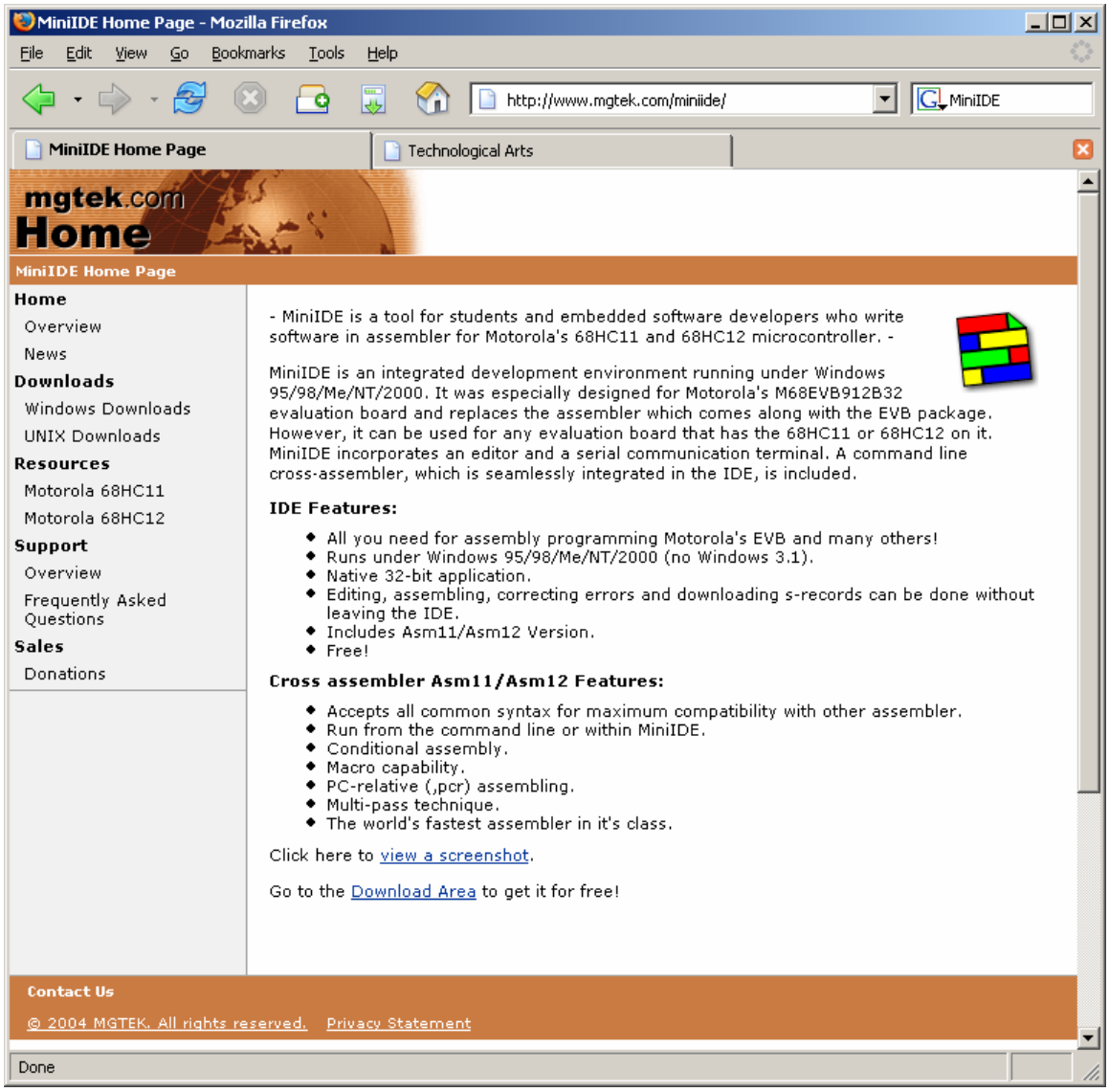
How to use MiniIDE with Adapt912DT60 and FLASH Loader

Download MiniIDE from http://www.mgtek.com/miniide/

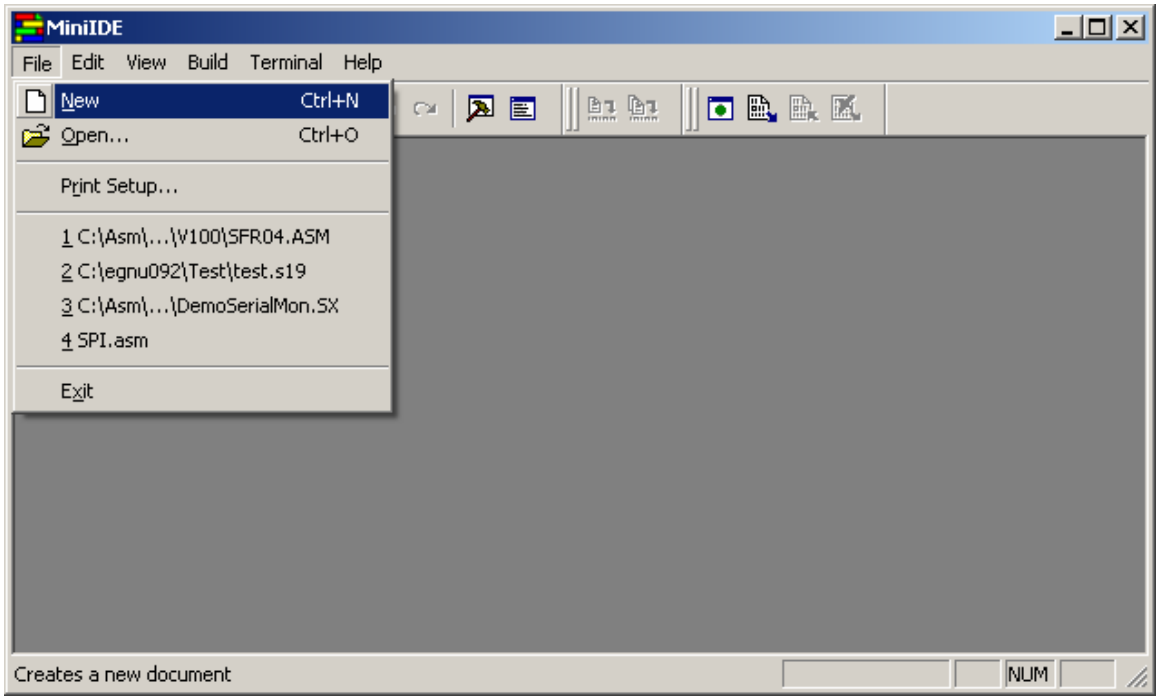


Click on the Download area to select your Operating System (OS) and download MiniIDE.

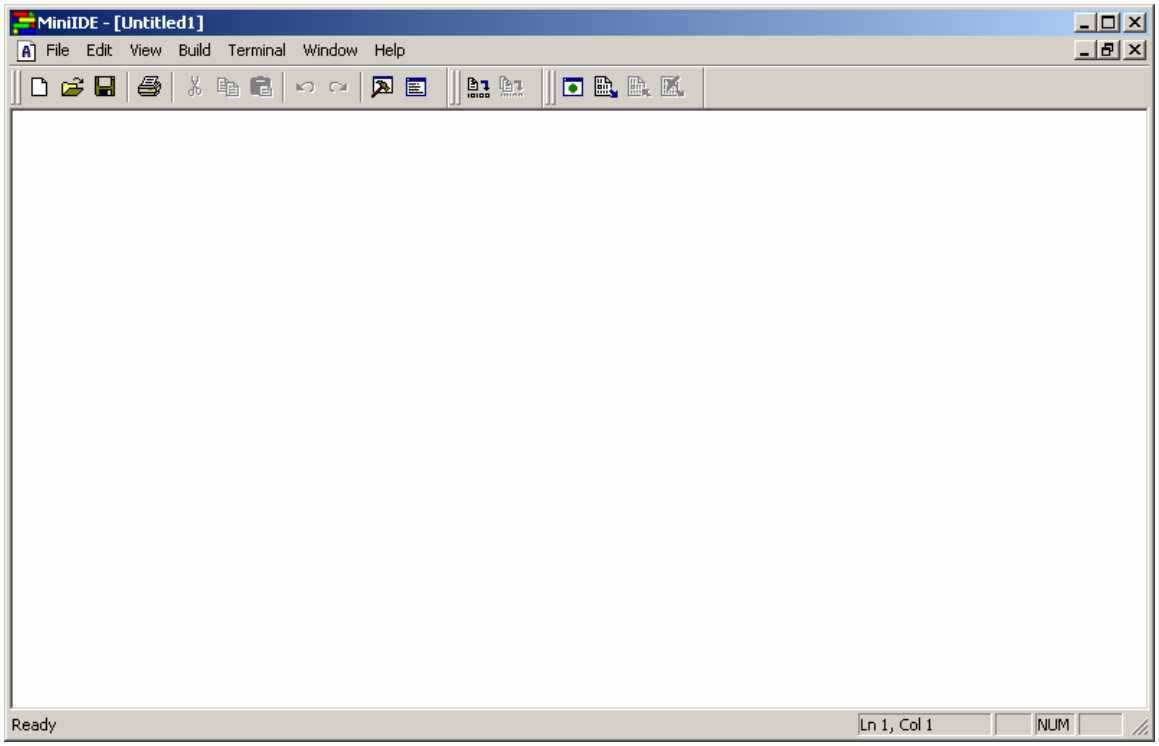
This document assumes that MiniIDE has been installed in your computer.

Getting Started:

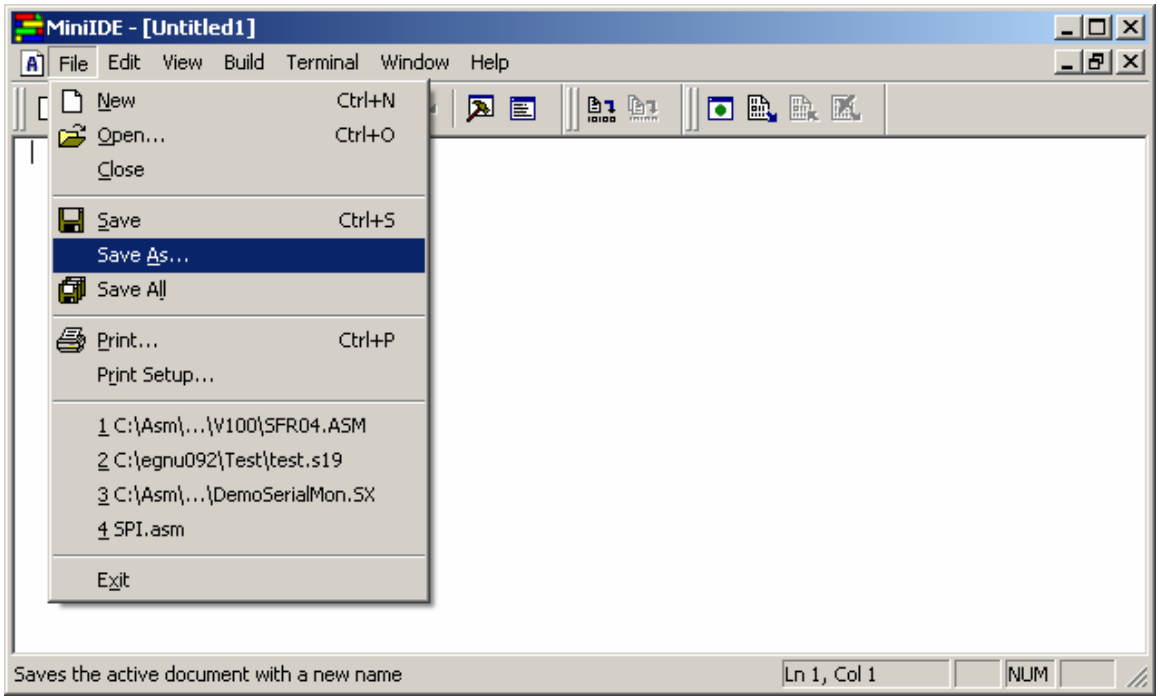
To create a new document click Menu – File - New



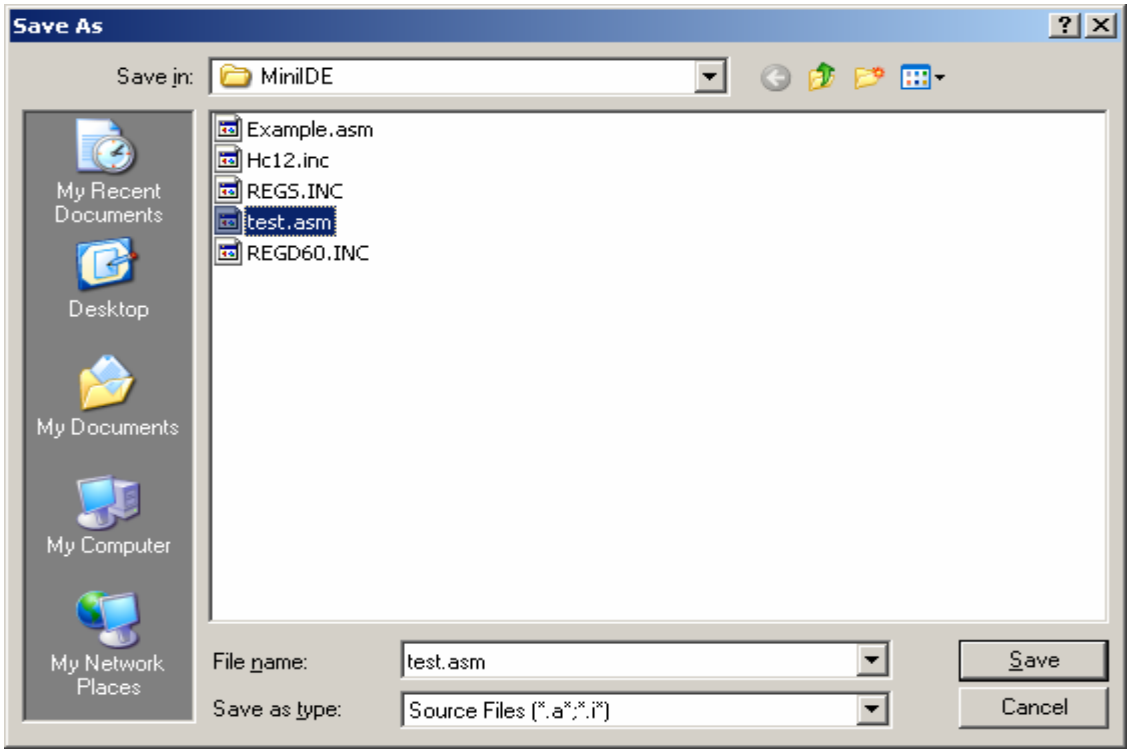
A new ***untitled*** file is created.



