AN642

Code Hopping Decoder using a PIC16C56

Author: *Steven Dawson*
Microchip Technology Inc.

OVERVIEW

This application note fully describes the working of a code hopping decoder implemented on a Microchip PIC16C56 microcontroller. Background is given on the various KEELOQ® code hopping encoders that can be used with the decoder, the decoder hardware described, and descriptions of the various software modules comprising the system. The software can be used to implement a stand alone decoder or integrated with full function security systems. The decoder supports the Microchip HCS200, HCS201, HCS300, HCS301, HCS360, HCS361 and HCS410 KEELOQ code hopping encoders.

KEY FEATURES

- Stand alone decoder
- Compatible with Microchip HCS200, HCS201, HCS300, HCS301, HCS360, HCS361 and HCS410 encoders
- Automatic baud rate detection
- Automatic encoder type detection
- Four function outputs
- Six learnable transmitters
- RC Oscillator

NOTICE:

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROPRIETARY AND CONFIDENTIAL INFORMATION OF MICROCHIP TECHNOLOGY INC. THEREFORE, ALL PARTIES ARE REQUIRED TO SIGN A NON-DISCLOSURE AGREEMENT BEFORE RECEIVING THIS DOCUMENT.

KEELOQ is a registered trademark of Microchip Technology, Inc.
Microchip's Secure Data Products are covered by some or all of the following patents:
Code hopping encoder patents issued in Europe, U.S.A., and R.S.A. — U.S.A.: 5,517,187; Europe: 0459781; R.S.A.: ZA93/4726
Secure learning patents issued in the U.S.A. and R.S.A. — U.S.A.: 5,686,904; R.S.A.: 95/5429

INTRODUCTION TO KEELOQ ENCODERS

All KEELOQ encoders use the KEELOQ code hopping technology to make each transmission by an encoder unique. The encoder transmissions have two parts. The first part changes each time the encoder is activated and is called the hopping code part. The second part is the serial number of the encoder, identifying it to a decoder.

Hopping Code

The Hopping Code contains function information, a discrimination value, and a synchronization counter. This information is encrypted by an encryption algorithm before being transmitted. A 64-bit encryption key is used by the encryption algorithm. If one bit in the data that is encrypted changes, the result is that an average of half the bits in the output will change. As a result, the hopping code changes dramatically for each transmission and can not be predicted.

Function Information

The encoder transmits up to four bits of function information. Up to 15 different functions are available (0000 is related to the reset condition in all the current encoders, and can never be transmitted).

Discrimination Value

Stored in the encoder EEPROM, this information is used to check integrity of decryption operation in the decoder. If known information is inserted into the transmitted string before encryption, the same information can be used at the decoder to check whether the information has been decrypted correctly. In the Microchip HCS encoders, up to 12 bits (including overflow bits) are available.

Synchronization Counters

The transmitted word contains a 16-bit synchronization counter. The synchronization information is used at the decoder to determine whether a transmission is valid, or a repetition of a previous transmission. Previous codes are rejected to safeguard against code grabbers. The HCS300/301 encoder transmits two overflow bits which may be used to extend the range of the synchronization counter from 65,536 to 196,608 button operations.

Fixed Code

Serial Number

The encoder's serial number is transmitted every time the button is pressed. The serial number is transmitted unencrypted as part of the transmission, and serves to identify the encoder to the decoder. The number can be used during learning operations to calculate the key to be used for decrypting the transmissions.

Other Status and Function Information

The HCS300/301 encoders include provision for four bits of function information and two status bits in the fixed code portion of its transmission. The two status bits indicate whether a repeated transmission is being sent, and whether the battery voltage is low. The HCS200/201 does not send repeated transmission information and the bit is permanently set to '0'.

Transmission Format

Table 1 contains a summary of the information contained in transmissions from each of the KEELOQ encoders that can be learned by the Microchip decoder.

FIGURE 1: BLOCK DIAGRAM

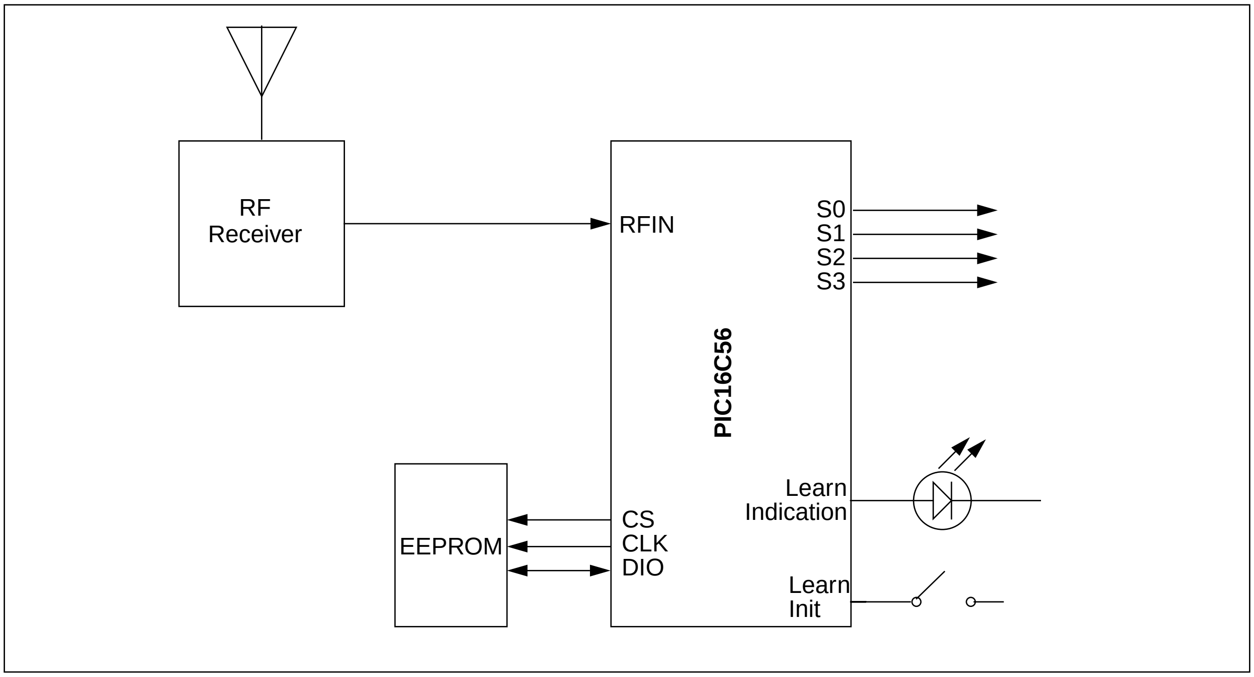


TABLE 1: KEELOQ ENCODER TRANSMISSION SUMMARY

	HCS200/201 # of bits	HCS300/301 # of bits	HCS360/361 # of bits	HCS410 # of bits
Total Transmission Length	66	66	67	69
Code Hopping Portion	32	32	32	32
Sync Counter	16	16	16	16
Discrimination bits	12	10	8	10
User Bits	0	0	2	0
Overflow Bits	0	2	1	2
Independent Mode	0	0	1	0
Function Code	4	4	4	4
Fixed Portion	34	34	35	37
Serial number	28	28	28/32	28/32
Function Code	4	4	4/0	4/0
Low Voltage Indicator	1	1	1	1
Repeat Bit	1	1	0	0
CRC	0	0	2	2
Queue Bits	0	0	0	2

TABLE 2: HCS200/201 AND HCS300/301 CODE HOPPING TRANSMISSION FORMAT

Code Hopping Portion			Fixed Portion		
Sync Counter	Discrimination	Func	Serial Number	Func	VLOW REPT

TABLE 3: HCS200/201 AND HCS300/301 SEED TRANSMISSION FORMAT

Seed Portion	Fixed Portion		
Seed	Serial Number	Func	VLOW REPT

TABLE 4: HCS360/361 CODE HOPPING TRANSMISSION FORMAT

Code Hopping Portion			Fixed Portion		
Sync Counter	Discrimination OVR, IND	Func	Serial Number (28/32 bits)	Func (4/0 bits)	VLOW REPT

TABLE 5: HCS360/361 SEED TRANSMISSION FORMAT

Seed Portion	Fixed Portion		
Seed (48 bits)	Serial Number (12/16 MS bits)	Func (4/0 bits)	VLOW REPT

TABLE 6: HCS410 CODE HOPPING TRANSMISSION FORMAT

Code Hopping Portion			Fixed Portion		
Sync Counter	Discrimination OVR	Func	Serial Number (28/32 bits)	Func (4/0 bits)	VLOW CRC QUE

TABLE 7: HCS410 SEED TRANSMISSION FORMAT

Seed Portion	Fixed Portion	
Seed (60 bits)	Func (4/0 bits)	VLOW CRC QUE

PWM Format

In general, all KEELOQ encoders share a common transmission format.

