

Communicating with 3-wire EEPROMs

Interfacing 93CX6 serial EEPROMs

INTRODUCTION

The 93CX6 Serial EEPROMs feature a three/four wire serial interface bus. The attractive price and simple interface make it the ideal device for additional memory space. This application note is intended for design engineers who wish to incorporate a pre-packaged serial EEPROM interface driver into their application.

THE HARDWARE CONNECTION

A typical 4-wire hardware connection is illustrated in Figure 1 and a typical 3-wire connection is illustrated in Figure 2. Since all I/O ports on the GMS87C1202 are configurable as input and/or output, a 3-wire interface makes optimum utilization of the I/O pins by having a common connection for the DI and DO lines of the serial EEPROM. The port pin on the GMS87C1202 connected to these pins, has a default setting as an output and is configured, when needed, as an input during program execution.

THE SOFTWARE CONNECTION

An example interface driver is listed in Appendix A. A flow diagram is given in Figure 3. The interface driver is written to minimize both overhead to the calling program as well as the program space necessary for its inclusion into the user's code. The driver has been written as a generic driver to service all 93 Series Serial EEPROMs.

Note: The four command/data passing registers have to be defined consecutively in order for the FSR to access them successfully in the program execution.

The user should take the following steps when using the routines provided in Appendix A.

- A) Specify and define a 3-/4-wire interface by defining the common connection to the DI/DO lines and setting the equate 'wire3' TRUE or FALSE (4-wired is automatically assumed if 3-wire is false).
- B) Specify and define if 16-bit or 8-bit data organization is used, by setting equate 'org8' TRUE or FALSE.
- C) The user should assemble the source file by specifying which type of serial EEPROM is being used. This is done by defining the equate 93C46 as TRUE. Only one device can be TRUE, the rest have to be defined FALSE.
- D) The user would invoke the driver as follows:

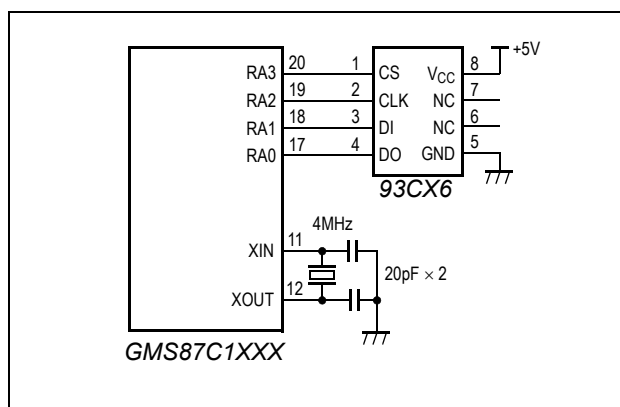


Figure 1. 4 Wire Connection

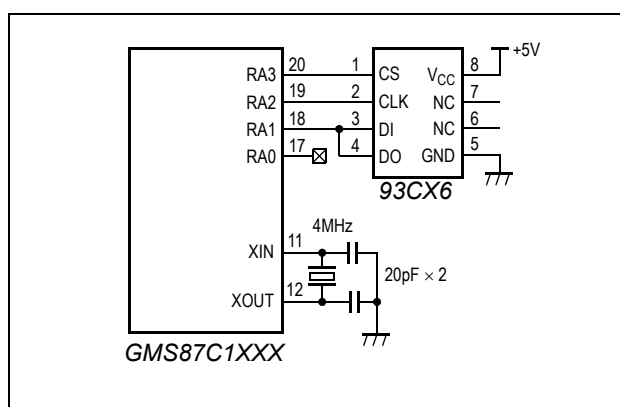


Figure 2. 3 Wire Connection

1. Load the register defined as 'cmd' with the 93CX6 Command Opcode (only the four upper bits are used in this register; see Figure 4).
2. If necessary, load the register defined 'addr' with the lower 8/7/6 bit address of the location.
3. If necessary, load the 9th bit of the address as bit 3 of the register defined as 'cmd' (see)
Note: READ, WRITE and ERASE commands need to have an address associated with the command and the 9th bit of the address is only required when using the 93C56/66 devices in the 8 bit mode (ORG tied to GND).
4. If necessary, load the register defined as 'highb' and 'lowb' with the 16 bits of data, the most significant byte loaded in 'highb'. In 8 bit mode, 'highb'

