

Keeloq 学习材料

一、Keeloq 的定义

Keeloq 实际上是一个“ASIC”的特别设计，内含加密及解密技术。

二、Keeloq 的安全性（以 HCS300 为例）

Keeloq 的编码含有一组 64-BIT “编码密码”

编码器共传送 66-BIT 的资料到解码器，其中有 32-BITS 的资料是一完全不可预测的跳码资料

Keeloq 的解码器知道这资料所用“编码密码”，所以可以用它来检验其接收的跳码资料是否正确

Keeloq 的传送资料是唯一，且不重履。。。。。

Keeloq 较长的编码技术可避免扫描机在短时间内就将相同的码传送出去：

即使以最快的方式来产生 HCS300 的编码(一秒约可传送 10 笔资料)，必须超过 3.7 年才可能产生全部的跳码部份(32 BIT)，若要产生全部 66 BIT 则需 2.2×10^{11} 年。

三、Keeloq 的应用领域

适用于遥控或命令辨别的应用场合，如安全锁、车库门遥控、秘密通讯、软件保护等。

四、Keeloq 编码技术原理介绍

3. 1 Keeloq 的核心

16-BIT 同步计数器，每当一按键被按下时，同步计数器会自动加一并存储在内部的 EEPROM。

Keeloq 演算法----一种非线性的推算公式，当输入数据进入这演算法时，其输出对输入而言是唯一的（不重覆）结果。（输入数据=16-BIT 同步计数值+10-BIT 识别码+2-BIT 溢出位+4-BIT 功能键值，输出数据=32-BIT 跳码）

3. 2 Keeloq 的核心组成元件

制造商代码（Manufacture's Code）

制造商/产品的辨别

由制造商自行决定此代码（不可泄露的原始密码）

遥控器的**制造商代码**必须与接收解码器相同

不同的制造商拥有不同的**制造商代码**

序号 (Serial Number)

每一编码 IC 或遥控器其**序号**均不相同

用来识别遥控器与接收器之间的关系

即使使用者同时有两支遥控器来控制同一接收器，其**序号**也不相同（但**制造商代码**必须相同）

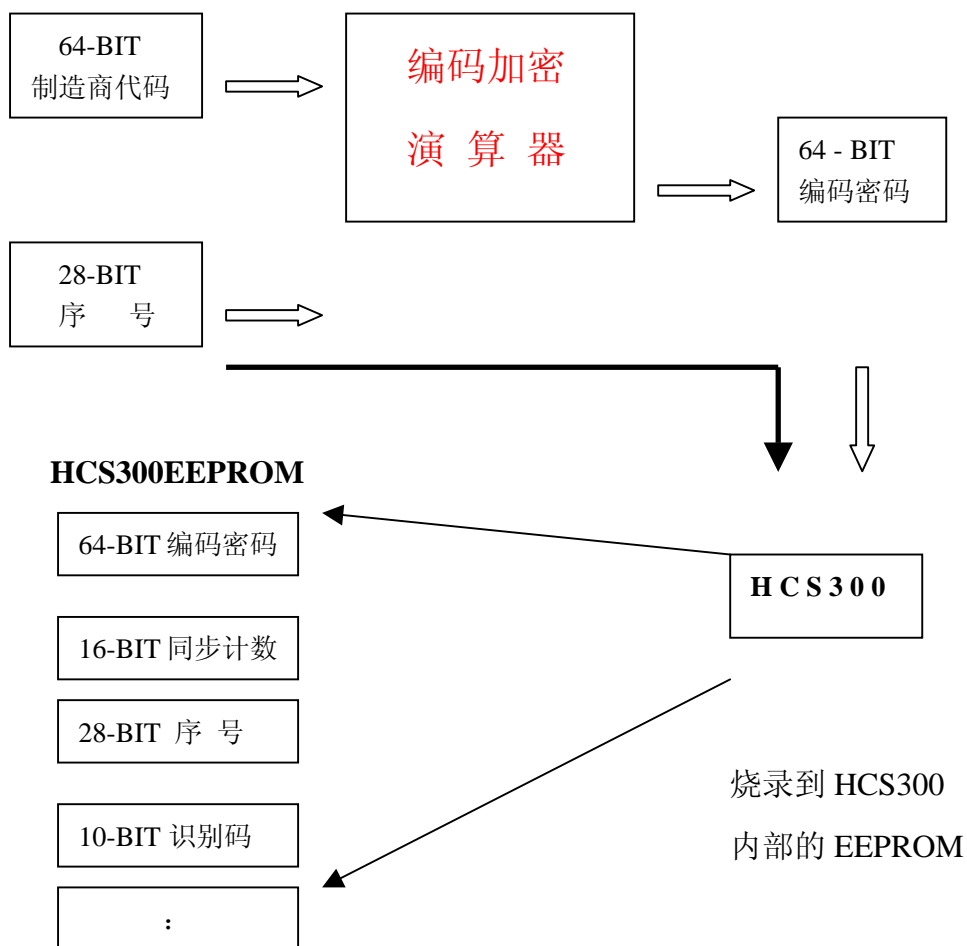
编码密码 (Encryption Key)

