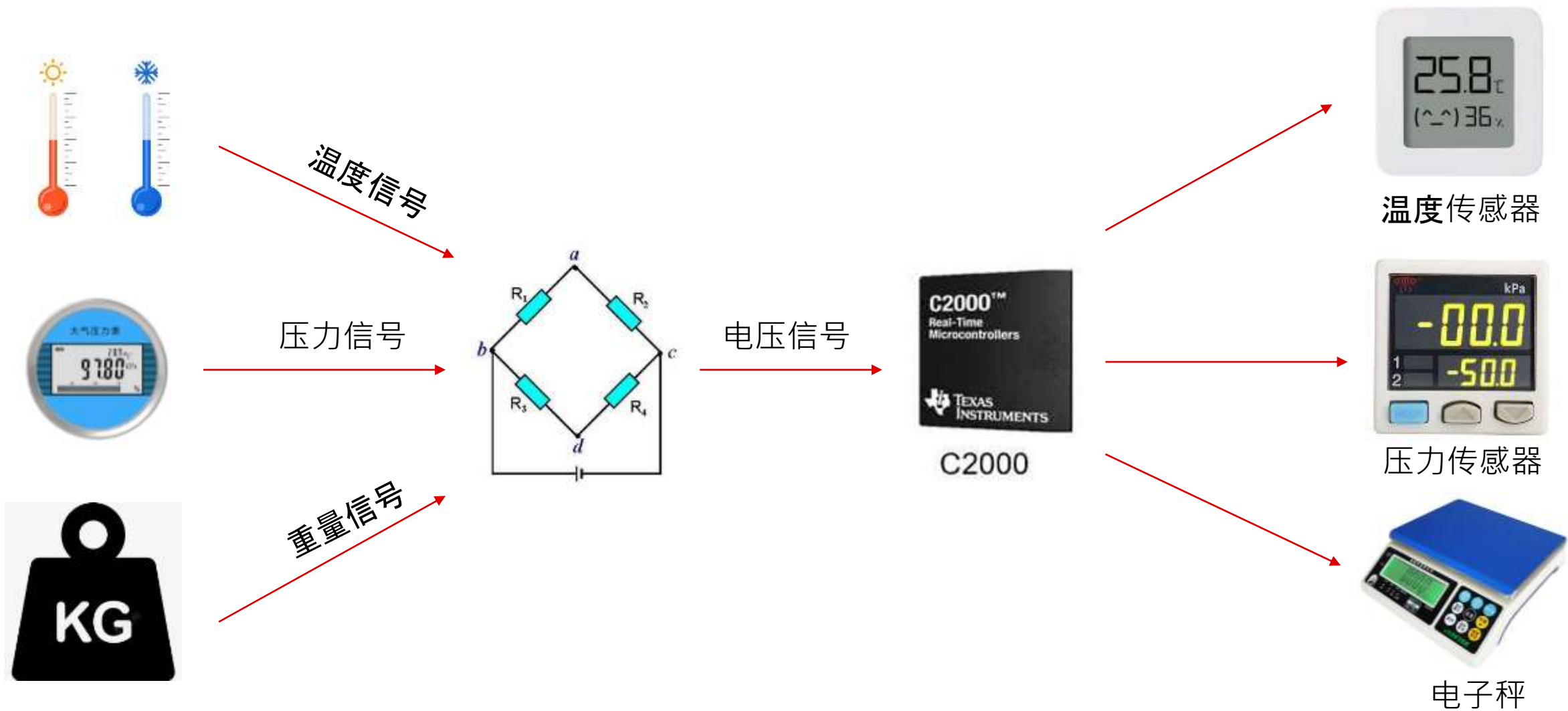


# F28P55x编程实例Labs-AD采样

- Code Composer Studio
- C2000Ware
- LaunchXL-F28P55x

