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

HCS362

KEELOQ® Code Hopping Encoder

FEATURES

Security

- Programmable 28/32-bit serial number
- Two programmable 64-bit encryption keys
- Programmable 60-bit seed
- Each transmission is unique
- 69-bit transmission code length
- 32-bit hopping code
- 37-bit fixed code (28/32-bit serial number, 4/0-bit function code, 1-bit status, 2-bit CRC/time, 2-bit queue)
- Encryption keys are read protected

Operation

- 2.0V – 6.3V operation
- Four button inputs
- 15 functions available
- Selectable baud rates and code word blanking
- Programmable minimum code word completion
- Battery low signal transmitted to receiver with programmable threshold
- Non-volatile synchronization data
- PWM and Manchester modulation

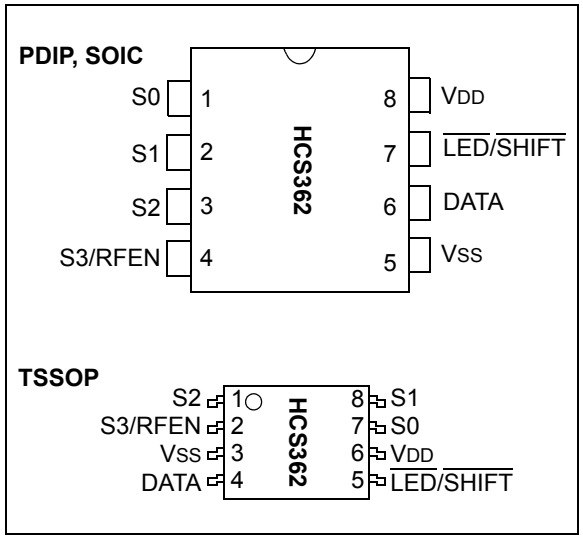
Other

- RF Enable output – PLL interface
- Easy to use programming interface
- On-chip EEPROM
- On-chip tunable oscillator and timing components
- Button inputs have internal pull-down resistors
- Current limiting on $\overline{\text{LED}}$ output
- Minimum component count

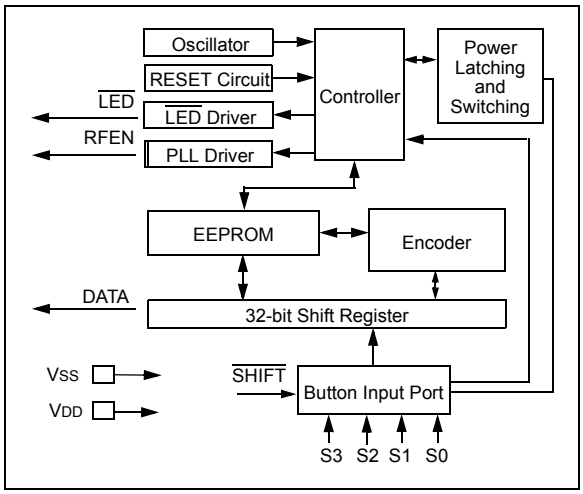
Enhanced Features Over HCS300

- 60-bit seed vs. 32-bit seed
- 2-bit CRC for error detection
- 28/32-bit serial number select
- Tunable oscillator (+/-10% over specified voltage ranges)
- Time bits option
- Queue bits
- TSSOP package
- Programmable Time-out and Guard Time

PACKAGE TYPES



HCS362 BLOCK DIAGRAM



Typical Applications

The HCS362 is ideal for Remote Keyless Entry (RKE) applications. These applications include:

- Automotive RKE systems
- Automotive alarm systems
- Automotive immobilizers
- Gate and garage door openers
- Identity tokens
- Burglar alarm systems

GENERAL DESCRIPTION

The HCS362 is a code hopping encoder designed for secure Remote Keyless Entry (RKE) systems. The HCS362 utilizes the KEELOQ® code hopping technology, which incorporates high security, a small package outline and low cost, to make this device a perfect solution for unidirectional remote keyless entry systems and access control systems.

The HCS362 combines a 32-bit hopping code generated by a nonlinear encryption algorithm, with a 28/32-bit serial number and 9/5 status bits to create a 69-bit transmission stream. The length of the transmission eliminates the threat of code scanning. The code hopping mechanism makes each transmission unique, thus rendering code capture and resend (code grabbing) schemes useless.

The crypt key, serial number and configuration data are stored in an EEPROM array which is not accessible via any external connection. The EEPROM data is programmable but read protected. The data can be verified only after an automatic erase and programming operation. This protects against attempts to gain access to keys or manipulate synchronization values. The HCS362 provides an easy to use serial interface for programming the necessary keys, system parameters and configuration data.

1.0 SYSTEM OVERVIEW

Key Terms

The following is a list of key terms used throughout this data sheet. For additional information on KEELOQ and Code Hopping, refer to Technical Brief 3 (TB003).

- **RKE** - Remote Keyless Entry
- **Button Status** - Indicates what button input(s) activated the transmission. Encompasses the 4 button status bits S3, S2, S1 and S0 (Figure 3-2).
- **Code Hopping** - A method by which a code, viewed externally to the system, appears to change unpredictably each time it is transmitted.
- **Code word** - A block of data that is repeatedly transmitted upon button activation (Figure 3-2).
- **Transmission** - A data stream consisting of repeating code words (Figure 8-1).
- **Crypt key** - A unique and secret 64-bit number used to encrypt and decrypt data. In a symmetrical block cipher such as the KEELOQ algorithm, the encryption and decryption keys are equal and will therefore be referred to generally as the crypt key.
- **Encoder** - A device that generates and encodes data.
- **Encryption Algorithm** - A recipe whereby data is scrambled using a crypt key. The data can only be interpreted by the respective decryption algorithm using the same crypt key.
- **Decoder** - A device that decodes data received from an encoder.
- **Decryption algorithm** - A recipe whereby data scrambled by an encryption algorithm can be unscrambled using the same crypt key.
- **Learn** – Learning involves the receiver calculating the transmitter's appropriate crypt key, decrypting the received hopping code and storing the serial number, synchronization counter value and crypt key in EEPROM. The KEELOQ product family facilitates several learning strategies to be implemented on the decoder. The following are examples of what can be done.
  - **Simple Learning**
    The receiver uses a fixed crypt key, common to all components of all systems by the same manufacturer, to decrypt the received code word's encrypted portion.
  - **Normal Learning**
    The receiver uses information transmitted during normal operation to derive the crypt key and decrypt the received code word's encrypted portion.
  - **Secure Learn**
    The transmitter is activated through a special button combination to transmit a stored 60-bit seed value used to generate the transmitter's crypt key. The receiver uses this seed value to derive the same crypt key and decrypt the received code word's encrypted portion.
- **Manufacturer's code** – A unique and secret 64-bit number used to generate unique encoder crypt keys. Each encoder is programmed with a crypt key that is a function of the manufacturer's code. Each decoder is programmed with the manufacturer code itself.

The HCS362 code hopping encoder is designed specifically for keyless entry systems; primarily vehicles and home garage door openers. The encoder portion of a keyless entry system is integrated into a transmitter, carried by the user and operated to gain access to a vehicle or restricted area. The HCS362 is meant to be a cost-effective yet secure solution to such systems, requiring very few external components (Figure 2-1).

Most low-end keyless entry transmitters are given a fixed identification code that is transmitted every time a button is pushed. The number of unique identification codes in a low-end system is usually a relatively small number. These shortcomings provide an opportunity for a sophisticated thief to create a device that 'grabs' a transmission and retransmits it later, or a device that quickly 'scans' all possible identification codes until the correct one is found.

The HCS362, on the other hand, employs the KEELOQ code hopping technology coupled with a transmission length of 66 bits to virtually eliminate the use of code 'grabbing' or code 'scanning'. The high security level of

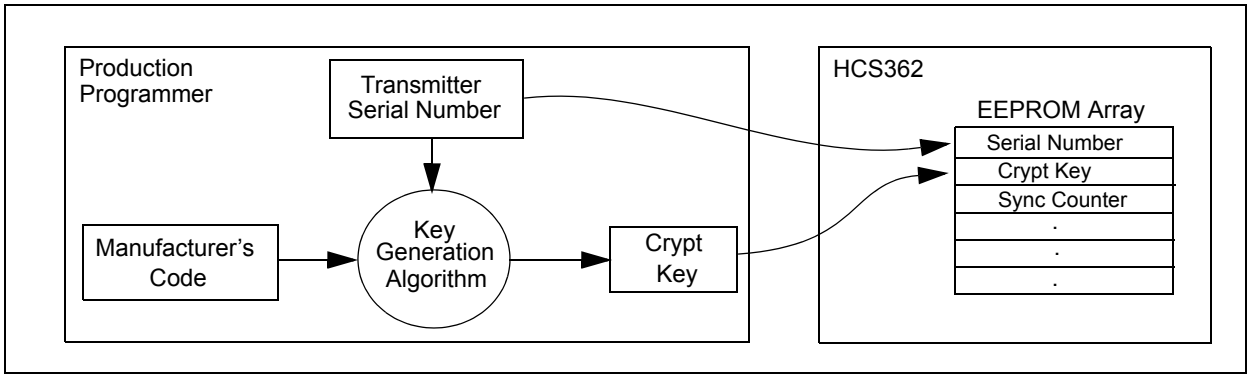
the HCS362 is based on the patented KEELOQ technology. A block cipher based on a block length of 32 bits and a key length of 64 bits is used. The algorithm obscures the information in such a way that even if the transmission information (before coding) differs by only one bit from that of the previous transmission, the next coded transmission will be completely different. Statistically, if only one bit in the 32-bit string of information changes, greater than 50 percent of the coded transmission bits will change.

As indicated in the block diagram on page one, the HCS362 has a small EEPROM array which must be loaded with several parameters before use; most often programmed by the manufacturer at the time of production. The most important of these are:

- A 28-bit serial number, typically unique for every encoder
- A crypt key
- An initial 16-bit synchronization value
- A 16-bit configuration value

The crypt key generation typically inputs the transmitter serial number and 64-bit manufacturer's code into the key generation algorithm (Figure 1-1). The manufacturer's code is chosen by the system manufacturer and must be carefully controlled as it is a pivotal part of the overall system security.

FIGURE 1-1: CREATION AND STORAGE OF CRYPT KEY DURING PRODUCTION



The 16-bit synchronization counter is the basis behind the transmitted code word changing for each transmission; it increments each time a button is pressed. Due to the code hopping algorithm's complexity, each increment of the synchronization value results in greater than 50% of the bits changing in the transmitted code word.

