One Day KeeLoq Workshop

課 程 介紹

- KEELOQ® 原理與介紹
- KEELOQ® 單向傳送編碼技術
- 產生編碼密碼(Encryption Key)
- KEELOQ® 資料接收、解碼與判斷
- KEELOQ® 紙上解碼介紹
- 練習解 Hopping 和判斷資料
- 解碼密碼是如何產生的？
- 簡易學習模式 - Simple Learn
- 標準學習模式 - Normal Learn
- 安全學習模式 - Secure Learn

無線遙控器傳輸

破解遙控器的方法

KeeLoq 的應用

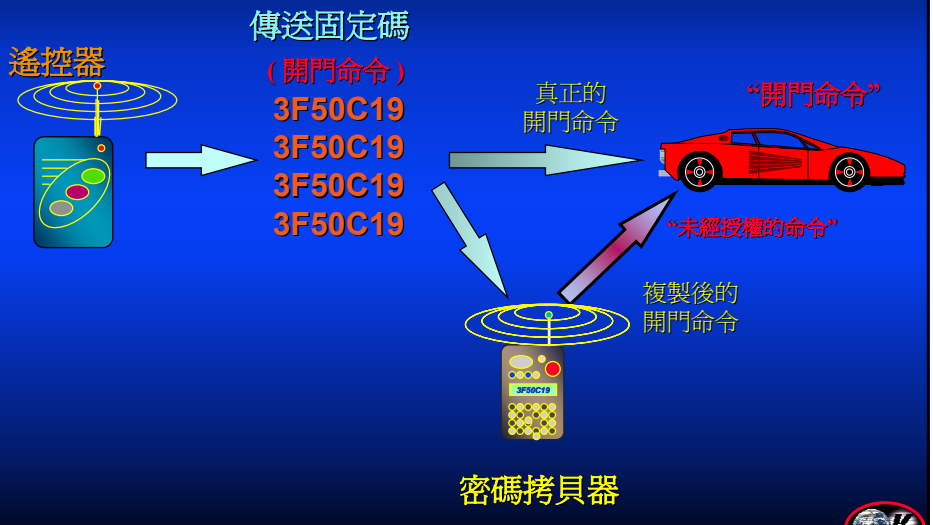
破解遙控器的方式

只要用簡單的固有技術，就能破解現今眾多用在汽車、車庫或門禁的無線遙控系統 .....

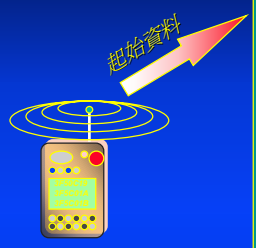
這些固有技術如下:

1. 密碼捕捉、拷貝機
2. 密碼掃瞄器
3. 電腦預測密碼的組合

密碼捕捉、拷貝機



密碼掃瞄器



“密碼掃瞄器”

1. 設定掃瞄起始值
2. 傳送掃瞄資料
3. 掃瞄資料加一
4. 重複步驟 2

0000000
-
-
3F50C11
3F50C12
3F50C13
3F50C14
3F50C15
3F50C16
3F50C17
3F50C18
3F50C19
3F50C1A
3F50C1B
3F50C1C
-
-
FFFFFFF

“所有可能的資料組合均可被傳送出去，直到密碼被接受爲止”

“防止密碼被盜取 ?”

- 利用跳碼方式來防止拷貝機:
  - 對傳送資料而言 - 似乎是一組隨機產生的資料，不可預測其資料的變化
- 避免掃碼機的方法:
  - 增加資料傳送的長度 - 適當的資料長度會讓掃碼機花費很多年的時間來猜測可能的密碼
  - HCS300 一秒約可傳 送10 筆資料，2^{66} 組可能出現的資料總共傳送時間度大約需要

2^{66} x 0.1 秒 = 2.2×10^{11} 年

十分之一的猜中機率需 ? 年

KEELOQ® 的定義

實際上是一個 “ASIC” 的

特別設計

內含 加密 及 解密 技術

適用於遙控或命令辨別的

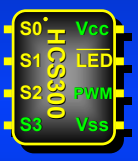
應用場合

KEELOQ® 資料傳送架構

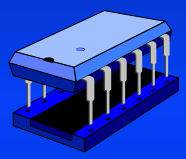
Keeloq®
Encoder/
Transcoder

Keeloq®
Receiver/
Transceiver

1.

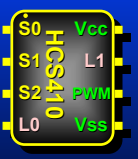


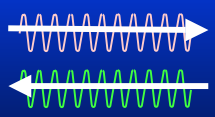
傳送媒介

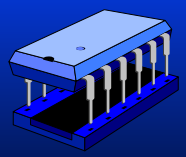


單向
傳送編碼技術
(Code Hopping)

2.







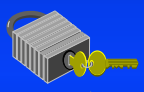
雙向
傳送編碼技術
(IFFi/w)

KEELOQ® 的應用領域

軟體保護

車庫門遙控

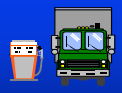
密秘通訊



安全鎖

KEELOQ
Rock Solid Security

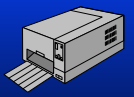
被動式感應控制



真僞鑑別

居家安全

停車場管制



防做冒零組件

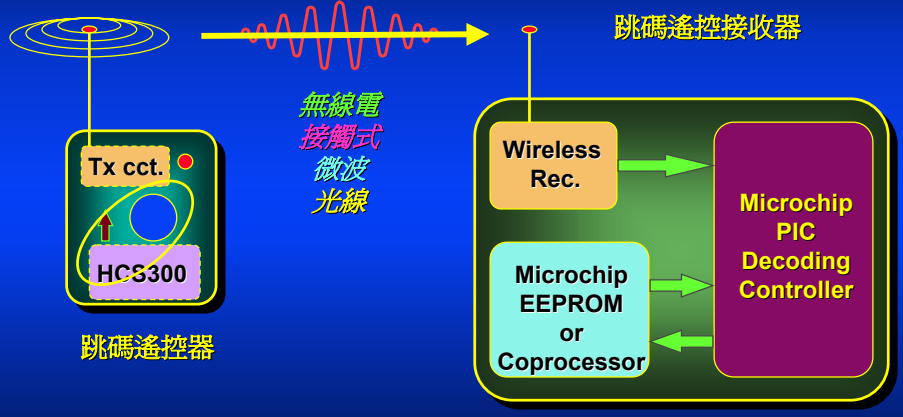
KeeLoq®

單向傳送編碼技術
HCS300 介紹

KeeLoq 的傳送可經由公共頻道

- KeeLoq 的編碼含有一組 64 bit "編碼密碼"
- 編碼器共傳送 66 bits 的資料到解碼器，其中有 32 bits 的資料是一完全不可預測的跳碼資料
- KeeLoq 的解碼器知道這資料所用"編碼密碼"，所以可以用它來檢驗其接收的跳碼資料是否正確.
- KeeLoq 的傳送資料是唯一，且不重覆....
- KeeLoq 較長的編碼技術可避免掃碼機在短時間內就將相同的碼傳送出去：
  - 即使以最快的方式來產生 HCS300 的編碼，必須超過 3.7 年才可能產生全部的跳碼部份 (32 bit)，若要產生全部 66 bit 則需 2.2×10^{11} 年。

KeeLoq® 單向傳送技術



演算法！

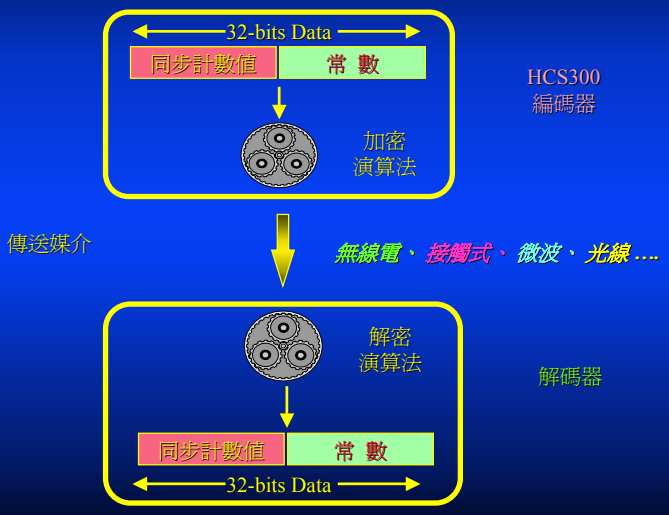
KeeLoq 的演算法則是一數學推算公式

當資料進入這演算法時，

其輸出對輸入而言是唯一的（不重覆）結果

這種一對一的關係來對應到輸入與輸出的資料

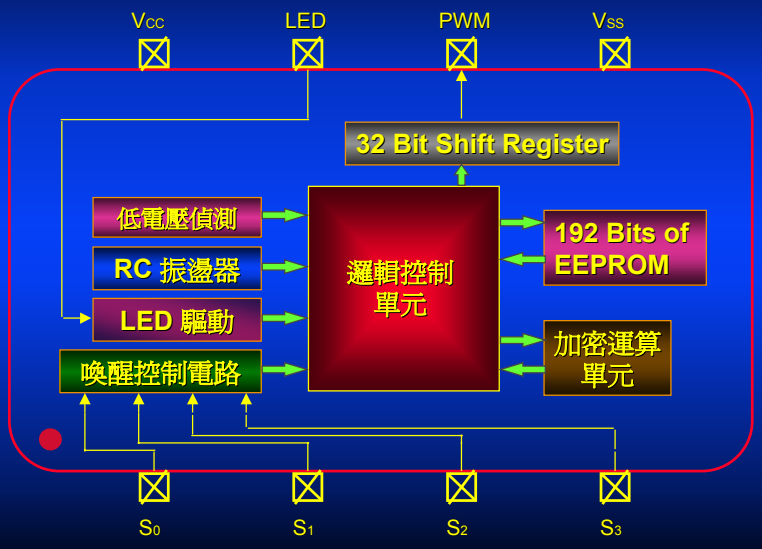
跳碼技術是簡易的!



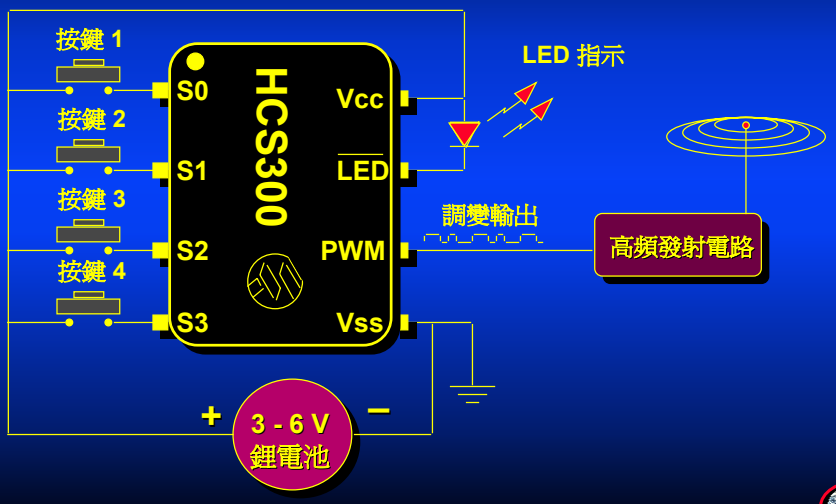
HCS300 編碼器

- HCS300內部有一個 EEPROM 用來儲存:
  - 64 bit 編碼密碼，這個密碼是不會被傳送出去的
  - 28 bit 序號，每個編碼器的序號應不相同。
  - 16 bit 同步計數值 ，每當編碼器傳送一筆 資料後，其值會自動增加一並存入EEPROM 中。
  - 10 bit 固定 識別碼。
- 一個用來編碼加密的核心電路

HCS300 方塊圖



HCS300 基本的線路



HCS300 內部資料儲存位址一覽表

儲存位址	助憶符號	說明
0	KEY_0	64-bit 編碼密碼 (word 0)
1	KEY_1	64-bit 編碼密碼 (word 1)
2	KEY_2	64-bit 編碼密碼 (word 2)
3	KEY_3	64-bit 編碼密碼 (word 3)
4	SYNC	16-bit 同步計數值
5	保留	設定爲 0000H
6	SER_0	32-bit 序號 0 (word 0)
7	SER_1(Note)	32-bit 序號 1 (word 1)
8	SEED_0	32-bit 種子編號 0 (word 0)
9	SEED_1	32-bit 種子編號 1 (word 1)
10	EN_KEY	16-bit Envelope Key (未使用)
11	CONFIG	工作設定字元組

