Hands-On Exercise (PWM_ADC_SYN)
Approach

- Use Processor Expert Beans to implement Target System application
  - Use 56F83xx PWM0 with Internal TimerC Channel 2 to trigger ADCA (Channel 0).
  - Download and Execute on 56F83xx EVM
ADC Synchronization from the PWM Module

56F83xx

PWM Module

Sync pulse

Timer C

Time delay

ADC Trigger

ADC

6 PWMs w/ dead-time

Optional External Sync

PWM Signal

Sync Pulse

ADC Trigger

Time delay

Convert anywhere within the PWM cycle
PWM, QUADTimer and ADC can communicate internally with no CPU support.
ADC Sampling helps to filtering the measured current - antialiasing.

Noise free ADC sampling when the power switch is not acting

ADC sample is taken when shunt resistor signal (information) is available
ADC Synchronization with the PWM

Competitor Solution → 568300 Solution

PWM Output → PWM Synch signal

ISR Latency

Timer Delay

ADC Convert

Control Algorithm Execution

PWM values Written to registers

PWM values Updated
Select “Processor Expert Stationery” and put the project name to there.
Select and double click ADC inside “CPU Internal Peripherals - Converter Beans”
Select Enable internal trigger.
Select “Yes” for Sync from PWM
Kick Trigger Source and select “Add new shared bean from template TriggerTimer”.

Additional note: This step involves a specific configuration in a software tool, likely for a microcontroller or digital signal processing application, related to adding a new trigger timer bean.
Kick "yes"
Select Timer C channel 2 “TMRC2”
### Operation mode

- **Count once**: Counter 0 input pin
- **Count length**: Counter 0 input pin
- **Count direction**:
  - TMF1: 0
  - TMF2: 0
- **Master mode**: Disabled
- **External OFlag force**: Disabled
- **Forced OFlag value**: Disabled
- **Force OFlag output**: Disabled
- **Output enable**: No
- **Output polarity**: True
- **Input polarity**: True
- **Capture channel**: Disabled
- **Input capture mode**: Disabled
- **OutputMode**: Toggle OFlag output on successful compare count
- **Compare load control 1**: Disabled
- **Compare load control 2**: Disabled
- **Pin**: 0

### Interrupts

- **Interrupt channel**: INT_TMFC2, INT_TMRC2
- **Interrupt compare interrupt**: Enabled
- **Timer overflow interrupt**: Enabled
- **Input edge interrupt**: Enabled
- **Timer compare 1 interrupt**: Enabled
- **Timer compare 2 interrupt**: Enabled
- **Interrupt priority**: Medium priority
### DDFAE Training

#### PWM_ADC_SYN_8323.mcp

**Properties**

<table>
<thead>
<tr>
<th>Bean</th>
<th>Visibility</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Events**

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Comment**

<table>
<thead>
<tr>
<th>Comment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Settings**

- **Clock settings**
  - **Primary source**: prescaler (IP BUS clock)
  - **Secondary source**: counter 2 input pin

- **Operation mode**: Triggered count mode
  - **Count once**: count repeatedly
  - **Count length**: count till compare, then reinitialize
  - **Count direction**: up

- **Master mode**: Disabled
- **External OFLAG force**: Disabled
- **Forced UFLAG value**: Disabled
- **Force OFLAG output**: Disabled
- **Output enable**: yes
- **Output polarity**: true
- **Input polarity**: true
- **Cochannel initialization**: Disabled
- **Input capture mode**: Disabled
- **Compare load control**: Enabled
- **Load upon successful compare**: with the value in TNRx_CMP1
- **Compare load control 2**: Disabled

**Pins**

- **Pins**

**Interrupts**

- **Timer Channel**
  - **Interrupt**: INT_TMFC2
  - **Timer compare interrupt**: Disabled
  - **Timer overflow interrupt**: Disabled
  - **Input edge interrupt**: Disabled
  - **Timer compare 1 interrupt**: Disabled
  - **Timer compare 2 interrupt**: Disabled

