C2000 Tips

Useful Website for C2000

- **General**

  www.ti.com/c2000
  www.ti.com/piccolo
  www.ti.com/delfino
  www.ti.com/controlsuite
  www.ti.com/c2000tools
  www.ti.com/c2000getstarted
  www.ti.com/c2000training
  www.ti.com/c2000community
  processors.wiki.ti.com/index.php/Category:C2000

- **Applications**

  www.ti.com/motorcontrol
  www.ti.com/digitalpower
  www.ti.com/hev
  www.ti.com/solar
  www.ti.com/metering
  www.ti.com/led
  www.ti.com/c2000dmc
  www.ti.com/dpslib

DPS Software Library & Header Files

Today's DPSLib components: * Will all roll into controlSuites

- **ePWM Demonstration** **SPRC228**
- **HRPWM Demonstration** **SPRC227**
- **DC/DC Buck Converter** **SPRC229**
- **DC/AC Single Phase Inverter** **SPRC303**
- **Two-phase PFC** **SPRC307**
- **DC/DC Phase-shifted Full-Bridge** **SPRC311**

Visit www.ti.com/dpslib for above library s/w and link to the h/w platforms

C2000 header files Simplify peripheral init. and other functions, takes care of register defs

- **Header file package consists of:**
  - **\DSP280x_headers\include → .h files**
  - **\DSP280x_common\src → .c source files**
  - **\DSP280x_headers\cmd → linker command files**
  - **\DSP280x_headers\gel → .gel files for CCS**
  - **\DSP280x_examples → example programs**
  - **\doc → documentation**

Visit www.ti.com/c2000getstarted

Free On-line Training

- **Using Multi-Phase DC/DC Power Supply Control**
  http://training.ti.com/courses/CourseDescription.asp?iCSID=53935
- **Enabling High-Freq Power Conversion Applications**
  http://training.ti.com/courses/CourseDescription.asp?iCSID=54032
- **Implementing High-BW, Low-Cycle Count Controllers**
  http://training.ti.com/courses/CourseDescription.asp?iCSID=54259

