

Lab 1: Aircraft Fault Display

1.0 Inputs

Eight (8) SPST switches are to be used as inputs. When each switch is closed, no fault is indicated. 5 volt CMOS logic levels are to be employed.

1.0.1 Input Enumeration

The faults are enumerated (in decreasing order of severity) as follows:

8: Right Engine	7: Left Engine
6: Hydraulic Pressure Low	5: Low Fuel
4: Gear Unlocked	3: Slats Disagree
2: Cabin Pressure Low	1: Landing Lights Fault

Code (0) is used to indicate no fault. The inputs must be designed such that if a harness wire fails open, the fault condition is indicated.

1.1 Output

The output is to be a single seven-segment LED.

1.2 Processing and Data Flow

1.2.1 Power On Self Test

When the system is powered up, it will count from 1 to 8 to test the LED display. It will then revert to fault detection mode.

1.2.2 Fault Detection Mode

In this mode the system will continually scan the fault input lines:

- a) When no fault is present, the numeral "0" will be continuously displayed.
- b) When any faults are present, the system will display the faults in sequence, starting with the most severe. Each fault will be displayed according to the numeric codes in section 1.0. Each fault should be displayed for 1 s, followed by a blank display for 0.5 s, followed by the next code (if any).
- c) When faults are corrected, they must be automatically removed from the display sequence.

Lab 2 LCD Display

1.0 Inputs

Eight (8) SPST switches are to be used as inputs. When each switch is closed, no fault is indicated. 5 volt CMOS logic levels are to be employed.

1.0.1 Input Enumeration

The faults are enumerated (in decreasing order of severity) as follows:

8: Right Engine	7: Left Engine
6: Hydraulic Pressure Low	5: Low Fuel
4: Gear Unlocked	3: Slats Disagree
2: Cabin Pressure Low	1: Landing Lights Fault

Code (0) is used to indicate no fault. The inputs must be designed such that if a harness wire fails open, the fault condition is indicated.

1.1 Output

The output is to be in two forms:

- a) A generic 16 character by two-line LCD.
- b) Two LEDs, green for no fault and red for failure. These LEDs are to be mutually exclusive.

1.2 Processing and Data Flow

1.2.1 Power On Self Test

When the system is powered up, it will count display "Fault Warning Self-Test" and alternately flash the green and red LEDs for two seconds to test them.

1.2.2 Fault Detection Mode

In this mode the system will continually scan the fault input lines:

- a) When no fault is present, the green LED will be lit continuously and the text "All Systems Normal" is to be displayed.
- b) When any faults are present, the system will display the faults in sequence, starting with the most severe. The red LED will be illuminated and the green LED will be extinguished. Each fault will be displayed according to the verbose codes in section 1.0. The text "Failure," followed by a flashing verbose description, will be displayed for each fault. Each fault should be displayed for 1 s, followed by a blank display for 0.5 s, followed by the next fault (if any).
- c) When faults are corrected, they must be automatically removed from the display sequence. When all faults are corrected, the system should return to state 1.2.2(a).

Lab 3 Keyboard Interfacing

Objectives:

- Interface the keypad to the microcontroller using your own choice of port pins (you will need five input port pins if you use the 74HC922 keyboard scan IC). Please refer to the data sheet for this part.
- Create general-purpose I/O routines for getting ASCII and numeric data from the keyboard so that a user interface can be created.
- Define the behavioral specifications for a simple application demonstrating the functionality of the keyboard and LCD.

Instructions:

In this experiment you will be defining what you would like the microcontroller to do. This can be almost anything -- even a simple game or simulation is fine. The objective is to get high-level routines for the keyboard and LCD display working together. These high-level routines will form the basis for more complex applications with the microcontroller.

Note that you must pay careful attention when writing these routines to ensure that they are robust. Watch C pointers, in particular, and don't allow them to write past the end of arrays, or you'll be pulling out your hair debugging. Use boundary checking for user input so that your program doesn't lock up or fail when the user makes incorrect entries.

