AN659

KEELOQ™ Simple Code Hopping Decoder

Author: Steven Dawson
Standard Microcontroller and
ASSP Division

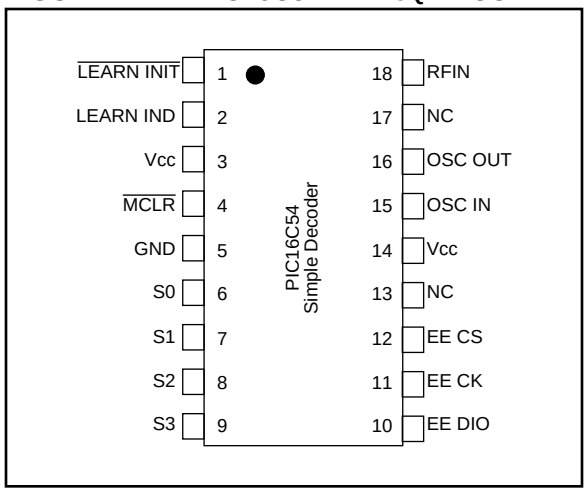
OVERVIEW

This application note fully describes the working of a code hopping decoder implemented on a Microchip PIC16C54 microcontroller. The PIC16C54 is smaller than the PIC16C56 used in the normal decoder (AN642) or the secure learn decoder (AN652). A simplified learning scheme with all encoders sharing a common key and a simplified counter synchronization scheme has been used to reduce the code space requirements. The use of a common key reduces the security of the system. The simple decoder can learn up to 15 encoders. This application note describes the various KEELOQ code hopping encoders that can be used with the decoder, the decoder hardware, and 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, HCS300, HCS301, HCS360, and HCS361 KEELOQ code hopping encoders.

KEY FEATURES

- PIC16C54
- Stand alone decoder
- Compatible with Microchip HCS200, HCS300, HCS301, HCS360, and HCS361 encoders
- Automatic bit rate detection
- Automatic encoder type detection
- Four function outputs
- Up to 15 learnable transmitters
- RC Oscillator
- Single press learn

FIGURE 1: PIC16C54 KEELOQ DECODER



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 and is encrypted. The second part is the unencrypted part of the transmission, principally containing the encoder's serial number identifying it to a decoder. Refer to DS91002, Introduction to KEELOQ.

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.

Discrimination Value

Stored in the encoder EEPROM, this information can be used to check integrity of decryption operation by a 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. 12 bits (including overflow bits) are available in the Microchip HCSXXX encoders.

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 is a repetition of a previous transmission. Previous codes are rejected to safeguard against code grabbers. The HSC300 and HCS301 encoders transmit two overflow bits which may be used to extend the range of the synchronization counter from 65,536 to 196,608 button operations. The HCS360 and HCS361 encoders transmit one overflow bit which can be used to extend the range of the synchronization counter from 65,536 to 131,071 button operations.

Unencrypted 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.

Other Status and Function Information

The HCS200, HCS300, and HCS301 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 does not send repeated transmission information, and the bit is permanently set to '0'.

The HCS360/361 encoders transmit two bits that are used as a Cyclic Redundancy check. These bits can be used to check the integrity of the reception. Additionally, the HCS360 and HCS361 encoders can extend the length of the serial number from 28 bits to 32 bits, replacing the unencrypted function code.

Transmission Format Summary

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 2: DECODER BLOCK DIAGRAM

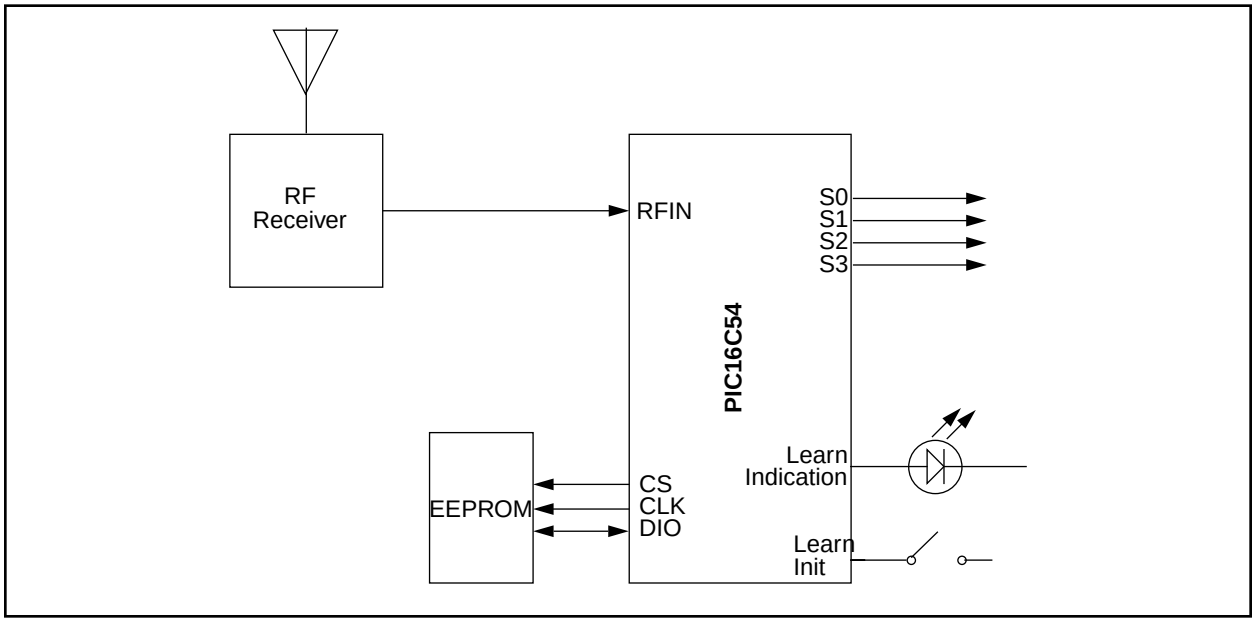


TABLE 1: KEELOQ ENCODER TRANSMISSION SUMMARY

	HCS200[1]	HCS300/301	HCS360/361
Total transmission length	66 bits	66 bits	67 bits
Hopping code portion (total length)	32 bits	32 bits	32 bits
Function bits	3 bits (+1[1])	4 bits	4 bits
Discrimination bits	12 bits	12 bits	12 bits
Synchronization	16 bits	16 bits	16 bits
Fixed portion (total length)	34 bits	34 bits	35 bits
Serial number	28 bits	28 bits	28/32 bits[2]
Function bits	3 bits (+1[1])	4 bits	4/0 bits[2]
Status bits			
VLOW	1 bit	1 bit	1 bit
RPT	0	1 bit	0
CRC	0	0	2 bits

[1]The HCS200 transmissions are padded to retain compatibility with the HCS300/301 encoders.
[2]User selectable with a 32-bit serial number or a 28-bit serial number, and 4 unencrypted function bits.

The code hopping portion is always transmitted first (LSB to MSB), followed by the fixed portion (LSB to MSB).

TABLE 2: HCS200† AND HCS300/301 TRANSMISSION FORMAT

Hop Code			Serial Number			Function	Status
LSB	32 bits	MSB	LSB	28 bits	MSB	4 bits	2 bits

†The HCS200 transmissions are padded to retain compatibility with the HCS300/301 encoders.

TABLE 3: HCS360/361 TRANSMISSION FORMAT

Hop Code			Serial Number			Function	Status
LSB	32 bits	MSB	LSB	28/32 bits	MSB	4/0 bits	3 bits

Bits are transmitted from left to right (i.e., the code hopping part first).

PWM Format

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

A **preamble** to improve biasing of decision thresholds in super-regenerative 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 66 or 67 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 3 shows the sampling points when sampling the data bits. The first and last elements are used exclusively to verify the integrity of the received symbol. 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, 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 $2^{-132} < 2 \times 10^{-40}$.