CCS3.3 Version, CSP, CGT and Flash Plug-In

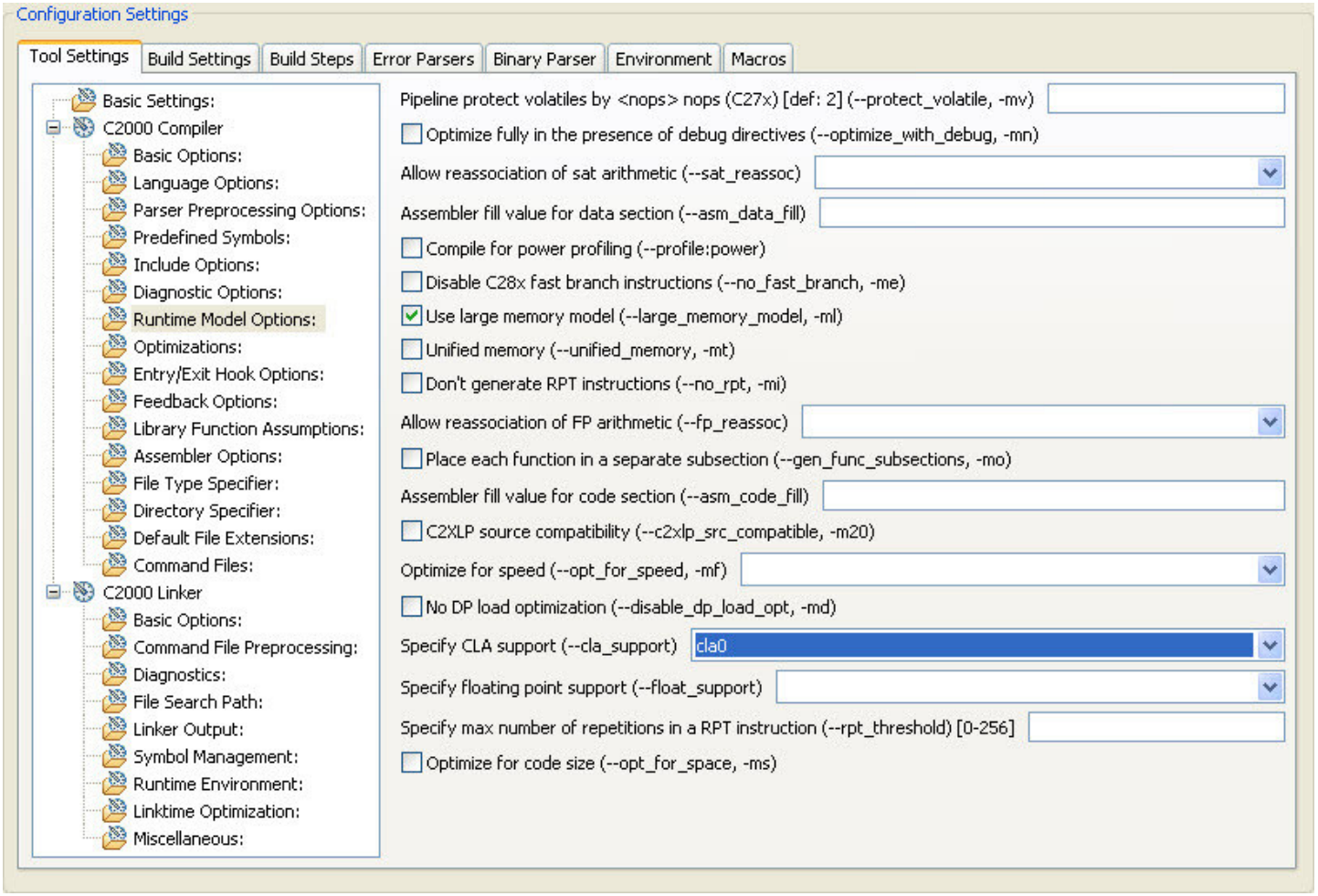
- Service Release: https://www-a.ti.com/downloads/sds_support/CCSv3.3ServiceReleases.htm
- Code Generation Tools: https://www-a.ti.com/downloads/sds_support/CodeGenerationTools.htm
- Delfino CSP:
  http://software-dl.ti.com/dsps/dsps_registered_sw/sdo_ccstudio/CCSv3/CSPS/Delfino_CSP_v3.3.2801.EXE
- F2803x CSP V3.3.2901/2903:
  http://software-dl.ti.com/dsps/dsps_registered_sw/sdo_ccstudio/CCSv3/CSPS/F2803x_CSP_v3.3.2901.exe
  http://software-dl.ti.com/dsps/dsps_registered_sw/sdo_ccstudio/CCSv3/CSPS/F2803x_CSP_v3.3.2903.exe
- CCS V3.3.83.20:
  http://software-dl.ti.com/dsps/dsps_public_sw/sdo_ccstudio/CCSv3/CCS_3_3/exports/CCS_3.3.83.20_Platinum.zip

CCS Version	Service Releases	Chip Support	Chip Support Package	Code Generation Tools	Flash Plug-In (API)
3.3.38.2	--	F281x F280x	F2823x CSP	4.1.3	--
3.3.81	SR11	F2823x	Delfino CSP	5.0.0	--
3.3.82.13	SR12	F2833x	F2802x CSP F2803x CSP V3.3.2901	5.1.1	F2803x Flash Plug-In
3.3.82.13	SR12	F280200/20/21 F28030/31	F2803x CSP V3.3.2903	5.1.1	F28026 API V2.00
3.3.83.16/19/20	--	F2802x F2803x (CLA)	--	5.2.0	--

CCS Versions & CGT Wiki

- C2000 Wiki: http://processors.wiki.ti.com/index.php/Category:C2000
- Development Tools for C2000: http://processors.wiki.ti.com/index.php/Category:Development_Tools_for_C2000
- Concerto College: http://www.ti.com/mcu/docs/mcuorphan.tsp?contentId=129766
- Piccolo F2806x Multi-day Workshop: http://processors.wiki.ti.com/index.php/C2000_Piccolo_Multi-Day_Workshop
- Download CCS: http://processors.wiki.ti.com/index.php/Download_CCS#Code_Composer_Studio_Version_3_Downloads
- CCS Wiki: http://processors.wiki.ti.com/index.php/Category:CCS
- CCS v3.3 Wiki: http://processors.wiki.ti.com/index.php/Category:Code_Composer_Studio_v3
- CCS v4 Wiki: http://processors.wiki.ti.com/index.php?title=Category:Code_Composer_Studio_v4
- CCS v5 Wiki: http://processors.wiki.ti.com/index.php/Category:Code_Composer_Studio_v5
- Compiler Wiki: http://processors.wiki.ti.com/index.php?title=Category:Compiler
- Code Generation Tools: https://www-a.ti.com/downloads/sds_support/TICodegenerationTools/download.htm

