SYSTEM DESIGN SPECIFICATIONS
LW9 SYSTEM

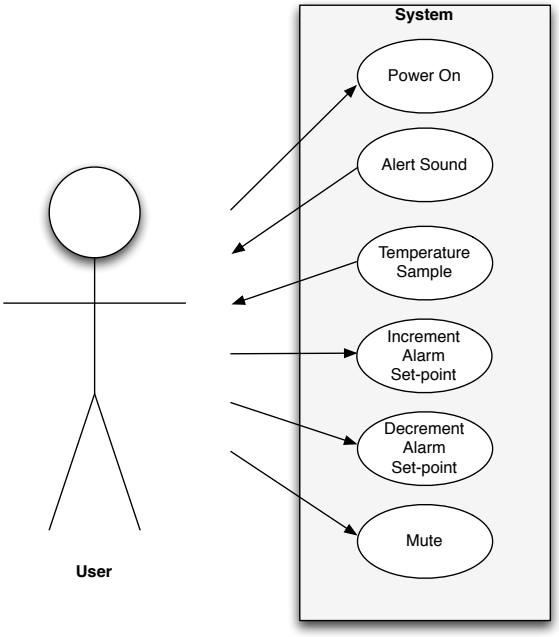
SYSTEM DESCRIPTION

The system is a temperature monitor. The system initializes at power-on reset and monitors temperature. An alert sound is generated when the temperature exceeds a programmed set-point. The user can mute the alert sound using a pushbutton. The user can adjust the set-point using pushbuttons. A fan is enabled whenever the temperature exceeds the set-point. Status information is displayed on an LCD panel.

REQUIREMENTS

1. The system must monitor temperature using a temperature sensor.
2. The system must enable a fan when temperature exceeds a set-point.
3. The system must generate alert sounds when temperature exceeds a set-point.
4. The system must respond to mute sound events.
5. The system must respond to increment and decrement set-point events.
6. The system must implement an 8 second periodic LCD display timeline.
7. The system must sample, store, and display temperature data.
8. The system must display a system operating heartbeat.
9. The system must operate in standard room temperatures.
10. The system must operate from standard U.S. line power or USB supplied power.
11. The system does not need power-failure or system failure recovery.

USE CASE DIAGRAM



USE CASE EVENTS

1. Power On

   A. The user applies power or resets the system.
   B. The system initializes hardware resources.
   C. The system displays a startup LCD screen for 2 seconds.
   D. The system enters an 8 second infinite loop system timeline.

2. Alert Sound

   A. The system identifies a temperature above the set-point.
   B. The system turns on an alert sound if the alarm is not muted.
   C. The system turns off the alert sound if temperature is below the set-point.
   D. The system turns on or off an LCD panel alert icon based on alarm status.

3. Temperature Sample

   A. The system identifies a 250 ms real-time event.
   B. If the system timeline is between 0s and 4s then the temperature sample is stored and the LCD panel is updated.
   C. If the system timeline is between 4s and 6s then the system does not store the temperature sample and instead displays a bar graph of the previously stored temperature data on the LCD panel. The top row represents samples above the set-point and the bottom row represents samples below the set-point.
   D. If the system timeline is between 6s and 8s then the system does not store the temperature sample and instead displays statistical information about the stored temperature data. The information displayed is average temperature, maximum temperature in the sample set, and minimum temperature in the sample set.
   E. If the system timeline is at 8s then the sample set maximum is compared against the system lifetime maximum stored at EEPROM location 0. If the sample set maximum exceeds the system lifetime maximum then the EEPROM location is updated to reflect the new maximum.

5. Increment Alarm Set-point

   A. The system identifies an increment alarm set-point event.
   B. The system increments the alarm set-point by 2°F.

6. Decrement Alarm Set-point

   A. The system identifies a decrement alarm set-point event.
   B. The system decrements the alarm set-point by 2°F.

7. Mute

   A. The system identifies a mute alarm event.
   B. The system toggles the alarm on or off.
   C. The system updates the LCD panel alarm status icon.

SPECIFICATION OF SYSTEM INPUTS AND OUTPUTS

1. System Inputs

   A. The system uses an LM34 temperature sensor.

      i. The LM34 is powered by the Atmega32 power supply rails.
      ii. The LM34 output is valid for 0°F to 100°F.
      iii. The LM34 output is connected directly to the Atmega32 ADC.

   B. The system uses three de-bounced pushbuttons for user input.

      i. Hardware debouncing prevents switch oscillation. Consult the supporting diagrams section at the end of this document.
      ii. External interrupt pin INT2 is used for alarm mute.
      iii. External interrupt pin INT1 is used for increment alarm set-point.
      iv. External interrupt pin INT0 is used for decrement alarm set-point.

2. System Outputs

   A. The system controls a sonic alert using Atmega32 port pin PD7.
   B. The system controls a fan using Atmega32 port pin PD6.
   C. The system controls a serial LCD panel using Atmega32 USART TX.

SYSTEM FUNCTIONAL SPECIFICATION

The system uses interrupt-driven, real-time software written in C for the Atmega32.

1. The system initializes at power-on reset.

A. The temperature set-point variable is initialized to 84 degrees Fahrenheit.
B. The ADC is initialized for left-adjusted result, ADC interrupt enabled, non-continuous conversion.
C. A one quarter second timer interrupt is initialized.
D. The LCD panel is initialized.
E. External interrupt pins are initialized.
F. CPU interrupt processing is enabled.
G. A startup screen with a welcome message and the system lifetime maximum temperature is displayed for 2 seconds.
H. The main function enters an infinite loop that implements a periodic 8 second system timeline.
I. The infinite loop monitors global volatile variables set by interrupts.
J. The infinite loop controls the alarm, fan, and LCD based on the global volatile variables.

