HCS500

KEELOQ® Code Hopping Decoder

FEATURES

Security

- Encrypted Storage of Manufacturer's Code
- Encrypted Storage of Crypt Keys
- Up to Seven Transmitters can be Learned
- KEELOQ® Code Hopping Technology
- Normal and Secure Learning Mechanisms

Operating

- 3.0V—5.5V Operation
- Internal Oscillator
- Auto Bit Rate Detection

Other

- Stand-Alone Decoder Chipset
- External EEPROM for Transmitter Storage
- Synchronous Serial Interface
- 1 Kbit user EEPROM
- 8-Pin PDIP/SOIJ 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 us to 400 us
- 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. HCS500 is a code hopping decoder designed for secure Remote Keyless Entry (RKE) systems. The HCS500 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 and access control systems. The HCS500 can be used as a stand-alone decoder or in conjunction with a microcontroller.

PIN DIAGRAM

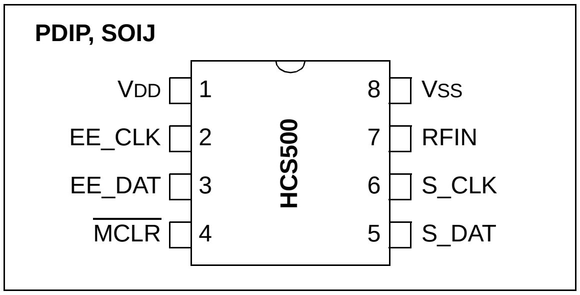
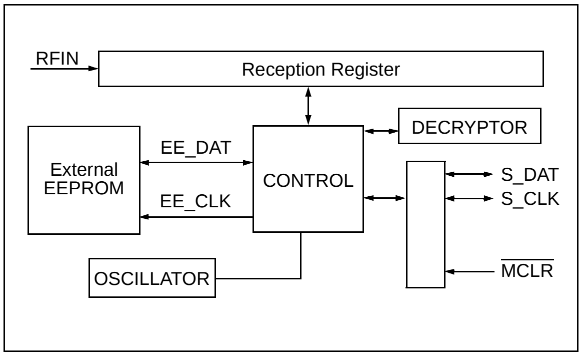


FIGURE 1: BLOCK DIAGRAM



The manufacturer's code, crypt keys, and synchronization information are stored in encrypted form in external EEPROM. The HCS500 uses the S_DAT and S_CLK inputs to communicate with a host controller device.

The HCS500 operates over a wide voltage range of 3.0 volts to 5.5 volts. The decoder employs automatic bit-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 four button Status bits S3, S2, S1 and S0 (Figure 7-2).
- **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 7-1).
- **Transmission** – A data stream consisting of repeating code words (Figure 7-1).
- **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

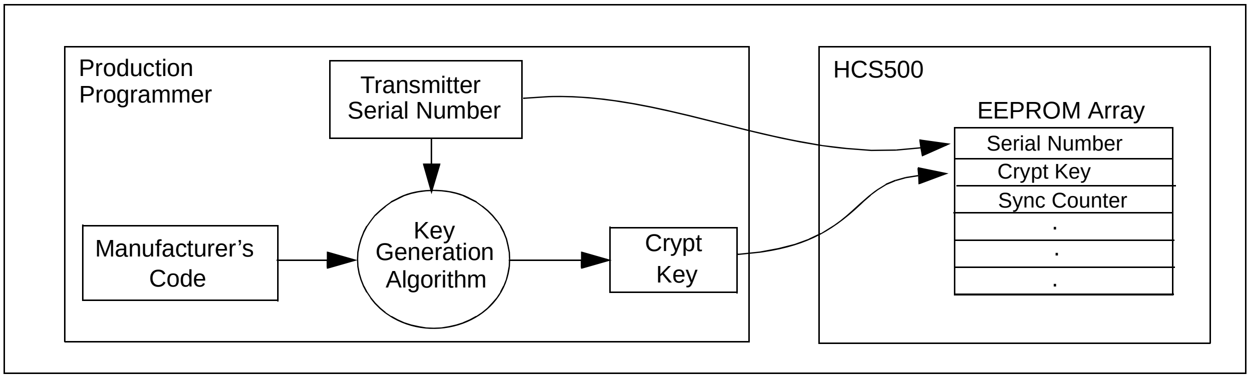
The HCS encoders have a small EEPROM array which must be loaded with several parameters before use. The most important of these values are:

- 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.

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 7.2 "Code Word Organization"**.

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 HCS500 based transmitter. **Section 3.0 "Decoder Operation"** provides detail on integrating the HCS500 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)

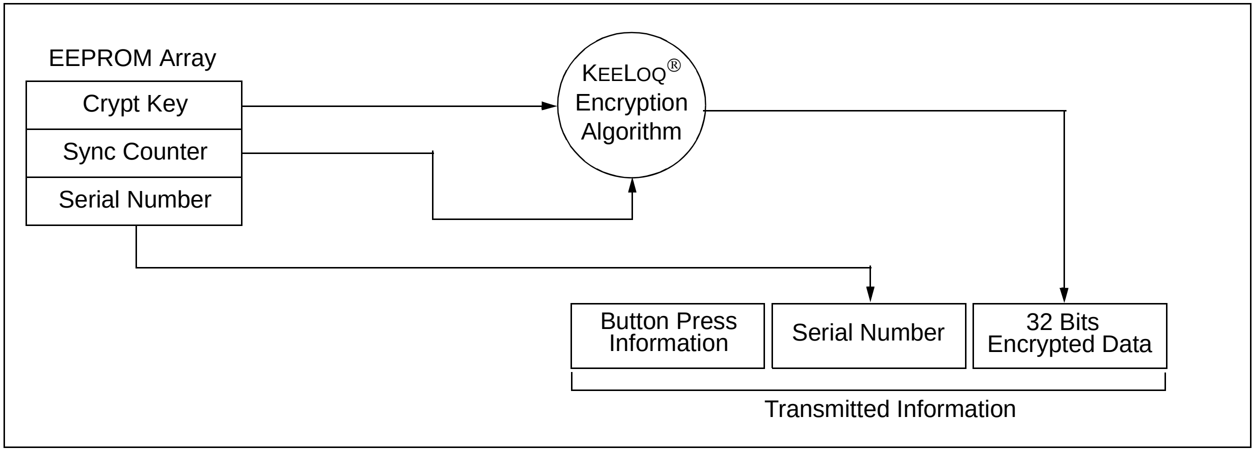
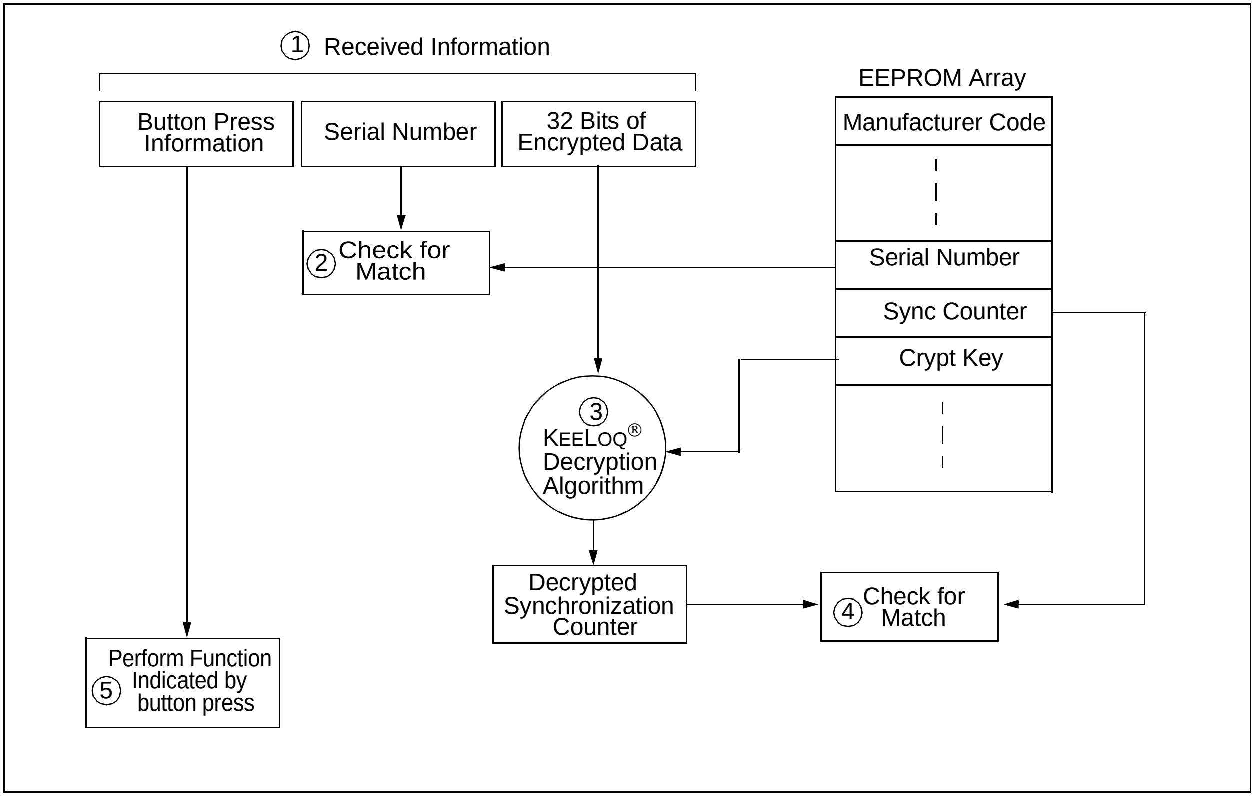


FIGURE 1-3: BASIC OPERATION OF RECEIVER (DECODER)



Note: Circled numbers indicate the order of execution.

2.0 PIN ASSIGNMENT

The description of the pins of the HCS500 decoder is provided in Table 2-1.

TABLE 2-1: DECODER PIN ASSIGNMENT

PIN	Decoder Function	I/O[(1)]	Buffer Type[(1)]	Description
1	VDD	P	—	Power Connection
2	EE_CLK	O	TTL	Clock to I²C™ EEPROM
3	EE_DAT	I/O	TTL	Data to I²C™ EEPROM
4	$\overline{\text{MCLR}}$	I	ST	Master clear input
5	S_DAT	I/O	TTL	Synchronous data from controller
6	S_CLK	I	TTL	Synchronous clock from controller
7	RFIN	I	TTL	RF input from receiver
8	GND	P	—	Ground connection

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

3.0 DECODER OPERATION

3.1 Learning a Transmitter to a Receiver (Normal or Secure Learn)

Before the transmitter and receiver can work together, the receiver must first ‘learn’ and store the following information from the transmitter in EEPROM:

- A check value of the serial number
- The crypt key
- The current synchronization counter value

The decoder must also store the manufacturer’s code (**Section 1.1 “HCS Encoder Overview”**) in protected memory. This code will typically be the same for all of the decoders in a system.

The HCS500 has seven memory slots, and, consequently, can store up to seven transmitters. During the learn procedure, the decoder searches for an empty memory slot for storing the transmitter’s information. When all of the memory slots are full, the decoder will overwrite the last transmitter’s information. To erase all of the memory slots at once, use the ERASE_ALL command (C3H).

