



# IoT 개발하기 (Live PiCamera)

feat. Raspberry Pi

Solution Lab. 안창남 책임

2017.05.12





# Agenda



- 국내 IoT 서비스 종류
- 무엇을 해볼 것인가?
- IoT 개발 환경 준비하기
- Raspbian OS 설치
- Raspberry Pi Configuration
- 영상, 음성 Streaming 준비하기
- Raspberry Pi에서 PC로 영상, 음성 전송
- Android에서 영상, 음성 수신
- Android에서 영상, 음성 송신 및 수신
- 대화창 구현
- Google Play Store 등록
- Live PiCamera Demo
- 추가 개발 계획

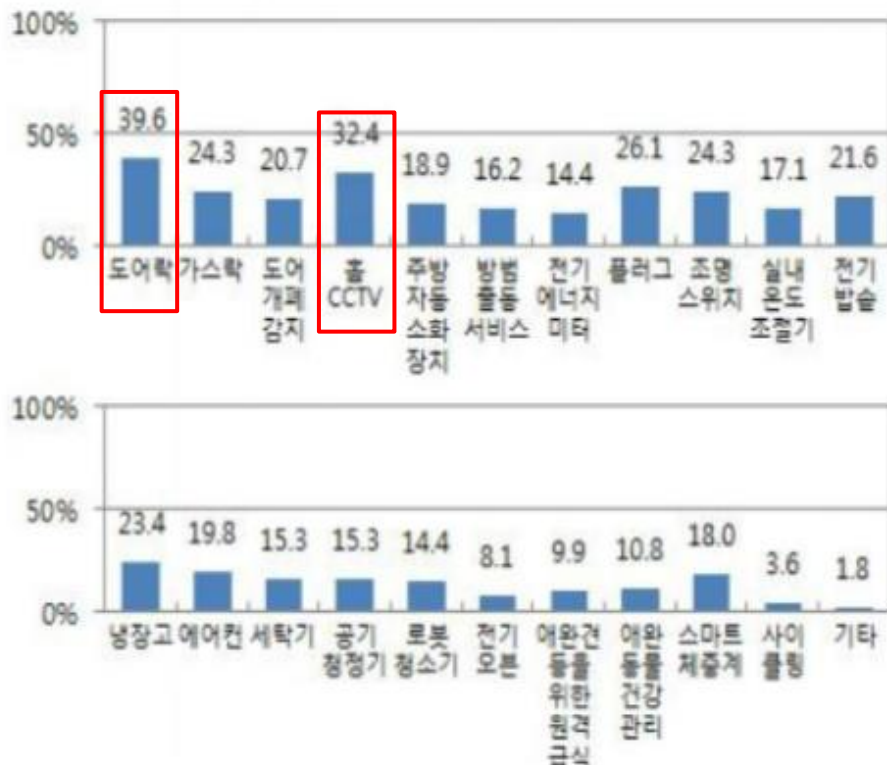


# 국내 IoT 서비스 종류

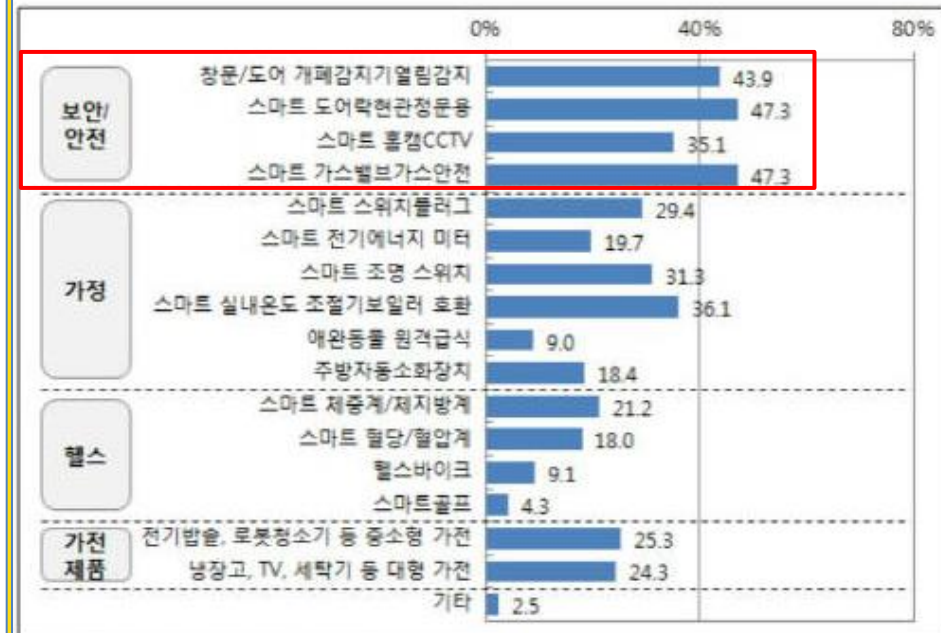


- 도어락과 홈CCTV 서비스가 높은 비중 차지
- 보안/안전 분야 서비스 사용의향이 높게 나타남

서비스 종류



사용의향 서비스



주: 스마트폰 사용자 BASE(N=1,525)임.