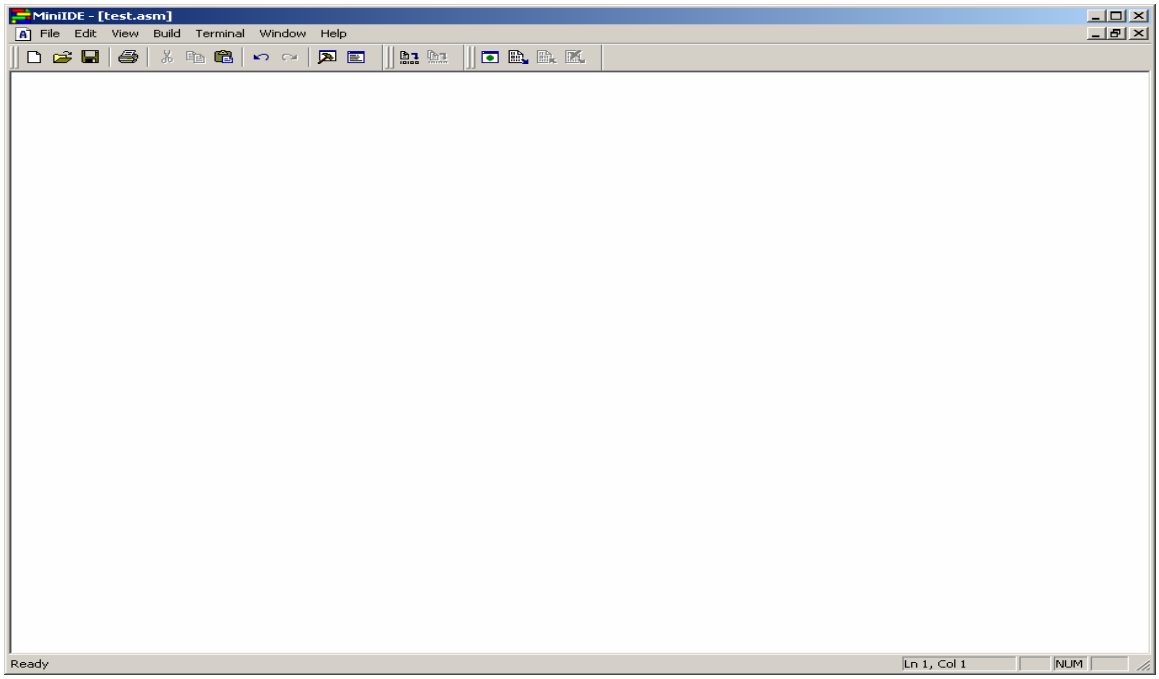
In this example a message "***Hello world***" is sent out thru the SCI port once and LED connected to PORTA bit 6 is toggled continuously. The file is **Save As *test.asm.***



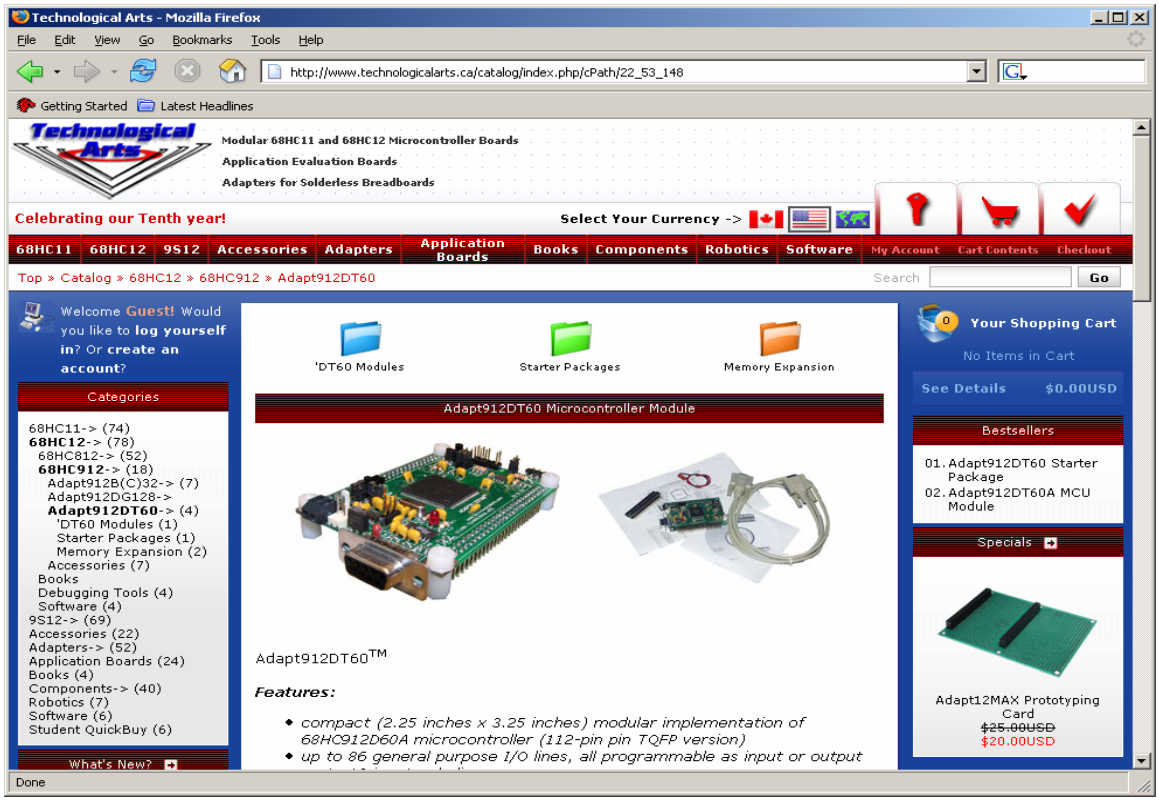
MiniIDE opening an Explorer window to help where the file should be saved to.



MiniIDE has changed the untitled file to ***test.asm***.

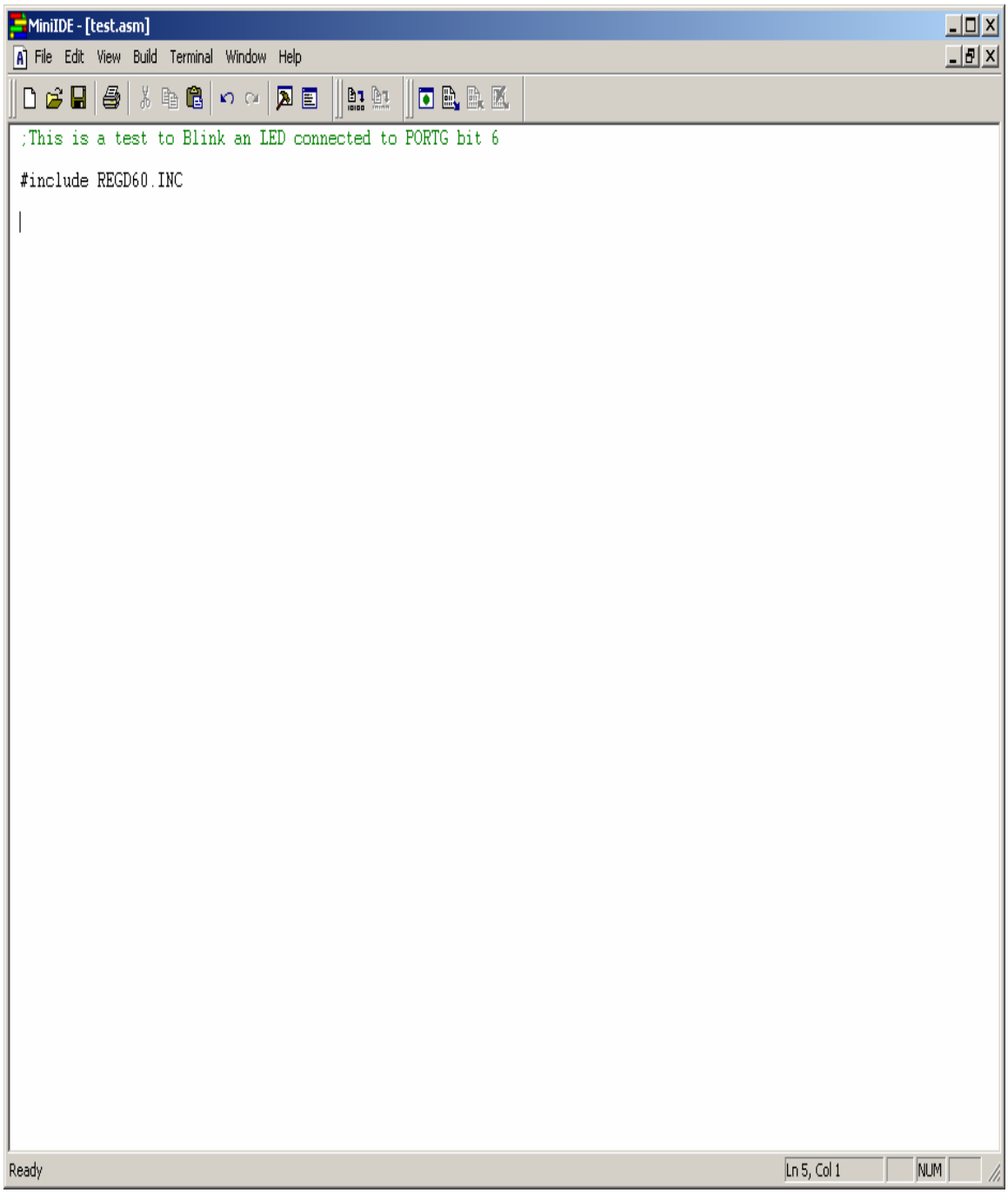


This document will use the Adapt912DT60 or Adapt912DT60A from Technological Arts.
http://www.technologicalarts.ca/catalog/index.php/cPath/22_53_148



In this example, various Register definitions of HC11 are in the include file called ***REGD60.INC***. Generally, these types of files are to be found at www.Freescale.com website. ***REGD60.INC*** is listed at the end of the document.

Also, this document assumes that one is familiar with what are PORTs and Registers. This document will only show how to use ***MiniIDE*** all the way to using ***FLASHLoader*** in programming the Adapt912DT60/A FLASH.



Parameters:

```
* Operational Parameters
RAM      equ      $200            ;68HCD60  internal RAM
STACK    equ      $0800           ;Stack at top of internal ram
EEPROM   equ      $0c00           ;68HC912D60  internal EEPROM
FLASH    equ      $8000           ;68HC912D60  internal FLASH memory
rbase    equ      $0000           ;68HC912D60 register block
CODE     equ      $8000           ;Start of Flash
```

Please note the use of ***equ.*** It simply means a string is equal to a value to connect both the meaning of the string and the value assigned to it. For example,

```
STACK:              equ        $0800              ;At end of RAM
```

Means that STACK = $0800

To define RAM variables by the use of ***ds*** as define segment of a variable. For example below, please note the start of RAM is defined to begin at $0000

```
            Org        RAM

dum         ds         1          ; 1 byte of dummy RAM variable
temp        ds         1          ; another byte of dummy RAM variable
```

Meaning dum = $0200 and temp = $0201.

Below assigns the start of code. For example,

```
            Org        FLASH              ;Start of CODE

ResetFunc:                        ;This is where the RESET vector points to
            sei                           ;Disable Any interrupts
```

The is assigned to start at $8000 as defined by

```
FLASH:              equ        $8000              ;for 32K upper half
```

Type the rest of the codes below and once that is done, the code can be assembled or build.

