

WSN 期中報告

組員

604410144 曾群丰

605410096 詹明翰

605410164 李俊億

題目

Experiments of Zigbee

Short Address Assignment

實驗環境

Windows 版本：Windows7 家用進階版

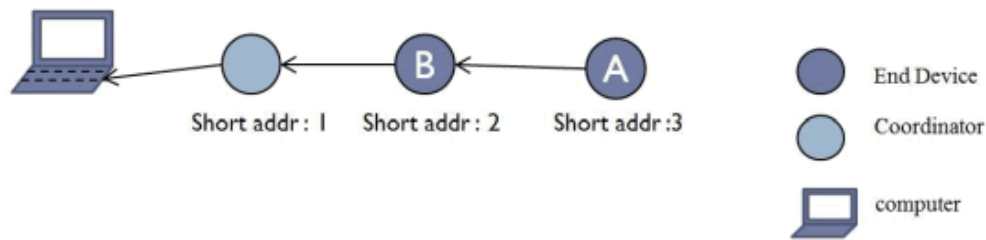
處理器：AMD Athlon(tm)IIX4 630 Processor 2.80GHz

記憶體：4.00GB

系統類型：64 位元作業系統

實驗流程

使用三台 Sensor 進行溫溼度的感測，分別為 coordinator、end device A、end device B。其中 coordinator 進行所有資料的收集，但本身不具有傳送資訊封包的功能；end device B 扮演中介者，將自身資訊以及 end device A 的感測資料一併傳給 coordinator；end device A 則是負責將感測資料傳給 end device B。



程式碼說明及修改

1. Final.h

```
/*  
*****  
***      Macro Definitions      ***  
*****  
/* PC UART parameters */  
#define PC_UARTPORT      E_AHI_UART_0  
#define PC_UARTEVENT     E_AHI_DEVICE_UART0  
#define PC_UARTBAUDRATE  E_AHI_UART_RATE_38400  
  
/* Network parameters */  
#define PAN_ID            0xABCD  
#define BROADCAST_ADR     0x0000  
#define COORDINATOR_ADR   0x0001  
#define END_DEVICE_ADR    0x0004  
#define MAX_NODE          20  
#define MAX_CHILD         1  
#define SCAN_CHANNELS     0x07FFF800UL  
#define CHANNEL_MIN       11  
#define ACTIVE_SCAN_DURATION 3  
#define ENERGY_SCAN_DURATION 3  
#define HELLO_INTERVAL    5  
#define DELAYED_INTERVAL  5  
#define DELAYED_TIMEOUT    2  
#define MAX_HOPCOUNT     4  
#define CSKIP_ADDR        0x0001
```

在 Final.h 中須將 MAX_CHILD 和 CSKIP_ADDR 設為 1，為了讓三台 Sensor 之

間點對點傳送。

2. Final.c

```
478 PRIVATE void vHandleMcpsDataInd(MAC_McpsDcfmInd_s *psMcpsInd)
479 {
480     MAC_RxFrameData_s *psFrame;
481     psFrame = &psMcpsInd->uParam.sIndData.sFrame;
482     /**change**/
483     u8LastLQI = psFrame->u8LinkQuality;
484     /**change**/
485     if (psFrame->sDstAddr.uAddr.u16Short == sNodeData.u16Address) {
486         /* Check application layer sequence number of frame and reject if it is the same as t1
487         if (psFrame->au8Sdu[0] >= sNodeData.u8RxPacketSeqNb)
488         {
489             sNodeData.u8RxPacketSeqNb++;
490             vProcessReceivedDataPacket(&psFrame->au8Sdu[1], (psFrame->u8SduLength) - 1);
491         }
492     } else if (psFrame->sDstAddr.uAddr.u16Short == BROADCAST_ADR) {
493         if (psFrame->au8Sdu[0] >= sNodeData.u8RxPacketSeqNb)
494         {
495             sNodeData.u8RxPacketSeqNb++;
496             /**change**/
497             if (psFrame->sSrcAddr.uAddr.u16Short > sNodeData.u16Address)
498             vProcessBroadcastDataPacket(&psFrame->au8Sdu[1], (psFrame->u8SduLength) - 1);
499             /**change**/
500         }
501     }
502 }
503
504
505
```

```
552 PRIVATE void vProcessBroadcastDataPacket(uint8 *pu8Data, uint8 u8Len)
553 {
554     static char command_b[100];
555     static int cur_cmb = 0;
556     /**change**/
557     char tmp_str[10];
558     v2ByteToDEC(u8LastLQI, tmp_str);
559     vUartPrint(PC_UARTPORT, tmp_str);
560     vUartPrint(PC_UARTPORT, " ");
561     /**change**/
562     int i;
563     for (i=0; i<u8Len; i++) {
564         if (pu8Data[i] == '\n' || pu8Data[i] == '\0') {
565             command_b[cur_cmb] = '\n';
566             command_b[cur_cmb+1] = '\0';
567             vUartPrint(PC_UARTPORT, "RX:");
568             vUartPrint(PC_UARTPORT, command_b);
569             vProcessor(command_b);
570             cur_cmb = 0;
571             command_b[cur_cmb] = '\0';
572         } else {
573             command_b[cur_cmb] = pu8Data[i];
574             cur_cmb++;
575         }
576     }
577     /**change**/
578     if (sNodeData.u16Address != 0x0001)
579     vTransmitDataPacket(pu8Data, u8Len, BROADCAST_ADR);
580     /**change**/
581 }
582
```

