Texas Instruments

Application Report

SPRACP6–December 2019

How to Maximize GPIO Usage in C2000™ Devices

Aki Li, Frank Ankapong

ABSTRACT

This application report provides several suggestions on maximizing the GPIO resources on the device to limit the need for IO expanders. Most of the content in this report is catered towards the TMS320F28004x series but can be adapted for other C2000 real-time controllers.

Contents

List of Figures

List of Tables

Trademarks

C2000, Code Composer Studio are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

1 Introduction

With the integration trend of microcontrollers, new C2000 devices now provide more analog peripherals, which reduce the system level cost and design effort. The F28004x device incorporates on-chip analog-to-digital converters (ADCs), buffered digital-to-analog converters (DACs), programmable gain amplifiers (PGAs), and comparator subsystems (CMPSS). However, the addition of all this extra analog along with their required pins has led to the reduction in the number of GPIOs. This document explores several options on how to maximize usage of GPIOs.

2 Zero Pin Boot Mode Option

On every reset, the device executes a boot sequence in ROM depending on the boot mode configuration. By default, the boot ROM uses two boot control GPIO pins to determine the boot mode configuration. Table 1 shows the default boot mode options of the F28004x device.

Table 1. F28004x Device Default Boot Modes

Boot Mode	GPIO24 (default boot mode select pin 1)	GPIO32 (default boot mode select pin 0)
Parallel IO	0	0
SCI/Wait boot	0	1
CAN	1	0
Flash	1	1

There are two ways the boot control GPIO pins can be re-purposed for use in an application:

- Utilize weak pull-ups on these GPIOs so they can be overdriven by the application
- Configure an alternate boot mode that does not require the boot control GPIO pins

2.1 *Weak Pull-Up Method*

By using weak pull-ups on the boot control GPIO pins, the device will always boot to Flash boot mode. However, after boot completion, the boot control GPIO pins can be configured for application use. When utilizing this approach, the internal pull-ups should not be enabled on these GPIOs. It is recommended to use pull-up resistors in the range of 16.5 kΩ to 27.5 kΩ.

NOTE: The boot control GPIO pins should not be driven by anything else while the device is booting as that can influence the mode the device boots into.

2.2 *Zero Pin Boot Mode Method*

Instead of dedicating a set of GPIOs to determine boot mode, the boot mode of the device can be configured as a zero pin option.

With the introduction of the F28004x and newer devices, the number of boot mode select pins can now be customized to support as many as three and as few as zero pins. Using the zero pin option restricts the device to a single boot mode but it frees up the factory default pins to be used for GPIO purposes. Contrary to previous devices which only allowed reconfiguration of the fourth boot mode entry, for the F28004x device, the whole boot mode selection table is customizable. The steps below outline how the device can be configured for zero pin boot to flash:

1. Customize boot pin configuration:
   a. Set BOOTPIN_CONFIG.BMSP0 to 0xFF
   b. Set BOOTPIN_CONFIG.BMSP1 to 0xFF
   c. Set BOOTPIN_CONFIG.BMSP2 to 0xFF
   d. Set BOOTPIN_CONFIG.KEY to 0x5A for boot ROM to treat these register bits as valid

2. Flash boot mode configuration.

   This is achieved by configuring BOOT_DEF_LOW (0xFFFFFF03) and BOOT_DEF_HIGH (0xFFFFFFFF) user OTP locations as shown in Table 2.

