

IoT Milestone 2

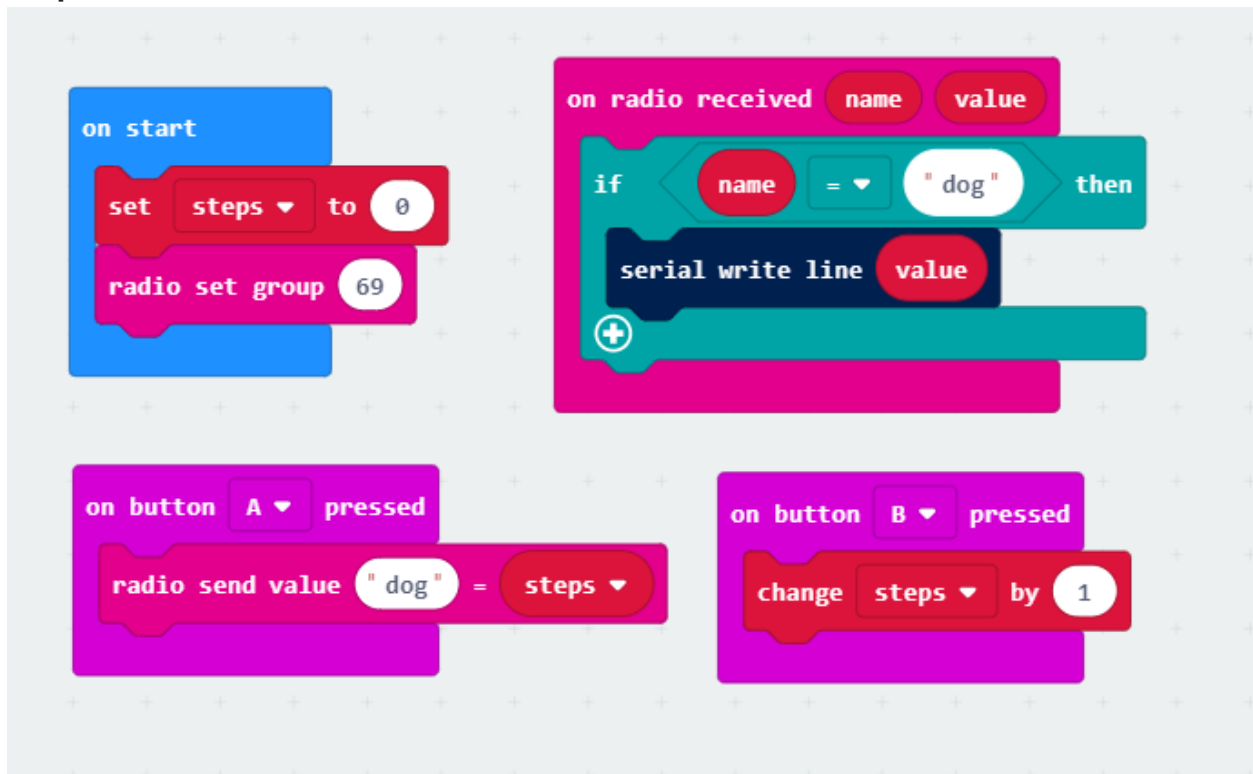
Step 1:

We added a password to the beginning of the message that we wanted to display on the other microbit. When the message was broadcasted to other microbits, the other microbits checked to see if the front of the message equaled the password we set. If the password matched, it would display the message that is after the password. If the password was not correct, the message would not show. By adding this password, we resolved the conflicts. We can do this with the same code on both microbits. We just have to make sure the correct passwords are placed to the front of both messages. We can also do this with different codes, as long as the same password is placed to the front of the codes. We prefer to use python over javascript because we are more familiar with python, and it is easier to use.

