosquitto_sub.exe
2016-07-12 오전 12:57 55,808 pthreadUC2.dll
2016-06-03 오전 06:20 355 pwfile.example
2016-06-03 오전 06:20 2,053 readme-windows.txt
2016-06-03 오전 06:20 807 readme.md
2016-05-03 오후 05:44 273,408 ssleay32.dll
2016-07-12 오후 04:38 62,913 Uninstall.exe
18개 파일 2,016,435 바이트
3개 디렉터리 3,945,463,808 바이트 남음

C:\Program Files\mosquitto>mosquitto -v
1468313503: mosquitto version 1.4.9 (build date 08/06/2016 11:59:29.51) starting

1468313503: Using default config.
1468313503: Opening ipv6 listen socket on port 1883.
1468313503: Opening ipv4 listen socket on port 1883.
```

PC - Mosquitto broker

- Cmd에서 “netstat -an”을 입력하면, 활성 연결이 나옵니다.

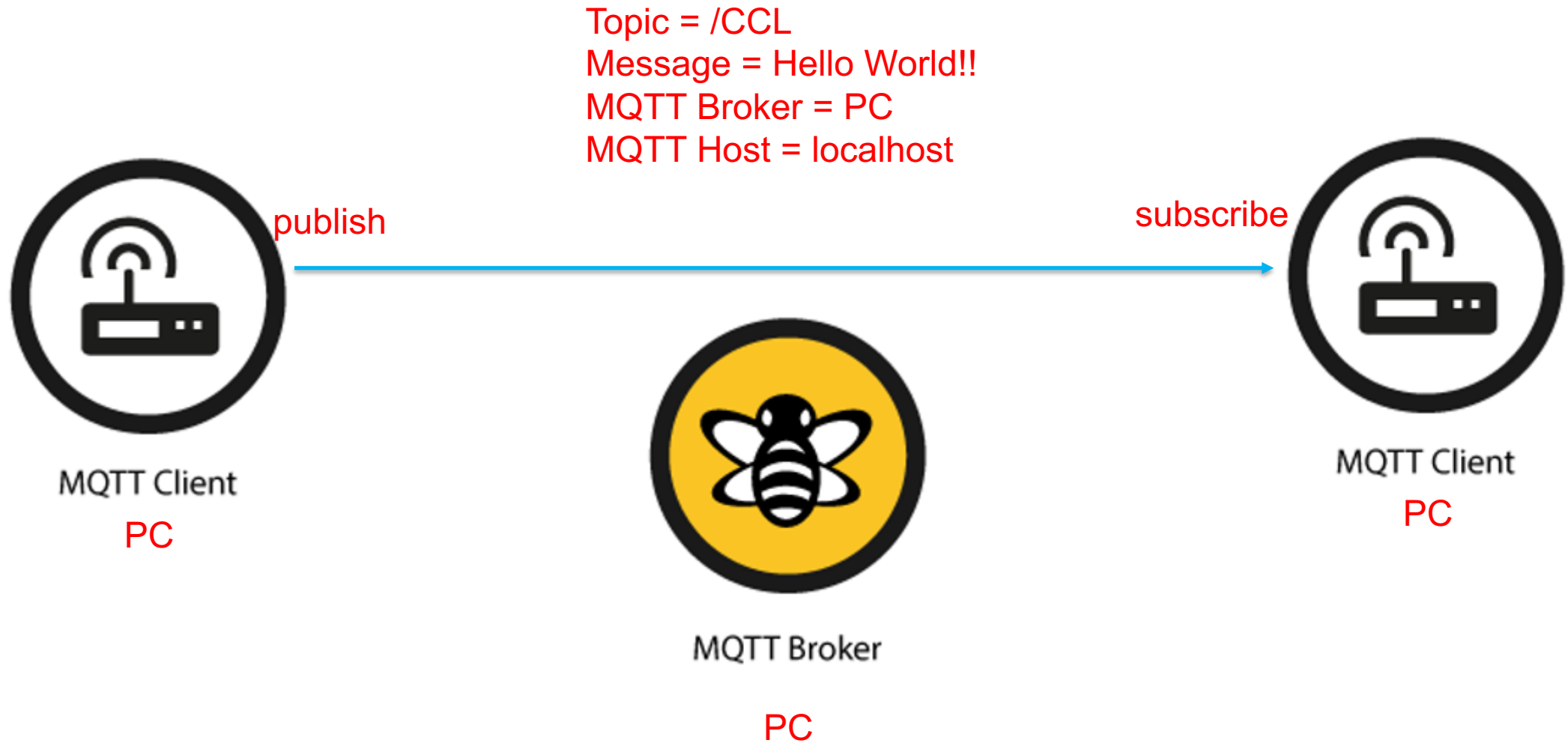