3.2 Learning Procedure

Learning is initiated by sending the ACTIVATE_LEARN (D2H) command to the decoder. The decoder acknowledges reception of the command by pulling the data line high.

For the HCS500 decoder to learn a new transmitter, the following sequence is required:

1. Activate the transmitter once.
2. Activate the transmitter a second time. (In Secure Learning mode, the seed transmission must be transmitted during the second stage of learn by activating the appropriate buttons on the transmitter.)

   The HCS500 will transmit a learn-status string, indicating that the learn was successful.
3. The decoder has now learned the transmitter.
4. Repeat steps 1-3 to learn up to seven transmitters

Note 1: Learning will be terminated if two nonsequential codes were received or if two acceptable codes were not decoded within 30 seconds.

2: If more than seven transmitters are learned, the new transmitter will replace the last transmitter learned. It is, therefore, not possible to erase lost transmitters by repeatedly learning new transmitters. To remove lost or stolen transmitters, ERASE_ALL transmitters and relearn all available transmitters.

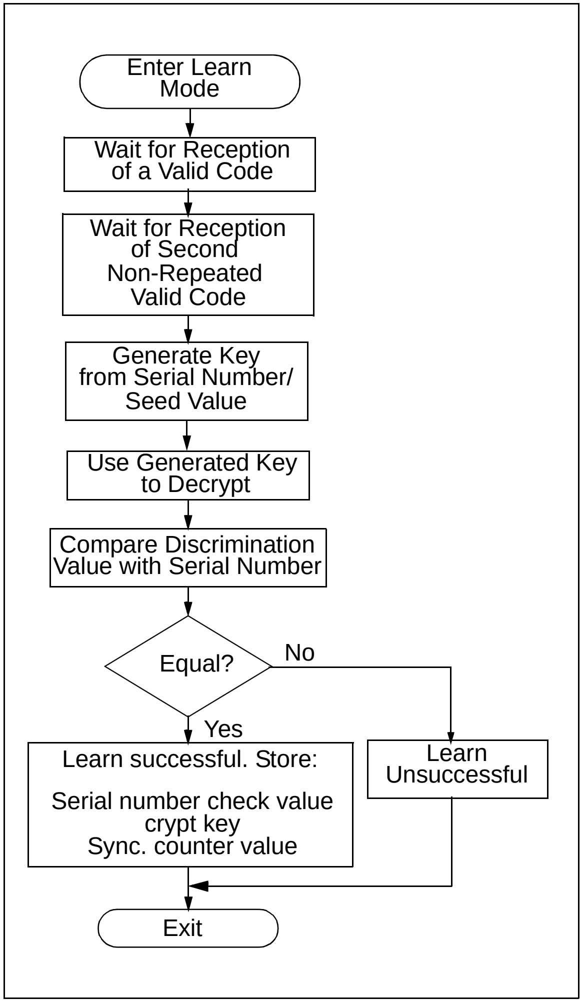
3: Learning a transmitter with a crypt key that is identical to a transmitter already in memory replaces the existing transmitter. In practice, this means that all transmitters should have unique crypt keys. Learning a previously learned transmitter does not use any additional memory slots.

The following checks are performed by 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 crypt key is generated according to the selected algorithm
- The hopping code is decrypted
- The discrimination value is checked
- If all the checks pass, the key, serial number check value, and synchronization counter values are stored in EEPROM memory

Figure 3-1 shows a flow chart of the learn sequence.

FIGURE 3-1: LEARN SEQUENCE


Enter Learn Mode
Wait for Reception of a Valid Code
Wait for Reception of Second Non-Repeated Valid Code
Generate Key from Serial Number/ Seed Value
Use Generated Key to Decrypt
Compare Discrimination Value with Serial Number
Equal?
No
Yes
Learn successful. Store:
Serial number check value
crypt key
Sync. counter value
Learn Unsuccessful
Exit

3.3 Validation of Codes

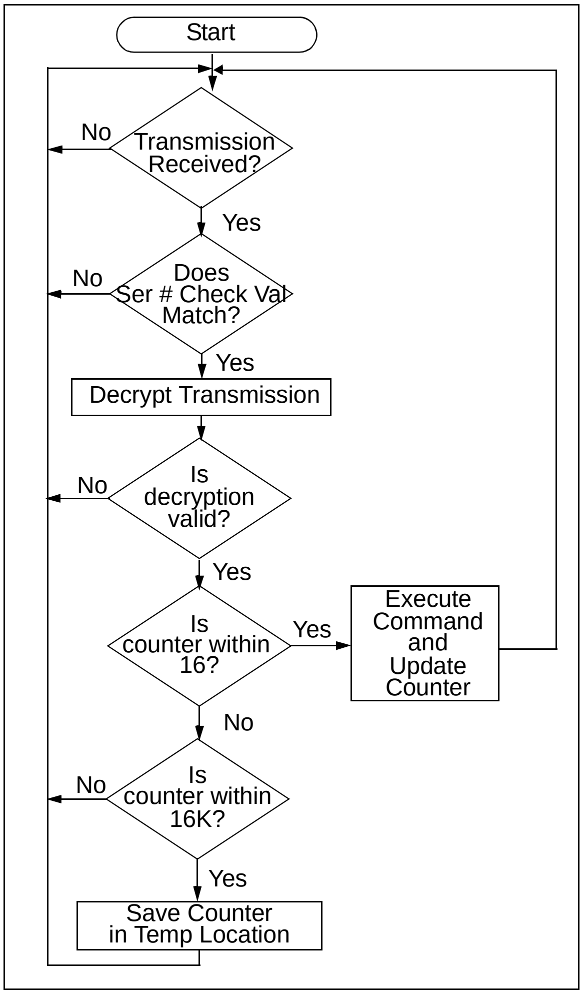
The decoder waits for a transmission and checks the serial number to determine if it is a learned transmitter. If it is, it takes the code hopping portion of the transmission and decrypts it, using the crypt key. It uses the discrimination value to determine if the decryption was valid. If everything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

3.4 Validation Steps

Validation consists of the following steps:

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

FIGURE 3-2: DECODER OPERATION



3.5 Synchronization with Decoder (Evaluating the Counter)

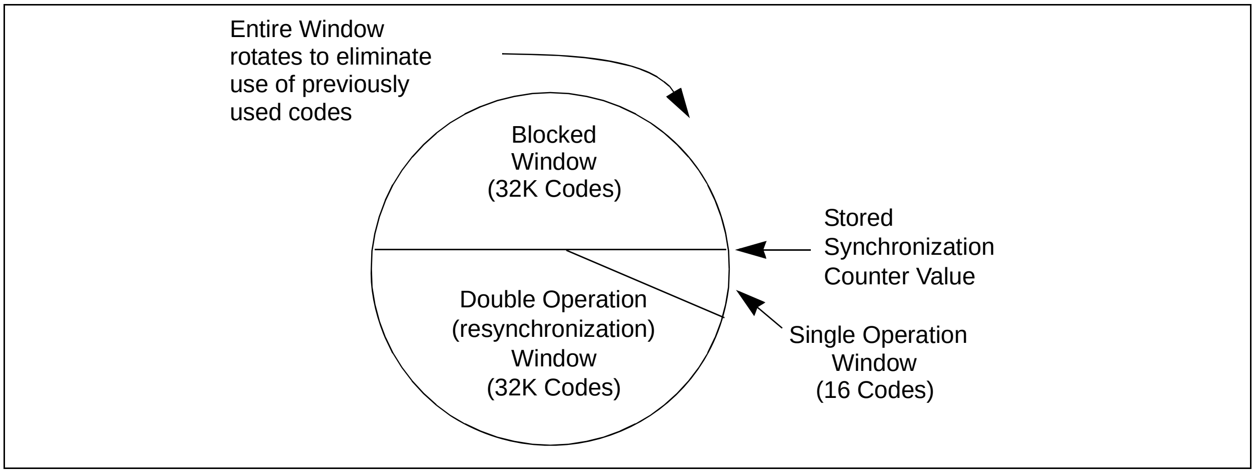
The KEELOQ technology patent scope includes a sophisticated synchronization technique that does not require the calculation and storage of future codes. The technique securely blocks invalid transmissions while providing transparent resynchronization to transmitters inadvertently activated away from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

Figure 3-3 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.

FIGURE 3-3: SYNCHRONIZATION WINDOW



4.0 INTERFACING TO A MICROCONTROLLER

The HCS500 interfaces to a microcontroller via a synchronous serial interface. A clock and data line are used to communicate with the HCS500. The microcontroller controls the clock line. There are two groups of data transfer messages. The first is from the decoder whenever the decoder receives a valid transmission. The decoder signals reception of a valid code by taking the data line high (maximum of 500 ms) The microcontroller then services the request by clocking out a data string from the decoder. The data string contains the function code, the Status bit, and block indicators. The second is from the controlling microcontroller to the decoder in the form of a defined command set.

Figure 4-1 shows the HCS500 decoder and the I/O interface lines necessary to interface to a microcontroller.

4.1 Valid Transmission Message

The decoder informs the microcontroller of a valid transmission by taking the data line high for up to 500 ms. The controlling microcontroller must acknowledge by taking the clock line high. The decoder then takes the data line low. The microcontroller can then begin clocking a data stream out of the HCS500. The data stream consists of:

- Start bit ‘0’.
- Two Status bits [REPEAT, V_{LOW}].
- 4-bit function code [S3 S2 S1 S0].
- Stop bit ‘1’.
- Four bits indicating which block was used [TX3…TX0].
- Four bits indicating the number of transmitters learned into the decoder [CNT3…CNT0].
- 64 bits of the received transmission with the hopping code decrypted.

Note: Data is always clocked in/out Least Significant Bit (LSb) first.

The decoder will terminate the transmission of the data stream at any point where the clock is kept low for longer than 1 ms. Therefore, the microcontroller can only clock out the required bits. A maximum of 80 bits can be clocked out of the decoder.

