

Creating and Modifying User Modules

And other Unnatural Acts

Dave Van Ess
Principal Applications Engineer



Objectives

Objective:

- Introductory the fundamental components of a PSoC User Module.
- Show how XML is used to describe the intraconnection of the PSoC resources.
- Create an improved UM (timer16X) using the old UM (timer16) as a template.

User Modules

A User Module is:

- The information required to connect PSoC blocks.
- Software (API) to control it
- A Datasheet
- An Icon

User Modules

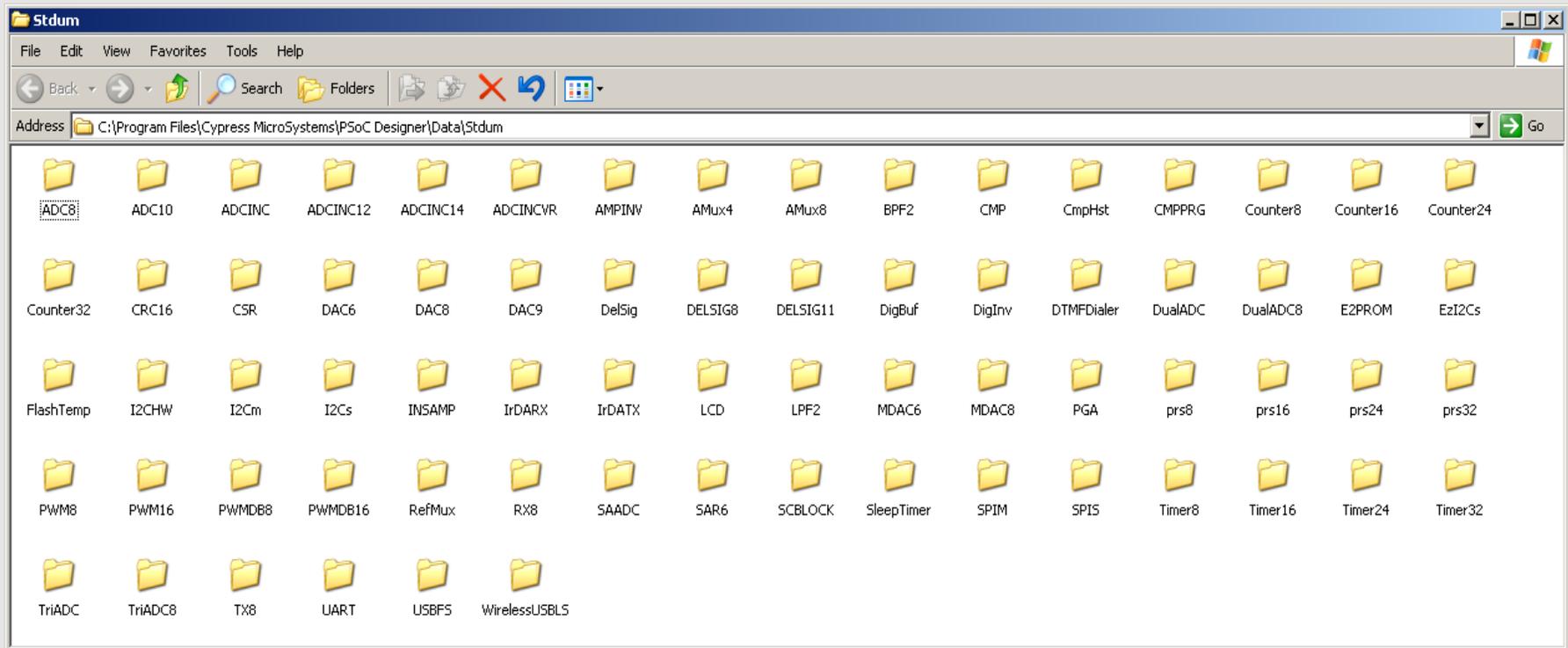
User Module

- At its most basic form is a directory and an XML file with the same name.
 - The same name must also appear as the NAME attribute in the <PSOC_USER_MODULE> element in the UM XML file.
- If done correctly the UM is added to the active UM library.
- Other files exist in the User Module directory, but their naming is controlled by attributes within the UM XML file.
- The other files which are relevant to the UM
 - *.ico – User Module icon
 - *.htm – User Module data sheet
 - *.emf – User Module block diagram

User Modules

User Module Directory

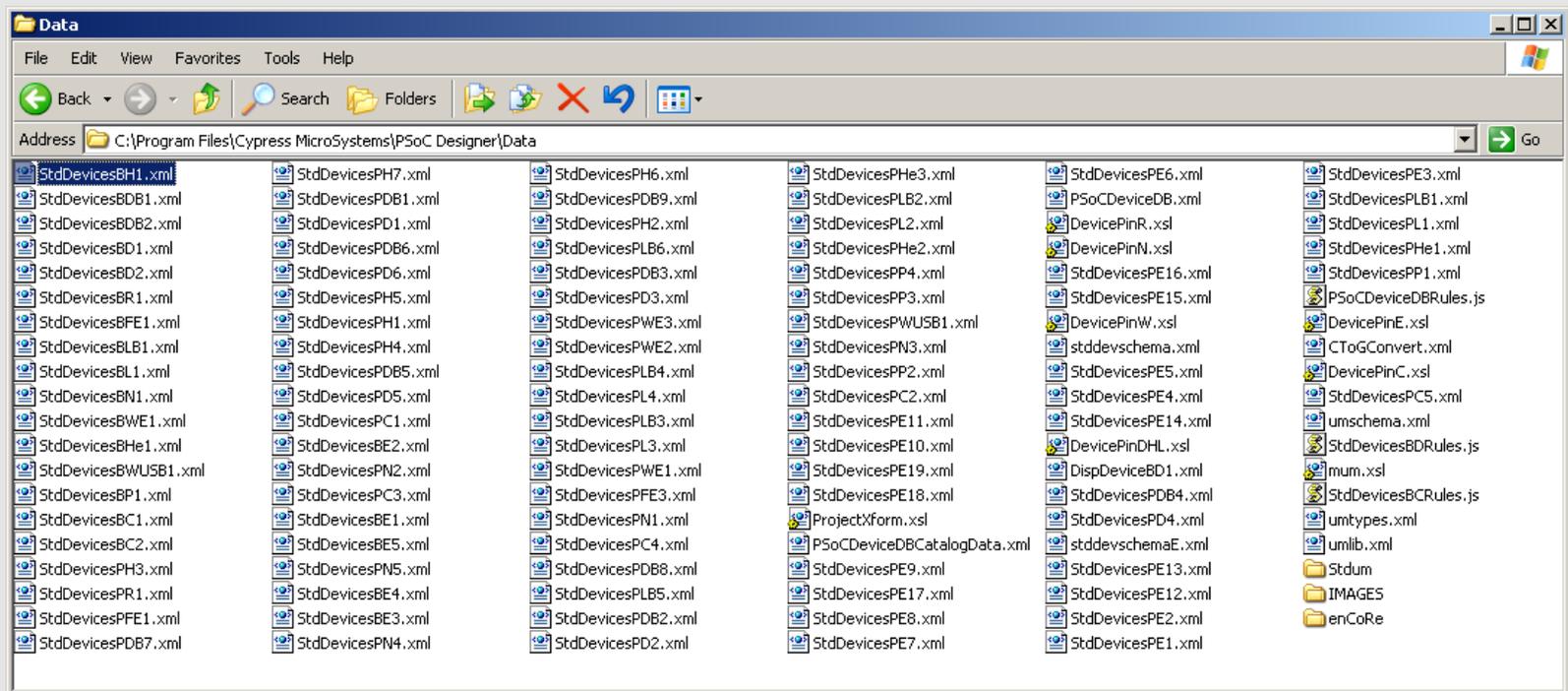
- All UMs located at <C:\Program Files\Cypress MicroSystems\PSoC Designer\Data\Stdum>



User Modules

User Module Directory

- All device descriptions located at <C:\Program Files\Cypress Microsystems\PSoC Designer\Data>
 - Think of each as a dictionary of acceptable register and bit field values for each particular device.



User Modules

Making New User Modules:

- Building a UM from scratch is extremely difficult to do.
- It is far easier to start with a copy of an established UM and modify it to meet your needs.
- Information from several different UMs can be combined to produce a new UM.
 - The counter8 XML and the DAC6 XML could be combined to produce a waveform generator UM.
 - Think of it as gene splicing.

User Modules

Project Goal

Build an improved Timer16

- The upper digital block of the present Timer16 has a parameter selection to interrupt on:
 - Capture
 - Terminal Count
 - Compare True
- The goal is to create a new UM (timer16X) that:
 - Generates a Terminal Count interrupt for the upper digital block.
 - Generates a Capture interrupt for the lower digital block.

User Modules

Step 1

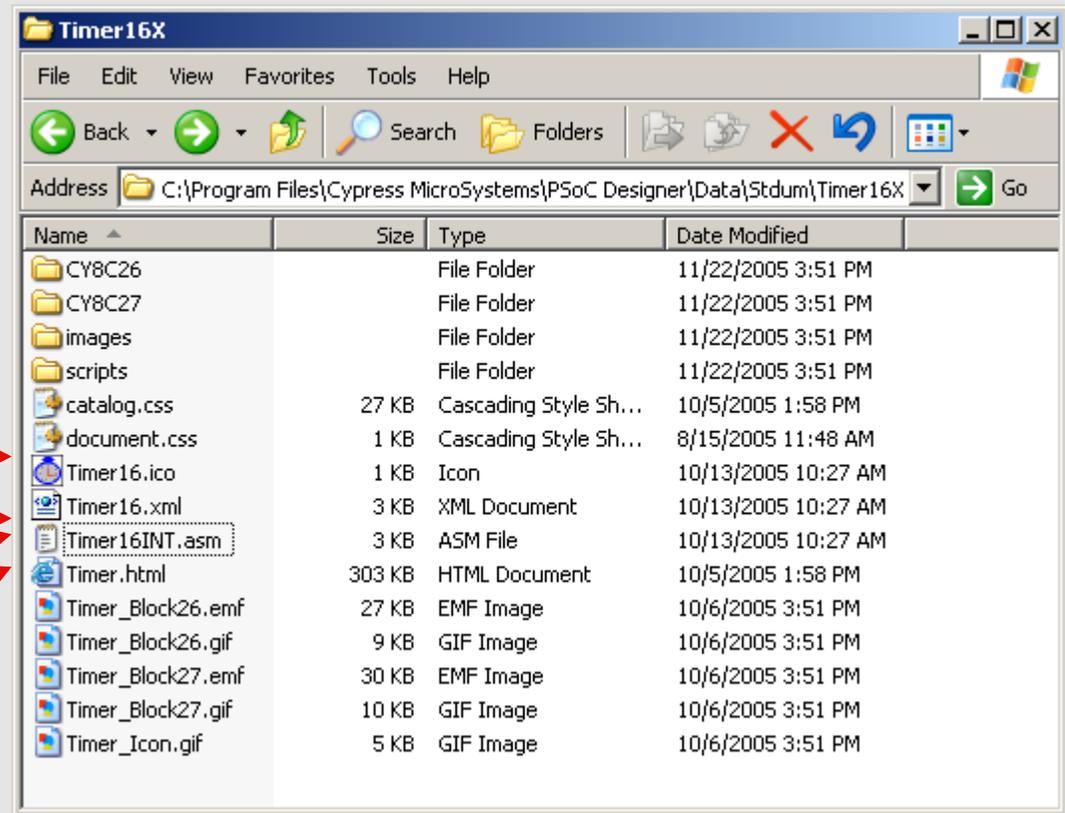
- Make a copy of **Timer16**.
- Rename it **Timer16X**
- Open it

