Post-Lab 3: IEEE 802.15.4 & Thread

What to submit?

Please use this document as a template, add your responses directly, and export it as a PDF to Gradescope. Each group should submit one post-lab.

Group name:
Team member names: Rajan Verma (Group Activities with Connors and Anand)
Link to GitHub repository: (for M : Please refer to Connors Jacksons)

E. Setting up a Receiver and Sender

1. **What did you set the 1st byte of the payload to?**

2. **Prove that you can receive the packets with the updated payload (A screenshot of serial output works here),**

F. Filtering out packets not meant for your receiver

1. **How long does it take to send an 802.15.4 preamble?**

2. **What is the updated extended address of your receiver?**

3. **What nrf_802154_ function did you use to change the channel?**

4. **Which channel did you use?**

5. **Prove you can receive packets with the updated address, channel, and PAN id changes (A screenshot of serial output works here).**

G. Getting acknowledgements from the receiver

1. **Include a screenshot of the updated sender packet showing acknowledgements being requested.**

2. **Demonstrate that the sender transmissions are successful (A screenshot of serial output works here).**

H. Reducing the packet size

1. **By how many bytes did you reduce the packet size by switching to short addresses?**

2. **What did you change the sender's and receiver's addresses to?**

3. **Set short address for the receiver (Include your updated sender code and receiver code. A GitHub link to a specific commit or lines of code works here.)**

I. Low Power Listening (LPL)

1. **Code for the transmitter and receiver. A GitHub link to the two main.c files is fine.**

2. **The terminal outputs from both the transmitter and receiver for the 10 packets using both check periods. This should be four separate text files. A GitHub link to these text files is fine.**

3. **A short (paragraph or two) description of your results. How do you know your receiver was correctly duty cycling to save energy? Did this version of LPL work? Did you run into any challenges?**

4. **CHECKOFF: We will test your two boards with each other and against our reference implementation. Make sure to #define the check period and parameters of the network so you can easily change it.**
   - **a.** We will change your check window to a large value to verify the board does not receive packets while sleeping.
   - b. We will make your check window reasonable to make sure it works correctly on both the receiver and sender. The sender should print how long the packet train was.

