

CROSSHAIRS []
EMBEDDED

Invaluable [insight] through precise software solutions

[**Crosshairs Embedded
F2803x Piccolo Experimenter's
Kit
User Guide**]

Contents

Contents	2
Requirements	3
Crosshairs [debugger] and Crosshairs [interface designer]	3
Code Composer Studio 4	3
F2803x Piccolo Experimenter's Kit	3
controlSUITE Installation	4
Configuration	6
Working with CCS4	8
Configuring your workspace	8
Importing example project.....	8
Understanding the example	9
Commros-files.....	9
Device support-files	9
A detailed look at Commros	9
Compiling	9
Modifying the example	10
Programming the Experimenter's Kit.....	11
Using the Crosshairs [debugger]	12
Creating a debug session.....	12
Debugging the application	13
Where to go from here.....	14
FAQ.....	16
Customer Contact.....	17

Requirements

Before proceeding, please make sure you have the following software and hardware properly installed and configured on your computer.

Crosshairs [debugger] and Crosshairs [interface designer]

The Crosshairs [debugger] and Crosshairs [interface designer] may be downloaded for free (30-day trial) from the following location:

<http://crosshairembedded.com/free-trial>

Please note that after registering for a free trial, the download links are only valid for a limited time. Download and install the Crosshairs [debugger] and Crosshairs [interface designer]. The stand-alone installation of the Crosshairs [engine] is not required for this tutorial.

Code Composer Studio 4

This may be downloaded for free from Texas Instruments at the following location:

<http://focus.ti.com/docs/toolsw/folders/print/ccstudio.html>

In this tutorial, Code Composer Studio 4 Microcontroller Limited edition was used (TMD FCCS-MCULTD.)

F2803x Piccolo Experimenter's Kit

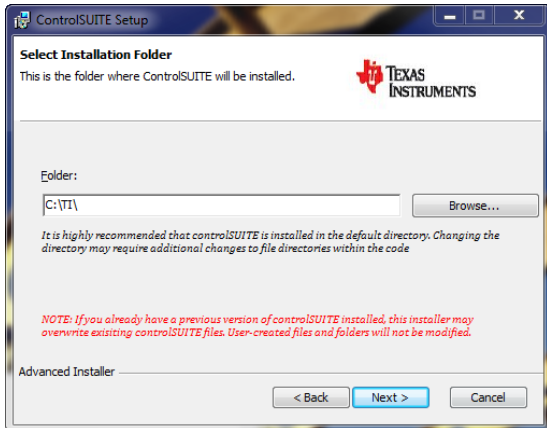
The Piccolo Experimenter's kits are available from Texas Instruments or one of its suppliers. In this particular tutorial, the F28035 Experimenter's kit was used. For ordering information, please refer to the following URL:

<http://focus.ti.com/docs/toolsw/folders/print/tmdxdock28035.html>

controlSUITE Installation

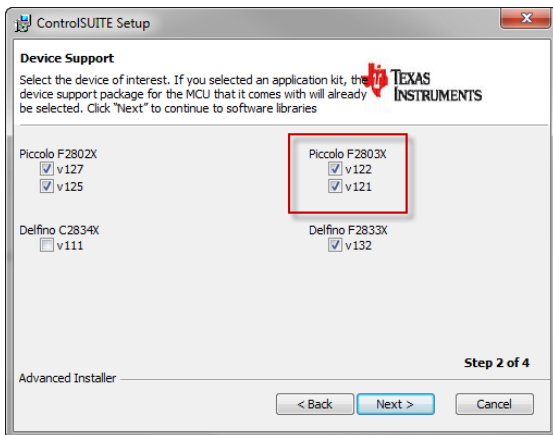
The relevant software and resource files needed for development on the F2803x Piccolo Experimenter's Kit (besides the CCS4 installation itself) is an installation of the TI controlSUITE software package. This is available from the link below:

<http://focus.ti.com/docs/toolsw/folders/print/controlsuite.html>

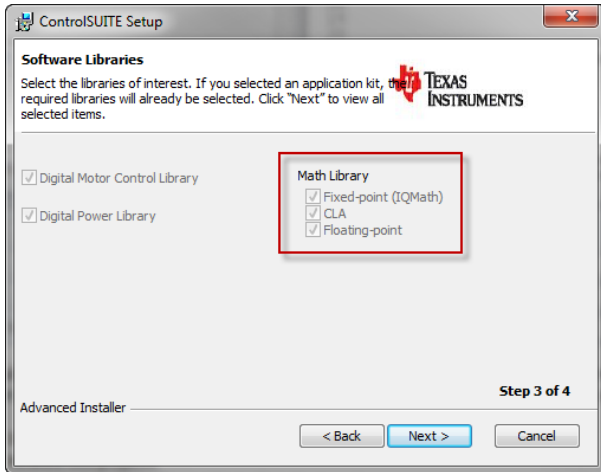


We recommend using the default path for installation "C:\TI\" for simplicity. If this installation path is changed certain include folders and library folders may need to be changed in the example projects for the example application to work.

