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

HCS512

KEELOQ® Code Hopping Decoder

FEATURES

Security

- Secure storage of Manufacturer's Code
- Secure storage of transmitter's keys
- Up to four transmitters can be learned
- KEELOQ® code hopping technology
- Normal and secure learning mechanisms

Operating

- 4.0V – 6.0V operation
- 4 MHz external RC oscillator
- Learning indication on LRNOUT
- Auto baud rate detection
- Power saving SLEEP mode

Other

- Stand-alone decoder
- On-chip EEPROM for transmitter storage
- Four binary function outputs–15 functions
- 18-pin DIP/SOIC package

Typical Applications

- Automotive remote entry systems
- Automotive alarm systems
- Automotive immobilizers
- Gate and garage openers
- Electronic door locks
- Identity tokens
- Burglar alarm systems

Compatible Encoders

All KEELOQ encoders and transponders configured for the following setting:

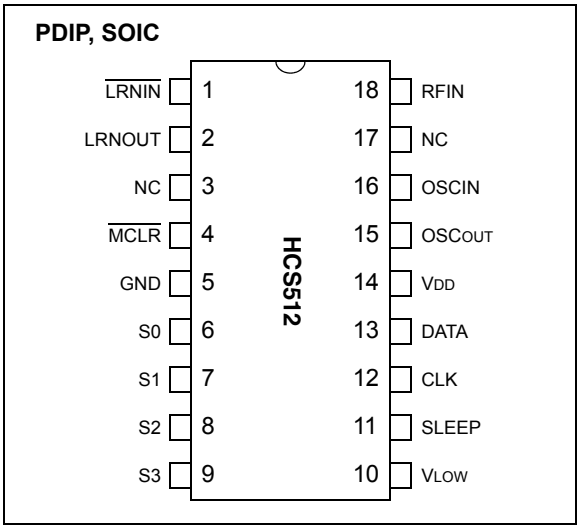
- PWM modulation format (1/3-2/3)
- TE in the range from 100 µs to 400 µs
- 10 x TE Header
- 28-bit Serial Number
- 16-bit Synchronization counter
- Discrimination bits equal to Serial Number 8 LSbs
- 66- to 69-bit length code word.

DESCRIPTION

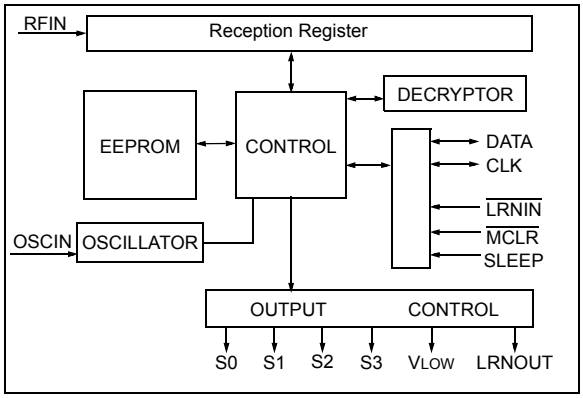
The Microchip Technology Inc. HCS512 is a code hopping decoder designed for secure Remote Keyless Entry (RKE) systems. The HCS512 utilizes the patented KEELOQ code hopping system and high security learning mechanisms to make this a canned solution when used with the HCS encoders to implement a unidirectional remote keyless entry system.

PACKAGE TYPE

PDIP, SOIC



BLOCK DIAGRAM



The Manufacturer's Code, transmitter keys, and synchronization information are stored in protected on-chip EEPROM. The HCS512 uses the DATA and CLK inputs to load the Manufacturer's Code which cannot be read out of the device.

© 2011 Microchip Technology Inc.

The HCS512 operates over a wide voltage range of 3.0 volts to 6.0 volts. The decoder employs automatic baud rate detection which allows it to compensate for wide variations in transmitter data rate. The decoder contains sophisticated error checking algorithms to ensure only valid codes are accepted.

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 8-2).
- **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 8-1).
- **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.

1.1 HCS Encoder Overview

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

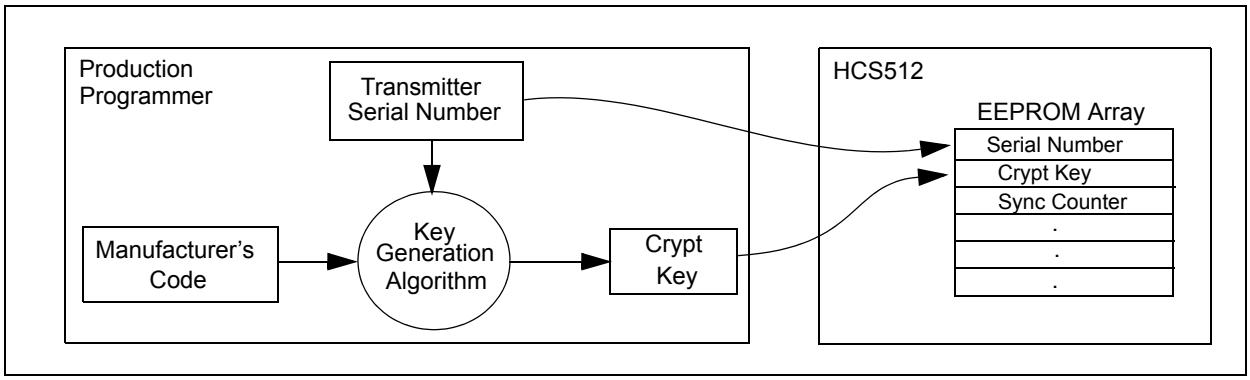
- A crypt key that is generated at the time of production
- A 16-bit synchronization counter value
- A 28-bit serial number which is meant to be unique for every encoder

The manufacturer programs the serial number for each encoder at the time of production, while the 'Key Generation Algorithm' generates the crypt key (Figure 1-1). Inputs to the key generation algorithm typically consist of the encoder's serial number and a 64-bit manufacturer's code, which the manufacturer creates.

Note:	The manufacturer code is a pivotal part of the system's overall security. Consequently, all possible precautions must be taken and maintained for this code.

© 2011 Microchip Technology Inc.

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 8.2.

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 HCS512 based transmitter. Section 5.0 provides detail on integrating the HCS512 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)