利用**制造商代码**及**序号**产生 **64-BIT** 的**编码密码**

这 **64-BIT** 的**编码密码**会被烧录在内部的 EEPROM

这 **64-BIT** 的**编码密码**是用来产生跳码的密码

3. 3 Keeloq 如何产生编码密码 (标准编码法)



简易编码法 (Simple Encode)

编码密码 = 制造商代码

编码密码不会随着序号改变

标准编码法 (Normal Encode)

编码密码 **不等于** 制造商代码

编码密码是由制造商代码及序号共同产生，任何一项改变，编码密码也会改变

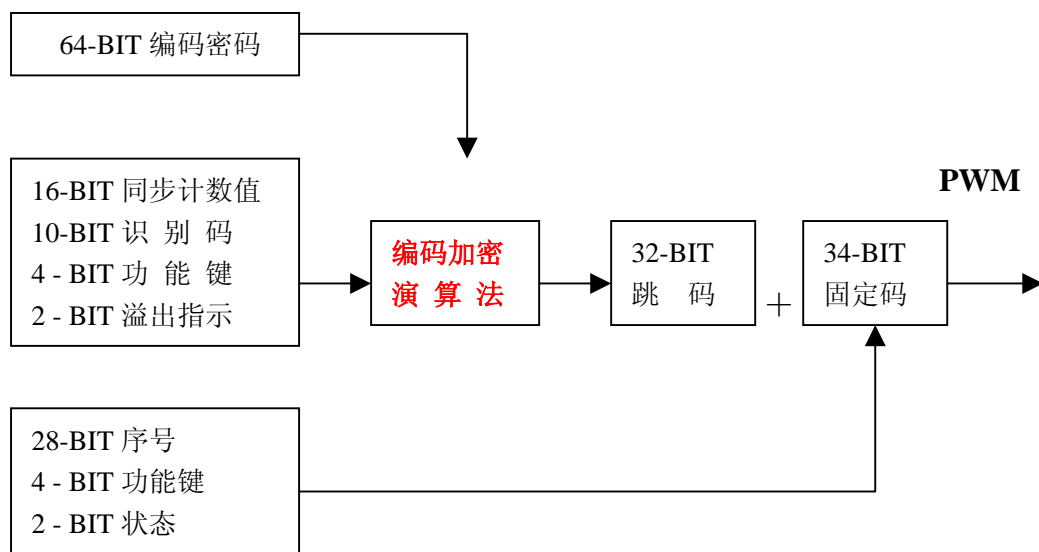
安全编码法 (Secure Encode)

编码密码 **不等于** 制造商代码

编码密码是由制造商代码及序号和种子码共同产生，任何一项改变，编码密码也会改变

五、HCS300 传送编码方式说明

每当遥控器的按键被按下时，HCS300 即将下列的资料传送出去



六、Keeloq 的解密

6. 1 Keeloq 解密方式

硬件解密（如使用 HCS500 等）操作简易，缺点增加成本

软件解密编程较复杂，以下重点讲解

6. 2 Keeloq 的学习模式

1. Keeloq 系统为何要学习？

因为生产配对方便，管理简单

因为解码器一开始时，除了制造商代码外，什么都不知道！

接收解码器须要众多的解码资讯储存在 EEPROM 中，而这些资讯的提供者是遥控发射器（编码器）：

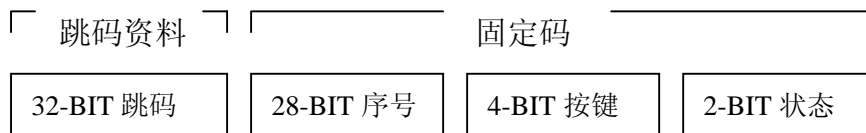
- ▲ 序号（Serial Number）
- ▲ 同步计数值（Current Sync.Counter Value）
- ▲ 识别码（Discrimination Value）
- ▲ 编码密码（Encryption Key）

