

ELEC5514 Networked Embedded Systems

Lab Report

Group 22	
Member Name	SID
Ke Zhang	450432501
Hao Qin	440413879
Xiangyu Yan	450069156
Yahong Liu	440358628

LAB 1

I. Introduction

In this lab, the general knowledgeable of ZigBee network and the hardware will be introduced, which is aim to familiar with ZigBee network and the software named Zena. And this lab divided into five parts: Introduce ZigBee, Introduce the hardware, Get familiar with the working environment, Learn how to use ZENA packet sniffer and Run the ZigBee demo.

II. Lab steps and results

1 Section 1. Introduction on ZigBee

In this section, we looked into “ZigBee 2006 Application note (AN1232). By reading this note, we have got general information about ZigBee, such as ZigBee is a wireless protocol implemented in the wireless network with low data rate sensors. And it uses the IEEE 802.15.4 and has network formation, device address assignments, routing, messaging and device discovery functions.

2 Section 2. Hardware

For this step, we have learnt about the hardware of ZigBee. In this lab we used demo board (PIC24fj256 MCU) with RF module (MRF24j40) as ZigBee node, Pickit 3 to connect the Demo board with the PC, RS232 cable and Zena network analyzer.

3 Section 3. Working environment

In this section, we built the demo applications by following the steps of lab notes and connected devices with PC.

4 Section 4. Zena network sniffer

We did every single steps of section 4 that is shown in QuickStartGuide and changed the channel to our group number 22. The result is shown below:

```
#define MAC_LONG_ADDR_BYTE7 0x00
#define MAC_LONG_ADDR_BYTE6 0x04
#define MAC_LONG_ADDR_BYTE5 0xA3
#define MAC_LONG_ADDR_BYTE4 0x00
#define MAC_LONG_ADDR_BYTE3 0x00
#define MAC_LONG_ADDR_BYTE2 0x00
#define MAC_LONG_ADDR_BYTE1 0x00
#define MAC_LONG_ADDR_BYTE0 0x00

#define PA_LEVEL 0x00 // -0.00 dBm
#define FREQUENCY_BAND FB_2400GHz
#define ALLOWED_CHANNELS 0x00400000
```

Figure 1.1

And the network topology is:

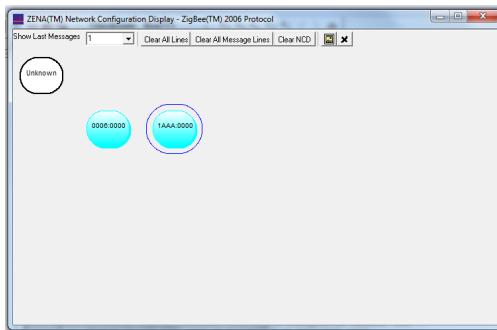


Figure 1.2

5 Section 5. Run the ZigBee demo

Firstly, we programed two devices, coordinator and one router.



Figure 1.3

Then switch the device power off. And connected a RS-232 cable from the coordinator device to the PC. Finally, opened up a Putty window on the PC to the COM port. Hit the 'Enter', the menu system showed as below:

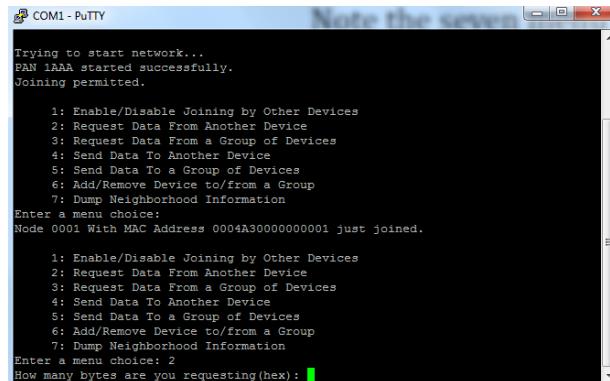
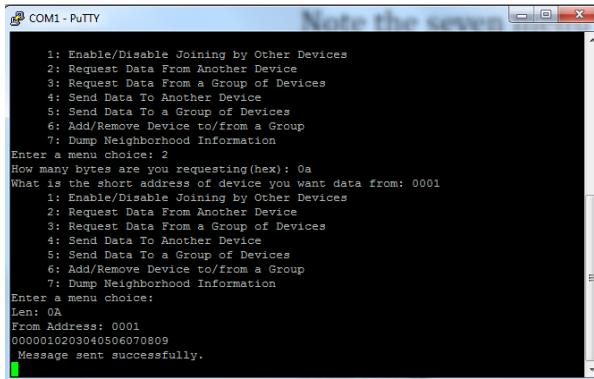


Figure 1.4

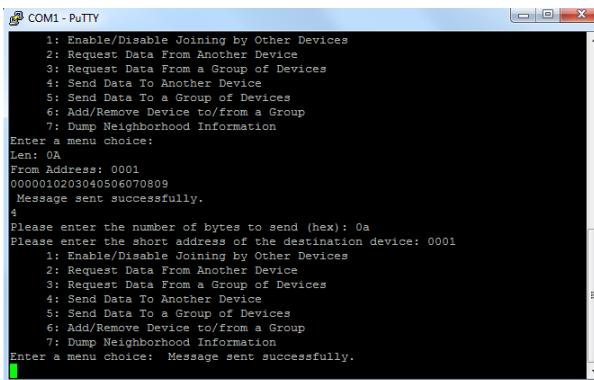
Then we typed 2 in the menu, the figure below shows the result:



```
COM1 - PuTTY
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
Enter a menu choice: 2
How many bytes are you requesting(hex): 0a
What is the short address of device you want data from: 0001
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
Enter a menu choice:
Len: 0A
From Address: 0001
0000010203040506070809
Message sent successfully.
```

Figure 1.5

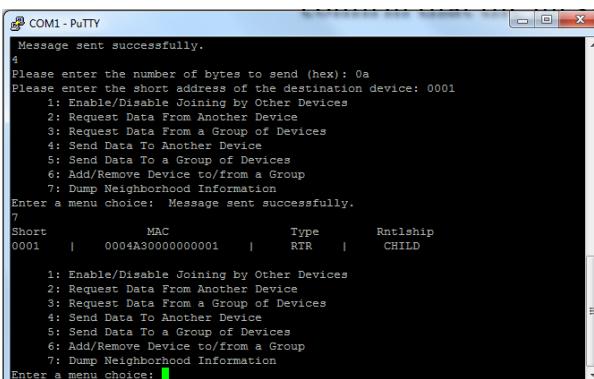
Then select option 4, the result is:



```
COM1 - PuTTY
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
Enter a menu choice:
Len: 0A
From Address: 0001
0000010203040506070809
Message sent successfully.
4
Please enter the number of bytes to send (hex): 0a
Please enter the short address of the destination device: 0001
1: Enable/Disable Joining by Other Devices
2: Request Data From a Group of Devices
3: Request Data From Another Device
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
Enter a menu choice: Message sent successfully.
```

Figure 1.6

The result of option 7 is shows as below:



```
COM1 - PuTTY
Message sent successfully.
4
Please enter the number of bytes to send (hex): 0a
Please enter the short address of the destination device: 0001
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
Enter a menu choice: Message sent successfully.
7
Short      MAC          Type      Rntship
0001 | 0004A30000000001 |   RTR | CHILD
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
Enter a menu choice:
```

Figure 1.7

III. Conclusion

In this lab, we have learnt the general knowledgeable of ZigBee network and the hardware of this system, and got familiar with the working environment, learnt how to use ZENA packet sniffer. Also, the demo code was understood.

LAB 2

I. Introduction

This report includes two sections:

1. ZigBee network layer basics.
2. Try ZigBee networking through PuTTy.

II. Lab steps and results

1 Section 1

The ZigBee coordinator use NWK (Network layer) to start new network, it can also provide new addressing to the device that joined the network.

There are three topologies for the ZigBee NWK: star, tree and mesh. For the star topology, the coordinator controls other end devices directly. In Figure2.1 the blue one represents the coordinator, in the PuTTy window we select option 1 and then choose enable join to let other devices join the coordinator, the pink one is an end device who join the network.

