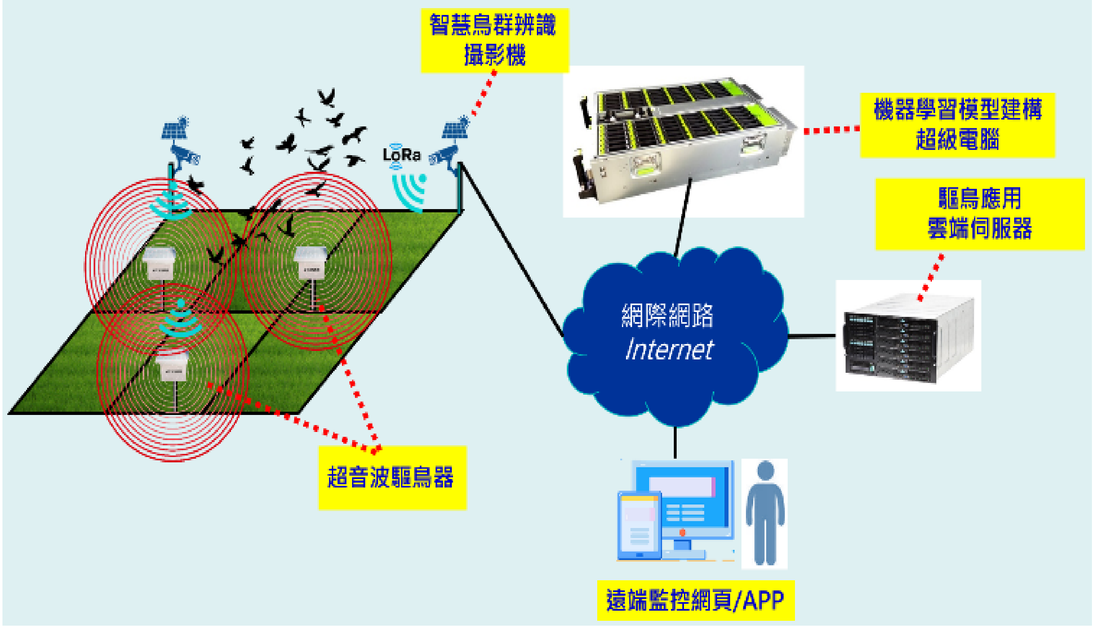
資訊工程學系 / 廖冠雄 教授 / 蔡子皓 / 陳柏陽

農業AIOT禽鳥驅離系統

研究動機與目的

目前農田中最常採用施放鞭炮的方式來驅離禽鳥，雖 然能有效驅趕禽鳥，但是需要花費更多人力，同時也會產生噪音及空氣汙染這方面的問題，於是我們提出人工智慧結合物聯網的系統，透過發射超音波來驅離禽鳥，以降低農損率，且不會影響周遭居民的生活品質。

