國立宜蘭大學電子工程學系

專題成果報告書

|  |  |
| --- | --- |
| 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: 博士

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

**摘要**

OTSM（Orthogonal Time Sequence Multiplexing）通訊技術是一種新型的多重存取技術，它利用正交的時間序列來實現多用戶的數據傳輸。SDR（Software-Defined Radio，軟件定義無線電）則是一種無線通信技術，利用軟件而非硬件來實現無線電通信的功能。將SDR應用於OTSM通訊技術，可以有效提升系統的靈活性和性能。

**目錄**

[摘要 I](#_Toc168571696)

[目錄 II](#_Toc168571697)

[表目錄 IV](#_Toc168571698)

[圖目錄 V](#_Toc168571699)

[第一章 簡介 1](#_Toc168571700)

[壹、 計畫緣由 1](#_Toc168571701)

[貳、 目的 1](#_Toc168571702)

[參、 成果介紹 1](#_Toc168571703)

[第二章 相關內容及原理 3](#_Toc168571705)

[壹、 SDR簡介 3](#_Toc168571706)

[貳、 Matlab簡介 6](#_Toc168571707)

[參、 QAM原理 7](#_Toc168571708)

[肆、 OTSM原理 8](#_Toc168571709)

[第三章 流程 10](#_Toc168571710)

[壹、 ADALM Pluto環境搭建 10](#_Toc168571711)

[第四章 進度控制與分配 18](#_Toc168571712)

[壹、 甘特表 18](#_Toc168571713)

[貳、 工作分配 22](#_Toc168571714)

[第五章 成果 23](#_Toc168571715)

[壹、 Tx傳送端 23](#_Toc168571716)

[貳、 Rx接收端 27](#_Toc168571717)

[參、 結果圖 28](#_Toc168571718)

[第六章 檢討與討論 29](#_Toc168571719)

[壹、 檢討 29](#_Toc168571720)

[貳、 結論 30](#_Toc168571721)

[參考文獻 31](#_Toc168571722)

[附錄 32](#_Toc168571723)

[附檔1 OTSM\_demo\_loop.m程式碼 32](#_Toc168571724)

[附檔2 Transmitter.m程式碼 41](#_Toc168571725)

[附檔3 Receiver.m程式碼 45](#_Toc168571726)

**表目錄**

[表 1 甘特表 18](#_Toc168514051)

**圖目錄**

[圖 1 Pluto SDR外觀 3](#_Toc168524706)

[圖 2 Pluto SDR內部結構 5](#_Toc168524707)

[圖 3 Matlab 6](#_Toc168524708)

[圖 4不同離散資訊符號域與相應調製方案之間的關係 8](#_Toc168524709)

[圖 5顯示了 OTSM 收發器的分步操作 9](#_Toc168524710)

[圖 6 ADALM Pluto SDR接上電腦 10](#_Toc168524711)

[圖 7 驅動安裝完成示意圖 11](#_Toc168524712)

[圖 8 Get More Apps示意圖 12](#_Toc168524713)

[圖 9 搜尋Pluto示意圖 13](#_Toc168524714)

[圖 10 選取並安裝示意圖 13](#_Toc168524715)

[圖 11 安裝中示意圖 14](#_Toc168524716)

[圖 12 重要步驟示意圖 14](#_Toc168524717)

[圖 13 重要步驟示意圖 15](#_Toc168524718)

[圖 14 測試Pluto示意圖 16](#_Toc168524719)

[圖 15 輸出結果圖 17](#_Toc168524720)

[圖 16 Tx傳送端展示圖 23](#_Toc168524721)

[圖 17 PilotBit數據輸入 24](#_Toc168524722)

[圖 18 Preamble data輸入 25](#_Toc168524723)

[圖 19 取得TxDataBits 25](#_Toc168524724)

[圖 20 TxDataBits數據 26](#_Toc168524725)

[圖 21 Rx接收端展示圖 27](#_Toc168524726)

[圖 22 三種偵測器的錯誤率 27](#_Toc168524727)

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
8. 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 1 Pluto SDR外觀

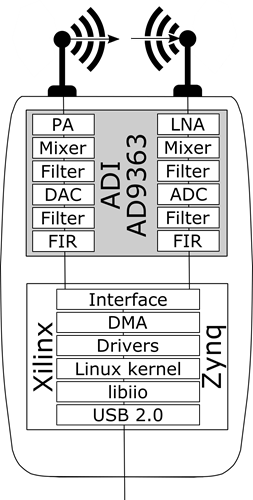
1. 優點：

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 Pluto SDR內部結構[2]

1. Matlab簡介

圖 3 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[3]。

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。

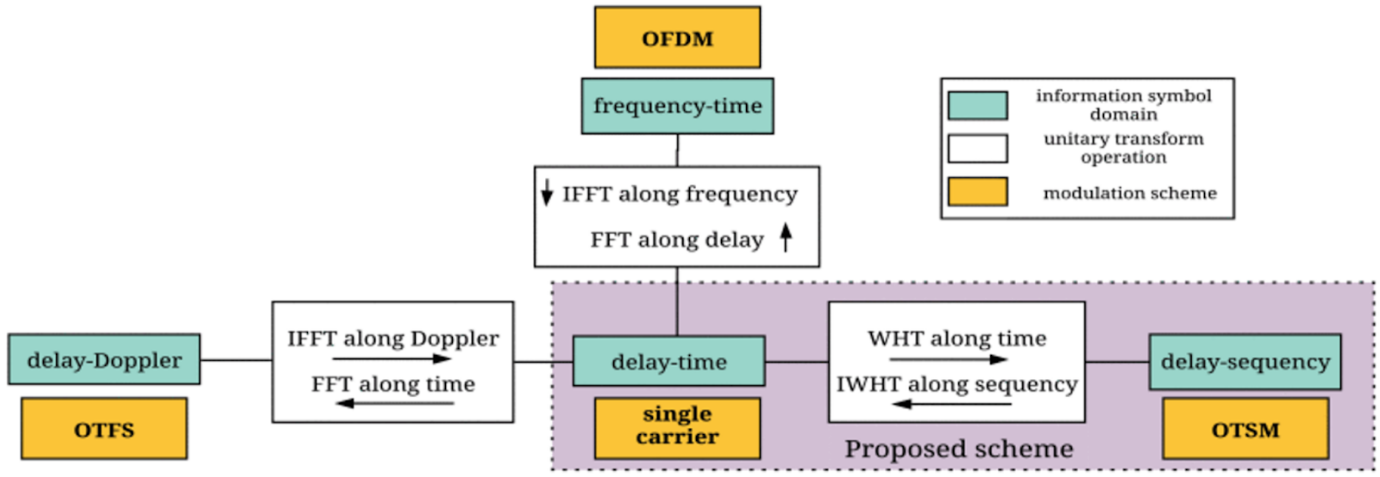
****

圖 4不同離散資訊符號域與相應調製方案之間的關係

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。

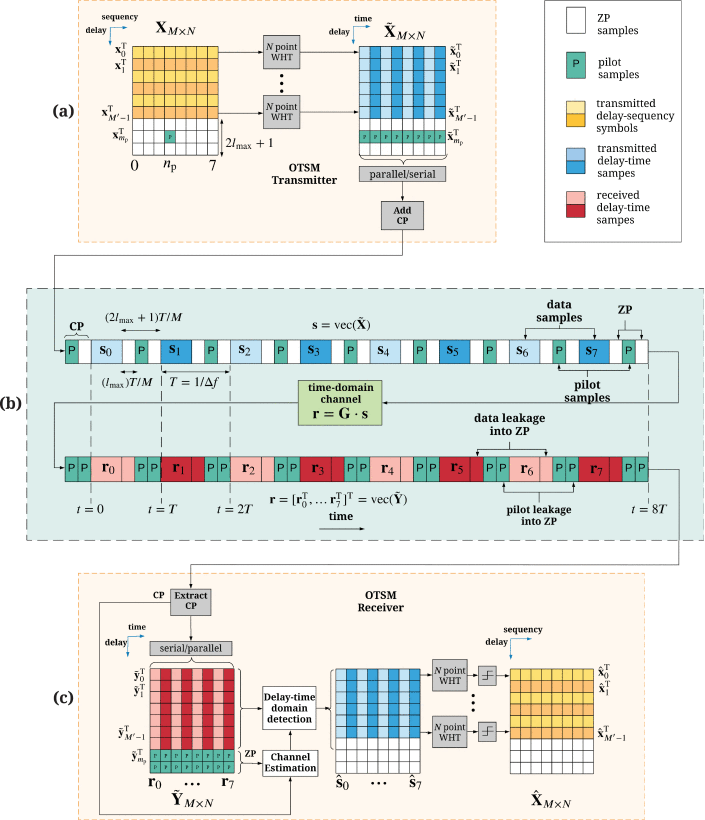


圖 5顯示了 OTSM 收發器的分步操作[4]