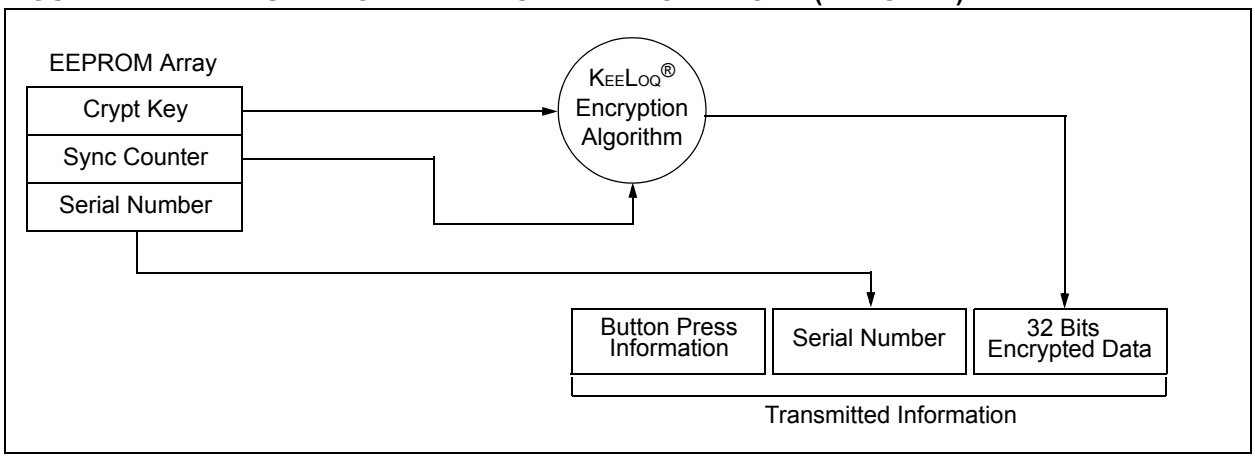
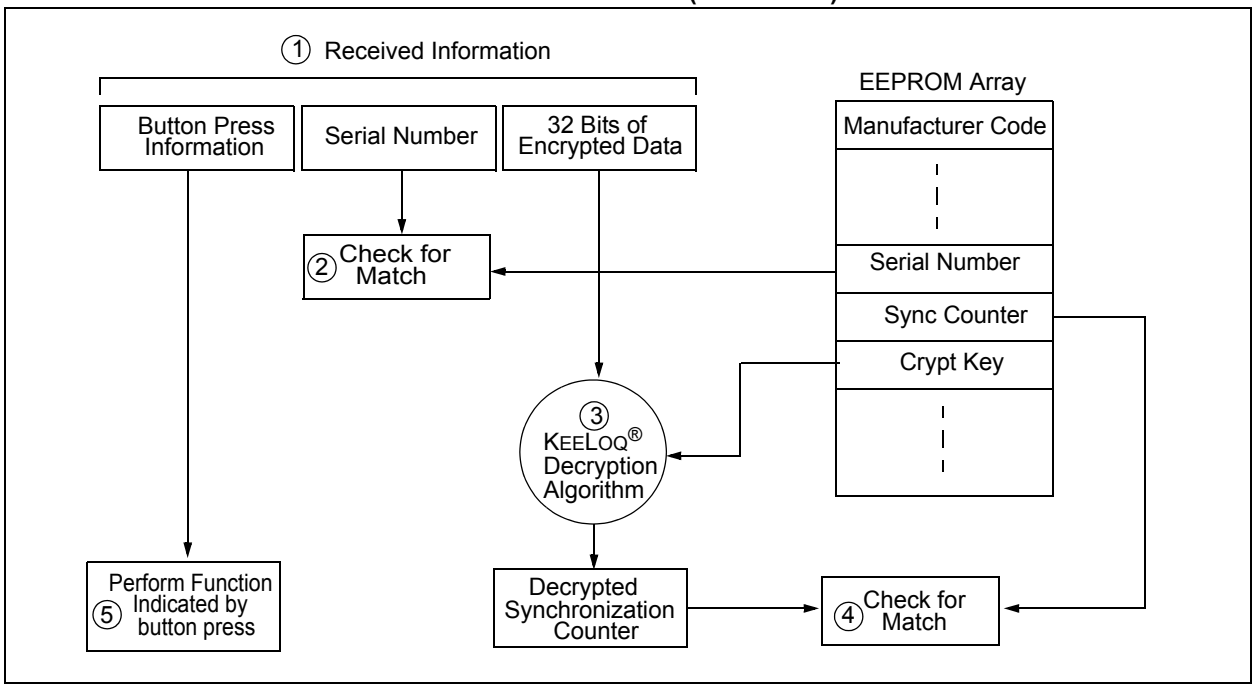


FIGURE 1-3: BASIC OPERATION OF RECEIVER (DECODER)



NOTE: Circled numbers indicate the order of execution.

2.0 PIN ASSIGNMENT

PIN	Decoder Function	I/O (1)	Buffer Type(1)	Description
1	$\overline{\text{LRNIN}}$	I	TTL	Learn input - initiates learning, 10K pull-up required on input
2	LRNOUT	O	TTL	Learn output - indicates learning
3	NC	—	TTL	Do not connect
4	$\overline{\text{MCLR}}$	I	ST	Master clear input
5	Ground	P	—	Ground connection
6	S0	O	TTL	Switch 0
7	S1	O	TTL	Switch 1
8	S2	O	TTL	Switch 2
9	S3	O	TTL	Switch 3
10	VLOW	O	TTL	Battery low indication output
11	SLEEP	I	TTL	Connect to RFIN to allow wake-up from SLEEP
12	CLK	I/O	TTL/ST (2)	Clock in Programming mode and Synchronous mode
13	DATA	I/O	TTL/ST (2)	Data in Programming mode and Synchronous mode
14	VDD	P	—	Power connection
15	OSCOUT (1MHz)	O	TTL	Oscillator out (test point)
16	OSCIN (4MHz)	I	ST	Oscillator in – recommended values 4.7 kΩ and 22 pF
17	NC	—	—	
18	RFIN	I	TTL	RF input from receiver

Note 1: P = power, I = in, O = out, and ST = Schmitt Trigger input.

2: Pin 12 and Pin 13 have a dual purpose. After RESET, these pins are used to determine if Programming mode is selected in which case they are the clock and data lines. In normal operation, they are the clock and data lines of the synchronous data output stream.

© 2011 Microchip Technology Inc.

3.0 DESCRIPTION OF FUNCTIONS

3.1 Parallel Interface

The HCS512 activates the S3, S2, S1 & S0 outputs when a new valid code is received. The outputs will be activated for approximately 500 ms. If a repeated code is received during this time, the output extends for approximately 500 ms.

3.2 Serial Interface

The decoder has a PWM/Synchronous interface connection to microcontrollers with limited I/O. An output data stream is generated when a valid transmission is received. The data stream consists of one START bit, four function bits, one bit for battery status, one bit to indicate a repeated transmission, two status bits, and one STOP bit. (Table 3-1). The DATA and CLK lines are used to send a synchronous event message.

A special status message is transmitted on the second pass of learn. This allows the controlling microcontroller to determine if the learn was successful (Result = 1) and if a previous transmitter was overwritten (Overwrite = 1). The status message is shown in Figure 3-2.

Table 3-1 show the values for TX1:0 and the number of transmitters learned.

TABLE 3-1: STATUS BITS

TX1	TX0	Number of Transmitters
0	0	One
0	1	Two
1	0	Three
1	1	Four

FIGURE 3-1: DATA OUTPUT FORMAT

START	S3	S2	S1	S0	VLOW	REPEAT	TX1	TX0	STOP

FIGURE 3-2: STATUS MESSAGE FORMAT

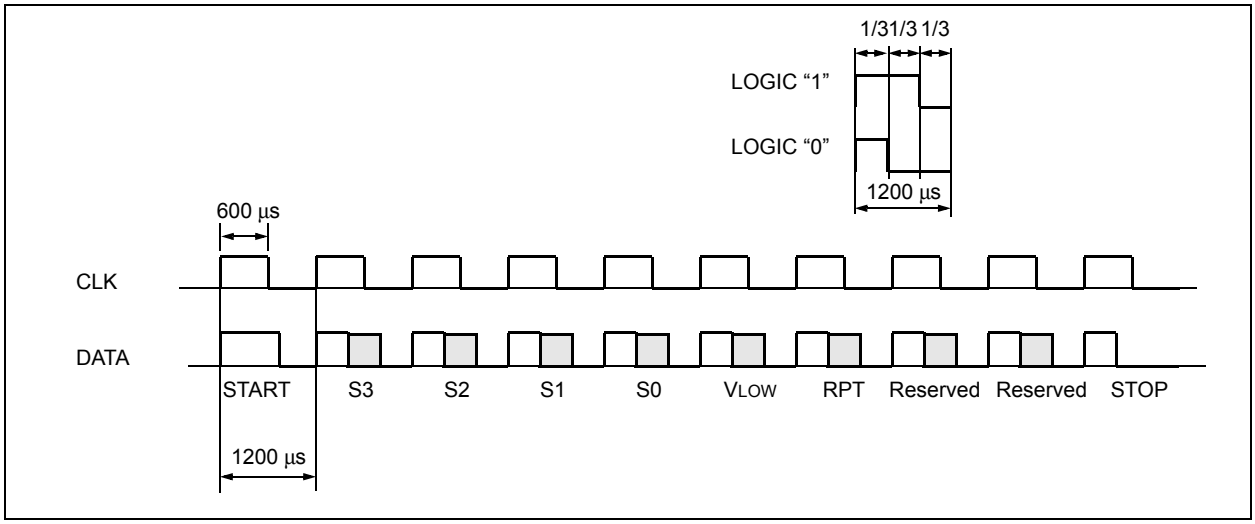
START	0	0	0	0	RESULT	OVRWR	TX1	TX0	STOP

A 1-wire PWM or 2-wire synchronous interface can be used.

In 1-wire mode, the data is transmitted as a PWM signal with a basic pulse width of 400 µs.