When installing controlSUITE using custom mode instead of full install please ensure that these checkboxes are checked:



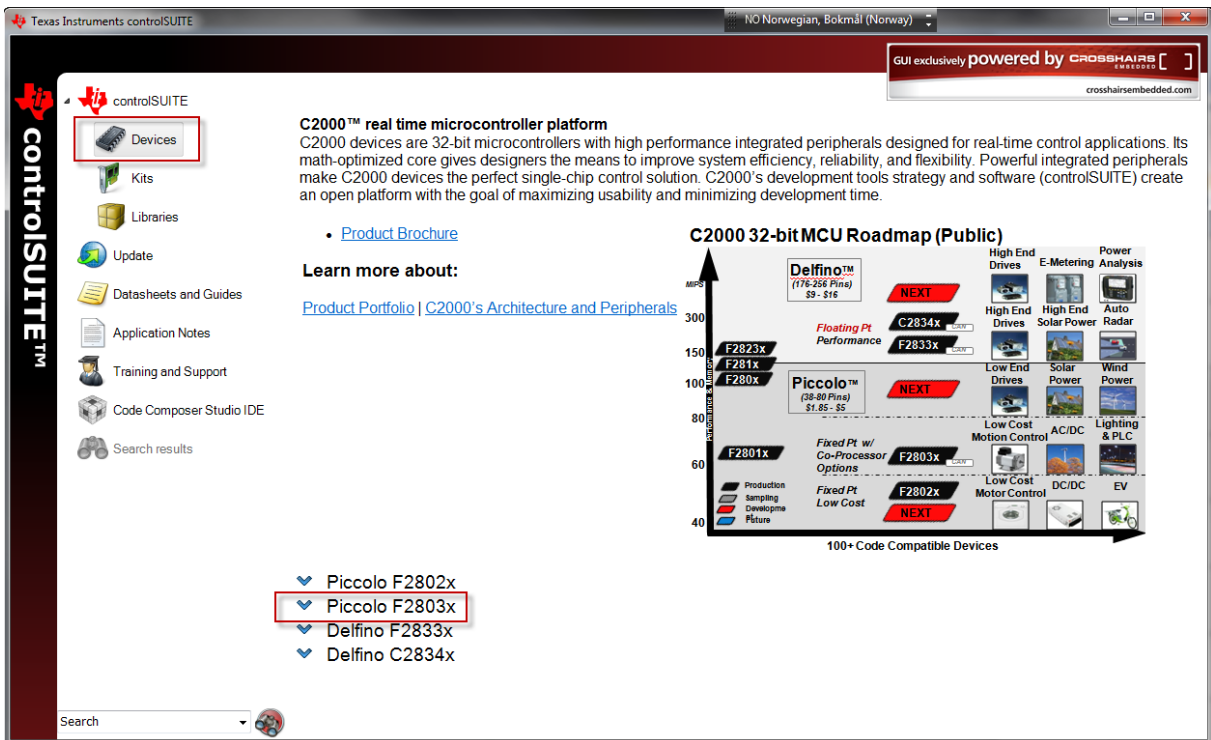
Piccolo F2803x support files



Digital Motor Control Library, Fixed-point Math library CLA and Floating-point Math library

After a successful install you can browse the installed files for the development kits and the support files for the different controlCARDS etc. through the controlSUITE Desktop application.

Open up the controlSUITE Desktop and expand the controlSUITE node and select Kits. In Kits press the arrow to expand C2000 controlCARDS

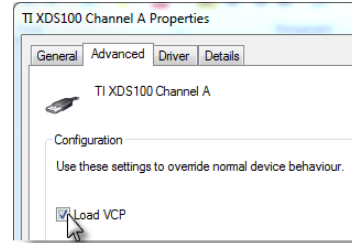


Here you will find relevant info about the Piccolo 28035 controlCARD.

Configuration

Serial using FTDI-Chip (RS232 over USB)

After installing CCS4 on your system drivers for the FTDI chip should be present on your system. Hook up the kit with the USB-cable, open the Windows Device Manager. Expand the Universal Serial Bus Controllers section and find the two TI XDS100 entries (Channel A and Channel B.)

