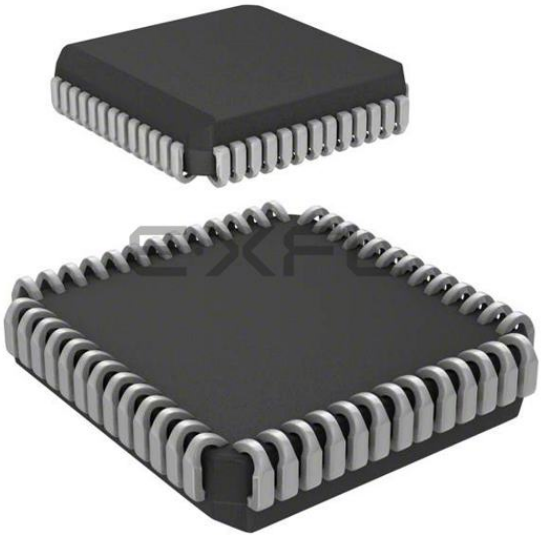

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	HC11
Core Size	8-Bit
Speed	2MHz
Connectivity	SCI, SPI
Peripherals	POR, WDT
Number of I/O	38
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LCC (J-Lead)
Supplier Device Package	52-PLCC (19.1x19.1)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc11e1cfn2

Table of Contents

Section 1. General Description

Section 2. Operating Modes and On-Chip Memory

Section 8. Serial Peripheral Interface (SPI)

Section 9. Timing System

Table of Contents

ADPU — A/D Power-Up Bit
0 = A/D powered down
1 = A/D powered up

CSEL — Clock Select Bit
0 = A/D and EEPROM use system E clock.
1 = A/D and EEPROM use internal RC clock.

IRQE — Configure $\overline{\text{IRQ}}$ for Edge-Sensitive Only Operation
Refer to **Section 5. Resets and Interrupts**.

DLY — Enable Oscillator Startup Delay Bit
0 = The oscillator startup delay coming out of stop is bypassed and the MCU resumes processing within about four bus cycles.
1 = A delay of approximately 4000 E-clock cycles is imposed as the MCU is started up from the stop power-saving mode. This delay allows the crystal oscillator to stabilize.

CME — Clock Monitor Enable Bit
Refer to **Section 5. Resets and Interrupts**.

Bit 2 — Not implemented
Always reads 0

CR[1:0] — COP Timer Rate Select Bits
Refer to **Section 5. Resets and Interrupts** and **Section 9. Timing System**.

3.4 Conversion Process

The A/D conversion sequence begins one E-clock cycle after a write to the A/D control/status register, ADCTL. The bits in ADCTL select the channel and the mode of conversion.

An input voltage equal to V_{RL} converts to $00 and an input voltage equal to V_{RH} converts to $FF (full scale), with no overflow indication. For ratiometric conversions of this type, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}.

3.5 Channel Assignments

The multiplexer allows the A/D converter to select one of 16 analog signals. Eight of these channels correspond to port E input lines to the MCU, four of the channels are internal reference points or test functions, and four channels are reserved. Refer to **Table 3-1**.

system is configured to perform a conversion on each of four channels where each result register corresponds to one channel.

NOTE: *When the multiple-channel continuous scan mode is used, extra care is needed in the design of circuitry driving the A/D inputs. The charge on the capacitive DAC array before the sample time is related to the voltage on the previously converted channel. A charge share situation exists between the internal DAC capacitance and the external circuit capacitance. Although the amount of charge involved is small, the rate at which it is repeated is every 64 μs for an E clock of 2 MHz. The RC charging rate of the external circuit must be balanced against this charge sharing effect to avoid errors in accuracy. Refer to M68HC11 Reference Manual, Motorola document order number M68HC11RM/AD, for further information.*

CD:CA — Channel Selects D:A Bits

Refer to **Table 3-2**. When a multiple channel mode is selected (MULT = 1), the two least significant channel select bits (CB and CA) have no meaning and the CD and CC bits specify which group of four channels is to be converted.