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 3: KEELOQ PWM TRANSMISSION FORMAT

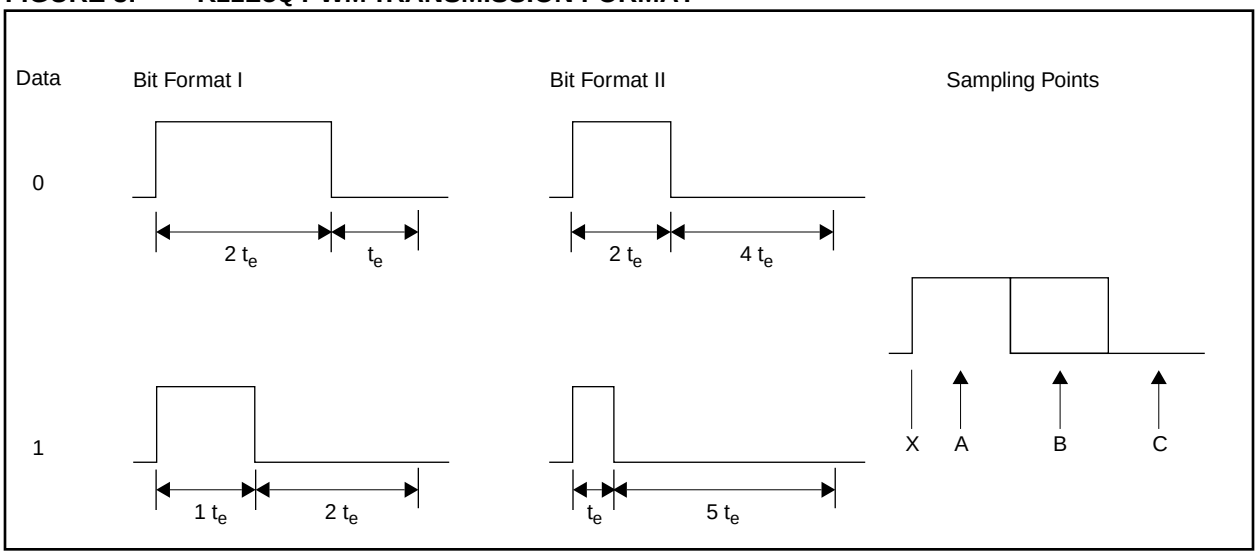


TABLE 4: TRANSMISSION FORMATS

Encoder	Preamble	Header Chopping	PWM Bit Format		Hopping Code Length	Unencrypted Code Length
			I	II		
HCS200	•	•	•	—	32 bits	34 bits
HCS300/301	•	•	•	—	32 bits	34 bits
HCS360	•	•	•	—	32 bits	35 bits
HCS361	•	•	•	•	32 bits	35 bits

DECODER IMPLEMENTATION

The decoder is implemented on a PIC16C54 RISC microcontroller and a 93C46 EEPROM as shown in the decoder schematic in Figure 14. However, this solution can be implemented in any PIC16/17 microcontroller with at least 500 words of program memory. 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 decoder activates the outputs dictated by the transmission.

Encoder information, such as serial number and synchronization information is stored externally in an EEPROM. The EEPROM used is a Microchip 93C46 Microwire® Serial EEPROM. The information stored in the EEPROM is encrypted to protect the contents. The EEPROM encryption is less secure than the KEELOQ code hopping algorithm.

As can be seen from the section on encoder transmissions there are differences in the transmission formats of the different encoders that can be used with the decoder. 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 described, the complete serial number (28 bits) is stored.

After matching the received and stored serial number, validation of a received transmission consists of two steps. The first includes checking the integrity of the decryption operation. Here the decoder compares the least significant 8-bits of the discrimination value received with the least significant 8-bits of the serial number. The discrimination value stored with the HCS300/301/360/361 includes overflow bits and user 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.

TABLE 5: 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.
$\overline{\text{LEARN INIT}}$	1	I	Input to initiate learning, active low.
LEARN IND	2	O	Output to show the status of the learn process (in an integrated system this will be combined with the system status indicator).
S0, S1, S2, S3	6, 7, 8, 9	O	Function outputs—corresponds to encoder input pins.

Microwire is registered trademark of Motorola.

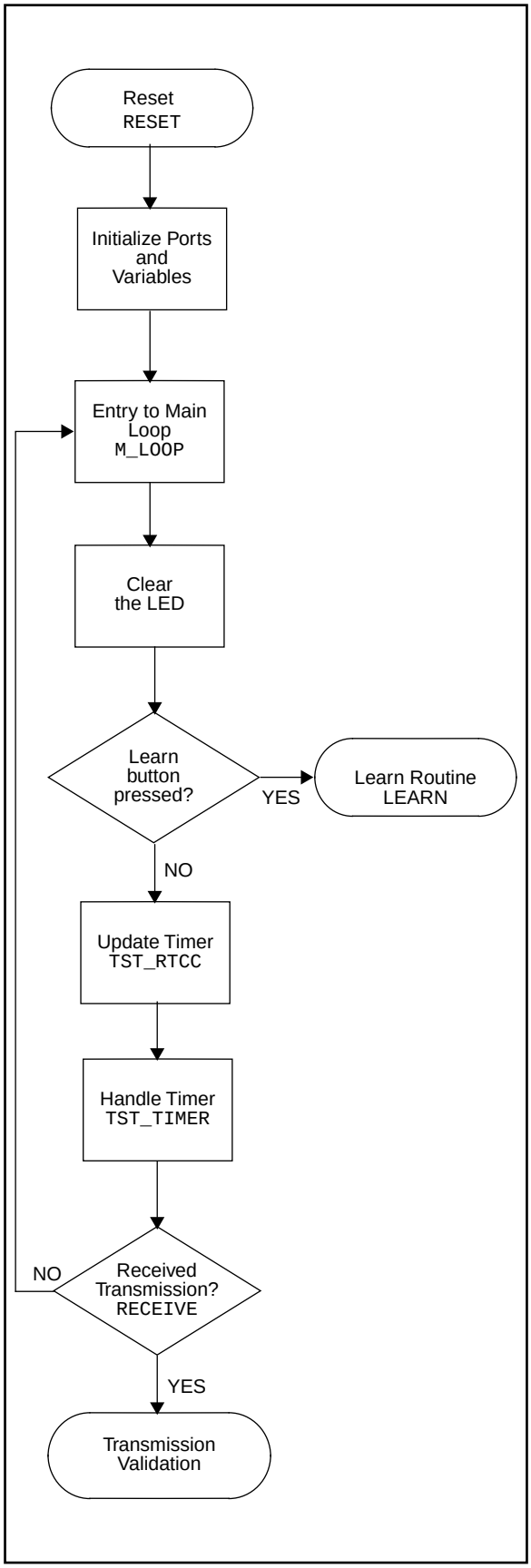
PROGRAM FLOW

The software for the Microchip decoder has been written for the PIC16C54 microcontroller. The compiler used is MPASM version 01.30.01. The operating frequency of the PIC16C54 is 4 MHz. The clock speed should be kept as close as possible to 4 MHz as the reception routine (RECEIVE) is dependent on the 4 MHz clock for correct functioning. Other decoder functions that rely on a 4 MHz clock speed are the hold times of the various outputs and time-outs. The main program flow is described here. Detailed descriptions of individual functions can be found further in the application note.

After startup, the encoder enters the main loop where it spends most of its time. The main loop checks to see if the learn button is being activated. If so, the decoder enters the learn mode described in the “Learn” section (page 15).

If learn has not been initiated, the microcontroller checks for transmissions from encoders (RECEIVE) as described on page 7. If a transmission from an encoder has successfully been received, the microcontroller validates the transmission received as described in the “Transmission Validation” section (page 10). If the transmission received is a valid transmission from an encoder learned into the system, the system sets the appropriate outputs (M_OUTPUT).

FIGURE 4: MICROCHIP DECODER MAIN PROGRAM FLOW



Microwire is a registered trademark of Motorola.

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 bit rate detection is used to compensate for variations in bit rate of different encoders of a specific type, as well as the differences in bit rate between different encoders (HCS200, HCS300, and HCS360). The reception routine is able to receive 64-bit transmissions. The decoder only uses the first 60 bits received. The remaining bits are ingnored.

