國立宜蘭大學電子工程學系

專題研究報告

|  |  |
| --- | --- |
| 題目： | 以SDR實現OTSM通訊技術 |

專題組員：B1042021陳昱升B1042029萬子謙

B1042051陳軍銓B1042039姚辰彥

B1042042卓家葳B1042027陳建霖

指導教授:梁耀仁博士

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 中華民國 |  | 年 |  | 月 |  |  |

**摘要**

目錄

[1 簡介 1](#_Toc164089861)

[1.1 計畫緣由 1](#_Toc164089862)

[1.2 目的 1](#_Toc164089863)

[1.3 成果介紹 2](#_Toc164089864)

[2 相關內容及原理 3](#_Toc164089865)

[2.1 SDR簡介 3](#_Toc164089866)

[2.2 Matlab簡介 6](#_Toc164089867)

[2.3 QAM原理 7](#_Toc164089868)

[2.4 OTSM原理 8](#_Toc164089869)

[3 流程 11](#_Toc164089870)

[3.1 ADALM Pluto環境搭建 11](#_Toc164089871)

[4 進度控制與分配 20](#_Toc164089874)

[4.1 甘特圖 20](#_Toc164089875)

[4.2 工作分配 25](#_Toc164089876)

[5 成果 27](#_Toc164089877)

[5.1 Tx傳送端和Rx接收27](#_Toc164089878)

[5.2 將Loop.m分成Tx與Rx兩部分 28](#_Toc164089879)

[6 檢討與討論 30](#_Toc164089883)

[6.1 檢討 30](#_Toc164089884)

[6.2 結論 30](#_Toc164089885)

[參考資料 32](#_Toc164089886)

[附檔 33](#_Toc164089887)

[附檔1 BER.m程式碼 33](#_Toc164089888)

[附檔2 Loop.m程式碼 34](#_Toc164089889)

[附檔3 Tx.m程式碼 43](#_Toc164089890)

[附檔4 Rx.m程式碼 48](#_Toc164089891)

[附檔5 Matlab\_iio程式修改 55](#_Toc164089892)

[附檔6 DataAnalysis.m程式碼 58](#_Toc164089893)

[附檔7 figplusfig.m程式碼 60](#_Toc164089894)

1. 簡介

本專題名稱為以SDR實現OTSM無線通訊技術。在計畫摘要中，我們將介紹壹計畫緣由、貳相關內容及原理、參成果介紹。

1. 計畫緣由

隨著通信技術不斷的進步，從有線到無線通信，為了能夠節省有限的頻寬資源、提供高質量通信服務並能抵抗頻率選擇性衰減通道，運用軟體定義無線電（Software Defined Radio）實現正交分時多工 (Orthogonal Time-Sequency Multiplexing)技術。

1. 目的

傳送資料透過Pluto送出，然後由接收端接收到訊息判斷接收端的品質並把錯誤率呈現出來。透過專題製作我們可以瞭解在通訊系統中的理論部分怎麼樣利用Matlab code一步一步實現OTSM通訊技術的傳送端和接收端，最後可以了解整個通訊系統並實作、驗證。

1. 成果介紹

我們利用Matlab以及傅立葉轉換的技術對接收到的訊號作後處理，再利用通道模型的公式作通道估測來推算通道響應值，來觀察通道。在使用SDR來取代向量網路分析儀來接收、分析訊號和訊號產生器發送訊號，以降低實作之成本，讓我們自己動手實踐出無線傳輸的過程。

下面將簡單介紹內容：所使用的工具:Pluto SDR\*2、Matlab、PC\*2。

1. 改寫Matlab code使原本只能單機Loop的功能改寫成可以一台 傳送一台接收。
2. 使用QAM通訊技術測試。
3. 使用OTSM通訊技術測試。
4. 產生實時更新的各種通訊分析圖。

新增功能：

1. 錯誤率偵測。
2. 不同條件下資料儲存。
3. 分析資料並畫統計圖。
4. 比較QAM與OTSM在不同距離下錯誤率與星座圖的差異。
5. 版本控制。
7. 相關內容及原理

本章將介紹2.1節SDR簡介、2.2節Matlab簡介、2.3節BPSK原理、2.4節OFDM原理，及其他相關內容以及原理。

1. SDR簡介

SDR(Software Defined Radio，軟體定義無線電)，是一種實現無線通訊的新概念和體制。在硬體中可以通過組件（例如混頻器，濾波器，放大器，調變器 / 解調器，檢測器等）實現，也可以通過軟體手段實施[1]。

圖2.1.1 Pluto SDR外觀

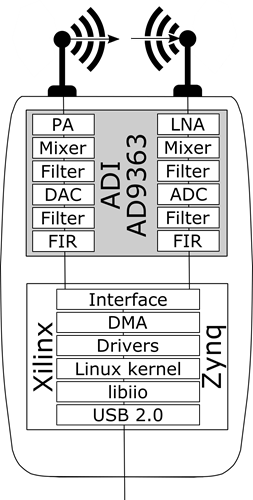
1. 優點：

靈活性高：SDR可以通過軟體編程更改運行模式，並能夠實現OTSM等多種通信方式，這使得SDR具有非常高的靈活性。

1. 容易升級：由於SDR採用軟體編程，因此可以更容易地升級軟件，而不需要更換硬件。
2. 低成本：由於SDR採用通用硬件，因此可以在各種平臺上運行，這降低了實現OTSM所需的硬件成本。
3. 較小的體積和重量：由於SDR採用通用硬件，因此相比傳統的OTSM系統，SDR系統的體積和重量要小得多。
4. 缺點：
5. 需要較強的處理能力：SDR系統需要在通用硬件上運行OTSM等多種通信方式，因此需要較強的處理能力。
6. 需要更多的功率：由於SDR系統需要較強的處理能力，因此需要更多的功率，這可能會導致更高的能耗。
7. 需要更高的技術水平：由於SDR系統需要軟體編程，因此需要更高的技術水平來實現OTSM等通信方式。

以下說明專題所使用的SDR(Pluto SDR)規格：

1. ADC 和 DAC 取樣速率： 65.2kSPS 到 61.44MSPS
2. 頻寬 : 200 kHz to 20 MHz
3. ADC 和 DAC 解析度：12位
4. 頻率精度：±25ppm
5. 頻率範圍：325MHz到3800MHz
6. Tx 輸出功率：7dBm
7. Rx 雜訊係數：<3.5dB
8. Rx 和 Tx 調製精度(EVM)：-34dB
9. USB：2.0 (for PWR/DATA)
10. 內核：雙ARM Cortex®-A9-667 MHz

圖2.1.2 Pluto SDR內部結構

1. Matlab簡介

圖2.2.1 Matlab

MATLAB（Matrix Laboratory，矩陣實驗室）是由美國The MathWorks公司出品的商業數學軟體。MATLAB是一種用於演算法開發、資料視覺化、資料分析以及數值計算的進階技術計算語言和互動式環境。除矩陣運算、繪製函數/資料圖像等常用功能外，MATLAB還可用來建立使用者介面，以及呼叫其它語言（包括C、C++、Java、Python、FORTRAN）編寫的程式。

MATLAB主要用於數值運算，但利用為數眾多的附加工具箱，它也適合不同領域的應用，例如控制系統設計與分析、影像處理、深度學習、訊號處理與通訊、金融建模和分析等。另外還有配套軟體套件Simulink提供視覺化開發環境，常用於系統類比、動態/嵌入式系統開發等方面[2]。

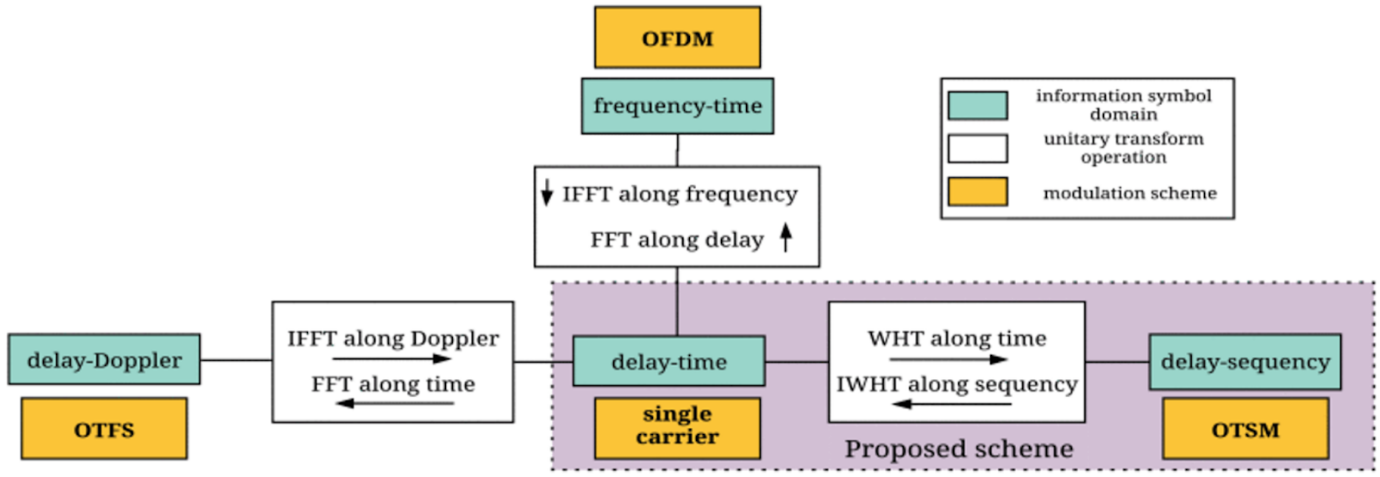
1. 功能
   1. 線性代數、統計、傅立葉分析、篩選、最佳化以及數值積分等的數學函數。
   2. 可用於視覺化資料的二維和三維圖形函數。
   3. 工具箱(Toolbox)：訊號處理和通訊，使電腦能和Pluto SDR溝通。
2. QAM原理

