Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

HCS410

KEELOQ® Code Hopping Encoder and Transponder

FEATURES

Security

- Two programmable 64-bit encoder keys
- 16/32-bit bi-directional challenge and response using one of two keys
- 69-bit transmission length
- 32-bit unidirectional code hopping, 37-bit non-encrypted portion
- Encoder keys are read protected
- Programmable 28/32-bit serial number
- 60/64-bit, read-protected seed for secure learning
- Three IFF encryption algorithms
- Delayed increment mechanism
- Asynchronous transponder communication
- Queuing information transmitted

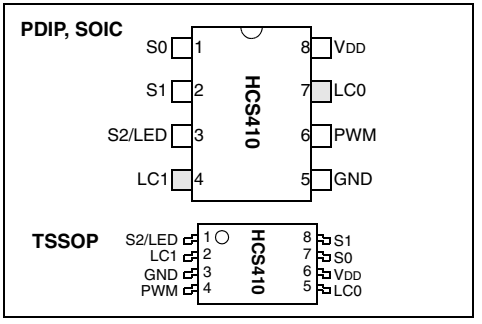
Operating

- 2.0V - 6.6V operation, 13V encoder only operation
- Three switch inputs [S2, S1, S0]—seven functions
- Batteryless bi-directional transponder
- Selectable baud rate and code word blanking
- Automatic code word completion
- Battery low signal transmitted
- Non-volatile synchronization
- PWM or Manchester RF encoding
- Combined transmitter, transponder operation
- Anti-collision of multiple transponders
- Passive proximity activation
- Device protected against reverse battery
- Intelligent damping for high Q LC-circuits

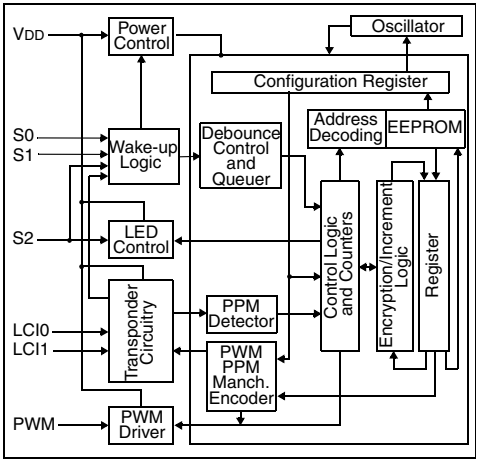
Other

- 37-bit nonencrypted part contains 28/32-bit serial number, 4/0-bit function code, 1-bit battery low, 2-bit CRC, 2-bit queue
- Simple programming interface
- On-chip tunable RC oscillator (±10%)
- On-chip EEPROM
- 64-bit user EEPROM in transponder mode
- Battery-low LED indication
- SQTP serialization quick-time programming
- 8-pin PDIP/SOIC/TSSOP and die

PACKAGE TYPES



BLOCK DIAGRAM



Typical Applications

- Automotive remote entry systems
- Automotive alarm systems
- Automotive immobilizers
- Gate and garage openers
- Electronic door locks (Home/Office/Hotel)
- Burglar alarm systems
- Proximity access control

DESCRIPTION

The HCS410 is a code hopping transponder device designed for secure entry systems. The HCS410 utilizes the patented KEELOQ® code hopping system and bi-directional challenge-and-response for logical and physical access control. High security learning mechanisms make this a turnkey solution when used with the KEELOQ decoders. The encoder keys and synchronization information are stored in protected on-chip EEPROM.

A low cost batteryless transponder can be implemented with the addition of an inductor and two capacitors. A packaged module including the inductor and capacitor will also be offered.

A single HCS410 can be used as an encoder for Remote Keyless Entry (RKE) and a transponder for immobilization in the same circuit and thereby dramatically reducing the cost of hybrid transmitter/transponder circuits.

1.0 SYSTEM OVERVIEW

1.1 Key Terms

- Anti-Collision – Allows two transponders to be in the files simultaneously and be verified individually.
- CH Mode – Code Hopping Mode. The HCS410 transmits a 69-bit transmission each time it is activated, with at least 32-bits changing each time the encoder is activated.
- Encoder Key – A unique 64-bit key generated and programmed into the encoder during the manufacturing process. The encoder key controls the encryption algorithm and is stored in EEPROM on the encoder device.
- IFF – Identify friend or foe is a means of validating a token. A decoder sends a random challenge to the token and checks that the response of the token is a valid response.
- KEELOQ Encryption Algorithm – The high security level of the HCS410 is based on the patented KEELOQ technology. A block cipher encryption algorithm based on a block length of 32 bits and a key length of 64 bits is used. The algorithm obscures the information in such a way that even if the unencrypted/challenge information differs by only one bit from the information in the previous transmission/challenge, the next coded transmission/response will be totally different. Statistically, if only one bit in the 32-bit string of information changes, approximately 50 percent of the coded transmission will change.
- Learn – The HCS product family facilitates several learning strategies to be implemented on the decoder. The following are examples of what can be done.

  Normal Learn –The receiver uses the same information that is transmitted during normal operation to derive the transmitter's encoder key, decrypt the discrimination value and the synchronization counter.

  Secure Learn* – The transmitter is activated through a special button combination to transmit a stored 60-bit value (random seed) that can be used for key generation or be part of the key. Transmission of the random seed can be disabled after learning is completed.
- Manufacturer's Code – A 64-bit word, unique to each manufacturer, used to produce a unique encoder key in each transmitter (encoder).
- Passive Proximity Activation – When the HCS410 is brought into in a magnetic field without a command given by the base station, the HCS410 can be programmed to give an RF transmission.
- Transport Code – A 32-bit transport code needs to be given before the HCS410 can be inductively programmed. This prevents accidental programming of the HCS410.

1.2 KEELOQ Code Hopping Encoders

When the HCS410 is used as a code hopping encoder device, it is ideally suited to keyless entry systems, primarily for vehicles and home garage door openers. It is meant to be a cost-effective, yet secure solution to such systems. The encoder portion of a keyless entry system is meant to be carried by the user and operated to gain access to a vehicle or restricted area.

Most keyless entry systems transmit the same code from a transmitter every time a button is pushed. The relative number of code combinations for a low end system is also a relatively small number. These shortcomings provide the means for a sophisticated thief to create a device that 'grabs' a transmission and retransmits it later or a device that scans all possible combinations until the correct one is found.

The HCS410 employs the KEELOQ code hopping technology and an encryption algorithm to achieve a high level of security. Code hopping is a method by which the code transmitted from the transmitter to the receiver is different every time a button is pushed. This method, coupled with a transmission length of 69 bits, virtually eliminates the use of code 'grabbing' or code 'scanning'.

The HCS410 has a small EEPROM array which must be loaded with several parameters before use. The most important of these values are:

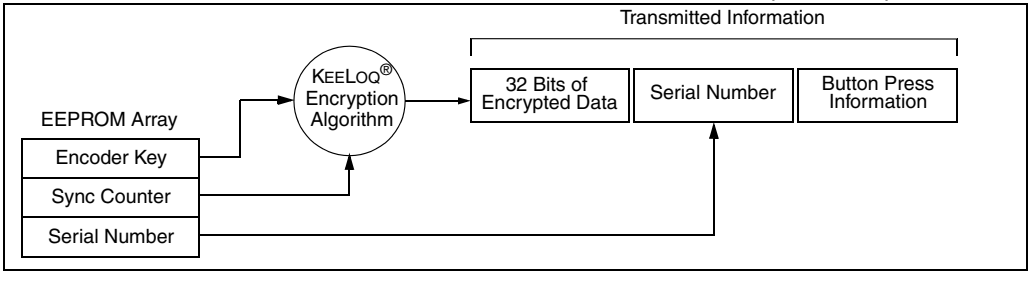
- A 28/32-bit serial number which is meant to be unique for every encoder
- 64-bit seed value
- A 64-bit encoder key that is generated at the time of production
- A 16-bit synchronization counter value.
- Configuration options

The 16-bit synchronization counter value is the basis for the transmitted code changing for each transmission, and is updated each time a button is pressed. Because of the complexity of the code hopping encryption algorithm, a change in one bit of the synchronization counter value will result in a large change in the actual transmitted code.

Once the encoder detects that a button has been pressed, the encoder reads the button and updates the synchronization counter. The synchronization counter value, the function bits, and the discrimination value are then combined with the encoder key in the encryption algorithm, and the output is 32 bits of encrypted information (Figure 1-1). The code hopping portion provides up to four billion changing code combinations. This data will change with every button press, hence, it is referred to as the code hopping portion of the code word.

The 32-bit code hopping portion is combined with the button information and the serial number to form the code word transmitted to the receiver. The code word format is explained in detail in Section 2.2.

FIGURE 1-1: BASIC OPERATION OF A CODE HOPPING TRANSMITTER (ENCODER)



1.3 KEELOQ IFF

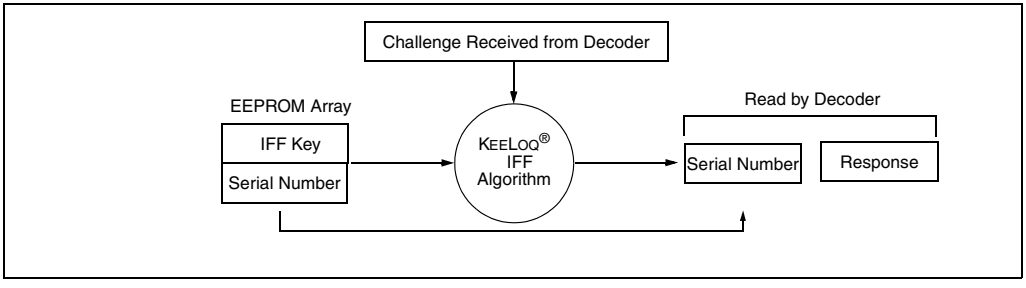
The HCS410 can be used as an IFF transponder for verification of a token. In IFF mode the HCS410 is ideally suited for authentication of a key before disarming a vehicle immobilizer. Once the key has been inserted in the car's ignition the decoder would inductively poll the key validating it before disarming the immobilizer.