Build Options & CGT Documents



SPRU430

SPRU513

SPRU514

H/W Design Suggestion

1. DS Chapter 2 & 6

2.2 Signal Descriptions

Table 2-2 describes the signals. With the exception of the JTAG pins, the GPIO function is the default at reset, unless otherwise mentioned. The peripheral signals that are listed under them are alternate

Table 2-2. Terminal Functions (1)

TERMINAL				I/O/Z	DESCRIPTION
NAME	PN PIN #	PAG PIN #	RSH PIN #		
JTAG					
$\overline{\text{TRST}}$	10	8	6	I	JTAG test reset with internal pulldown. $\overline{\text{TRST}}$, when driven high, gives the scan system control of the operations of the device. If this signal is not connected or driven low, the device operates in its functional mode, and the test reset signals are ignored. **NOTE:** $\overline{\text{TRST}}$ is an active high test pin and must be maintained low at all times during normal device operation. An external pull-down resistor is required on this pin. The value of this resistor should be based on drive strength of the debugger pods applicable to the design. A 2.2-kΩ resistor generally offers adequate protection. Since this is application-specific, it is recommended that each target board be validated for proper operation of the debugger and the application. (↓)
TCK	See GPIO38			I	See GPIO38. JTAG test clock with internal pullup. (↑)

2. Hardware Design Guidelines for TMS320F28xx and TMS320F28xxx DSCs (Rev. A): spraas1b

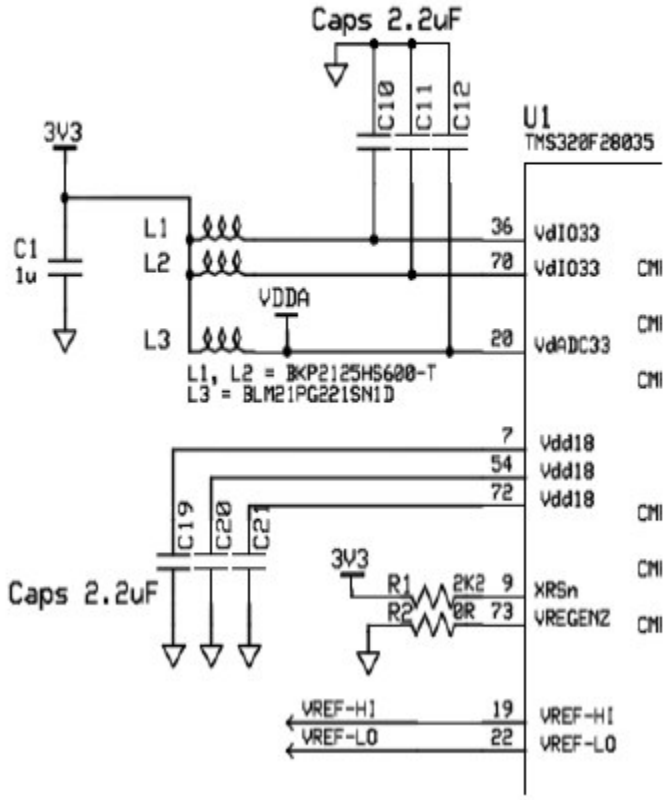
3. ControlCARD schematic

Vdd / Vss in Piccolo

Table 2-2. Terminal Functions [1] (continued)