Table 3-2. A/D Converter Channel Selection

Channel Select Control Bits CD:CC:CB:CA	Channel Signal	Result in ADRx if MULT = 1
0000	AN0	ADR1
0001	AN1	ADR2
0010	AN2	ADR3
0011	AN3	ADR4
0100	AN4	ADR1
0101	AN5	ADR2
0110	AN6	ADR3
0111	AN7	ADR4
10XX	Reserved	—
1100	V_{RH}(1)	ADR1
1101	V_{RL}(1)	ADR2
1110	(V_{RH})/2(1)	ADR3
1111	Reserved(1)	ADR4

1. Used for factory testing

Table 5-3. Highest Priority Interrupt Selection

PSEL[3:0]	Interrupt Source Promoted
0 0 0 0	Timer overflow
0 0 0 1	Pulse accumulator overflow
0 0 1 0	Pulse accumulator input edge
0 0 1 1	SPI serial transfer complete
0 1 0 0	SCI serial system
0 1 0 1	Reserved (default to $\overline{\text{IRQ}}$)
0 1 1 0	$\overline{\text{IRQ}}$ (external pin or parallel I/O)
0 1 1 1	Real-time interrupt
1 0 0 0	Timer input capture 1
1 0 0 1	Timer input capture 2
1 0 1 0	Timer input capture 3
1 0 1 1	Timer output compare 1
1 1 0 0	Timer output compare 2
1 1 0 1	Timer output compare 3
1 1 1 0	Timer output compare 4
1 1 1 1	Timer input capture 4/output compare 5

5.5 Interrupts

The MCU has 18 interrupt vectors that support 22 interrupt sources. The 15 maskable interrupts are generated by on-chip peripheral systems. These interrupts are recognized when the global interrupt mask bit (I) in the condition code register (CCR) is clear. The three non-maskable interrupt sources are illegal opcode trap, software interrupt, and $\overline{\text{XIRQ}}$ pin. Refer to **Table 5-4**, which shows the interrupt sources and vector assignments for each source.

For some interrupt sources, such as the SCI interrupts, the flags are automatically cleared during the normal course of responding to the interrupt requests. For example, the RDRF flag in the SCI system is cleared by the automatic clearing mechanism consisting of a read of the SCI status register while RDRF is set, followed by a read of the SCI data register. The normal response to an RDRF interrupt request would be to read the SCI status register to check for receive errors, then to read the received data from the SCI data register. These steps satisfy the automatic clearing mechanism without requiring special instructions.

Section 7. Serial Communications Interface (SCI)

7.1 Introduction

The serial communications interface (SCI) is a universal asynchronous receiver transmitter (UART), one of two independent serial input/output (I/O) subsystems in the M68HC11 E series of microcontrollers. It has a standard non-return-to-zero (NRZ) format (one start bit , eight or nine data bits, and one stop bit). Several baud rates are available. The SCI transmitter and receiver are independent, but use the same data format and bit rate.

All members of the E series contain the same SCI, with one exception. The SCI system in the MC68HC11E20 and MC68HC711E20 MCUs have an enhanced SCI baud rate generator. A divide-by-39 stage has been added that is enabled by an extra bit in the BAUD register. This increases the available SCI baud rate selections. Refer to **Figure 7-8** and **7.7.5 Baud Rate Register**.

7.2 Data Format

The serial data format requires these conditions:

1. An idle line in the high state before transmission or reception of a message
2. A start bit, logic 0, transmitted or received, that indicates the start of each character
3. Data that is transmitted and received least significant bit (LSB) first
4. A stop bit, logic 1, used to indicate the end of a frame. A frame consists of a start bit, a character of eight or nine data bits, and a stop bit.
5. A break, defined as the transmission or reception of a logic 0 for some multiple number of frames

Selection of the word length is controlled by the M bit of SCI control register (SCCR1).

7.3 Transmit Operation

The SCI transmitter includes a parallel transmit data register (SCDR) and a serial shift register. The contents of the serial shift register can be written only through the SCDR. This double buffered operation allows a character to be shifted out serially while another character is waiting in the SCDR to be transferred into the

these flags is automatic. Functions that are normally performed in response to the status flags also satisfy the conditions of the clearing sequence.

TDRE and TC flags are normally set when the transmitter is first enabled (TE set to 1). The TDRE flag indicates there is room in the transmit queue to store another data character in the TDR. The TIE bit is the local interrupt mask for TDRE. When TIE is 0, TDRE must be polled. When TIE and TDRE are 1, an interrupt is requested.

The TC flag indicates the transmitter has completed the queue. The TCIE bit is the local interrupt mask for TC. When TCIE is 0, TC must be polled. When TCIE is 1 and TC is 1, an interrupt is requested.

