Course Project

This document is to provide guidance for the ECE2220 course design project. Note:

- The project is open ended, meaning its complexity can be increased as needed, and there is no single right solution to the project at any level.
- Larger groups will be expected to be responsible for more difficult project variations.
- Each group should divide the problem into multiple independent components, each of which will be investigated by one or two members of a group.

Week One: In the lab:

I. Form into groups consisting of 3 or 4 (max.) people.

II. Choose a group name, and a group leader. The group leader:

A. is responsible for putting together the final project and handing in the project report,
B. has final say in settling any design issues or other differences of opinions within the group.

III. In writing, submit to the TAs:

A. The name of your group.
B. A list of the names and student numbers for the members of the group.
C. Indicate who is the group leader.
D. State any assumptions that you think will be necessary to complete the project (these must be justified and approved by your TA).
E. Indicate how the design will be partitioned between the group members.

IV. Begin working on the design project.

A. Determine exactly what you are going to do, how in depth, and what aspects of the problem you are going to tackle, (a state diagram is a good start),
B. Determine in advance the public interface to each component, i.e. what are the inputs and what are the outputs,
C. Make sure these are standardized among all sub components,
D. Specify a timeline for the project, remember to allocate sufficient time to prepare a presentation and report.
E. Be sure to leave time for verification. Design without verification is meaningless.

Week Two: In the lab:

V. Provide verbal progress report to TAs - Be prepared to discuss the state diagrams for your project, and any other interesting design issues with the TAs, time estimate 10 minutes / per group.

VI. Continue working on implementation - State diagrams for main sub-systems should be completed.

Week Three: In the lab:

VII. Continue working on implementation.

Week Four: In the lab:

VIII. Project report to class – Present an overview of your implementation to the class. Discuss any specific features the group has implemented, problems encountered and their solutions, and any other interesting features in which you learned something which would be of interest to your fellow students. Time: approx. 15-20 minutes/group.

IX. Written project report is due. Final due date is 03 December 2007 at 4:30 p.m.. Projects may be turned in to your TAs, or to the ECE department office (E2-390).

Your report should be professional in quality (you will be marked on presentation, spelling, etc. as well as content). The report should include (but is not limited to):

- Design issues regarding the chosen design,
- Ideas (whether used or not) on how the design issues were tackled.
- The reasons why an idea was used (or more importantly, not used),
- Any problems which may have been encountered, and how they were (hopefully) solved,
- The design specification, from simple English to mathematics, tables, circuit diagrams, state and timing diagrams, etc.
- Simulations, and results,
- And of course what (if anything) you learned.

While the basic discussion of the design should come from the given specification, please do not simply repeat it in the report. Put it in your own words, and then move on to your contributions, design and analysis of the problem.

Most of the final grade will come from the effort shown on the project. The problem is open ended and you are allowed to make assumptions as you go along. However, be sure that any assumptions are clearly stated, and are at the very least, reasonable. The scope of the work required to implement the project is left up to each group provided that the minimum design specifications are met.

Mark Distribution:

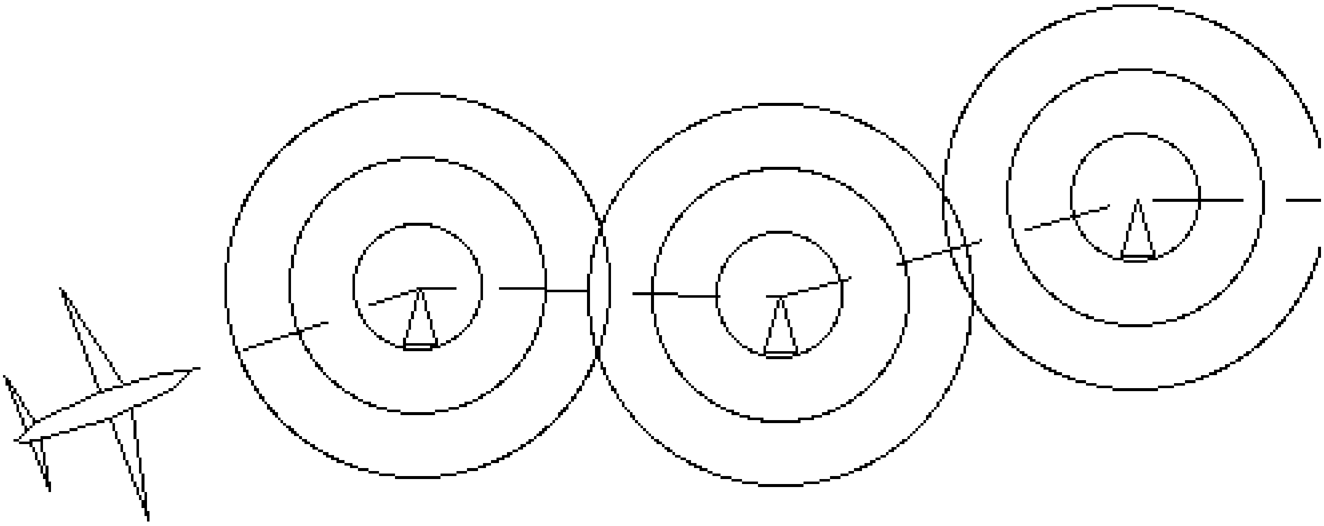
Proposal =	10
Oral Report =	30
Written Report =	30
Peer Evaluations =	10
Demonstration =	10
Complexity =	10