2. 流程
3. ADALM Pluto環境搭建
4. 硬體安裝

圖 6 ADALM Pluto SDR接上電腦

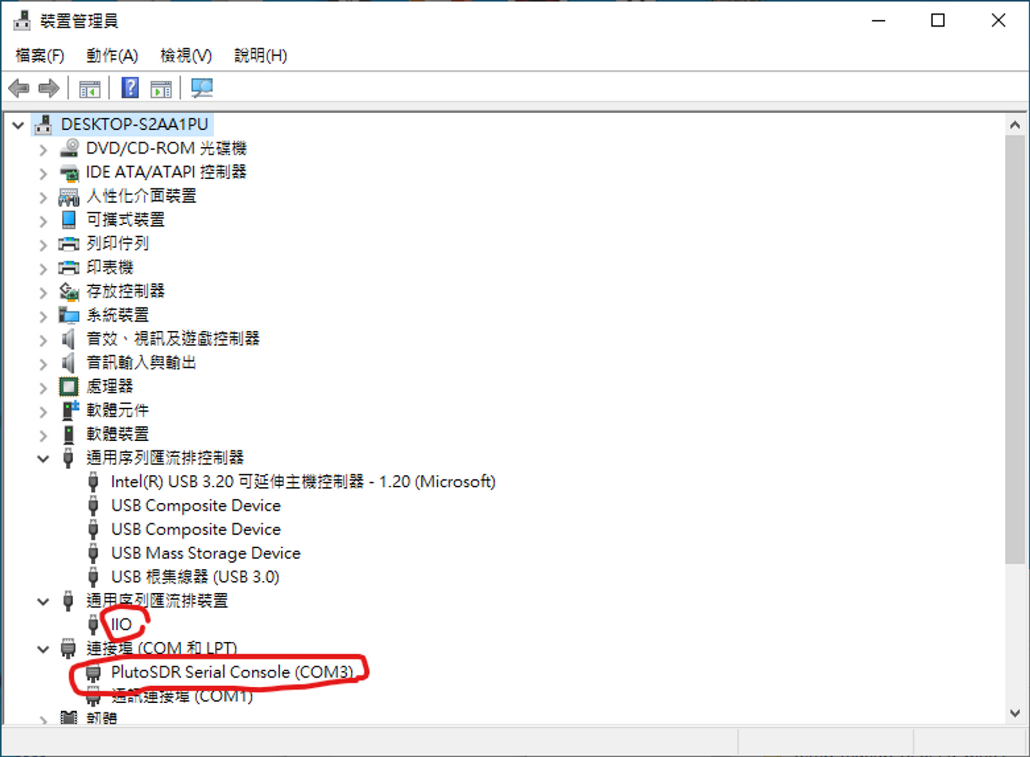
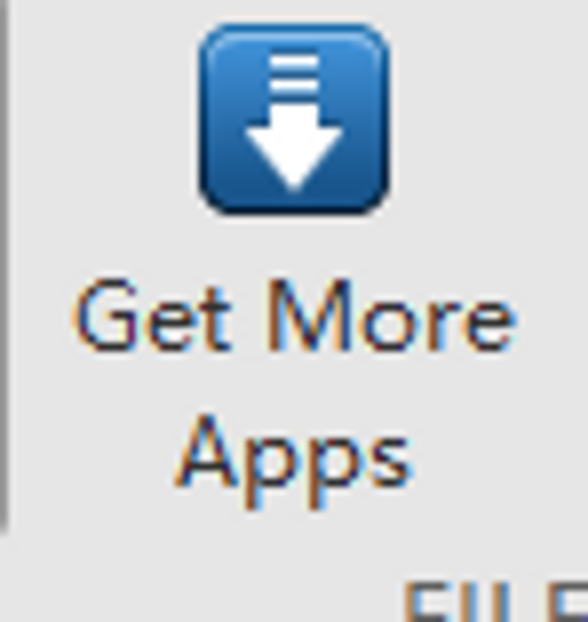
1. 安裝usb驅動
   1. 下載PlutoSDR-M2k-USB-Drivers.exe
   2. 安裝驅動
   3. 驅動安裝完成

圖 7 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 8 Get More Apps示意圖

* 1. 搜尋Pluto

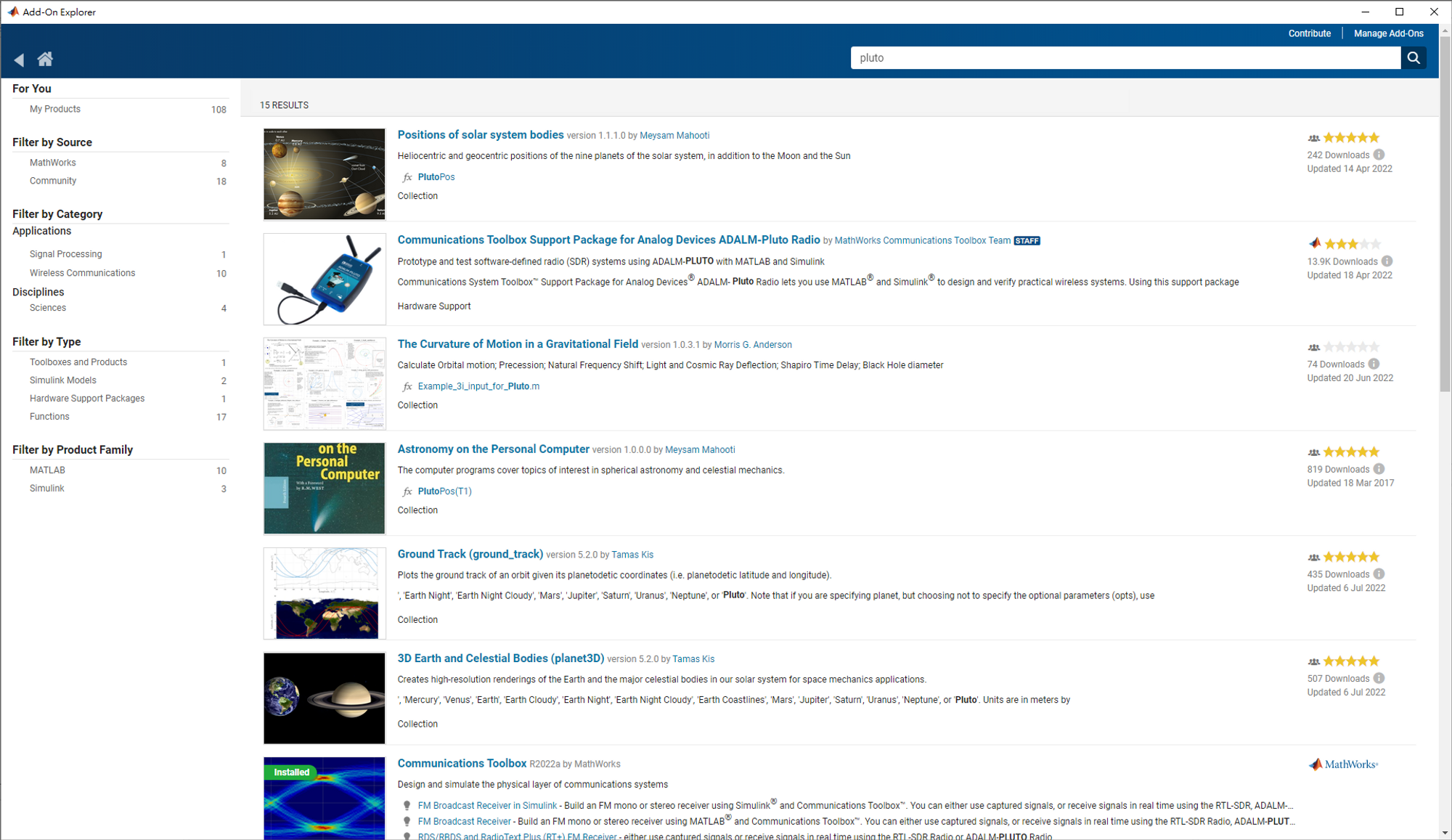


圖 9 搜尋Pluto示意圖

* 1. 選取並安裝(過程中有跳出甚麼就按”是”)

