MICROCHIP

HCS500

KEELOQ™ Code Hopping Decoder

FEATURES

Security

- Encrypted storage of manufacturer's code
- Encrypted storage of encoder 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 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

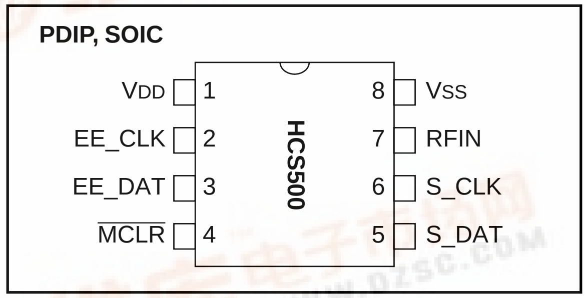
Compatible Encoders

- HCS200, HCS300, HCS301, HCS360, HCS410 (PWM Mode)

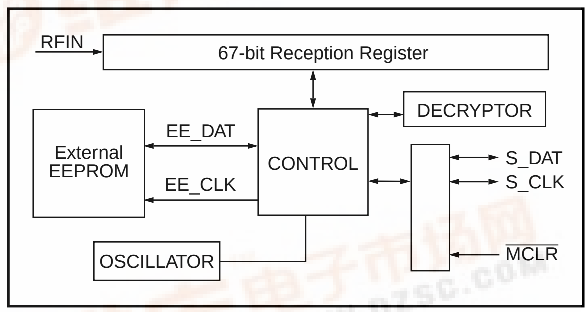
DESCRIPTION

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.

PACKAGE TYPE



BLOCK DIAGRAM



The manufacturer's code, encoder 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.

KEELOQ is a registered trademark of Microchip Technology Inc.
*Code hopping patents issued in Europe, U.S.A; and R.S.—US:5,517,187; Europe: 0459781

1.0 KEELOQ SYSTEM OVERVIEW

1.1 Key Terms

- Manufacturer's Code – A 64-bit word, unique to each manufacturer, used to produce a unique encoder key in each transmitter.
- Encoder Key – A 64-bit key, unique for each transmitter. The encoder key controls the KeeLoq decryption algorithm and is stored in EEPROM on the decoder device.
- Learn – The receiver uses information that is transmitted to derive the transmitter's encoder key, decrypt the discrimination value, and the synchronization counter in learning mode. The encoder key is a function of the manufacturer's code and the device serial number and/or seed value.

The HCS encoders and decoders employ the KeeLoq code hopping technology and a KeeLoq encryption algorithm to achieve a high level of security. Code hopping is a method by which the code transmitted from the transmitter to the receiver is different every time a button is pushed. This method, coupled with a transmission length of 66 bits, virtually eliminates the use of code 'grabbing' or code 'scanning'.

1.2 HCS Encoder Overview

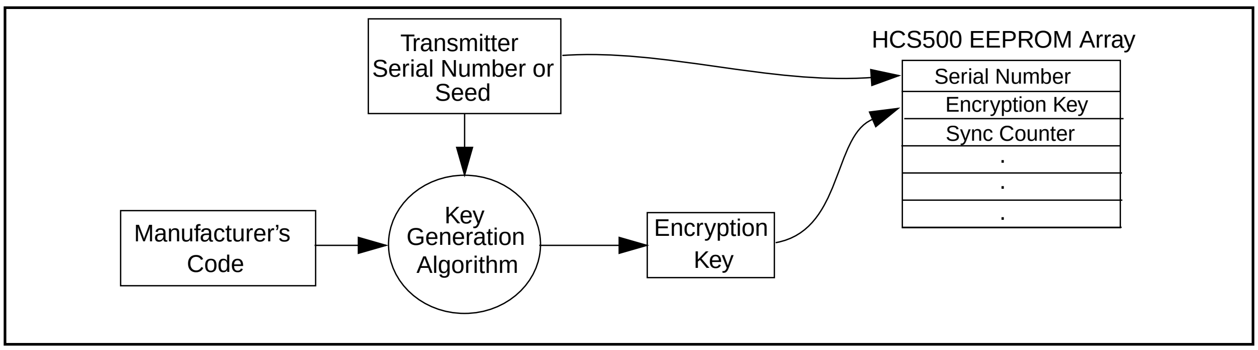
The HCS encoders have a small EEPROM array which must be loaded with several parameters before use. The most important of these values are:

- An encoder 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 encoder 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 ENCRYPTION KEY DURING PRODUCTION



The 16-bit synchronization counter is the basis for the transmitted code changing for each transmission and is updated each time a button is pressed. Because of the complexity of the KEELOQ encryption algorithm, a change in one bit of the synchronization counter value will result in a large change in the actual transmitted code. There is a relationship (Figure 1-2) between the encoder key values in EEPROM and how they are used in the encoder. Once the encoder detects that a button has been pressed, the encoder reads the button and updates the synchronization counter. The synchronization value is then combined with the encoder key in the KEELOQ encryption algorithm, and the output is 32 bits of encrypted information. This data will change with every button press, hence, it is referred to as the code hopping portion of the code word. The 32-bit code hopping portion is combined with the button information and the serial number to form the code word transmitted to the receiver.

1.3 HCS Decoder Overview

Before a transmitter and receiver can work together, the receiver must first 'learn' and store certain information from the transmitter. This information includes a 'check value' of the serial number, the encoder key, and current synchronization counter value.

When a validly formatted message is detected, the receiver first compares the serial number. If the serial number check value is from a learned transmitter, the message is decrypted. Next, the receiver checks the decrypted synchronization counter value against what is stored in memory. If the synchronization counter value is verified, then a valid transmission message is sent. 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: BASIC OPERATION OF A CODE HOPPING TRANSMITTER (ENCODER)