Figure 1-2 shows how the key values in EEPROM are used in the encoder. Once the encoder detects a button press, it reads the button inputs and updates the synchronization counter. The synchronization counter and crypt key are input to the encryption algorithm and the output is 32 bits of encrypted information. This data will change with every button press, its value appearing externally to 'randomly hop around', hence it is referred to as the hopping portion of the code word. The 32-bit hopping code is combined with the button information and serial number to form the code word transmitted to the receiver. The code word format is explained in greater detail in Section 3.1.

A receiver may use any type of controller as a decoder, but it is typically a microcontroller with compatible firmware that allows the decoder to operate in conjunction with an HCS362 based transmitter. Section 6.0 provides detail on integrating the HCS362 into a system.

A transmitter must first be 'learned' by the receiver before its use is allowed in the system. Learning includes calculating the transmitter's appropriate crypt key, decrypting the received hopping code and storing the serial number, synchronization counter value and crypt key in EEPROM.

In normal operation, each received message of valid format is evaluated. The serial number is used to determine if it is from a learned transmitter. If from a learned transmitter, the message is decrypted and the synchronization counter is verified. Finally, the button status is checked to see what operation is requested. Figure 1-3 shows the relationship between some of the values stored by the receiver and the values received from the transmitter.

FIGURE 1-2: BUILDING THE TRANSMITTED CODE WORD (ENCODER)

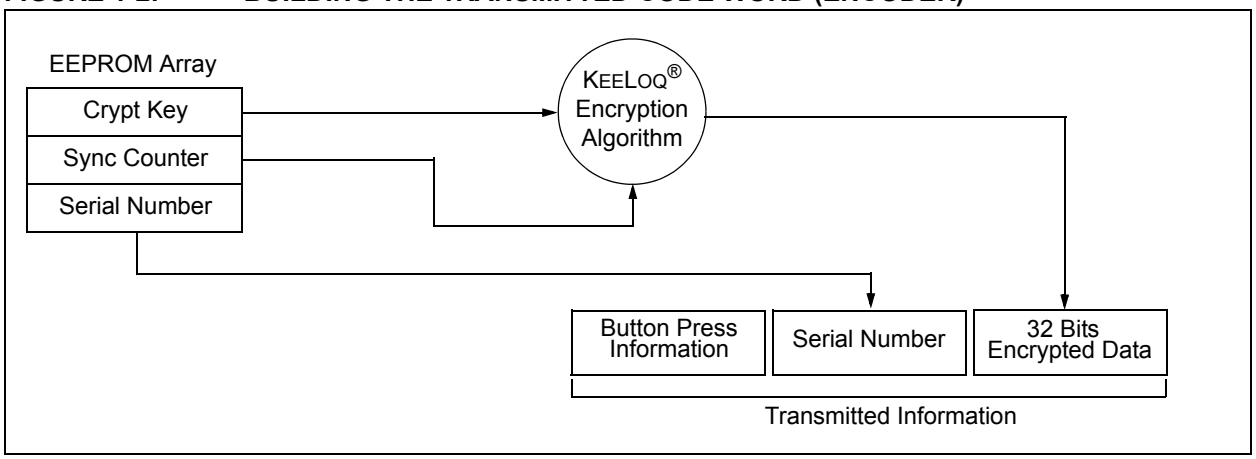
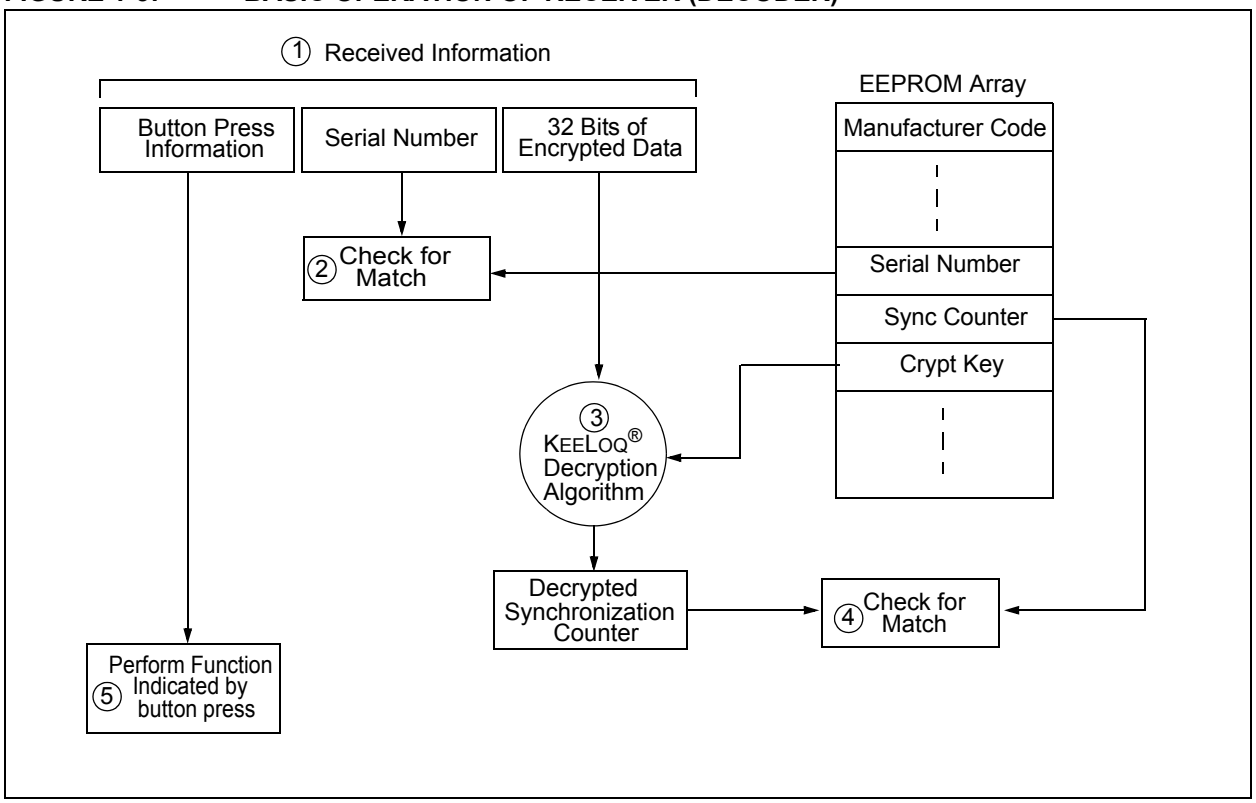


FIGURE 1-3: BASIC OPERATION OF RECEIVER (DECODER)



NOTE: Circled numbers indicate the order of execution.

2.0 DEVICE DESCRIPTION

As shown in the typical application circuits (Figure 2-1), the HCS362 is a simple device to use. It requires only the addition of buttons and RF circuitry for use as the transmitter in your security application. See Table 2-1 for a description of each pin and Figure 2-1 for typical circuits. Figure 2-2 shows the device I/O circuits.

TABLE 2-1: PIN DESCRIPTIONS

Name	Pin Number	Description
S0	1	Switch input 0
S1	2	Switch input 1
S2	3	Switch input 2 / Clock pin when in Programming mode
S3/ RFEN	4	Switch input 3 / RF enable output
VSS	5	Ground reference connection
DATA	6	Data output pin / DATA I/O pin for Programming mode
$\overline{\text{LED}}$/ $\overline{\text{SHIFT}}$	7	Cathode connection for $\overline{\text{LED}}$ and DUAL mode $\overline{\text{SHIFT}}$ input
VDD	8	Positive supply voltage

FIGURE 2-1: TYPICAL CIRCUITS

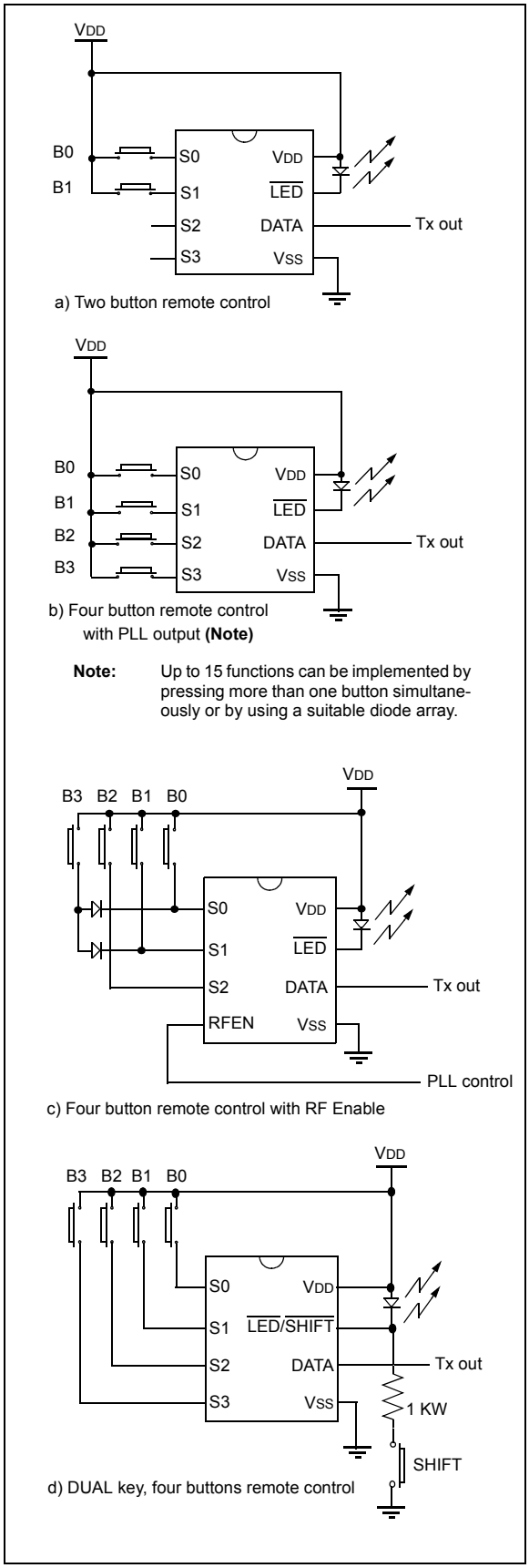
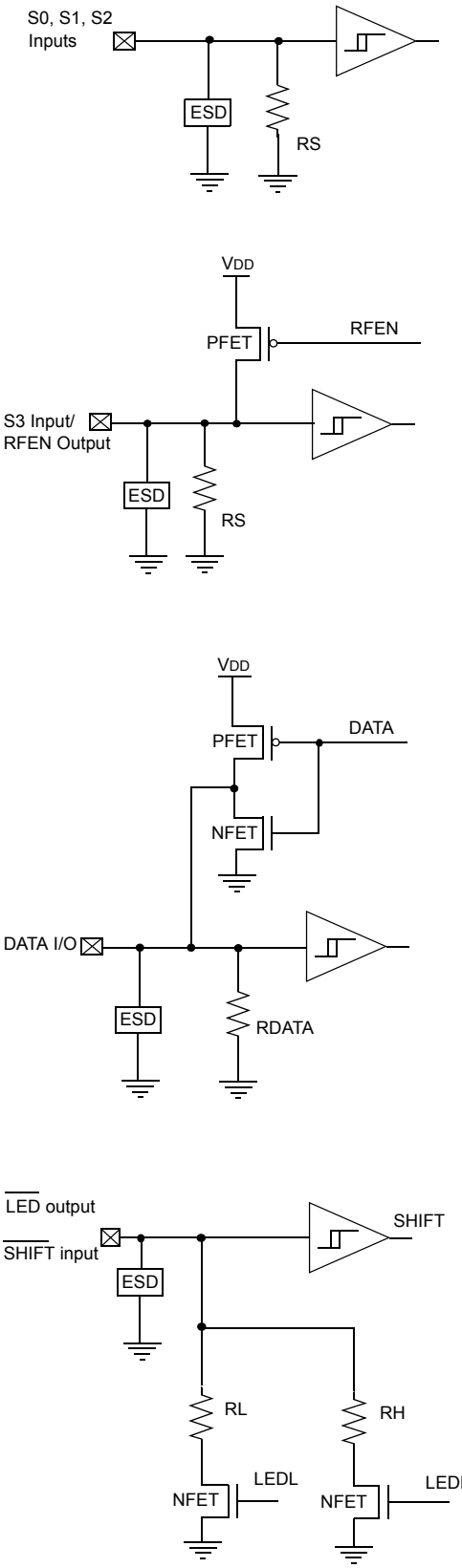