Writing a 0 to TE requests that the transmitter stop when it can. The transmitter completes any transmission in progress before actually shutting down. Only an MCU reset can cause the transmitter to stop and shut down immediately. If TE is written to 0 when the transmitter is already idle, the pin reverts to its general-purpose I/O function (synchronized to the bit-rate clock). If anything is being transmitted when TE is written to 0, that character is completed before the pin reverts to general-purpose I/O, but any other characters waiting in the transmit queue are lost. The TC and TDRE flags are set at the completion of this last character, even though TE has been disabled.

7.9 Receiver Flags

The SCI receiver has five status flags, three of which can generate interrupt requests. The status flags are set by the SCI logic in response to specific conditions in the receiver. These flags can be read (polled) at any time by software. Refer to **Figure 7-10**, which shows SCI interrupt arbitration.

When an overrun takes place, the new character is lost, and the character that was in its way in the parallel RDR is undisturbed. RDRF is set when a character has been received and transferred into the parallel RDR. The OR flag is set instead of RDRF if overrun occurs. A new character is ready to be transferred into RDR before a previous character is read from RDR.

The NF and FE flags provide additional information about the character in the RDR, but do not generate interrupt requests.

The last receiver status flag and interrupt source come from the IDLE flag. The RxD line is idle if it has constantly been at logic 1 for a full character time. The IDLE flag is set only after the RxD line has been busy and becomes idle, which prevents repeated interrupts for the whole time RxD remains idle.

