Introduction
In this experiment, you will familiarize yourself with the Ubicom SX28 microcontroller and the SX-Key Development system.

Document Conventions
This document makes use of four symbols that dictate your activities:

✓ Instruction for the student to follow.
◆ Required lab notes.
✓ Demonstrate to lab instructor/TA.
◆ Lab report item.

Pre-Lab:
✓ Use the links in the Resources section at the end of this document to obtain:
  (1) SX-Key.exe, (2) The SX-Key/Blitz Development System Manual v1.1.
◆ Enter all code examples into SX-Key.exe and save on a floppy (bring to lab).
✓ Look up and attempt to understand all assembly mnemonics and directives that you are not already familiar with. Using Adobe Acrobat Reader’s search function will assist in quickly locating terms in the manual. The Manual’s Appendix B: SX Instruction Set contains explanations of all assembly instructions for the SX microcontroller used with the SX-Key development system.
Applications
The SX28 microcontroller and SX-Key development system has many features common to microcontroller development environments:

- A programming tool – the SX-Key
- In-circuit debugging tools – the SX-Key in conjunction with the SX-Key software is used to view register values, I/O pin values, the clock, flag bits, and events.

The pushbutton and LED circuits used in this laboratory are common user interface components found in many electronic products such as alarm systems, sprinkler timers, cell phones, and PDAs.

The concept of establishing a time base using the interrupt service routine extends far beyond the user interface. Once you have a time base, the microcontroller can be programmed to control multiple processes such as serial communication, sampling the user interface, motor control with PWM, communication with peripherals such as sensors, memory, and special purpose integrated circuits.

Parts

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Part Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>470 Ω resistor (Yellow, Violet, Brown)</td>
</tr>
<tr>
<td>1</td>
<td>LED (Light Emitting Diode)</td>
</tr>
<tr>
<td>1</td>
<td>SX-Key</td>
</tr>
<tr>
<td>1</td>
<td>SX Tech Board</td>
</tr>
<tr>
<td>1</td>
<td>7.5 V, 1000 mA AC adaptor</td>
</tr>
</tbody>
</table>

Activity #1: SX-Key Hardware, Debugging Environment, Test Circuit, and First Program

Figure 1.1 shows the hardware setup for the SX-Tech Tool Kit. Plug the SX-key into the 4-pin header on the SX Tech Board exactly as shown in the figure. The other end of the SX-Key should be connected to a serial cable. You will need to plug the other end of the serial cable into an available serial port on your PC and the 7.5 V, 1000 mA supply connected to the SX-Tech board must also be plugged into a wall outlet.
Connect 7.5 V, 1000 mA AC Adaptor here.

Connect 9-pin serial cable here.

Figure 1.1: Setting up the SX-Tech Tool Kit.

Build the circuit shown in Figure 1.2. This is the LED circuit pictured in Figure 1.1.

Figure 1.2: LED circuit.

- If you have not already done so, download and run SX-Key.exe. Instructions for this can be found in the Resources section at the end of this document.

- Load Program Listing 1.1.
; Program Listing 1.1
; ---- Assembler Directives -----------------------------------------------
device SX28L,oscxt2 ; SX28L, external feedback for
                   ; 4 MHz res.
device stackx_optionX,turbo ; Extend stack & option
                           ; registers & turbo.
freq 4_000_000 ; Debug frequency.
reset start ; Go to 'Start' label
            ; on reset.

; ---- Watch Directives ----------------------------------------------------
watch timer_1,8,udec ; Watch timers.
watch timer_2,8,udec
watch rb,8,ubin ; Watch I/O port B.

; ---- I/O Pin Definitions --------------------------------------------------
LED_line = RB.6 ; Alias name for port B pin 6.

; ---- RAM Declarations -----------------------------------------------------
org 8 ; Global Registers.
timer_1 ds 1 ; Two 8-bit registers for
timer_2 ds 1 ; 16-bit counter.