Start of 3B Material

K. Join the class Thread network

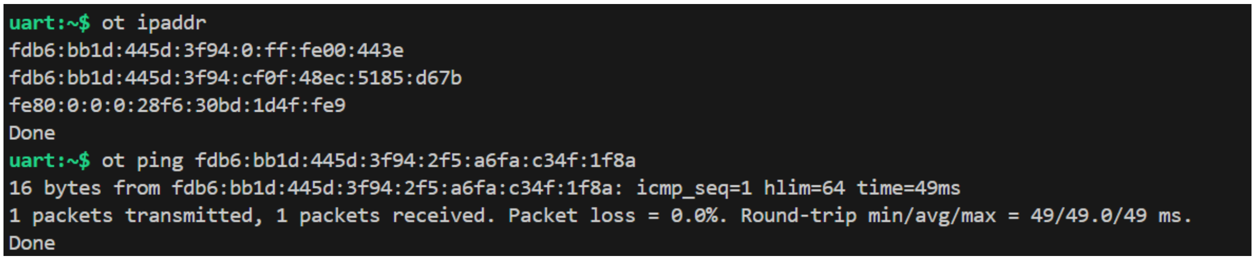
1. **IP addresses of both boards:**

2. **Router table:**

3. **Child table:**

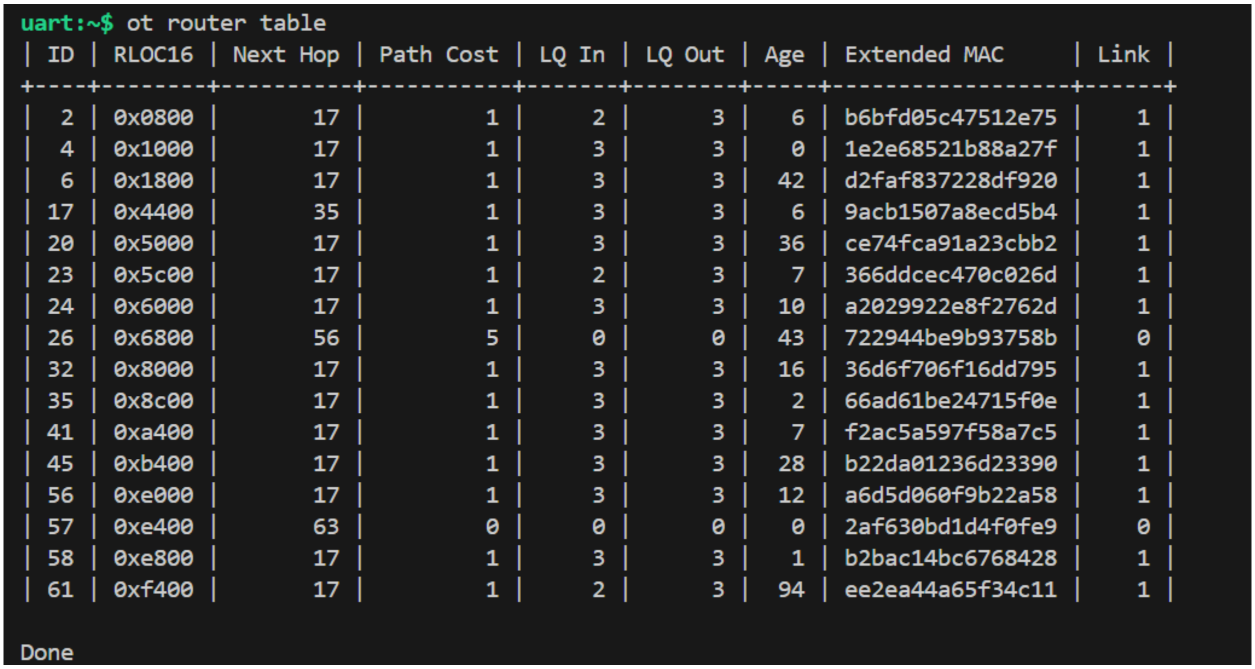
4. **Neighbor table:**

TASK 1: List the IP addresses of both connected boards and include them in your report.



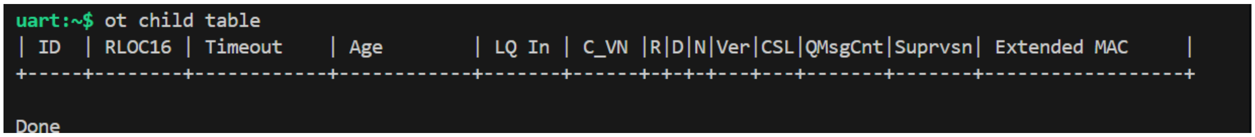
```
uart:~$ ot ipaddr
fdb6:bb1d:445d:3f94:0:ff:fe00:443e
fdb6:bb1d:445d:3f94:cf0f:48ec:5185:d67b
fe80:0:0:0:28f6:30bd:1d4f:fe9
Done
uart:~$ ot ping fdb6:bb1d:445d:3f94:2f5:a6fa:c34f:1f8a
16 bytes from fdb6:bb1d:445d:3f94:2f5:a6fa:c34f:1f8a: icmp_seq=1 hlim=64 time=49ms
1 packets transmitted, 1 packets received. Packet loss = 0.0%. Round-trip min/avg/max = 49/49.0/49 ms.
Done
```

TASK 2 : Run the ot router table command in the CLI app and copy or screenshot the output.