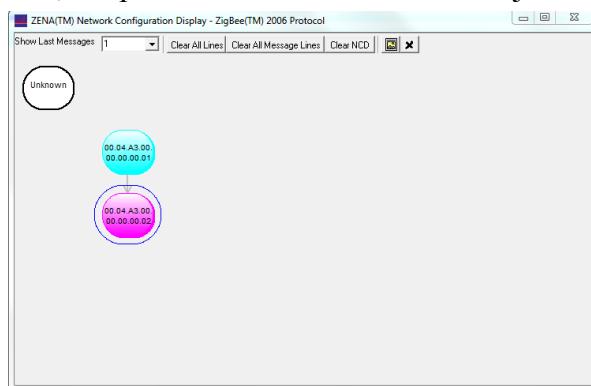


Figure 2.1

After that, we disable the join option of the coordinator, then setup a new device.

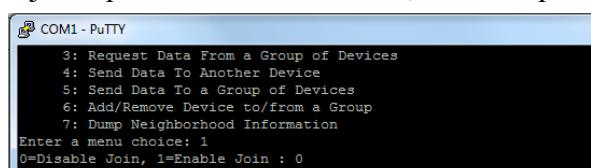


Figure 2.2

As shown in Figure2.3, the yellow one is the new device, it connects with the device we use in the last step. Otherwise, it will connect the coordinator if we do not disable the join for the coordinator, which shown in Figure2.4

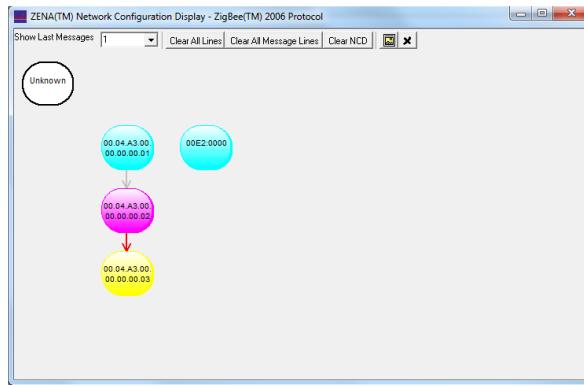


Figure 2.3

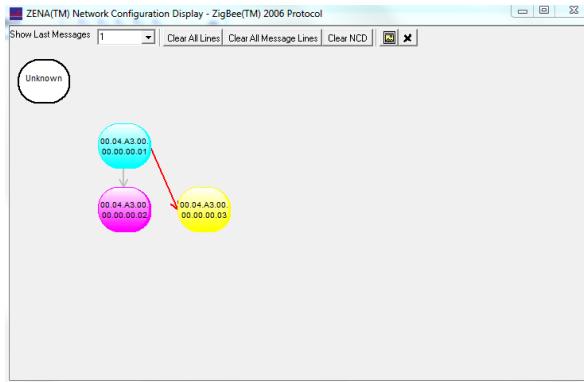


Figure 2.4

Here the figure2.3 refers to tree topology and the Figure2.4 refers to star topology.

2 Section 2

In this section, we establish a real ZigBee connection which include sending and receiving messages

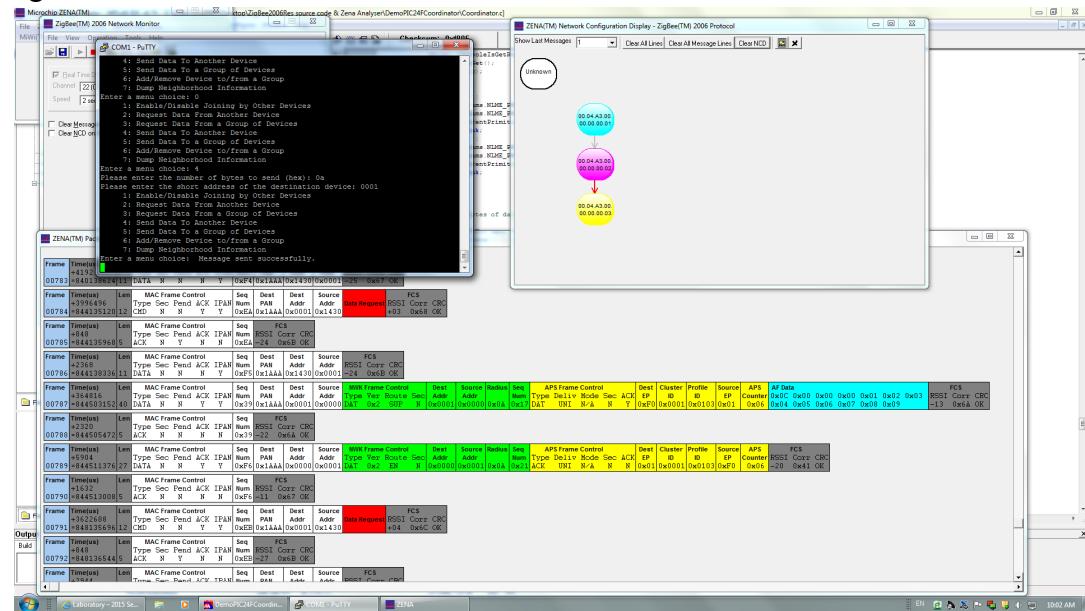


Figure 2.5

As shown in Figure2.5, we can see MAC header (white part) and Network header (green part), in these two parts, we can easily find the Dest address and Source address.

There are three routing modes: suppress, enable and force. In this section, we only focus on the enable and suppress modes. Firstly, we program the enable modes, as shown in figure2.6 we write the codes from lab guide to change it.

```

1602 ms APSDE_DATA_request.DstAddrMode = APS_ADDRESS_16_BIT;
1603 tf "\r\nPlease enter the short address of the destination device: ";
1605 ms APSDE_DATA_request.DstAddress.ShortAddr.v[1] = GetMACByte();
1606 ms APSDE_DATA_request.DstAddress.ShortAddr.v[0] = GetMACByte();
1607
1608 ms APSDE_DATA_request.SrcEndpoint = 1;
1609 ms APSDE_DATA_request.DstEndpoint = 240;
1610 ms APSDE_DATA_request.ProfilePath.Val = MY_PROFILE_ID;
1611
1612 rams APSDE_DATA_request.asduLength; TxData
1613 ms APSDE_DATA_request.RadiusCounter = DEFAULT_RADIUS;
1614 ms APSDE_DATA_request.DiscoverRoute = TRUE;
1615 ms APSDE_DATA_request.TxOptions.bits.acknowledged = 1;
1616
1617
1618 ms APSDE_DATA_request.DiscoverRoute = ROUTE_DISCOVERY_ENABLE;
1619

```

Figure 2.6

And now we can get the enable modes, we can check the result in Zena windows, in the Network header, we can see the EN which means the enable modes.

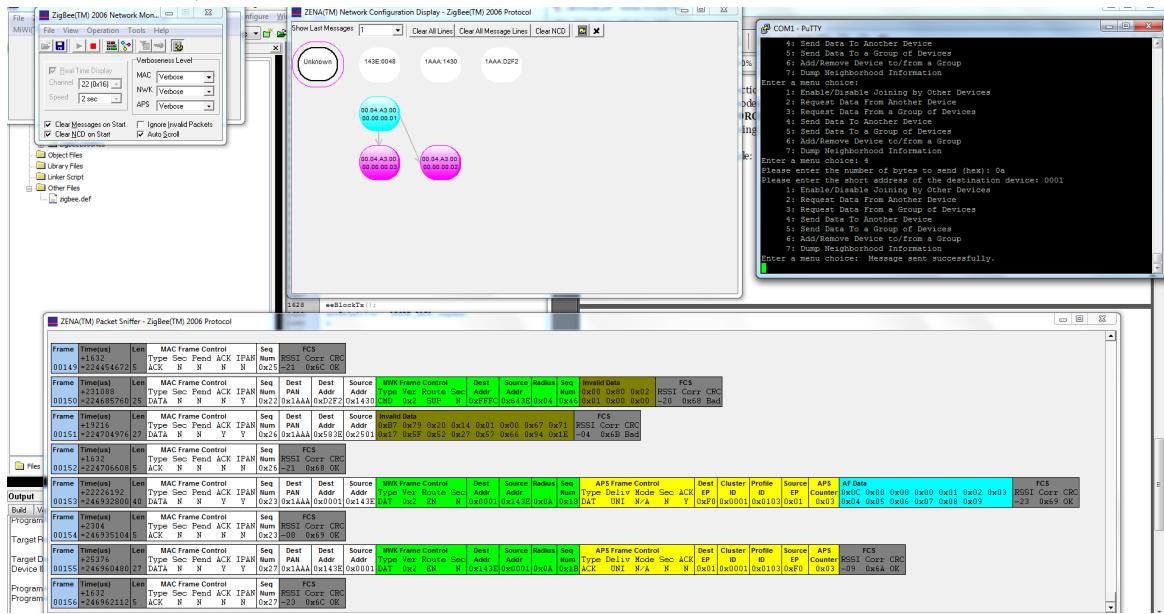


Figure 2.7

Next, we program the suppress modes by using the same way, as shown in figure2.8.

```

1609 params APSDE_DATA_request.SrcEndpoint = 1;
1610 params APSDE_DATA_request.DstEndpoint = 240;
1611 params APSDE_DATA_request.ProfilePath.Val = MY_PROFILE_ID;
1612
1613 //params APSDE_DATA_request.asduLength; TxData
1614 params APSDE_DATA_request.RadiusCounter = DEFAULT_RADIUS;
1615 params APSDE_DATA_request.DiscoverRoute = TRUE;
1616 params APSDE_DATA_request.TxOptions.bits.acknowledged = 1;
1617
1618 params APSDE_DATA_request.DiscoverRoute = ROUTE_DISCOVERY_SUPPRESS;
1619

```

Figure 2.8

And now we can get a suppress modes, just check the Zena result shown in figure2.9.