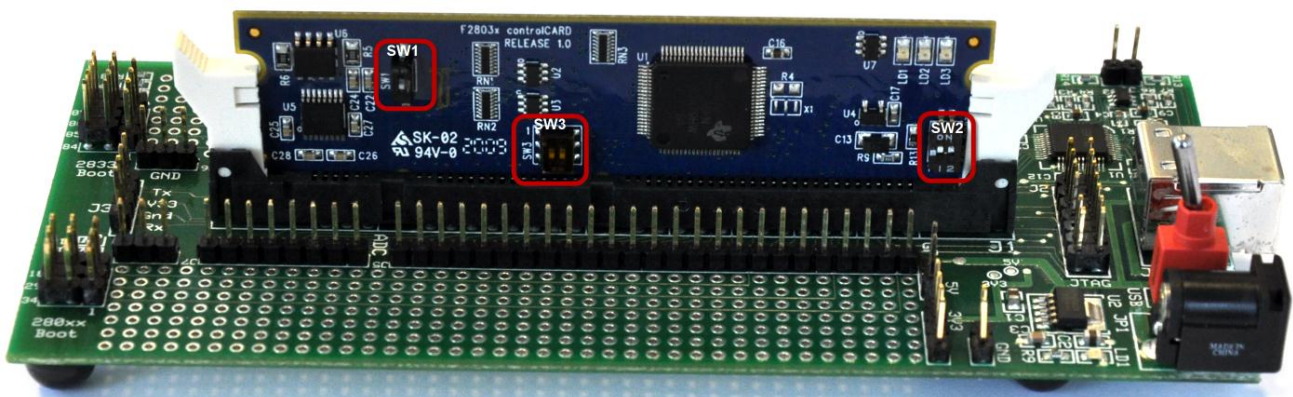


For both entries, open the Properties dialog and select the Advanced tab. Click to enable "Load VCP" as shown in the screenshot. This enables the virtual serial port for the device.

After enabling VCP for both channels, unplug and replug the Piccolo Experimenter's Kit. You should now see two new COM-ports in your Device Manager under the Ports (COM & LPT) section.

Jumpers and Switches

Please make sure that the following jumpers and switches are in the correct positions



SW1	
POS	1
ON	
OFF	

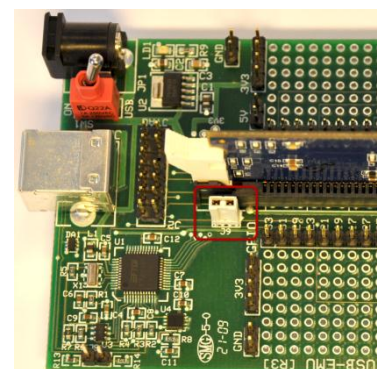
SW3		
POS	1	2
ON		
OFF		

SW2		
POS	1	2
ON		
OFF		

SW1: Controls whether GPIO-28 is used for RS232 or not. Turning this **off** releases GPIO-28 and lets us use the FTDI-chip

SW2: Controls boot mode, currently set to boot from FLASH.

Note that J9 (see below) does not have pinouts by default. In order to properly close J9, pins will have to be soldered on and populated with a jumper.

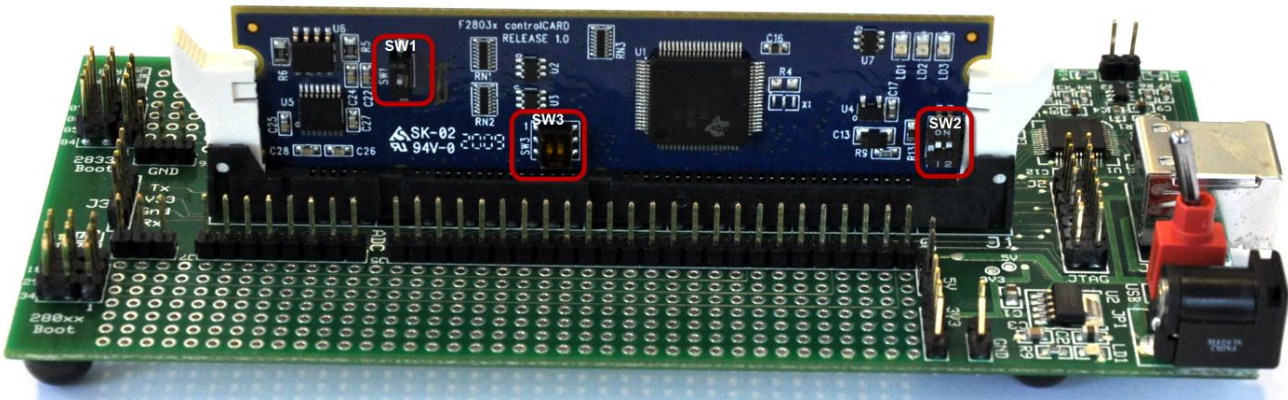


J9 enables or disables routing serial communication through the FTDI-chip.