# ADC模数转换器



# ADC模数转换器

## 参考电压分配

**ADC :**  
VREFHx/VSSA  
内部: (0-3.3)V/(0-2.5)V

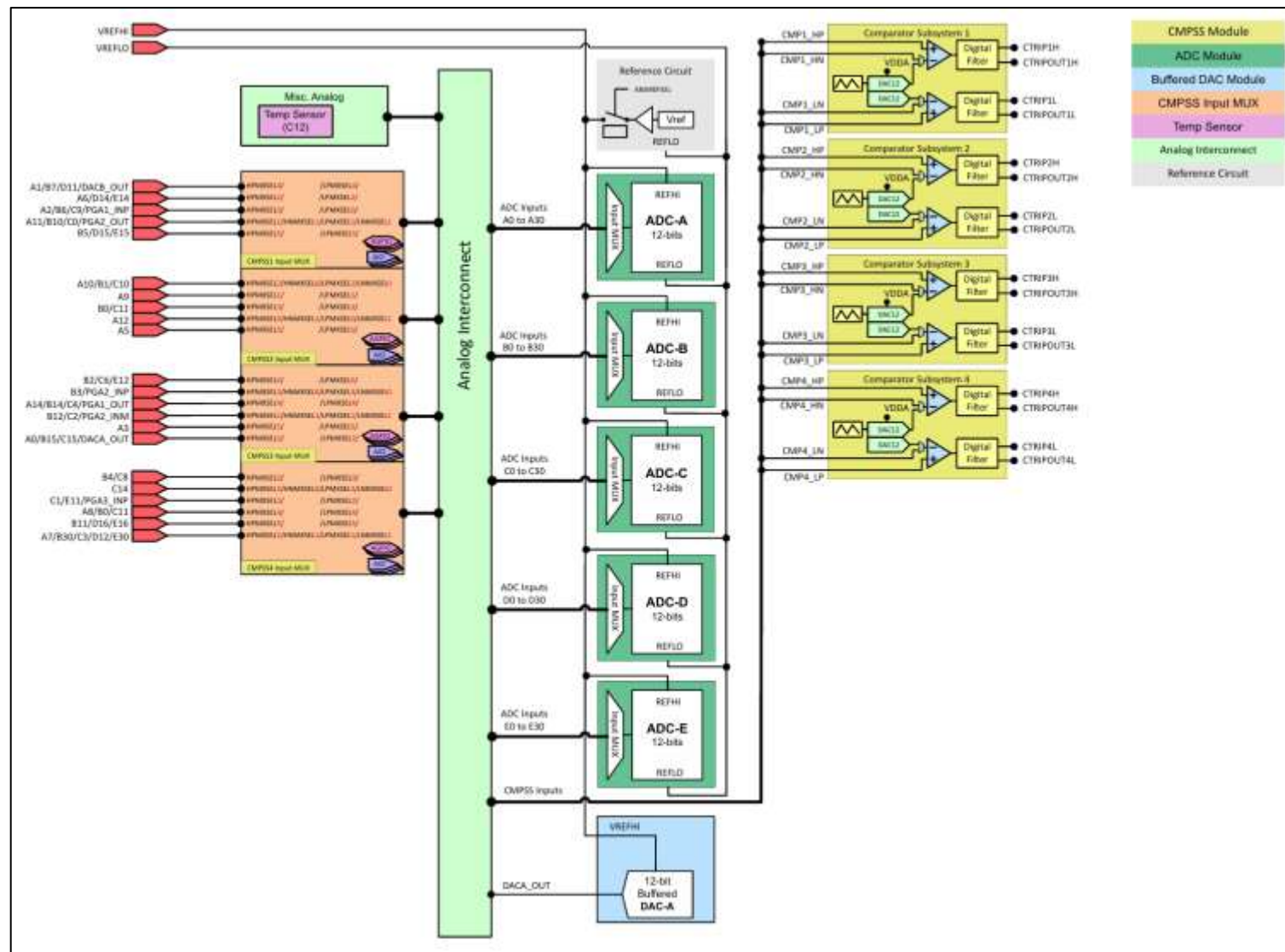
**DAC:**  
VREFHx/VSSA :