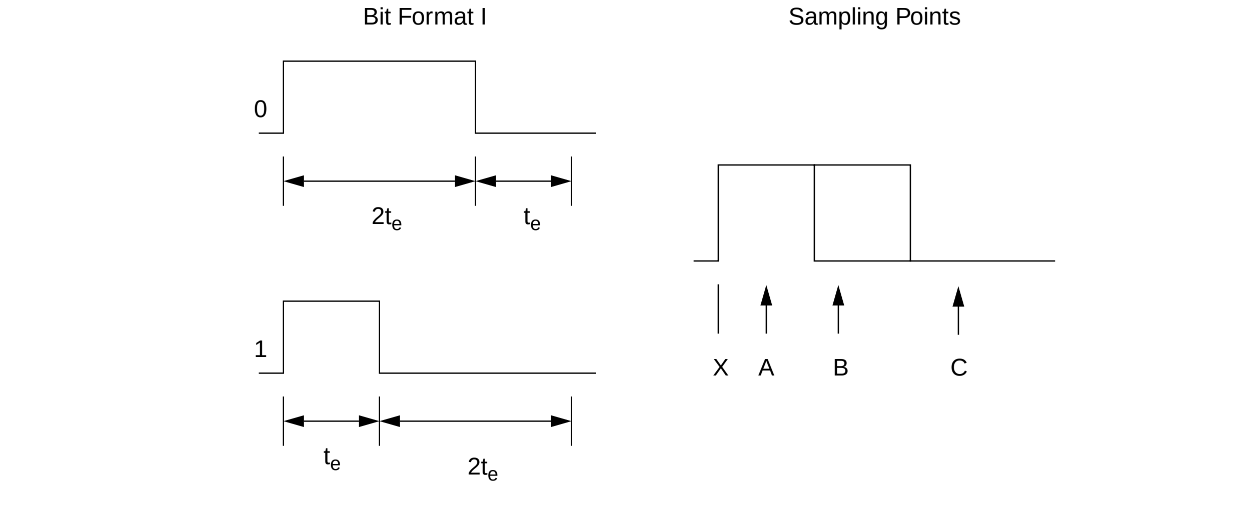
- A preamble to improve biasing of decision thresholds in superregenerative receivers. The preamble consists of alternate on and off periods, each lasting as long as a single elemental period.
- A calibration header, consisting of a low period of 10 elemental periods. Calibration actions should be performed on the low period of the header to ensure correct operation with header chopping.
- A string of pulse width modulated bits, each consisting of three elements. The first element is high, the second contains the data transmitted and is either high or low, the third element is always low.
- A guard period is usually left between the transmissions. During this period nothing is transmitted by the encoder.

Figure 2 shows the sampling points when sampling data. The first and last elements are used exclusively to verify the integrity of the received signal. The first element (sample point A) is always high, the second (sample point B) is the complement of the data bit being sent, and the final element (sample point C) is always low. Because the period between the low portion of a bit (sample point C) and the rising edge of the following bit (sample point X) can vary somewhat, the rising edge of the first element (sample point X) is used to resynchronize the receiving routine to each incoming bit.

If random noise is being received, the probability of a set of three samples producing a valid combination is only 2^{-2} = 1/4. For a string of 66 bits, the corresponding figure is 2^{-134}. For longer strings, the probability is considerably less.

Integrity checking on incoming signals is important. Code hopping signals require significant processing, as well as EEPROM access, to decrypt. Unnecessary processing can be avoided by not attempting to decrypt incoming codes that have bit errors.

FIGURE 2: KEELOQ PWM TRANSMISSION FORMAT



IMPLEMENTATION

The Microchip decoder's primary hardware components are a PIC16C56 RISC microcontroller and a 93LC46B EEPROM. However, this solution can be implemented in any PIC16/17 microcontroller with at least 1K words of programming. The operating frequency of the controller is 4 MHz. The microcontroller is used to capture transmissions from the various encoders, decrypt transmissions captured, and check the validity of the transmission based on the information in the decrypted transmission and information stored in the EEPROM. If a transmission from a valid encoder is received, the Microchip decoder activates the outputs dictated by the transmission.

Encoder information, such as serial number, synchronization information, and decryption key are stored externally in an EEPROM. The EEPROM used is a Microchip 93LC46B CMOS serial EEPROM. The information stored in the EEPROM is encrypted to protect the decoders from cloning. The EEPROM encryption is less secure than the KEELOQ code hopping algorithm.

A more secure implementation of the decoder would be to modify the software in the application note and use a PIC16CXX with an internal EEPROM such as a PIC16C84. In this way communication between the PIC16C56 and EEPROM cannot be monitored.

As can be seen from the section on encoder transmissions there are differences in the transmission formats of the different encoders that are compatible with the system. The following section summarizes how the differences in transmitted data are dealt with by the decoder.

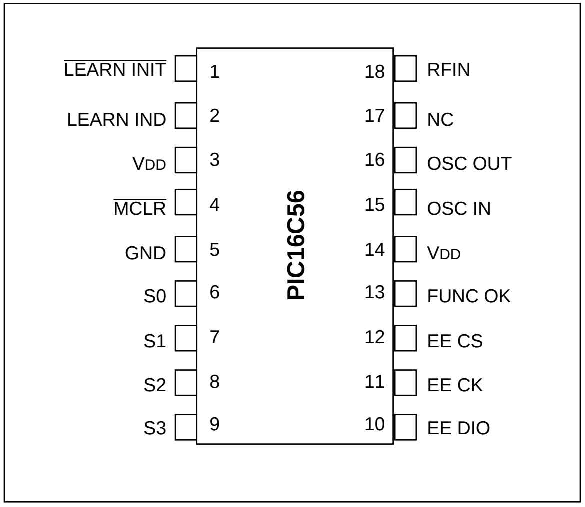
As the **serial number** information follows after the code hopping portion of the transmission, any number of serial number bits can be received and processed. In the Microchip decoder being described, 28 bits of the serial number are stored.

The serial number is used to identify the memory block used to store the 64-bit decryption key for a particular encoder because of the relationship between serial number and decryption key. In other words, the serial number is stored with the key. When a transmission is received, the decoder finds the correct memory block by checking all blocks until a matching serial number is found. The key is then retrieved from that particular memory block. A serial number of 0000000_{16} is considered invalid.

Validation of a received transmission consists of two parts. The first includes checking the integrity of the decryption operation. Here the decoder compares the 12-bit discrimination value received with the stored discrimination value. The discrimination value stored for the HCS300/301 includes the overflow bits.

The second portion of validation involves checking **synchronization** information for that particular encoder. The synchronization counter transmitted by all encoders is 16 bits long. Two copies of the full synchronization counter are stored for all valid encoders. The storing of two copies of the synchronization information protects the decoder from loosing synchronization with an encoder if one of the counters is corrupted.

FIGURE 3: PINOUTS OF MICROCHIP KEELOQ DECODER



FUNCTIONAL INPUTS AND OUTPUTS

TABLE 8: MICROCHIP DECODER FUNCTIONAL INPUTS AND OUTPUTS

Mnemonic	Pin Number	Input / Output	Function
RF IN	18	I	Demodulated PWM signal from RF receiver. The decoder uses this input to receive encoder transmissions.
LEARN INIT	1	I	Input to initiate learning.
LEARN INDICATION	2	O	Output to show the status of the learn process (in an integrated system this will be combined with the system status indicator).
FUNC OK	13	O	Indication that the received button code matches the learned button code.
S0, S1, S2, S3	6, 7, 8, 9	O	Function outputs, correspond to encoder input pins.
EE DIO	10	I/O	EEPROM Data.
EE CK	11	O	EEPROM Clock.
EE CS	12	1	EEPROM Chip Select.

PROGRAM FLOW

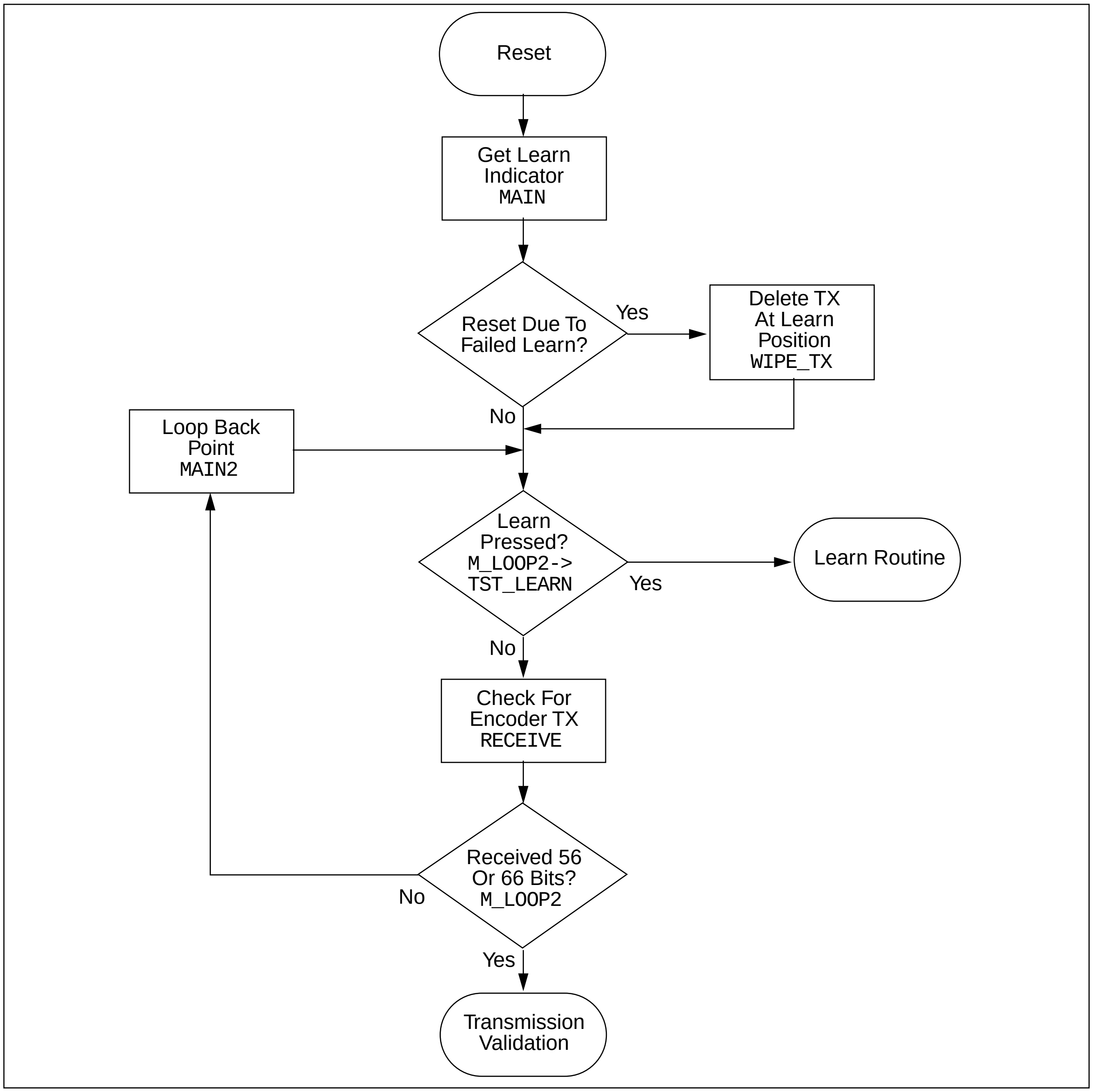
The software for the Microchip decoder has been written for the PIC16C56 microcontroller. The compiler used is MPASM. The operating frequency of the PIC16C56 is 4 MHz. The clock speed is important as the reception routine (RECEIVE) has some critical timing specifications. Other decoder functions that rely on a 4 MHz clock speed are the hold times of the various outputs, time-outs, etc.