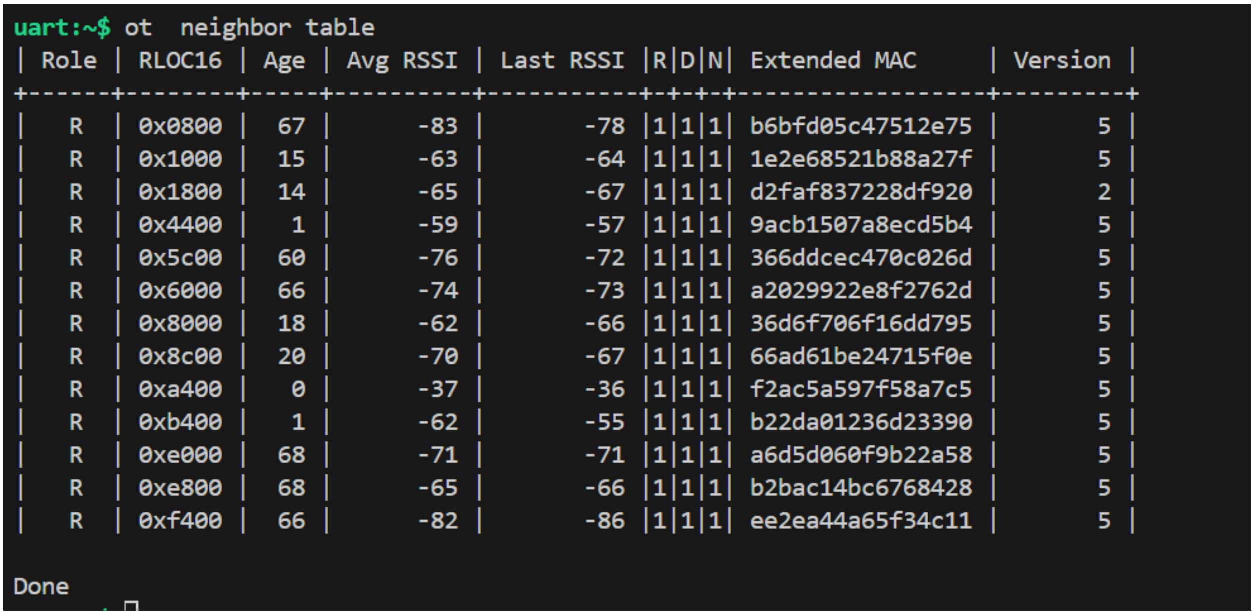
```
uart:~$ ot router table
| ID | RLOC16 | Next Hop | Path Cost | LQ In | LQ Out | Age | Extended MAC     | Link |
+----+--------+----------+-----------+-------+--------+-----+------------------+------+
|  2 | 0x0800 |       17 |         1 |     2 |      3 |   6 | b6bfd05c47512e75 |    1 |
|  4 | 0x1000 |       17 |         1 |     3 |      3 |   0 | 1e2e68521b88a27f |    1 |
|  6 | 0x1800 |       17 |         1 |     3 |      3 |  42 | d2faf837228df920 |    1 |
| 17 | 0x4400 |       35 |         1 |     3 |      3 |   6 | 9acb1507a8ecd5b4 |    1 |
| 20 | 0x5000 |       17 |         1 |     3 |      3 |  36 | ce74fca91a23cbb2 |    1 |
| 23 | 0x5c00 |       17 |         1 |     2 |      3 |   7 | 366ddcec470c026d |    1 |
| 24 | 0x6000 |       17 |         1 |     3 |      3 |  10 | a2029922e8f2762d |    1 |
| 26 | 0x6800 |       56 |         5 |     0 |      0 |  43 | 722944be9b93758b |    0 |
| 32 | 0x8000 |       17 |         1 |     3 |      3 |  16 | 36d6f706f16dd795 |    1 |
| 35 | 0x8c00 |       17 |         1 |     3 |      3 |   2 | 66ad61be24715f0e |    1 |
| 41 | 0xa400 |       17 |         1 |     3 |      3 |   7 | f2ac5a597f58a7c5 |    1 |
| 45 | 0xb400 |       17 |         1 |     3 |      3 |  28 | b22da01236d23390 |    1 |
| 56 | 0xe000 |       17 |         1 |     3 |      3 |  12 | a6d5d060f9b22a58 |    1 |
| 57 | 0xe400 |       63 |         0 |     0 |      0 |   0 | 2af630bd1d4f0fe9 |    0 |
| 58 | 0xe800 |       17 |         1 |     3 |      3 |   1 | b2bac14bc6768428 |    1 |
| 61 | 0xf400 |       17 |         1 |     2 |      3 |  94 | ee2ea44a65f34c11 |    1 |

Done
```

TASK 3: Run the ot child table command in the CLI app and copy or screenshot the output.



```
uart:~$ ot child table
| ID  | RLOC16 | Timeout    | Age        | LQ In | C_VN |R|D|N|Ver|CSL|QMsgCnt|Suprvsn| Extended MAC     |
+-----+--------+------------+------------+-------+------+-+-+-+---+---+-------+-------+------------------+

Done
```