You may find it helpful to implement the following C functions, or at least something like them.

```
// global input buffer to be held in SRAM
char input_buffer[32];

// Scan the keyboard, returning an 74HC922 scan code (or -1 if no key pressed)
int scan_key();
// Scan the keyboard, returning (-1) if no new key pressed, or the new 74HC922 key code
char getkey();
// Scan the keyboard; if a new key is pressed, return the ASCII key code, or 0 (NUL) if
  none.
char getkey_ASCII();
// Scan the keyboard; if no keys pressed, wait until is one is pressed, then return the
  ASCII
  // keycode.
char waitkey_ASCII();
// Input a string from the user into the INPUT_BUFFER while displaying a szPROMPT
// on the LCD.
int inputStringWithPrompt( char flash* szPrompt,int nMaxLen);
// Input an integer from the user, while displaying szPROMPT
int inputIntegerWithPrompt(char flash* szPrompt);
```

Remember that developing working input routines is the goal of this lab. This is not always an easy task, but it is a very important one. Remember to write and test one short section of code at a time. Allow yourself plenty of time to understand and complete this lab!

Lab 4 Analog to Digital Conversion (Temperature Lab)

1.0 Inputs

One LM34 temperature measurement IC is to be used to read the ambient temperature. The ambient temperature will not be allowed to exceed 240 degrees F.

1.1 Output

The output is to be in two forms:

- a) A generic 16 character by two-line LCD.
- b) A 12 volt DC fan. When the temperature is too high, the fan will be activated.

1.2 Processing and Data Flow

1.2.1 Temperature Display

The system will continually display to the nearest 0.1 degree, the sensor temperature in both Fahrenheit and Celsius. The top line of the display is to be used for this purpose. The display will be updated in a flicker-free manner from two to four times per second.

The temperature is always displayed regardless of the fault condition (or lack thereof).

1.2.2 Fault Indication

When the ambient temperature is below 78.0 F, the display will show "Temperature OK" on the bottom line.

When the temperature is 78.0 F or higher, the display will show "High temp" on the bottom line. The DC fan motor will be turned on, and 3 degrees F of hysteresis will be provided such that the fan motor will remain on until the ambient temperature falls below 75.0 F.

1.3 Option 1

The user may optionally be allowed to program the temperature limit and hysteresis. The programming mode is entered by pressing or holding a specified key during normal operation of the instrument.

The user interface of the programming mode is specified by the student.

Lab 5 UART Communications

Lab 5 is an extension of the ADC lab (Lab 4), so all I/O specifications from Lab 4 apply to this design.

This design will allow a personal computer to retrieve the temperature data. No data will be required from the PC in order to initiate the communications -- the data flow will be purely simplex.

The revised output specification is as follows:

1.1 Output

The output is to be in three forms:

- a) A generic 16 character by two-line LCD.
- b) A 12 volt DC fan. When the temperature is too high, the fan will be activated.
- c) An RS232D serial interface supporting the RxData line for direct transmission of data to a personal computer. The data will be sent using the parameters: 9600 bps, 8,N,1.

1.2 Input

In addition to the input types supported in Lab 4, the following input will be supported:

- a) RS232D TxData line for reception of command data from the attached personal computer.
- b) The RS232 RTS, RLSD, and DCE READY signals will be hard-wired to the "on" or "asserted" condition regardless of the communications state.

The display specification is revised with the addition of the following section:

1.3 Serial Data Echo

About once per second the system will output a status indication to the personal computer. The status indication will include the fan on/off status, the current temperature to the nearest 0.1 degree F, and the upper temperature limit.

When the temperature is normal and the fan is off, the status message "Temperature Normal" will be included in the status message.

When the temperature is above the limit, the status message "High Temperature" will be included. If the temperature has been above the limit and has fallen below the limit (but the cooling fan is still on, due to hysteresis), the status message "Cooling down" may also be appended.

1.4 Remote Command Capability

The system will continually check the RxData line for incoming command information from the PC.

It is recommended that the RETURN character ('\r', 0xd) be used to activate a menu system that will allow the user to remotely configure the temperature limit and hysteresis from the personal computer. Any other character should be ignored to prevent menu activation by spurious noise or characters.

All inputs to the menu system must be bounds-checked, and if no input is received within approximately 20 seconds, the menu should automatically exit and allow the system to continue running again.