**CMPSS:**  
VDDA/VSSA

**VREFLO:**  
用于ADC的失调电压自我校准

单端采样模式 : ADCINx、GND

差分采样模式 : ADCINxP、ADCINxN



# ADC模数转换器

## 功能实现

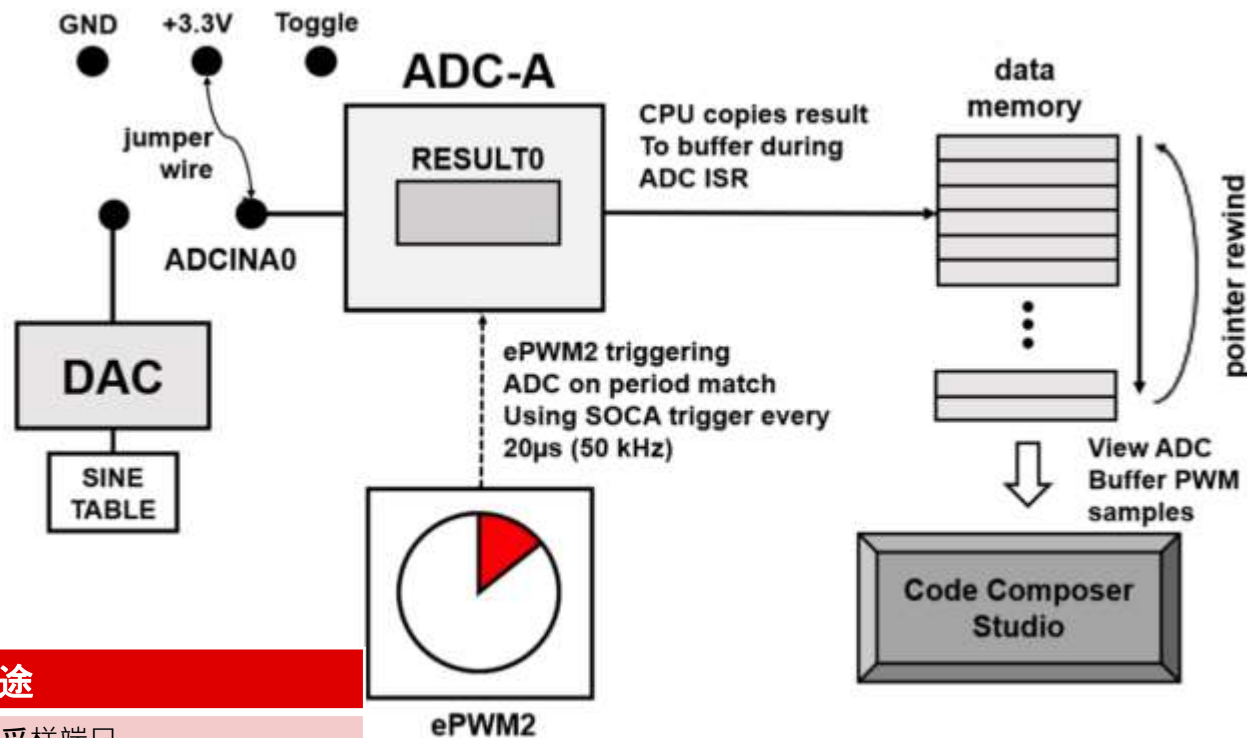
单片机的EPWM1产生方波，用ADCA0通道采集3.3V和EWPM1，以Debug的方式查看数据采集的变化情况；

GPIO16—高电平  
GPIO3 —电平翻转

## 实现步骤

- 复制空白工程
- Sysconfig配置GPIO
- Sysconfig配置EPWM
- Sysconfig配置ADCA0

GPIO	PIN脚	用途
ADCINA0	70	AD采样端口
GPIO16	51	输出高电平
GPIO3	37	输出PWM
LED5	--	运行指示
EPWM2	--	用于触发ADC的SOC



# Sysconfig配置GPIO

GPIO	PIN脚	用途
ADCINA0	70	AD采样端口
<b>GPIO16</b>	<b>51</b>	<b>输出高电平</b>
GPIO3	37	输出PWM
LED5	--	运行指示
EPWM2	--	用于触发ADC的SOC

GPIO (3 of 66 Added) ⓘ

- ✓ myGPIOHigh
- ✓ myGIOToggle
- ✓ myBoardLED0\_GPIO

 ADD  REMOVE ALL



Name	myGPIOHigh
Use Hardware	None
Analog Mode	Pin is in digital mode
GPIO Direction	Pin is a GPIO output
Pin Type	Push-pull output/pull-up enabled on input
Qualification Mode	Synchronization to SYSCLK
<b>External Interrupts</b> Connect to an XINT for interrupts	
Core Select	CPU1 selected as controller core
Write Initial Value	<input checked="" type="checkbox"/>
Initial Value	1: GPIO state is HIGH
<b>PinMux</b> Peripheral and Pin Configuration	
GPIO	C24, D2, E2, GPIO16/54 (Header)

# Sysconfig配置GPIO

GPIO	PIN脚	用途
ADCINA0	70	AD采样端口
GPIO16	51	输出高电平
GPIO3	37	输出PWM
LED5	--	运行指示
EPWM2	--	用于触发ADC的SOC