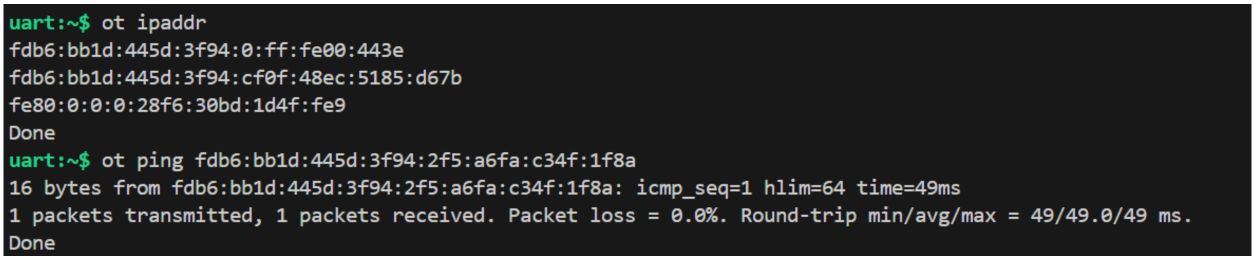
TASK 4: Run the ot neighbor table command in the CLI app and copy or screenshot the output.



```
uart:~$ ot  neighbor table
| Role | RLOC16 | Age | Avg RSSI | Last RSSI |R|D|N| Extended MAC     | Version |
+------+--------+-----+----------+-----------+-+-+-+------------------+---------+
|   R  | 0x0800 |  67 |      -83 |       -78 |1|1|1| b6bfd05c47512e75 |       5 |
|   R  | 0x1000 |  15 |      -63 |       -64 |1|1|1| 1e2e68521b88a27f |       5 |
|   R  | 0x1800 |  14 |      -65 |       -67 |1|1|1| d2faf837228df920 |       2 |
|   R  | 0x4400 |   1 |      -59 |       -57 |1|1|1| 9acb1507a8ecd5b4 |       5 |
|   R  | 0x5c00 |  60 |      -76 |       -72 |1|1|1| 366ddcec470c026d |       5 |
|   R  | 0x6000 |  66 |      -74 |       -73 |1|1|1| a2029922e8f2762d |       5 |
|   R  | 0x8000 |  18 |      -62 |       -66 |1|1|1| 36d6f706f16dd795 |       5 |
|   R  | 0x8c00 |  20 |      -70 |       -67 |1|1|1| 66ad61be24715f0e |       5 |
|   R  | 0xa400 |   0 |      -37 |       -36 |1|1|1| f2ac5a597f58a7c5 |       5 |
|   R  | 0xb400 |   1 |      -62 |       -55 |1|1|1| b22da01236d23390 |       5 |
|   R  | 0xe000 |  68 |      -71 |       -71 |1|1|1| a6d5d060f9b22a58 |       5 |
|   R  | 0xe800 |  68 |      -65 |       -66 |1|1|1| b2bac14bc6768428 |       5 |
|   R  | 0xf400 |  66 |      -82 |       -86 |1|1|1| ee2ea44a65f34c11 |       5 |

Done
```

Task 5 : TASK 5: Run a ping and show the output.

Ping output:

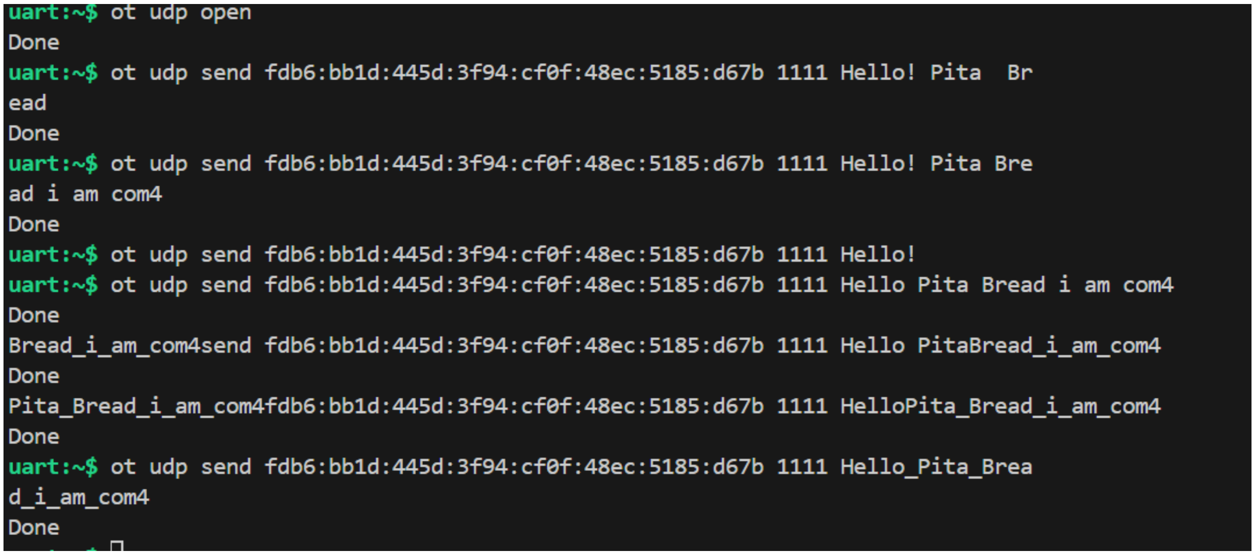


```
uart:~$ ot ipaddr
fdb6:bb1d:445d:3f94:0:ff:fe00:443e
fdb6:bb1d:445d:3f94:cf0f:48ec:5185:d67b
fe80:0:0:0:28f6:30bd:1d4f:fe9
Done
uart:~$ ot ping fdb6:bb1d:445d:3f94:2f5:a6fa:c34f:1f8a
16 bytes from fdb6:bb1d:445d:3f94:2f5:a6fa:c34f:1f8a: icmp_seq=1 hlim=64 time=49ms
1 packets transmitted, 1 packets received. Packet loss = 0.0%. Round-trip min/avg/max = 49/49.0/49 ms.
Done
```

L. Send UDP messages between your devices

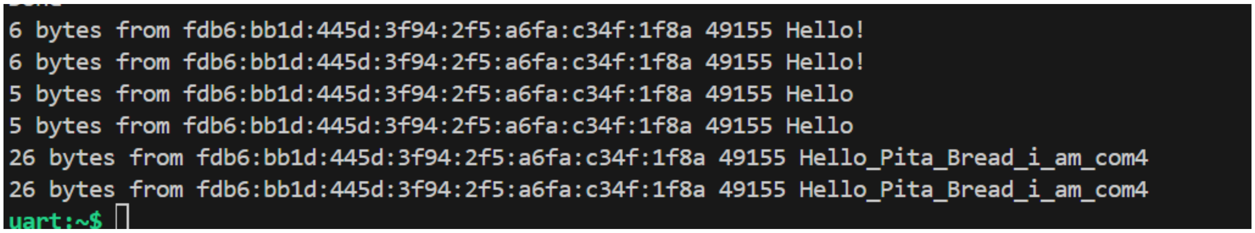
Show evidence (with sufficient explanation) to show your devices sending and receiving UDP packets.

Sender :Com4



```
uart:~$ ot udp open
Done
uart:~$ ot udp send fdb6:bb1d:445d:3f94:cf0f:48ec:5185:d67b 1111 Hello! Pita  Br
ead
Done
uart:~$ ot udp send fdb6:bb1d:445d:3f94:cf0f:48ec:5185:d67b 1111 Hello! Pita Bre
ad i am com4
Done
uart:~$ ot udp send fdb6:bb1d:445d:3f94:cf0f:48ec:5185:d67b 1111 Hello!
uart:~$ ot udp send fdb6:bb1d:445d:3f94:cf0f:48ec:5185:d67b 1111 Hello Pita Bread i am com4
Done
Bread_i_am_com4send fdb6:bb1d:445d:3f94:cf0f:48ec:5185:d67b 1111 Hello PitaBread_i_am_com4
Done
Pita_Bread_i_am_com4fdb6:bb1d:445d:3f94:cf0f:48ec:5185:d67b 1111 HelloPita_Bread_i_am_com4
Done
uart:~$ ot udp send fdb6:bb1d:445d:3f94:cf0f:48ec:5185:d67b 1111 Hello_Pita_Brea
d_i_am_com4
Done
```