Icon →

Base XML →

Interrupt template →

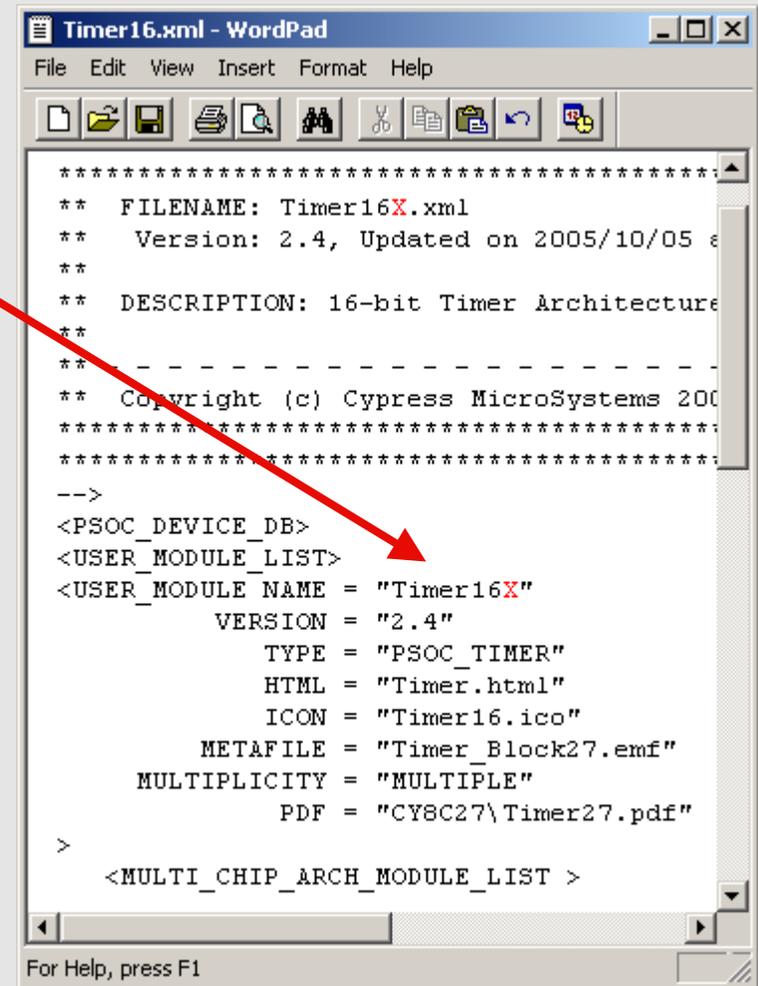
Datasheet →



User Modules

Step 2

- Open **Timer16.xml**
 - Use your preferred text editor
- Change **USER_MODULE_NAME** to **TimerX**.
- Might as well change the **FILENAME**: while you're here.
 - Its only a comment but it can't hurt.
- Close the file and reopen with Internet Explorer.
 - Should open with no problems.
 - This checking process will be done frequently.
 - Lets call it CRET



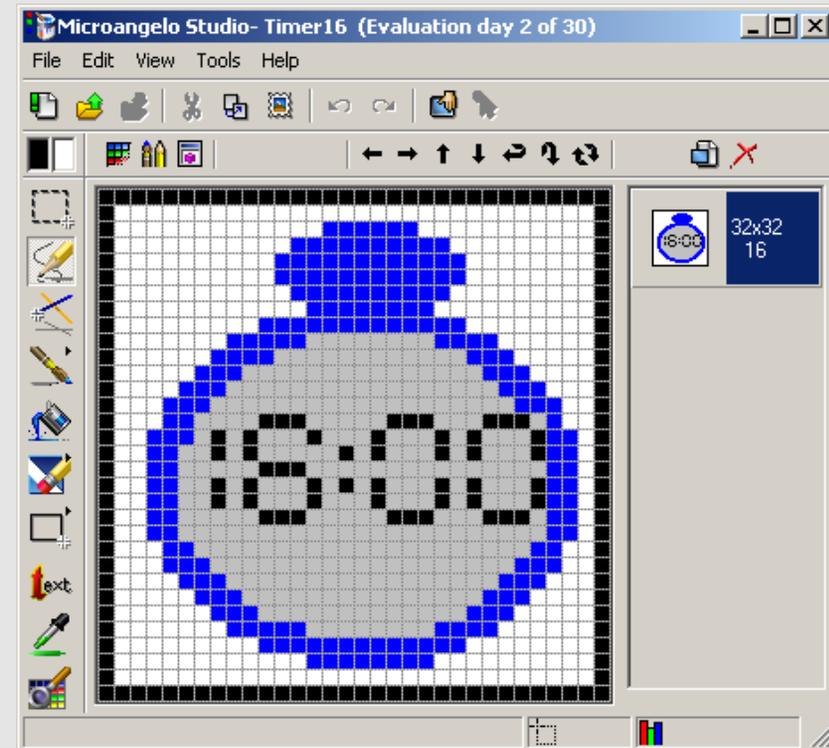
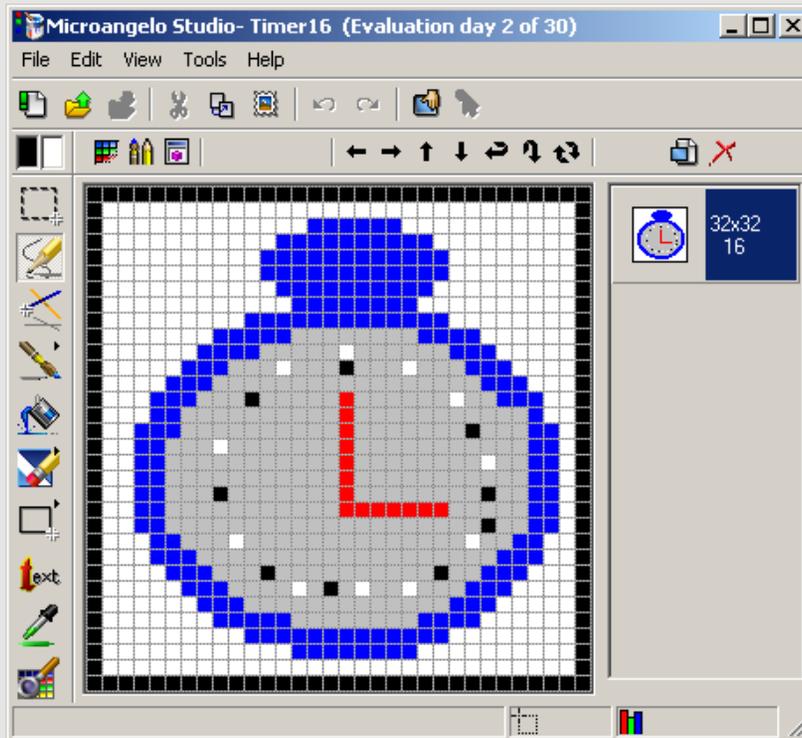
```
*****
**  FILENAME: Timer16X.xml
**  Version: 2.4, Updated on 2005/10/05 e
**
**  DESCRIPTION: 16-bit Timer Architecture
**
**  -----
**  Copyright (c) Cypress MicroSystems 200
*****
*****
-->
<PSOC_DEVICE_DB>
<USER_MODULE_LIST>
<USER_MODULE_NAME = "Timer16X"
    VERSION = "2.4"
    TYPE = "PSOC_TIMER"
    HTML = "Timer.html"
    ICON = "Timer16.ico"
    METAFILE = "Timer_Block27.emf"
    MULTIPLICITY = "MULTIPLE"
    PDF = "CY8C27\Timer27.pdf"
>
<MULTI_CHIP_ARCH_MODULE_LIST >
```

Close, Reopen with Explorer to Test

User Modules

Step 3

- Open **Timer16.ico**
- Alter Icon



User Modules

Sanity Check!

Open a 27x project with Designer

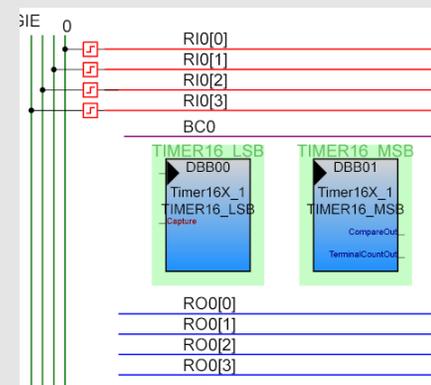
- Verify that UM is present.
- Verify that it places.

For many UMs this is all the XML changes that are required.

- For UMs that have significant changes in software but no hardware changes.
 - This is your chance to change the APIs of your favorite UM to the way that you (*and God*) think they should be, while leaving the base UM hardware configuration alone!

Timer16X_1

User Module Parameters	Value
Clock	?
Capture	?
TerminalCountOut	?
CompareOut	?
Period	0
CompareValue	0
CompareType	?
InterruptType	?
ClockSync	?
TC_PulseWidth	Full Clock
InvertCapture	Normal



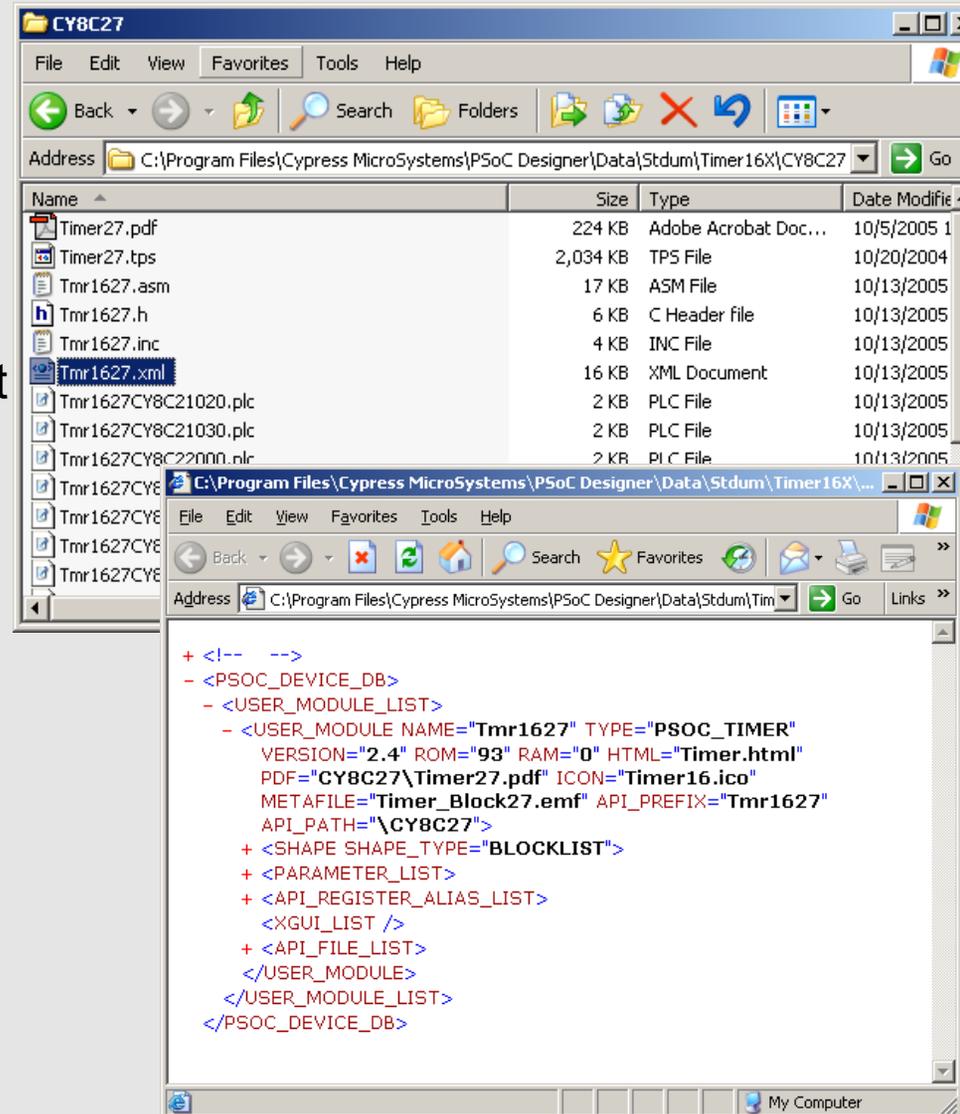
User Modules

Open **CY8C27**

Open **Tmr1627.xml**

XML has 4 main sections

- **SHAPE**
 - For bit fields are set that don't change.
- **PARAMETER_LIST**
 - For bit fields are set by user selection.
- **API_REGISTER_ALIS_LIST**
 - Allows appropriate names of registers.
- **API_FILE LIST**
 - Lists the API files to be generated.