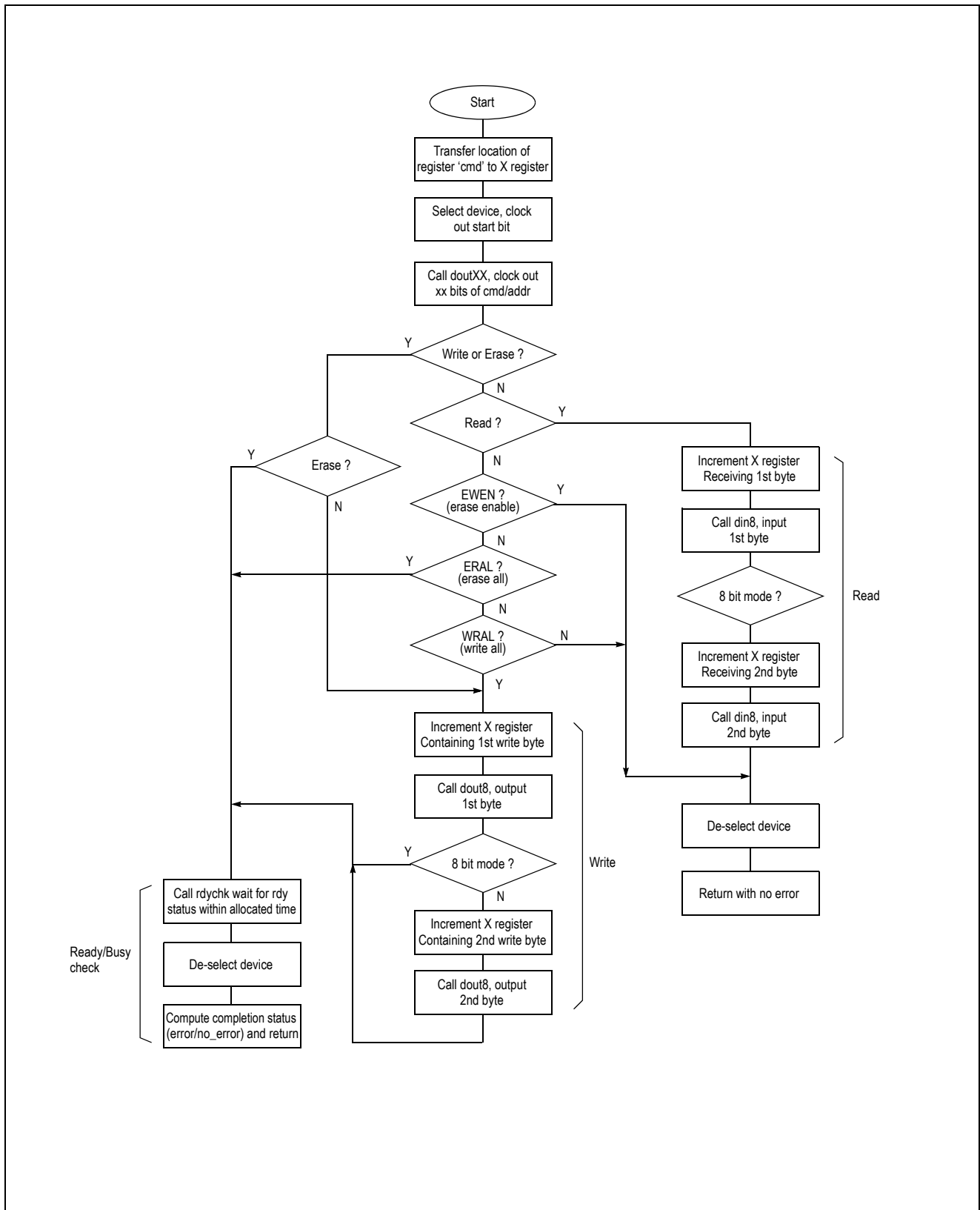


Figure 3. Flow Chart

should be loaded with the 8 bit data.