The main program flow is described here. More detailed descriptions of the modules can be found further in the application note. On power-up the decoder reads the learn indicator from the external EEPROM. The status flags are checked to see if a learn routine was interrupted when the microcontroller was reset. If so, it is assumed the learn cycle was not successfully completed and the encoder at the learn indicator subsequently deleted (WIPE_TX).

The encoder then enters the main loop where it spends most of its time. The main loop checks to see if the learn button is being activated (TST_LEARN). If so, the decoder enters the learn mode.

If learn has not been initiated, the microcontroller then checks for transmissions from encoders (RECEIVE). If 64 bits (HCS encoders) are received, the microcontroller validates the transmission received. If the transmission received is a valid transmission from an encoder learned into the system, the system sets the appropriate outputs (M_BUT).

FIGURE 4: MICROCHIP DECODER MAIN PROGRAM FLOW


Reset
Get Learn Indicator MAIN
Reset Due To Failed Learn?
Yes
Delete TX At Learn Position WIPE_TX
No
Loop Back Point MAIN2
Learn Pressed? M_LOOP2-> TST_LEARN
Yes
Learn Routine
No
Check For Encoder TX RECEIVE
Received 56 Or 66 Bits? M_LOOP2
No
Yes
Transmission Validation

FUNCTIONAL MODULES

Reception

The reception routine (called RECEIVE) is based on a reliable algorithm which has successfully been used in previous implementations of KEELOQ decoders. Automatic baud rate detection is used to compensate for variations in baud rate of different encoders of a specific type, as well as the difference in baud rate between different encoders (HCS200, and HCS300). The reception routine is able to handle 64-bit transmissions. This is easily extented to receive more bits. The reception routine is able to determine the type of encoder by the number of bits in the transmission.

The reception algorithm performs the following functions when an output is detected from the receiver:

1. Calibrate on the header low period to determine the actual elemental period for the transmission being received. The required elemental period is 10% of the low header period. In the diagram below (Figure 5) the header calibration sample points are marked 1 through 3. The calibration flow chart (Figure 6) shows at what points in the source code samples 1, 2, and 3 are taken. Elemental periods outside the capture range of the algorithm (either too long or too short) should be rejected, since they would be due either to noise or to reception of an incomplete signal.
2. Using the determined elemental period, three samples after the first rising edge following the header are taken. The first sample is taken half an elemental period after the rising edge (sample 4); the second, one elemental period later (sample 5), and the third, another one elemental period later (sample 6). The first sample must be high, the second could be either high or low, and the third sample must be low. If either the first or the third sample is not as expected, the attempt at capturing a transmission is abandoned. In the diagram below (Figure 5), the data sample points are points 4 through 6. The flow chart describing data reception (Figure 7) shows where in the code the samples are taken.
3. If all 64 bits have been captured, each with the correct first and third elements, the transmission can be assumed to be correct, and decryption can commence.

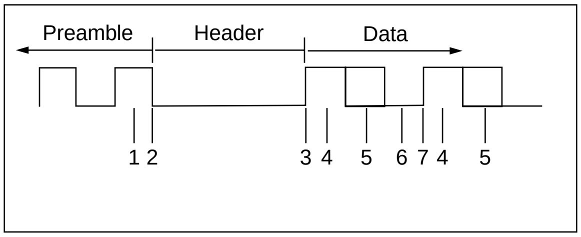
The receiving routine should be called often enough to ensure that the high portion in the header is not missed (Sample 1, Figure 5).

In systems where the receive routine is called to check if there is activity on the receiver input, the routine should poll the input for a valid transmission for at least the time taken to complete one transmission if activity is detected on the input line. This makes provision for the receive routine being called while a transmission is in progress. Having missed the first header, the first transmission will be invalid and be discarded. The decoder should continue sampling the input through the guard time in order to catch the next header and transmission (i.e., for a decoder designed to capture HCS300 transmissions the time spent polling for a valid transmission should be at least 100 ms if activity is detected in the input line).

The diagram below (Figure 5) gives all the major sampling points in the receive algorithm.

Note: The sample points are labeled in the receive routine flow diagrams that follow.

FIGURE 5: SAMPLING POINTS USED IN RECEIVE ALGORITHM



Flow Diagrams

The first flow diagram (Figure 6) describes the calibration routine which is used to determine the actual transmission rate of the encoder so that the decoder can compensate for deviations from nominal timing. There are four different exit points, each of which should branch to a point in the program where housekeeping and input monitoring can be resumed. There is only one exit point for a valid calibration operation (RCV7). At this point, it is assumed that a valid header has been received and that a string of data bits will follow.

The second flow diagram (Figure 7) handles the reception of bits once the calibration routine has been successfully completed. The data bits are all sampled three times each to ensure that a noise free transmission has been received. The receive routine uses the calibrated elemental period, determined in the calibration routine, to ensure that the samples are spaced correctly. The routine resynchronizes itself on the rising flank of each bit. Provision for identification of the encoder type, based on the number of bits received, is included. KEELOQ encoders transmit at least 66 bits. The decoder only receives 64 bits of the transmission, the remaining status bits aren't used by this decoder. The last two bits of the HCS300 transmission, VLOW and Repeat, are ignored by the Microchip decoder.

If all of the control samples in all of the bits are sampled correctly (i.e., the first element is high and the last element is low), the routine checks whether 56 bits have been received correctly. If not, the routine returns to the calling procedure.

FIGURE 6: CALIBRATION FLOW CHART

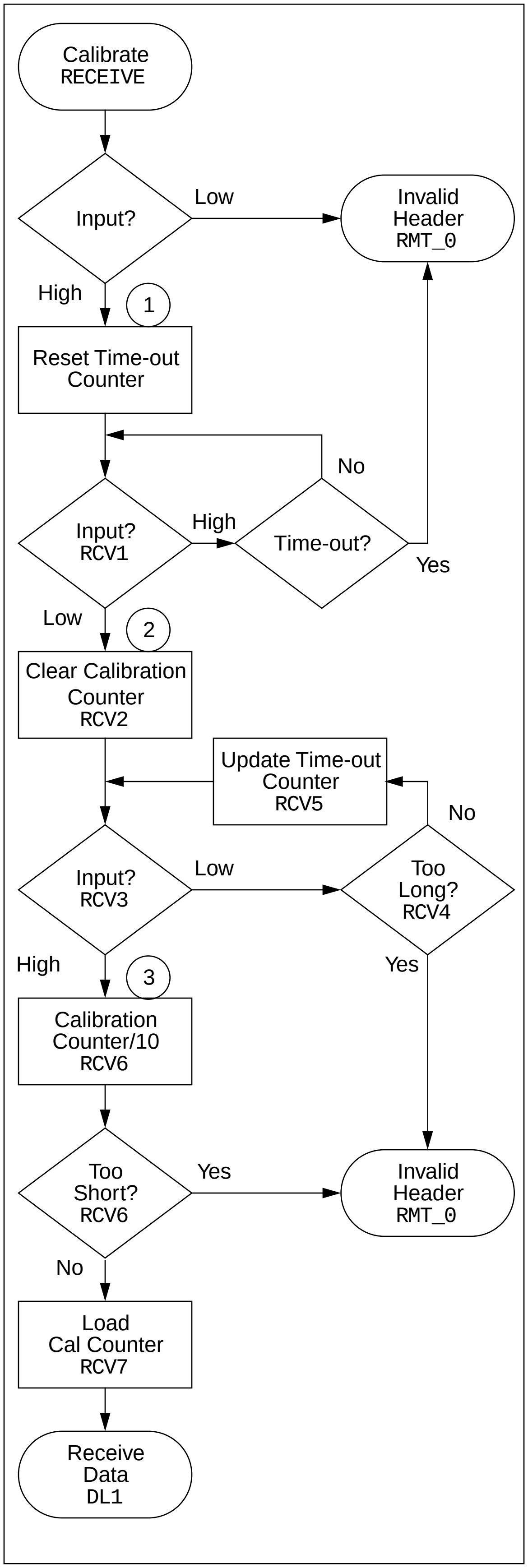
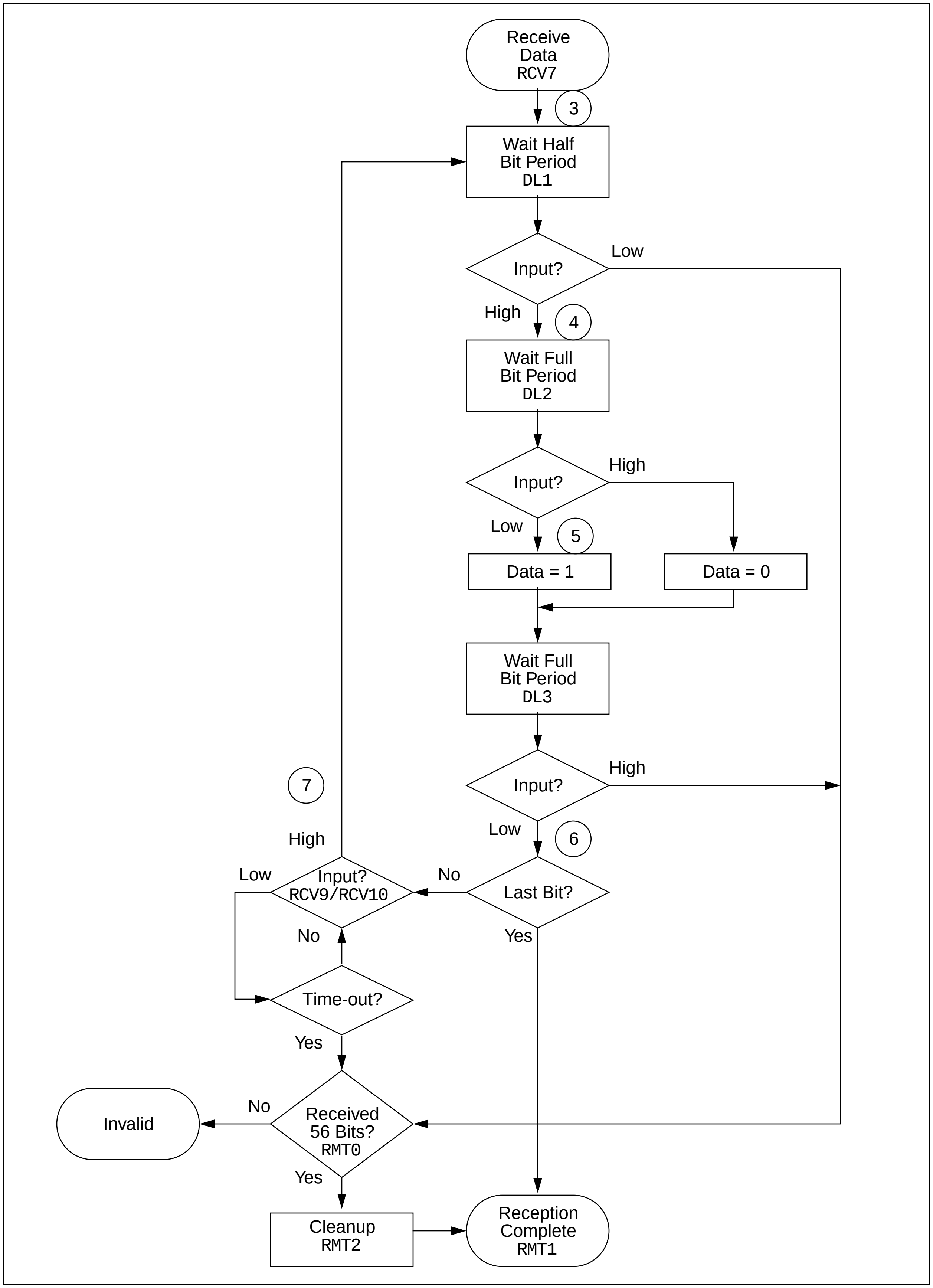