The screenshot shows a Windows Explorer window titled 'CY8C27' with the address bar pointing to 'C:\Program Files\Cypress Microsystems\PSoC Designer\Data\Stdum\Timer16X\CY8C27'. The file list includes:

Name	Size	Type	Date Modified
Timer27.pdf	224 KB	Adobe Acrobat Doc...	10/5/2005 1
Timer27.tps	2,034 KB	TPS File	10/20/2004
Tmr1627.asm	17 KB	ASM File	10/13/2005
Tmr1627.h	6 KB	C Header file	10/13/2005
Tmr1627.inc	4 KB	INC File	10/13/2005
Tmr1627.xml	16 KB	XML Document	10/13/2005
Tmr1627CY8C21020.plc	2 KB	PLC File	10/13/2005
Tmr1627CY8C21030.plc	2 KB	PLC File	10/13/2005
Tmr1627CY8C22000.plc	2 KB	PLC File	10/13/2005
Tmr1627CY8C...			

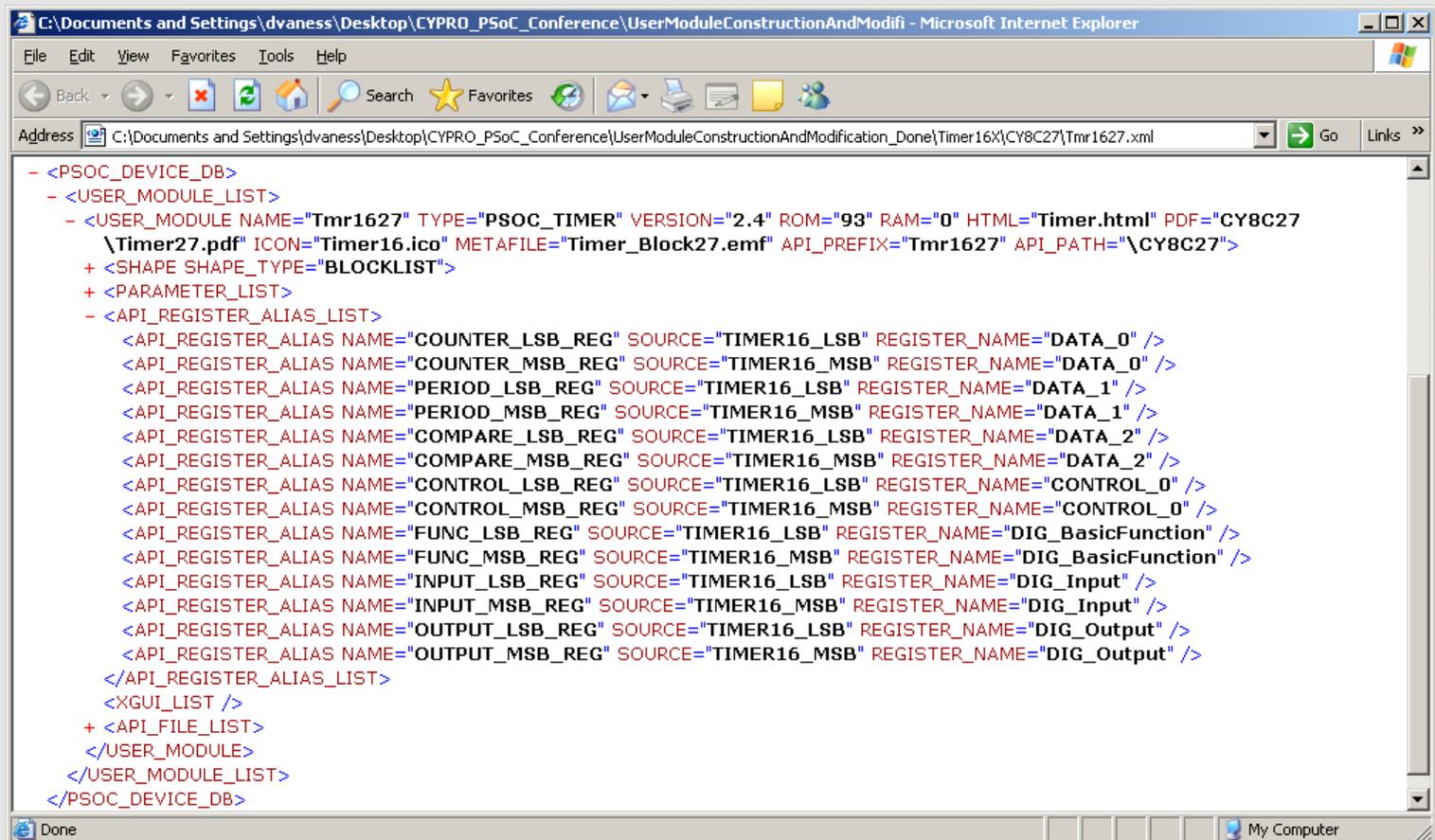
An inset window shows the XML content of 'Tmr1627.xml':

```
+ <!-- -->
- <PSOC_DEVICE_DB>
- <USER_MODULE_LIST>
  - <USER_MODULE NAME="Tmr1627" TYPE="PSOC_TIMER"
    VERSION="2.4" ROM="93" RAM="0" HTML="Timer.html"
    PDF="CY8C27\Timer27.pdf" ICON="Timer16.ico"
    METAFILE="Timer_Block27.emf" API_PREFIX="Tmr1627"
    API_PATH="\CY8C27">
    + <SHAPE SHAPE_TYPE="BLOCKLIST">
    + <PARAMETER_LIST>
    + <API_REGISTER_ALIAS_LIST>
    <XGUI_LIST />
    + <API_FILE_LIST>
  </USER_MODULE>
</USER_MODULE_LIST>
</PSOC_DEVICE_DB>
```

User Modules

API_REGISTER_ALIAS_LIST

- Allows you to define register names.



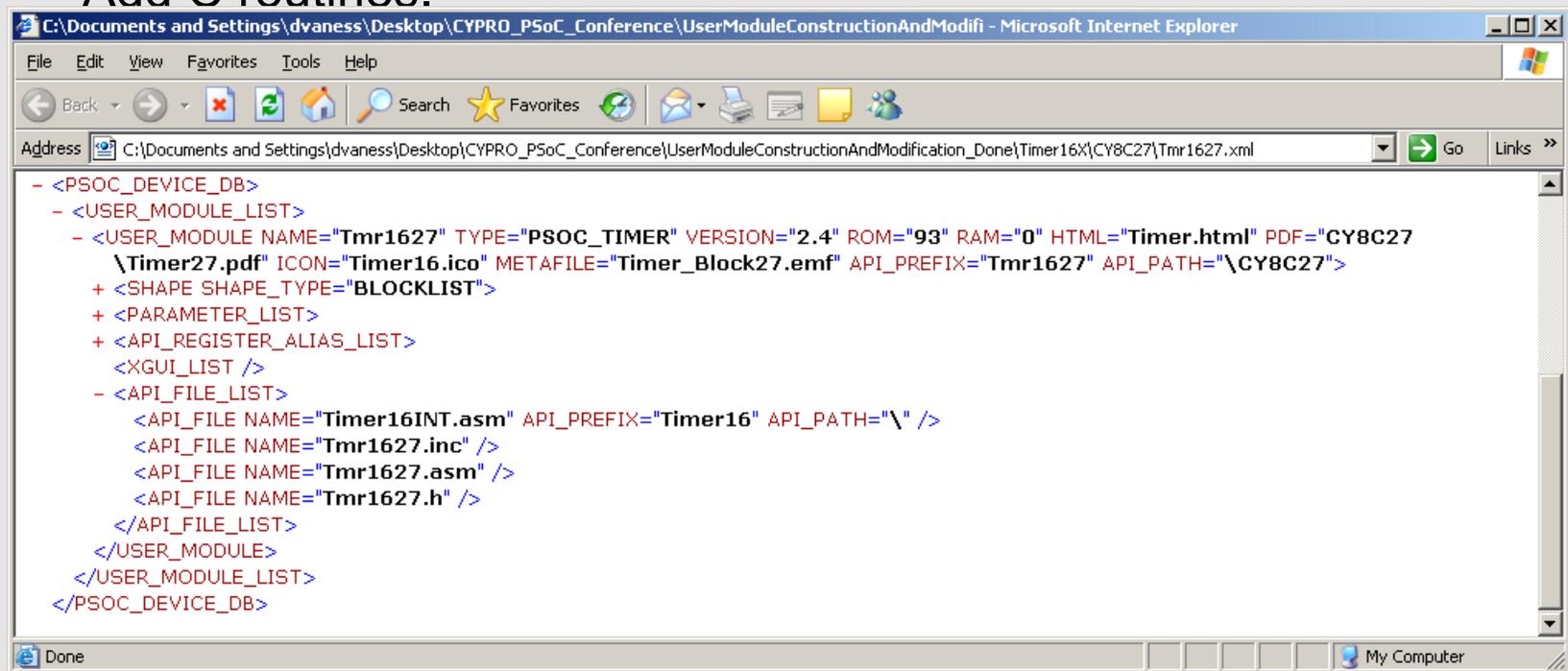
The screenshot shows a Microsoft Internet Explorer browser window displaying XML code. The address bar shows the file path: C:\Documents and Settings\dvaness\Desktop\CYPRO_PSoC_Conference\UserModuleConstructionAndModification_Done\Timer16\CY8C27\Tmr1627.xml. The code is as follows:

```
- <PSOC_DEVICE_DB>
- <USER_MODULE_LIST>
  - <USER_MODULE NAME="Tmr1627" TYPE="PSOC_TIMER" VERSION="2.4" ROM="93" RAM="0" HTML="Timer.html" PDF="CY8C27
    \Timer27.pdf" ICON="Timer16.ico" METAFILE="Timer_Block27.emf" API_PREFIX="Tmr1627" API_PATH="\CY8C27">
    + <SHAPE SHAPE_TYPE="BLOCKLIST">
    + <PARAMETER_LIST>
    - <API_REGISTER_ALIAS_LIST>
      <API_REGISTER_ALIAS NAME="COUNTER_LSB_REG" SOURCE="TIMER16_LSB" REGISTER_NAME="DATA_0" />
      <API_REGISTER_ALIAS NAME="COUNTER_MSB_REG" SOURCE="TIMER16_MSB" REGISTER_NAME="DATA_0" />
      <API_REGISTER_ALIAS NAME="PERIOD_LSB_REG" SOURCE="TIMER16_LSB" REGISTER_NAME="DATA_1" />
      <API_REGISTER_ALIAS NAME="PERIOD_MSB_REG" SOURCE="TIMER16_MSB" REGISTER_NAME="DATA_1" />
      <API_REGISTER_ALIAS NAME="COMPARE_LSB_REG" SOURCE="TIMER16_LSB" REGISTER_NAME="DATA_2" />
      <API_REGISTER_ALIAS NAME="COMPARE_MSB_REG" SOURCE="TIMER16_MSB" REGISTER_NAME="DATA_2" />
      <API_REGISTER_ALIAS NAME="CONTROL_LSB_REG" SOURCE="TIMER16_LSB" REGISTER_NAME="CONTROL_0" />
      <API_REGISTER_ALIAS NAME="CONTROL_MSB_REG" SOURCE="TIMER16_MSB" REGISTER_NAME="CONTROL_0" />
      <API_REGISTER_ALIAS NAME="FUNC_LSB_REG" SOURCE="TIMER16_LSB" REGISTER_NAME="DIG_BasicFunction" />
      <API_REGISTER_ALIAS NAME="FUNC_MSB_REG" SOURCE="TIMER16_MSB" REGISTER_NAME="DIG_BasicFunction" />
      <API_REGISTER_ALIAS NAME="INPUT_LSB_REG" SOURCE="TIMER16_LSB" REGISTER_NAME="DIG_Input" />
      <API_REGISTER_ALIAS NAME="INPUT_MSB_REG" SOURCE="TIMER16_MSB" REGISTER_NAME="DIG_Input" />
      <API_REGISTER_ALIAS NAME="OUTPUT_LSB_REG" SOURCE="TIMER16_LSB" REGISTER_NAME="DIG_Output" />
      <API_REGISTER_ALIAS NAME="OUTPUT_MSB_REG" SOURCE="TIMER16_MSB" REGISTER_NAME="DIG_Output" />
    </API_REGISTER_ALIAS_LIST>
    <XGUI_LIST />
    + <API_FILE_LIST>
  </USER_MODULE>
</USER_MODULE_LIST>
</PSOC_DEVICE_DB>
```

