

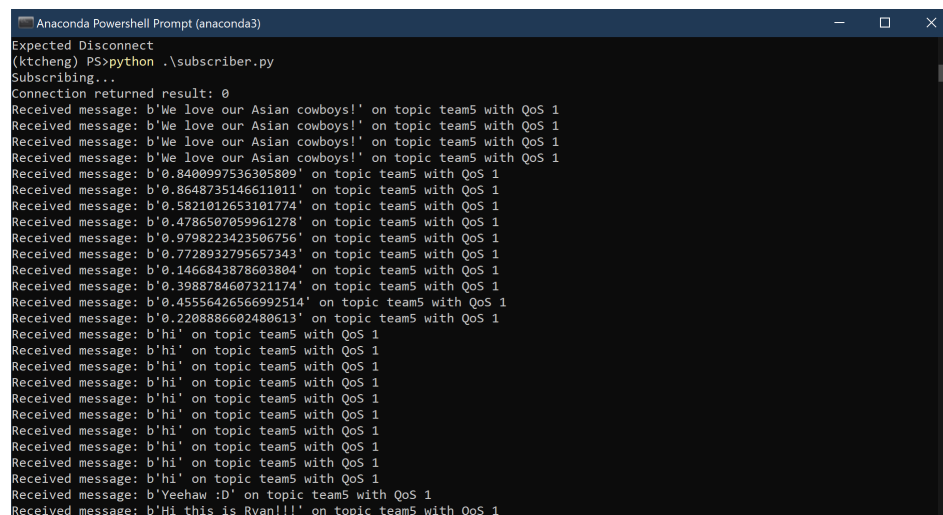
EC ENGR 180DA, Winter 2022: Kellen Cheng (905155544)
Warm-up Lab (Partners: Ryan Doan, Newton Yee, Grace Zhao)
Due 3 February, 2022 at 6:00 PM

1 Communication Speech Tasks

- (a) **Solution:** For reference regarding task 1 (and the rest of the subsequent tasks), refer to the Github repository that was created in lab 1. As a refresher, here is the link:

<https://github.com/ktcheng/180DA-WarmUp>

- (b) **Solution:** For reference regarding the MQTT task, please see the screenshot below of the terminal, showing the ability for our entire group of 4 to be able to receive and publish messages to our own custom topic, titled: "team5". A consideration of the MQTT module is the ability we now have for our project to potentially use communications to a better effect. Combined with technology like speech processing, it allows possible the ability to publish messages from individual players to a game "server", which opens up many different ways of setting up our design project. One difficulty of using MQTT is that the messages that we sent had to be done via running code in the terminal, and if we wanted to use MQTT in the future, it would be worthwhile automating this procedure. Incorporating something like this might need some leeway of a couple seconds for messages to transfer and signal as received on our device. Communications choice for our project will most likely depend on the complexity and lag time tradeoff.



```
Anaconda Powershell Prompt (anaconda3)
Expected Disconnect
(kcheng) PS>python .\subscriber.py
Subscribing...
Connection returned result: 0
Received message: b'We love our Asian cowboys!' on topic team5 with QoS 1
Received message: b'We love our Asian cowboys!' on topic team5 with QoS 1
Received message: b'We love our Asian cowboys!' on topic team5 with QoS 1
Received message: b'We love our Asian cowboys!' on topic team5 with QoS 1
Received message: b'0.8400997536305809' on topic team5 with QoS 1
Received message: b'0.8648735146611011' on topic team5 with QoS 1
Received message: b'0.5821012653101774' on topic team5 with QoS 1
Received message: b'0.4786507059961278' on topic team5 with QoS 1
Received message: b'0.9798223423506756' on topic team5 with QoS 1
Received message: b'0.7728932795657343' on topic team5 with QoS 1
Received message: b'0.1466843878603804' on topic team5 with QoS 1
Received message: b'0.3988784607321174' on topic team5 with QoS 1
Received message: b'0.45556426566992514' on topic team5 with QoS 1
Received message: b'0.2208886602480613' on topic team5 with QoS 1
Received message: b'hi' on topic team5 with QoS 1
Received message: b'hi' on topic team5 with QoS 1
Received message: b'hi' on topic team5 with QoS 1
Received message: b'hi' on topic team5 with QoS 1
Received message: b'hi' on topic team5 with QoS 1
Received message: b'hi' on topic team5 with QoS 1
Received message: b'hi' on topic team5 with QoS 1
Received message: b'hi' on topic team5 with QoS 1
Received message: b'hi' on topic team5 with QoS 1
Received message: b'hi' on topic team5 with QoS 1
Received message: b'Yeehaw :D' on topic team5 with QoS 1
Received message: b'Hi this is Ryan!!!' on topic team5 with QoS 1
```

- (c) **Solution:** For reference regarding the speech processing task, please see the screenshot below of the terminal, which displays the ability of Python speech recognition using the "Sphinx" library. Below illustrates one such example using rhyming words: "cat", "rat", "bat", as well as the performance of the algorithm. Note that with the third word, there was background white noise inserted (phone playing the sound of rain).

```
Anaconda Powershell Prompt (anaconda3)
(base) PS> .\speech.py
(base) PS>python .\speech.py
say something!
sphinx could not understand audio
(base) PS>python .\speech.py
say something!
sphinx thinks you said cat
(base) PS>python .\speech.py
say something!
sphinx thinks you said rat
(base) PS>python .\speech.py
say something!
sphinx thinks you said that
(base) PS>_
```

2 Python Code

- (a) **Solution:** For code regarding the MQTT task, please reference "subscriber.py" and "publisher.py".
For code regarding the speech recognition task, please reference "speech.py".