FIGURE 2-2: I/O CIRCUITS



2.1 Architectural Overview

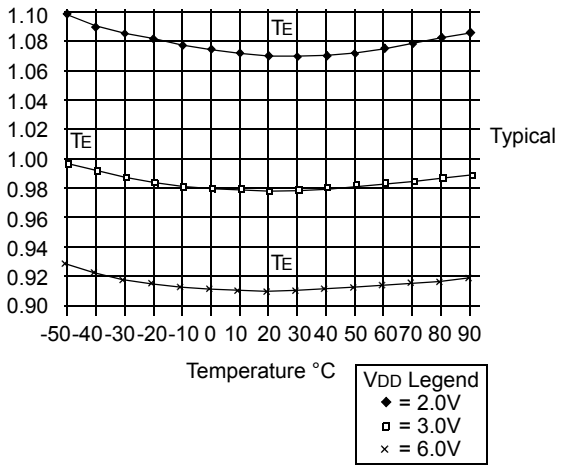
2.1.1 ONBOARD EEPROM

The HCS362 has an onboard non-volatile EEPROM, which is used to store user programmable data. The data can be programmed at the time of production and include the security-related information such as encoder keys, serial numbers, discrimination and seed values. All the security related options are read protected. The HCS362 has built in protection against counter corruption. Before every EEPROM write, the internal circuitry also ensures that the high voltage required to write to the EEPROM is at an acceptable level.

2.1.2 INTERNAL RC OSCILLATOR

The HCS362 has an onboard RC oscillator that controls all the logic output timing characteristics. The oscillator frequency varies within ±10% of the nominal value (once calibrated over a voltage range of 2V – 3.5V or 3.5V – 6.3V). All the timing values specified in this document are subject to the oscillator variation.

FIGURE 2-3: HCS362 NORMALIZED TE VS. TEMPERATURE



Note: Values are for calibrated oscillator

2.1.3 LOW VOLTAGE DETECTOR

A low battery voltage detector onboard the HCS362 can indicate when the operating voltage drops below a predetermined value. There are eight options available depending on the VLOW[0..2] configuration options. The options provided are:

000 - 2.0V	100 - 4.0V
001 - 2.1V	101 - 4.2V
010 - 2.2V	110 - 4.4V
011 - 2.3V	111 - 4.6V

FIGURE 2-4: HCS362 VLOW DETECTOR (TYPICAL)

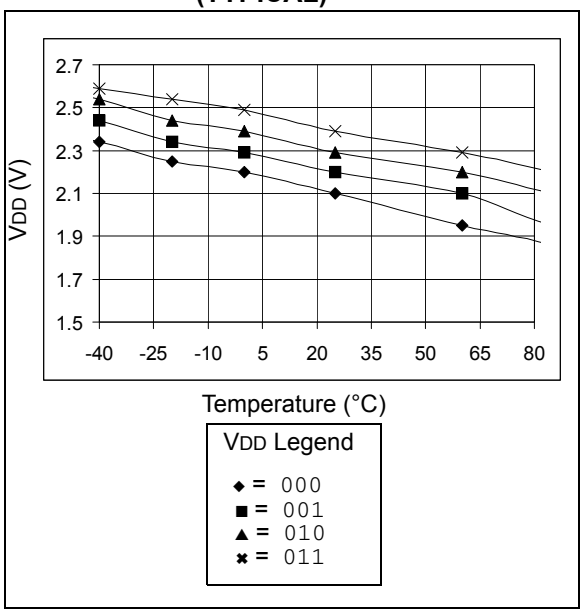
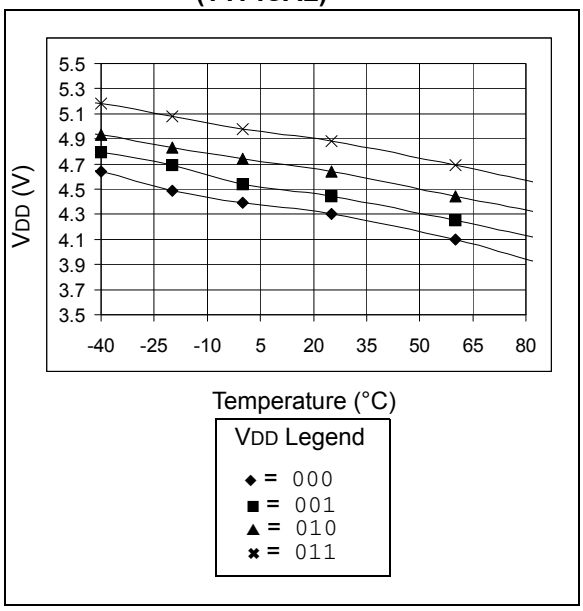


FIGURE 2-5: HCS362 VLOW DETECTOR (TYPICAL)



The output of the low voltage detector is transmitted in each code word, so the decoder can give an indication to the user that the transmitter battery is low. Operation of the $\overline{\text{LED}}$ changes as well to further indicate that the battery is low and needs replacing.

2.2 Dual Encoder Operation

The HCS362 contains two crypt keys (possibly derived from two different Manufacturer's Codes), but only one Serial Number, one set of Discrimination bits, one 16-bit Synchronization Counter and a single 60-bit Seed value. For this reason the HCS362 can be used as an encoder in multiple (two) applications as far as they share the same configuration: transmission format, baud rate, header and guard settings. The SHIFT input pin (multiplexed with the LED output) is used to select between the two crypt keys.

A logic 1 on the $\overline{\text{SHIFT}}$ input pin selects the first crypt key.

A logic 0 on the $\overline{\text{SHIFT}}$ input pin will select the second crypt key.

3.0 DEVICE OPERATION

The HCS362 will wake-up upon detecting a switch closure and then delay for switch debounce (Figure 3-1). The synchronization information, fixed information and switch information will be encrypted to form the hopping code. The encrypted or hopping code portion of the transmission will change every time a button is pressed, even if the same button is pushed again. Keeping a button pressed for a long time will result in the same code word being transmitted until the button is released or time-out occurs.

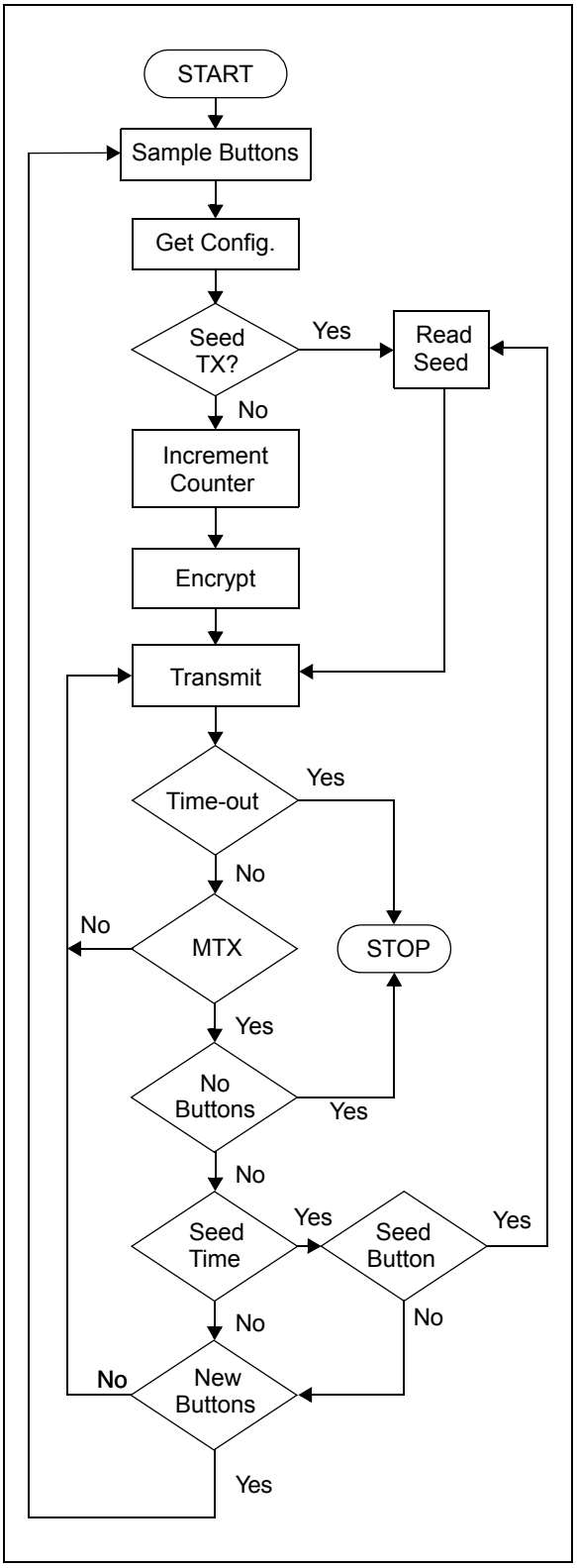
The time-out time can be selected with the Time-out (TIMOUT[0..1]) configuration option. This option allows the time-out to be disabled or set to 0.8 s, 3.2 s or 25.6 s. When a time-out occurs, the device will go into SLEEP mode to protect the battery from draining when a button gets stuck.

