



Embedded Systems Course



Laboratory Practical 5: TI Piccolo MCU controlSTICK




NOTE:
Short report to be
written... see last
page!!

TEAM:	Two
DURATION:	2 hours
DOC REF:	Prac5_Piccolo.docx

1	Introduction.....	1
2	The Piccolo MCU controlSTICK	1
3	Getting started with the Piccolo controlSTICK.....	2
3.1	Software Installation	2
3.2	Hello Piccolo - BlinkingLED.....	2
4	Simple DAC-ADC	4
4	Submission for this prac	4

1 Introduction

This practical builds on the student’s knowledge of microcontrollers as discussed in the course work and employed in practices using the CSB337 board during the last semester. Specifically, the focus of this practical is to introduce the student to a new evaluation kit, the TI Piccolo MCU controlSTICK.

The practice covers the following tasks:

- Setting up the kit ready for embedded development
- Running the BlinkingLED demo project
- Running the SimpleDACtoADC demo project and extending it to produce a simple siren-like sound
- Writing a report on the work done

2 The Piccolo MCU controlSTICK

The Piccolo microcontroller controlSTICK is a fully functional evaluation kit, slightly larger than a memory stick, which allows quick and simple evaluation of all the functionalities of Texas Instruments’ new Piccolo microcontroller. The kit is shown in figure 2.1 below with key components labeled.

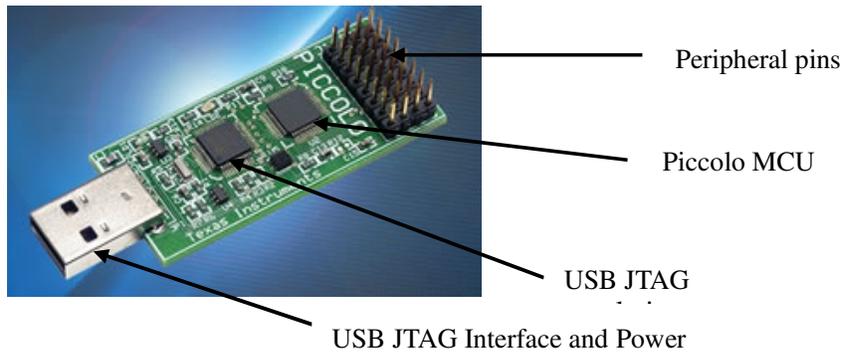


Figure 2.1 The Piccolo MCU ControlSTICK

3 Getting started with the Piccolo controlSTICK

3.1 Software Installation

This subsection guides you through installing the necessary software to ready the kit for embedded development. The two software tools that you have to install are the Code Composer Studio (CCStudio) version 4 and the controlSUITE.

The CCStudio software is a C/C++ integrated development environment (IDE) for all TI's embedded development and debugging. To install this software, follow the simple steps outlined below:

- Insert the CCStudio DVD into your PC
- Open the DVD , locate the setup program and run it
 - This will install the CCS software onto your machine

The next software that we need to install is the controlSUITE. The suite comprises a cohesive set of software infrastructure and tools that simplifies embedded software development. To install the software, download its zip package from <http://focus.ti.com/docs/toolsw/folders/print/controlsuite.html> under the additional information title and copy it to C:\.

3.2 Hello Piccolo – BlinkingLED

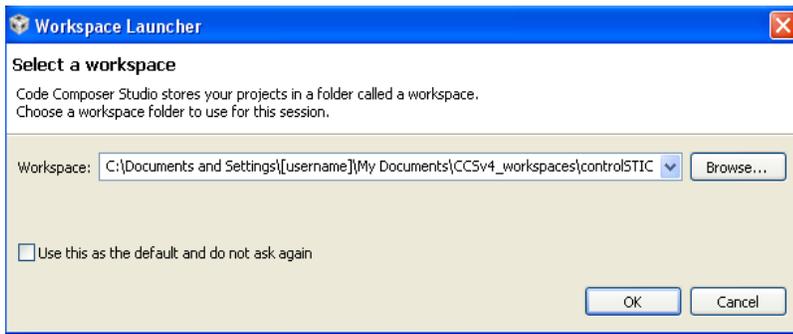
Now that you have successfully completed the software installations necessary to begin development on the controlSTICK, this subsection guides you through the running of one of the controlSTICK example projects, BlinkingLED.