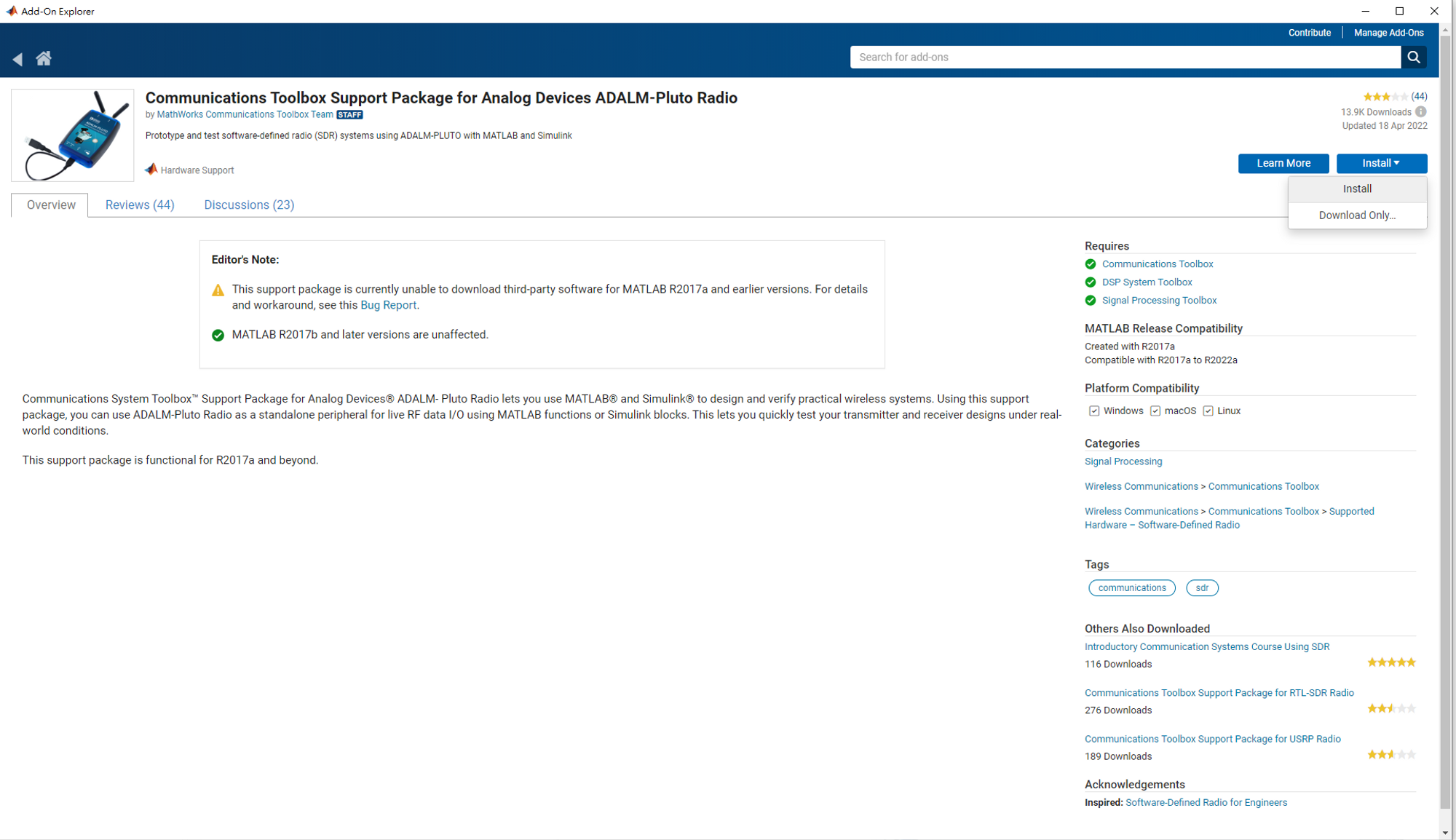
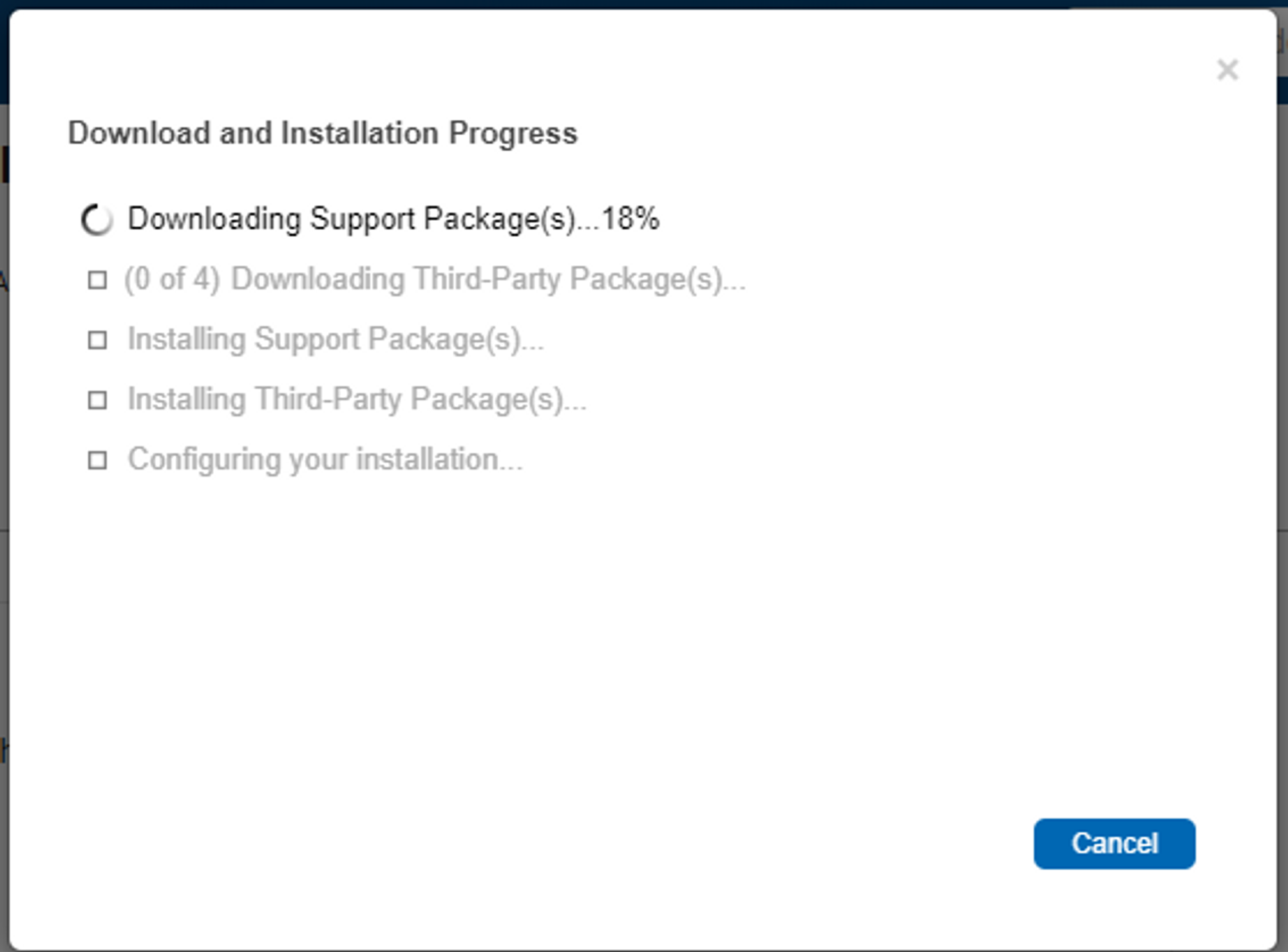