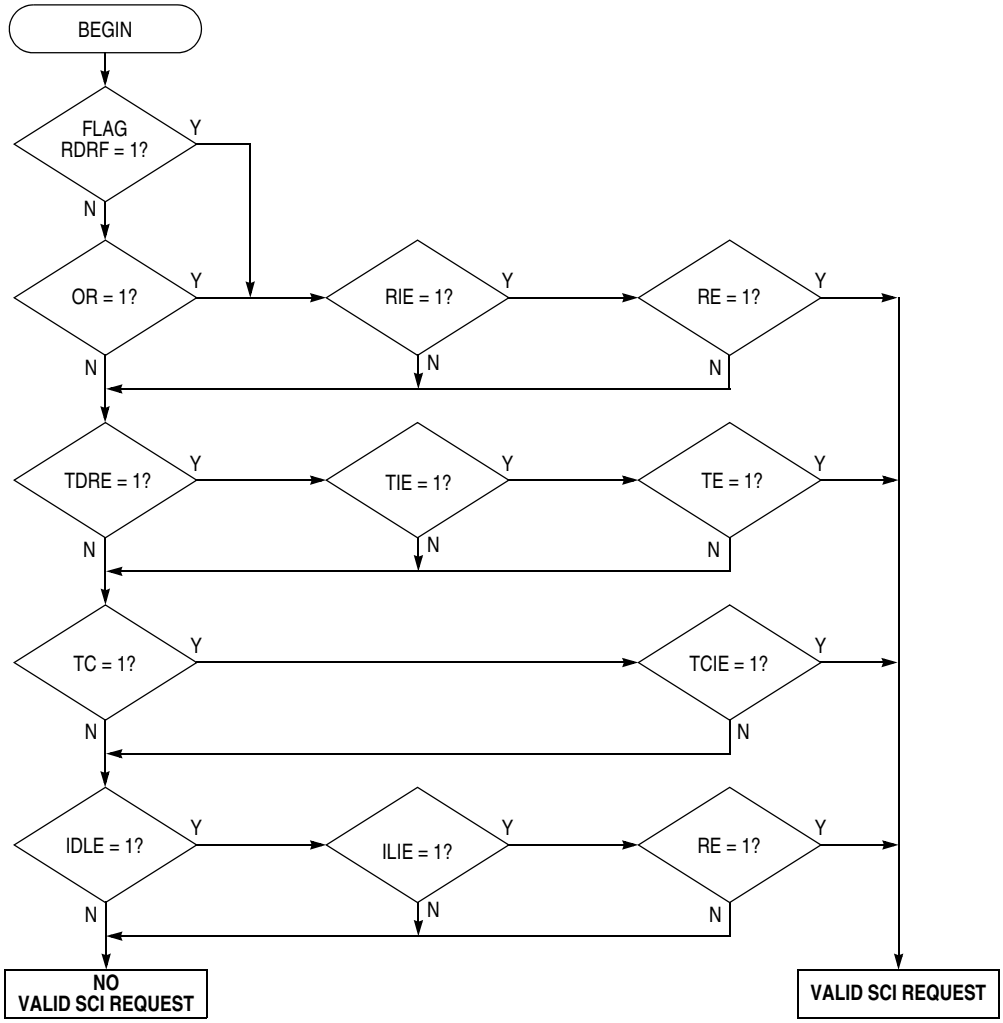

Figure 7-10. Interrupt Source Resolution Within SCI

9.3.3 Timer Input Capture 4/Output Compare 5 Register

Use TI4/O5 as either an input capture register or an output compare register, depending on the function chosen for the PA3 pin. To enable it as an input capture pin, set the I4/O5 bit in the pulse accumulator control register (PACTL) to logic level 1. To use it as an output compare register, set the I4/O5 bit to a logic level 0. Refer to **9.7 Pulse Accumulator**.

Register name: Timer Input Capture 4/Output Compare 5 (High) Address: $101E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Register name: Timer Input Capture 4/Output Compare 5 (Low) Address: $101F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	1	1	1	1	1	1	1	1

Figure 9-7. Timer Input Capture 4/Output Compare 5 Register Pair (TI4/O5)

9.4 Output Compare

Use the output compare (OC) function to program an action to occur at a specific time — when the 16-bit counter reaches a specified value. For each of the five output compare functions, there is a separate 16-bit compare register and a dedicated 16-bit comparator. The value in the compare register is compared to the value of the free-running counter on every bus cycle. When the compare register matches the counter value, an output compare status flag is set. The flag can be used to initiate the automatic actions for that output compare function.

To produce a pulse of a specific duration, write a value to the output compare register that represents the time the leading edge of the pulse is to occur. The output compare circuit is configured to set the appropriate output either high or low, depending on the polarity of the pulse being produced. After a match occurs, the output compare register is reprogrammed to change the output pin back to its inactive level at the next match. A value representing the width of the pulse is added to the original value, and then written to the output compare register. Because the pin state changes occur at specific values of the free-running counter, the pulse width can be controlled accurately at the resolution of the free-running counter, independent of software latencies. To generate an output signal of a specific frequency and duty cycle, repeat this pulse-generating procedure.

The five 16-bit read/write output compare registers are: TOC1, TOC2, TOC3, and TOC4, and the TI4/O5. TI4/O5 functions under software control as either IC4 or OC5. Each of the OC registers is set to $FFFF on reset. A value written to an OC

Section 10. Electrical Characteristics

10.1 Introduction

This section contains electrical specifications for the M68HC11 E-series devices.

10.2 Maximum Ratings for Standard and Extended Voltage Devices

Maximum ratings are the extreme limits to which the MCU can be exposed without permanently damaging it.

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to **10.5 DC Electrical Characteristics**, **10.6 Supply Currents and Power Dissipation**, **10.7 MC68L11E9/E20 DC Electrical Characteristics**, and **10.8 MC68L11E9/E20 Supply Currents and Power Dissipation** for guaranteed operating conditions.*

Rating	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to +7.0	V
Input voltage	V_{In}	–0.3 to +7.0	V
Current drain per pin(1) excluding V_{DD}, V_{SS}, AV_{DD}, V_{RH}, V_{RL}, and $\overline{XIRQ}/V_{PPE}$	I_D	25	mA
Storage temperature	T_{STG}	–55 to +150	°C

1. One pin at a time, observing maximum power dissipation limits

NOTE: *This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{In} and V_{Out} be constrained to the range $V_{SS} \le (V_{In}$ or $V_{Out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD}).*