Table 2. Custom Boot Table

Boot Mode Number	BOOTDEF Name	Value
0	BOOT_DEF0	03(Flash boot)
1	BOOT_DEF1	0xFF
2	BOOT_DEF2	0xFF
3	BOOT_DEF3	0xFF
4	BOOT_DEF4	0xFF
5	BOOT_DEF5	0xFF
6	BOOT_DEF6	0xFF
7	BOOT_DEF7	0xFF

3. Writing the values to user OTP.

   All the above custom boot configurations take effect by programming the related locations in user-configurable DCSM OTP. The steps below outline one way that can be achieved:

   a. Pick any of the C2000Ware examples project and add the following code snippet above the main function (it can be anywhere in the file just outside the functions):

```
#pragma RETAIN(otp_z1_data_1)
#pragma DATA_SECTION(otp_z1_data_1,"dcsm_zsel_z1_1");
const long otp_z1_data_1 = 0x5AFFFFFF;
#pragma RETAIN(otp_z1_data_2)
#pragma DATA_SECTION(otp_z1_data_2,"dcsm_zsel_z1_2");
const long otp_z1_data_2 = 0xFFFFFF03;
```

   b. In the linker command file (e.g 28004x_generic_ram_lnk.cmd), add the following lines:

```
MEMORY
{
PAGE 0:
DCSM_ZSEL_Z1_P0: origin = 0x07800C, length = 0x000002
DCSM_ZSEL_Z1_P1: origin = 0x07801C, length = 0x000002
}
SECTIONS
{
dcsm_zsel_z1_1 : > DCSM_ZSEL_Z1_P0, PAGE = 0
dcsm_zsel_z1_2 : > DCSM_ZSEL_Z1_P1, PAGE = 0
}
```

   c. Re-compile the example and load to the target via JTAG using Code Composer Studio™ (CCS). The loader and flash API plugin in CCS will take care of writing these values to OTP location.

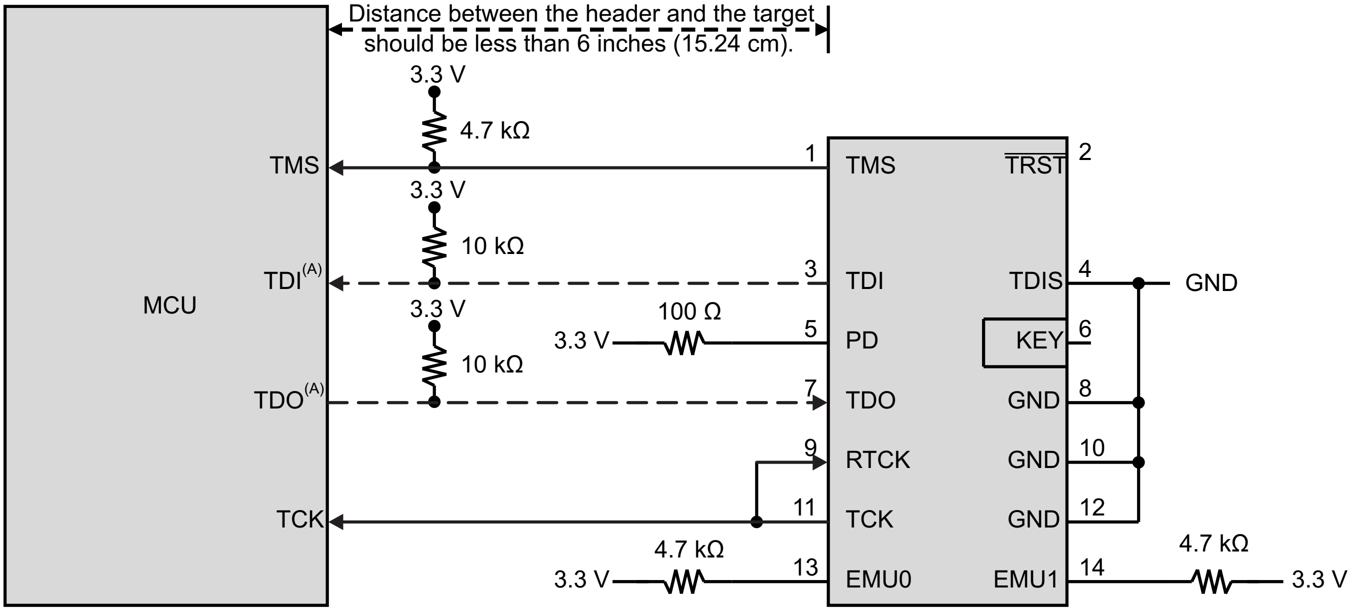
For detailed information on the above configurations, see the *Device Boot Modes* chapter in the TMS320F28004x Piccolo Microcontrollers Technical Reference Manual.