Note: Only the WRITE and WRAL commands require data to be loaded in the 'highb', 'lowb' locations.

5. Call the driver sub-routine 'see'.
6. Upon completion, the driver will return a completion status in the Accumulator register (error/no error). Only commands requiring a status check are capable of returning a valid error/no error status, in all other cases a no error is returned.
7. If the READ command is executed, the 16/8 bit data will be loaded in the 'highb' and 'lowb' registers, where 'highb' contains the MSB in the 16 bit mode and 8 bit data in the 8-bit mode.

The Example interface assumes a 4MHz oscillator clock which gives us a 1µS instruction cycle. If a higher clock speed is used, additional NOPs have to be included in the code in order to meet the minimum clock speed requirements of the 93 Series Serial EEPROMs (see data sheet for further

details).

Listing in Appendix B is for an interface to 93C46 Serial EEPROM only.

SUMMARY

The 93 Series Serial EEPROMs are a simple and versatile method of increasing read/write memory capability in the GMS87C1XXX Series MCU application. The 'generic' code in Appendix A makes it easy to incorporate in any GMS800 series application. Any of the 93CXX Series Serial EEPROMs can be applied, while at the same time using a minimal amount of I/O, code RAM resources.

Code size: 6 bytes of RAM

Appendix A listing: 127 byte program code (max.)