Note: 1. 序號總長共 32 bits，但只有低 28 bits 有效且會傳送出去。
2. 序號的最高位元 (Bit 31)，爲自動關機控制位元。若不小心按鍵持續壓著約 25 秒後，HCS300 會自動關機以節省耗電。

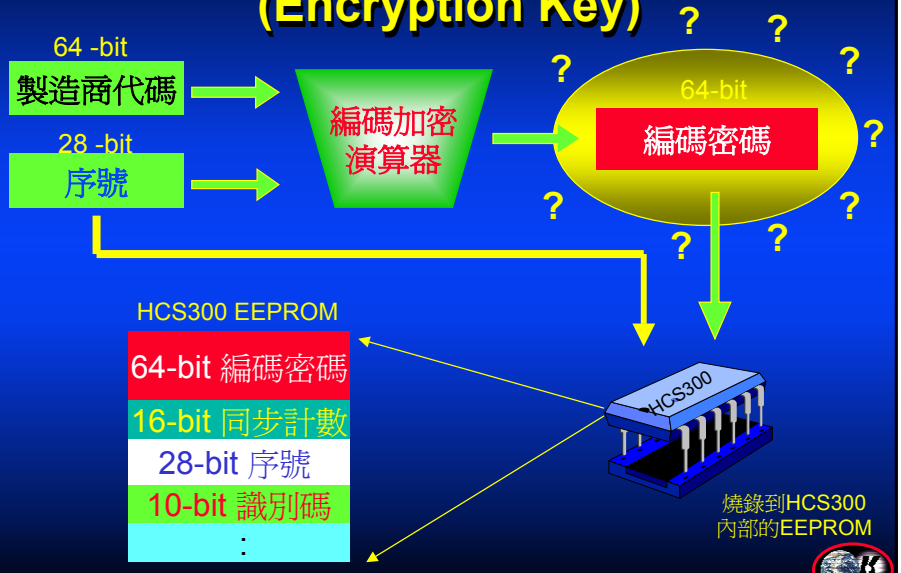
HCS300 工作設定字元組

位元位址	說　明
bit 0	識別碼 Bit 0
bit 1	識別碼 Bit 1
bit 2	識別碼 Bit 2
bit 3	識別碼 Bit 3
bit 4	識別碼 Bit 4
bit 5	識別碼 Bit 5
bit 6	識別碼 Bit 6
bit 7	識別碼 Bit 7
bit 8	識別碼 Bit 8
bit 9	識別碼 Bit 9
bit 10	Overflow Bit 0 (OVR0)
bit 11	Overflow Bit 1 (OVR1)
bit 12	Low Voltage Trip Point Select
bit 13	傳送速率選擇 Bit 0 (BSL0)
bit 14	傳送速率選擇 Bit 1 (BSL1)
bit 15	Envelope Encryption Select (EENC)

KEELOQ®

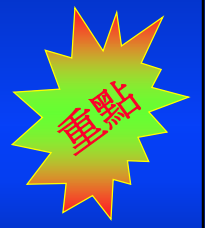
第一步驟
編碼密碼（Encryption Key）
是如何產生

如何產生編碼密碼
(Encryption Key)



KEELOQ 核心組成元件

- **製造商代碼 (Manufacture's Code)**
  - 製造商 / 產品的辨別
  - 由製造商自行決定此代碼 (不可洩漏的**原始密碼**)
  - 遙控器的**製造商代碼**必須與接收解碼器相同
  - 不同的製造商擁有不同的**製造商代碼**
- **序號(Serial Number)**
  - 每一編碼 IC 或 遙控器 其 **序號** 均不相同
  - 用來識別遙控器與接收器之間的關係
  - 即使使用者同時有兩支遙控器來控制同一接收 器，其 **序號**也不相同 (但 **製造商代碼** 必須相同)
- **編碼密碼 (Encryption Key)**
  - 利用 **製造商代碼** 及 **序號** 產生 64 bits 的 **編碼密碼**
  - 這 64 bits 的 **編碼密碼** 會被燒錄在內部的 EEPROM
  - 這 64 bits 的 **編碼密碼** 是用來產生跳碼的密碼



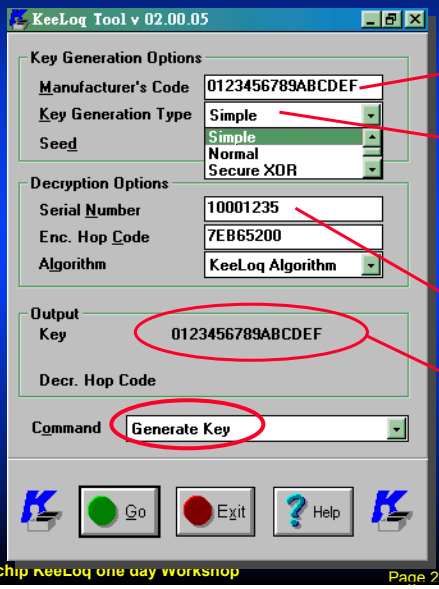
產生簡易模式的編碼密碼

- 利用 Microchip 所提供的編解碼工具
  - KeeLoq Tool v2.00.05
  - C:\…\One Day KeeLoq\Keeloq Tools\Encode tool

Icon for KeeLoq Tool

- 利用此軟體工具練習產生各種模式的編碼密碼
  - Simple Encode
  - Normal Encode

產生密碼編碼工具 (KeeLoq Tool)



64-bit 製造商代碼

編碼方式選擇
- Simple (簡易編碼法)
- Normal (標準編碼法)
- Secure (安全編碼法)

4-bit按鍵＋28-bit序號

64-bit 編碼密碼

產生編碼密碼 (Encryption Key)

實驗 一

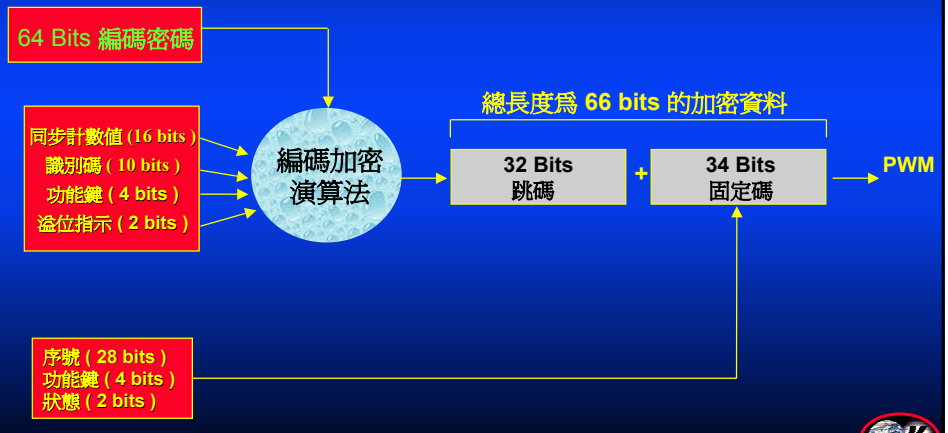
- (簡易編碼法) Simple Encoded
  - 製造商代碼 = 0123456789ABCDEF
  - 序號 = 00001234
  - 產生的 64-bit 編碼密碼 = _____
  - ♣ 試著改變序號 = 00001235
  - ♣ 新產生的 64-bit 編碼密碼 = _____
- (標準編碼法) Normal Encoded
  - 製造商代碼 = 0123456789ABCDEF
  - 序號 = 00001234
  - 產生的 64-bit 編碼密碼 = _____
  - ♣ 試著改變序號 = 00001235
  - ♣ 新產生的 64-bit 編碼密碼 = _____

KeeLoq®

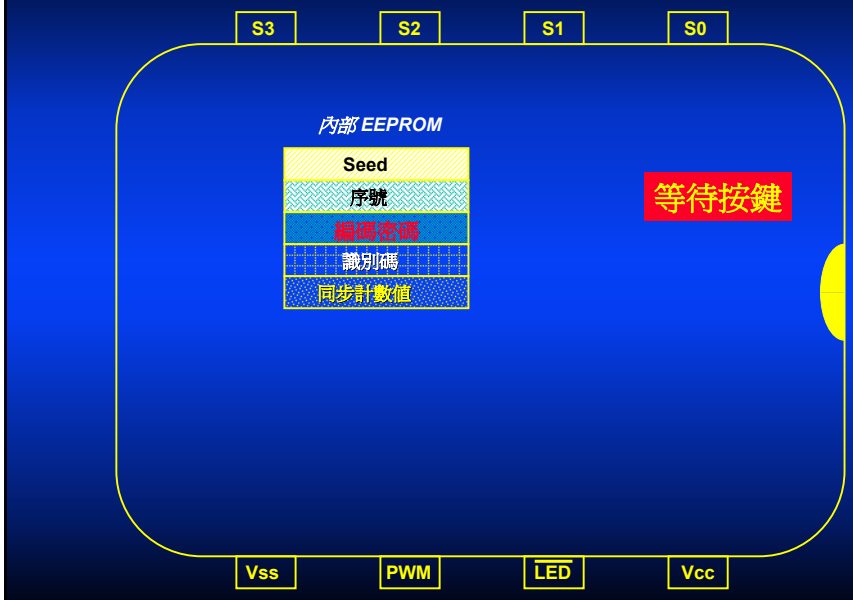
HCS300 動作流程

HCS300傳送編碼方式說明

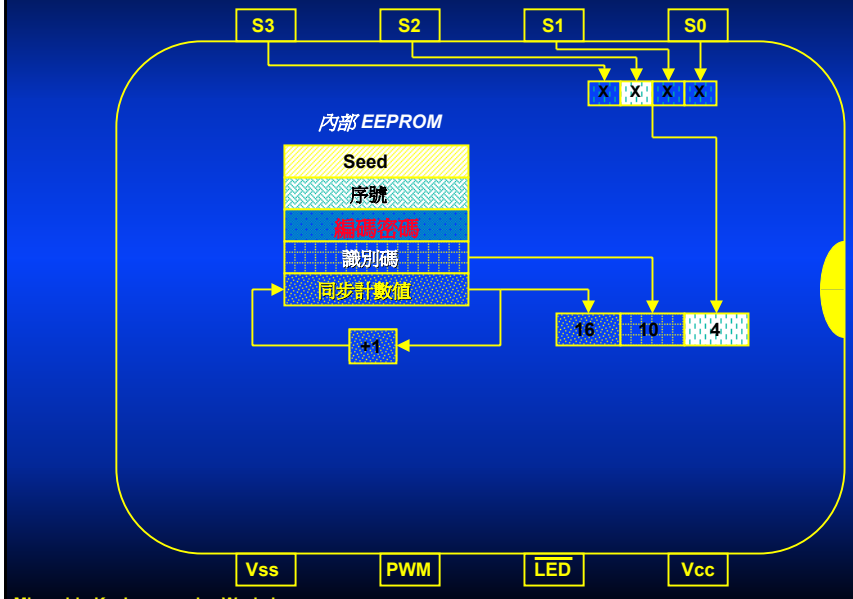
每當遙控器的按鍵被按下時，HCS300 即會將下列的資料傳送出去



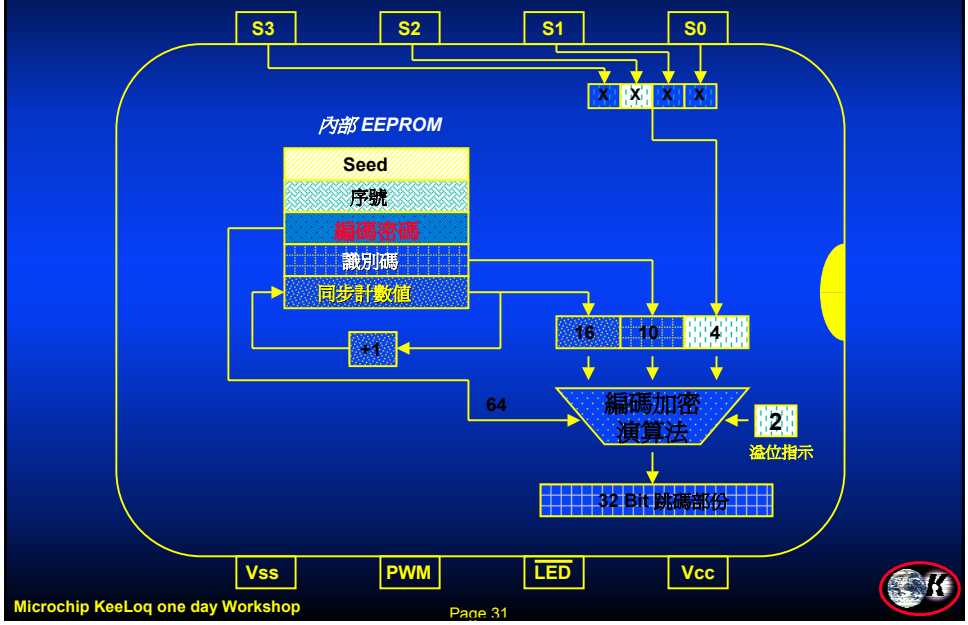
HCS 編碼流程詳述 (一)