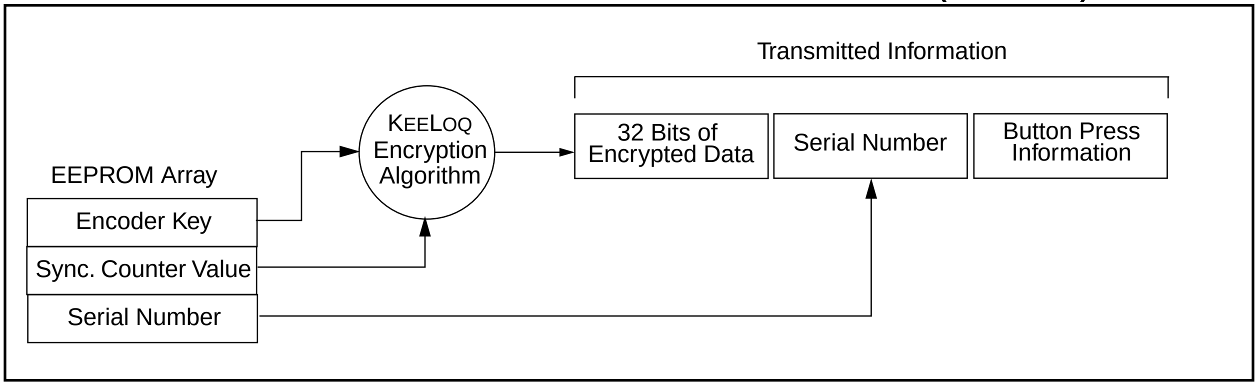
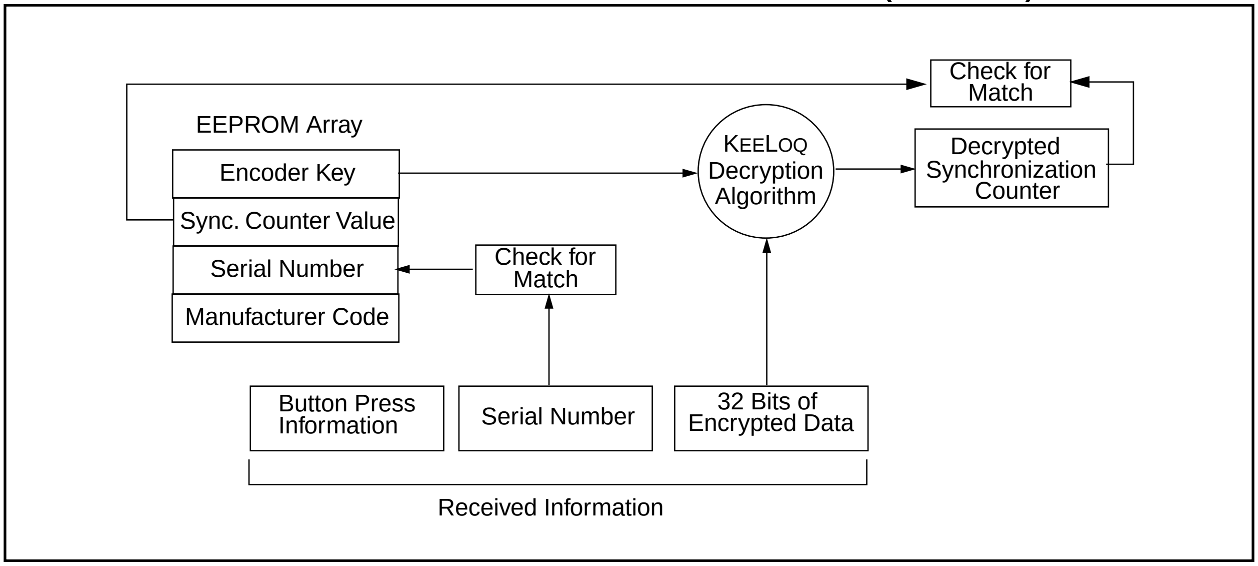


FIGURE 1-3: BASIC OPERATION OF A CODE HOPPING RECEIVER (DECODER)



2.0 PIN ASSIGNMENT

PIN	Decoder Function	I/O(1)	Buffer Type(1)	Description
1	V_{DD}	P	—	Power Connection
2	EE_CLK	O	TTL	Clock to I^2C™ EEPROM
3	EE_DAT	I/O	TTL	Data to I^2C EEPROM
4	$\overline{MCLR}$	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 encoder key
- The current synchronization counter value

The decoder must also store the manufacturer's code (Section 1.2) 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).

3.1.1 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.

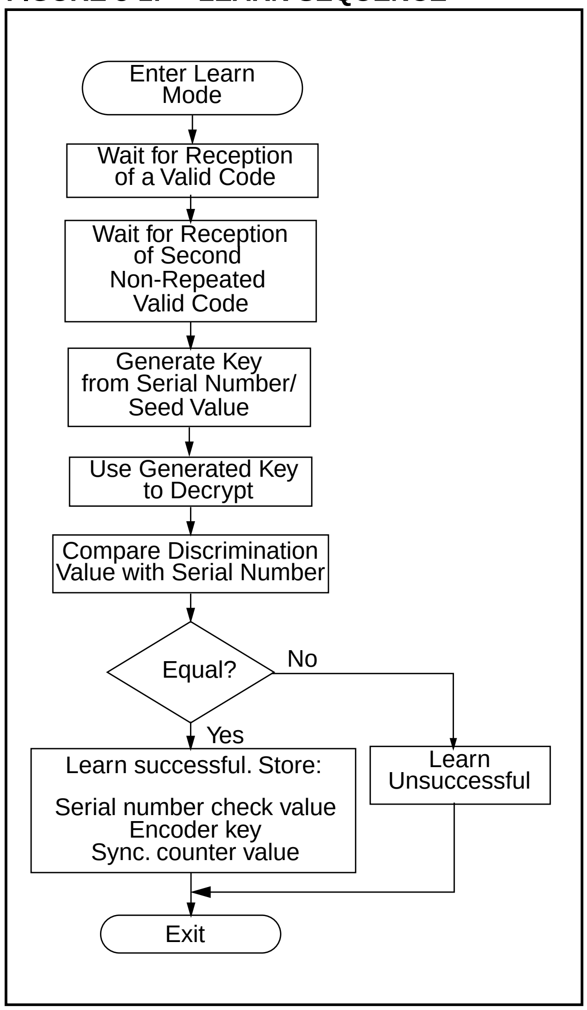
3: Learning a transmitter with an encoder key that is identical to a transmitter already in memory replaces the existing transmitter. In practice, this means that all transmitters should have unique encoder keys. Learning a previously learned transmitter does not use any additional memory slots.

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 encoder 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



3.2 Validation of Codes

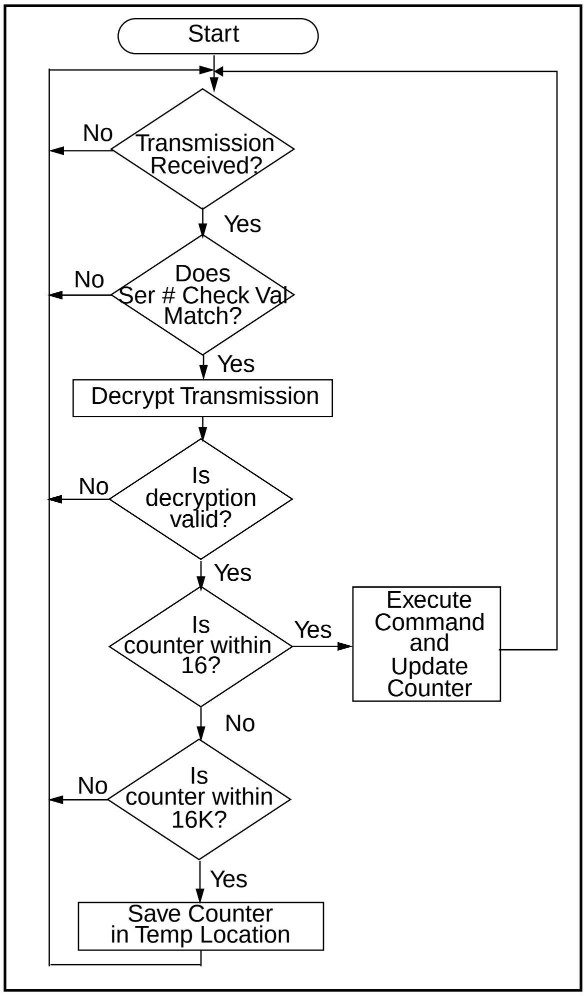
The decoder waits for a transmission and checks the serial number to determine if it is a learned transmitter. If it is, it takes the code hopping portion of the transmission and decrypts it, using the encoder key. It uses the discrimination value to determine if the decryption was valid. If everything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

3.3 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.