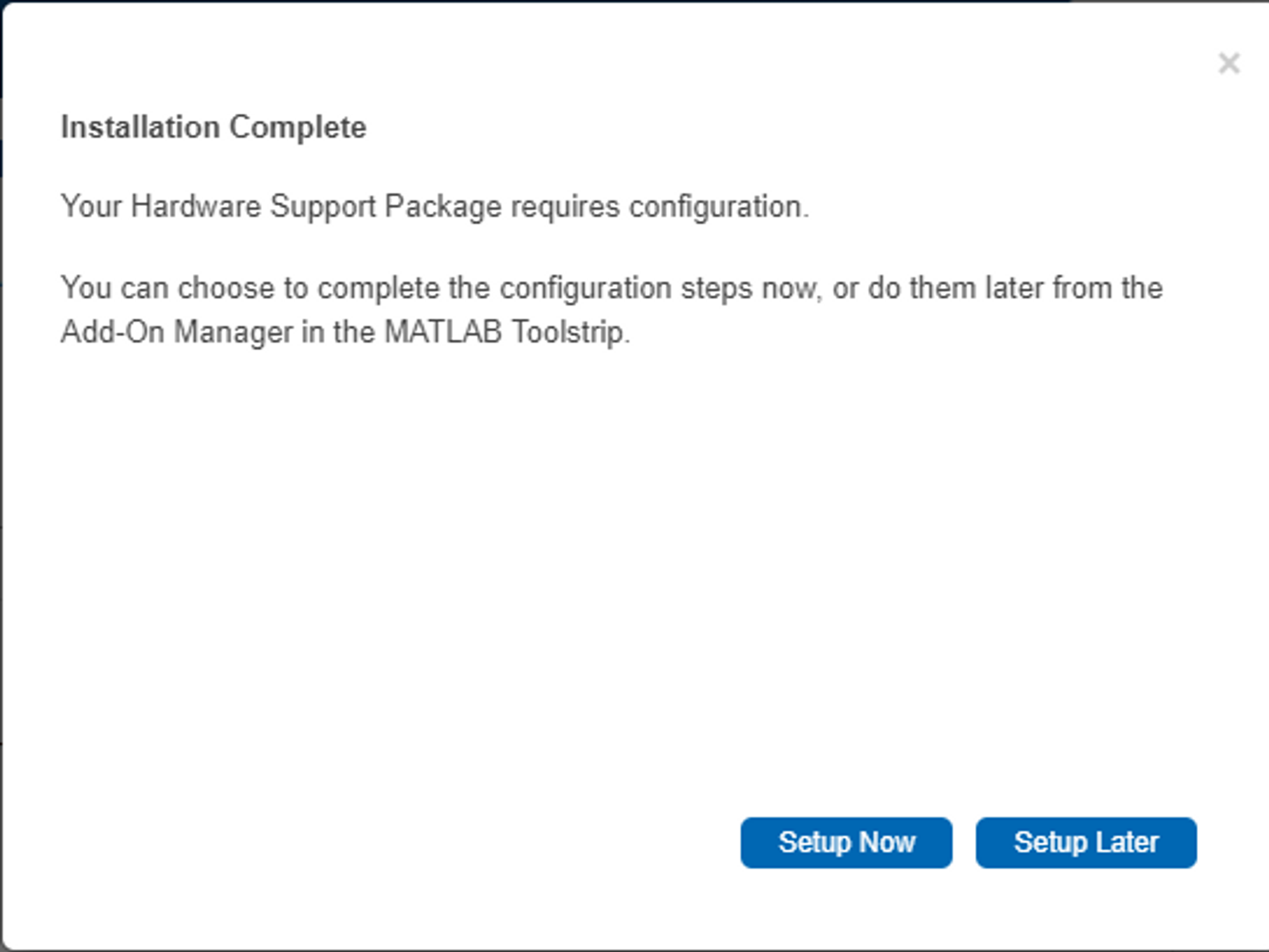
圖 10 選取並安裝示意圖

圖 11 安裝中示意圖



* 1. 按"setup now”

圖 12 重要步驟示意圖



* 1. 下面要取消勾選!!