In 2-wire mode, Synchronous mode PWM bits start on the rising edge of the clock, and the bits must be sampled on the falling edge. The START bit is a ‘1’ and the STOP bit is ‘0’.

FIGURE 3-2: PWM OUTPUT FORMAT(1)



Note: The Decoder output PWM format logic (“1” / “0”) is reversed with respect of the Encoder modulation format.

4.0 DECODER OPERATION

4.1 Learning a Transmitter to a Receiver

Either the serial number-based learning method or the seed-based learning method can be selected. The learning method is selected in the configuration byte. In order for a transmitter to be used with a decoder, the transmitter must first be 'learned'. When a transmitter is learned to a decoder, the decoder stores the crypt key, a check value of the serial number and current synchronization value in EEPROM. The decoder must keep track of these values for every transmitter that is learned. The maximum number of transmitters that can be learned is four. The decoder must also contain the Manufacturer's Code in order to learn a transmitter. The Manufacturer's Code will typically be the same for all decoders in a system.

The HCS512 has four memory slots. After an "erase all" procedure, all the memory slots will be cleared. Erase all is activated by taking $\overline{\text{LRNIN}}$ low for approximately 10 seconds. When a new transmitter is learned, the decoder searches for an empty memory slot and stores the transmitter's information in that memory slot. When all memory slots are full, the decoder randomly overwrites existing transmitters.

4.1.1 LEARNING PROCEDURE

Learning is activated by taking the $\overline{\text{LRNIN}}$ input low for longer than 64 ms. This input requires an external pull-up resistor.

To learn a new transmitter to the HCS512 decoder, the following sequence is required:

1. Enter Learning mode by pulling $\overline{\text{LRNIN}}$ low for longer than 64 ms. The LRNOUT output will go high.
2. Activate the transmitter until the LRNOUT output goes low indicating reception of a valid code (hopping message).
3. Activate the transmitter a second time until the LRNOUT toggles for 4 seconds (in Secure Learning mode, the seed transmission must be transmitted during the second stage of learn by activating the appropriate buttons on the transmitter).

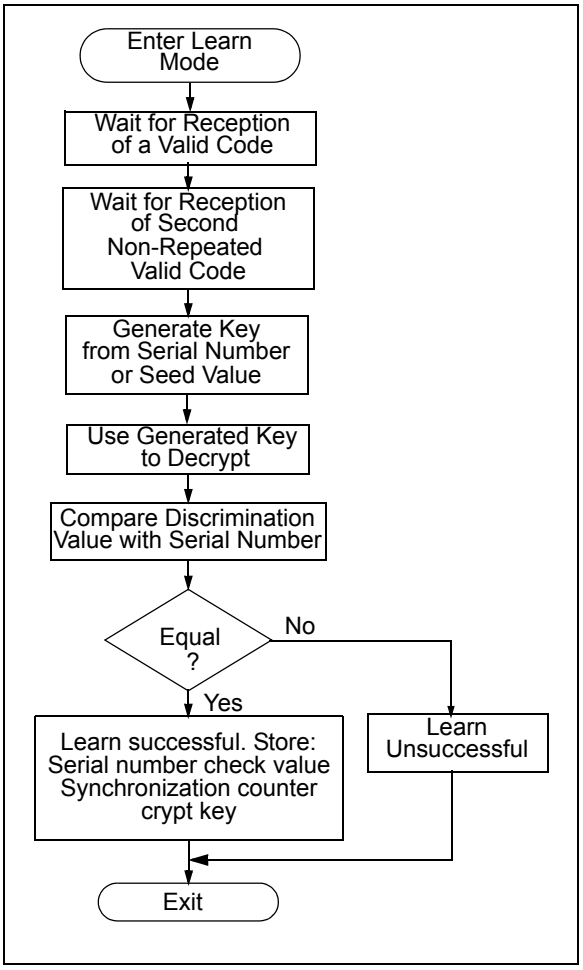
   If $\overline{\text{LRNIN}}$ is taken low momentarily during the learn status indication, the indication will be terminated. Once a successful learning sequence is detected, the indication can be terminated allowing quick learning in a manufacturing setup.
4. The transmitter is now learned into the decoder.
5. Repeat steps 1-4 to learn up to four transmitters.
6. Learning will be terminated if two non-sequential codes were received or if two acceptable codes were not decoded within 30 seconds.

The following checks are performed on the decoder to determine if the transmission is valid during learn:

- The first code word is checked for bit integrity.
- The second code word is checked for bit integrity.
- The hopping code is decrypted.
- If all the checks pass, the serial number and synchronization counters are stored in EEPROM memory.

Figure 4-1 shows a flow chart of the learn sequence.

FIGURE 4-1: LEARN SEQUENCE



© 2011 Microchip Technology Inc.

4.2 Validation of Codes

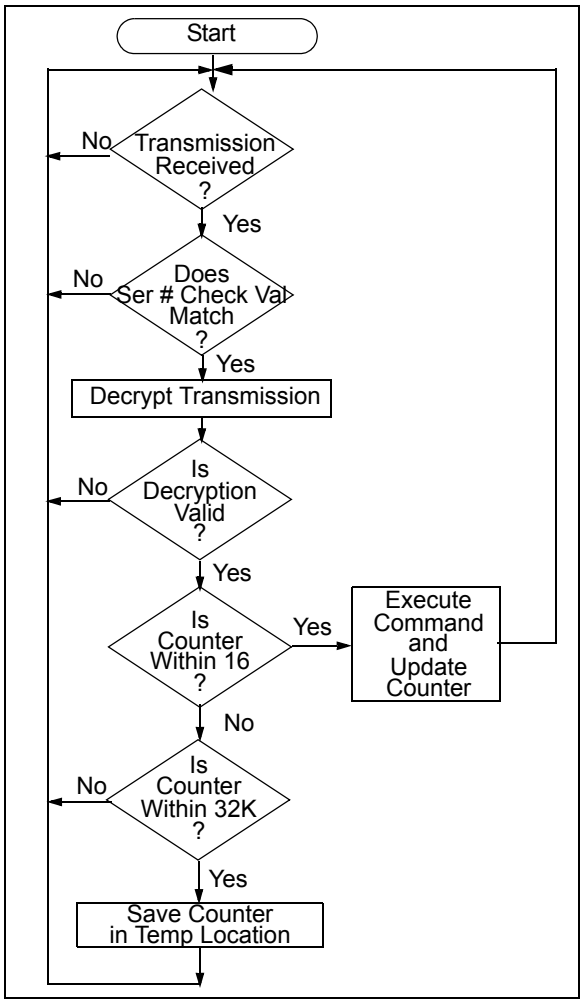
The decoder waits for a transmission and checks the serial number to determine if the transmitter has been learned. If learned, the decoder decrypts the encrypted portion of the transmission using the crypt key. It uses the discrimination bits to determine if the decryption was valid. If everything up to this point is valid, the synchronization value is evaluated.

4.3 Validation Steps

Validation consists of the following steps:

- Search EEPROM to find the Serial Number Check Value Match
- Decrypt the Hopping Code
- Compare the 10 bits of discrimination value with the lower 10 bits of serial number
- Check if the synchronization counter falls within the first synchronization window.
- Check if the synchronization counter falls within the second synchronization window.
- If a valid transmission is found, update the synchronization counter, else use the next transmitter block and repeat the tests.

FIGURE 4-2: DECODER OPERATION



4.4 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.