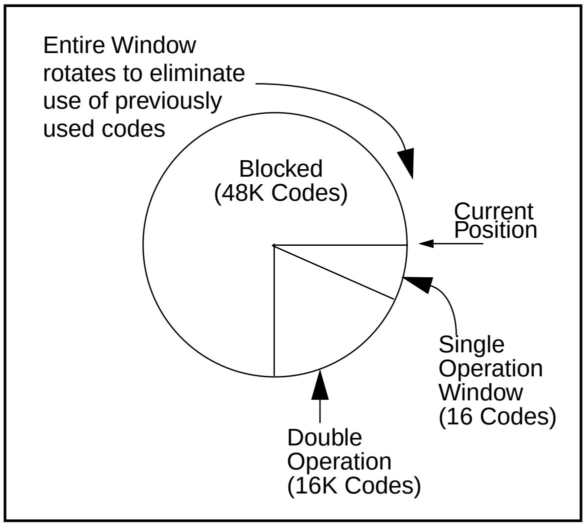
FIGURE 3-2: DECODER OPERATION



3.4 Synchronization with Decoder

The KEELOQ technology features a sophisticated synchronization technique (Figure 3-3) which does not require the calculation and storage of future codes. If the stored synchronization counter value for that particular transmitter and the synchronization counter value that was just decrypted are within a formatted window of 16, the counter is stored, and the command is executed. If the synchronization counter value was not within the single operation window, but is within the double operation window of the 16K window, the transmitted synchronization counter value is stored in a temporary location, and the decoder goes back to waiting for another transmission. When the next valid transmission is received, it will check the new synchronization counter value with the one in temporary storage. If the two values are sequential, it is assumed that the counter had just gotten out of the single operation 'window', but is now back in synchronization, so the new synchronization counter value is stored, and the command is executed. If a transmitter has somehow gotten out of the double operation window, the transmitter will not work and must be relearned. Since the entire window rotates after each valid transmission, codes that have been used become part of the 'blocked' (48K) codes and are no longer valid. This eliminates the possibility of grabbing a previous code and retransmitting to gain entry.

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’.
- 2 status bits [REPEAT, VLOW].
- 4-bit function code [S3 S2 S1 S0].
- Stop bit ‘1’.
- 4 bits indicating which block was used [TX3…TX0].
- 4 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

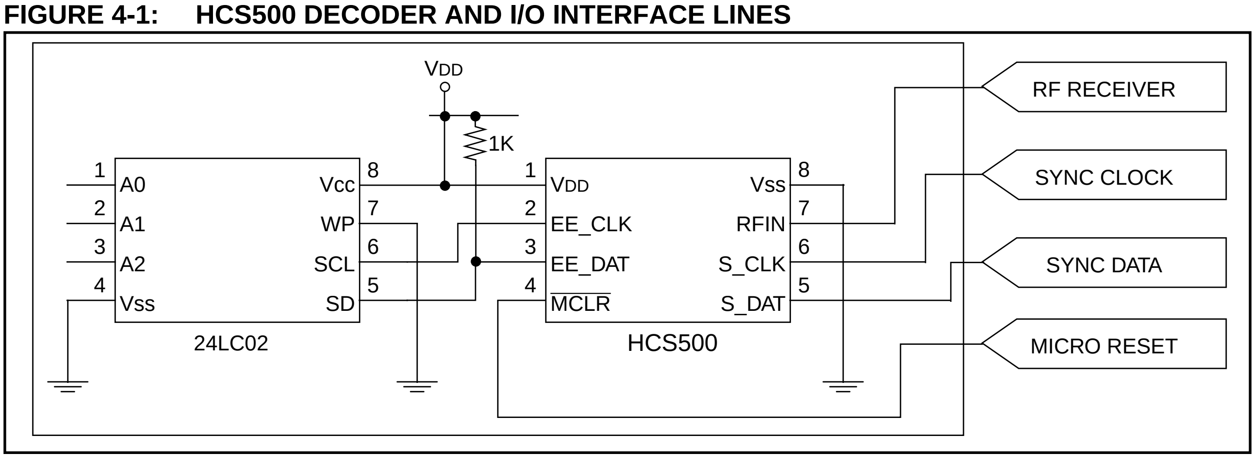
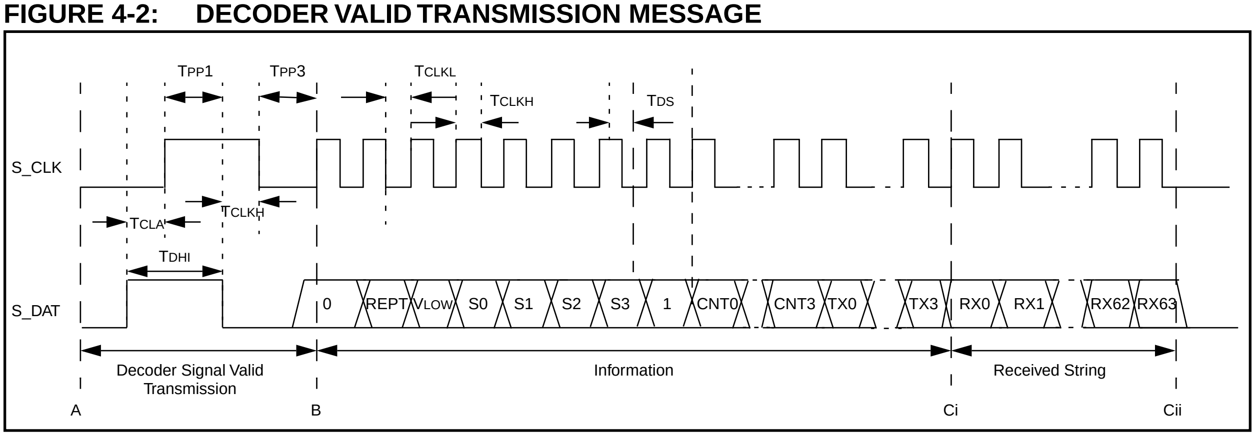