 ADD  REMOVE ALL

GPIO (3 of 66 Added) 


- myGPIOHigh
- myGPIONToggle
- myBoardLED0\_GPIO






Name	myGPIONToggle
Use Hardware	None
Analog Mode	Pin is in digital mode
GPIO Direction	Pin is a GPIO output
Pin Type	Push-pull output/pull-up enabled on input
Qualification Mode	Synchronization to SYSCLK

**External Interrupts** Connect to an XINT for interrupts 

Core Select	CPU1 selected as controller core
Write Initial Value	<input checked="" type="checkbox"/>
Initial Value	1: GPIO state is HIGH

**PinMux** Peripheral and Pin Configuration 

GPIO  

 Connected to hardware(Un-suppress).

# Sysconfig配置GPIO

GPIO	PIN脚	用途
ADCINA0	70	AD采样端口
GPIO16	51	输出高电平
GPIO3	37	输出PWM
LED5	--	运行指示
EPWM2	--	用于触发ADC的SOC

GPIO (3 of 66 Added) ⓘ

✓ myGPIOHigh

✓ myGPIONToggle

✓ myBoardLED0\_GPIO

+ ADD

REMOVE ALL



Name	myBoardLED0_GPIO
Use Hardware	LED5
Analog Mode	Pin is in digital mode
GPIO Direction	Pin is a GPIO output
Pin Type	Push-pull output/floating input
Qualification Mode	Synchronization to SYSCLK
<b>External Interrupts</b> Connect to an XINT for interrupts	
Core Select	CPU1 selected as controller core
Write Initial Value	<input type="checkbox"/>
<b>PinMux</b> Peripheral and Pin Configuration	
GPIO	Any(A18, B18, C18, GPIO21/49 (LED5))

# Sysconfig配置EPWM

GPIO	PIN脚	用途
ADCINA0	70	AD采样端口
GPIO16	51	输出高电平
GPIO3	37	输出PWM
LED5	--	运行指示
EPWM2	--	用于触发ADC的SOC

EPWM (1 of 12 Added) ⓘ

+ ADD

REMOVE ALL

✓ myEPWM1



Name myEPWM1

Use Hardware None

Load EPWM Settings From Device Memory Export

Copy Settings

Template Code Generation

EPWM Global Load



# Sysconfig配置EPWM

GPIO	PIN脚	用途
ADCINA0	70	AD采样端口
GPIO16	51	输出高电平
GPIO3	37	输出PWM
LED5	--	运行指示
EPWM2	--	用于触发ADC的SOC

**EPWM Time Base**

Emulation Mode

Time Base Clock Divider

High Speed Clock Divider

Time Base Period Load Mode

Time Base Period Load Event

Time Base Period

Time Base Period Link

Enable Time Base Period Global Load

Initial Counter Value

Counter Mode

Enable Phase Shift Load

Force a Sync Pulse

Sync In Pulse Source

Sync Out Pulse

One-Shot Sync Out Trigger

EPWMxSYNCPER Source Select

Divide clock by 1

**i** For perfectly synchronized TBCLKs across multiple EPWM modules, the prescaler bits in the TBCTL register of each EPWM module must be set identically

Divide clock by 1

PWM Period register access is through shadow register

Shadow to active load occurs when time base counter reaches 0

1999

Disable Linking

0

Up - count mode

Sync-in source is EPWM1 sync-out signal

None

Trigger is OSHT sync

Counter equals Period

# Sysconfig配置EPWM

GPIO	PIN脚	用途
ADCINA0	70	AD采样端口
GPIO16	51	输出高电平
GPIO3	37	输出PWM
LED5	--	运行指示
EPWM2	--	用于触发ADC的SOC

## EPWM Event-Trigger

Enable EPWM Interrupt	<input checked="" type="checkbox"/>
Register Interrupt Handler	<input type="checkbox"/>
Interrupt Event Sources	Time-base counter is disabled
Interrupt Event Count	Disabled
Interrupt Event Count Initial Value Load Enable	<input type="checkbox"/>
Force Interrupt Event Count Initial Value	<input type="checkbox"/>

## ADC SOC Trigger

SOCA Trigger Enable	<input checked="" type="checkbox"/>
SOCA Trigger Source	Time-base counter equal to period
SOCA Trigger Event Count	1 Event Generates Interrupt
SOCA Trigger Event Count Initial Value Load Enable	<input type="checkbox"/>
SOCB Trigger Enable	<input type="checkbox"/>