QAM（Quadrature Amplitude Modulation）是一種數位調變技術，常用於數位通訊系統中，特別是在有限頻寬的載波上進行資料傳輸。 QAM透過同時變化正弦波的振幅和相位來傳輸數位訊號。

QAM調變利用訊號的振幅（振幅）和相位來編碼多個位元資料。 對於QAM，它在複平面上繪製了一個訊號點的星座圖，其中的每個點代表特定的訊號模式，該模式可以表示多個位元。

1. OTSM原理
2. 定義

這是一種新型的單載波調變方案將資訊符號多路復用在延時-序列域中，其中序列度定義為單位時間間隔內的零交叉次數。O是一種將訊息符號在延時-序列域中進行多路復用，並將其在時間上進行行列交錯的調製方式。這種組合的方法結合了序列多重化和時分複用的特性。透過這種方式，訊息符號經過Walsh-Hadamard變換後，按照特定的序列度進行排列，然後在時間上進行交錯排布，以實現傳輸。被視為一種低複雜度的調變方案，適用於高移動性的通訊環境，同時具有良好的效能。

****

1. 運作模式
2. 傳送端處理：在訊號傳輸之前，訊息符號被分塊並編碼為符號序列。
3. Walsh-Hadamard變換（WHT）：每個資訊符號區塊經過Walsh-Hadamard變換，對應到延遲-序列域。WHT的每一列代表一個Walsh函數，具有其獨特的序列度，類似於依照增加諧波數排序的傅立葉分量。
4. 行列交錯：對通過WHT的訊號進行行列交錯，這將在時域和序列域中對訊號進行交錯。這種行列交錯操作實現了時分複用和序列多重化的功能。
5. 頻道傳輸：經過行交錯後的訊號透過無線頻道傳輸。在傳輸過程中，訊號可能受到多路徑衰落和多普勒效應等影響。
6. 接收端處理：接收端接收到經由頻道傳送的訊號後，進行行列解交錯，即將時域與序列域的交錯逆向操作。接著進行逆WHT操作，將訊號映射回原始的資訊符號塊。

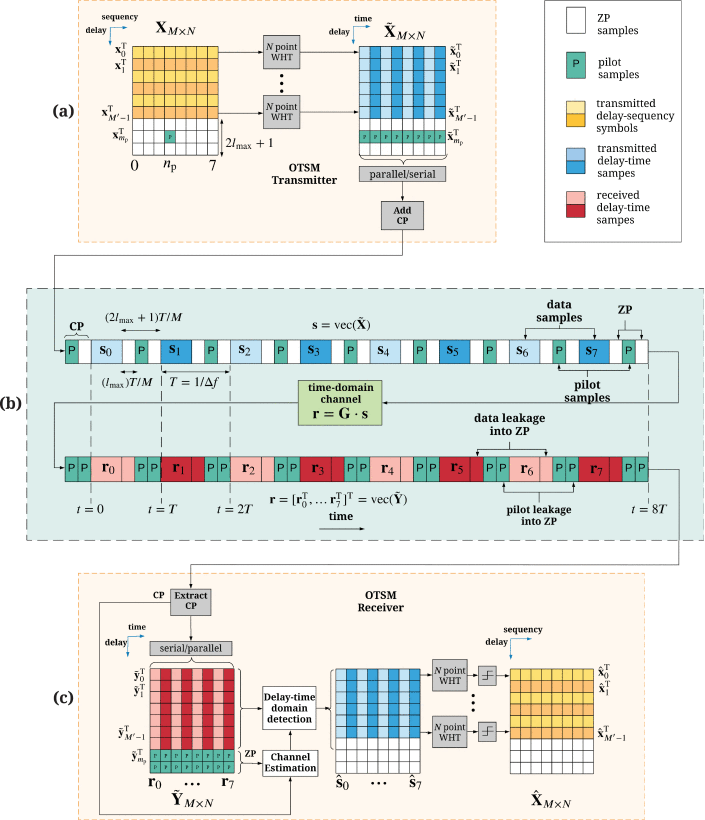


圖1.4.1顯示了 OTSM 收發器的分步操作。[3]

2. 流程
3. ADALM Pluto環境搭建
4. 硬體安裝

圖3.1.1 ADALM Pluto SDR接上電腦

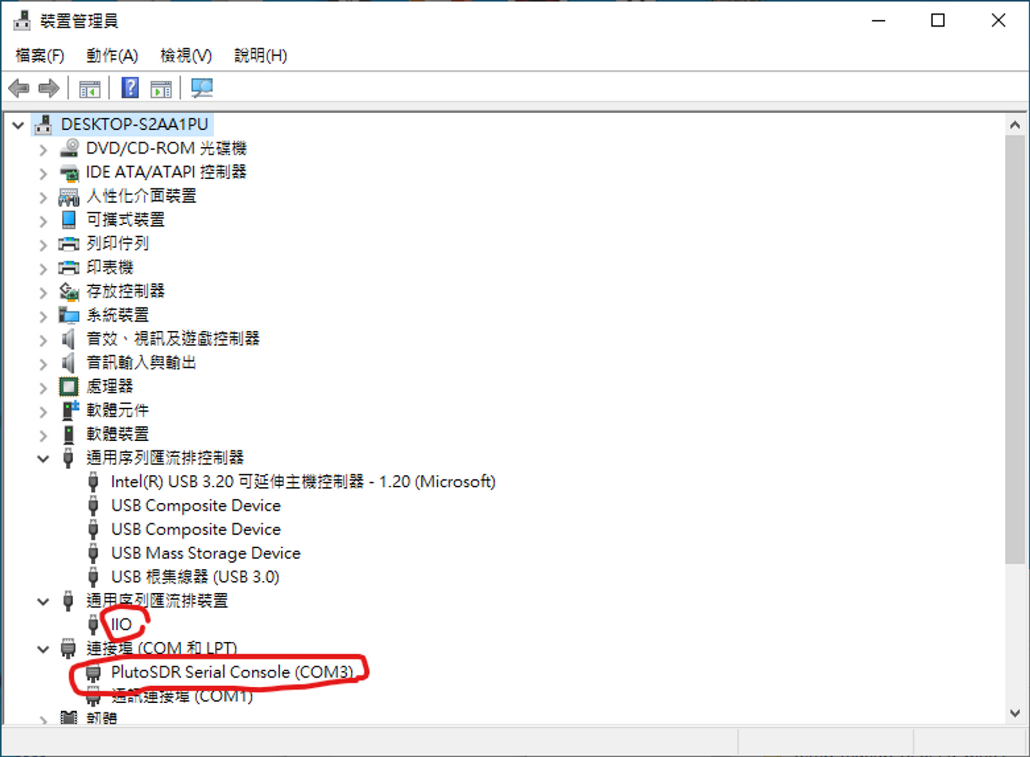
1. 安裝usb驅動
   1. 下載PlutoSDR-M2k-USB-Drivers.exe
   2. 安裝驅動
   3. 驅動安裝完成

圖3.1.2 驅動安裝完成示意圖

1. 安裝機器與安裝IIO程式
   1. 下載軟體adi-osc-setup.exe
   2. 安裝
2. 安裝Matlab端驅動
   1. 開啟MatLab依照圖片操作
   2. 按下APPS然後按下Get More Apps