IFF validation of the token involves a random challenge being sent by a decoder to a token. The token then generates a response to the challenge and sends this response to the decoder (Figure 1-2). The decoder calculates an expected response using the same challenge. The expected response is compared to the response received from the token. If the responses match, the token is identified as a valid token and the decoder can take appropriate action.

The HCS410 can do either 16 or 32-bit IFF. The HCS410 has two encryption algorithms that can be used to generate a response to a challenge. In addition there are up to two encoder keys that can be used by the HCS410. Typically each HCS410 will be programmed with a unique encoder key(s).

In IFF mode, the HCS410 will wait for a command from the base station and respond to the command. The command can either request a read/write from user EEPROM or an IFF challenge response. A given 16 or 32-bit challenge will produce a unique 16/32-bit response, based on the IFF key and IFF algorithm used.

FIGURE 1-2: BASIC OPERATION OF AN IFF TOKEN



2.0 DEVICE OPERATION

The HCS410 can either operate as a normal code hopping transmitter with one or two IFF keys (Figure 2-1) or as purely an IFF token with two IFF keys (Figure 2-2 and Figure 2-3). When used as a code hopping transmitter the HCS410 only needs the addition of buttons and RF circuitry for use as a transmitter. Adding the transponder function to the transmitter requires the addition of an inductor and two capacitors as shown in Figure 2-1 and Figure 2-2. A description of each pin is given in Table 2-1. Table 2-2 shows the function codes for using the HCS410.

FIGURE 2-1: COMBINED TRANSMITTER/ TRANSPONDER CIRCUIT

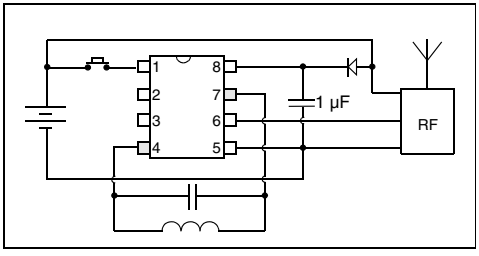


FIGURE 2-2: TRANSPONDER CIRCUIT

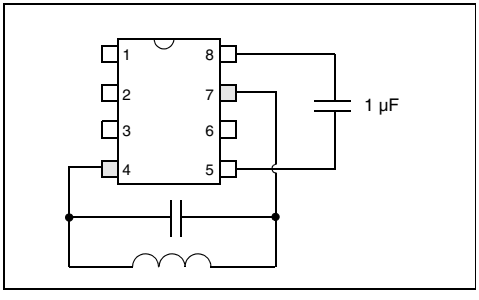


FIGURE 2-3: 2-WIRE, 1 OR 2-KEY IFF TOKEN

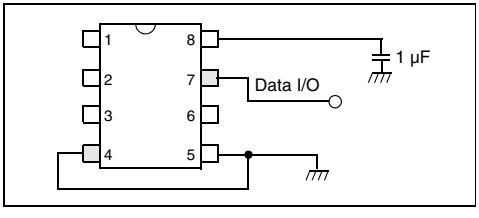


Figure 2-4 shows how to use the HCS410 with a 12V battery as a code hopping transmitter. The circuit uses the internal regulator, normally used for charging a capacitor/battery in LC mode, to generate a 6V supply for the HCS410.

FIGURE 2-4: HCS410 ENCODER WITH 12V BATTERY

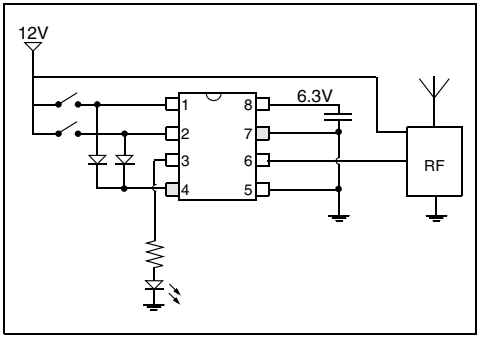


FIGURE 2-5: LED CONNECTION TO S2/LED OUTPUT

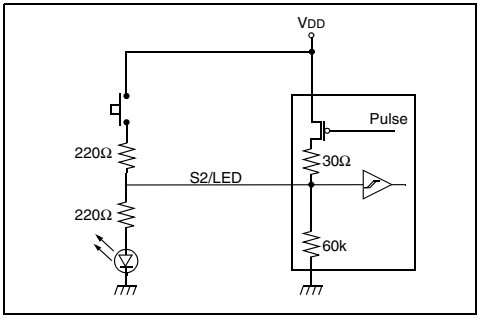
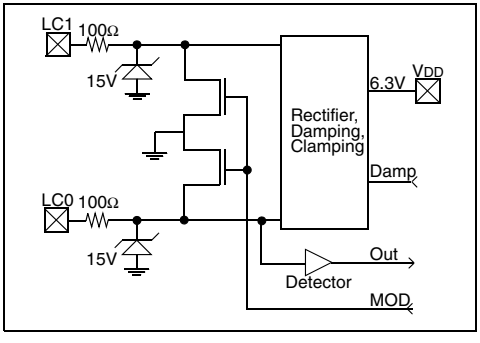


FIGURE 2-6: LC PIN BLOCK DIAGRAM



2.1 Pinout Description

The HCS410 has the same footprint as all of the other devices in the KEELOQ family, except for the two pins that are reserved for transponder operations and the LED that is now located at the same position as the S2 switch input.

- S[0:1] – are inputs with Schmitt Trigger detectors and an internal 60k¾ (nominal) pull-down resistors.
- S2/LED – uses the same input detection circuit as S0/S1 but with an added PMOS transistor connected to VDD capable of sourcing enough current to drive an LED.
- LC[0:1] – is the transponder interface pins to be connected to an LC circuit for inductive communication. LC0 is connected to a detector for data input. Data output is achieved by clamping LC0 and LC1 to GND through two NMOS transistors. These pins are also connected to a rectifier and a regulator, providing power to the rest of the logic and for charging an external power source (Battery/Capacitor) through VDD.

  The input impedance of the LC pins is a function of input voltage. At low voltages, the input impedance is in the order of mega-ohms. **When laying out a PC board, care should be taken to ensure that there is no cross coupling between the LC pins and other traces on the board.** Glitches on the LC lines will cause the device to reset. A high-value resistor (220 KW) between LC0 and GND can be added to reduce sensitivity.

TABLE 2-1: PINOUT DESCRIPTION

Name	Pin Number	Description
S0	1	Switch input 0
S1	2	Switch input 1
S2/LED	3	Switch input 2/LED output, Clock pin for programming mode
LC1	4	Transponder interface pin
VSS	5	Ground reference connection
PWM	6	Pulse width modulation (PWM) output pin/Data pin for programming mode
LC0	7	Transponder interface pin
VDD	8	Positive supply voltage connection

TABLE 2-2: FUNCTION CODES

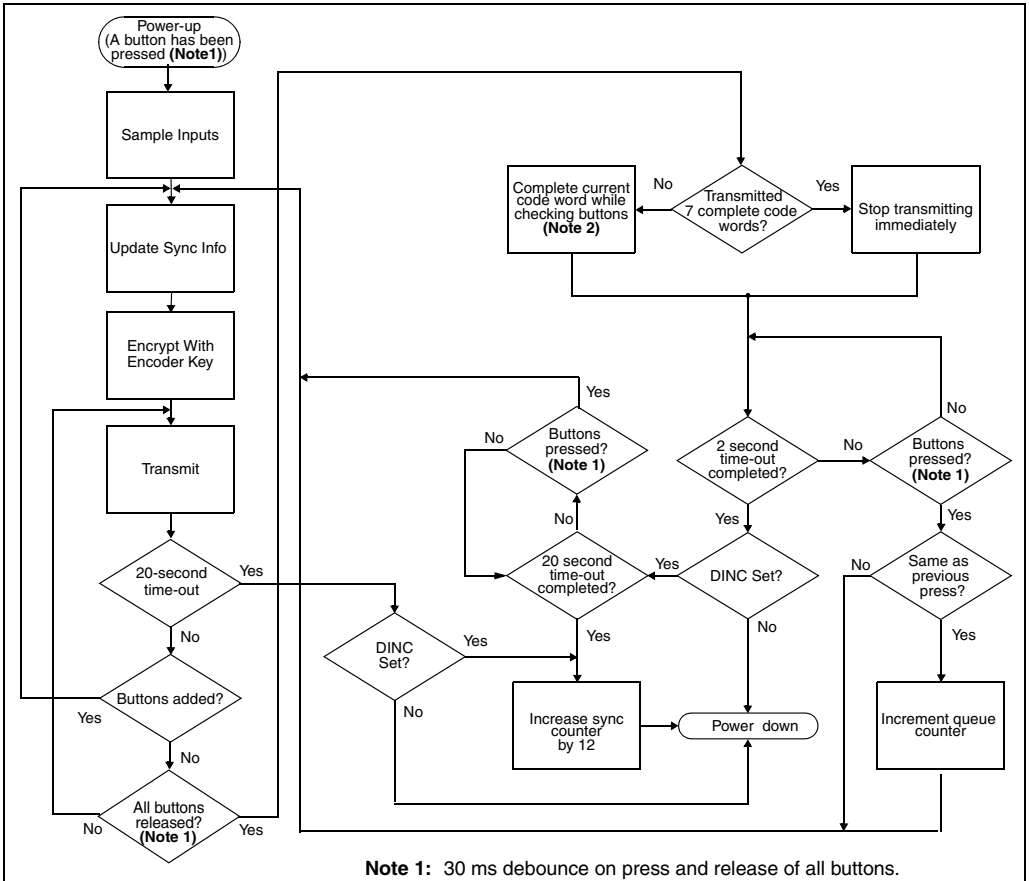
	LC0	S2	S1	S0	Comments
1	0	0	0	1	Normal Code Hopping transmission
2	0	0	1	0	Normal Code Hopping transmission
3	0	0	1	1	Delayed seed transmission if allowed by SEED and TMPSD/Normal Code Hopping transmission
4	0	1	0	0	Normal Code Hopping transmission
5	0	1	0	1	Normal Code Hopping transmission
6	0	1	1	0	Normal Code Hopping transmission
7	0	1	1	1	Immediate seed transmission if allowed by SEED and TMPSD/Normal Code Hopping transmission
8	1	0	0	0	Transponder mode