系統架構



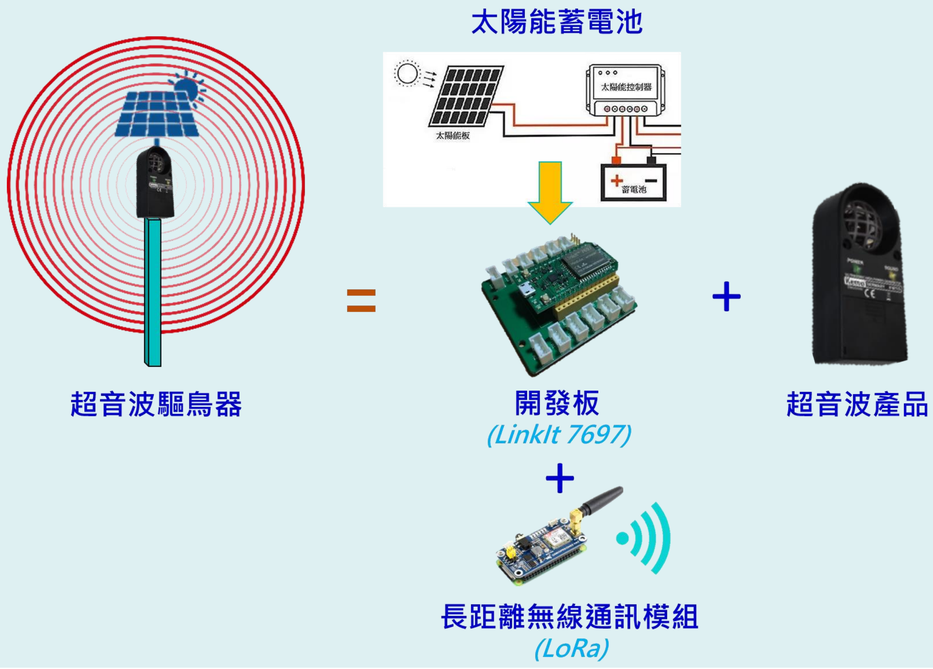
機構介紹

智慧鳥群辨識攝影機：安裝於農地適當位置，監看特定範圍，自動辨識出現的有害鳥群，直接遙控範圍內的超音波驅鳥器，發出超音波以驅離鳥群。本裝置的主控板採用NVIDIA Jetson Nano，其上配置以下兩項元件：

(一) 戶外數位攝影機：連續擷取農地畫面，並將影像傳至NVIDIA Jetson Nano中。
(二) LoRa長距離無線通訊模組：單板電腦將透過此通訊模組遙控農地現場的超音波驅鳥器、傳送各項統計數據、以及接受來自雲端伺服器的訊息。

超音波驅鳥器：可被遠端遙控持續發出引起禽鳥不適感的特定頻率超音波，以驅離現場對作物有害鳥群。本裝置的組成如圖5所示，LinkIt 7697當作超音波驅鳥器的開發板，其上配置以下兩項元件：

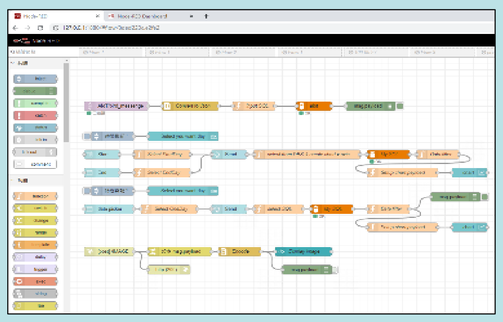
(一) 超音波產品：用來發射出特定頻段的超音波。
(二) LoRa長距離無線通訊模組：開發板將透過此通訊模組接收智慧鳥群辨識攝影機送來的訊息，以啟動農地現場的超音波驅鳥器。



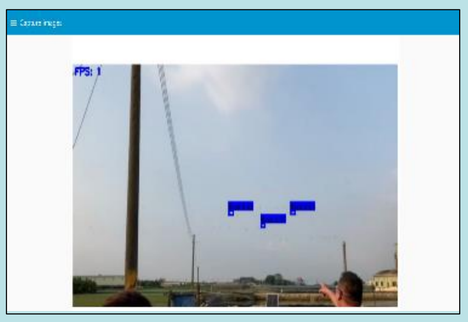
成果展示

智慧鳥群辨識攝影機

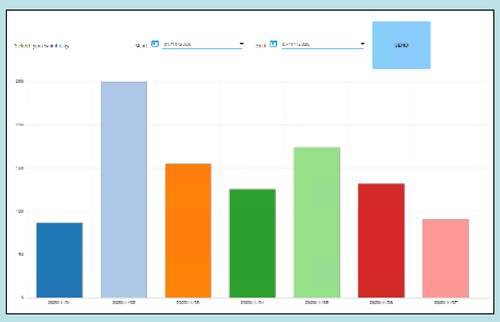
超音波驅鳥器

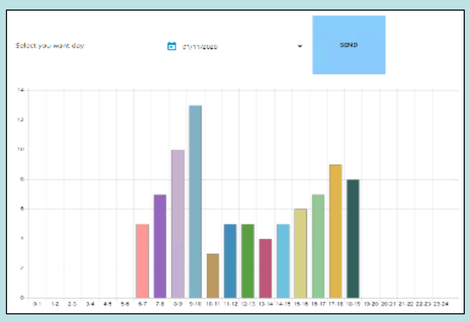


Node-RED 開發工具



捕捉禽鳥物件之畫面





鳥隻數量統計圖

未來展望

未來可對鳥類辨識模型進行優化，像是在資料集中添加實際農田中出現鳥類的圖片，或是添加像是飛機、風箏等...類別的圖片作為訓練，這樣或許就能解決辨識時的誤判，進而提升對鳥類物件上偵測的效果，除了資料量上的擴增以及類別數目的增加外，也可以導入不同類型的神經網路框架以重新訓練模型，再去評估甚麼樣子的模型框架更合適用此系統上，使得小物件上的偵測會更加準確，由於硬體上的進步使得人工智慧技術在開發上能更快速的進行實驗與分析，而未來人工智慧技術結合跨領域的發展會是一大趨勢。