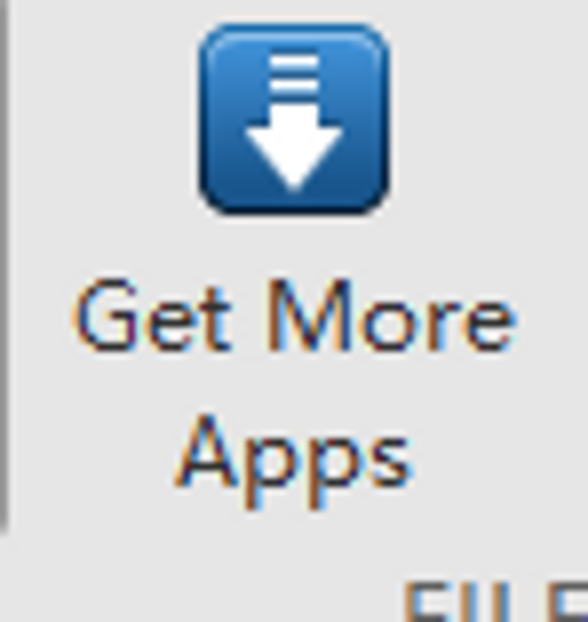


圖3.1.3 Get More Apps示意圖

* 1. 搜尋Pluto

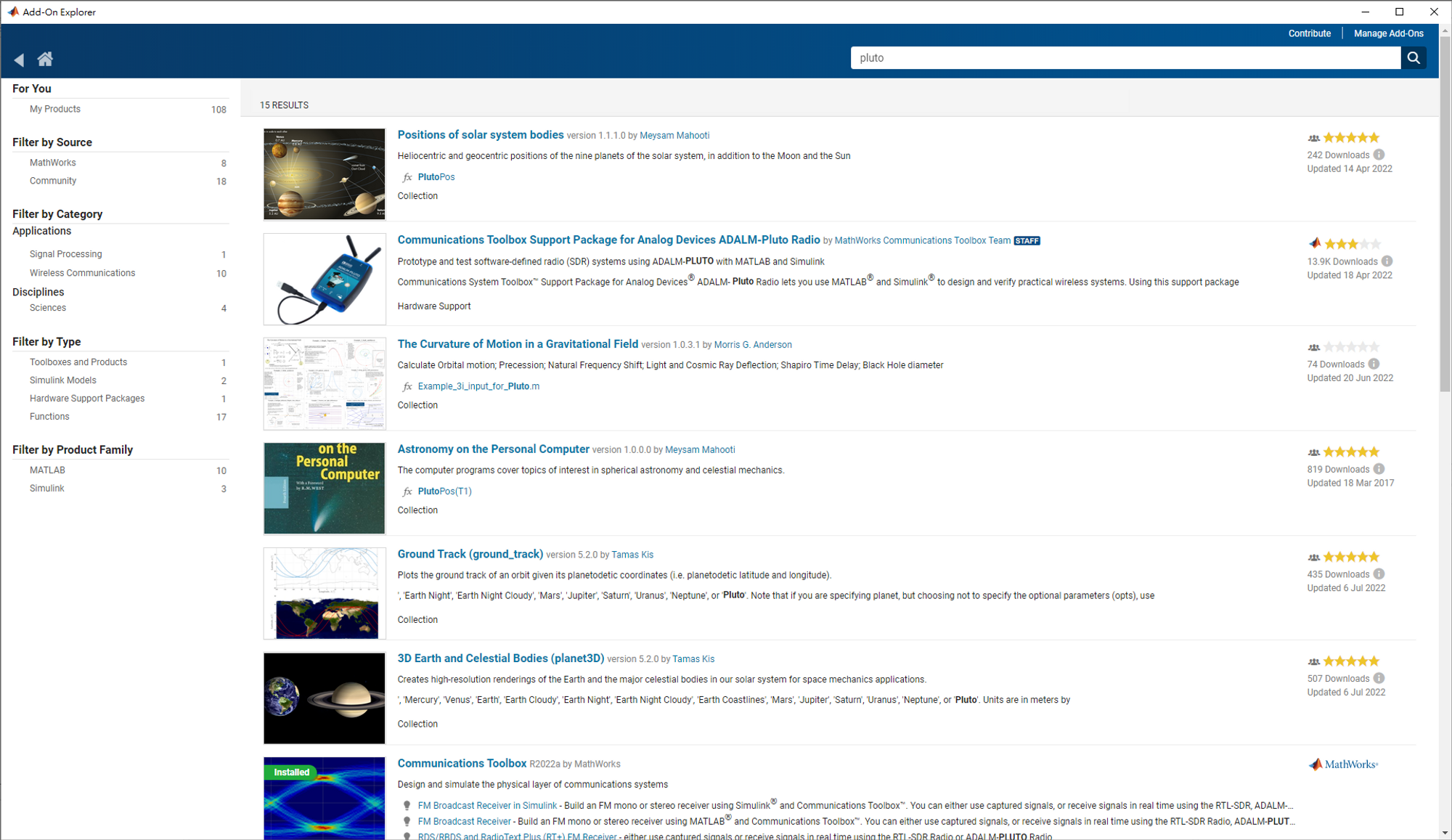


圖3.1.4搜尋Pluto示意圖

* 1. 選取並安裝(過程中有跳出甚麼就按”是”)

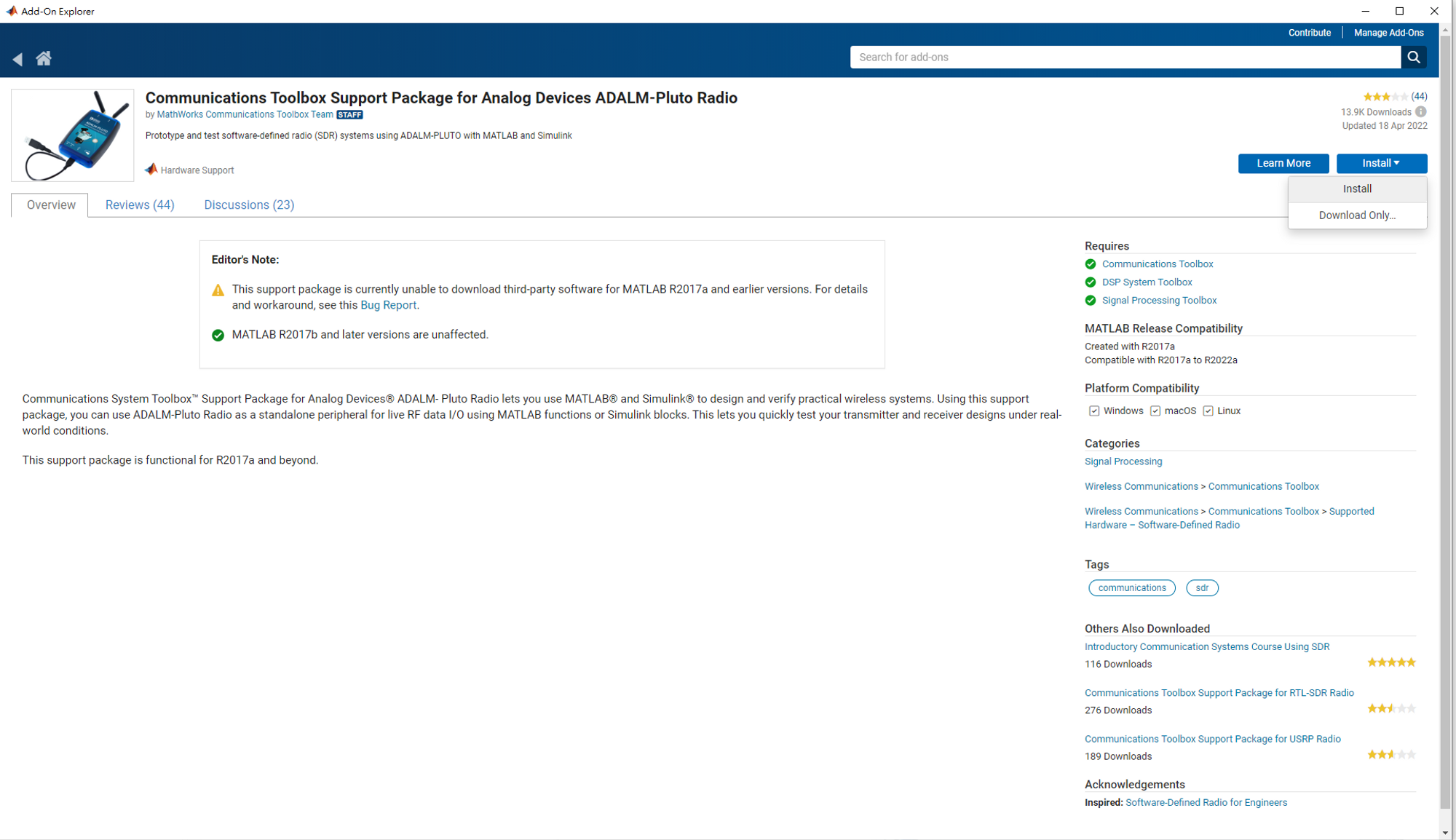
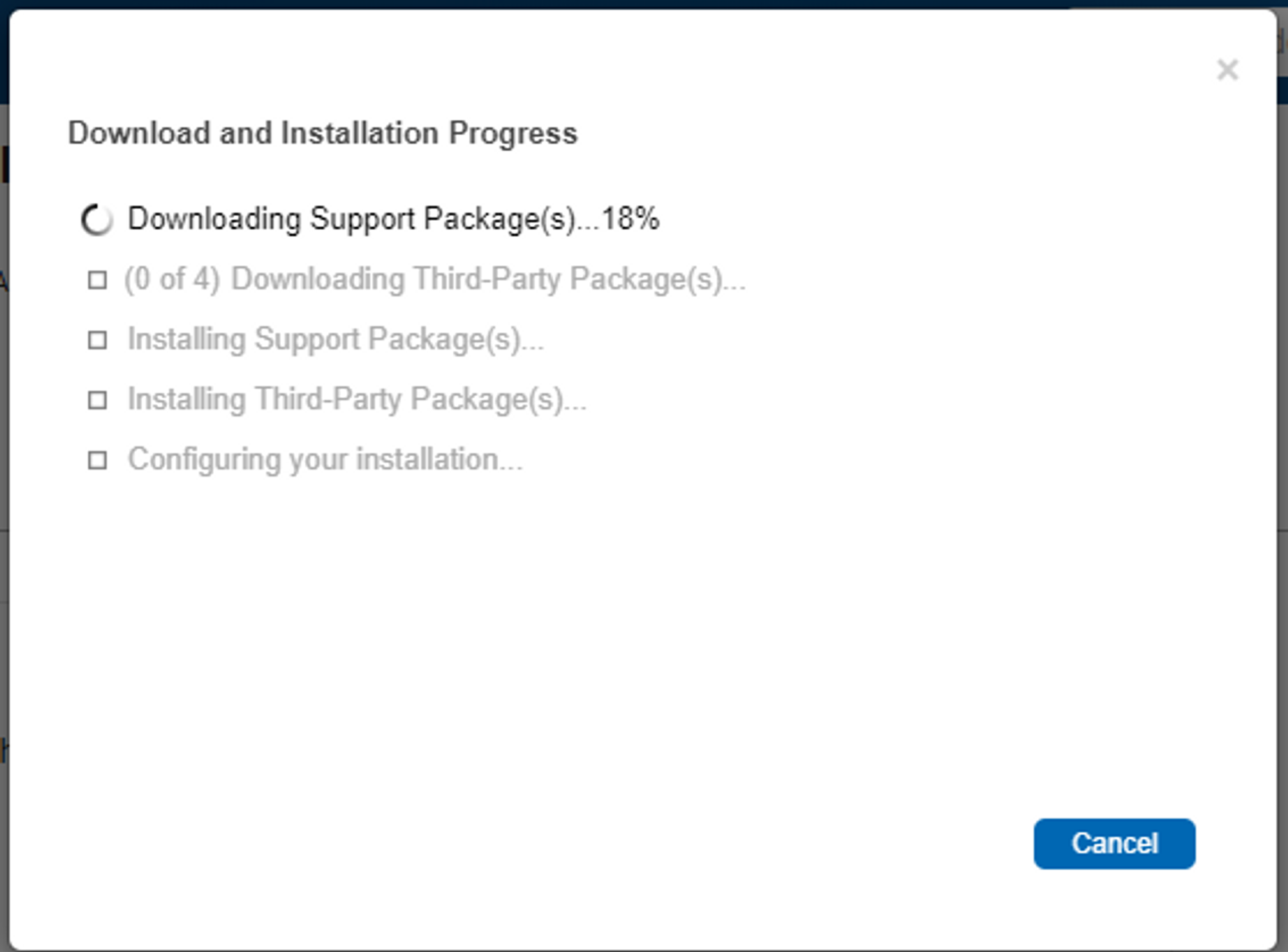