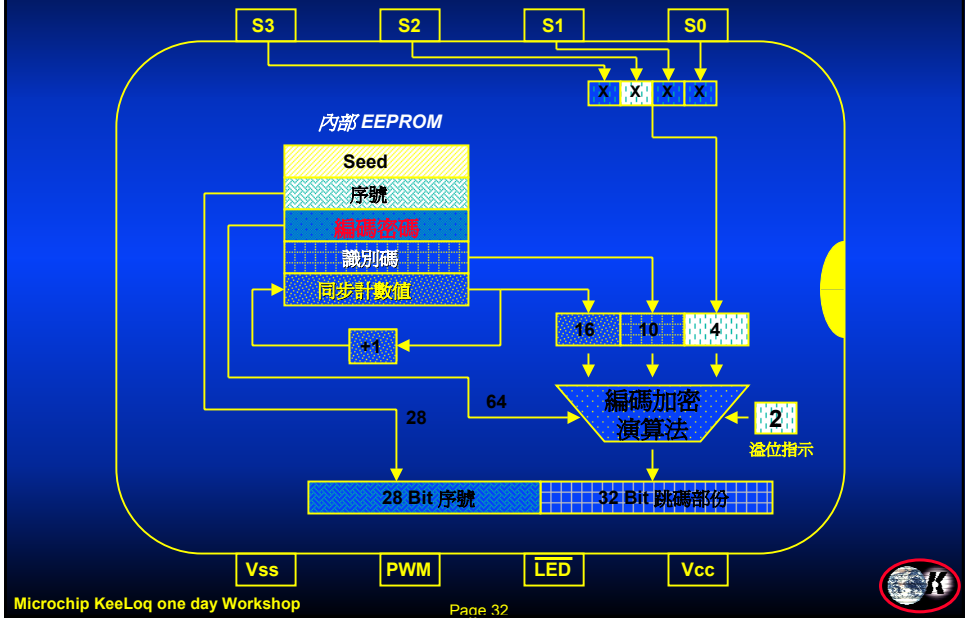
HCS 編碼流程詳述 (二)



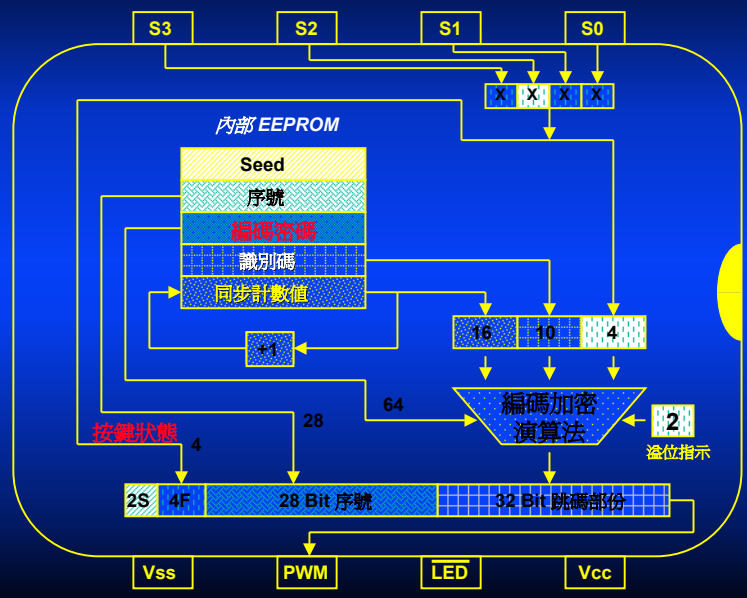
HCS 編碼流程詳述 (三)



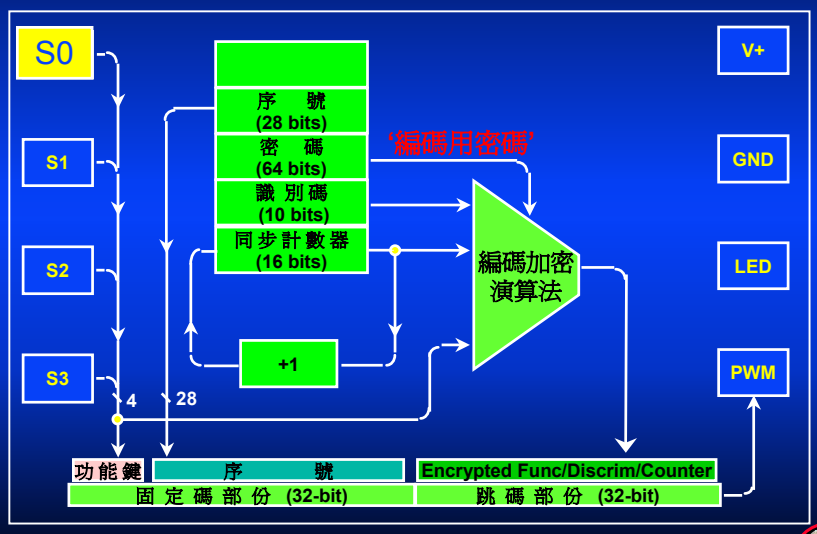
HCS 編碼流程詳述 (四)



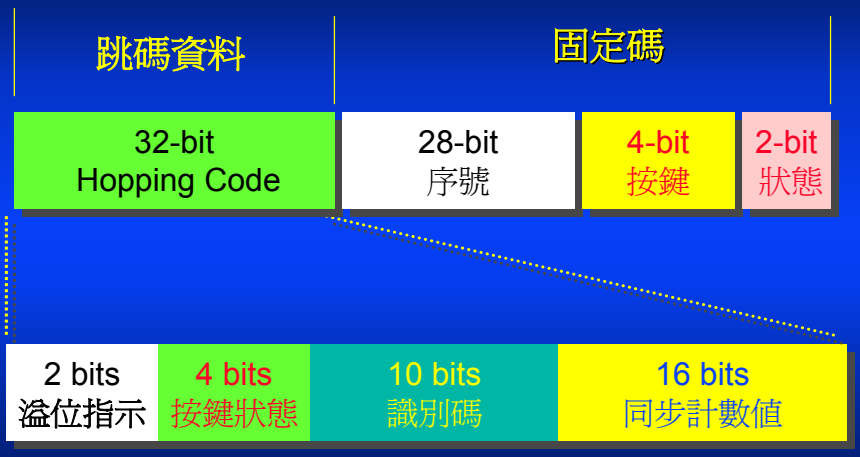
HCS 編碼流程詳述 (五)



編碼方式再看一次



32 bits Hopping Code



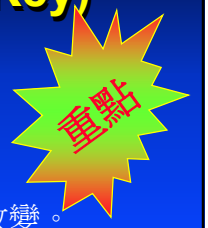
32 bits Hopping資料的組成原件

編碼的重要觀念

- 要啓動編碼器 (HCS300) 只需將按鍵按下即可
- 每次按鍵均會產生一組新的編碼，內部 IC 的基本動作會如下：
  - 同步計數值會自動加一後再存入其 EEPROM
  - 同步計數值，識別碼和功能鍵會重新編碼加密後以產生一組新的跳碼 (Hopping Code)
- 新產生的 66 bits 資料碼，會被傳送到接收器進行解碼的動作。

重要觀念

產生編碼密碼 (Encryption Key)



- (簡易編碼法) Simple Encode
  - 編碼密碼 (Encryption Key) = 製造商代碼
  - 編碼密碼 (Encryption Key) 不會隨著序號改變。
- (標準編碼法) Normal Encode
  - 編碼密碼 (Encryption Key) 不等於 製造商代碼
  - 編碼密碼 (Encryption Key) 是由製造商代碼及序號共同產生，任何一項改變編碼密碼 (Encryption Key) 也會更著改變。

實驗二：產生 Hopping Code

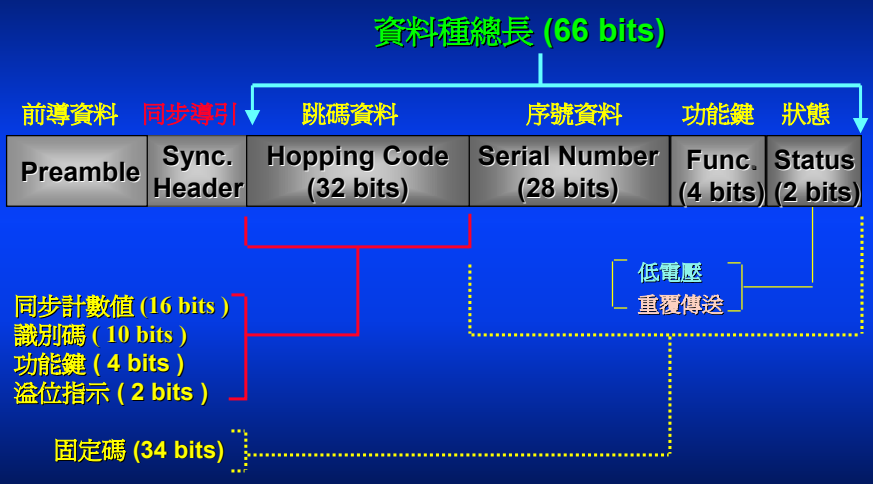
- 底下的 Hopping Code 是使用 (0~F) 的製造商代碼以簡易編碼方式 (Simple Encode) 來記錄 **HCS300** 所送出的編碼作爲紙上解碼的輸入
- HCS300 燒錄設定：[序號 0001234] ， [同步計數起始值 0000] [識別碼序號的最低10-bit]

跳碼 (Hopping Code)	+ 固定碼 (Fixed Code)
A128D1A4	0001234 F=2 V=0 R=1
AF24B45D	0001234 F=2 V=0 R=1
23CDC957	0001234 F=4 V=0 R=0
B4D9F196	0001234 F=8 V=0 R=0
20B5E2BE	0001234 F=1 V=0 R=0
0522C4DC	0001234 F=2 V=0 R=1
813EFA40	0001234 F=4 V=0 R=1
44BB29DE	0001234 F=8 V=0 R=0
D4934126	0001234 F=8 V=0 R=0

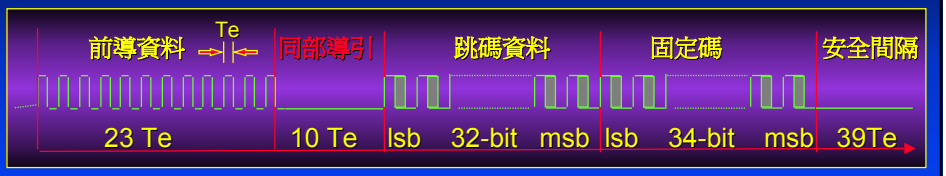
KeeLoq®

PWM 資料接收

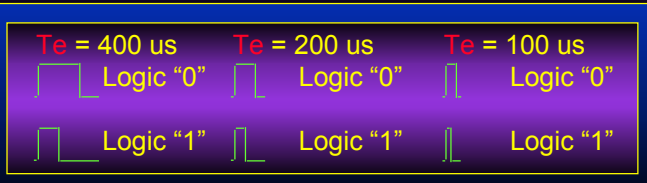
傳送資料格式



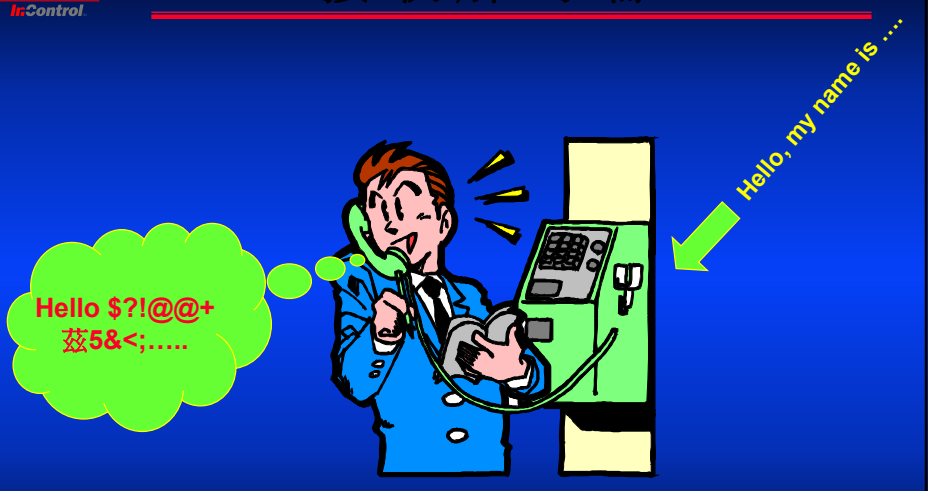
傳送資料格式 (continued)