Appendix B listing: 236 byte program code

*Author: Jackson Shin
MCU application team
e-mail: jackson.shin@hynix.com*

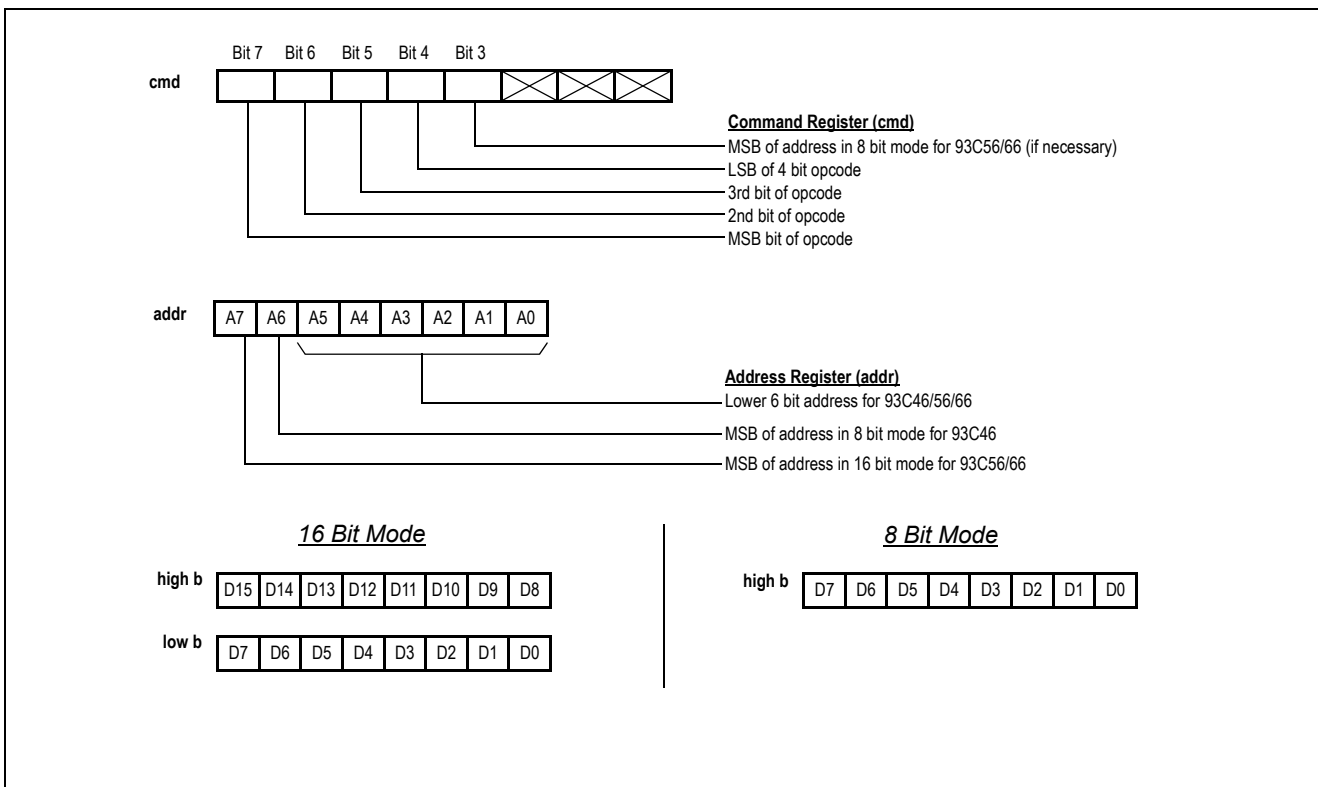


Figure 4. 'CMD', 'ADDR' data byte definition

APPENDIX A

GMS800 series MICOM ASSEMBLER Thu Feb 01 18:09:37 2001
(PAGE 1)

```

1          PAGE      10000
2          ;*****
3          ; PROGRAM DESCRIPTION:
4          ; Interface 93C46 Serial EEPROM with GMS87C1202
5          ;-----
6          ; MICROCONTROLLER : GMS87C1202 (2K ROM, 20 PIN DIP)
7          ; OPERATING FREQUENCY : 4MHz
8          ; SERIAL EEPROM   : 93C46
9          ;-----
10         ; programmed by Jongpil Shin (HEI) on 2001.1.30      Version 1.00
11         ;*****
12         ;
13         ;***** USER RAM ASSIGNMENT *****
14         ;
15         loopcnt DS      1
16         ;
17         ; The following three registers must be
18         ; located consecutively in memory.
19         cmd      DS      1
20         highb   DS      1
21         lowb    DS      1
22         ;
23         ;***** CONTROL REGISTER ASSIGNMENT *****
24         ;
25         RA      EQU      0C0H
26         RAIO    EQU      0C1H
27         ;
28         ;***** BIT ASSIGNMENT *****
29         ; The following assignments are for 3-wire setup.
30         ; For 4-wire please assign DATAOUT equ 0.
31         ;
32         DATAOUT EQU    1          ; Port pin tied to DO pin on 93C46, 3 wire setup
33         DATAIN  EQU    1          ; Port pin tied to DI pin on 93C46, 3 wire setup
34         CLOCK   EQU    2          ; Port pin tied to CLK on 93C46
35         CS      EQU    3          ; Port pin tied to CS on 93C46
36         ;
37         ;***** SYMBOL ASSIGNMENT *****
38         ;
39         SERIAL  EQU      RA        ; Port RA used for 93C46
40         INPUT   EQU      0000_1101B ; Make RA1 port as an input mode
41         NO_ERROR EQU      0
42         ERROR   EQU      1
43         ;
44         ;After issuing a WRITE, ERASE, ERAL, or WRAL command the approximate number of machine
45         ;cycle X 256 to wait for the RDY status.
46         ;This value must be adjusted as 10ms for operating frequencies other than 4MHz
47         TRIES   EQU      8
48         ;
49         ; 93C46 Command
50         READ    EQU      1000_0000B ; read command op code
51         WRITE   EQU      0100_0000B ; write command op code
52         ERASE   EQU      1100_0000B ; erase command op code
53         EWEN    EQU      0011_0000B ; erase enable command op code
54         EWDS    EQU      0000_0000B ; erase disable command op code
55         ERAL    EQU      0010_0000B ; erase all command op code
56         WRAL    EQU      0001_0000B ; write all command op code
57         ;
58         ;***** MACRO DEFINITION *****