; ---- Interrupt Service Routine ---------------------------------------------
org 0 ; Program memory origin.

; ---- Boot Routine-----------------------------------------------------------
start mov !option,#%10001000 ; RTCC enabled, reg 0 = w
     mov !rb,#%10111111 ; RC.7 => output.
     clr timer_1 ; Clear timers
clr timer_2

; ---- Main Routine-----------------------------------------------------------
main inc timer_1 ; Increment timer low byte.
snz ; Increment timer high byte
     ; on rollover.
inc timer_2
jnz main ; Go to main if timer_2 != 0.
not LED_line ; Invert RB.6.
jmp main ; Infinite loop.

Select Debug from the SX-Key editor’s Run menu.

Figure 1.3 (next page) shows the SX-Key Editor’s in circuit debugging environment, which is comprised of four windows:

1) Registers – Shows:
   a. all SX28 chip’s registers
   b. Program addresses, machine codes, and assembly commands.
2) Debug – Buttons to control debugging and bring windows to the foreground.
3) Watch – View register values specified in assembly code by the “watch” directive.
4) Code – Shows source code with a blue bar to denote the command about to be executed and a red bar showing a breakpoint.

Click the Run button in the Debug window
✓ Verify that the LED blinks on and off at about 4 Hz. Trouble-scout the circuit if needed.
✓ Click the Stop button and make a note of the values in the watch window.
✓ Compare these values to the RB register and global registers 08 and 09.
✓ Click the line of code with the not LED line command in the Code window. This should cause the line to be highlighted in red signifying a breakpoint.
✓ Click the Poll button.

Note that the program is running, but the debug information in the registers window is updated every time the program passes the breakpoint. The value of the RB register should be updated in the Watch and Registers window.

Can you determine from this the relationship between RB and the LED?
Click Stop to halt the polling process.
Click Run, and the program will run full speed until it gets to the breakpoint.
Click Step 10 or 15 times to step through the program one assembly command at a time.
Click the number to the right of the timer_1 label in the Watch window.
Enter a value of 253.
Click the number to the right of the timer_2 label in the Watch window.
Enter a value of 255.
Record each assembly command that was executed and how the register values change for the next 20 clicks of the step button.
Explain how the value of the LED was toggled by incrementing the timer in terms of the assembly code that was executed.
Click the Walk button.
Record what happens.
Click the Stop button.
Click the jmp main line to move the breakpoint to that line.
Use the Watch window to set the value of the timer_1 and timer_2 to 250 and 255 respectively.
Click the walk button again.
Explain what just occurred.
Clear the breakpoint by clicking the red, highlighted line `jmp main` command in the code window.
Click run.
Click pole a few times.
Explain the differences in the debugging environment’s behavior:
- When the Poll button is selected when a breakpoint is set.
- When the Poll button is selected when there is no breakpoint and the program is running.
Click Stop.
Select your own breakpoint in the Code window.
Click pole.
Click reset. What effect did this have on the active command line?
What happens with the first few steps?
Devise a sequence of setting breakpoints and use of poll, walk, and step to effectively demonstrated and explain the theory of operation of a 16-bit counter to your instructor.
Click Quit to exit the debug session.

Resources
Documentation and software for the SX-Key and SX chips is available for free download from www.parallax.com. The web address that takes you directly to the SX Tech Downloads page is: http://www.parallax.com/sx/downloads.asp

Below are the links you should download to complete these products:
- Self-extracting archive of the SX-Key v1.30
  340 KB
  Software (for Rev. E/F SX-Keys) and addendum information. Supports the 18/28/48/52-pin chips in one executable.

- SX-Key and SX-Blitz Manual v. 1.1
  4.8 MB

There are also free, downloadable tutorials for the SX-Tech toolkit:
- Introduction to Assembly Programming with the SX
  1.3 MB
  Version 1.2

- I/O Control with the SX Microcontroller
  1.2 MB