자료: 정보통신정책연구원(2016) 설문결과



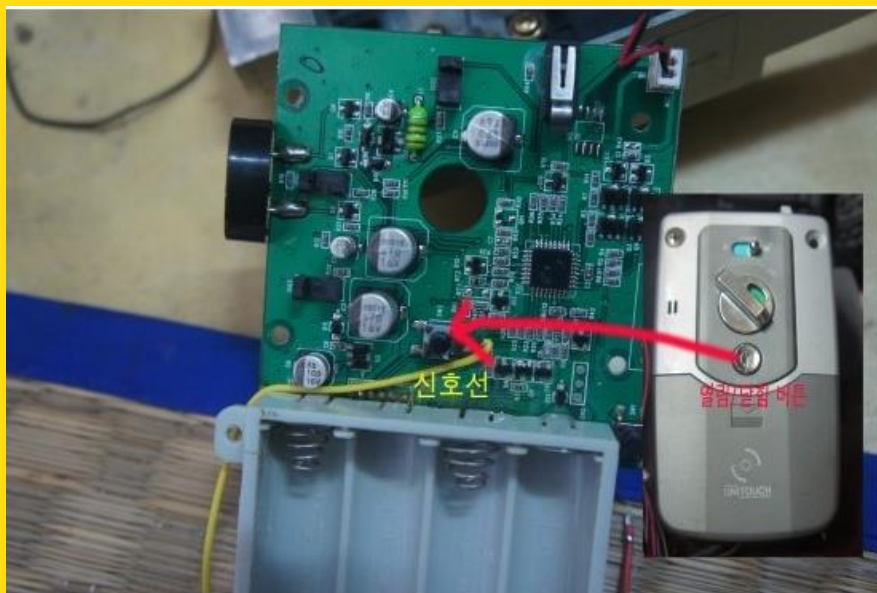
# 무엇을 해볼 것인가?



도어락 with OTP

vs.

CCTV or Streaming



- 부족한 자료
- 도어락 PCB 직접 제어 필요
- 실패에 대한 부담감 (도어락 비용?)



- 다양하고 충분한 자료
- 카메라, 사운드카드 등 추가 모듈 필요

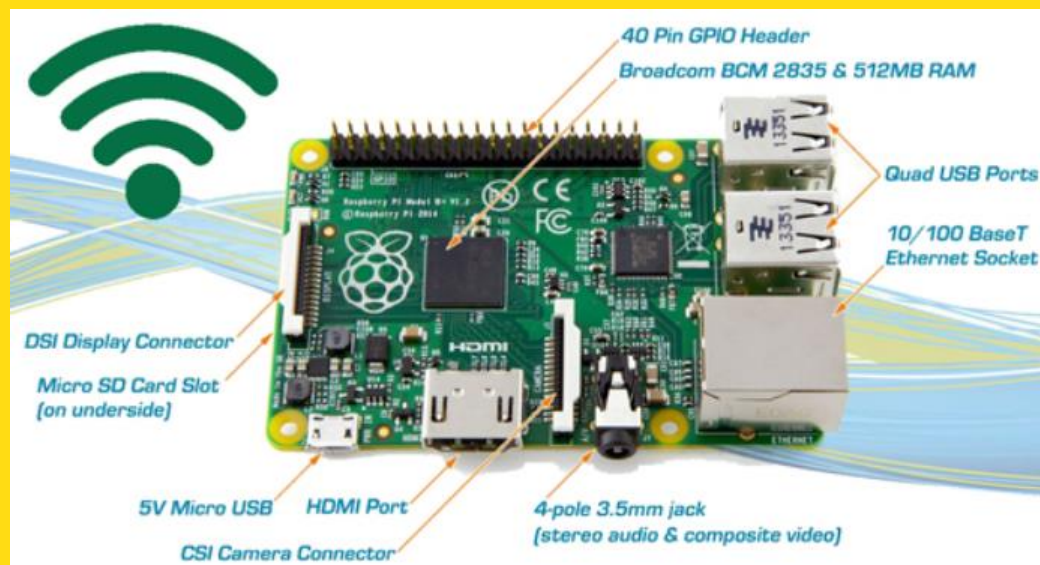


# IoT 개발 환경 준비하기



## ■ 보드 및 관련 모듈 구매

- Raspberry Pi3 Model B, Pi Camera, USB sound card (python 7.1CH), MIC (ETM-001)  
Micro SD Card 8GB



