COMP9032 Project

Description:

Microcontrollers have been used in many application domains. One application is for toys. Figure 1 below is a helicopter model. The operation of the helicopter is controlled by a remote controller that sends control signals to the helicopter via inferred.

Figure 1

In this project, you will be working individually to develop a system, using the AVR Development board, to emulate a controller to remotely control the helicopter flying in a big building hall. The area of the hall is 50x50 square meters and its ceiling is 10 meters high.

Assume the helicopter can fly at four speeds: 1m/s, 2m/s, 3m/s, and 4m/s (m/s stands for meters/second). The helicopter will be crashed if it hits the wall or ceiling.

The location of the helicopter is represented by (x,y,z) in the hall. The helicopter is initially grounded at the center of the hall (25, 25, 0).

During flight, the helicopter's position is determined by the speed and previous location of the helicopter. If the helicopter is in a position outside of the hall, it means the helicopter is crashed. You need to develop an algorithm to determine the flight location.

The design for the control system is specified as follows (note: here the operation of the helicopter is simplified):

Inputs

- Six keys on the key pad are used for the six flight directions: U-up, D-down, F-forward, B-backward, L-left, and R-right.
- Pushing buttons (or another two keys on the key pad), PB0 and PB1, are used for the speed up and down control. Within the given speed range. If PB0 is pressed, the helicopter speed is increased by one level; if PB1 is pressed, the helicopter speed is decreased by one level. The speed changing from one level to another level takes 0.5 seconds.
- Key '#' is used for taking off and landing. When this key is pressed, the helicopter goes from the ground upward at 2m/s. When this key is pressed again, the flight starts to descend vertically at the speed of 1m/s, when it touches to the ground. The helicopter stops.
- Key '*' is used for hovering control. When this key is pressed, the helicopter is put in the hovering state with its position unchanged. When this key is pressed again, the flight resumes the previous flight speed and direction.

Outputs

LCD, LEDs and motor are used to indicate the status of the helicopter.

- When the simulation is started or after the reset button on the board is pressed, "start" is displayed on LCD, as shown Figure 2(a):

Start:

Figure 2 (a)

- During flight, the location, current direction, and the speed of the helicopter are displayed on the LCD. The flight speed (specifically the propeller speed) is simulated with the motor. The higher the speed, the faster the motor spins. An example of the LCD display is given in Figure

2(b), where the helicopter is flight upward at the speed 3m/s and the current location is (2, 3, 1).

position	direction	speed
(2, 3, 1)	U	3m/s

Figure 2 (b)

- For a successful flight (landing safely), the distance and the total time of the flight is displayed on LCD, as illustrated in Figure 2(c).

Distance: 25 m
Duration: 40 s

Figure 2 (c)

- When the helicopter crashes, LCD displays the location of the helicopter and the LED bar flashes.

Note:

1. The LCD panel can only display limited number of characters. You may need to design how the related information can be displayed effectively. For example, to save the space, use ‘pos’ for ‘position’.
2. The motor simulation of the propeller speed is optional. But you are encouraged to implement it.

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 a user uses your emulating system, including how to wire up the AVR lab board. Descriptions must be given on each button action and how the LED and LCD displays should be interpreted.
3. Design manual. The design manual describes how you design the emulating system. 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,* which shows how information and data are represented in the design.

c. *Module specification*, describing the functions, the input and the output of each module.

d. *Algorithms, used in the software design*

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

Grading

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

- Implementation (55%):
  - Adherence to specification
  - Implementation of all input/output functionality
- Code Style (10%):
  - Easy to read
  - Well documented
- User Manual (15%)
  - Accurately describes set up of board
  - Correctly describes functionality
- Design Manual (20%)
  - Adherence to specification
  - Readability and Completeness

Demonstration: your lab class in week 13.
Submission: Friday, week 13