FIGURE 7: DATA RECEPTION IN RECEIVE



Validation

Once a complete transmission has been received from an encoder, the transmission needs to be validated before any further action is taken. Validation consists of the following steps:

1. Check the serial number (24, 28 bits) against the stored encoder serial numbers (M_SERIAL).
2. Decrypt the transmission received (M_HOP).
3. Compare the discrimination value in the decrypted hopping portion of the transmission against the stored discrimination value (M_DIS).
4. Check if the synchronization counter falls within the resynchronization window (M_CHECK1).
5. Check if the synchronization counter falls within the open window. If not, then decoder resynchronization is necessary (M_CHECK2).
6. If resynchronization is necessary wait for a second transmission from the encoder with a consecutive synchronization counter.
7. Update the synchronization counter in EEPROM (M_UPDATE).
8. Set the appropriate outputs (M_BUT).
9. Return to MAIN routine and continue normal housekeeping chores.

Discrimination Values

After decryption, the Code Shift Register (CSR) used by the KEELOQ decryption algorithm contains the same 32 bits of information originally encrypted in the encoder before transmission. 12 of these bits are discrimination bits.

The decryption operation can be checked by comparing parts of the decrypted 32-bit word (the discrimination values) with known values.

For the HCS300/301 the user can program the discrimination bits to contain any value. In the HCS360/361 the discrimination bits are the least significant 8 bits of the serial number. The discrimination bits are stored in the external EEPROM during learn. By comparing the discrimination bits to the bits expected, the integrity of the decryption can be easily verified.

Note: The overflow bits (when available in an encoder) are treated as part of the discrimination value. For example, these bits can be set when the HCS300/301 encoders are programmed. When the encoder's counter overflows the overflow bits are individually cleared extending the counter range from 65,536 to 196,608. The clearing of the overflow bits will result in an 'erroneous' discrimination value being decrypted because the overflow bits are stored as part of the discrimination value. The transmission will be treated as an invalid transmission by this decoder and the transmission discarded. In order to avoid this, the HCS300/301's overflow bits should both be programmed as '0' and the synchronization counter started at '0'.

TABLE 9: HCS200/201, HCS300/301 DECRYPTED HOPPING CODE TRANSMISSION FORMAT

Function*	MSB Encoder disc. bits LS	MSB Synchronization counter LSB
(4 bits)	(12 bits including overflow bits)	(16 bits)

* The HCS200/201 has padding in S3 button position since no S3 button is present. In addition the HCS200/201 does not have overflow bits present and these are also padded.

Synchronization Checking

The synchronization information is used at the decoder to determine whether the transmission is valid or whether it is a repetition of a previous transmission. Repetitious codes are rejected to safeguard the system against code grabbers.

The transmitting encoder has a 16-bit synchronization counter, stored in EEPROM, which is incremented every time the encoder is activated. The synchronization counter value is stored in the decoder's EEPROM every time a valid transmission is received from a particular encoder. When a following transmission is received from the same transmitter it is possible to quickly verify whether the transmission is valid. For example, a grabbed code from the legitimate user's previous transmission will result in a synchronization counter value, that has already been received, being decrypted.

Note:	Two copies of the synchronization counter are stored. The reason for this is should the power go down during an EEPROM write, a corrupted counter value would be read when the device is later powered up, resulting in encoder transmissions erroneously being discarded as invalid.

Provision must be made for the transmitter being pressed while out of range of the decoder. The Microchip decoder does this by allowing two 'synchronization windows'. The **open window** is a reception of a transmission where the synchronization counter is 1 to 16 higher than the previous counter value received. The reception of such a signal will result in an immediate counter update by the decoder and the appropriate outputs being activated.

If the transmitter is pressed more than 16 times out of range of the receiver, resynchronization needs to take place. The **resynchronization window** is a half of the total counter range, 32K big. During resynchronization the decoder waits for two consecutive transmissions from the encoder before resynchronization takes place and the resynchronized counters updated in the decoder's EEPROM. When the decoder receives a transmission with a synchronization counter value more than 16 above the stored counter value and less than 32,768 counts above the stored value, the decoder temporarily stores the value of the synchronization counter received. If the next transmission received has a sequential synchronization counter value the decoder resynchronizes on the last transmission received and activates the appropriate outputs.

If any of the above tests fail the transmission received is discarded. It is easy to change the size of the various windows in the source code. Modifications to the synchronization windows can be made in the M_CHECK routine.

FIGURE 8: DECODER WINDOW OPERATION

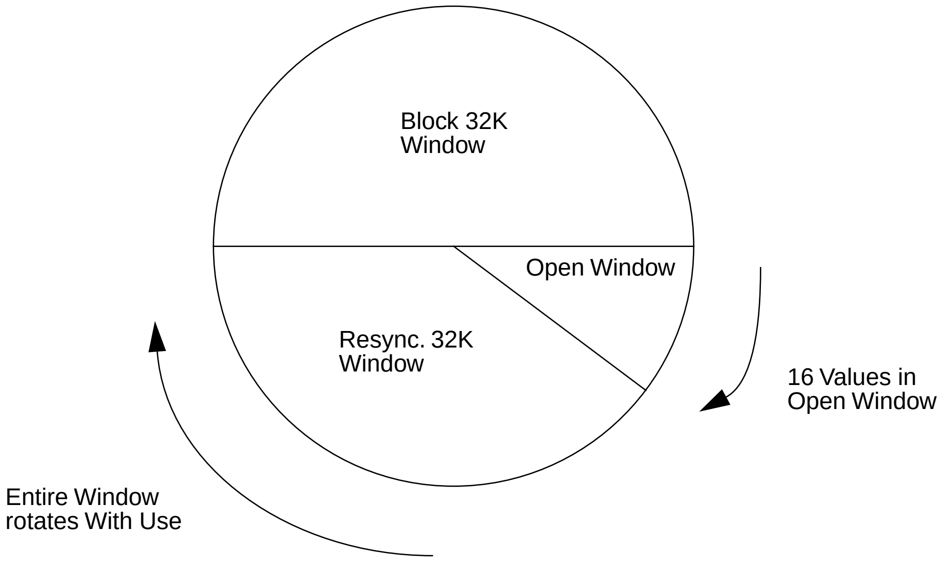
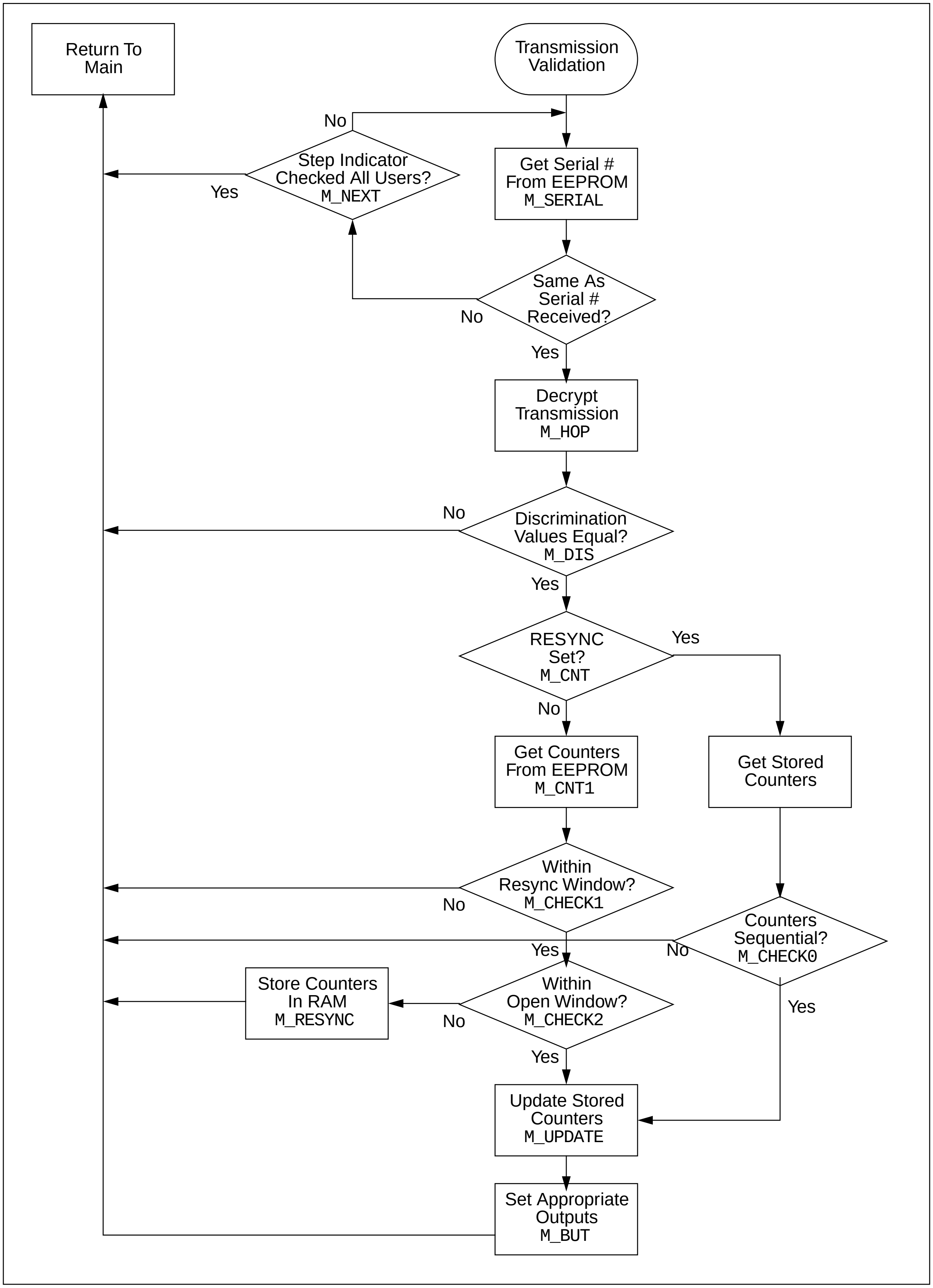