BSL1 : BSL0	0 : 0	0 : 1	1 : 0	1 : 1
Te	400 us	200 us	100 us	100 us
Frame Time	108 ms	54 ms	27 ms	27 ms
BACW	1x 100 %	1x 50 %	2x 25 %	1x 25 %

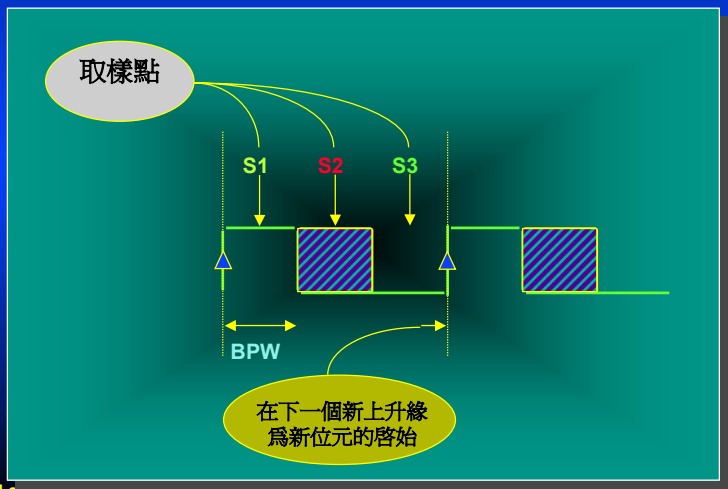


接收解碼端



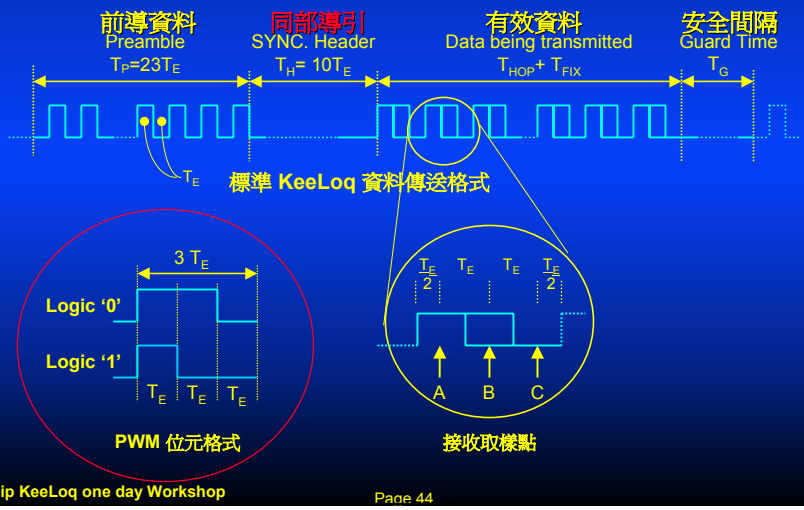
接收解碼的取樣點

接收端需攫取 66 bits 的資料存入 Buffer 以供解碼之用



如何接收 PWM 資料

- 取樣規則



PWM 接收方法

1. 利用前導資料的指引，準備接收資料
2. 利用同部導引 (Sync. Header 10 T_E) 做爲接收速率校正，並得到 T_E 的時間
3. 在偵測的第一個上升緣，等待 1/2 T_E 時間後立即取樣 (取樣點 A)並測試是否爲 1，如果爲 0 則接收資料失敗。
4. 等待 1 T_E 時間後，立即取樣 (取樣點 B) 做爲資料位元
5. 再等 1 T_E 時間後，再取樣 (取樣點 C)並測試，如果爲 1 則接收資料失敗。
6. 等待下一個上升緣的到來，如等待時間超過 1 T_E 則接收資料失敗。
7. 重複上述 3 ~ 6 的動作，直到完成 66 bits 接收

KEELOQ®

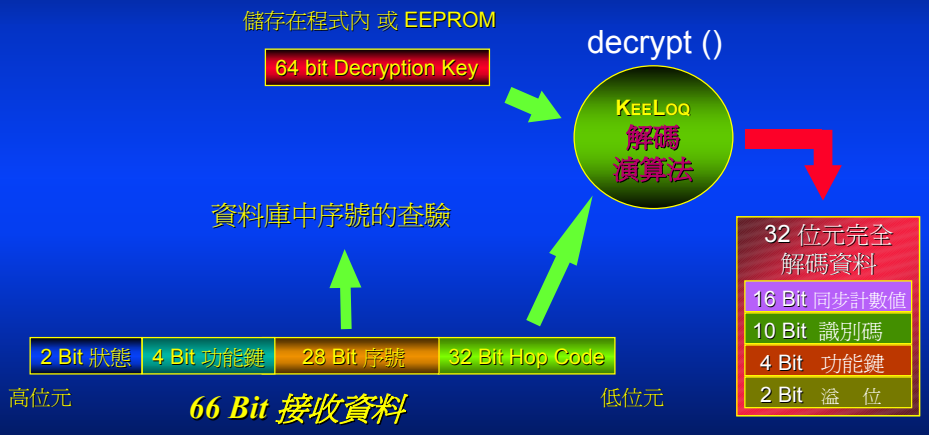
Hopping Code
解碼 與 判斷

KeeLoq 解碼功能方塊

KeeLoq 解碼副程式

- Microchip Technology 提供 KeeLoq 解碼副程式以幫助客戶開發產品。
- KeeLoq 解碼副程式可從 “KeeLoq License Disk” 磁片中獲得。
- KeeLoq 解碼副程式包含下列內容：
  - PWM 接收副程式
  - 解碼副程式 { decrypt ( ) }
  - 同步計數檢驗副程式
  - EEPROM 讀 / 寫 副程式

基本解碼方法



解碼的確認

重點

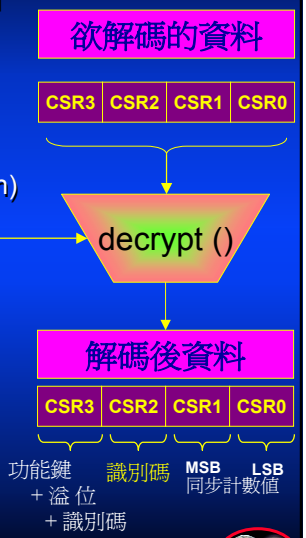
- 接收到一有效的 KEELOQ 資料，共 66 bits 。
- 檢查接收資料的固定碼部份是否與資料庫中的序號相同。
- 自資料庫中取出 64-bit 的 “Encryption key” 。
- 將接收到的資料加以解碼產生四種資料：
  - 功能鍵，溢位，識別碼，同步計數值
- 檢驗 10-bit 的 “識別碼”：
  - 識別碼的值(內定) 與 序號 (Serial Number) 的低 10 位元相等
- 比較固定碼中的“功能鍵”值 與 解碼後的“功能鍵”值是否相等：
  - 注意 - 按鍵排列順序爲 (MSB) S2 , S1, S0 , S3 (LSB)
- 檢查“同步計數值”的變化是否正確。

KeeLoq 解碼程式說明

- 採用 32 bit 非線性演算法則 [Decrypt ( )]
  - 不用管它(一大堆旋轉與互斥運算不太容易融會貫通，拿來解碼就是了)。
- 64 bit 解碼用的密碼
  - 直接就使用製造商代碼 (Simple Learn)
  - 可從編碼器的傳送資料中獲得 (Normal Learn)

KEY7	KEY6	KEY5	KEY4	KEY3	KEY2	KEY1	KEY0
MSB							LSB

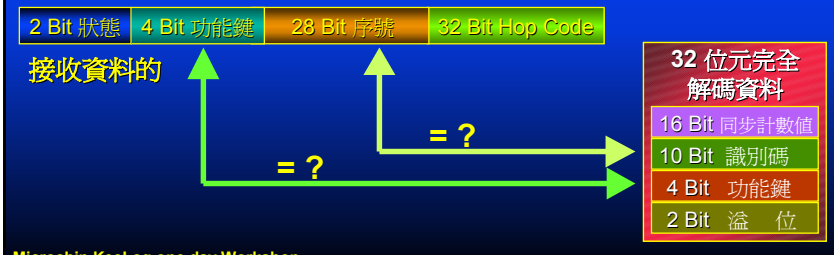
- 解密用的密碼 (共64 bit) 放在：(KEY7,KEY6,KEY5,…,KEY0)
- 32-bit 的 Hopping Code (接收到的)必須放在 (CSR3 , CSR2 , CSR1 , CSR1)
  - CSR3 的值爲最高位元組 (MSB)



解碼後資料的判斷

重點

- 比較解碼後的 “識別碼” 與接收資料的 “序號” (Serial Number)中較低的 10 位元是否相等。
- 比較解碼後的 “功能鍵” 與接收資料的 “功能鍵” 是否相等。
- 判斷解碼後的 “同步計數值” 與資料庫中相對應的舊 “同步計數值” 是否合理增加。



同步計數值的概念

127

− 123

答案是？

假設新接收到的 (解碼後) 同步計數值爲 127

EEPROM 儲存的同步計數值爲 123
(此值爲上一次的接收同步計數值)

- 接收器 "知道" 上一次的同步計數值
- 新同步計數值 "減去" 舊同步計數值
- 說明這個答案是 .....????

同部計數值的檢查

重點

- 上一次的接收同步計數值來自資料庫中
- 新接收到的 (解碼後) 同步計數值必須 大於 舊的同部計數值
- 假設新接收到的同步計數值 等於 或 小於舊的同步計數值時，表示有仿冒資料入侵。
- 具有自動同步功能:
  - 當新接收到的同步計數值 超過 舊同步計數值甚多時，則須在一定的時間內重新再接收一筆新的資料，且其同步計數值之差必須 小於 4
  - 同步計數值會自動採用此新值

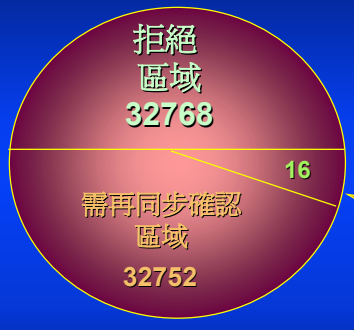
同步計數值的檢查範例

儲存的計數值	新接收的計數值	差距	後續的動作
100	98	-2	不動作
100	105	5	動作
105	125	20	不動作 等下一筆命令
105 (125 temp stored in RAM)	126	21 (1)	重新同步並動作

假設同步確認確認區域的差設定爲 16

KeeLoq 內定的同步計數器

旋轉式的同步計數檢驗方式

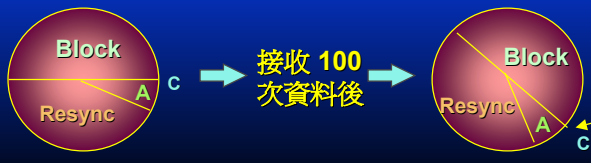


16-Bit 同步計數器 = 2^{16} = 65536

目前同步計數器的位置

接受動作區域

範例:



KeeLoq®

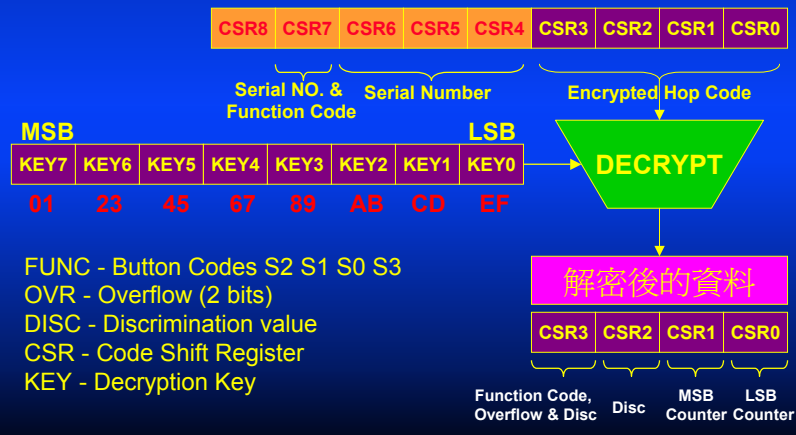
紙上解碼介紹

RECEIVE 的動作說明

- RECEIVE 接收由Keeloq Encoder 傳來的資料
- 接收到的資料將被置於 CSR0 .. CSR8 中
- 只要在程式中以 “call RECEIVE” 指令即可叫用此副程式
- 執行結果將反映於 W 暫存器中；C=1 資料接收完成；C=0 無可用資料或錯誤

DECRYPT 副程式說明

- 主程式以 “call DECRYPT” 指令來叫用解密副程式
  - CSR3 .. CSR0 爲欲解密的 Hopping Code
  - KEY7 .. KEY0 爲使用的 DECRYPTION KEY !!