```
C:\Program Files\Mosquitto>netstat -an

활성 연결

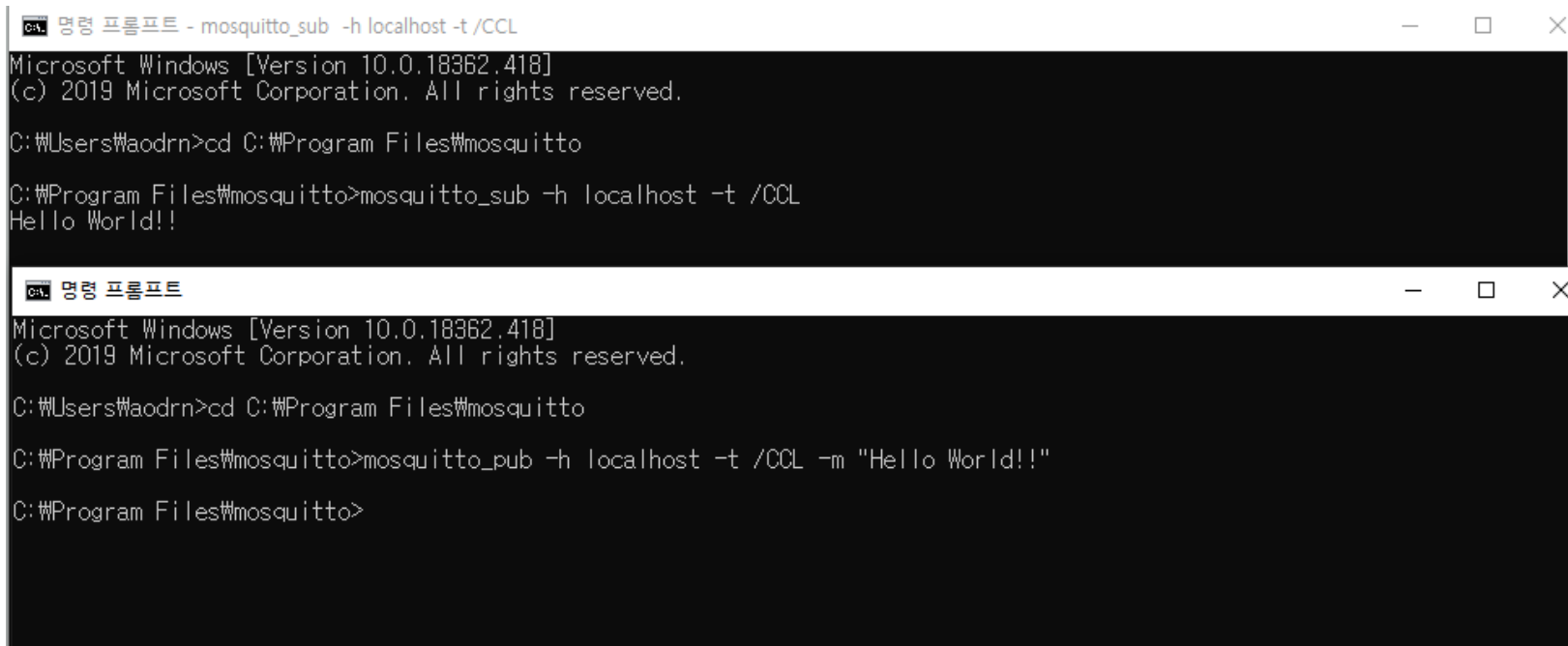
프로토콜  로컬 주소          외부 주소          상태
TCP      0.0.0.0:135      0.0.0.0:0          LISTENING
TCP      0.0.0.0:445      0.0.0.0:0          LISTENING
TCP      0.0.0.0:902      0.0.0.0:0          LISTENING
TCP      0.0.0.0:912      0.0.0.0:0          LISTENING
TCP      0.0.0.0:1536     0.0.0.0:0          LISTENING
TCP      0.0.0.0:1537     0.0.0.0:0          LISTENING
TCP      0.0.0.0:1538     0.0.0.0:0          LISTENING
TCP      0.0.0.0:1539     0.0.0.0:0          LISTENING
TCP      0.0.0.0:1540     0.0.0.0:0          LISTENING
TCP      0.0.0.0:1541     0.0.0.0:0          LISTENING
TCP      0.0.0.0:1548     0.0.0.0:0          LISTENING
TCP      0.0.0.0:1883     0.0.0.0:0          LISTENING
TCP      0.0.0.0:5040     0.0.0.0:0          LISTENING
TCP      0.0.0.0:5357     0.0.0.0:0          LISTENING
TCP      0.0.0.0:7283     0.0.0.0:0          LISTENING
TCP      0.0.0.0:14430    0.0.0.0:0          LISTENING
TCP      0.0.0.0:14440    0.0.0.0:0          LISTENING
TCP      0.0.0.0:16720    0.0.0.0:0          LISTENING
TCP      0.0.0.0:22083    0.0.0.0:0          LISTENING
TCP      0.0.0.0:22084    0.0.0.0:0          LISTENING
TCP      0.0.0.0:24138    0.0.0.0:0          LISTENING
TCP      0.0.0.0:24139    0.0.0.0:0          LISTENING
TCP      0.0.0.0:24140    0.0.0.0:0          LISTENING
TCP      0.0.0.0:24141    0.0.0.0:0          LISTENING
TCP      0.0.0.0:55920    0.0.0.0:0          LISTENING
```

PC Server - MQTT Test



PC에서 확인

- Mosquitto 폴더로 이동해서 실행
- `mosquitto_sub -h 호스트ip -t 토픽명(ex:mosquitto_sub -h 192.168.0.12 -t /CCL)`
- `Mosquitto_pub -h 호스트ip -t 토픽명 -m 메시지`
- (ex:mosquitto_pub -h 192.168.0.12 -t /CCL -m "Hello World!!")



```
명령 프롬프트 - mosquitto_sub -h localhost -t /CCL
Microsoft Windows [Version 10.0.18362.418]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\baodrn>cd C:\Program Files\mosquitto

C:\Program Files\mosquitto>mosquitto_sub -h localhost -t /CCL
Hello World!!

명령 프롬프트
Microsoft Windows [Version 10.0.18362.418]
(c) 2019 Microsoft Corporation. All rights reserved.