Electrical Characteristics

Description	CONFIG	Temperature	Frequency	MC Order Number
52-pin windowed ceramic leaded chip carrier (CLCC)				
EPROM	$0F	–40°C to +85°C	2 MHz	MC68HC711E9CFS2
			3 MHz	MC68HC711E9CFS3
		–40°C to +105°C	2 MHz	MC68HC711E9VFS2
		–40°C to +125°C	2 MHz	MC68HC711E9VFS2
20 Kbytes EPROM	$0F	0°C o +70°°C	3 MHz	MC68HC711E20FS3
		–40°C to +85°C	2 MHz	MC68HC711E20CFS2
			3 MHz	MC68HC711E20CFS3
		–40°C to +105°C	2 MHz	MC68HC711E20VFS2
		–40°C to +125°C	2 MHz	MC68HC711E20MFS2
48-pin dual in-line package (DIP) — MC68HC811E2 only				
No ROM, 2 Kbytes EEPROM	$FF	0°C to +70°°C	2 MHz	MC68HC811E2P2
		–40°C to +85°C	2 MHz	MC68HC811E2CP2
		–40°C to +105°C	2 MHz	MC68HC811E2VP2
		–40°C to +125°C	2 MHz	MC68HC811E2MP2
56-pin dual in-line package with 0.70-inch lead spacing (SDIP)				
BUFFALO ROM	$0F	–40°C to +85°C	2 MHz	MC68HC11E9BCB2
			3 MHz	MC68HC11E9BCB3
No ROM	$0D	–40°C to +85°C	2 MHz	MC68HC11E1CB2
			3 MHz	MC68HC11E1CB3
		–40°C to +105°C	2 MHz	MC68HC11E1VB2
		–40°C to +125°C	2 MHz	MC68HC11E1MB2
No ROM, no EEPROM	$0C	–40°C to +85°C	2 MHz	MC68HC11E0CB2
			3 MHz	MC68HC11E0CB3
		–40°C to +105°C	2 MHz	MC68HC11E0VB2
		–40°C to +125°C	2 MHz	MC68HC11E0MB2

Bootstrap mode is useful both at the component level and after the MCU has been embedded into a finished user system.

At the component level, Motorola uses bootstrap mode to control a monitored burn-in program for the on-chip electrically erasable programmable read-only memory (EEPROM). Units to be tested are loaded into special circuit boards that each hold many MCUS. These boards are then placed in burn-in ovens. Driver boards outside the ovens download an EEPROM exercise and diagnostic program to all MCUs in parallel. The MCUs under test independently exercise their internal EEPROM and monitor programming and erase operations. This technique could be utilized by an end user to load program information into the EPROM or EEPROM of an M68HC11 before it is installed into an end product. As in the burn-in setup, many M68HC11s can be gang programmed in parallel. This technique can also be used to program the EPROM of finished products after final assembly.

Motorola also uses bootstrap mode for programming target devices on the M68HC11 evaluation modules (EVM). Because bootstrap mode is a privileged mode like special test, the EEPROM-based configuration register (CONFIG) can be programmed using bootstrap mode on the EVM.

The greatest benefits from bootstrap mode are realized by designing the finished system so that bootstrap mode can be used after final assembly. The finished system need not be a single-chip mode application for the bootstrap mode to be useful because the expansion bus can be enabled after resetting the MCU in bootstrap mode. Allowing this capability requires almost no hardware or design cost and the addition of this capability is invisible in the end product until it is needed.

The ability to control the embedded processor through downloaded programs is achieved without the disassembly and chip-swapping usually associated with such control. This mode provides an easy way to load non-volatile memories such as EEPROM with calibration tables or to program the application firmware into a one-time programmable (OTP) MCU after final assembly.

Another powerful use of bootstrap mode in a finished assembly is for final test. Short programs can be downloaded to check parts of the

M68HC11 Family members which have 256 bytes of RAM, the download length is fixed at exactly 256 bytes plus the leading $FF character.

The intercharacter delay counter is started [8] by loading the delay constant from TOC1 into the X index register. The 19-E-cycle wait loop is executed repeatedly until either a character is received [9] or the allowed intercharacter delay time expires [10]. For 7812 baud, the delay constant is 10,241 E cycles (539 x 19 E cycles per loop). Four character times at 7812 baud is 10,240 E cycles (baud prescale of 4 x baud divider of 4 x 16 internal SCI clocks/bit time x 10 bit times/character x 4 character times). The delay from reset to the initial $FF character is not critical since the delay counter is not started until after the first character ($FF) is received.

To terminate the bootloading sequence and jump to the start of RAM without downloading any data to the on-chip RAM, simply send $FF and nothing else. This feature is similar to the jump to EEPROM at [4] except the $FF causes a jump to the start of RAM. This procedure requires that the RAM has been loaded with a valid program since it would make no sense to jump to a location in uninitialized memory.

After receiving a character, the downloaded byte is stored in RAM [11]. The data is transmitted back to the host [12] as an indication that the download is progressing normally. At [13], the RAM pointer is incremented to the next RAM address. If the RAM pointer has not passed the end of RAM, the main download loop (from [7] to [14]) is repeated.