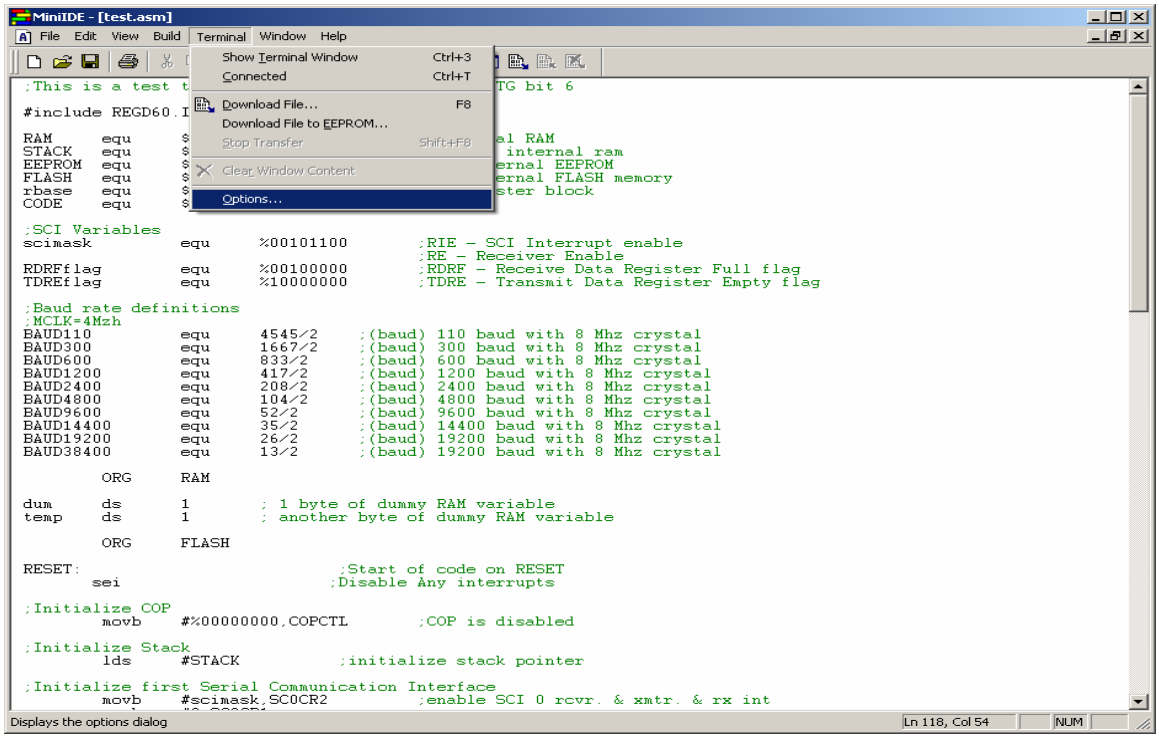
```
;This is a test to Blink an LED connected to PORTG bit 6

#include REGD60.INC

RAM      equ      $200            ;68HCD60  internal RAM
STACK    equ      $0800           ;Stack at top of internal ram
EEPROM   equ      $0c00           ;68HC912D60  internal EEPROM
FLASH    equ      $8000           ;68HC912D60  internal FLASH memory
rbase    equ      $0000           ;68HC912D60 register block
CODE     equ      $8000           ;Start of Flash
```

```
;SCI Variables
scimask         equ     %00101100       ;RIE - SCI Interrupt enable
                                        ;RE - Receiver Enable
RDRFflag        equ     %00100000       ;RDRF - Receive Data Register
Full flag
TDREflag        equ     %10000000       ;TDRE - Transmit Data Register
Empty flag

;Baud rate definitions
;MCLK=4Mzh
BAUD110         equ     4545/2    ;(baud) 110 baud with 8 Mhz crystal
BAUD300         equ     1667/2    ;(baud) 300 baud with 8 Mhz crystal
BAUD600         equ     833/2     ;(baud) 600 baud with 8 Mhz crystal
BAUD1200        equ     417/2     ;(baud) 1200 baud with 8 Mhz crystal
BAUD2400        equ     208/2     ;(baud) 2400 baud with 8 Mhz crystal
BAUD4800        equ     104/2     ;(baud) 4800 baud with 8 Mhz crystal
BAUD9600        equ     52/2      ;(baud) 9600 baud with 8 Mhz crystal
BAUD14400       equ     35/2      ;(baud) 14400 baud with 8 Mhz crystal
BAUD19200       equ     26/2      ;(baud) 19200 baud with 8 Mhz crystal
BAUD38400       equ     13/2      ;(baud) 19200 baud with 8 Mhz crystal

        ORG     RAM

dum     ds      1       ; 1 byte of dummy RAM variable
temp    ds      1       ; another byte of dummy RAM variable

        ORG     FLASH

RESET:                          ;Start of code on RESET
       sei                     ;Disable Any interrupts

;Initialize COP
        movb    #%00000000,COPCTL       ;COP is disabled

;Initialize Stack
        lds     #STACK          ;initialize stack pointer

;Initialize first Serial Communication Interface
        movb    #scimask,SC0CR2         ;enable SCI 0 rcvr. & xmtr. & rx
int
        movb    #0,SC0CR1
;
        movw    #BAUD9600,SC0BDH        ;Set baud rate to 9600
        ldab    SC0SR1                  ;read register to clear flag
RDRF
        ldab    SC0DRL                  ;read receive buffer

;Initialize Port G bit 6
        ldaa    #$FF
        staa    DDRG

        ldx     #HelloMsg               ;Send message
        jsr     OutStr

main
        com     PORTG
        bsr     delay

        bra     main


;---------------------------------------------------------------
delay
        ldx     #0
```

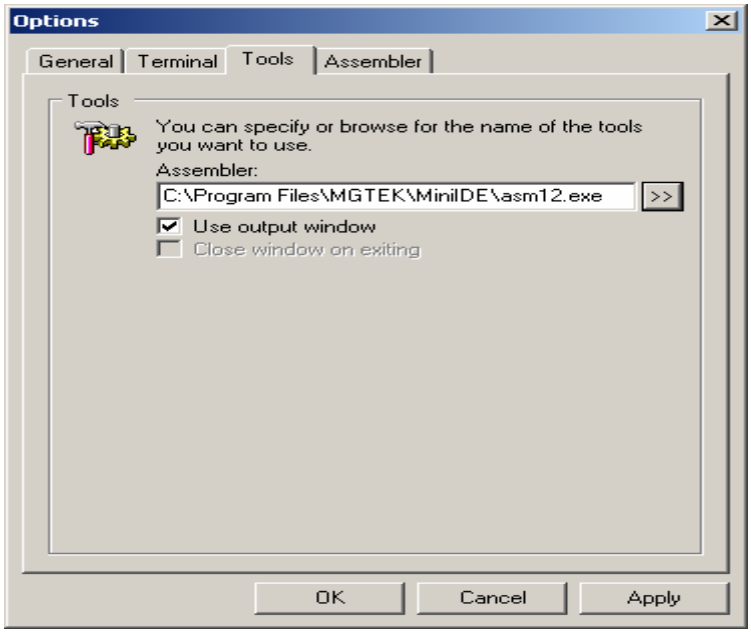
```
delay_loop
        dbne    x,delay_loop
        rts


;---------------------------------------------------------------
OutStr
        ldaa    1,x+            ;get a char, advance pointer, null?
        beq     OutStrDone      ;yes, return

        bsr     putchar         ;no, sent it out
        bra     OutStr          ;go send next char

OutStrDone
        rts

putchar
        brclr   SC0SR1,TDREflag,*  ;loop waitin for the TDRE bit to be
set
        staa    SC0DRL             ;send char
        rts

;---------------------------------------------------------------------------
HelloMsg         dc.b     $0a,$0d,'Hello World',0

;---------------------------------------------------------------------------

PsuedoVECTOR  EQU     $DFC2

        org     PsuedoVECTOR

        FDB     RESET           ;30 - CGM PLL
        FDB     RESET           ;29 - MSCAN1 TRANSMIT
        FDB     RESET           ;28 - MSCAN0 RECEIVE
        FDB     RESET           ;27 - MSCAN0 ERROR
        FDB     RESET           ;26 - Pulse Accumulator B overflow
        FDB     RESET           ;25 - Modulus Down Counter Underflow
        FDB     RESET           ;24 - KEY WAKEUP FOR J OR H
        FDB     RESET           ;23 - MSCAN0 WAKE UP
        FDB     RESET           ;22 - ANALOG TO DIGITAL
        FDB     RESET           ;21 - SERIAL COMMUNICATION 1
        FDB     RESET    ;20 - SERIAL COMMUNICATION 0
        FDB     RESET           ;19 - SPI SERIAL TRANSFER COMPLETE
        FDB     RESET           ;18 - PULSE ACCUMULATOR INPUT EDGE
        FDB     RESET           ;17 - PULSE ACCUMULATOR OVERFLOW
        FDB     RESET           ;16 - TIMER OVERFLOW
        FDB     RESET           ;15 - TIMER CHANNEL 7
        FDB     RESET          ;14 - TIMER CHANNEL 6
        FDB     RESET           ;13 - TIMER CHANNEL 5
        FDB     RESET           ;12 - TIMER CHANNEL 4
        FDB     RESET           ;11 - TIMER CHANNEL 3
        FDB     RESET           ;10 - TIMER CHANNEL 2
        FDB     RESET           ;9 - TIMER CHANNEL 1
        FDB     RESET           ;8 - TIMER CHANNEL 0
        FDB     RESET     ;7 - REAL TIME INTERRUPT
        FDB     RESET           ;6 - IRQ
        FDB     RESET           ;5 - XIRQ
        FDB     RESET       ;4 - SWI
        FDB     RESET           ;3 - RESERVED
        FDB     RESET           ;2 - COP FAILURE RESET
        FDB     RESET           ;1 - COP CLOCK MONITOR FAIL RESET
        FDB     RESET          ;0 - RESET
```

Assemble or Build a file:

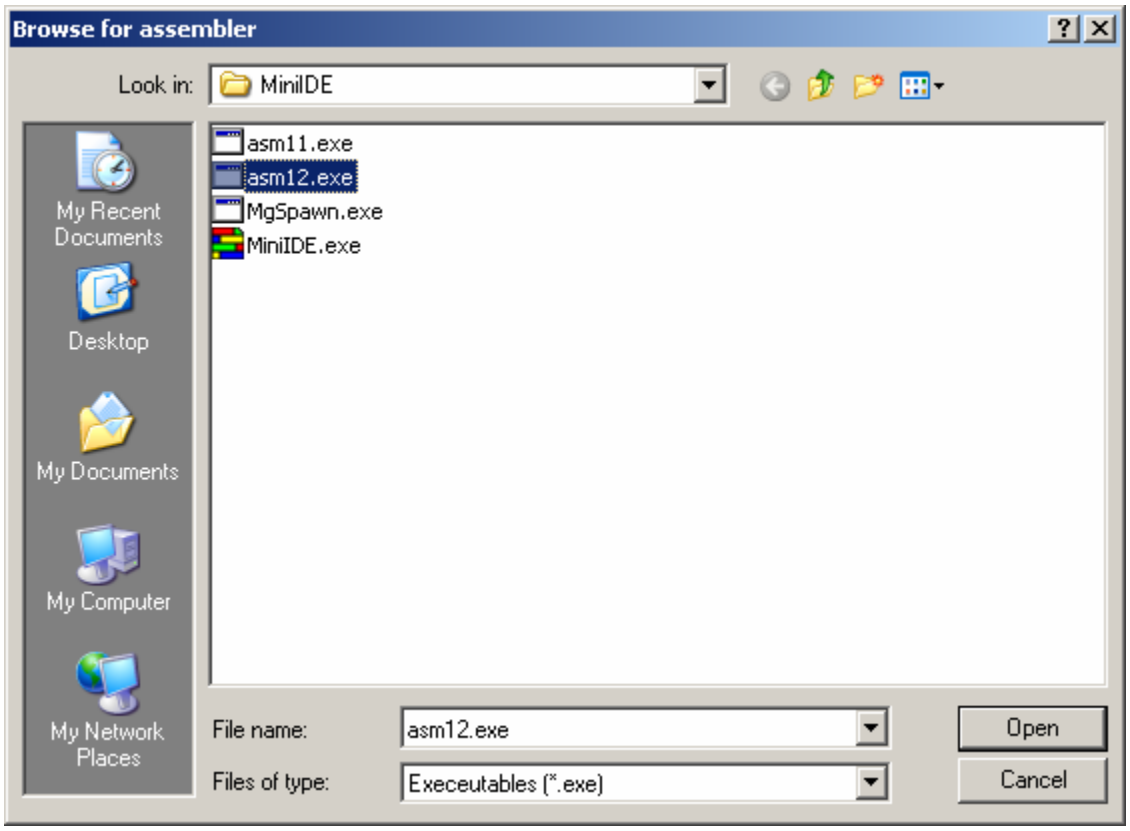
The first thing to do is check the options to make sure it is set for HC12 assembler. Click on Terminal Menu – Options and then Tools tab.