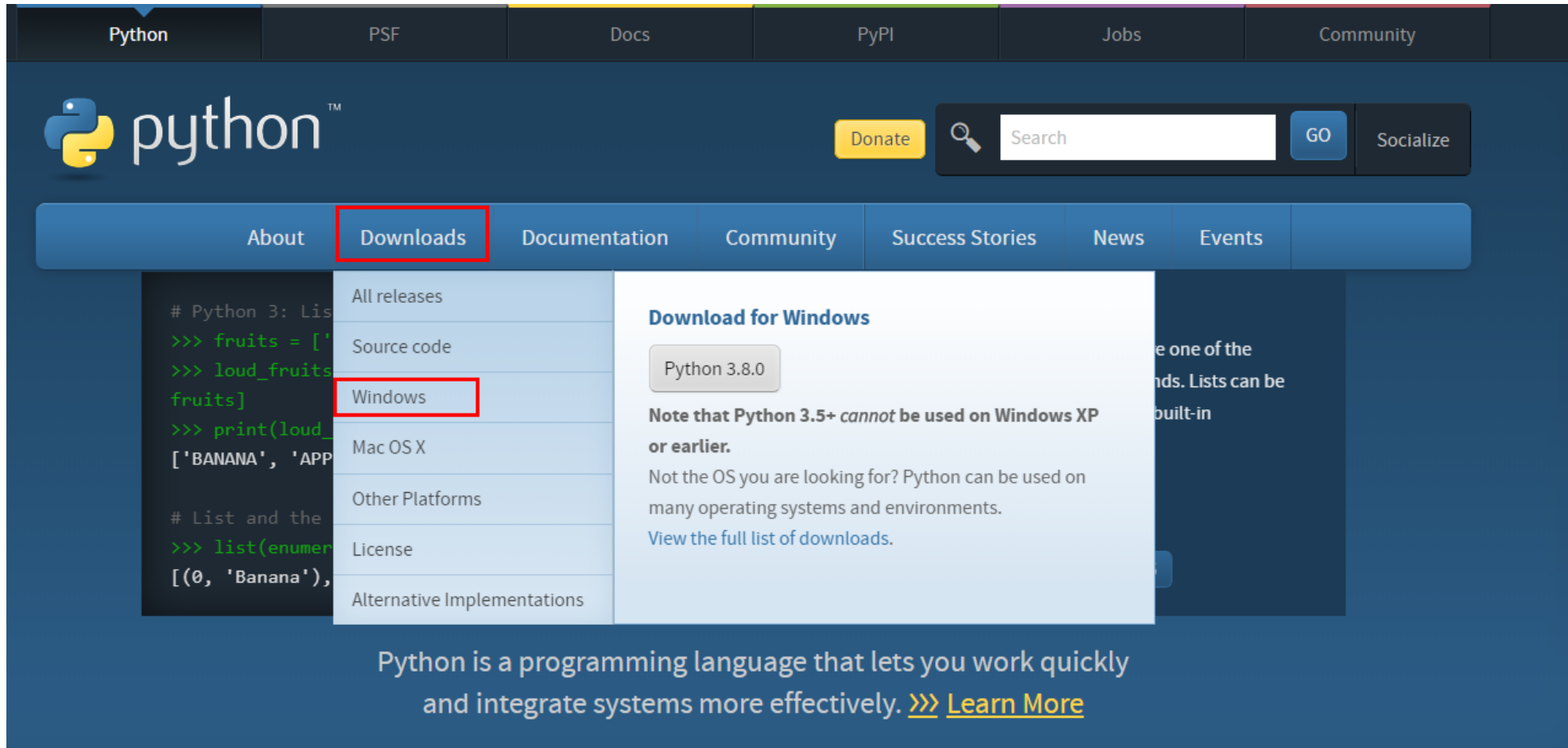
C:\Users\baodrn>cd C:\Program Files\mosquitto

C:\Program Files\mosquitto>mosquitto_pub -h localhost -t /CCL -m "Hello World!!"

