Application Report
SPRABY7–June 2015

Using the Piccolo™ CAN Bootloader at High Temperature

Adam Haun

ABSTRACT

The Controller Area Network (CAN) protocol requires strict tolerances from reference bit clocks, which are generally met using quartz crystal oscillators or ceramic resonators. The TMS320F2803x, TMS320F2805x, and TMS320F2806x microcontrollers use an on-chip zero-pin oscillator with no external components as a reference clock for the CAN bootloader. This oscillator's tolerance is sufficient at room temperature, but not across the full operating temperature range of the device. To use the CAN bootloader at all temperatures, it is necessary to re-trim the oscillator using the on-chip temperature sensor. The re-trim code can be programmed into the OTP memory to allow the entire Flash memory to be reprogrammed during the boot load process.

Project collateral and source code discussed in this application report can be downloaded from the following URL: http://www.ti.com/lit/zip/spraby7.

Contents

1 Introduction

Application code on the Piccolo series of devices can invoke the CAN bootloader in ROM as part of its firmware update routine. The advantage of using the ROM bootloader is that the entire Flash memory can be erased and programmed without needing to implement RAM functions in the application or use a debugger and a IEEE Standard 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture (JTAG) interface.

When run as part of the boot ROM startup process, the CAN bootloader uses the on-chip oscillator with its default room temperature trim. This is sufficient for temperature-controlled manufacturing processes, but run-time firmware updates may need to work beyond 25°C. To support this, the application can re-trim the on-chip oscillator using the on-chip temperature sensor. To maintain the simplicity of using the ROM bootloader, the re-trim routine can be programmed into the device's OTP memory.

This application report will demonstrate the procedure for TMS320F2806x devices. Trim functions are available as part of controlSUITE™. A description of the oscillator compensation method can be found in the *Oscillator Compensation Guide* (SPRAB84). A description of the CAN bootloader functionality can be found in the *Boot ROM* section of the *TMS320x2806x Piccolo Technical Reference Manual* (SPRUH18).

2 Implementation

There are four steps in the temperature-compensated boot load process that must be performed with the PLL bypassed. The code shown here uses the support functions and register definitions included in controlSUITE.

The first step is to connect the ADC to the temperature sensor and sample the temperature:

```
// Initialize the ADC and connect it to the temperature sensor
InitAdc();
EALLOW;
AdcRegs.ADCCTL2.bit.ADCNONOVERLAP = 1;       //Enable non-overlap mode
AdcRegs.ADCCTL1.bit.TEMPCONV  = 1;           //Connect channel A5 internally to the temperature
sensor
AdcRegs.ADCSOC0CTL.bit.CHSEL  = 5;           //Set SOC0 channel select to ADCINA5
AdcRegs.ADCSOC0CTL.bit.ACQPS  = 25;          //Set SOC0 acquisition period to 26 ADCCLK
AdcRegs.INTSEL1N2.bit.INT1SEL = 0;           //Connect ADCINT1 to EOC0
AdcRegs.INTSEL1N2.bit.INT1E  =  1;           //Enable ADCINT1

//Wait for end of conversion.
while(AdcRegs.ADCINTFLG.bit.ADCINT1 == 0){}  //Wait for ADCINT1
AdcRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;        //Clear ADCINT1
asm(" RPT #255 || NOP");

//Get temp sensor sample result from SOC1
sample = AdcResult.ADCRESULT0;
```

The second step is to call the oscillator compensation functions with the sample value:

```
Osc1Comp(sample);
Osc2Comp(sample);
```

The third step is to invoke the CAN bootloader in ROM. On F2803x devices, the CAN bootloader function starts at address 0x3ff641. On F2805x devices, the function starts at address 0x3fff01. On F2806x devices, the function starts at address 0x3ff4fc. This function takes no arguments and returns a 32-bit value containing the entry point from the loader data stream. The easiest way to invoke the bootloader is with a function pointer:

```
Uint32 (*CAN_Boot_ptr)(void) = (Uint32 (*)(void))0x3ff4fc;

//After calling Osc1Comp() and Osc2Comp()
entryAddr = (*CAN_Boot_ptr)();
```

To better support flash reprogramming, it is convenient to program these compensation and wrapper functions into the flash OTP memory. The two provided files show how to do this. HT_CAN_Boot_Wrapper.c contains the compensation and bootloader wrapper functionality described above. HT_CAN_Boot_Wrapper.cmd is a linker command file that assigns these functions to OTP addresses.

Specifying the load address for a function involves three steps.

The first step is to define a memory region in page 0 in the linker command file. Here, two regions are defined: one for the main wrapper function and one for the ADC and compensation routines. This allows the main wrapper function to have a known address that can be referred to in the application code:

```
OTP_CALLADDR : origin = 0x3D7840, length = 0x000040 /* on-chip OTP --
 specific address for the main wrapper function */
OTP_FUNCS    : origin = 0x3D7880, length = 0x000380 /* on-chip OTP --
 wrapper support functions */
```

The second step is to create an assembly section for the function or functions that will be loaded into the memory region. Here, two sections are defined: one for each region:

```
/* High-temp CAN bootloader wrapper functions */
LoaderWrapper      : > OTP_CALLADDR,  PAGE = 0
LoaderWrapperFuncs : > OTP_FUNCS,     PAGE = 0
```

The final step is to use the CODE_SECTION #pragma to assign individual functions to assembly sections:

```
#pragma CODE_SECTION(CompensatedCANBootLoader, "LoaderWrapper")
void CompensatedCANBootLoader(void)
{
     ...
}

#pragma CODE_SECTION(InitAdc, "LoaderWrapperFuncs")
void InitAdc(void)
{
     ...
}

#pragma CODE_SECTION(Osc1Comp, "LoaderWrapperFuncs")
void Osc1Comp (int16 sensorSample)
{
     ...
}

#pragma CODE_SECTION(Osc2Comp, "LoaderWrapperFuncs")
void Osc2Comp (int16 sensorSample)
{
     ...
}

#pragma CODE_SECTION(GetOscTrimValue, "LoaderWrapperFuncs")
Uint16 GetOscTrimValue(int Coarse, int Fine)
{
     ...
}
```

With this done, the resulting .out file will have the desired load and run addresses for the functions.

3 References

- *Oscillator Compensation Guide* (SPRAB84)
- *TMS320x2806x Piccolo Technical Reference Manual* (SPRUH18)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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