User Modules

API_FILE

- Allow you to define name and location of all API files.
 - Is possible to add multiple assembly files
 - Add C routines.



The screenshot shows a Microsoft Internet Explorer window displaying XML code for a user module configuration. The address bar shows the file path: C:\Documents and Settings\dvaness\Desktop\CYPRO_PSoC_Conference\UserModuleConstructionAndModification_Done\Timer16\CY8C27\Tmr1627.xml. The code is as follows:

```
- <PSOC_DEVICE_DB>
- <USER_MODULE_LIST>
- <USER_MODULE NAME="Tmr1627" TYPE="PSOC_TIMER" VERSION="2.4" ROM="93" RAM="0" HTML="Timer.html" PDF="CY8C27
  \Timer27.pdf" ICON="Timer16.ico" METAFILE="Timer_Block27.emf" API_PREFIX="Tmr1627" API_PATH="\CY8C27">
+ <SHAPE SHAPE_TYPE="BLOCKLIST">
+ <PARAMETER_LIST>
+ <API_REGISTER_ALIAS_LIST>
  <XGUI_LIST />
- <API_FILE_LIST>
  <API_FILE NAME="Timer16INT.asm" API_PREFIX="Timer16" API_PATH="\\" />
  <API_FILE NAME="Tmr1627.inc" />
  <API_FILE NAME="Tmr1627.asm" />
  <API_FILE NAME="Tmr1627.h" />
</API_FILE_LIST>
</USER_MODULE>
</USER_MODULE_LIST>
</PSOC_DEVICE_DB>
```

User Modules

To modify this UM

- Lower Block must be set to also be an interrupt source in the SHAPE section.
- The parameter used to select interrupt type must be removed from the PARAMETER_LIST area.
- Bit Values must be added to the SHAPE section to set the interrupt type to:
 - Terminal Count for the Upper Block
 - Capture for the Lower Block

User Modules

Step 4

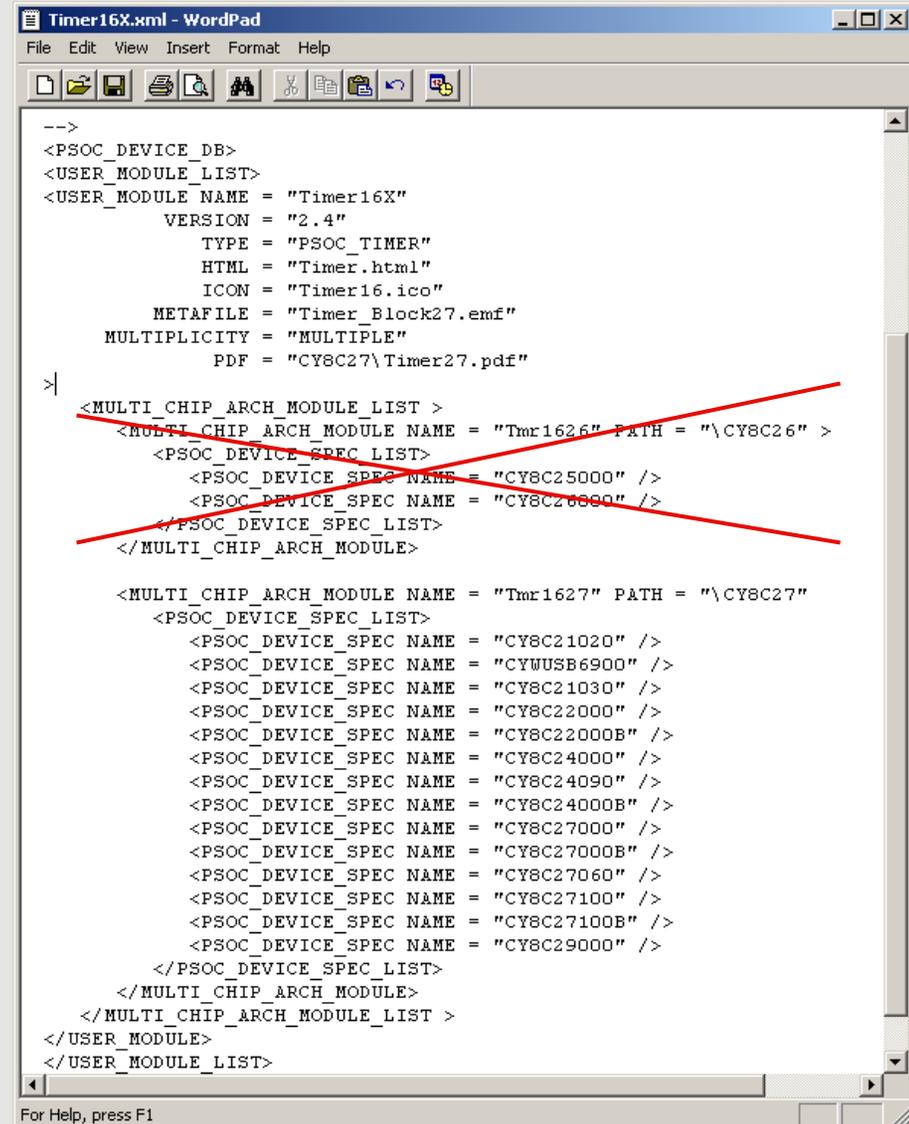
This user module will not work for 25x and 26x family of parts. So

- Open **Timer16X.xml**
Remove

```
<MULTI_CHIP_ARCH_MODULE NAME = "Tmr1626" PATH = "\\CY8C26" >  
  <PSOC_DEVICE_SPEC_LIST>  
    <PSOC_DEVICE_SPEC NAME = "CY8C25000" />  
    <PSOC_DEVICE_SPEC NAME = "CY8C26000" />  
  </PSOC_DEVICE_SPEC_LIST>  
</MULTI_CHIP_ARCH_MODULE>
```

CRET

- The CY8C26 directory is no longer needed.
 - Remove it!



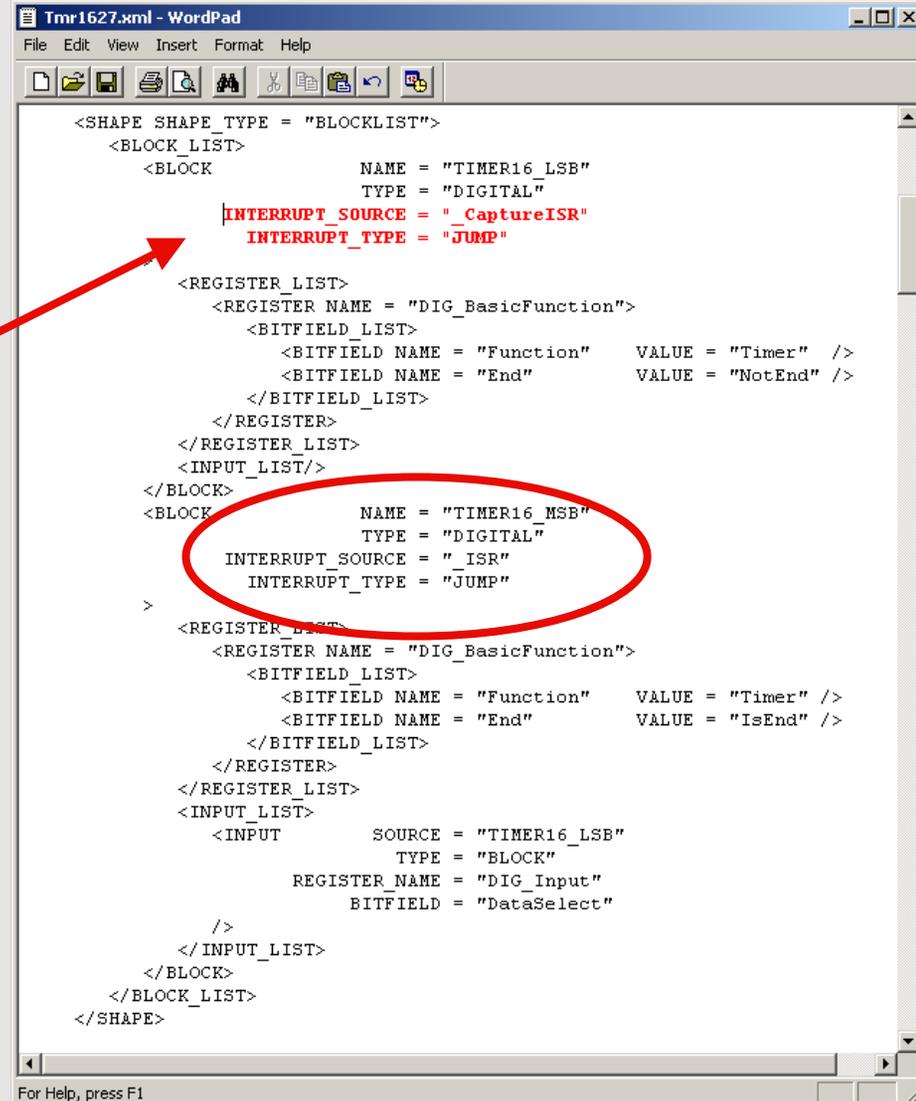
```
-->  
<PSOC_DEVICE_DB>  
<USER_MODULE_LIST>  
<USER_MODULE NAME = "Timer16X"  
  VERSION = "2.4"  
  TYPE = "PSOC_TIMER"  
  HTML = "Timer.html"  
  ICON = "Timer16.ico"  
  METAFILE = "Timer_Block27.emf"  
  MULTIPLICITY = "MULTIPLE"  
  PDF = "CY8C27\Timer27.pdf"  
>  
<MULTI_CHIP_ARCH_MODULE_LIST >  
  <MULTI_CHIP_ARCH_MODULE NAME = "Tmr1626" PATH = "\\CY8C26" >  
    <PSOC_DEVICE_SPEC_LIST>  
      <PSOC_DEVICE_SPEC NAME = "CY8C25000" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C26000" />  
    </PSOC_DEVICE_SPEC_LIST>  
  </MULTI_CHIP_ARCH_MODULE>  
  
  <MULTI_CHIP_ARCH_MODULE NAME = "Tmr1627" PATH = "\\CY8C27"  
    <PSOC_DEVICE_SPEC_LIST>  
      <PSOC_DEVICE_SPEC NAME = "CY8C21020" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C21030" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C22000" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C22000B" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C24000" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C24000B" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C24090" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C27000" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C27000B" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C27060" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C27100" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C27100B" />  
      <PSOC_DEVICE_SPEC NAME = "CY8C29000" />  
    </PSOC_DEVICE_SPEC_LIST>  
  </MULTI_CHIP_ARCH_MODULE>  
</MULTI_CHIP_ARCH_MODULE_LIST >  
</USER_MODULE>  
</USER_MODULE_LIST>
```