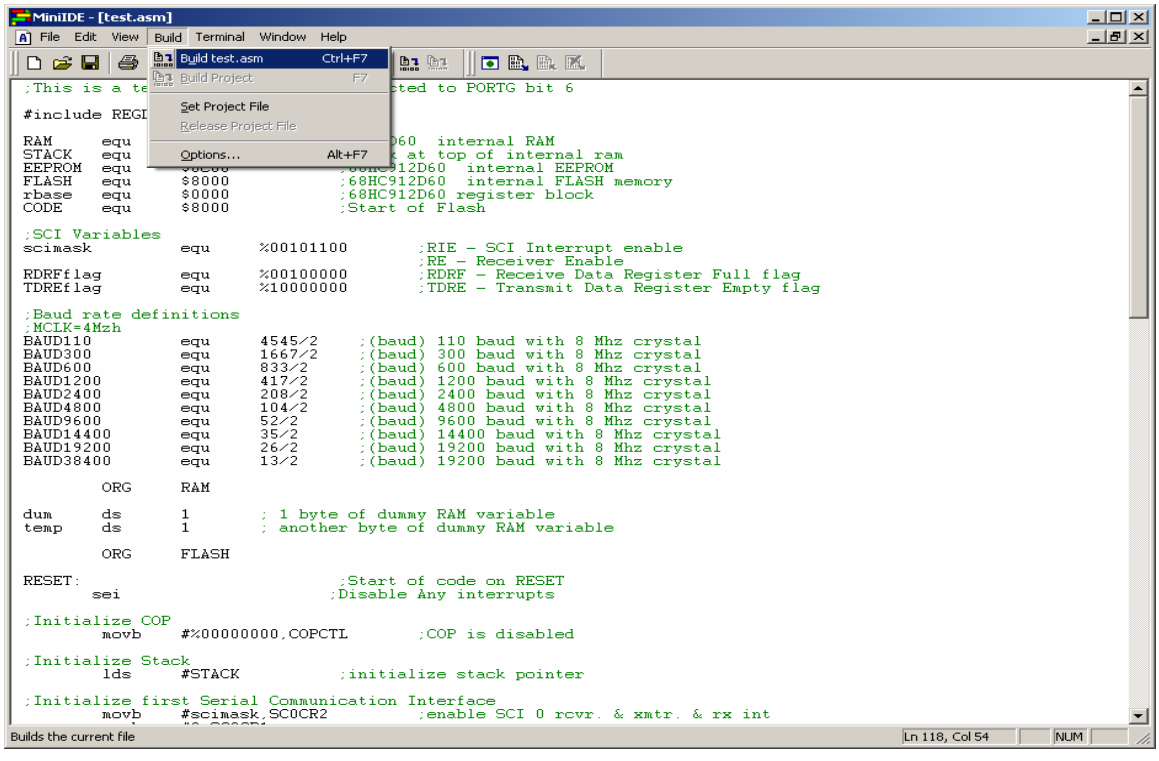
Make sure to select and use the ***asm12.exe*** as the assembler for HC12 MCUs.



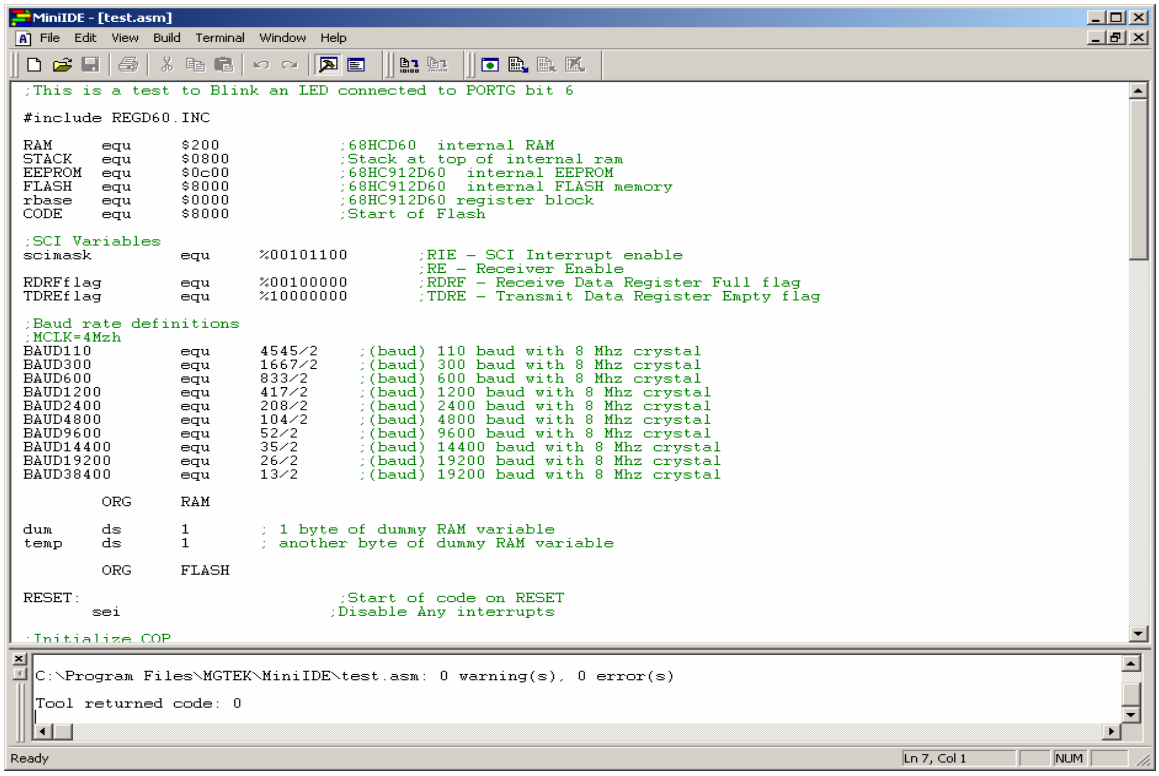
As one can note, the ***asm12.exe*** are for HC12 MCUs.



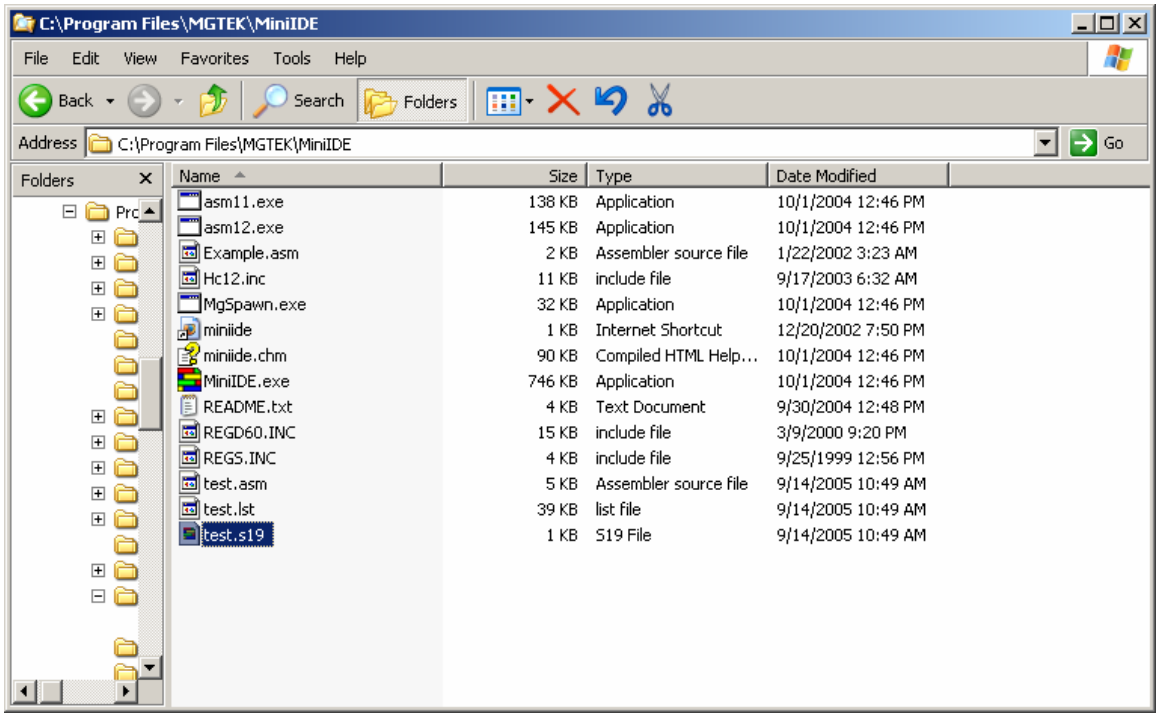
To build the file, select Build menu – Build ***test.asm*** as shown.



After the build – Please check for any error(s).

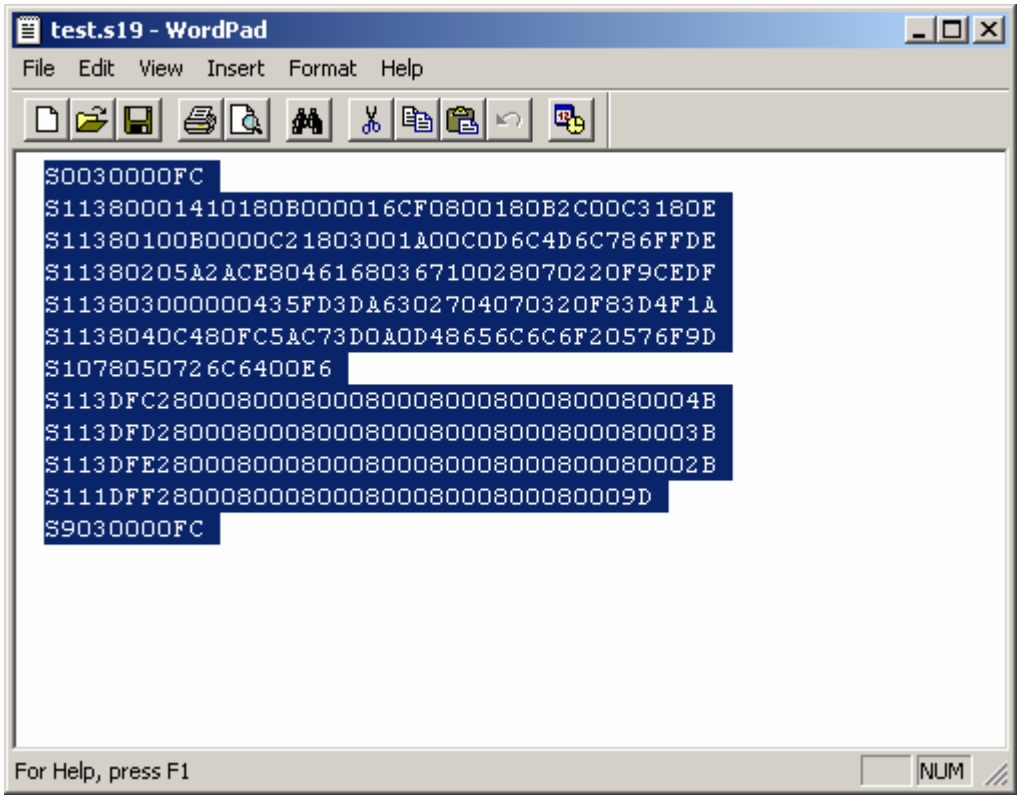


Locate the test.s19 record to examine the generated code.



Examining S-record:

Use wordpad to open the S-record as shown.



If one looks closely at the S-record one can see S1 to be of different lengths. This is atypical S-record generated by ICC12. S1 records are programmed in the ***$1000 - $DFFF*** memory blocks.

As stated previously, FLASH Loader occupies $E000 to $FFFF therefore the vector address at below $E000.

Note the content of the memory address at $DFFE:$DFFF is $8000, the pseudo vector pointing to the start of code to begin at $8000

S113DFC280008000800080008000800080008004B
S113DFD280008000800080008000800080008003B
S113DFE280008000800080008000800080008002B
S111DFF28000800080008000800080008000***8000***9D

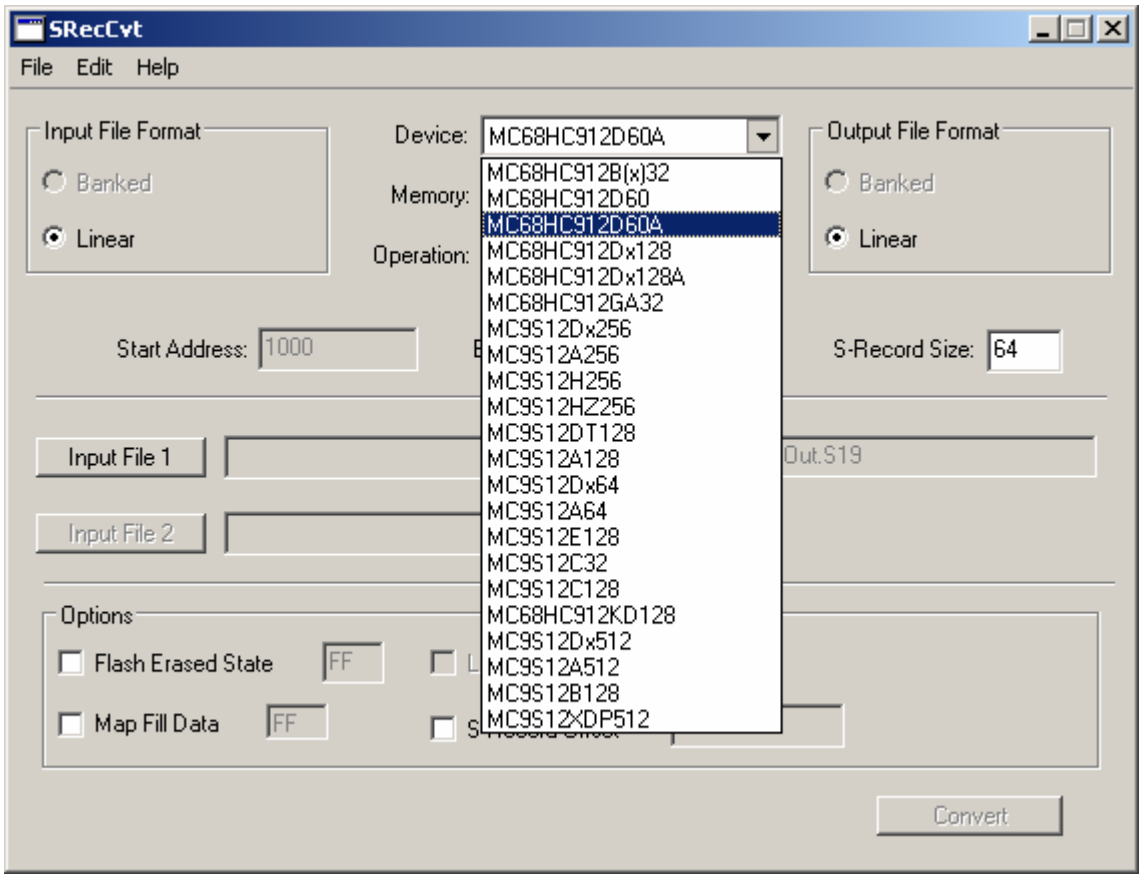
The S-record below is the start of code. The content of address beginning at $8000 to $8050.