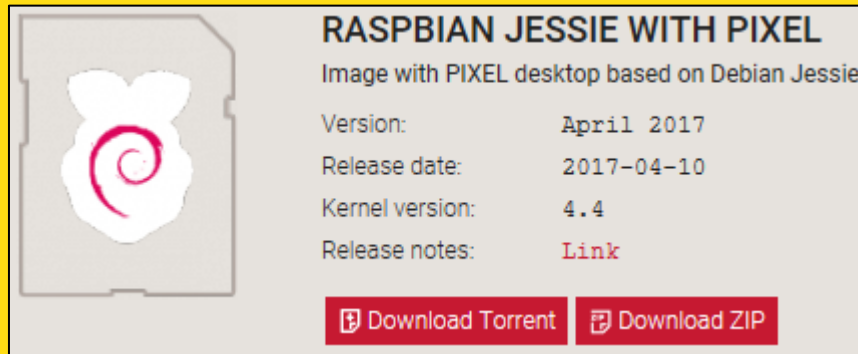
- 구매: <https://www.eleparts.co.kr/>



# Raspbian OS 설치

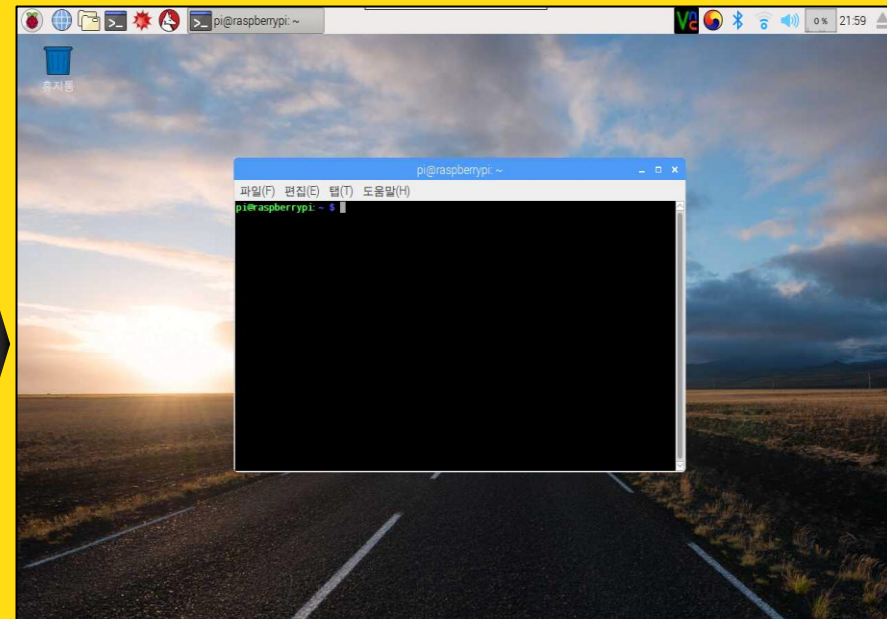
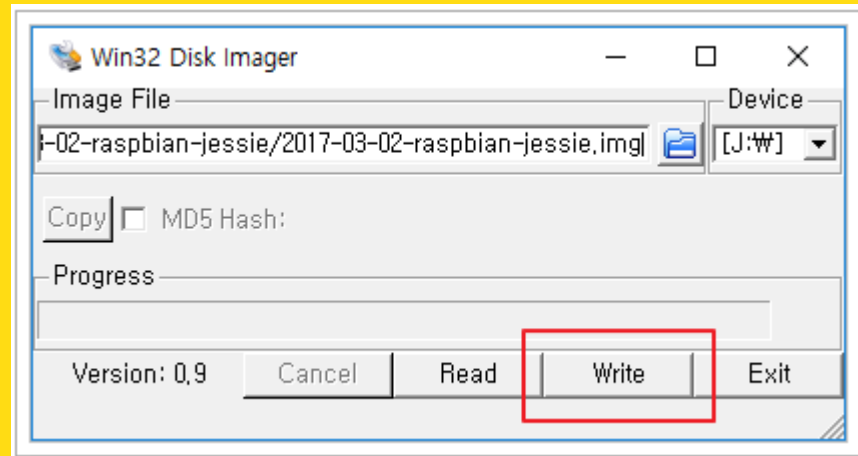
## ■ Download Raspbian OS image

- <https://www.raspberrypi.org/downloads/raspbian/>



## ■ Installing OS image using Windows

- <https://sourceforge.net/projects/win32diskimager/>



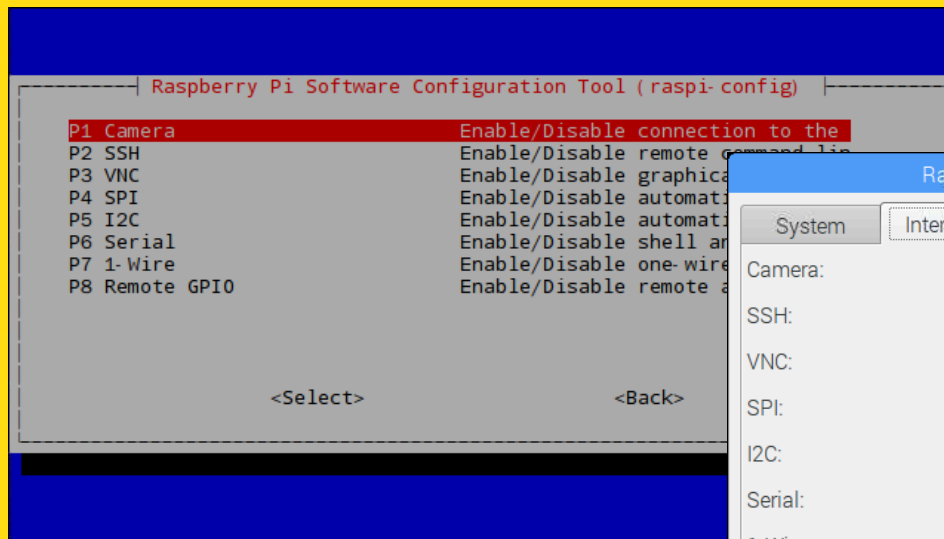
# Raspberry Pi Configuration

- Raspbian update & upgrade

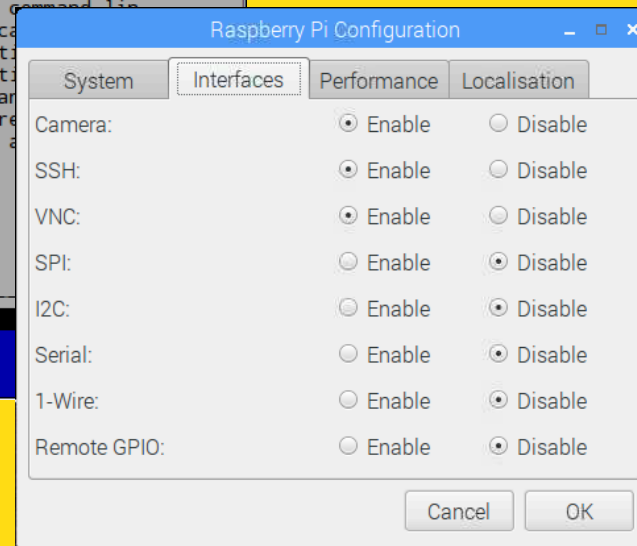
\$ sudo apt-get update

\$ sudo apt-get upgrade

- Camera, VNC, SSH enable (\$sudo raspi-config)



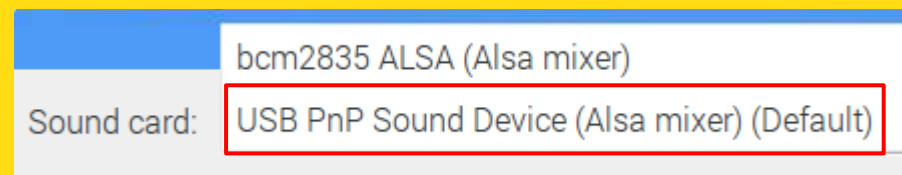
OR



- WiFi 설정



- USB Sound Card를 default로 설정





# 영상, 음성 Streaming 준비하기



## ▪ Server Solution 검토

- **MJPEG-Streamer**:jpg 촬영을 연속으로 수행해서 마치 영상처럼 보이도록 해주는 솔루션, 3~4초 지연, 음성 Streaming 지원 안됨.
- **GStreamer**: **지연 없는 영상/음성 Streaming 지원, 다양한 OS 지원**
- **UV4L (User space Video4Linux)**: encrypted live data, 지연 없는 audio and video streaming, mirroring or conferencing over the web and for the IoT, RESTful API 지원

