BC45F0023/BC68F3132 Hopping Code Engine Application Note and Example

D/N: AN0626EN

Introduction

In anti-theft remote control applications such as car alarms, motorcycle alarms and electric rolling door remote controls, manufacturers usually encrypt the remote control output data in order to ensure that the remote control is not easily copied. Traditional remote controls use fixed codes for their control method, however this makes it not too difficult to replicate the code using a few simple copy devices. This offers no user safety guarantee.

To counteract this, Holtek's BC45F0023 and BC68F3132 devices contain an EEPROM and a hopping code engine, which can be used to easily generate encrypted control data. The sync count value in the hopping code can be stored in the EEPROM and the synchronisation state can be maintained even after battery changes. This application note will take the BC45F0023 hopping code encryption/decryption device as an example to illustrate how to generate hopping codes using the hopping code engine with the synchronisation process. Users can also refer to this example to design their own hopping code remote controls.

Operating Principle

A rolling code, also known as a hopping code, means that in situations where the input data is the same, the output results are different for different triggers. To achieve this function, a variable is required to be added into the encryption process, which changes each time it is triggered. A common hopping code has two encryption types, Simple Learn and Normal Learn. Refer to Figure 1 to find out more about their differences.

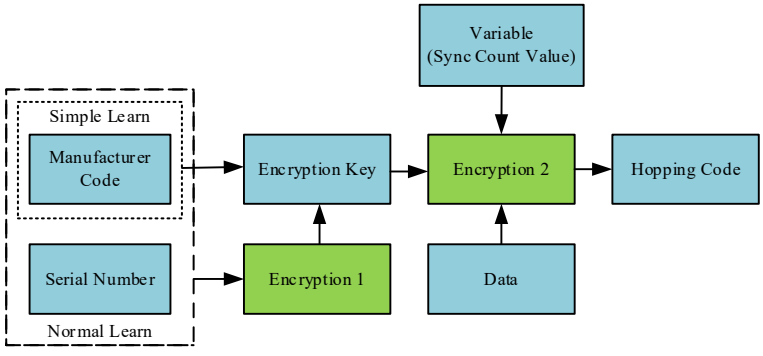

Figure 1

An anti-theft remote control usually adds two pieces of information, one is the manufacturer code, which usually represents the product company. For example, anti-theft remote controls made by company A and company B can be distinguished using this code. The second is the Serial Number – SN, which mostly indicates different remote controls. For example, there can be two rolling door remote controls with different serial numbers, but both of them can control the same rolling door. In this case the rolling door will be the receiver and must recognise the remote controls with these two serial numbers as valid remote controls.

During the Simple Learn process, the manufacturer code is directly used as an encryption key. During the Normal Learn process, the result is used as a key after calculating the code using the Serial Number. The Manufacturer Code is a main key component during encryption, therefore it is very important and needs to be managed carefully.

Variable

The sync count value is a key variable for generating the hopping code. It will increment every time a hopping code is generated. When the receiver is not nearby, multiple bytes of transmitted data will not necessarily be received. This will then cause the receiver to be out of sync. Therefore, it is necessary to set a range of received values as "valid values" during decoding. If the received value range is set to 16, the sync count value can be the previous received value plus 1~16. All values in this range can be considered as valid values. In addition to valid values, another range value can be defined as the reconfirmed value. If there is a value that is required to be reconfirmed, it will reduce most of the out-of-sync situations due to abnormal power failure, battery replacement or being out of synch for too long. The following figure shows how the 16-bit sync count value changes.

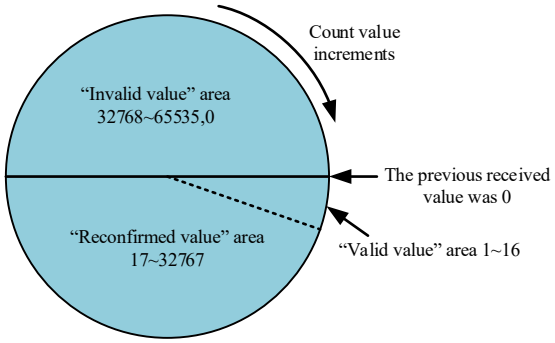

Figure 2

Encryption

Encryption is an important process during the hopping code generation and uses an encryption method known as "Nonlinear Feedback". This method is simple in structure, reliable in operation and also can generate a nonlinear sequence. This makes it impossible for anyone to obtain the encryption formula from the output data.