Serial using GPIO-28 / J3 pinouts

Jumpers and Switches

Please make sure that the following jumpers and switches are in the correct positions



SW1	
POS	1
ON	<input type="checkbox"/>
OFF	<input type="checkbox"/>

SW3		
POS	1	2
ON	<input type="checkbox"/>	<input type="checkbox"/>
OFF	<input type="checkbox"/>	<input type="checkbox"/>

SW2		
POS	1	2
ON	<input type="checkbox"/>	<input type="checkbox"/>
OFF	<input type="checkbox"/>	<input type="checkbox"/>

SW1: Controls whether GPIO-28 is used for RS232 or not. Turning this **on** enables RS232 pins 2 and 42 on the DIMM-100 socket. A RS232 cable must be connected to J3.

SW2: Controls boot mode, currently set to boot from FLASH.

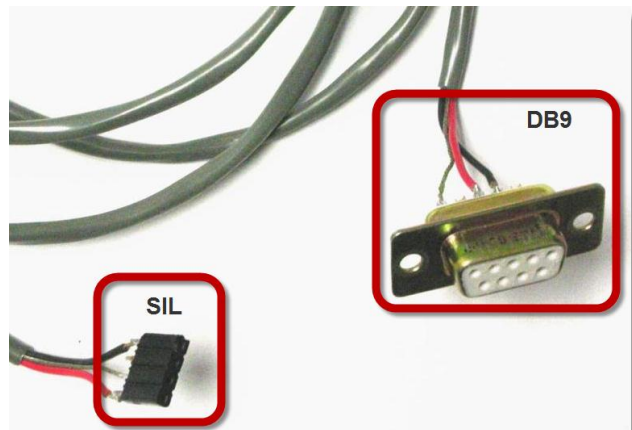
Jumper J9 must not be populated with a jumper!

Serial cable, J3-pinouts

A serial cable has to be connected to J3 on the docking card. Connections should be as follows

DB9-side pins	SIL pins
2 (Red)	1 (TX)
3 (Black)	4 (RX)
5 (Gray)	3 (GND)

Do not connect pin 2 on the SIL-side.



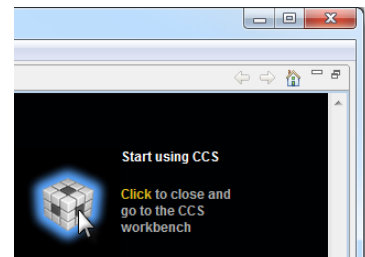
Working with CCS4

After successfully installing Code Composer Studio 4 and completing the Requirements section above, please start CCS4.

Configuring your workspace

When you start CCS4, you will by default be prompted to select the workspace you would like to use. Select a folder you would like to use and click OK. Please note that if you are an existing CCS4 user, you might want to create a new workspace, for example `\crosshairs_examples`.

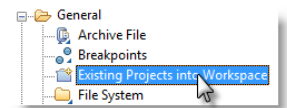
When CCS4 loads and the welcome screen is visible, click the cube in the top-right corner to access the CCS Workbench.



Importing example project

With a clean workspace, open the File menu and select **Import...**

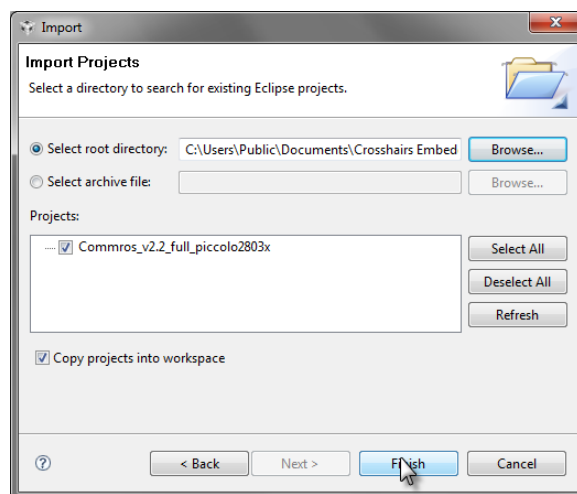
Expand the General folder, select Existing Projects into Workspace and click Next.



In the next step, we browse to and select the appropriate example from your Crosshairs installation. The example locations depend on if you installed for all users (Public or Shared Documents) or just the current user (My Documents.)

Browse to `<Documents folder>\Crosshairs Embedded\
crosshairs[commros]\TMS320F28xxx\Examples\CCS4\Commros_v2.2_full_piccolo2803x` and click OK.

Note that by default, CCS4 does not make a copy of imported projects and will rather modify the original versions. We recommend checking the Copy projects into workspace checkbox before clicking Finish.