**Output pins**

<table>
<thead>
<tr>
<th>Output pins</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2004
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>TMRC2</td>
</tr>
<tr>
<td>Settings</td>
<td></td>
</tr>
<tr>
<td>Clock settings</td>
<td></td>
</tr>
<tr>
<td>Primary source</td>
<td>prescaler [IP BUS clock]</td>
</tr>
<tr>
<td>Secondary source</td>
<td></td>
</tr>
<tr>
<td>Operation mode</td>
<td>Triggered count mode</td>
</tr>
<tr>
<td>Count once</td>
<td>count repeatedly</td>
</tr>
<tr>
<td>Counter length</td>
<td>count till compare, then reinitialize</td>
</tr>
<tr>
<td>Counter direction</td>
<td>up</td>
</tr>
<tr>
<td>Master mode</td>
<td>Disabled</td>
</tr>
<tr>
<td>External OFLAG enable</td>
<td>Disabled</td>
</tr>
<tr>
<td>Forced OFLAG enable</td>
<td>Disabled</td>
</tr>
<tr>
<td>Force OFLAG enable</td>
<td>Disabled</td>
</tr>
<tr>
<td>Output enable</td>
<td>false</td>
</tr>
<tr>
<td>Output polarity</td>
<td>true</td>
</tr>
<tr>
<td>Input polarity</td>
<td>true</td>
</tr>
<tr>
<td>Compare load control 1</td>
<td>Enabled</td>
</tr>
<tr>
<td>Compare load control 2</td>
<td>Disabled</td>
</tr>
<tr>
<td>Interrupts</td>
<td></td>
</tr>
<tr>
<td>Timer Channel</td>
<td></td>
</tr>
<tr>
<td>Registers</td>
<td></td>
</tr>
<tr>
<td>Timer Compare register</td>
<td>10</td>
</tr>
<tr>
<td>Timer Compare register 2</td>
<td>0</td>
</tr>
<tr>
<td>Timer Load register</td>
<td>0</td>
</tr>
<tr>
<td>Timer Capture register</td>
<td>0</td>
</tr>
<tr>
<td>Timer Comparison Load register 1</td>
<td>10</td>
</tr>
<tr>
<td>Timer Comparison Load register 2</td>
<td>0</td>
</tr>
</tbody>
</table>
Double click AD1:ADC for PWM
Select PWM source
Kick "..." box and select "Add new shared bean PWMMC."
<table>
<thead>
<tr>
<th>Properties</th>
<th>Methods</th>
<th>Events</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bean name</td>
<td>PWM1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device</td>
<td>PWM_A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Align</td>
<td>center-aligned mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode of PWM Pair 0</td>
<td>independent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode of PWM Pair 1</td>
<td>independent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode of PWM Pair 2</td>
<td>independent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Top-Side PWM Fair 0 Polarity</td>
<td>Positive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Top-Side PWM Fair 1 Polarity</td>
<td>Positive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Top-Side PWM Fair 2 Polarity</td>
<td>Positive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottom-Side PWM Fair 0 Polarity</td>
<td>Positive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottom-Side PWM Fair 1 Polarity</td>
<td>Positive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottom-Side PWM Fair 2 Polarity</td>
<td>Positive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write Protect</td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output pads</td>
<td>Disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enable in Wait mode</td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enable in E.nt mode</td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>...</td>
<td>Unassigned timing</td>
<td></td>
</tr>
<tr>
<td>Output Frequency</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Same frequency in modes</td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PWMA</td>
<td>PWM_A prescaler</td>
<td>Auto selected prescaler</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Roboc</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Half cycle reload</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Hardware acceleration</td>
<td>Disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dead time</td>
<td>...</td>
<td>Unassigned timing</td>
<td></td>
</tr>
<tr>
<td>Correction</td>
<td>Disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interrupt service/event</td>
<td>Disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channel 0</td>
<td>Channel</td>
<td>PWMA0</td>
<td>PWMA0</td>
</tr>
<tr>
<td></td>
<td>PWM pin</td>
<td>PWMA0_GPIOA0</td>
<td>PWMA0_GPIOA0</td>
</tr>
<tr>
<td></td>
<td>PWM pin signal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Duty</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fin PWM0 active</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Output software control</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BASC</td>
<td>ADVANCED</td>
<td>EXPERT</td>
</tr>
</tbody>
</table>
Kick “…” box to input PWM frequency.
Input 50kHz
Kick “…” box to input dead time value.
Input 0
● Disable Channel 1-5
● Disable all Fault inputs
DDFAE Training
```c
/**
 * Module PWM_ADC_SYN_8323

# include "Cpu.h"
# include "Events.h"
```