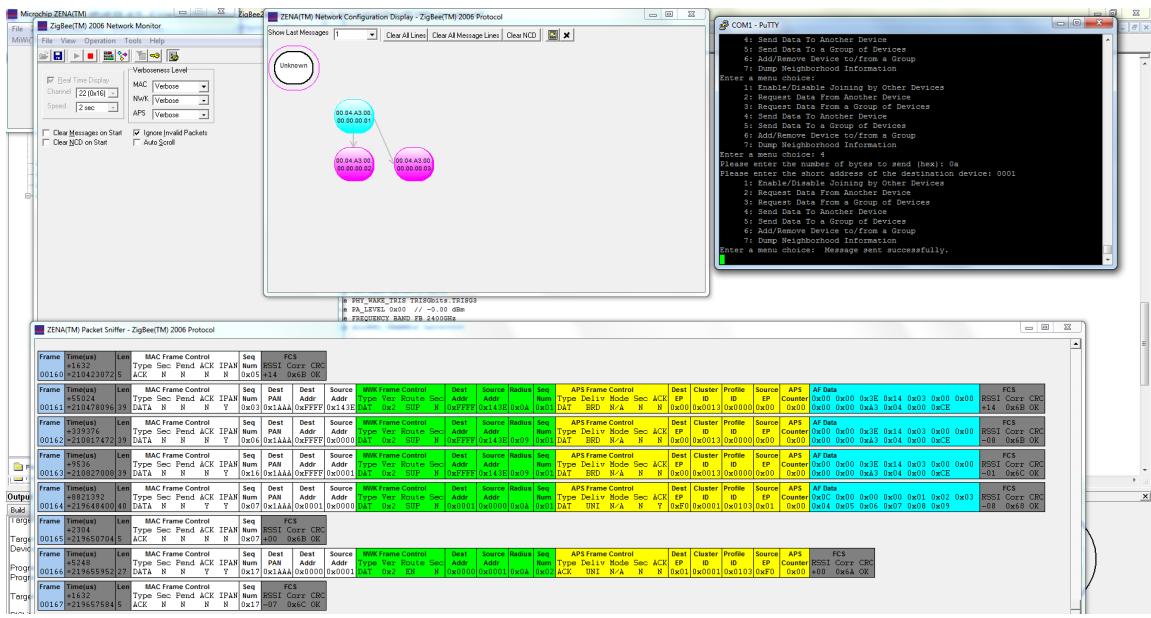


Figure 2.9

And now we get the suppress mode and we can see the packet capturing clearly.

III. Conclusion

In this lab, we learned how to construct a ZigBee network and the basic wireless sensor network constructing methods. In addition, we got familiar with the operation with the devices.

Lab 3

I. Introduction

In this lab, we will start to learn the ZigBee source code structure, and how to program the ZigBee node by C language.

II. Lab steps and results

1 Section 1

In this section, we investigate the source code of the ZigBee stack, we need to read the “Coordinator.c” file and try to understand the structure in the code.

2 Section 2

For this section, we first read the menu of the “Coordinator.c” file to understand the how its written and how the functions work.

Figure 3.1 is the picture of menu selections.

```

249  /* Menu System */
250  ROM char * const menu =
251      "\r\n    1: Enable/Disable Joining by Other Devices"
252      "\r\n    2: Request Data From Another Device"
253      "\r\n    3: Request Data From a Group of Devices"
254      "\r\n    4: Send Data To Another Device"
255      "\r\n    5: Send Data To a Group of Devices"
256      "\r\n    6: Add/Remove Device to/from a Group"
257      "\r\n    7: Dump Neighborhood Information"
258  ;

```

Figure 3.1

Figure 3.2 shows the cases for each menu selection.

```

1607
1608
1609 void ProcessMenu(void)
1610 {
1611     BYTE c;
1612
1613     DISABLE_NDI();
1614
1615     /* Get the user's input from the keyboard. */
1616     c = ConsoleGet();
1617     ConsolePut(c);
1618
1619     switch(c)
1620     {
1621         /* Enable or Disable Joining by other devices */
1622         case '1':
1623             ConsoleOutString(ROM char * const:"\r\n0=Disable Join, 1=Enable Join : ");
1624             while(ConsoleIsGetReady());
1625             c = ConsoleGet();
1626             ConsolePut(c);
1627             switch(c)
1628             {
1629                 case '0':
1630                     params.NLME_PERMIT_JOINING_request.PermitDuration = 0x00;
1631                     params.NLME_PERMIT_JOINING_request._updatePayload = FALSE;
1632                     currentPrimitive = NLME_PERMIT_JOINING_request;
1633                 case '1':
1634                     params.NLME_PERMIT_JOINING_request.PermitDuration = 0xFF;
1635                     params.NLME_PERMIT_JOINING_request._updatePayload = FALSE;
1636                     currentPrimitive = NLME_PERMIT_JOINING_request;
1637                 break;
1638             }
1639         break;
1640
1641         /* Request 16-bytes of data from Another Device */
1642         case '2':
1643             printf("\r\nHow many bytes are you requesting(hex):");
1644             Tdprintf("Tdata++"); Get16Byte();
1645             ZigbeeUnlockTx();
1646
1647             params.ARSDK_DATA_request.DstAddress = ARS_ADDRESS_16_BIT;
1648             params.ARSDK_DATA_request.ToTheShortAddress_of_Device_You_Want_Data_From = " ";
1649             params.ARSDK_DATA_request.DstAddress.ShortAddr.v[1] ~ Get16Byte();
1650             params.ARSDK_DATA_request.DstAddress.ShortAddr.v[0] ~ GetMACByte();
1651             params.ARSDK_DATA_request.RadiusCounter = DEFAULT_RADIUS;
1652
1653             params.ARSDK_DATA_request.DiscoveryRoute = ROUTE_DISCOVERY_SUPPRESS;
1654
1655
1656         #ifdef _SUPPORT_SECURITY
1657             params.ARSDK_DATA_request.TxOptions.Val = 1;
1658         #else
1659             params.ARSDK_DATA_request.TxOptions.Val = 0;
1660         #endif

```

Figure 3.2

Then we try to modify the menu, we add an extra choice on the menu to display a string of characters. As shown, we add case 8 option.

```

249  /* Menu System */
250  ROM char * const menu =
251      "\r\n    1: Enable/Disable Joining by Other Devices"
252      "\r\n    2: Request Data From Another Device"
253      "\r\n    3: Request Data From a Group of Devices"
254      "\r\n    4: Send Data To Another Device"
255      "\r\n    5: Send Data To a Group of Devices"
256      "\r\n    6: Add/Remove Device to/from a Group"
257      "\r\n    7: Dump Neighborhood Information"
258      "\r\n    8: Display HELLO WORLD on your screen"
259  ;

```

Figure 3.3

This is our detailed C code for case 8 in order to display a string of characters.

