AN-738
APPLICATION NOTE

ANALOG DEVICES
One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106 • Tel: 781/329-4700 • Fax: 781/326-8703 • www.analog.com

Using the AD7877 Touch Screen Controller and the Intel PXA250 Processor Under Windows CE.NET

by Paul Perrault and Susan Pratt

INTRODUCTION

The AN-738 Application Note details one way to connect an Analog Devices AD7877 touch screen controller to an Intel® PXA25x processor for use under a Windows® CE operating system. The user will be able to quickly develop a touch screen driver based on the example software accompanying this note. The exact code base will vary depending on the processor. Table 1 lists the software with a brief description.

DRIVER OVERVIEW

The driver supports the basic touch controller functionality of the AD7877. The driver makes use of several advanced features, including programmable acquisition delays, averaging, conversion sequencing, and contact resistance calculation. The driver requires an SPI interface on the host processor and a single GPIO pin.

The sample AD7877 touch driver is written for the Intel PXA25x CPU family (XScale®), and communicates with the CPU via the hardware SPI controller on the PXA25x. The Intel PXA25x processor family (XScale) has been very popular with Windows CE developers in recent years, and this driver is written specifically for that processor family.

However, any processor with a hardware SPI controller and an available GPIO could use this driver for touch screen interactions with minimal changes. The methods described within this application note have been tested to pass the Windows CE Test Kit (CETK) using a slightly modified PXA25x integrated development platform (IDP) and an AD7877 evaluation board. The software is intended to be used as a sample driver, or as a basis to build up a production-grade driver for your system, but is not intended for use “as is” in production systems.

The touch driver is interrupt driven, using the $\overline{\text{PENIRQ}}$ output to signal the processor when the screen is touched. Subsequent samples are scheduled using the OS timer. The driver queries the $\overline{\text{PENIRQ}}$ output at each sample point to determine if the screen remains touched.

Table 1. Files Included and Referenced in the AN-738 Application Note

File Name	Description
Tchpdd.cpp	The PDD (platform dependent driver) layer of the touch screen driver. The MDD (model device driver) layer of the driver is written by Microsoft. Together, these two files form a driver for Windows CE.
Pxa255.h	This is a header file to describe the register definitions of the PXA255 processor. This also defines the bit definitions for the synchronous serial port (SSP) on this processor so that the SPI interface works with the AD7877.
ad7877.h	This is a header file to deal with the registers inside the AD7877.
sources	A sources file is a text file that sets the macro definitions for the source code in a subdirectory. Build.exe uses these macro definitions to determine how to compile and link the source code.
makefile	In an application development environment, such as Microsoft® Platform Builder, a makefile contains all commands, macro definitions, and options needed. In this application, the makefile simply references the standard Windows CE makefile.

Data conversions utilize the built-in channel sequencing support in slave mode. A conversion is initiated at each sample point so that it will be complete by the next sample point. Thus, at each sample point, the driver reads the results of the conversion initiated at the prior sample point. This mechanism minimizes time spent in the driver, eliminates the need to use GPIO resources for the $\overline{\text{DAV}}$ status signal, and requires only that the sample rate guarantees enough time for whatever acquisition delays, averaging, and channel sequencing are in effect.

DRIVER CONFIGURATIONS

The touch driver requires interrupt support from the OAL (OEM adaptation layer). The OAL must return a valid sysintr number when $\overline{\text{PENIRQ}}$ asserts. This sysintr number must be known to the touch driver at driver initialization. The sysintr value is currently specified via a #define configuration option in the driver and defaults to 0x00000019. This value must match the value returned by the OAL. Implementation of the OAL requirements is beyond the scope of this document and will vary from code base to code base. It is sufficient to note that the OAL impacts will be limited to the pin operation of the GPIOs on the PXA25x and the interrupt support. Table 2 shows sample GPIO register settings to enable the AD7877 to be used with the PXA25x.