FIGURE 4-1: HCS500 DECODER AND I/O INTERFACE LINES

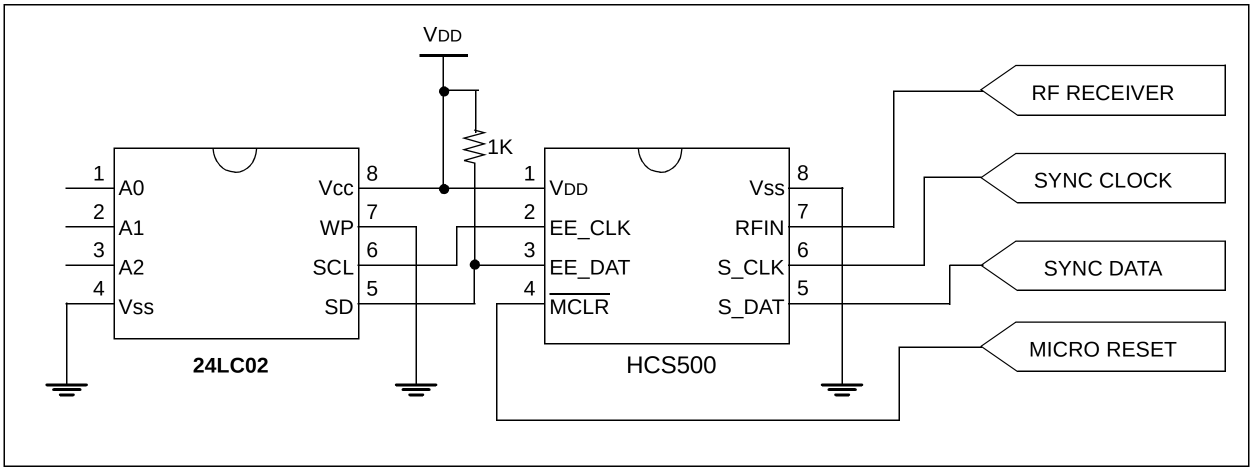
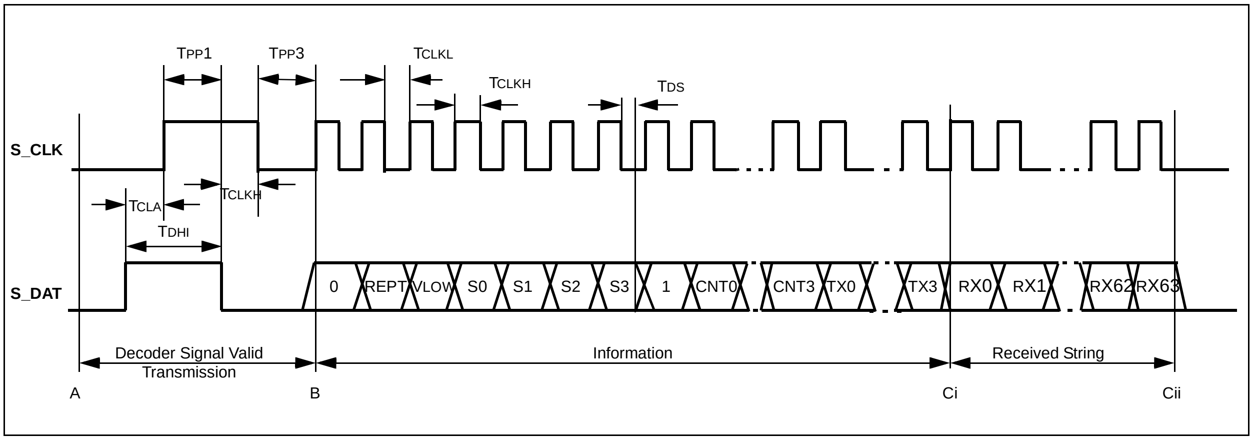


FIGURE 4-2: DECODER VALID TRANSMISSION MESSAGE



4.2 Command Mode

4.2.1 MICROCONTROLLER COMMAND MODE ACTIVATION

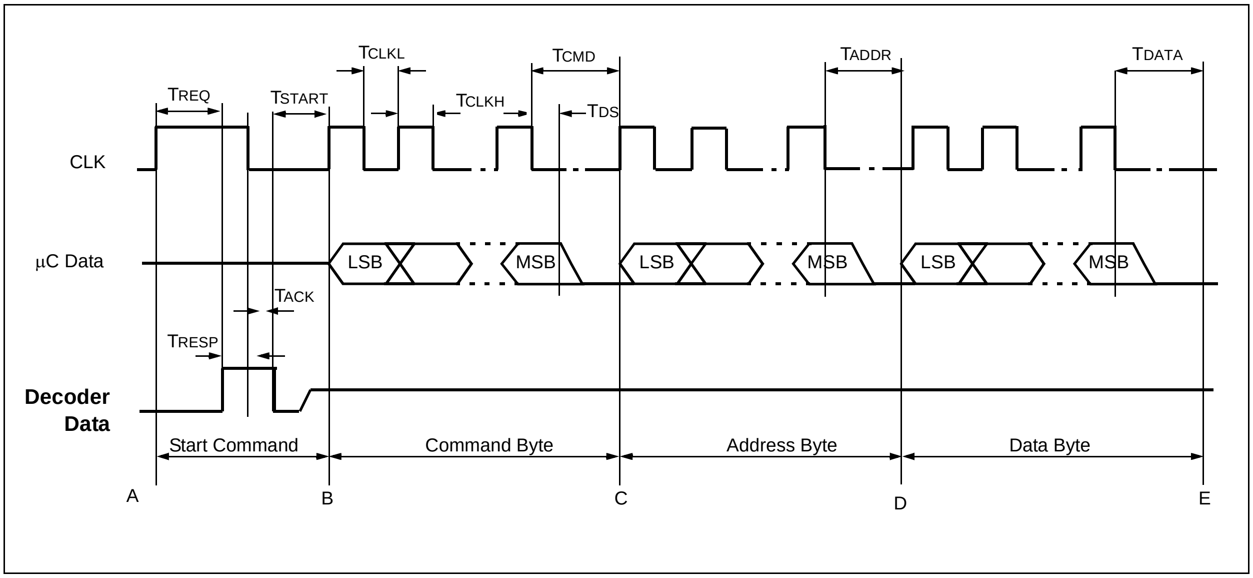
The microcontroller command consists of four parts. The first part activates the Command mode, the second part is the actual command, the third is the address accessed, and the last part is the data. The microcontroller starts the command by taking the clock line high for up to 500 ms. The decoder acknowledges the start-up sequence by taking the data line high. The microcontroller takes the clock line low, after which the decoder will take the data line low, tri-state the data line and wait for the command to be clock in. The data must be set up on the rising edge and will be sampled on the falling edge of the clock line.

4.2.2 COLLISION DETECTION

The HCS500 uses collision detection to prevent clashes between the decoder and microcontroller. Whenever the decoder receives a valid transmission the following sequence is followed:

- The decoder first checks to see if the clock line is high. If the clock line is high, the valid transmission notification is aborted, and the microcontroller Command mode request is serviced.
- The decoder takes the data line high and checks that the clock line does not go high within 50 µs. If the clock line goes high, the valid transmission notification is aborted and the Command mode request is serviced.
- If the clock line goes high after 50 µs but before 500 ms, the decoder will acknowledge by taking the data line low.
- The microcontroller can then start to clock out the 80-bit data stream of the received transmission.

FIGURE 4-3: MICROCONTROLLER COMMAND MODE ACTIVATION



4.2.3 COMMAND ACTIVATION TIMES

The command activation time (Table 4-1) is defined as the maximum time the microcontroller has to wait for a response from the decoder. The decoder will abort and service the command request. The response time depends on the state of the decoder when the Command mode is requested.

4.2.4 DECODER COMMANDS

The command byte specifies the operation required by the controlling microcontroller. Table 4-2 lists the commands.

TABLE 4-1: COMMAND ACTIVATION TIMES

Decoder State	Min	Max
While receiving transmissions	—	2.5 ms BPW_{MAX} = 2.7 ms
During the validation of a received transmission	—	3 ms
During the update of the sync counters	—	40 ms
During learn	—	170 ms

Note: *These parameters are characterized but not tested.

TABLE 4-2: DECODER COMMANDS

Instruction	Command Byte	Operation
`READ`	$F0_{16}$	Read a byte from user EEPROM
`WRITE`	$E1_{16}$	Write a byte to user EEPROM
`ACTIVATE_LRN`	$D2_{16}$	Activate a learn sequence on the decoder
`ERASE_ALL`	$C3_{16}$	Activate an erase all function on the decoder
`PROGRAM`	$B4_{16}$	Program manufacturer’s code and Configuration byte

4.2.5 READ BYTE/S FROM USER EEPROM

The Read command (Figure 4-4) is used to read bytes from the user EEPROM. The offset in the user EEPROM is specified by the address byte which is truncated to seven bits (C to D). After the address, a dummy byte must be clocked in (D to E). The EEPROM data byte is clocked out on the next rising edge of the clock line with the Least Significant bit first (E to F). Sequential reads are possible by repeating sequence E to F within 1 ms after the falling edge of the previous byte's Most Significant Bit (MSb) bit. During the sequential read, the address value will wrap after 128 bytes. The decoder will terminate the Read command if no clock pulses are received for a period longer than 1.2 ms.

4.2.6 WRITE BYTE/S TO USER EEPROM

The Write command (Figure 4-5) is used to write a location in the user EEPROM. The address byte is truncated to seven bits (C to D). The data is clocked in Least Significant bit first. The clock line must be asserted to initiate the write. Sequential writes of bytes are possible by clocking in the byte and then asserting the clock line (D – F). The decoder will terminate the Write command if no clock pulses are received for a period longer than 1.2 ms. After a successful write sequence, the decoder will acknowledge by taking the data line high and keeping it high until the clock line goes low.

FIGURE 4-4: READ BYTES FROM USER EEPROM

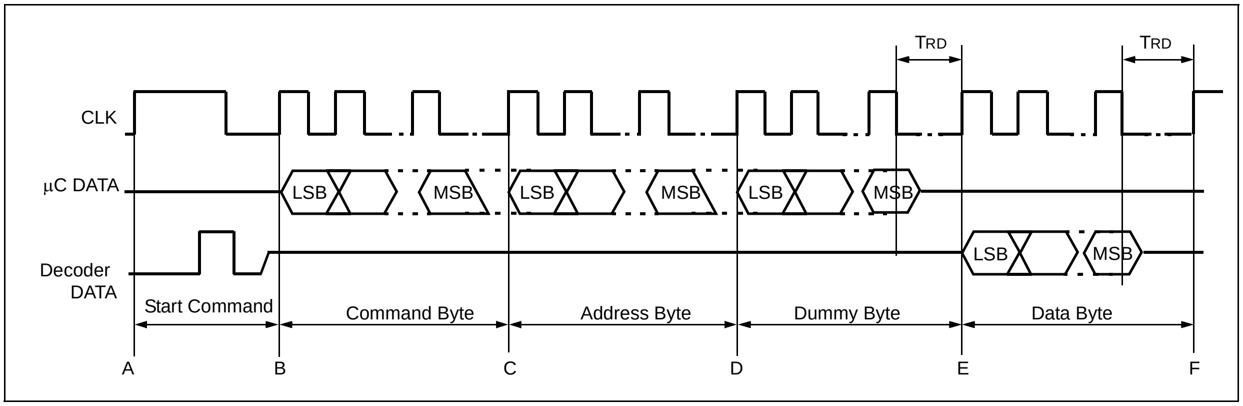
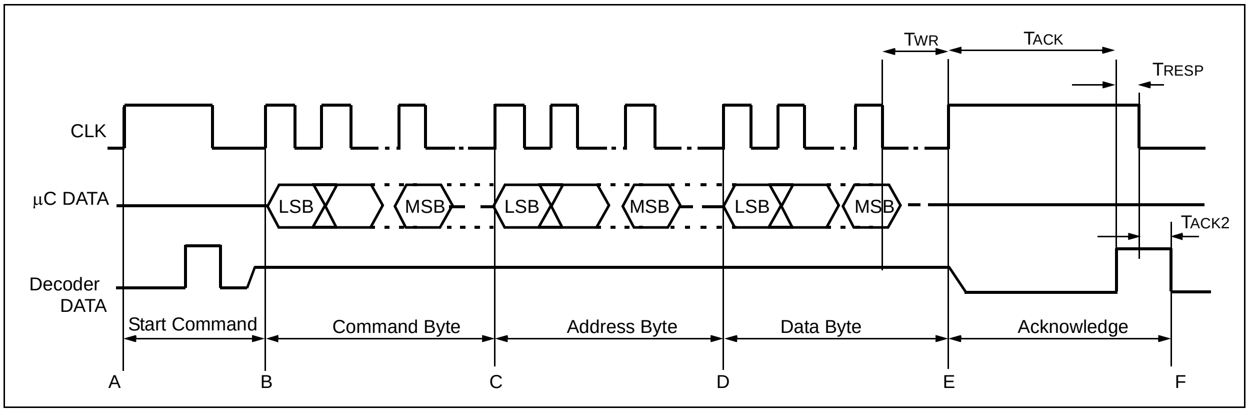


FIGURE 4-5: WRITE BYTES TO USER EEPROM



4.2.7 ACTIVATE LEARN

The activate Learn command (Figure 4-6) is used to activate a transmitter learning sequence on the decoder. The command consists of a Command mode activation sequence, a command byte, and two dummy bytes. The decoder will respond by taking the data line high to acknowledge that the command was valid and that learn is active.