If in the transmit process, it is detected that a new button is pressed, the current code word will be aborted. A new code word will be transmitted and the time-out counter will RESET. If all the buttons are released, the minimum code words will be completed. The minimum code words can be set to 1,2,4 or 8 using the Minimum Code Words (MTX[0..1]) configuration option. If the time for transmitting the minimum code words is longer than the time-out time, the device will not complete the minimum code words.

Note:	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.

A code that has been transmitted will not occur again for more than 64K transmissions. This will provide more than 18 years of typical use before a code is repeated based on 10 operations per day. Overflow information programmed into the encoder can be used by the decoder to extend the number of unique transmissions to more than 192K.

FIGURE 3-1: BASIC FLOW DIAGRAM OF THE DEVICE OPERATION



3.1 Transmission Modulation Format

The HCS362 transmission is made up of several code words. Each code word starts with a preamble and a header, followed by the data (see Figure 3-1 and Figure 3-2).

The code words are separated by a **Guard Time** that can be set to 0 ms, 6.4 ms, 25.6 ms or 76.8 ms with the Guard Time Select (GUARD[0..1]) configuration option. All other timing specifications for the modulation formats are based on a basic timing element (TE). This **Timing Element** can be set to 100 µs, 200 µs, 400 µs or 800 µs with the Baud Rate Select (BSEL[0..1]) configuration option. The **Header Time** can be set to 3 TE or 10 TE with the Header Select (HEADER) Configuration option.

There are two different modulation formats available on the HCS362 that can be set according to the Modulation Select (MOD) configuration option:

- Pulse Width Modulation (PWM)
- Manchester Encoding

The various formats are shown in Figure 3-3 and Figure 3-4.

FIGURE 3-2: CODE WORD TRANSMISSION SEQUENCE

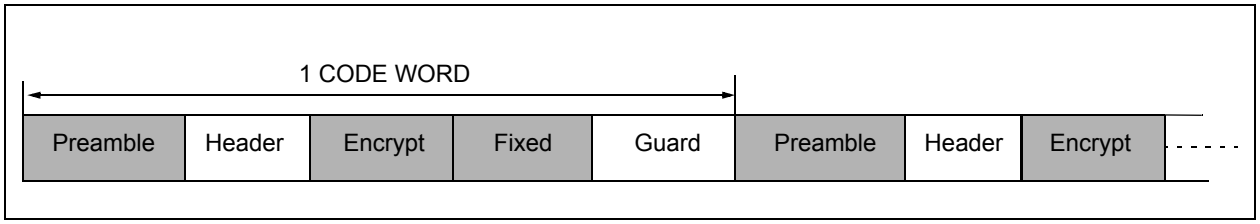


FIGURE 3-3: TRANSMISSION FORMAT (PWM)

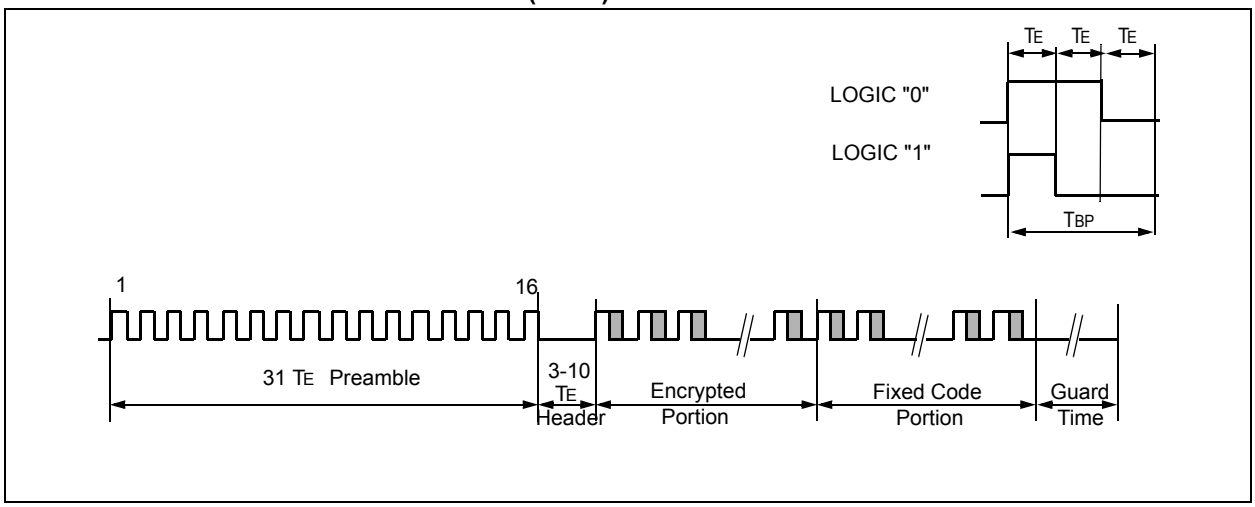
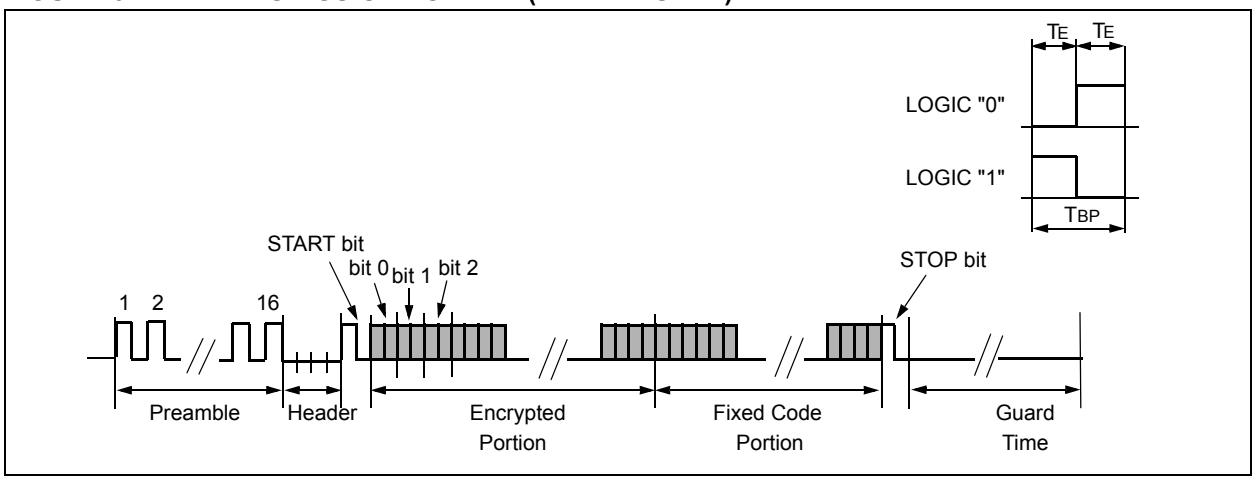


FIGURE 3-4: TRANSMISSION FORMAT (MANCHESTER)



3.1.1 CODE HOPPING DATA

The hopping portion is calculated by encrypting the counter, discrimination value and function code with the Encoder Key (KEY). The counter is a 16-bit counter. The discrimination value is 10 bits long and there are 2 counter overflow bits (OVR) that are cleared when the counter wraps to 0. The rest of the 32 bits are made up of the function code also known as the button inputs.

3.1.2 FIXED CODE DATA

The 32 bits of fixed code consist of 28 bits of the serial number (SER) and another copy of the function code. This can be changed to contain the whole 32-bit serial number with the Extended Serial Number (XSER) configuration option.

3.1.3 STATUS INFORMATION

The status bits will always contain the output of the Low Voltage detector (VLOW), the Cyclic Redundancy Check (CRC) bits (or TIME bits depending on CTSEL) and the Button Queue information.

3.1.3.1 Low Voltage Detector Status (VLOW)

The output of the low voltage detector is transmitted with each code word. If VDD drops below the selected voltage, a logic '1' will be transmitted. The output of the detector is sampled before each code word is transmitted.

3.1.3.2 Button Queue Information (QUEUE)

The queue bits indicate a button combination was pressed again within 2 s after releasing the previous activation. Queuing or repeated pressing of the same buttons (or button combination) is detected by the HCS362 button debouncing circuitry.

The Queue bits are added as the last two bits of the standard code word. The queue bits are a 2-bit counter that does not wrap. The counter value starts at '00b' and is incremented, if a button is pushed within 2 s of the previous button press. The current code word is terminated when the buttons are queued. This allows additional functionality for repeated button presses.

The button inputs are sampled every 6.4 ms during this 2 s period.