資訊工程學系／陳延華教授／洪勤翔、蔡莉雯、林映臻、王彬宇、邱郁涵

AI糾錯碼實踐於嵌入式裝置

摘要

本專題採用二元平方剩餘碼(Quadratic Residue code, QR code)和極化碼(Polar code)作為資料保護工具，進而提高資料傳遞時的可靠性和安全性，並藉此提高整體的糾錯能力，再利用機器學習(Machine Learning, ML) 進行解碼訓練以獲得新的解碼方式，最終運行於嵌入式系統上。

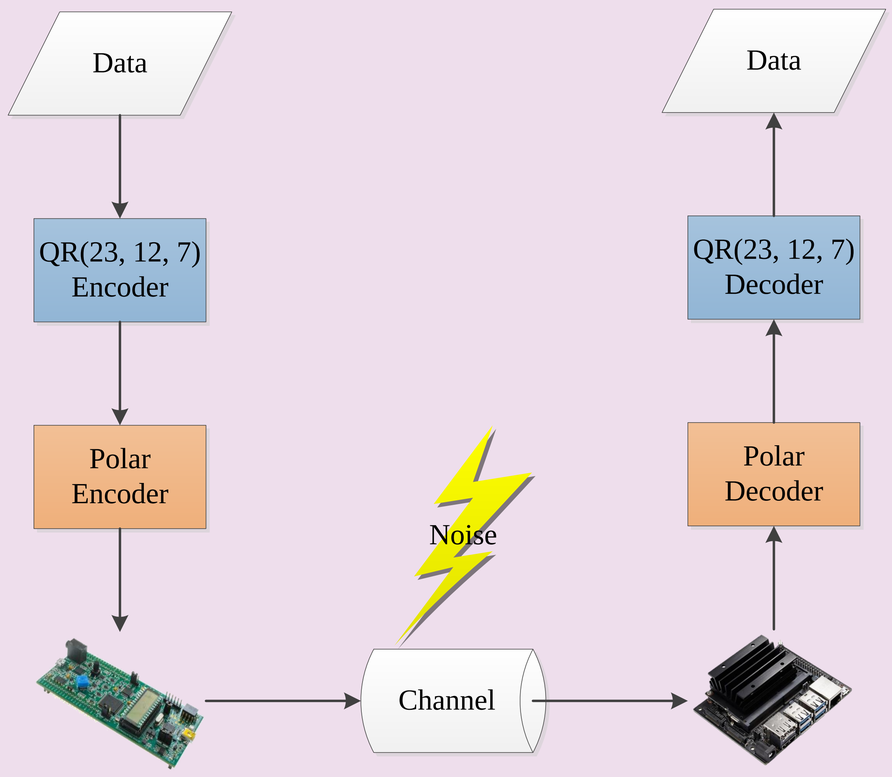


圖1 資料傳遞過程

研究方法

QR(23, 12, 7) code編碼需定義Q_n集合如下：

$$Q_n = \{j \mid j \equiv x^2 \,\%\, n, for\ 1 \le x \le n-1\}$$

生成多項式$g_{(x)}$定義如下：

$$g_{(x)} = \prod_{i \epsilon Q_n} (x - \beta^i)$$

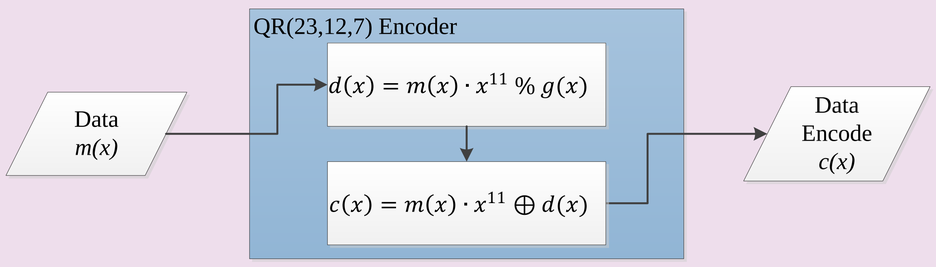


圖2 QR(23, 12, 7) code編碼過程

本專題最終採用近鄰演算法(K-Nearest Neighbors, KNN)進行訓練，利用症狀子值與錯誤位置的碼字與錯誤樣式之間具有一對一的性質建立資料。