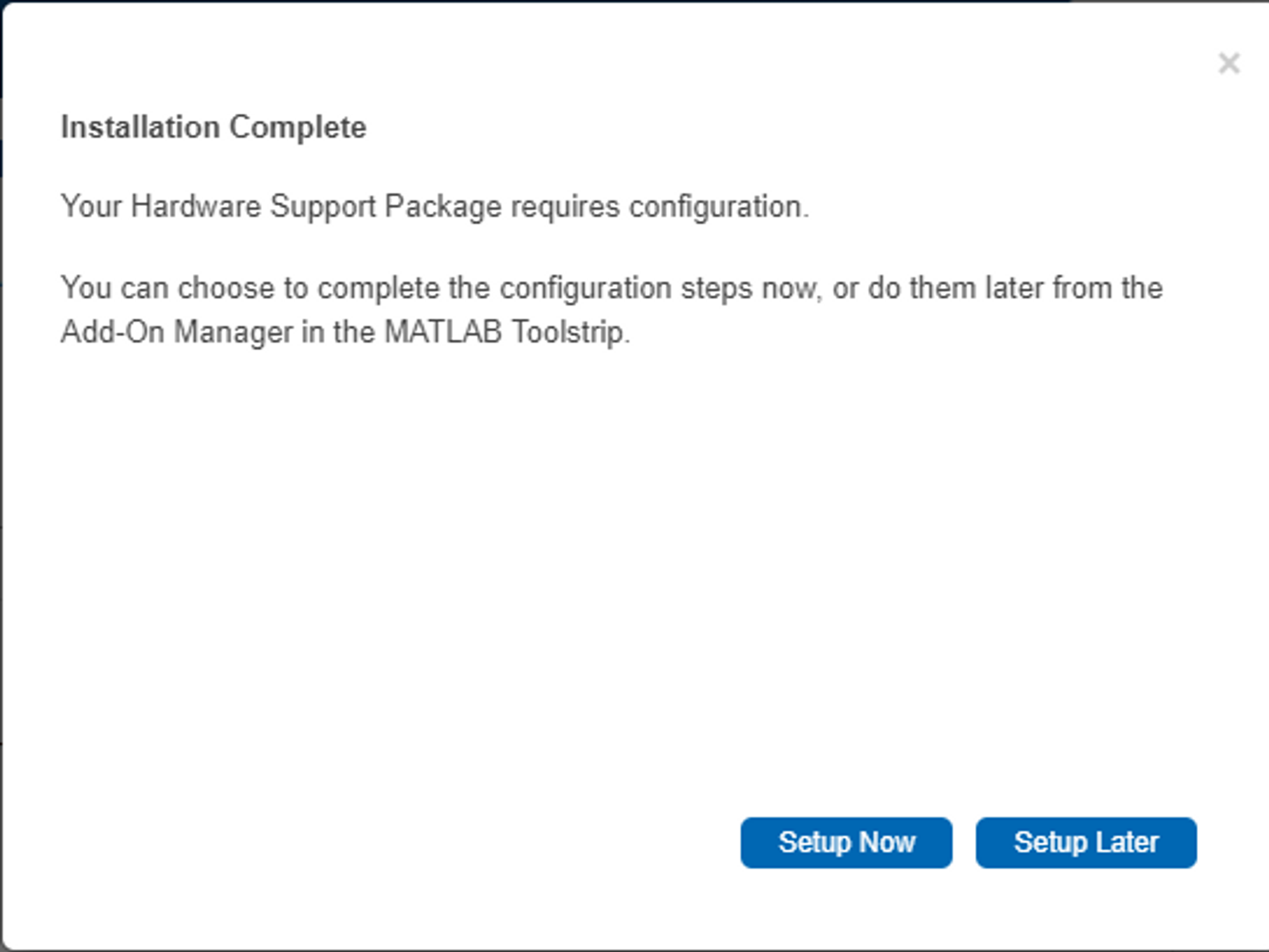
圖3.1.5選取並安裝示意圖

圖3.1.6安裝中示意圖



* 1. 按"setup now”

圖3.1.7重要步驟示意圖



* 1. 下面要取消勾選!!

