text.Replace(" ", string.Empty); 字串取代

int pos\_account = text.IndexOf("Account:"); 回傳 A的位置

my\_account = text.Substring(pos\_account + 8, temp - (pos\_account + 8));

擷取子集合 (開始位置 長度)

my\_account = my\_account.Trim(); (移除空格)

宣告字串 array

string[] array = new string[6];

六筆資料

rtb\_msg.AppendText("retrun find \"test\" fail\n");

讀取外部檔案

string[] lines = System.IO.File.ReadAllLines(@"d:\8195AM\_WlanLab\_vk008.map");

string data = "";

int[,] final\_data = new int[32,16] ; // 二為矩陣宣告

//0x95 0x81 0xc2 0x16

for(int i = 0; i<lines.Length; i++)

{

data = lines[i];

data = data.Replace(" ", "");

for(int j=0; j <16; j = j + 2)

{

final\_data[i,j+1] = Convert.ToInt32( data.Substring(j\*2, 2),16);讀入部分字串 並轉16進位

final\_data[i,j] = Convert.ToInt32(data.Substring(j\*2+2, 2), 16);

}

}

RX 擷取字串

if (rx\_data.Contains("packet OK"))

good\_packet = rx\_data.Substring(rx\_data.IndexOf("OK:") + 3, rx\_data.IndexOf("CRC") - rx\_data.IndexOf("OK:") - 3).Trim();

error\_pcacket = rx\_data.Substring(rx\_data.IndexOf("error:") + 6, rx\_data.IndexOf("FA") - rx\_data.IndexOf("error:") - 6).Trim();}

傳遞多個變數 切割

在 **getPara** 中抓取UI的數值 將其連接成一字串

在 另一個Class裡 parameter\_pase中分析切割字串 存入 array 再取值

string **tx\_para** = **getPara**();

tx.tx\_prepare(srp, tx\_para);

private string **getPara**()

{

string **txPara** = "ch = " + dataMap[cbb\_Freq.Text] + " ; bw = " + dataMap[cbb\_BandWidth.Text] + " ; data\_rate = " + dataMap[cbb\_DataRate.Text] + " ; preamble = " + dataMap[cbb\_Preamble.Text] +

" ; ant = " + dataMap[cbb\_Antenna.Text] + " ; crystal = " + cbb\_XtalCap.Text + " ; powerIndex = " + cbb\_PowerIndex.Text + " ; count = " + txt\_TxPacketCount.Text + " ; txItem = " + dataMap[cbb\_TxItem.Text];

return **txPara**;

}

parameter\_parse(tx\_parameter, out channel, out bandwidth, out datarate, out preamble, out crystal, out powerIndex, out \_count, out \_txItem, out ant);

void parameter\_parse(string **input\_para**, out int channel, out int bandwidth, out int datarate, out int preamble, out int crystal, out int powerIndex, out int count, out int txItem, out string ant)

{

**input\_para** = input\_para.Replace(" ", "");

char[] delimiterChars = { ' ', '=', ';' };

string[] para\_Array = **input\_para**.Split(delimiterChars);

int index = Array.FindIndex(para\_Array, element => element.Contains("ch"));

channel = Convert.ToInt32(para\_Array[index + 1]);

index = Array.FindIndex(para\_Array, element => element.Contains("bw"));

bandwidth = Convert.ToInt32(para\_Array[index + 1]);

index = Array.FindIndex(para\_Array, element => element.Contains("data\_rate"));

datarate = Convert.ToInt32(para\_Array[index + 1]);

index = Array.FindIndex(para\_Array, element => element.Contains("preamble"));

preamble = Convert.ToInt32(para\_Array[index + 1]);

index = Array.FindIndex(para\_Array, element => element.Contains("ant"));

ant = para\_Array[index + 1];

index = Array.FindIndex(para\_Array, element => element.Contains("crystal"));

crystal = Convert.ToInt32(para\_Array[index + 1]);

index = Array.FindIndex(para\_Array, element => element.Contains("powerIndex"));

powerIndex = Convert.ToInt32(para\_Array[index + 1]);

index = Array.FindIndex(para\_Array, element => element.Contains("count"));

count = Convert.ToInt32(para\_Array[index + 1]);

index = Array.FindIndex(para\_Array, element => element.Contains("txItem"));

txItem = Convert.ToInt32(para\_Array[index + 1]);

}