Using an EEPROM— IIC Interface NM24C02/03/04/05/08/09/16/17

Fairchild
Application Note 794

FAIRCHILD
SEMICONDUCTOR™

INTRODUCTION

Fairchild Semiconductor's NM24C EEPROMs are designed to interface with Inter-Integrated Circuit (IIC) buses and hardware. Fairchild's electrically erasable programmable read only memories (EEPROMs) offer valuable security features (write protection), two write modes, three read modes and a wide variety of memory sizes. Applications for the IIC bus and NM24C memories are included in SANs (small-area networks), stereos, televisions, automobiles and other scaled-down systems that don't require tremendous speeds but instead cost efficiency and design simplicity.

IIC BACKGROUND

The IIC bus configuration is an amalgam of microcontrollers and peripheral controllers. By definition: a device that transmits signals onto the IIC bus is the "transmitter" and a device that receives signals is the "receiver"; a device that controls signal transfers on the line in addition to controlling the clock frequency is the "master" and a device that is controlled by the master is the "slave". The master can transmit or receive signals to or from a slave, respectively, or control signal transfers between two slaves, where one is the transmitter and the other is the receiver. It is possible to combine several masters, in addition to several slaves, onto an IIC bus to form a multimaster system. If more than one master simultaneously tries to control the line, an arbitration procedure decides which master gets priority. The maximum number of devices connected to the bus is dictated by the maximum allowable capacitance on the lines, 400 pF, and the protocol's addressing limit of 16k; typical device capacitance is 10 pF. Up to eight E^2PROMs can be connected to an IIC bus, depending on the size of the memory device implemented.

Simplicity of the IIC system is primarily due to the bidirectional 2-wire design, a serial data line (SDA) and serial clock line (SKL), and to the protocol format. Because of the efficient 2-wire configuration used by the IIC interface compared to that of the MICROWIRE™ and SPI interface, reduced board space and pin count allows the designer to have more creative flexibility while reducing interconnecting cost.

OPERATING Fairchild SEMICONDUCTOR'S NM24Cs

The NM24C E^2PROMs require only six simple operating codes for transmitting or receiving bits of information over the 2-wire IIC bus. These fields are explained in greater detail below and briefly described hereafter: a start bit, a 7-bit slave address, a read/write bit which defines whether the slave is a transmitter or receiver, an acknowledge bit, message bits divided into 8-bit segments and a stop bit.

For efficient and faster serial communication between devices, the NM24C Family features page write and sequential read.

The NM24C03/C05/C09/C16/C17 Family offers a security feature in addition to standard features found in the NM24C02/C04/C08/C16 Family. The security feature is beneficial in that it allows Read Only Memory (ROM) to be implemented in the upper half of the memory to prevent any future programming in that particular chip section; the remaining memory that has not been write protected can still be programmed. The security feature in the NM24C03/C05/C09/C17 Family does not require immediate implementation when the device is interfaced to the IIC bus, which gives the designer the option to choose this feature at a later date. Table 1 displays the following parameters: memory content, write protect and the maximum number of individual IIC E^2PROMs allowed on an IIC bus at one time if the total line capacitance is kept below 400 pF.

Code used to interface the NM24Cs with Fairchild Semiconductor's COP8™ Microcontroller Family is listed in a latter section of this application note for further information to the reader.

TABLE 1.

Part No.	Number of 256x8 Page Blocks	Write Protect Feature	Max. Parts
NM24C02	1	No	8
NM24C03	1	Yes	8
NM24C04	2	No	4
NM24C05	2	Yes	4
NM24C08	4	No	2
NM24C09	4	Yes	2
NM24C16	8	No	1
NM24C17	8	Yes	1