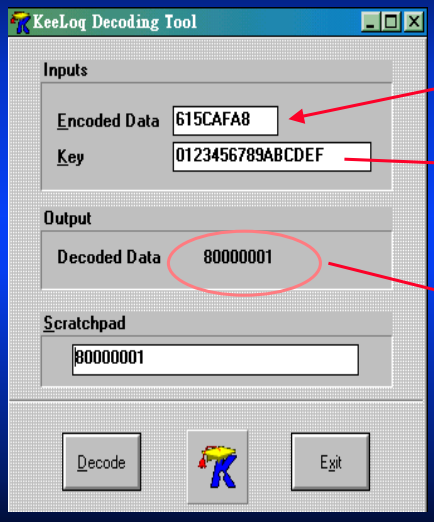
FUNC - Button Codes S2 S1 S0 S3
OVR - Overflow (2 bits)
DISC - Discrimination value
CSR - Code Shift Register
KEY - Decryption Key

紙上解碼演練 - 實驗三

- 以實驗二所燒錄的 HCS300 做爲編碼器
  - 製造商代碼：0123456789ABCDEF
  - 使用簡易編解碼方式
- 將實驗二所收到的 “Hopping” 碼填入 “KeeLoq Decoding Tool” 中的解碼資料欄中來解碼
- 嘗試將解碼後的值進一步分解，並注意其資料變化？

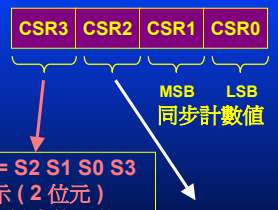
KeeLoq Decoding Tool Icon

Hopping 解碼說明



填入“Hopping” 碼

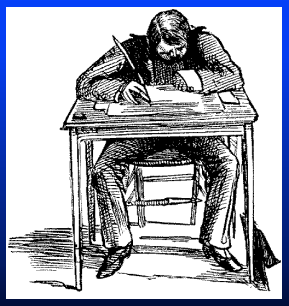
製造商代碼



紙上解碼練習

跳碼 (Hopping Code)	經過解碼值	功能鍵	溢位	識別碼	同步計數值
A128D1A4	22340001	0010b	00b	234h	0001h
_AF24B45D____	_____	_____	____	_____	_____
_23CDC957____	_____	_____	____	_____	_____
_B4D9F196____	_____	_____	____	_____	_____
_20B5E2BE____	_____	_____	____	_____	_____
_0522C4DC____	_____	_____	____	_____	_____
_813EFA40____	_____	_____	____	_____	_____
_44BB29DE____	_____	_____	____	_____	_____
_D4934126____	_____	_____	____	_____	_____

KeeLoq 的學習模式
&
解碼密碼是如何產生的？



KeeLoq 三種學習模式

- 簡易學習模式 - Simple Learn
- 標準學習模式 - Normal Learn
- 安全學習模式 - Secure Learn

KEELOQ 的學習步驟就是：

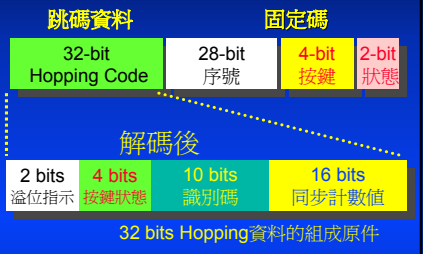
1. 接收跳碼資料
2. 進行資料解碼
3. 核對解碼後資料
4. 儲存學習資料

KEELOQ 系統爲何要學習？

- 因爲生產配對方便，管理簡單
- 因爲解碼器一開始時，除了製造商代碼外，什麼都不知道！！！
- 接收解碼器須要眾多的解碼資訊儲存在 EEPROM 中，而這些資訊的提供者是遙控發射器 (編碼器):
  - 序號 (Serial Number)
  - 同步計數值 (Current Sync. Counter Value)
  - 識別碼 (Discrimination Value)
  - 編碼密碼 (Encryption Key)

HCS300 所提供的資訊

- 序號 (Serial Number)
  - 28 bit 的序號是放在固定碼中
  - 可自接收資料中直接取得
- 同步計數值 (Sync. Counter Value)
  - 16 bit 的長度，隱藏在跳碼裡
  - 無法直接得到，須透過解碼後才可取得
- 識別碼 (Discrimination Value)
  - 10 bit 的長度，隱藏在跳碼裡
  - 無法直接得到，須透過解碼後才可取得
  - 如不特別指定，識別碼的值是等於序號中較低的 10 bit



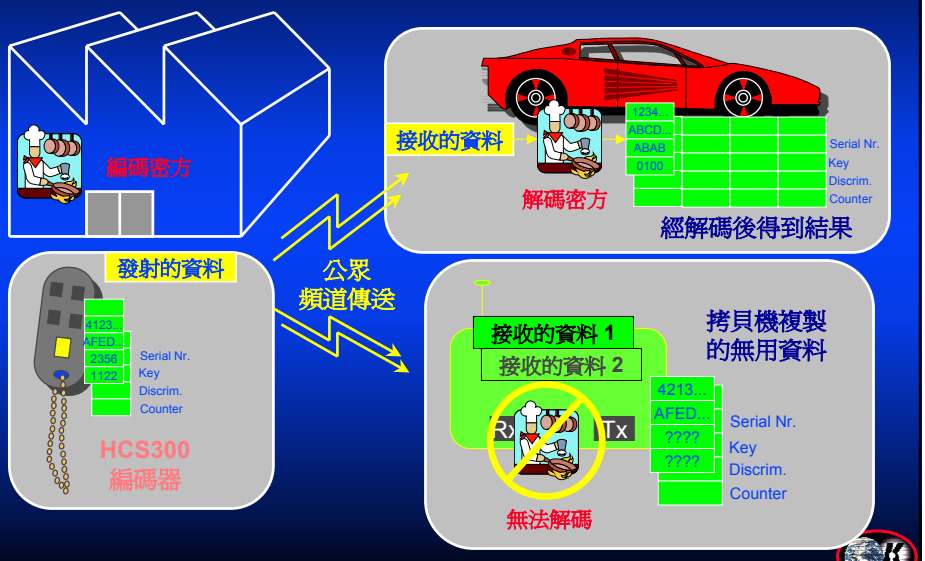
奇怪啦！解碼密碼怎麼不在傳送的資料裡呢？

解碼密碼原理

做菜大家都會

- 雞蛋、豬肉、高麗菜、洋蔥 大家都看的到（序號，按鍵，跳碼 ）
- 但是加入了 調味料，使每個人煮出來的味道都有差異；如果加入的是獨家調味密方則這結果就是獨一無二的。
- 這獨家調味密方就是：
  - 製造商代碼 (Manufacture's Code) - Simple Learn
  - 解碼密碼 (Encryption Key) - Normal Lrarn

遙控資料是透過公眾頻道



解碼密碼在那裡？

- 因資料是在公用頻道上傳送，所以必須要有一些保密絕招來防止資料被破解。
- 實際上，跳碼部份是由 編碼密碼 來產生的，只是無法預測及目視出答案。

重點 ☛ 編碼密碼 == 解碼密碼。

重點 ☛ 解碼密碼 的產生須透過 製造商代碼 與 序號 的運算後才能得到。(對接收機而言，序號從那裡來？)

- 由此可見製造商代碼是系統裡，唯一不可洩漏的真正密碼。
- 如果製造商代碼被知道，則系統就有可能被破解；否則是不可能被破解的。

KeeLoq 基本解碼元件摘要

① 原料一：製造商代碼 (Manufacturer's Code)

- 總長有 64 bit - 對製造商的產品而言是唯一 (不重複)

② 原料二：

- 簡易學習模式 (Simple Learn) - 不需要
- 標準學習模式 (Normal Learn) - 序號 (Serial Number)
- 安全學習模式 (Secure Learn) - 種子編號 (SEED) + 序號 (Serial Number)

③ 解碼副程式：decrypt ( )

- 解碼演算法則 或 XOR 運算法則
- 將原料一及原料二的資料加以解碼
- 此解碼程式由 Microchip 提供

Simple Learn

簡易學習模式
Simple Learn

透過一次的學習，取得下列資料:

序號 (Serial Number)

識別碼 (Discrimination Value)

同步計數值 (Sync. Counter Value)

Simple Learn

- 在簡易學習模式，編碼密碼 (Encryption Key) 是怎麼取得的？

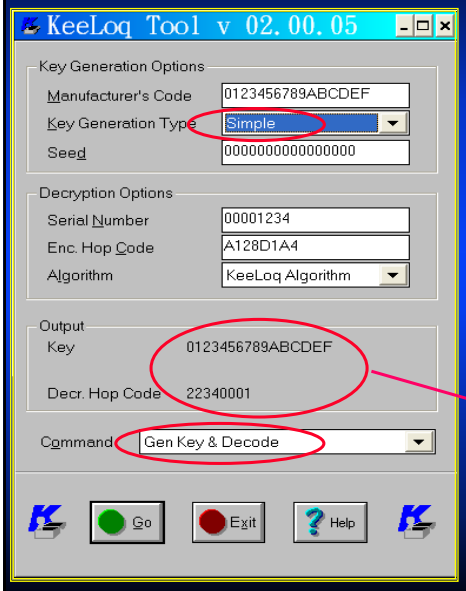
答案：製造商代碼 就等於 解碼密碼 (Encryption Key)

優點：簡單易懂、解碼程式較短。

缺點：假如系統是採用簡易學習模式的話，如果製造商代碼被知道，則其它使用相同製造商代碼的系統就有可能被破解。

動手實驗 - 簡易學習模式一

Simple Learn



假設編碼器的設定爲：

製造商代碼 = 0123456789ABCDEF

序號 = 00001234

S0 的按鍵被按下 (按鍵碼 = 2)

在簡易學習模式下傳送出去的資料：

序號 = 00001234

跳碼 = A128D1A4

使用 "KeeLoq Tool" 練習解碼：

1. 編碼密碼 (Encryption Key)
   = _____
2. 同步計數值 (Sync. Counter Value)
   = _____
3. 識別碼 (Discrimination Value)
   = _____
4. 按鍵值 (Button) S2,S1,S0,S3
   = _____

動手實驗 - 簡易學習模式二

Simple Learn

- 以上一實驗爲基楚:
  - 改變序號 = 00000345，則解碼結果是否改變？

    _____
  - - 改變製造商代碼 = 1123456789ABCDEF ，

      則解碼結果是否改變？ _____

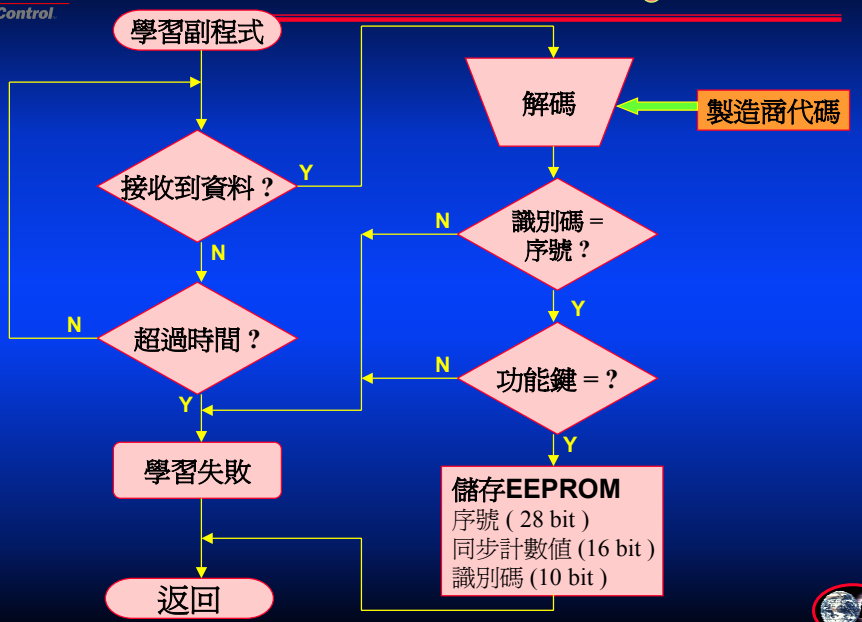
    - 是否依舊可以將跳碼解碼出來？ _____
  - 是否可說明識別碼 (Discrimination Value) 的

    功能？(假設識別碼是自己指定的值≒序號)