When all data has been downloaded, the bootloader goes to [16] because of an intercharacter delay timeout [10] or because the entire 512-byte RAM has been filled [15]. At [16], the X and Y index registers are set up for calling the PROGRAM utility routine, which saves the user from having to do this in a downloaded program. The PROGRAM utility is fully explained in **EPROM Programming Utility**. The final step of the bootloader program is to jump to the start of RAM [17], which starts the user's downloaded program.

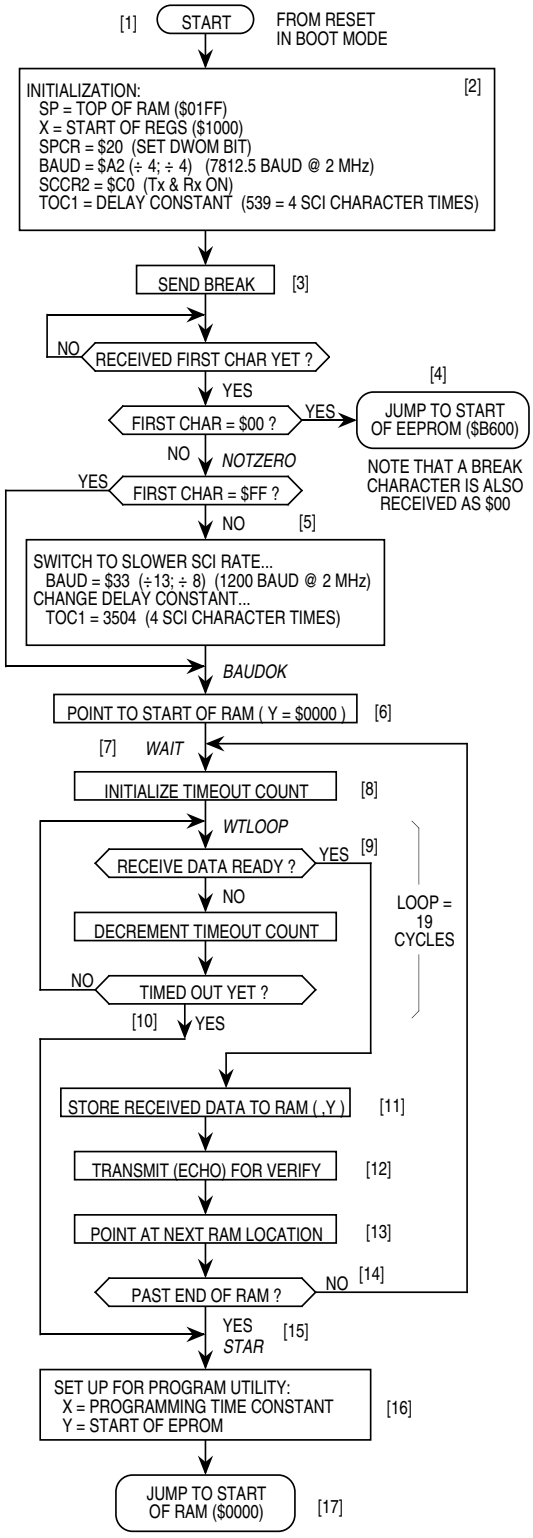

Figure 3. MC68HC711E9 Bootloader Flowchart

Modifications

This example programmed version 3.4 of the BUFFALO monitor into the EPROM of an MC68HC711E9; the changes to the BASIC program to download some other program are minor.

The necessary changes are:

1. In line 30, the length of the program to be downloaded must be assigned to the variable CODESIZE%.
2. Also in line 30, the starting address of the program is assigned to the variable ADRSTART.
3. In line 9570, the start address of the program is stored in the third and fourth items in that DATA statement in hexadecimal.
4. If any changes are made to the number of bytes in the boot code in the DATA statements in lines 9500–9580, then the new count must be set in the variable "BOOTCOUNT" in line 25.

Operation

Configure the EVBU for boot mode operation by putting a jumper at J3. Ensure that the trace command jumper at J7 is not installed because this would connect the 12-V programming voltage to the OC5 output of the MCU.

Connect the EVBU to its dc power supply. When it is time to program the MCU EPROM, turn on the 12-volt programming power supply to the new circuitry in the wire-wrap area.

Connect the EVBU serial port to the appropriate serial port on the host system. For the Macintosh, this is the modem port with a modem cable. For the MS-DOS® computer, it is connected to COM1 with a straight through or modem cable. Power up the host system and start the BASIC program. If the program has not been compiled, this is accomplished from within the appropriate BASIC compiler or interpreter. Power up the EVBU.