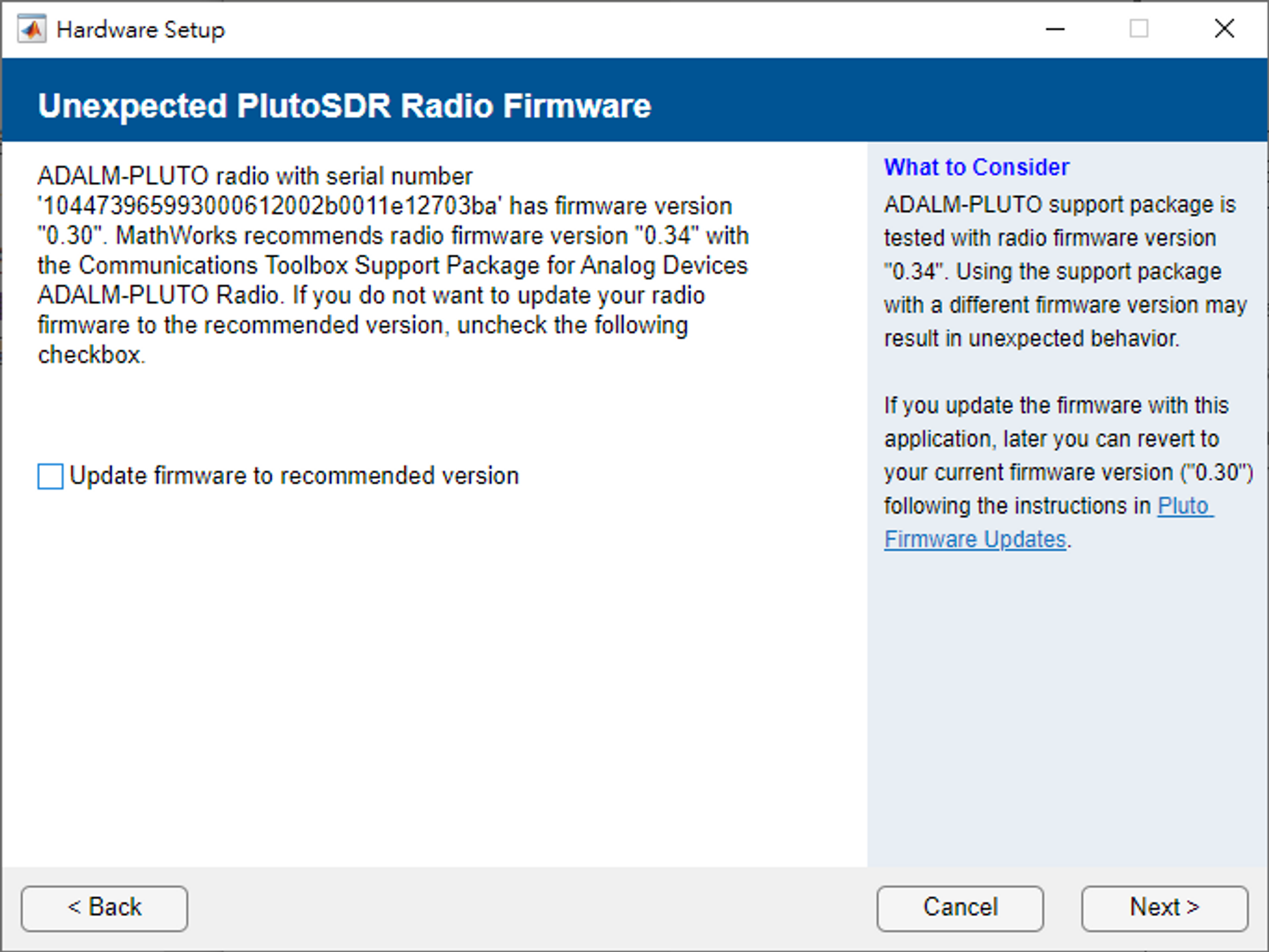


圖 13 重要步驟示意圖

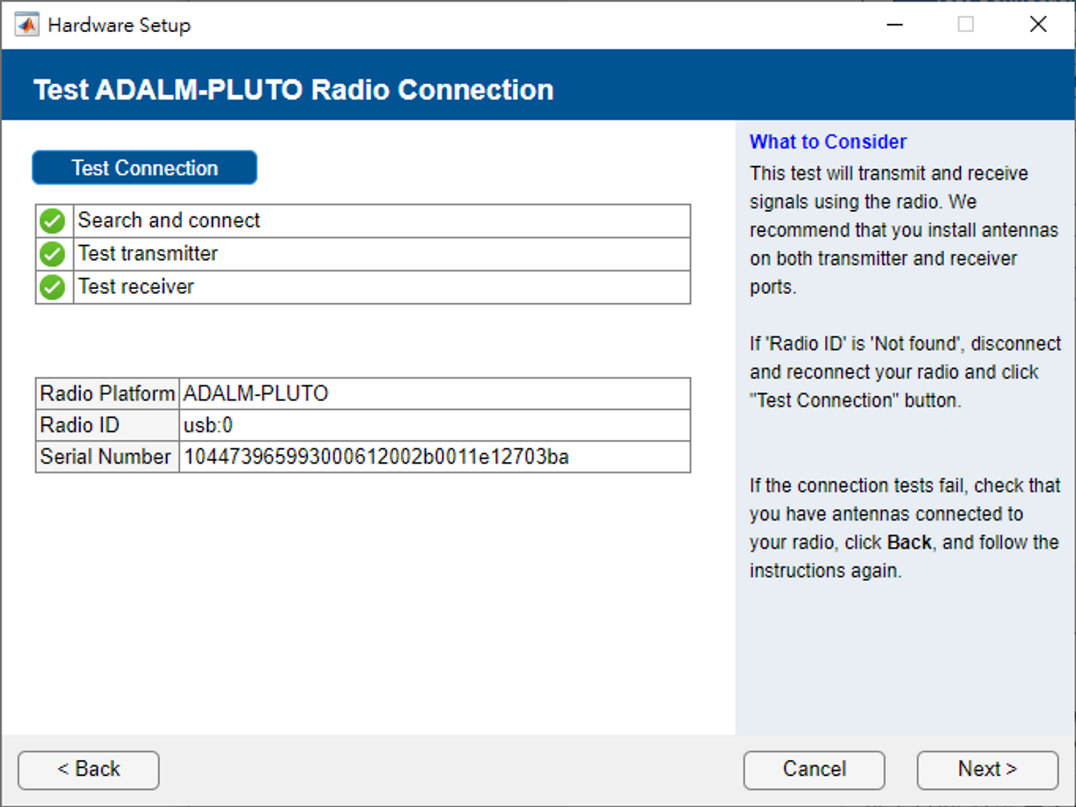
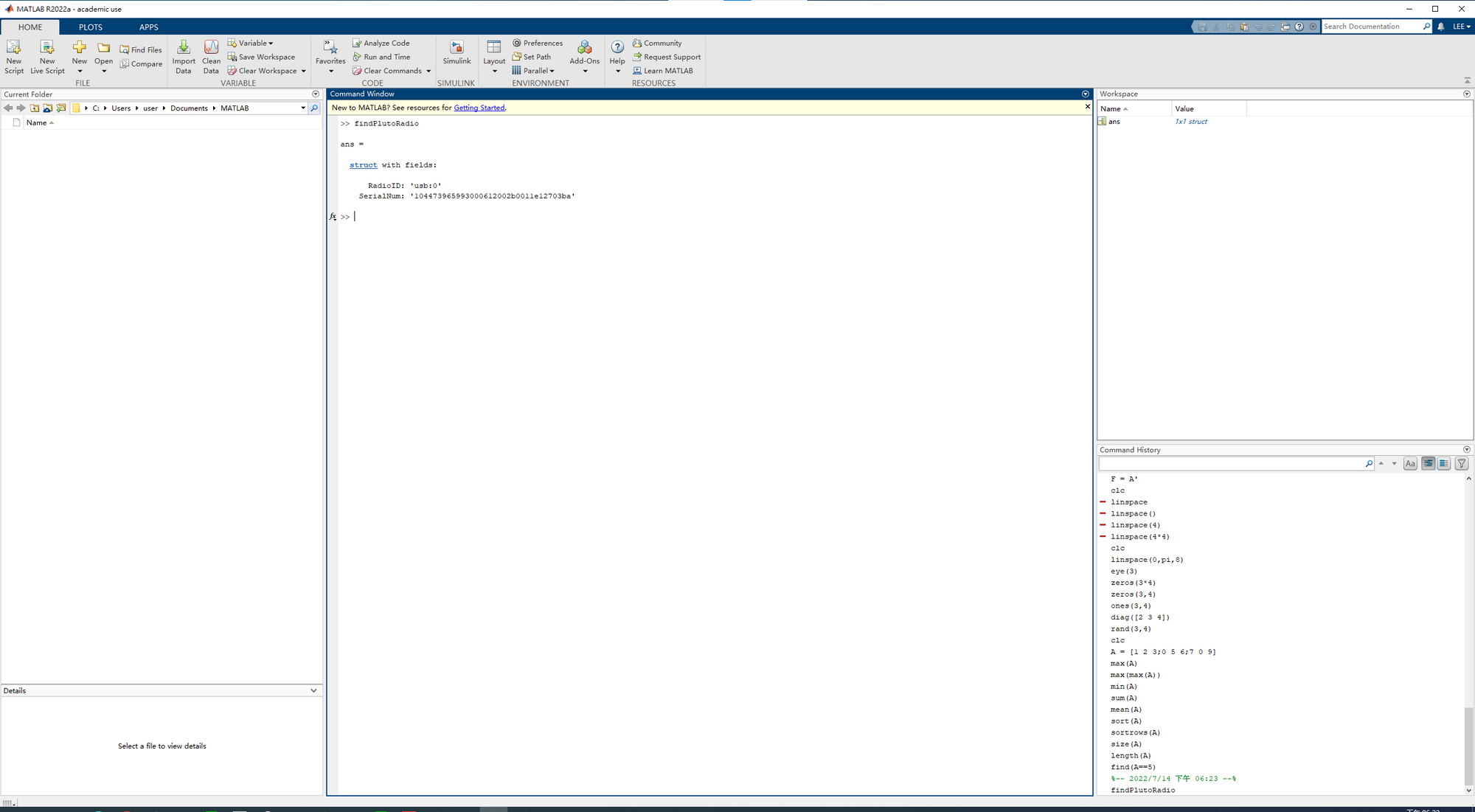
* 1. 測試完成

圖 14 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15

圖 15 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 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：嘗試拆開Tx及Rx的MATLAB code、幫忙CFO校正、海報製作

陳昱升：驅動安裝、OTSM程式替換、傳送資料加Pilot、Sync、通道設定、同步接收資料、通道估測(CFO校正)、OTSM數據整理、畫圖、版本控制

陳軍銓：協助OTSM程式替換OFDM

陳建霖：資料收集、協助通道估測(CFO校正)

姚辰彥：製作PPT、協助比較偵測器的錯誤率、協助通道設定

卓家葳：協助拆開Tx及Rx的MATLAB code、處理Github問題、協助同步接收資料

1. 成果

本章將展示matlab各個系統的實體照片,並詳細說明。在這裡我們將OTSM\_demo\_loop.m分成Tx 與Rx兩部分的成果展示, 進行程式的同步,第壹節為Tx傳送端經過WHT和QAM調變，第貳節為Rx接收端，延遲-多普勒網格符號放置、經過WHT 矩陣，再經過Pilot切割和解調變，第參節為結果圖，呈現三種偵測器的錯誤率。