User Modules

Step 5 *Add Interrupt*

- Open **Tmr1627.xml**
- Use interrupt code from **TIMER16_MSB** block to add an interrupt to Timer16LSB block.
 - Name this new interrupt source **_CaptureISR**

CRET



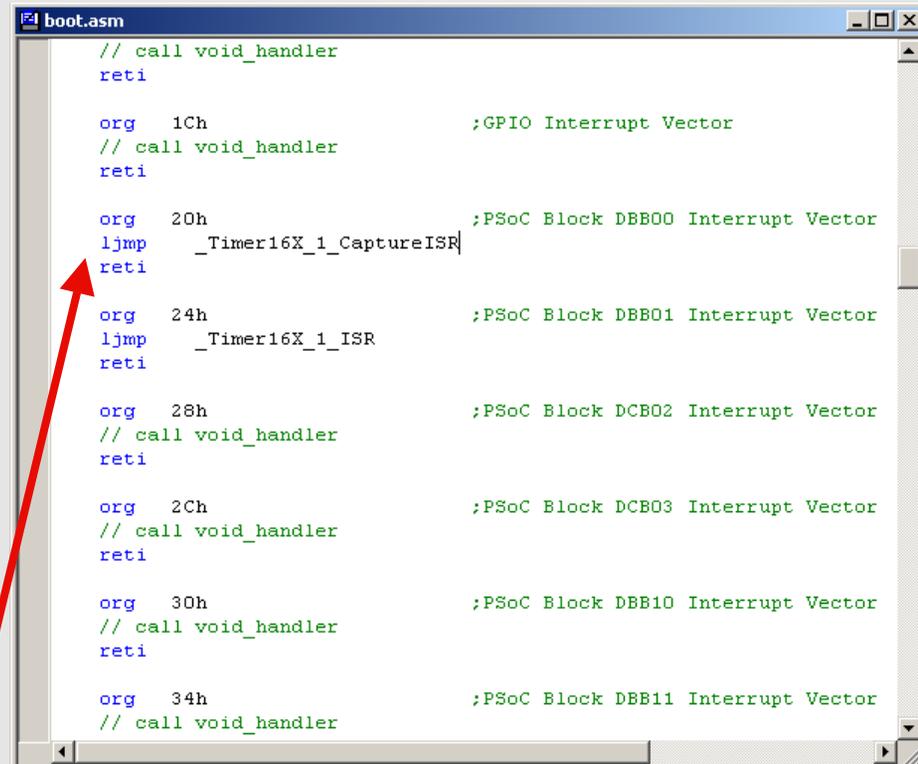
```
<SHAPE SHAPE_TYPE = "BLOCKLIST">
  <BLOCK_LIST>
    <BLOCK
      NAME = "TIMER16_LSB"
      TYPE = "DIGITAL"
      INTERRUPT_SOURCE = "_CaptureISR"
      INTERRUPT_TYPE = "JUMP"
    >
    <REGISTER_LIST>
      <REGISTER NAME = "DIG_BasicFunction">
        <BITFIELD_LIST>
          <BITFIELD NAME = "Function" VALUE = "Timer" />
          <BITFIELD NAME = "End" VALUE = "NotEnd" />
        </BITFIELD_LIST>
      </REGISTER>
    </REGISTER_LIST>
    <INPUT_LIST/>
  </BLOCK>
  <BLOCK
    NAME = "TIMER16_MSB"
    TYPE = "DIGITAL"
    INTERRUPT_SOURCE = "_ISR"
    INTERRUPT_TYPE = "JUMP"
  >
  <REGISTER_LIST>
    <REGISTER NAME = "DIG_BasicFunction">
      <BITFIELD_LIST>
        <BITFIELD NAME = "Function" VALUE = "Timer" />
        <BITFIELD NAME = "End" VALUE = "IsEnd" />
      </BITFIELD_LIST>
    </REGISTER>
  </REGISTER_LIST>
  <INPUT_LIST>
    <INPUT
      SOURCE = "TIMER16_LSB"
      TYPE = "BLOCK"
      REGISTER_NAME = "DIG_Input"
      BITFIELD = "DataSelect"
    />
  </INPUT_LIST>
</BLOCK_LIST>
</SHAPE>
```

User Modules

Sanity Check!

Open a 27x project with Designer

- Place Timer16X
- Generate application.
- Go to application editor
 - Open boot.asm
- Verify that the new interrupt vector has been added.



```
boot.asm
// call void_handler
reti

org 1Ch ;GPIO Interrupt Vector
// call void_handler
reti

org 20h ;PSoC Block DBB00 Interrupt Vector
ljmp _Timer16X_1_CaptureISR
reti

org 24h ;PSoC Block DBB01 Interrupt Vector
ljmp _Timer16X_1_ISR
reti

org 28h ;PSoC Block DCB02 Interrupt Vector
// call void_handler
reti

org 2Ch ;PSoC Block DCB03 Interrupt Vector
// call void_handler
reti

org 30h ;PSoC Block DBB10 Interrupt Vector
// call void_handler
reti

org 34h ;PSoC Block DBB11 Interrupt Vector
// call void_handler
```

User Modules

Step 6 *Remove Interrupt Type Parameter*

- Open **Tmr1627.xml**
- Remove the **InterruptType** parameter.
 - Save removed portion in a temporary file.
- For **ClockSync** parameter
 - Change order from 8 to 7
- For **TC_Width** parameter
 - Change order to from 9 to 8

CRET

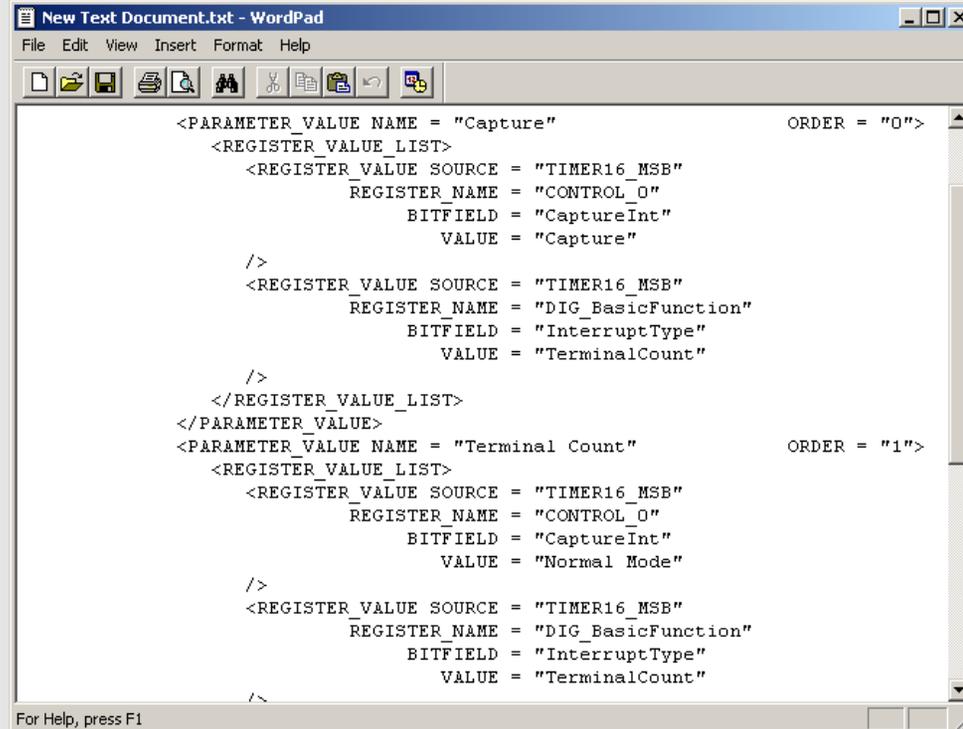
```
</REGISTER_VALUE_LIST>
</PARAMETER_VALUE>
</PARAMETER_VALUE_LIST>
</PARAMETER>
<PARAMETER NAME = "InterruptType"
ORDER = "7"
TYPE = "BLOCK"
>
  <PARAMETER_VALUE_LIST>
    <PARAMETER_VALUE NAME = "Capture" ORDER = "0">
      <REGISTER_VALUE_LIST>
        <REGISTER_VALUE SOURCE = "TIMER16_MSB"
REGISTER_NAME = "CONTROL_0"
BITFIELD = "CaptureInt"
VALUE = "Capture"
/>
        <REGISTER_VALUE SOURCE = "TIMER16_MSB"
REGISTER_NAME = "DIG_BasicFunction"
BITFIELD = "InterruptType"
VALUE = "TerminalCount"
/>
      </REGISTER_VALUE_LIST>
    </PARAMETER_VALUE>
    <PARAMETER_VALUE NAME = "Terminal Count" ORDER = "1">
      <REGISTER_VALUE_LIST>
        <REGISTER_VALUE SOURCE = "TIMER16_MSB"
REGISTER_NAME = "CONTROL_0"
BITFIELD = "CaptureInt"
VALUE = "Normal Mode"
/>
        <REGISTER_VALUE SOURCE = "TIMER16_MSB"
REGISTER_NAME = "DIG_BasicFunction"
BITFIELD = "InterruptType"
VALUE = "TerminalCount"
/>
      </REGISTER_VALUE_LIST>
    </PARAMETER_VALUE>
    <PARAMETER_VALUE NAME = "Compare True" ORDER = "2">
      <REGISTER_VALUE_LIST>
        <REGISTER_VALUE SOURCE = "TIMER16_MSB"
REGISTER_NAME = "CONTROL_0"
BITFIELD = "CaptureInt"
VALUE = "Normal Mode"
/>
        <REGISTER_VALUE SOURCE = "TIMER16_MSB"
REGISTER_NAME = "DIG_BasicFunction"
BITFIELD = "InterruptType"
VALUE = "CompareTrue"
/>
      </REGISTER_VALUE_LIST>
    </PARAMETER_VALUE>
  </PARAMETER_VALUE_LIST>
</PARAMETER>
<PARAMETER NAME = "ClockSync"
ORDER = "7"
TYPE = "BLOCK"
>
```

User Modules

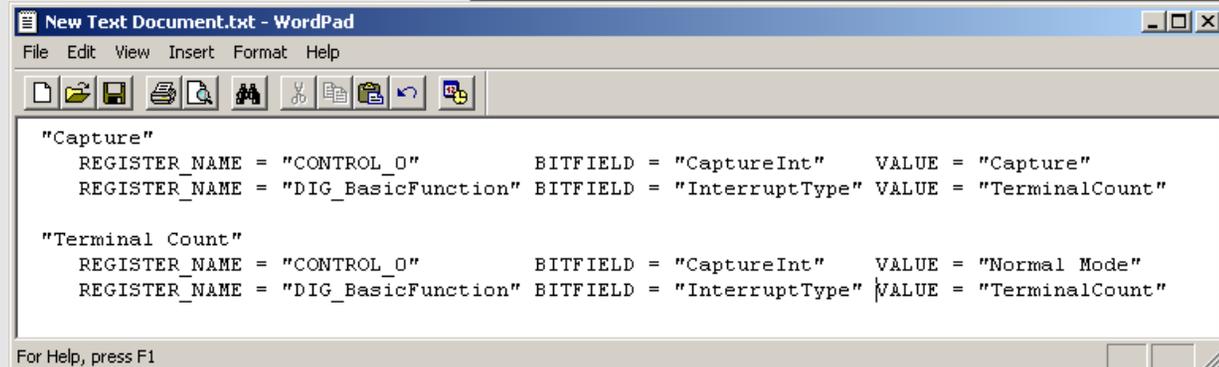
Step 7 *Add Interrupts to SHAPE*

Open temporary file.