The BC45F0023 hopping code encryption/decryption device provides an integrated hopping code engine, which is a nonlinear-feedback system. It contains a nonlinear-feedback shift register - NLFSR, a nonlinear function register - NLF, a loop counter register - LOOPCNT and a hopping code engine

key register - HPKEY. The user simply writes the corresponding value into these registers. After the engine has started, when the feedback counter has finished counting, a set of nonlinear values can be obtained, together with the sync count value. This is the hopping code system. Refer to the corresponding MCU datasheet for the detailed operation process. The following sections will introduce the hopping code generated process in the hopping code engine.

Example Program Description

Users have different requirements for encryption complexity. The following is a conceptual architecture and users can design the protocol format according to their requirements. The following section will describe the programming process for both the Simple Learn and Normal Learn methods. The example program provided in this application note should be used with the HT-IDE3000 simulation development tool to open the project.

Simple Learn Process

Simple Learn: Use a 64-bit manufacturer code as the encryption key. A 32-bit input data packet consists of a 16-bit sync count code, a 12-bit serial number and a 4-bit control command. Then select a 5-bit variable nonlinear function and aggregate its operation result into a 32-bit value for use by the algorithm. Finally, the feedback times have to be determined. The higher the feedback time value, the more time will be required by the encryption algorithm. The following figure shows the parameter relationships for the Simple Learn.

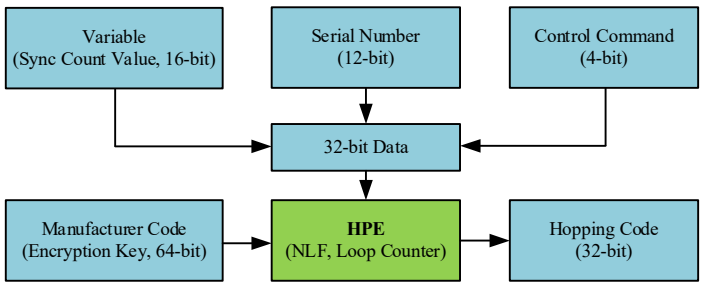

Figure 3

The follows describes how to use the hopping code engine. Write the feedback times into the LOOPCNT register, the manufacturer code into the HPKEY register and the nonlinear function into the NLF register. Write the sync count code, the serial number and the control command into the NLFSR register. When the ENCDEC bit is cleared to 0 and then the HPEEN bit is set to 1, the engine will then start to encrypt. When the HPEEN bit is automatically cleared to 0, the value stored in the NLFSR register will now be the required hopping code. The complete process, including the key trigger action, is shown as follows.

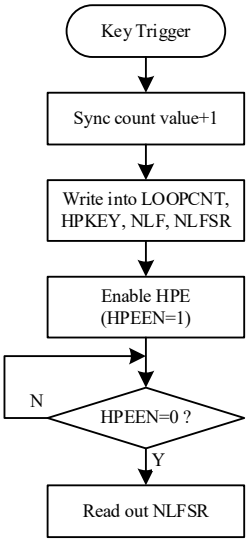

Figure 4

Simple Decryption Process

Simple Decryption: Write the received hopping codes into the NLFSR register and set the ENCDEC bit to 1. The other registers must be the same as the written values during encryption. When the HPEEN bit is set to 1, the engine will start to decrypt. When the HPEEN bit is automatically cleared to 0, the value in the NLFSR register will be the original data. Then respectively read out the sync count code, serial number and control command according to the corresponding position when encrypted. Combined with the receive execution commands, the complete process is shown in the following figure.