### Raspberry Pi

#### ▪ GStreamer Installation

\$ sudo apt-get install gstreamer1.0

```
Errors were encountered while processing:  
/var/cache/apt/archives/gnome-packagekit-data_3.14.0-1_all.deb  
/var/cache/apt/archives/gnome-packagekit-session_3.14.0-1+b1_armhf.de  
b  
E: Sub-process /usr/bin/dpkg returned an error code (1)
```

>> 위 에러 발생시: \$ sudo apt-get clean 후 다시 install 실행

Gstreamer-concepts and examples





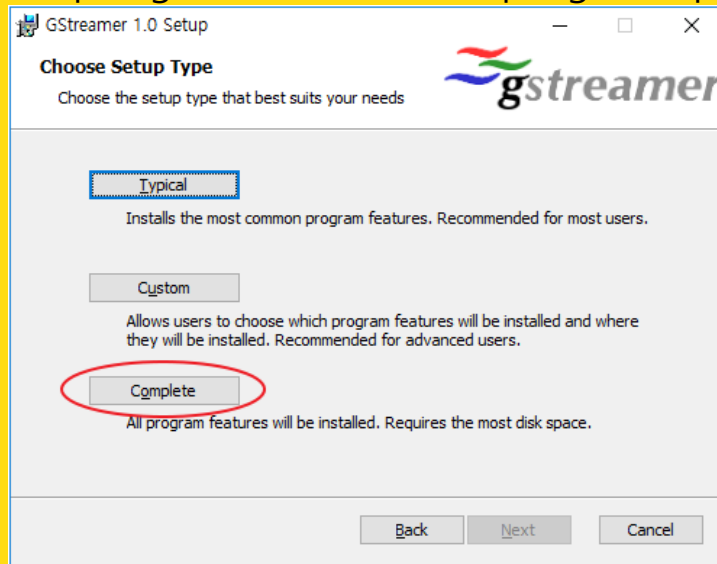
# 영상, 음성 Streaming 준비하기(Contd.)

## Windows용 GStreamer 설치

### Windows

#### Gstreamer Download & Installation

- [https://gstreamer.freedesktop.org/data/pkg/windows/1.10.4/gstreamer-1.0-x86\\_64-1.10.4.msi](https://gstreamer.freedesktop.org/data/pkg/windows/1.10.4/gstreamer-1.0-x86_64-1.10.4.msi)



- 만일 Typical로 설치 후 실행할 경우 아래와 같은 video codec 에러 발생함.

```
WARNING: erroneous pipeline: no element "avdec_h264"
```

# Raspberry Pi에서 PC로 영상, 음성 전송

## 영상, 음성 전송/수신 Pipeline 만들기

### Raspberry Pi

#### Video 전송 Pipeline

```
raspivid -n -t 0 -vf -hf -h 240 -w 320 -fps 25 -b 2000000 -o - | gst-launch-1.0 -v fdsrc ! h264parse !  
rtph264pay config-interval=1 pt=96 ! gdppay ! tcpserversink host=192.168.xxx.xxx port=5000
```

#### Audio 전송 Pipeline

```
gst-launch-1.0 -v alsasrc device=plughw:1,0 ! audio/x-raw, format=S16LE, endianness=1234,  
signed=true, rate=16000, depth=16, width=16, channels=1 ! tcpserversink host=192.168.xxx.xxx  
port=5001
```

### Windows


#### Video 수신 Pipeline

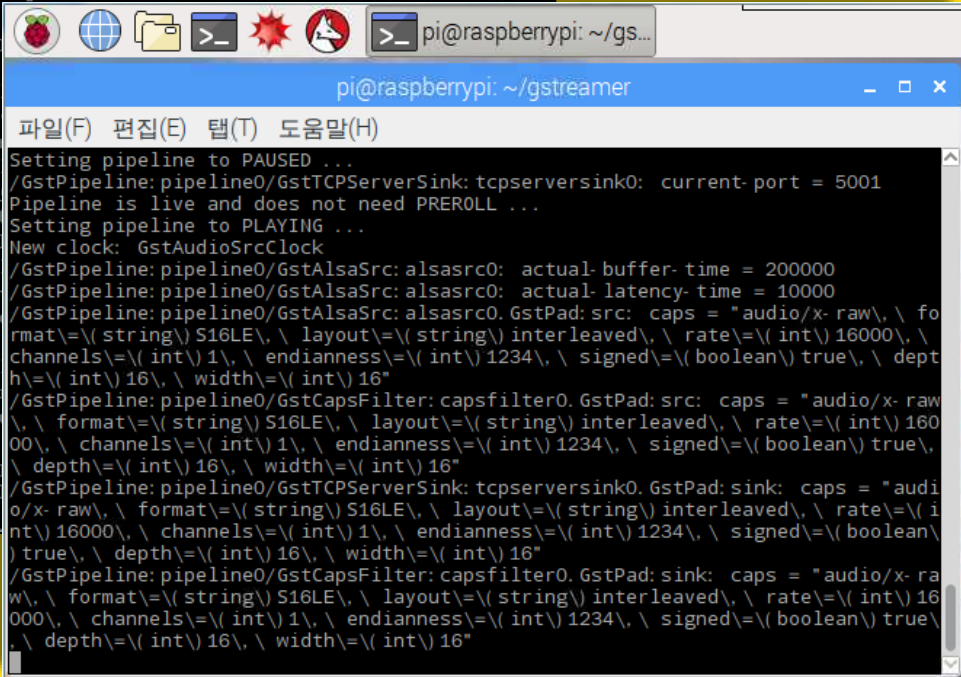
```
gst-launch-1.0 -v tcpclientsrc host=192.168.xxx.xxx port=5000 ! gdpdepay ! rtph264depay !  
avdec_h264 ! videoconvert ! autovideosink sync=false
```

#### Audio 수신 Pipeline

```
gst-launch-1.0 -v tcpclientsrc host=192.168.xxx.xxx port=5001 ! audio/x-raw, format=S16LE,  
channels=1, rate=16000, width=16 ! autoaudiosink sync=false
```

\* Video, Audio를 동시에 송/수신하기 위해서는 두 Pipeline을 '|'로 연결하여 한번에 실행해야 함.