紅框是直接讀取 psFrame->u8LinkQuality 即可印出 LQI 值，而藍框內是判斷接收到的資料的地址大小，來決定要不要轉送給下一層。

```

810 PRIVATE void vTickTimerHandler(void) {
811     int i;
812     char tmp_str[100];
813     uint16 nexthop;
814
815     //led_toggle(LED0);
816
817     if (ul6HelloInt-- == 0) {
818         ul6HelloInt = HELLO_INTERVAL;
819         // Hello message
820         if (sNodeData.ul6Address > 1)
821         {
822             genHELLO(tmp_str, sNodeData.ul6Address);
823             vTransmitDataPacket((uint8*)tmp_str, strlen(tmp_str), BROADCAST_ADR);
824         }
825         //return;
826     }
827
828     vSerialRxString((uint8*)tmp_str);
829     if (strlen(tmp_str) != 0)
830     {
831         vProcessor(tmp_str);
832         //return;
833     }
834
835     if (ul6DelayedChk-- == 0) {
836         ul6DelayedChk = DELAYED_INTERVAL;
837         // Delayed check
838         for (i=0; i<100; i++)

```

而 Coordinator 本身只收資料並不發送任何資訊，故紅色框框內判斷如果本身地址為 1 則不發送任何封包。

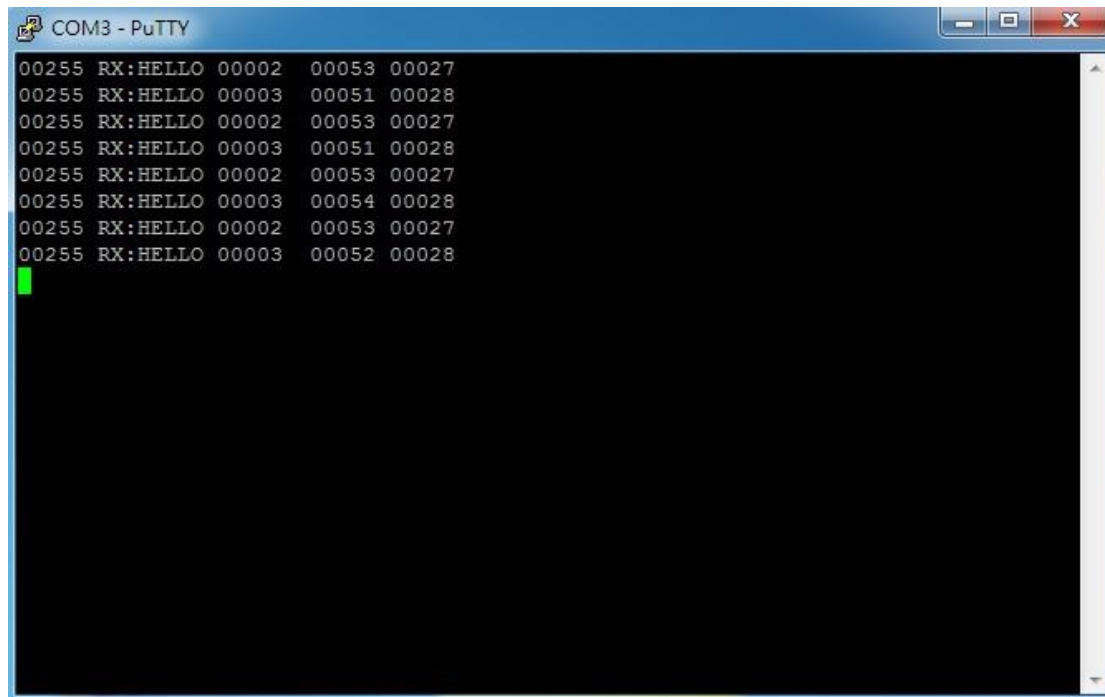
```

423 if (sNodeData.ul6NbrChild < MAX_CHILD)
424 {
425     /* Store end device address data */
426     ul6EndDeviceIndex = sNodeData.ul6NbrChild;
427     ul6ShortAddr = BROADCAST_ADR; //unused
428     sNodeData.sChildData[ul6EndDeviceIndex].ul6ShortAddr = sNodeData.ul6Address + ul6EndDeviceIndex*CSKIP_ADDR + 1;
429     sNodeData.sChildData[ul6EndDeviceIndex].u32ExtAddrL = psMlmeInd->uParam.sIndAssociate.sDeviceAddr.u32L;
430     sNodeData.sChildData[ul6EndDeviceIndex].u32ExtAddrH = psMlmeInd->uParam.sIndAssociate.sDeviceAddr.u32H;
431     sNodeData.ul6NbrChild++;
432
433     sMlmeReqResp.uParam.sRspAssociate.u8Status = 0; /* Access granted */
434 }
435 else
436     sMlmeReqResp.uParam.sRspAssociate.u8Status = 2; /* Denied */
437
438 /* Create association response */
439 sMlmeReqResp.u8Type = MAC_MLME_RSP_ASSOCIATE;
440 sMlmeReqResp.u8ParamLength = sizeof(MAC_MlmeRspAssociate_s);
441 sMlmeReqResp.uParam.sRspAssociate.sDeviceAddr.u32H = psMlmeInd->uParam.sIndAssociate.sDeviceAddr.u32H;
442 sMlmeReqResp.uParam.sRspAssociate.sDeviceAddr.u32L = psMlmeInd->uParam.sIndAssociate.sDeviceAddr.u32L;
443 sMlmeReqResp.uParam.sRspAssociate.ul6AssocShortAddr = sNodeData.ul6Address + ul6EndDeviceIndex*CSKIP_ADDR + 1;
444
445 sMlmeReqResp.uParam.sRspAssociate.u8SecurityEnable = FALSE;
446
447 /* Send association response. There is no confirmation for an association response, hence no need to check */
448 vAppApiMlmeRequest(&sMlmeReqResp, &sMlmeSyncCfm);