00 - first activation

01 - second activation

10 - third activation

11 - from fourth activation on

3.1.3.3 Cyclic Redundancy Check (CRC)

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

EQUATION 3-1: CRC Calculation

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

and

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

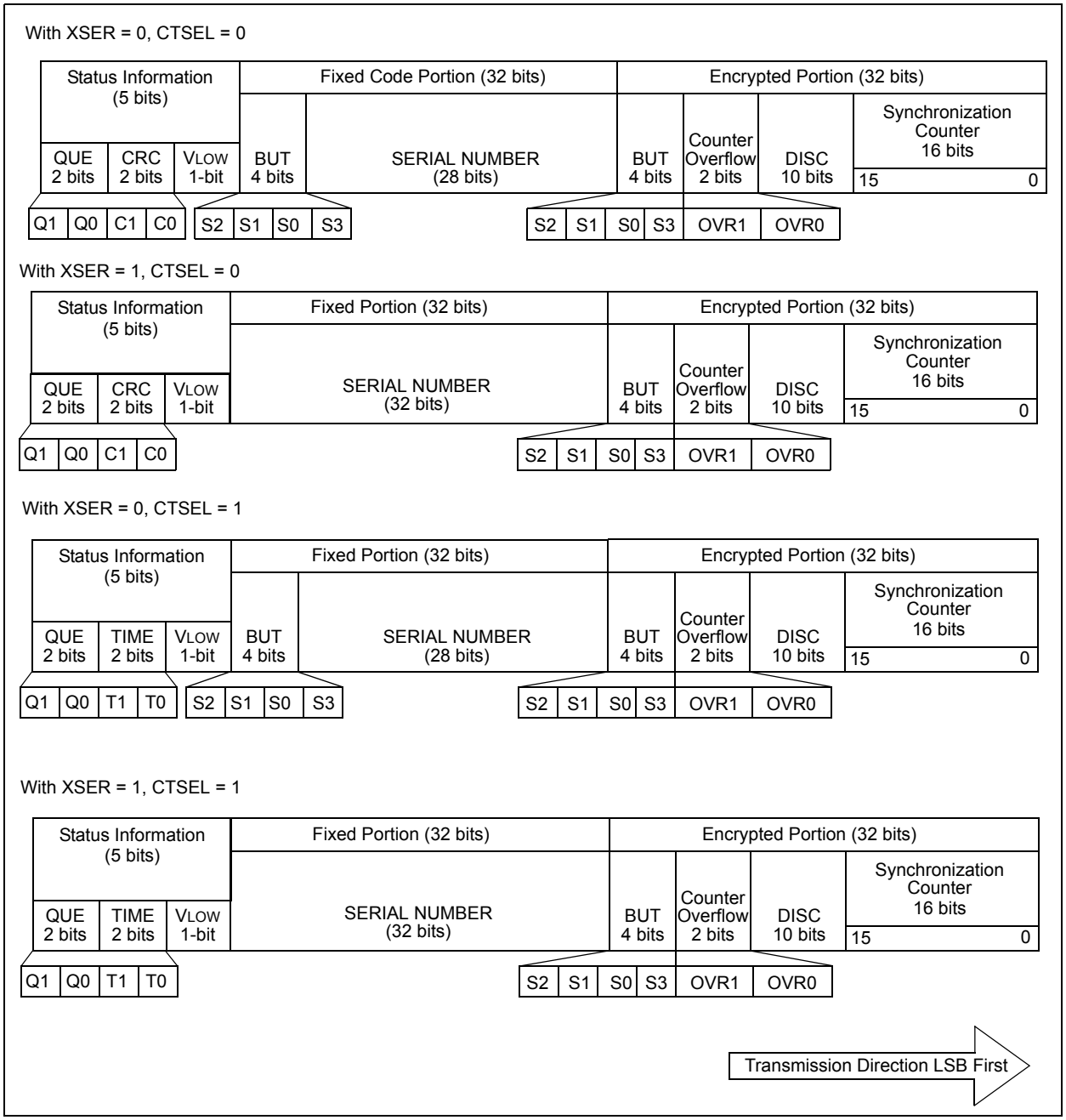
with

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

and Di_n the nth transmission bit $0 \leq n \leq 64$

Note:	The CRC may be wrong when the battery voltage is around either of the VLOW trip points. This may happen because VLOW is sampled twice each transmission, once for the CRC calculation (PWM is LOW) and once when VLOW is transmitted (PWM is HIGH). VDD tends to move slightly during a transmission which could lead to a different value for VLOW being used for the CRC calculation and the transmission. **Work around:** If the CRC is incorrect, recalculate for the opposite value of VLOW.

FIGURE 3-5: CODE WORD DATA FORMAT



3.1.4 MINIMUM CODE WORDS

MTX[0..1] configuration bits selects the minimum number of code words that will be transmitted. If the button is released after 1.6 s (or greater) and MTX code words have been transmitted, the code word being transmitted will be terminated. The possible values are:

00 - 1

01 - 2

10 - 4

11 - 8

3.1.5 TIME BITS

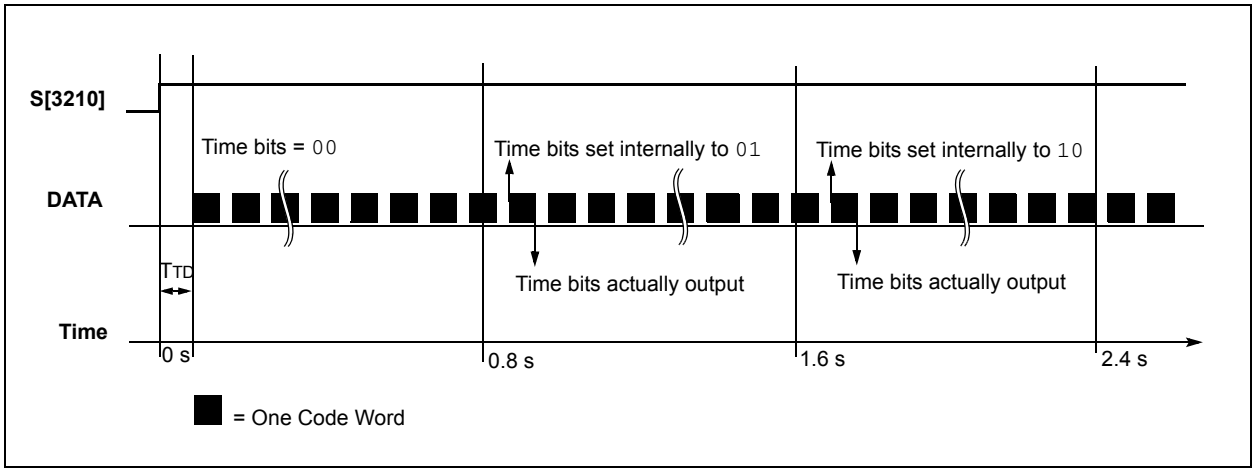
The time bits indicate the duration that the inputs were activated:

00 - immediate
01 - after 0.8 s
10 - after 1.6 s
11 - after 2.4 s

The TIME bits are incremented every 0.8 s and does not wrap once it reaches '11'.

Time information is alternative to the CRC bits availability and is selected by the CTSEL configuration bit.

FIGURE 3-6: TIME BITS OPERATION

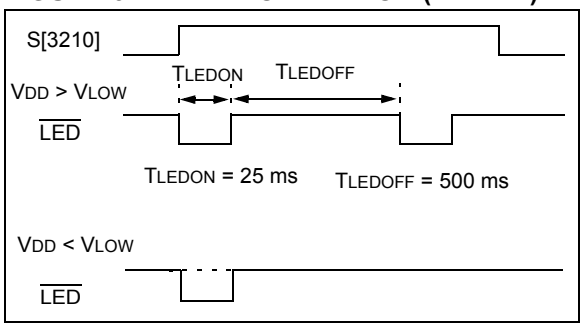


3.2 $\overline{\text{LED}}$ Output

The $\overline{\text{LED}}$ pin will be driven LOW periodically while the HCS362 is transmitting data, in order to switch on an external LED.

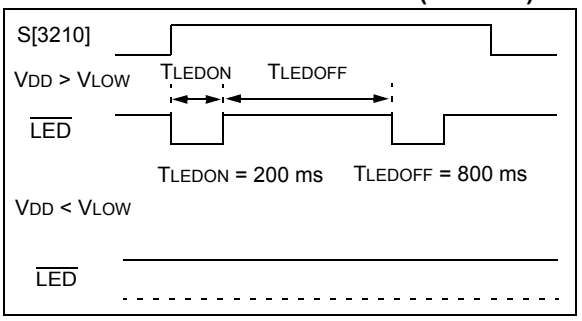
The duty cycle (TLEDON/TLEDOFF) can be selected between two possible values by the configuration option (LED).

FIGURE 3-7: LED OPERATION (LED = 1)



The same configuration option determines whether when the VDD Voltage drops below the selected VLOW trip point, the LED will blink only once or stop blinking.

FIGURE 3-8: LED OPERATION (LED = 0)



Note: When the HCS362 encoder is used as a Dual Encoder the $\overline{\text{LED}}$ pin is used as a $\overline{\text{SHIFT}}$ input (Figure 2-2). In such a configuration the LED is always ON during transmission. To keep power consumption low, it is recommended to use a series resistor of relatively high value. VLOW information is not available when using the second Encryption Key.

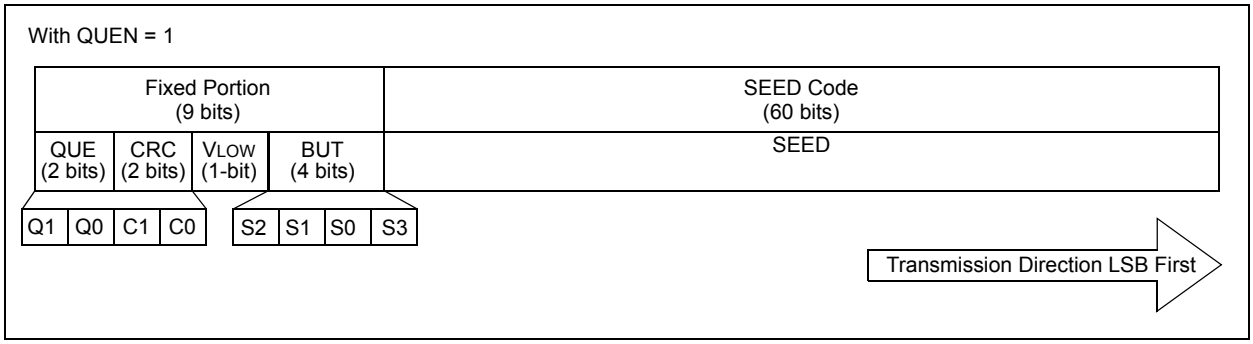
3.3 Seed Code Word Data Format

A seed transmission transmits a code word that consists of 60 bits of fixed data that is stored in the EEPROM. This can be used for secure learning of encoders or whenever a fixed code transmission is required. The seed code word further contains the function code and the status information (VLOW, CRC and QUEUE) as configured for normal code hopping code words. The seed code word format is shown in Figure 3-9. The function code for seed code words is always '1111b'.

Seed code words can be configured as follows:

- Enabled permanently.
- Disabled permanently.
- Enabled until the synchronization counter is greater than 7Fh, this configuration is often referred to as **Limited Seed**.
- The time before the seed code word is transmitted can be set to 1.6 s or 3.2 s, this configuration is often referred to as **Delayed Seed**. When this option is selected, the HCS362 will transmit a code hopping code word for 1.6 s or 3.2 s, before the seed code word is transmitted.

FIGURE 3-9: SEED CODE WORD FORMAT



3.3.1 SEED OPTIONS

The button combination (S[3210]) for transmitting a Seed code word can be selected with the Seed and SeedC (SEED[0..1] and SEEDC) configuration options as shown in Table 3-1 and Table 3-2:

TABLE 3-1: SEED OPTIONS (SEEDC = 0)

SEED	Seed S[3210]	1.6 s Delayed Seed S[3210]
00	–	–
01	0101*	0001*
10	0101	0001
11	0101	–
Note:	*Limited Seed	

TABLE 3-2: SEED OPTIONS (SEEDC = 1)

SEED	Seed S[3210]	3.2 s Delayed Seed S[3210]
00	–	–
01	1001*	0011*
10	1001	0011
11	1001	–
Note:	*Limited Seed	

Example A): Selecting SEEDC = 1 and SEED = 11: makes SEED transmission available every time the combination of buttons S3 and S0 is pressed simultaneously, but Delayed Seed mode is not available.