The reception algorithm performs a number of functions when an output is detected from the receiver. Figure 5 gives all the major sampling points in the reception algorithm.

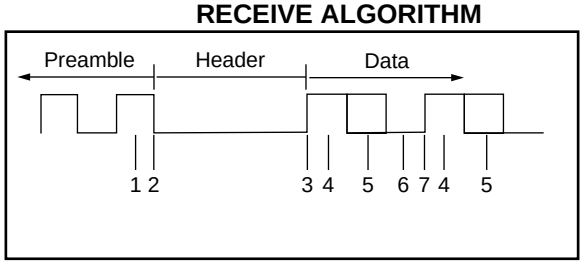
The reception algorithm calibrates on the low period of the header to determine the actual elemental period for the transmission being received. The required elemental period is 10% of the low header period. In Figure 5 the header calibration sample points are marked 1 through 3. The calibration flow chart (Figure 6) shows at what points in the program samples 1, 2, and 3 are taken.

Elemental periods outside the capture range of the algorithm (either too long or too short) are rejected, since they are due either to noise or to reception of an incomplete signal.

Using the determined elemental period, three samples after the first rising edge (sample 3) 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 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.

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).

FIGURE 5: SAMPLING POINTS USED IN RECEIVE ALGORITHM



In systems where the reception 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 reception 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).

Reception Algorithm Flow Chart

The flow chart in 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 chart (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 reception routine uses the calibrated elemental period, determined in the calibration routine, to ensure that the samples are correctly spaced. The routine resynchronizes itself on the rising flank of each bit. Only 60 bits of the data received are used by the decoder described, the decoder ignores the unencrypted function code and the status bits.

If the control samples in a given bit are sampled correctly (i.e., the first element is high and the last element is low), the routine checks whether 66 bits have been received correctly. If not, the routine returns to the calling procedure.

FIGURE 6: CALIBRATION FLOW CHART

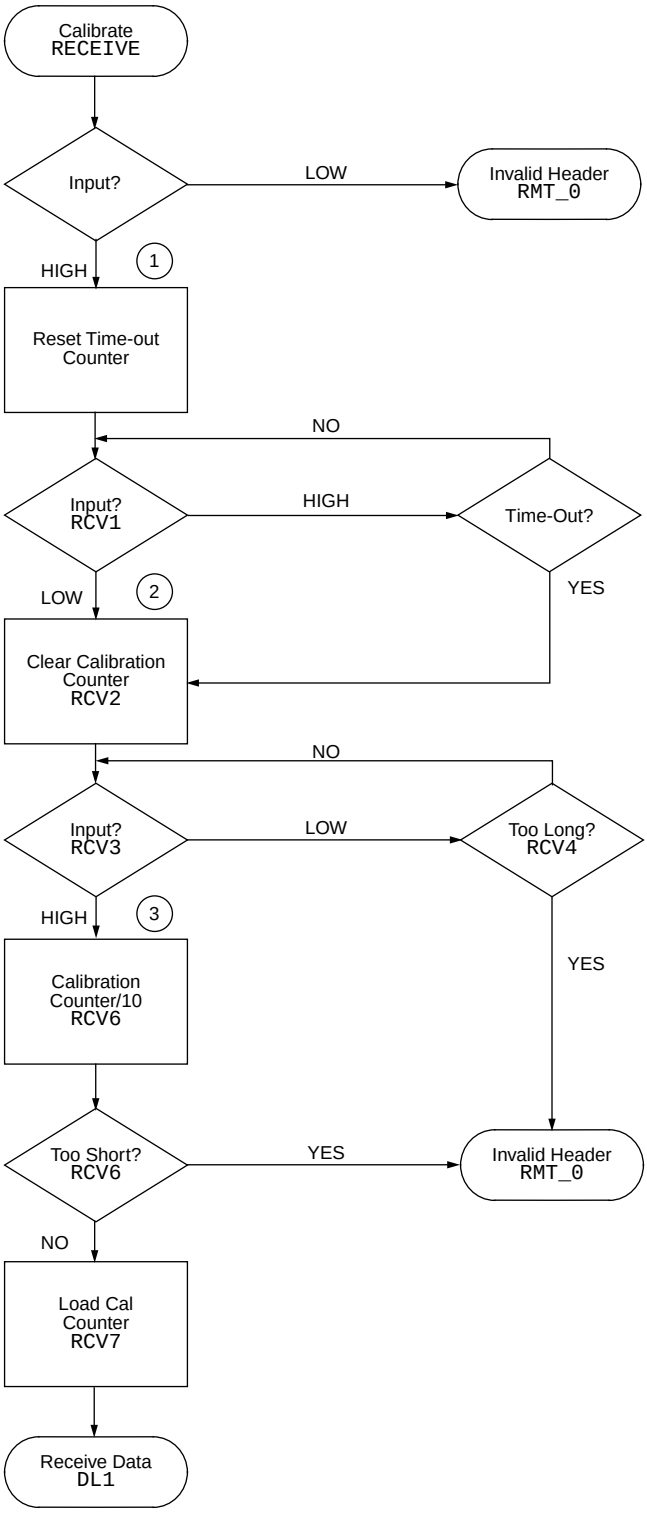
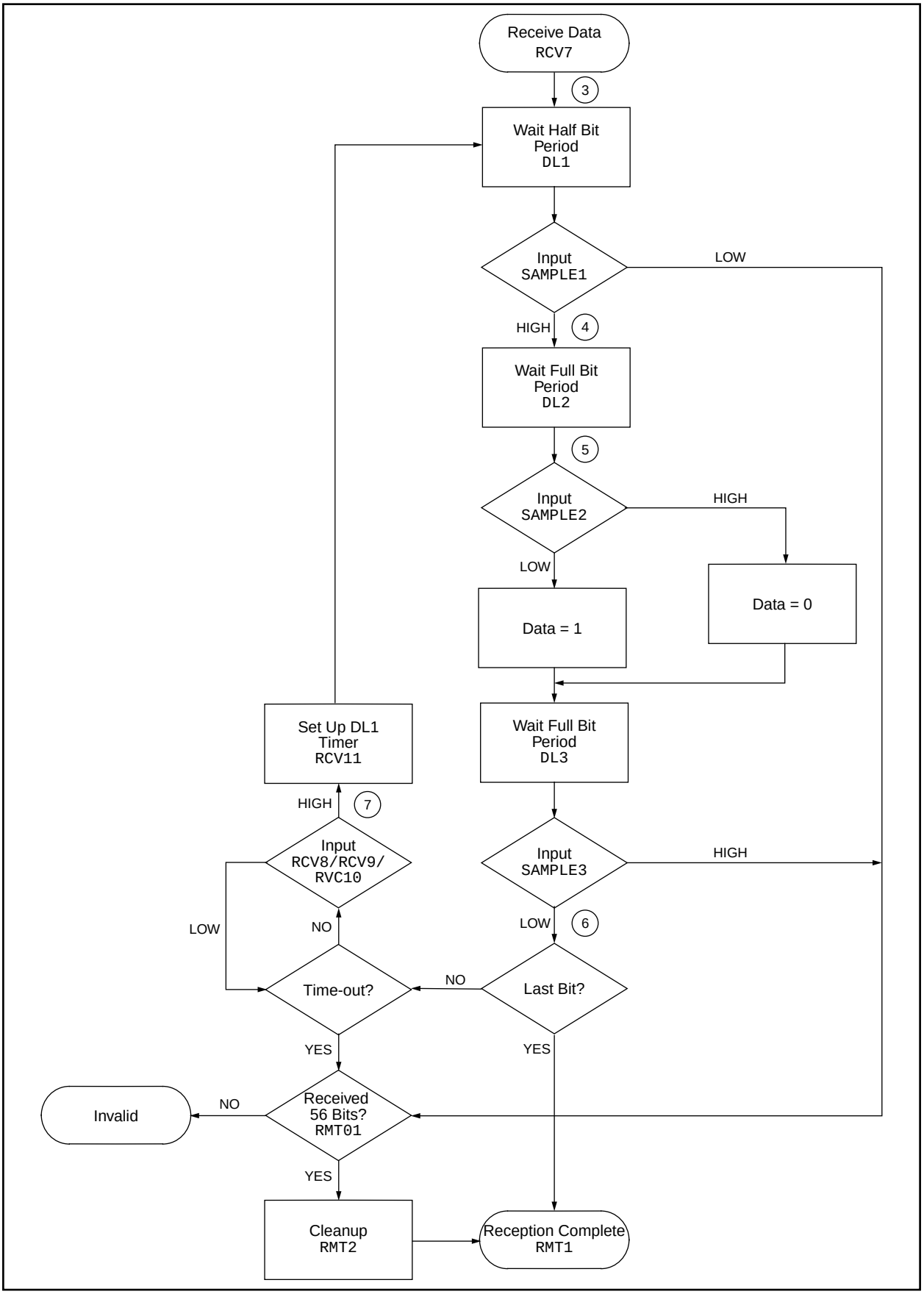