Upon reception of the first transmission, the decoder will respond with a learn status message (Figure 4-7).

During learn, the decoder will acknowledge the reception of the first transmission by taking the data line high for 60 ms. The controlling microcontroller can clock out at most eight bits, which will all be zeros. All of the bits of the status byte are zero, and this is used to distinguish between a learn time-out status string and the first transmission received string. The controlling microcontroller must ensure that the clock line does not go high 60 ms after the falling edge of the data line, for this will terminate learn.

Upon reception of the second transmission, the decoder will respond with a learn status message (Figure 4-8).

The learn status message after the second transmission consists of the following:

- 1 Start bit.
- The function code [S3:S0] of the message is zero, indicating that this is a status string.
- The RESULT bit indicates the result of the learn sequence. The RESULT bit is set if successful and cleared otherwise.
- The OVR bit will indicate whether an exiting transmitter is over written. The OVR bit will be set if an existing transmitter is learned over.
- The [CNT3…CNT0] bits will indicate the number of transmitters learned on the decoder.
- The [TX3…TX0] bits indicate the block number used during the learning of the transmitter.

FIGURE 4-6: LEARN MODE ACTIVATION

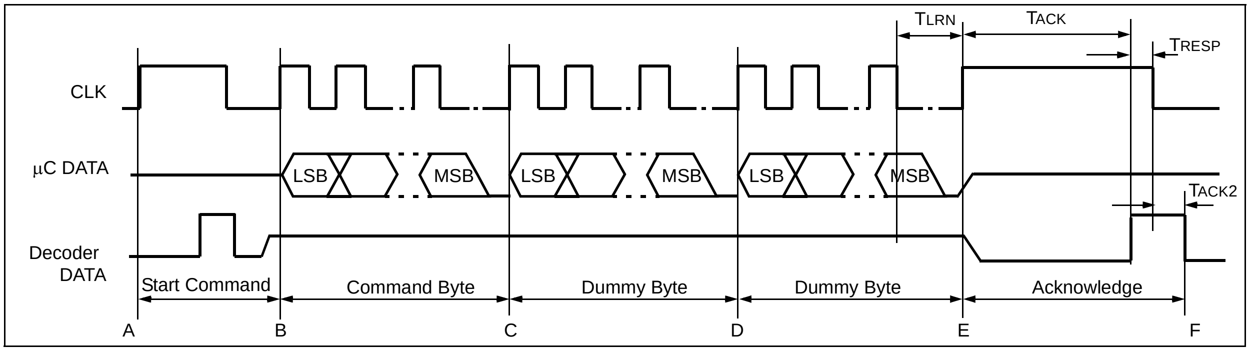


FIGURE 4-7: LEARN STATUS MESSAGE AFTER FIRST TRANSMISSION

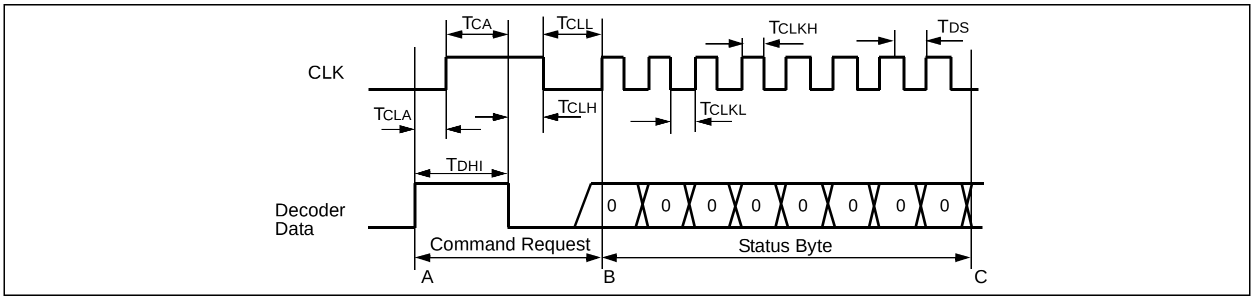
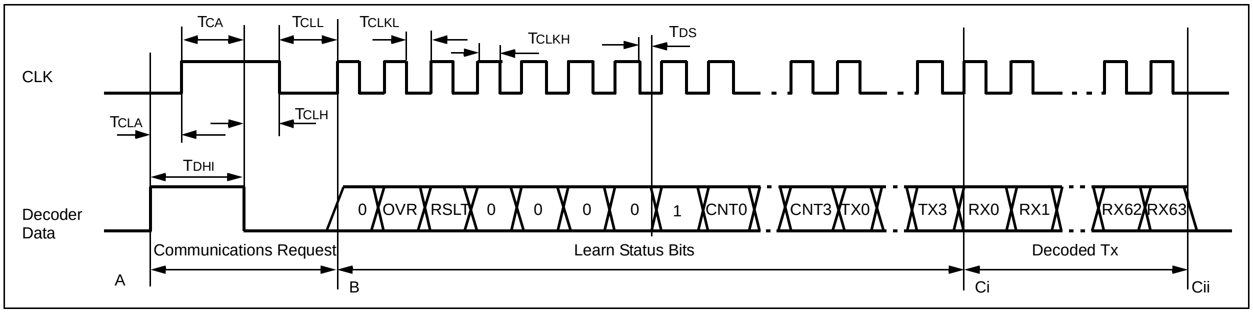


FIGURE 4-8: LEARN STATUS MESSAGE AFTER SECOND TRANSMISSION



4.2.8 ERASE ALL

The Erase All command (Figure 4-9) erases all the transmitters in the decoder. After the command and two dummy bytes are clocked in, the clock line must be asserted to activate the command. After a successful completion of an erase all command, the data line is asserted until the clock line goes low.

4.3 Stand-Alone Mode

The HCS500 decoder can also be used in stand-alone applications. The HCS500 will activate the data line for up to 500 ms if a valid transmission was received, and this output can be used to drive a relay circuit. To activate Learn or Erase All commands, a button must be connected to the CLK input. User feedback is indicated on an LED connected to the DATA output line. If the CLK line is pulled high, using the learn button, the LED will switch on. After the CLK line is kept high for longer than two seconds, the decoder will switch the LED line off, indicating that learn will be entered if the button is released. If the CLK line is kept high for another six seconds, the decoder will activate an ERASE_ALL Command.

Learn mode can be aborted by taking the clock line high until the data line goes high (LED switches on). During learn, the data line will give feedback to the user and, therefore, must not be connected to the relay drive circuitry.

Note: The REPS bit must be cleared in the Configuration byte in Stand-Alone mode.

After taking the clock low and before a transmitter is learn, any low-to-high change on the clock line may terminate learn. This has learn implications when a switch with contact bounce is used.

4.4 Erase All Command and Erase Command

The Table 4-3 describes two versions of the Erase All command.

TABLE 4-3: ERASE ALL COMMAND

Command Byte	Subcommand Byte	Description
$C3_{16}$	00_{16}	Erase all transmitters.
$C3_{16}$	01_{16}	Erase all transmitters except ‘1’. The first transmitter in memory is not erased.

Subcommand 01 can be used where a transmitter with permanent status is implemented in the microcontroller software. Use of subcommand 01 ensures that the permanent transmitter remains in memory even when all other transmitters are erased. The first transmitter learned after any of the following events is the first transmitter in memory and becomes the permanent transmitter:

1. Programming of the manufacturer’s code.
2. Erasing of all transmitters (subcommand 00 only).

4.5 Test mode

A special test mode is activated after:

1. Programming of the manufacturer’s code.
2. Erasing of all transmitters.

Test mode can be used to test a decoder before any transmitters are learned on it. Test mode enables testing of decoders without spending the time to learn a transmitter. Test mode is terminated after the first successful learning of an ordinary transmitter. In Test mode, the decoder responds to a test transmitter. The test transmitter has the following properties:

1. crypt key = manufacturer’s code.
2. Serial number = any value.
3. Discrimination bits = lower ten bits of the serial number.
4. Synchronization counter value = any value (synchronization information is ignored).

Because the synchronization counter value is ignored in Test mode, any number of test transmitters can be used, even if their synchronization counter values are different.

4.6 Power Supply Supervisor

Reliable operation of the HCS500 requires that the contents of the EEPROM memory be protected against erroneous writes. To ensure that erroneous writes do not occur after supply voltage “brown-out” conditions, the use of a proper power supply supervisor device (like Microchip part MCP100-450) is imperative.