Example B): Selecting SEEDC = 0 and SEED = 01: makes SEED transmission available only for a limited time (only up to 128 times). The combination of buttons S2 and S0 produces an immediate transmission of the SEED code. Pressing and holding for more than 1.6 seconds the S0 button alone, produces the SEED code word transmission (Delayed Seed).

3.4 RF Enable and PLL Interface

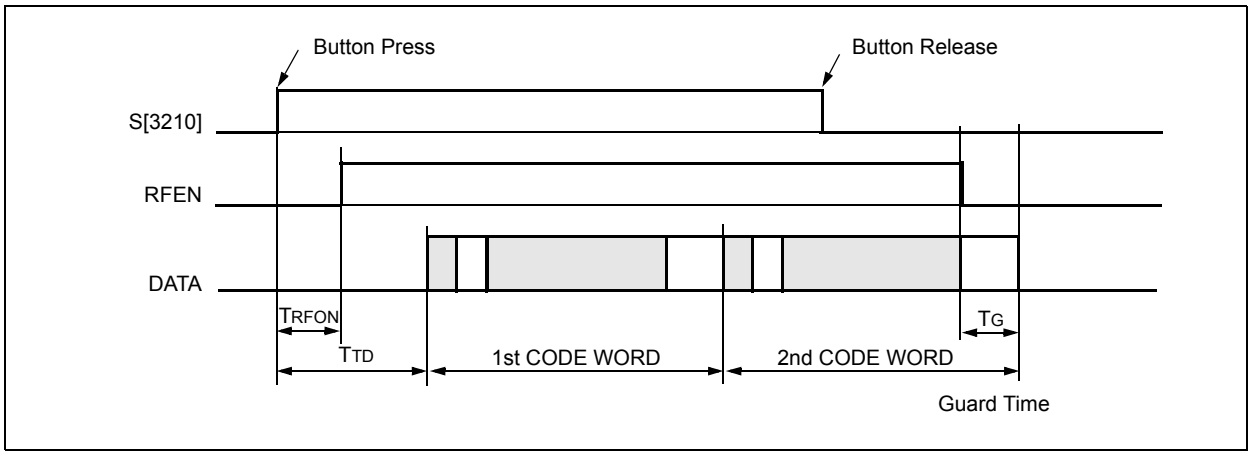
The S3/RFEN pin of the HCS362 can be configured to function as an RF Enable output signal. This is selected by the RF Enable Output (RFEN) configuration option. When enabled, this pin will be driven HIGH before data is transmitted through the DATA pin.

The RF Enable and DATA output are synchronized so to interface with RF PLL circuits operating in ASK mode. Figure 3-10 shows the startup sequence. The RFEN signal will go LOW at the end of the last code word, including the Guard time.

When the RF Enable output is selected, the S3 pin can still be used as a button input. The debouncing logic will be affected though, considerably reducing the responsiveness of the button input.

Note: When the RF Enable output feature is used and a four (or more) buttons input configuration is required, the use of a scheme similar to Figure 2-1 (scheme C) is recommended.

FIGURE 3-10: PLL INTERFACE



4.0 EEPROM MEMORY ORGANIZATION

The HCS362 contains 288 bits (18 x 16-bit words) of EEPROM memory (Table 4-1). This EEPROM array is used to store the encryption key information and synchronization value. Further descriptions of the memory array is given in the following sections.

TABLE 4-1: EEPROM MEMORY MAP

Word Address	Field	Description
0	KEY1_0	64-bit Encryption Key1 (Word 0) LSB
1	KEY1_1	64-bit Encryption Key1 (Word 1)
2	KEY1_2	64-bit Encryption Key1 (Word 2)
3	KEY1_3	64-bit Encryption Key1 (Word 3) MSB
4	KEY2_0	64-bit Encryption Key2 (Word 0) LSB
5	KEY2_1	64-bit Encryption Key2 (Word 1)
6	KEY2_2	64-bit Encryption Key2 (Word 2)
7	KEY2_3	64-bit Encryption Key2 (Word 3) MSB
8	SEED_0	Seed value (Word 0) LSB
9	SEED_1	Seed value (Word 1)
10	SEED_2	Seed value (Word 2)
11	SEED_3	Seed value (Word 3) MSB
12	CONFIG_0	Configuration Word (Word 0)
13	CONFIG_1	Configuration Word (Word 1)
14	SERIAL_0	Serial Number (Word 0) LSB
15	SERIAL_1	Serial Number (Word 1) MSB
16	SYNC	Synchronization counter
17	RES	Reserved – Set to zero

4.1 KEY_0 - KEY_3 (64-bit Crypt Key)

The 64-bit crypt key is used to create the encrypted message transmitted to the receiver. This key is calculated and programmed during production using a key generation algorithm. The key generation algorithm may be different from the KEELOQ algorithm. Inputs to the key generation algorithm are typically the transmitter's serial number and the 64-bit manufacturer's code. While the key generation algorithm supplied from Microchip is the typical method used, 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.

4.2 SYNC (Synchronization Counter)

This is the 16-bit synchronization value that is used to create the hopping code for transmission. This value will be incremented after every transmission.

4.3 SEED_0, SEED_1, SEED_2, and SEED 3 (Seed Word)

This is the four word (60 bits) seed code that will be 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.

Note:	Upper four Significant bits of SEED_3 contains extra configuration information (see Table 4-4).

4.4 SERIAL_0, SERIAL_1 (Encoder Serial Number)

SER_0 and SER_1 are the lower and upper words of the device serial number, respectively. There are 32 bits allocated for the serial number and a selectable configuration bit determines whether 32 or 28 bits will be transmitted. The serial number is meant to be unique for every transmitter.

4.5 Configuration Words

There are 36 configuration bits stored in the EEPROM array. They are used by the device to determine transmission speed, format, delays and Guard times. They are grouped in three Configuration Words: CONFIG_0, CONFIG_1 and the upper nibble of the SEED_3 word. A description of each of the bits follows this section.

TABLE 4-2: CONFIG_0

Bit Address	Field	Description	Values
0	OSC_0	Oscillator adjust	0000 - nominal 1000 - fastest 0111 - slowest
1	OSC_1		
2	OSC_2		
3	OSC_3		
4	VLOW_0	VLOW select	nominal values 000 - 2.0V 100 - 4.0V 001 - 2.1V 101 - 4.2V 010 - 2.2V 110 - 4.4V 011 - 2.3V 111 - 4.6V
5	VLOW_1		
6	VLOW_2		
7	BSEL_0	Bitrate select	00 - TE = 100 µs 01 - TE = 200 µs 10 - TE = 400 µs 11 - TE = 800 µs
8	BSEL_1		
9	MTX_0	Minimum number of code words	00 - 1 01 - 2 10 - 4 11 - 8
10	MTX_1		
11	GUARD_0	Guard time select	00 - 0 ms (1 TE) 01 - 6.4 ms + 2 TE 10 - 25.6 ms + 2 TE 11 - 76.8 ms + 2 TE
12	GUARD_1		
13	TIMOUT_0	Time-out select	00 - No Time-out 01 - 0.8 s to 0.8 s + 1 code word 10 - 3.2 s to 3.2 s + 1 code word 11 - 25.6 s to 25.6 s + 1 code word
14	TIMOUT_1		
15	CTSEL	CTSEL	0 = TIME bits 1 = CRC bits

4.5.1 OSC

The internal oscillator can be tuned to ±10%. (0000 selects the nominal value, 1000 the fastest value and 0111 the slowest). When programming the device, it is the programmer's responsibility to determine the optimal calibration value.

4.5.2 VLOW[0..2]

The low voltage threshold can be programmed to be any of the values shown in the following table:

4.5.3 BSEL[0..1]

The basic timing element TE, determines the actual transmission Baud Rate. This translates to different code word lengths depending on the encoding format selected (Manchester or PWM), the Header length selection and the Guard time selection, from approximately 40 ms up to 220 ms. Refer to Table 8-4 and Table 8-5 for a more complete description.

4.5.4 MTX[0..1]

MTX selects the minimum number of code words that will be transmitted. A minimum of 1, 2, 4 or 8 code words will be transmitted.

Note:	If MTX and BSEL settings in combination require a transmission sequence to exceed the TIMOUT setting, TIMOUT will take priority.

4.5.5 GUARD

The Guard time between code words can be set to 0 ms, 6.4 ms, 25.6 ms and 76.8 ms. If during a series of code words, the output changes from Hopping Code to Seed the Guard time will increase by 3 x TE.

4.5.6 TIMOUT[0..1]

The transmission time-out can be set to 0.8 s, 3.2 s, 25.6 s or no time-out. After the time-out period, the encoder will stop transmission and enter a low power Shutdown mode.

TABLE 4-3: CONFIG_1

Bit Address	Field	Description	Values
0	DISC_0	Discrimination bits	DISC[9:0]
1	DISC_1		
2	DISC_2		
...	...		
8	DISC_8		
9	DISC_9		
10	OVR_0	Overflow	OVR[1:0]
11	OVR_1		
12	XSER	Extended Serial Number	0 - Disable 1 - Enable
13	SEEDC	Seed Control	0 = Seed transmission on: S[3210] = 0001 (delay 1.6 s) S[3210] = 0101 (immediate) 1 = Seed transmission on: S[3210] = 0011 (delay 3.2 s) S[3210] = 1001 (immediate)
14	SEED_0	Seed options	00 - No Seed 01 - Limited Seed (Permanent and Delayed) 10 - Permanent and Delayed Seed 11 - Permanent Seed only
15	SEED_1		

4.5.7 DISC[0..9]

The discrimination bits are used to validate the decrypted code word. The discrimination value is typically programmed with the 10 Least Significant bits of the serial number or a fixed value.

4.5.8 OVR[0..1]

The overflow bits are used to extend the possible code combinations to 192K. If the overflow bits are not going to be used they can be programmed to zero.

4.5.9 XSER

If XSER is enabled a 32-bit serial number is transmitted. If XSER is disabled a 28-bit serial number and a 4-bit function code are transmitted.

4.5.10 SEED[0..1]

The seed value which is transmitted on key combinations (0011) and (1001) can be disabled, enabled or enabled for a limited number of transmissions determined by the initial counter value.

