SEMICONDUCTOR APPLICATION NOTE

AN1241

Interfacing the MC68HC705J1A to 9356/9366 EEPROMs

By Mark Glenewinkel CSIC Applications

MOTOROLA

INTRODUCTION

This application note describes the hardware and software interface used to communicate between the Motorola MC68HC705J1A MCU and 9356/9366 EEPROM chips. The 93XX series of EEPROMs are an industry standard used widely to store nonvolatile bits of information. The software listing in this application note will work with 9356 and 9366 EEPROMs. The EEPROM bits are arranged in 128 or 256 16-bit registers, respectively. With some modification, the software will work with other 93XX series EEPROMs.

Some of the applications in which EEPROMs can be utilized are listed below.

- · ID number for remote addressing or security
- Storage of telecommunication information like phone number recall and speed dialing
- Power down information storage for consumer electronics like TVs and VCRs
- Reprogrammable calibration data for test/measurement equipment

The 93XX EEPROMs communicate with the outside world using a serial link. Since the MC68HC705J1A does not have the hardware on chip to communicate to the EEPROM, a software driver is used. This method bit programs an I/O port to properly transfer data to and from the EEPROM. A National NM93C56N was used for testing the software routines in this application note.

HARDWARE INTERFACE

The 9356 is a very simple 8-pin device. Appendix A shows a typical connection between the MC68HC705J1A and the 9356. The serial interface connection uses only four pins of the 9356. They are as follows:

- CS Chip Select
- SK Serial Clock
- DO Serial Data Output
- DI Serial Data Input

These signals must be clocked in a certain way in order to transfer the correct serial data to and from the MC68HC705J1A.



SOFTWARE INTERFACE

Communication between the MC68HC705J1A and the 9356 is done with a synchronous serial protocol. As mentioned earlier, the MC68HC705J1A bit programs its I/O pins to communicate with the 9356. A timing diagram of the serial link can be found in the 9356 data sheet if needed.

The 9356 will accept seven different commands. They are as follows:

- READ Read a 16-bit data word from an address in memory 1)
- WRITE Write a 16-bit data word from an address in memory 2)
- WRALL Write all addresses with the same 16-bit data word 3)
- ERASE Erase a 16-bit data word from an address in memory 4)
- 5) ERAL — Erase all addresses within the memory map
- 6) WEN — Erase/write enable the EEPROM memory
- 7) WDS — Erase/write disable the EEPROM memory

The 9356 transmission format is a frame of data bits containing an opcode, an address, and if needed, a word of data. The opcode is three bits long, the address is eight bits long, and the data word is 16 bits long. Table 1 illustrates the bit information each instruction needs.

Instruction	Opcode	Address	Data
	opecae	, (44, 666	Data
READ	110	A7–A0	
WEN	100	11XXXXXX	
ERASE	111	A7–A0	
ERAL	100	10XXXXXX	
WRITE	101	A7–A0	D15–D0
WRALL	100	01XXXXXX	D15–D0
WDS	100	00XXXXXX	

Та	bl	е	1.
		-	

IMPLEMENTATION AND TEST

Software was written to provide subroutines to perform each of the seven commands. A total of four bytes of RAM are needed to support the subroutines. These bytes are described below.

- 1) OP CODE Contains the opcode needed for the command
- Contains the address for the command 2) ADDR
- DATA H The high byte for the data word 3)
- 4) DATA L — The low byte for the data word

As needed, each EEPROM command subroutine will call other supporting subroutines to execute the transmission of data between the MC68HC705J1A and the 9356. Appendix B contains flowcharts for all of these subroutines.

Each EEPROM command subroutine has input data and output data. This data is inherent with some commands while others need the information passed to them before the subroutine is called. Table 2 lists the input data needed and output data generated for each of the seven commands.