FIGURE 4-2: DECODER VALID TRANSMISSION MESSAGE



4.2 Command Mode

4.2.1 MICROCONTROLLER COMMAND MODE ACTIVATION

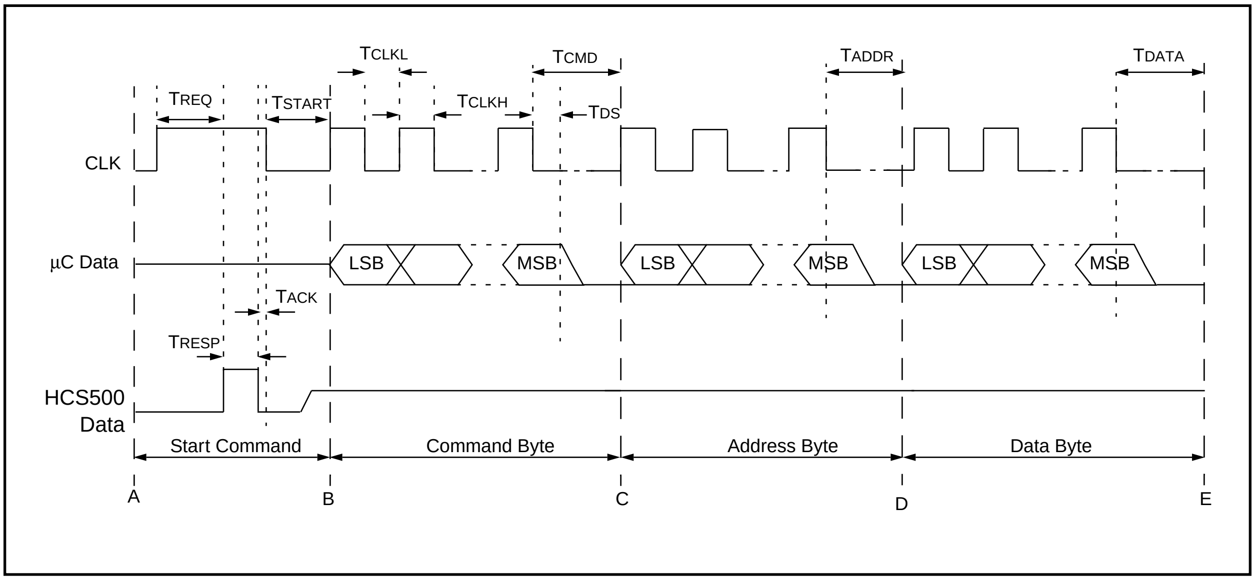
The microcontroller command consists of four parts. The first part activates the command mode, the second part is the actual command, the third is the address accessed, and the last part is the data. The microcontroller starts the command by taking the clock line high for up to 500 ms. The decoder acknowledges 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 doesn't 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 1/2 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

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

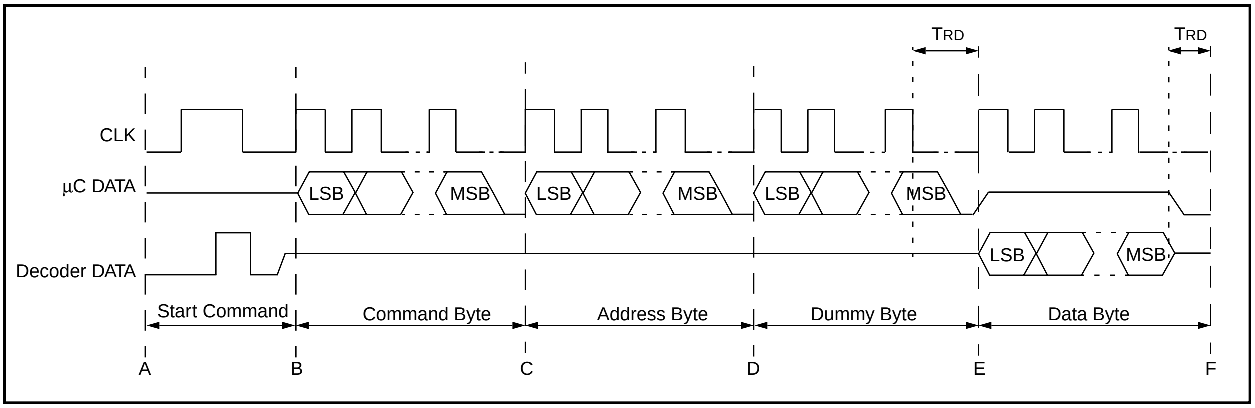
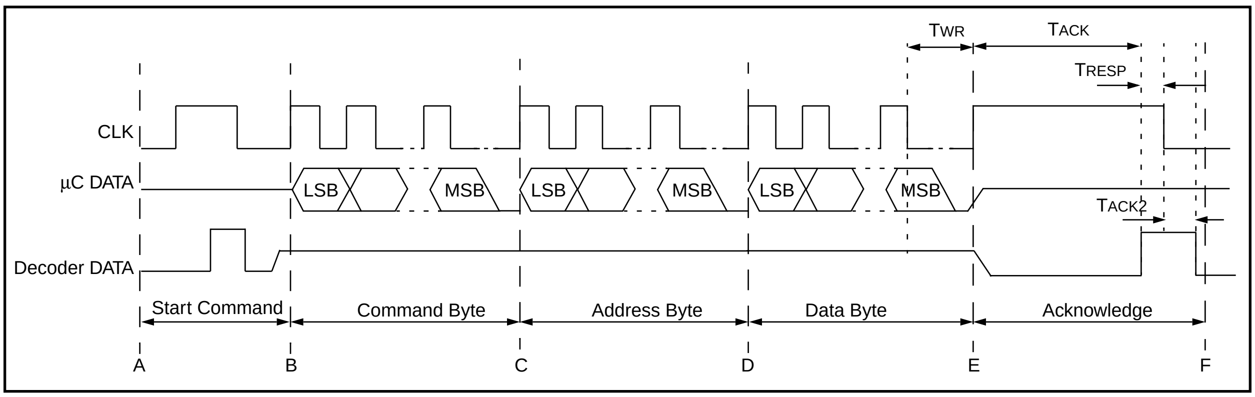