2. The system uses a one-quarter second timer interrupt service routine.

A. The ADC sample is started.
B. Global volatile variables are set if needed.
C. The interrupt service routine exits.

3. The system uses an ADC interrupt service routine to monitor temperature.

A. The left-adjusted result is written to a global volatile variable.
B. The left-adjusted result is compared to the alarm set-point.
C. The interrupt service routine exits.

5. The system uses interrupt service routines to handle pushbutton events.

A. A global volatile variable is set to the appropriate pushbutton value.
B. The interrupt service routine exits.

6. The system uses a serial LCD panel to report information to users.

A. The system reports temperature information on the LCD panel in °F.
B. The system reports power-on as a periodic one-second heartbeat icon.
C. The system reports the alarm condition using an appropriate icon.
D. The system reports the temperature set-point on the LCD in °F.
E. The system reports mute status with either text or icon.
F. The system displays a bar graph using appropriate symbols (▮, - , _).
G. Heartbeat pulses and status icons are disabled during bar graph display.

LCD USER INTERFACE EXAMPLES

C	E	2	8	1	0				T	E	M	P	M	O	N
S	Y	S	M	A	X	=	9	5	°	F				♫	♥

T		=	8	0	°	F		M	A	X	=	8	8	°	F
S	P	=	8	4	°	F								♫	♥

A	V	E	=	8	0	°	F		H	I	=	8	8	°	F
L	O	=	7	4	°	F								♫	♥

DEVICE DRIVER REQUIREMENTS

1. The firmware must include ADC, USART, and LCD device drivers (.h, .c files)
2. The following USART device driver functions are required.

    A. usart2810_init(void) // default: 9600, 1 start, no stop, no parity
    B. usart2810_baud(unsigned char highByte, unsigned chart lowByte)
    C. usart_send_data(unsigned char theData)

3. The following LCD device driver functions are required.

    A. lcd2810_init(void) // default: LCD on, cursor off, no blink
    B. lcd2810_gotoxy(unsigned char row, unsigned char column)
    C. lcd2810_init_heartbeat_icon(void)
    D. lcd2810_init_sound_icon(void)
    E. lcd2810_send_data(unsigned char theData)

4. The following ADC device driver functions are required.

A. initializers
B. setters for ADMUX values, ADCSRA values
C. getters for ADCH and ADCL
D. adc2810_start_convert(void)

5. The following global interrupt configuration device driver functions are required.

A. initializers
B. setters for enable and disable interrupts
C. setters for trigger edge

6. The following EEPROM device driver functions are required.

A. unsigned char eeprom2810_read(void)
B. void eeprom2810_write(unsigned char databyte)

SUPPORTING INFORMATION

Review the datasheets for the Atmega32, LM34, and LCD panel. The datasheets are posted on the course website. **Consult** the ADC, EEPROM, and USART tech notes if needed.

SUPPORTING DIAGRAMS

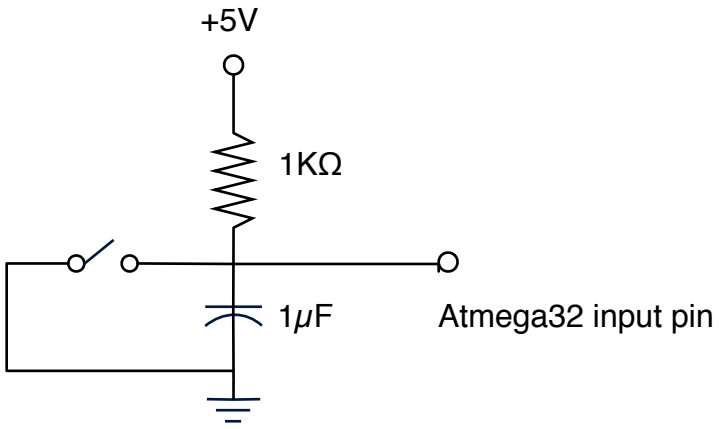


Figure 1: Hardware-based Switch Debouncing

Pushbutton switches act like mechanical springs. This causes the switch output to oscillate until the spring action settles. There are many techniques to compensate for switch oscillation. Techniques in software de-bouncing and hardware de-bouncing exist.

One hardware de-bouncing technique is the addition of an RC time constant to the switch circuit. The RC time constant prevents the bounce because the capacitor cannot change voltage rapidly. Thus, the switch output becomes a more gradual capacitive charge-discharge shaped waveform. The RC time constant determines how rapidly the charge-discharge occurs. Standard values of 1KΩ and 1μF work well in most applications. On microcontrollers with built-in port pullup resistors, the pullup resistor can be enabled in place of the 1KΩ external resistor. Note that in any case, the capacitance value may need to be adjusted to guarantee the timer constant exceeds the switch bounce time. Use 1μF as a starting value and evaluate the system for performance. Increase the capacitance value if bouncing behavior is seen.

DELIVERABLES

1. **Demonstrate** the system to Dr. Meier by Friday of week 10.
2. **Email** your well-commented source code to Dr. Meier immediately after your demonstration (meier@msoe.edu). You may zip the files for submission. **Use** the subject header **CE2810 LW9** when submitting your code.