S113***8000***1410180B000016CF0800180B2C00C3180E
S11380100B0000C21803001A00C0D6C4D6C786FFDE
S11380205A2ACE80461680367100280702 20F9CEDF
S113803000000435FD3DA6302704070320F83D4F1A
S1138040C480FC5AC73D0A0D48656C6C6F20576F9D
S1078050726C6400E6

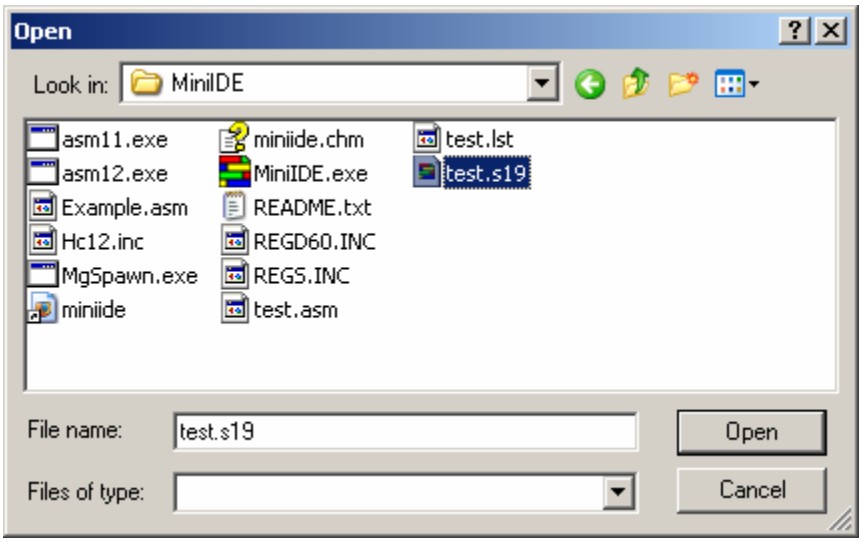
For the 912D60A, the S- records needs to be formatted for 64 bytes lengths. The SrecCVT program can be downloaded at www.freescale.com website. The actual file is locate at this link.
http://www.freescale.com/files/soft_dev_tools/software/app_software/dbug_rom_monitors/DB12S12FW.zip

Unzip the file to locate a folder ***SRecCvt-GUI***. This folder contains 3 files. The ***SRecCvt-GUI.exe*** will be used to re-format the s-record. Double click on ***SRecCvt-GUI.exe*** to execute GUI.

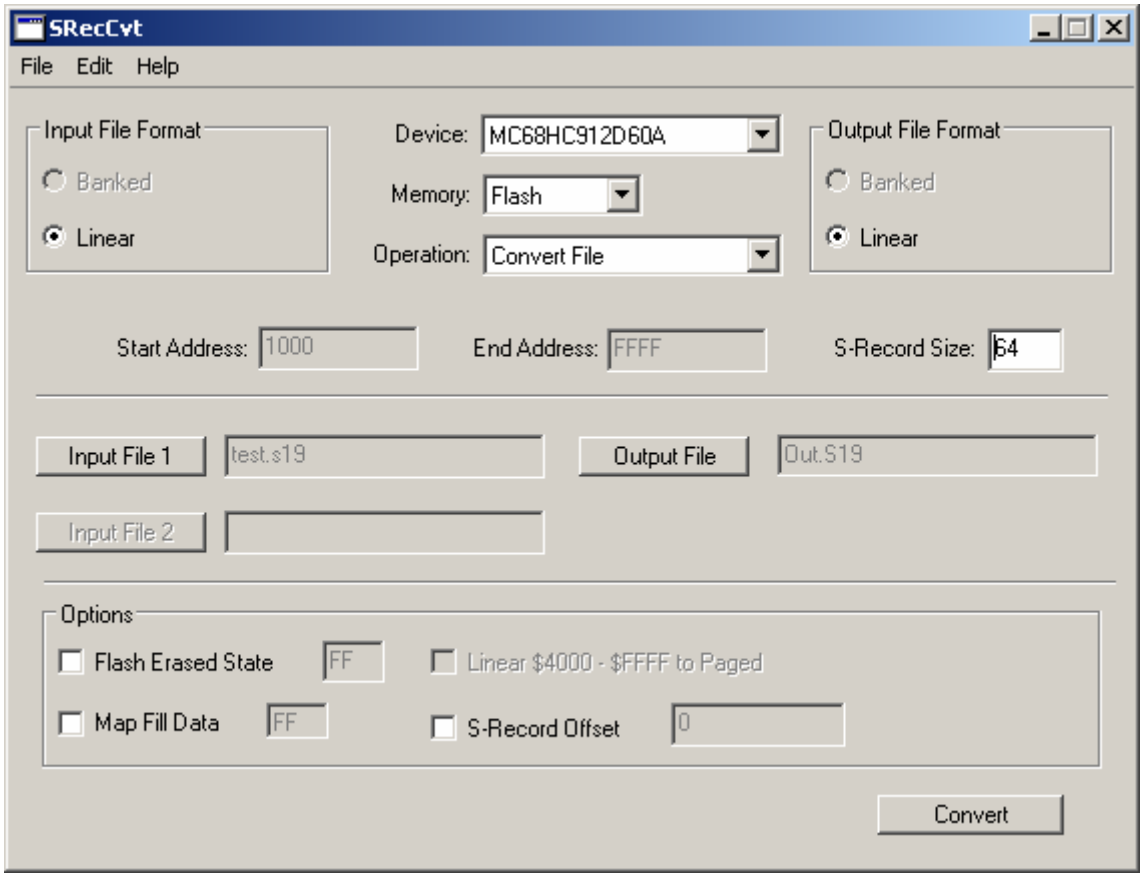
On the Device pulldown select MC68HC912D60A as shown.



Click on the Input File 1 button to help locate ***test.s19***. An explorer window will open to help locate said file as shown.

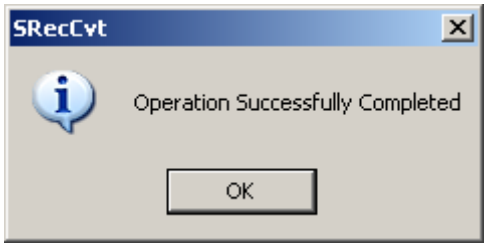


Click on ***test.s19*** then press Open button to accept selection as shown. Note that ***Out.s19*** is the default generated output file of the ***SrecCVT***.



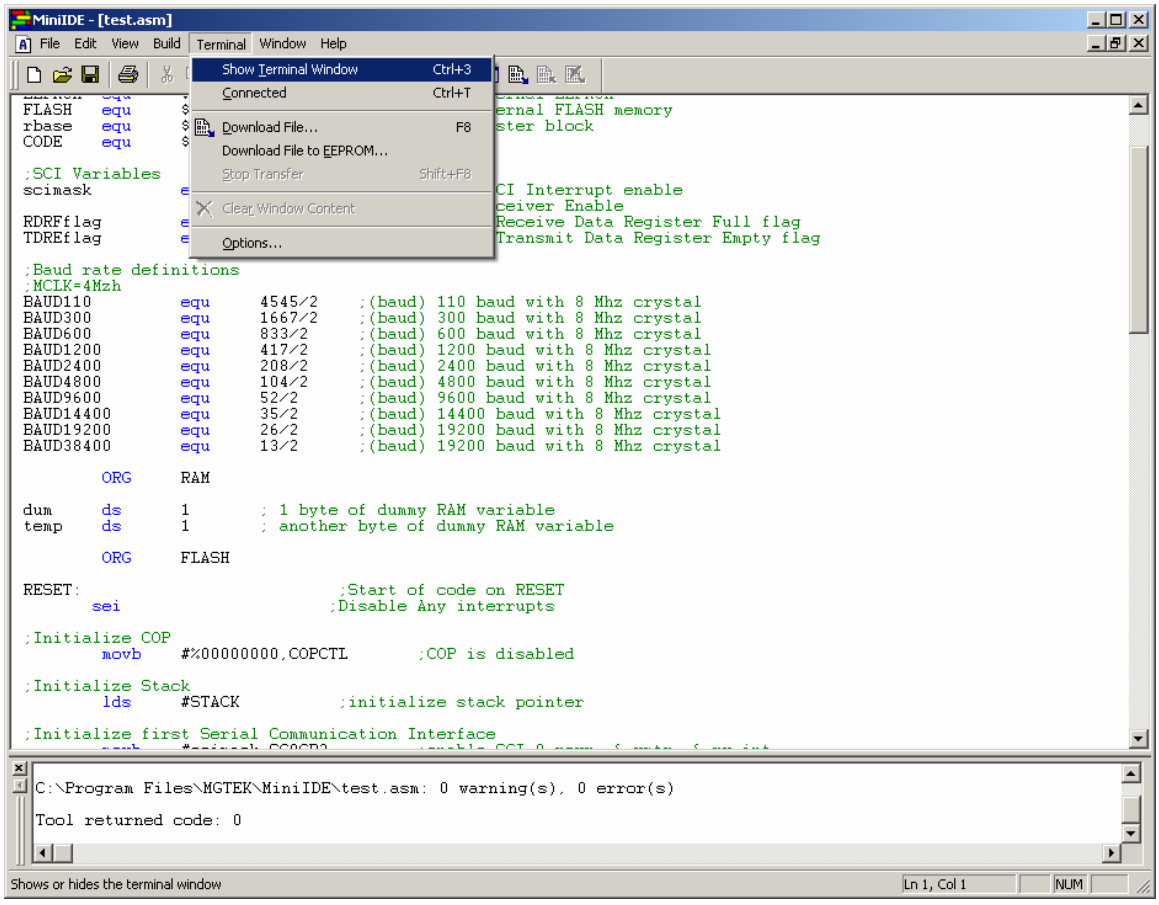
To Convert:

Initiate Conversion by pressing the ***Convert*** button. The message “Operation Successfully Completed” will pop-up as shown.

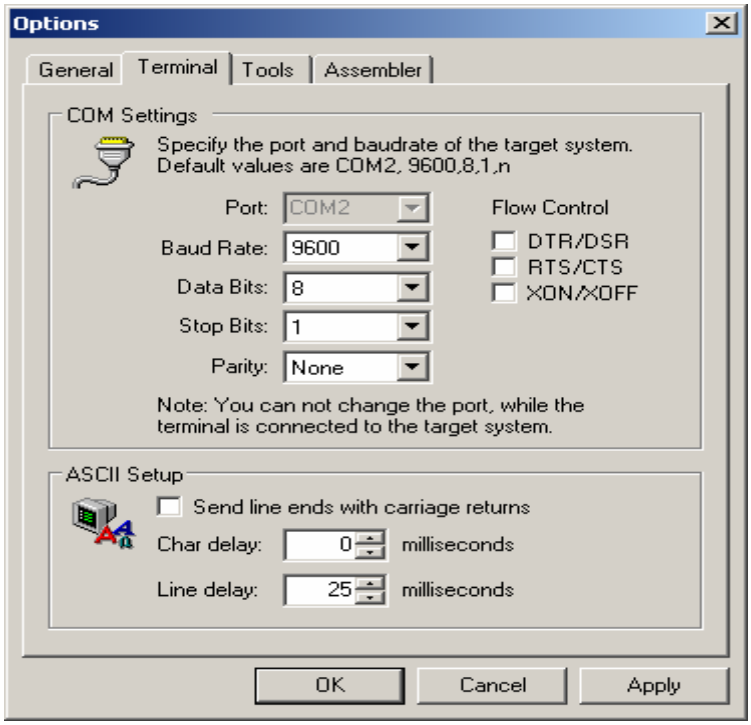


Programming the Adapt912DT60/A:

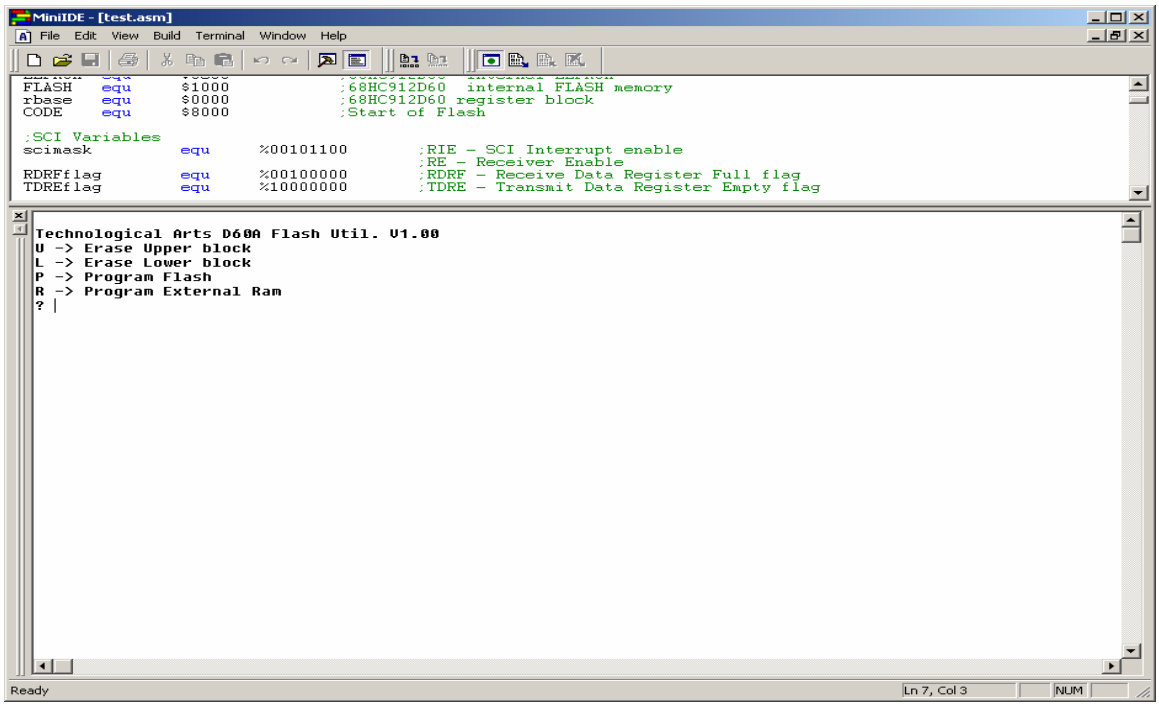
Once the S-record is formatted, time now to program the MCU. Open the MiniIDE terminal window. Click on Terminal Menu and ***Select Show Terminal*** as shown.



Under Terminal Option the BAUD, COM and Line Delay needs to be setup as shown. Choose which COM are available for use. Set Line delay = 25 as shown.



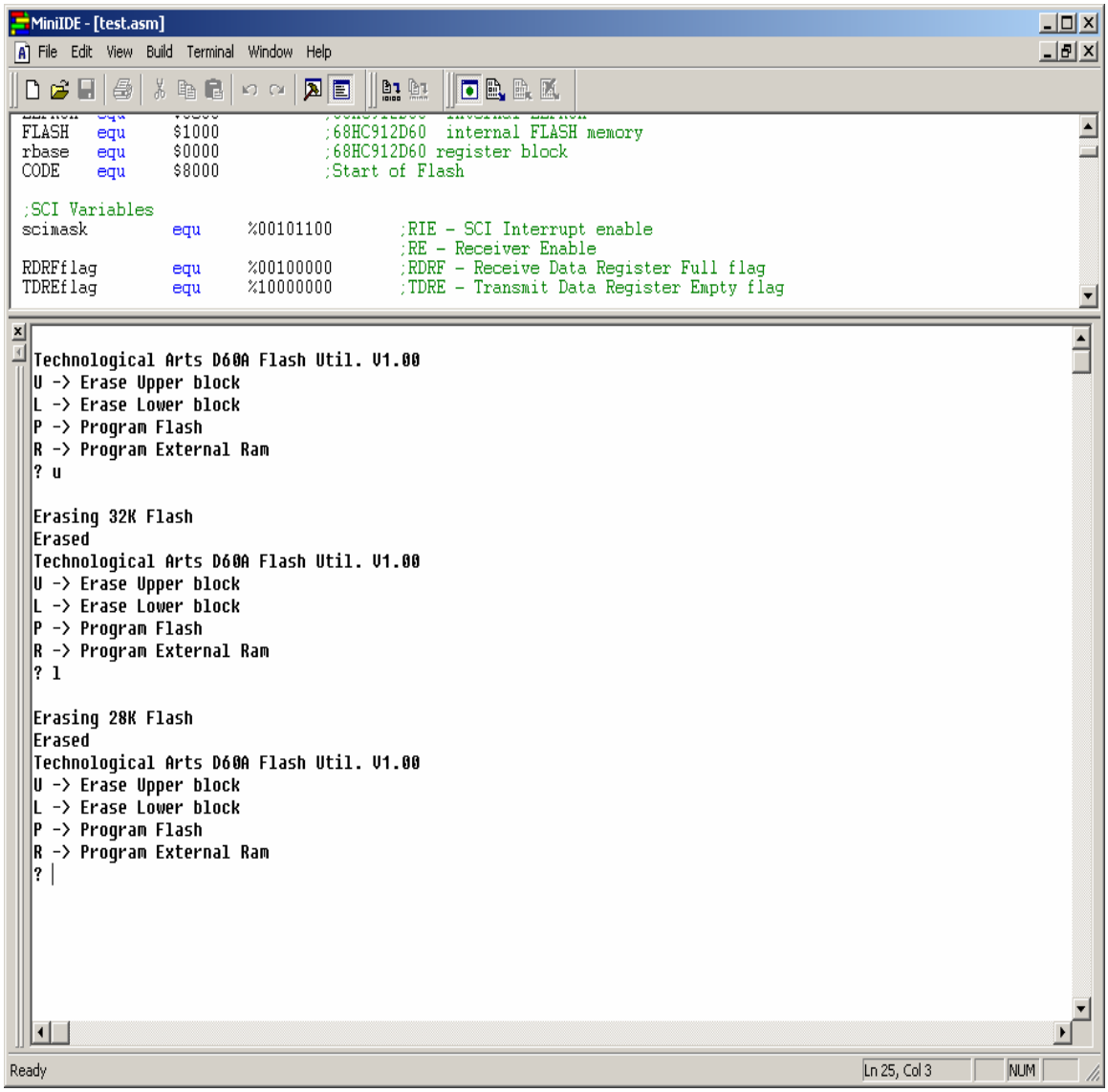
Move SW2 Load/Run switch to Load. Power up unit or press the RESET button if already powered up.



Please note the in the 912D60/A the Flash modules are in 2 location. One is below $8000, beginning at $1000 to $7FFF (Lower block). The other is at $8000 to $FFFF (upper block). The FlashLoader allows one to erase either Lower or upper memory blocks. Since the code starts from $1000 and the Psuedo vector at $DFD0, it is important that both blocks be erased.

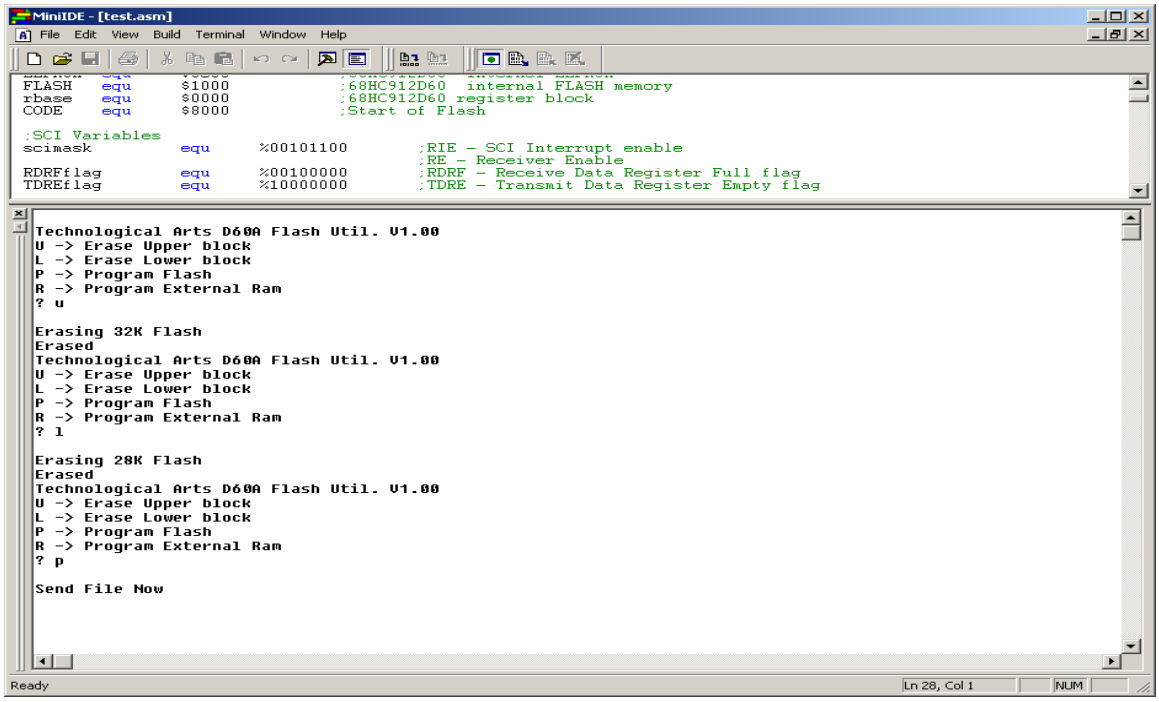
Erase command:

The command to erase upper memory block is the letter ***U*** and ***L*** for lower the memory block as shown. Here the upper and lower blocks are erased.

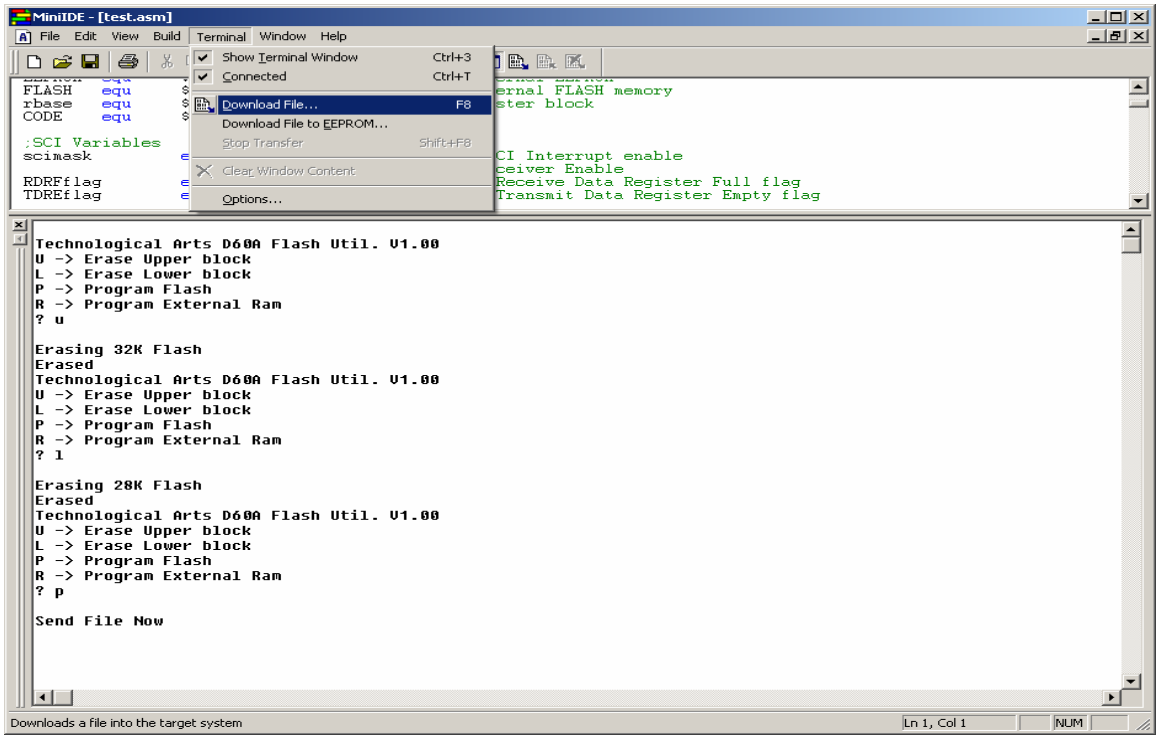


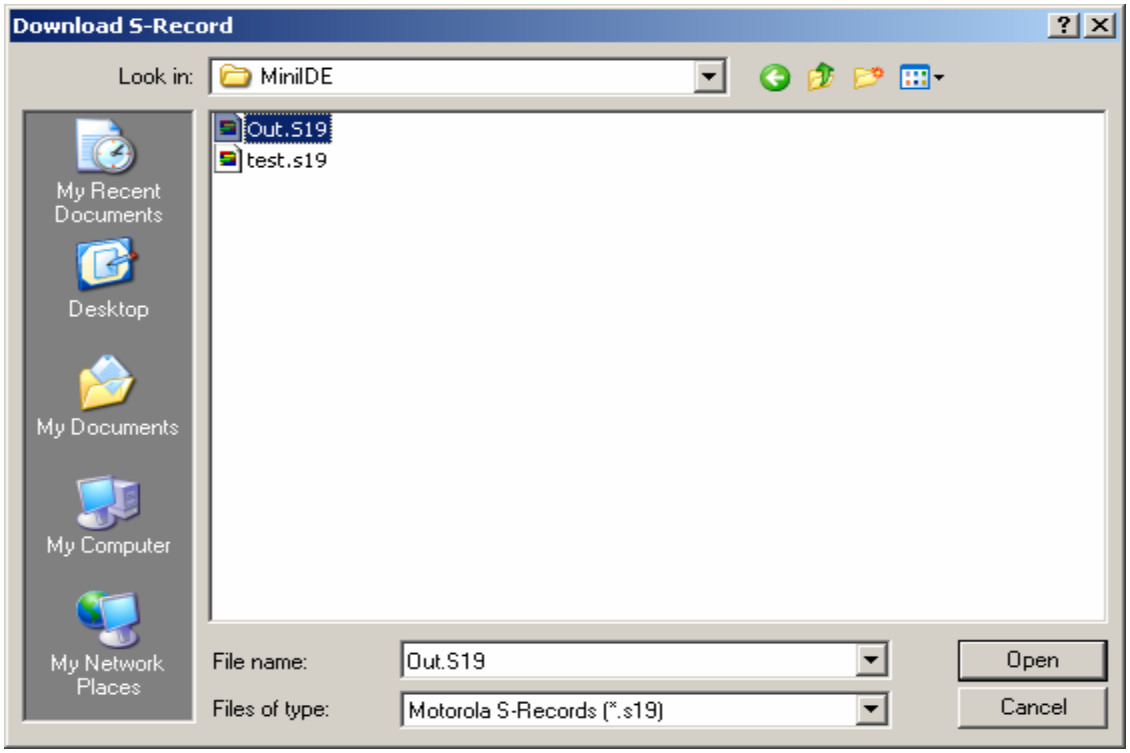
Programming:

To program select P command as shown.