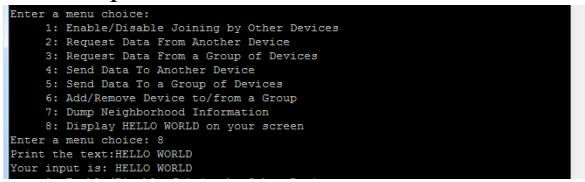
```

1691
1692
1693     case '8':
1694         printf("\r\nPrint the text:");
1695         char charTest[11];
1696         int intTest;
1697
1698         for(intTest=0; intTest<11; intTest++){
1699             charTest[intTest]=ConsoleGet();
1700             ConsolePut(charTest[intTest]);
1701
1702         printf("\r\nYour input is: ");
1703         for(intTest=0; intTest<11; intTest++){
1704             ConsolePut(charTest[intTest]);
1705         }
1706
1707         break;
1708

```

Figure 3.4

Figure3.5 shows the console output.



```
Enter a menu choice:
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Display HELLO WORLD on your screen
Enter a menu choice: 8
Print the text:HELLO WORLD
Your input is: HELLO WORLD
```

Figure 3.5

III. Conclusion

In this lab, we learned some information and knowledge about the structure of the source code and did some modification on the functions.

LAB 4

I. Introduction

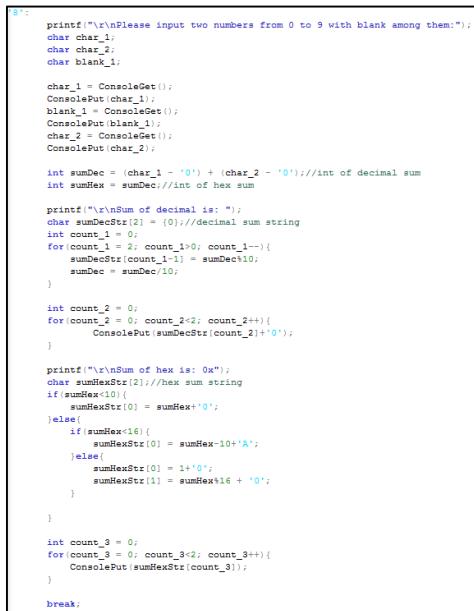
In this lab, we have two program tasks, first is to calculate and display the sum of 2 numbers and second is convert decimal number to a hex number.

II. Lab steps and results

1 Section 1:

In this part we need to calculate and display the sum of two decimal input numbers which from 0 to 9, and turn it into hex value.

The code is displayed as follow.



```
#include <stdio.h>
#include <conio.h>

int main()
{
    printf("\n\nPlease input two numbers from 0 to 9 with blank among them:");
    char char_1;
    char char_2;
    char blank_1;
    char blank_2;

    char_1 = ConsoleGet();
    ConsolePut(char_1);
    blank_1 = ConsoleGet();
    ConsolePut(blank_1);
    char_2 = ConsoleGet();
    ConsolePut(char_2);

    int sumDec = (char_1 - '0') + (char_2 - '0');//int of decimal sum
    int sumHex = sumDec//int of hex sum

    printf("\n\nSum of decimal is: ");
    char sumDecStr[2] = {0};//decimal sum string
    int count_1 = 0;
    for(count_1 = 2; count_1>0; count_1--){
        sumDecStr[count_1-1] = sumDec%10;
        sumDec = sumDec/10;
    }

    int count_2 = 0;
    for(count_2 = 0; count_2<2; count_2++){
        ConsolePut(sumDecStr[count_2]+'0');
    }

    printf("\n\nSum of hex is: 0x");
    char sumHexStr[2];//hex sum string
    if(sumHex<10){
        sumHexStr[0] = sumHex+'0';
    }else{
        if(sumHex<16){
            sumHexStr[0] = sumHex-10+'A';
        }else{
            sumHexStr[0] = 14+'0';
            sumHexStr[1] = sumHex%16 + '0';
        }
    }

    int count_3 = 0;
    for(count_3 = 0; count_3<2; count_3++){
        ConsolePut(sumHexStr[count_3]);
    }

    break;
}
```

Figure 4.1

And the outcome is shown in the putty interface, when we input 4 and 5 the sum is displayed as 9 along with its hex value 0x9.

```

1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Calculate and display the decimal and hex sum of two numbers from 0 - 9
0: Display the hex value of an decimal number from 256 - 999
Enter a menu choice: 8
Please input two numbers from 0 to 9 with blank:8 9
Sum of decimal is: 17
Sum of hex is: 0x11
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Calculate and display the decimal and hex sum of two numbers from 0 - 9
0: Display the hex value of an decimal number from 256 - 999
Enter a menu choice: 8
Please input two numbers from 0 to 9 with blank:4 5
Sum of decimal is: 09
Sum of hex is: 0x9

```

Figure 4.2

2 Section 2:

In this section we will try to input a decimal value with wider range (from 256 to 999) and then convert it into hex form and output it.

The code we programmed is shown as follow.

```

/*:
printf("\r\nPlease input a number from 256 to 999:");
char char_1_new;
char char_2_new;
char char_3_new;

char char_1_new = ConsoleGet();
ConsolePut(char_1_new);
char char_2_new = ConsoleGet();
ConsolePut(char_2_new);
char char_3_new = ConsoleGet();
ConsolePut(char_3_new);

int decimal = ((char_1_new-'0')*100 + (char_2_new-'0')*10 + (char_3_new-'0'))/1000; //input decimal number
char hexStr[8]; //hex number string

int count_1_new = 0;
for (count_1_new = 8; count_1_new>0; count_1_new--){
    hexStr[count_1_new-1] = decimal%16;
    if (decimal%16 > 9)
        hexStr[count_1_new-1] = decimal%16 + '0';
    else{
        hexStr[count_1_new-1] = decimal%16 - 10 + 'A';
    }
    decimal = decimal/16;
}

printf("\r\nHex value of input is: %s", hexStr);
int count_2_new = 0;
for count_2_new = 0; count_2_new < 8; count_2_new++){
    ConsolePut(hexStr[count_2_new]);
}
break;
*/

```

Figure 4.3

As shown in the putty interface, when we input decimal value 999, 345 and 500, we get 0x000003E7, 0x00000159 and 0x000001F4 respectively.

```

1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Calculate and display the decimal and hex sum of two numbers from 0 - 9
0: Display the hex value of an decimal number from 256 - 999
Enter a menu choice: 0
Please input a number from 256 to 999:999
Hex value of input is: 0x000003E7
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Calculate and display the decimal and hex sum of two numbers from 0 - 9
0: Display the hex value of an decimal number from 256 - 999
Enter a menu choice: 0
Please input a number from 256 to 999:345
Hex value of input is: 0x00000159
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Calculate and display the decimal and hex sum of two numbers from 0 - 9
0: Display the hex value of an decimal number from 256 - 999
Enter a menu choice: 0
Please input a number from 256 to 999:500
Hex value of input is: 0x000001F4

```

Figure 4.4

III. Conclusion

In this lab, we reviewed the C language and used that to add some new function for the

source code.

LAB 5

I. Introduction

We learned about the ZigBee endpoints and clusters structures in this lab. We learned about the actual condition when transmitting packets through ZigBee networks. This lab allows us to set up the endpoints and clusters in a ZigBee network.

II. Lab steps and results

1 Section 1:

We learned about the structure of the ZigBee network application layer. Generally, the ZigBee network consists of three-level addresses. The above is the network address and next level is the endpoint address, following by the cluster address.

We need to specify these two addresses before setting up the ZigBee networks.

2 Section 2:

In this part, we tried to realize the two tasks of lab4 based on the endpoints and clusters.