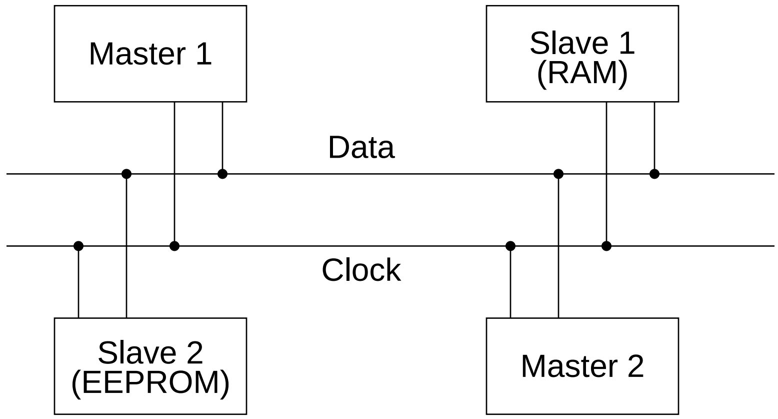


FIGURE 1. IIC-Bus Configurations

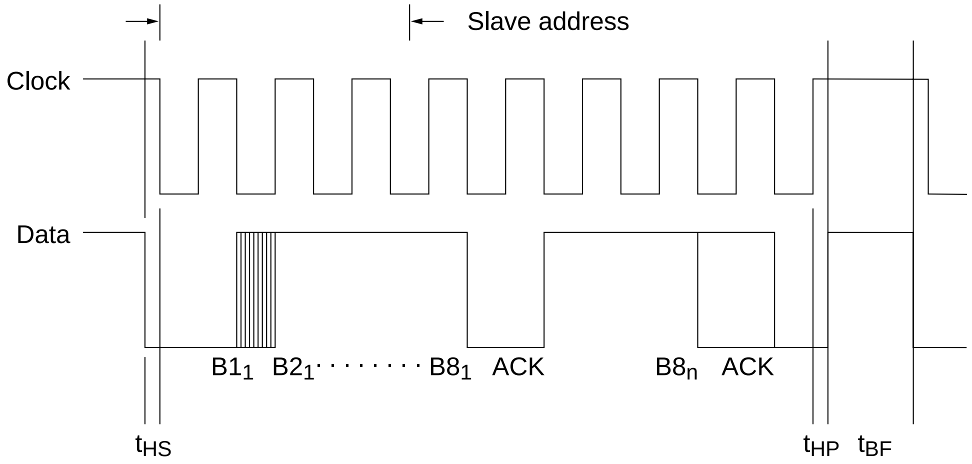


FIGURE 2. IIC Bus Timing

Start Condition

— Clock and Data line high (Bus free)

— Change Data line from high to low

— After $t_{HS(Min)}$ = 4 µs the master supplies the clock

Acknowledge

— Transmitting device releases the Data line

— The receiving device pulls the Data line low during the ACK-clock if there is no error

— If there is no ACK, the master will generate a Stop Condition to abort the transfer

Stop Condition

— Clock line goes high

— After $t_{HP(Min)}$ = 4.7 µs the Data lines go high

— The master maintains the Data and Clock line high

— Next Start Condition after $t_{FB(Min)}$ = 4.7 µs is possible

START/STOP CONDITIONS

If both the data and clock lines are HIGH, the bus is not busy. To attain control of the bus, a start condition is needed from a master; and to release the lines, a stop condition is required.

Start Condition: HIGH-to-LOW transition of the data line while the clock line is in a HIGH state.

Stop Condition: LOW-to-HIGH transition of the data line while the clock line is in a HIGH state.

The master always generates the start and stop conditions. After the start condition the bus is in the busy state. The bus becomes free after the stop condition.

DATA BIT TRANSFER

After a start condition “S” one databit is transferred during each clock pulse. The data must be stable during the HIGH-period of the clock. The data line can only change when the clock line is at a LOW level.

Normally each data transfer is done with 8 data bits and 1 acknowledge bit (byte format with acknowledge).

ACKNOWLEDGE

Each data transfer needs to be acknowledged. The master generates the acknowledge clock pulse. The transmitter releases the data line (SDA = HIGH) during the acknowledge clock pulse. If there was no error detected, the receiver will pull down the SDA-line during the HIGH period of the acknowledge clock pulse.

If a slave receiver is not able to acknowledge, the slave will keep the SDA line HIGH and the master can then generate a STOP condition to abort the transfer.

If a master receiver keeps the SDA line HIGH, during the acknowledge clock pulse the master signals the end of data transmission and the slave transmitter release the data line to allow the master to generate a STOP-condition.

ARBITRATION

Only in multimaster systems.

If more than one device are potential masters and more than one desires access to the bus, an arbitration procedure takes place: if a master transmits a HIGH level and another master transmits a LOW level, the master with the LOW level will get the bus and the other master will release the bus; and the clock line switches immediately to the slave receiver mode. This arbitration could carry on through many bits (address bits and data bits are used for arbitration).