Understanding the example

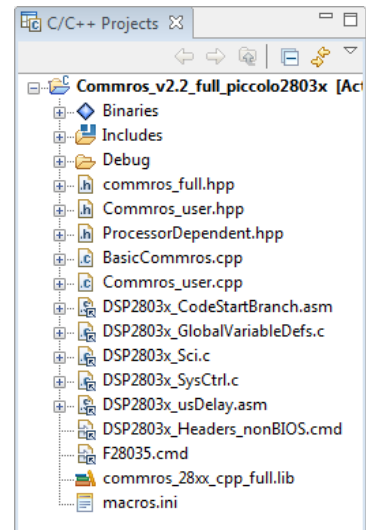
In the C/C++ Projects tab, you will now see the Piccolo2803x example. A description of the files follows below:

Commros-files

- commros_full.hpp
 - Include for the full version of Commros
- Commros_user.hpp / .cpp
 - User-specific Commros implementation
- BasicCommros.cpp
 - Application entry point
- Commros_28xx_cpp_full.lib
 - Commros library reference

Device support-files

- ProcessorDependent.h
- DSP2803x_*.c / .asm / .cmd
- F28035.cmd



A detailed look at Commros

Before proceeding, please read the Crosshairs [commros] User Guide that is installed with all Crosshairs products. This is found in the Crosshairs Embedded program group on the start menu.

The entry point of the application is found inside BasicCommros.cpp. Here you will find a reference to Commros_user.hpp as well as calls to initialize and service Commros.

```
1 #include "DSP2803x_Device.h" // DSP280x Headerfile Include File
2 #include "DSP2803x_Examples.h" // DSP280x Examples Include File
3
4 #include "Commros_user.hpp" // Commros configuration
5
6 int main(void)
7 {
8     InitSysCtrl();
9     DINT;
10    InitCommros(); // Initialize commros
11
12    for(;;)
13    {
14        commros.ServiceRoutine(); // Call commros
15    }
16 }
```

The call to `InitCommros()` on line 10 takes care of configuring the Commros kernel, the communication callbacks as well as the Serial communication.

To look at the implementation, open the file `Commros_user.cpp` and scroll to line 96. Please note that the initialization (in particular for Serial communication) is tailored specifically to the Piccolo ControlSTICK and may not work without modification for other development kits.

The call to `commros.ServiceRoutine()` on line 14 handles outstanding communication requests to the Commros kernel from Crosshairs client applications. This call may be placed in the main-loop, in a low-priority OS task or an interrupt service routine.

Compiling

To compile the example, right-click on the Project and select Build Project or open the Project menu and select Build Project from there. If there are compile errors, please refer to the FAQ on page 13 or contact support@crosshairembedded.com.

Modifying the example

In order to have something interesting to look at in the Crosshairs [debugger] and [interface designer] we modify the example with the following changes (all changes are in BasicCommros.cpp)

Before `int main(void)` add the following line (line 5)

```
int testVarA, testVarB, testVarC;
```

After the call to `InitCommros`, add the following line (line 13)

```
testVarA = 0; testVarB = 0; testVarC = 0;
```

Inside the for-loop, add the following four lines (line 18)

```
testVarA++;  
if((testVarA > 30000))  
{  
    testVarA = 0;  
    testVarB++;  
  
    if((testVarB % 40) == 0)  
    {  
        testVarC = testVarB;  
    }  
}
```

The application should now look like this:

```
#include "DSP2803x_Device.h" // DSP2803x Headerfile Include File  
#include "DSP2803x_Examples.h" // DSP2803x Examples Include File  
  
#include "Commros_user.hpp" // Commros configuration  
int testVarA, testVarB, testVarC;  
  
int main(void)  
{  
    InitSysCtrl();  
    DINT;  
    InitCommros(); // Initialize commros  
  
    testVarA = 0; testVarB = 0; testVarC = 0;  
    for(;;)  
    {  
        testVarA++;  
        if((testVarA > 30000))  
        {  
            testVarA = 0;  
            testVarB++;  
  
            if((testVarB % 40) == 0)  
            {  
                testVarC = testVarB;  
            }  
        }  
        commros.ServiceRoutine(); // Call commros  
    }  
}
```

After making the above changes, compile the application again.

Programming the Experimenter's Kit

To program the application to the F28035 Experimenter's kit, open the Target menu and select Debug Active Project.

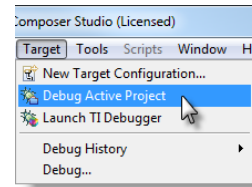
If prompted to create a new target configuration, click Yes and Finish. In the target configuration, change the Connection and Device as follows:

- Connection: Texas Instruments XDS100v1 USB Emulator
- Device: Experimenter's Kit – Piccolo F28035

Click Save and launch the debug session again (open the Target menu and select Debug Active Project.)

If the debug session is launched correctly, you should now be in the Debug perspective and the program counter should be on line 10 in BasicCommros.cpp. To verify that the application runs correctly, click run and halt a few times.

Before proceeding to the Crosshairs products, disconnect the debug session inside CCS4. You may also close CCS4 at this point. Please note that if you halted the application before exiting, unplug and replug the Experimenter's Kit in order to restart the application.



Using the Crosshairs [debugger]

For more comprehensive usage instructions, please refer to the Crosshairs [debugger] User Guide that is installed with the product.