FIGURE 4-5: WRITE BYTES TO USER EEPROM



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

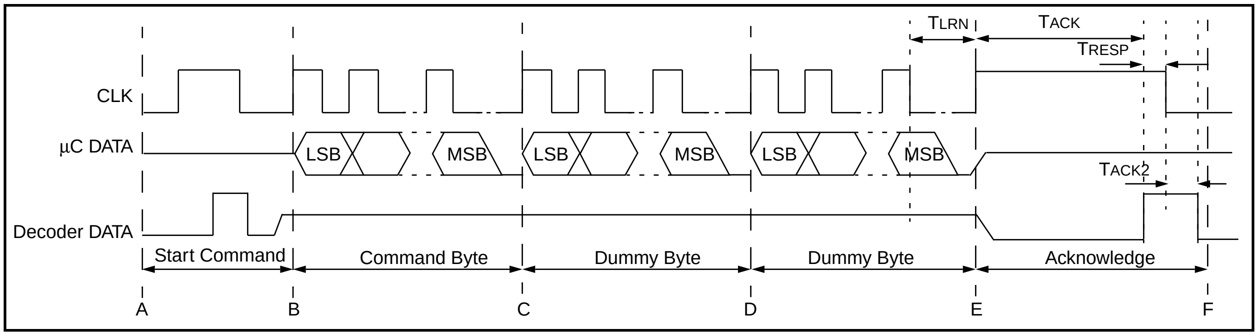


FIGURE 4-7: LEARN STATUS MESSAGE AFTER FIRST TRANSMISSION

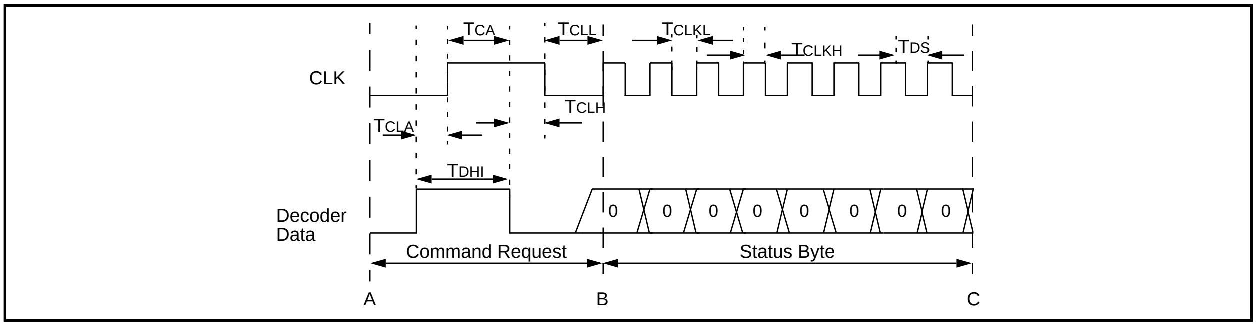
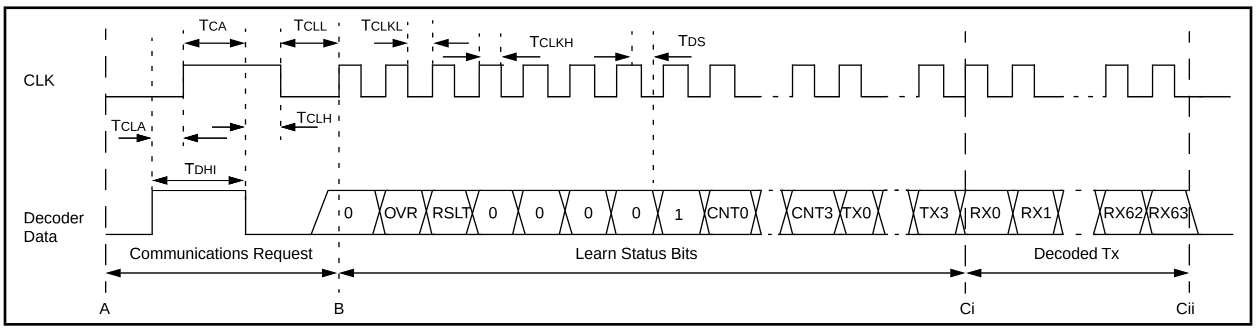


FIGURE 4-8: LEARN STATUS MESSAGE AFTER SECOND TRANSMISSION



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 2 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 6 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. Encoder key = manufacturer's code.
2. Serial number = any value.
3. Discrimination bits = lower 10 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 is imperative (Figure 4-10 and Figure 8-2).

FIGURE 4-9: ERASE ALL

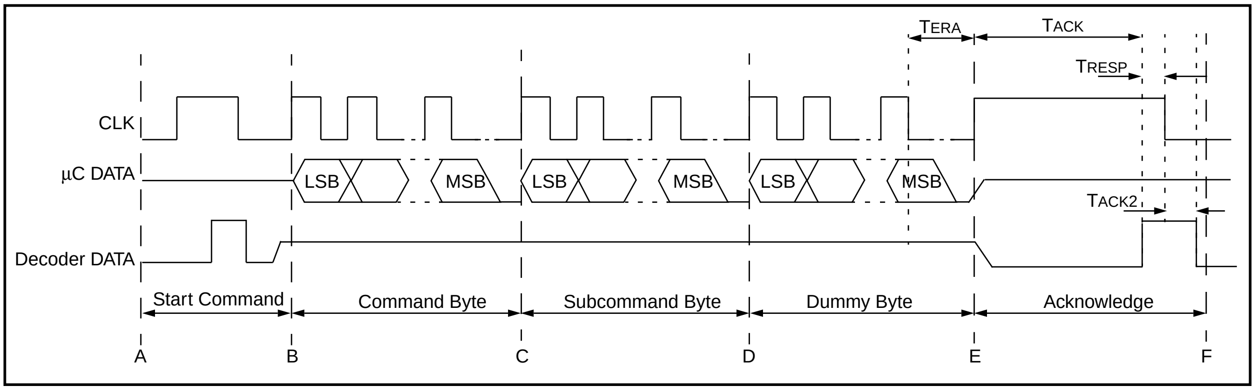


FIGURE 4-10: STAND-ALONE MODE LEARN/ERASE-ALL TIMING

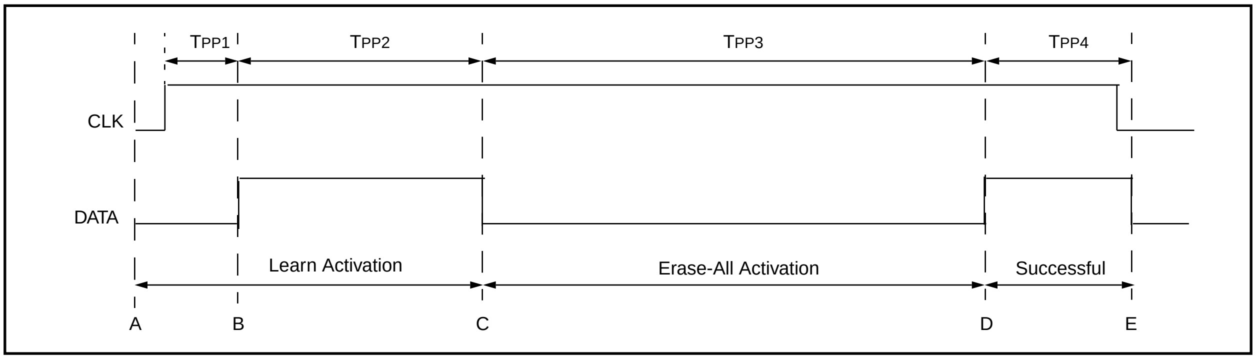
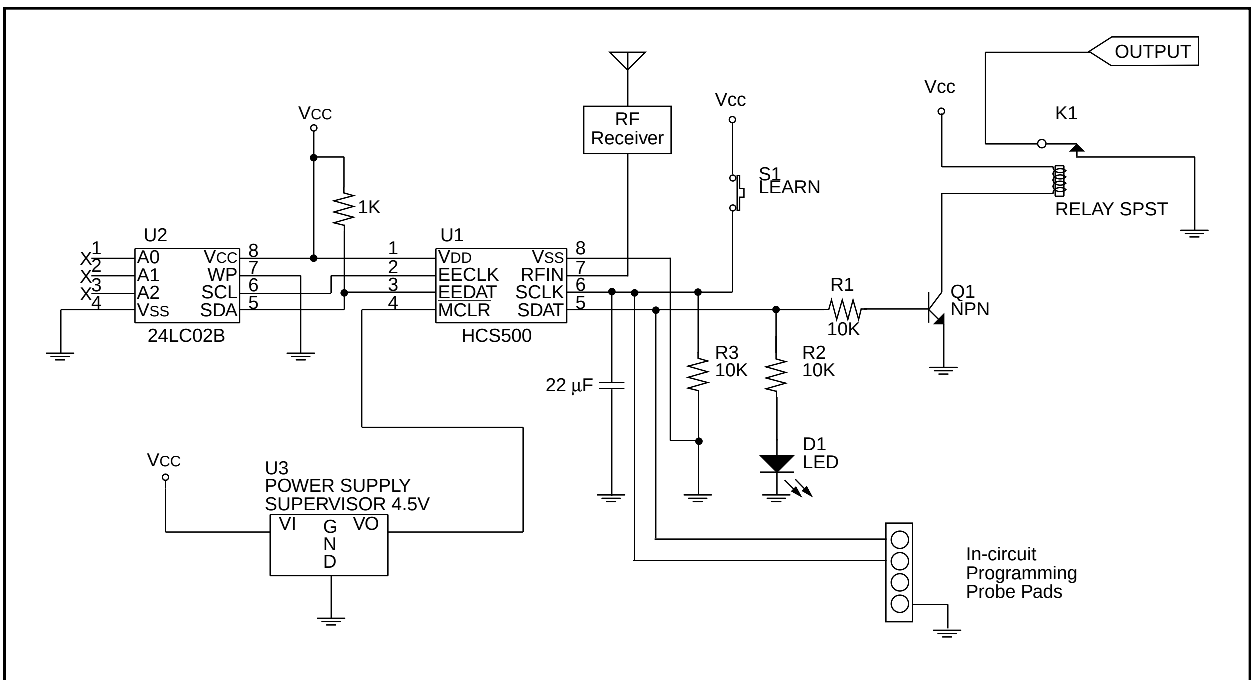