- 





# Android에서 영상, 음성 수신



## ■ GStreamer Android tutorials

- <https://gstreamer.freedesktop.org/documentation/tutorials/android/index.html>
- 5가지 tutorial 설명만 제공 (공식 gstreamer에서는 tutorial 소스를 받을 수 없음.)

Android tutorial 1: Link against GStreamer

Android tutorial 2: A running pipeline

Android tutorial 3: Video

Android tutorial 4: A basic media player

Android tutorial 5: A Complete media player



<https://cgit.freedesktop.org/~slomo/gst-sdk-tutorials/>

## ■ Android 개발 환경 구축

- Android NDK Download
- gstreamer ndk build를 위한 binary package download (gstreamer-1.0-android-universal-1.10.4.tar.bz2)

<https://gstreamer.freedesktop.org/download/>

Android

Binaries for GStreamer on Android that integrate into the Android NDK can be found [here](#). T

iOS

- tutorial 소스가 eclipse용으로 Android Studio용으로 migration 필요  
(환경변수 경로, 빌드 옵션 등 몇 가지 빌드 에러 수정)



# Android에서 영상, 음성 수신



## ▪ Tutorial 4 소스 활용

- 구조: Java에서는 Play, Pause 등 control만 수행, Native에서 Pipeline 수행을 통해 Streaming 실행. 이를 위해 Native와 Java간 Data structure를 공유함.
- 예제 코드 이슈: 영상 또는 음성 한 가지 Pipeline에 대해서만 정상 동작
- 예제 코드 수정: native의 init()에서 영상과 음성 Data structure를 구분하여 2개의 thread로 동작하도록 수정. 단, 음성 Pipeline을 먼저 수행해야만 영상, 음성 동시 Play 됨.

```

/* Instruct the native code to create its internal data structure, pipeline and thread */
static void gst_native_init (JNIEnv* env, jobject thiz, jstring cmdAudio, jstring cmdVideo) {
    CustomData *data = g_new0 (CustomData, 1);
    SET_CUSTOM_DATA (env, thiz, custom_data_field_id_a, data);
    data->streamType = GST_AUDIO;

    CustomData *data_v = g_new0 (CustomData, 1);
    SET_CUSTOM_DATA (env, thiz, custom_data_field_id_v, data_v);
    data_v->streamType = GST_VIDEO;

    const gchar *char_uri_a = (gchar*)(env)->GetStringUTFChars (env, cmdAudio, NULL);
    const gchar *char_uri_v = (gchar*)(env)->GetStringUTFChars (env, cmdVideo, NULL);

    pipeline_a = g_strjoin ("", char_uri_a, " ! audio/x-raw, format=S16LE, rate=16000, depth=16, width=16, channels=1 ! autoaudiosink sync=false", NULL);
    pipeline_v = g_strjoin ("", char_uri_v, " ! gdpdepay ! rtph264depay ! avdec_h264 ! videoconvert ! autovideosink sync=false", NULL);

    (env)->ReleaseStringUTFChars (env, cmdAudio, char_uri_a);
    (env)->ReleaseStringUTFChars (env, cmdVideo, char_uri_v);

    GST_DEBUG_CATEGORY_INIT (debug_category, "tutorial-4", 0, "Android tutorial 4");
    gst_debug_set_threshold_for_name("tutorial-4", GST_LEVEL_DEBUG);

    GST_DEBUG ("Created CustomData at %p", data);
    data->app = (env)->NewGlobalRef (env, thiz);
    GST_DEBUG ("Created GlobalRef for app object at %p", data->app);
    pthread_create (&gst_app_thread[0], NULL, &app_function_a, data);

    GST_DEBUG ("Created CustomData at %p", data_v);
    data_v->app = (env)->NewGlobalRef (env, thiz);
    GST_DEBUG ("Created GlobalRef for app object at %p", data_v->app);
    pthread_create (&gst_app_thread[1], NULL, &app_function_v, data_v);
}

```

Java에 Native의 CustomData 공유

영상, 음성 Pipeline 생성

영상, 음성 Streaming Thread 생성

# Android에서 영상, 음성 송신 및 수신

## ▪ Raspberry Pi Camera 활용 이슈

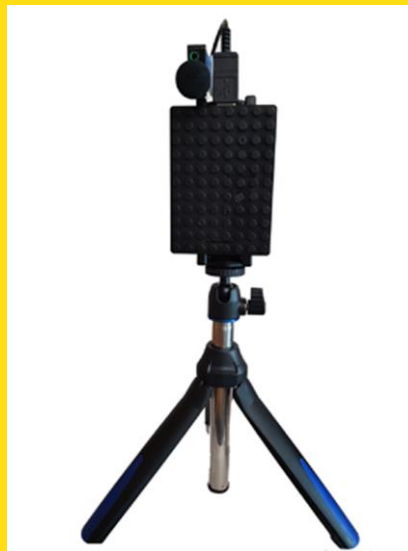
- Raspberry Pi Camera로 촬영 할 경우 고정 아이피를 사용 할 수 있는 환경이어야 함.
- Port forwarding을 위해 공유기 설정 필요 함.
- 위 두가지 조건이 충족되지 않는다면 Streaming Service를 할 수 없음.

