

*TMS320 DSP  
DESIGNER'S NOTEBOOK*

# ***TMS320C/F240 Evaluation Board Initialization Software***

---

---

---

*APPLICATION REPORT: SPRA287*

*Pierre Voultoury  
Digital Signal Processing Products  
Semiconductor Group*

*Texas Instruments  
November 1997*



## **IMPORTANT NOTICE**

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

## **TRADEMARKS**

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

## **CONTACT INFORMATION**

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com

## Contents

<b>Abstract .....</b>	<b>7</b>
<b>Product Support.....</b>	<b>8</b>
World Wide Web .....	8
Email.....	8
<b>Design Problem.....</b>	<b>9</b>
<b>Solution.....</b>	<b>9</b>
Corresponding Code .....	10

# TMS320C/F240 Evaluation Board Initialization Software

---

---

---

## Abstract

The TMS320C/F240 Evaluation Module provides a tool that makes it easier to design the software elements of a system when taking the initial steps in the development of a new application.

This note allows the user, once the necessary tools have been installed, to quickly start developing code based on the TMS320C/F240 Evaluation Module. This document describes how to use the tools and provides a lengthy code example.



## Product Support

### World Wide Web

Our World Wide Web site at [www.ti.com](http://www.ti.com) contains the most up to date product information, revisions, and additions. Users registering with TI&ME can build custom information pages and receive new product updates automatically via email.

### Email

For technical issues or clarification on switching products, please send a detailed email to [dsph@ti.com](mailto:dsph@ti.com). Questions receive prompt attention and are usually answered within one business day.



## Design Problem

The TMS320C/F240 Evaluation Module provides a tool that makes it easier to design the software elements of a system when taking the initial steps in the development of a new application.

This note allows the user, once the necessary tools have been installed, to quickly start developing code based on the TMS320C/F240 Evaluation Module. A list of the necessary tools is summarized in the documentation given with this board.

## Solution

First set the jumper JP5 between pins number 2&3. This causes the  $V_{ccp}$  pin voltage to be equal to  $V_{cc}$  (+5V), thus allowing the watchdog unit to be disabled by software. (Note that the watchdog unit should be disabled only during the development and debug stages.) The jumper JP6 in the code development step should be set between pins number 1 & 2. This action causes the TMS320F240 device to run in microprocessor mode and all program memory accesses are off-chip (i.e. on-board RAM is used instead of on-chip Flash memory). The following references list the pages in the TMS320C/F240 User's Guide that are relevant to the initial set-up procedure:

- ☐ PLL unit to get a CPUCLK equal to 20MHz  
(pages 10-15 10-18).
- ☐ Disable the watchdog  
(pages 6-12).
- ☐ Manage the shared I/O ports  
(pages 11-11 11-18).
- ☐ Initialization of the ADC Unit  
(p 3-6 3-10).
- ☐ Initialization of the Capture Unit  
(p 2-75 2-78).
- ☐ Manage incoming interrupts.

The second file included is the memory mapping necessary to link the software. Regarding the included file C240reg.h, please refer to the Designer's Note Page titled Programmable Registers Addresses.

Shown below is the code listing.