The endpoint and cluster are set as below:

Coordinator Endpoint ID: 171

Coordinator Cluster ID: 0x0002

Router Endpoint ID: 172

Router Cluster ID: 0x1111

2.1 Task 1:

The program is as below:

Coordinator transmits two numbers to router:

```

1718     case 'B':
1719         printf("\r\nPlease input two numbers with blanks:");
1720         /* Load the payload with the data to send */
1721         char sending_data = 0;
1722         for(i = 0; i < 3; i++)
1723         {
1724             sending_data = ConsoleGet();
1725             ConsolePut(sending_data);
1726             TxBuffer[TxData+i] = sending_data; // write the data into transmit buffer
1727         }
1728         params APSDE_DATA.request.DstAddrMode = APS_ADDRESS_16_BIT; // use network address
1729         params APSDE_DATA.request.DstAddress.ShortAddr.v[1] = 0x00; //MSB of Dest address
1730         params APSDE_DATA.request.DstAddress.ShortAddr.v[0] = 0x01; //LSB of Dest address
1731         params APSDE_DATA.request.SrcEndpoint = 171; // set the endpoints
1732         params APSDE_DATA.request.ProfileId.Val = 0x0000;
1733         params APSDE_DATA.request.ProfileId.Val |= MY_PROFILE_ID; // ignore this
1734         //params APSDE_DATA.request.radiusLength = TdData
1735         params APSDE_DATA.request.RadiusCounter = DEFAULT_RADIUS; // ignore this
1736         params APSDE_DATA.request.TxOptions.bits.acknowledged = 1; // use ACK or not
1737         params APSDE_DATA.request.DiscoverRoute = ROUTE_DISCOVERY_SUPPRESS; // select therouting mode
1738         // ignore the security
1739         #ifdef I_SUPPORT_SECURITY
1740         params APSDE_DATA.request.TxOptions.Val = 1;
1741         #else
1742         params APSDE_DATA.request.ClusterId.Val = 0x1111; //set the cluster ID
1743         ZigBeeBlockTx(); // block other transmission until finish
1744         currentPrimitive = APSDE_DATA_request; // give the command of transmission
1745         break;
1746     case 'a':
1747         break;