FIGURE 9: VALIDATION FLOW CHART TRANSMISSION VALIDATION FLOWCHART


Return To Main
Transmission Validation
No
Step Indicator Checked All Users? M_NEXT
Yes
Get Serial # From EEPROM M_SERIAL
Same As Serial # Received?
No
Yes
Decrypt Transmission M_HOP
No
Discrimination Values Equal? M_DIS
Yes
RESYNC Set? M_CNT
Yes
No
Get Counters From EEPROM M_CNT1
Get Stored Counters
Within Resync Window? M_CHECK1
No
Counters Sequential? M_CHECK0
Yes
No
Store Counters In RAM M_RESYNC
Within Open Window? M_CHECK2
No
Yes
Yes
Update Stored Counters M_UPDATE
Set Appropriate Outputs M_BUT

Function Interpretation

In a single-chip system, where the code hopping decoder and the control program are combined into one device, the function code is interpreted to determine what the system must do. One function can be used to arm the system and lock the vehicle, a second to disarm the system and unlock the vehicle, and a third to open the trunk.

The four function bits in the encrypted portion of a transmission can be used to determine the button(s) pressed on the transmitter. Up to 15 functions can be implemented in this way, 0000 being related to a reset state.

The four function bits transmitted by the KEELOQ encoders are labeled F2, F1, F0 and F3. These correspond to S2, S1, S0, and S3 in the HCS300/301, S2, S1, S0, and S2 in the HCS200/201. The Sn bits in turn correspond with the values of the control inputs of the encoders.

In the Microchip decoder the function code received from the encoder is put onto the function outputs (S0 to S3) if a valid transmission is received (M_BUT). In addition, if the button code that was learned into the system is received, the FUNC OK output is activated.

Output Activation

The Microchip decoder has five momentary outputs namely S0, S1, S2, S3, and FUNC OK. As described in the section on Function Interpretation these outputs are a function of the inputs activated on the encoder. The momentary outputs are activated for 524 ms and extended for 524 ms if a repeated transmission is received. If a new valid transmission with a different function code is received during output activation, the outputs are switched off for 131 ms and the new function output activated.

FUNC OK is set if the function code received by the decoder is the same as the function code received during the learn cycle. The routine displaying the function code information and checking whether the function code received is the same as the one used during learn is called M_BUT.

Key Generation

Before transmitting the hopping portion of the code an encoder uses a 64-bit read protected encryption key to encrypt the information. In order to read the information contained in the hopping portion of the code it is necessary for the decoder to decrypt the data.

The use of a manufacturer’s code in key generation allows a unique relationship between encoder serial number and encryption/decryption key pair. This enables each manufacturer to produce encoders that cannot be cloned by competitors. Security of the manufacturer’s code is critical to product security and as a result the manufacturer’s code (MKEY) is stored in ROM in the PIC16C56 microcontroller.

Two key generation techniques are used in the KEELOQ decoders. The first is a normal key generation algorithm which is used in the Microchip decoder described. The second key generation technique uses a feature on the Microchip HCS encoders called SEED transmission. In this mode the transmitter transmits a SEED programmed during learn instead of the code hopping.

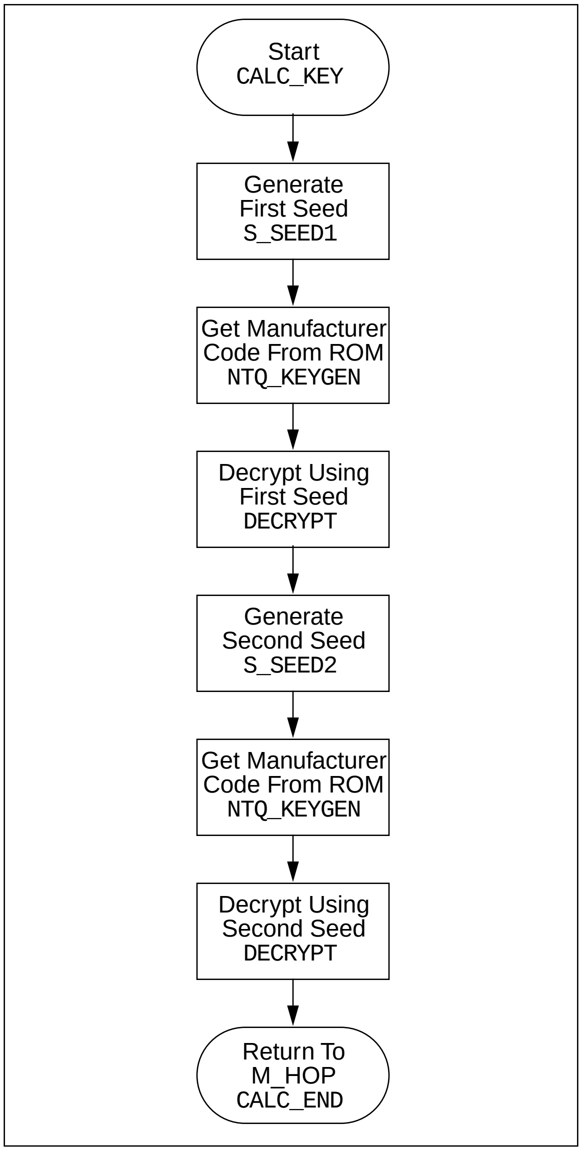
Normal Key Generation

The Microchip decoder described uses the normal key generation technique to generate decryption keys for the encoders. Normal key generation is performed in two steps as shown in the flow chart in Figure 10. The first step generates the least significant 32 bits of the decryption key, the second the most significant 32 bits of the decryption key. The relationship between decryption key and serial number during key generation is achieved by using the serial number of the encoder as a seed for the decryption routine. The length of the serial number is 28 bits for the HCS200/201 and the HCS300/301. The serial number of the encoder is padded to make the key generation seed 32 bits. To allow the most significant 32 bits of the decryption key to differ from the least significant 32 bits the serial number is padded to 32 bits differently on each step. The manufacturer’s code is used as the key for the decryption operation during key operation. Table 10 shows the different padding techniques used.

TABLE 10: SEED GENERATION PADDING FOR VARIOUS ENCODERS

	HCS encoders
Seed1 padding	2*******16
Seed2 padding	6*******16

FIGURE 10: NORMAL KEY GENERATION FLOW CHART

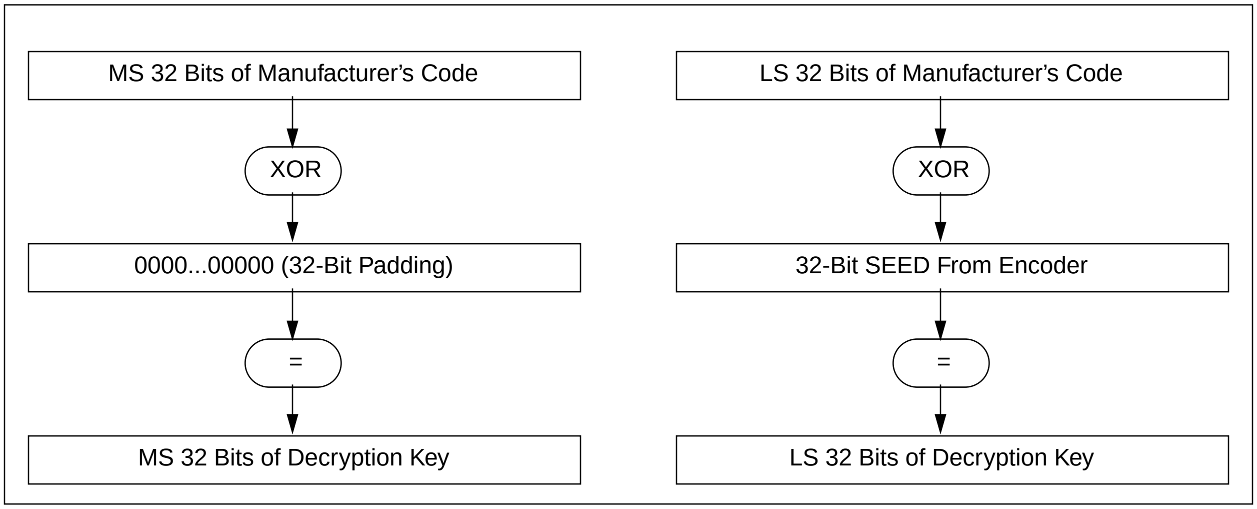


Secure Key Generation

Secure key generation relies on the encoder to supply the seed used to generate a key. The Microchip HCS encoders are able to transmit a fixed 32-bit value (SEED) in place of the code hopping during the learn process. The SEED is padded with zeros to make up 64 bits and then XOR'ed with the manufacturer's code. The value that results is the 64-bit decryption key that is used to decrypt the hopping portion of a transmission.