## Corresponding Code

```
*****
; File Name : easy.asm
; Target System : c240 evm
; Description : This software gives a basic software
; configuration sufficient to get quickly started with
; the C/F240 EVM.
*****
        .include "C240reg.h"
; Variable definitions
        .bss      capt,1
; Reset & interrupt vectors
        .sect     "vectors"
RSVECT B _c_int0
INT1      B      COMINT
INT2      B      GPT1INT
INT3      B      GPT23INT
INT4      B      CAPINT
          .space  16*2
INT6      B      ADCINT
          .space  16*38

        .text
        .global  _c_int0
_c_int0

        SETC      CNF
        CLRC      OVM      ; Reset overflow mode
        CLRC      SXM      ; Reset sign extension mode
        CLRC      XF
        SETC      INTM      ; Set global interrupt mask

;Disable watchdog (Vccp=5v), watchdog counter reset p6-12
        LDP        #00E0h
        SPLK       #0006Fh, WD_CNTL
        SPLK       #05555h, WD_KEY
        SPLK       #0AAAAh, WD_KEY

;set up PLL clockin=10Mhz, CPUCLOCK=20Mhz, SYSCLK=10Mhz
;CKCR1 must be set before CKCR0
        SPLK       #00b1h,CKCR1
        SPLK       #0081h,CKCR0

; Set up CLKOUT to be SYSCLK p6-6
        SPLK       #40C0h,SYSCR

;I/O setting p11-11
        LDP        #00E1h
        SPLK       #0ffffh, OCRA
        SPLK       #0FF70h, OCRB
        SPLK       #0f1f1h, PCDATDIR

;Clear EV control registers
        LDP        #0e8h
        SPLK       #0000h,T1CON      ;no timer enable
        SPLK       #0000h,T1PER
        SPLK       #0000h,T1CNT
        SPLK       #0000h,T1CMP
        SPLK       #0000h,T2CON
```



```

        SPLK      #0000h,T2PER
        SPLK      #0000h,T2CNT
        SPLK      #0000h,T2CMP
        SPLK      #0000h,T3CON
        SPLK      #0000h,T3PER
        SPLK      #0000h,T3CNT
        SPLK      #0000h,T3CMP
        SPLK      #0000h,COMCON
        SPLK      #0000h,DBTCON
        SPLK      #0000h,ACTR
        SPLK      #0000h,SACTR
        SPLK      #0000h,CMPR1
        SPLK      #0000h,CMPR2
        SPLK      #0000h,CMPR3
        SPLK      #0000h,SCMPR1
        SPLK      #0000h,SCMPR2
        SPLK      #0000h,SCMPR3
        SPLK      #0000h,CAPCON      ;no capture
        SPLK      #0000h,FIFO1
        SPLK      #0000h,FIFO2
        SPLK      #0000h,FIFO3
        SPLK      #00ffh,CAPFIFO

;Capture Unit Setting
        SPLK      #0b0fch,CAPCON
        SPLK      #00ffh,CAPFIFO

;Core Mask Setting
        LDP        #0
        LACC       #028h
        SACL       IMR
        LACC       IFR
        SACL       IFR

;EV Mask Setting, Vector & Flag reset p11-46
        LDP        #0E8h
        LACC       IFRA
        SACL       IFRA
        LACC       IFRB
        SACL       IFRB
        LACC       IFRC
        SACL       IFRC
        SPLK       #0,IMRA
        SPLK       #0,IMRB
        SPLK       #7,IMRC
        LACC       IVRA
        LACC       IVRB
        LACC       IVRC

;ADC Unit setting
        LDP        #0E0h
        SPLK       #0403h,ADCTRL2
        SPLK       #1b00h,ADCTRL1

        CLRC       INTM

LOOP      B        LOOP

COMINT

        CLRC       ;core of this interrupt
        CLRC       INTM
        RET

```



```
GPT1INT
                                ;core of this interrupt
                                ;example of context saving
; start context saving
    MAR    *, AR1      ; make AR1 pointer active
    MAR    *+          ; skip one position
    SST    #1, *+      ; save ST1
    SST    #0, *+      ; save ST0
    SACH    *+          ; save H of ACC
    SACL    *+          ; save L of ACC
    SPH    *+          ; save H of PREG
    SPL    *+          ; save L of PREG
    MPY    #1          ; get TREG
    SPL    *+          ; save TREG (16 bit)
    POPD    *          ; save RET value
; end context saving
; start context restoring
    MAR    *, AR1      ;make stack pointer active
    PSHD    *-          ; restore RET value
    MAR    *-          ; skip TREG
    LT      *+          ; load L PREG
    MPY    *#1          ; restore L PREG
    LT      *-          ; restore TREG
    MAR    *-          ;
    LPH    *-          ; restore H PREG
    LACL    *-          ; load L ACC
    ADD     *- , 16     ; load H ACC
    LST     *#0, *-     ; load ST0
    LST     *#1, *-     ; load ST1
; end context restoring
    CLRC    INTM
    RET

GPT23INT
                                ;core of this interrupt
    CLRC    INTM
    RET

CAPINT
                                ;core of this interrupt
    CLRC    INTM
    RET

ADCINT
                                ;core of this interrupt
    CLRC    INTM
    RET

*****
* LINKER COMMAND FILE - MEMORY SPECIFICATION for C240 *
* File Name init.cmd *
*****
MEMORY
{
PAGE 0 :  VECS    :origin = 0h      , length = 040h
          PROG    :origin = 40h     , length = 0800h
PAGE 1 :  MMRS    :origin = 0h      , length = 05Fh
          B2      :origin = 0060h   , length = 020h /*DARAM */
          0       :origin = 0100h   , length = 0200h /*DARAM */
          B1      :origin = 0300h   , length = 0200h /*DARAM */
}
*****
* SECTIONS ALLOCATION *
```



```
*****
SECTIONS
{
    .vectors : { } > VECS PAGE 0 /*INTERRUPT VECTOR TABLE*/
    .text    : { } > PROG PAGE 0 /* CODE */
    .mmrs    : { } > MMRS PAGE 1 /*Memory Mapped Reg*/
    .blk0    : { } > B0  PAGE 1  /* Block B0 - page 4 */
    .bss     : { } > B2  PAGE 1  /* Block B2 - page 0 */
    .blk1    : { } > B1  PAGE 1  /* Block B1 - page ? */
}

```