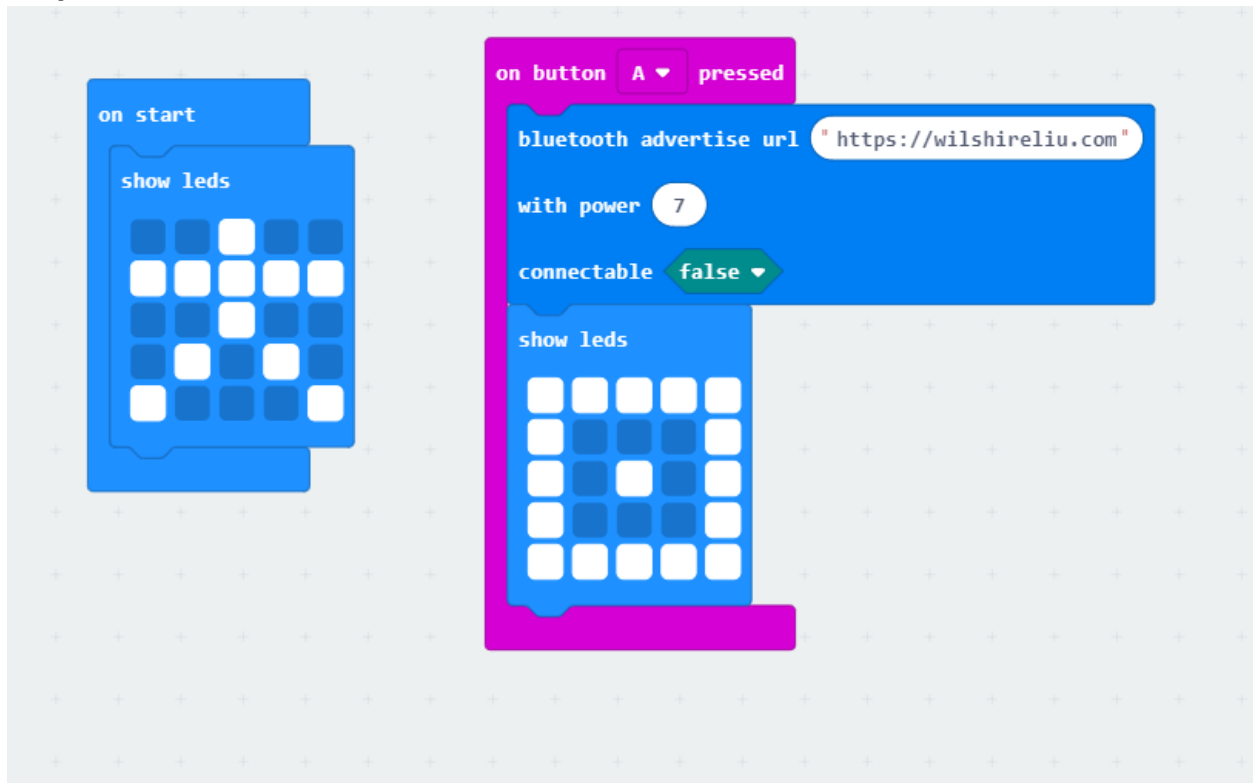
Step 2:

The baud rate is the rate that information is transferred in one communication channel. Higher baud rate means faster speed of transferring information. For example, 5000 baud rate states that 5000 bits of information is transferred in a second through the channel.

Step 3:



Step 4:



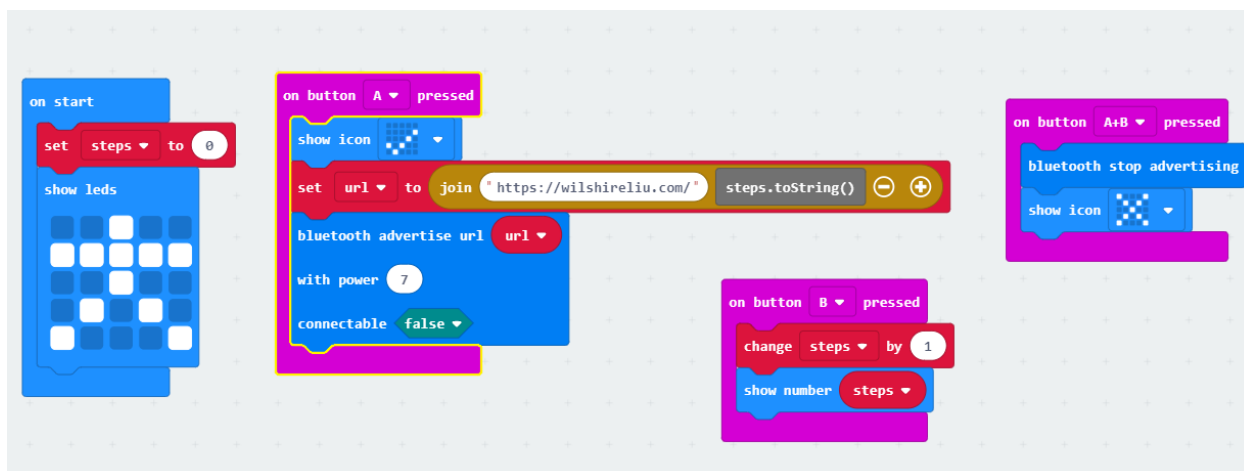
Step 5:

The script first tries to get the event loop: If there is no current event loop set in the current OS thread and `set_event_loop()` has not yet been called, `asyncio` will create a new event loop and set it as the current one. It then creates a Bluetooth socket and creates a connection with that socket. It then starts running the event loop. Inside this loop, every time it receives a EddyStone beacon, it calls the `_process_packet` method and then decodes that beacon and prints it out in the format that is specified in the method. The loop "runs forever" until a keyboard interrupt happens which will stop the scanning.

Step 6:

We need to solve the issue of continuously sending out messages. We made a statement to stop the advertising when buttons A+B are pressed. When the microbit starts advertising the eddybeacon, a check mark will display on the microbit display. And then the terminal of the Raspberry Pi will show the URL + step count based on how many times we clicked button B on the microbit. The microbit will stop advertising once we press A+B, and the microbit display will show a cross mark. Below are the screenshots.

Wilshire Liu
Stanley Wen
Yunwei Yan



```
192.168.4.1 (raspberrypi) - VNC Viewer
pi@raspberrypi: ~/Desktop
File Edit Tabs Help
60, 'url': 'https://wilshireliu.com/4'}
Google beacon: {'mac address': 'de:82:79:56:6f:d6', 'tx_power': -10, 'rssi': -
61, 'url': 'https://wilshireliu.com/4'}
Google beacon: {'mac address': 'de:82:79:56:6f:d6', 'tx_power': -10, 'rssi': -
59, 'url': 'https://wilshireliu.com/4'}
Google beacon: {'mac address': 'de:82:79:56:6f:d6', 'tx_power': -10, 'rssi': -
59, 'url': 'https://wilshireliu.com/5'}
Google beacon: {'mac address': 'de:82:79:56:6f:d6', 'tx_power': -10, 'rssi': -
59, 'url': 'https://wilshireliu.com/6'}
Google beacon: {'mac address': 'de:82:79:56:6f:d6', 'tx_power': -10, 'rssi': -
56, 'url': 'https://wilshireliu.com/6'}
Google beacon: {'mac address': 'de:82:79:56:6f:d6', 'tx_power': -10, 'rssi': -
48, 'url': 'https://wilshireliu.com/6'}
Google beacon: {'mac address': 'de:82:79:56:6f:d6', 'tx_power': -10, 'rssi': -
60, 'url': 'https://wilshireliu.com/7'}
Google beacon: {'mac address': 'de:82:79:56:6f:d6', 'tx_power': -10, 'rssi': -
49, 'url': 'https://wilshireliu.com/7'}
Google beacon: {'mac address': 'de:82:79:56:6f:d6', 'tx_power': -10, 'rssi': -
44, 'url': 'https://wilshireliu.com/7'}
Google beacon: {'mac address': 'de:82:79:56:6f:d6', 'tx_power': -10, 'rssi': -
47, 'url': 'https://wilshireliu.com/7'}
Google beacon: {'mac address': 'de:82:79:56:6f:d6', 'tx_power': -10, 'rssi': -
67, 'url': 'https://wilshireliu.com/7'}
Google beacon: {'mac address': 'de:82:79:56:6f:d6', 'tx_power': -10, 'rssi': -
64, 'url': 'https://wilshireliu.com/7'}
```