- The **REGISTER_NAME**, **BITFIELD**, and **VALUE** for each type of interrupt can be used to define the personalize the requires registers.



```
<PARAMETER_VALUE NAME = "Capture" ORDER = "0">
  <REGISTER_VALUE_LIST>
    <REGISTER_VALUE SOURCE = "TIMER16_MSB"
      REGISTER_NAME = "CONTROL_0"
      BITFIELD = "CaptureInt"
      VALUE = "Capture"
    />
    <REGISTER_VALUE SOURCE = "TIMER16_MSB"
      REGISTER_NAME = "DIG_BasicFunction"
      BITFIELD = "InterruptType"
      VALUE = "TerminalCount"
    />
  </REGISTER_VALUE_LIST>
</PARAMETER_VALUE>
<PARAMETER_VALUE NAME = "Terminal Count" ORDER = "1">
  <REGISTER_VALUE_LIST>
    <REGISTER_VALUE SOURCE = "TIMER16_MSB"
      REGISTER_NAME = "CONTROL_0"
      BITFIELD = "CaptureInt"
      VALUE = "Normal Mode"
    />
    <REGISTER_VALUE SOURCE = "TIMER16_MSB"
      REGISTER_NAME = "DIG_BasicFunction"
      BITFIELD = "InterruptType"
      VALUE = "TerminalCount"
    />
  </REGISTER_VALUE_LIST>
</PARAMETER_VALUE>
```



```
"Capture"
REGISTER_NAME = "CONTROL_0" BITFIELD = "CaptureInt" VALUE = "Capture"
REGISTER_NAME = "DIG_BasicFunction" BITFIELD = "InterruptType" VALUE = "TerminalCount"

"Terminal Count"
REGISTER_NAME = "CONTROL_0" BITFIELD = "CaptureInt" VALUE = "Normal Mode"
REGISTER_NAME = "DIG_BasicFunction" BITFIELD = "InterruptType" VALUE = "TerminalCount"
```

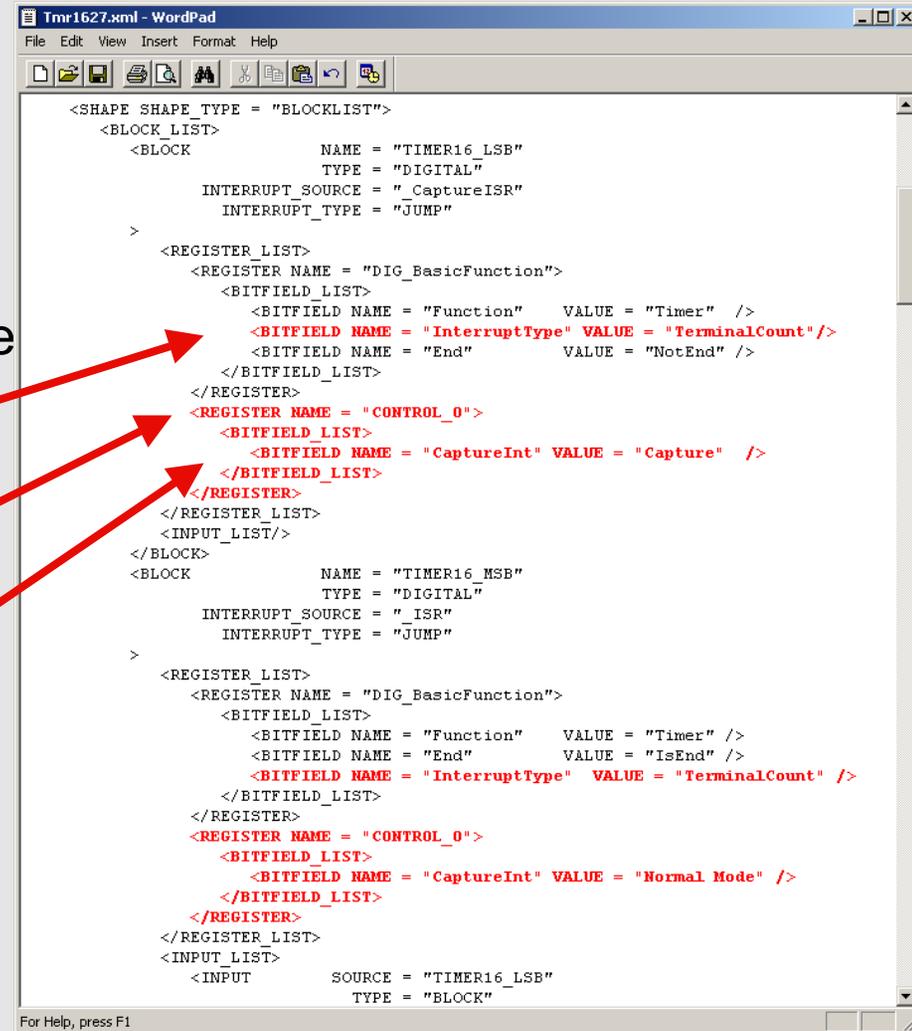
User Modules

Step 7b *Add Interrupts to SHAPE*

- Open **Tmr1627.XML**

For **TIMER16_LSB**

- Add **“InterruptType”** bit field to the **“DIG_BasicFunction”** register.
 - Set its value to **“TerminalCount”**
- Add **CONTROL_0** register
 - Add **“CaptureInt”** bit field to it.
 - Set its value to **“Capture”**.



```
Tmr1627.xml - WordPad
File Edit View Insert Format Help

<SHAPE SHAPE_TYPE = "BLOCKLIST">
  <BLOCK_LIST>
    <BLOCK
      NAME = "TIMER16_LSB"
      TYPE = "DIGITAL"
      INTERRUPT_SOURCE = "_CaptureISR"
      INTERRUPT_TYPE = "JUMP"
    >
      <REGISTER_LIST>
        <REGISTER NAME = "DIG_BasicFunction">
          <BITFIELD_LIST>
            <BITFIELD NAME = "Function" VALUE = "Timer" />
            <BITFIELD NAME = "InterruptType" VALUE = "TerminalCount" />
            <BITFIELD NAME = "End" VALUE = "NotEnd" />
          </BITFIELD_LIST>
        </REGISTER>
        <REGISTER NAME = "CONTROL_0">
          <BITFIELD_LIST>
            <BITFIELD NAME = "CaptureInt" VALUE = "Capture" />
          </BITFIELD_LIST>
        </REGISTER>
      </REGISTER_LIST>
    </BLOCK>
    <BLOCK
      NAME = "TIMER16_MSB"
      TYPE = "DIGITAL"
      INTERRUPT_SOURCE = "_ISR"
      INTERRUPT_TYPE = "JUMP"
    >
      <REGISTER_LIST>
        <REGISTER NAME = "DIG_BasicFunction">
          <BITFIELD_LIST>
            <BITFIELD NAME = "Function" VALUE = "Timer" />
            <BITFIELD NAME = "End" VALUE = "IsEnd" />
            <BITFIELD NAME = "InterruptType" VALUE = "TerminalCount" />
          </BITFIELD_LIST>
        </REGISTER>
        <REGISTER NAME = "CONTROL_0">
          <BITFIELD_LIST>
            <BITFIELD NAME = "CaptureInt" VALUE = "Normal Mode" />
          </BITFIELD_LIST>
        </REGISTER>
      </REGISTER_LIST>
    </BLOCK>
  </BLOCK_LIST>
  <INPUT_LIST>
    <INPUT
      SOURCE = "TIMER16_LSB"
      TYPE = "BLOCK"
    >
  </INPUT_LIST>
</SHAPE>
```

User Modules

Step 7c Add Interrupts to SHAPE

For TIMER16_MSB

- Add “**InterruptType**” bit field to the “**DIG_BasicFunction**” register.
 - Set its value to “**TerminalCount**”
- Add **CONTROL_0** register
- Add “**CaptureInt**” bit field to it.
 - Set its value to “**Normal Mode**”

CRET

All XML changes are done!

```
<SHAPE SHAPE_TYPE = "BLOCKLIST">
  <BLOCK_LIST>
    <BLOCK
      NAME = "TIMER16_LSB"
      TYPE = "DIGITAL"
      INTERRUPT_SOURCE = "_CaptureISR"
      INTERRUPT_TYPE = "JUMP"
    >
      <REGISTER_LIST>
        <REGISTER NAME = "DIG_BasicFunction">
          <BITFIELD_LIST>
            <BITFIELD NAME = "Function" VALUE = "Timer" />
            <BITFIELD NAME = "InterruptType" VALUE = "TerminalCount" />
            <BITFIELD NAME = "End" VALUE = "NotEnd" />
          </BITFIELD_LIST>
        </REGISTER>
        <REGISTER NAME = "CONTROL_0">
          <BITFIELD_LIST>
            <BITFIELD NAME = "CaptureInt" VALUE = "Capture" />
          </BITFIELD_LIST>
        </REGISTER>
      </REGISTER_LIST>
    </BLOCK>
    <BLOCK
      NAME = "TIMER16_MSB"
      TYPE = "DIGITAL"
      INTERRUPT_SOURCE = "_ISR"
      INTERRUPT_TYPE = "JUMP"
    >
      <REGISTER_LIST>
        <REGISTER NAME = "DIG_BasicFunction">
          <BITFIELD_LIST>
            <BITFIELD NAME = "Function" VALUE = "Timer" />
            <BITFIELD NAME = "End" VALUE = "IsEnd" />
            <BITFIELD NAME = "InterruptType" VALUE = "TerminalCount" />
          </BITFIELD_LIST>
        </REGISTER>
        <REGISTER NAME = "CONTROL_0">
          <BITFIELD_LIST>
            <BITFIELD NAME = "CaptureInt" VALUE = "Normal Mode" />
          </BITFIELD_LIST>
        </REGISTER>
      </REGISTER_LIST>
    </BLOCK>
  </BLOCK_LIST>
  <INPUT_LIST>
    <INPUT
      SOURCE = "TIMER16_LSB"
      TYPE = "BLOCK"
    >
  </INPUT_LIST>
</SHAPE>
```

User Modules

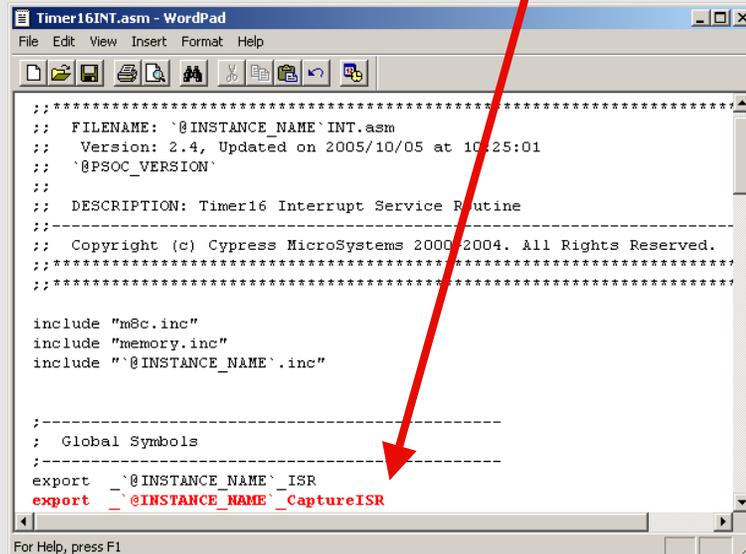
To modify software for this UM

- A new interrupt handler must be added for the new interrupt.
- API must be changed to pass the intended interrupt to enable or disable type .
- An new interrupt mask must be generated for this interrupt.
 - For both C (.h) and assembly (.inc)
- Placement file must be change for 29x parts so both interrupt masks are always in the same interrupt register.