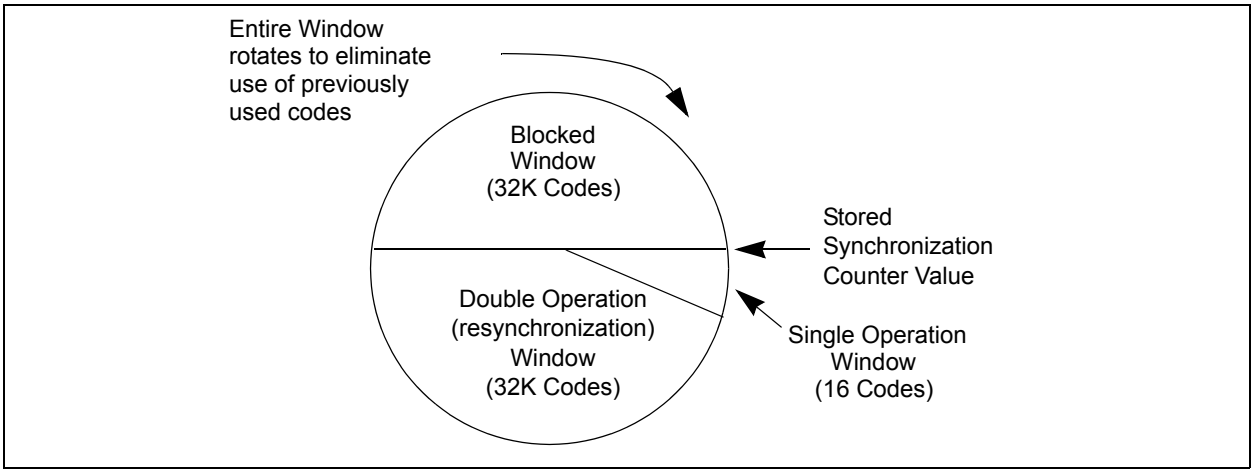
Figure 4-3 shows a 3-partition, rotating synchronization window. The size of each window is optional but the technique is fundamental. Each time a transmission is authenticated, the intended function is executed and the transmission's synchronization counter value is stored in EEPROM. From the currently 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 4-3: SYNCHRONIZATION WINDOW



4.5 SLEEP Mode

The SLEEP mode of the HCS512 is used to reduce current consumption when no RF input signal is present. SLEEP mode will only be effective in systems where the RF receiver is relatively quiet when no signal is present. During SLEEP, the clock stops, thereby significantly reducing the operating current. SLEEP mode is enabled by the SLEEP bit in the configuration byte.

The HCS512 will enter SLEEP mode when:
- The RF line is low
- After a function output is switched off
- Learn mode is terminated (time-out reached)

The device will not enter SLEEP mode when:
- A function output is active
- Learn sequence active
- Device is in Programming mode

The device will wake-up from SLEEP when:
- The SLEEP input pin changes state
- The CLOCK line changes state

Note:	During SLEEP mode the CLK line will change from an output line to an input line that can be used to wake-up the device. Connect CLK to $\overline{\text{LRNIN}}$ via a 100K resistor to reliably enter the Learn mode whenever SLEEP mode is active.

© 2011 Microchip Technology Inc.

5.0 INTEGRATING THE HCS512 INTO A SYSTEM

The HCS512 can act as a stand-alone decoder or be interfaced to a microcontroller. Typical stand-alone applications include garage door openers and electronic door locks. In stand-alone applications, the HCS512 will handle learning, reception, decryption, and validation of the received code; and generate the appropriate output. For a garage door opener, the HCS512 input will be connected to an RF receiver, and the output, to a relay driver to connect a motor controller.

Typical systems where the HCS512 will be connected to a microcontroller include vehicle and home security systems. The HCS512 input will be connected to an RF receiver and the function outputs to the microcontroller. The HCS512 will handle all the decoding functions and the microcontroller, all the system functions. The Serial Output mode with a 1- or 2-wire interface can be used if the microcontroller is I/O limited.

6.0 DECODER PROGRAMMING

The PG306001 production programmer will allow easy setup and programming of the configuration byte and the manufacturer's code.

6.1 Configuration Byte

The configuration byte is used to set system configuration for the decoder. The LRN bits determine which algorithm (Decrypt or XOR) is used for the key generation. SC_LRN determines whether normal learn (key derived from serial number) or secure learn (key derived from seed value) is used.

TABLE 6-1: CONFIGURATION BYTE

Bit	Name	Description
0	LRN0	Learn algorithm select
1	LRN1	Not used
2	SC_LRN	Secure Learn enable (1 = enabled)
3	SLEEP	SLEEP enable (1 = enabled)
4	RES1	Not used
5	RES2	Not used
6	RES3	Not used
7	RES4	Not used

TABLE 6-2: LEARN METHOD LRN0, LRN1 DEFINITIONS

LRN0	Description
0	Decrypt algorithm
1	XOR algorithm

6.2 Programming the Manufacturer's Code

The manufacturer's code must be programmed into EEPROM memory through the synchronous programming interface using the DATA and CLK lines. Provision must be made for connections to these pins if the decoder is going to be programmed in circuit.

Programming mode is activated if the CLK is low for at least 1 ms and then goes high within 64 ms after power-up, stays high for longer than 8 ms but not longer than 128 ms. After entering Programming mode the 64-bit manufacturer's code, 8-bit configuration byte, and 8-bit checksum is sent to the device using the synchronous interface. After receiving the 80-bit message the checksum is verified and the information is written to EEPROM. If the programming operation was successful, the HCS512 will respond with an Acknowledge pulse.

After programming the manufacturer's code, the HCS512 decoder will automatically activate an Erase All function, removing all transmitters from the system.

6.3 Download Format

The manufacturer's code and configuration byte must be downloaded Least Significant Byte, Least Significant bit first as shown in Table 6-3.

6.4 Checksum

The checksum is used by the HCS512 to check that the data downloaded was correctly received before programming the data. The checksum is calculated so that the 10 bytes added together (discarding the overflow bits) is zero. The checksum can be calculated by adding the first 9 bytes of data together and subtracting the result from zero. Throughout the calculation the overflow is discarded.

Given a manufacturer's code of 01234567-89ABCDEF$_{16}$ and a Configuration Word of 1$_{16}$, the checksum is calculated as shown in Figure 6-1. The checksum is 3F$_{16}$.

6.5 Test Transmitter

The HCS512 decoder will automatically add a test transmitter each time an Erase All Function is done. A test transmitter is defined as a transmitter with a serial number of zero. After an Erase All, the test transmitter will always work without learning and will not check the synchronization counter of the transmitter. Learning of any new transmitters will erase the test transmitter.

Note 1: A transmitter with a serial number of zero cannot be learned. Learn will fail after the first transmission.

2: Always learn at least one transmitter after an Erase All sequence. This ensures that the test transmitter is erased.

TABLE 6-3: DOWNLOAD DATA

Byte 9	Byte 8	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Checksum	Config	Man Key_7	Man Key_6	Man Key_5	Man Key_4	Man Key_3	Man Key_2	Man Key_1	Man Key_0

Byte 0, right-most bit downloaded first.

FIGURE 6-1: CHECKSUM CALCULATION

$01_{16} + 23_{16} = 24_6$

$24_{16} + 45_{16} = 69_{16}$

$69_{16} + 67_{16} = D0_{16}$

$D0_{16} + 89_{16} = 159_{16}$

$59_{16} + AB_{16} = 104_{16}$ (Carry is discarded)

$04_{16} + CD_{16} = D1_{16}$ (Carry is discarded)

$D1_{16} + EF_{16} = 1C0_{16}$

$C0_{16} + 1_{16} = C1_{16}$ (Carry is discarded)

$(FF_{16} - C1_{16}) + 1_{16} = 3F_{16}$

© 2011 Microchip Technology Inc.