FIGURE 4-9: ERASE ALL

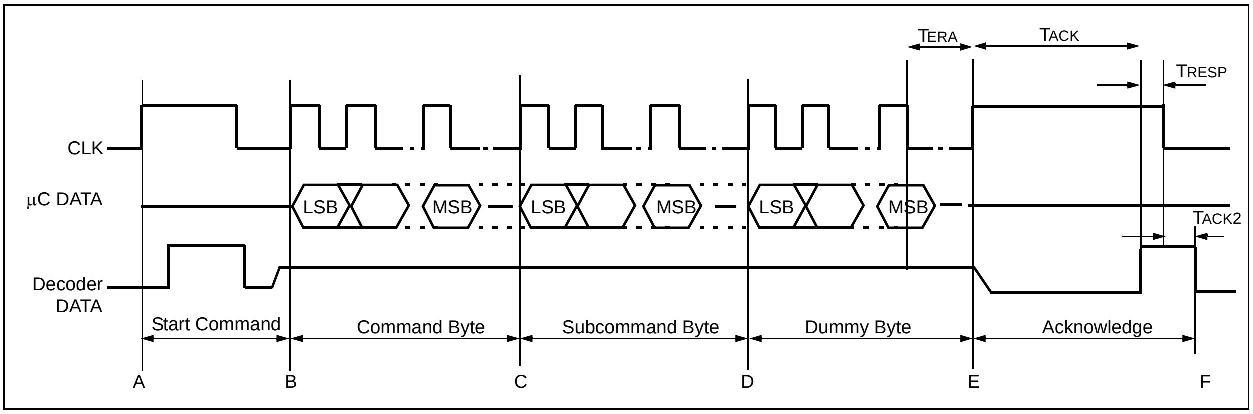


FIGURE 4-10: STAND-ALONE MODE LEARN/ERASE-ALL TIMING

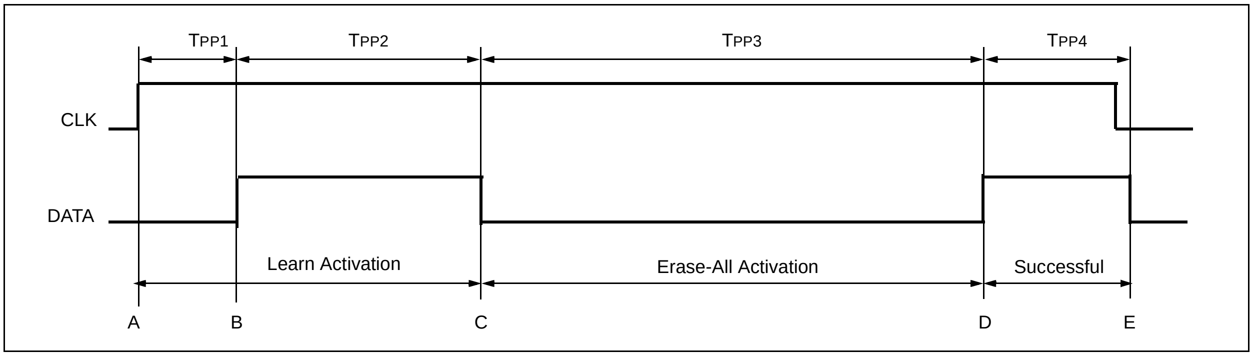
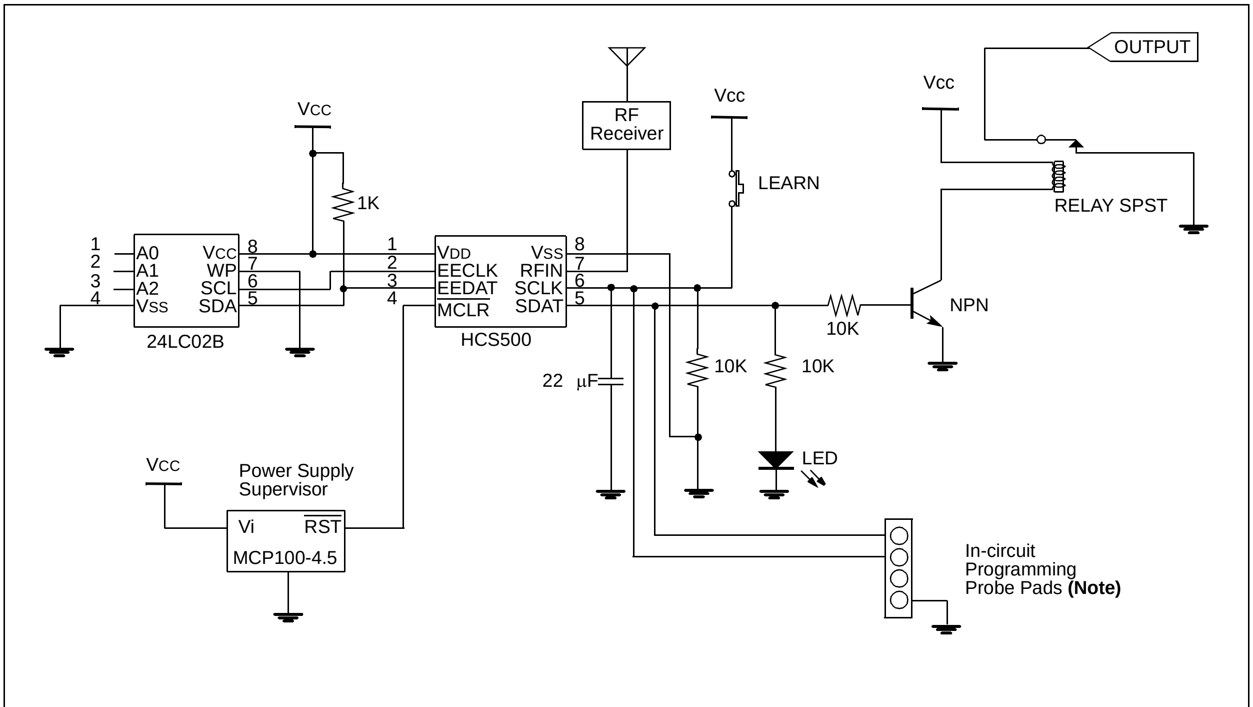


FIGURE 4-11: TYPICAL STAND-ALONE APPLICATION CIRCUIT



Note: Because each HCS500 is individually matched to its EEPROM, in-circuit programming is strongly recommended.

5.0 DECODER PROGRAMMING

The decoder uses a 2K, 24LC02B serial EEPROM. The memory is divided between system memory that stores the transmitter information (read protected) and user memory (read/write). Commands to access the user memory are described in Sections 4.2.5 and 4.2.6.

The following information stored in system memory needs to be programmed before the decoder can be used:

- 64-bit manufacturer's code
- Decoder Configuration byte

Note 1: These memory locations are read protected and can only be written to using the program command with the device powered up.

2: The contents of the system memory is encrypted by a unique 64-bit key that is stored in the HCS500. To initialize the system memory, the HCS500's program command must be used. The EEPROM and HCS500 are matched, and the devices must be kept together. In-circuit programming is therefore recommended.

5.1 Configuration Byte

The decoder is configured during initialization by setting the appropriate bits in the Configuration byte. The following table list the options:

TABLE 5-1: DECODER INITIALIZATION USING CONFIGURATION BYTE

Bit	Mnemonic	Description

0	LRN_MODE	Learning mode selection
		LRN_MODE = '0'—Normal Learn
		LRN_MODE = '1'—Secure Learn

1	LRN_ALG	Algorithm selection
		LRN_ALG = '0'—KEELOQ Decryption Algorithm
		LRN_ALG = '1'—XOR Algorithm

2	REPEAT	Repeat Transmission enable 0 = Disable 1 = Enabled
3	Not Used	Reserved
4	Not Used	Reserved
5	Not Used	Reserved
6	Not Used	Reserved
7	Not Used	Reserved

5.1.1 LRN_MODE

LRN_MODE selects between two learning modes. With LRN_MODE = 0, the Normal (serial number derived) mode is selected; with LRN_MODE = 1, the Secure (seed derived) mode is selected. See **Section 6.0 "Key Generation"** for more detail on learning modes.

5.1.2 LRN_ALG

LRN_ALG selects between the two available algorithms. With LRN_ALG = 0 selected, the KEELOQ decryption algorithm is selected; with LRN_ALG = 1, the XOR algorithm is selected. See **Section 6.0 "Key Generation"** for more detail on learning algorithms.

5.1.3 REPEAT

The HCS500 can be configured to indicate repeated transmissions. In a stand-alone configuration, repeated transmissions must be disabled.

5.2 Programming Waveform

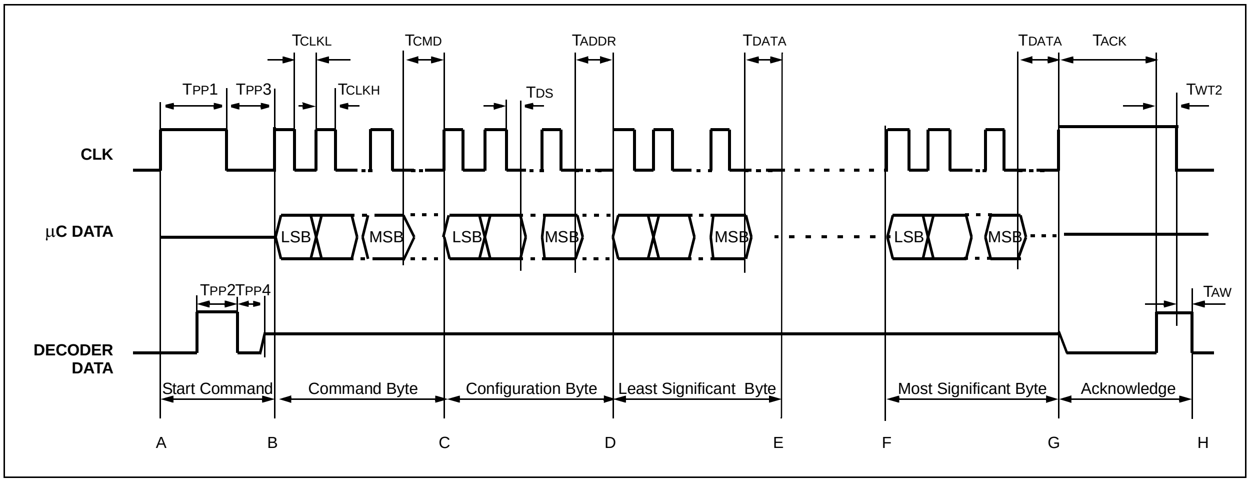
The programming command consists of the following:

- Command Request Sequence (A to B)
- Command Byte (B to C)
- Configuration Byte (C to D)
- Manufacturer's Code Eight Data Bytes (D to G)
- Activation and Acknowledge Sequence (G to H)

5.3 Programming Data String

A total of 80 bits are clocked into the decoder. The 8-bit command byte is clocked in first, followed by the 8-bit Configuration byte and the 64-bit manufacturer's code. The data must be clocked in Least Significant Bit (LSB) first. The decoder will then encrypt the manufacturer's code using the decoder's unique 64-bit EEPROM crypt key. After completion of the programming EEPROM, the decoder will acknowledge by taking the data line high (G to H). If the data line goes high within 30 ms after the clock goes high, programming also fails.

FIGURE 5-1: PROGRAMMING WAVEFORM



6.0 KEY GENERATION

The HCS500 supports three learning schemes which are selected during the initialization of the system EEPROM. The learning schemes are:

- Normal learn using the KEELOQ decryption algorithm
- Secure learn using the KEELOQ decryption algorithm
- Secure learn using the XOR algorithm

6.1 Normal (Serial Number derived) Learn using the KEELOQ Decryption Algorithm

This learning scheme uses the KEELOQ decryption algorithm and the 28-bit serial number of the transmitter to derive the crypt key. The 28-bit serial number is patched with predefined values as indicated below to form two 32-bit seeds.

SourceH = 60000000 00000000H + Serial Number |$_{\text{28 Bits}}$
SourceL = 20000000 00000000H + Serial Number |$_{\text{28 Bits}}$

Then, using the KEELOQ decryption algorithm and the manufacturer's code the crypt key is derived as follows:

KeyH$_{\text{Upper 32 bits}}$ = F$_{\text{KEELOQ Decryption}}$ (SourceH) |$_{\text{64-Bit Manufacturer's Code}}$
KeyL$_{\text{Lower 32 bits}}$ = F$_{\text{KEELOQ Decryption}}$ (SourceL) |$_{\text{64-Bit Manufacturer's Code}}$

6.2 Secure (Seed Derived) Learn using the KEELOQ Decryption Algorithm

This scheme uses the secure seed transmitted by the encoder to derive the two input seeds. The decoder always uses the lower 64 bits of the transmission to form a 60-bit seed. The upper four bits are always forced to zero.

For 32-bit seed encoders (HCS200, HCS201, HCS300, HCS301):

SourceH = Serial Number$_{\text{Lower 28 bits}}$
SourceL = Seed$_{\text{32 bits}}$

For 48-bit seed encoders (HCS360, HCS361):

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

For 60-bit seed encoders (HCS362, HCS365, HCS370, HCS410, HCS412, HCS473):

SourceH = Seed$_{\text{Upper 32 bits}}$ (with upper four bits set to zero)
SourceL = Seed$_{\text{Lower 32 bits}}$

The KEELOQ decryption algorithm and the manufacturer's code is used to derive the crypt key as follows:

KeyH$_{\text{Upper 32 bits}}$ = Decrypt (SourceH)$_{\text{64 Bit Manufacturer's Code}}$
KeyL$_{\text{Lower 32 bits}}$ = Decrypt (SourceL)$_{\text{64 Bit Manufacturer's Code}}$

6.3 Secure (Seed Derived) Learn using the XOR Algorithm

This scheme uses the seed transmitted by the encoder to derive the two input seeds. The decoder always use the lower 64 bits of the transmission to form a 60-bit seed. The upper four bits are always forced to zero.