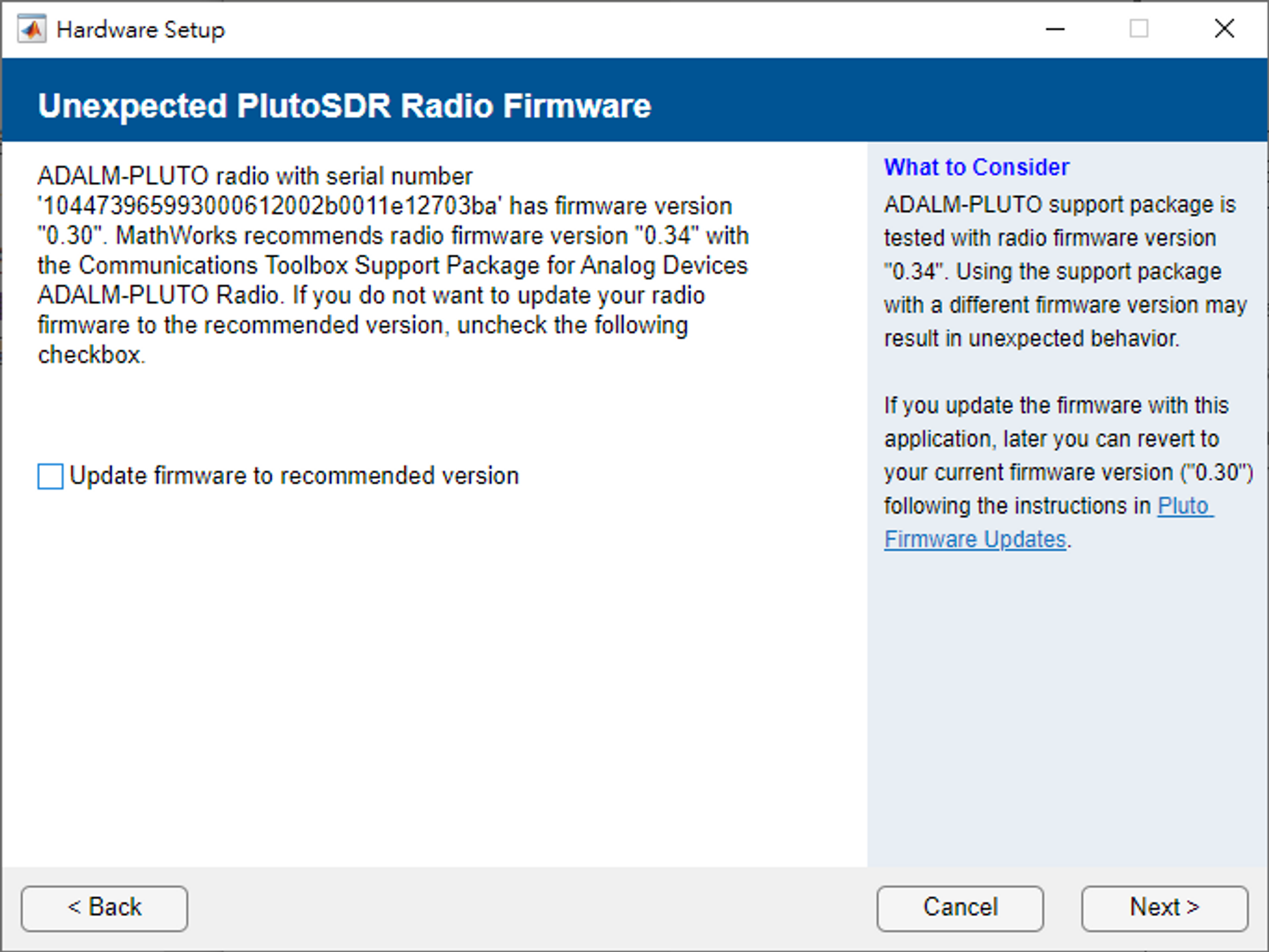


圖3.1.8重要步驟示意圖

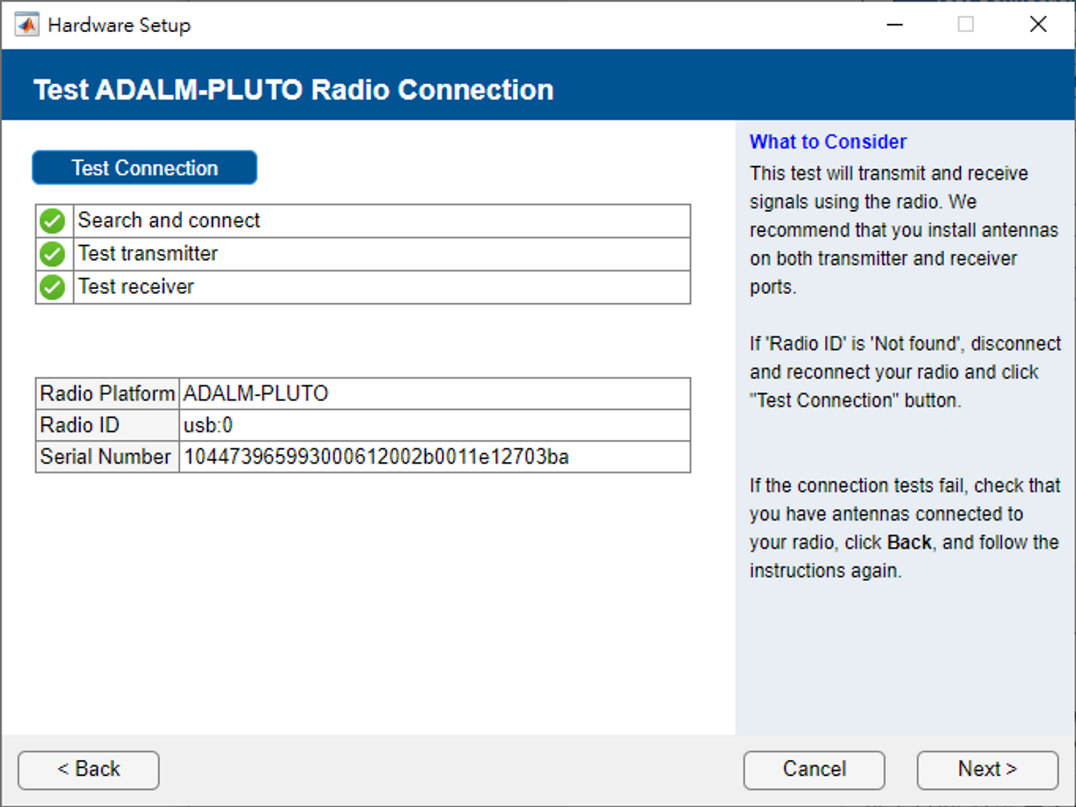
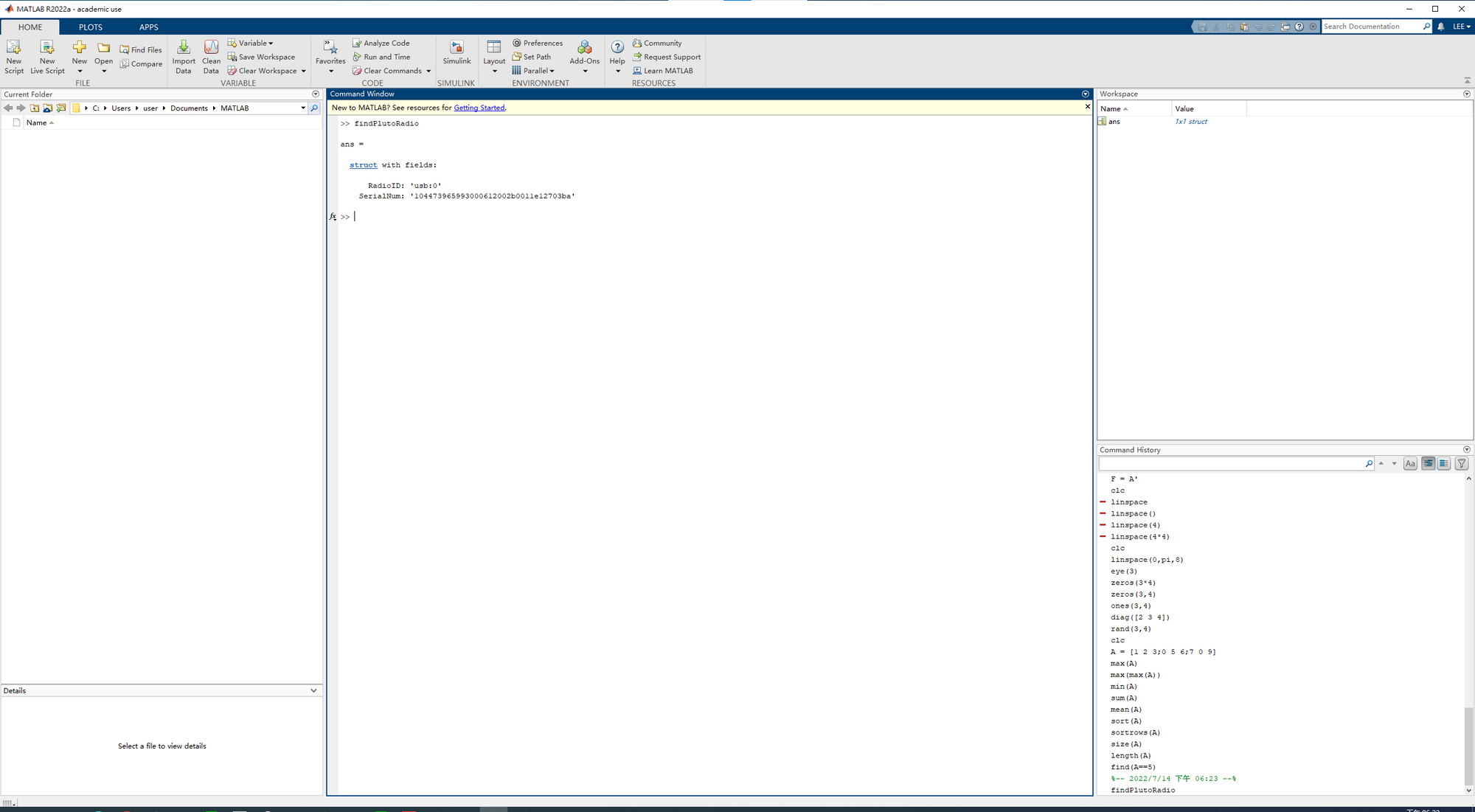
* 1. 測試完成

圖3.1.9測試Pluto示意圖

* 1. Matlab端輸入指令測試是否正確連線

終端機輸入findPlutoRadio查看輸出是否如圖3.1.10

圖3.1.10輸出結果圖



1. IIO driver安裝
   1. 下載libiio-0.17.g5bdc242-Windows-setup.exe
   2. 安裝驅動
2. Matlab中Mingw安裝
   1. 下載mingw.mlpkginstall
   2. 安裝驅動
3. 進度控制與分配
4. 甘特表