2.2 Code Hopping Mode (CH Mode)

The HCS410 wakes up upon detecting a switch closure and then delays approximately 30 ms for switch debounce (Figure 2-7). The synchronization counter value, fixed information, and switch information are encrypted to form the code hopping portion. The encrypted or code hopping portion of the transmission changes every time a button is pressed, even if the same button is pushed again. Keeping a button pressed for a long time results in the same code word being transmitted until the button is released or time-out occurs. A code that has been transmitted will not occur again for more than 64K transmissions. Overflow information programmed into the encoder can be used by the decoder to extend the number of unique transmissions to more than 192K.

If, during the transmit process, it is detected that a new button(s) has been added, a reset will immediately be forced and the code word will not be completed. Please note that buttons removed will not have any effect on the code word unless no buttons remain pressed in which case the current code word will be completed and the power down will occur. If, after a button combination is pressed, and the same button combination is pressed again within 2 seconds of the first press, the current transmission will be aborted and a new trans-

FIGURE 2-7: CODE HOPPING ENCODER OPERATION



Note 1: 30 ms debounce on press and release of all buttons.
2: Completes a minimum of 3 code words if MTX3 is set.

2.2.1 TRANSMISSION DATA FORMAT

The HCS410 transmission (CH Mode) is made up of several parts (Figure 2-10 and Figure 2-11). Each transmission is begun with a preamble and a header, followed by the encrypted and then the fixed data. The actual data is 69 bits which consists of 32 bits of encrypted data and 37 bits of fixed data. Each transmission is followed by a guard period before another transmission can begin. Refer to Table 6-4 and Table 6-5 for transmission timing specifications. The combined encrypted and nonencrypted sections increase the number of combinations to 1.47 x 10^{20}.

The HCS410 transmits a 69-bit code word when a button is pressed. The 69-bit word is constructed from a Fixed Code portion and Code Hopping portion (Figure 2-8).

The **Encrypted Data** is generated from 4 function bits, 2 overflow bits, and 10 discrimination bits, and the 16-bit synchronization counter value (Figure 2-8).

The **Nonencrypted Code Data** is made up of 2 QUE bits, 2 CRC bits, a VLOW bit, 4 function bits, and the 28-bit serial number. If the extended serial number (32 bits) is selected, the 4 function code bits will not be transmitted (Figure 2-8).

FIGURE 2-8: HOP CODE WORD ORGANIZATION (RIGHT-MOST BIT IS CLOCKED OUT FIRST)

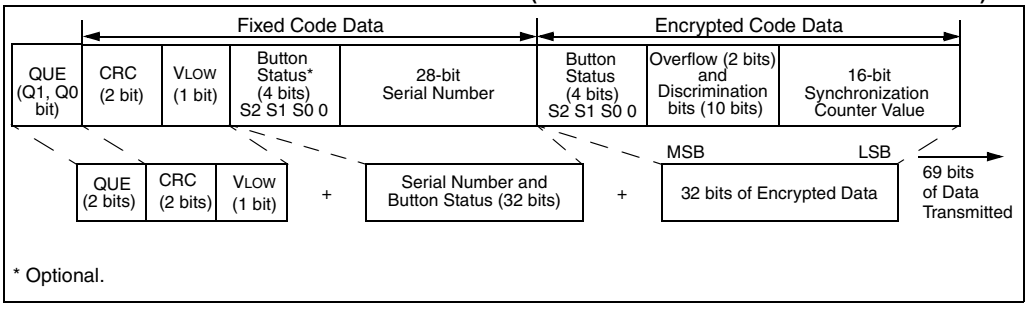
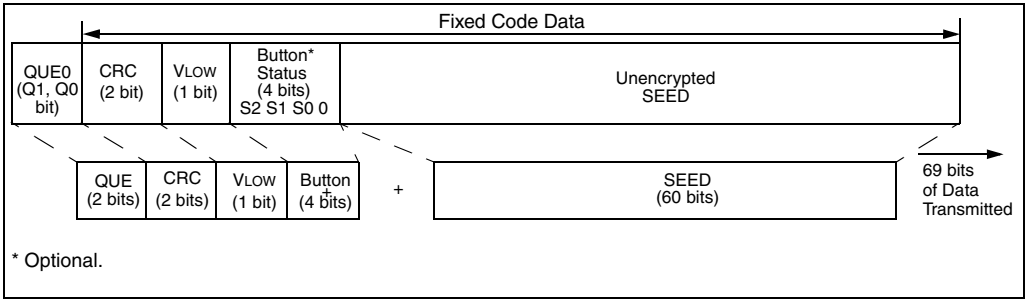


FIGURE 2-9: SEED CODE WORD ORGANIZATION



2.2.2 TRANSMISSION DATA MODULE

The Data Modulation Format is selectable between Pulse Width Modulation (PWM) format and Manchester encoding. Both formats are preceded by a preamble and synchronization header, followed by the 69-bits of data. Manchester encoding has a leading and closing '1' for each code word.

The same code word is continuously sent as long as the input pins are kept high with a guard time separating the code words. All of the timing values are in multiples of a Basic Timing Element (TE), which can be changed using the baud rate option bits.

FIGURE 2-10: TRANSMISSION FORMAT—MANCH = 0

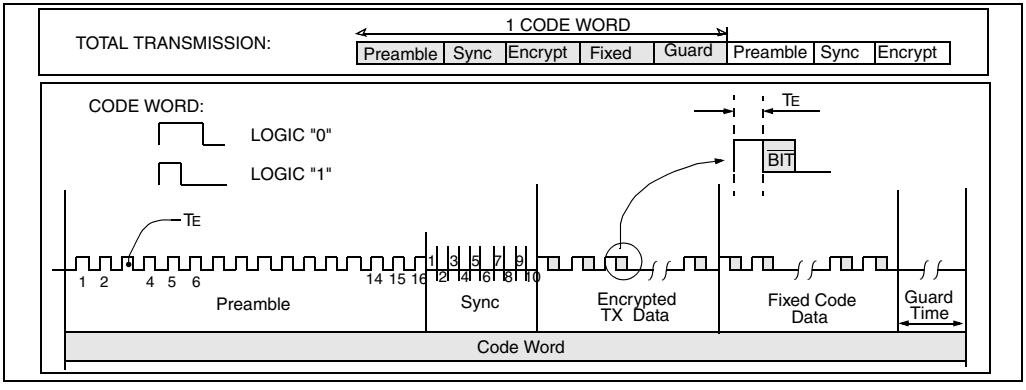
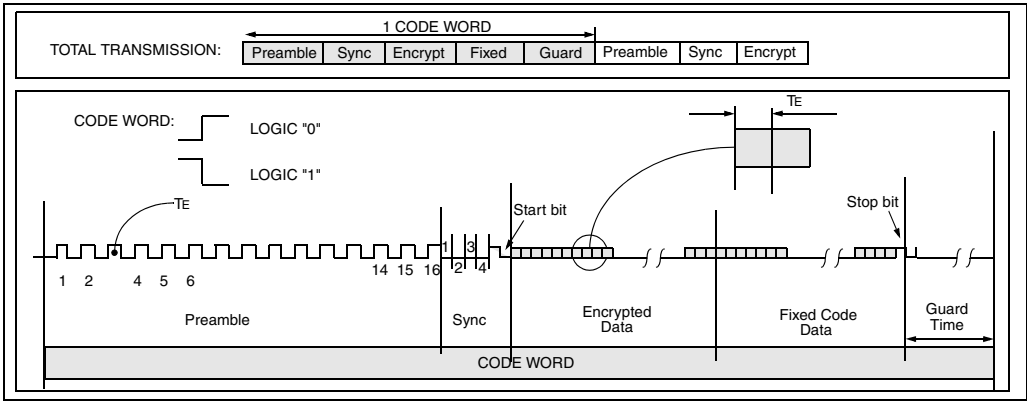


FIGURE 2-11: TRANSMISSION FORMAT—MANCH = 1



2.3 Code Hopping Mode Special Features

2.3.1 CODE WORD COMPLETION

Code word completion is an automatic feature that ensures that the entire code word is transmitted, even if the button is released before the transmission is complete. The HCS410 encoder powers itself up when a button is pushed and powers itself down after the command is finished (Figure 2-7). If MTX3 is set in the configuration word, a minimum of three transmissions will be transmitted when the HCS410 is activated, even if the buttons are released.