Table 2. Sample GPIO Register Settings of PXA255 for Use with the AD7877

Intel PXA255 Register	Bits	Setting	Description
SSCR0	SSE [7]	0	Disable SSP so that SSP changes can happen
SSCR0	DSS [3:0]	0b1111	Set the data size to 16 bits
SSCR0	FRF [5:4]	0b00	Set the frame format to Motorola's SPI
SSCR0	SCR [15:8]	0b0000	Set the SSP clock rate to 1.8 MHz
SSCR0	ECS [6]	0	Use the on-chip clock for SSPCLK
SSCR1	RIE [0]	0	Disable receive FIFO interrupt
SSCR1	TIE [1]	0	Disable transmit FIFO interrupt
SSCR1	LBM [2]	0	Disable loopback mode
SSCR1	SPO [3]	0	The SPI clock idles low
SSCR1	SPH [4]	0	SPI clock polarity
GPSR0	PD24	1	Set GPIO24 (SSPSFRM) output high
GPDR0	PD26	0	Set GPIO26 (SSPRXD) pin direction (input)
GPDR0	PD25	1	Set GPIO25 (SSPTXD) pin direction (output)
GPDR0	PD24	1	Set GPIO24 (SSPSFRM) pin direction (output)
GPDR0	PD23	1	Set GPIO23 (SSPSCLK) pin direction (output)
GAFR0_U	AF23 [15:14]	0b10	Set GPIO23 to be SSP CLK
GAFR0_U	AF24 [17:16]	0b10	Set GPIO24 to be SSP SFRM
GAFR0_U	AF25 [19:18]	0b10	Set GPIO25 to be SSPTXD
GAFR0_U	AF26 [21:20]	0b01	Set GPIO26 to be SSP RXD
SSCR0	SSE [7]	1	Enable SSP

Other configuration options in the driver include a macro to read the input from $\overline{\text{PENIRQ}}$ within the driver (to determine if the screen is touched), STOPACQ active level, sample rate configuration, and touch pressure/contact resistance define statements. If the particular application has an input device that has a larger area than a stylus, then the touch pressure feature is very useful, and is a good opportunity for the user to add value to their platform. This type of enhancement has been left to the user to implement on an as-needed basis. Some of these options are detailed in the code comments of tchpdd.cpp.

CONNECTING THE PXA25X TO THE AD7877

The AD7877 uses the SPI serial protocol, and the SSP (synchronous serial protocol) of the PXA25x must be connected for implementation. A single GPIO pin is also required to implement the driver correctly, as this pin connects to the $\overline{\text{PENIRQ}}$ input on the part. Neither pull-up nor pull-down resistors are required to make these connections since these resistors are included on-chip in the AD7877.

Table 3. Hardware connections from the PXA25x to the AD7877

Intel PXA255	Analog Devices AD7877
GPIO26 (SSPRXD) ball A9	DOUT Pin 27
GPIO23 (SSPSCLK) ball F9	DCLK Pin 26
GPIO25 (SSPTXD) ball D9	DIN Pin 19
GPIO24 (SSPSFRM) ball E9	$\overline{\text{CS}}$ Pin 18
GPIO Pin 5 (used for interrupt) ball J11	$\overline{\text{PENIRQ}}$ Pin 17

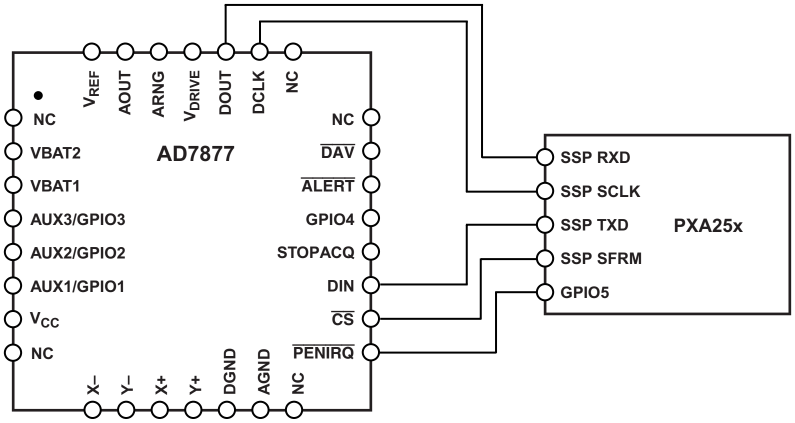

Figure 1. Connection Diagram.

REFERENCES

Intel PXA255 Applications Processors Developer's Manual:
ftp://download.intel.com/design/pca/applicationsprocessors/manuals/27869302.pdf

Specifically, section 8-1 (Synchronous Serial Port Controller), section 4-1 (GPIO Registers), and section 4-2 (Interrupt Controller) are useful reading.

Intel PXA255 Applications Processor Electrical, Mechanical, Thermal Specifications:
http://www.intel.com/design/pca/applicationsprocessors/manuals/27878002.pdf

Analog Devices AD7877 Touch Screen Digitizer Data Sheet:
www.analog.com/UploadedFiles/Data_Sheets/43218542AD7877_prf.pdf

Microsoft Windows CE.NET development site:
http://msdn.microsoft.com/embedded/ce.net/

Contact susan.pratt@analog.com with any questions or comments on the AD7877 hardware and software.