如表4.1.1是我們專題所有工作的甘特表，包括每一個部分完成的時間。使用灰階表示完成程度，淺灰表示1-50%、深灰表示50-99%、黑色表示100%

表4.1.1甘特表

|  | 2023年 | | | | | | | | 2024年 | | | | | 負責人 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 製  作  項  目 | 五月 | 六月 | 七月 | 八月 | 九月 | 十月 | 十一月 | 十二月 | 一月 | 二月 | 三月 | 四月 | 五月 |
| 題  目  討  論 |  |  |  |  |  |  |  |  |  |  |  |  |  | 全員 |
| 計  劃  書  撰  寫 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 資  料  收  集 |  |  | |  |  |  |  |  |  |  |  |  |  |
| 熟  悉  Mat  Lab  語  法 |  | | |  |  |  |  |  |  |  |  |  |  |
| 設  備  材  料  準  備 |  | | | |  |  |  |  |  |  |  |  |  |
| OTSM  調  變  研  究 |  |  |  |  | |  |  |  |  |  |  |  |  |
| FB  操作 |  |  |  |  | |  |  |  |  |  |  |  |  |
| Mat  lab  傳  送  端  與  接  收  端 |  |  |  |  | |  |  |  |  |  |  |  |  |
| 簡  易  通  道  測  試 |  |  |  |  | |  |  |  |  |  |  |  |  |
| 天  線  傳  送 |  |  |  |  |  | |  |  |  |  |  |  |  |
| 示  波  器  判  讀  資  料 |  |  |  |  |  |  | |  |  |  |  |  |  |
| 通  道  估  測 |  |  |  |  |  |  |  | | |  |  |  |  |
| 解  決  無  線  通  訊  同  步  問  題 |  |  |  |  |  |  |  |  |  | |  |  |  |
| 測試 |  |  |  |  |  |  |  |  |  |  | |  |  |
| 報  告  整  理  與  製  作 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 專  題  發  表 |  |  |  |  |  |  |  |  |  |  |  |  | |

1. 工作分配

萬子謙：版本控制、修改OFDM同步程式、創建.mat使資料能被保存

陳昱升：將Loop.m分成Tx.m與Rx.m

陳軍銓：新增BER.m、新增DataAnalysis.m、新增figplusfig.m

陳建霖：SDR資料查詢、OFDM資料查詢

姚辰彥：降低錯誤率、修正等化器

卓家葳：通道參數設定、通道估測

1. 成果
2. Tx傳送端和Rx接收端

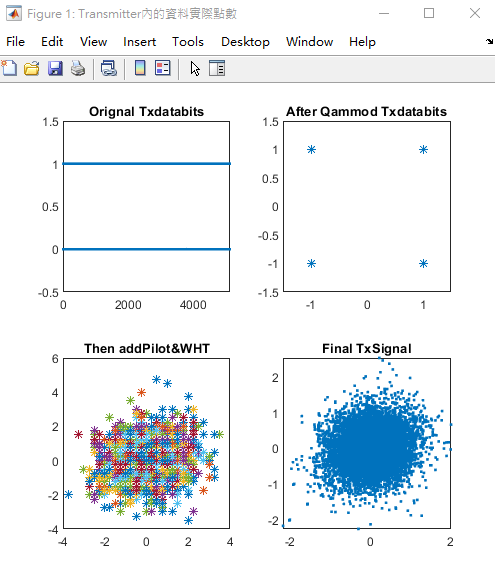


圖5.1.1 Tx傳送端展示圖



圖5.1.2 Rx接收端展示圖

結果圖

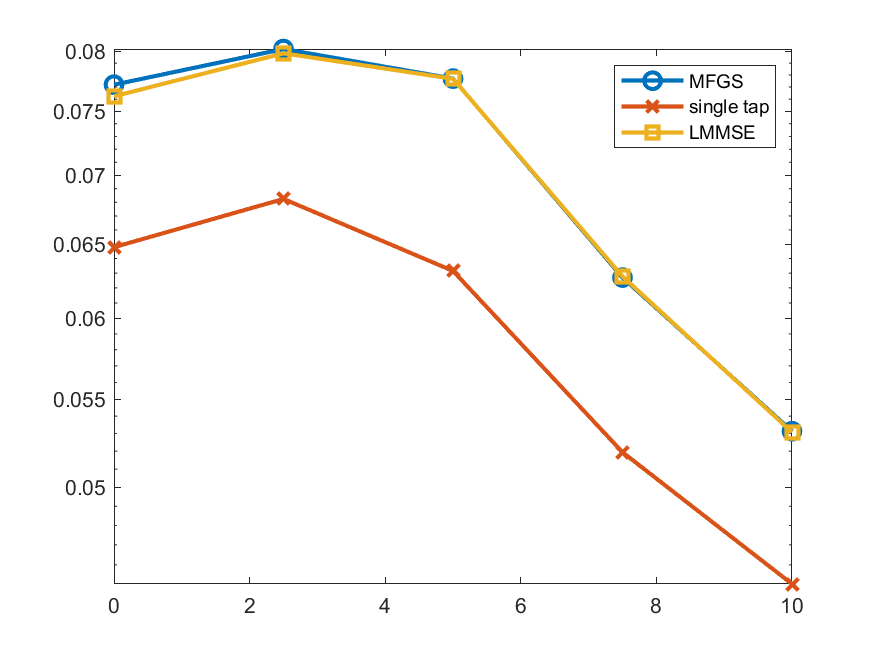


圖5.2.1 三種偵測器的錯誤率

1. 檢討與討論

本章將詳細敘述上學期評量結果回應表之修正部分，以及組員在專題中所遇到的種種問題。

1. 檢討

評審評語1：僅單人報告

解決方法1: 以多人報告呈現

評審評語2：建議過寬的(近代通訊技術)主題稍加微調至 明確的名字

解決方法2：改成明確的專題名稱，以SDR實現OTSM技術

評審評語3：未把SDR與OFDM及OTSM的關係說明清楚

解決方法3：著重在OTSM的東西上面

評審評語4：貢獻未說明清楚

解決方法4：仔細地把貢獻羅列出來

評審評語5：建議在成果上能以硬體輸入做評估

解決方法5：SDR為軟體定義無線電，衡量之後，還是以寫軟體的東西為主，硬體方面只有用到PLUTO

評審評語6：OTSM程式碼在網路上可找到。

解決方法6：網路上可以找到的是理想的數學模型，專題做的是會跑實際通道的模型，衡量之後，不做修改

1. 結論

參考學長與網路上的MATLAB程式(僅有OFDM收發機&OTSM數學模型)，自行消化理解後，改變OFDM改成OTSM系統並修改Pluto相關硬體設定。

嘗試將Pluto實現一台傳送一台接收(未完成)。

使用MATLAB生成比較圖形，可以更直觀的表示不同情況下的數據變化。比如，可以繪製不同接收演算法的錯誤率比較

參考資料

1. 軟體無線電wiki：<https://zh.wikipedia.org/zh-tw/%E8%BD%AF%E4%BB%B6%E6%97%A0%E7%BA%BF%E7%94%B5>
2. Pluto:<https://www.mouser.tw/images/marketingid/2017/img/175161332_Analog_Devices_ADALM-PLUTOActiveLearningModule.png?v=072922.0344>
3. Matlab簡介：<https://zh.wikipedia.org/zh-tw/MATLAB>
4. QAM:<https://upload.wikimedia.org/wikipedia/commons/9/90/QAM16_Demonstration.gif>
5. OTSM論文: <https://ieeexplore.ieee.org/abstract/document/9460770>

附檔

附檔1 BER.m程式碼

function BER ( TxDataBits , RxDataBits , NoFoundDataTimes , AllFoundDataTimes)

TxDataBitsReshape = reshape(TxDataBits, [], 1);

RxDataBitsReshape = reshape(RxDataBits, [], 1);

BER = sum(abs(TxDataBitsReshape-RxDataBitsReshape));

ER = BER/length(TxDataBitsReshape);

subplot(235);

axis off;

text(1.5,1.0,['BER: ',num2str(ER\*100),' %']);

text(1.5,0.8,['ER: ',num2str(BER),' bits']);

text(1.5,0.6,['NoFoundData: ',num2str(NoFoundDataTimes),' 次']);

text(1.5,0.4,['AllFoundData: ',num2str(AllFoundDataTimes),' 次']);

text(1.5,0.2,['無收訊率: ',num2str(NoFoundDataTimes / AllFoundDataTimes \*100),' %']);

global BERData;

BERData = ER;

end

附檔2 Loop.m程式碼

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*此範例僅適用於單機自收自發使用-立鎂科技\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clearvars -except times;close all;warning off; %預設環境

set(0,'defaultfigurecolor','w');

%加入path

addpath ..\..\library

addpath ..\..\library\matlab

addpath ..\..\code\matlab\OFDM

%刪除.mat

if(0)

Delete\_mat;

end

%設定接收執行次數

set\_looptimes = 1000;

% 載入資料

if exist('ScattorData.mat','file')

load('ScattorData.mat');

else

AllRxDataSymbEqAverage = zeros(0,0); %儲存Scattor資料

end

if exist('BERData.mat','file')

load('BERData.mat');

else

AllBERData = zeros(0,0); %儲存BER資料

end

%設定pluto IP

ip = '192.168.2.1';

%設定與進入TX函式

upsample=4; %過取樣取4倍，數位還原類比後比較可以不失真

txdata = Transmitter(upsample);

txdata = round(txdata .\* 2^15); %[(32768\*4096)/1024] -2/2

%% Transmit and Receive using MATLAB libiio 串接pluto

% System Object Configuration

s = iio\_sys\_obj\_matlab; % MATLAB libiio Constructor

s.ip\_address = ip;

s.dev\_name = 'ad9361';

s.in\_ch\_no = 2;

s.out\_ch\_no = 2;

s.in\_ch\_size = length(txdata);

s.out\_ch\_size = length(txdata).\*10;

s = s.setupImpl();

input = cell(1, s.in\_ch\_no + length(s.iio\_dev\_cfg.cfg\_ch));

output = cell(1, s.out\_ch\_no + length(s.iio\_dev\_cfg.mon\_ch));

% Set the attributes of AD9361

input{s.getInChannel('RX\_LO\_FREQ')} = 2400e6;

input{s.getInChannel('RX\_SAMPLING\_FREQ')} = 40e6;

input{s.getInChannel('RX\_RF\_BANDWIDTH')} = 20e6;

input{s.getInChannel('RX1\_GAIN\_MODE')} = 'manual';%% slow\_attack manual

input{s.getInChannel('RX1\_GAIN')} = 1;

input{s.getInChannel('TX\_LO\_FREQ')} = 2400e6;

input{s.getInChannel('TX\_SAMPLING\_FREQ')} = 40e6;

input{s.getInChannel('TX\_RF\_BANDWIDTH')} = 20e6;

%% PLOT TX for Evan\_debug 畫出TX的時域圖與頻譜圖

TimeScopeTitleStr = 'OFDM-TX-Baseband I/Q Signal';

SpectrumTitleStr = 'OFDM-TX-Baseband Signal Spectrum';

samp\_rate = 40e6;

scope1 = dsp.TimeScope('SampleRate', samp\_rate, ...

