EE 4770 **Homework 3** **Due: 10 March 1997**

Problem 1: Design a circuit and interface routine to convert process variable $x \in [190\,\mathrm{K}, 320\,\mathrm{K}]$, temperature, to a floating-point number $H(x) = x/\,\mathrm{K}$ using an RTD with response $H_\mathrm{t}(x) = R_0(1 + \alpha_1 x + \alpha_2 x^2)$, where $R_0 = 100\,\Omega$, $\alpha_1 = 0.00398/\,^\circ\mathrm{C}$ and $\alpha_2 = -5.84 \times 10^{-7}/\,^\circ\mathrm{C}^2$. Precision should be $\pm 0.1\,\mathrm{K}$. To reduce the chance of clipping use only 90% of the ADC's dynamic range. That is, if a 5 V ADC is selected, its input should be within $[0.25\,\mathrm{V}, 4.75\,\mathrm{V}]$ over the given measurement range. Be sure to account for the RTD's nonlinear response and the Celsius units used in the model function.

Problem 2: Design a system to determine the location of a turtle confined to a square fenced-off area and write the location, in meters, to structure `Point turtle_loc` where

```
typedef struct {
  double x;  /* X-coordinate in meters. */
  double y;  /* Y-coordinate in meters. */
} Point;
```

The turtle is free to move around the area which is one meter square. (There is plenty of room, it's a small turtle.) The turtle has a 1 light-watt lamp mounted on its shell; the light radiates uniformly in all directions and is higher than the fence. The location is to be determined using photodiodes with response $H_{\mathrm{t}1}(x) = x\frac{50\,\mu\mathrm{A}}{\mathrm{mW/\,cm^2}}$. Assume that the photodiodes are sensitive only to the light directly radiated from the lamp. (That is, light from the turtle that is reflected off other surfaces will not affect the photodiodes, nor will light from other sources.) Use as few photodiodes as possible. Assume that components operate perfectly and that the turtle will do nothing to reduce the sensitivity of the light source or detectors.

The following geometry functions are available:

```
 Point Line_Intersection(Line line1, Line line2);
```

returns the point at which `line1` and `line2` intersect (if any), where

```
typedef struct {
 Point p1,p2;
} Line;
```

Function

```
 Point *Circle_Intersection(Point center1, double radius1,
                            Point center2, double radius2);
```

returns an array containing the zero, one, or two points where the circles intersect. You can assume the existence of other basic functions, for example to determine if a point is in a rectangle.

Note: this assignment will not be graded on syntax; if you're not sure how to use a function make a reasonable assumption and state the assumption on your solution. (Of course, you can also ask the instructor or TA.)