FORMATS

There are three data transfer formats supported:

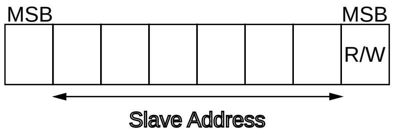
— Master transmitter writes to slave receiver; no direction change

— Master reads immediately after sending the address byte

— Combined format with multiple read or write tranfers.

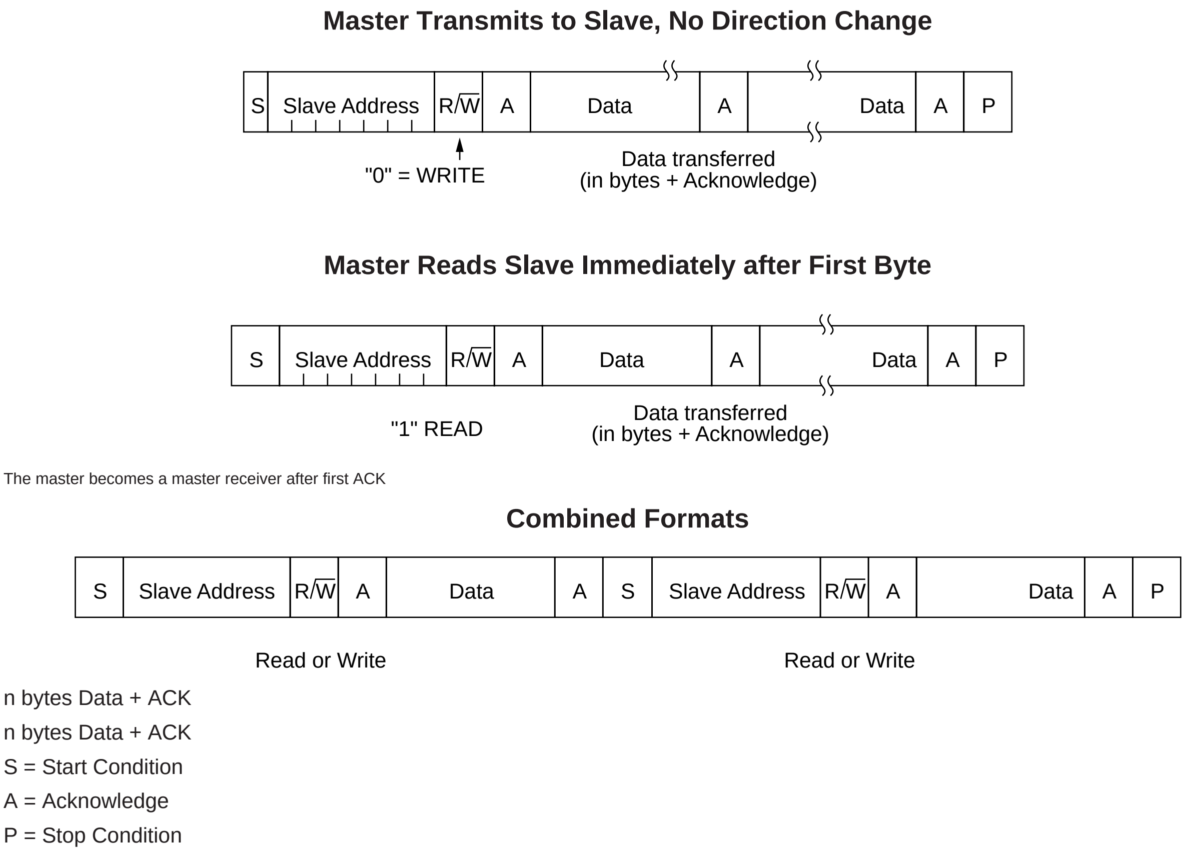
ADDRESSING

The 7-bit address of an IIC device and the direction of the following data is coded in the first byte after the start condition:



A “0” on the least significant bit indicates that the master will write information to the selected Slave address device; a “1” indicates that the master will read data from the slave.

Some slave addresses are reserved for future use. These are all addresses with the bit combinations 1111XXX and 0000XXX. The address 00000000 is used for a general call address, for example, to initialize all I^{2}C devices (refer to I^{2}C bus specification for detailed information).