If less than seven words have been transmitted when the buttons are released, the HCS410 will complete the current word. If more than seven words have been transmitted, and the button is released, the PWM output is immediately switched off.

2.3.2 CODE WORD BLANKING ENABLE

Federal Communications Commission (FCC) part 15 rules specify the limits on fundamental power and harmonics that can be transmitted. Power is calculated on the worst case average power transmitted in a 100ms window. It is therefore advantageous to minimize the duty cycle of the transmitted word. This can be achieved by minimizing the duty cycle of the individual bits and by blanking out consecutive words. Code Word Blanking Enable (CWBE) is used for reducing the average power of a transmission (Figure 2-12). Using the CWBE allows the user to transmit a higher amplitude transmission if the transmission length is shorter. The FCC puts constraints on the average power that can be transmitted by a device, and CWBE effectively prevents continuous transmission by only allowing the transmission of every second or fourth word. This reduces the average power transmitted and hence, assists in FCC approval of a transmitter device.

The HCS410 will either transmit all code words, 1 in 2 or 1 in 4 code words, depending on the baud rate selected and the code word blanking option. See Section 3.7 for additional details.

2.3.3 CRC (CYCLE REDUNDANCY CHECK) BITS

The CRC bits are calculated on the 65 previously transmitted bits. The CRC bits can be used by the receiver to check the data integrity before processing starts. The CRC can detect all single bit and 66% of double bit errors. The CRC is computed as follows:

EQUATION 2-1: CRC CALCULATION

$$CRC[1]_{n+1} = CRC[0]_n \oplus Di_n$$

and

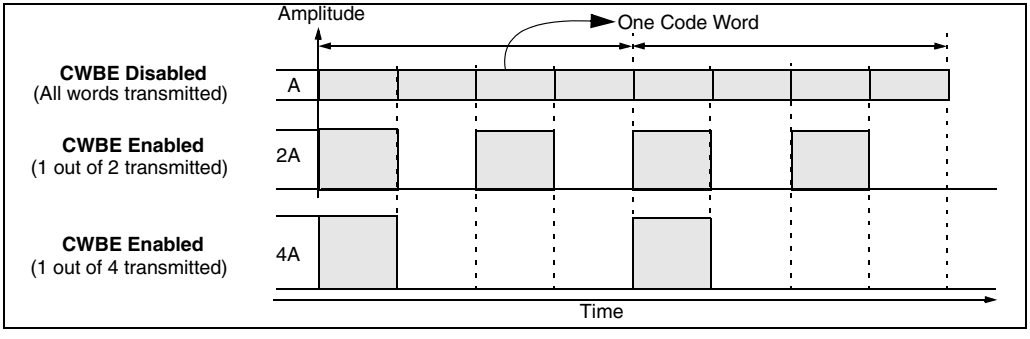
$$CRC[0]_{n+1} = (CRC[0]_n \oplus Di_n) \oplus CRC[1]_n$$

with

$$CRC[1,\ 0]_0 = 0$$

and Di_n the nth transmission bit 0 ð n ð 64

FIGURE 2-12: CODE WORD BLANKING ENABLE



•Patents have been applied for.

2.3.4 SEED TRANSMISSION

In order to increase the level of security in a system, it is possible for the receiver to implement what is known as a secure learning function. This can be done by utilizing the seed value on the HCS410 which is stored in EEPROM. Instead of the normal key generation method being used to create the encoder key, this seed value is used and there should not be any mathematical relationship between serial numbers and seeds for the best security. See Section 3.7.3 for additional details.

2.3.5 PASSIVE PROXIMITY ACTIVATION

If the HCS410 is brought into a magnetic field it enters IFF mode. In this mode it sends out ACK pulses on the LC lines. If the HCS410 doesn't receive any response to the first set of ack pulses within 50 ms the HCS410 will transmit a normal code hopping transmission for 2 seconds if XPRF is set in the configuration word. The function code during this transmission is S2:S0 = 000.

2.3.6 AUTO-SHUTOFF

The Auto-shutoff function automatically stops the device from transmitting if a button inadvertently gets pressed for a long period of time. This will prevent the device from draining the battery if a button gets pressed while the transmitter is in a pocket or purse. Time-out period is approximately 20 seconds.

2.3.7 VLOW: VOLTAGE LOW INDICATOR

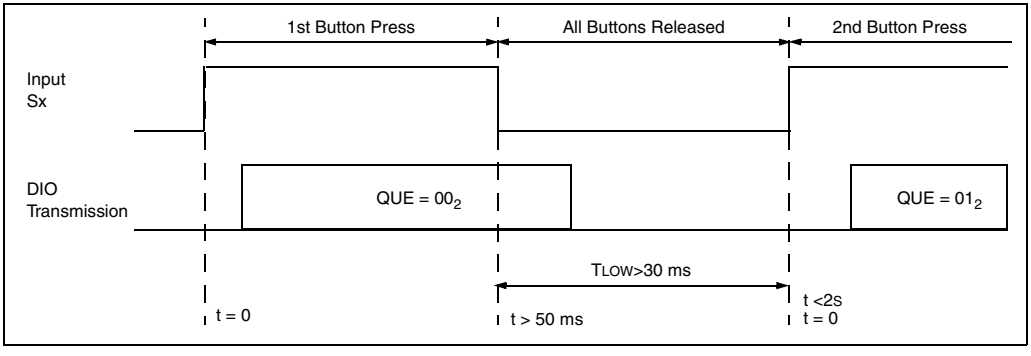
The VLOW bit is transmitted with every transmission (Figure 2-8). VLOW is set when the operating voltage has dropped below the low voltage trip point, approximately 2.2V or 4.4V selectable at 25°C. This VLOW signal is transmitted so the receiver can give an indication to the user that the transmitter battery is low.

2.3.8 QUE0:QUE1: QUEUING INFORMATION

If a button is pressed, released for more than 30 ms, and pressed again within 2 seconds of the first press, the QUE counter is incremented (Figure 2-7). The transmission that the HCS410 is busy with is aborted and a new transmission is begun with the new QUE bits set. These bits can be used by the decoder to perform secondary functions using only a single button without the requirement that the decoder receive more than one completed transmission. For example if none of the QUE bits are set the decoder only unlocks the driver's door, if QUE0 is set (double press on the transmitter) the decoder unlocks all the doors.

Note 1:	The QUE will not overflow.
2:	The button must be pressed for more than 50 ms.

FIGURE 2-13: QUE COUNTER TIMING DIAGRAM



2.3.9 LED OUTPUT

The S2/LED line can be used to drive a LED when the HCS410 is transmitting. If this option is enabled in the configuration word the S2 line is driven high periodically when the HCS410 is transmitting as shown in Figure 2-14. The LED output operates with a 30 ms on and 480 ms off duty cycle when the supply voltage is above the level indicated by the VLOW bit in the configuration word. When the supply voltage drops below the voltage indicated by the VLOW bit the HCS410 will indicate this by turning the LED on for 200ms at the start of a transmission and remain off for the rest of the transmission.

2.3.10 DELAYED INCREMENT

The HCS410 has a delayed increment feature that increments the counter by 12, 20 seconds after the last button press occurred. The 20-second time-out is reset and the queue counter will increment if another press occurs before the 20 seconds expires. The queue counter is cleared after the buttons have been released for more than 2 seconds. Systems that use this feature will circumvent the latest jamming-code grabbing attackers.

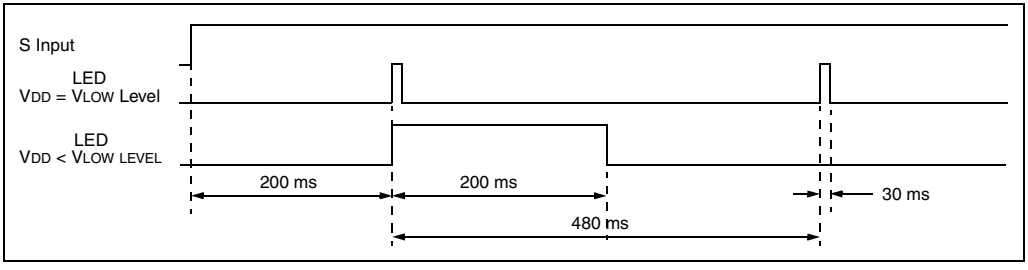
2.3.11 OTHER CONFIGURABLE OPTIONS

Other configurable code hopping options include an

- Transmission-rate selection
- Extended serial number.

These are described in more detail in Section 3.7.

FIGURE 2-14: LED INDICATION DURING TRANSMISSION



2.4 IFF Mode

IFF mode allows the decoder to perform an IFF validation, to write to the user EEPROM and to read from the user EEPROM. Each operation consists of the decoder sending an opcode data and the HCS410 giving a response.

There are two IFF modes: IFF1 and IFF2. IFF1 allows only one key IFF, while IFF2 allows two keys to be used.

Note:	When IFF2 is enabled, seed transmissions will not be allowed.