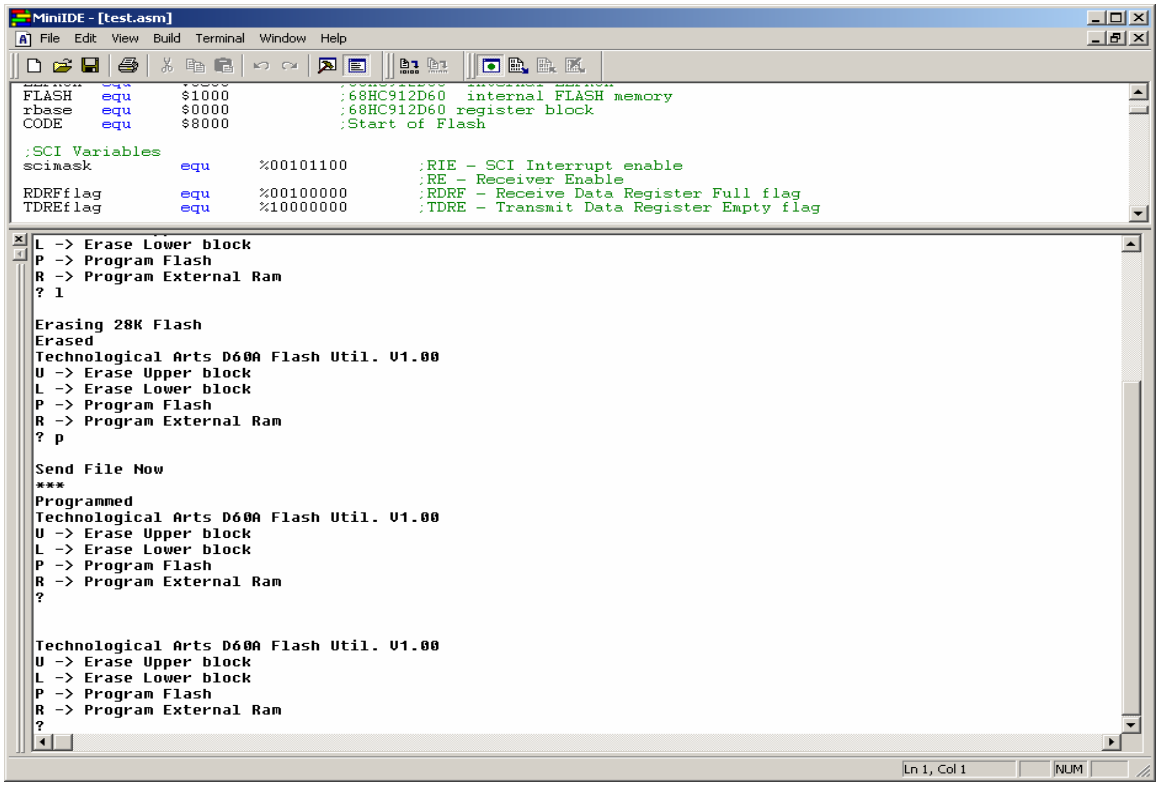


Under the Terminal menu select Download File, an explorer window will pop up to help locate the ***out.s19*** file. Select ***out.s19*** then press the ***open*** button.





Once the correct file is selected, press the Download button to initiate upload to Adapt912DT60. Character *** will appear and the message ***Programmed*** will be displayed to show the MCU was programmed.



Move SW2 Run/Load switch to ***Run*** and press the RESET button. The LED on the ***Adapt912DT60/A*** that is connected to ***PG7*** will begin to blink.

This concludes the use of MiniIDE with Adapt912DT60/A to using the FLASH Loader.