最後，創建 mat 使資料能被保存為成果展示。

1. Tx傳送端
2. Tx數據經過WHT

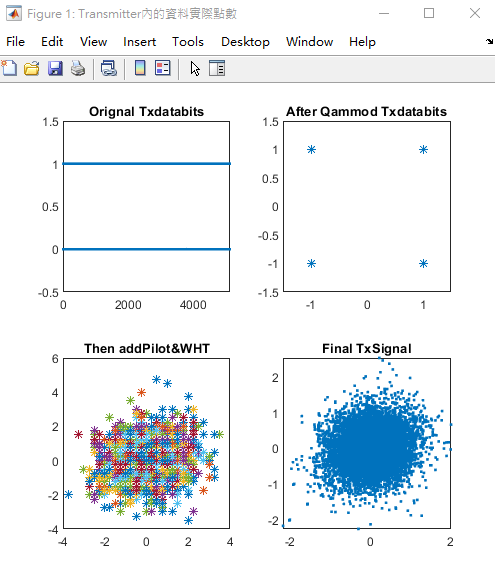


圖 16 Tx傳送端展示圖

1. Tx內的PilotBits矩陣設置

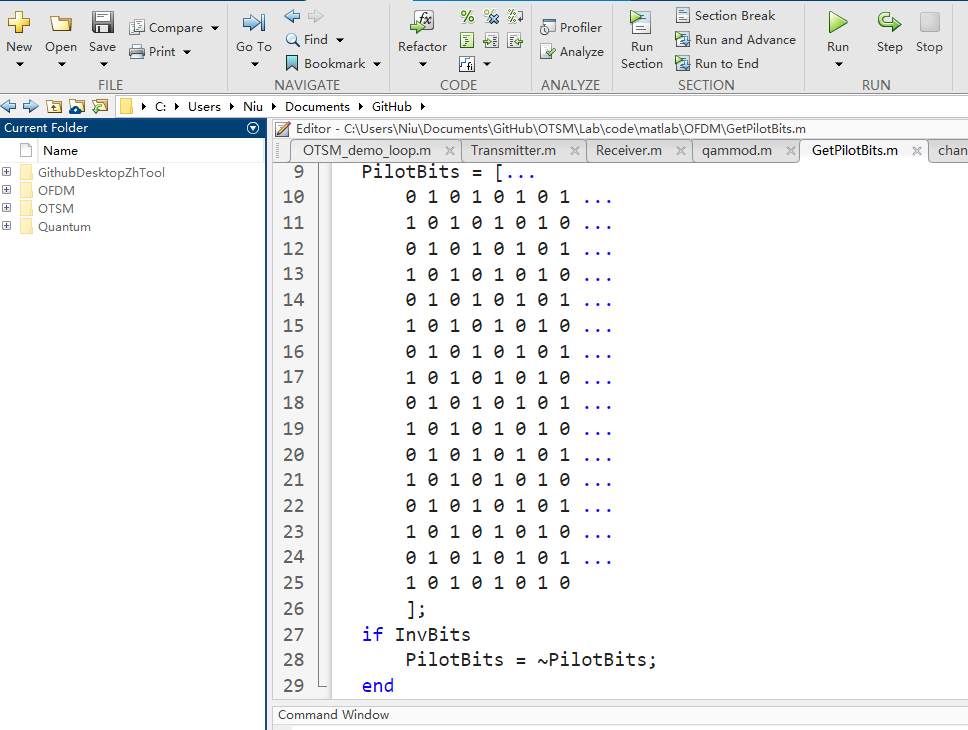


圖 17 PilotBit數據輸入

1. Transmitter內的QAMMOD 正交振幅調制

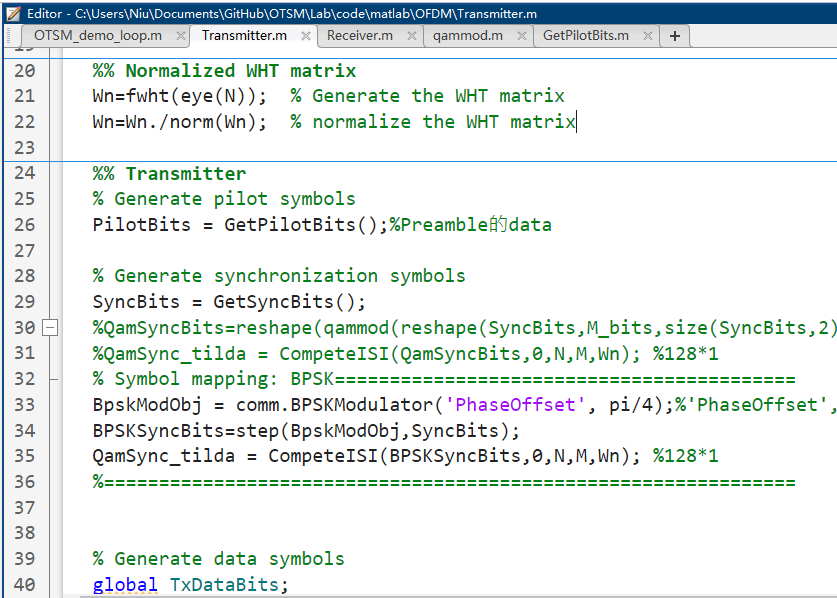


圖 18 Preamble data輸入

向量 X 的 N 點離散Walsh-Hadamard變換，其中 N 必須是 2 的冪。

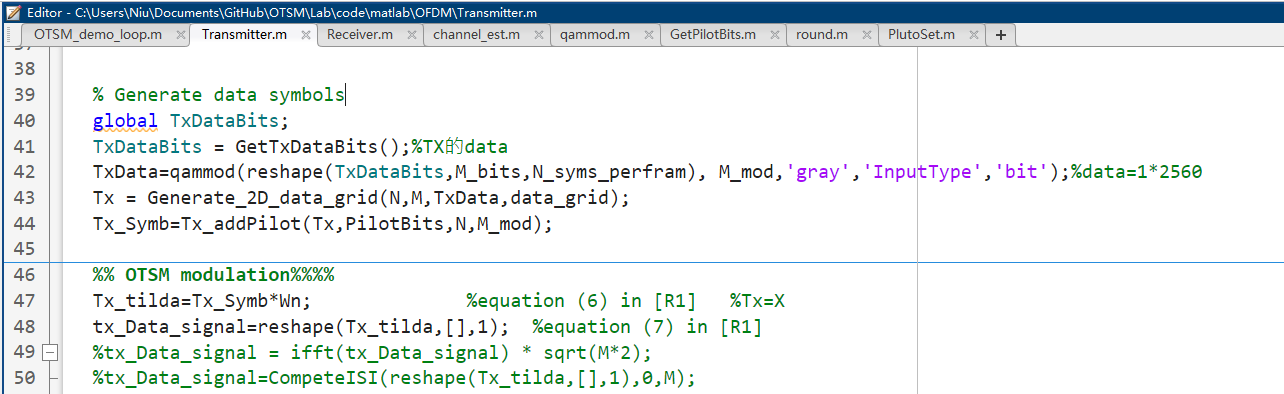


圖 19 取得TxDataBits

這裡使用reshape，透過重新排列TxDataBits數據來重塑數組。

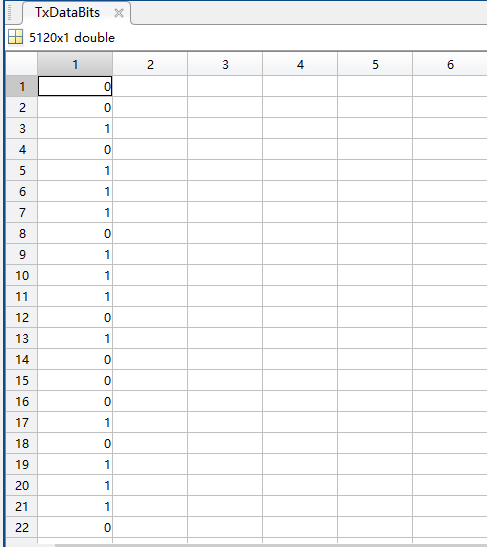


圖 20 TxDataBits數據

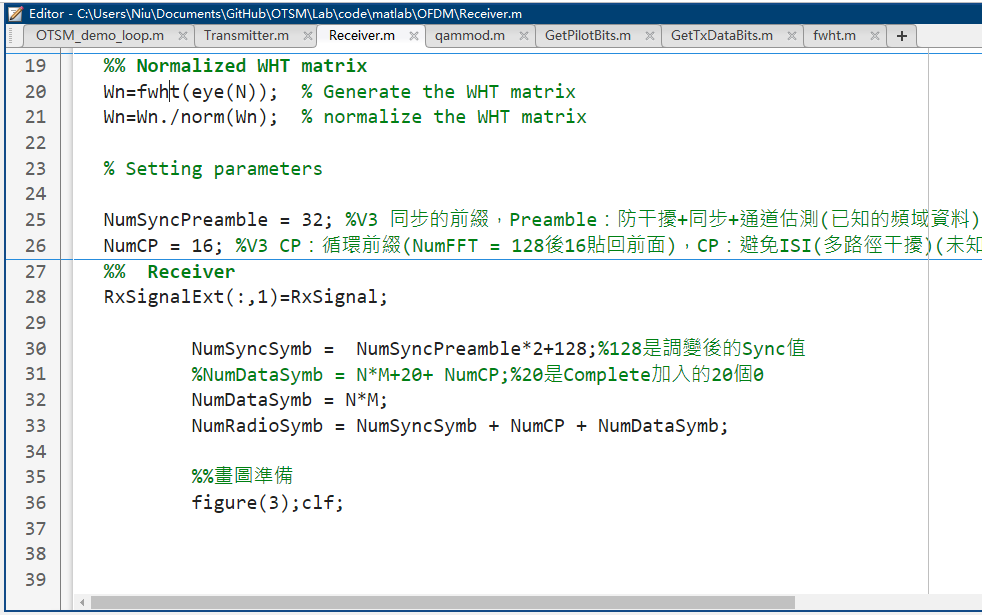


圖 21 TxDataBits數據

1. Rx接收端

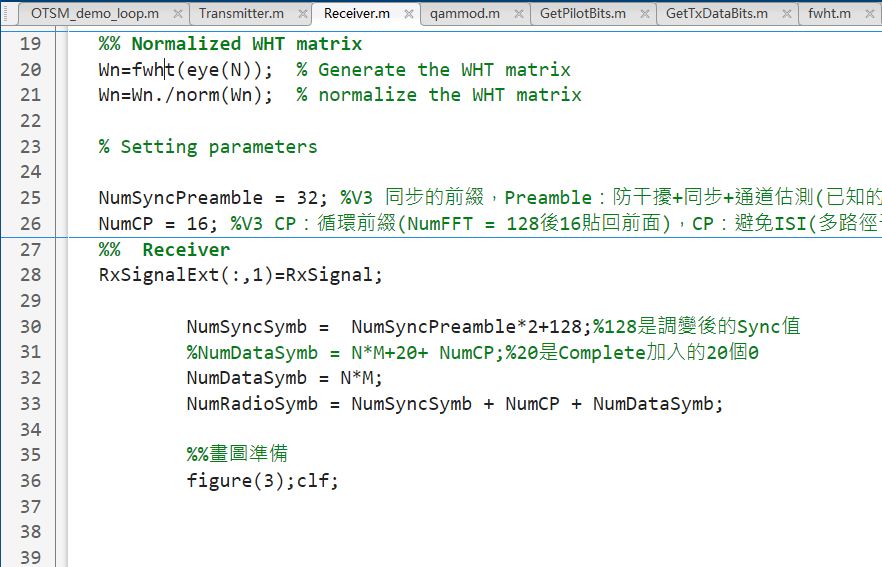


圖 22 Rx接收端的WHT矩陣



圖 22 Rx接收端展示圖

1. 結果圖

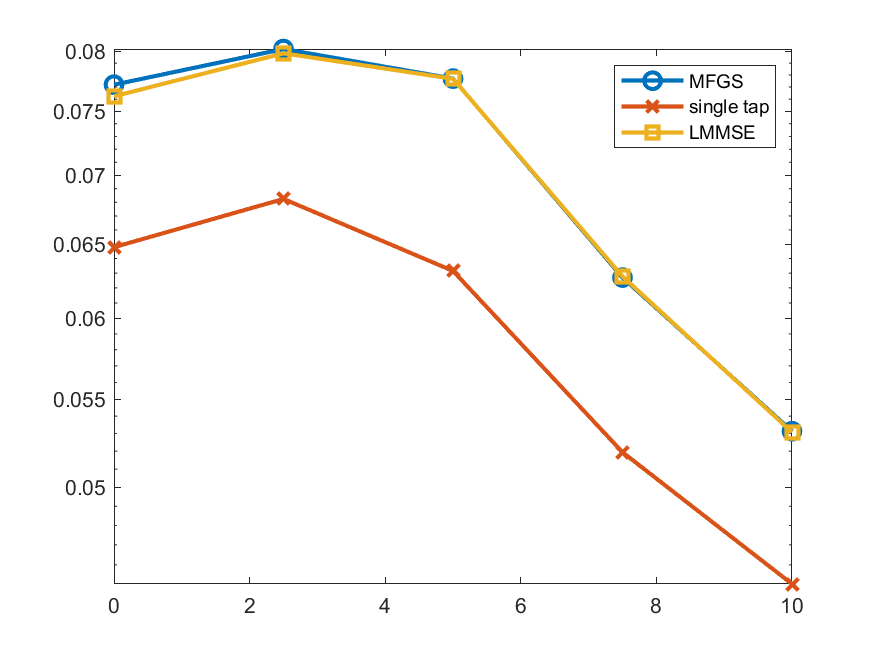


圖 22 三種偵測器的錯誤率

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

參考文獻

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>

**附錄**

**附檔1 OTSM\_demo\_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

global NoFoundDataTimes;