NOTE: OTP locations cannot be re-written, so it is strongly suggested that the Emulation Boot Mode be used to validate the custom modification first. EMU-BOOTPIN-CONFIG and EMU-BOOTDEF can be programmed to experiment with different boot modes without writing to OTP.

3 cJTAG Option

Contrary to previous devices, the F28004x device supports cJTAG connection as shown in Figure 1. JTAG port has four dedicated pins: TMS, TDI, TDO, and TCK, while cJTAG port is a compact JTAG interface requiring only two pins (TMS and TCK). Thus, the traditional GPIO35 (TDI) and GPIO37 (TDO) pins can be muxed as other peripheral functions.

Compared to JTAG, cJTAG provides lower performance because it uses only two pins: TMS and TCK. Basically, TDI, TDO and TMS functionality is implemented on single pin, TMS. The maximum supported clock frequency on TCK is 15 MHz for JTAG, and for cJTAG, it decreases to 10 MHz. Note that in order to use cJTAG, a supported debug probe is required, such as XDS100V3, XDS110, XDS200, and so forth.



A TDI and TDO connections are not required for cJTAG option and these pins can be used as GPIOs instead.

Figure 1. Connecting to the 14-Pin JTAG Header

4 Zero Pin Internal Oscillator Option

If using the internal oscillator as the clock source, the X2 pin can be used as a GPIO. Note that if using X2 as a GPIO, X1 has to be pulled down with a 1kΩ resistor for correct functionality.

Unlike legacy devices, the F28004x device does not require oscillator compensation since the trimming during factory testing takes care of this. Typically, the internal oscillator frequency is less than ±1% from target across voltage/temperature. However, when the lifetime of the device is taken into consideration, the frequency stability can not be guaranteed, so the maximum frequency shift will be ±3%, as shown in the Table 3.

Table 3. Internal Oscillators Characteristics

Parameter		Test Conditions	MIN	TYP	MAX	UNIT
f_{INTOSC}	Frequency, INTOSC1 and INTOSC2		9.7	10	10.3	MHz
$f_{INTOSC-STABILITY}$	Frequency stability at room temperature	30°C, Nominal VDD		±0.1%		
	Frequency stability over VDD	30°C		±0.2%		
	Frequency stability		–3%		3%	
$t_{INTOSC-ST}$	Start-up and settling time				20	µs

Note that since timing requirements for CAN is critical, it is recommended to use an external oscillator if the CAN module is being used in the application. The required frequency tolerance for the CAN bit clock depends on the bit timing setup and network configuration, and can be as tight as 0.1%. Furthermore, the ADC has non-negligible sensitivity with the clock source. When using INTOSC as the system clock, the ADC will incur performance degradation, like lower SNR. Flash operation will also be affected, since an extra flash wait state is needed if INTOSC is used as the clock source for F28004x. For more details, see the *TMS320F28004x Piccolo™ Microcontrollers Data Manual*.

5 CMPSS to Save Extra GPIO for Fast Protection

Generally, it is common to use external comparators to achieve protections such as OVP and OCP. These external comparators then communicate their trip information to the device through GPIO pins which are then used to trip the ePWMs. The F28004x device has internal comparators and hence eliminates the need for external comparators and required GPIOs. The CMPSS outputs can be brought out on GPIOs for debug purposes but this is not required for normal operation.