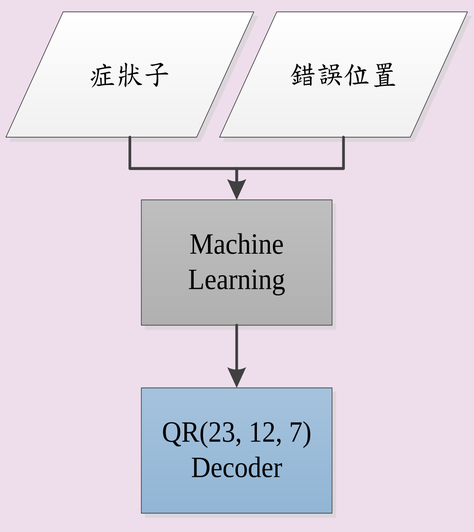


圖3 訓練過程

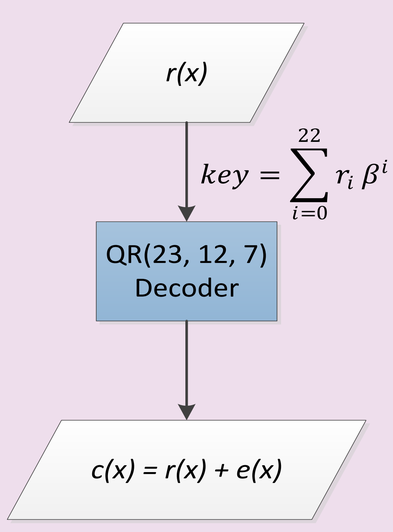


圖4 解碼過程

研究方法

Polar(8, 4, 4) code編碼定義如下：

利用Kronecker product ⊗生成編碼矩陣G_3

$$\begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix}$$

圖5 利用G_3對資料進行編碼

本專題的Polar code採用凍結位元(frozen bits)來進行解碼，接收到資料向量後取出凍結位元(u_0, u_1, u_2, u_4)判斷是否都為0，如果是代表原始資料沒有錯；反之，利用查表法找出錯誤的位元進行糾錯。

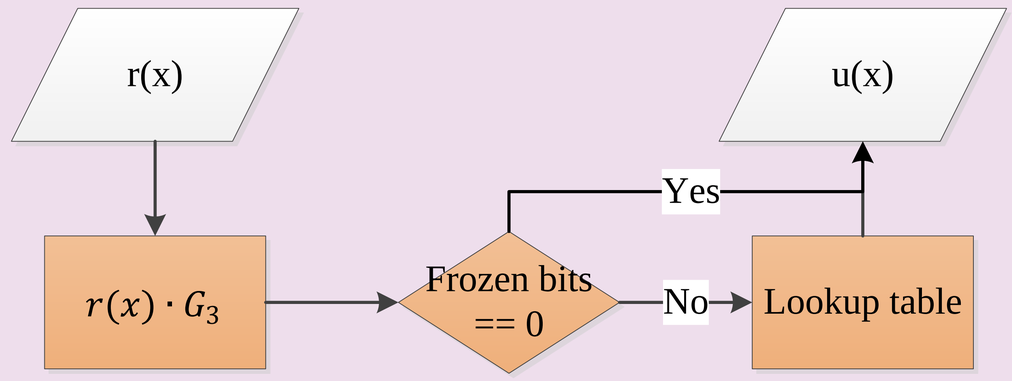


圖6 Polar code解碼過程

結論

選用二元平方剩餘碼與極化碼進行連結碼的研究，其中改良了以往常見的查表法以及代數法，使用AI訓練的方式，將糾錯碼進行解錯糾正。執行各種錯誤測試顯示結果，透過機器學習KNN的模式下也可以正確分出各種錯誤的資料特徵，且在二元平方剩餘碼的錯誤範圍內，全部可以正確解出訊息，最後使用連結碼強化單獨使用二元平方剩餘碼的整體解錯能力。