Master Transmits to Slave, No Direction Change

Master Reads Slave Immediately after First Byte

The master becomes a master receiver after first ACK

Combined Formats

n bytes Data + ACK

n bytes Data + ACK

S = Start Condition

A = Acknowledge

P = Stop Condition

FIGURE 3. IIC-Bus Transfer Formats

TIMING

The master can generate a maximum clock frequency of 100 KHz. The minimum LOW period is defined as 4.7 µs; the minimum HIGH period width is 4 µs; the maximum rise time on SDA and SCL is 1 µs; and the maximum fall time on SDA and SCL is 300 ns. Figure 4 shows the detailed timing requirements.

Symbol	Parameter	Min	Max	Units
f_{SCL}	SCL Clock Frequency	0	100	kHz
t_{BUF}	Time the Bus Must Be Free before a New Transmission Can Start	4.7		µs
$t_{HD:STA}$	Hold Time Start Condition. After this Period the First Clock Pulse is Generated	4.0		µs
t_{LOW}	The LOW Period of the Clock	4.7		µs
$t_{SU:STA}$	Setup Time for Start Condition (Only Relevant for a Repeated Start Condition)	4.7		µs
$t_{HD:DAT}$	Data in Hold Time	5 0 (Note 1)		µs µs
$t_{SU:DAT}$	Setup Time Data	250		ns
t_r	Rise Time of Both SDA and SCL Lines		1	µs
t_f	Fall time of Both SDA and SCL Lines		300	ns
$t_{SU:STO}$	Setup Time for Stop Condition	4.7		µs

Note 1: Note that a transmitter must internally provide at least a hold time to bridge the undefined region (max. 300 ns) of the falling edge of SCL.

FIGURE 4. IIC-Bus Timing Requirements

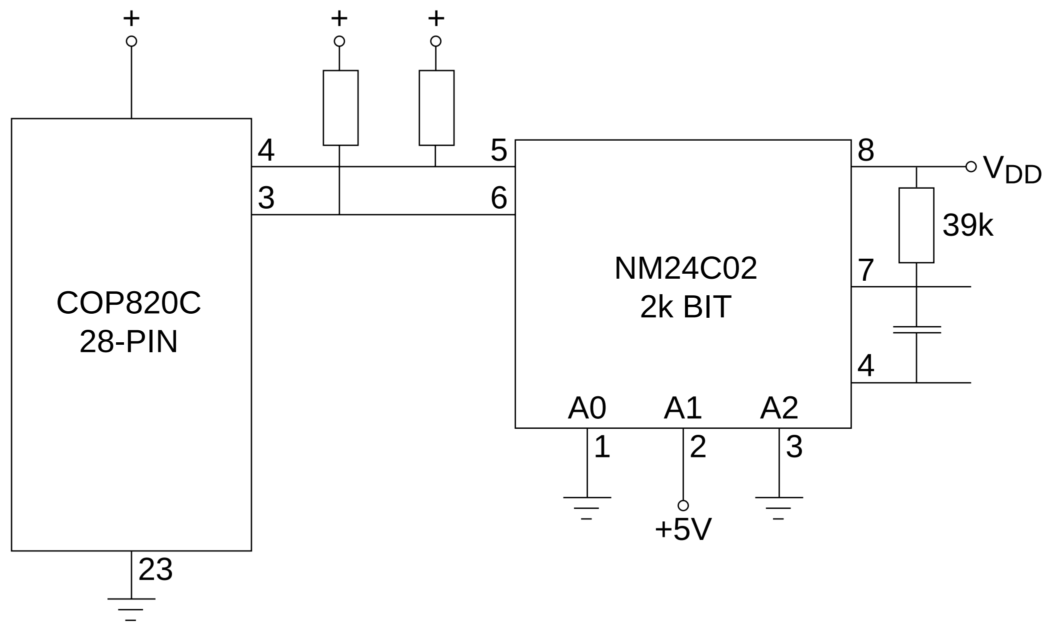


FIGURE 5. IIC Bus EEPROM/µController Configuration Used for Sam ple Code

SOFTWARE TASKS

I. Write fixed values to E^2PROM cells

II. Read values back from E^2PROM and save in RAM locations from COP

Note: IIC Bus Modes Used:

Master	SDA →	Slave Receiver
Transmitter	SCL →	
	← SDA	
Master Receiver		Slave Receiver
	SCL →	

REMARKS

— The IIC bus, 2-wire serial interface generally requires a pull-up resistor on the SDA line and the SCL line, depending on whether TTL or CMOS hardware interfacing exists.