Since the random seed is only transmitted during the learning process, and is required to generate the key a normal hop transmission cannot be intercepted, a key generated and the hop code decrypted to predict the next hopping code. Figure 11 shows how the secure key generation technique described can be implemented.

FIGURE 11: SECURE KEY GENERATION


MS 32 Bits of Manufacturer's Code
XOR
0000...00000 (32-Bit Padding)
=
MS 32 Bits of Decryption Key
LS 32 Bits of Manufacturer's Code
XOR
32-Bit SEED From Encoder
=
LS 32 Bits of Decryption Key

Decryption

After receiving a valid transmission the decoder decrypts the code hopping portion of the transmitted code. This is done with the help of the KEELOQ decryption algorithm. Each encoder has its own encryption key which is related to the serial number of the encoder and is calculated when the encoder is programmed. In order to decrypt the transmission a decryption key is calculated by the decoder during the learn cycle and stored in the EEPROM. The decryption routine is called DECRYPT.

The KEELOQ decryption algorithm is used to decrypt the 32-bit code hopping portion of KEELOQ transmissions. A 32-bit Code Shift Register (CSR) contains the received code, and a 64-bit register contains the decryption key

The 64-bit decryption key is retrieved from EEPROM into RAM for every decryption operation. Several 64-bit keys are stored in memory, one for each valid transmitter. The key particular to a given encoder is retrieved to decrypt the code hopping portion of a particular encoder. The key to be retrieved is identified by the serial number of the encoder that is transmitted.

The block diagram (Figure 12) explains the operation during each iteration of the decryption algorithm. A non-linear function (NLF) is used to produce a single bit from five bits in the CSR. This output is combined, via an exclusive-OR function, with two CSR bits and a single-bit from the key register to form an output. At the end of each cycle, the key register is rotated left, and the CSR is rotated left. The MSB (bit 3,7) of the CSR is discarded, and the output from the exclusive-OR function is inserted into the LSB (bit 0,0) of the CSR.

The decryption operation requires 528 iterations. In other words, the operation in the block diagram should be executed 528 times before the decrypted data will appear in the CSR.

The non-linear function (NLF, Table 11) is intended to obscure any linear relationships that might otherwise exist in the encrypted output. The NLF is listed in the form of a 5-bit lookup table, in which the five input bits are $I_4 = CSR_{3,6}$, $I_3 = CSR_{3,1}$, $I_2 = CSR_{2,3}$, $I_1 = CSR_{1,0}$, and $I_0 = CSR_{0,0}$.

FIGURE 12: THE KEELOQ DECRYPTION ALGORITHM

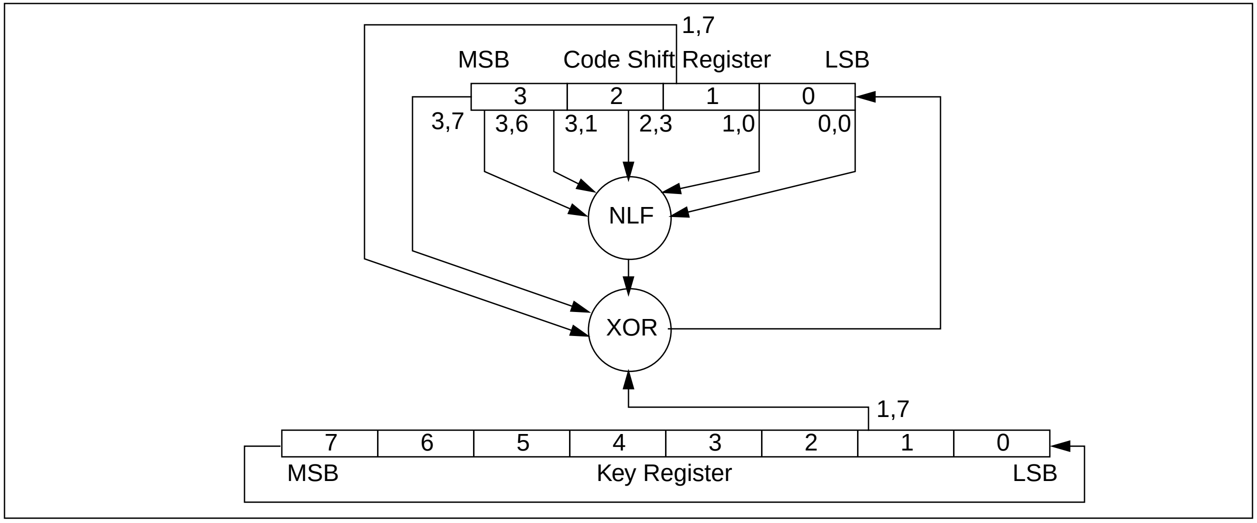


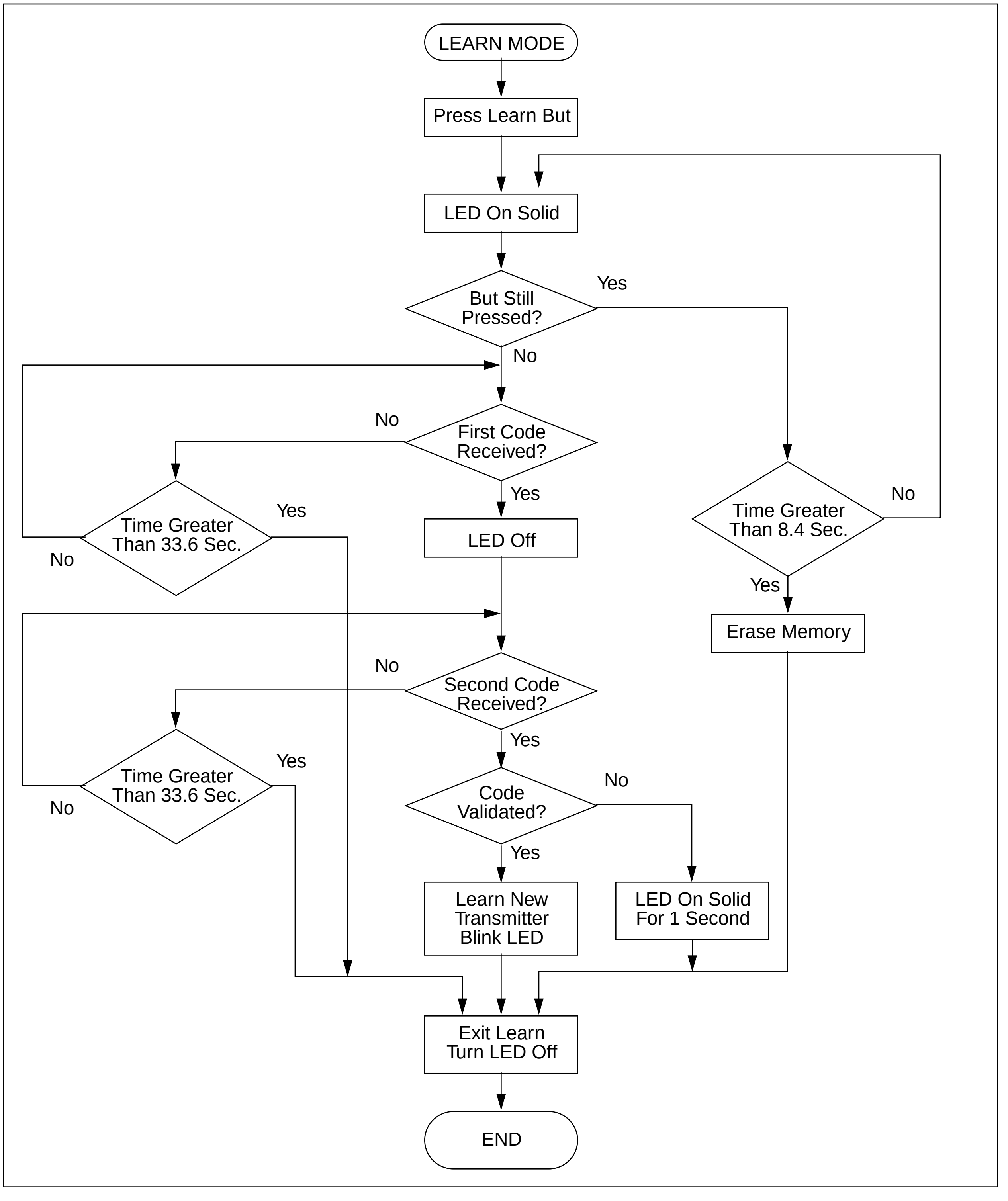
TABLE 11: NON-LINEAR FUNCTION OUTPUT

I4	I3	I2	I1	I0	NLF	I4	I3	I2	I1	I0	NLF
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0	1	0
0	0	0	1	0	1	1	0	0	1	0	1
0	0	0	1	1	1	1	0	0	1	1	1
0	0	1	0	0	0	1	0	1	0	0	1
0	0	1	0	1	1	1	0	1	0	1	0
0	0	1	1	0	0	1	0	1	1	0	1
0	0	1	1	1	0	1	0	1	1	1	0
0	1	0	0	0	0	1	1	0	0	0	0
0	1	0	0	1	0	1	1	0	0	1	1
0	1	0	1	0	1	1	1	0	1	0	0
0	1	0	1	1	0	1	1	0	1	1	1
0	1	1	0	0	1	1	1	1	0	0	1
0	1	1	0	1	1	1	1	1	0	1	1
0	1	1	1	0	1	1	1	1	1	0	0
0	1	1	1	1	0	1	1	1	1	1	0

Learn

A learn indicator is used by the Microchip decoder to keep track of the next position a learn is to take place. The encoder positions in EEPROM form a rotating buffer where the next transmitter to be written over is the transmitter at the tail of the buffer. The LEARN INIT input is active low and the LEARN IND output active high. Learn is initiated by momentarily pressing the LEARN button. The decoder uses the current learn position as a scratch pad area. This means that an unsuccessful learn deletes the information stored at that learn position. The learn indicator is not incremented if the learn was unsuccessful. The following flow diagram shows the learning operation.