In limited Seed mode, the device will output the seed if the sync counter (Section 4.2) is from 00hex to 7Fhex. For a counter higher than 7F, a normal hopping code will be output.

Note:	Whenever a SEED code word is output, the 4 function bits (Figure 8-4) will be set to all ones [1,1,1,1].

4.5.11 SEEDC

SEEDC selects between seed transmission on 0001 and 0101 (SEEDC = 0) and 0011 and 1001 (SEEDC = 1). The delay before seed transmission is 1.6 s for (SEEDC = 0) and 3.2 s for (SEEDC = 1).

TABLE 4-4: SEED_3

Bit Address	Field	Description	Values
0	SEED_48	Seed Most Significant word	—
1	SEED_49		
2	SEED_50		
...	...		
9	SEED_57		
10	SEED_58		
11	SEED_59		
12	LED	LED output timing	0 = VBOT>VLOW LED blink 200/800 ms VBOT<VLOW LED not blinking 1 = VBOT>VLOW LED blink 25/500 ms VBOT<VLOW LED blink once
13	MOD	Modulation Format	0 = PWM 1 = MANCHESTER
14	RFEN	RF Enable/S3 multiplexing	0 - Enabled (S3 only sensed 2 seconds after the last button is released) 1 - Disabled (S3 same as other S inputs)
15	HEADER	Header Length	0 = short Header, TH = 3 x TE 1 = standard Header, TH = 10 x TE

4.5.12 HEADER

When PWM mode is selected the header length (low time between preamble and data bits start) can be set to 10 x TE or 3 x TE. The 10 x TE mode is recommended for compatibility with previous KEELOQ encoder models. In Manchester mode, the header length is fixed and set to 4 x TE.

4.5.13 RFEN

RFEN selects whether the RFEN output is enabled or disabled. If enabled, S3 is only sampled 2 s after the last button is released and at the start of the first transmission. If disabled S3 functions the same as the other S inputs.

4.6 SYNCHRONOUS MODE

In Synchronous mode, the code word can be clocked out on DATA using S2 as a clock. To enter Synchronous mode, DATA and S0 must be taken HIGH and then S2 is taken HIGH. After Synchronous mode is entered, S0 must be taken LOW. The data is clocked out on DATA on every rising edge of S2. Auto-shutoff timer is not disabled in Synchronous mode. This can be used to implement RF testing.

FIGURE 4-1: SYNCHRONOUS TRANSMISSION MODE

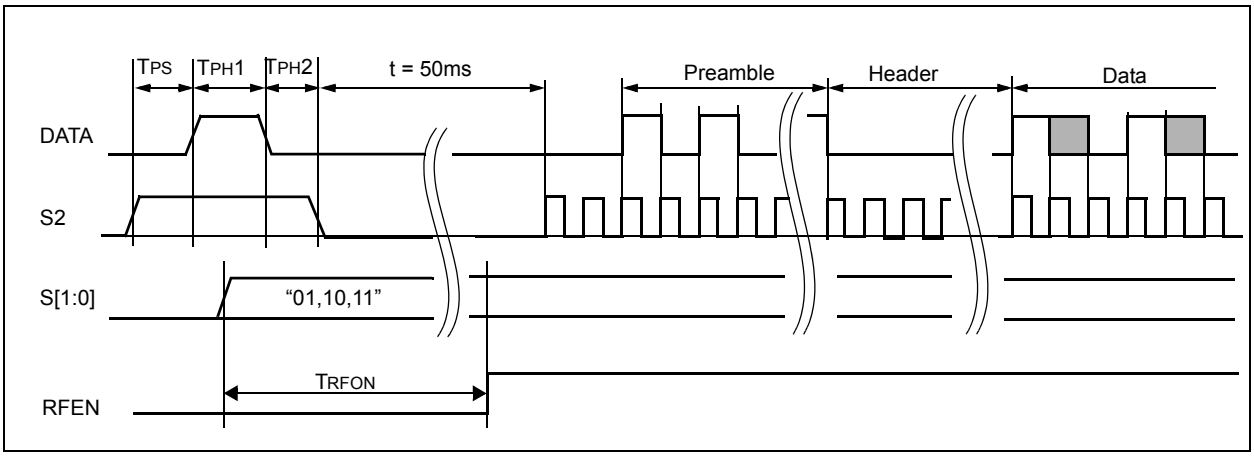
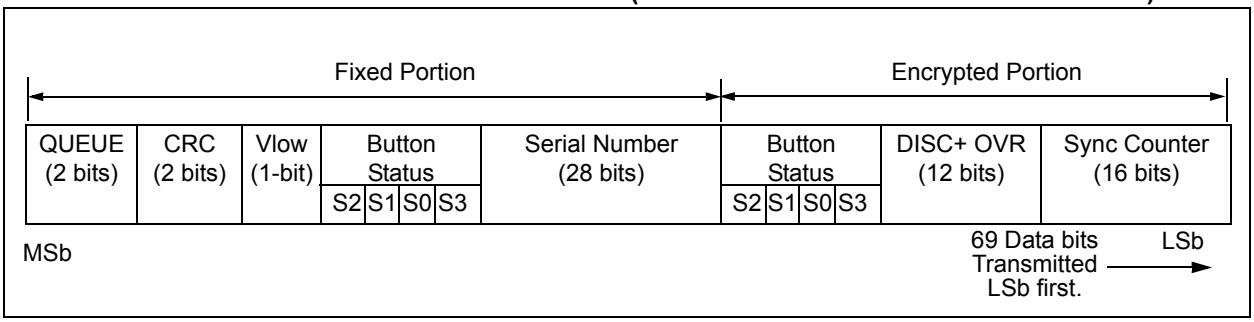


FIGURE 4-2: CODE WORD ORGANIZATION (SYNCHRONOUS TRANSMISSION MODE)



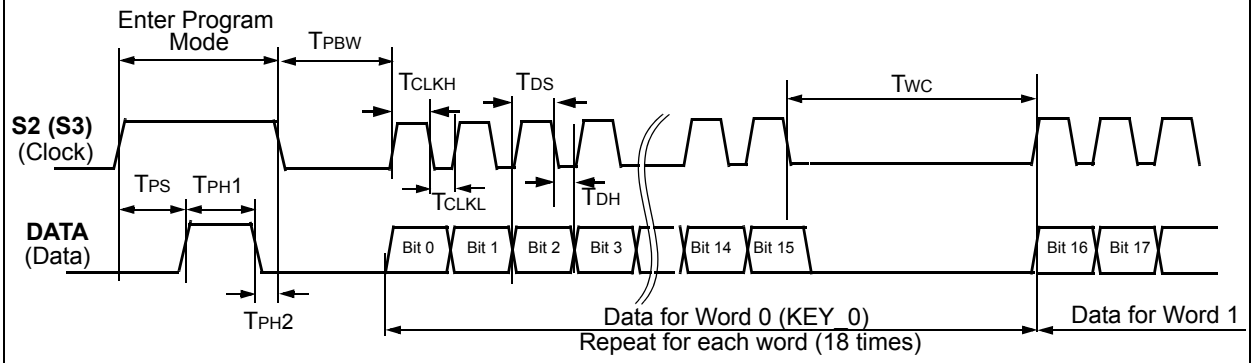
5.0 PROGRAMMING THE HCS362

When using the HCS362 in a system, the user will have to program some parameters into the device, including the serial number and the secret key before it can be used. The programming cycle allows the user to input all 288 bits in a serial data stream, which are then stored internally in EEPROM. Programming will be initiated by forcing the DATA line HIGH, after the S2 line has been held HIGH for the appropriate length of time (Table 5-1 and Figure 5-1). After the Program mode is entered, a delay must be provided to the device for the automatic bulk write cycle to complete. This will write all locations in the EEPROM to an all zeros pattern including the OSC calibration bits.

The device can then be programmed by clocking in 16 bits at a time, using S2 as the clock line and DATA as the data in-line. After each 16-bit word is loaded, a programming delay is required for the internal program cycle to complete. This delay can take up to Twc. At the end of the programming cycle, the device can be verified (Figure 5-2) by reading back the EEPROM. Reading is done by clocking the S2 line and reading the data bits on DATA. For security reasons, it is not possible to execute a verify function without first programming the EEPROM. **A Verify operation can only be done once, immediately following the Program cycle**.

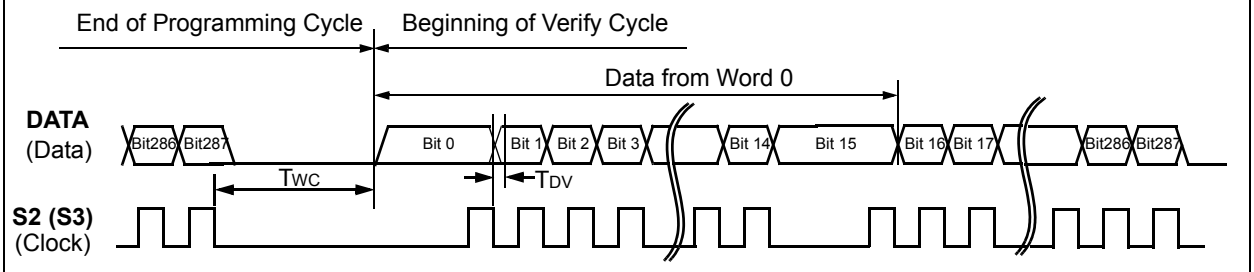
Note:	To ensure that the device does not accidentally enter Programming mode, PWM should never be pulled high by the circuit connected to it. Special care should be taken when driving PNP RF transistors.

FIGURE 5-1: PROGRAMMING WAVEFORMS



Note 1: Unused button inputs to be held to ground during the entire programming sequence.
2: The VDD pin must be taken to ground after a Program/Verify cycle.

FIGURE 5-2: VERIFY WAVEFORMS



Note: If a Verify operation is to be done, then it must immediately follow the Program cycle.

TABLE 5-1: PROGRAMMING/VERIFY TIMING REQUIREMENTS

V_{DD} = 5.0V ± 10% 25 °C ± 5 °C				
Parameter	**Symbol**	**Min.**	**Max.**	**Units**
Program mode setup time	T_{PS}	3.5	4.5	ms
Hold time 1	T_{PH1}	3.5	—	ms
Hold time 2	T_{PH2}	50	—	µs
Bulk Write time	T_{PBW}	4.0	—	ms
Program delay time	T_{PROG}	4.0	—	ms
Program cycle time	T_{WC}	50	—	ms
Clock low time	T_{CLKL}	50	—	µs
Clock high time	T_{CLKH}	50	—	µs
Data setup time	T_{DS}	0	—	µs
Data hold time	T_{DH}	30	—	µs
Data out valid time	T_{DV}	—	30	µs

6.0 INTEGRATING THE HCS362 INTO A SYSTEM

Use of the HCS362 in a system requires a compatible decoder. This decoder is typically a microcontroller with compatible firmware. Microchip will provide (via a license agreement) firmware routines that accept transmissions from the HCS362 and decrypt the hopping code portion of the data stream. These routines provide system designers the means to develop their own decoding system.