## ▪ Raspberry PI를 중계 서버로 활용

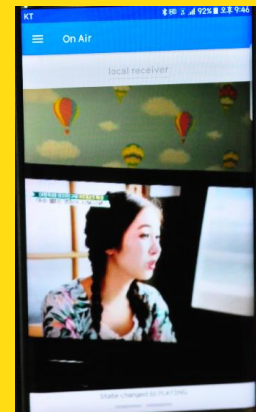
- Android 송신쪽에서 udp 통신으로 Raspberry Pi로 송신
- Raspberry Pi는 udp 통신으로 수신된 data를 src로 사용하여 tcp통신으로 송신
- Android 수신쪽은 기존과 동일하게 수신 처리



udp



tcp







# Android에서 영상, 음성 송신 및 수신



- udp 영상, 음성 송신/수신 Pipeline 만들기

## Raspberry Pi

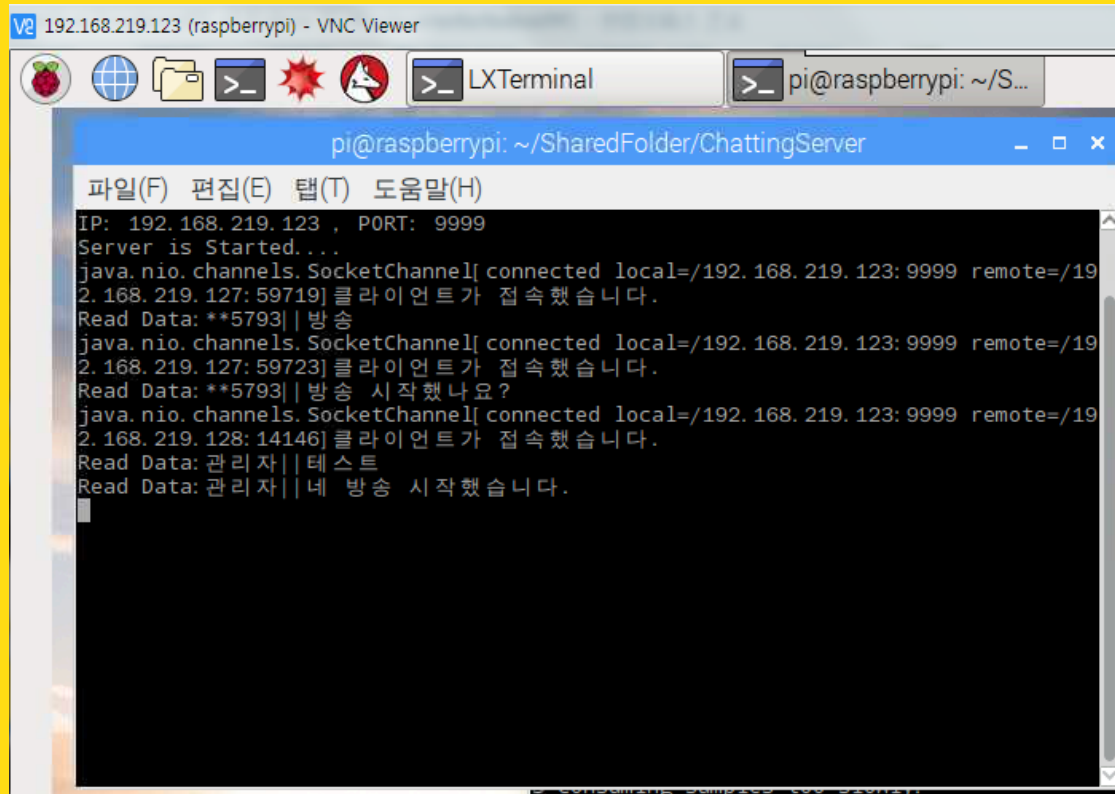
```
▪ gst-launch-1.0 -v udpsrc port=$udp_port ! h264parse ! rtph264pay pt=96 config-interval=1 !  
gdppay ! tcpserversink host=$ipadd port=$port_v | gst-launch-1.0 -v udpsrc port=$udp_port !  
audio/x-raw,format=S16LE,endianness=1234,signed=true,rate=16000,depth=16,width=16,  
channels=1 ! tcpserversink host=$ipadd port=$port_a
```

## Android 영상, 음성 송신

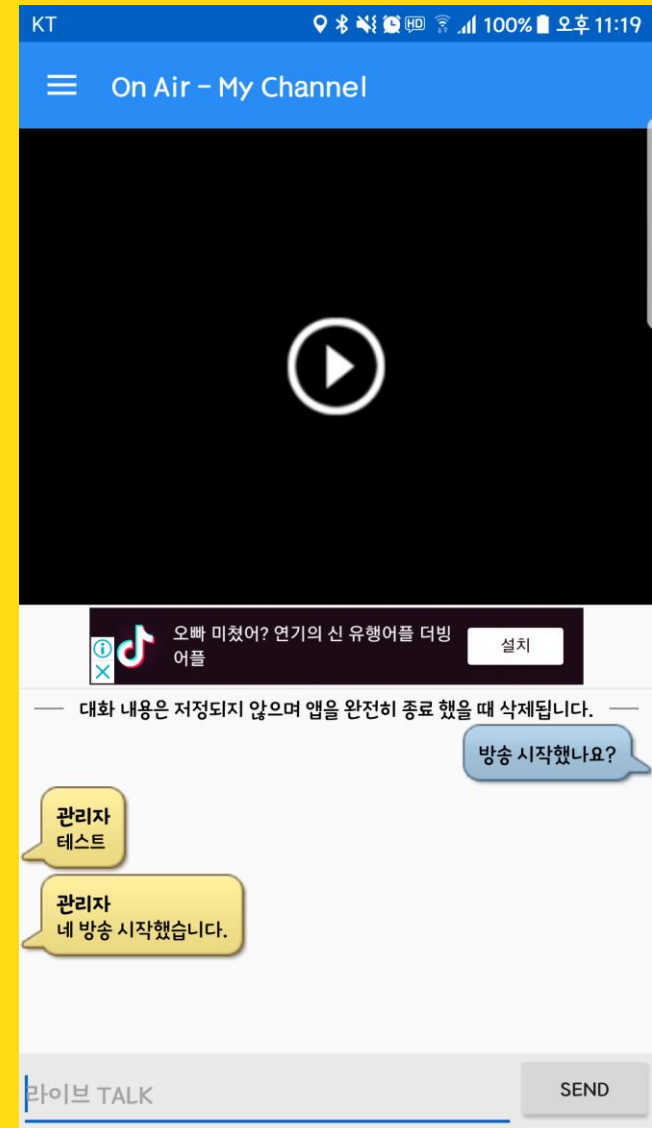
```
pipeline_a = g_strjoin ("", "openslessrc ! queue ! audio/x-raw, format=S16LE, rate=16000,  
depth=16, width=16, channels=1 ! queue ! ", char_uri_a, NULL);  
pipeline_v = g_strjoin ("", "ahcsrc device=0 ! queue ! video/x-raw, width=320, height=240 !  
tee name=t t. ! queue ! videoflip method=clockwise ! autovideosink sync=false t. ! queue !  
videoflip method=clockwise ! x264enc tune=zerolatency ! rtph264pay ! ", char_uri_v, NULL);
```