FIGURE 4-11: TYPICAL STAND-ALONE APPLICATION CIRCUIT



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:

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 for more detail on learning modes.

5.1.2 LRN_ALG

LRN_ALG selects between the two available algorithms. With LRN_ALG = 0, is selected the KEELOQ decryption algorithm is selected; with LRN_ALG = 1, the XOR algorithm is selected. See Section 6.0 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

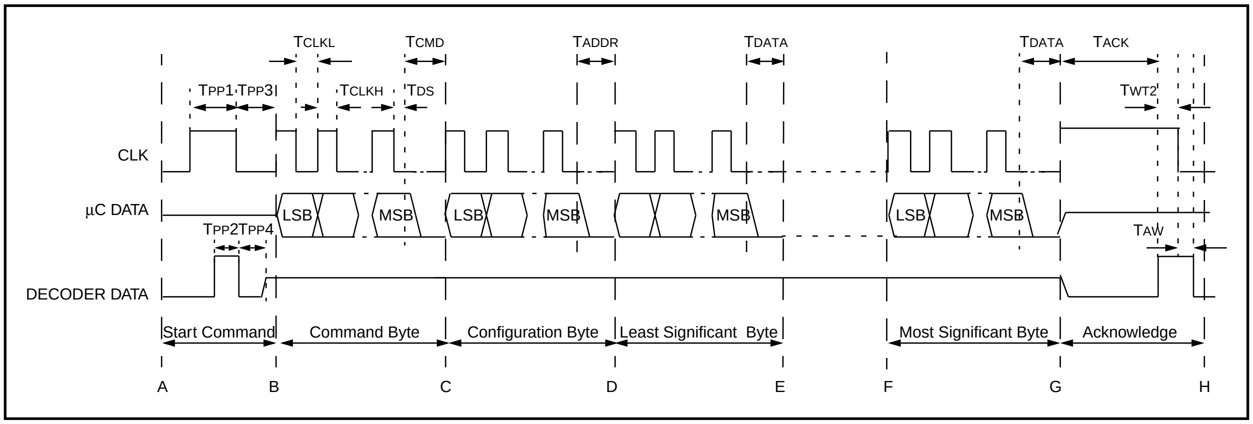
The programming command consists of the following:

- 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 encoder 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 encoder 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 encoder 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 4 bits are always forced to zero.

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

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

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

SourceH = Seed $_{\text{Upper 16 bits}}$ + Serial Number $_{\text{Upper 16 bits}}$ with upper 4 bits set to zero
SourceL = Seed $_{\text{Lower 32 bits}}$

For 60-bit seed encoders (HCS410):

SourceH = Seed $_{\text{Upper 32 bits}}$ with upper 4 bits set to zero
SourceL = Seed $_{\text{Lower 32 bits}}$

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

KeyH $_{\text{Upper 32 bits}}$ = F $_{\text{KEELOQ Decrypt}}$ (SourceH) $|_{\text{64 Bit Manufacturer's Code}}$
KeyL $_{\text{Lower 32 bits}}$ = F $_{\text{KEELOQ Decrypt}}$ (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 4 bits are always forced to zero.

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

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

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

SourceH = Seed $_{\text{Upper 16 bits}}$ + Serial Number $_{\text{Upper 16 bits}}$ with upper 4 bits set to zero
SourceL = Seed $_{\text{Lower 32 bits}}$

For 60-bit seed encoders (HCS410):

SourceH = Seed $_{\text{Upper 32 bits}}$ with upper 4 bits set to zero
SourceL = Seed $_{\text{Lower 32 bits}}$

Then, using the KEELOQ decryption algorithm and the manufacturer's code the encoder 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 66/67 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 66/67-bit code word when a button is pressed. The 66/67-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 2 status bits, 4 function bits, and the 28/32-bit serial number.