By design, the CMPSS outputs are routed directly to the EPWM through the EPWM X-BAR for tripping purposes. There are 7 Comparator Subsystems (CMPSS) in the F28004x device, with two comparators each for a total of 14 comparators. As shown in Figure 2, the delay from CMPINx input change to ePWM output trip action is approximately within 55 ns, which is comparable to discrete comparators. Furthermore, since the CMPSS inputs are multiplexed with the ADC inputs, in-time protection happens as the ADC samples the input signal. For detailed information on the CMPSS, see the *CMPSS* chapter in the *TMS320F28004x Piccolo Microcontrollers Technical Reference Manual*. And, the software examples can be found in C2000Ware folder located at: C:\ti\c2000\C2000Ware_<version_number>\driverlib\f28004x\examples\cmpss.

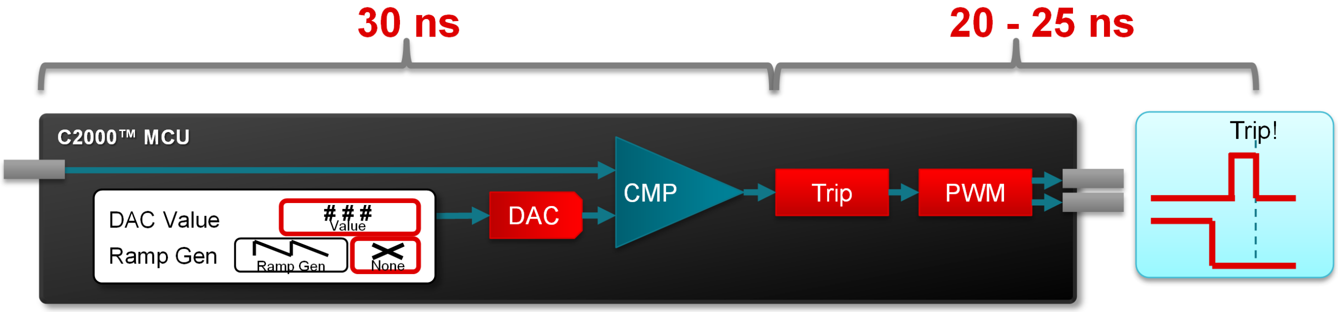


Figure 2. CMPSS Protection Timing

6 AIO Option

The analog signals in the F28004x device are multiplexed with digital input functionality. In the F28004x device, the Input X-BAR has access to every GPIO, including AIOs, so Input X-BAR can be used to route signals from a GPIO to many different IP blocks such as the ADC(s), eCAP(s), ePWM(s), and external interrupts, as shown in Figure 3.