For 32-bit seed encoders (HCS200, HCS201, HCS300, HCS301):

SourceH = Serial Number$_{\text{Lower 28 bits}}$
SourceL = Seed$_{\text{32 bits}}$

For 48-bit seed encoders (HCS360/HCS361):

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

For 60-bit seed encoders (HCS362, HCS365, HCS370, HCS410, HCS412, HCS473):

SourceH = Seed$_{\text{Upper 32 bits}}$ with upper four bits set to zero
SourceL = Seed$_{\text{Lower 32 bits}}$

Then, using the manufacturer's code the crypt key is derived as follows:

KeyH$_{\text{Upper 32 bits}}$ = SourceH **XOR** 64-Bit Manufacturer's Code$_{\text{Upper 32 bits}}$
KeyL$_{\text{Lower 32 bits}}$ = SourceL **XOR** 64-Bit Manufacturer's Code$_{\text{Lower 32 bits}}$

7.0 KEELOQ ENCODERS

7.1 Transmission Format (PWM)

The KEELOQ encoder transmission is made up of several parts (Figure 7-1). 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 34/35 bits of non-encrypted data. Each transmission is followed by a guard period before another transmission can begin. The code hopping 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-code hopping 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}.

7.2 Code Word Organization

The HCS encoder transmits a 66/69-bit code word when a button is pressed. The 66/69-bit word is constructed from a code hopping portion and a non-code hopping portion (Figure 7-2).

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

The **Non-encrypted Data** is made up from two Status bits, four function bits, and the 28/32-bit serial number.

FIGURE 7-1: TRANSMISSION FORMAT (PWM)

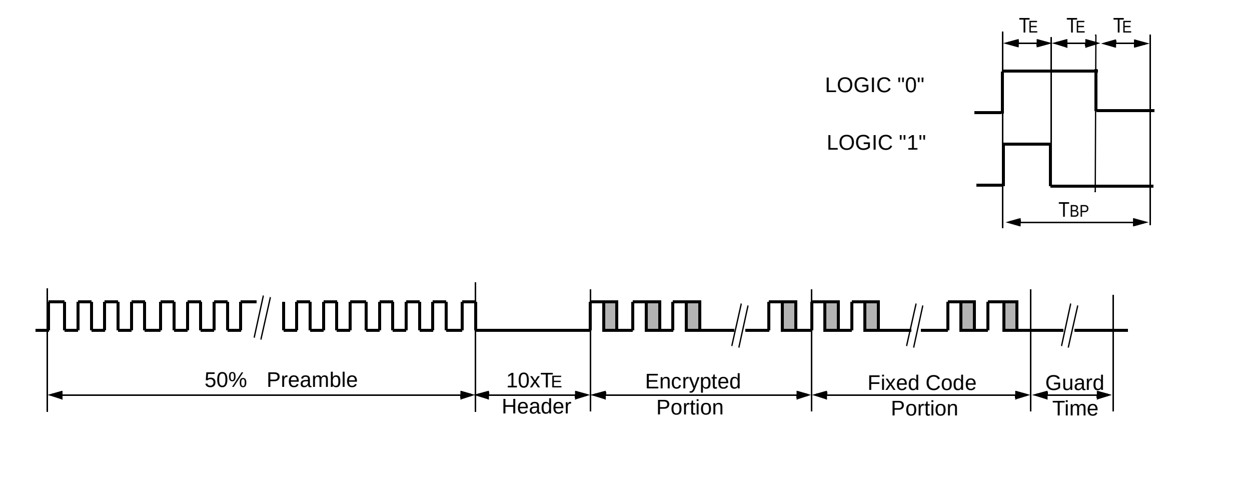
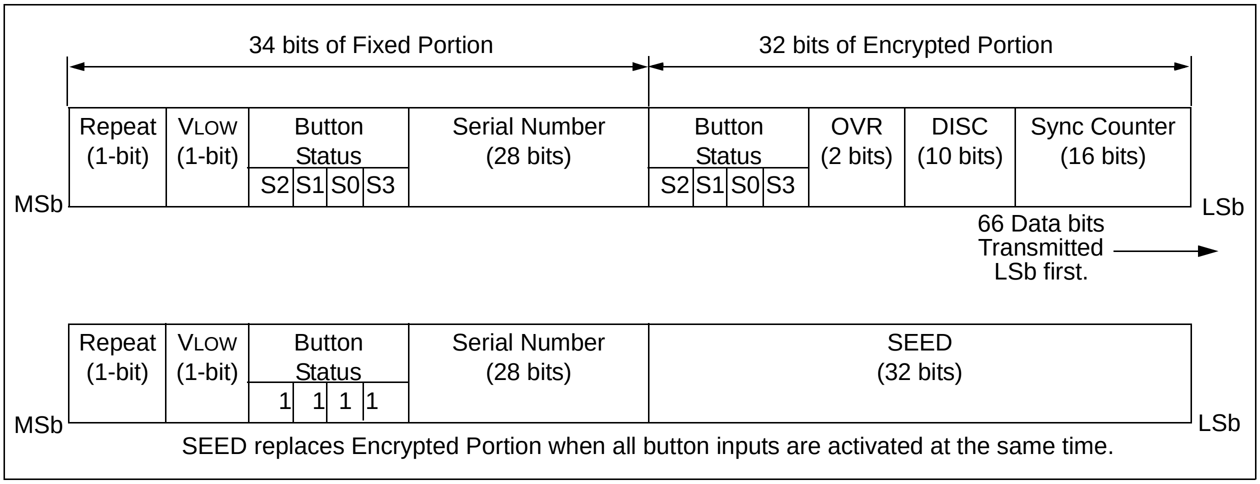


FIGURE 7-2: CODE WORD ORGANIZATION



Repeat (1-bit)	VLOW (1-bit)	Button Status				Serial Number (28 bits)	Button Status				OVR (2 bits)	DISC (10 bits)	Sync Counter (16 bits)
		S2	S1	S0	S3		S2	S1	S0	S3			

Repeat (1-bit)	VLOW (1-bit)	Button Status				Serial Number (28 bits)	SEED (32 bits)
		1	1	1	1		

SEED replaces Encrypted Portion when all button inputs are activated at the same time.

8.0 DEVELOPMENT SUPPORT

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

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

8.1 MPLAB Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature

Built-in support for Bugzilla issue tracker

8.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

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

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. 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. MPLAB XC Compiler uses the assembler to produce its object 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 X IDE compatibility

8.3 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 X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

8.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. 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

8.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC 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 X IDE compatibility

8.6 MPLAB X SIM Software Simulator

The MPLAB X 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 X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC 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

8.7 MPLAB REAL ICE In-Circuit Emulator System

The 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 all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

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 (RJ-11) 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 X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

8.8 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB 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.

8.9 PICkit 3 In-Circuit Debugger/ Programmer

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 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 a 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™ (ICSP™).

8.10 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.

8.11 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.

8.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex

Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

9.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings†

Ambient temperature under bias	-40°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)**	700 mW
Maximum current out of V_{SS} pin	200 mA
Maximum current into V_{DD} pin	150 mA
Input clamp current, I_{IK} (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	25 mA

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

† **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.

9.1 Standard Operating Conditions

The standard operating conditions for any device are defined as:

Operating Voltage: $V_{DDMIN} \le V_{DD} \le V_{DDMAX}$
Operating Temperature: $T_{A\_MIN} \le T_A \le T_{A\_MAX}$

V_{DD} — Operating Supply Voltage[(1)]

HCS500

V_{DDMIN} (Fosc ≤ 16 MHz)	+2.3V
V_{DDMIN} (Fosc ≤ 32 MHz)	+2.5V
V_{DDMAX}	+5.5V

T_A — Operating Ambient Temperature Range

Commercial Temperature (C)

T_{A_MIN}	0°C
T_{A_MAX}	70°C

Industrial Temperature (I)

T_{A_MIN}	-40°C
T_{A_MAX}	+85°C

Note 1: See DC Characteristics: Supply Voltage.

TABLE 9-1: DC CHARACTERISTICS

Section 9.1 "Standard Operating Conditions"					
Symbol	**Parameters**	**Min.**	**Typ.**	**Max.**	**Units**
VDD	Supply voltage	3.0	—	5.5	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 0.3	2.4 5	mA µA
IPD	Power-Down Current	— —	0.25 0.3	4 5	µA µA
VIL	Input low voltage	Vss Vss Vss	— — —	0.8 0.15 VDD 0.15 VDD	V V V
VIH	Input high voltage	2.0 0.25 VDD + 0.8 0.85 VDD	— — —	VDD VDD VDD	V V V
VOL	Output low voltage	—	—	0.6	V

† 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.

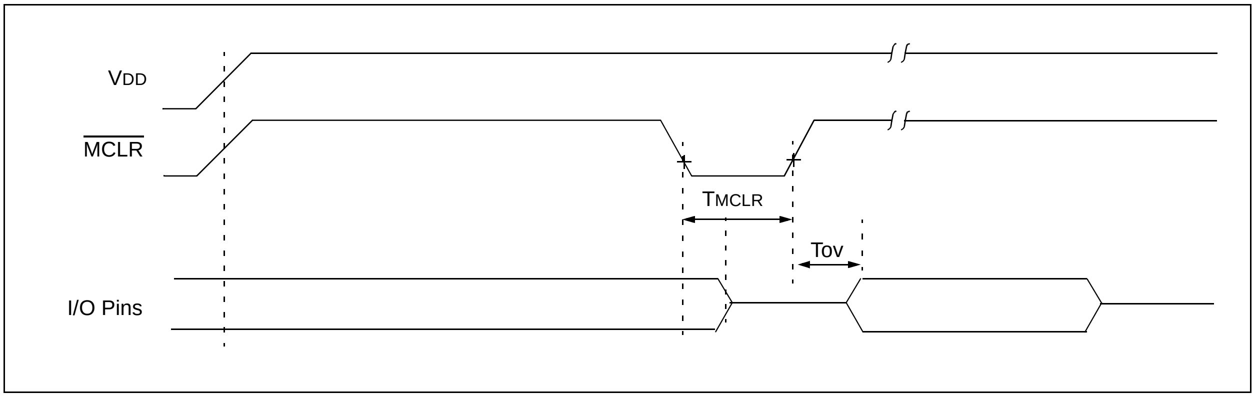
Note: Negative current is defined as coming out of the pin.

TABLE 9-2: AC CHARACTERISTICS

Section 9.1 "Standard Operating Conditions"					
Symbol	**Parameters**	**Min.**	**Typ.**	**Max.**	**Units**
TE	Transmit elemental period	65	—	660	µs
TOD	Output delay	48	75	237	ms
TMCLR	$\overline{\text{MCLR}}$ low time	150	—	—	ns
TOV	Time output valid	—	150	222	ms

* These parameters are characterized but not tested.