```

```

59      ;
60      startbit MACRO
61          SET1    SERIAL.DATAIN
62          CLKIT
63          ENDM
64
65      dsel    MACRO                ; De-select the 93cX6 device
66          CLR1    SERIAL.CS      ; Chip Select (CS) = '0' to de-select the device
67          ENDM
68
69      sel     MACRO                ; select the 93cX6 device
70          SET1    SERIAL.CS      ; Chip Select (CS) = '1' to select the device
71          ENDM
72
73      clkkit  MACRO                ; Clock a serial data bit into or out of the 93CX6 device.
74          SET1    SERIAL.CLOCK   ; Clock (CLK) = '1'
75          NOP                    ; Adjust the number of nop instructions between the
76                                ; assertion and de-assertion of CLK in proportion to the
77                                ; GMS87C1202 operating frequency.
78          CLR1    SERIAL.CLOCK
79          ENDM
80      ;
81      ;***** VECTOR AREA *****
82      ;
83          ORG     0FFFEH
84  FFFE 00F8      DW     RESET      ;Reset Vector
85
86          ORG     0F800H
87
88  F800 1E7F      RESET:  LDX     #07FH      ;Load X with 07Fh.
89  F802 8E        TXSP                    ;Initialize SP = #7FH.
90  F803 E400C0    LDM     RA,#0000_0000B ;Initialize Port RA pin.
91  F806 E40FC1    LDM     RAIO,#0000_1111B ;Set RA0-RA3 as output mode
92  F809 1B9AF8    JMP     start
93      ;
94      ;*****
95      ;                               DOUT8
96      ;*****
97      ; Dout8 will output 8 bits of data to the 93C46. Before calling this routine,
98      ; the X register must point to the byte being transmitted.
99      ;
100     dout8:
101  F80C D4        LDA     {X}          ;Load data to be transmitted.
102  F80D 3E08      LDY     #8           ;Initialize bit counter.
103  F80F 28        d_o_8:  ROL     A           ;Get bit information.
104  F810 EBC030    STC     SERIAL.DATAIN ;Set or Clear DATAOUT port pin.
105              CLKIT                ;Clock the 93C46.
106  F813 41C0     @      SET1    SERIAL.CLOCK
107  F815 FF       @      NOP
108              @
109              @
110  F816 51C0     @      CLR1    SERIAL.CLOCK
111  F818 BE       DEC     Y           ;Repeat until bit coueter Y = 0.
112  F819 70F4     BNE     d_o_8       ;Y still > 0
113  F81B 31C0     CLR1    SERIAL.DATAIN ;make sure DI pin low before return.
114  F81D C400     LDA     #NO_ERROR ;Exit with good status.
115  F81F 6F       RET
116
117      ;*****
118      ;                               Datin 8
119      ;*****
120      ; Din8 will input 8 bits of data from the 93C46. Before calling this routine,