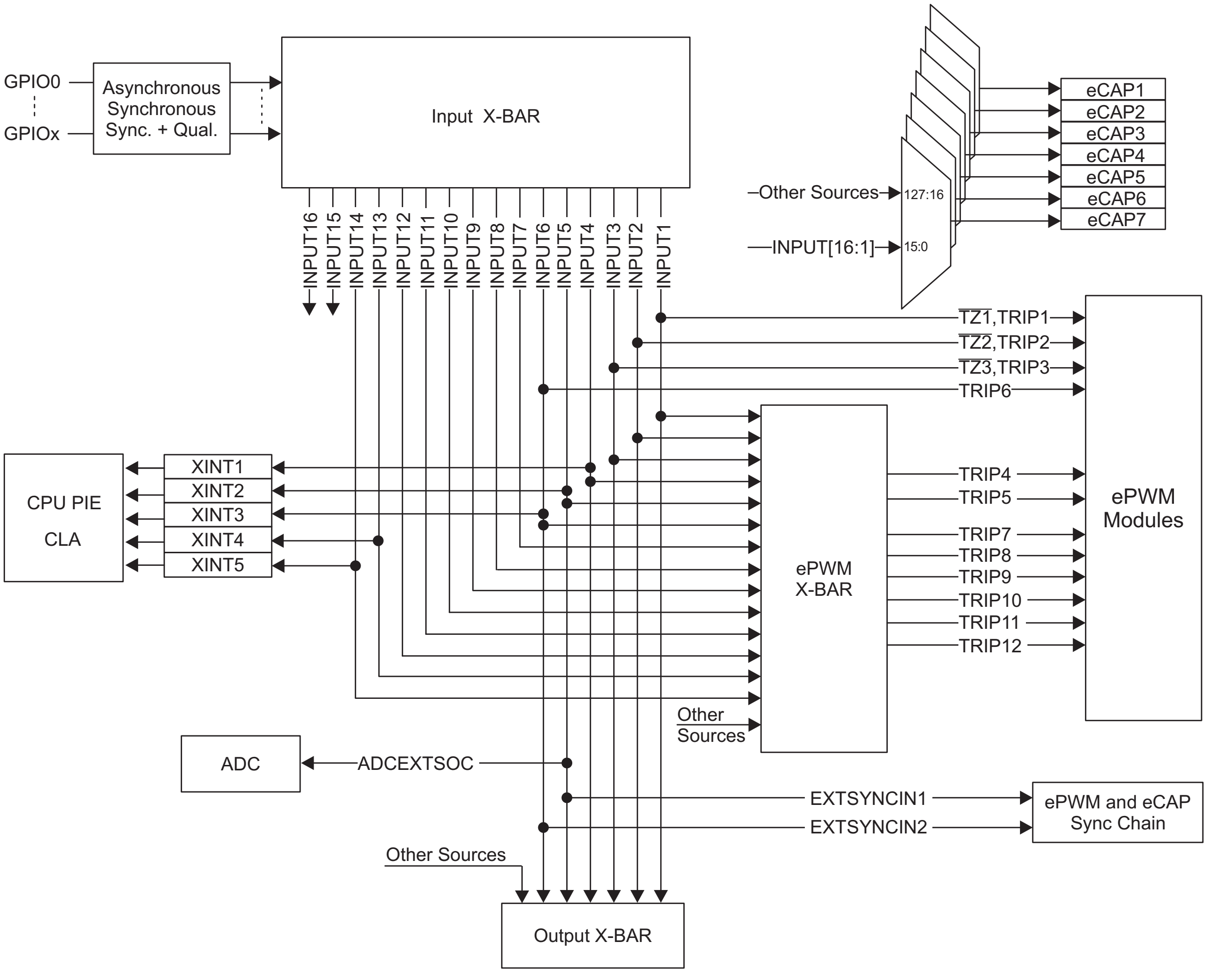


Figure 3. Input X-BAR

For example, the following code outlines using a digital AIO to receive Fault IO signal, then to trigger ePWM protection actions. As shown in Table 4, ADC B5 channel refers to AIO243. In this example, both Input X-BAR and ePWM X-BAR will be used.

Table 4. AIO Pins Mapping

Analog Group 4								CMP4				
B5	G4_ADCAB				B5			HPMXSEL = 3	HNMXSEL = 0	LPMXSEL = 3	LNMXSEL = 0	AIO243
B4/C8/PGA4_OF	PGA4_OF	39	24	22	B4	C8	PGA4_OF	HPMXSEL = 0		LPMXSEL = 0		AIO227
C3	G4_ADCC					C3		HPMXSEL = 1	HNMXSEL = 1	LPMXSEL = 1	LNMXSEL = 1	AIO245
PGA4_IN	PGA4_IN						PGA4_IN	HPMXSEL = 2		LPMXSEL = 2		
PGA4_GND	PGA4_GND	32	20	18			PGA4_GND					
-	PGA4_OUT				B11	C9	PGA4_OUT	HPMXSEL = 4		LPMXSEL = 4		

```
// enable digital mode for AIO243(B5)
    GPIO_setAnalogMode(243, GPIO_ANALOG_DISABLED);
    GPIO_setPinConfig(GPIO_243_GPIO243);
    GPIO_setDirectionMode(243, GPIO_DIR_MODE_IN);
    GPIO_setPadConfig(234U, GPIO_PIN_TYPE_STD);

    //Configure AIO243 as INPUT X-BAR 1
   XBAR_setInputPin(XBAR_INPUT1, 243);

   // Configure INPUT X-BAR 1 as EPWM X-BAR TRIP4
    XBAR_setEPWMMuxConfig(XBAR_TRIP4, XBAR_EPWM_MUX01_INPUTXBAR1);
    XBAR_enableEPWMMux(XBAR_TRIP4, XBAR_MUX01);

//To see the remainder of the code, refer to the example in
C:\ti\c2000\C2000Ware__<version_number>\\driverlib\f28004x\examples\cmpss\cmpss_ex1_asynch
```