FIGURE 7: DATA RECEPTION FLOW CHART



Transmission 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 and is shown in Figure 8:

1. Check the serial number (28 bits) received against the stored encoder serial numbers (M_SERIAL).
2. Decrypt the transmission received using the system key (M_HOP).
3. Compare the eight least significant bits of the serial number to the 8 least significat bits of the discrimination value in the decrypted hopping portion of the transmission (M_DIS).
4. Get the first counter from the EEPROM (M_CNT) and check if the counter falls within the 512 count open window (M_CHECK0). If not get the second counter from EEPROM (M_CNT) and check whether the counter falls within the open window.
5. Check whether the received counter is the same as the stored counter (M_CHECK2). If so reset the output timer (M_TZERO)
6. If the transmission is a new one update the counters in EEPROM (M_UPDATE).
7. Set the appropriate function code on the outputs (M_OUTPUT) and reset the output timer (M_TZERO).
8. Return to the main routine (M_LOOP) and continue with 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. Twelve 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.

TABLE 6: HCS200 AND HCS300/301 DECRYPTED HOPPING CODE TRANSMISSION FORMAT

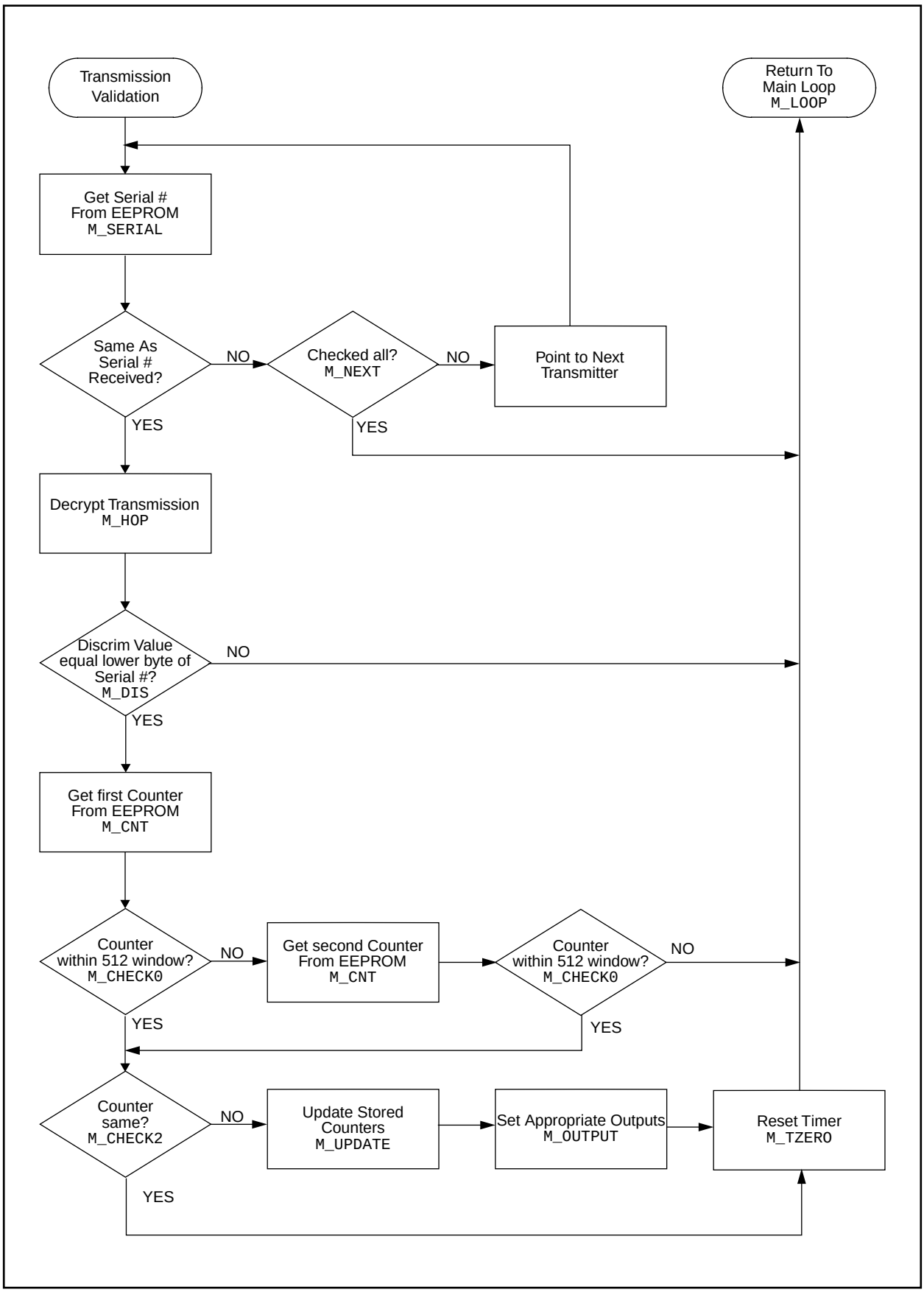
Function[1]	MSB	Encoder disc. bits	LSB	MSB	Synchronization counter	LSB
(4 bits)		(12 bits)			(16 bits)	

[1]The HCS200 has padding in S3 button position since no S3 button is present.

TABLE 7: HCS360/361 DECRYPTED HOPPING CODE TRANSMISSION FORMAT

Function	MSB	Encoder disc. bits	LSB	MSB	Synchronization counter	LSB
(4 bits)		(12 bits)			(16 bits)	

FIGURE 8: TRANSMISSION VALIDATION FLOWCHART



Synchronization Checking

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

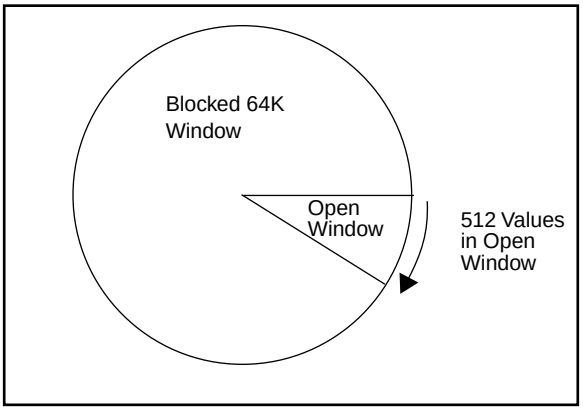
The transmitting encoder has a 16-bit synchronization counter, stored in EEPROM, which is incremented every time the encoder is activated. The synchronization counter value received is stored in the decoder's EEPROM every time a valid transmission is received from a particular encoder. When a following 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 is less than the current count value.

The simple decoder described in the application note has a single synchronization window. The window is 512 counts big. If the synchronization counter received is less than 512 counts above the last counter received the decoder updates the counters in EEPROM and activates the appropriate outputs. If the counter is greater that 512 above the stored counter the transmitter will have to be relearned. Modification to the synchronization window can be made in M_SUB.

The decoder stores two copies of the synchronization counter. If the power supply is interrupted during a counter update the synchronization counter can be corrupted. The decoder first checks received synchronization counter with the first counter. If the counter received is outside the open window the second counter is read and compared with the received counter. When the counters are updated in EEPROM they are written singly and both counters should never be corrupted at the same time.