Command	Subroutine Input	Subroutine Output
READ	ADDR	DATA_H/L
EWEN	—	—
ERASE	ADDR	—
ERAL	_	_
WRITE	ADDR & DATA_H/L	—
WRALL	DATA_H/L	—
EWDS	_	_

Table 2:

Code was written and tested with a level of quality equal to the Carnegie-Mellon Software Engineering Institute (SEI) Level 2. A test routine consisting of writing and reading the EEPROM is listed in Appendix C. Refer to Appendix A for the schematic used in the design and test of the software. An LED is used to verify that the test code works properly. The test routine executes the following:

- 1) Initializes the port on the HC705J1A for serial transmission. LED is turned off.
- 2) Writes EEPROM address \$00 with \$AA55.
- 3) Writes EEPROM address \$20 with \$1234.
- 4) Reads EEPROM address \$00 and stores it to RAM location TEST1 and TEST2.
- 5) Reads EEPROM address \$20 and stores it to RAM location TEST3 and TEST4.
- 6) Checks if TEST1 = \$AA, TEST2 = \$55, TEST3 = \$12, and TEST4 = \$34.
- 7) If check is good, then light the LED. If check is bad, do not light the LED.
- 8) Continue to run in an infinite loop until reset.

For increased reliability, the software watchdog on the MC68HC705J1A is used. Also, a low voltage inhibit circuit, the MC34064, is used to decrease susceptibility to brown out or short power failure conditions.

SUMMARY

This application note has described the interface needed to successfully communicate between the MC68HC705J1A and the 9356. For more information on the MC68HC705J1A, please consult the Technical Data Manual, MC68HC705J1A/D. Contact National Semiconductor or SGS Thompson for technical data on the 93XX series of EEPROM memories.

An electronic copy of the code listing in Appendix C and a listing of the test program to fully test all the EEPROM commands can be found on the Motorola MCU BBS. The BBS number is (512) 891-3733. The filename is j1a_9356.arc and is on the CSIC BBS under the APPNOTES directory.

Also, Motorola Application Note AN1221/D further details the software and hardware interfaces needed between the 93XX series and other HC05 MCUs.

APPENDIX A



APPENDIX B











APPENDIX C

* Main Routine J1A 9356 - 705J1A to 9356 EEPROM * * File Name: J1A_9356.RTN Copyright (c) Motorola 1995 * Full Functional Description Of Routine Design: * Program flow: * Reset: Initializes ports for bit banging. * Calls EWEN sub to enable write to EEPROM. * Calls ERAL to erase all EEPROM * Writes \$AA55 to EEPROM \$00 * Writes \$1234 to EEPROM \$20 * Reads EEPROM \$00 and \$20 * Check for correct data, light LED if correct * Execute endless loop * Part Specific Framework Includes Section #nolist #INCLUDE 'H705J1A.FRK' ;Include the equates for the HC705J1A ; so that all labels can be used. #list MOR Bytes Definitions for Main Routine org MOR db \$21 ;COP enabled, osc resistor enabled ; If used on a mask rom part, ; be sure to specify this option.

* * * * * Equates and RAM Storage * * CS equ 0 ;bit # for chip select equ 1 equ 2 SER CLK ;bit # for serial clock ;bit # for serial data out SER OUT SER_IN 3 ;bit # for serial data in equ * * * RAM storage variables *** ;start of static RAM at \$C0 org RAM OPCODE rmb 1 ;command byte ;EEPROM address byte ADDR rmb 1 DATA H 1 ;MSByte of data rmb DATA_L 1 ;LSByte of data rmb rmb 1 ;test byte #1 TEST1 TEST2 ;test byte #2 rmb 1 TEST3 rmb 1 ;test byte #3 test4 rmb 1 ;test byte #4 * * * * Program Initialization * This section sets up the port for bit banging. * * * To prevent floating inputs and associated high current draw, * * the HC705J1A has pulldown devices on all I/O pins. This * initialization should enable these pulldowns on unused I/O * pins. RESET enables the pulldowns, so no code is required. orq EPROM J9356 START lda #\$80 ;init portA PORTA sta ; sta COPR ;kick the wdog #\$87 ;init i/o of port A lda sta DDRA