```

Figure 5.1

Router receives the packet and calculate the summary of two numbers, then transmit it back to coordinator:

```

1044     case 171:
1045         /*firstly you may want to read the payload first*/
1046         char received_data = -125; //APLGet() function will return the
1047         /* data received one by one.
1048         char received_data = -125; //if you want to read the buffer there is any
1049         char sending_data = 0;
1050         int num;
1051         int sum;
1052         // then you can do some processing or other work
1053         WORD_VAL clusterID = params.APSDE_DATA.indication.ClusterId;
1054         printf("ClusterID = %d\r\n", clusterID);
1055
1056         switch(clusterID)
1057         {
1058             int num = 0;
1059             int sum = 0;
1060             int i;
1061             case 0x1111:
1062                 num = received_data - '0';
1063                 num = received_data - '0';
1064                 sum = num + num;
1065             sending_data = received_data;
1066             sending_data = received_data;
1067             sending_data = sum + sending_data;
1068             sending_data = sum + sending_data;
1069             break;
1070         }
1071         /* you may also want to send some data back*/
1072         TxBuffer[TxData] = sending_data; //the data you want to send back into
1073         /* the buffer, if you want to read the buffer there is any
1074         TxBuffer[TxData] = sending_data;
1075         params APSDE_DATA.request.DstAddress.ShortAddr = APS_ADDRESS_16_BIT; // use network address
1076         params APSDE_DATA.request.DstAddress.ShortAddr.v[1] = 0x00; //MSB of Dest address
1077         params APSDE_DATA.request.DstAddress.ShortAddr.v[0] = 0x01; //LSB of Dest address
1078         params APSDE_DATA.request.SrcEndpoint = 172; // set the endpoints
1079         params APSDE_DATA.request.ProfileId.Val = 0x0000;
1080         #ifdef I_SUPPORT_SECURITY
1081         params APSDE_DATA.request.TxOptions.Val = 1;
1082         #else
1083         params APSDE_DATA.request.ClusterId.Val = 0x0000;
1084         ZigBeeBlockTx(); // block other transmission until finish
1085         currentPrimitive = APSDE_DATA_request; // give the command of transmission
1086         break;
1087     case 172:
1088         #ifdef I_SUPPORT_SECURITY
1089         params APSDE_DATA.request.TxOptions.Val = 1;
1090         #else
1091         params APSDE_DATA.request.ClusterId.Val = 0x0000;
1092         ZigBeeBlockTx();
1093         currentPrimitive = APSDE_DATA_request; // give the command of transmission
1094         break;
1095     case 173:
1096         break;
1097     case 174:
1098         break;
1099     case 175:
1100         break;
1101     case 176:
1102         break;
1103     case 177:
1104         break;
1105     case 178:
1106         break;
1107     case 179:
1108         break;
1109     case 17A:
1110         break;
1111     case 17B:
1112         break;
1113     case 17C:
1114         break;
1115     case 17D:
1116         break;
1117     case 17E:
1118         break;
1119     case 17F:
1120         break;
1121     case 17G:
1122         break;
1123     case 17H:
1124         break;
1125     case 17I:
1126         break;
1127     case 17J:
1128         break;
1129     case 17K:
1130         break;
1131     case 17L:
1132         break;
1133     case 17M:
1134         break;
1135     case 17N:
1136         break;
1137     case 17O:
1138         break;
1139     case 17P:
1140         break;
1141     case 17Q:
1142         break;
1143     case 17R:
1144         break;
1145     case 17S:
1146         break;
1147     case 17T:
1148         break;
1149     case 17U:
1150         break;
1151     case 17V:
1152         break;
1153     case 17W:
1154         break;
1155     case 17X:
1156         break;
1157     case 17Y:
1158         break;
1159     case 17Z:
1160         break;
1161     case 17A1:
1162         break;
1163     case 17A2:
1164         break;
1165     case 17A3:
1166         break;
1167     case 17A4:
1168         break;
1169     case 17A5:
1170         break;
1171     case 17A6:
1172         break;
1173     case 17A7:
1174         break;
1175     case 17A8:
1176         break;
1177     case 17A9:
1178         break;
1179     case 17AA:
1180         break;
1181     case 17AB:
1182         break;
1183     case 17AC:
1184         break;
1185     case 17AD:
1186         break;
1187     case 17AE:
1188         break;
1189     case 17AF:
1190         break;
1191     case 17B0:
1192         break;
1193     case 17B1:
1194         break;
1195     case 17B2:
1196         break;
1197     case 17B3:
1198         break;
1199     case 17B4:
1200         break;
1201     case 17B5:
1202         break;
1203     case 17B6:
1204         break;
1205     case 17B7:
1206         break;
1207     case 17B8:
1208         break;
1209     case 17B9:
1210         break;
1211     case 17BA:
1212         break;
1213     case 17BB:
1214         break;
1215     case 17BC:
1216         break;
1217     case 17BD:
1218         break;
1219     case 17BE:
1220         break;
1221     case 17BF:
1222         break;
1223     case 17C0:
1224         break;
1225     case 17C1:
1226         break;
1227     case 17C2:
1228         break;
1229     case 17C3:
1230         break;
1231     case 17C4:
1232         break;
1233     case 17C5:
1234         break;
1235     case 17C6:
1236         break;
1237     case 17C7:
1238         break;
1239     case 17C8:
1240         break;
1241     case 17C9:
1242         break;
1243     case 17CA:
1244         break;
1245     case 17CB:
1246         break;
1247     case 17CC:
1248         break;
1249     case 17CD:
1250         break;
1251     case 17CE:
1252         break;
1253     case 17CF:
1254         break;
1255     case 17D0:
1256         break;
1257     case 17D1:
1258         break;
1259     case 17D2:
1260         break;
1261     case 17D3:
1262         break;
1263     case 17D4:
1264         break;
1265     case 17D5:
1266         break;
1267     case 17D6:
1268         break;
1269     case 17D7:
1270         break;
1271     case 17D8:
1272         break;
1273     case 17D9:
1274         break;
1275     case 17DA:
1276         break;
1277     case 17DB:
1278         break;
1279     case 17DC:
1280         break;
1281     case 17DD:
1282         break;
1283     case 17DE:
1284         break;
1285     case 17DF:
1286         break;
1287     case 17E0:
1288         break;
1289     case 17E1:
1290         break;
1291     case 17E2:
1292         break;
1293     case 17E3:
1294         break;
1295     case 17E4:
1296         break;
1297     case 17E5:
1298         break;
1299     case 17E6:
1300         break;
1301     case 17E7:
1302         break;
1303     case 17E8:
1304         break;
1305     case 17E9:
1306         break;
1307     case 17EA:
1308         break;
1309     case 17EB:
1310         break;
1311     case 17EC:
1312         break;
1313     case 17ED:
1314         break;
1315     case 17EF:
1316         break;
1317     case 17F0:
1318         break;
1319     case 17F1:
1320         break;
1321     case 17F2:
1322         break;
1323     case 17F3:
1324         break;
1325     case 17F4:
1326         break;
1327     case 17F5:
1328         break;
1329     case 17F6:
1330         break;
1331     case 17F7:
1332         break;
1333     case 17F8:
1334         break;
1335     case 17F9:
1336         break;
1337     case 17FA:
1338         break;
1339     case 17FB:
1340         break;
1341     case 17FC:
1342         break;
1343     case 17FD:
1344         break;
1345     case 17FE:
1346         break;
1347     case 17FF:
1348         break;
1349     case 1700:
1350         break;
1351     case 1701:
1352         break;
1353     case 1702:
1354         break;
1355     case 1703:
1356         break;
1357     case 1704:
1358         break;
1359     case 1705:
1360         break;
1361     case 1706:
1362         break;
1363     case 1707:
1364         break;
1365     case 1708:
1366         break;
1367     case 1709:
1368         break;
1369     case 170A:
1370         break;
1371     case 170B:
1372         break;
1373     case 170C:
1374         break;
1375     case 170D:
1376         break;
1377     case 170E:
1378         break;
1379     case 170F:
1380         break;
1381     case 1710:
1382         break;
1383     case 1711:
1384         break;
1385     case 1712:
1386         break;
1387     case 1713:
1388         break;
1389     case 1714:
1390         break;
1391     case 1715:
1392         break;
1393     case 1716:
1394         break;
1395     case 1717:
1396         break;
1397     case 1718:
1398         break;
1399     case 1719:
1400         break;
1401     case 171A:
1402         break;
1403     case 171B:
1404         break;
1405     case 171C:
1406         break;
1407     case 171D:
1408         break;
1409     case 171E:
1410         break;
1411     case 171F:
1412         break;
1413     case 17100:
1414         break;
1415     case 17101:
1416         break;
1417     case 17102:
1418         break;
1419     case 17103:
1420         break;
1421     case 17104:
1422         break;
1423     case 17105:
1424         break;
1425     case 17106:
1426         break;
1427     case 17107:
1428         break;
1429     case 17108:
1430         break;
1431     case 17109:
1432         break;
1433     case 1710A:
1434         break;
1435     case 1710B:
1436         break;
1437     case 1710C:
1438         break;
1439     case 1710D:
1440         break;
1441     case 1710E:
1442         break;
1443     case 1710F:
1444         break;
1445     case 17110:
1446         break;
1447     case 17111:
1448         break;
1449     case 17112:
1450         break;
1451     case 17113:
1452         break;
1453     case 17114:
1454         break;
1455     case 17115:
1456         break;
1457     case 17116:
1458         break;
1459     case 17117:
1460         break;
1461     case 17118:
1462         break;
1463     case 17119:
1464         break;
1465     case 1711A:
1466         break;
1467     case 1711B:
1468         break;
1469     case 1711C:
1470         break;
1471     case 1711D:
1472         break;
1473     case 1711E:
1474         break;
1475     case 1711F:
1476         break;
1477     case 17120:
1478         break;
1479     case 17121:
1480         break;
1481     case 17122:
1482         break;
1483     case 17123:
1484         break;
1485     case 17124:
1486         break;
1487     case 17125:
1488         break;
1489     case 17126:
1490         break;
1491     case 17127:
1492         break;
1493     case 17128:
1494         break;
1495     case 17129:
1496         break;
1497     case 1712A:
1498         break;
1499     case 1712B:
1500         break;
1501     case 1712C:
1502         break;
1503     case 1712D:
1504         break;
1505     case 1712E:
1506         break;
1507     case 1712F:
1508         break;
1509     case 17130:
1510         break;
1511     case 17131:
1512         break;
1513     case 17132:
1514         break;
1515     case 17133:
1516         break;
1517     case 17134:
1518         break;
1519     case 17135:
1520         break;
1521     case 17136:
1522         break;
1523     case 17137:
1524         break;
1525     case 17138:
1526         break;
1527     case 17139:
1528         break;
1529     case 1713A:
1530         break;
1531     case 1713B:
1532         break;
1533     case 1713C:
1534         break;
1535     case 1713D:
1536         break;
1537     case 1713E:
1538         break;
1539     case 1713F:
1540         break;
1541     case 17140:
1542         break;
1543     case 17141:
1544         break;
1545     case 17142:
1546         break;
1547     case 17143:
1548         break;
1549     case 17144:
1550         break;
1551     case 17145:
1552         break;
1553     case 17146:
1554         break;
1555     case 17147:
1556         break;
1557     case 17148:
1558         break;
1559     case 17149:
1560         break;
1560     case 1714A:
1561         break;
1562     case 1714B:
1563         break;
1564     case 1714C:
1565         break;
1566     case 1714D:
1567         break;
1568     case 1714E:
1569         break;
1570     case 1714F:
1571         break;
1572     case 17150:
1573         break;
1574     case 17151:
1575         break;
1576     case 17152:
1577         break;
1578     case 17153:
1579         break;
1580     case 17154:
1581         break;
1581     case 17155:
1582         break;
1583     case 17156:
1584         break;
1585     case 17157:
1586         break;
1587     case 17158:
1588         break;
1589     case 17159:
1590         break;
1591     case 1715A:
1592         break;
1593     case 1715B:
1594         break;
1595     case 1715C:
1596         break;
1597     case 1715D:
1598         break;
1599     case 1715E:
1600         break;
1601     case 1715F:
1602         break;
1603     case 17160:
1604         break;
1605     case 17161:
1606         break;
1607     case 17162:
1608         break;
1609     case 17163:
1610         break;
1611     case 17164:
1612         break;
1613     case 17165:
1614         break;
1615     case 17166:
1616         break;
1617     case 17167:
1618         break;
1619     case 17168:
1620         break;
1621     case 17169:
1622         break;
1623     case 1716A:
1624         break;
1625     case 1716B:
1626         break;
1627     case 1716C:
1628         break;
1629     case 1716D:
1630         break;
1631     case 1716E:
1632         break;
1633     case 1716F:
1634         break;
1635     case 17170:
1636         break;
1637     case 17171:
1638         break;
1639     case 17172:
1640         break;
1641     case 17173:
1642         break;
1643     case 17174:
1644         break;
1645     case 17175:
1646         break;
1647     case 17176:
1648         break;
1649     case 17177:
1650         break;
1651     case 17178:
1652         break;
1653     case 17179:
1654         break;
1655     case 1717A:
1656         break;
1657     case 1717B:
1658         break;
1659     case 1717C:
1660         break;
1661     case 1717D:
1662         break;
1663     case 1717E:
1664         break;
1665     case 1717F:
1666         break;
1667     case 17180:
1668         break;
1669     case 17181:
1670         break;
1671     case 17182:
1672         break;
1673     case 17183:
1674         break;
1675     case 17184:
1676         break;
1677     case 17185:
1678         break;
1679     case 17186:
1680         break;
1681     case 17187:
1682         break;
1683     case 17188:
1684         break;
1685     case 17189:
1686         break;
1687     case 1718A:
1688         break;
1689     case 1718B:
1690         break;
1691     case 1718C:
1692         break;
1693     case 1718D:
1694         break;
1695     case 1718E:
1696         break;
1697     case 1718F:
1698         break;
1699     case 17190:
1700         break;
1701     case 17191:
1702         break;
1703     case 17192:
1704         break;
1705     case 17193:
1706         break;
1707     case 17194:
1708         break;
1709     case 17195:
1710         break;
1711     case 17196:
1712         break;
1713     case 17197:
1714         break;
1715     case 17198:
1716         break;
1717     case 17199:
1718         break;
1719     case 1719A:
1720         break;
1721     case 1719B:
1722         break;
1723     case 1719C:
1724         break;
1725     case 1719D:
1726         break;
1727     case 1719E:
1728         break;
1729     case 1719F:
1730         break;
1731     case 171900:
1732         break;
1733     case 171901:
1734         break;
1735     case 171902:
1736         break;
1737     case 171903:
1738         break;
1739     case 171904:
1740         break;
1741     case 171905:
1742         break;
1743     case 171906:
1744         break;
1745     case 171907:
1746         break;
1747     case 171908:
1748         break;
1749     case 171909:
1750         break;
1751     case 17190A:
1752         break;
1753     case 17190B:
1754         break;
1755     case 17190C:
1756         break;
1757     case 17190D:
1758         break;
1759     case 17190E:
1760         break;
1761     case 17190F:
1762         break;
1763     case 171910:
1764         break;
1765     case 171911:
1766         break;
1767     case 171912:
1768         break;
1769     case 171913:
1770         break;
1771     case 171914:
1772         break;
1773     case 171915:
1774         break;
1775     case 171916:
1776         break;
1777     case 171917:
1778         break;
1779     case 171918:
1780         break;
1781     case 171919:
1782         break;
1783     case 17191A:
1784         break;
1785     case 17191B:
1786         break;
1787     case 17191C:
1788         break;
1789     case 17191D:
1790         break;
1791     case 17191E:
1792         break;
1793     case 17191F:
1794         break;
1795     case 1719100:
1796         break;
1797     case 1719101:
1798         break;
1799     case 1719102:
1800         break;
1801     case 1719103:
1802         break;
1803     case 1719104:
1804         break;
1805     case 1719105:
1806         break;
1807     case 1719106:
1808         break;
1809     case 1719107:
1810         break;
1811     case 1719108:
1812         break;
1813     case 1719109:
1814         break;
1815     case 171910A:
1816         break;
1817     case 171910B:
1818         break;
1819     case 171910C:
1820         break;
1821     case 171910D:
1822         break;
1823     case 171910E:
1824         break;
1825     case 171910F:
1826         break;
1827     case 1719110:
1828         break;
1829     case 1719111:
1830         break;
1831     case 1719112:
1832         break;
1833     case 1719113:
1834         break;
1835     case 1719114:
1836         break;
1837     case 1719115:
1838         break;
1839     case 1719116:
1840         break;
1841     case 1719117:
1842         break;
1843     case 1719118:
1844         break;
1845     case 1719119:
1846         break;
1847     case 171911A:
1848         break;
1849     case 171911B:
1850         break;
1851     case 171911C:
1852         break;
1853     case 171911D:
1854         break;
1855     case 171911E:
1856         break;
1857     case 171911F:
1858         break;
1859     case 1719120:
1860         break;
1861     case 1719121:
1862         break;
1863     case 1719122:
1864         break;
1865     case 1719123:
1866         break;
1867     case 1719124:
1868         break;
1869     case 1719125:
1870         break;
1871     case 1719126:
1872         break;
1873     case 1719127:
1874         break;
1875     case 1719128:
1876         break;
1877     case 1719129:
1878         break;
1879     case 171912A:
1880         break;
1881     case 171912B:
1882         break;
1883     case 171912C:
1884         break;
1885     case 171912D:
1886         break;
1887     case 171912E:
1888         break;
1889     case 171912F:
1890         break;
1891     case 1719130:
1892         break;
1893     case 1719131:
1894         break;
1895     case 1719132:
1896         break;
1897     case 1719133:
1898         break;
1899     case 1719134:
1900         break;
1901     case 1719135:
1902         break;
1903     case 1719136:
1904         break;
1905     case 1719137:
1906         break;
1907     case 1719138:
1908         break;
1909     case 1719139:
1910         break;
1911     case 171913A:
1912         break;
1913     case 171913B:
1914         break;
1915     case 171913C:
1916         break;
1917     case 171913D:
1918         break;
1919     case 171913E:
1920         break;
1921     case 171913F:
1922         break;
1923     case 1719140:
1924         break;
1925     case 1719141:
1926         break;
1927     case 1719142:
1928         break;
1929     case 1719143:
1930         break;
1931     case 1719144:
1932         break;
1933     case 1719145:
1934         break;
1935     case 1719146:
1936         break;
1937     case 1719147:
1938         break;
1939     case 1719148:
1940         break;
1941     case 1719149:
1942         break;
1943     case 171914A:
1944         break;
1945     case 171914B:
1946         break;
1947     case 171914C:
1948         break;
1949     case 171914D:
1950         break;
1951     case 171914E:
1952         break;
1953     case 171914F:
1954         break;
1955     case 1719150:
1956         break;
1957     case 1719151:
1958         break;
1959     case 1719152:
1960         break;
1961     case 1719153:
1962         break;
1963     case 1719154:
1964         break;
1965     case 1719155:
1966         break;
1967     case 1719156:
1968         break;
1969     case 1719157:
1970         break;
1971     case 1719158:
1972         break;
1973     case 1719159:
1974         break;
1975     case 171915A:
1976         break;
1977     case 171915B:
1978         break;
1979     case 171915C:
1980         break;
1981     case 171915D:
1982         break;
1983     case 171915E:
1984         break;
1985     case 171915F:
1986         break;
1987     case 1719160:
1988         break;
1989     case 1719161:
1990         break;
1991     case 1719162:
1992         break;
1993     case 1719163:
1994         break;
1995     case 1719164:
1996         break;
1997     case 1719165:
1998         break;
1999     case 1719166:
2000         break;
2001     case 1719167:
2002         break;
2003     case 1719168:
2004         break;
2005     case 1719169:
2006         break;
2007     case 171916A:
2008         break;
2009     case 171916B:
2010         break;
2011     case 171916C:
2012         break;
2013     case 171916D:
2014         break;
2015     case 171916E:
2016         break;
2017     case 171916F:
2018         break;
2019     case 1719170:
2020         break;
2021     case 1719171:
2022         break;
2023     case 1719172:
2024         break;
2025     case 1719173:
2026         break;
2027     case 1719174:
2028         break;
2029     case 1719175:
2030         break;
2031     case 1719176:
2032         break;
2033     case 1719177:
2034         break;
2035     case 1719178:
2036         break;
2037     case 1719179:
2038         break;
2039     case 171917A:
2040         break;
2041     case 171917B:
2042         break;
2043     case 171917C:
2044         break;
2045     case 171917D:
2046         break;
2047     case 171917E:
2048         break;
2049     case 171917F:
2050         break;
2051     case 1719180:
2052         break;
2053     case 1719181:
2054         break;
2055     case 1719182:
2056         break;
2057     case 1719183:
2058         break;
2059     case 1719184:
206
```