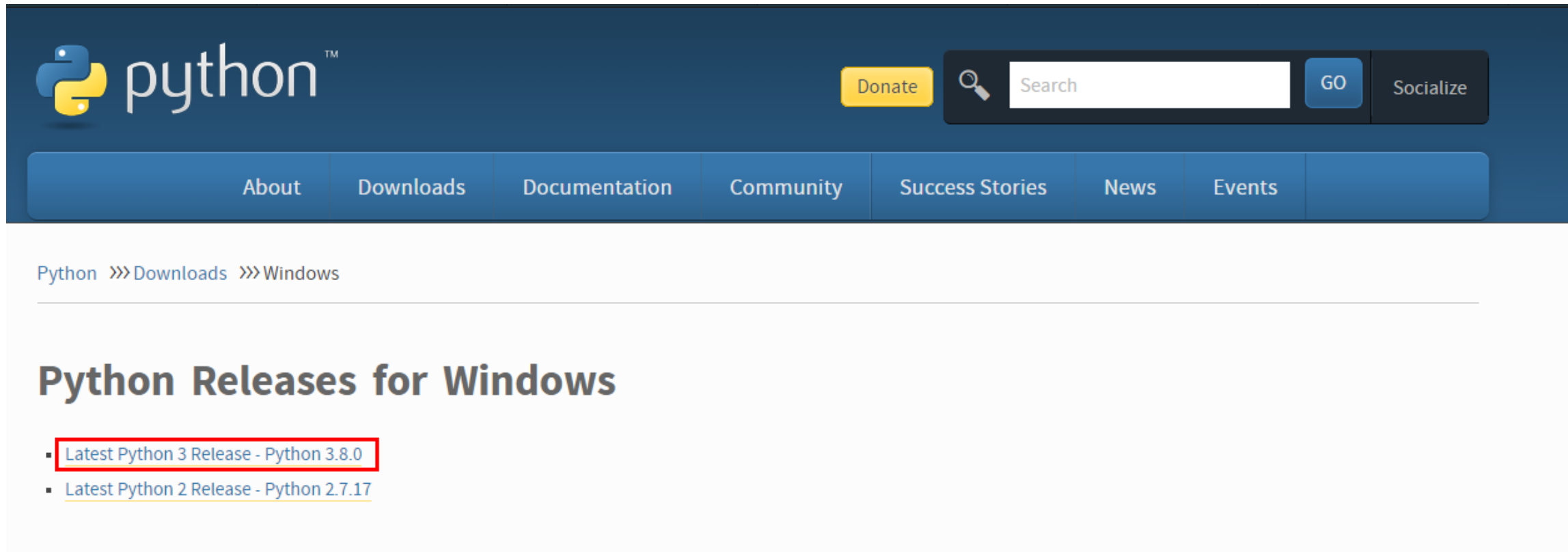
C:\Program Files\mosquitto>
```

Python 3 설치(python 3버전 없는 사람만)

- www.python.org에서 다운로드 메뉴 - Python 3.5이상 선택해서 다운로드



Python 3 설치



The screenshot shows the Python.org website. The top navigation bar includes the Python logo, a 'Donate' button, a search bar with a 'GO' button, and a 'Socialize' button. Below this is a secondary navigation bar with links for 'About', 'Downloads', 'Documentation', 'Community', 'Success Stories', 'News', and 'Events'. The main content area shows the breadcrumb 'Python >>> Downloads >>> Windows' followed by the heading 'Python Releases for Windows'. Under this heading, there is a list of two items: 'Latest Python 3 Release - Python 3.8.0' (highlighted with a red box) and 'Latest Python 2 Release - Python 2.7.17' (underlined).

python™

Donate

Search

GO

Socialize

About Downloads Documentation Community Success Stories News Events

Python >>> Downloads >>> Windows

Python Releases for Windows

- Latest Python 3 Release - Python 3.8.0
- Latest Python 2 Release - Python 2.7.17

Python 3 설치

Files

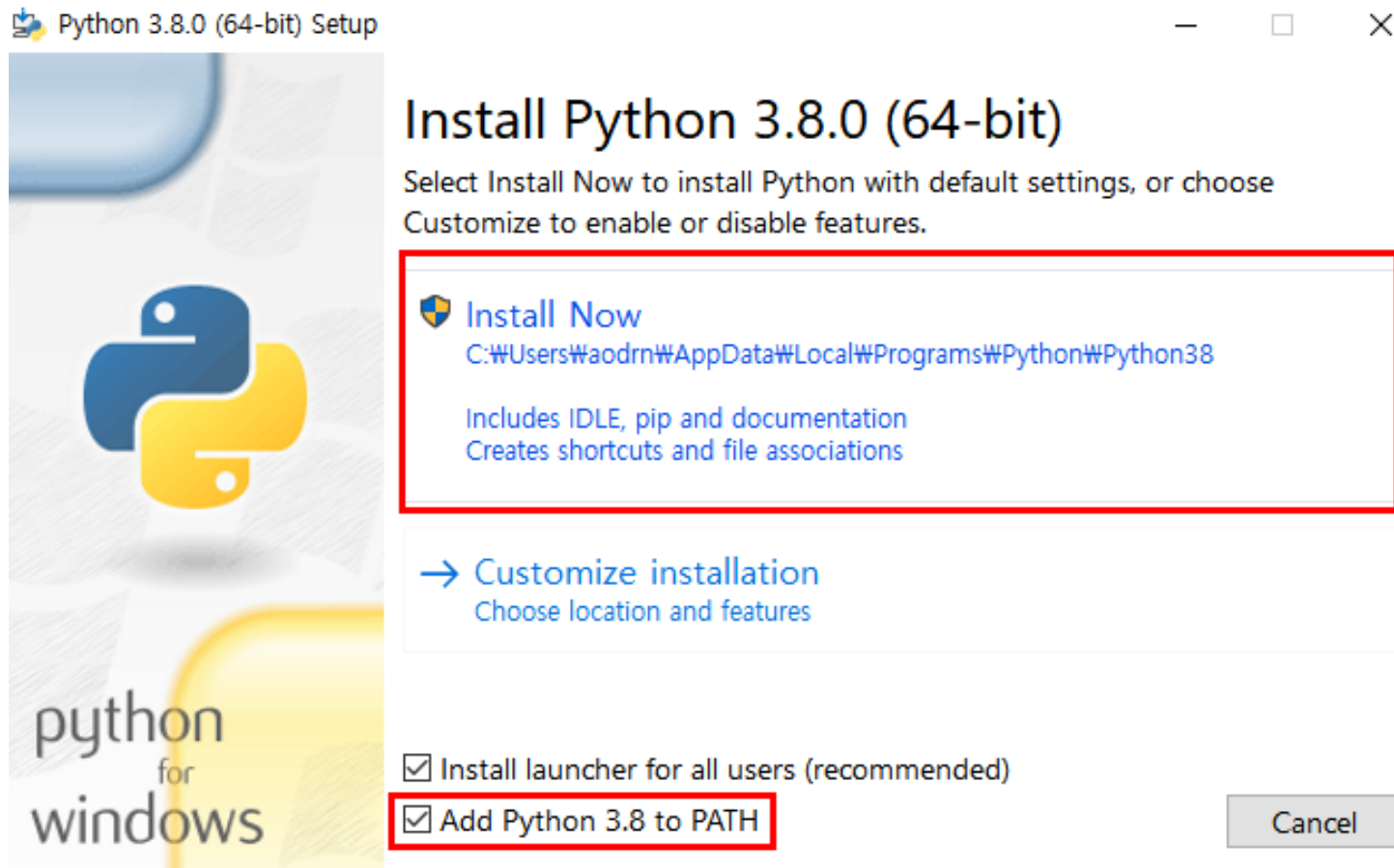
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		e18a9d1a0a6d858b9787e03fc6fdaa20	23949883	SIG
XZ compressed source tarball	Source release		dbac8df9d8b9edc678d0f4cacdb7dbb0	17829824	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	f5f9ae9f416170c6355cab7256bb75b5	29005746	SIG