Total =	100

Project 2007: Aircraft Navigation Beacon Monitor.

The goal of this project is to design a prototype system that is able to autonomously monitor the output transmission of an aircraft navigation beacon.

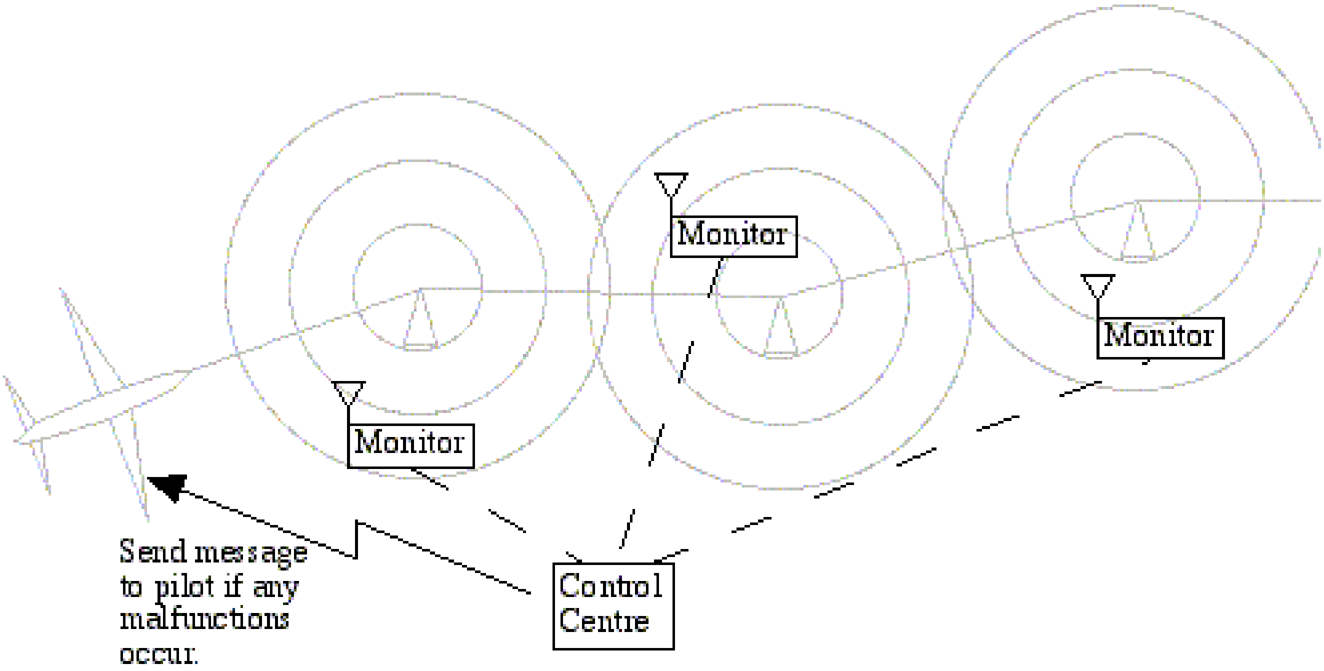
Background
When flying, an aircraft does not necessarily take a straight line path from its point of departure to its destination. As a passenger, often you can feel the plane banking and changing heading. This is because, the aircraft is navigating using a number of intermediate points between the two airports. Even in though we have GPS (Global Positioning System) satellite navigation available, aircraft navigation is normally accomplished via listening to radio transmitters on the ground acting a fixed beacons. As a result the aircraft navigates from one navigation beacon to the next.



The system uses what are called non-directional beacons (NDBs). The term non-directional means that they do not transmit direction information to the aircraft. However, since the transmitters are fixed at a known location, pilots can calculate their own position based on their ability to receive the transmitted signal. Since the pilots need to be able to identify the specific transmitter, each NDB transmits its own Morse code callsign/identifier.