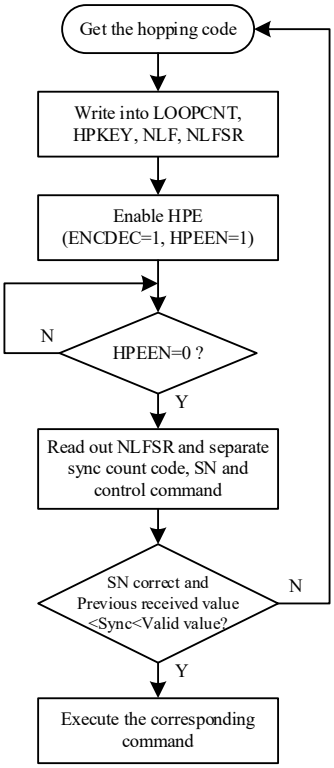

Figure 5

Normal Learn Process

Normal Learn: The manufacturer code and serial number are used to execute the decryption algorithm with its result being used as the encryption key. First, add 0x60000000 to the 28-bit serial number to obtain a 32-bit value. This is combined with the 64-bit manufacturer code and the decoding algorithm is then used to calculate the highest 32-bits of the key. In a similar way, add 0x20000000 to the 28-bit serial number and using the 64-bit manufacturer code and the decoding algorithm, the lower 32-bits of the key is calculated. In this way, a complete 64-bit value is obtained as the key. After that, encrypt and calculate using the Simple Learn, to generate the rolling code.

The following figure shows the parameter relationships during the Simple Learn.

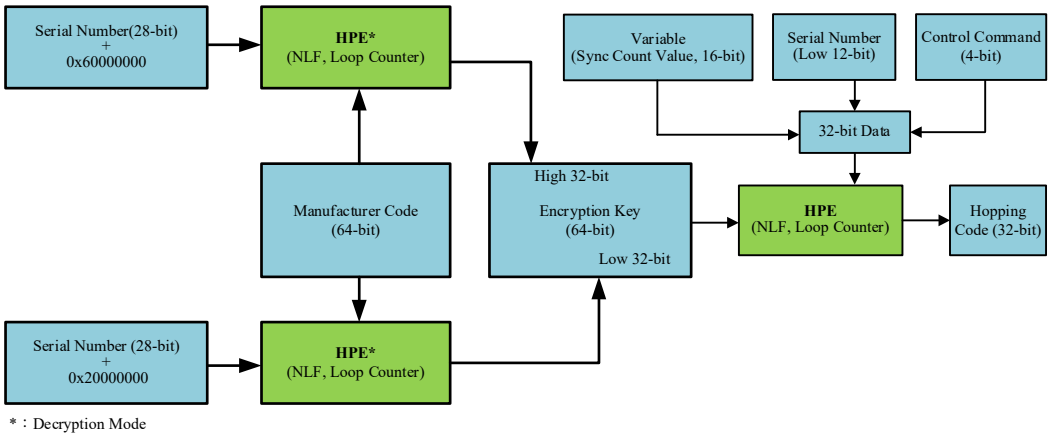

Figure 6

Combined with the key trigger action, first calculate the key value, namely New HPKEY using the manufacturer code and the serial number, and then store it. When a trigger event occurs, the Simple Learn process will be executed. The complete process is shown in the following figure.