— I IC bus compatible µC's or peripherals have OPEN DRAIN outputs at SDA and SCL.

— COP800 does not have OPEN DRAIN outputs, but the "bus requirements" can be met by switching SDA and SCL connections into TRI-STATE® for the following cases:

The bus is not accessed

A slave has to send an acknowledge bit.

— MICROWIRE can not be used for I^2C bus operations.

— Current sink capability on SDA and SCL must be 3 mA to maintain "Low Level" (an IIC bus spec.).

```
.TITLE IIC - EEPROM ROUTINES
.INCLD COP800.INC
.CHIP 840
.LIST X ‘21
    * * *TASK RELATED RAM - DECLARE* * *
EEADR                                                       = 002       ;ADDRESS OF EEPROM
EEWRD                                                       = 003       ;WORD ADDRESS EEPR.
EEDAT1                                                      = 004       ;DATA TO EECELL
EEDAT2                                                      = 005       ;SECOND BYTE
FLAG                                                        = 010       ;FLAG-WORD
EEREAD                                                      = 012       ;READ-DATA FROM EE
                                                            = 013       ;SECOND BYTE
                                                            = 014       ;THIRD BYTE
                                                            = 015       ;FOURTH BYTE
BITCO                                                       = 0F0       ;COUNTER FOR BITSHFT

INIT:
    LD SP,                                                  #06F
    LD B,                                                   PORTLD      ;INIT LS, LE FOR EE-
    LD [B+],                                                #00C        ;OPERATIONS
    LD [B],                                                 #00C
    LD B,                                                   #EEDAT2     ;INIT RAMS
    LD [B-],                                                #034        ;FIXEED VALUES FOR
    LD [B-],                                                #012        ;EEWRITE (2 BYTES)
    LD [B-],                                                #0A0        ;MIRROR OF #05
    LD [B]                                                  #025        ;MIRROR OF “A5”

        ;* * * * * * * * * * * * * * * * * * * * * * * * * * *
        ;EXAMPLE: IF ADDRESS BYTES IS “1010 01X THEN          *
        ;STORE: “X010 0101                                   *
        ;INTO RAM (X=0/1; WRITE/READ)                         *
        ;* * * * * * * * * * * * * * * * * * * * * * * * * * *

    LD PSW,                                                 #00         ;LOAD PSW
    LD CNTRL,                                               #00         ;AND CNTRL REG.
    LD FLAG,                                                #0          .FORM

        ; * * * * * * * * * * * * * * * * * * * *
        ; * * * *  DO WRITE TO EE-PROM  * * * *
        ; * * * * * * * * * * * * * * * * * * * *

   ;(2 BYTE SUCCESSIVE WRITE)

    SBIT 0,                                                 FLAG        ;SET FLAG FOR WRITE
    LD B,                                                   PORTLD      ;POINT LPORT DAT REG.
                                                                        ;TO MODIFY “SDA, SCL”
    RBIT 2 [B],                                                         ;PREPARE FOR START
    JSR STACON                                                          ;CONDITION.
    JSR WAIT                                                            ;AFTER WRITE TO EE.
                                                                        ;WAIT FOR > THAT 40
        * * * * * * * * * * * * * * * * *
        ;* * DO THE START CONDITION * *
        ;* * AND SHIFT OUT ADRESS * * *
        ;* * BYTE AND WORD-ADRESS * * *
        * * * * * * * * * * * * * * * * *
STACON:
    RBIT 3,                                     PORTLD      ;FINISH START COND.
    LD B,                                       #EEADR      ;PREPARE TO CLOCK
                                                            ;OUT ADDRESS.
LOPA:
    LD BITCO,                                   #008        ;DO SETS OF 8 BITS

LOPA 1:
    IFBIT 0, [B]                                            ;SWITCH SDA BEFORE
    JP ONE,                                                 ;SCL
    RBIT 2,                                     PORTLD      ;SET BIT LEVLE “0”
    JP CLK

ONE:
    SBIT 2,                                     PORTLD      ;SET BIT LEVEL “1”
    JP CLK                                                  ;ENSURE SAME BIT
                                                            ;LENGTH

CLK:
    SBIT 3,                                     PORTLD      ;DO CLOCK PULSE
    NOP
    NOP
    RBIT 3,                                     PORTLD      ;ENSURE>4USEC
    RBIT 2,                                     PORTLD      ;SWITCH ALSO SDA LOW
    .FORM

    LD A, [B]                                               ;ROTATE BYTE ONE