# 대화창 구현

## Java NIO를 이용한 Server, Client 구현




```
192.168.219.123 (raspberrypi) - VNC Viewer
pi@raspberrypi: ~/SharedFolder/ChattingServer
파일(F) 편집(E) 탭(T) 도움말(H)
IP: 192.168.219.123 , PORT: 9999
Server is Started...
java.nio.channels.SocketChannel[ connected local=/192.168.219.123:9999 remote=/192.168.219.127:59719] 클라이언트가 접속했습니다.
Read Data: **5793| 방송
java.nio.channels.SocketChannel[ connected local=/192.168.219.123:9999 remote=/192.168.219.127:59723] 클라이언트가 접속했습니다.
Read Data: **5793| 방송 시작했나요?
java.nio.channels.SocketChannel[ connected local=/192.168.219.123:9999 remote=/192.168.219.128:14146] 클라이언트가 접속했습니다.
Read Data: 관리자 || 테스트
Read Data: 관리자 || 네 방송 시작했습니다.
```






# Google Play Store 등록





 **Google Play**

검색

 앱

카테고리 ▾ | 홈 인기 차트 신규 출시

내 앱

쇼핑하기

< 게임

키즈

에디터 추천

계정

내 구독


코드 사용

ギフト 카드 구매

내 위시리스트

내 게임 활동

자녀 보호 안내



## Live PiCamera

irisxii 커뮤니케이션


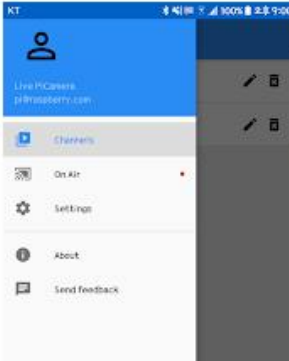


★★★★★ 7人

12

광고 포함

사용 중인 모든 기기와 호환되는 앱입니다.