User Modules

Step 8 *Add interrupt handler.*

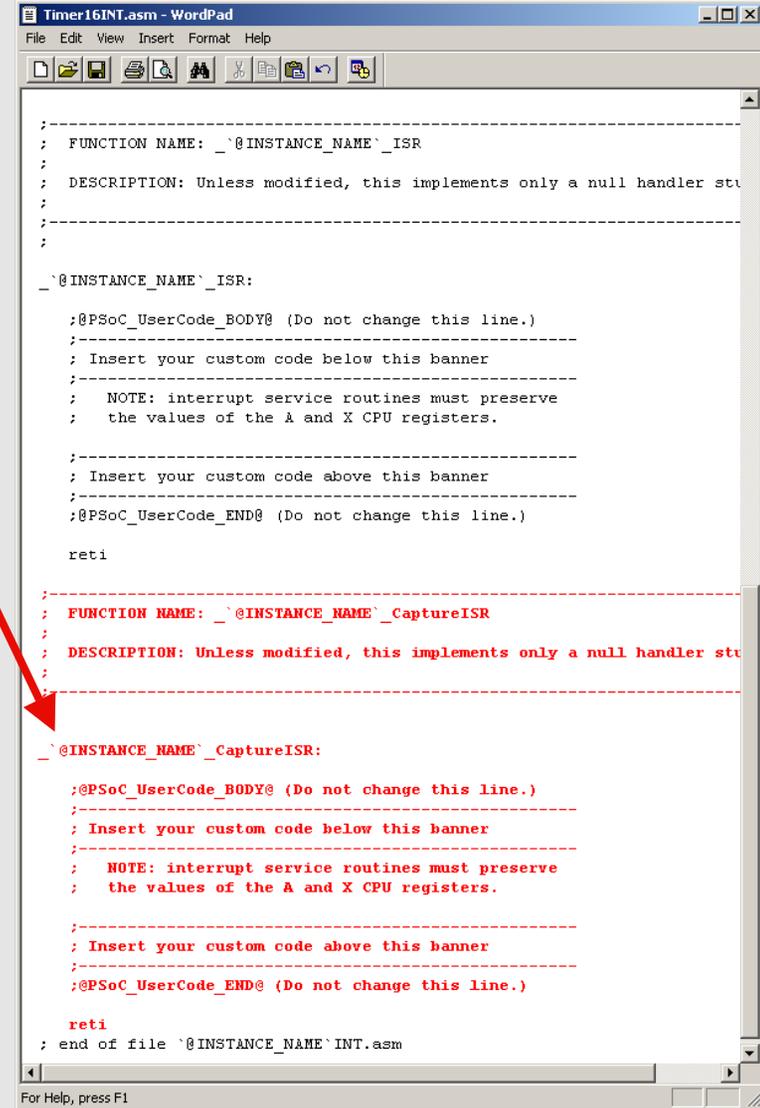
- Back up a directory
- Open **Timer16INT.asm**
- Duplicate the **_ISR** function and rename to **_CaptureISR**
- Add export for new Label



```
Timer16INT.asm - WordPad
File Edit View Insert Format Help

;*****
; FILENAME: `@INSTANCE_NAME`INT.asm
; Version: 2.4, Updated on 2005/10/05 at 10:25:01
; `@PSOC_VERSION`
;
; DESCRIPTION: Timer16 Interrupt Service Routine
;-----
; Copyright (c) Cypress MicroSystems 2000/2004. All Rights Reserved.
;*****
include "m0c.inc"
include "memory.inc"
include "`@INSTANCE_NAME`.inc"

;-----
; Global Symbols
;-----
export _`@INSTANCE_NAME`_ISR
export _`@INSTANCE_NAME`_CaptureISR
```



```
Timer16INT.asm - WordPad
File Edit View Insert Format Help

;-----
; FUNCTION NAME: _`@INSTANCE_NAME`_ISR
;
; DESCRIPTION: Unless modified, this implements only a null handler stu
;-----
;
;-----
_`@INSTANCE_NAME`_ISR:
;@PSoC_UserCode_BODY@ (Do not change this line.)
;-----
; Insert your custom code below this banner
;-----
; NOTE: interrupt service routines must preserve
; the values of the A and X CPU registers.
;-----
; Insert your custom code above this banner
;-----
;@PSoC_UserCode_END@ (Do not change this line.)

reti

;-----
; FUNCTION NAME: _`@INSTANCE_NAME`_CaptureISR
;
; DESCRIPTION: Unless modified, this implements only a null handler stu
;-----
;
;-----
_`@INSTANCE_NAME`_CaptureISR:
;@PSoC_UserCode_BODY@ (Do not change this line.)
;-----
; Insert your custom code below this banner
;-----
; NOTE: interrupt service routines must preserve
; the values of the A and X CPU registers.
;-----
; Insert your custom code above this banner
;-----
;@PSoC_UserCode_END@ (Do not change this line.)

reti
; end of file `@INSTANCE_NAME`INT.asm
```

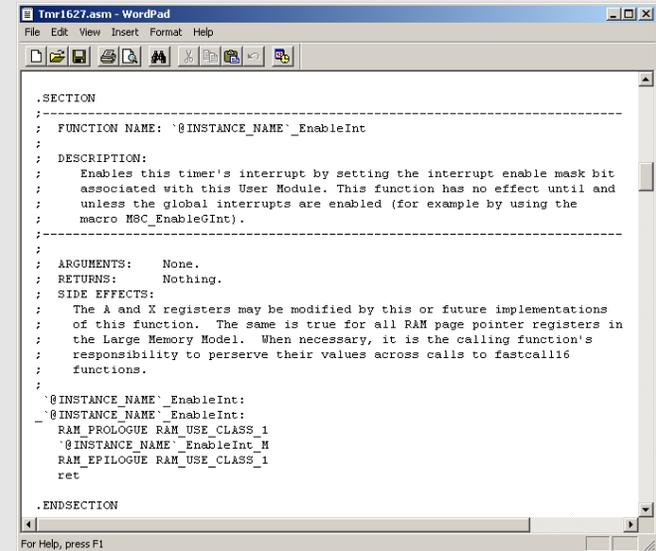
User Modules

Step 9 *Change APIs*

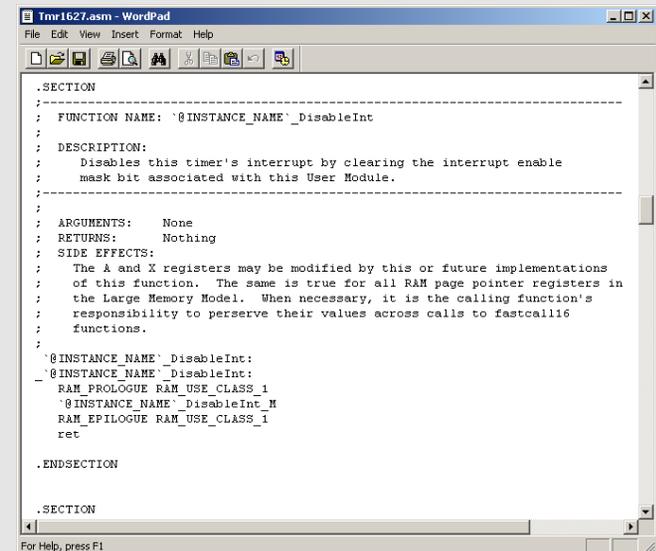
- Open **CY8C27**
- Open **Tmr1627.asm**

Both APIs are macros. These will both be changed when Tmr1627.inc is changed.

Although not needed for this particular example. This step is still shown. Many UMs you develop will require some assembly code alteration.



```
.SECTION
-----
; FUNCTION NAME: `@INSTANCE_NAME`_EnableInt
;
; DESCRIPTION:
;   Enables this timer's interrupt by setting the interrupt enable mask bit
;   associated with this User Module. This function has no effect until and
;   unless the global interrupts are enabled (for example by using the
;   macro MSC_EnableGInt).
;-----
; ARGUMENTS:   None.
; RETURNS:     Nothing.
; SIDE EFFECTS:
;   The A and X registers may be modified by this or future implementations
;   of this function. The same is true for all RAM page pointer registers in
;   the Large Memory Model. When necessary, it is the calling function's
;   responsibility to preserve their values across calls to fastcall16
;   functions.
;
;_@INSTANCE_NAME`_EnableInt:
-@INSTANCE_NAME`_EnableInt:
  RAM_PROLOGUE RAM_USE_CLASS_1
  @INSTANCE_NAME`_EnableInt_M
  RAM_EPILOGUE RAM_USE_CLASS_1
  ret
.ENDSECTION
```



```
.SECTION
-----
; FUNCTION NAME: `@INSTANCE_NAME`_DisableInt
;
; DESCRIPTION:
;   Disables this timer's interrupt by clearing the interrupt enable
;   mask bit associated with this User Module.
;-----
; ARGUMENTS:   None
; RETURNS:     Nothing
; SIDE EFFECTS:
;   The A and X registers may be modified by this or future implementations
;   of this function. The same is true for all RAM page pointer registers in
;   the Large Memory Model. When necessary, it is the calling function's
;   responsibility to preserve their values across calls to fastcall16
;   functions.
;
;_@INSTANCE_NAME`_DisableInt:
-@INSTANCE_NAME`_DisableInt:
  RAM_PROLOGUE RAM_USE_CLASS_1
  @INSTANCE_NAME`_DisableInt_M
  RAM_EPILOGUE RAM_USE_CLASS_1
  ret
.ENDSECTION

.SECTION
```

User Modules

Step 10 *Alter include file.*

- Open **Tmr1627.inc**
- Duplicate the **_INT_MASK** declaration and change for new **_CaptureINT_MASK** declaration.
- Change enable and disable interrupt macros to use A as the enabling and disabling mask.