It is possible to use the HCS410 as an IFF token without using a magnetic field for coupling. The HCS410 can be directly connected to the data line of the decoder as shown in Figure 2-3. The HCS410 gets its power from the data line as it would in normal transponder mode. The communication is identical to the communication used in transponder mode.

2.4.1 IFF MODE ACTIVATION

The HCS410 will enter IFF mode if the capacitor/inductor resonant circuit generates a voltage greater than approximately 1.0 volts on LC0. After the verified application of power and elapse of the normal reset period, the device will start responding by pulsing the DATA line (LC0/1) with pulses as shown in Figure 2-17. This action will continue until the pulse train is terminated by receiving a start signal of duration 2TE, on the LC inputs before the next expected marker pulse. The device now enters the IFF mode and expects to receive an 'Opcode' and a 0/16/32-bit Data-stream to react on. The data rate (TE) is determined by the TBSL bits in the configuration word. See Section 3.0 for additional details.

2.4.2 IFF DECODER COMMANDS

As shown in Figure 2-15, a logic 1 and 0 are differentiated by the time between two rising edges. A long pulse indicates a 1; a short pulse, a 0.

FIGURE 2-15: MODULATION FOR IFF COMMUNICATION

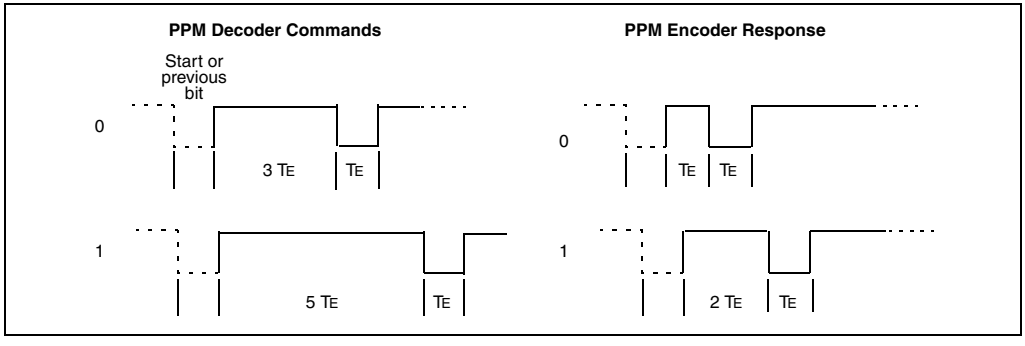


FIGURE 2-16: OVERVIEW OF IFF OPERATION

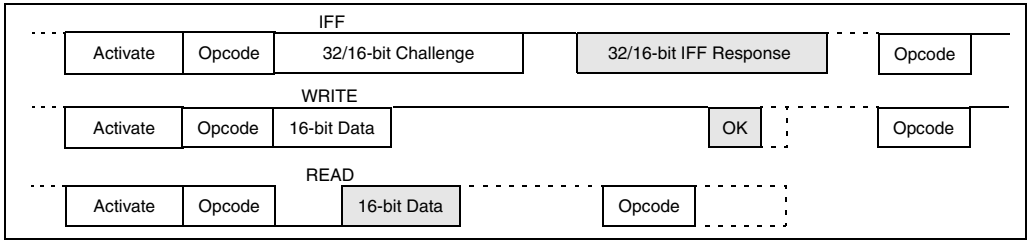


FIGURE 2-17: DECODER IFF COMMANDS AND WAVEFORMS

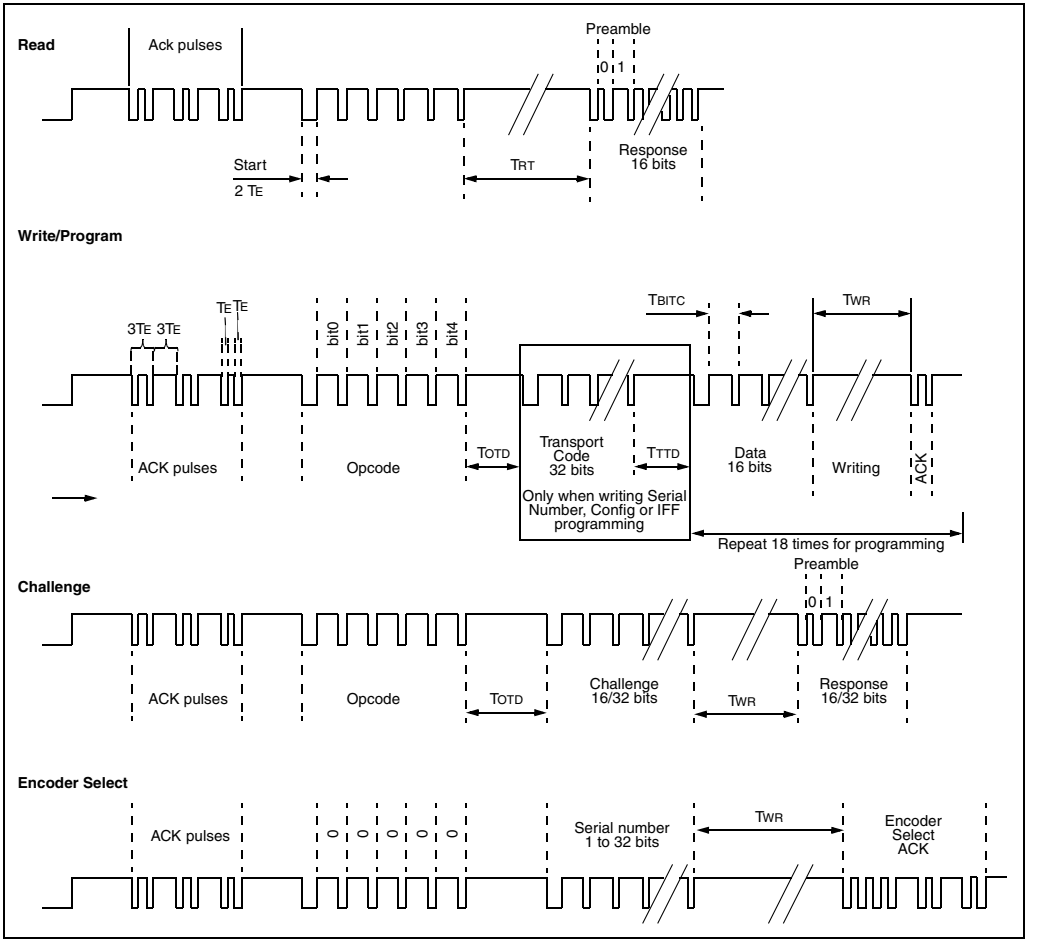


TABLE 2-3: IFF TIMING PARAMETERS

Parameter	Symbol	Minimum	Typical	Maximum	Units
Time Element IFFB = 0 IFFB = 1	TE	— —	200 100	— —	µs
PPM Command Bit Time Data = 1 Data = 0	TBITC	3.5 5.5	4 6	— —	TE
PPM Response Bit Time Data = 1 Data = 0	TBITR	— —	2 3	— —	TE
PPM Command Minimum High Time	TPMH	1.5	—	—	TE
Response Time (Minimum for Read)	TRT	6.5	—	—	ms
Opcode to Data Input Time	TOTD	1.8	—	—	ms
Transport Code to Data Input Time	TTTD	6.8	—	—	ms
IFF EEPROM Write Time (16 bits)	TWR	—	—	30	ms

2.4.3 HCS410 RESPONSES

The responses from the HCS410 are in PPM format. See Figure 2-17 for additional information. Every response from the HCS410 is preceded by a “2 bit preamble” of 01_2, and then 16/32 bits of data.

2.4.4 IFF RESPONSE

The 16/32-bit response to a 16/32-bit challenge, is transmitted once, after which the device is ready to accept another command. The same applies to the result of a Read command. The opcode written to the device specifies the challenge length and algorithm used. The response always starts with a leading preamble of 01_2 followed by the 16/32 bits of data.

2.4.5 IFF WRITE

The decoder can write to USER[0:3], SER[0:1], and the configuration word in the EEPROM.

After the HCS410 has written the word into the EEPROM, it will give two acknowledge pulses (T_E wide and T_E apart) on the LC pins.

When writing to the serial number or configuration word, the user must send the transport code before the write will begin (Section 3.4) .

Note:	If the configuration word is written, the device must be reset to allow the new configuration settings to come into effect.

2.4.6 IFF READ

The decoder can read USER[0:3], SER[0:1], and the configuration word in the EEPROM. After the data has been read, the device is ready to receive a command again.

Each read command is followed by a 16-bit data response. The response always starts with a leading preamble of 01_2 and then the 16-bits of data.

2.4.7 IFF PROGRAMMING

Upon receiving a programming opcode and the transport code, the EEPROM is erased (Section 3.4). Thereafter, the first 16 bits of data can be written. After indicating that a write command has been successfully completed the device is ready to receive the next 16 bits. After a complete memory map was received, it will be transmitted in PPM format on the LC pins as 16-bit words. This enables wireless programming of the device.

After the EEPROM is erased, the configuration word is reloaded. This results in oscillator tuning bits of 0000 being used during programming. When using IFF programming, the user should read the configuration word and store the oscillator bits in the memory map to be programmed. A program command should be sent and the next set of ACK pulses transmitted by the HCS410 should be used to determine the T_E. A second program command can then be sent, and the device programmed using the T_E just calibrated.

