

I. Content

Use Mbed to read the sensor value from 3D Accelerator and 3D gyro. Send the value by json format to Socket Server through Wifi. Use Python to open the Socket Server and visualize the received values.

II. Discussion

1. Mbed OS version

Mbed OS version needs to be 6.15.0, so that the sensor can operate successfully.

2. Wifi and Socket Interface

Python serves as the Socket server and Mbed serves as the socket client. The two need to connect under the same Wifi, so that the communication can be reached to one another successfully.

Check *\$ipconfig* in the terminal, we can get the IP address of the host under this Wifi. Since the server IP and the port is already known, we can directly use *socketaddress.set_ip_address* and *socketaddress.set_port* to connect to the desired target.

4. Read from sensor

Since the module has been simplified for memory concern, it only supports minimal *printf()* function, which may limit the usage of the floating point. However, we need the sensor value to be in floating point format. Thus, some code associating *BufferSerial.h* is added and the *print_lib* in *targets.json* is also modified to “*std*”. In other words, we need to enable *printf()* to support the floating point format.

5. Json format

After reading the sensor value from 3D Accelerator and 3D gyro, we use *sprintf()* to enclose the value in json format. This would help simplify the code in python. The socket would transfer the data in string.

6. Max message length

The *MAX_MESSAGE_RECEIVED_LENGTH* needs to be set according to the message length. In this case, our json string length is approximately 200. If the variable is set less than the length of the actual sent message, it might result in memory management fault. This error happens because the excess bit might use some memory which should

not be used.

6. send the message

We use TCPsocket to transmit the data. Port numbers from 0 to 1023 are reserved for common TCP/IP applications. So, we set the port number to 6531.

III. Result

1. on Mbed terminal

```
Starting socket demo
Connecting to the network...
Socket connect to 172.20.10.3 || Port: 6531
Start sensor init

Loop 0

GYRO_X = -70.00
GYRO_Y = -70.00
GYRO_Z = -70.00

ACCELBRO_X = -11
ACCELBRO_Y = -1
ACCELBRO_Z = 97
sent 197 bytes

Loop 1
```

```
ACCELBRO_X = 121
ACCELBRO_Y = -832
ACCELBRO_Z = -544
sent 205 bytes

Loop 4

GYRO_X = 25830.00
GYRO_Y = -92820.00
GYRO_Z = 33460.00

ACCELBRO_X = 305
ACCELBRO_Y = 135
ACCELBRO_Z = 867
sent 206 bytes
Complete 5 sample message sent
Complete from sensor
Demo concluded successfully
```

2. on Python terminal

Socket server successfully receive the message sent by the client.

```
===== RESTART: D:\嵌入式\HW2.py =====
listen!
Connected by ('172.20.10.2', 28993)
Received data from socket client: {'Gyro': {'x': -70.0, 'y': -70.0, 'z': -70.0}, 'Acc
e': {'x': -11.0, 'y': -1.0, 'z': 97.0}, 's': 1}
Received data from socket client: {'Gyro': {'x': 70.0, 'y': -1330.0, 'z': 700.0}, 'Ac
ce': {'x': 29.0, 'y': -15.0, 'z': 999.0}, 's': 2}
Received data from socket client: {'Gyro': {'x': 193830.0, 'y': 177100.0, 'z': -32900
.0}, 'Acce': {'x': 587.0, 'y': -375.0, 'z': 785.0}, 's': 3}
Received data from socket client: {'Gyro': {'x': 8470.0, 'y': 4690.0, 'z': -4900.0},
'Acce': {'x': 121.0, 'y': -832.0, 'z': -544.0}, 's': 4}
Received data from socket client: {'Gyro': {'x': 25830.0, 'y': -92820.0, 'z': 33460.0
}, 'Acce': {'x': 305.0, 'y': 135.0, 'z': 867.0}, 's': 5}
```

IV. Visualization

We use *mplot3d* module in *matplotlib* to plot the sensor values, since it composed of data from x, y and z axis.