```

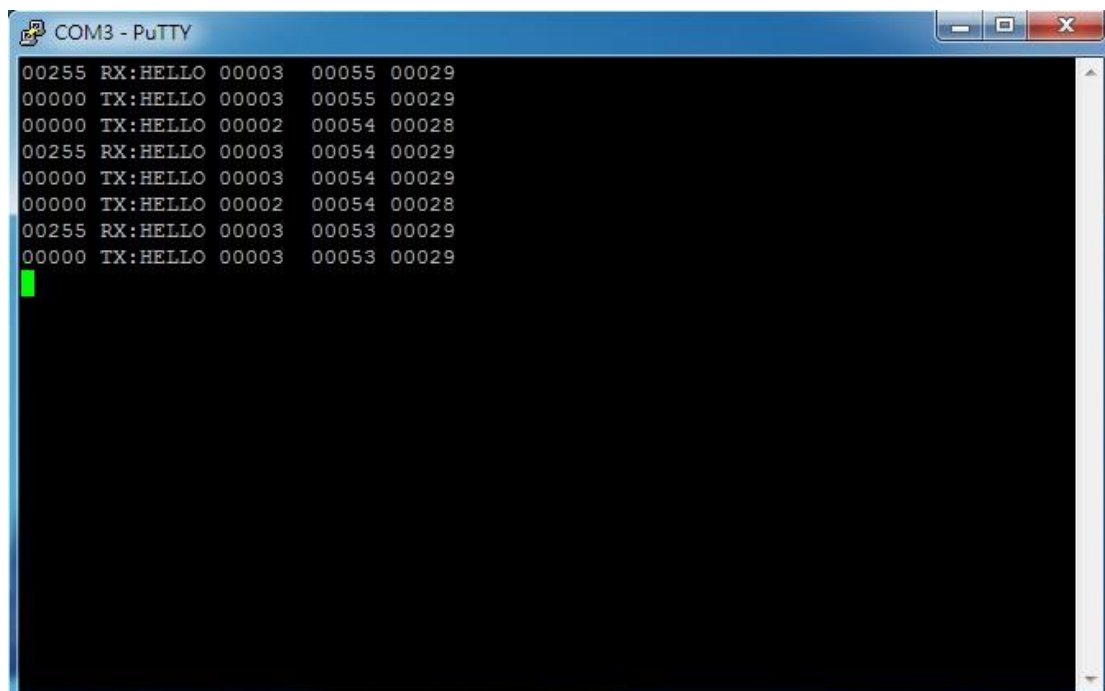
分配 short address。

實驗結果



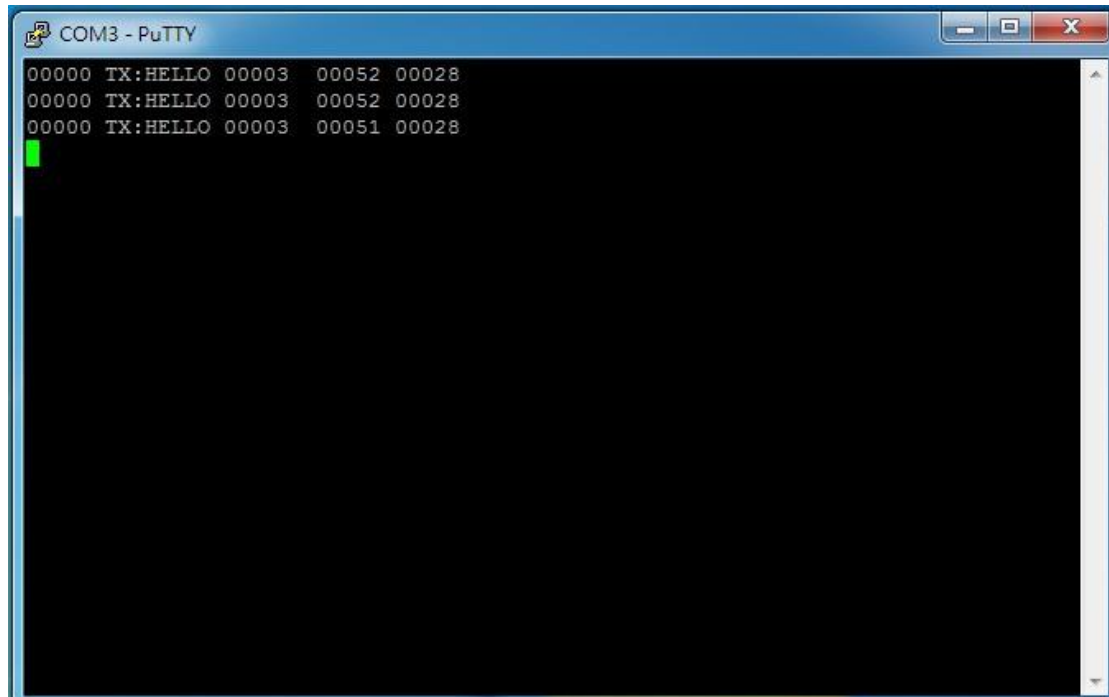
```
COM3 - PuTTY
00255 RX:HELLO 00002 00053 00027
00255 RX:HELLO 00003 00051 00028
00255 RX:HELLO 00002 00053 00027
00255 RX:HELLO 00003 00051 00028
00255 RX:HELLO 00002 00053 00027
00255 RX:HELLO 00003 00054 00028
00255 RX:HELLO 00002 00053 00027
00255 RX:HELLO 00003 00052 00028
█
```

Coordinator



```
COM3 - PuTTY
00255 RX:HELLO 00003 00055 00029
00000 TX:HELLO 00003 00055 00029
00000 TX:HELLO 00002 00054 00028
00255 RX:HELLO 00003 00054 00029
00000 TX:HELLO 00003 00054 00029
00000 TX:HELLO 00002 00054 00028
00255 RX:HELLO 00003 00053 00029
00000 TX:HELLO 00003 00053 00029
█
```

Parent



```
COM3 - PuTTY
00000 TX:HELLO 00003 00052 00028
00000 TX:HELLO 00003 00052 00028
00000 TX:HELLO 00003 00051 00028
█
```

End device