ControlSTICK Hardware Setup:

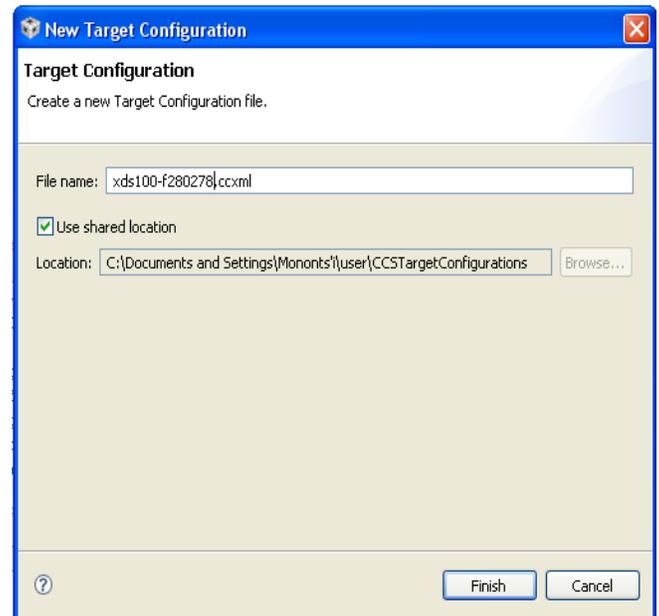
1. Browse to the folder “C:\TI\controlSUITE\development_kits\Piccolo controlSTICK\” and open the Timer-BlinkingLED project folder. Inside this folder’s ~doc directory, open up the project’s user guide for further reference.
 - The user guide contains details of the software and hardware configurations for each project.
2. Plug in the controlSTICK to an open USB port and let windows scan and install its drivers

Setting up CCS to work with the controlSTICK:

3. Open CCS version 4
4. The workspace launcher should appear,
 - click on browse and by creating new folders as necessary, set your workspace to “C:\Documents and Settings\[username]\My Documents\CCSv4_workspaces\controlSTICK”.
 - Uncheck the box that says “Use this as the default and do not ask again”
 - Click OK



5. The next step involves configuring the CCS to know what specific Piccolo MCU it will be connecting to. We will create a target configuration file to specify this.
 - Click Target -> New Target Configuration and change the new configuration to xds100-f28027.ccxml. Check the “use shared location” checkbox and click finish.
 - On the new “General Setup”
6. On the new “General Setup” tab, enter the following options and save the configuration
 - Connection – Texas Instruments XDS100v1
 - Device – TMS320F28027
7. Add all the projects into your current workspace by
 - Browse to the root directory of the controlSTICK where you have copied the controlSUITE files. This will be: “C:\TI\controlSUITE\development_kits\Piccolo controlSTICK”.
 - Click finish



Configuring the BlinkingLED project:

8. Expand the file structure of the BlinkingLED from the C/C++ Projects tab. Right click on the project's name and select “Set as active Project”.
9. Each project can be configured to create code and run in either flash or RAM. For this exercise we will run in RAM. Right click on the BlinkingLED project and select “Active Build Configuration -> F2802x_RAM”

Building and loading the project:

10. Click the debug button. This will compile the project and the output file should be automatically loaded into the F28027 device.
11. Select Tools -> Debugger Options -> Generic Debugger options. In real-time options box, check the following boxes:
 - Enable silicon real-time mode
 - Enable polite real-time mode

Setting up the watch window:

12. Open a watch window to view the variables used in the project.
Click: View -> Watch on the menu bar

In the watch window add the CpuTimer0Regs.PRD.all variable defined under the software setup section of the BlinkingLED.pdf file opened in step 1

13. Click on the continuous refresh button in the watch window. This enables the window to run with real-time mode.

Running the code

14. To run the code, press the green Run button in the Debug window.
 - This should cause the LED on the controlSTICK to start blinking

Exercise

Experiment more with the project by:

- Varying the frequency of the LED (The current configuration is such that the LED blinks approximately once every second, change it to once every two seconds). In your report, describe how you achieved this.
- If the kit had an extra LED (say called GPIO-35), how would you change the program so that it produces a rolling (LED2 on, LED3 off, LED2 on LED3 off ...) blinking pattern?

4 Simple DAC-ADC

The purpose of this section is to demonstrate how to perform continuous conversion of one ADC channel using a simple DAC as the input signal. The Piccolo microcontroller does not have a DAC among its peripherals. Instead, the ePWM peripheral is configured and used as a DAC.

Hardware Setup:

On the peripheral pins, use the jumper wire to connect the output of the filtered ePWM (PWM1A-DAC) to channel 0 of the ADC (ADC-A0). Refer to the pin layout shown in the SimpleDACtoADC project manual to locate correct pin connections.

Software Setup:

Add the following variables to the watch window.

- adc_result (format = hexadecimal)
- duty_cycle (format = decimal)

Debug and run the project by following the same steps as outlined for the BlinkingLED program in section 3.

Exercise

Earlier in this section, it was mentioned that the Piccolo MCU does not have a DAC as one of its on-core peripherals. Describe how the PWM peripheral is used to “fake” a DAC in this project. Use diagrams (circuits?) where necessary to clarify your description.

Given a simple low-power speaker, along with basic electrical elements such as resistors and capacitors, describe how you would use them to extend this project to produce a siren-like sound. Implement your prototype, mention and comment on your findings.

5 Submission for this Prac

The submission for this practice is a report that covers the following:

- Answers to the questions in sections 4 and 5 of the prac
- A half page report that (i) Gives your understanding of what micro-controllers are and (ii) includes a discussion of five areas where they are applied with a specific example given for each area.