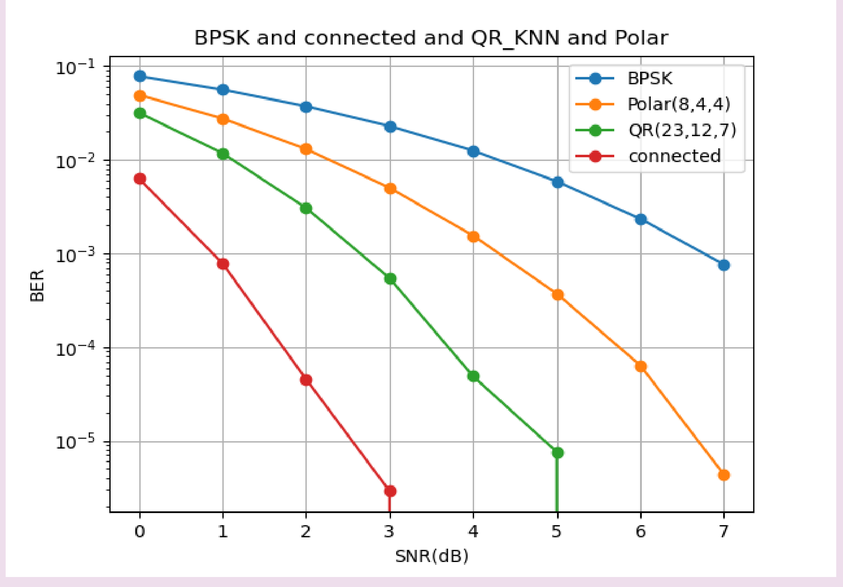


圖1 資料傳遞過程

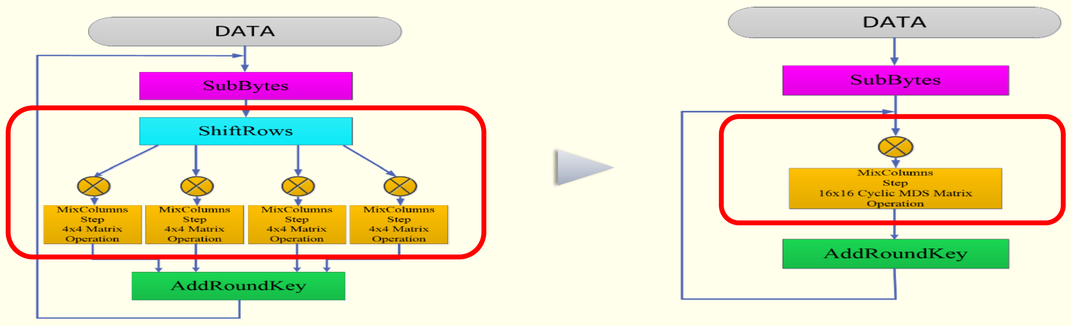
資訊工程學系 / 指導老師:陳延華教授 / 學生:曾雅靖、陳宏維、楊佳偉、石程鑫、戴佩仙

利用MDS循環矩陣加強 AES MIXCOLUMNS 與 INVMIXCOLUMNS 轉換程序之安全性

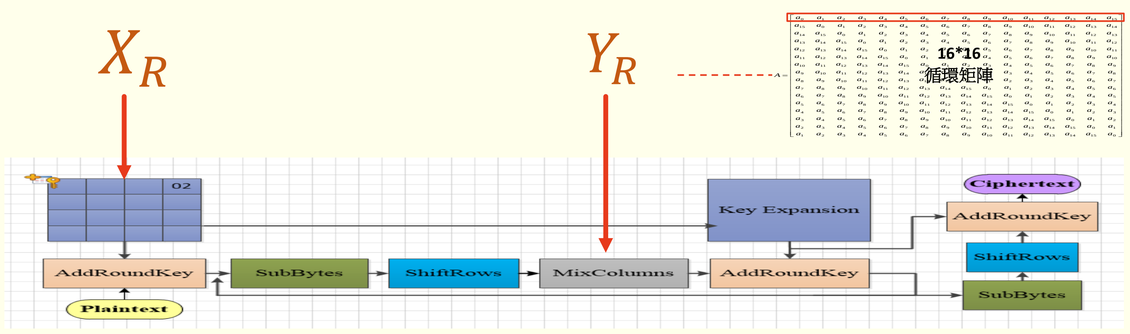
摘要

原AES MixColumns矩陣運算的4x4矩陣，利用具有高分岔性與邏輯運算特性的16x16 MDS矩陣替代(圖一)。16x16 矩陣運算需要很多乘法運算步驟，本專題利用邏輯運算特性減少有限體乘法數目進而降低運算時間。