Windows help file	Windows		1c33359821033ddb3353c8e5b6e7e003	8457529	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	99cca948512b53fb165084787143ef19	8084795	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	29ea87f24c32f5e924b7d63f8a08ee8d	27505064	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	f93f7ba8cd48066c59827752e531924b	1363336	SIG
Windows x86 embeddable zip file	Windows		2ec3abf05f3f1046e0dbd1ca5c74ce88	7213298	SIG
Windows x86 executable installer	Windows		412a649d36626d33b8ca5593cf18318c	26406312	SIG
Windows x86 web-based installer	Windows		50d484ff0b08722b3cf51f9305f49fdc	1325368	SIG

64비트

32비트

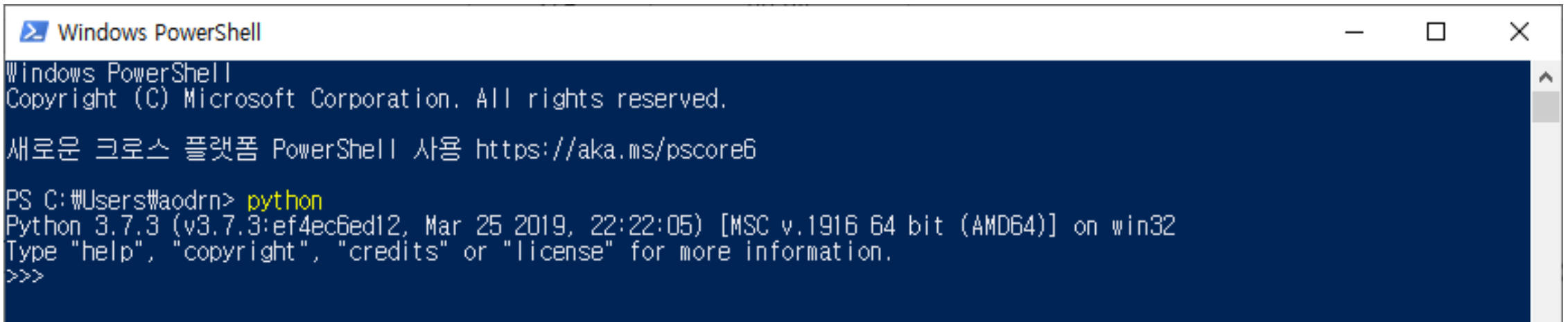
Python 3 설치

- 설치 과정에서 Add Python 3.8 to PATH 반드시 체크



Python 3 설치

- 설치 확인
- 윈도우키 + R을 눌러서 나오는 창에 powershell이라고 치고, 확인을 눌러서 powershell 실행
- Powershell 화면에서 python이라고 치고 입력해서 오류가 나오지 않으면 설치 성공

A screenshot of a Windows PowerShell window. The title bar says "Windows PowerShell". The terminal text shows the standard PowerShell startup messages, followed by the command "python" being entered. The output shows "Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32" and the prompt ">>>".

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Users\waodrn> python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Python 3 설치

- Python 실행이 안되면
 - 환경 변수 설정
 - Path에 자신의 Python 폴더와 Python \ Scripts 폴더 추가
-
- ex) C:\Users\oadrn\AppData\Local\Programs\Python\Python37
 - ex) C:\Users\oadrn\AppData\Local\Programs\Python\Python37\Scripts

PC에 Python 라이브러리 설치

- PC에 Python이 설치되어 있지 않다면 Python 먼저 설치!!!!!!

Cmd 창