Frame	Time(us)	Len	MAC Frame Control	Seq Num	Dest PAN	Dest Addr	Source Addr	HWK Frame Control	Dest Addr	Source Addr	Radius	Seq Num	APS Frame Control	Type	Dlev Mode	Sec ACK	EP ID	Cluster ID	Profile ID	Source EP	APS Counter	AF Data	FCS	
000013	+4607024	30	DATA N N Y	0x04	0x1AAA	0x0001	0x0000	DAT 0x2 SUP N	0x0000	0x0000	0x0A	0xD	DAT UNI N/A N H	EP 0x0AC	0x1111	0x0103	0x0AB	0x00	0x00	0x33	0x32 0x20	-14	0x6B OK	
Frame	+1792	Len	MAC Frame Control	Seq Num	FCS			HWK Frame Control					APS Frame Control									RSSI Corr CRC		
000014	-12656496	5	Type Sec Pend ACK IPAN	N	RSSI Corr CRC	ACK N N N	0x04	-17	0x69													-17	0x6A OK	
Frame	+7760	Len	MAC Frame Control	Seq Num	PAN	Dest Addr	Source Addr	HWK Frame Control	Dest Addr	Source Addr	Radius	Seq Num	APS Frame Control	Type	Dlev Mode	Sec ACK	EP ID	Cluster ID	Profile ID	Source EP	APS Counter	AF Data	FCS	
000015	-12654256	30	DATA N N Y	0x04	0x1AAA	0x0001	0x0000	DAT 0x2 SUP N	0x0000	0x0001	0x0A	0xD	DAT UNI N/A N H	EP 0x0AB	0x0002	0x0103	0x0AC	0x01	0x00	0x35	0x32 0x33	RSSI Corr CRC	-17	0x6A OK
Frame	+1776	Len	MAC Frame Control	Seq Num	FCS			HWK Frame Control					APS Frame Control									RSSI Corr CRC		
000016	-12666032	5	Type Sec Pend ACK IPAN	N	RSSI Corr CRC	ACK N N N	0x04	-17	0x6C													-17	0x6A OK	
Frame	+5942256	Len	MAC Frame Control	Seq Num	PAN	Dest Addr	Source Addr	HWK Frame Control	Dest Addr	Source Addr	Radius	Seq Num	APS Frame Control	Type	Dlev Mode	Sec ACK	EP ID	Cluster ID	Profile ID	Source EP	APS Counter	AF Data	FCS	
000017	-18608288	30	DATA N N Y	0x05	0x1AAA	0x0001	0x0000	DAT 0x2 SUP N	0x0000	0x0001	0x0A	0xD	DAT UNI N/A N H	EP 0x0AC	0x1111	0x0103	0x0AB	0x01	0x00	0x35	0x34 0x20	RSSI Corr CRC	-15	0x69 OK
Frame	+1776	Len	MAC Frame Control	Seq Num	FCS			HWK Frame Control					APS Frame Control									RSSI Corr CRC		
000018	-18610064	5	Type Sec Pend ACK IPAN	N	RSSI Corr CRC	ACK N N N	0x05	-17	0x6B													-17	0x6A OK	
Frame	+12760	Len	MAC Frame Control	Seq Num	PAN	Dest Addr	Source Addr	HWK Frame Control	Dest Addr	Source Addr	Radius	Seq Num	APS Frame Control	Type	Dlev Mode	Sec ACK	EP ID	Cluster ID	Profile ID	Source EP	APS Counter	AF Data	FCS	
000019	-18622883	30	DATA N N Y	0x05	0x1AAA	0x0001	0x0000	DAT 0x2 SUP N	0x0000	0x0001	0x0A	0xD	DAT UNI N/A N H	EP 0x0AB	0x0002	0x0103	0x0AC	0x02	0x00	0x39	0x34 0x35	RSSI Corr CRC	-17	0x6A OK
Frame	+1792	Len	MAC Frame Control	Seq Num	FCS			HWK Frame Control					APS Frame Control									RSSI Corr CRC		
000020	-18624624	5	Type Sec Pend ACK IPAN	N	RSSI Corr CRC	ACK N N N	0x05	-14	0x6B													-17	0x6A OK	