使用橢圓曲線迪菲-赫爾曼金鑰交換得到之私鑰X和Y值，分別作為AES Key值與MixColumns矩陣係數(圖二)，達到整體過程安全性得提高。



圖一



圖二

研究方法

改良AES--循環矩陣化簡

$$\mathbf{A}=\left[\begin{array}{cccc|cccc|c} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & d^0 \\ a_{15} & a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & d^1 \\ a_{14} & a_{15} & a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & d^2 \\ a_{13} & a_{14} & a_{15} & a_0 & a_1 & a_2 & a_3 & a_4 & d^3 \\ \hline a_{12} & a_{13} & a_{14} & a_{15} & a_0 & a_1 & a_2 & a_3 & d^4 \\ a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_0 & a_1 & a_2 & d^5 \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_0 & a_1 & d^6 \\ a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_0 & d^7 \\ \hline a_8 & a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & d^8 \\ a_7 & a_8 & a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & d^9 \\ a_6 & a_7 & a_8 & a_9 & a_{10} & a_{11} & a_{12} & a_{13} & d^{10} \\ a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} & a_{11} & a_{12} & d^{11} \\ a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} & a_{11} & d^{12} \\ a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} & d^{13} \\ a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & d^{14} \\ a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & d^{15} \end{array}\right]$$

A_0 A_1 D_0 A_1 A_0 D_1

$$\begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} = \begin{bmatrix} A_0 & A_1 \\ A_1 & A_0 \end{bmatrix} \begin{bmatrix} D_0 \\ D_1 \end{bmatrix} \longrightarrow \begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} = \begin{bmatrix} A_0(D_0+D_1)+(A_0+A_1)D_1 \\ A_0(D_0+D_1)+(A_0+A_1)D_0 \end{bmatrix} = \begin{bmatrix} E+F \\ E+G \end{bmatrix}$$

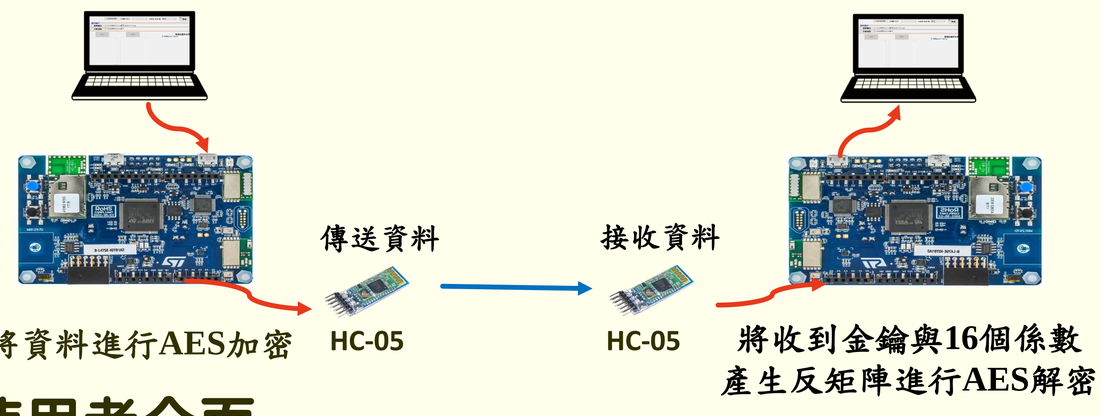
改良AES--非循環矩陣化簡

$$\mathbf{E}=\left[\begin{array}{cccc|cccc|c} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & d_0+d_8 \\ a_{15} & a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & d_1+d_9 \\ a_{14} & a_{15} & a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & d_2+d_{10} \\ a_{13} & a_{14} & a_{15} & a_0 & a_1 & a_2 & a_3 & a_4 & d_3+d_{11} \\ \hline a_{12} & a_{13} & a_{14} & a_{15} & a_0 & a_1 & a_2 & a_3 & d_4+d_{12} \\ a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_0 & a_1 & a_2 & d_5+d_{13} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_0 & a_1 & d_6+d_{14} \\ a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_0 & d_7+d_{15} \end{array}\right]$$