REG60.INC

```
;MC68HC912D60 REGISTER MAP
;MEMORY BLOCK BEGINS AT 0000H AND ENDS AT 1FFH

REG     EQU     $0000

PORTA   EQU     $0000          ;PORTA
PORTB   EQU     $0001          ;PORTB
DDRA    EQU     $0002          ;PORTA - DATA DIRECTION REGISTER
DDRB    EQU     $0003          ;PORTB - DATA DIRECTION REGISTER

PORTE   EQU     $0008          ;PORTE
DDRE    EQU     $0009          ;DDRE - DATA DIRECTION REGISTER
PEAR    EQU     $000A          ;PEAR -
MODE    EQU     $000B          ;MODE - MODE REGISTER
PUCR    EQU     $000C          ;PUCR - PULL UP CONTROL REGISTER
RDRIV   EQU     $000D          ;RDRIV - REDUCED DRIVE OF I/O LINES

INITRM  EQU     $0010          ;INITRM - INITIALIZATION OF INTERNAL RAM
POSITION REGISTER
INITRG  EQU     $0011          ;INITRG - INITIALIZATION OF INTERNAL
REGISTER POSITION REGISTER
INITEE  EQU     $0012          ;INITEE - INITIALIZATION OF INTERNAL
EEPROM POSITION REGISTER
MISC    EQU     $0013          ;MISC - MISCELLANEOUS MAPPING CONTROL
REGISTER
RTICTL  EQU     $0014          ;RTICTL - REAL TIME INTERRUPT CONTROL
REGISTER
RTIFLG  EQU     $0015          ;RTIFLG - REAL TIME INTERRUPT FLAG
REGISTER
COPCTL  EQU     $0016          ;COPCTL - COP CONTROL REGISTER
COPRST  EQU     $0017          ;COPRST - ARM/RESET COP TIMER
REGISTER

ITST0   EQU     $0018          ;ITST0
ITST1   EQU     $0019          ;ITST1
ITST2   EQU     $001A          ;ITST2
ITST3   EQU     $001B          ;ITST3

INTCR   EQU     $001E          ;INTCR - INTERRUPT CONTROL REGISTER
HPRIO   EQU     $001F          ;HPRIO - HIGHEST PRIORITY I INTERRRUPT

BRKCT0  EQU     $0020          ;BRKCT0
BRKCT1  EQU     $0021          ;BRKCT1
BRKAH   EQU     $0022          ;BRKAH
```

```
BRKAL   EQU     $0023         ;BRKAL
BRKDH   EQU     $0024         ;BRKDH
BRKDL   EQU     $0025         ;BRKDL

PORTG   EQU     $0028         ;PORTG
PORTH   EQU     $0029         ;PORTH
DDRG    EQU     $002A         ;DDRG - PORTG DATA DIRECTION
DDRH    EQU     $002B         ;DDRH - PORTH DATA DIRECTION
KWIEG   EQU     $002C         ;KWIEJ - KEY WAKEUP PORTG INTERRUPT
ENABLE REGISTER
KWIEH   EQU     $002D         ;KWIEH - KEY WAKEUP PORTH INTERRUPT
ENABLE REGISTER
KWIFG   EQU     $002E         ;KWIFJ - KEY WAKEUP PORTG FLAG
REGISTER
KWIFH   EQU     $002F         ;KWIFH - KEY WAKEUP PORTH FLAG
REGISTER
KWPJ    EQU     $0030         ;KWPJ -
KWPH    EQU     $0031         ;KWPH -

SYNR    EQU     $0038         ;SYNR -
REFDV   EQU     $0039         ;REFDV -
CGTFLG  EQU     $003A         ;CGTFLG -
PLLFLG  EQU     $003B         ;PLLFLG -
PLLCR   EQU     $003C         ;PLLCR -
CLKSEL  EQU     $003D         ;CLKSEL -
SLOW    EQU     $003E         ;SLOW -
CGTCTL  EQU     $003F         ;CGTCTL -

PWCLK   EQU     $0040         ;PWCLK -
PWPOL   EQU     $0041         ;PWPOL -
PWEN    EQU     $0042         ;PWEN -
PWPRES  EQU     $0043         ;PWPRES -
PWSCAL0 EQU     $0044         ;PWSCAL0
PWSCNT0 EQU     $0045         ;PWSCNT0
PWSCAL1 EQU     $0046         ;PWSCAL1
PWSCNT1 EQU     $0047         ;PWSCNT1
PWCNT0  EQU     $0048         ;PWCNT0
PWCNT1  EQU     $0049         ;PWCNT1
PWCNT2  EQU     $004A         ;PWCNT2
PWCNT3  EQU     $004B         ;PWCNT3
PWPER0  EQU     $004C         ;PWPER0
PWPER1  EQU     $004D         ;PWPER1
PWPER2  EQU     $004E         ;PWPER2
PWPER3  EQU     $004F         ;PWPER3
PWDTY0  EQU     $0050         ;PWDTY0
PWDTY1  EQU     $0051         ;PWDTY1
```

```
PWDTY2  EQU    $0052        ;PWDTY2
PWDTY3  EQU    $0053        ;PWDTY3
PWCTL   EQU    $0054        ;PWCTL
PWTST   EQU    $0055        ;PWTST

PORTP   EQU    $0056        ;PORTP - PORTP REGISTER
DDRP    EQU    $0057        ;DDRP - DATA DIRECTION REGISTER

ATD0CTL0      EQU    $0060  ;ATDCTL0 - RESERVED
ATD0CTL1      EQU    $0061  ;ATDCTL1 - RESERVED
ATD0CTL2      EQU    $0062  ;ATDCTL2 - ATD CONTROL REGISTER
ATD0CTL3      EQU    $0063  ;ATDCTL3 - ATD CONTROL REGISTER
ATD0CTL4      EQU    $0064  ;ATDCTL4 - ATD CONTROL REGISTER
ATD0CTL5      EQU    $0065  ;ATDCTL5 - ATD CONTROL REGISTER

ATD0STAT0     EQU    $0066  ;ATDSTAT - ATD STATUS HIGH REGISTER
ATD0STAT1     EQU    $0067  ;ATDSTAT - LOW REGISTER
ATD0TESTH     EQU    $0068  ;ATDTEST - ATD TEST HIGH REGISTER
ATD0TESTL     EQU    $0069  ;ATDTEST - LOW REGISTER

PORTAD0 EQU    $006F        ;PORTAD - PORT AD DATA INPUT REGISTER
ADR00H  EQU    $0070        ;ADR00H
ADR00L  EQU    $0071        ;ADR00L
ADR01H  EQU    $0072        ;ADR01H
ADR01L  EQU    $0073        ;ADR01L
ADR02H  EQU    $0074        ;ADR02H
ADR02L  EQU    $0075        ;ADR02L
ADR03H  EQU    $0076        ;ADR03H
ADR03L  EQU    $0077        ;ADR03L
ADR04H  EQU    $0078        ;ADR04H
ADR04L  EQU    $0079        ;ADR04L
ADR05H  EQU    $007A        ;ADR05H
ADR05L  EQU    $007B        ;ADR05L
ADR06H  EQU    $007C        ;ADR06H
ADR06L  EQU    $007D        ;ADR06L
ADR07H  EQU    $007E        ;ADR07H
ADR07L  EQU    $007F        ;ADR07L

TIOS    EQU    $0080        ;TIOS - TIMER INPUT CAPTURE/OUTPUT
COMPARE SELECT
CFORC   EQU    $0081        ;CFORC - TIMER COMPARE FORCE
REGISTER
OC7M    EQU    $0082        ;OC7M - OUTPUT COMPARE 7 MASK
REGISTER
OC7D    EQU    $0083        ;OC7D - OUTPUT COMPARE 7 DATA REGISTER
```

```
TCNT    EQU     $0084

TSCR    EQU     $0086          ;TSCR - TIMER SYSTEM CONTROL REGISTER
TQCR    EQU     $0087          ;TQCR - RESERVED
TCTL1   EQU     $0088          ;TCTL1 - TIMER CONTROL REGISTER 1
TCTL2   EQU     $0089          ;TCTL2 - TIMER CONTROL REGISTER 2
TCTL3   EQU     $008A          ;TCTL3 - TIMER CONTROL REGISTER 3
TCTL4   EQU     $008B          ;TCTL4 - TIMER CONTROL REGISTER 4

TMSK1   EQU     $008C          ;TMSK1 - TIMER INTERRUPT MASK 1
TMSK2   EQU     $008D          ;TMSK2 - TIMER INTERRUPT MASK 2

TFLG1   EQU     $008E          ;TFLG1 - TIMER INTERRUPT FLAG 1
TFLG2   EQU     $008F          ;TFLG2 - TIMER INTERRUPT FLAG2

TC0     EQU     $0090          ;TC0 - TIMER INPUT/CAPTURE COMPARE HIGH
REGISTER0
TC1     EQU     $0092          ;TC1 - TIMER INPUT/CAPTURE COMPARE HIGH
REGISTER0
TC2     EQU     $0094          ;TC2 - TIMER INPUT/CAPTURE COMPARE HIGH
REGISTER0
TC3     EQU     $0096          ;TC3 - TIMER INPUT/CAPTURE COMPARE HIGH
REGISTER0
TC4     EQU     $0098          ;TC4 - TIMER INPUT/CAPTURE COMPARE HIGH
REGISTER0
TC5     EQU     $009A          ;TC5 - TIMER INPUT/CAPTURE COMPARE HIGH
REGISTER0
TC6     EQU     $009C          ;TC6 - TIMER INPUT/CAPTURE COMPARE HIGH
REGISTER0
TC7     EQU     $009E          ;TC7 - TIMER INPUT/CAPTURE COMPARE HIGH
REGISTER0

PACTL   EQU     $00A0          ;PATCL - PULSE ACCUMULATOR CONTROL
REGISTER
PAFLG   EQU     $00A1          ;PAFLG - PULSE ACCUMULATOR FLAG
REGISTER

PACN3   EQU     $00A2          ;PACN3
PACN2   EQU     $00A3          ;PACN2
PACN1   EQU     $00A4          ;PACN1
PACN0   EQU     $00A5          ;PACN0

MCCTL   EQU     $00A6          ;MCCTL
MCFLG   EQU     $00A7          ;MCFLG
ICPACR  EQU     $00A8          ;ICPAR
DLYCT   EQU     $00A9          ;DLYCT
```

```
ICOVW   EQU     $00AA          ;ICOVW
ICSYS   EQU     $00AB          ;ICSYS

TIMTST  EQU     $00AD          ;TIMTST - TIMER TEST REGISTER
PORTT   EQU     $00AE          ;PORTT
DDRT    EQU     $00AF          ;DDRT - DATA DIRECTION REGISTER

PBCTL   EQU     $00B0          ;PBCTL
PBFLG   EQU     $00B1          ;PBFLG
PA3H    EQU     $00B2          ;PA3H
PA2H    EQU     $00B3          ;PA2H
PA1H    EQU     $00B4          ;PA1H
PA0H    EQU     $00B5          ;PA0H
MCCNTH  EQU     $00B6          ;MCCNTH
MCCNTL  EQU     $00B7          ;MCCNTL

TC0H    EQU     $00B8          ;TC0H
TC0L    EQU     $00B9          ;TC0L
TC1H    EQU     $00BA          ;TC1H
TC1L    EQU     $00BB          ;TC1L
TC2H    EQU     $00BC          ;TC2H
TC2L    EQU     $00BD          ;TC2L
TC3H    EQU     $00BE          ;TC3H
TC3L    EQU     $00BF          ;TC3L

SC0BDH  EQU     $00C0          ;SC0BDH - SCI BAUD RATE CONTROL
REGISTER
SC0BDL  EQU     $00C1          ;SC0BDL - SCI BAUD RATE CONTROL
REGISTER
SC0CR1  EQU     $00C2          ;SC0CR1 - SCI CONTROL REGISTER
SC0CR2  EQU     $00C3          ;SC0CR2 - SCI CONTROL REGISTER
SC0SR1  EQU     $00C4          ;SC0SR1 - SCI STATUS REGISTER
SC0SR2  EQU     $00C5          ;SC0SR2 - SCI STATUS REGISTER
SC0DRH  EQU     $00C6          ;SC0DRH - SCI DATA REGISTER
SC0DRL  EQU     $00C7          ;SC0DRL - SCI DATA REGISTER
SC1BDH  EQU     $00C8          ;SC1BDH -
SC1BDL  EQU     $00C9          ;SC1BDL -
SC1CR1  EQU     $00CA          ;SC1CR1 -
SC1CR2  EQU     $00CB          ;SC1CR2 -
SC1SR1  EQU     $00CC          ;SC1SR1 -
SC1SR2  EQU     $00CD          ;SC1SR2 -
SC1DRH  EQU     $00CE          ;SC1DRH -
SC1DRL  EQU     $00CF          ;SC1DRL -

SP0CR1  EQU     $00D0          ;SP0CR1 - SPI CONTROL REGISTER
SP0CR2  EQU     $00D1          ;SP0CR2 - SPI CONTROL REGISTER
```

```
SP0BR   EQU    $00D2        ;SP0BR - SPI BAUD RATE REGISTER
SP0SR   EQU    $00D3        ;SP0SR - SPI STATUS REGISTER

SP0DR   EQU    $00D5        ;SP0DR - SPI DATA REGISTER
PORTS   EQU    $00D6        ;PORTS
DDRS    EQU    $00D7        ;PORTS - DATA DIRECTION REGISTER

PURDS   EQU    $00D9        ;PURDS -

IBAD    EQU    $00E0        ;IBAD -
IBFD    EQU    $00E1        ;IFBD -
IBCR    EQU    $00E2        ;IBCR -
IBSR    EQU    $00E3        ;IBSR -
IBDR    EQU    $00E4        ;IBDR -
IBPURD  EQU    $00E5        ;IBPURD -
PORTIB  EQU    $00E6        ;PORTIB -
DDRIB   EQU    $00E7        ;DDRIB -

EEMCR   EQU    $00F0        ;EEMCR - EEPROM MODULE
CONFIGURATION
EEPROT  EQU    $00F1        ;EEPROT - EEPROM BLOCK PROTECT
EETST   EQU    $00F2        ;EETST - EEPROM TEST
EEPROG  EQU    $00F3        ;EEPROG - EEPROM CONTROL

FEE32LCK    EQU    $00F4        ;FEE32LCK -
FEE32MCR    EQU    $00F5        ;FEE32MCR -
FEE32CTL    EQU    $00F7        ;FEE32CTL -
FEE28LCK    EQU    $00F8        ;FEE28LCK -
FEE28MCR    EQU    $00F9        ;FEE28MCR -
FEE28CTL    EQU    $00FB        ;FEE28CTL -

MTST0   EQU    $00F8        ;MTST0 -
MTST1   EQU    $00F9        ;MTST1 -
MTST2   EQU    $00FA        ;MTST2 -
MTST3   EQU    $00FB        ;MTST3 -
PORTK   EQU    $00FC        ;PORTK -
DDRK    EQU    $00FD        ;DDRK -

PPAGE   EQU    $00FF        ;PAGE -

CMCR0   EQU    $0100        ;C0MCR0 -
CMCR1   EQU    $0101        ;C0MCR1 -
CBTR0   EQU    $0102        ;C0BTR0 -
CBTR1   EQU    $0103        ;C0BTR1 -
CRFLG   EQU    $0104        ;C0RFLG -
CRIER   EQU    $0105        ;C0RIER -
```

```
CTFLG   EQU    $0106        ;C0TFLG -
CTCR    EQU    $0107        ;C0TCR -
CIDAC   EQU    $0108        ;C0IDAC -

CRXERR  EQU    $010E        ;C0RXERR -
CTXERR  EQU    $010F        ;C0TXERR -
CIDAR0  EQU    $0110        ;C0IDAR0 -
CIDAR1  EQU    $0111        ;C0IDAR1 -
CIDAR2  EQU    $0112        ;C0IDAR2 -
CIDAR3  EQU    $0113        ;C0IDAR3 -
CIDMR0  EQU    $0114        ;C0IDMR0 -
CIDMR1  EQU    $0115        ;C0IDMR1 -
CIDMR2  EQU    $0116        ;C0IDMR2 -
CIDMR3  EQU    $0117        ;C0IDMR3 -
CIDAR4  EQU    $0118        ;C0IDAR4 -
CIDAR5  EQU    $0119        ;C0IDAR5 -
CIDAR6  EQU    $011A        ;C0IDAR6 -
CIDAR7  EQU    $011B        ;C0IDAR7 -
CIDMR4  EQU    $011C        ;C0IDMR4 -
CIDMR5  EQU    $011D        ;C0IDMR5 -
CIDMR6  EQU    $011E        ;C0IDMR6 -
CIDMR7  EQU    $011F        ;C0IDMR7 -

PCTLCAN     EQU    $013D   ;PCTLCAN0 -
PORTCAN     EQU    $013E   ;PORTCAN0 -
DDRCAN      EQU    $013F   ;DDRCAN0 -

RXFG    EQU    $0140   ;RXFG0 -
TX0     EQU    $0150   ;TX00 -
TX1     EQU    $0160   ;TX01 -
TX2     EQU    $0170   ;TX01 -

ATD1CTL0    EQU    $01E0   ;ATD1CTL0 -
ATD1CTL1    EQU    $01E1   ;ATD1CTL1 -
ATD1CTL2    EQU    $01E2   ;ATD1CTL2 -
ATD1CTL3    EQU    $01E3   ;ATD1CTL3 -
ATD1CTL4    EQU    $01E4   ;ATD1CTL4 -
ATD1CTL5    EQU    $01E5   ;ATD1CTL5 -
ATD1STAT0   EQU    $01E6   ;ATD1STAT0 -
ATD1STAT1   EQU    $01E7   ;ATD1STAT1 -
ATD1TESTH   EQU    $01E8   ;ATD1TESTH -
ATD1TESTL   EQU    $01E9   ;ATD1TESTL -

PORTAD1 EQU    $01EF   ;PORTAD1 -
ADR10H  EQU    $01F0   ;ADR10H -
ADR10L  EQU    $01F1   ;ADR10L -
```

```
ADR11H  EQU     $01F2   ;ADR11H -
ADR11L  EQU     $01F3   ;ADR11L -
ADR12H  EQU     $01F4   ;ADR12H -
ADR12L  EQU     $01F5   ;ADR12L -
ADR13H  EQU     $01F6   ;ADR13H -
ADR13L  EQU     $01F7   ;ADR13L -
ADR14H  EQU     $01F8   ;ADR14H -
ADR14L  EQU     $01F9   ;ADR14L -
ADR15H  EQU     $01FA   ;ADR15H -
ADR15L  EQU     $01FB   ;ADR15L -
ADR16H  EQU     $01FC   ;ADR16H -
ADR16L  EQU     $01FD   ;ADR16L -
ADR17H  EQU     $01FE   ;ADR17H -
ADR17L  EQU     $01FF   ;ADR17L -

C1MCR0  EQU     $0300   ;C1MCR0 -
C1MCR1  EQU     $0301   ;C1MCR1 -
C1BTR0  EQU     $0302   ;C1BTR0 -
C1BTR1  EQU     $0303   ;C1BTR1 -
C1RFLG  EQU     $0304   ;C1RFLG -
C1TFLG  EQU     $0305   ;C1TFLG -
C1TFCR  EQU     $0306   ;C1TCR -
C1IDAC  EQU     $0307   ;C1IDAC -

C1RXERR EQU     $030E   ;C1RXERR -
C1TXERR EQU     $030F   ;C1TXERR -

C1IDAR0 EQU     $0310   ;C1IDAR0 -
C1IDAR1 EQU     $0311   ;C1IDAR1 -
C1IDAR2 EQU     $0312   ;C1IDAR2 -
C1IDAR3 EQU     $0313   ;C1IDAR3 -
C1IDMR0 EQU     $0314   ;C1IDMR0 -
C1IDMR1 EQU     $0315   ;C1IDMR1 -
C1IDMR2 EQU     $0316   ;C1IDMR2 -
C1IDMR3 EQU     $0317   ;C1IDMR3 -
C1IDAR4 EQU     $0318   ;C1IDAR4 -
C1IDAR5 EQU     $0319   ;C1IDAR5 -
C1IDAR6 EQU     $031A   ;C1IDAR6 -
C1IDAR7 EQU     $031B   ;C1IDAR7 -
C1IDMR4 EQU     $031C   ;C1IDMR4 -
C1IDMR5 EQU     $031D   ;C1IDMR5 -
C1IDMR6 EQU     $031E   ;C1IDMR6 -
C1IDMR7 EQU     $031F   ;C1IDMR7 -

PCTLCAN1        EQU     $033D   ;PCTLCAN1 -
PORTCAN1        EQU     $033E   ;PORTCAN1 -
```

```
DDRCAN1       EQU    $033F  ;DDRCAN1 -

RXFG1         EQU    $0340  ;RXFG1 -
TX10          EQU    $0350  ;TX10 -
TX11          EQU    $0360  ;TX11 -
TX12          EQU    $0370  ;TX12 -
```