'Title', TimeScopeTitleStr, ...

'TimeSpan', 1/samp\_rate\*(length(txdata)+100), ...

'YLimits', [-5 5], ...

'ShowLegend', true, ...

'ShowGrid', true, ...

'Position', [0 300 400 400]);

step(scope1,txdata);

release(scope1);

spectrum1 = dsp.SpectrumAnalyzer('SampleRate', samp\_rate, ...

'SpectrumType', 'Power density', ...

'SpectralAverages', 10, ...

'YLimits', [-70 60], ...

'Title', SpectrumTitleStr, ...

'YLabel', 'Power spectral density', ...

'Position', [500 300 400 400]);

% Show power spectral density of captured burst

step(spectrum1,txdata);

release(spectrum1);

global NoFoundDataTimes;

NoFoundDataTimes = 0; %未找到data的次數(沒有通過同步)

global AllFoundDataTimes;

AllFoundDataTimes= 0; %嘗試接收data的次數(也是環圈執行次數)

AllRxDataSymbEq = zeros(0,0); %暫時儲存scattor資料

AllBERDataCol = zeros(0,0); %暫時儲存BER資料

for loop\_times = 1:set\_looptimes %確保重複監測

%% PLOT RX 畫出RX的圖

for i=i:4 %由於PLUTO-USB數據量受限~因此RX使用此FOR-LOOP等待TX數據進入 by Evan 2019-04-16

fprintf('Transmitting Data Block %i ...\n',i);

input{1} = real(txdata);

input{2} = imag(txdata);

output = stepImpl(s, input);

fprintf('Data Block %i Received...\n',i);

end

I = output{1};

Q = output{2};

Rx = I+1i\*Q;

figure(1); clf;

set(gcf,'name','立鎂科技-RX實際I/Q接收狀態'); % EVAN for debug OK

subplot(121);

plot(I);

hold on;

plot(Q);

subplot(122);

pwelch(Rx, [],[],[], 40e6, 'centered', 'psd');

% 20230301新增將PSD圖疊起來

hold on;

pwelch(txdata, [],[],[], 40e6, 'centered', 'psd');

legend('Rx', 'Tx')

%% PLOT RX

Rx = Rx(:,1);

global RxDataSymbEq;

Receiver(Rx(1:4:end));

% 儲存scattor資料(一筆資料108\*18所以每18列一筆)

AllRxDataSymbEq = [AllRxDataSymbEq RxDataSymbEq];

global TxDataBits;

global RxDataBits;

global BERData;

AllFoundDataTimes = AllFoundDataTimes + 1;

BER ( TxDataBits , RxDataBits , NoFoundDataTimes , AllFoundDataTimes);

AllBERDataCol = [AllBERDataCol BERData];

%pause(0.1);

%break;

end

%% 結束

fprintf('Transmission and reception finished\n');

% Read the RSSI attributes of both channels

rssi1 = output{s.getOutChannel('RX1\_RSSI')};

% rssi2 = output{s.getOutChannel('RX2\_RSSI')};

s.releaseImpl();

% 平均某一固定距離的BER跟星座圖

AllBERData = [AllBERData;AllBERDataCol];

for i=1:108

for j=1:18

for k=1:set\_looptimes

AllRxDataSymbEq(i,j) = AllRxDataSymbEq(i,j)+AllRxDataSymbEq(i,j+18\*(k-1));

end

end

end

AverageData = zeros(108,18);

AverageData = AllRxDataSymbEq(1:108,1:18)./set\_looptimes;

AllRxDataSymbEqAverage = [AllRxDataSymbEqAverage AverageData];

% 將資料導出到.mat中

save('ScattorData.mat', 'AllRxDataSymbEqAverage');

save('BERData.mat', 'AllBERData');

附檔3 Tx.m程式碼

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*此範例僅適用於單機自收自發使用-立鎂科技\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clearvars -except times;close all;warning off;

set(0,'defaultfigurecolor','w');

addpath ..\..\library

addpath ..\..\library\matlab

addpath ..\..\code\matlab\OFDM

ip = '192.168.2.1';

upsample=4;

txdata = Transmitter(upsample);

txdata = round(txdata .\* 2^15); %[(32768\*4096)/1024] -2/2

%% Transmit and Receive using MATLAB libiio

% System Object Configuration

s = iio\_sys\_obj\_matlab; % MATLAB libiio Constructor

s.ip\_address = ip;

s.dev\_name = 'ad9361';

s.in\_ch\_no = 2;

s.out\_ch\_no = 2;

s.in\_ch\_size = length(txdata);

s.out\_ch\_size = length(txdata).\*10;

s = s.setupImpl();

input = cell(1, s.in\_ch\_no + length(s.iio\_dev\_cfg.cfg\_ch));

output = cell(1, s.out\_ch\_no + length(s.iio\_dev\_cfg.mon\_ch));

% Set the attributes of AD9361

input{s.getInChannel('RX\_LO\_FREQ')} = 2400e6;

input{s.getInChannel('RX\_SAMPLING\_FREQ')} = 40e6;

input{s.getInChannel('RX\_RF\_BANDWIDTH')} = 20e6;

input{s.getInChannel('RX1\_GAIN\_MODE')} = 'manual';%% slow\_attack manual

input{s.getInChannel('RX1\_GAIN')} = 1;

input{s.getInChannel('TX\_LO\_FREQ')} = 2400e6;

input{s.getInChannel('TX\_SAMPLING\_FREQ')} = 40e6;

input{s.getInChannel('TX\_RF\_BANDWIDTH')} = 20e6;

%改gain

%txPluto = sdrtx('Pluto','Gain',0)

%% PLOT TX for Evan\_debug

TimeScopeTitleStr = 'OFDM-TX-Baseband I/Q Signal';

SpectrumTitleStr = 'OFDM-TX-Baseband Signal Spectrum';

samp\_rate = 40e6;

scope1 = dsp.TimeScope('SampleRate', samp\_rate, ...

'Title', TimeScopeTitleStr, ...

'TimeSpan', 1/samp\_rate\*(length(txdata)+100), ...

'YLimits', [-5 5], ...

'ShowLegend', true, ...

'ShowGrid', true, ...

'Position', [0 300 400 400]);

step(scope1,txdata);

release(scope1);

spectrum1 = dsp.SpectrumAnalyzer('SampleRate', samp\_rate, ...

'SpectrumType', 'Power density', ...

'SpectralAverages', 10, ...

'YLimits', [-70 60], ...

'Title', SpectrumTitleStr, ...

'YLabel', 'Power spectral density', ...

'Position', [500 300 400 400]);

% Show power spectral density of captured burst

step(spectrum1,txdata);

release(spectrum1);

loop\_num\_of\_time = 0;%%宣告迴圈次數

a=0;%% 輸出迴圈次數

while(1)

%% PLOT RX

for i=i:4 %由於PLUTO-USB數據量受限~因此RX使用此FOR-LOOP等待TX數據進入 by Evan 2019-04-16

fprintf('Transmitting Data Block %i ...\n',i);

input{1} = real(txdata);

input{2} = imag(txdata);

writeTxData(s, input);

a=a+1;

fprintf('%d\n',a);

fprintf('Data Block %i Received...\n',i);

end

% I = output{1};

% Q = output{2};

% Rx = I+1i\*Q;

% figure(1); clf;

% set(gcf,'name','立鎂科技-RX實際I/Q接收狀態'); % EVAN for debug OK

% subplot(121);

% plot(I);

% hold on;

% plot(Q);

% subplot(122);

% pwelch(Rx, [],[],[], 40e6, 'centered', 'psd');

% %% PLOT RX

% Rx = Rx(:,1);

% Receiver(Rx(1:4:end));

% pause(0.1);

% %break;

end

fprintf('Transmission and reception finished\n');

% Read the RSSI attributes of both channels

rssi1 = output{s.getOutChannel('RX1\_RSSI')};

% rssi2 = output{s.getOutChannel('RX2\_RSSI')};

s.releaseImpl();

附檔4 Rx.m程式碼

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*此範例僅適用於單機自收自發使用-立鎂科技\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clearvars -except times;close all;warning off; %預設環境

set(0,'defaultfigurecolor','w');