* * * * * * *	* * * * * * * * * * * * * * * *	* * * * * * * * * * * * * * * * * * *	******	* * *
*				*
*	J1A_9356 Main	Program Loop		*
*		1		*
		n the test routin		*
* prope	er serial transm	MISSION. THE LED	is lit if the test passes.	*
*****	* * * * * * * * * * * * * * * * * *	* * * * * * * * * * * * * * * * *	*****	**
* * *	Enable erase/w	write mode of EEF	PROM	
	jsr	J9356_EWEN	;call ewen routine	
* * *	Erase all EEPP	ROM memory map		
	jsr	J9356_ERAL	;call eral routine	
* * *	Write \$AA55 to	<u>, 400</u>		
		#\$00	;load address	
	sta	ADDR	, iouu uuui coo	
	lda	#\$AA	;load data byte high	
	sta	DATA H		
	lda	#\$55	;load data byte low	
	sta	DATA L		
	jsr	J9356_WRITE	;call write routine	
* * *		- <u>400</u>		
	Write \$1234 to lda	#\$20	;load address	
	sta	#\$20 ADDR	/IOad address	
	lda	#\$12	;load data byte high	
	sta	DATA H	/ioad data byte migh	
	lda	#\$34	;load data byte low	
	sta	DATA L	, ioda data byte iow	
	jsr	J9356 WRITE	;call write routine	
	5~2	0,000	/0411 11100 1040110	
* * *	Read \$00			
	lda	#\$00	;load address	
	sta	ADDR		
	jsr	J9356_READ	;call read routine	
	lda	DATA_H		
	sta	TEST1	;store away data_h to test1	
	lda	DATA_L		
	sta	TEST2	;store away data_l to test2	
* * *	Read \$20			
	lda	#\$20	;load address	
	sta	ADDR		
	jsr	J9356_READ	;call read routine	
	lda	DATA_H	-	
	sta	TEST3	;store away data_h to test3	
	lda	DATA_L		
	sta	TEST4	;store away data_l to test4	

*** Check J9356_CKSUM	results o lda cmpa	f write and read TEST1 #\$AA	, light LED if good ;check test1
	bne	J9356_BRANCH	;branch if no good, no LED
	lda cmpa	TEST2 #\$55	;check test2
	bne	J9356_BRANCH	;branch if no good, no LED
	lda cmpa	TEST3 #\$12	;check test3
	bne	J9356_BRANCH	;branch if no good, no LED
	lda cmpa	TEST4 #\$34	;check test4
	bne		;branch if no good, no LED
	bclr	7, porta	;EEPROM write and read is good ; light LED
J9356_BRANCH	clra		

_	sta	COPR	;kick	the wdog
	bra	J9356_BRANCH		

* * * EEPROM Command SubRoutines * * These 7 subroutines execute each of the 7 commands * * that the EEPROM will respond to * * * J9356 EWEN lda #\$80 ;load opcode OPCODE sta #\$C0 lda ;load address sta ADDR bset CS, PORTA ;CS line is high jsr J9356_WR_OP ;write opcode jsr J9356_WR_ADDR ;write address bclr CS,PORTA ;CS line is low rts ;return * * * J9356_EWDS lda #\$80 ;load opcode OPCODE sta clr ADDR ;load addr bset CS,PORTA ;CS line is hig jsr J9356_WR_OP ;write opcode jsr J9356_WR_ADDR ;write address ;CS line is high bclr CS,PORTA ;CS line is low rts ;return * * * J9356 WRITE lda #\$A0 ;load opcode OPCODE sta bset CS, PORTA ;CS line is high J9356_WR_OP ;write opcode jsr J9356_WR_ADDR ;write address jsr jsr J9356_WR_DATA ;write data bclr SER_OUT, PORTA ;CS line is low bclr CS, PORTA jsr J9356_WAIT ;wait until EEPROM is ready rts ;return