    _____

簡易學習模式 - 學習流程圖

Simple Learn



簡易學習模式 - 辨識資料

Simple Learn

- 一個正確的簡易學習模式，必須由解碼端起動 (例如按了某鍵來起動學習程序) 。同樣的解碼器也應在一定的時間內接收到一組經相同製造商代碼編碼過的遙控資料。

解 碼 ❄ 接收資料經製造商代碼解碼後，會得到 功能鍵、溢 位、識別碼 及 同步計數值

檢驗 1 ☛ 識別碼 (假設使用內定值) 與 所接收的 序號 中較低的 10 位元，是否相等？

檢驗 2 ☛ 解碼後的 功能鍵 與 所接收資料中的固定碼之 按鍵值 ，是否相等？

結 束 ✳ 檢查結束，最後可得到三種學習資料：

- 序號 (28 bit)
- 同步計數值 (16 bit)
- 識別碼 (10 bit)

簡易學習模式 - 資料儲存

Simple Learn

- 解碼器在經過正確的學習後，必須將下列三種資料記錄在 EEPROM 中，做爲解碼資料庫用：
  - 序號 (Serial Number) - 28 bit
  - 同步計數值 (Sync. Counter Value) - 16 bit
  - 識別碼 (Discrimination Value) - 10 bit
- 另 解碼密碼 (Encryption Key) 或 製造商代碼 (manufacture's Code) 一般而言，是以表格 (Table) 的方式儲存在程式裡。
- Microchip 所提供的範例中，對簡易學習模式只學習一次。實際上，使用者可依實際之需要加以修改。

實驗四 – 簡易學習模式程式演練

- 啓動 MPLAB IDE v6.xx，載入KeeLoq.mcp
- 此 Project file 共有三個檔案
  - Main.asm – 主程式
  - Keeloq.asm – KeeLoq 編解碼副程式
  - 16f877i.lkr – MPLINK的連結描述檔

- 設定製造商代碼(0~F)共64位元
- 將實驗二的Hopping Code填入到變數位址 CSR3~CSR0
- 利用設定中斷執行點的除錯方式觀察解碼後的值

Normal Learn

標準學習模式
Normal Learn

第一次學習，取得下列資料:

- 解碼密碼 (Encryption Key)
- 序號 (Serial Number)
- 識別碼 (Discrimination Value)
- 同步計數值 (Sync. Counter Value)

第二次的學習，檢查同步計數值後儲存學習結果到 EEPROM

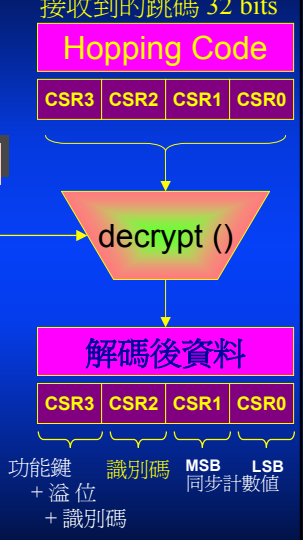
Normal Learn

- 標準學習模式 與 簡易學習模式的解碼方式其實是一樣地，只是使用不同的 解碼密碼 (Encryption Key)。這時候：

製造商代碼 就不等於 解碼密碼 (Encryption Key)

KEY7	KEY6	KEY5	KEY4	KEY3	KEY2	KEY1	KEY0

MSB 解碼密碼 (64 bit Encryption Key) LSB

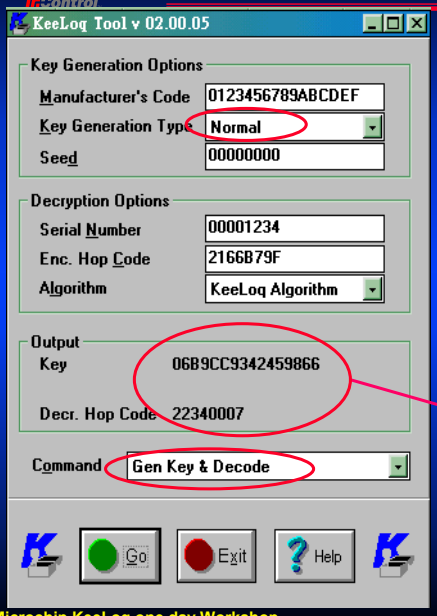


- 在標準學習模式中，解碼密碼 (Encryption Key) 是怎麼取得的？

答: *解碼密碼是必須利用製造商代碼與序號 經過兩次的運算後方可得到這 64 bit 的解碼密碼。*

動手實驗 - 標準學習模式一

Normal Learn



假設編碼器的設定爲：

製造商代碼 = 0123456789ABCDEF

序號 = 00001234

S0 的按鍵被按下 (按鍵碼 = 2)

標準學習模式下傳送出去的資料：

序號 = 00001234

跳碼 = 2166B79F

使用 “KeeLoq Tool” 練習解碼：

1. 解碼密碼 (Encryption Key)
   = _____
2. 同步計數值 (Sync. Counter Value)
   = _____
3. 識別碼 (Discrimination Value)
   = _____
4. 按鍵值 (Button) S2,S1,S0,S3
   = _____

動手實驗 - 標準學習模式二

Normal Learn

假設另外一個編碼器的設定爲：

製造商代碼 = 0123456789ABCDEF

序號 = 00000345 (被改變過)

S0 的按鍵被按下 (按鍵碼 = 2)

標準學習模式下傳送出去的資料：

序號 = 00000345

跳碼 = 4D80ED88

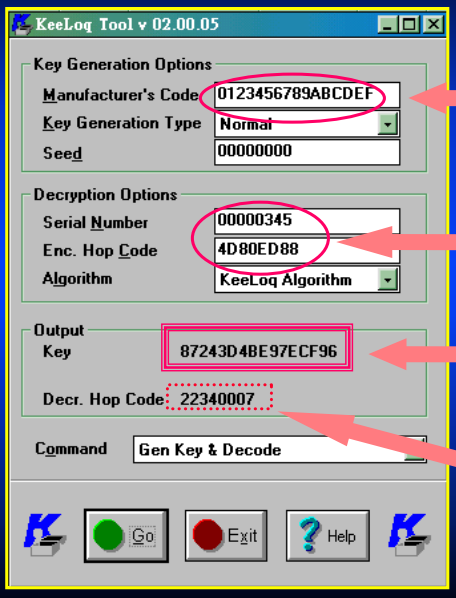
使用 “KeeLoq Tool” 練習解碼：

1. 新的解碼密碼 (Encryption Key) = _____
2. 同步計數值 (Sync. Counter Value) = _____
3. 識別碼 (Discrimination Value) = _____
4. 按鍵值 (Button Code) S2, S1, S0, S3 = _____

試著用舊的解碼密碼，是否可能解出正確的碼？

動手實驗 - 標準學習模式

Normal Learn



製造商代碼通常是儲存在程式裡，並可在經自己編碼保密

所接收到的 66 bit 傳輸資料

新的解碼密碼 (Encryption Key) = 87243D4B E97ECF96
注意：序號改變，解碼密碼也改變

同步計數值 = 0007h
識別碼 = 234h
按鍵值 = 2h

解碼密碼 (Encryption Key) 的產生

Normal Learn

- 由前面的實驗，我們可以知道 解碼密碼 是和 序號 是有關係的。
- 實際上，解碼密碼是利用 製造商代碼 與 序號 經過兩次的運算後方可得到這 64 bit 的解碼密碼：
  - 第一次運算，先取得 32 bit (LSB)
  - 第二次運算，再取得 32 bit (MSB)
- 所謂"運算"，就是使用前面所使用過的解碼副程式：decrypt( )
  - 可以用 decrypt( ) 來解跳碼 (與 製造商代碼 或 解碼密碼)
  - 也可以用 decrypt( ) 來產生 解碼密碼 (與 序號 或 種子編號)

解碼副程式 decrypt( )

Normal Learn

解跳碼 (Hopping Code)

32 bit 跳碼

Simple

製造商代碼 或 解碼密碼

Normal

Dectypt( )

32 bit 解碼資料

產生解碼密碼 (Encryption Key)

補 2 6 +28 bit 序號 輸入32 bit

Dectypt( )

製造商代碼 64 bit

32 bit 解碼資料 補 2 得 32 bit (LSB)

32 bit 解碼資料 補 6 得 32 bit (MSB)

合起來共 64 bit 的解碼密碼

產生解碼密碼的密訣 (Encryption Key)

Normal Learn

步驟 1 : 設定 X = 2 ，產生 32 bit 的 LSB

步驟 2 : 設定 X = 6 ，產生 32 bit 的 MSB

製造商代碼 64 bit

01	23	45	67	89	AB	CD	EF
KEY7	KEY6	KEY5	KEY4	KEY3	KEY2	KEY1	KEY0

MSB LSB

序號

X+ 0000345

CSR3	CSR2	CSR1	CSR0

decrypt ()

CSR3	CSR2	CSR1	CSR0

新產生 64 bit 的解碼密碼

87 24 3D 4B E9 7E CF 96

87243D4B (MSB)

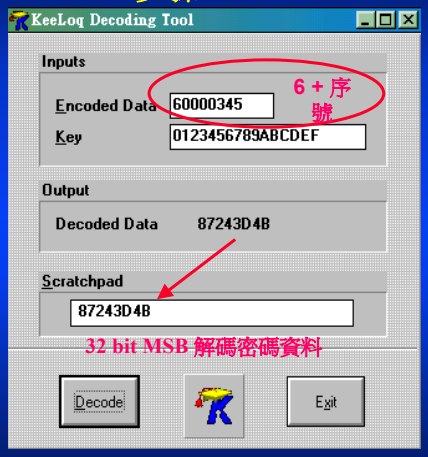
+

E97ECF96 (LSB)

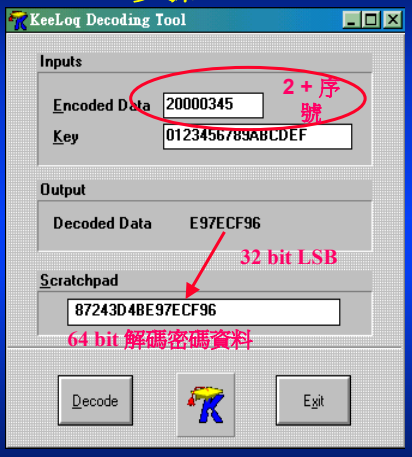
產生解碼密碼實驗

Normal Learn

步驟 1



步驟 2



有了解碼密碼後才能解跳碼以獲得

- 序號 (Serial Number)
- 識別碼 (Discrimination Value)
- 同步計數值 (Sync. Counter Value)

進入標準學習模式 (Normal Learn)

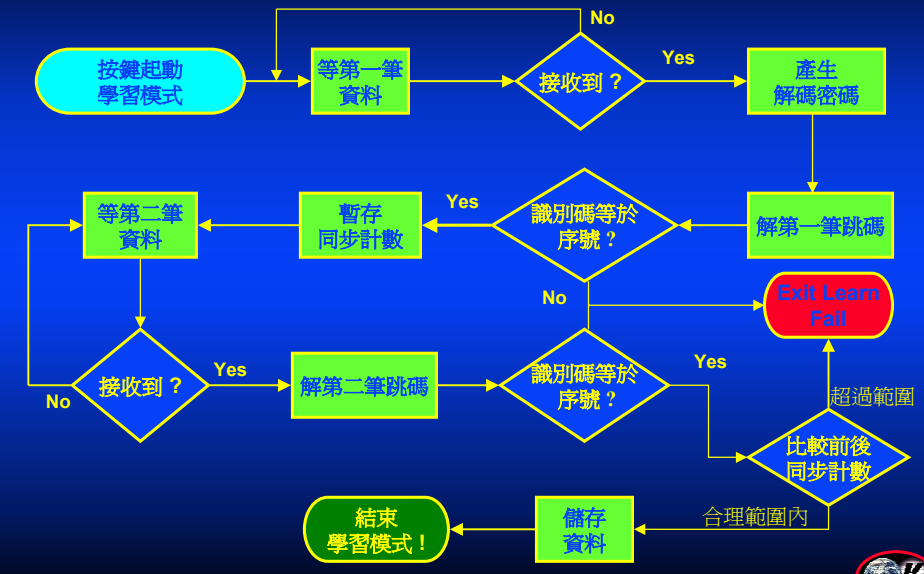
Normal Learn

已經知道解碼密碼後開始進入學習模式

- 將第一次接收的資料進行解碼後，則得到下列數據，並加以判斷：
  - 序號 (Serial Number)
  - 識別碼 (Discrimination Value) = 序號 ?
  - 同步計數值 (Sync. Counter Value)
- 等待第二次的接收資料，解碼後
  - 識別碼 (Discrimination Value) = 序號 ?
  - 檢查 同步計數值 是否加一
  - 儲存學習結果到 EEPROM

