

# 2019년 데이터통신

- HW 03 -

제출일자	2019.03.08
이름	노효근
학번	201502049
분반	02

## \* 과제 목표

- apk파일을 설치하여 어플을 이용하여 소리로 데이터를 전송하고 받는 방법을 이해한다.

## \* decode.py

- complete.apk 파일을 이용하여 소리로 데이터를 전송하고 받는다.  
complete.apk 에서 작동하는 decode.py 파일을 확인하여 보면  
기본적으로 설정되어있는 HANDSHAKE\_START\_HZ = 4096 , HANDSHAKE\_END\_HZ = 5120+1024로 되어있다. HANDSHAKE\_START\_HZ는 decode 파일이 실행되고, 소리를 받는 시작 주파수를 의미하고 HANDSHAKE\_END\_HZ는 소리를 받고 끝내는 주파수를 의미한다. 오차범위 +20Hz로 실행이 가능하다. HANDSHAKE\_START\_HZ를 받게 되면 in\_paket이 True로 바뀌며, 소리를 받는 함수가 실행한다. 아스키코드에 맞는 Hz를 맞게 되면, 해당 Hz를 파악하여 문자로 해석하여 화면에 출력하게 해준다.

## \* decode\_1.py

- 1-release.apk 파일을 이용하여 소리로 데이터를 전송하고 받는다.  
1-release.apk 에서 작동하는 decode\_1.py 파일을 만든다.  
기본적으로 설정되어있는 HANDSHAKE\_START\_HZ = 4096 , HANDSHAKE\_END\_HZ = 5120+1024로 되어있다. HANDSHAKE\_START\_HZ는 decode 파일이 실행되고, 소리를 받는 시작 주파수를 의미하고 HANDSHAKE\_END\_HZ는 소리를 받고 끝내는 주파수를 의미한다. 오차범위 +20Hz로 실행이 가능하다. dom을 통해 어플에서 보내는 Hz를 확인하여 해당 decode 파일을 실행 할 수 있는 HANDSHAKE\_START\_HZ와 HANDSHAKE\_END\_HZ를 찾아본다.  
그리하여 시작하는 HANDSHAKE\_START\_HZ는 1024Hz 이었고, HANDSHAKE\_END\_HZ는 1024+512Hz로 확인 할 수 있었다. 그리하여 시작 주파수와 종료 주파수를 바꾸고 실행하니 다음과 같이 화면에 문자가 출력되는걸 확인 할 수 있었다.



## ★ decode\_2.py

- assignment2.apk 파일을 이용하여 소리로 데이터를 전송하고 받는다.

assignment2.apk 에서 작동하는 decode\_2.py 파일을 만든다.

기본적으로 설정되어있는 HANDSHAKE\_START\_HZ = 4096 , HANDSHAKE\_END\_HZ = 5120+1024로 되어있다. HANDSHAKE\_START\_HZ는 decode 파일이 실행되고, 소리를 받는 시작 주파수를 의미하고 HANDSHAKE\_END\_HZ는 소리를 받고 끝내는 주파수를 의미한다. 오차범위 +20Hz로 실행이 가능하다.

dom을 통해 어플에서 보내는 Hz를 확인하여 해당 decode 파일을 실행 할 수 있는 HANDSHAKE\_START\_HZ와 HANDSHAKE\_END\_HZ를 찾아본다. 그리하여 시작하는 HANDSHAKE\_START\_HZ는 4096Hz 이었고, HANDSHAKE\_END\_HZ는 5120+512Hz로 확인 할 수 있었다. 그리하여 시작 주파수와 종료 주파수를 바꾸고 실행하였지만, 화면에 문자가 출력되지 않았다.

해당 apk 파일은 다른 apk파일과 달리 소리를 더 길게 전달하였고, 주파수는 그의 절반이었다.

sin함수의 1주기는 4비트이지만, 이 apk파일은 소리의 주파수를 절반으로 보내는 형태이고 따라서 1주기에 2비트만 보내는 것으로 알 수 있다.

이후 extract\_packet 함수를 확인하여 보면, Hz를 freqs를 이용하여 받는다. bit\_chunks를 이용하여 받은 Hz를 bytearray에 값을 넣어, 문자열로 출력하여준다. 위의 결과들을 종합하면, 1주기에 2비트를 전송하므로 BITS의 수를 4에서 2로 변경하여 값을 출력하도록 한다.



## ★ 실행결과 영상

- decode\_1 : <https://youtu.be/uR7vr7tKYdl>
- decode\_2 : [https://youtu.be/J8f1qD\\_iTDs](https://youtu.be/J8f1qD_iTDs)