    RRC A,                                                  ;BIT POS. RIGHT
    X A, [B]                                                ;AND SAVE
    DRSZ BITCO                                              ;CHECK IF 8 BITS
    JP LOPA1,                                               ;SHIFTED
    LD A, [B+]                                  FLAG        ;DECREMENT 8
    IFBIT 1,                                    GETDAT      ;CHECK IF READ
    JMP,                                                    ;3RD BYTE IS NEXT?
                                                            ;IF SO, THEN READ.
    JSR ACK,                                                ;GET ACKNOWLEDGED
                                                            ;WHEN 8 BITS ARE
                                                FLAG        ;SHIFTED.
    IFBIT 0,                                                CHECK IF READ.
    JP CEC1                                                 ;OR WRITE OPERATION.
    IFBNE                                       #04         ;ON READ (HERE)
    JMP LOPA                                                ;AFTER EE-ADDRESS AND
    RET                                                     ;WORD ADDRESS ARE SHFT

CEC1:
    IFBNE                                       #06         ;1ST AND 2ND DATA-
    JMP LOPA                                                ;BYTE (3RD + 4TH)
```

```
                                                                       ;NSEC TO PROPERLY
                                                                       ;ERASE WRITE.

LD B,                                                       #EEDAT2    ;INIT RAMS
LD [B-],                                                    #078       ;ANOTHER 2 BYTES
LD [B-],                                                    #056       ;OF FIXED DATA
LD [B-],                                                    #0E0       ;MIRROR OF #07
LD [B],                                                     #025       ;MIRROR OF “A5”
            ;TO MODIFY “SDA, SCL”
RBIT 2, [B],                                                           ;PREPARE FOR START
JSR STACON,                                                            ;CONDITION.
JSR WAIT,                                                              ;AFTER WRITE TO EE.
                                                                       ;WAIT FOR > THAN 40
                                                                       ;MSEC TO PROPERLY
                                                                       ;ERASE WRITE.
.FORM
    ;* * * * * * * * * * * * * * * * * * *
    ;* * * * DO READ FROM EE-PROM * * * *
    ;* * * * * * * * * * * * * * * * * * *
          (READ 4 SUCCESSIVE BYTES)
RBIT 0                                                      FLAG       ;INDICATE READ
LD B,                                                       #EEWRD     ;INIT RAMS
LD [B-],                                                    #0A0       ;MIRROR OF #05
LD [B],                                                     #025       ;MIRROR OF “A5”

    ;* * * * * * * * * * * * * * * * * * * *
    ;* * FIRST 2 BYTES SAME AS IF WRITE * *
    ;* * * * * * * * * * * * * * * * * * * *

                                      (IN TERMS OF TRNSMIT)
LD B,                                                       #PRTLD     ;PREPARE
RBIT 2 [B]                                                             ;FOR
JSR STACON,                                                            ;START COND.
                                                                       ;AND SHIFT 1ST
                                                                       ;2 BYTES
SBIT 2,                                                     PORTLD     ;PREPARE FOR
NOP,                                                                   ;ANOTHER START-
NOP,                                                                   ;CONDITION
SBIT 3,                                                     PORTLD     ;SDA HIGH FIRST.
SBIT 1,                                                     FLAG       ;INDICATE THAT
                                                                       ;3RD BYTE IS NEXT
LD B,                                                       #EEWRD     ;INIT RAMS
LD [B-],                                                    #0A0       ;MIRROR OF #05
LD [B],                                                     #0A5       ;MIRROR OF “A5”
                                                                       ;PERFORM ANOTHER
RBIT 2, [B],                                                PORTLD     ;START
JSR STACON
RBIT 1,                                                     FLAG
JMP INIT                                                               ;CLOSE THE LOOP WHEN
JMP INIT                                                               ;FINISHED
.FORM
```

```
STP:
    SBIT 3,                                                 PORTLD      ;ESTABLISH STOP
    NOP,                                                                ;CONDITION