- **Filename**: PWM_ADC_SYN_8323.c
- **Project**: PWM_ADC_SYN_8323
- **Processor**: 56F8323
- **Version**: Driver 01.09
- **Compiler**: Metrowerks DSP C Compiler
- **Date/Time**: 10/24/2005, 5:09 PM
- **Abstract**:
  - **Main Module**
  - Here is to be placed user's code.
- **Settings**:
  - **Contents**:
    - No public methods
- **Copyright**: (c) Copyright UNIS. spol. s r.o. 1997-2004
- **UNIS. spol. s r.o.**
  - Jandruska 33
  - 624 00 Brno
  - Czech Republic
- **Website**: www.processorexpert.com
- **Email**: info@processorexpert.com

---

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.
Add the code for ADC conversion timing test

Put ADC PE bean EnableIntTrigger by Drag-n-Drop method.
Open Event.c event

```c
/* MODULE Event */
#include "Cpu.h"
#include "Events.h"

/*
 Event : AD1_OnEnd (module Events)
 From bean : AD1 [ADC]
 Description:
 This event is called after the measurement (which consists of <1 or more conversions>) is/are finished.
 Parameters : None
 Returns : Nothing

***************************************************************************/

/*
#pragma interrupt called /* Comment this line if the appropriate 'Interru;
 is set to 'yes' */#pragma interrupt saveall is

void AD1_OnEnd(void)
{
    /* Write your code here ... */
}

/* END Events */

***************************************************************************/

This file was created by UNIS Processor Expert 2.96 [03.65]
for the Freescale 68800 series of microcontrollers.

***************************************************************************/
Add Global variables

Add GPIO pin for timing test

Put ADC PE bean GetValue16 by Drag-n-Drop method.
DDFAE Training

Slide 42

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.
© Freescale Semiconductor, Inc. 2004

Click debug
/**
 * @brief Example code for PWM_ADC_SYN_0323

 * This file was created by UNIS Processor Expert 2.96 [03.65]
 * for the Freescale 56800 series of microcontrollers.
 */

void AD1_OnEnd(void)
{
    /* Write your code here ... */

    unsigned int x=0x8;
    if(AD1_GetValue16(&x)!ERR_OK)
    {
        /* Code goes here */
    }
    else
    {
        /* Code goes here */
    }

    /* END Events */

    /*
     * 
     * This file was created by UNIS Processor Expert 2.96 [03.65]
     * for the Freescale 56800 series of microcontrollers.
     */
    
    /*
     ***********************************************************/
Program "sdm_pROM_xRAM.elf" is executing. Choose Break from the Debug menu to stop it.
Result 1:

In this example (PWM-Center Aligned mode): ADC conversion time + timer C channel 2 delay $\approx 4.4$ usec
Change to Edge Aligned
Click debug
Click Run

Program "sdm_pROM_xRAM.elf" is executing. Choose Break from the Debug menu to stop it.
Result 2:

In this example (PWM- Edge Aligned mode):
ADC conversion time +
timer C channel 2 delay $\approx 4.4$ usec

Port C bit 0

PWM Syn Pulse

PWM Syn Pulse

PWM 0

$\approx 4.4$ usec

$\approx 1.9$ usec

$\approx 3.7$ usec

$10$ usec
Summary

Hands-On Training
Summary

- Understand the hardware and software support available for the 56800E Hybrid Controller product line.

- Demonstrated the ease of developing applications using CodeWarrior development tools with Processor Expert™ technology for PWM_ADC_SYN conversion.
Thank You!!!