NoFoundDataTimes = 0;

%%% OTFS parameters%%%%%%%%%%

% N: number of symbols in time

N = 64;

% M: number of subcarriers in frequency

M = 64;

% M\_mod: size of QAM constellation

M\_mod = 4;

M\_bits = log2(M\_mod);

% average energy per data symbol

eng\_sqrt = (M\_mod==2)+(M\_mod~=2)\*sqrt((M\_mod-1)/6\*(2^2));

% SNR and variance of the noise

% SNR = P/\sigma^2; P: avg. power of albhabet transmitted

SNR\_dB = 0:2.5:10;

SNR = 10.^(SNR\_dB/10);

sigma\_2 = (abs(eng\_sqrt)^2)./SNR;

%%%

%% Initializing simulation error count variables

err\_ber\_MFGS = zeros(1,length(SNR\_dB));%bit error rate

err\_ber\_1tap = zeros(1,length(SNR\_dB));

err\_ber\_LMMSE = zeros(1,length(SNR\_dB));

avg\_ber\_MFGS=zeros(1,length(SNR\_dB));

avg\_ber\_1tap=zeros(1,length(SNR\_dB));

avg\_ber\_LMMSE=zeros(1,length(SNR\_dB));

det\_iters\_MFGS=0;

no\_of\_detetor\_iterations\_MFGS= zeros(length(SNR\_dB),1); %no\_of\_detetor\_iterations\_MFGS= zeros(1,set\_looptimes);

avg\_no\_of\_iterations\_MFGS=zeros(1,length(SNR\_dB));

%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,N,M,M\_mod);

txdata = round(txdata.\*2^15);

%% 設定Pluto

[input,s]=PlutoSet(txdata);

%% Initializing simulation error count variables

N\_fram = 10;

% global iesn0

global ifram

for iesn0 = 1:length(SNR\_dB)

sigma = sqrt(sigma\_2(iesn0));

for ifram = 1:N\_fram

current\_frame\_number=zeros(1,iesn0);

current\_frame\_number(iesn0)=ifram;

%%PLOT TX & 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

input{1} = real(txdata);

input{2} = imag(txdata);

output = stepImpl(s, input);%調用pluto的通道資料

end

I = output{1};

Q = output{2};

Rx = I+1i\*Q;

figure(2); clf;%clear figure

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

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; %'centered' 表示計算雙邊頻,'psd'表示頻譜類型

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

legend('Rx', 'Tx')

%% Receiver

[est\_info\_bits\_MFGS,det\_iters\_MFGS,est\_info\_bits\_1tap,est\_info\_bits\_LMMSE] = Receiver(Rx(1:upsample:end), sigma, N, M, M\_mod,M\_bits);

%% errors count%%%%%

global TxDataBits;

errors\_MFGS = sum(xor(est\_info\_bits\_MFGS,TxDataBits));

errors\_1tap = sum(xor(est\_info\_bits\_1tap,TxDataBits));

errors\_LMMSE = sum(xor(est\_info\_bits\_LMMSE,TxDataBits));

err\_ber\_MFGS(1,iesn0) = err\_ber\_MFGS(1,iesn0) + errors\_MFGS;

err\_ber\_1tap(1,iesn0) = err\_ber\_1tap(1,iesn0) + errors\_1tap;

err\_ber\_LMMSE(1,iesn0) = err\_ber\_LMMSE(1,iesn0) + errors\_LMMSE;

no\_of\_detetor\_iterations\_MFGS(iesn0)=no\_of\_detetor\_iterations\_MFGS(iesn0)+det\_iters\_MFGS;

%% Error count

avg\_no\_of\_iterations\_MFGS(iesn0)=no\_of\_detetor\_iterations\_MFGS(iesn0)/ifram;

avg\_ber\_MFGS(1,iesn0)=err\_ber\_MFGS(1,iesn0).'/length(TxDataBits)/ifram;

avg\_ber\_1tap(1,iesn0)=err\_ber\_1tap(1,iesn0).'/length(TxDataBits)/ifram;

avg\_ber\_LMMSE(1,iesn0)=err\_ber\_LMMSE(1,iesn0).'/length(TxDataBits)/ifram;

%% DISP error performance details

clc

disp('####################################################################')

fprintf('OTSM-(N,M,QAM size)');disp([N,M,M\_mod]);

display(current\_frame\_number,'Number of frames');

display(SNR\_dB,'SNR (dB)');

display(avg\_ber\_MFGS,'Average BER - Matched Filtered Gauss Seidel');

display(avg\_ber\_1tap,'Average BER - single tap equalizer');

display(avg\_ber\_LMMSE,'Average BER - LMMSE equalizer');

display(avg\_no\_of\_iterations\_MFGS,'Average number of iterations for the MFGS detector');

disp('####################################################################')

end

end

%% 結束

figure(4)

semilogy(SNR\_dB,avg\_ber\_MFGS,'-o','LineWidth',2,'MarkerSize',8)

hold on

semilogy(SNR\_dB,avg\_ber\_1tap,'-x','LineWidth',2,'MarkerSize',8)

hold on

semilogy(SNR\_dB,avg\_ber\_LMMSE,'-square','LineWidth',2,'MarkerSize',8)

legend('MFGS','single tap','LMMSE')

**附檔2 Transmitter.m程式碼**

function tx\_signal2 = Transmitter(upsample,N,M,M\_mod)

%NumFFT = 64;%V3 FFT轉換的點數

NumSyncPreamble = 32;%V3 同步的前綴，Preamble：防干擾+同步+通道估測(已知的頻域資料)

NumCP = 16;%V3 CP：循環前綴，CP：避免ISI(多路徑干擾)(未知的時域訊號)

%% OTFS parameters%%%%%%%%%%

M\_bits = log2(M\_mod);

%% delay-Doppler grid symbol placement

% max delay spread in the channel

delay\_spread = M/(8/3);%40\*64是資料部分 剩下是Pilot跟Sync

% data positions of OTFS delay-Doppler domain data symbols in the 2-D grid

M\_data = M-delay\_spread;

data\_grid=zeros(M,N);

data\_grid(1:M\_data,1:N)=1;

% number of symbols per frame

N\_syms\_perfram = sum(sum(data\_grid));

%% Normalized WHT matrix

Wn=fwht(eye(N)); % Generate the WHT matrix

Wn=Wn./norm(Wn); % normalize the WHT matrix

%% Transmitter

% Generate pilot symbols

PilotBits = GetPilotBits();%Preamble的data

% Generate synchronization symbols

SyncBits = GetSyncBits();

%QamSyncBits=reshape(qammod(reshape(SyncBits,M\_bits,size(SyncBits,2)/M\_bits), M\_mod,'gray','InputType','bit'),[],1);

%QamSync\_tilda = CompeteISI(QamSyncBits,0,N,M,Wn); %128\*1

% Symbol mapping: BPSK==========================================

BpskModObj = comm.BPSKModulator('PhaseOffset', pi/4);%'PhaseOffset', pi/4：0度偏移成45度、180度偏移成225度

BPSKSyncBits=step(BpskModObj,SyncBits);

QamSync\_tilda = CompeteISI(BPSKSyncBits,0,N,M,Wn); %128\*1

%===============================================================

% Generate data symbols

global TxDataBits;

TxDataBits = GetTxDataBits();%TX的data

TxData=qammod(reshape(TxDataBits,M\_bits,N\_syms\_perfram), M\_mod,'gray','InputType','bit');%data=1\*2560

Tx = Generate\_2D\_data\_grid(N,M,TxData,data\_grid);

Tx\_Symb=Tx\_addPilot(Tx,PilotBits,N,M\_mod);

%% OTSM modulation%%%%

Tx\_tilda=Tx\_Symb\*Wn; %equation (6) in [R1] %Tx=X

tx\_Data\_signal=reshape(Tx\_tilda,[],1); %equation (7) in [R1]

%tx\_Data\_signal = ifft(tx\_Data\_signal) \* sqrt(M\*2);

%tx\_Data\_signal=CompeteISI(reshape(Tx\_tilda,[],1),0,M);

TxSignal = [ ...