FIGURE 13: LEARN OPERATION



Learn (cont'd)

The following checks are performed on the received codes to determine if the transmitter is valid:

1. The first code that is received is checked for bit integrity (RECEIVE).
2. The stored serial numbers are then searched to check if a transmitter being is re-learned. If a re-learn is taking place that position is used or else the position pointed to by the learn indicator is used (M_SERIAL).
3. The serial number is stored in EEPROM and used to generate a decryption key (CALC_KEY).
4. The hop code is decrypted (M_HOP) and the result stored temporarily (M_SL_UPDATE). The counter and serial number are stored (M_UPDATE).
5. The serial number of the second code that is received is compared to the first received serial number (M_SERIAL).
6. The second hop code is decrypted (M_HOP) and the discrimination values compared (M_DIS).
7. The synchronization counters of the decrypted codes are compared to check that they are sequential codes (M_UPDATE).
8. If all the checks pass the learn was successful and the learn indicator is incremented else the position is erased.

Operation of Learn

1. Press and release the LEARN button. Indicator LED will turn on to indicate learn mode.
2. Press transmitter button. The LED will turn off.
3. Press transmitter a second time. The LED will blink to indicate that the transmitter was learned successfully.
4. Repeat steps 1-3 to learn up to six transmitters. The seventh transmitter will overwrite the first transmitter that was learned.
5. Learn will be terminated if two non-sequential codes were received or if two acceptable codes were not decrypted within 33.6 seconds. An invalid learn will be indicated by the LED turning on solid for one second.
6. Erasing all the transmitters is accomplished by pressing and holding the LEARN button for 8.4 seconds. The LED will turn off at the end of the 8.4 seconds to indicate that all the transmitters were erased. The learn indicator is reset to the first position.

TIMER0 (RTCC) Multiplexing

A time keeping scheme is needed to ensure that the system timing is not abandoned while receiving an incoming signal, during learn cycles, key generation and decryption. The system timing is used to allow periodic monitoring of sensors and pulsing outputs with a specific period.

TIMER0 is used to keep track of system time. TIMER0 is an 8-bit timer on the PIC16C56. On the Microchip decoder described, TIMER0 is prescaled to increment every 256 instruction cycles. This makes TIMER0 very useful for keeping track of real time. While various routines are being run, including receive routines and decryption, TIMER0 is periodically checked for a time-out value calculated at the beginning of a certain period (i.e., switch off time of a LED).

The routine checking TIMER0 is called TST_RTCC. The most significant bit (MSB) of TIMER0 changes every 32 ms. In order to extend the range of TIMER0 2 additional 8-bit counters are used, CNT_LW and CNT_HI, which extends the range TIMER0 to 134 seconds. The MSB of TIMER0 is mirrored in the MSB of the STATUS register during startup. During TST_RTCC the 2 bits are compared. If the bits differ, the MSB of TIMER0 has changed indicating that 32 ms has passed. The MSB of STATUS is changed to match the MSB of TIMER0 and the extended counter(CNT_LW and CNT_HI) incremented.

The second portion of the TST_RTCC routine checks appropriate time-out values based on the system status bits in SREG (i.e. To check for the 30s time-out in the learn routine TST_RTCC checks to see if bit three of CNT_HI is set).

ROM MEMORY MAP (8-BIT BYTES)

TABLE 12: ROM MEMORY MAP (8-BIT BYTES)

Word Address	Mnemonic	Description
40	MKEY_0	64-bit Manufacturer's Code (Used to generate decryption keys)
41	MKEY_1	
42	MKEY_2	
43	MKEY_3	
44	MKEY_4	
45	MKEY_5	
46	MKEY_6	
47	MKEY_7	
48	Unused	
49	Unused	
4A	EKEY_0	64-bit EEPROM Key (Used to encrypt EEPROM data)
4B	EKEY_1	
4C	EKEY_2	
4D	EKEY_3	
4E	EKEY_4	
4F	EKEY_5	
50	EKEY_6	
51	EKEY_7	

EEPROM MEMORY MAP (16-BIT WORDS)

TABLE 13: EEPROM MEMORY MAP (16-BIT WORDS)

Address	Mnemonic	Address	Mnemonic
00	USER0	20	CNT20
01	Learn Ind.	21	CNT21
02	DIS0	22	SER20
03	DIS1	23	SER21
04	USER2	24	KEY20
05	USER3	25	KEY21
06	USER4	26	KEY22
07	USER5	27	KEY23
08	DIS2	28	CNT30
09	DIS3	29	CNT31
0A	DIS4	2A	SER30
0B	DIS5	2B	SER31
0C	USER6	2C	KEY30
0D	USER7	2D	KEY31
0E	USER8	2E	KEY32
0F	USER9	2F	KEY33
10	CNT00	30	CNT40
11	CNT01	31	CNT41
12	SER00	32	SER40
13	SER01	33	SER41
14	KEY00	34	KEY40
15	KEY01	35	KEY41
16	KEY02	36	KEY42
17	KEY03	37	KEY43
18	CNT10	38	CNT50
19	CNT11	39	CNT51
1A	SER10	3A	SER50
1B	SER11	3B	SER51
1C	KEY10	3C	KEY50
1D	KEY11	3D	KEY51
1E	KEY12	3E	KEY52
1F	KEY13	3F	KEY53

USER These words are reserved for user storage.

SER The encoder serial number storage

KEY These words contain the decryption key for each encoder.

DIS Discrimination values and function code storage.

CNT Two copies of the synchronization counter are stored for each encoder to prevent loss of synchronization information due to EEPROM write failure.

RAM MEMORY MAP (8 BIT BYTES)

TABLE 14: RAM MEMORY MAP (8 BIT BYTES)

Address	Mnemonic	Description
07	FLAGS	Decoder flags
08	ADDRESS	Address register - points to address in EEPROM
09	TXNUM	Current transmitter
0A	OUTBYT	General data register, mask register used in decryption
0B	CNT0	Loop counters
0C	CNT1	
OD	CNT2	
OE	CNT_HI	16-bit clock counter
OF	CNT_LO	
10	TMP1	Temporary registers
11	TMP2	
12	TMP3	
13	TMP4	
14	CSR4	64-bit shift register Used in reception, decryption and key generation
15	CSR5	
16	CSR6	
17	CSR7	
18	CSR0	
19	CSR1	
1A	CSR2	
1B	CSR3	
1C	OLD_BUT	Store previous button code
1D	RAM_HI	16-bit RAM counter (used in resynchronization)
1E	RAM_LW	
1F	SREG	Program state register

ALTERNATE NAMES AND FUNCTIONS

Many of the memory locations in RAM are used by multiple routines. A list of alternate names and functions are given in the table below.

TABLE 15: ALTERNATE NAMES AND FUNCTIONS

Address	Mnemonic	Also known as	Description
11	KEY0	TMP2	64-bit shift register holds decryption key
10	KEY1	TMP1	
12	KEY2	TMP3	
13	KEY3	TMP4	
14	KEY4	CSR4	
15	KEY5	CSR5	
16	KEY6	CSR6	
17	KEY7	CSR7	
18	HOP1	CSR0	32-bit hop code register
19	HOP2	CSR1	
1A	HOP3	CSR2	
1B	HOP4	CSR3	
1B	DAT1	CSR3	32-bit data register
1A	DAT2	CSR2	
19	DAT3	CSR1	
18	DAT4	CSR0	
OD	ETMP1	CNT2	Extended 32-bit buffer used during key generation as a 32-bit buffer
1C	ETMP2	OLD_BUT	
1D	ETMP3	RAM_HI	
1E	ETMP4	RAM_LW	
17	SER_0	CSR7	24/28-bit serial number, stores received transmission open 32-bits
16	SER_1	CSR6	
15	SER_2	CSR5	
14	SER_3	CSR4	
1B	FUNC	CSR3	Button code and user nibble of discrimination value
1A	DISC	CSR2	Discrimination value
19	CNTR_HI	CSR1	16-bit received counter
18	CNTR_LW	CSR0	

DEVICE PINOUTS

The device used in the application note is a PIC16C56 PDIP.

TABLE 16: DEVICE PINOUTS

PIN	PIC16C56 function	Decoder function	PIN	PIC16C56 function	Decoder function
1	PORTA Bit2	LEARN Input Active Low	18	PORTA Bit1	RF Input
2	PORTA Bit3	LRN IND Output Active High	17	PORTA Bit0	Not used
3	TIME	Connect to VDD	16	Osc In	RC osc (4 MHz)
4	/MCLR	Brown out detect	15	Osc Out	
5	GND	Ground	14	VDD	+5V supply
6	PORTB Bit0	S0	13	PORTB Bit7	FUNC OK
7	PORTB Bit1	S1	12	PORTB Bit6	EEPROM CS (1)
8	PORTB Bit2	S2	11	PORTB Bit5	EEPROM CLK (2)
9	PORTB Bit3	S3	10	PORTB Bit4	EEPROM DIO (3+4)

TIMING PARAMETERS

TABLE 17: TIMING PARAMETERS

Parameter	Typical	Unit
Output activation duration	524	ms
Output pause if new function code received	131	ms
Erase all duration	8.4	s
Learn mode time-out	33.6	s
Learn successful LED flash duration	4.2	s
Learn successful LED flash rate	3.8	Hz
Learn failure LED on duration	1	s

SOURCE CODE LISTING

A diskette is supplied containing source code for the Microchip decoder in the file mcdec12.asm. The code has been compiled using MPASM v01.30.01. Certain functions are dependent on the oscillator speed for correct functioning. Examples of time dependent functions include RECEIVE and TST_RTCC. The PIC16C56 Microcontroller should run at 4 MHz.