Since the AIOs share the same reference as other analog peripherals, if digital signals with sharp edges (high dv/dt) are connected to the AIOs, cross-talk might occur with adjacent analog signals. Therefore, you should limit the edge rate of signals connected to AIOs, so it is suggested to use series resistors at the driver input to an AIO to slow down the edge rate and reduce the coupling. It is also recommended the AIOs be used for signals with minimal toggling, like a FAULT indicator. The AIO input circuit ground pin should be connected to the analog ground VSSA of F28004x.

These analog I/O (AIO) pins do not have digital output capability. For 100-pin PZ package, there are to up to 21 AIOs that share the same pins with ADC channels. They are assigned to a single port: Port H of GPIO224-GPIO247.

7 Using the GPDACs as General Purpose Outputs

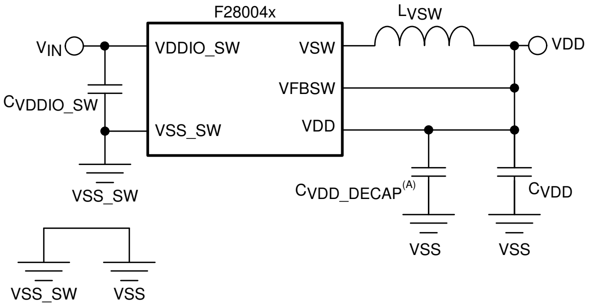
The F28004x device has two general-purpose 12-bit DACs (GPDAC). The output of these GPDACs are multiplexed with two ADC channels. If the application requires more GPIOs but is not utilizing the GPDACs and has some ADC channels left over, the GPDACs can be treated as general purpose outputs by only writing 0 and 4095 to the DACVALS register. There are limitations to using the GPDACs as general-purpose outputs. The GPDAC has a maximum switch rate of 500KHz. However, this should be sufficient to toggle a LED.

8 Optional 1.2-V Internal DCDC

The F28004x device has three options for supplying the 1.2 V VDD voltage:

- An external supply (not available for 56-pin RSH package configurations)
- Internal 1.2-V LDO Voltage Regulator (VREG)
- Internal 1.2-V Switching Regulator (DC-DC)

The internal DC-DC regulator offers increased efficiency over the LDO for converting 3.3 V to 1.2 V. For detailed information on the DC-DC, see the *TMS320F28004x Piccolo™ Microcontrollers Data Manual*. Figure 4 shows the schematic implementation. If the internal DC-DC regulator is not used, VFBSW and VSW pin can be configured as GPIO22 and GPIO23, respectively. Note that VSW pin has an internal capacitance of approximately 100 pF, so it is suggested to use this pin only for applications that do not require a fast switching response.



Copyright © 2017, Texas Instruments Incorporated

Figure 4. DC-DC Circuit Schematic

9 References

Texas Instruments: *TMS320F28004x Piccolo Microcontrollers Technical Reference Manual*

Texas Instruments: *TMS320F28004x Piccolo™ Microcontrollers Data Manual*

Texas Instruments: *The TMS320F28004x Microcontroller: A Comparison to the TMS320F2806x and TMS320F2803x Microcontrollers*

Texas Instruments: *C2000™ Software Controlled Firmware Update Process*

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265