FIGURE 9-1: RESET WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



9.2 AC Electrical Characteristics

9.2.1 COMMAND MODE ACTIVATION

TABLE 9-3: COMMAND MODE ACTIVATION SPECIFICATIONS

Section 9.1 "Standard Operating Conditions"					
Symbol	**Parameters**	**Min.**	**Typ.**	**Max.**	**Units**
TREQ	Command request time	0.0150	—	500	ms
TRESP	Microcontroller request acknowledge time	—	—	1	ms
TACK	Decoder acknowledge time	—	—	30	µs
TSTART	Start Command mode to first command bit	20	—	1000	µs
TCLKH	Clock high time	20	—	1000	µs
TCLKL	Clock low time	20	—	1000	µs
FCLK	Clock frequency	500	—	25000	Hz
TDS	Data hold time	14	—	—	µs
TCMD	Command validate time	—	—	10	µs
TADDR	Address validate time	—	—	10	µs
TDATA	Data validate time	—	—	10	µs

* These parameters are characterized but not tested.

9.2.2 READ FROM USER EEPROM COMMAND

TABLE 9-4: READ FROM EEPROM COMMAND

Section 9.1 "Standard Operating Conditions"					
Symbol	**Parameters**	**Min.**	**Typ.**	**Max.**	**Units**
TRD	Decoder EEPROM read time	400	—	1500	µs

* These parameters are characterized but not tested.

9.2.3 WRITE TO USER EEPROM COMMAND

TABLE 9-5: WRITE TO USER EEPROM COMMAND

Section 9.1 "Standard Operating Conditions"					
Symbol	**Parameters**	**Min.**	**Typ.**	**Max.**	**Units**
TWR	Write command activation time	20	—	1000	µs
TACK	EEPROM write acknowledge time	—	—	10	ms
TRESP	Microcontroller acknowledge response time	20	—	1000	µs
TACK2	Decoder response acknowledge time	—	—	10	µs

* These parameters are characterized but not tested.

9.2.4 ACTIVATE LEARN COMMAND IN MICRO MODE

TABLE 9-6: ACTIVATE LEARN COMMAND IN MICRO MODE

Section 9.1 "Standard Operating Conditions"					
Symbol	**Parameters**	**Min.**	**Typ.**	**Max.**	**Units**
TLRN	Learn command activation time	20	—	1000	µs
TACK	Decoder acknowledge time	—	—	20	µs
TRESP	Microcontroller acknowledge response time	20	—	1000	µs
TACK2	Decoder data line low	—	—	10	µs

* These parameters are characterized but not tested.

9.2.5 ACTIVATE LEARN COMMAND IN STAND-ALONE MODE

TABLE 9-7: ACTIVATE LEARN COMMAND IN STAND-ALONE MODE

Section 9.1 "Standard Operating Conditions"					
Symbol	**Parameters**	**Min.**	**Typ.**	**Max.**	**Units**
TPP1	Command request time	—	—	100	ms
TPP2	Learn command activation time	—	—	2	s
TPP3	Erase-all command activation time	—	—	6	s

* These parameters are characterized but not tested.

9.2.6 LEARN STATUS STRING

TABLE 9-8: LEARN STATUS STRING

Section 9.1 "Standard Operating Conditions"					
Symbol	**Parameters**	**Min.**	**Typ.**	**Max.**	**Units**
TDHI	Command request time	—	—	500	ms
TCLA	Microcontroller command request time	0.005	—	500	ms
TCA	Decoder request acknowledge time	—	—	10	µs
TCLH	Clock high hold time			1.2	ms
TCLL	Clock low hold time	0.020	—	1.2	ms
TCLKH	Clock high time	20	—	1000	µs
TCLKL	Clock low time	20	—	1000	µs
FCLK	Clock frequency	500	—	25000	Hz
TDS	Data hold time	—	—	5	µs

* These parameters are characterized but not tested.

9.2.7 ERASE ALL COMMAND

TABLE 9-9: TIMING SPECIFICATION FOR ERASE ALL COMMAND

Section 9.1 "Standard Operating Conditions"					
Symbol	**Parameters**	**Min.**	**Typ.**	**Max.**	**Units**
TERA	Learn command activation time	20	—	1000	µs
TACK	Decoder acknowledge time	20	—	210	ms
TRESP	Microcontroller acknowledge response time	20	—	1000	µs
TACK2	Decoder data line low	—	—	10	µs

* These parameters are characterized but not tested.

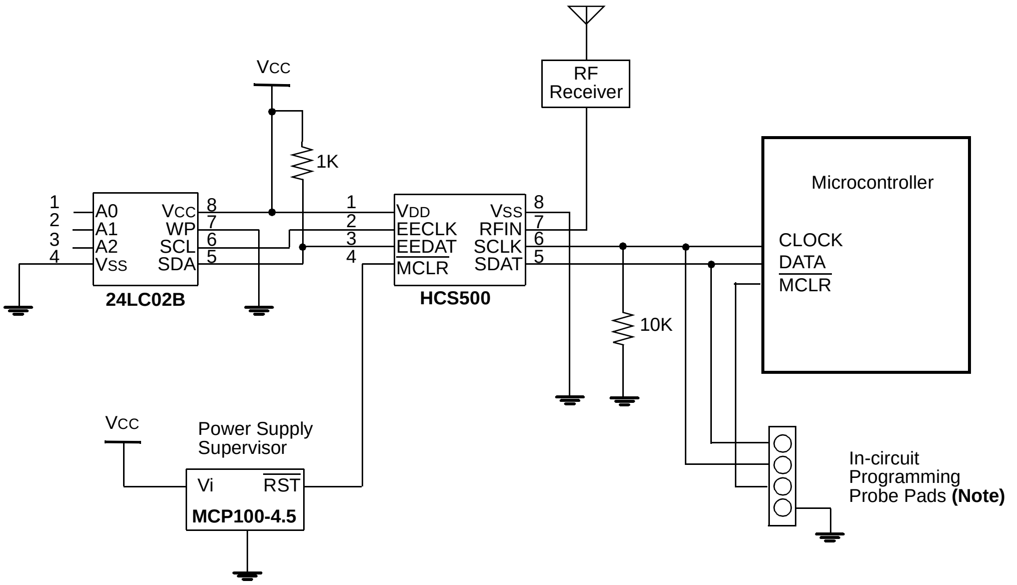
9.2.8 PROGRAMMING COMMAND

TABLE 9-10: TIMING SPECIFICATION FOR PROGRAMMING COMMAND

Section 9.1 "Standard Operating Conditions"					
Symbol	**Parameters**	**Min.**	**Typ.**	**Max.**	**Units**
TPP1	Command request time	—	—	500	ms
TPP2	Decoder acknowledge time	—	—	1	ms
TPP3	Start Command mode to first command bit	20	—	1000	µs
TPP4	Data line low before tri-stated	—	—	5	µs
TCLKH	Clock high time	20	—	1000	µs
TCLKL	Clock low time	20	—	1000	µs
FCLK	Clock frequency	500	—	25000	Hz
TDS	Data hold time	—	—	5	µs
TCMD	Command validate time	—	—	10	µs
TACK	Command acknowledge time	30	—	240	ms
TWT2	Acknowledge respond time	20	—	1000	µs
TALW	Data low after clock low	—	—	10	µs

* These parameters are characterized but not tested.

FIGURE 9-2: TYPICAL MICROCONTROLLER INTERFACE CIRCUIT

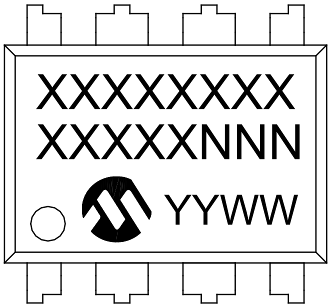


Note: Because each HCS500 is individually matched to its EEPROM, in-circuit programming is strongly recommended.

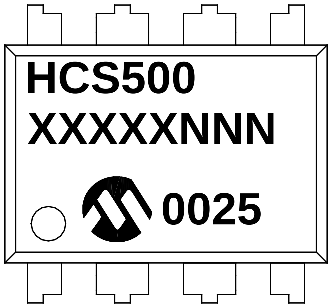
10.0 PACKAGING INFORMATION

10.1 Package Marking Information

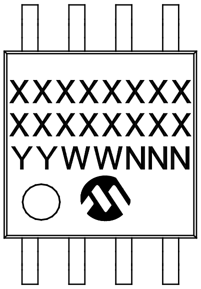
8-Lead PDIP (300 mil)



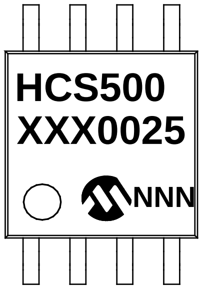
Example



8-Lead SOIJ (5.28 mm)



Example



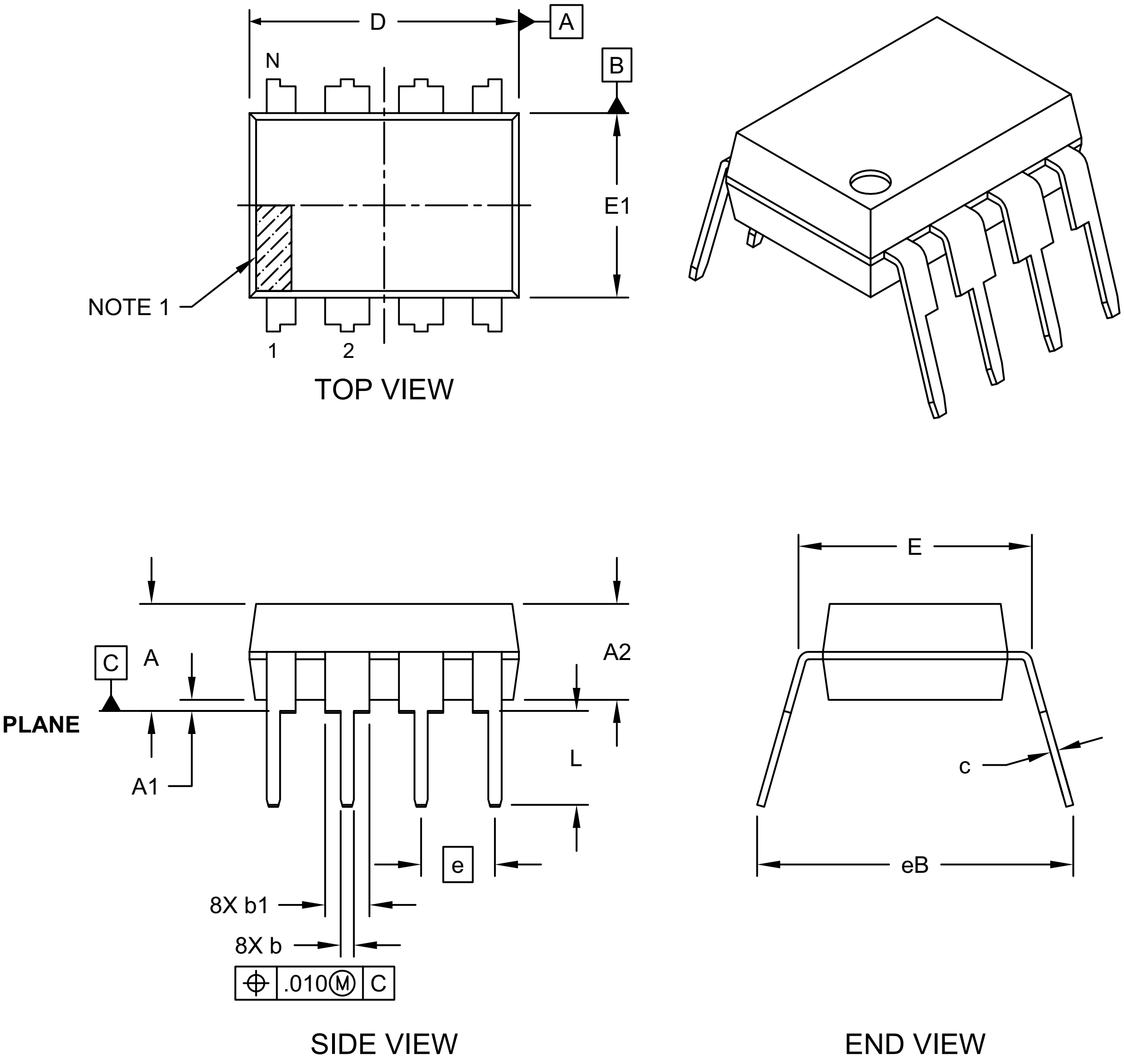
Legend:	XX...X	Customer specific information*
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code

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.