Receiver : Pita Bread

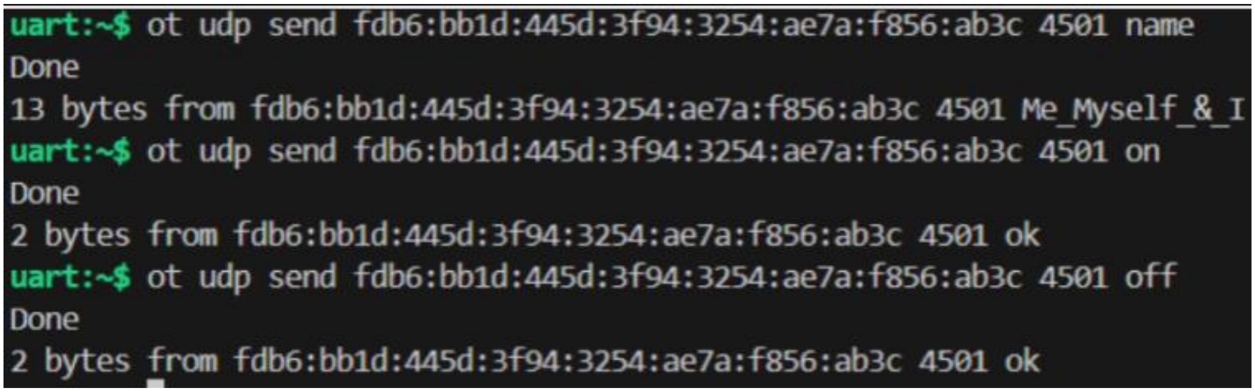


```
6 bytes from fdb6:bb1d:445d:3f94:2f5:a6fa:c34f:1f8a 49155 Hello!
6 bytes from fdb6:bb1d:445d:3f94:2f5:a6fa:c34f:1f8a 49155 Hello!
5 bytes from fdb6:bb1d:445d:3f94:2f5:a6fa:c34f:1f8a 49155 Hello
5 bytes from fdb6:bb1d:445d:3f94:2f5:a6fa:c34f:1f8a 49155 Hello
26 bytes from fdb6:bb1d:445d:3f94:2f5:a6fa:c34f:1f8a 49155 Hello_Pita_Bread_i_am_com4
26 bytes from fdb6:bb1d:445d:3f94:2f5:a6fa:c34f:1f8a 49155 Hello_Pita_Bread_i_am_com4
uart:~$ 
```