FIGURE 7-1: CODE WORD TRANSMISSION FORMAT

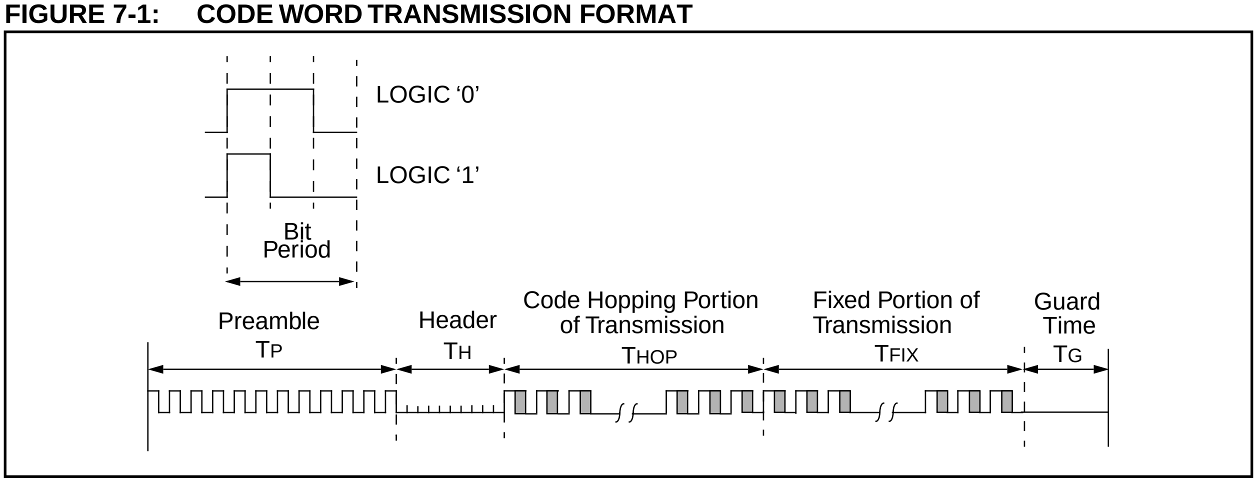
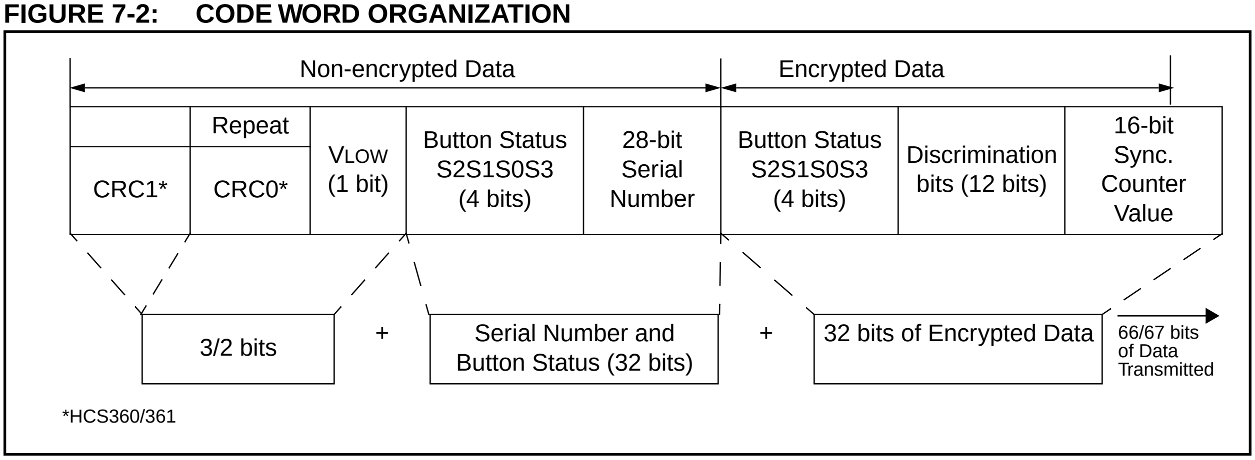


FIGURE 7-2: CODE WORD ORGANIZATION



8.0 ELECTRICAL CHARACTERISTICS FOR HCS500

Absolute Maximum Ratings†

Ambient temperature under bias .... -40°C to +85°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 +7.0V

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.

TABLE 8-1: DC CHARACTERISTICS

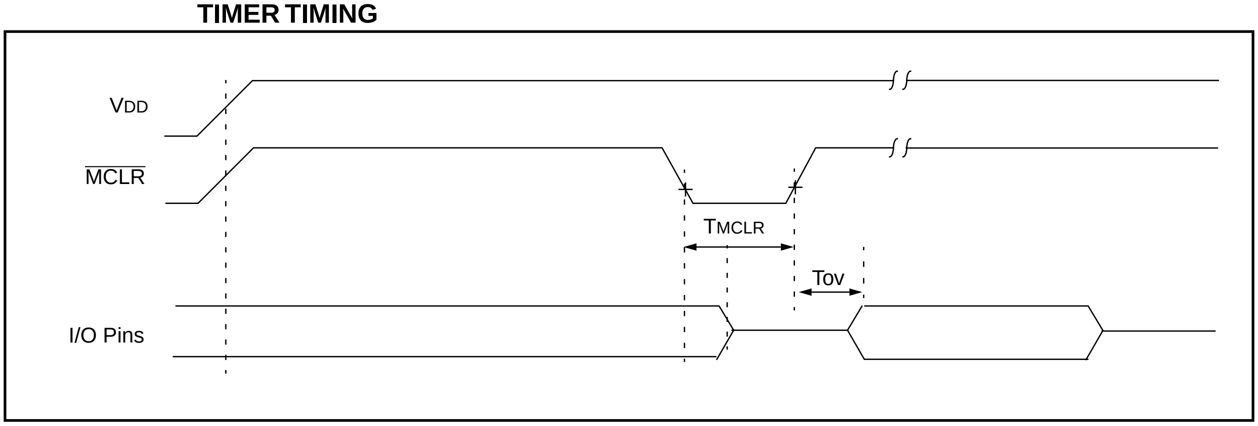
		Standard Operating Conditions (unless otherwise stated) Operating temperature Commercial (C): $0°C \le T_A \le +70‡°C$ Industrial (I): $-40°C \le T_A \le +85‡°C$				
Symbol	**Parameters**	**Min**	**Typ(†)**	**Max**	**Units**	**Conditions**
V_{DD}	Supply voltage	3.0	—	5.5	V	
V_{POR}	V_{DD} start voltage to ensure Reset	—	Vss	—	V	
S_{VDD}	V_{DD} rise rate to ensure reset	0.05*	—	—	V/ms	
I_{DD}	Supply current	— —	1.8 0.3	2.4 5	mA µA	F_{OSC} = 4 MHz, V_{DD} = 5.5V Sleep mode (no RF input)
I_{PD}	Power Down Current	— —	0.25 0.3	4 5	µA µA	V_{DD} = 3.0V, Commercial V_{DD} = 3.0V, Industrial
V_{IL}	Input low voltage	V_{SS} V_{SS}	— —	0.15 V_{DD} 0.8	V V	Except $\overline{MCLR}$ = 0.15 V_{DD} V_{DD} between 4.5V and 5.5V
V_{IH}	Input high voltage	0.25 V_{DD} 2.0	— —	V_{DD} V_{DD}	V V	Except $\overline{MCLR}$ = 0.85 V_{DD} V_{DD} between 4.5V and 5.5V
V_{OL}	Output low voltage	—	—	0.6	V	I_{OL} = 8.7 mA, V_{DD} = 4.5V
V_{OH}	Output high voltage	V_{DD} - 0.7	—	—	V	I_{OH} = -5.4 mA, V_{DD} = 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 8-2: AC CHARACTERISTICS

		Standard Operating Conditions (unless otherwise specified): Commercial (C): $0°C \le TA \le +70°C$ Industrial (I): $-40°C \le TA \le +85°C$				
Symbol	**Parameters**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
T_E	Transmit elemental period	65	—	660	µs	
T_{OD}	Output delay	48	75	237	ms	
T_{MCLR}	$\overline{MCLR}$ low time	150	—	—	ns	
T_{OV}	Time output valid	—	150	222	ms	

FIGURE 8-1: RESET WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



8.1 AC Electrical Characteristics

8.1.1 COMMAND MODE ACTIVATION

		Standard Operating Conditions (unless otherwise specified): Commercial (C): 0°C ≤ TA ≤ +70°C Industrial (I): -40°C ≤ TA ≤ +85°C			
Symbol	**Parameters**	**Min**	**Typ**	**Max**	**Units**
TREQ	Command request time	0.0050	—	500	ms
TRESP	Microcontroller request acknowledge time	—	—	1	ms
TACK	Decoder acknowledge time	—	—	4	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

8.1.2 READ FROM USER EEPROM COMMAND

		Standard Operating Conditions (unless otherwise specified): Commercial (C): 0°C ≤ TA ≤ +70°C Industrial (I): -40°C ≤ TA ≤ +85°C			
Symbol	**Parameters**	**Min**	**Typ**	**Max**	**Units**
TRD	Decoder EEPROM read time	400	—	1500	µs