설치됨

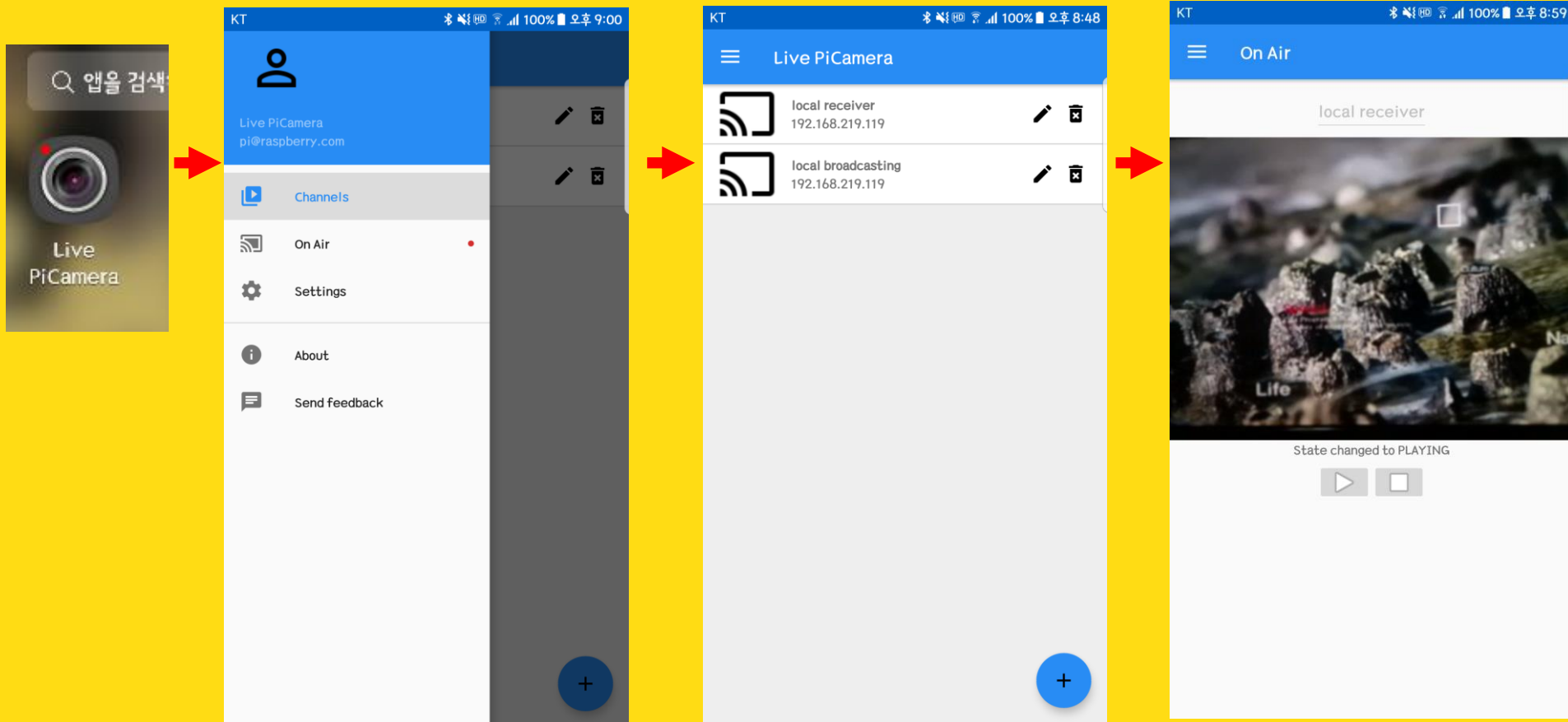




# Live PiCamera Demo



- 외부에서 접속하기 위해 공유기 port forwarding 필요
- Android 영상, 음성 수신 모드

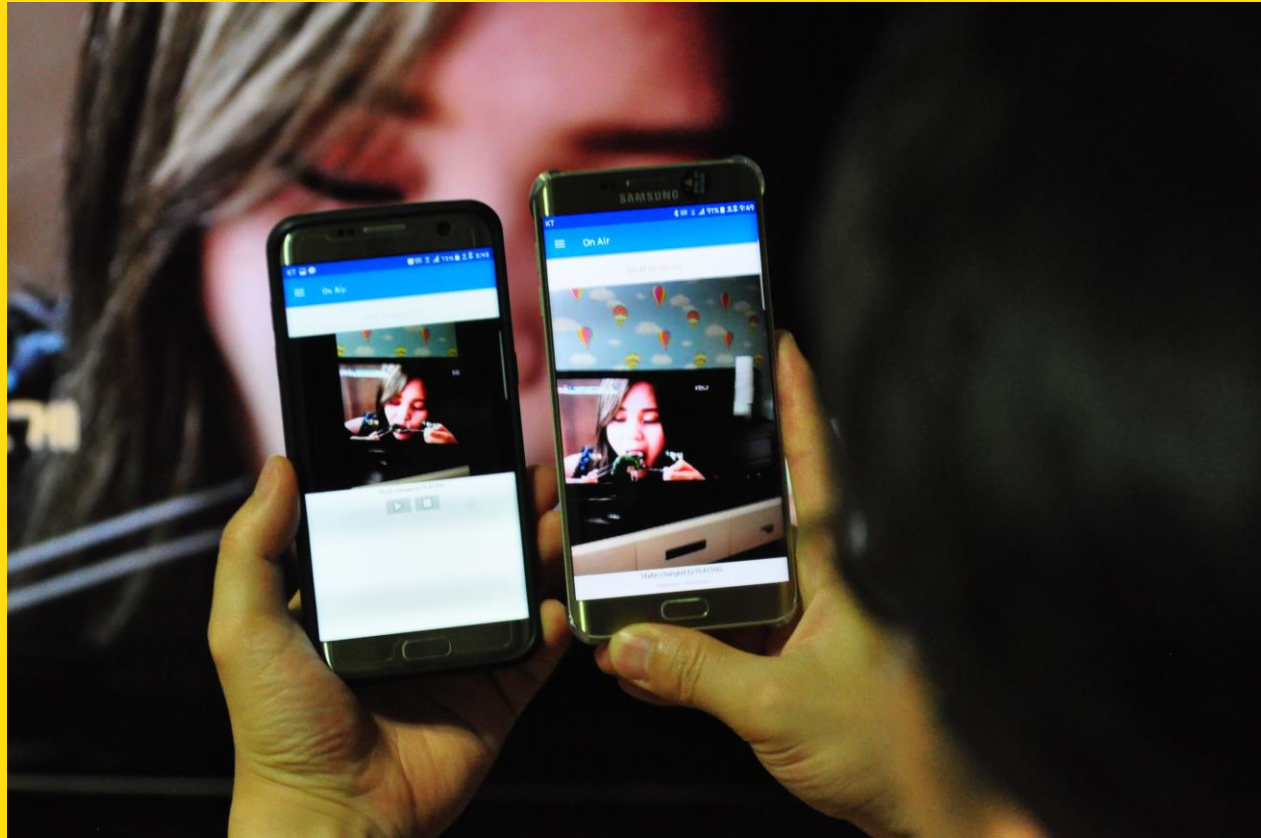
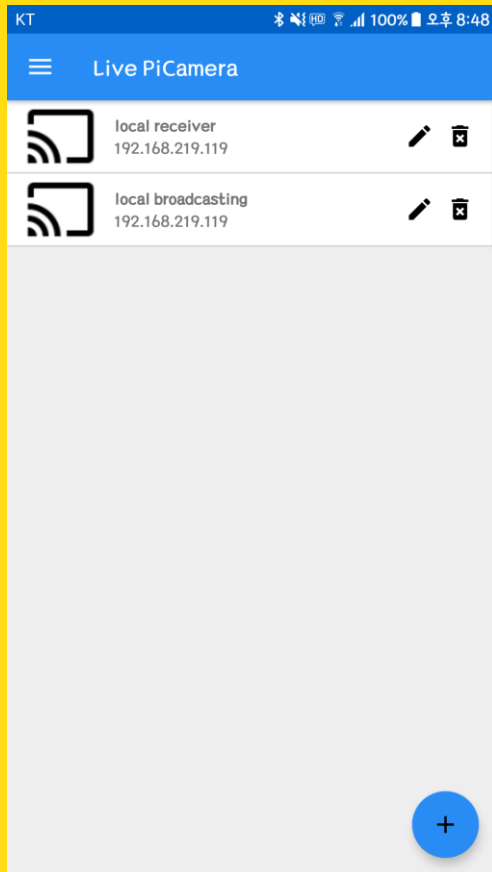


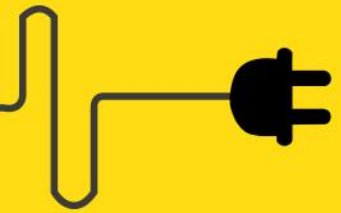


# Live PiCamera Demo

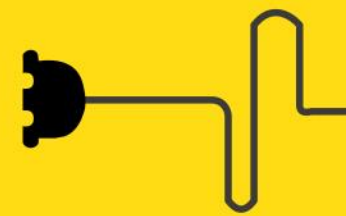


## ■ Android간 영상, 음성 송/수신 모드





# 추가 개발 계획



- 구글 OTP 또는 Firebase를 이용한 인증처리
- FCM (Firebase Cloud Messaging)을 이용한 푸시알람 구현
- Streaming 영상 저장 및 특정 사이트 또는 클라우드에 업로드
- OpenCV를 이용한 얼굴 인식