FIGURE 6-2: PROGRAMMING WAVEFORMS

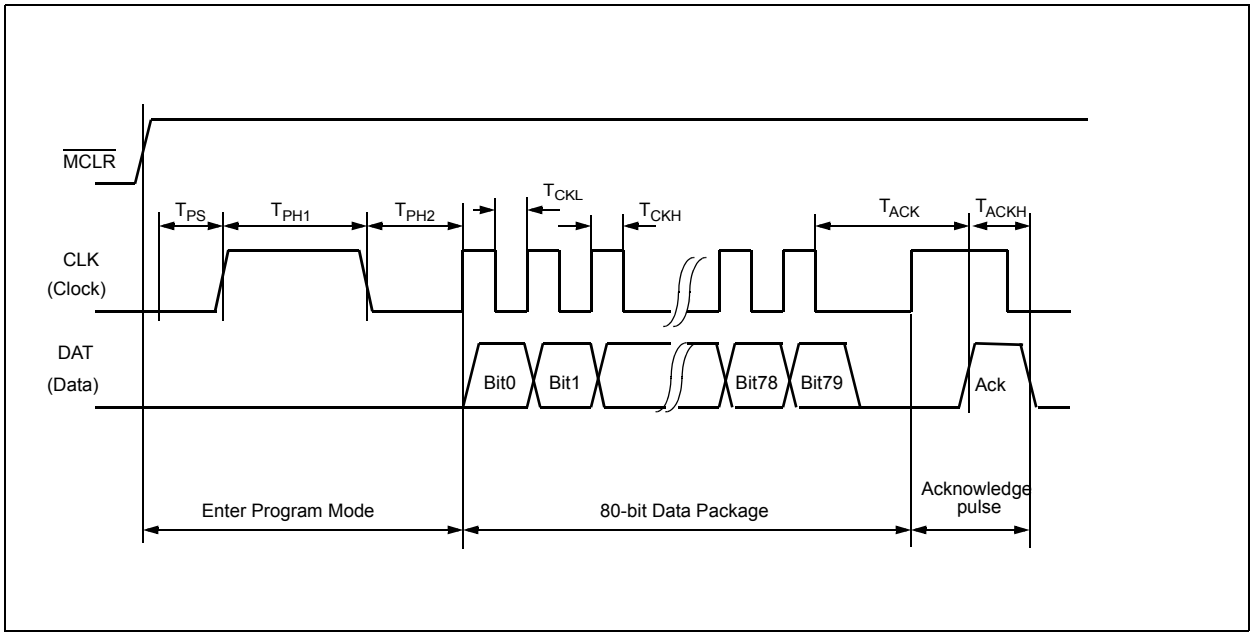


TABLE 6-4: PROGRAMMING TIMING REQUIREMENTS

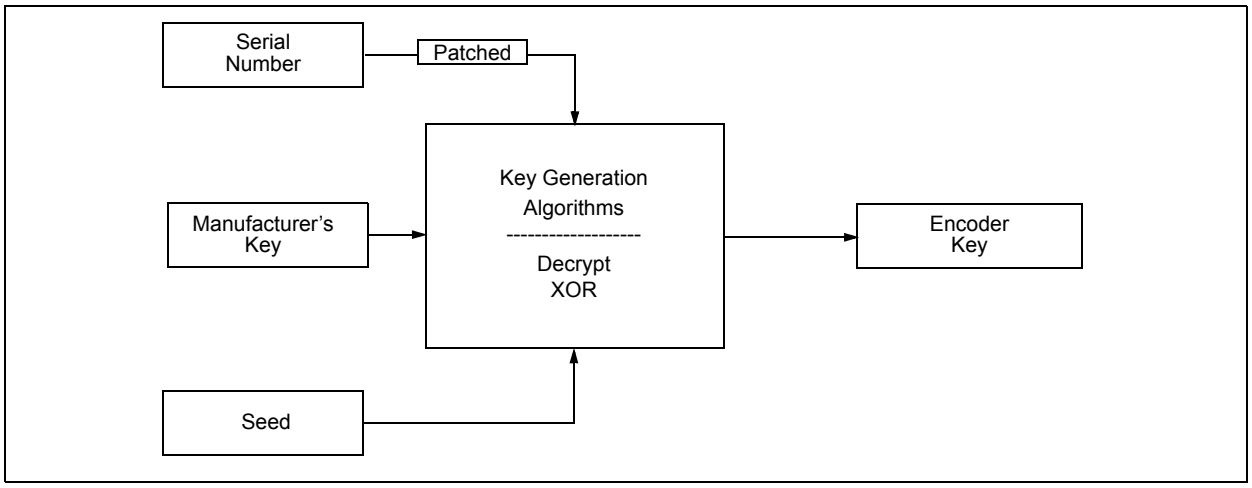
Parameter	Symbol	Min.	Max.	Units
Program mode setup time	TPS	1	64	ms
Hold time 1	TPH1	8	128	ms
Hold time 2	TPH2	0.05	320	ms
Clock High Time	TCKH	0.05	320	ms
Clock Low Time	TCKL	0.050	320	ms
Acknowledge Time	TACK	—	80	ms
Acknowledge duration	TACKH	1	—	ms

Note: F_{OSC} equals 4 MHz.

7.0 KEY GENERATION SCHEMES

The HCS512 decoder has two key generation schemes. Normal learning uses the transmitter's serial number to derive two input seeds which are used as inputs to the key generation algorithm. Secure learning uses the seed transmission to derive the two input seeds. Two key generation algorithms are available to convert the inputs seeds to secret keys. The appropriate scheme is selected in the Configuration Word.

FIGURE 7-1:



7.1 Normal Learning (Serial Number Derived)

The two input seeds are composed from the serial number in two ways, depending on the encoder type. The encoder type is determined from the number of bits in the incoming transmission. SourceH is used to calculate the upper 32 bits of the crypt key, and SourceL, for the lower 32 bits.

For 28-bit serial number encoders (66 / 67-bit transmissions):

SourceH = 6H + 28 bit Serial Number
SourceL = 2H + 28 bit Serial Number

7.2 Secure Learning (Seed Derived)

The two input seeds are composed from the seed value that is transmitted during secure learning. The lower 32 bits of the seed transmission is used to compose the lower seed, and the upper 32 bits, for the upper seed. The upper 4 bits (function code) are set to zero.

For 32-bit seed encoders:

SourceH = Serial Number $_{\text{Lower 28 bits}}$ (with upper 4 bits always zero)
SourceL = Seed $_{\text{32 bits}}$

For 48-bit seed encoders:

SourceH = Seed $_{\text{Upper 16 bits}}$ + Serial Number $_{\text{Upper 16 bits}}$ (with upper 4 bits always zero) << 16
SourceL = Seed $_{\text{Lower 32 bits}}$

For 60-bit seed encoders:

SourceH = Seed $_{\text{Upper 28 bits}}$ (with upper 4 bits always zero)
SourceL = Seed $_{\text{Lower 32 bits}}$

© 2011 Microchip Technology Inc.

7.3 Key Generation Algorithms

There are two key generation algorithms implemented in the HCS512 decoder. The KEELOQ decryption algorithm provides a higher level of security than the XOR algorithm. Section 6.1 describes the selection of the algorithms in the configuration byte.

7.3.1 KEELOQ DECRYPT ALGORITHM

This algorithm uses the KEELOQ decryption algorithm and the manufacturer's code to derive the crypt key as follows:

$Key_{Upper\ 32\ bits}$ = Decrypt (SourceH) $_{64\ Bit\ Manufacturers\ Code}$

$Key_{Lower\ 32\ bits}$ = Decrypt (SourceL) $_{64\ Bit\ Manufacturers\ Code}$

7.3.2 XOR WITH THE MANUFACTURER'S CODE

The two 32-bits seeds are XOR with the manufacturer's code to form the 64 bit crypt key.