Creating a debug session

After launching the Crosshairs [debugger] open the File menu and select New Project. Give your project a name and provide an (optional) description.

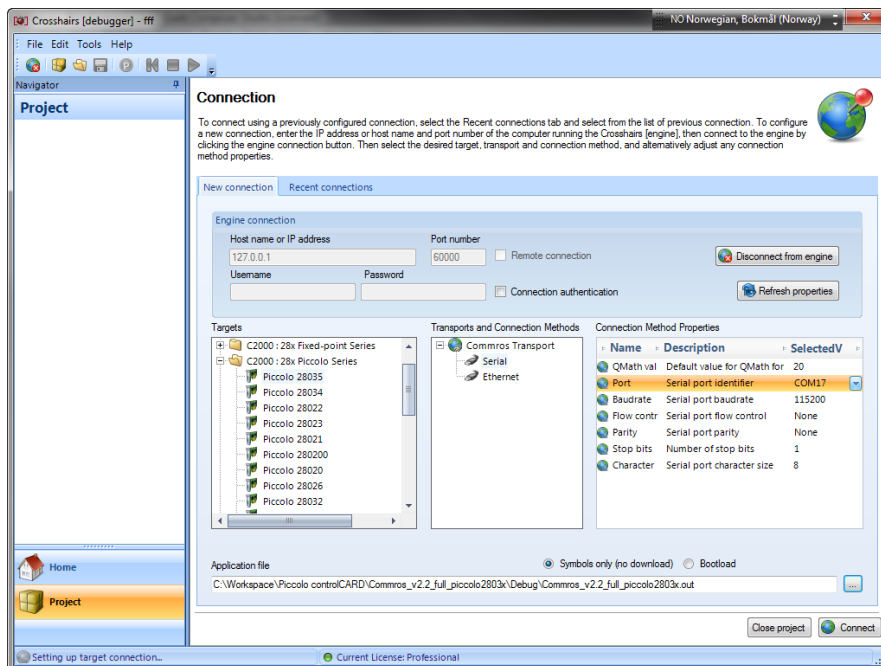
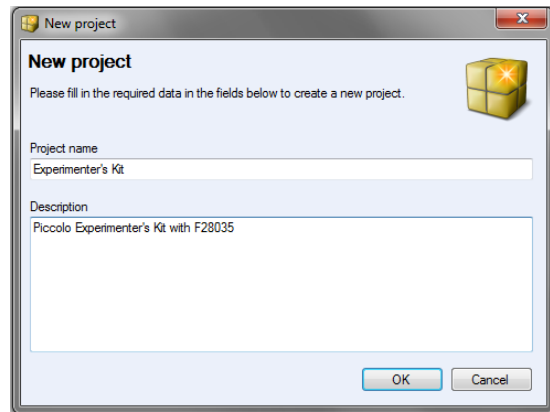
After clicking OK, you are taken to the Connection configuration screen. This is where the connection to the Crosshairs [engine] and, in turn, the hardware is established.

If you are connecting to an engine running on your local computer, leave the Engine Connection settings as they are and click the Connect to Engine button. If prompted to start the engine, click Yes.

In the Targets list, expand the C2000:28x Piccolo Series and select Piccolo 28035.

Select Commros Serial as the transport and select the correct COM-port in the Connection method properties pane.

Next, select the Application file by clicking the [...] button in the lower right corner. Browse to the \Debug folder of the CCS4 project that was created in the previous sections. Once everything is configured properly, click the Connect button.

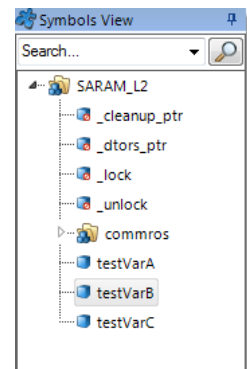


If the connection fails, please refer to the [FAQ](#) or contact support@crosshairembedded.com for assistance.

Debugging the application

Once the connection is established, you will be in the Monitor module of the Crosshairs [debugger].



On the left side, you will find the Symbols View, expand the SARAM_L2 section as shown here.