10.2 Package Details

8-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



TOP VIEW

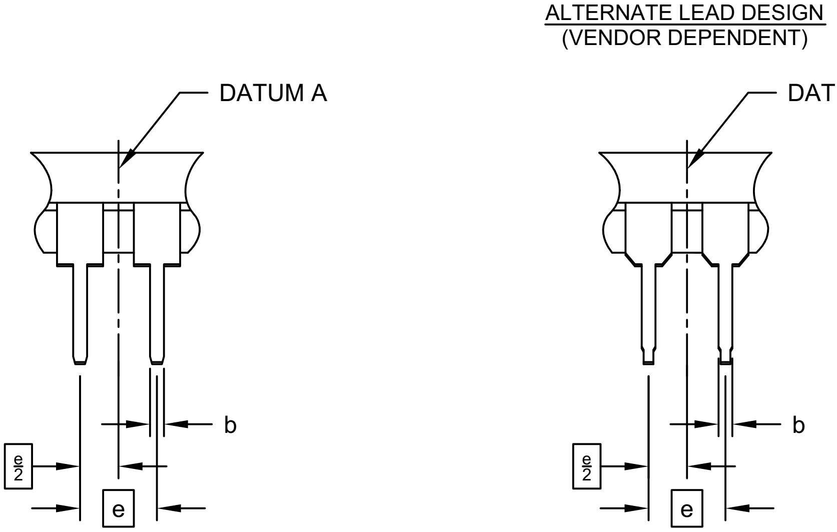
SIDE VIEW

END VIEW

Microchip Technology Drawing No. C04-018D Sheet 1 of 2

8-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		8	
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	.290	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.348	.365	.400
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	-	-	.430

Notes:

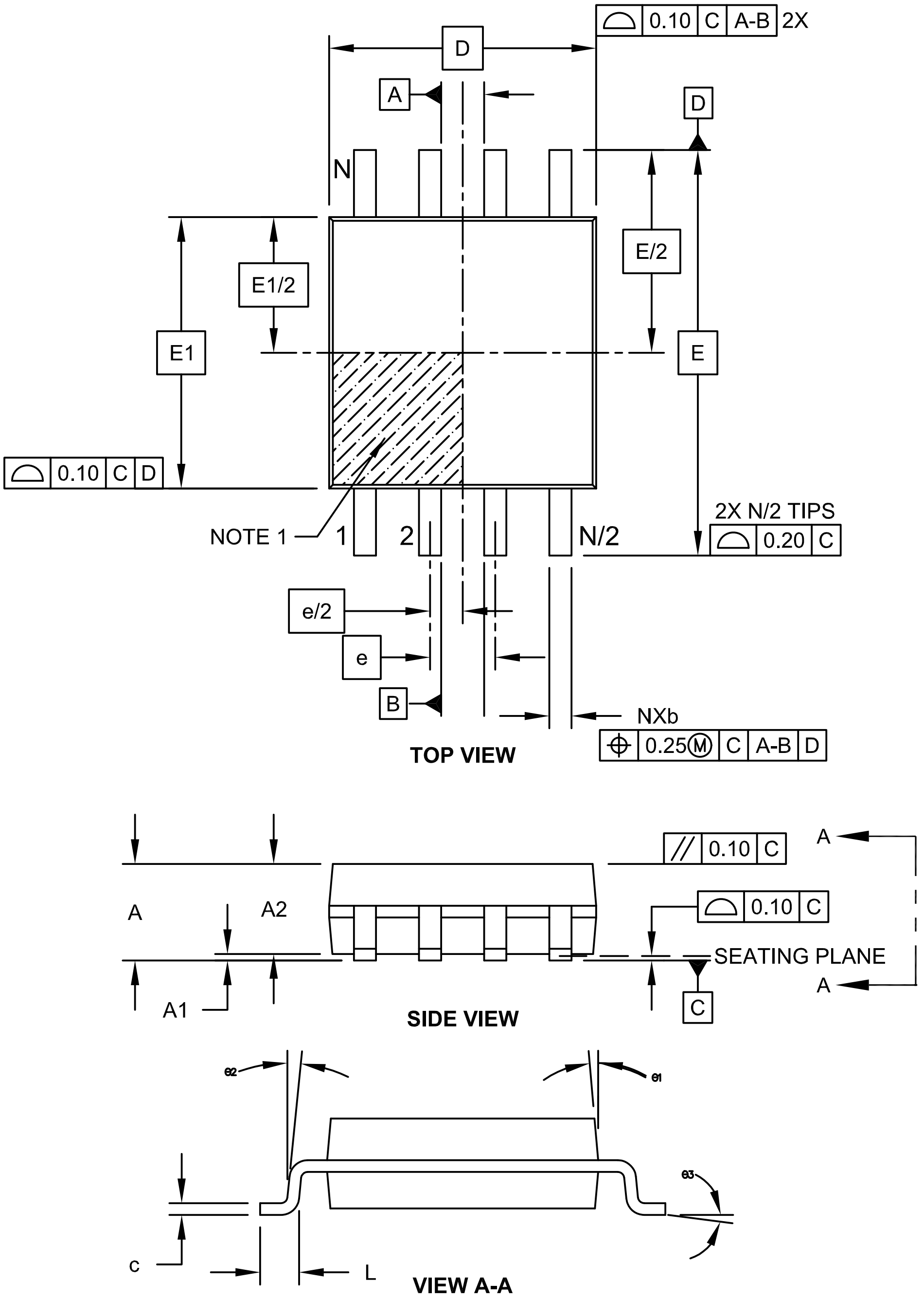
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-018D Sheet 2 of 2

8-Lead Plastic Small Outline (SM) - Medium, 5.28 mm Body [SOIJ]

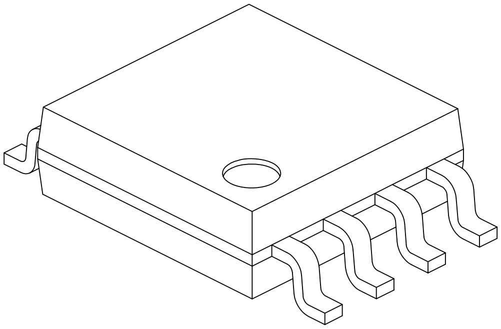
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-056C Sheet 1 of 2

8-Lead Plastic Small Outline (SM) - Medium, 5.28 mm Body [SOIJ]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	8		
Pitch	e	1.27 BSC		
Overall Height	A	1.77	-	2.03
Standoff §	A1	0.05		0.25
Molded Package Thickness	A2	1.75	-	1.98
Overall Width	E	7.94 BSC		
Molded Package Width	E1	5.25 BSC		
Overall Length	D	5.26 BSC		
Foot Length	L	0.51	-	0.76
Lead Thickness	c	0.15	-	0.25
Lead Width	b	0.36	-	0.51
Mold Draft Angle	Θ1	-	-	15°
Lead Angle	Θ2	0°	-	8°
Foot Angle	Θ3	0°	-	8°

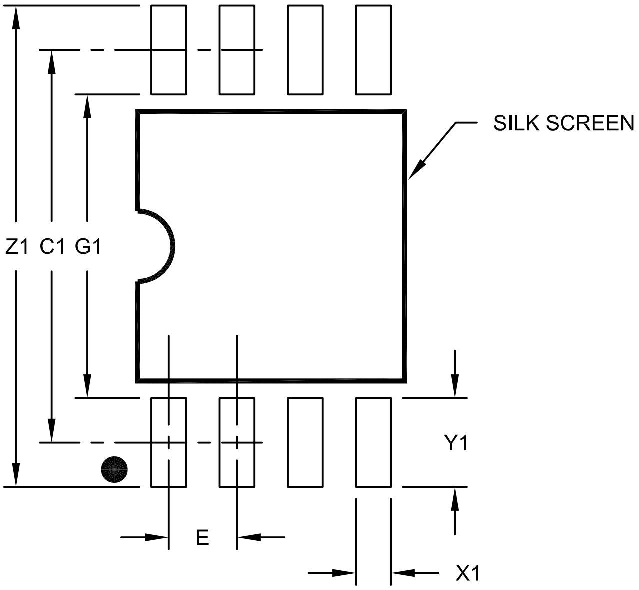
Notes:

1. SOIJ, JEITA/EIAJ Standard, Formerly called SOIC
2. § Significant Characteristic
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25mm per side.

Microchip Technology Drawing No. C04-056C Sheet 2 of 2

8-Lead Plastic Small Outline (SM) - Medium, 5.28 mm Body [SOIJ]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Overall Width	Z1			9.00
Contact Pad Spacing	C1		7.30	
Contact Pad Width (X8)	X1			0.65
Contact Pad Length (X8)	Y1			1.70
Distance Between Pads	G1	5.60		
Distance Between Pads	G	0.62		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2056C

APPENDIX A: REVISION HISTORY

Revision A (September 1998)

Original Release

Revision B (September 1998)

Revision C (December 2001)

Revision D (June 2011)

Updated the following sections: Development Support, The Microchip Web Site, Reader Response and HCS500 Product Identification System; Added new section: Appendix A; Minor formatting and text changes were incorporated throughout the document.

Revision E (February 2015)

Updated Packaging Information Chapter; Other minor corrections.

THE MICROCHIP WEB SITE

Microchip provides online support via our web site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X](1)	-	X	/XX	XXX
Device	Tape and Reel Option		Temperature Range	Package	Pattern

Device:	**HCS500**	
Tape and Reel Option:	Blank T	= Standard packaging (tube or tray) = Tape and Reel(1)
Temperature Range:	I C	= -40°C to +85°C (Industrial) = 0°C to +70°C (Extended)
Package:	PDIP SOIJ	= P = SM
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

a) HCS500T - I/ST 301
Tape and Reel,
Industrial temperature,
PDIP package
QTP pattern #301

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC^{32} logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KleerNet, KleerNet logo, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-63277-079-0

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983
Indianapolis

Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Pforzheim
Tel: 49-7231-424750

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

01/27/15