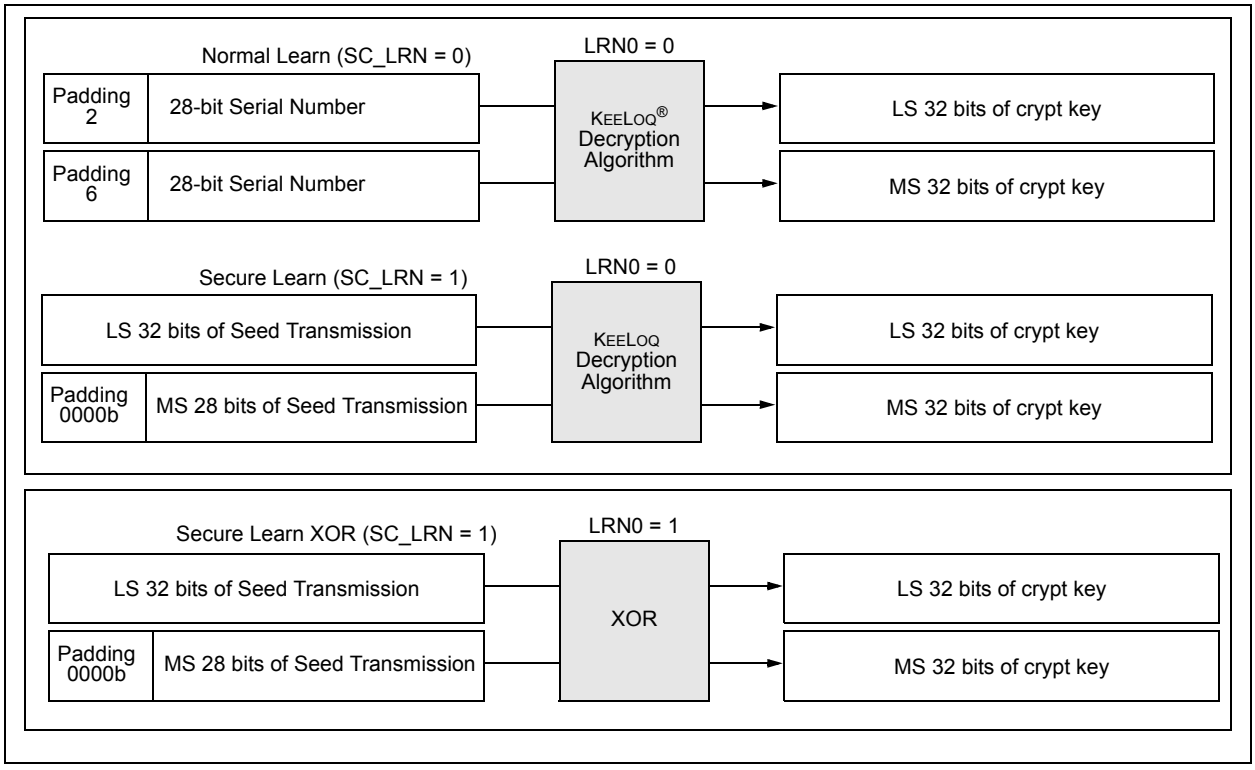
$Key_{Upper\ 32\ bits}$ = SourceH **XOR** Manufacturers Code $_{Upper\ 32\ bits}$

$Key_{Lower\ 32\ bits}$ = SourceL **XOR** Manufacturers Code $_{Lower\ 32\ bits}$

After programming the manufacturer's code, the HCS512 decoder will automatically activate an Erase All function, removing all transmitters from the system.

If $\overline{LRNIN}$ is taken low momentarily during the learn status indication, the indication will be terminated. Once a successful learning sequence is detected, the indication can be terminated, allowing quick learning in a manufacturing setup.

FIGURE 7-2: HCS512 KEY GENERATION



8.0 KEELOQ ENCODERS

8.1 Transmission Format (PWM)

The KEELOQ encoder transmission is made up of several parts (Figure 8-1). Each transmission begins with a preamble and a header, followed by the encrypted and then the fixed data. The actual data is 66/69 bits which consists of 32 bits of encrypted data and 34/37 bits of non-encrypted data. Each transmission is followed by a guard period before another transmission can begin. The encrypted portion provides up to four billion changing code combinations and includes the button status bits (based on which buttons were activated) along with the synchronization counter value and some discrimination bits. The non-encrypted portion is comprised of the status bits, the function bits, and the 28-bit serial number. The encrypted and non-encrypted combined sections increase the number of combinations to 7.38 x 10^{19}.

8.2 Code Word Organization

The HCSXXX encoder transmits a 66/69-bit code word when a button is pressed. The 66/69-bit word is constructed from an encryption portion and a non-encrypted code portion (Figure 8-2).

The **Encrypted Data** is generated from four button bits, two overflow counter bits, ten discrimination bits, and the 16-bit synchronization value.

The **Non-encrypted Data** is made up from 2 status bits, 4 function bits, and the 28/32-bit serial number.

FIGURE 8-1: TRANSMISSION FORMAT (PWM)

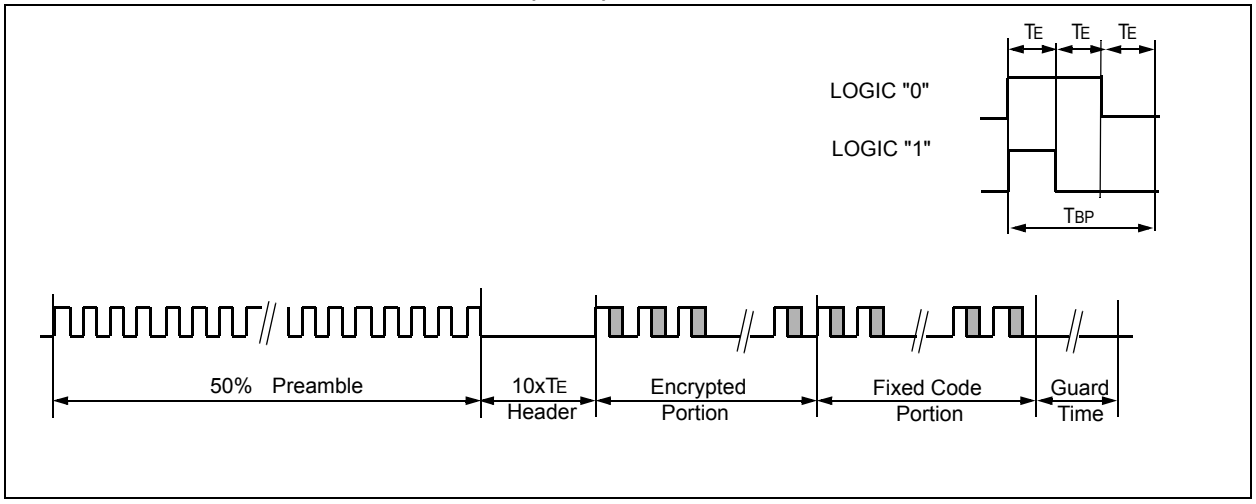
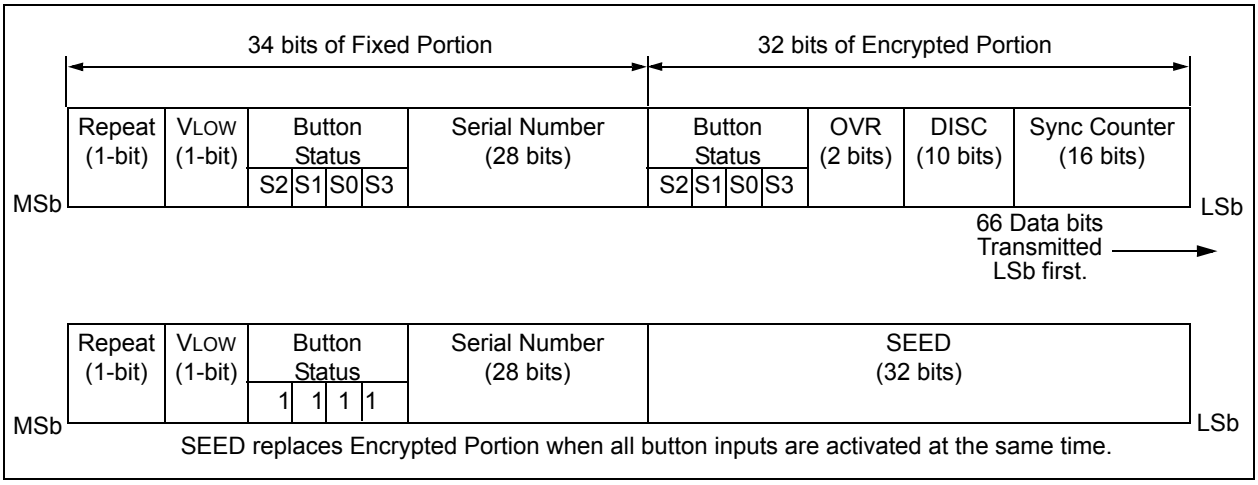


FIGURE 8-2: CODE WORD ORGANIZATION



© 2011 Microchip Technology Inc.

9.0 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

9.1 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.

9.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

9.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

9.4 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

9.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

9.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

© 2011 Microchip Technology Inc.

9.7 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC® MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

9.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

9.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

9.10 PICkit 3 In-Circuit Debugger/ Programmer and PICkit 3 Debug Express

The MPLAB PICkit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the reset line to implement in-circuit debugging and In-Circuit Serial Programming™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

© 2011 Microchip Technology Inc.

9.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

9.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

9.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

© 2011 Microchip Technology Inc.