The simplified synchronization scheme used in the decoder uses less code space than the conventional synchronization scheme at the expense of a lower level of security.

FIGURE 9: DECODER WINDOW OPERATION



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 on the current encoders.

The four function bits transmitted by the KEELOQ encoders correspond to the S2, S1, S0, and S3 inputs on the HCS300/301/360/361 encoders and S2, S1, S0, and S2 inputs on the HCS200.

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_OUTPUT).

Output Activation

The Microchip decoder has four momentary outputs namely S0, S1, S2, and S3. 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 is activated.

Decryption

After receiving a complete transmission the decoder decrypts the code hopping portion of the transmitted code. All of the encoders have the same encryption key.

The KEELOQ decryption algorithm is used to decrypt the 32-bit code hopping portion of KEELOQ transmissions. The decryption routine is called DECRYPT. A 32–bit Code Shift Register (CSR) contains the received code, and a 64-bit register contains the decryption key. In the simple decoder, all encoders have a common decryption key (KEY 0; KEY 7) which is securely stored in the program memory.

The block diagram (Figure 10) explains the operation during each iteration of the decryption algorithm. A nonlinear function (NLF – Table 8) 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 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 NLF 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:

- I4 = CSR3,6
- I3 = CSR3,1
- I2 = CSR2,3
- I1 = CSR1,0
- I0 = CSR0,0.

FIGURE 10: THE KEELOQ DECRYPTION ALGORITHM

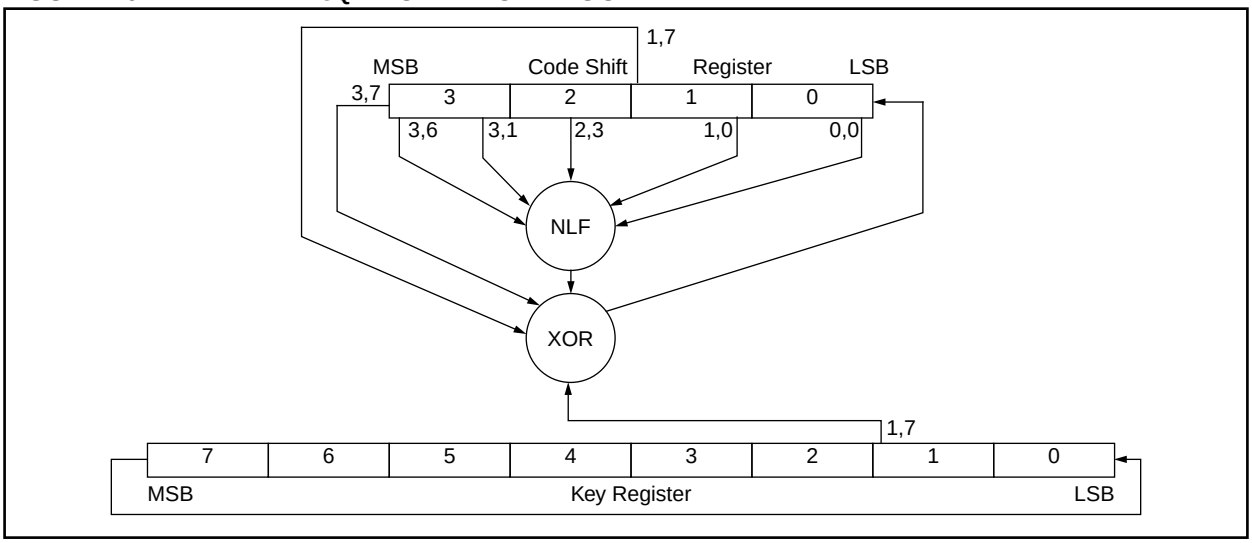


TABLE 8: NON-LINEAR FUNCTION OUTPUT

I_4	I_3	I_2	I_1	I_0	NLF
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	0
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	0
1	1	1	1	1	0

Learn

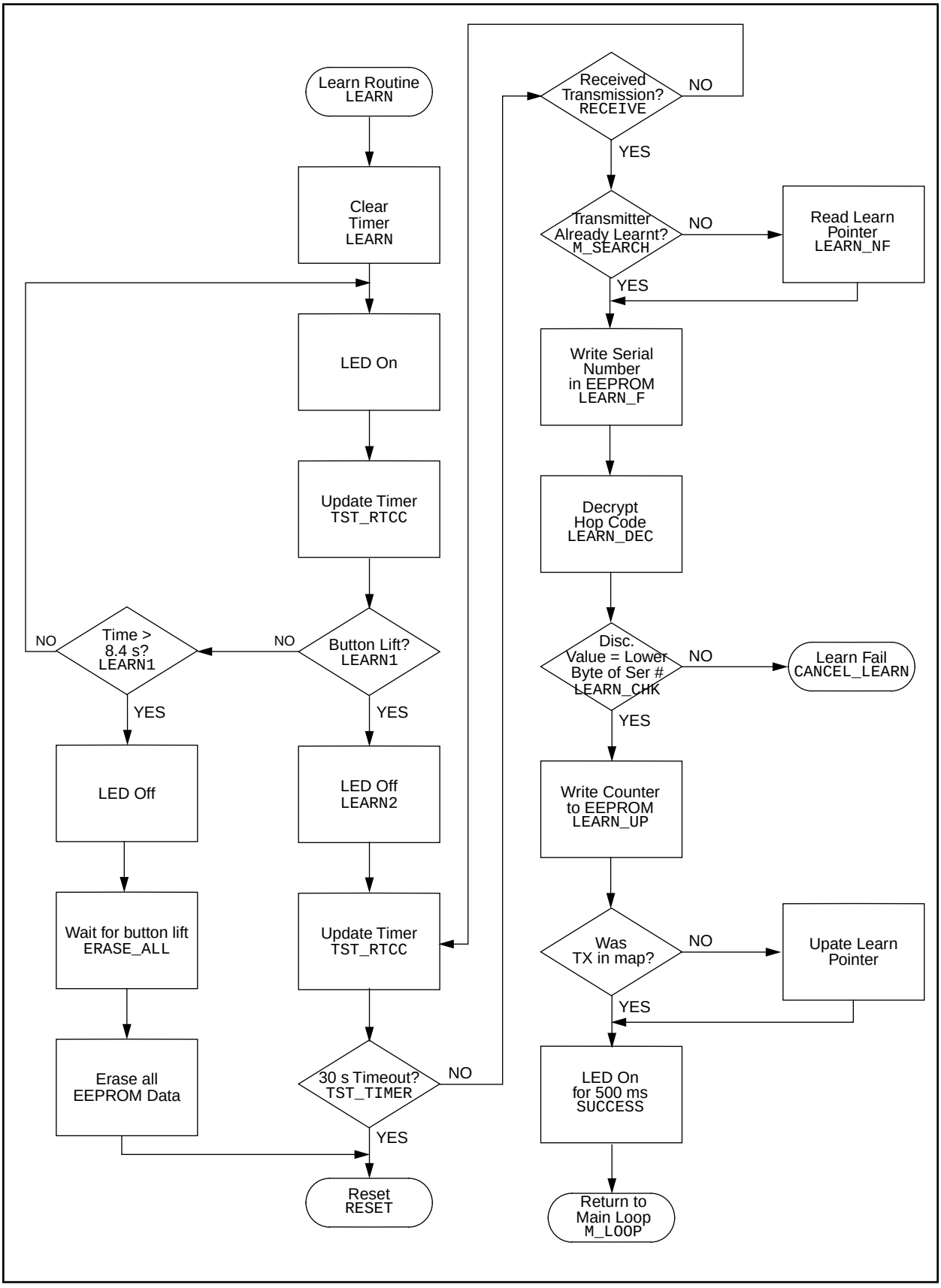
The decoder is able to learn up to 15 encoders. Internally a learn pointer is used by the decoder to keep track of the next internal memory position where a learn is to take place. The 15 encoder positions in EEPROM form a rotating buffer where the next transmitter to be written over is the transmitter at the tail of the buffer. The LEARN INIT input on the decoder 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 pointer is not incremented if the learn was unsuccessful.