2. Keeloq 的学习步骤就是：

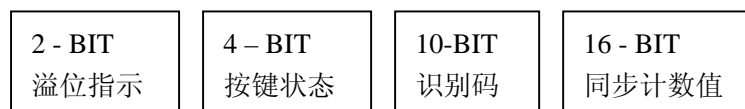
- 接收跳码资料
- 进行资料解码
- 核对解码后资料
- 储存学习资料

3. 遥控发射器 HCS300 所提供的资讯

- 序号（Serial Number）
 - ▲ 28-BIT 的序号是放在固定码中
 - ▲ 可自接收资料中直接取得
- 同步计数值（Current Sync.Counter Value）
 - ▲ 16-BIT 的长度，隐藏在跳码中
 - ▲ 无法直接得到，须透过解码后才可取得
- 识别码（Discrimination Value）
 - ▲ 10-BIT 的长度，隐藏在跳码中
 - ▲ 无法直接得到，须透过解码后才可取得
 - ▲ 如不特别指定，识别码的值 = 序号中较低的 10 BIT



解 码 后



4. Keeloq 基本解码元件

原料一：制造商代码

▲ 总长有 64-BIT---对制造商的产品而言唯一的，一般固化在程序代码中

原料二：

▲ 简易学习模式---不需要

▲ 标准学习模式---序号

▲ 安全学习模式---种子编号+序号

解码程序：decrypt()

▲ 解码演算法则或 XOR 运算法则

▲ 将原料一及原料二的资料加以解码

▲ 此解码程序由 Microchip 提供

5. 简易学习模式

透过一次的学习，取得下列资料：

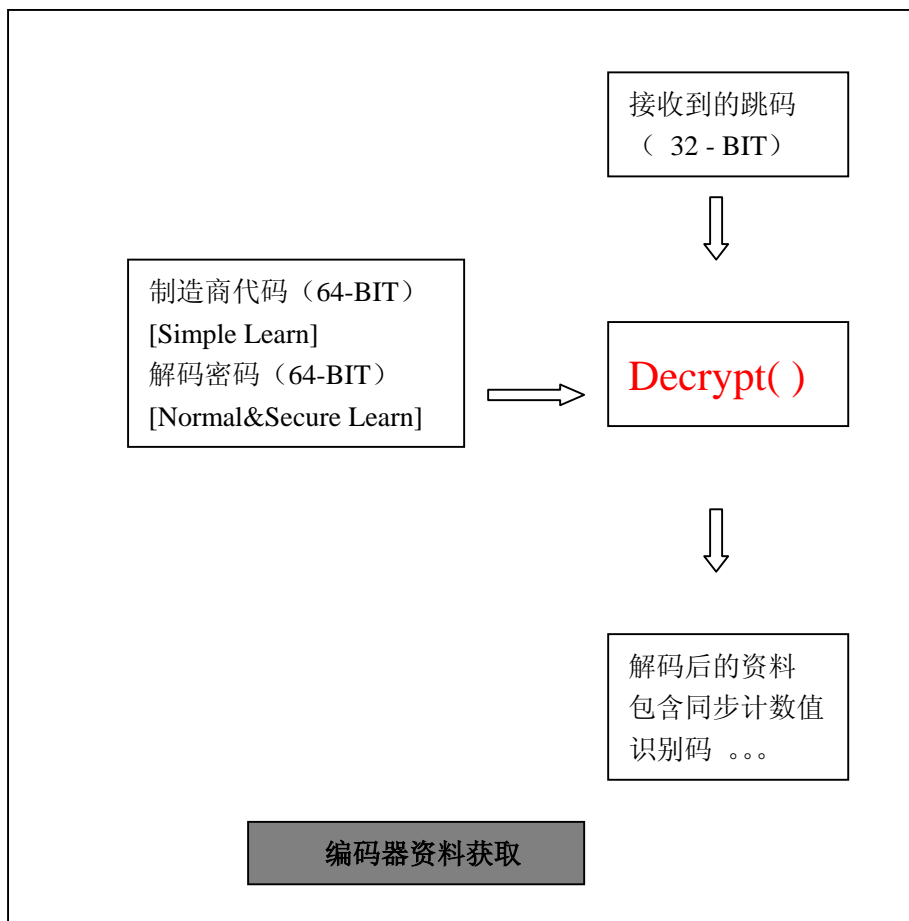
▲ 序号 (Serial Number)