The top screenshot shows the original code in `Tmr1627.inc`. The line `@INSTANCE_NAME`_INT_MASK: @CTAB44`equ 0x@TIMER16_MSB_ISR_MASK` is highlighted in red. A red arrow points from the second list item to this line.

```
;;*****  
;; FILENAME: `@INSTANCE_NAME`.inc  
;; Version: 2.4, Updated on 2005/10/05 at 10:25:01  
;; `@PSOC_VERSION`  
;;  
;; DESCRIPTION: Assembler declarations for the Timer16 user module interface  
;;               for the 22/24/27/29xxx PSoC family of devices  
;;-----  
;; Copyright (c) Cypress MicroSystems 2000-2004. All Rights Reserved.  
;;*****  
include "m8c.inc"  
  
;-----  
; Constants for `@INSTANCE_NAME` API's.  
;-----  
  
`@INSTANCE_NAME`_CONTROL_REG_START_BIT: `@CTAB44`equ 0x01 ; Control register start bit  
`@INSTANCE_NAME`_INT_REG: `@CTAB44`equ 0x@TIMER16_MSB_ISR_ADDR`  
`@INSTANCE_NAME`_INT_MASK: `@CTAB44`equ 0x@TIMER16_MSB_ISR_MASK`  
`@INSTANCE_NAME`_CaptureINT_MASK: `@CTAB44`equ 0x@TIMER16_LSB_CaptureISR_MASK`
```

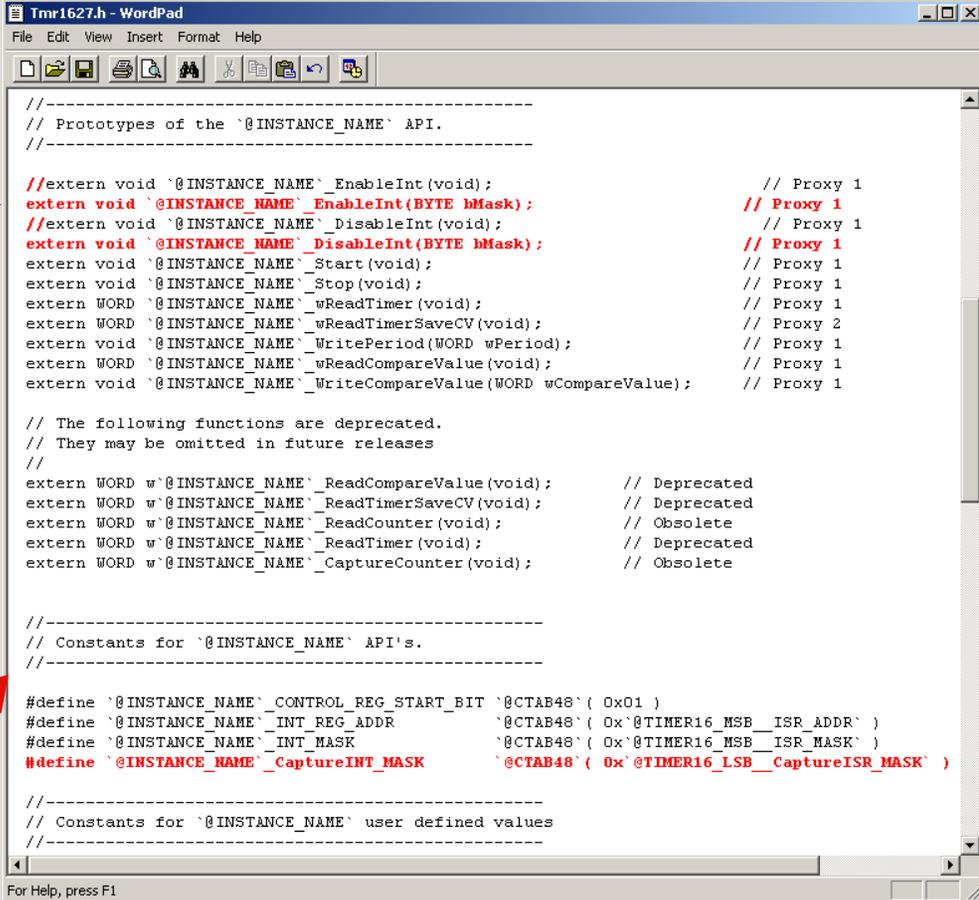
The bottom screenshot shows the modified code. The new macro `_CaptureINT_MASK` is added, and the original `_INT_MASK` macro is updated to use register A. Red arrows point from the third list item to these changes.

```
macro `@INSTANCE_NAME`_EnableInt_M  
;MSC_EnableIntMask `@INSTANCE_NAME`_INT_REG, `@INSTANCE_NAME`_INT_MASK  
mov X,SP  
push A  
mov A,reg[`@INSTANCE_NAME`_INT_REG]  
or A,[X]  
mov reg[`@INSTANCE_NAME`_INT_REG],A  
pop A  
endm  
  
macro `@INSTANCE_NAME`_DisableInt_M  
;MSC_DisableIntMask `@INSTANCE_NAME`_INT_REG, `@INSTANCE_NAME`_INT_MASK  
mov X,SP  
cpl A  
push A  
mov A,reg[`@INSTANCE_NAME`_INT_REG]  
and A,[X]  
mov reg[`@INSTANCE_NAME`_INT_REG],A  
pop A  
endm
```

User Modules

Step 11 *Alter .h file.*

- Open Tmr1627.h
- Change EnableInt and DisableInt function prototypes to require a BYTE argument.
- Duplicate the **_INT_MASK** #define and change for new **_CaptureINT_MASK** define



```
Tmr1627.h - WordPad
File Edit View Insert Format Help

//-----
// Prototypes of the `@INSTANCE_NAME` API.
//-----

//extern void `@INSTANCE_NAME`_EnableInt(void); // Proxy 1
extern void `@INSTANCE_NAME`_EnableInt(BYTE bMask); // Proxy 1
//extern void `@INSTANCE_NAME`_DisableInt(void); // Proxy 1
extern void `@INSTANCE_NAME`_DisableInt(BYTE bMask); // Proxy 1
extern void `@INSTANCE_NAME`_Start(void); // Proxy 1
extern void `@INSTANCE_NAME`_Stop(void); // Proxy 1
extern WORD `@INSTANCE_NAME`_wReadTimer(void); // Proxy 1
extern WORD `@INSTANCE_NAME`_wReadTimerSaveCV(void); // Proxy 2
extern void `@INSTANCE_NAME`_WritePeriod(WORD wPeriod); // Proxy 1
extern WORD `@INSTANCE_NAME`_wReadCompareValue(void); // Proxy 1
extern void `@INSTANCE_NAME`_WriteCompareValue(WORD wCompareValue); // Proxy 1

// The following functions are deprecated.
// They may be omitted in future releases
//
extern WORD w`@INSTANCE_NAME`_ReadCompareValue(void); // Deprecated
extern WORD w`@INSTANCE_NAME`_ReadTimerSaveCV(void); // Deprecated
extern WORD w`@INSTANCE_NAME`_ReadCounter(void); // Obsolete
extern WORD w`@INSTANCE_NAME`_ReadTimer(void); // Deprecated
extern WORD w`@INSTANCE_NAME`_CaptureCounter(void); // Obsolete

//-----
// Constants for `@INSTANCE_NAME` API's.
//-----

#define `@INSTANCE_NAME`_CONTROL_REG_START_BIT `@CTAB48` ( 0x01 )
#define `@INSTANCE_NAME`_INT_REG_ADDR `@CTAB48` ( 0x`@TIMER16_MSB_ISR_ADDR` )
#define `@INSTANCE_NAME`_INT_MASK `@CTAB48` ( 0x`@TIMER16_MSB_ISR_MASK` )
#define `@INSTANCE_NAME`_CaptureINT_MASK `@CTAB48` ( 0x`@TIMER16_LSB_CaptureISR_MASK` )

//-----
// Constants for `@INSTANCE_NAME` user defined values
//-----

For Help, press F1
```

User Modules

Step 12

Alter 29x placement file

Placement files list all legal placements for a particular chip family.

Only the 29x family allows the placement so the each block has a different interrupt register. *DCB13 DBB20*

- Open **Tmr1627CY8C2900.plc**
- Remove **RESOURCE_PLACEMENT_INDEX="7"**
- Renumber all following indexes.

```
Tmr1627CY8C2900.plc - WordPad
File Edit View Insert Format Help
</RESOURCE_PLACEMENT_LIST>
</RESOURCE_LOCATION>
<RESOURCE_LOCATION RESOURCE_PLACEMENT_INDEX="6">
  <RESOURCE_PLACEMENT_LIST>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_LSB" TARGET_RESOURCE_NAME="DCB12"/>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_MSB" TARGET_RESOURCE_NAME="DCB13"/>
  </RESOURCE_PLACEMENT_LIST>
</RESOURCE_LOCATION>
<del><RESOURCE_LOCATION RESOURCE_PLACEMENT_INDEX="7">
  <RESOURCE_PLACEMENT_LIST>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_LSB" TARGET_RESOURCE_NAME="DCB13"/>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_MSB" TARGET_RESOURCE_NAME="DBB20"/>
  </RESOURCE_PLACEMENT_LIST>
</del></RESOURCE_LOCATION>
<RESOURCE_LOCATION RESOURCE_PLACEMENT_INDEX="7">
  <RESOURCE_PLACEMENT_LIST>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_LSB" TARGET_RESOURCE_NAME="DBB20"/>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_MSB" TARGET_RESOURCE_NAME="DBB21"/>
  </RESOURCE_PLACEMENT_LIST>
</RESOURCE_LOCATION>
<RESOURCE_LOCATION RESOURCE_PLACEMENT_INDEX="8">
  <RESOURCE_PLACEMENT_LIST>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_LSB" TARGET_RESOURCE_NAME="DBB21"/>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_MSB" TARGET_RESOURCE_NAME="DCB22"/>
  </RESOURCE_PLACEMENT_LIST>
</RESOURCE_LOCATION>
<RESOURCE_LOCATION RESOURCE_PLACEMENT_INDEX="9">
  <RESOURCE_PLACEMENT_LIST>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_LSB" TARGET_RESOURCE_NAME="DCB22"/>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_MSB" TARGET_RESOURCE_NAME="DCB23"/>
  </RESOURCE_PLACEMENT_LIST>
</RESOURCE_LOCATION>
<RESOURCE_LOCATION RESOURCE_PLACEMENT_INDEX="10">
  <RESOURCE_PLACEMENT_LIST>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_LSB" TARGET_RESOURCE_NAME="DCB23"/>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_MSB" TARGET_RESOURCE_NAME="DBB30"/>
  </RESOURCE_PLACEMENT_LIST>
</RESOURCE_LOCATION>
<RESOURCE_LOCATION RESOURCE_PLACEMENT_INDEX="11">
  <RESOURCE_PLACEMENT_LIST>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_LSB" TARGET_RESOURCE_NAME="DBB30"/>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_MSB" TARGET_RESOURCE_NAME="DBB31"/>
  </RESOURCE_PLACEMENT_LIST>
</RESOURCE_LOCATION>
<RESOURCE_LOCATION RESOURCE_PLACEMENT_INDEX="12">
  <RESOURCE_PLACEMENT_LIST>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_LSB" TARGET_RESOURCE_NAME="DBB31"/>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_MSB" TARGET_RESOURCE_NAME="DCB32"/>
  </RESOURCE_PLACEMENT_LIST>
</RESOURCE_LOCATION>
<RESOURCE_LOCATION RESOURCE_PLACEMENT_INDEX="13">
  <RESOURCE_PLACEMENT_LIST>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_LSB" TARGET_RESOURCE_NAME="DCB32"/>
    <RESOURCE_PLACEMENT UM_RESOURCE_NAME="TIMER16_MSB" TARGET_RESOURCE_NAME="DCB33"/>
  </RESOURCE_PLACEMENT_LIST>
</RESOURCE_LOCATION>
</RESOURCE_LOCATION_LIST>
</SHAPE_LOCATOR>
</SHAPE_LOCATOR_LIST>
</PSOC_DEVICE_DB>
For Help, press F1
```

User Modules

Final Step.

- Start up Designer for a 29x part
- Verify the correct placement combinations.
- Generate application
- Verify that API compile correctly

Summary

- Each User Module is a combination of information on the intraconnections of PSoC resources, the software to control it, an icon, and a data sheet.
- It is possible to generate new or modify existing ones
- Different UMs can be combined to produce a new UM
- New UMs can be old ones with no change to the hardware. Only changes to the your custom APIs.

Questions