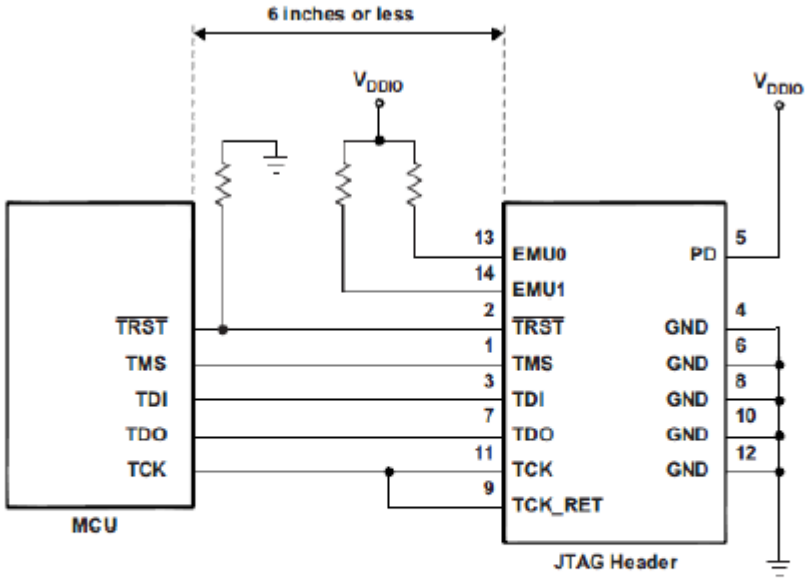
TERMINAL			I/O/Z	DESCRIPTION
NAME	PN PIN #	PAG PIN #		
CPU AND I/O POWER				
V_{DDA}	20	16		Analog Power Pin. Tie with a 2.2-µF capacitor (typical) close to the pin.
V_{SSA}	21	17		Analog Ground Pin. NOTE: VREFLO is always connected to V_{SSA} on the 64-pin PAG device.
V_{DD}	7	5		CPU and Logic Digital Power Pins – no supply source needed when using internal VREG. Tie with 1.2 µF (minimum) ceramic capacitor (10% tolerance) to ground when using internal VREG. Higher value capacitors may be used, but could impact supply-rail ramp-up time.
V_{DD}	54	43		
V_{DD}	72	59		
V_{DDIO}	36	29		Digital I/O and Flash Power Pin – Single Supply source when VREG is enabled
V_{DDIO}	70	57		
V_{SS}	8	6		Digital Ground Pins
V_{SS}	35	28		
V_{SS}	53	42		
V_{SS}	71	58		

Release 0.2



Piccolo JTAG Connection

1. EMU0/1 pull-up, TRST pull-down
2. JTAG multiplexing with GPIO
3. 300nf~1pf capacitor for EMC/EMI



See Figure 4-16 for JTAG/GPIO multiplexing.

Figure 6-4. Emulator Connection Without Signal Buffering for the MCU

NOTE

The 2803x devices do not have EMU0/EMU1 pins. For designs that have a JTAG Header on-board, the EMU0/EMU1 pins on the header must be tied to V_{DDIO} through a 4.7-kΩ (typical) resistor.

CCS fails to connect with target

- Target device?
- Power on?
- CCS support?
- JTAG hardware design & continuity?
- Oscillator works?
- Reset signal normal?
- Emulator type & vendor?
- Swap test?

CSM / Device Locked

1. New device with no password (password is 0xFFFFFFFF).
2. Device will be locked permanently if password is programmed to 0x00000000.
3. User must remember its password.
4. Don’t power down or cancel the procedure when flash programming is going on.
5. Don’t solder the chip to board with high-temp for long time.
6. Make sure the power stability.
7. Flash memory is not easy to fail.

Boot Mode

1. Make sure the boot pins are set to the right status.
2. It's better don't add external circuit.
3. For standalone system which is boot to flash, check these pins are connect to high or left unconnected.
4. These GPIO can be used normally after boot to the right mode.

Table 3-6. Boot Mode Selection

MODE	GPIO37/TDO	GPIO34/COMP2OUT/ COMP3OUT	$\overline{\text{TRST}}$	MODE
3	1	1	0	GetMode
2	1	0	0	Wait (see Section 3.3.10 for description)
1	0	1	0	SCI
0	0	0	0	Parallel IO
EMU	x	x	1	Emulation Boot

Codes run in RAM

Why codes need to run in RAM?