```
pip install --upgrade pip
```

```
pip install paho-mqtt
```

- <https://github.com/eclipse/paho.mqtt.python>

503 commits 2 branches 10 releases 33 contributors View license

Branch: master New pull request Create new file Upload files Find file **Clone or download**

PierreF Merge pull request #390 from naveenrobo/master

examples	Merge branch 'fixes' into develop
src/paho	Merge pull request #390 from naveenrobo/master
test	Fix test following previous commit
tests	Start test_broker in unitset set-up

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

Open in Desktop **Download ZIP**

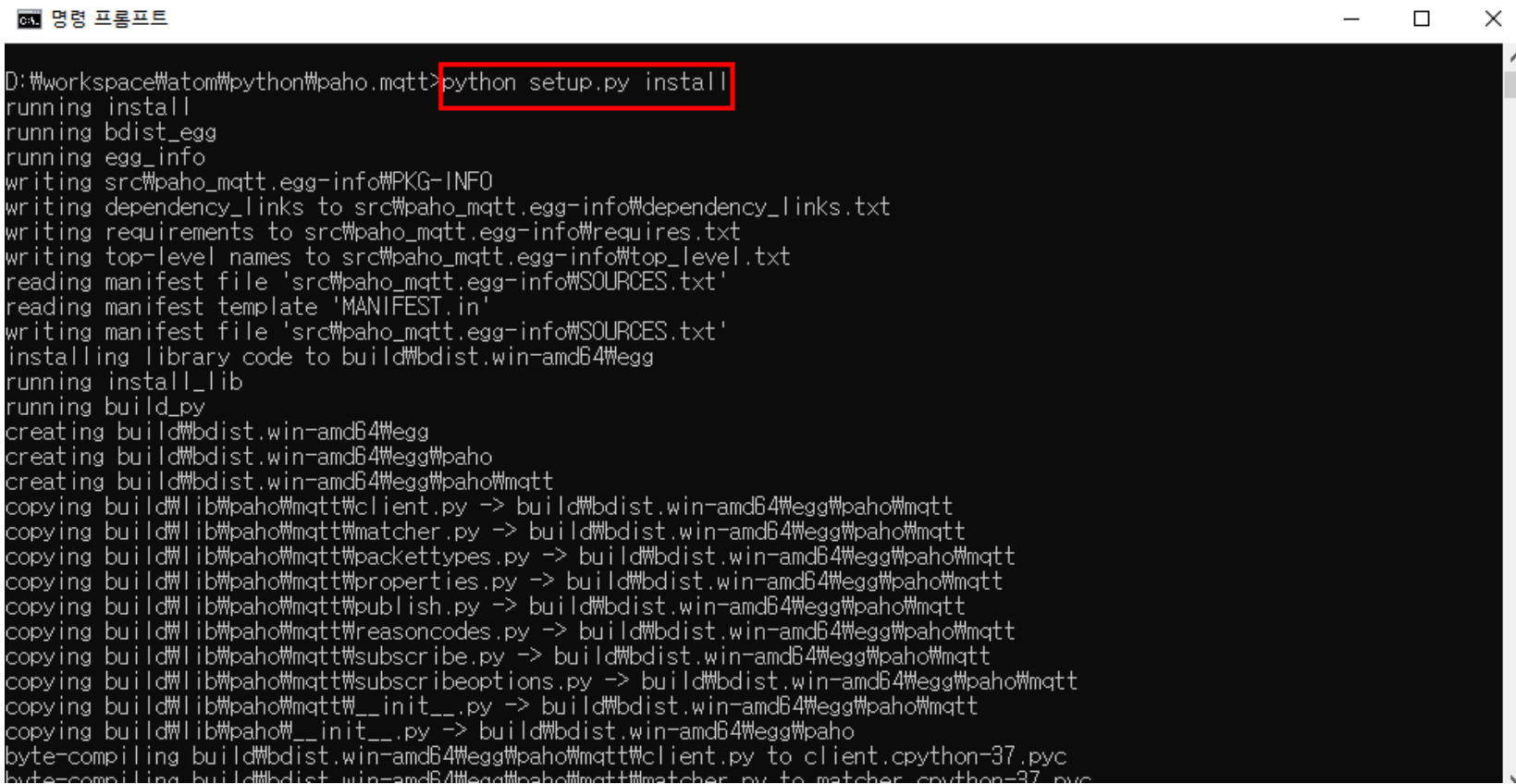
Python 라이브러리 설치

압축 풀고 다운로드한 폴더로 이동

cd paho.mqtt.python

설치 파일 실행

Python setup.py install



