

2025년 1학기 데이터통신 3주차 과제 보고서

- 모스부호 변환 및 해석 -



202001156 정보통계학과 김수영

과제2. 해결 방법

```
def audio2file(audio,filename):
    with wave.open(filename, 'wb') as w:
        w.setnchannels(1)
        w.setsampwidth(4)
        w.setframerate(48000)
        for a in audio:
            w.writeframes(struct.pack('<l', a))

def file2morse(filename):
    with wave.open(filename, 'rb') as w:
        audio = []
        framrate = w.getframerate()
        frames = w.getnframes()
        for i in range(frames):
            frame = w.readframes(1)
            audio.append(struct.unpack('<i',frame)[0])
        morse = ''
        unit = int(0.1*48000)
        for i in range(1,math.ceil(len(audio)/unit)+1):
            stdev = statistics.stdev(audio[(i-1)*unit:i*unit])
            if stdev > 10000:
                morse = morse + '.'
            else:
                morse = morse + ' '
        morse = morse.replace('...', '-')
    return morse
```

audio2file(), file2morse()는 실습 과제에 있는 것을 활용하여 구현하였다. 하지만 file2morse의 출력 형식을 살펴보면 다음과 같다.

A screenshot of a Windows PowerShell terminal window. The title bar shows "Windows PowerShell". The command prompt displays the following commands and output:

```
PS F:\학교\4학년\1학기\데이터통신\과제\3주차\assignment> python .\202001156_morse_converter.py decoding  
Input wav file name to decode: output_202001156_김수영.wav  
Original Unit: . _ . _ . _ . _ . _ . _ . _ . _ . _ .
```

file2morse()의 연산은 해당 wav 파일의 유닛을 구하는 것이다. 즉, 국제 표준 적용인

- dot: 1 unit
- dash: 3 units
- dot dash 사이: 1 unit
- 문자 사이: 3 units
- 단어 사이 (공백, 스페이스): 7 units

이 규칙을 표현한 것이다. 일반적인 모스부호는 ‘·’, ‘-’, ‘/’, ‘.’ 4가지로 구분하여 표현되기에 이러한 모스부호의 형식을 맞춘 후 제대로 변환된 모스부호 string을 탐색하여 이를 dictionary와 대조한 뒤 decoding을 수행하는 것이 적절할 것이라 판단하였다.

첫번째로 공백으로 구분되어 있는 문자는 사실 ‘-.’ 이나 ‘.’ 처럼 원래는 붙어 있는 문자열로 이는 각 문자 하나를 의미하기에 적절하게 치환을 해주어야 한다. 다행히 dot와 dash 간 유닛은 1unit 즉, 공백 하나를 의미하고 문자와 문자 간 공백은 3unit 차이가 나므로 이러한 사실을 이용하여 문자별 모스부호를 구분할 수 있다.

```
91 def unit2text(input):
92     output = ""
93     for i in range(0, len(input), 2):
94         if input[i] == ".":
95             output = output + "."
96         elif input[i] == "-":
97             output = output + "-"
98         elif input[i] == " ":
99             output = output + " "
100     return output
```

[illegible]

해당 코드를 활용하여 유닛 단위로 조각난 문자를 구별해냈다. 하지만 아직 공백 즉, ‘/’에 해당하는 부분은 적절한 변환이 이루어지지 않았으므로 이를 변환시킬 코드를 다시 작성해준다. 공백 즉, ‘/’는 7unit을 간격으로 사용하고 unit2text()를 통해 3unit으로 압축되었으므로 이를 활용해주면 된다.

```
102 def tab2slash(input):
103     input_list = list(input)
104     for i in range(0, len(input_list)-1):
105         if input_list[i] == " " and input_list[i+1] == " ":
106             input_list[i+1] = "/"
107     return "".join(input_list)
```

[illegible]

이러한 코드를 통해 이제 형식에 맞는 모스부호를 얻었다. 남은 것은 형식화된 모스부호를 올바르게 디코딩 하는 코드를 짜기만 하면 인코딩된 모스부호를 원래 문자열로 바꿀 수 있다.