*** WRAL - J9356_WRAL	subrouti lda sta lda sta bset jsr jsr jsr bclr bclr jsr rts	ne to write all #\$80 OPCODE #\$40 ADDR CS,PORTA J9356_WR_OP J9356_WR_ADDR J9356_WR_DATA SER_OUT,PORTA CS,PORTA J9356_WAIT	<pre>EEPROM ************************************</pre>
*** READ - J9356_READ	subrouti lda sta bset jsr jsr bset bclr jsr bclr bclr rts	ne to read EEPRC #\$C0 OPCODE CS,PORTA J9356_WR_OP J9356_WR_ADDR SER_CLK,PORTA SER_CLK,PORTA J9356_RD_DATA SER_OUT,PORTA CS,PORTA	<pre>M ************************************</pre>
*** ERASE - J9356_ERASE	- subrout lda sta bset jsr jsr bclr jsr rts	ine to erase EEP #\$E0 OPCODE CS,PORTA J9356_WR_OP J9356_WR_ADDR CS,PORTA J9356_WAIT	ROM ************************************
*** ERAL - J9356_ERAL	subrouti lda sta sta bset jsr jsr bclr jsr rts	ne to erase all #\$80 OPCODE ADDR CS,PORTA J9356_WR_OP J9356_WR_ADDR CS,PORTA J9356_WAIT	EEPROM ************************************

* * * EEPROM Supporting SubRoutines * These subroutines support the functions called from the Command subs * * * * J9356 WR OP ; init counter for LOOP1 ldx #3T Write to the serial output pin brclr 7,OPCODE,J9356_L1_2 ; if opcode bit7 = 0, goto L1_2 J9356_LOOP1 bset SER_OUT,PORTA ;ser_out = 1 bra J9356 L1 3 ;qoto L1 3 J9356 L1 2 bclr SER OUT, PORTA ;ser out = 0 Clock the serial clock pin J9356_L1_3 bset SER_CLK,PORTA ;ser_clk = 1 bclr SER CLK, PORTA ;ser clk = 0 OPCODE asl ;rotate the opcode decx ;decrease counter loop bne J9356_LOOP1 ; is LOOP1 finished? clra COPR ;kick the wdog sta ;return rts J9356_WR_ADDR ldx #8T ; init counter for LOOP2 Write to the serial output pin brclr 7,ADDR,J9356_L2_2 ; if addr bit7 = 0, goto L2_2 J9356 LOOP2 bset SER_OUT,PORTA ;ser_out = 1 bra J9356_L2_3 ;goto L2_3 SER_OUT,PORTA ;ser_out = 0 bclr J9356_L2_2 Clock the serial clock pin J9356_L2_3 bset SER_CLK,PORTA ;ser_clk = 1 bclr SER CLK, PORTA ;ser clk = 0 ADDR asl ;rotate the addr decx ;decrease counter loop J9356_LOOP2 ; is LOOP2 finished? bne rts ;return

#16T J9356_WR_DATA ldx ; init counter for LOOP4 Write the serial output pin with data brclr 7,DATA H,J9356 L4 2 ; if addr bit7 = 0, goto L4 2 J9356 LOOP4 SER_OUT,PORTA ;ser_out = 1 bset ;goto L4_3 bra J9356 L4 3 bclr SER_OUT,PORTA ;ser_out = 0 J9356_L4_2 Clock the serial clock pin J9356 L4 3 bset SER_CLK, PORTA ;ser clk = 1 bclr SER_CLK,PORTA ;ser_clk = 0 DATA_L asl ;rotate the DATA_L DATA_H rol ;rotate the DATA_H decx ;decrease counter loop ; is LOOP4 finished? bne J9356 LOOP4 rts ;return * * * J9356 RD DATA ldx #16T ; init counter for LOOP3 Read the serial input pin brclr SER_IN,PORTA,J9356_L3 ;carry bit = serial in J9356 LOOP3 J9356_L3 rol DATA_L ;rotate left result rol DATA H Clock the serial clock pin bset SER CLK, PORTA ;ser clk = 1 bclr SER_CLK,PORTA ;ser_clk = 0 decx ;decrease counter loop J9356 LOOP3 ; is LOOP3 finished? bne ;return rts Wait until write cycle is over bset CS, PORTA J9356 WAIT ;CS line is high J9356 W2 clra sta COPR ;kick the wdog brclr SER_IN,PORTA,J9356_W2 bclr CS,PORTA ;CS line is low ;return rts

org	RESET
fdb	J9356_START

NOTES

NOTES

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola and (Aire registered trademarks of Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

Literature Distribution Centers:

USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036. EUROPE: Motorola Ltd.; European Literature Centre; 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England. JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141 Japan. ASIA-PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No.2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.