QamSync\_tilda(1:NumSyncPreamble);

QamSync\_tilda(1:NumSyncPreamble);

QamSync\_tilda;

zeros(128,1);

tx\_Data\_signal(size(tx\_Data\_signal,1)-NumCP+1:end, :)

tx\_Data\_signal];

flt1=rcosine(1,upsample,'fir/sqrt',0.05,64);%pulse shaper

tx\_signal2=rcosflt(TxSignal,1,upsample, 'filter', flt1); %4240(TxSignal)\*4(upsample)+(513(flt1)-1)：因為捲積所以要-1

%%陳昱升想測試畫圖%%

figure(1);

set(gcf,'name','Transmitter內的資料實際點數','Position', [20 400 500 500]);

subplot(221);

plot(TxDataBits,'.');title('Orignal Txdatabits');axis([-10,N\_syms\_perfram\*M\_bits+10,-0.5,1.5]);

subplot(222);

plot(TxData,'\*');title('After Qammod Txdatabits');axis([-1.5,1.5,-1.5,1.5]);

subplot(223);

plot(Tx\_tilda,'\*');title('Then addPilot&WHT');

subplot(224);

plot(tx\_signal2,'.');title('Final TxSignal');

**附檔3 Receiver.m程式碼**

function [est\_info\_bits\_MFGS,det\_iters\_MFGS,est\_info\_bits\_1tap,est\_info\_bits\_LMMSE] = Receiver(RxSignal, sigma, N, M, M\_mod,M\_bits)

%%廣域變數宣告

% global iesn0

%% OTFS parameters%%%%%%%%%%

%% delay-Doppler grid symbol placement

% max delay spread in the channel

delay\_spread = M/(8/3);%40\*64是資料部分 剩下是Pilot跟Sync

% data positions of OTFS delay-Doppler domain data symbols in the 2-D grid

M\_data = M-delay\_spread;%64-24=40

data\_grid=zeros(M,N);

data\_grid(1:M\_data,1:N)=1;

DelayPilotSymb=sqrt(size(GetPilotBits,2)/2);

%% Normalized WHT matrix

Wn=fwht(eye(N)); % Generate the WHT matrix

Wn=Wn./norm(Wn); % normalize the WHT matrix

% Setting parameters

NumSyncPreamble = 32; %V3 同步的前綴，Preamble：防干擾+同步+通道估測(已知的頻域資料)

NumCP = 16; %V3 CP：循環前綴(NumFFT = 128後16貼回前面)，CP：避免ISI(多路徑干擾)(未知的時域訊號)

%% Receiver

RxSignalExt(:,1)=RxSignal;

NumSyncSymb = NumSyncPreamble\*2+128;%128是調變後的Sync值

%NumDataSymb = N\*M+20+ NumCP;%20是Complete加入的20個0

NumDataSymb = N\*M;

NumRadioSymb = NumSyncSymb + NumCP + NumDataSymb;

%%畫圖準備

figure(3);clf;

%StartIdx = SyncRxSignalImproved2(RxSignalExt,M\_mod,N,M);

StartIdx = SyncRxSignalImproved1(RxSignalExt, 1 ,M\_mod,N,M,Wn);

%%JF加入重新賦值=1避免StartIdx == -1時直接中斷程式：

global NoFoundDataTimes;

if(StartIdx == -1)

StartIdx = 1;

NoFoundDataTimes=NoFoundDataTimes+1;

elseif(StartIdx+NumRadioSymb-1 >= length(RxSignalExt))

StartIdx = 1;

NoFoundDataTimes=NoFoundDataTimes+1;

end

RxSignalRadioFrame = RxSignalExt(StartIdx + NumSyncSymb+NumCP+128 :StartIdx+NumRadioSymb-1+128);

%RxSignalRadioFrame =fft(RxSignalRadioFrame ) / sqrt(1);

%RxSignalRadioFrame = ICompeteISI(RxSignalRadioFrame,N\*M,M,0);

%% OTSM Reobtain Pilot%%%%

%Pilot side

RxSignalGrid=reshape(RxSignalRadioFrame,N,M);

% %%把兩個Pilot分別放入2行

% Y\_OTSM\_Pilot=zeros(N,2);

% for i=1:sqrt(size(Y\_OTSM\_Pilot,1))

% %Y\_OTSM\_Pilot=reshape(RxSignalGrid(M\_data+DelayPilotSymb+1:M\_data+8\*2,1:DelayPilotSymb),[],1);%傳送資料時Pilot和Sync皆以8\*8的形式傳送

% Y\_OTSM\_Pilot((i-1)\*8+1:(i-1)\*8+8,1)=RxSignalGrid(M\_data+1:M\_data+DelayPilotSymb,i);

% Y\_OTSM\_Pilot((i-1)\*8+1:(i-1)\*8+8,2)=RxSignalGrid(M\_data+DelayPilotSymb\*2+1:M\_data+DelayPilotSymb\*3,i);

% end

%%把兩個Pilot切割成4\*n後交叉放入2行

Y\_OTSM\_Pilot=zeros(N,2);

for i=1:sqrt(size(Y\_OTSM\_Pilot,1))

%Y\_OTSM\_Pilot=reshape(RxSignalGrid(M\_data+DelayPilotSymb+1:M\_data+8\*2,1:DelayPilotSymb),[],1);%傳送資料時Pilot和Sync皆以8\*8的形式傳送

Y\_OTSM\_Pilot((i-1)\*4+1:(i-1)\*4+4,1)=RxSignalGrid(M\_data+1:M\_data+4,i);

Y\_OTSM\_Pilot((i-1)\*4+1:(i-1)\*4+4,2)=RxSignalGrid(M\_data+5:M\_data+8,i);

end

for i=1:sqrt(size(Y\_OTSM\_Pilot,1))

%Y\_OTSM\_Pilot=reshape(RxSignalGrid(M\_data+DelayPilotSymb+1:M\_data+8\*2,1:DelayPilotSymb),[],1);%傳送資料時Pilot和Sync皆以8\*8的形式傳送

Y\_OTSM\_Pilot((i+7)\*4+1:(i+7)\*4+4,1)=RxSignalGrid(M\_data+DelayPilotSymb+1:M\_data+DelayPilotSymb+4,i);

Y\_OTSM\_Pilot((i+7)\*4+1:(i+7)\*4+4,2)=RxSignalGrid(M\_data+DelayPilotSymb+5:M\_data+DelayPilotSymb+8,i);

end

%把pilot的8\*8資料 拆成4\*8、4\*8

% Estimate carrier frequency offset

[RxDataSymbEq,RxSignalRadioFrameCmpCFO,G] = channel\_est(N,M,M\_mod,RxSignalRadioFrame,Y\_OTSM\_Pilot,0);

r = reshape(RxSignalRadioFrameCmpCFO,[],1);

%% OTSM demodulation%%%%

%主要解調變

Y\_tilda=reshape(r,M,N); %equation (11) in [R1]

%Y\_tilda(M\_data+1:end,:)=0; %去除Pilot

Y = Y\_tilda\*Wn; %equation (12) in [R1]

%% Generate the block-wise channel matrices in the delay-time and the time-frequency domain

[Gn\_block\_matrix,Tn\_block\_matrix,zn\_block\_vector,H\_t\_f]=Generate\_Matched\_Filter\_GS\_matrices(N,M,G,r);

%%偵錯

if (isnan(sum(zn\_block\_vector(:))))

est\_info\_bits\_MFGS=zeros(M\_data\*M\*M\_bits,1);

est\_info\_bits\_1tap=zeros(M\_data\*M\*M\_bits,1);

est\_info\_bits\_LMMSE=zeros(M\_data\*M\*M\_bits,1);

det\_iters\_MFGS=0;

return;

end

%% GS SOR Iterative detection

n\_ite\_MRC=50; % maximum number of detector iterations

omega=1; %damping parameter - reducing omega improves error performance at the cost of increased detector iterations

if(M\_mod==64)

omega=0.25;

end

figure(5);

set(gcf,'name','三種偵測器EQ圖','Position', [650 50 600 400]);

decision=1; %1-hard decision, 0-soft decision

[est\_info\_bits\_MFGS,det\_iters\_MFGS,~] = Matched\_Filter\_GS\_detector(N,M,M\_mod,sigma,data\_grid,Y,H\_t\_f,n\_ite\_MRC,omega,Tn\_block\_matrix,Gn\_block\_matrix,zn\_block\_vector,r,Wn,decision);

[est\_info\_bits\_1tap,~] = TF\_single\_tap\_equalizer(N,M,M\_mod,sigma,data\_grid,Y,H\_t\_f,Wn);

[est\_info\_bits\_LMMSE,~] = Block\_LMMSE\_detector(N,M,M\_mod,sigma,data\_grid,Gn\_block\_matrix,r,Wn);