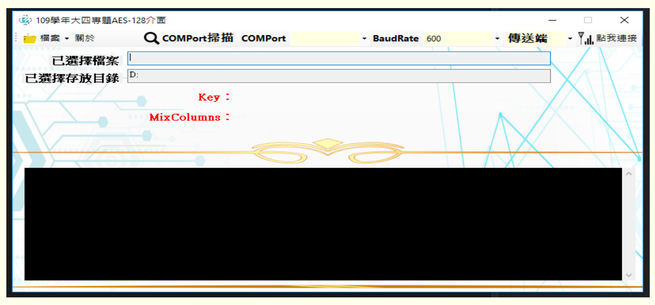
B_0 C_1 S_0 C_2 B_0 S_1

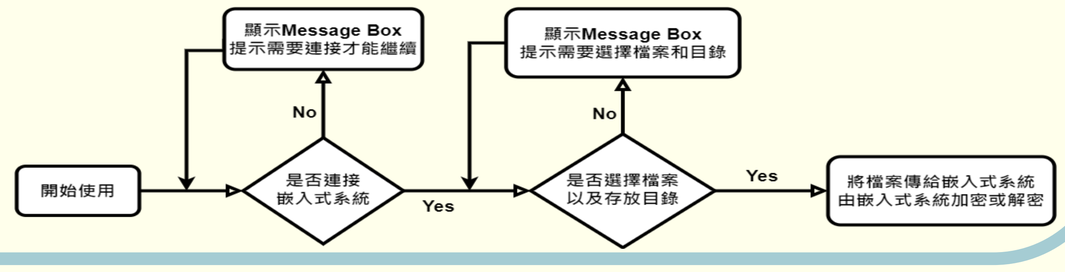
$$\begin{bmatrix} E_0 \\ E_1 \end{bmatrix} = \begin{bmatrix} B_0 & C_1 \\ C_2 & B_0 \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \end{bmatrix} \longrightarrow \begin{bmatrix} E_0 \\ E_1 \end{bmatrix} = \begin{bmatrix} B_0(S_0+S_1)+(B_0+C_1)S_1 \\ B_0(S_0+S_1)+(B_0+C_2)S_0 \end{bmatrix} = \begin{bmatrix} T+U \\ T+V \end{bmatrix}$$

嵌入式系統應用--STM B-L475E



使用者介面





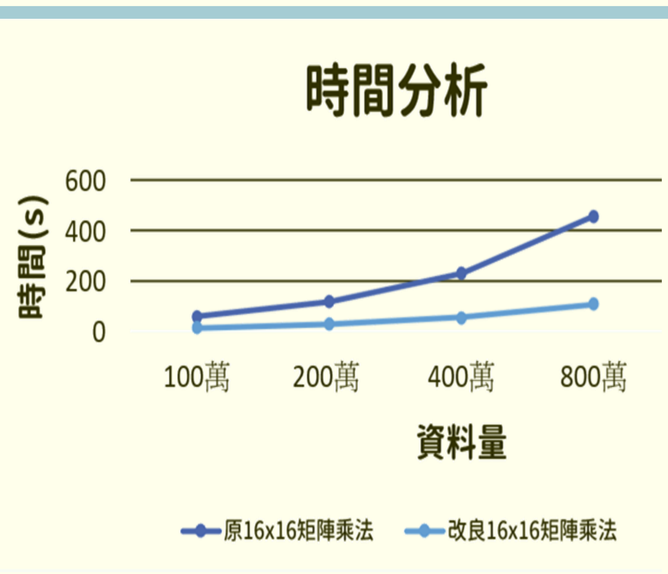
結論

原16x16矩陣乘法
乘法數 256
加法數 240

→

改良16x16矩陣乘法
乘法數 69
加法311

利用邏輯運算特性



效能比原16x16矩陣乘法減少76.50%矩陣乘法時間