%加入path

addpath ..\..\library

addpath ..\..\library\matlab

addpath ..\..\code\matlab\OFDM

%刪除.mat

if(0)

Delete\_mat;

end

%設定接收執行次數

set\_looptimes = 1000;

% 載入資料

if exist('ScattorData.mat','file')

load('ScattorData.mat');

else

AllRxDataSymbEqAverage = zeros(0,0); %儲存Scattor資料

end

if exist('BERData.mat','file')

load('BERData.mat');

else

AllBERData = zeros(0,0); %儲存BER資料

end

%設定pluto IP

ip = '192.168.2.1';

%設定與進入TX函式

upsample=4; %過取樣取4倍，數位還原類比後比較可以不失真

txdata = Transmitter(upsample);

txdata = round(txdata .\* 2^15); %[(32768\*4096)/1024] -2/2

%% Transmit and Receive using MATLAB libiio 串接pluto

% System Object Configuration

s = iio\_sys\_obj\_matlab; % MATLAB libiio Constructor

s.ip\_address = ip;

s.dev\_name = 'ad9361';

s.in\_ch\_no = 2;

s.out\_ch\_no = 2;

s.in\_ch\_size = length(txdata);

s.out\_ch\_size = length(txdata).\*10;

s = s.setupImpl();

input = cell(1, s.in\_ch\_no + length(s.iio\_dev\_cfg.cfg\_ch));

output = cell(1, s.out\_ch\_no + length(s.iio\_dev\_cfg.mon\_ch));

% Set the attributes of AD9361

input{s.getInChannel('RX\_LO\_FREQ')} = 2400e6;

input{s.getInChannel('RX\_SAMPLING\_FREQ')} = 40e6;

input{s.getInChannel('RX\_RF\_BANDWIDTH')} = 20e6;

input{s.getInChannel('RX1\_GAIN\_MODE')} = 'manual';%% slow\_attack manual

input{s.getInChannel('RX1\_GAIN')} = 1;

input{s.getInChannel('TX\_LO\_FREQ')} = 2400e6;

input{s.getInChannel('TX\_SAMPLING\_FREQ')} = 40e6;

input{s.getInChannel('TX\_RF\_BANDWIDTH')} = 20e6;

global NoFoundDataTimes;

NoFoundDataTimes = 0; %未找到data的次數(沒有通過同步)

global AllFoundDataTimes;

AllFoundDataTimes= 0; %嘗試接收data的次數(也是環圈執行次數)

AllRxDataSymbEq = zeros(0,0); %暫時儲存scattor資料

AllBERDataCol = zeros(0,0); %暫時儲存BER資料

for loop\_times = 1:set\_looptimes %確保重複監測

for i=i:4 %由於PLUTO-USB數據量受限~因此RX使用此FOR-LOOP等待TX數據進入 by Evan 2019-04-16

fprintf('Transmitting Data Block %i ...\n',i);

output = readRxData(s);

fprintf('Data Block %i Received...\n',i);

end

I = output{1};

Q = output{2};

Rx = I+1i\*Q;

figure(1); clf;

set(gcf,'name','立鎂科技-RX實際I/Q接收狀態'); % EVAN for debug OK

subplot(121);

plot(I);

hold on;

plot(Q);

subplot(122);

pwelch(Rx, [],[],[], 40e6, 'centered', 'psd');

% 20230301新增將PSD圖疊起來

hold on;

pwelch(txdata, [],[],[], 40e6, 'centered', 'psd');

legend('Rx', 'Tx')

%% PLOT RX

Rx = Rx(:,1);global RxDataSymbEq;

Receiver(Rx(1:4:end));

% 儲存scattor資料(一筆資料108\*18所以每18列一筆)

AllRxDataSymbEq = [AllRxDataSymbEq RxDataSymbEq];

global TxDataBits;

global RxDataBits;

global BERData;

AllFoundDataTimes = AllFoundDataTimes + 1;

BER ( TxDataBits , RxDataBits , NoFoundDataTimes , AllFoundDataTimes);

AllBERDataCol = [AllBERDataCol BERData];

%pause(0.1);

%break;

end

%% 結束

fprintf('Transmission and reception finished\n');

% Read the RSSI attributes of both channels

rssi1 = output{s.getOutChannel('RX1\_RSSI')};

% rssi2 = output{s.getOutChannel('RX2\_RSSI')};

s.releaseImpl();

% 平均某一固定距離的BER跟星座圖

AllBERData = [AllBERData;AllBERDataCol];

for i=1:108

for j=1:18

for k=1:set\_looptimes

AllRxDataSymbEq(i,j) = AllRxDataSymbEq(i,j)+AllRxDataSymbEq(i,j+18\*(k-1));

end

end

end

AverageData = zeros(108,18);

AverageData = AllRxDataSymbEq(1:108,1:18)./set\_looptimes;

AllRxDataSymbEqAverage = [AllRxDataSymbEqAverage AverageData];

% 將資料導出到.mat中

save('ScattorData.mat', 'AllRxDataSymbEqAverage');

save('BERData.mat', 'AllBERData');

附檔5 Matlab\_iio程式修改

% 下--2023323

function writeTxData(obj, varargin)

if(obj.sys\_obj\_initialized == 0)

return;

end

% Implement the device configuration flow

for i = 1 : length(obj.iio\_dev\_cfg.cfg\_ch)

if(~isempty(varargin{1}{i + obj.in\_ch\_no}))

if(length(varargin{1}{i + obj.in\_ch\_no}) == 1)

new\_data = (varargin{1}{i + obj.in\_ch\_no} ~= obj.num\_cfg\_in(i));

else

new\_data = ~strncmp(char(varargin{1}{i + obj.in\_ch\_no}'), char(obj.str\_cfg\_in(i,:)), length(varargin{1}{i + obj.in\_ch\_no}));

end

if(new\_data == 1)

if(length(varargin{1}{i + obj.in\_ch\_no}) == 1)

obj.num\_cfg\_in(i) = varargin{1}{i + obj.in\_ch\_no};

str = num2str(obj.num\_cfg\_in(i));

else

for j = 1:length(varargin{1}{i + obj.in\_ch\_no})

obj.str\_cfg\_in(i,j) = varargin{1}{i + obj.in\_ch\_no}(j);

end

obj.str\_cfg\_in(i,j+1) = 0;

str = char(obj.str\_cfg\_in(i,:));

end

writeAttributeString(obj.iio\_dev\_cfg.cfg\_ch(i).ctrl\_dev, obj.iio\_dev\_cfg.cfg\_ch(i).port\_attr, str);

end

end

end

writeData(obj.libiio\_data\_in\_dev, varargin{1});

end

function ret = readRxData(obj)

varargout = cell(1, obj.out\_ch\_no + length(obj.iio\_dev\_cfg.mon\_ch));

[~, data] = readData(obj.libiio\_data\_out\_dev);

for i = 1 : obj.out\_ch\_no

varargout{i} = data{i};

end

% Implement the parameters monitoring flow

for i = 1 : length(obj.iio\_dev\_cfg.mon\_ch)

[~, val] = readAttributeDouble(obj.iio\_dev\_cfg.mon\_ch(i).ctrl\_dev, obj.iio\_dev\_cfg.mon\_ch(i).port\_attr);

varargout{obj.out\_ch\_no + i} = val;

end

ret=varargout;

end

%上2023323

附檔6 DataAnalysis.m程式碼

load('ScattorData.mat');

load('BERData.mat');

AllBERDataAverage = sum(AllBERData')./length(AllBERData(1,:));

subplot(121);plot(0:length(AllBERDataAverage(1,:))-1,AllBERDataAverage);axis([-0.1,length(AllBERDataAverage(1,:))-1+0.1,0,1]);

subplot(122);

for i = 1:length(AllRxDataSymbEqAverage(1,:))/18-1:length(AllRxDataSymbEqAverage(1,:))/18

plot(reshape(AllRxDataSymbEqAverage(1:108,18\*(i-1)+1:18\*i),1,108\*18).\*exp(-1i\*pi/4),'.');axis([-1.5,1.5,-1.5,1.5]);title('scattor after equalization'); axis square;

hold on;

legend\_text{i}=[' 距離 = ',num2str(i-1),' cm '];

end

legend(legend\_text{1},legend\_text{8});

% for i = 1:length(AllRxDataSymbEqAverage(1,:))/18

% plot(reshape(AllRxDataSymbEqAverage(1:108,18\*(i-1)+1:18\*i),1,108\*18).\*exp(-1i\*pi/4),'.');axis([-1.5,1.5,-1.5,1.5]);title('scattor after equalization'); axis square;

% hold on;

% legend\_text{i}=[' 距離 = ',num2str(i-1),' cm '];

%

% end

% legend(legend\_text);

附檔7 figplusfig.m程式碼

% 打開第一個.fig文件

fig1 = openfig('BPSK.fig');

fig2 = openfig('OFDM.fig');

% 獲取第一個圖形的子圖

axes1 = findobj(fig1, 'Type', 'Axes');

axes2 = findobj(fig2, 'Type', 'Axes');

figure;

subplot(1,2,1);

copyobj(allchild(axes1(2)), gca);

hold on;

copyobj(allchild(axes2(2)), gca);

subplot(1,2,2);

copyobj(allchild(axes1(1)), gca);

hold on;

copyobj(allchild(axes2(1)), gca);

全文完