Learn Program Flow

The simple decoder has a single transmission learn. This is because there is no need to generate a decryption key. On receiving a transmission from an encoder the decoder validates the encoder simply by checking that the discrimination value is identical to the serial number. The program flow during learn is shown in Figure 11 below. Learn in the simple decoder takes place as follows.

1. The decoder detects that the learn button has been pressed (M_LOOP) and enters learn mode.
2. The time-out counter is cleared (LEARN)and the LED is switched on until the button is released (LEARN1). If the learn button is activated for 8 seconds, all of the transmitters are erased and the LED is switched off (ERASE_ALL).
3. The transmitter waits till it receives a valid transmission (LEARN3). If no transmission is received within 30 seconds the decoder times out and exits learn (TST_TIMER).
4. Once the decoder has received a transmission it checks whether the transmitter has already been learnt (M_SEARCH). If so the user position is overwritten, otherwise the learn pointer is used as the learn position (LEARN_NF) and the serial number written to EEPROM.
5. The HOP code is decrypted (LEARN_DEC) and the least significant 8 bits of the serial number compared to the least significant 8 bits of the discrimination value (LRN_CHK).
6. If the discrimination bytes match those of the serial number the counter is written to the EEPROM (LEARN_UP) otherwise the decoder exits the learn routine (CANCEL_LEARN).
7. If the transmitter was being learnt to the position pointed to by the learn pointer the learn pointer is updated (LEARN_UP).
8. The LED is switched on for 500ms to show that the learn was successfully completed and the decoder returns to normal program flow (SUCCESS).

FIGURE 11: PROGRAM FLOW DURING LEARN



Operation of Learn

The following steps need to be followed by a user to learn an encoder onto the simple decoder. The steps are shown in Figure 21.

1. Press and release the learn button. The LEARN LED will be on while the button is pressed.
2. Activate the transmitter. The LED will turn on for 500 ms to indicate that the transmitter was successfully learnt.
3. To learn up to 15 transmitters, repeat steps 1-2. The 16th transmitter will overwrite the first transmitter that was learned.

Learn will be terminated if the eight least significant bits of the serial number do not match the eight least significant bits of the discrimination value.

Erasing all the transmitters is accomplished by pressing and holding the LEARN button for more than 8.4 seconds. The LED will turn off at this time indicating that the decoder is in erase all mode. The transmitters are erased when the learn button is released.

TIMER0 (RTCC) Multiplexing

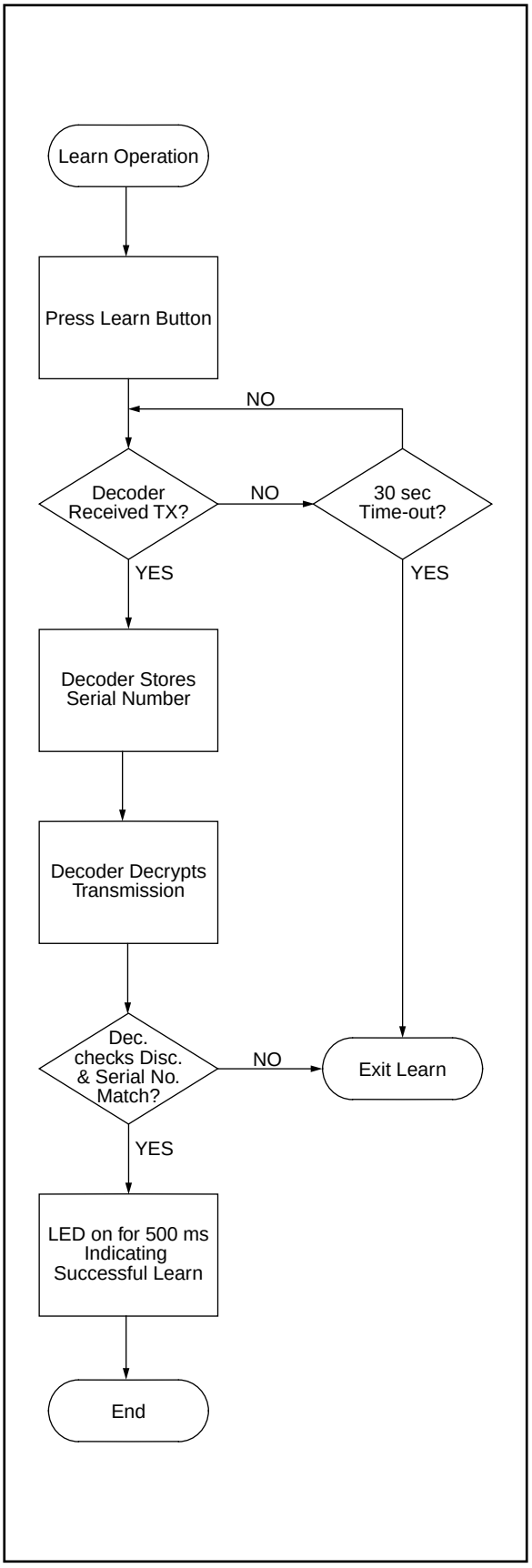
A time keeping scheme is needed to ensure that the system timing is not abandoned while receiving an incoming signal, during learn cycles and decryption. The system timing is used to allow periodic monitoring of sensors and pulsing outputs with a specific period.

TIMER0 is used to keep track of system time. TIMER0 is an 8-bit timer on the PIC16C54. On the 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 reception 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, two additional 8-bit counters are used, CNT_LW and CNT_HI, which extend the range of TIMER0 to 134 seconds. If the MSB of TIMER0 is set, the extended counter (CNT_LW and CNT_HI) is incremented and the function returns to the calling program.

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

FIGURE 12: LEARN OPERATION



EEPROM MEMORY MAP (16-BIT BYTES)

TABLE 9: EEPROM MEMORY MAP (16-BIT WORDS)

Address	Mnemonic	Address	Mnemonic
00	USER0	20	CNT7_0
01	Learn pointer	21	CNT7_1
02	USER1	22	SER7_0
03	USER2	23	SER7_1
04	CNT0_0	24	CNT8_0
05	CNT0_1	25	CNT8_1
06	SER0_0	26	SER8_0
07	SER0_1	27	SER8_1
08	CNT1_0	28	CNT9_0
09	CNT1_1	29	CNT9_1
0A	SER1_0	2A	SER9_0
0B	SER1_1	2B	SER9_1
0C	CNT2_0	2C	CNT10_0
0D	CNT2_1	2D	CNT10_1
0E	SER2_0	2E	SER10_0
0F	SER2_1	2F	SER10_1
10	CNT3_0	30	CNT11_0
11	CNT3_1	33	CNT11_1
12	SER3_0	32	SER11_0
13	SER3_1	33	SER11_1
14	CNT4_0	34	CNT12_0
15	CNT4_1	35	CNT12_1
16	SER4_0	36	SER12_0
17	SER4_1	37	SER12_1
18	CNT5_0	38	CNT13_0
19	CNT5_1	39	CNT13_1
1A	SER5_0	3A	SER13_0
1B	SER5_1	3B	SER13_1
1C	CNT6_0	3C	CNT14_0
1D	CNT6_1	3D	CNT14_1
1E	SER6_0	3E	SER14_0
1F	SER6_1	3F	SER14_1

Note: The number of users can be limited by changing ‘MAX_USERS’ defined at the top of the code.

RAM MEMORY MAP

TABLE 10: RAM MEMORY MAP

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.
0B	CNT0	Loop counters.
0C	CNT1	
OD	CNT2	
OE	CNT_HI	16-bit clock counter.
OF	CNT_LO	
10	CSR0	64-bit shift register
11	CSR1	
12	CSR2	
13	CSR3	
14	CSR4	
15	CSR5	
16	CSR6	
17	CSR7	
18	TMP1	
19	TMP2	
1A	REG0	Not Used
1B	REG1	Not Used
1C	KEY0	Least significant 32 bits of the key shift registor.
1D	KEY1	
1E	KEY2	
1F	KEY3	

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 Table 11 below.

