COMP9032 Project

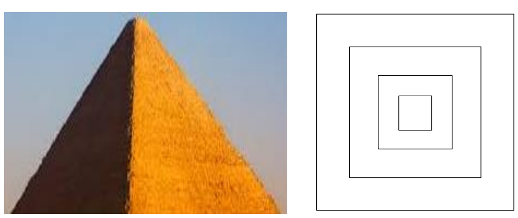
Description

Drone is an unmanned aerial vehicle, increasingly used in missions that are difficult and dangerous for humans.

Figure 1*

In this project, you will be working individually to develop a simulation system with the AVR Development board, emulating the control of a drone to search an accident scene in a mountainous area, as illustrated in Figure 1.

The mountains have no caves and can be described by a 2D array of surface points. For each x-y coordinate location, the related surface height, H(x,y), is given in the array. A simple example of such representation is shown in Figure 2, where a pyramid on a 7 meter by 7 meter area with the peak of 6 meters is (approximately) represented by a 7x7 array.



$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 & 2 & 2 & 0 \\ 0 & 2 & 4 & 4 & 4 & 2 & 0 \\ 0 & 2 & 4 & 6 & 4 & 2 & 0 \\ 0 & 2 & 4 & 4 & 4 & 2 & 0 \\ 0 & 2 & 2 & 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(a) (b) (c)

Figure 2: (a) Pyramid* (b) Top View (c) 2D Representation

For this project, you can, roughly based on Figure 1, generate your own mountains. The accident scene, if any, can be located on some surface point. It is assumed that x and y are in the range of 64 meters and that the highest peak and the lowest valley of the area are, respectively, 64 meters and 10 meters. The values in the 2D presentation can be precise to 1 meter.

The drone is initially grounded. Once launched, it flights to the area and searches for the target. Here, we assume the drone can see an object (e.g. the target) in a distance of 10 meters. Namely, the drone needs to flight high or low enough in order to see peaks and valleys.

The drone position can be represented by (x, y, z). At a location, the local peak around the drone can block its view to other side of the peak. For simplicity, it is assumed that during the search, drone can be in two states: 1) halt to inspect the local area, and 2) flight to a new location to continue search.

You need to develop an algorithm for the drone to search all spots in the area. Once the target is found or the whole area is searched, the drone will flight back with the search result.

Your design will have the below features:

- The accident location can be set after the simulation starts.
- The search can be interrupted by the user if the mission needs to be aborted early.
- The design offers the following input/output functions:

Input:

- Push Button:
  - Used to launch the search
    - When the button is pressed, the drone will flight to the area and start search.
  - Used to abort the search
- Keypad:
  - Used to set the accident scene
    - Before the search starts, set the x-y location of the accident (x, y).

Output:

- LCD:
  - Used to display
    - Drone status
      - E.g. G(grounded), S(search), A(aborted), and other information at your discretion.
    - Drone location: (x, y, z)
    - Search result: “not found” or location of the accident (x, y, z)
- Motor:
  - When the drone flights, the motor spins at a fast speed; When the drone halts in the air, the motor spins at a low speed;
  - When the drone grounded, the motor stops.
- LED:
  - Used to indicate the start of simulation
    - E.g. flashing for a few seconds.

Submission Information

The following items should be submitted:

1. Source code. Your program should be well commented.
2. User manual. The user manual describes how to your emulating system, including how to wire up the AVR lab board. Descriptions must be given on each input action and how output displays should be interpreted.
3. Design manual. The design manual describes how the emulating system is designed. The following components you may want to include in your design manual.
    a. System control flow at the module level using a diagram, which shows interaction between modules and input/output operations.
    b. Data structures and algorithms used in the software design

Both manuals should be well written. A person with knowledge about the subject and the lab board should understand how your system is designed and how to operate and modify your system after reading the given manuals.

Grading

The project is worth 15% of your final result and will be marked under the following criteria:

- Implementation to be demonstrated in lab (65%):
- Code Style (5%):
    - Easy to understand
    - Well documented
- User Manual (10%)
    - Good presentation
    - Clear and easy to understand
- Design Manual (20%)
    - Good presentation
    - Clear and easy to understand

Demonstration time: your lab class in Week 13.
Submission time: Friday, Week 13