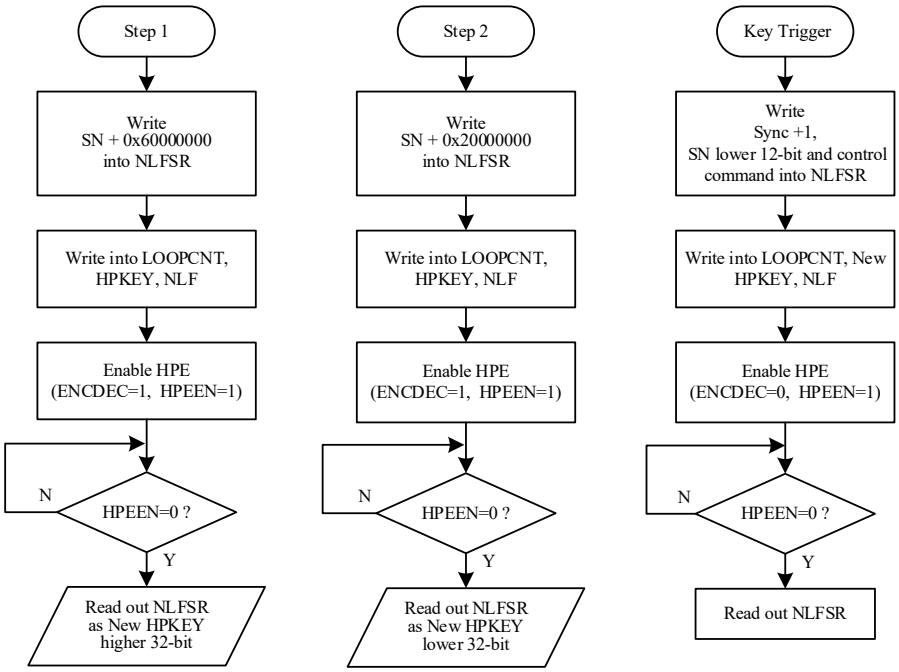

Figure 7

Normal Decryption Process

The normal decryption process is consistent with the simple decryption process, except that the decryption key value must also be the same as the calculated key value during the Normal Learn. The complete process is shown in the following figure.

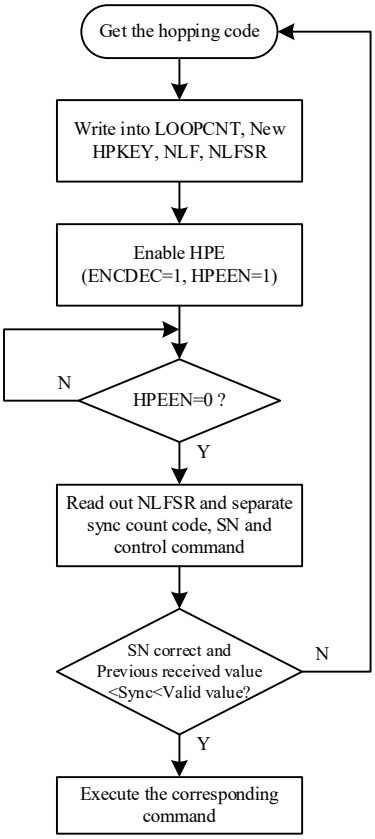

Figure 8

Conclusion

This application note has taken the BC45F0023 as an example to introduce how to use the hopping code engine to generate hopping codes. Users can obtain simple example programs for this from the Holtek official website.

Reference File

Reference File: BC45F0023, BC68F3132 Datasheet.

For more information, consult the Holtek official website: www.holtek.com.

Revision and Modification Information

Date	Author	Issue	Modification Information
2022.08.25	何信智	V1.00	First Version

Disclaimer

All information, trademarks, logos, graphics, videos, audio clips, links and other items appearing on this website ('Information') are for reference only and is subject to change at any time without prior notice and at the discretion of Holtek Semiconductor Inc. and its related companies (hereinafter 'Holtek', 'the company', 'us', 'we' or 'our'). Whilst Holtek endeavors to ensure the accuracy of the Information on this website, no express or implied warranty is given by Holtek to the accuracy of the Information. Holtek shall bear no responsibility for any incorrectness or leakage.

Holtek shall not be liable for any damages (including but not limited to computer virus, system problems or data loss) whatsoever arising in using or in connection with the use of this website by any party. There may be links in this area, which allow you to visit the websites of other companies. These websites are not controlled by Holtek. Holtek will bear no responsibility and no guarantee to whatsoever Information displayed at such sites. Hyperlinks to other websites are at your own risk.

Limitation of Liability

In no event shall Holtek Limited be liable to any other party for any loss or damage whatsoever or howsoever caused directly or indirectly in connection with your access to or use of this website, the content thereon or any goods, materials or services.

Governing Law

The Disclaimer contained in the website shall be governed by and interpreted in accordance with the laws of the Republic of China. Users will submit to the non-exclusive jurisdiction of the Republic of China courts.

Update of Disclaimer

Holtek reserves the right to update the Disclaimer at any time with or without prior notice, all changes are effective immediately upon posting to the website.