TABLE 11: ALTERNATE NAMES AND FUNCTIONS

Address	Mnemonic	Also known as	Description
0A	MASK	OUTBYT	Mask used in decryption.
0A	TMP_CNT	TMP1	Counter used in the reception routine.
10	HOP1	CSR0	32-bit hop code register.
11	HOP2	CSR1	
12	HOP3	CSR2	
13	HOP4	CSR3	
17	SER_0	CSR7	28-bit serial number is stored here by the reception routine.
16	SER_1	CSR6	
15	SER_2	CSR5	
14	SER_3	CSR4	
13	FUNC	CSR3	Function code and user nibble of discrimination value.
12	CODE	CSR2	Discrimination value.
11	CNTR_HI	CSR1	16-bit received counter.
10	CNTR_LW	CSR0	
11	CSR8	TMP2	Most significant byte of the code shift register.
0D	KEY4	CNT2	Most significant 32 bits of the key shift register.
15	KEY5	CSR5	
16	KEY6	CSR6	
17	KEY7	CSR7	

DEVICE PINOUTS

The device used in the application note is a PIC16C54 PDIP or SOIC.

TABLE 12: DEVICE PINOUTS

PIN	PIC16C54 Function	Decoder Function	PIN	PIC16C54 Function	Decoder Function
1	Port A Bit 2	LEARN Input	18	Port A Bit 1	RF Input
2	Port A Bit 3	LEARN Indicator	17	Port A Bit 0	Not used
3	TIMER0	Connect to VDD	16	Osc In	RC osc (4 MHz)
4	$\overline{\text{MCLR}}$	Brown out detect	15	Osc Out	
5	GND	Ground	14	VDD	+5V supply
6	Port B Bit 0	S0	13	Port B Bit 7	Not Used
7	Port B Bit 1	S1	12	Port B Bit 6	CS (93C46, pin 1))
8	Port B Bit 2	S2	11	Port B Bit 5	CLK (93C46, pin 2)
9	Port B Bit 3	S3	10	Port B Bit 4	DIO (93C46, pin 3 & 4)

TIMING PARAMETERS

TABLE 13: 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 failure LED on duration	1	s

SOURCE CODE LISTING

A diskette is supplied containing source code for the Microchip decoder in the file simdec**.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_TIMER. The PIC16C54 Microcontroller should run at 4 MHz.

TABLE 14: LIST OF IMPORTANT FUNCTIONS

Function Name	Description
DECRYPT	Decryption routine for Hop Code.
EEREAD	The data in the EEPROM at ADDRESS is read to TMP1 and TMP2 (Note).
EEWRITE	The data in TMP1, and TMP2 is written to the EEPROM at ADDRESS (Note).
M_DIS	Check discrimination value.
M_CNT	Check synchronization (counter) values.
RECEIVE	Start of the RF reception routine.
LEARN	Learn mode.
TST_RTCC	Check Timer0 and update CNT_LW and CNT_HI.
TST_TIMER	Check CNT_LW and CNT_HI and do whatever real-time tasks that are required.

Note: TMP1, TMP2 and ADDRESS are user defined registers.