2.5 IFF Opcodes

TABLE 2-4: LIST OF IFF COMMANDS

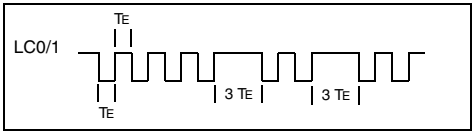
Command	Description	Expected data In	Response
00000	Select HCS410, used if Anti-collision enabled	1 to 32 bits of the serial number (SER)	Encoder select acknowledge if SER match
00001	Read configuration word	None	16-bit configuration word
00010	Read low serial number	None	Lower 16 bits of serial number (SER0)
00011	Read high serial number	None	Higher 16 bits of serial number (SER1)
00100	Read user area 0	None	16 Bits of User EEPROM USR0
00101	Read user area 1	None	16 Bits of User EEPROM USR1
00110	Read user area 2	None	16 Bits of User EEPROM USR2
00111	Read user area 3	None	16 Bits of User EEPROM USR3
01000	Program HCS410 EEPROM	Transport code (32 bits); Complete memory map: 18 x 16 bit words (288 bits)	Write acknowledge pulse after each 16-bit word, 288 bits transmitted in 18 bursts of 16-bit words
01001	Write configuration word	Transport code (32 bits); 16 Bit configuration word	Write acknowledge pulse
01010	Write low serial number	Transport code (32 bits); Lower 16 bits of serial number (SER0)	Write acknowledge pulse
01011	Write high serial number	Transport code (32 bits); Higher 16 bits of serial number (SER1)	Write acknowledge pulse
01100	Write user area 0	16 Bits of User EEPROM USR0	Write acknowledge pulse
01101	Write user area 1	16 Bits of User EEPROM USR1	Write acknowledge pulse
01110	Write user area 2	16 Bits of User EEPROM USR2	Write acknowledge pulse
01111	Write user area 3	16 Bits of User EEPROM USR3	Write acknowledge pulse
1X000	IFF1 using key-1 and IFF algorithm	32-Bit Challenge	32-Bit Response
1X001	IFF1 using key-1 and HOP algorithm	32-Bit Challenge	32-Bit Response
1X100	IFF2 32-bit using key-2 and IFF algorithm	32-Bit Challenge	32-Bit Response
1X101	IFF2 32-bit using key-2 and HOP algorithm	32-Bit Challenge	32-Bit Response

2.6 IFF Special Features

2.6.1 ANTI-COLLISION (ACOLI)

When the ACOLI bit is set in the configuration word, anti-collision mode is entered. The HCS410 will start sending ACK pulses when it enters a magnetic field. The ACK pulses stop as soon as the HCS410 detects a start bit from the decoder. A 'select encoder' opcode (00000) is then sent out by the decoder, followed by a 32-bit serial number. If the serial number matches the HCS410's serial number, the HCS410 will acknowledge with the acknowledge sequence as shown in Figure 2-18. The HCS410 can then be addressed as normal. If the serial number does not match, the IFF encoder will stop transmitting ACK pulses until it is either removed from the field or the correct serial number is given.

FIGURE 2-18: SERIAL NUMBER CORRECT ACKNOWLEDGE SEQUENCE



2.6.2 TRANSPONDER IN/RF OUT

When in transponder mode with ACOLI and XPRF set, the outputs of the HCS410's LC0:LC1 pins are echoed on the PWM output line. After transmitting the data on the LC pins, the data is then transmitted on the PWM line. The transmission format mirrors a code hopping transmission. The response replaces the 32-bit code hopping portion of the transmission. If the response is a 16-bit response, the 16 bits are duplicated to make up the 32-bit code hopping portion. The preamble, serial number, CRC, and queuing bits are all transmitted as normal (Figure 2-19).

This feature will be used in applications which use RF for long distance unidirectional authentication and short distance IFF.

Note:	If code word blanking is enabled, the HCS410 will not give any ACK pulses after a read, write or IFF.

2.6.3 INTELLIGENT DAMPING

If the LC circuit on the transponder has a high Q-factor, the circuit will keep on resonating for a long time after the field has been shut down by the decoder. This makes fast communication from the decoder to the HCS410 difficult. If the IDAMP bit is set to 0, the HCS410 will clamp the LC pins for 5 µs every 1/4 TE, whenever the HCS410 is expecting data from the decoder. The intelligent dumping pulses start 64 TE after the acknowledge pulses have been sent and continue for 64 TE. If the HSC410 detects data from the base station while sending out dump pulses, the dump pulses will continue to be sent. This option can be set in the configuration word.

2.7 LED Indicator

If a signal is detected on LC0, the LED pin goes high for 30 ms every 8s (IFFB = 0) or 4s (IFFB 1) to indicate that the power source is charging.

FIGURE 2-19: IFF INDUCTIVE IN RF OUT

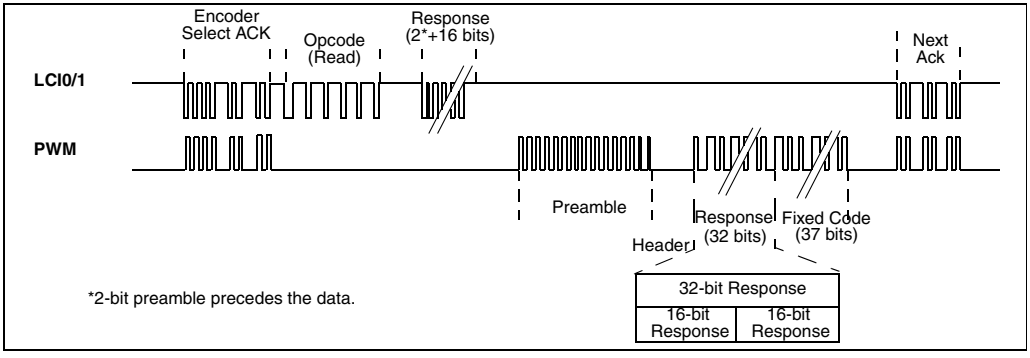
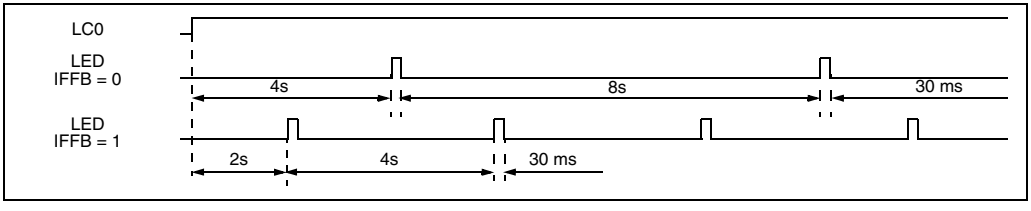


FIGURE 2-20: LED INDICATOR WHEN CHARGING POWER SOURCE



*Patents have been applied for.

3.0 EEPROM ORGANIZATION AND CONFIGURATION

The HCS410 has nonvolatile EEPROM memory which is used to store user programmable options. This information includes encoder keys, serial number, and up to 64-bits of user information.

The HCS410 has two modes in which it operates as specified by the configuration word. In the first mode the HCS410 has a single encoder key which is used for encrypting the code hopping portion of a CH Mode transmission and generating a response during IFF validation. Seed transmissions are allowed in this mode. In the second mode the HCS410 is a transponder device with two encoder keys.

The two different operating modes of the HCS410 lead to different EEPROM memory maps.

In IFF1 mode, the HCS410 can act as a code hopping encoder with Seed transmission, and as an IFF token with one key.

IFF1 Mode
64-bit Encoder Key 1
64-bit Seed/Transport Code (SEED0, SEED1, SEED2, SEED3)
32-bit Serial Number (SER0, SER1)
64-bit User Area (USR0, USR1, USER2, USR3)
10-bit Discrimination Value and 2 Overflow Bits.
16-bit Synchronization Counter
Configuration Data

In IFF2 mode, the HCS410 is able to act as a code hopping transmitter and an IFF token with two encoder keys.

IFF2 Mode
64-bit Encoder Key 1
64-bit Encoder Key 2/Transport Code
32-bit Serial Number (SER0, SER1)
64-bit User EEPROM (USR0, USR1, USER2, USR3)
10-bit Discrimination Value and 2 Overflow Bits.
16-bit Synchronization Counter
Configuration Data

3.1 Encoder Key 1 and 2

The 64-bit encoder key1 is used by the transmitter to create the encrypted message transmitted to the receiver in Code Hopping Mode. An IFF operation, can use encoder key1 or key2 to generate the response to a challenge received. The key(s) is created and programmed at the time of production using a key generation algorithm. Inputs to the key generation algorithm are the serial number or seed for the particular transmitter being used and a secret manufacturer's code. While a number of key generation algorithms are supplied by Microchip, a user may elect to create their own method of key generation. This may be done providing that the decoder is programmed with the same means of creating the key for decryption purposes. If a seed is used (CH Mode), the seed will also form part of the input to the key generation algorithm.

3.2 Discrimination Value and Overflow

The discrimination value forms part of the code hopping portion of a code hopping transmission. The least significant 10 bits of the discrimination value are typically set to the least significant bits of the serial number. The most significant 2 bits of the discrimination value are the overflow bits (OVR1: OVR0). These are used to extend the range of the synchronization counter. When the synchronization counter wraps from $FFFF_{16}$ to 0000_{16} OVR0 is cleared and the second time a wrap occurs OVR1 is cleared.

Once cleared, the overflow bits cannot be set again, thereby creating a permanent record of the counter overflow.