▲ 识别码 (Discrimination Value)

▲ 同步计数值 (Sync.Counter Value)

在简易学习模式，解码密码是怎么取得？

答案：制造商代码就等于解码密码。



6. 标准学习模式

第一次的学习，取得以下资料：

- ▲ 解码密码 (Encryption Key)
- ▲ 序号 (Serial Number)
- ▲ 识别码 (Discrimination Value)
- ▲ 同步计数值 (Sync.Counter Value)

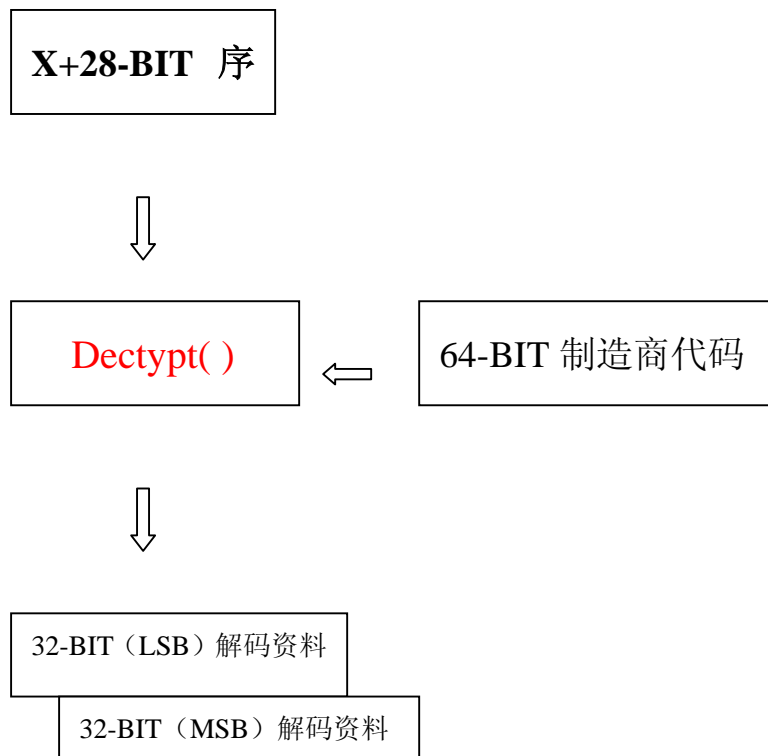
第二次的学习，检查同步计数值后储存学习结果到 EEPROM

★ 标准学习模式与简易学习模式的解码方式其实的一样地，只是使用不同的解码密码。

产生解码密码的秘诀 (Normal Learn)

步骤 1: 设定 X=2, 产生 32-BIT 的 LSB

步骤 2: 设定 X=6, 产生 32-BIT 的 MSB



合起来共 64 BIT 的解码密码

7. 安全学习模式

如何产生种子编号？

四个按键（S0，S1，S2，S3）同时被按下时，种子编号将会取代跳码的部分。

★ 安全学习模式与标准学习模式的解码方式其实的一样地，只是解码密码的产生方式不同。

产生解码密码的秘诀（Normal Learn）

步骤 0: 起动接收器进入学习模式后，遥控器四键必须同时按下，发送 SEED 码及序号。

步骤 1: 用 SEED 码，产生 32-BIT 的 LSB

步骤 2: 用 0+序号，产生 32-BIT 的 MSB

已经知道解码密码后开始进入安全学习模式

▲ 等待接收一般的跳码资料（重新按遥控器键）。

▲ 检查新收到的序号是否与旧序号相同。

▲ 利用新的解码密码与跳码进行解码后会得到下列数据，判断之：

-----第 7 页

识别码 = 序号?

同步计数值

▲ 储存解码密码, SEED, 序号, 识别码, 同步计数值到 EEPROM。

6. 3 Keeloq 的解码确认

接收到一有效的 Keeloq 资料。

检查接收资料的固定码部份是否与资料库中的序号相同。

自资料库中取出 64-BIT 的“解码密码”。

将接收到的资料加以解码产生四种资料:

功能键、溢位、识别码、同步计数值

检验 10-BIT 的“识别码:

识别码的值(默认)与序号的低 10-BIT 相等

比较固定码中的“功能键”值与解码后的“功能键”值是否相等

检查“同步计数值”的变化是否正确。

七、开发 Keeloq 所需的资源

1. Keeloq 编芯片烧写套件 Keeloq_kit
2. Keeloq 解码软件