APPENDIX A: SCHEMATIC DIAGRAMS

FIGURE 13: SCHEMATIC DIAGRAM OF MICROCHIP KEELOQ DECODER

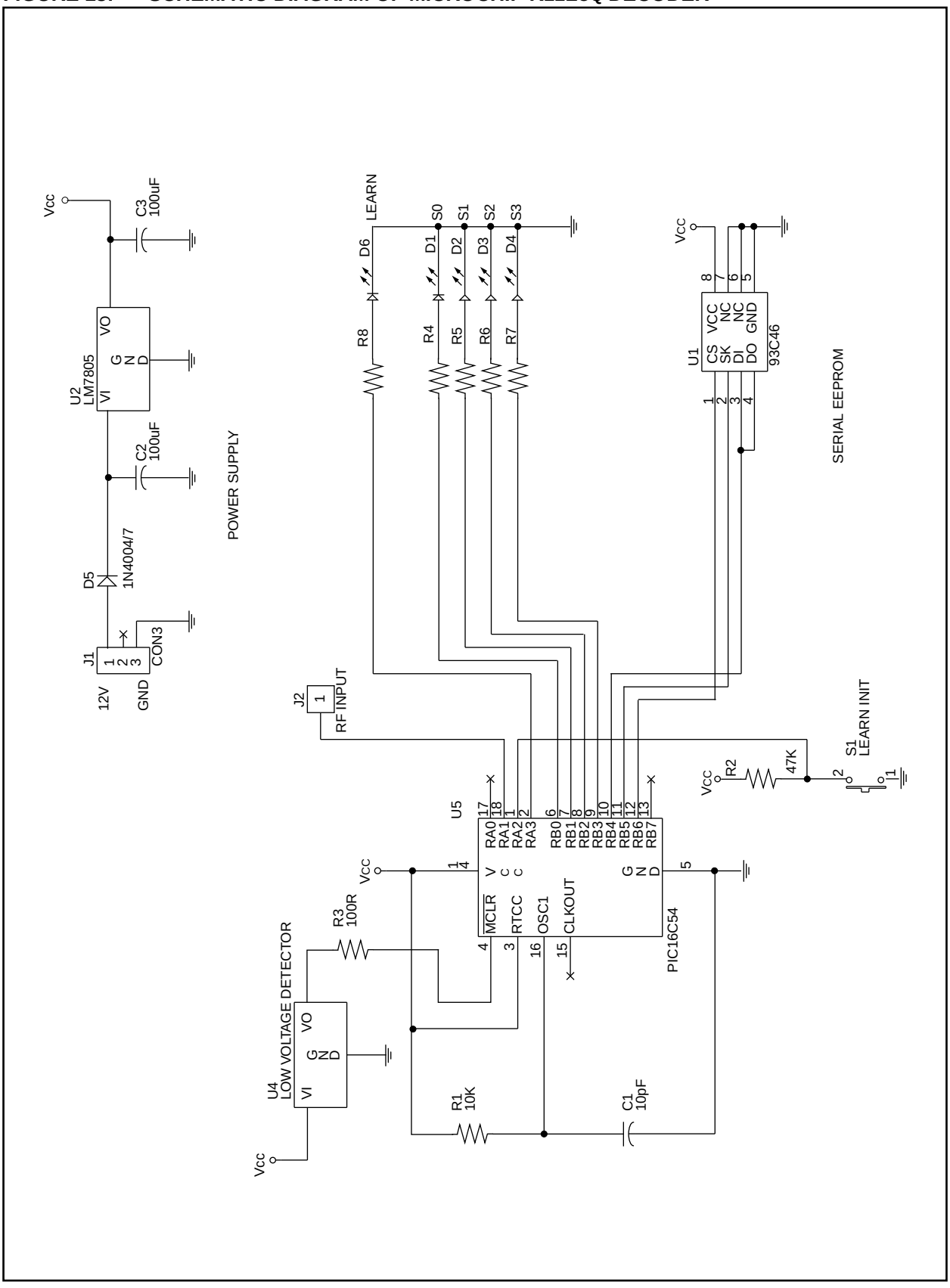


FIGURE 14: TYPICAL GARAGE DOOR OPENER SCHEMATIC

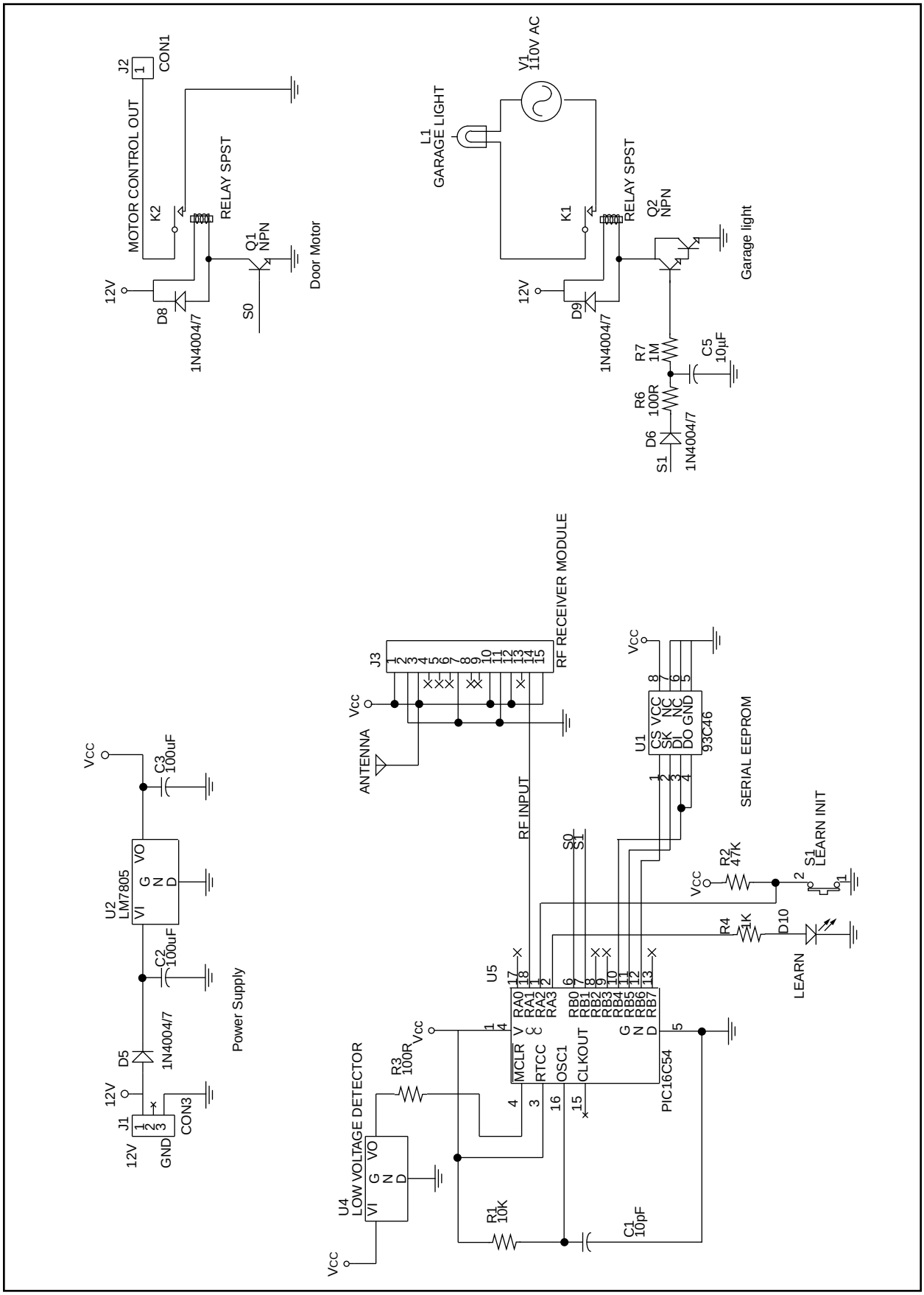
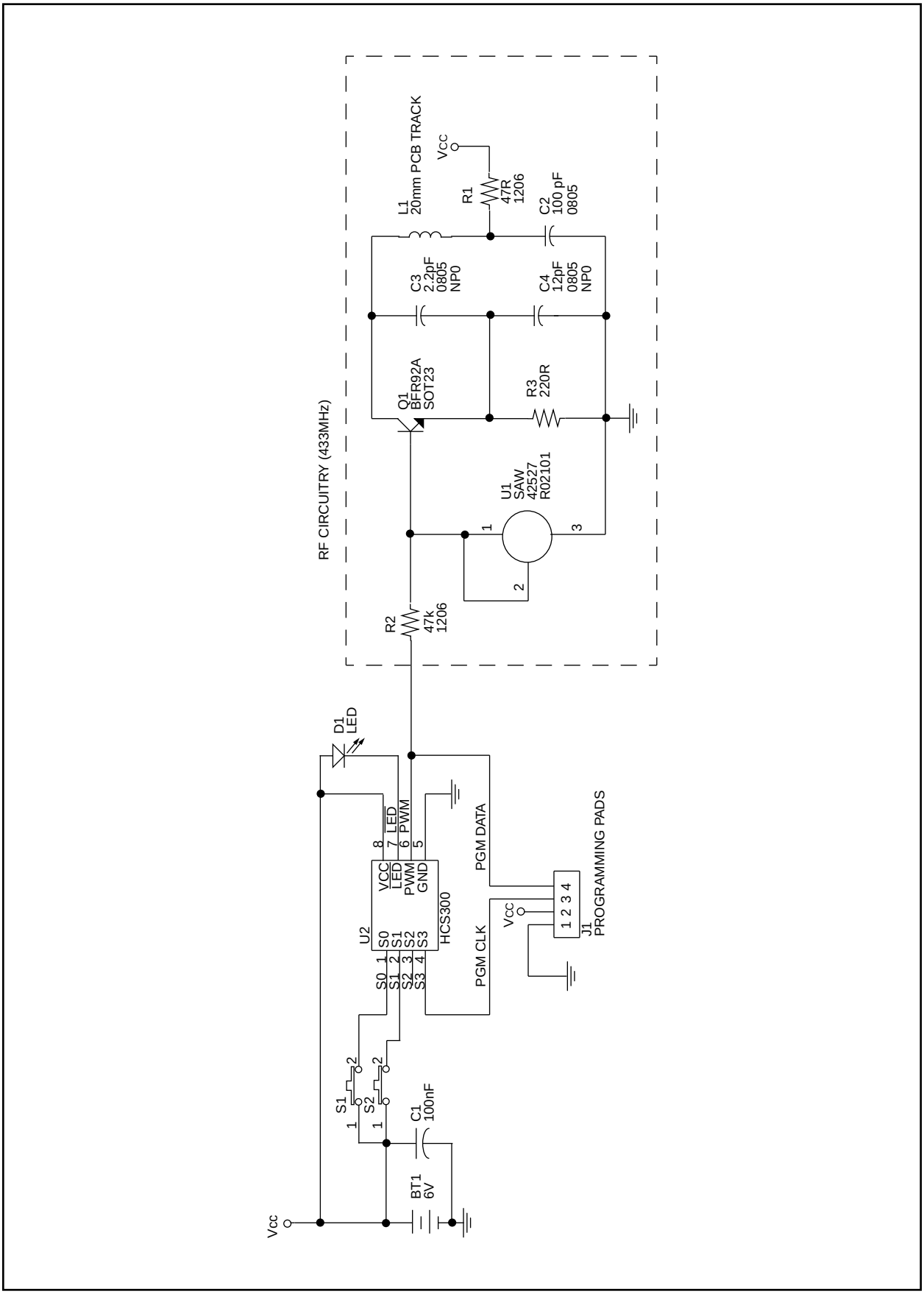


FIGURE 15: HCS200/300/301/360/361 TRANSMITTER DESIGN



NOTES:

NOTES:

NOTES:

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: 708 285-0071 Fax: 708 285-0075

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.
Suite 150
Two Prestige Place
Miamisburg, OH 45342
Tel: 513 291-1654 Fax: 513 291-9175

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 Technology
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
No. 6, Legacy, Convent Road
Bangalore 560 025 India
Tel: 91 80 526 3148 Fax: 91 80 559 9840

Korea
Microchip Technology
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
Unit 406 of Shanghai Golden Bridge Bldg.
2077 Yan'an Road West, Hongiao District
Shanghai, Peoples Republic of China
Tel: 86 21 6275 5700
Fax: 011 86 21 6275 5060

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

Taiwan, R.O.C
Microchip Technology
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 850303 Fax: 44 1628 850178

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 Muenchen, Germany
Tel: 49 89 627 144 0 Fax: 49 89 627 144 44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleone Pas Taurus 1
Viale Colleoni 1
20041 Agrate Brianza
Milan Italy
Tel: 39 39 6899939 Fax: 39 39 689 9883

JAPAN

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

11/7/96

MICROCHIP

Printed on recycled paper.