3.3 16-bit Synchronization Counter

This is the 16-bit synchronization counter value that is used to create the code hopping portion for transmission. This value will be changed after every transmission. The synchronization counter is not used in IFF mode.

*Patents have been applied for.

3.4 60/64-bit Seed Word/Transport Code

This is the 60-bit seed code that is transmitted when seed transmission is selected. This allows the system designer to implement the secure learn feature or use this fixed code word as part of a different key generation/tracking process or purely as a fixed code transmission. The seed is not available in IFF2-mode. A Seed transmission can be initiated in two ways, depending on the button inputs (Figure 3-1).

Seed transmission is available for function codes (Table 2-2) S[2:0] = 111 and S[2:0] = 011 (delayed). The delayed seed transmission starts with a normal code hopping transmission being transmitted for 3 seconds, before switching to a seed transmission. The two seed transmissions are shown in Figure 3-1.

The least significant 32-bits of the seed are used as the transport code. The transport code is used to write-protect the serial number, configuration word, as well as preventing accidental programming of the HCS410 when in IFF mode.

Note:	If both SEED and TMPSD are set, IFF2 mode is enabled.

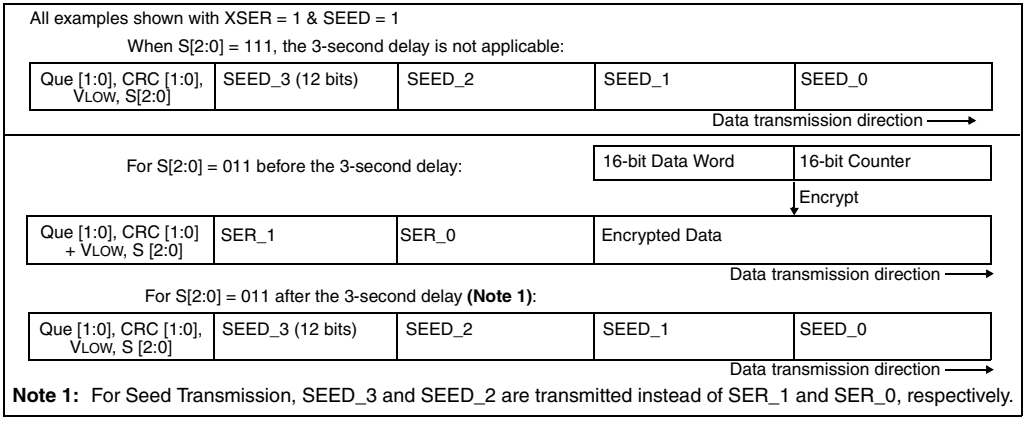
3.5 Encoder Serial Number

There are 32 bits allocated for the serial number and a selectable configuration bit (XSER) determines whether 32 or 28 bits will be transmitted. The serial number is meant to be unique for every transmitter.

3.6 User Data

The 64-bit user EEPROM can be reprogrammed and read at any time using the IFF interface.

FIGURE 3-1: SEED TRANSMISSION



Note 1: For Seed Transmission, SEED_3 and SEED_2 are transmitted instead of SER_1 and SER_0, respectively.

3.7 Configuration Data

The configuration data is used to select various encoder options. Further explanations of each of the bits are described in the following sections.

TABLE 3-1: CONFIGURATION OPTIONS SEED

Symbol	Description
CWBE	Code Word Blanking Enable
IDAMP	Intelligent Damping for High Q LC Tank.
SEED/ IFF2	Enable Seed Transmissions
TMPSD/ IFF2	Temporary Seed Transmissions
OSC0:3	Onboard Oscillator Tuning Bits
MTX3	Minimum 3 Code Words Transmitted
VLOW	Low Voltage Trip Point Selection
LED	Enable LED output
BSL0:1	Baudrate Select
TBSL	Transponder Baud Rate
MANCH	Manchester Modulation Mode
ACOLI	Anti Collision Communication Enable
XPRF	Passive Proximity Activation
DINC	Delayed Increment Enable
XSER	Extended Serial Number

3.7.1 **CWBE:** CODE WORD BLANKING ENABLE
BSL: BAUD RATE SELECT

Selecting this option allows code blanking as shown in Table 3-3. If this option is not selected, all code words are transmitted.

3.7.2 **IDAMP:** INTELLIGENT DAMPING

If IDAMP is set to '1' intelligent damping is disabled.

3.7.3 **SEED, TMPSD:** SEED TRANSMISSION

SEED	TMPSD	Description
0	0	No Seed/1 IFF Key
0	1	Seed Limited*
1	0	Always Enabled
1	1	IFF2/No Seed/2 IFF Keys

* Seed transmissions are allowed till the sychronization counter crosses a $XX7F_{16}$ boundary. e.g. If the counter is initialized to 0000_{16} when the device is programmed, seed transmissions will be allowed until the counter wraps from $007F_{16}$ to 0080_{16} giving the user 127 transmissions before seed transmissions are disabled.

3.7.4 **OSC:** OSCILLATOR TUNING BITS

These bits allow the onboard oscillator to be tuned to within 10% of the nominal oscillator speed over both temperature and voltage.

TABLE 3-2: OSCILLATOR TUNING

OSC	Description
1000	Fastest
1001 1010 • • • 1111	Faster
0000	Nominal
0001 0010 • • • 0110	Slower
0111	Slowest

TABLE 3-3: BAUD RATE SELECTION

		Code Hopping Transmissions (TE)			Transponder Communication (TE)	
BSL 1	BSL 0	PWM	Manchester	Codes Word Transmitted*	TBSL	PPM
0	0	400 µs	800 µs	All	0	200 µs
0	1	200 µs	400 µs	1 of 2	—	—
1	0	100 µs	200 µs	1 of 2	—	—
1	1	100 µs	200 µs	1 of 4	1	100 µs

Note: *If code word blanking is enabled.

3.7.5 **MTX3:** MINIMUM CODE WORDS COMPLETED

If this bit is set, the HCS410 will transmit a minimum of 3 words before it powers itself down. If this bit is cleared, the HCS410 will only complete the current transmission. This feature will only work if V_{DD} is connected directly to the battery as shown in Figure 2-1.

3.7.6 **VLOW:** LOW VOLTAGE TRIP POINT

The low voltage trip point select bit is used to tell the HCS410 what Vdd level is being used. This information will be used by the device to determine when to send the voltage low signal to the receiver. When this bit is set, the Vdd level is assumed to be operating from a 5 volt or 6 volt supply. If the bit is cleared, then the Vdd level is assumed to be 3.0 volts. Refer to Figure 6-3 for voltage trip point. When the battery reaches the Vlow point, the LED will flash once for 200 ms on during a code hopping transmission.

3.7.7 **LED:** OUTPUT ENABLE

If this bit is set, the S2 doubles as an LED output line. If this bit is cleared (0), S2 is only used as an input.

3.7.8 **TBSL:** TRANSPONDER BAUD RATE SELECT

This option selects the baud rate for IFF communication between a T_E of 100 µs or 200 µs.

3.7.9 **MANCH:** MANCHESTER CODE ENCODING

MANCH selects between Manchester code modulation and PWM modulation in code hopping mode. If MANCH = 1, Manchester code modulation is selected. If MANCH is cleared, PWM modulation is selected.

3.7.10 **ACOLI:** ANTI-COLLISION COMMUNICATION AND **XPRF**: TRANSPONDER ECHOING ON PWM OUTPUT

ACOLI = 1, XPRF = 0

If ACOLI is set the anti-collision operation during bi-directional transponder mode (IFF) is enabled. This feature is useful in situations where multiple transponders enter the magnetic field simultaneously.

ACOLI = 0, XPRF = 1

If XPRF is set, and ACOLI is cleared, proximity activation is enabled. the HCS410 starts sending out ACK pulses when it detects a magnetic field. If the HCS410 doesn't receive a start bit from the decoder within 50 ms of sending the first set of ACK pulses, the HCS410 will transmit a code hopping transmission PWM pin for 2 seconds.

ACOLI = 1, XPRF = 1

If both the ACOLI and XPRF are set, all of the HCS410 transponder responses are echoed on the PWM output, as described in Section 2.6.2.

3.7.11 **DINC:** DELAYED INCREMENT

If DINC is set to '1', the delayed increment feature is enabled. If DINC is cleared, the counter only increments once each time the button is pressed.

3.7.12 **XSER:** EXTENDED SERIAL NUMBER

If XSER is set, bits 60 to 63 of the transmission are the most significant bits of the serial number or seed. If XSER bit is cleared, bits 60 to 63 of the transmission are set to the function code used to activate the device (S2:S1:S0:0).

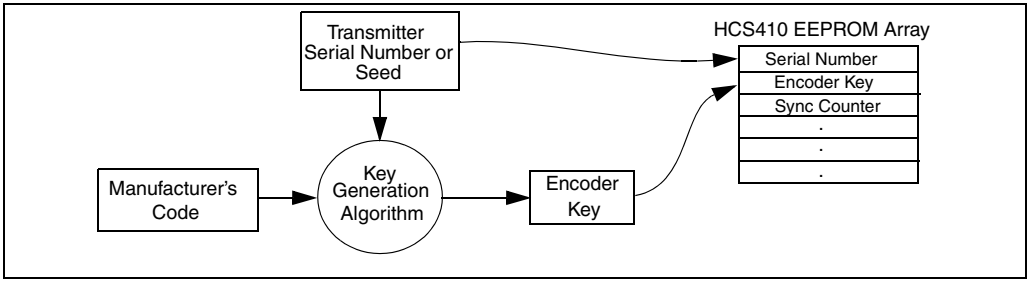
4.0 INTEGRATING THE HCS410 INTO A SYSTEM

Use of the HCS410 in a system requires a compatible decoder. This decoder is typically a microcontroller with compatible firmware. Firmware routines that accept transmissions from the HCS410, decrypt the code hopping portion of the data stream and perform IFF functions are available. These routines provide system designers the means to develop their own decoding system.