標準學習模式 - 學習流程圖

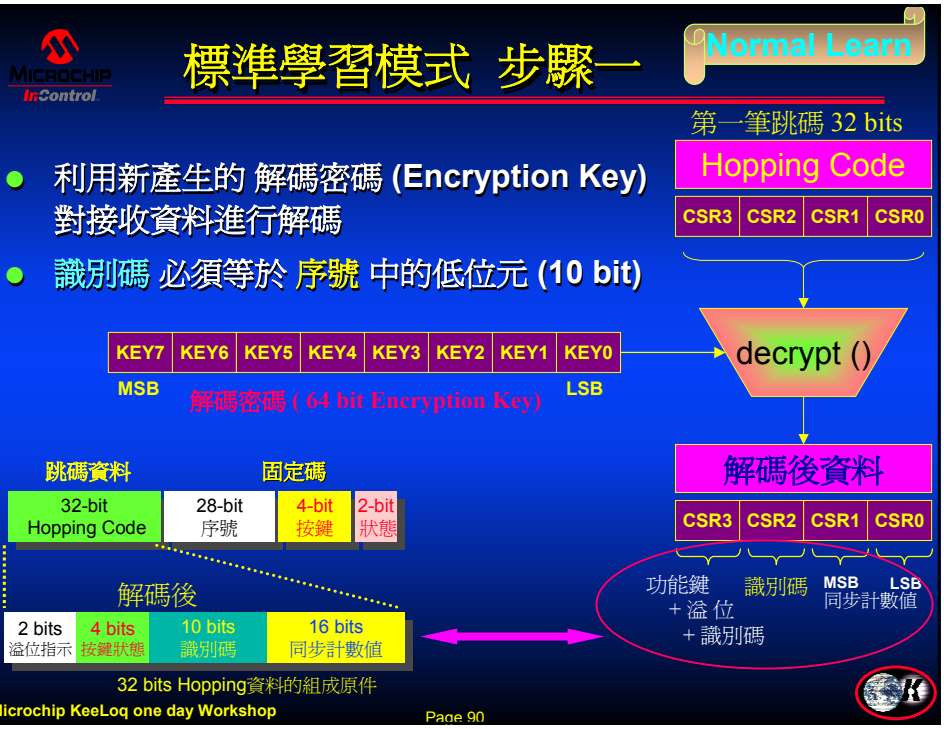
Normal Learn



標準學習模式 步驟一

Normal Learn

- 利用新產生的 解碼密碼 (Encryption Key) 對接收資料進行解碼
- 識別碼 必須等於 序號 中的低位元 (10 bit)



標準學習模式 步驟二

Normal Learn

- 在一定時間內，等待第二筆的資料輸入
- 對第二筆資料解碼 - 利用相同的 解碼密碼

解碼密碼 (64 bit Encryption Key)

KEY7	KEY6	KEY5	KEY4	KEY3	KEY2	KEY1	KEY0

MSB LSB

- 識別碼 必須等於 序號 中的低位元 (10 bit)
- 檢查目前的 同步計數值 是否比上一筆的 同步計數值 大於一
- 儲存學習結果 (序號 、識別碼、新同步計數值 及 解碼密碼) 到 EEPROM
- 使用 新同步計數值 爲同步點

第二筆跳碼 32 bits

Hopping Code

CSR3	CSR2	CSR1	CSR0

decrypt ()

解碼後資料

CSR3	CSR2	CSR1	CSR0

功能鍵 + 溢 位 + 識別碼　識別碼　MSB LSB 同步計數值

標準學習模式 解碼說明 一

Normal Learn

解 碼 器

學習模式 : 標準
製造商代碼 : 0123456789ABCDEF

接收到的資料
0 41234567 D4A34B0E

解碼器資料庫

序　號 (28 bits)	01234567
解碼密碼 (64 bits)	???
識 別 碼 (10 bits)	???
同步計數器 (16 bits)	???

製造商代碼
01234567 89ABCDEF

KEY7 ... KEY0
MSB LSB

decrypt ()

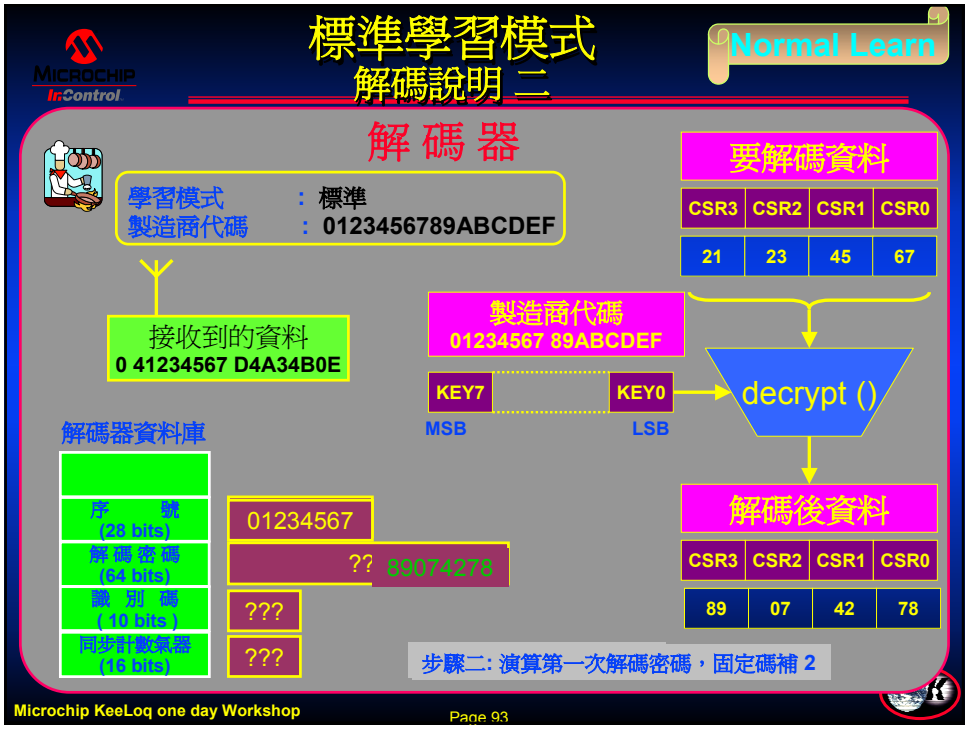
要解碼資料

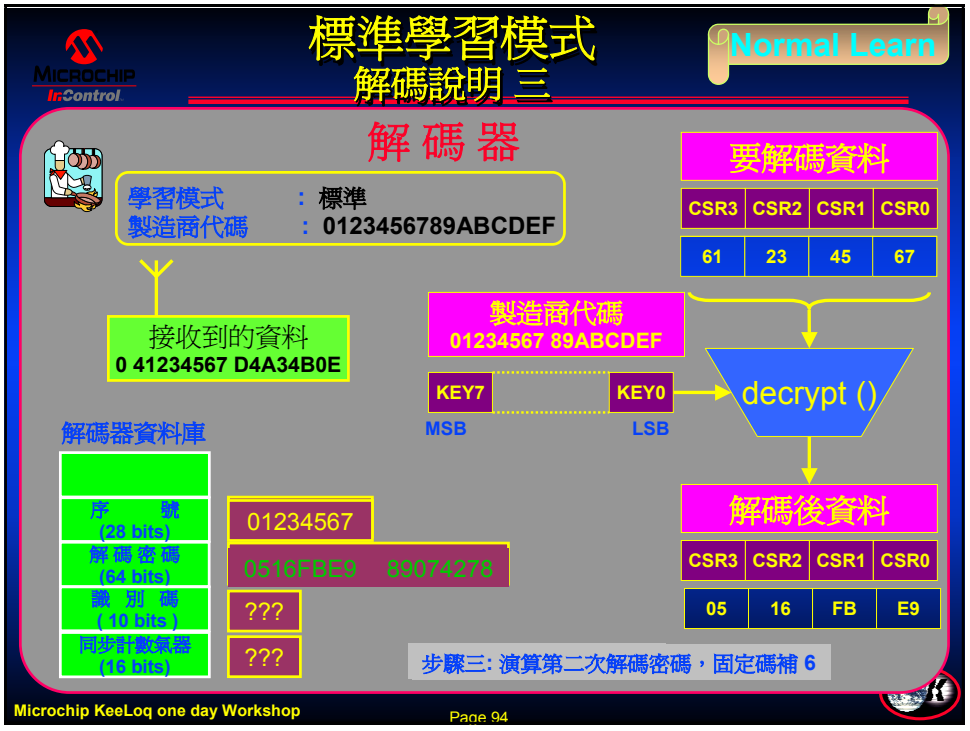
CSR3	CSR2	CSR1	CSR0

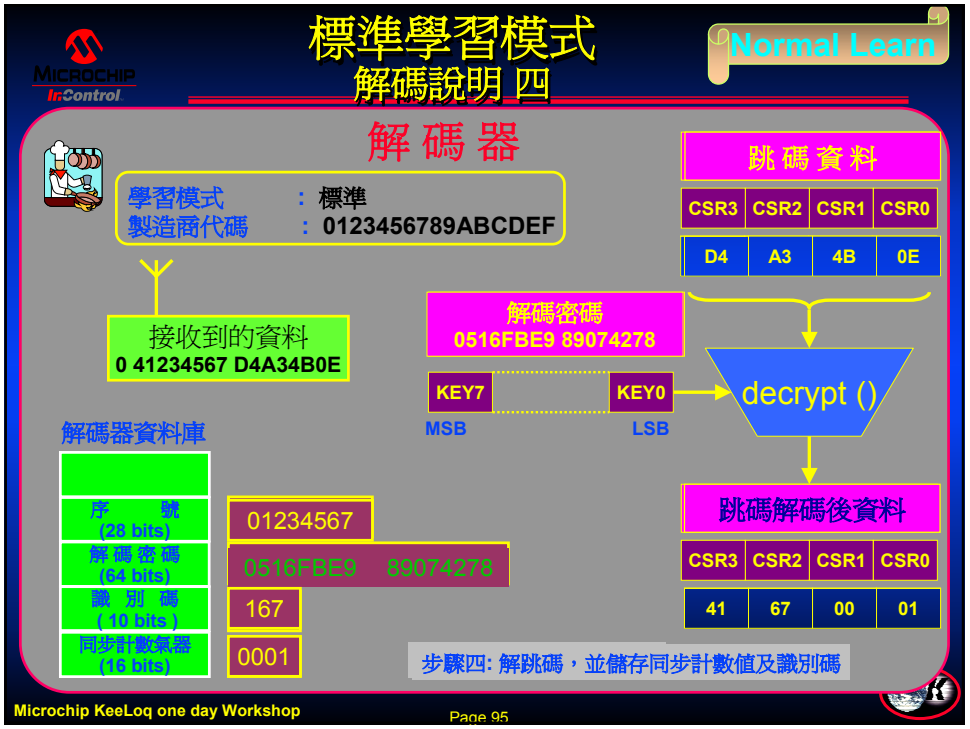
解碼後資料

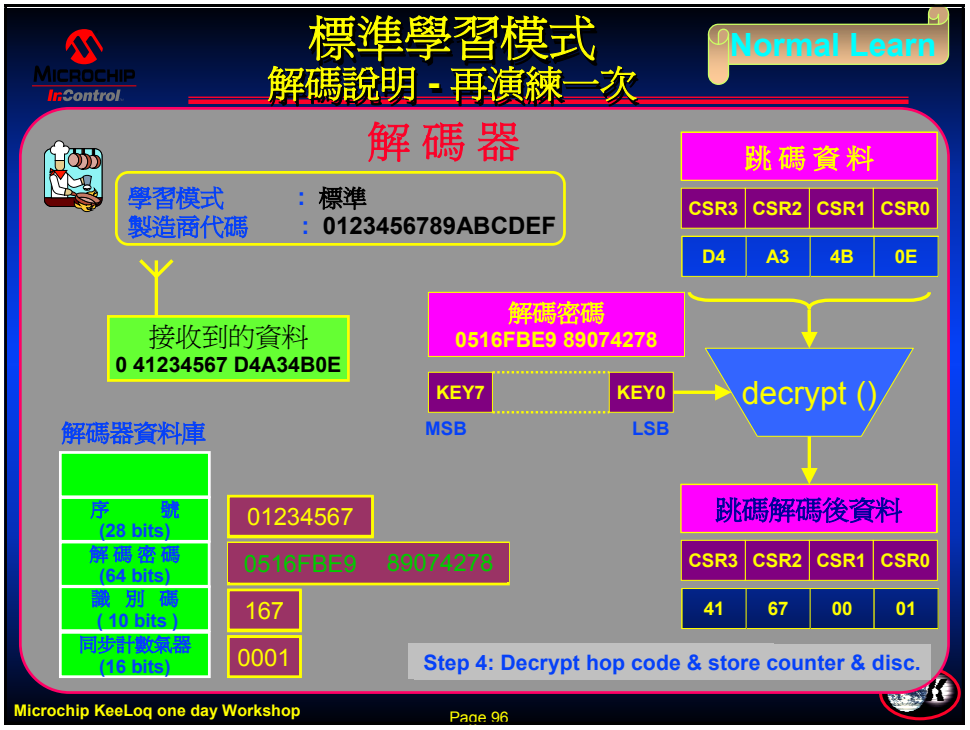
CSR3	CSR2	CSR1	CSR0

步驟一：載入 製造商代碼


標準學習模式
解碼說明 二
Normal Learn
解 碼 器
學習模式 : 標準
製造商代碼 : 0123456789ABCDEF
接收到的資料
0 41234567 D4A34B0E
要解碼資料
CSR3 CSR2 CSR1 CSR0
21 23 45 67
製造商代碼
01234567 89ABCDEF
KEY7 KEY0
MSB LSB
decrypt ()
解碼後資料
CSR3 CSR2 CSR1 CSR0
89 07 42 78
解碼器資料庫
序 號 (28 bits)
解碼密碼 (64 bits)
識 別 碼 (10 bits)
同步計數器 (16 bits)
01234567
?? 89074278
???
???
步驟二: 演算第一次解碼密碼，固定碼補 2
Microchip KeeLoq one day Workshop
Page 93