검증

1. 모스부호 복호화 검증

```
Windows PowerShell
```

```
PS F:\학교\4학년\1학기\데이터통신\과제\3주차\assignment> python .\202001156_morse_converter.py decoding  
Input wav file name to decode: output_202001156_김수영.wav  
Original Unit: . - . - . - . - . - . - . - . - . -  
  
Converted Unit: .- -. -. -. --. . -.. --... . -.  
  
Morse Code: .- -. -. -. --. . / -. ..-- --... . -.  
  
Text: ARRANGE NUMBER  
PS F:\학교\4학년\1학기\데이터통신\과제\3주차\assignment>
```

Morse Code를 복사하여 넣어보면

Morse Code Translator

International Morse
Translator
Machine
Training

T-Shirts, Mugs, Clocks and Cards

Visit the Morse Code World Shop for a unique gift

Input:

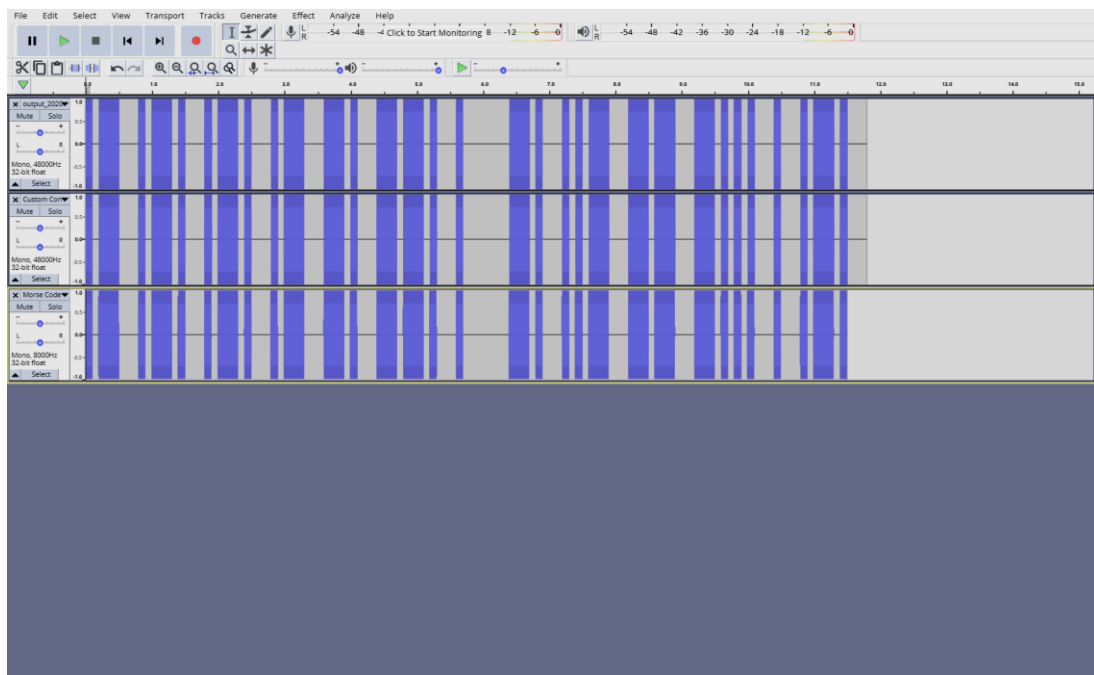
.. .- .- .- .- -. -. . / -. .- .- -. .- .- .-

Output:

ARRANGE NUMBER

ARRANGE NUMBER로 변환되는 것을 확인할 수 있다. 즉, 복호화는 정상적으로 작동함을 확인할 수 있다.

2. 파형 검증



마지막으로 정말 해당 wav파일이 ARRANGE NUMBER가 맞는지 확인하기 위해 원본 파일의 파형 (맨 위 파형), ARRANGE NUMBER를 커스텀 변환기에 넣었을 때의 파형 (두 번째 파형), ARRANGE WORLD를 모스 코드 월드에서 얻은 파형 (세 번째 파형)을 비교한 것이다. 모두 유닛의 간격이 일치함을 시각적으로 검증할 수 있다. 즉, ARRANGE WORLD가 올바른 문장이란 것을 확인할 수 있었으며 커스텀 변환기 역시 제대로 동작함을 확인할 수 있다.