```
CA: 명령 프롬프트
D:\workspace\atom\python\paho.mqtt>python setup.py install
running install
running bdist_egg
running egg_info
writing src\paho_mqtt.egg-info\PKG-INFO
writing dependency_links to src\paho_mqtt.egg-info\dependency_links.txt
writing requirements to src\paho_mqtt.egg-info\requires.txt
writing top-level names to src\paho_mqtt.egg-info\top_level.txt
reading manifest file 'src\paho_mqtt.egg-info\SOURCES.txt'
reading manifest template 'MANIFEST.in'
writing manifest file 'src\paho_mqtt.egg-info\SOURCES.txt'
installing library code to build\bdist.win-amd64\egg
running install_lib
running build_py
creating build\bdist.win-amd64\egg
creating build\bdist.win-amd64\egg\paho
creating build\bdist.win-amd64\egg\paho\mqtt
copying build\lib\paho\mqtt\client.py -> build\bdist.win-amd64\egg\paho\mqtt
copying build\lib\paho\mqtt\matcher.py -> build\bdist.win-amd64\egg\paho\mqtt
copying build\lib\paho\mqtt\packettypes.py -> build\bdist.win-amd64\egg\paho\mqtt
copying build\lib\paho\mqtt\properties.py -> build\bdist.win-amd64\egg\paho\mqtt
copying build\lib\paho\mqtt\publish.py -> build\bdist.win-amd64\egg\paho\mqtt
copying build\lib\paho\mqtt\reasoncodes.py -> build\bdist.win-amd64\egg\paho\mqtt
copying build\lib\paho\mqtt\subscribe.py -> build\bdist.win-amd64\egg\paho\mqtt
copying build\lib\paho\mqtt\subscribeoptions.py -> build\bdist.win-amd64\egg\paho\mqtt
copying build\lib\paho\mqtt\__init__.py -> build\bdist.win-amd64\egg\paho\mqtt
copying build\lib\paho\__init__.py -> build\bdist.win-amd64\egg\paho
byte-compiling build\bdist.win-amd64\egg\paho\mqtt\client.py to client.cpython-37.pyc
byte-compiling build\bdist.win-amd64\egg\paho\mqtt\matcher.py to matcher.cpython-37.pyc
```

paho

- The Eclipse Paho project provides open-source client implementations of MQTT and MQTT-SN messaging protocols aimed at new, existing, and emerging applications for the Internet of Things (IoT).
- MQTT is a light-weight publish/subscribe messaging protocol, originally created by IBM and Arcom (later to become part of Eurotech) around 1998. The [MQTT 3.1.1 specification](#) has now been standardised by the [OASIS consortium](#). The standard is available in a variety of [formats](#).
- Eclipse Paho
 - <https://www.eclipse.org/paho/>
- Eclipse Paho Python Client – documentation
 - <https://www.eclipse.org/paho/clients/python/docs/>

RP+PC간의 MQTT 통신 시나리오

온도,습도를
RP에서 publish
PC 에서 온도 Subscribe

`{'Temperature':oo, 'Humidity': oo}`

PC가 온도를 체크하여
일정 온도가 넘으면
PC에서 “on” publish

“on” 메시지

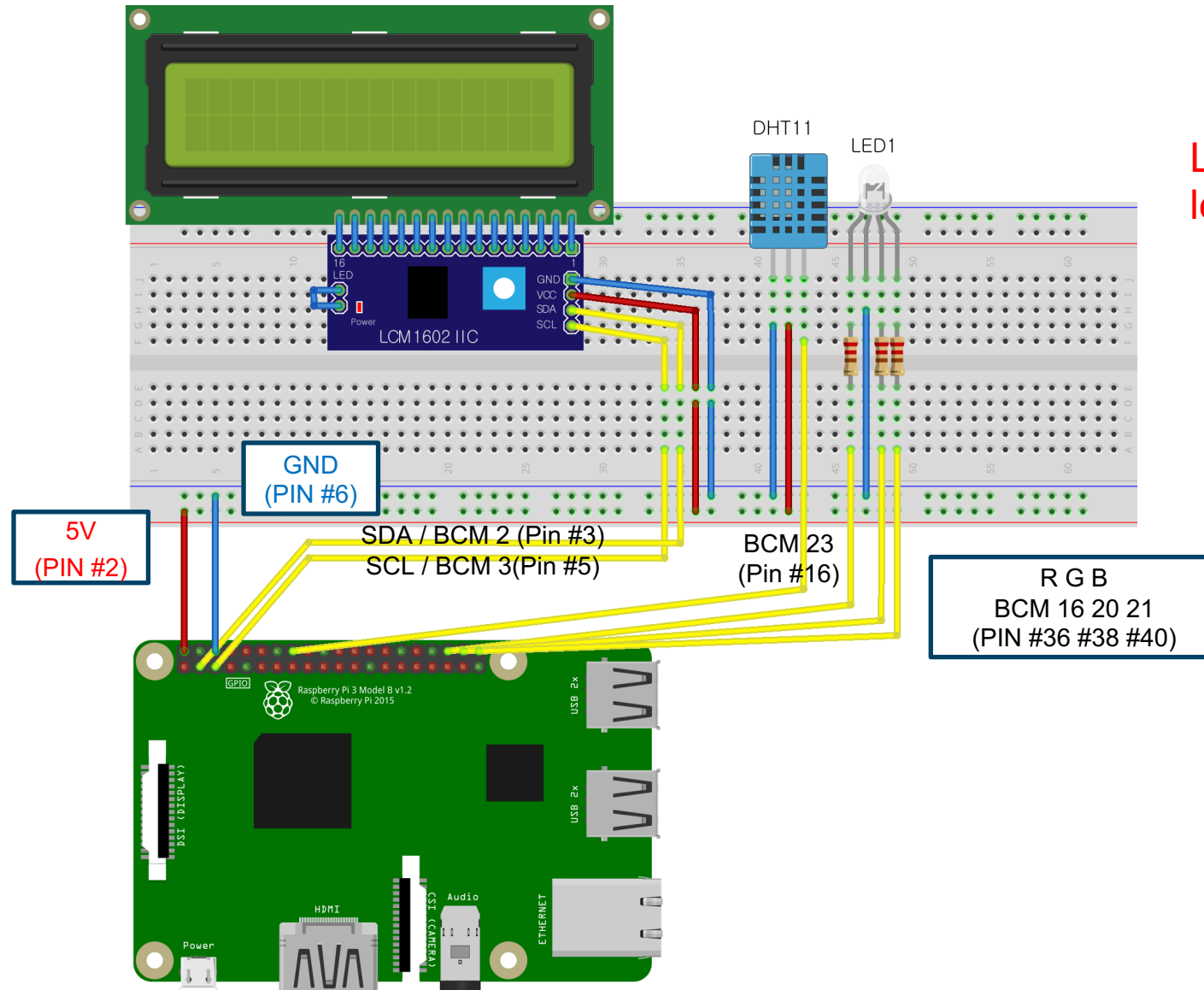
RP에서
LED 빨간색 ON

PC가 온도를 체크하여
일정 온도로 내려가면
PC에서 “off” publish

“off” 메시지

RP에서
LED 빨간색 OFF

RP+PC간의 MQTT 통신 시나리오

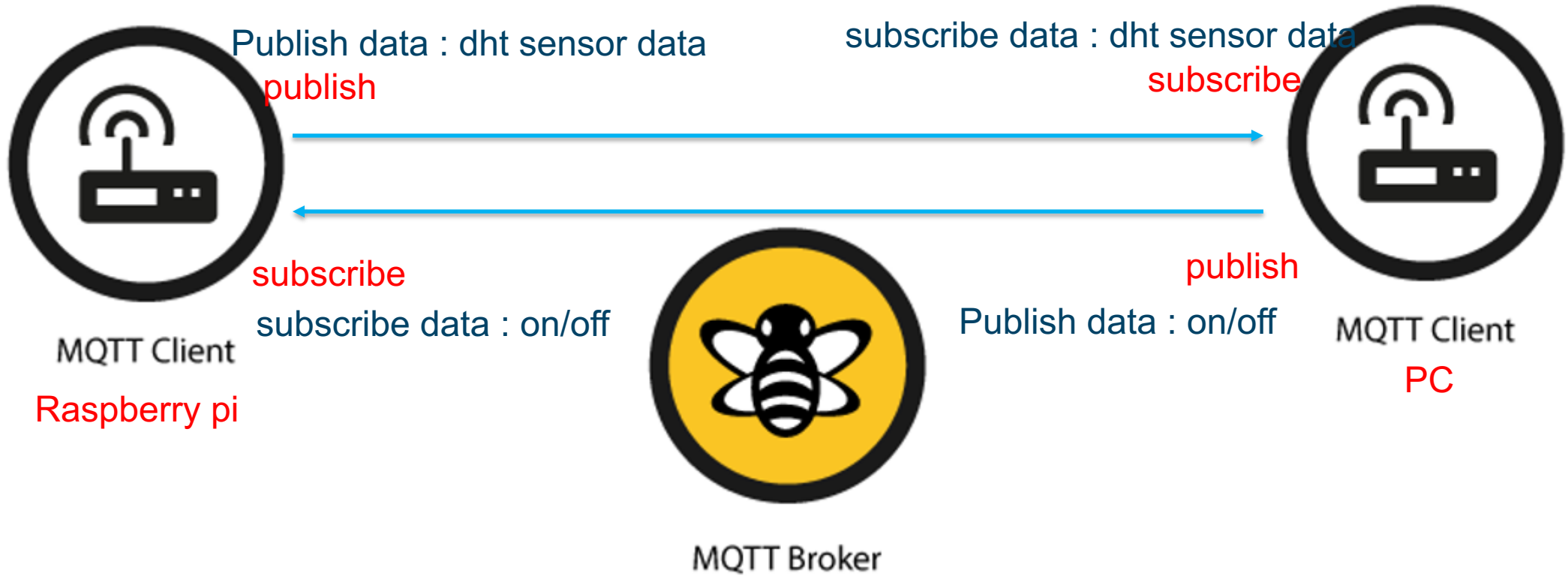


LCD 제외
led, DHT 센서만 연결!

Module	BCM	PIN
LED R	16	36
DHT	23	16

MQTT 환경

Topic = dht/CCL
MQTT Broker = PC
MQTT Host = PC ip



MQTT 이용한 Smart Home(Raspberry pi)

dht_pubsub.py