10.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings †

Ambient temperature under bias ..... -55°C to +125°C

Storage temperature ..... -65°C to +150°C

Voltage on any pin with respect to V_{SS} (except V_{DD}) ..... -0.6V to V_{DD} +0.6V

Voltage on V_{DD} with respect to Vss ..... 0 to +7.5V

Total power dissipation **(Note 1)** ..... 800 mW

Maximum current out of V_{SS} pin ..... 150 mA

Maximum current into V_{DD} pin ..... 100 mA

Input clamp current, Iik (V_I < 0 or V_I > V_{DD}) ..... ± 20 mA

Output clamp current, IOK (V_O < 0 or V_O > V_{DD}) ..... ± 20 mA

Maximum output current sunk by any I/O pin ..... 25 mA

Maximum output current sourced by any I/O pin ..... 20 mA

Note: Power dissipation is calculated as follows: $Pdis = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum\{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$

† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 10-1: DC CHARACTERISTICS

		Standard Operating Conditions (unless otherwise stated) Operating temperature Commercial (C): 0° C ≤TA ≤+70° C for commercial Industrial (I): -40° C ≤TA ≤+85° C for industrial				
Symbol	**Characteristic**	**Min**	**Typ(†)**	**Max**	**Units**	**Conditions**
VDD	Supply Voltage	4.0	—	6.0	V	
VPOR	VDD start voltage to ensure RESET	—	VSS	—	V	
SVDD	VDD rise rate to ensure RESET	0.05*	—	—	V/ms	
IDD	Supply Current	— — 	1.8 7.3 15	4.5 10 32	mA mA µA	FOSC = 4 MHz, VDD = 5.5V (During EEPROM programming) In SLEEP mode
VIL	Input Low Voltage	VSS	—	0.16 VDD	V	except $\overline{\text{MCLR}}$ = 0.2 VDD
VIH	Input High Voltage	0.48 VDD	—	VDD	V	except $\overline{\text{MCLR}}$ = 0.85 VDD
VOL	Output Low Voltage	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V
VOH	Output High Voltage	VDD-0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V

† Data in “Typ” column is at 5.0V, 25° C unless otherwise stated. These parameters are for design guidance only and are not tested.

* These parameters are characterized but not tested.

Note: Negative current is defined as coming out of the pin.

TABLE 10-2: AC CHARACTERISTICS

Symbol	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
FOSC	Oscillator frequency	2.7	4	6.21	MHz	REXT = 10K, CEXT = 10 pF
TE	PWM elemental pulse width	65	—	1080	µs	4.5V < VDD < 5.5V Oscillator components tolerance < 6%.
		130	—	1080	µs	3V < VDD < 6V Oscillator components tolerance <10%
TOD	Output delay	70	90	115	ms	
TA	Output activation time	322	500	740	ms	
TRPT	REPEAT activation time	32	50	74	ms	
TLRN	$\overline{\text{LRNIN}}$ activation time	21	32	—	ms	
TMCLR	$\overline{\text{MCLR}}$ low time	150	—	—	ns	
TOV	Time output valid	—	150	222	ms	

* These parameters are characterized but not tested.

© 2011 Microchip Technology Inc.

FIGURE 10-1: RESET WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

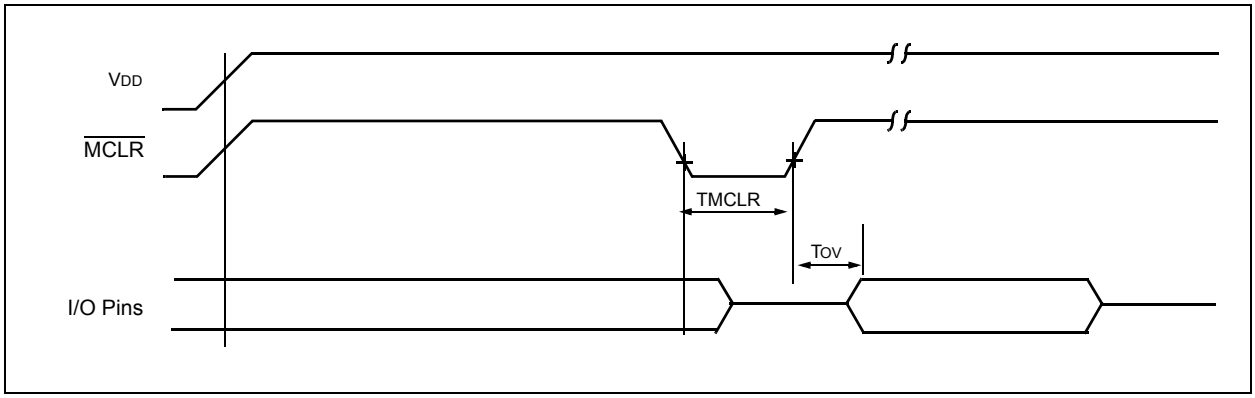
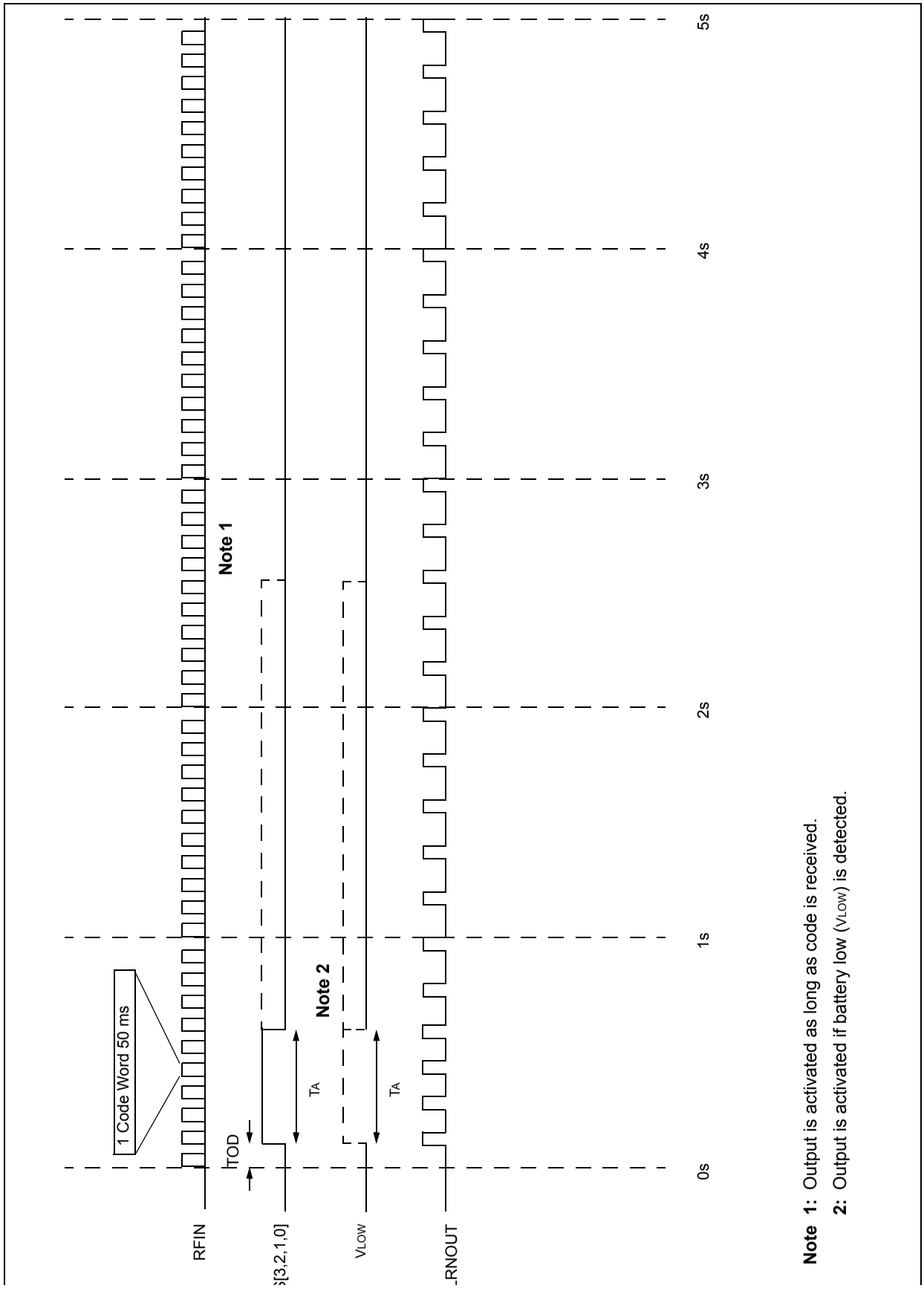
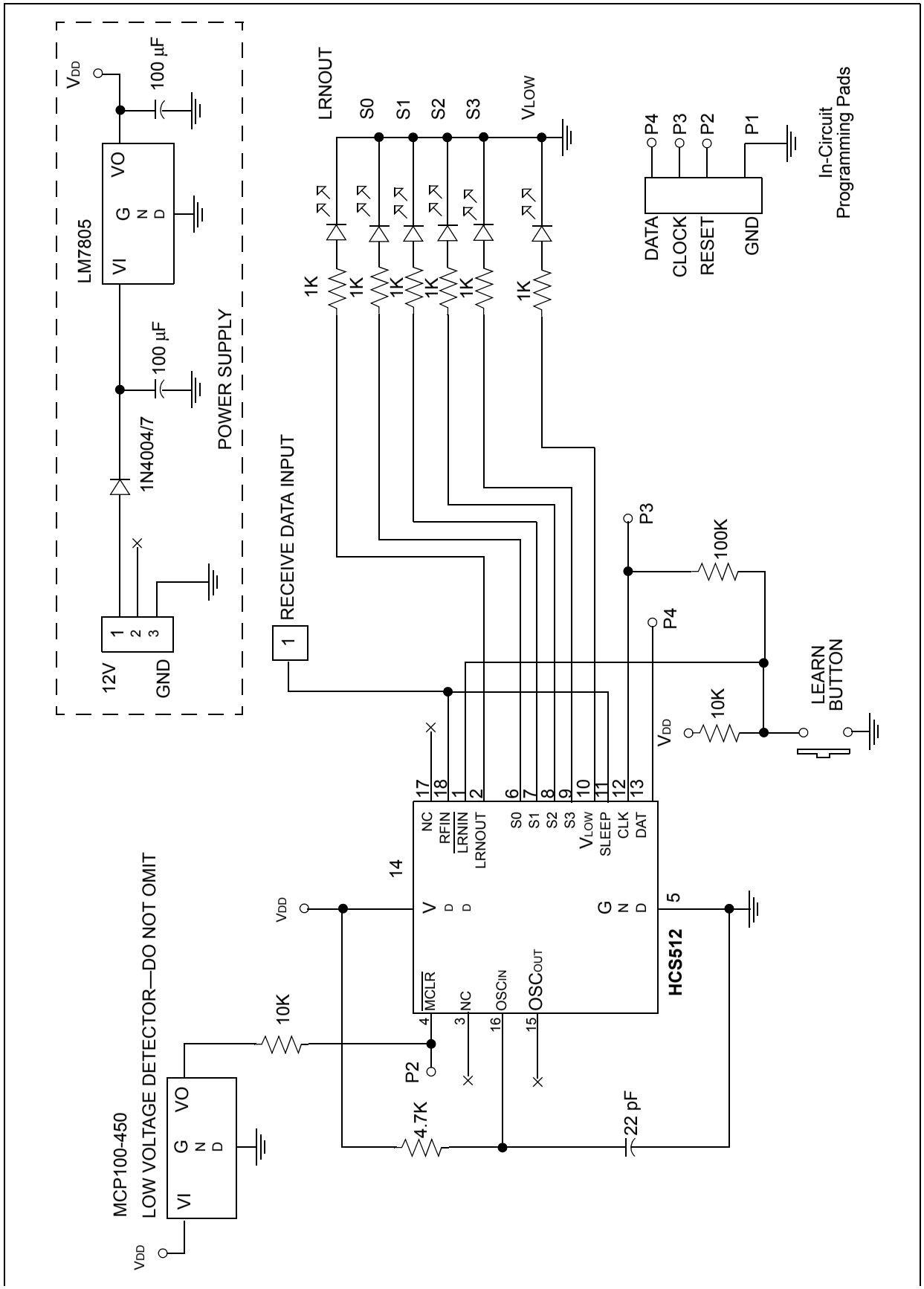