LIST OF IMPORTANT FUNCTIONS

Function Name	Description	Function length
CALC_KEY	Key generation routine.	62
DECRYPT	Decryption routine for Code Hop.	61
EEREAD	The data in the EEPROM at ADDRESS is read and decrypted to TMP1 and TMP2. Note that TMP1, TMP2 and ADDRESS are user defined registers.	31
EEWRITE	The data in TMP1 and TMP2 is encrypted and written to the EEPROM at ADDRESS. Note that TMP1, TMP2 and ADDRESS are user defined registers.	48
M_DIS	Check discrimination value.	19
M_CNT	Check synchronization (counter) values.	69
RECEIVE	Start of the RF receive routine.	121
TST_LEARN1	Check for learn mode and entry to learn.	21
TST_RTCC	Check TIMER0 and do whatever real time tasks are required.	38

APPENDIX SCHEMATIC DIAGRAMS

FIGURE 14: SCHEMATIC DIAGRAM OF MICROCHIP KEELOQ DECODER

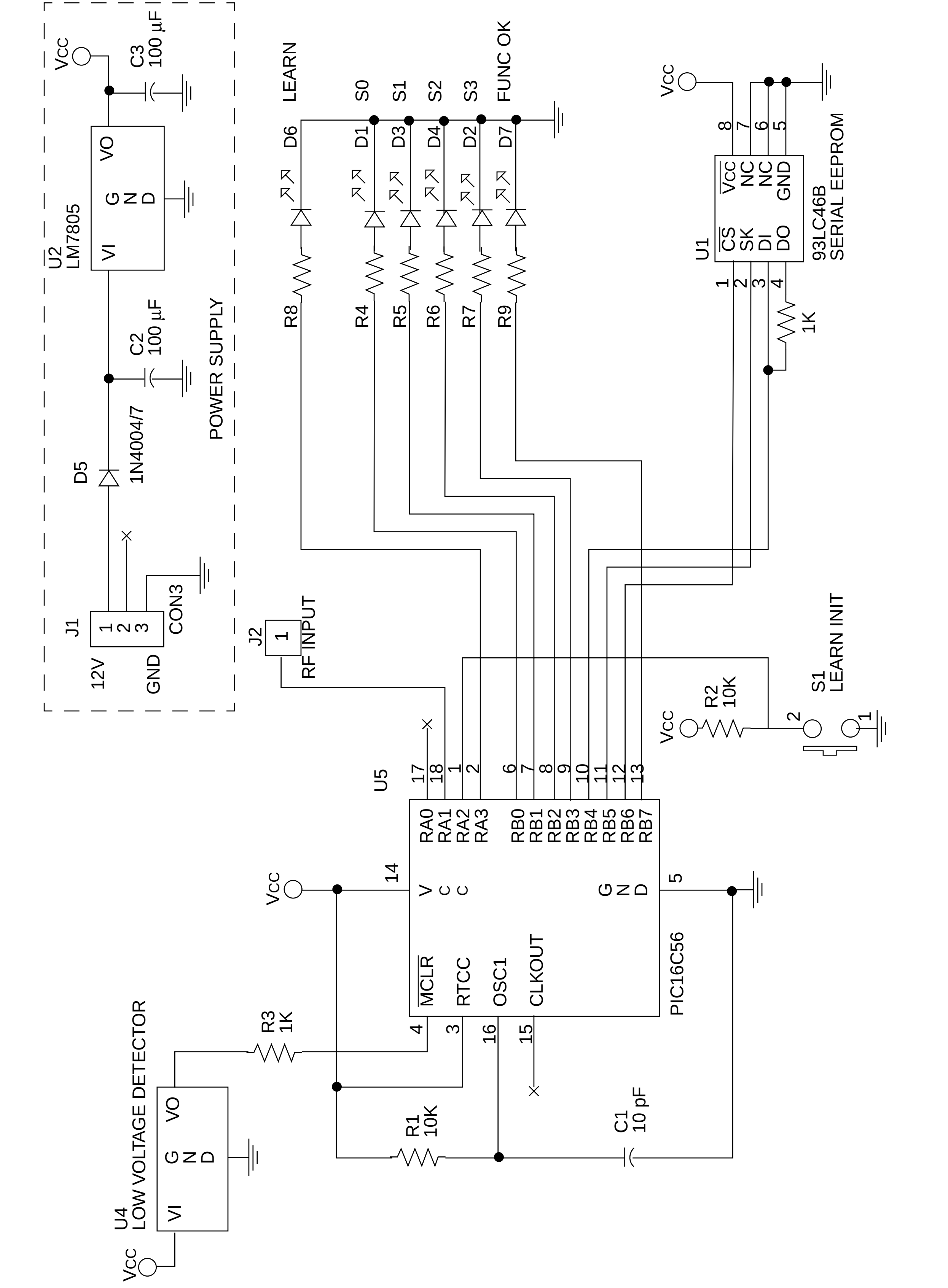
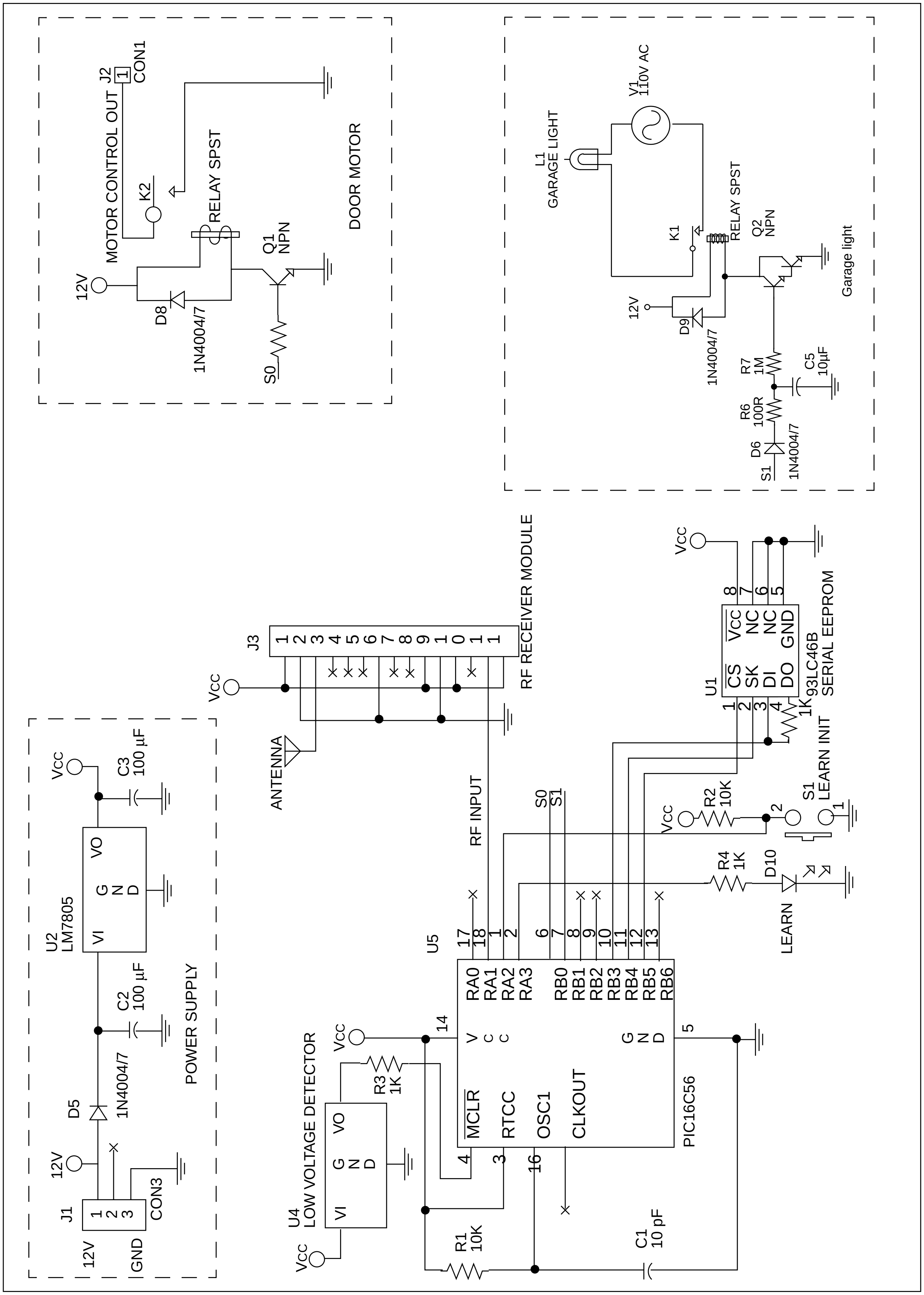


FIGURE 15: TYPICAL GARAGE DOOR OPENER SCHEMATIC



NOTES:

NOTES:

NOTES:

MICROCHIP®

WORLDWIDE SALES AND 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

Detroit

Microchip Technology Inc.
42705 Grand River, Suite 201
Novi, MI 48375-1727
Tel: 248-374-1888 Fax: 248-374-2874

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 202
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

AMERICAS (continued)

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

Japan

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa 222-0033 Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

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, Hong Qiao District
Shanghai, PRC 200335
Tel: 86-21-6275-5700 Fax: 86 21-6275-5060

ASIA/PACIFIC (continued)

Singapore

Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 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: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

United Kingdom

Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-1189-21-5858 Fax: 44-1189-21-5835

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 München, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy

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

9/29/98

DNV Certification, Inc.
USA
ACCREDITED REGISTRAR
ANSI-RAB

DNV

DNV MSC
The Netherlands
Accredited by the RvA

ISO 9001
REGISTERED FIRM

Microchip received ISO 9001 Quality System certification for its worldwide headquarters, design, and wafer fabrication facilities in January, 1997. Our field-programmable PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, related specialty memory products and development systems conform to the stringent quality standards of the International Standard Organization (ISO).

Information contained in this publication regarding device applications and the like is intended for suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.