```
import paho.mqtt.client as mqtt
import time
import Adafruit_DHT as dht
import json
import RPi.GPIO as GPIO
```

```
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(16, GPIO.OUT)
```

```
dht_type = 11
dht_pin = 23
```

```
# Define Variables
```

```
MQTT_HOST = " 192.9.44.252"
```

#자신의 pc ip

```
MQTT_PORT = 1883
```

```
MQTT_KEEPALIVE_INTERVAL = 60
```

```
MQTT_TOPIC = "dht/CCL"
```

```
# Define on_publish event function
```

```
def on_publish(client, userdata, mid):
    print ("Message Published...")
```

```
def on_connect ( client, userdata , flags, rc ):
    print("Connect with result code" + str (rc))
    client.subscribe("dht/CCL")
def on_message(client, userdata, msg):
    print(msg.topic+" "+str(msg.payload.decode('utf-8')))
    if msg.payload == "on":
        GPIO.output(16, True)
    elif msg.payload == "off":
        GPIO.output(16, False)
```

```
# Initiate MQTT Client
```

```
client = mqtt.Client()
```

```
# Register publish callback function
```

```
client.on_publish = on_publish
```

```
client.on_connect = on_connect
```

```
client.on_message = on_message
```

```
# Connect with MQTT Broker
```

```
client.connect(MQTT_HOST, MQTT_PORT, MQTT_KEEPALIVE_INTERVAL)
```

```
client.loop_start()
```

```
while True:
```

```
    humidity, temperature = dht.read_retry(dht_type, dht_pin)
```

```
    if humidity is not None and temperature is not None:
```

```
        data = {'temperature':round(temperature, 1), 'humidity' : round(humidity, 1)}
```

```
        client.publish(MQTT_TOPIC, str(data))
```

```
        print('Published. Sleeping ...')
```

```
    else:
```

```
        print('Failed to get reading. Skipping ...')
```

MQTT 이용한 Smart Home(Window PC)

dht_MQ.py

```
import paho.mqtt.client as mqtt
import paho.mqtt.publish as publish
import time

normal_temp = 25.0    #원하는 온도 설정
MQTT_Broker = "192.9.44.252" #자신의 pc(broker) ip

def on_connect ( client, userdata , flags, rc ) :
    print("Connect with result code" + str (rc) )
    client.subscribe("dht/CCL") #Topic

def on_message ( client, userdata , msg ) :

    x = str(msg.payload.decode('utf-8'))    #dht 센서 데이터
    print(msg.topic + " " + x)
    y = eval(x)    #dht 센서 데이터를 Dic타입으로 변환 파싱

    if y["temperature"] > normal_temp:
        publish.single("dht/CCL", "on",hostname = MQTT_Broker)

    elif y["temperature"] <= normal_temp:
        publish.single("dht/CCL", "off",hostname =
MQTT_Broker)
```

```
def on_publish(client, userdata, mid):
    print("message publish..")

def on_disconnect(client, userdata, rc):
    print("Disconnected")

client = mqtt.Client ()
client.on_connect = on_connect
client.connect(MQTT_Broker, 1883, 60)
client.on_message = on_message
client.on_publish = on_publish
client.on_disconnect = on_disconnect
client.loop_forever()
```

Gachon
Cognitive Computing Lab