    SBIT 2,                                                 PORTLD
    RET,
    .FORM

    * * * * * * * * * * * * * * * * * * * *
    ;* *  GET 8BIT OF DATA FROM EE-PROM * *
    * * * * * * * * * * * * * * * * * * * *

GETDAT:
    JSR ACK,                                                            ;GET ACKNOWLEDGEMENT
    LD B,                                                   #EEREAD     ;POINT FIRST READ RAM
    JP                                                      GETDT1      ;AND READ IN

GETDAT:
    JSR ACK,                                                            ;ACKNOWLEDGEMENT TO EE-
                                                                        ;PROM WHEN 8 BITS
                                                                        ;ARE SHIFTED IN.

GETDAT1:
    LD BITCO,                                               #008        ;INIT BIT COUNTER
    RBIT 2,                                                 PORTLC      ;BEFORE READING, PUT
    RBIT 2,                                                 PORTLD      ;‘SDA’ INTO HIGH-Z

LOPB:
    SBIT 3,                                                 PORTLD      ;DO CLOCK HIGH
    RBIT 7, [B]                                                         ;READ IN EEDATA
    IFBIT 2,                                                PORTLD      ;IN SETS OF 8 BITS
    SBIT 7, [B]
    RBIT 3,                                                 PORTLD      ;DO CLOCK LOW
    DRSZ BITCO,                                                         ;CHECK IF 8 BITS
    JP SHFT                                                             ;ARE SHIFTED
    LD A, [B+],                                                         ;INCREMENT B
    IFBNE                                                   #06         ;CHECK IF 4 BYTES
    JMP GETDT,                                              PORTLC      ;PUT L2=0
    SBIT 2,                                                             ;WHEN TRUE, DO STOP
                                                                        ;CONDITION AND
                                                                        ;RETURN

    .FORM

SHFT:
    LD A [B],                                                           ;ROTATE BITS ONE
    RRC A                                                               ;POSITION RIGHT
    X A, [B]
    JP LOPB

    * * * * * * * * * * * * * * * * * * * * * *
    ; * * SIMPLE ROUTING TO DO 40 MSEC DELAY * *
    * * * * * * * * * * * * * * * * * * * * * *
```

```
WAIT:
    LD OF 1                                                 #0.20      ;SIMPLE WAIT LOOP

LOPD:
                                                                       ;TO PRODUCE>40SEC
    LD OF 2,                                                #OFF       ;TIMEOUT

LOPC:
                                                                       ;TO PROPERLY PROGRAM
    DRSZ OF2,                                                          ‘EEPROM. TIME REQUIRED
    SP LOPC,                                                           ;TO ERASE/WRITE
    DRSZOF1,                                                           ;THE EEPART
    JP LOPD
    RET

ACK1:
    SBIT 2                                                  PORTLC     ;INDICATE TO EE-PROM
    JP ACLK                                                            ;(PUT DATA LINE LOW)

ACK:
    RBIT 2,                                                 PORTLC     PUT DATA-LINE HI-Z

ACLK:
                                                                       ;AND GET ACKNOWLEDGE
    SBIT 3,                                                 PORTLD     ;8 BITS ARE SHIFTED,
    NOP                                                                ;DO A DUMMY CLOCK
    NOP
    NOP
    RBIT 3,                                                 PORTLD     ;(FOR ACKNOWLEDGE)

    SBIT 2,                                                 PORTLC
    RET
    .END
```

Life Support Policy

Fairchild's products are not authorized for use as critical components in life support devices or systems without the express written approval of the President of Fairchild Semiconductor Corporation. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Fairchild Semiconductor
Americas
Customer Response Center
Tel. 1-888-522-5372

Fairchild Semiconductor
Europe
Fax: +44 (0) 1793-856858
Deutsch Tel: +49 (0) 8141-6102-0
English Tel: +44 (0) 1793-856856
Français Tel: +33 (0) 1-6930-3696
Italiano Tel: +39 (0) 2-249111-1

Fairchild Semiconductor
Hong Kong
8/F, Room 808, Empire Centre
68 Mody Road, Tsimshatsui East
Kowloon. Hong Kong
Tel; +852-2722-8338
Fax: +852-2722-8383

Fairchild Semiconductor
Japan Ltd.
4F, Natsume Bldg.
2-18-6, Yushima, Bunkyo-ku
Tokyo, 113-0034 Japan
Tel: 81-3-3818-8840
Fax: 81-3-3818-8841

Fairchild does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and Fairchild reserves the right at any time without notice to change said circuitry and specifications.