6.1 Learning a Transmitter to a Receiver

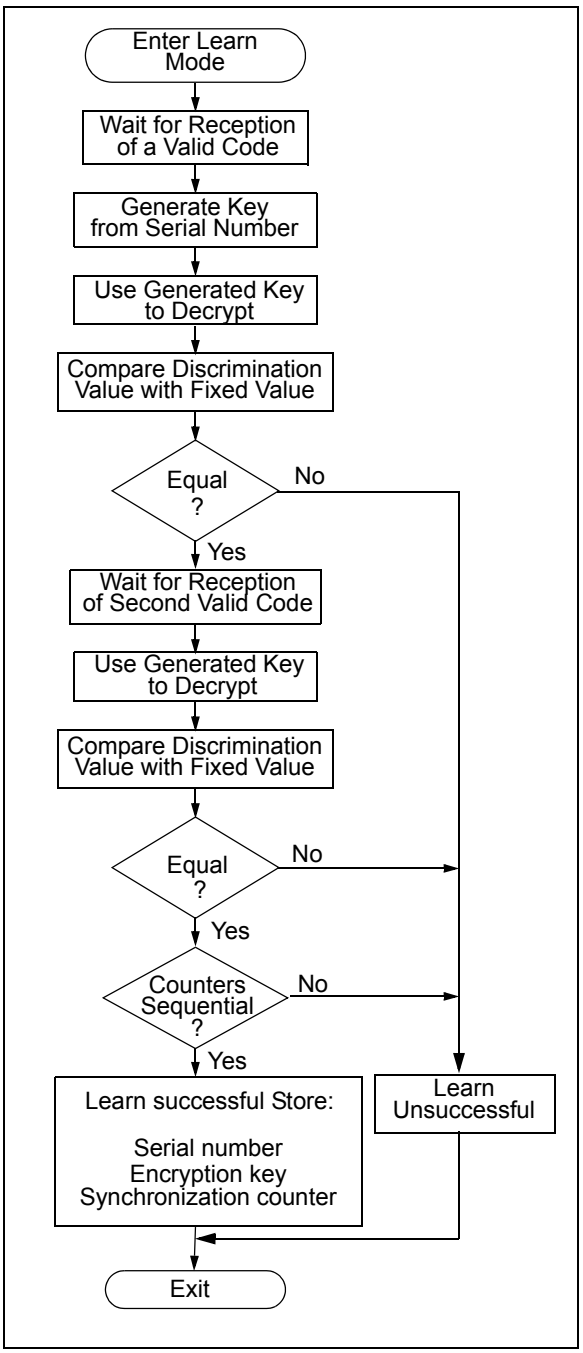
A transmitter must first be 'learned' by a decoder before its use is allowed in the system. Several learning strategies are possible, Figure 6-1 details a typical learn sequence. Core to each, the decoder must minimally store each learned transmitter's serial number and current synchronization counter value in EEPROM. Additionally, the decoder typically stores each transmitter's unique crypt key. The maximum number of learned transmitters will therefore be relative to the available EEPROM.

A transmitter's serial number is transmitted in the clear but the synchronization counter only exists in the code word's encrypted portion. The decoder obtains the counter value by decrypting using the same key used to encrypt the information. The KEELOQ algorithm is a symmetrical block cipher so the encryption and decryption keys are identical and referred to generally as the crypt key. The encoder receives its crypt key during manufacturing. The decoder is programmed with the ability to generate a crypt key as well as all but one required input to the key generation routine; typically the transmitter's serial number.

Figure 6-1 summarizes a typical learn sequence. The decoder receives and authenticates a first transmission; first button press. Authentication involves generating the appropriate crypt key, decrypting, validating the correct key usage via the discrimination bits and buffering the counter value. A second transmission is received and authenticated. A final check verifies the counter values were sequential; consecutive button presses. If the learn sequence is successfully complete, the decoder stores the learned transmitter's serial number, current synchronization counter value and appropriate crypt key. From now on the crypt key will be retrieved from EEPROM during normal operation instead of recalculating it for each transmission received.

Certain learning strategies have been patented and care must be taken not to infringe.

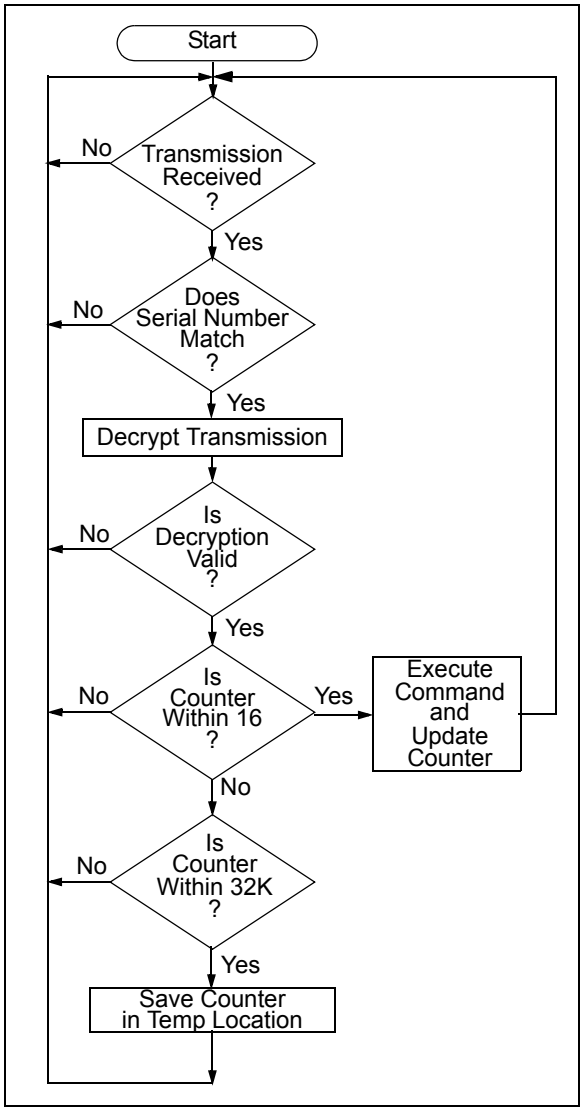
FIGURE 6-1: TYPICAL LEARN SEQUENCE



6.2 Decoder Operation

Figure 6-2 summarizes normal decoder operation. The decoder waits until a transmission is received. The received serial number is compared to the EEPROM table of learned transmitters to first determine if this transmitter's use is allowed in the system. If from a learned transmitter, the transmission is decrypted using the stored crypt key and authenticated via the discrimination bits for appropriate crypt key usage. If the decryption was valid the synchronization value is evaluated.

FIGURE 6-2: TYPICAL DECODER OPERATION



6.3 Synchronization with Decoder (Evaluating the Counter)

The KEELOQ technology patent scope includes a sophisticated synchronization technique that does not require the calculation and storage of future codes. The technique securely blocks invalid transmissions while providing transparent resynchronization to transmitters inadvertently activated away from the receiver.

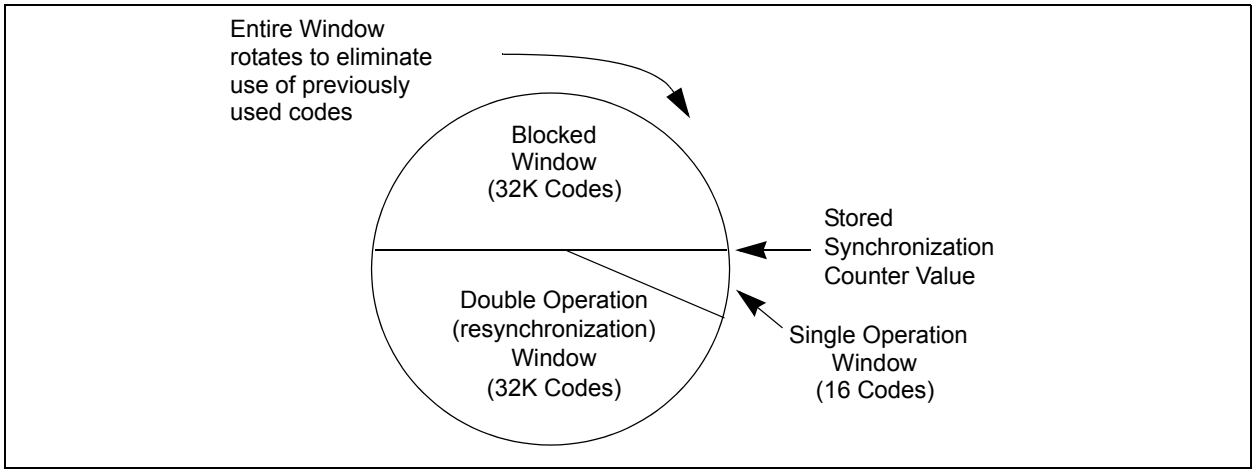
Figure 6-3 shows a 3-partition, rotating synchronization window. The size of each window is optional but the technique is fundamental. Each time a transmission is authenticated, the intended function is executed and the transmission's synchronization counter value is stored in EEPROM. From the currently stored counter value there is an initial "Single Operation" forward window of 16 codes. If the difference between a received synchronization counter and the last stored counter is within 16, the intended function will be executed on the single button press and the new synchronization counter will be stored. Storing the new synchronization counter value effectively rotates the entire synchronization window.

A "Double Operation" (resynchronization) window further exists from the Single Operation window up to 32K codes forward of the currently stored counter value. It is referred to as "Double Operation" because a transmission with synchronization counter value in this window will require an additional, sequential counter transmission prior to executing the intended function. Upon receiving the sequential transmission the decoder executes the intended function and stores the synchronization counter value. This resynchronization occurs transparently to the user as it is human nature to press the button a second time if the first was unsuccessful.

The third window is a "Blocked Window" ranging from the double operation window to the currently stored synchronization counter value. Any transmission with synchronization counter value within this window will be ignored. This window excludes previously used, perhaps code-grabbed transmissions from accessing the system.

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 6-3: SYNCHRONIZATION WINDOW



7.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

7.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.