Figure 5.4

The Console output is:

```

1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Transfer a packet
Enter a menu choice: 8
Please input two numbers with blanks:2 3
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Transfer a packet
Enter a menu choice: Message sent successfully.

Your input is:2
and:3
Sum is:5
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Transfer a packet
Enter a menu choice: 8
Please input two numbers with blanks:4 5
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Transfer a packet
Enter a menu choice: Message sent successfully.

Your input is:4
and:5
Sum is:9

```

Figure 5.5

2.2 Task 2:

The program is as below:

Coordinator transmits the number from 256-999 to router:

```

1744
1745
    case 'B':
1746        printf("\r\nPlease input a number from 256 to 999:");
1747        /* Load the payload with the data to send */
1748        char sending_data = 0;
1749        for(i = 0; i < 8; i++)
1750        {
1751            sending_data = ConsoleGet();
1752            ConsolePut(sending_data);
1753            TxBuffer[TxBufIndex] = sending_data; // write the data into transmit buffer
1754        }
1755        params APSDE_DATA_request.DstAddrMode = APS_ADDRESS_16_BIT; // use network address
1756        params APSDE_DATA_request.DstAddress.ShortAddr.v[1] = 0x00; //MSB of Dest address
1757        params APSDE_DATA_request.DstAddress.ShortAddr.v[0] = 0x01; //LSB of Dest address
1758        params APSDE_DATA_request.TxOptions.bridged = 1; // set the endpoints
1759        params APSDE_DATA_request.ProfileId.Val = MY_PROFILE_ID; // ignore this
1760        //params APSDE_DATA_request.sduLength; TxData
1761        params APSDE_DATA_request.RadiusCounter = DEFAULT_RADIUS; // ignore this
1762        params APSDE_DATA_request.TxOptions.bits.acknowledged = 1; // use ACK or not
1763        params APSDE_DATA_request.DiscoverRoute = ROUTE_DISCOVERY_SUPPRESS; // select the routing mode
1764        // ignore the security
1765        #ifdef I_SUPPORT_SECURITY
1766            params APSDE_DATA_request.TxOptions.Val = 1;
1767        #else
1768            params APSDE_DATA_request.TxOptions.Val = 0;
1769        #endif
1770        params APSDE_DATA_request.ClusterId.Val = 0x1111; //set the cluster ID
1771        ZigBeeBlockTx(); // block other transmission until finish
1772        currentPrimitive = APSDE_DATA_request; // give the command of transmission
1773        break;
1774
1775

```

Figure 5.6

Router receives the packet and calculate the hex value of the number, then transmit it back to

coordinator:

```

1044
1045     case 172:
1046         /*firstly you want to read the payload first*/
1047         char received_data1 = -APLGet(); //APLGet() function will return the
1048         char received_data2 = -APLGet(); // data in payload one by one.
1049         char received_data3 = -APLGet();
1050         char sending_data1 = 0;
1051         char sending_data2 = 0;
1052         char sending_data3 = 0;
1053         //then you can do some processing or other work
1054         WORD_VAR clusterID = params APSDE_DATA_indication.ClusterId;
1055         switch(clusterID.Val)
1056     {
1057         int num1 = 0;
1058         int num2 = 0;
1059         int num3 = 0;
1060         int decimal = 0;
1061         char hexStr[3] = {0};
1062         int count_1_new = 0;
1063         case 0x111:
1064             num1 = received_data1 - '0';
1065             num2 = received_data2 - '0';
1066             num3 = received_data3 - '0';
1067             decimal = num1 * 100 + num2 * 10 + num3 * 1;
1068
1069             for (count_1_new = 3, count_1_new>0; count_1_new-- )
1070             {
1071                 hexStr[count_1_new-1] = decimal%16;
1072                 if(decimal%16<10){
1073                     hexStr[count_1_new-1] = decimal%16 + '0';
1074                 }
1075                 else{
1076                     hexStr[count_1_new-1] = decimal%16 - 10 + 'A';
1077                 }
1078                 decimal = decimal/16;
1079             }
1080             sending_data1 = hexStr[0];
1081             sending_data2 = hexStr[1];
1082             sending_data3 = hexStr[2];
1083
1084             break;
1085
1086     }

```

Figure 5.7

```

1087 /*/ we may also want to send some data back*/
1088 Tdbuffer(TxData+1) = sending_data; //writs the data you want to send back into
1089 TxData[0]; // the buffer, if there is any
1090 Tdbuffer(TxData+1) = receiving_data;
1091
1092 params APSD_DATA_request DestAddressMode = APS_ADDRESS_16_BIT; //use network address
1093 params APSD_DATA_request DestAddress ShortAddr=params APSD_DATA_indication SrcAddress ShortAddr; // use the source address of the received packet as Dest address
1094
1095 params APSD_DATA_request SrcEndpoint = 172; // set the endpoints
1096 params APSD_DATA_request EndpointId = 171;
1097 params APSD_DATA_request Profilid Val = MY_PROFILE_ID; // ignore this one
1098
1099 //params APSD_DATA_request asduLength TxData
1100 params APSD_DATA_request RadiusCounter = DEFAULT_RADIUS; // ignore it
1101 params APSD_DATA_request TxOptions bits acknowledged = 1; // use ACK or not
1102 params APSD_DATA_request DiscoverRoute = ROUTE_DISCOVERY_SUPPRESS; //set the routing mode
1103
1104 // ignore the security
1105 #ifndef I_SUPPORT_SECURITY
1106 params APSD_DATA_request TxOptions Val = 1;
1107 #else
1108 params APSD_DATA_request TxOptions Val = 0;
1109 #endif
1110
1111 params APSD_DATA_request ClusterId Val= 0x0002; //set the cluster ID
1112
1113 ZigBeeBlockTx(); // block other transmission until finish
1114 currentPrimitive = APSD_DATA_request; // give the command of transmission
1115 break;
1116

```

Figure 5.8

Coordinate receives the hex value and display it on the console:

```
776
777     case 171:
778     {
779         WORD_VAL clusterID = params APSDE_DATA_indication.ClusterId;
780         switch(clusterID.Val)
781         {
782             char Rx1 = 0;
783             char Rx2 = 0;
784             char Rx3 = 0;
785
786             case 0x0000:
787                 Rx1 = APISet();
788                 Rx2 = APISet();
789                 Rx3 = APISet();
790
791                 printf("The Max number of your input is:");
792                 ConsolePut(Rx1);
793                 ConsolePut(Rx2);
794                 ConsolePut(Rx3);
795                 break;
796         }
797     }
798 }
```

Figure 5.9

Here are the transmission details displayed on ZENA:

Figure 5.10

The Console output is:

```
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Transfer a packet
Enter a menu choice: 8
Please input a number from 256 to 999:567
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Transfer a packet
Enter a menu choice: Message sent successfully.

Hex number of your input is:237
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Transfer a packet
Enter a menu choice: 8
Please input a number from 256 to 999:999
1: Enable/Disable Joining by Other Devices
2: Request Data From Another Device
3: Request Data From a Group of Devices
4: Send Data To Another Device
5: Send Data To a Group of Devices
6: Add/Remove Device to/from a Group
7: Dump Neighborhood Information
8: Transfer a packet
Enter a menu choice: Message sent successfully.

Hex number of your input is:3E7
```

Figure 5.11

III. Conclusion

We combined the knowledge we learned in the previous labs. We got deep understand of the wireless sensor network and knew the basic working modes of the ZigBee networks.