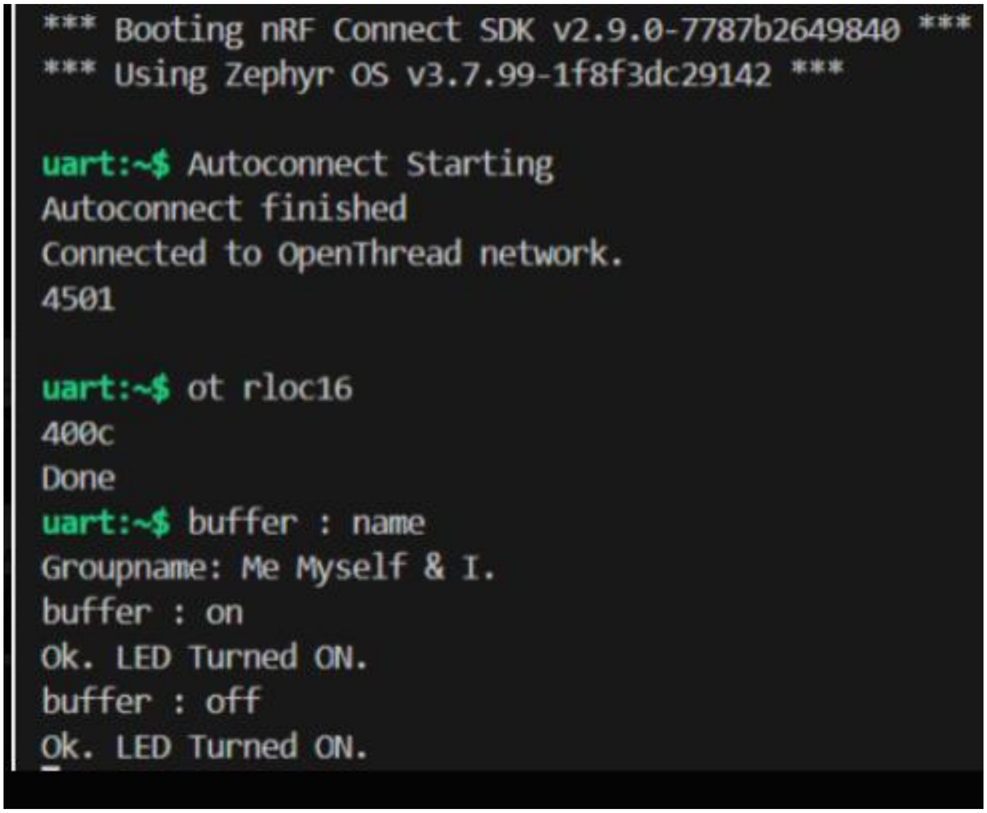
M. UDP Server

1. **Your code for your UDP device. A GitHub link is fine here.**

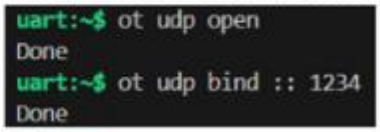
2. **Terminal output showing your UDP device working (with all three commands).**



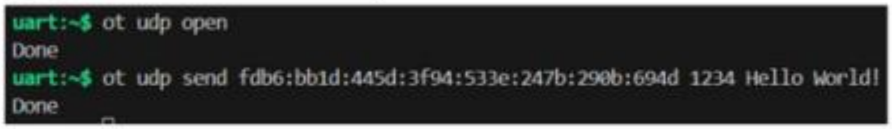
```
uart:~$ ot udp send fdb6:bb1d:445d:3f94:3254:ae7a:f856:ab3c 4501 name
Done
13 bytes from fdb6:bb1d:445d:3f94:3254:ae7a:f856:ab3c 4501 Me_Myself_&_I
uart:~$ ot udp send fdb6:bb1d:445d:3f94:3254:ae7a:f856:ab3c 4501 on
Done
2 bytes from fdb6:bb1d:445d:3f94:3254:ae7a:f856:ab3c 4501 ok
uart:~$ ot udp send fdb6:bb1d:445d:3f94:3254:ae7a:f856:ab3c 4501 off
Done
2 bytes from fdb6:bb1d:445d:3f94:3254:ae7a:f856:ab3c 4501 ok
```



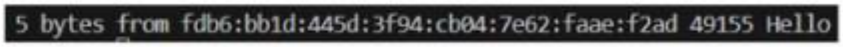
```
*** Booting nRF Connect SDK v2.9.0-7787b2649840 ***
*** Using Zephyr OS v3.7.99-1f8f3dc29142 ***

uart:~$ Autoconnect Starting
Autoconnect finished
Connected to OpenThread network.
4501

uart:~$ ot rloc16
400c
Done
uart:~$ buffer : name
Groupname: Me Myself & I.
buffer : on
Ok. LED Turned ON.
buffer : off
Ok. LED Turned ON.
```



```
uart:~$ ot udp open
Done
uart:~$ ot udp bind :: 1234
Done
```



```
uart:~$ ot udp open
Done
uart:~$ ot udp send fdb6:bb1d:445d:3f94:533e:247b:290b:694d 1234 Hello World!
Done
```



```
5 bytes from fdb6:bb1d:445d:3f94:cb04:7e62:faae:f2ad 49155 Hello
```

EC2. OPTIONAL: Visualize the network topology

This section is worth bonus points if you complete it.

1. **Show the network topology.**

2. **Show the network topology after changing your device's role (highlighting the change).**

EC3. OPTIONAL: LED Service

This section is worth bonus points if you complete it.

1. **Your code for your LED service. A GitHub link is fine here.**

2. **A video showing the button presses controlling the other board with the three commands.**