To a pilot it is important that the chain of transmitters be monitored for proper operation since a malfunctioning system could result in the pilot becoming lost. This is one of the key reason for not using GPS satellite navigation is that it is not easy to properly monitor the GPS signals in terms of accuracy and availability whereas it is simple to monitor a fixed transmitter. This is because GPS satellites are not geostationary and are therefore constantly moving with relation to the receiver. As a result, it is difficult to know which satellites you are using to calculate your position, and to know if any of the satellites you are listening to may be transmitting a corrupted or erroneous signal

Thus for the safe operation of the air navigation system is necessary to be able to monitor these NDB transmitters to ensure they are properly transmitting. This is normally done by placing an antenna and receiver within range of the transmitter and monitoring their outputs.



System Requirements
The task is to build a digital monitoring circuit capable of accepting and interpreting a Morse code input in order to determine its correctness.

Because the system you are designing is meant to be a prototype, we will make the following assumptions:

- It is assumed that if we can correctly decode the Morse code identifier the transmitter is operating properly.
- The Morse code can be entered using one of the push buttons on the lab board.
- The Morse code identifier will made up from the first initials of various group members.
- The system needs to be able to decode two different two-letter identifiers.
- For the two letters in each identifier there should be a minimum of four symbols consisting of both dots and dashes. eg. The identifiers EE(.   .), II (. .   . .), ET (.   -), etc, are not acceptable.

The operation of system should be as follows:

- After a reset the system should monitor the keypad and wait for an input indicating which identifier should be monitored. (Eg.key1 = indentifier 1, key 2 = indentifier 2);
- The system monitors the Morse code input:
  - If nothing is for a period longer than the time between word an error is displayed.
  - While the code is being received the system should indicate it is receiving a signal but that it cannot determine if it is correct.
  - At the end of the input sequence the system should provide and indication if the identifier is correct else it must indicate an error condition.
- If an error condition is output the system waits for a reset.

The built in LEDs and seven-segment displays can be used to generate any output information. Addional output may be added by interfacing output signals to the breadboard section of the experiment board.

Morse code details:
Morse code is a method for transmitting information using sequences of short and long symbols (dots and dashes) to represent letters, numerals, etc. The following are the letters of the alphabet encode in Morse code:

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M
Morse	.-	-...	-.-.	-..	.	..-.	--.	....	..	.---	-.-	.-..	--
Letter	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Morse	-.	---	.--.	--.-	.-.	...	-	..-	...-	.--	-..-	-.--	--..

If the length of time for a dot is taken to be one unit of time, then a dash is three units. The time between the symbols of one character is one unit, the time between characters is three units and the time between words seven units.

In the following table each numbered square represents one unit of time. Therefore if the identifier is 'BY' the transmission would be a repeating pattern of:

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
				B														Y													

The grey boxes represent the time that the input button should be held down.

As was stated in the course project description:

Larger groups will be expected to be responsible for more difficult project variations.

You may want to expand your implementation by adding features. These could be details such as:

- Having the ability to detect a long sequence such as 'JQ';
- Indicating if one of the two characters in the identifier are correct;
- The ability to accommodate random errors, for example, if two out of the last three sequences are correct, it is assumed the transmitter is okay;
- Decode and display the input as it is being recieved. i.e., if the sequence you are monitoring is BY, for example, after the 'B' is received display a 'B' on the seven segment display and then after the 'Y' is received show 'BY' on two of the seven segment displays;
- Design a Morse code generator which can be used to provide the input to your monitoring circuit;
- etc.

As a ***minimum*** your implementation will require the use of:

- User input devices (buttons/keyboard);
- Output displays - you may use the built in LEDs or may make use of external devices such as the seven segment display used in lab 3;
- FSM(s) to control the interaction of the parts;
- Combinational/sequential circuits to complete the implementation.

Hints/Tips:

The following may be helpful in getting started:

- Try to break up the task in to smaller, easier to accomplish units. This will allow for the even distribution of work and should avoid potential bottlenecks where one persons work depends on the completion of a second persons work;
- Try to make the design modular so you can reuse portions of your design;
- Use an LED to flash on the unit time (like a metronome);
- You can design your circuit so that it samples the input in the middle of each unit of time, therefore it won't be necessary to hold the button for precisely one or three units of time;
- Implement a circuit which can identify a single letter before expanding the design to the full implementation.

References:

NDB (Transport Canada):
<http://www.tc.gc.ca/CivilAviation/RegServ/terminology/glossary/n.htm #non_directional_beacon>

NDB (FAA):
<http://www.faa.gov/airports_airtraffic/air_traffic/publications/ATpubs/PCG/N.HTM>

NDB (Wikipedia):
<http://en.wikipedia.org/wiki/Non-directional_beacon>

Morse code translator:
<http://morsecode.scphillips.com/translator.html>