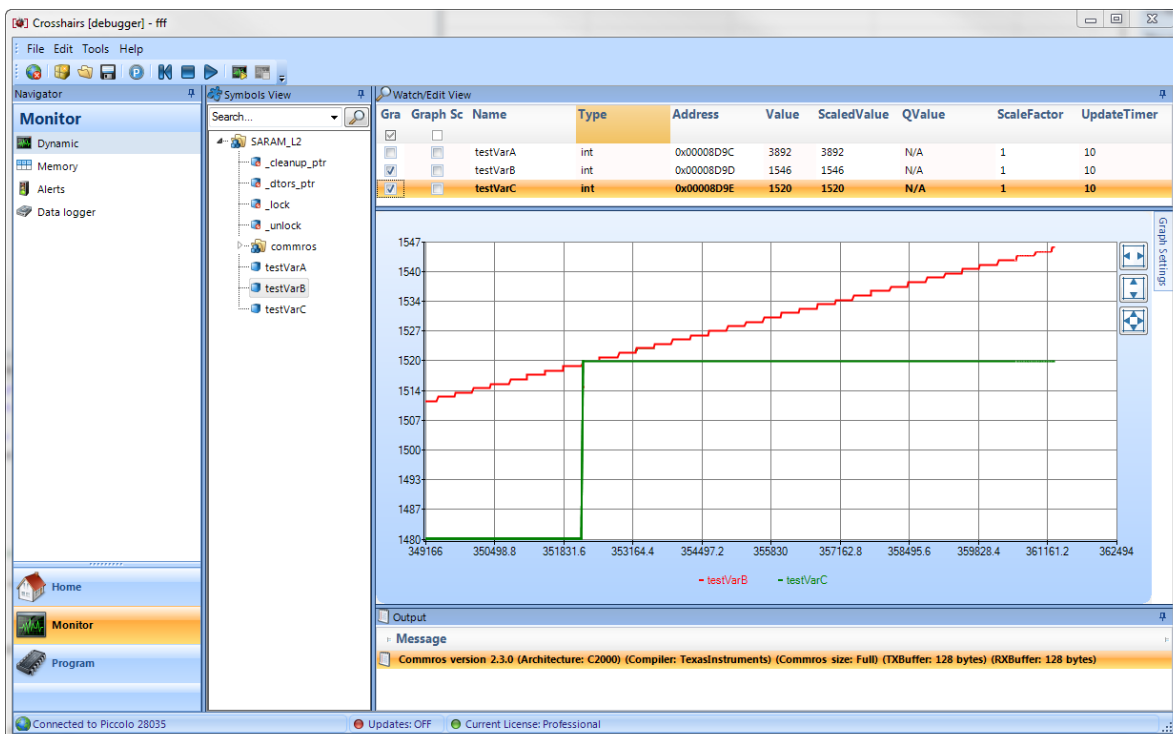


Add testVarA, testVarB and testVarC to the Watch/Edit view by either dragging them to the view, or right-clicking and selecting Add to Watch View.

The three variables should now be visible in the Watch/Edit View. The leftmost column of the Watch/Edit view is used to control which variables are graphed once updating is turned on. Check the box next to testVarB and testVarC.

Graph	Graph Sc	Name	Type	Address	Valu	ScaledValue	QValue	ScaleFa	UpdateTi
<input checked="" type="checkbox"/>	<input type="checkbox"/>								
<input type="checkbox"/>	<input type="checkbox"/>	testVarA	int	0x00008D9C	N/A		N/A	1	10
<input checked="" type="checkbox"/>	<input type="checkbox"/>	testVarB	int	0x00008D9D	N/A		N/A	1	10
<input checked="" type="checkbox"/>	<input type="checkbox"/>	testVarC	int	0x00008D9E	N/A		N/A	1	10

Next, let's turn on the real-time update, either by clicking the Start update button (), or by opening the Tools menu and selecting Start Update. You should now see values updating as well as the graph plotting testVarB and testVarC. Note that if the graph plotted is too flat, click the Fit to window button () on the graph.



Where to go from here

This example show the bare-bone integration of Crosshairs [Commros] in an application to run on the Piccolo controlCARD in the Piccolo controlCARD Experimenter's Kit. Bundled with the installers for the Crosshairs [Debugger] and the Crosshairs [Interface Designer] are several other real-world examples of the usage:

Commros_v2.2_full_dcmotor_CC2803X (28035) – This example showcases a simulator of a standard dc motor control algorithm using IQ math. The main purpose of the example is to showcase how the Crosshairs [Debugger] can be used to control and tune a motor control algorithm. The version of Commros that has been integrated supports multiple variable reads in one call to the library