標準學習模式
解碼說明 三
Normal Learn
解 碼 器
學習模式 : 標準
製造商代碼 : 0123456789ABCDEF
接收到的資料
0 41234567 D4A34B0E
要解碼資料
CSR3 CSR2 CSR1 CSR0
61 23 45 67
製造商代碼
01234567 89ABCDEF
KEY7 KEY0
MSB LSB
decrypt ()
解碼後資料
CSR3 CSR2 CSR1 CSR0
05 16 FB E9
解碼器資料庫
序 號 (28 bits)
解碼密碼 (64 bits)
識 別 碼 (10 bits)
同步計數器 (16 bits)
01234567
0516FBE9 89074278
???
???
步驟三: 演算第二次解碼密碼，固定碼補 6
Microchip KeeLoq one day Workshop
Page 94


標準學習模式
解碼說明 四
Normal Learn
解 碼 器
學習模式 : 標準
製造商代碼 : 0123456789ABCDEF
接收到的資料
0 41234567 D4A34B0E
跳 碼 資 料
CSR3 CSR2 CSR1 CSR0
D4 A3 4B 0E
解碼密碼
0516FBE9 89074278
KEY7 KEY0
MSB LSB
decrypt ()
解碼器資料庫
序 號 (28 bits)
解 碼 密 碼 (64 bits)
識 別 碼 (10 bits)
同步計數器 (16 bits)
01234567
0516FBE9 89074278
167
0001
跳碼解碼後資料
CSR3 CSR2 CSR1 CSR0
41 67 00 01
步驟四: 解跳碼，並儲存同步計數值及識別碼
Microchip KeeLoq one day Workshop
Page 95


標準學習模式
解碼說明 - 再演練一次
Normal Learn
解 碼 器
學習模式 : 標準
製造商代碼 : 0123456789ABCDEF
接收到的資料
0 41234567 D4A34B0E
跳 碼 資 料
CSR3 CSR2 CSR1 CSR0
D4 A3 4B 0E
解碼密碼
0516FBE9 89074278
KEY7 KEY0
MSB LSB
decrypt ()
解碼器資料庫
序 號 (28 bits)
解 碼 密 碼 (64 bits)
識 別 碼 (10 bits)
同步計數器 (16 bits)
01234567
0516FBE9 89074278
167
0001
跳碼解碼後資料
CSR3 CSR2 CSR1 CSR0
41 67 00 01
Step 4: Decrypt hop code & store counter & disc.
Microchip KeeLoq one day Workshop
Page 96

實驗五 – 標準學習模式程式演練

- 同實驗四的方式 -- 啓動 MPLAB IDE v6.xx，載入KeeLoq.mcp
- 練習將 “標準學習模式” 投影片的解碼說明實際用軟體解碼的方式來產生解碼密碼
  - **解碼資料：序號 = 41234567，跳碼 = D4A34B0E**
  - 先將序號做 **補 2** 及 **補 6** 運算產生一組解碼密碼
  - 將新的解碼密碼和 Hopping Code 進行解碼
  - 驗證序號，同步計數值，功能鍵

Secure Learn

安全學習模式
Secure Learn

解碼密碼取得的方式是：

- **利用種子編號產生32 bit 的 LSB**
- **利用序號利用產生32 bit 的 MSB**

如何產生種子編號

Secure Learn

☛ 一般按鍵被按下時所傳送的資料：

32-bit 跳碼資料 (Hopping)	28-bit 序 號	4-bit 按鍵	2-bit 狀態

☛ 四個按鍵 (S0, S1, S2 ,S3)同時被按下時，種子編號 將會取代跳碼的部份：

32-bit 種子編號 (SEED)	28-bit 序 號	4-bit 按鍵	2-bit 狀態

安全學習模式 vs 標準學習模式

Secure Learn

- 假設 製造商代碼 不慎洩露，侵入者可以：
  - ☆ 簡易學習模式：製造商代碼 = 解碼密碼
  - ☆ 標準學習模式：只要攔截到一筆發射資料，就可利用序號來產生 解碼密碼
- 有了解碼密碼 就可以解出跳碼，進而得到 同步計數值、識別碼 及按鍵值；再加上序號則可以燒錄到 HC300，成爲一複製的遙控器。
  - ♥ 安全學習模式：無法自一般傳送的資料中得到 種子編號 (SEED)的資料 (SEED碼可以是唯一且 不重複的亂數)

安全學習模式 Secure Learn

Secure Learn

- 安全學習模式 與 標準學習模式的解碼方式其實是一樣地，只是 解碼密碼 (Encryp -tion Key) 產生的方式不同。

KEY7	KEY6	KEY5	KEY4	KEY3	KEY2	KEY1	KEY0

MSB 解碼密碼 (64 bit Encryption Key) LSB

- 在安全學習模式中，解碼密碼 (Encryption Key) 是怎麼取得的？

答: 解碼密碼是必須利用製造商代碼 與 SEED 運算取得 32bit LSB , 再與 序號運算後再取得 32bit MSB後方 可得到這 64 bit 的解碼密碼。

接收到的跳碼 32 bits

Hopping Code

CSR3	CSR2	CSR1	CSR0

decrypt ()

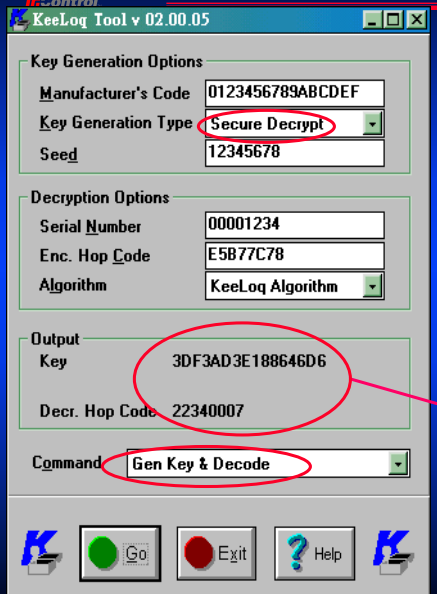
解碼後資料

CSR3	CSR2	CSR1	CSR0

功能鍵 + 溢 位 + 識別碼　識別碼　MSB 同步計數值　LSB

動手實驗 - 安全學習模式一

Secure Learn



假設編碼器的設定爲：

製造商代碼 = 0123456789ABCDEF

序號 = 00001234

S0 的按鍵被按下 (按鍵碼 = 2)

標準學習模式下傳送出去的資料：

序號 = 0000 1234

跳碼 = E5B7 7C78

SEED = 1234 5678

使用 "KeeLoq Tool" 練習解碼：

1. 解碼密碼 (Encryption Key) = _____
2. 同步計數值 (Sync. Counter Value) = _____
3. 識別碼 (Discrimination Value) = _____
4. 按鍵值 (Button) S2,S1,S0,S3 = _____

動手實驗 - 安全學習模式二

Secure Learn

假設另外一個編碼器的設定爲：

製造商代碼 = 0123456789ABCDEF

序號 = 00001234

S0 的按鍵被按下 (按鍵碼 = 2)

傳送出去的資料：

序號 = 00001234

跳碼 = B6110B11

改變 SEED 碼 = 01ABCDEF

使用 “KeeLoq Tool” 練習解碼：

1. 新的解碼密碼 (Encryption Key) = _____

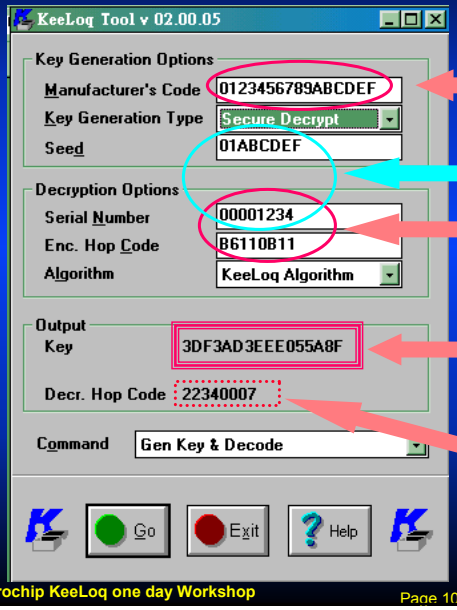
2. 同步計數值 (Sync. Counter Value) = _____

3. 識別碼 (Discrimination Value) = _____

4. 按鍵值 (Button Code) S2, S1, S0, S3 = _____

動手實驗 - 安全學習模式

Secure Learn



製造商代碼

含 SEED 的 66 bit 資料

一般所接收的 66 bit 資料

新的解碼密碼 (Encryption Key)
= 3DF3AD3E EE055A8F
注意：序號 或 SEED 任何一方改變
解碼密碼也改變

同步計數值 = 0007h
識別碼 = 234h
按鍵值 = 2h

產生解碼密碼的密訣 (Encryption Key)

Secure Learn

步驟 0：起動接收器進入學習模式後；
遙控器四鍵必須同時按下，
發送 SEED 碼及序號

製造商代碼 64 bit

01	23	45	67	89	AB	CD	EF
KEY7	KEY6	KEY5	KEY4	KEY3	KEY2	KEY1	KEY0

MSB LSB

SEED or 序號

CSR3	CSR2	CSR1	CSR0

decrypt ()

CSR3	CSR2	CSR1	CSR0

步驟 1：用 SEED 碼 ，產生 32 bit 的 LSB
步驟 2：用 0 + 序號 ，產生 32 bit 的 MSB

EE055A8F (LSB)

+

3DF3AD3E (MSB)

3D F3 AD 3E EE 05 5A 8F

新產生 64 bit 的解碼密碼

進入安全學習模式 (Secure Learn)

Secure Learn

已經知道解碼密碼後開始進入學習模式

- 等待接收一般的跳碼資料 (重新按遙控器鍵)。
- 檢查 新序號 (新收到的) 是否與 舊序號 相同。
- 利用新的 解碼密碼 與 跳碼 進行解碼後會得到下列數據，並判斷之：
  - 識別碼 (Discrimination Value) = 序號 ?
  - 同步計數值 (Sync. Counter Value)
- 儲存 解碼密碼，SEED，序號，識別碼，同步計數值到 EEPROM。

安全學習模式
解碼說明

Secure Learn

解 碼 器

學習模式　　：安全
製造商代碼　：0123456789ABCDEF

一般傳送資料
0 41234567 F73388E8

解碼密碼
C3C83DA8 A7190D1C

KEY7 ... KEY0
MSB　　LSB

要解碼資料

CSR3	CSR2	CSR1	CSR0
F7	33	88	E8

decrypt ()

解碼後資料

CSR3	CSR2	CSR1	CSR0
41	67	00	01

解碼器資料庫

種子編號 (32 bits)	AABBCCDD
序　號 (28 bits)	01234567
解碼密碼 (64 bits)	C3C83DA8　A7190D1C
識 別 碼 (10 bits)	167
同步計數器 (16 bits)	0001

步驟 4：利用解碼密碼與接收的一般跳碼資料解碼

課　程　結　束　！

如有疑問請洽：

02-2717-7175 #805

e-mail: taiwan.techhelp@microchip.com