Answer the prompt for filename with either a [RETURN] to accept the default shown or by typing in a new filename and pressing [RETURN].

® MS-DOS is a registered trademark of Microsoft Corporation in the United States and other countries.

The port pins that are used for EPROM data I/O lines may be inputs or outputs, depending on the pin that is emulating the EPROM output enable pin ($\overline{OE}$). To make these data pins appear as high-impedance inputs as they would on a non-EPROM part in reset, connect the $\overline{PB7}$/($\overline{OE}$) pin to a pullup resistor.

Bootloading a Program to Perform a ROM Checksum

The bootloader ROM must be turned off before performing the checksum program. To remove the boot ROM from the memory map, clear the RBOOT bit in the HPRIO register. This is normally a write-protected bit that is 0, but in bootstrap mode it is reset to 1 and can be written. If the boot ROM is not disabled, the checksum routine will read the contents of the boot ROM rather than the user's mask ROM or EPROM at the same addresses.

Inherent Delays Caused by Double Buffering of SCI Data

This problem is troublesome in cases where one MCU is bootloading to another MCU.

Because of transmitter double buffering, there may be one character in the serial shifter as a new character is written into the transmit data register. In cases such as downloading in which this 2-character pipeline is kept full, a 2-character time delay occurs between when a character is written to the transmit data register and when that character finishes transmitting. A little more than one more character time delay occurs between the target MCU receiving the character and echoing it back. If the master MCU waits for the echo of each downloaded character before sending the next one, the download process takes about twice as long as it would if transmission is treated as a separate process or if verify data is ignored.

Listing 3. MC68HC711E9 Bootloader ROM

```
                ****************************************************
                * BOOTLOADER FIRMWARE FOR 68HC711E9 - 21 Aug 89
                ****************************************************
                * Features of this bootloader are...
                *
                * Auto baud select between 7812.5 and 1200 (8 MHz)
                * 0 - 512 byte variable length download
                * Jump to EEPROM at $B600 if 1st download byte = $00
                * PROGRAM - Utility subroutine to program EPROM
                * UPLOAD - Utility subroutine to dump memory to host
                * Mask I.D. at $BFD4 = $71E9
                ****************************************************
                * Revision A -
                *
                * Fixed bug in PROGRAM routine where the first byte
                * programmed into the EPROM was not transmitted for
                * verify.
                * Also added to PROGRAM routine a skip of bytes
                * which were already programmed to the value desired.
                *
                * This new version allows variable length download
                * by quitting reception of characters when an idle
                * of at least four character times occurs
                *
                ****************************************************

                * EQUATES FOR USE WITH INDEX OFFSET = $1000
                *
 0008           PORTD    EQU    $08
 000E           TCNT     EQU    $0E
 0016           TOC1     EQU    $16
 0023           TFLG1    EQU    $23
                * BIT EQUATES FOR TFLG1
 0080           OC1F     EQU    $80
                *
 0028           SPCR     EQU    $28            (FOR DWOM BIT)
 002B           BAUD     EQU    $2B
 002D           SCCR2    EQU    $2D
 002E           SCSR     EQU    $2E
 002F           SCDAT    EQU    $2F
 003B           PPROG    EQU    $3B
                * BIT EQUATES FOR PPROG
 0020           ELAT     EQU    $20
 0001           EPGM     EQU    $01
                *
```

Order this document
by EB188

Motorola Semiconductor Engineering Bulletin

EB188

Enabling the Security Feature on M68HC811E2 Devices with PCbug11 on the M68HC711E9PGMR

By Edgar Saenz
Austin, Texas

Introduction

The PCbug11 software, needed along with the M68HC711E9PGMR to program MC68HC811E2 devices, is available from the download section of the Microcontroller Worldwide Web site http://www.motorola.com/semiconductors/.

Retrieve the file pcbug342.exe (a self-extracting archive) from the MCU11 directory.

Some Motorola evaluation board products also are shipped with PCbug11.

NOTE: *For specific information about any of the PCbug11 commands, see the appropriate sections in the PCbug11 User's Manual (part number M68PCBUG11/D2), which is available from the Motorola Literature Distribution Center, as well as the Worldwide Web at http://www.motorola.com/semiconductors/. The file is also on the software download system and is called pcbug11.pdf.*

MOTOROLA