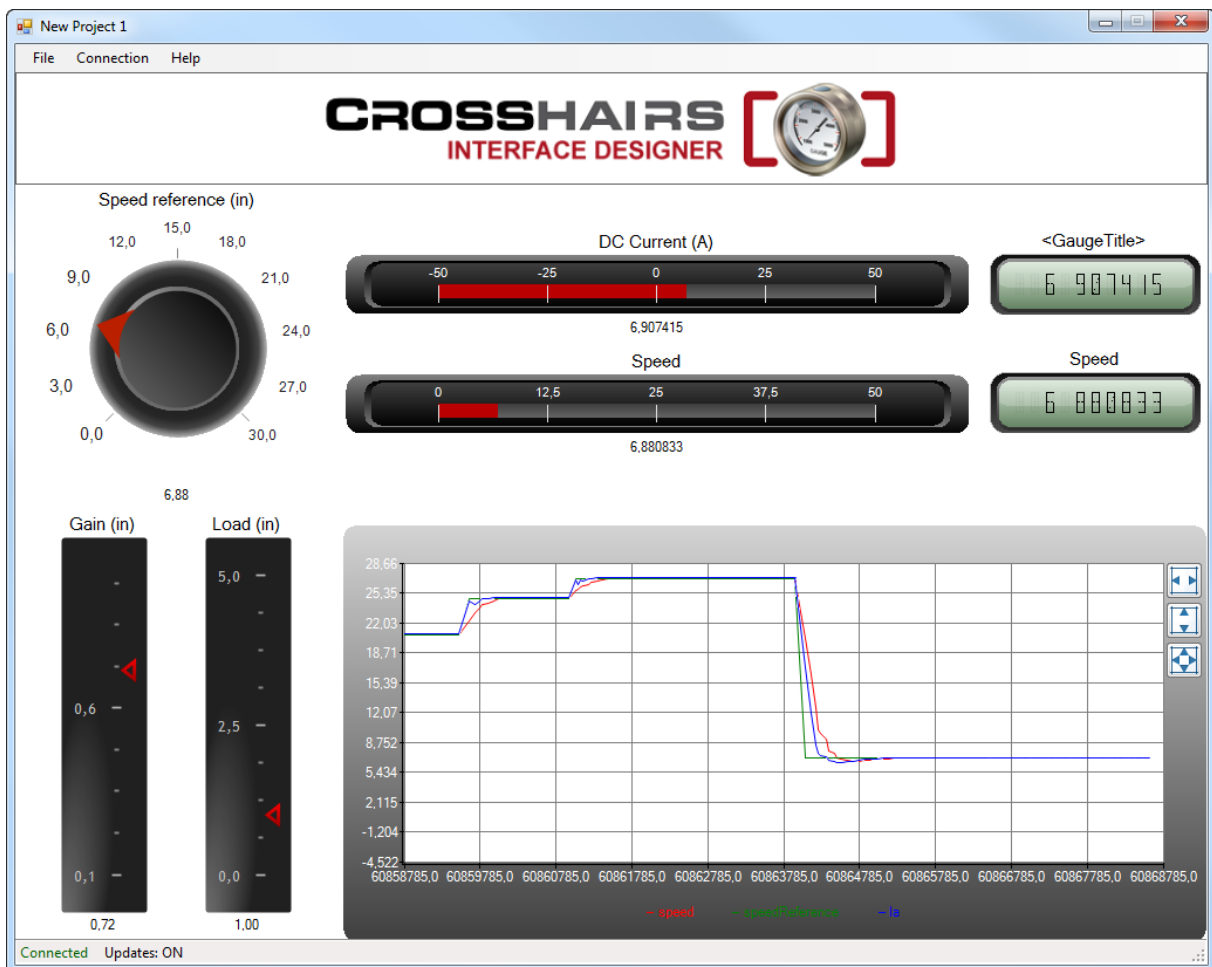
Commros_v2.2_small_dcmotor_piccolo2803x – This example is similar to the previous one but with a smaller version of Commros that doesn't support multiple variable reads in one call to the library

Crosshairs [interface designer]

The Crosshairs [interface designer] is available for trial download from the Crosshairs Embedded website. When using the Crosshairs [interface designer] you have a wealth of high quality graphical components available for your use in designing crisp, user friendly dashboards for your hardware.

For more information, please visit the following link: <http://crosshairseembedded.com/interface-designer>

For trial download, please register at the following link or contact sales@crosshairseembedded.com
<http://crosshairseembedded.com/free-trial>



FAQ

Q: Why can't I connect to the Piccolo controlCARD Experimenter's Kit through the Crosshairs[debugger] after I've programmed the example on the controlCARD?

A: Ensure that jumper J9 is populated if you are using Serial over the USB cable. If you are using Serial Pin-outs J9 must not be populated. Also ensure that the switches on the controlCARD is set up according to the communication method selected in [Configuration](#).

Q: I downloaded the controlSUITE software from Texas Instruments but the installer failed to run properly. Is there another way to get the controlSUITE resources installed?

A: Yes, you may also download a ZIP archive of the controlSUITE for manual install from this link: <http://www.ti.com/litv/zip/sprca85>

Customer Contact

If you have any questions regarding this document or any of the products and/or services described herein, please do not hesitate to contact us at support@crosshairseembedded.com

All other enquiries including license sales and enterprise solutions please contact us at sales@crosshairseembedded.com.

During business hours we may also be reached at the following phone numbers

- Sales +47 984 15 109
- General information +47 736 05 906

We are committed to total customer satisfaction and a member of our team will be in touch within one working day.

You may also find up-to-date news, company and product information on our website at www.crosshairseembedded.com