FIGURE 10-2: OUTPUT ACTIVATION



© 2011 Microchip Technology Inc.

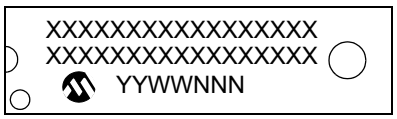
FIGURE 10-3: TYPICAL DECODER APPLICATION CIRCUIT



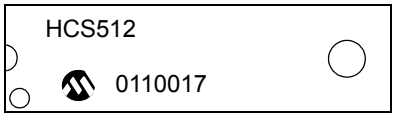
11.0 PACKAGING INFORMATION

11.1 Package Marking Information

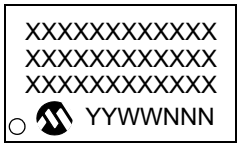
18-Lead PDIP



Example



18-Lead SOIC



Example

Legend:	XX...X	Customer specific information*
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

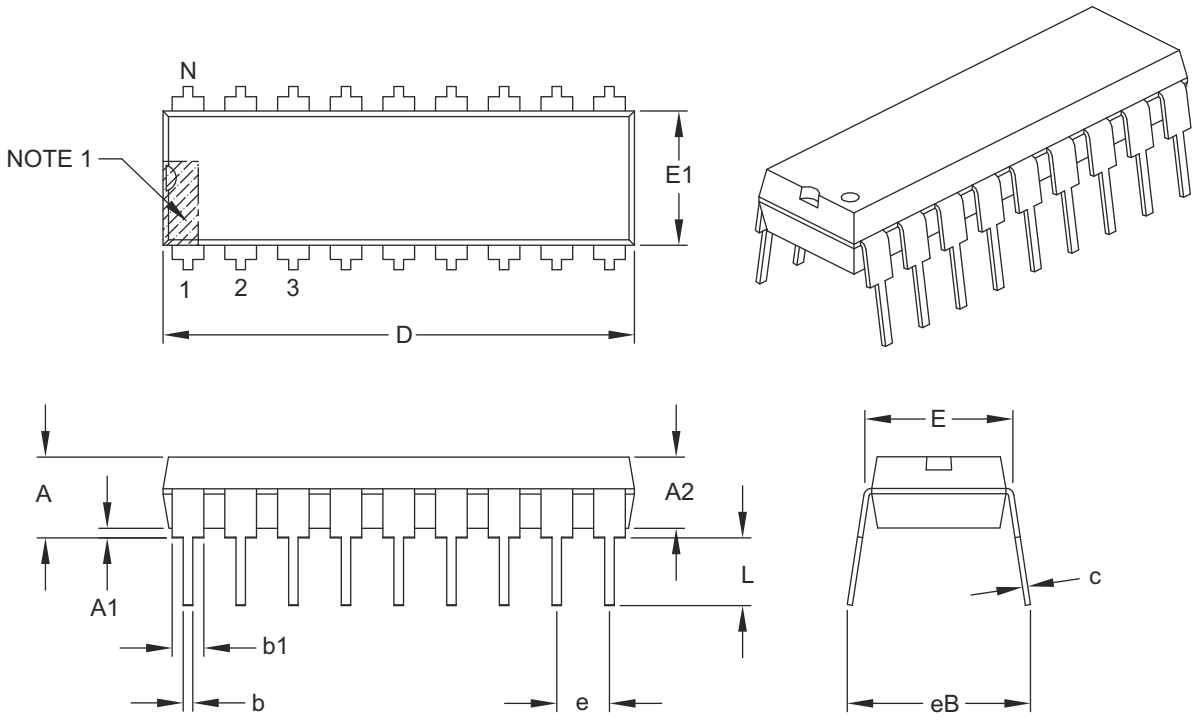
***** Standard PIC MCU device marking consists of Microchip part number, year code, week code, and traceability code. For PIC MCU device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

© 2011 Microchip Technology Inc.

11.2 Package Details

18-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		18	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.300	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.880	.900	.920
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.014
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-007B