```

```

121          ; the X register must point to the register being used to hold the incoming data.
122          ;
123          din8:
124 F820 E40DC1      LDM    RAO, #INPUT    ; set up the RA1 as an input before proceeding
125 F823 3E08        LDY    #8
126 F825 D4          LDA    {X}
127          d_i_8:
128          CLKIT          ;Clock the 93C46.
129 F826 41C0 @      SET1   SERIAL.CLOCK
130 F828 FF @        NOP
131 @
132 @
133 F829 51C0 @      CLR1   SERIAL.CLOCK
134 F82B CBC020      LDC    SERIAL.DATAOUT    ;Read the incoming bit, then store it into Carry bit.
135 F82E 28          ROL    A          ;Make room for the incoming bit in the destination register.
136 F82F BE          DEC    Y          ;Repeat until bit counter Y = 0
137 F830 70F4        BNE    d_i_8          ;Y still > 0
138 F832 F4          STA    {X}          ;Store read word into highb or/and lowb.
139 F833 E40FC1      LDM    RAO, #0000_1111B    ;Setup RA1 back to output.
140 F836 C400        LDA    #NO_ERROR    ;Exit with good status.
141 F838 6F          RET
142          ;
143          ;*****
144          ;                               RDYCHK
145          ;*****
146          ; Rdychk will read the 93C46 READY/BUSY status and wait for RDY status within
147          ; the allotted number of processor cycles. If RDY status is not present after
148          ; this set period, the routine will return with an error status.
149          ;
150          rdychk:
151 F839 3E08        LDY    #TRIES    ;Initialize time-out counter upper byte of 16 bits
152 F83B C400        LDA    #0        ;Initialize lower byte of 16 bits (11.28ms)
153
154 F83D E40DC1      LDM    RAO, #INPUT    ;Set up the RA1 as an input before proceeding
155          dsel          ;De-select the 93C46
156 F840 71C0 @      CLR1   SERIAL.CS
157          ; Note: Check the 93C46 data sheet for minimum CS low time. Depending upon
158          ; processor frequency, a NOP(s) may be between the assertion and de-assertion of
159          ; Chip Select.
160          ;NOP
161          sel          ;Re-select the 93C46
162 F842 61C0 @      SET1   SERIAL.CS
163 F844 23C00C      notrdy: BBS    SERIAL.DATAOUT, rdynoerr    ;Tset if DO PIN='1' 93c46 has completed.
164 F847 A8          DEC    A          ;No, not yet ready. Decrement the A
165 F848 70FA        BNE    notrdy    ;16 bit timer and check for expiration.
166 F84A BE          DEC    Y          ;still some left time. Try again.
167 F84B 70F7        BNE    notrdy    ;no, try again.
168
169 F84D E40FC1      LDM    RAO, #0000_1111B    ;setup RA1 back as output
170 F850 C401        LDA    #ERROR    ;RDY status was not present in the allotted time,
171 F852 6F          RET          ;...return with error status.
172
173 F853 E40FC1      rdynoerr: LDM    RAO, #0000_1111B    ;setup RA1 back to output
174 F856 C400        LDA    #NO_ERROR
175 F858 6F          RET
176          ;
177          ;*****
178          ;                               SEE
179          ;*****
180          ; See will control the entire operation of a 93C46 device. Prior to calling the
181          ; routine, load a valid command/memory address into location cmd, and for WRITE
182          ; or WRAL commands, load registers highb and lowb with 16 bits of write data.

```

```

183         ; Upon exit, the accumulator will contain the completion status.
184         ; Only 93C46 instruction which require a status check can return with an error
185         ; as the completion status. The values that denote the completion status are
186         ; defined as variables 'error' and 'no_eerror' in the general assignments section.
187         ;
188         see:
189 F859 1E01          LDX      #cmd          ;Load X with the loacation of the cmd register.
190         sel        ;Chip Select (CS) = '1' to select the 93C46.
191 F85B 61C0  @      SET1     SERIAL.CS
192         startbit   ;Sene a start bit.
193 F85D 21C0  @      SET1     SERIAL.DATAIN
194         @ CLKIT
195 F85F 41C0  @      SET1     SERIAL.CLOCK
196 F861 FF    @      NOP
197         @
198         @
199 F862 51C0  @      CLR1     SERIAL.CLOCK
200
201         ; READ, WRITE, ERASE instruction are processed by following routine
202
203 F864 3B0CF8       CALL     dout8          ;Transmit the 2 bit command and six bit address
204
205 F867 C3010E       BBS      cmd.6,see2      ;Check for a WRITE or ERASE command, yes -> see2
206 F86A E30119       BBS      cmd.7,read_     ;Check for a READ command, if yes, process READ command.
207 F86D A30111       BBS      cmd.5,see3      ;Check for a EWEN or ERAL command.
208 F870 83011D       BBS      cmd.4,write_    ;Check for a WRAL command, if yes, porcess WRAL/WRITE command.
209
210         exit_: dsel          ;No futher processing required; 93C46
211 F873 71C0  @      CLR1     SERIAL.CS
212 F875 C400         LDA      #NO_ERROR    ;Return with good completion status.
213 F877 6F          RET
214
215 F878 F30115       see2:   BBC      cmd.7,write_   ;WRITE command
216 F87B 3B39F8       exit2_: CALL     rdychk
217         dsel
218 F87E 71C0  @      CLR1     SERIAL.CS
219 F880 6F          RET
220
221 F881 9301F7       see3:   BBC      cmd.4,exit2_   ;ERAL command
222 F884 2FED         BRA      exit_          ;WRAL command
223
224 F886 8F          read_:  INC      X              ;Increment the X index register to point to the register...
225         ;...receiving the upper byte of the incomming 93C46 data word.
226 F887 3B20F8       CALL     din8          ;input the upper byte.
227 F88A 8F          INC      X              ;Increment the Index Register to point to the register...
228         ;...receiving the lower byte.
229 F88B 3B20F8       CALL     din8          ;Input 8 more bits.
230 F88E 2FE3         BRA      exit_          ;No, futher processing required, exit now.
231
232 F890 8F          write_: INC      X              ;Increment the X index Register to point to the upper byte of
233         ;...the 16 bit 93C46 data word to be transmitted.
234 F891 3B0CF8       CALL     dout8         ;Output that byte.
235 F894 8F          INC      X              ;Increment the X index Register to point to the lower byte.
236 F895 3B0CF8       CALL     dout8         ;Output the lower byte of the 16 bit 93C46 data word.
237 F898 2FE1         BRA      exit2_        ;Exit with a status check.
238
239         ;
240         ;*****
241         ;                               Test Program
242         ;*****
243         ; We've include a sample program to exercise the GMS87C1202 to 93C46 interface using a simple
244         ; erase, write and verify routine.
245         ;