8.1.3 WRITE TO USER EEPROM COMMAND

		Standard Operating Conditions (unless otherwise specified): Commercial (C): 0°C ≤ TA ≤ +70°C Industrial (I): -40°C ≤ TA ≤ +85°C			
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

8.1.4 ACTIVATE LEARN COMMAND IN MICRO MODE

		Standard Operating Conditions (unless otherwise specified): Commercial (C): 0°C ≤ TA ≤ +70 °C Industrial (I): -40°C ≤ TA ≤ +85°C			
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

8.1.5 ACTIVATE LEARN COMMAND IN STAND-ALONE MODE

		Standard Operating Conditions (unless otherwise specified): Commercial (C): 0°C ≤ TA ≤ +70°C Industrial (I): -40°C ≤ TA ≤ +85°C			
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

8.1.6 LEARN STATUS STRING

		Standard Operating Conditions (unless otherwise specified): Commercial (C): 0°C ≤ TA ≤ +70°C Industrial (I): -40°C ≤ TA ≤ +85°C			
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

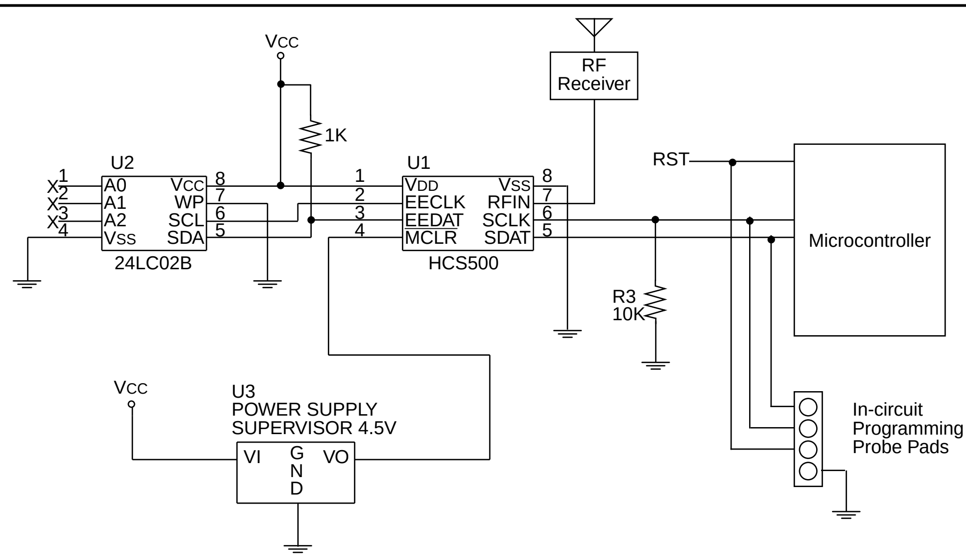
8.1.7 ERASE ALL COMMAND

		Standard Operating Conditions (unless otherwise specified): Commercial (C): 0°C ≤ TA ≤ +70°C Industrial (I): -40°C ≤ TA ≤ +85°C			
Symbol	**Parameters**	**Min**	**Typ**	**Max**	**Units**
T_{ERA}	Learn command activation time	20	—	1000	µs
T_{ACK}	Decoder acknowledge time	20	—	210	ms
T_{RESP}	Microcontroller acknowledge response time	20	—	1000	µs
T_{ACK2}	Decoder data line low	—	—	10	µs

8.1.8 PROGRAMMING COMMAND

		Standard Operating Conditions (unless otherwise specified): Commercial (C): 0°C ≤ TA ≤ +70°C Industrial (I): -40°C ≤ TA ≤ +85°C			
Symbol	**Parameters**	**Min**	**Typ**	**Max**	**Units**
T_{PP1}	Command request time	—	—	500	ms
T_{PP2}	Decoder acknowledge time	—	—	1	ms
T_{PP3}	Start command mode to first command bit	20	—	1000	µs
T_{PP4}	Data line low before tri-stated	—	—	5	µs
T_{CLKH}	Clock high time	20	—	1000	µs
T_{CLKL}	Clock low time	20	—	1000	µs
F_{CLK}	Clock frequency	500	—	25000	Hz
T_{DS}	Data hold time	—	—	5	µs
T_{CMD}	Command validate time	—	—	10	µs
T_{ACK}	Command acknowledge time	30	—	240	ms
T_{WT2}	Acknowledge respond time	20	—	1000	µs
T_{ALW}	Data low after clock low	—	—	10	µs

FIGURE 8-2: TYPICAL MICROCONTROLLER INTERFACE CIRCUIT



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

PRODUCT IDENTIFICATION SYSTEM

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

HCS500 — __ /P

Package:	P	=	Plastic DIP (300 mil Body), 8-lead
	SM	=	Plastic SOIC (150 mil Body), 8-lead
Temperature Range:	Blank	=	0°C to +70°C
	I	=	–40°C to +85°C
Device:	HCS500		Code Hopping Decoder
	HCS500T		Code Hopping Decoder (Tape and Reel)

Sales and Support

Data Sheets
Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office.
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277.
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office

Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602-786-7200 Fax: 602-786-7277
Technical Support: 602 786-7627
Web: http://www.microchip.com

Atlanta

Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston

Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508-480-9990 Fax: 508-480-8575

Chicago

Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas

Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 972-991-7177 Fax: 972-991-8588

Dayton

Microchip Technology Inc.
Two Prestige Place, Suite 150
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

Los Angeles

Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 714-263-1888 Fax: 714-263-1338

New York

Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516-273-5305 Fax: 516-273-5335

San Jose

Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto

Microchip Technology Inc.
5925 Airport Road, Suite 200
Mississauga, Ontario L4V 1W1, Canada
Tel: 905-405-6279 Fax: 905-405-6253

ASIA/PACIFIC

Hong Kong

Microchip Asia Pacific
RM 3801B, Tower Two
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2-401-1200 Fax: 852-2-401-3431

India

Microchip Technology India
No. 6, Legacy, Convent Road
Bangalore 560 025, India
Tel: 91-80-229-0061 Fax: 91-80-229-0062

Korea

Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Shanghai

Microchip Technology
RM 406 Shanghai Golden Bridge Bldg.
2077 Yan'an Road West, Hongiao District
Shanghai, PRC 200335
Tel: 86-21-6275-5700
Fax: 86 21-6275-5060

Singapore

Microchip Technology Taiwan
Singapore Branch
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan, R.O.C

Microchip Technology Taiwan
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2-717-7175 Fax: 886-2-545-0139

EUROPE

United Kingdom

Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44-1628-851077 Fax: 44-1628-850259

France

Arizona Microchip Technology SARL
Zone Industrielle de la Bonde
2 Rue du Buisson aux Fraises
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany

Arizona Microchip Technology GmbH
Gustav-Heinemann-Ring 125
D-81739 Müchen, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy

Arizona Microchip Technology SRL
Centro Direzionale Colleone
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-39-6899939 Fax: 39-39-6899883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81-4-5471- 6166 Fax: 81-4-5471-6122

5/8/97

MICROCHIP

Printed on recycled paper.