4.1 Key Generation

The serial number for each transmitter is programmed by the manufacturer at the time of production. The generation of the encoder key is done using a key generation algorithm (Figure 4-1). Typically, inputs to the key generation algorithm are the serial number of the transmitter or seed value, and a 64-bit manufacturer's code. The manufacturer's code is chosen by the system manufacturer and must be carefully controlled. The manufacturer's code is a pivotal part of the overall system security.

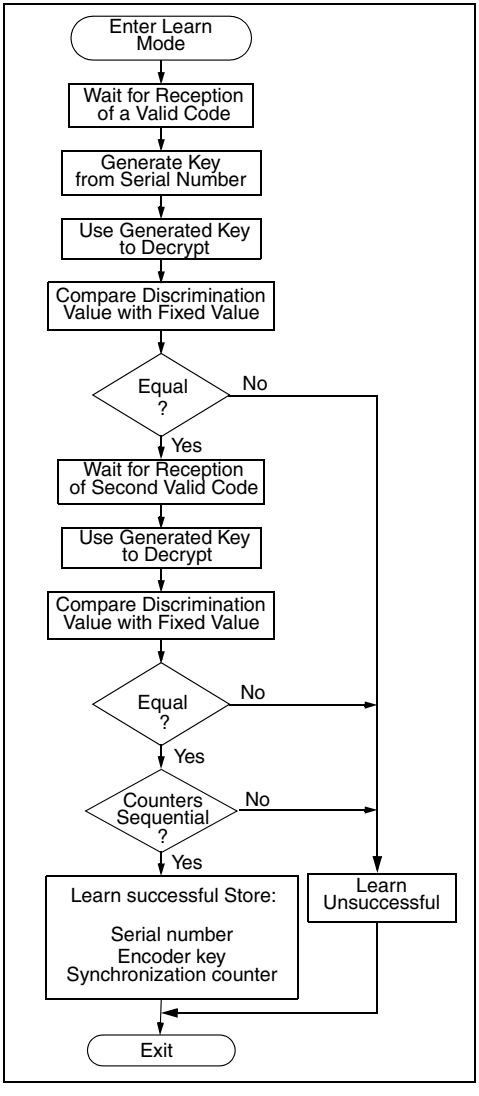
FIGURE 4-1: CREATION AND STORAGE OF ENCODER KEY DURING PRODUCTION



4.2 Learning an HCS410 to a Receiver

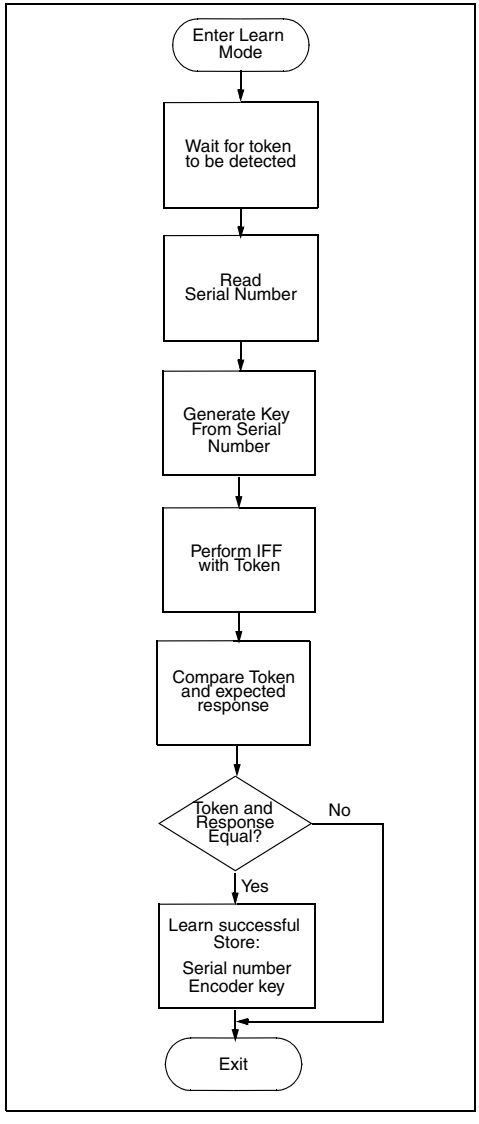
In order for a transmitter to be used with a decoder, the transmitter must first be 'learned'. Several learning strategies can be followed in the decoder implementation. When a transmitter is learned to a decoder, it is suggested that the decoder stores the serial number and current synchronization counter value (synchronization counter stored in CH Mode only) in EEPROM. The decoder must keep track of these values for every transmitter that is learned (Figure 4-2 and Figure 4-3).

FIGURE 4-2: TYPICAL CH MODE LEARN SEQUENCE



The maximum number of transmitters that can be learned is only a function of how much EEPROM memory storage is available. The decoder must also store the manufacturer's code in order to learn an HCS410, although this value will not change in a typical system so it is usually stored as part of the microcontroller ROM code. Storing the manufacturer's code as part of the ROM code is also better for security reasons.

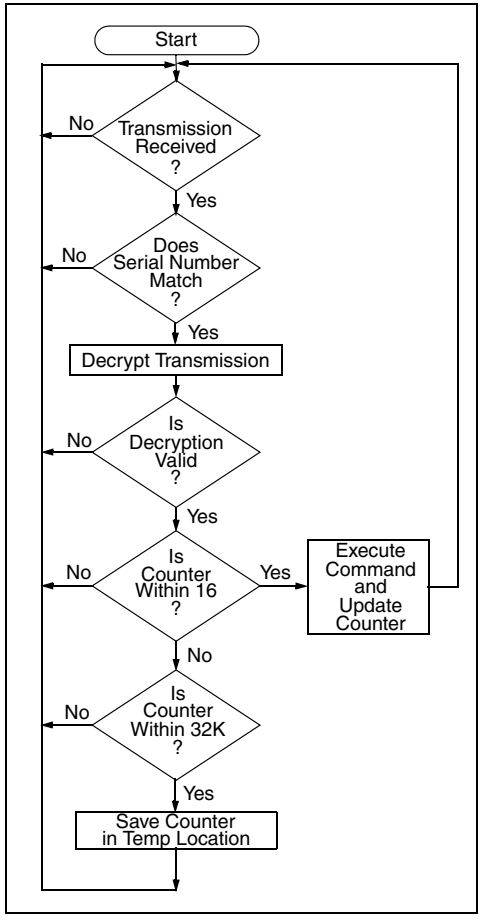
FIGURE 4-3: TYPICAL IFF LEARN SEQUENCE



4.3 CH Mode Decoder Operation

In a typical decoder operation (Figure 4-4), the key generation on the decoder side is done by taking the serial number from a transmission and combining that with the manufacturer's code to create the same encoder key that is stored in the HCS410. Once the encoder key is obtained, the rest of the transmission can be decrypted. The decoder waits for a transmission and immediately checks the serial number to determine if it is a learned transmitter. If it is, the code hopping portion of the transmission is decrypted using the stored key. It uses the discrimination bits to determine if the decryption was valid. If everything up to this point is valid, the synchronization counter value is evaluated.

FIGURE 4-4: TYPICAL CH MODE DECODER OPERATION



4.3.1 SYNCHRONIZATION WITH DECODER

The KEELOQ technology features a sophisticated synchronization technique (Figure 4-5) which does not require the calculation and storage of future codes. If the stored counter value for that particular transmitter and the counter value that was just decrypted are within a window of say 16, the counter is stored and the command is executed. If the counter value was not within the single operation window, but is within the double operation window of say 32K window, the transmitted synchronization counter value is stored in temporary location and it goes back to waiting for another transmission. When the next valid transmission is received, it will compare the new value with the one in temporary storage. If the two values are sequential, it is assumed that the counter had just gotten out of the single operation 'window', but is now back in sync, so the new synchronization counter value is stored and the command executed. If a transmitter has somehow gotten out of the double operation window, the transmitter will not work and must be relearned. Since the entire window rotates after each valid transmission, codes that have been used are part of the 'blocked' (32K) codes and are no longer valid. This eliminates the possibility of grabbing a previous code and retransmitting to gain entry.

Note:	The synchronization method described in this section is only a typical implementation and because it is usually implemented in firmware, it can be altered to fit the needs of a particular system

FIGURE 4-5: SYNCHRONIZATION WINDOW

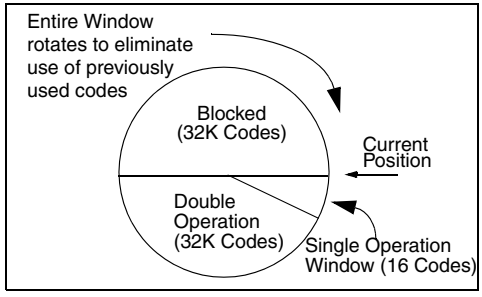


FIGURE 4-6: BASIC OPERATION OF A CODE HOPPING RECEIVER (DECODER)