# Sysconfig配置ADCA0

ADC (1 of 5 Added)

myADCA

Name: myADCA  
ADC Instance: ADCA  
ADC Clock Prescaler: ADCCLK = (input clk)  
Enable alternate timings (tDMA):   
Use External MUX:   
High Priority Mode SOCs: Round robin mode is used for all

**SOC Configurations** Start of Conversion Configurations

Enable SOCs SOC/EOC number 0

**SOC0** Start of Conversion 0

SOC0 Name: SOC0  
SOC0 Independent Name Mode:   
SOC0 Channel: single-ended, ADCIN0  
SOC0 Module Channel Name: A0  
SOC0 Device Pin Name: 23: AD/ B15/ C15/ DACA\_OUT  
SOC0 External Channel Selected via MUX: ADC\_CH\_ADCINX\_0

**SOC Triggers**

Trigger Mode: Single Trigger  
SOC0 Trigger: ePWM1, ADCSOCA  
SOC0 Interrupt Trigger: No ADCINT will trigger the SOC

**Sample Time Calculator**

SOC0 Sample Window [SYSCLK counts]: 20  
SOC0 Sample Time [ns]: 133.33333333333334

GPIO	PIN脚	用途
ADCINA0	70	AD采样端口
GPIO16	51	输出高电平
GPIO3	37	输出PWM
LED5	--	运行指示
EPWM2	--	用于触发ADC的SOC

# Sysconfig配置ADCA0

GPIO	PIN脚	用途
ADCINA0	70	AD采样端口
GPIO16	51	输出高电平
GPIO3	37	输出PWM
LED5	--	运行指示
EPWM2	--	用于触发ADC的SOC

**ADC INT Configurations** Interrupt Configurations

ADC Interrupt Pulse Mode Occurs at the end of the conversion

Enable ADC Interrupts ADCINT1 Interrupt

**INT1** ADC Interrupt 1

Enable ADC Interrupt 1

Interrupt 1 SOC Source SOC/EOC0

Continuous Interrupt Mode

**PPB Configurations** Post Processing Blocks Configurations

**Burst Mode** ADC Burst Mode

**Register PIE Interrupt Handlers**

**Analog PinMux** myANALOGPinMux0

Name myANALOGPinMux0

Use Case CUSTOM

Pins Used A0, B15, C15, DACA\_OUT

**PinMux** Peripheral and Pin Configuration

ANALOG Peripheral Any(ANALOG)

A0, B15, C15, DACA\_OUT Any(A0, B15, C15, DACA\_OUT/23 (Header))

```

#include "board.h"

#define ADC_BUF_LEN      50
uint16_t DEBUG_TOGGLE = 1; // Used for real-time mode
uint16_t AdcBuf[ADC_BUF_LEN]; // ADC buffer allocation

//
// Function Declarations
//
__interrupt void INT_myADCA_1_ISR(void);

void main(void)
{
    // CPU Initialization
    Device_init();
    Interrupt_initModule();
    Interrupt_initVectorTable();

    // Configure the GPIOs/ADC/PWM through SysConfig generated files
    Board_init();

    // Enable global interrupts and real-time debug
    EINT;
    ERTM;

    // Main Loop
    while(1){}
}

```

```

interrupt void INT_myADCA_1_ISR(void)
{
    static uint16_t *AdcBufPtr = AdcBuf;
    uint16_t LED_count = 0;

    // Read the ADC Result
    *AdcBufPtr++ = ADC_readResult(myADCA_RESULT_BASE, myADCA_SOC0);

    // Brute Force the circular buffer
    if (AdcBufPtr == (AdcBuf + ADC_BUF_LEN))
    {
        AdcBufPtr = AdcBuf;
    }

    // Toggle the pin
    if(DEBUG_TOGGLE == 1)
    {
        GPIO_togglePin(myGIPIOToggle);
    }

    if(LED_count++ > 25000) // Toggle slowly to see the LED blink
    {
        GPIO_togglePin(myBoardLED0_GPIO); // Toggle the pin
        LED_count = 0; // Reset the counter
    }

    Interrupt_clearACKGroup(INT_myADCA_1_INTERRUPT_ACK_GROUP);
    ADC_clearInterruptStatus(myADCA_BASE, ADC_INT_NUMBER1);
} // End of ADC ISR

```