1. Flash setup code must run in RAM
2. Time critical code run in RAM can improve efficiency

To find code example in
C:\tidcs\c28\DSP2803x\v121\DSP2803x_examples\flash\Example_2803xFlash.pjt

Main.c:

```
#pragma CODE_SECTION(InitFlash, "ramfuncs");

MemCopy(&RamfuncsLoadStart, &RamfuncsLoadEnd, &RamfuncsRunStart);
// Call Flash Initialization to setup flash waitstates
// This function must reside in RAM
InitFlash();
```

F28035.cmd:

```
ramfuncs : LOAD = FLASHD,
RUN = RAML0,
LOAD_START(_RamfuncsLoadStart),
LOAD_END(_RamfuncsLoadEnd),
RUN_START(_RamfuncsRunStart),
PAGE = 0
```

Piccolo ADC

It's indicated in silicon errata, and will be fixed in Rev. A.

Advisory	***ADC: Initial Conversion***
Revision(s) Affected	0
Details	When the ADC conversions are initiated by any source of trigger, the first sample may not be the correct conversion result.
Workaround(s)	Discard the first sample at the beginning of every series of conversions. For instance, if the application calls for a given series of conversions, SOC0→SOC1→SOC2, to initiate periodically, then setup the series instead as SOC0→SOC1→SOC2→SOC3 and only use the last three conversions, ADCRESULT1, ADCRESULT2, ADCRESULT3, thereby discarding ADCRESULT0. User application should validate if this workaround is acceptable in their application.
Advisory	***ADC: DC Specifications: Linearity Limitation***
Revision(s) Affected	0
Details	The linearity degrades with increasing temperature in the upper half of the transfer function.
Workaround(s)	The impact from this limitation will be addressed in the next revision (revision A) of the silicon. The following features will be added: 1. ADC clock divider-by-2 enable bit. At 60 MHz, effective sample rate will be 2.3 MSPS. This offers 30-MHz ADC and 60-MHz system clock, and improves linearity. 2. A non-pipeline mode enable bit will be available that offers 3 MSPS at 60-MHz ADC clock with linearity specifications better than can be obtained in the pipeline mode. 3. Existing pipeline mode with 4.6 MSPS at 60-MHz system clock will have improved linearity compared to revision 0 silicon. Details of enabling Options 1 and 2 will be released in future documentation along with the revision A device.

Piccolo Internal Oscillator & Temp Sensor

On-chip Temp sensor helps to Calibrate

1. Analyze thermal behavior of the device in Application set up
2. Analyze package thermal model with different PCB and board/system design
3. On-chip Oscillator for CAN communications for 1% range
4. Control PWM clock period when using HRPWM resolution
5. Control communication speeds in SCI or SPI, if desired.

To find application note on it:

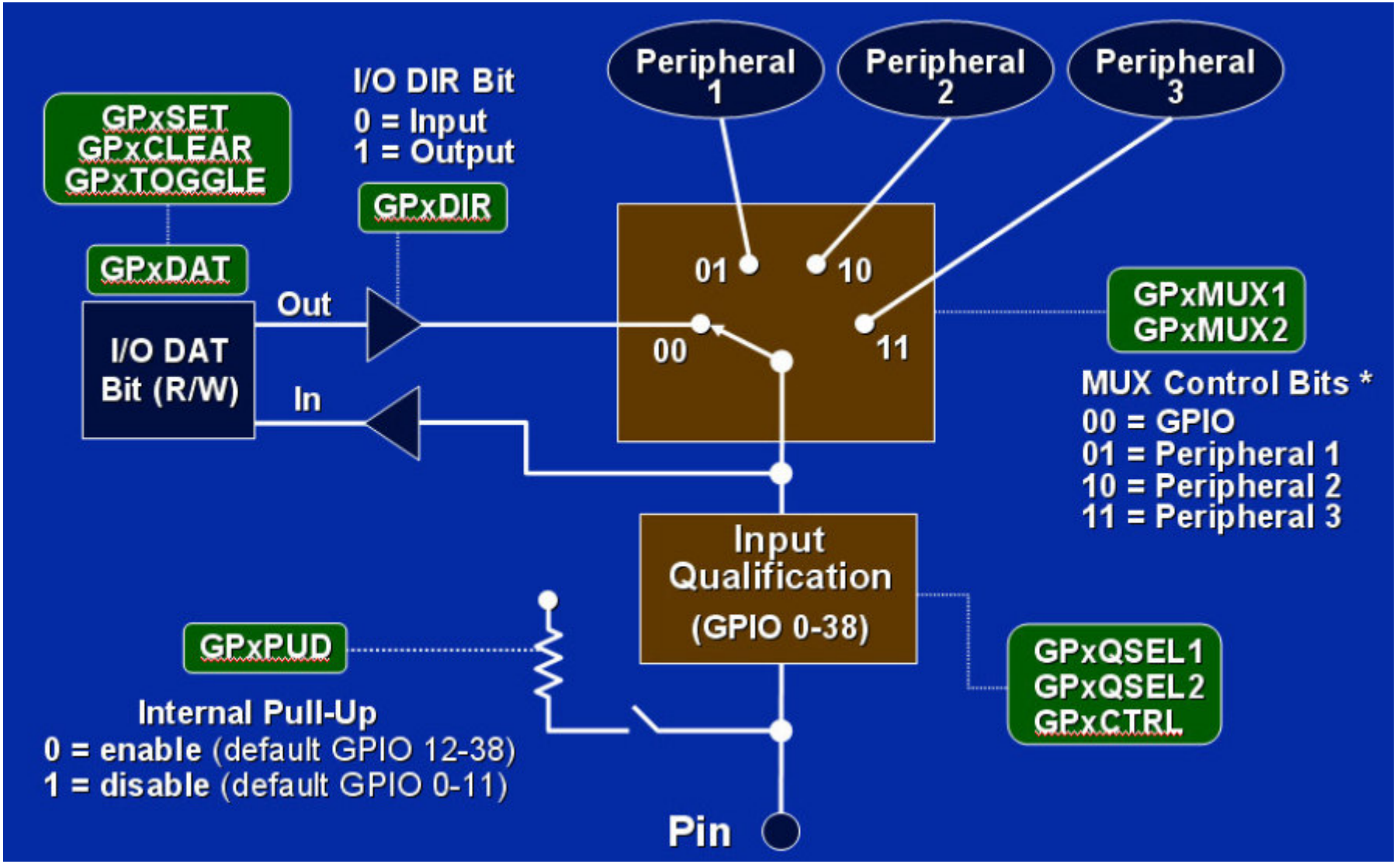
1. Oscillator Compensation Guide (SPRAB84A)
2. Piccolo MCU CAN Module Operation Using the On-Chip Zero-Pin Oscillator (SPRABI7)

Table 6-5. Internal Zero-Pin Oscillator (INTOSC1/INTOSC2) Characteristics

PARAMETER		MIN	TYP	MAX	UNIT
Internal zero-pin oscillator 1 (INTOSC1)(1)(2)	$t_{c(ZPOSC1)}$, Cycle time	97.09	100	103.09	ns
	Frequency	9.7	10	10.3	MHz
Internal zero-pin oscillator 2 (INTOSC2)(1)(2)	$t_{c(ZPOSC2)}$, Cycle time	97.09	100	103.09	ns
	Frequency	9.7	10	10.3	MHz
Step size (coarse trim)			55		kHz
Step size (fine trim)			14		kHz
Temperature drift(3)			3		kHz/°C
Voltage (V_{DD}) drift(3)			175		Hz/mV

(1) In order to achieve better oscillator accuracy (10 MHz ± 1% or better) than shown, refer to the oscillator calibration example in *2803x C/C++ Header Files and Peripheral Examples* (literature number SPRC892), and the *Oscillator Compensation Guide Application Report* (literature number SPRAB84).
(2) Frequency range ensured only when VREG is enabled, $\overline{VREGENZ}$ = V_{SS}.
(3) Output frequency of the internal oscillators follows the direction of both the temperature gradient and voltage (V_{DD}) gradient. For example:
- Increase in temperature will cause the output frequency to increase per the temperature coefficient.
- Decrease in voltage (V_{DD}) will cause the output frequency to decrease per the voltage coefficient.