```

```

245          start:
246 F89A E400C0      LDM    serial,#0      ;Clear the port tied to the 93C46 device.
247 F89D E40FC1      LDM    RAIO,#0000_1111B    ;Initialize the data direction register for that port.
248 F8A0 E43001      LDM    cmd,#EWEN      ;Transfer Erase/Write Enable command into cmd register.
249 F8A3 3B59F8      CALL   see          ;Enable the 93C46 device.
250
251 F8A6 E42001      LDM    cmd,#ERAL      ;Transfer Erase All command into 'cmd' register.
252 F8A9 3B59F8      CALL   see          ;Erase the 93C46.
253 F8AC 4401        CMP    #ERROR        ;Check completion status.
254 F8AE F037        BEQ    errloop       ;Yes, bad completion status, error-out.
255
256          TSTPTRN EQU    0AAH          ;Define the test pattern to be written.
257
258 F8B0 E44000      LDM    loopcnt,#64
259 F8B3 E44001      LDM    cmd,#WRITE     ;Load with Write command.
260 F8B6 E4AA02      LDM    highb,#TSTPTRN ;Initialize the 93C46 data register with write data.
261 F8B9 E4AA03      LDM    lowb,#TSTPTRN
262 F8BC 3B59F8      test1: CALL   see          ;Write data word into 93C46 device.
263 F8BF 4401        CMP    #ERROR        ;Check if error ?
264 F8C1 F024        BEQ    errloop       ;Yes, bad completion status, error-out.
265 F8C3 8901        INC    cmd           ;No, increment the 6 bit memory address filed.
266 F8C5 A900        DEC    loopcnt
267 F8C7 70F3        BNE    test1        ;Is Write precessing finished up tp 0FFH ?
268          ;
269          ; Read loop
270          ;
271 F8C9 E44000      LDM    loopcnt,#64   ;Initialize loop counter.
272 F8CC E48001      LDM    cmd,#READ     ;Load with Read command.
273 F8CF 3B59F8      test2: CALL   see          ;Read addressed data word from 93C46 device.
274 F8D2 C502        LDA    highb         ;Load Acc. with the pattern written.
275 F8D4 44AA        CMP    #TSTPTRN     ;Same ? Verify the data read against what was written.
276 F8D6 700F        BNE    errloop       ;Not same, error-out.
277 F8D8 C503        LDA    lowb         ;Repeat with the lower byte read.
278 F8DA 44AA        CMP    #TSTPTRN     ;Same ?
279 F8DC 7009        BNE    errloop       ;No, error-out.
280 F8DE 8901        INC    cmd           ;Yes, both byte correct, increment the 6 bit memory
281          ;address field.
282 F8E0 A900        DEC    loopcnt      ;Have we read all 64 locations ?
283 F8E2 70EB        BNE    test2        ;No, read another location.
284
285 F8E4 1BE4F8      all_ok: JMP    all_ok      ;Thank you !!!
286
287 F8E7 1BE7F8      errloop: JMP    errloop ;Emm..... Bad news !
288          ;
289          END

```

-- 0 Error(s) --

--- Total Machine Code : 236 Bytes --