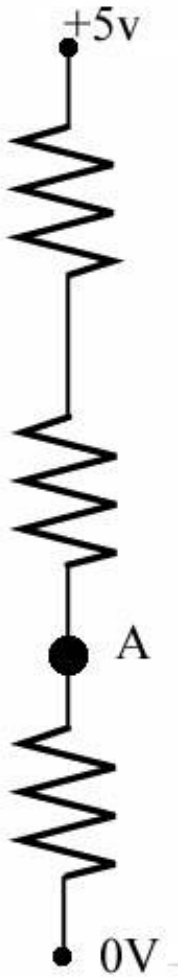


Electronics-Robotics Mid-Year Examination The Fire Alarm Project

1.

Note: Because you are working in groups and under a deadline, it is important for you to be in class and on time each day. You put the group at a disadvantage if you are late or absent.



Start with three resistors wired in series as indicated in the schematic diagram to the left. The top resistor and the center resistor **MUST** be different values.

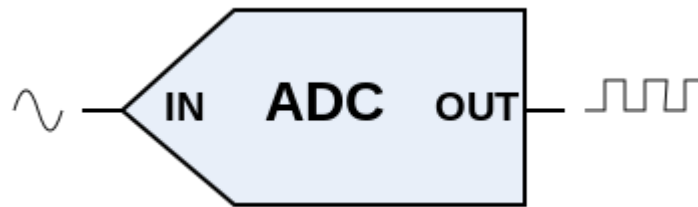
You also will need to wire up a red LED and a green LED. Each of those should have the appropriate resistor in series. You will also need a Piezo Speaker. The speaker does not require a resistor.

Eventually you will also need to have two buttons. One will "Short out" the top resistor. Another will "Short out" the center resistor. The bottom resistor will never be shorted out.

It would be smart to put all those schematics on one piece of paper to which you can refer when you are writing your program, and which you can submit with the program.

2.

You will use an ADC to monitor the voltage at a test point. Variations in voltage will indicate whether the condition of the circuit is normal, trouble, or alarm.



Analog to digital conversion:

Analog-to-digital conversion is the process which is performed by an analog-to-digital converter (**ADC, A/D, A to D, A2D**). It converts a continuous physical quantity (voltage, current, location) to a digital number that represents the amplitude of the quantity.

The conversion always introduces a small amount of error. The ADC will usually “sample” the variable at small time intervals, produce a number representing its magnitude, and then take another sample.

An ADC is defined by its bandwidth (the range of frequencies it can measure) and its signal to noise ratio (how accurately it can measure a signal relative to the noise it introduces). The inverse operation (Digital to Analog) is performed by a digital-to-analog converter (DAC).

The Activity Board has a 12-bit 4-channel SPI ADC. That means it can accept a signal between 0 and +5 volts and convert it to a 12 bit digital number. (See the TI Spec sheet for the ADC124S021.)

Using the ADC on the Activity Board requires several steps:

- (1) Include the library of functions already developed for the ADC
`#include "adcDCpropab.h"`
- (2) Set certain variables
`adc_init(21, 20, 19, 18); // CS=21, SCL=20, DO=19, DI=18`
- (3) declare a floating variable to hold the reading
`float voltage;`
- (4) read the ADC and store the reading in your variable
`voltage = adc_volts(3); // you are reading channel 3`

3.

Parallax developed a library which makes it easy to use the ADC on the Activity Board. Here is some C code which works:

```
#include "adcDCpropab.h" // Include adcDCpropAB
#include "simpletools.h" // Include simpletools
int main() // Main function
{
  adc_init(21, 20, 19, 18); // CS=21, SCL=20, DO=19, DI=18 (set up the a-d converter)
  float voltage; // declare voltage as a float
  while(1) // Loop repeats indefinitely
  {
    voltage = adc_volts(3); // Check A/D 3
    putChar(HOME); // move cursor to top left
    print("Voltage = %.2f V %c\n",voltage, CLREOL); // Display voltage and clear the rest of the
line
    pause(1000); // Wait 1 second before looping
  } //end while
} //end main function
```

At this point if you have not yet done so, you should construct the series resistor circuit shown on the first page and determine the voltage reading when the condition is normal, when a wire is broken (trouble), when one button is pushed, and when the other button is pushed. This is an important step! You will need these numbers in order to calibrate the alarm.

“Normal” Voltage (Circuit is set up and there are no buttons pushed) = _____

“Trouble” Voltage (The wire connecting the top resistor to 5V is removed) = _____

“Button 1” Voltage (A wire or button shorts out the top resistor) = _____

“Button 2” Voltage (A wire or button shorts out the second resistor) = _____

The instructor will lead you through the process of converting those voltages to the proper values to be entered into the “decision chain” part of your program.

4.

```
/*
This is working code for the Fire Alarm project   RWH   12/13/14
schematic: +5V --- 470 ohm--1K ohm --A-- 1k Ohm -- ground
Test lead from point A to ADC term 3
shorting 470 causes alarm x and shorting 1k causes alarm y, breaking circuit causes trouble
Alarm or trouble halts system.
***Note: You must calibrate the decision chain. Also flashing lights and beeping must be added.
*/
#include "simpletools.h"
#include "adcDCpropab.h" // Include adc library
int main() // Main function
{
adc_init(21, 20, 19, 18); // initialize the ADC ChipSelect=21, SerialClock=20, DO=19, DI=18
float voltage; // a floating variable is needed
while(1) // Loop repeats indefinitely
{
voltage = adc_volts(3); // Check A/D ch 3
//the following two lines are for testing --should be eliminated from final version
putChar(HOME); // move cursor to top left
print("Voltage = %.2f V %c\n",voltage, CLREOL); // Display voltage

// beginning of the decision chain
if(voltage < 1.02)
    {print("Trouble 1 \n", CLREOL); break;} //break drops out of the loop
else if(voltage < 2.23)
    {print("System Normal %c\n", CLREOL); } // no break if system normal
else if(voltage < 2.97)
    {print("Floor x %c\n", CLREOL); break;}
else if(voltage < 3.45)
    {print("Floor y %c\n", CLREOL);break;}
//end of the decision chain

pause(1000); // Wait 1 s before repeating
} //end while
print ("System has stopped, \n you must restart the program");
} //end main
```