Output GPIO Hi/Lo with SET/CLEAR

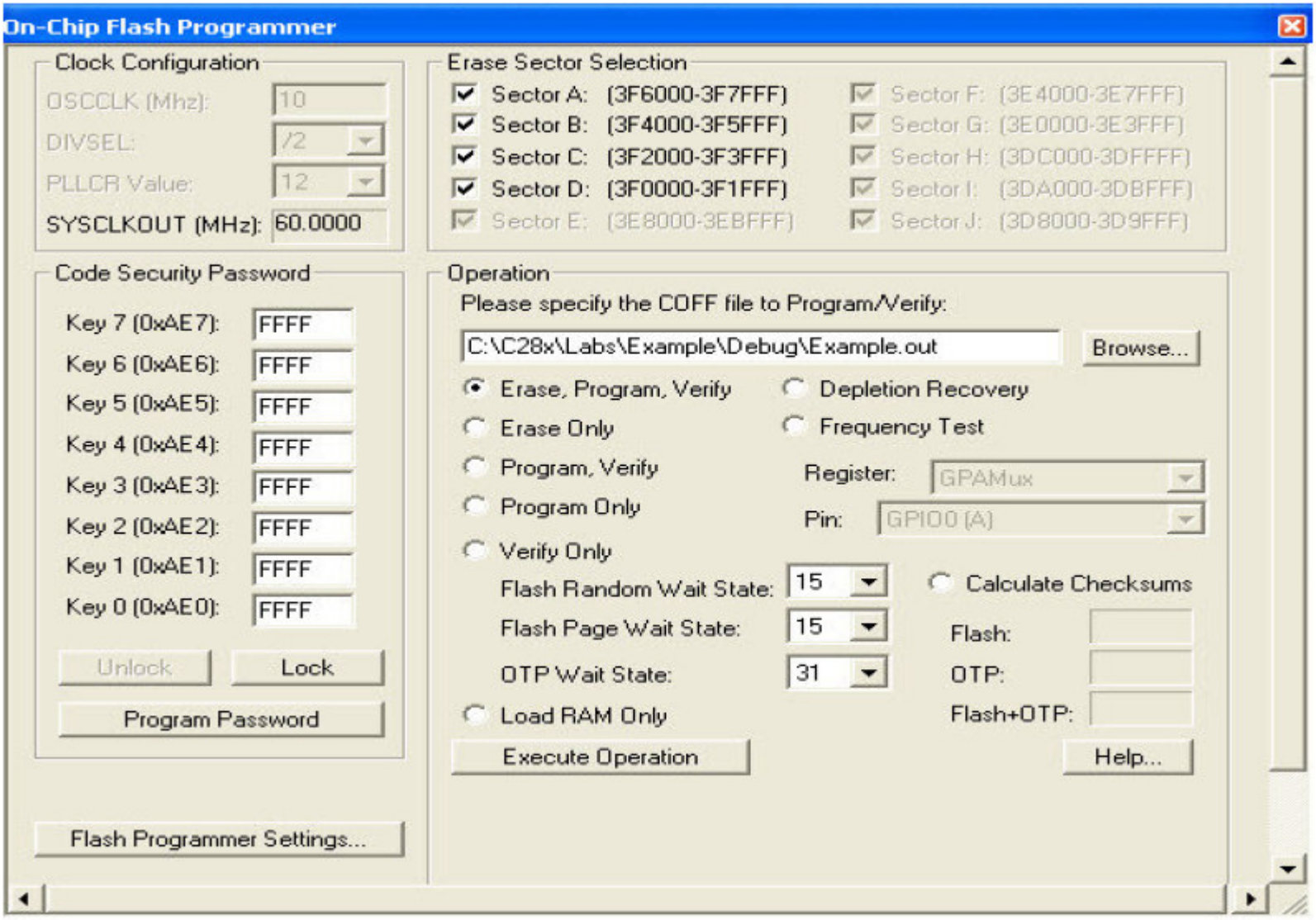


XINTF Difference Between F281x and F2833x

Initial XINTF in DSP2833x_CodeStartBranch.asm if external RAM used.

	F2833x	F281x
1.Data Bus Width	16-bit or 32-bit	16-bit
2.Address Bus Reach	20bit 1M*16bit	19bit 512k*16bit
3.Direct Memory Access	DMA access	not include DMA
4.XINTF Clock Enable	dedicated control	always enable
5.XINTF Pin Muxing	MUXed with GPIO	dedicated pins
6.NO.of Zones&Chip Select Signals	Zone 0、6、 7	Zone 0 1、 2、 6 7
7.Zone7 Memory Mapping	Zone6 7 not share	Zone 7 mirror 6
8.Zone Memory Loactions	Zone 0 4k*16bit	Zone 0 8k*16bit
9.EALLOW protection	EALLOW protect	not

Flash Programming with Plug-In



Q & A

Thanks!