Rochester Institute of Technology

Real Time and Embedded Systems: Project 6

Overview:

Design and implement an embedded, real-time stand-alone system to play a primitive "mirror" game using 2 servo motors, and the gyroscopic sensor for input.

Details:

The game: The goal of the game is to have the player "chase" the position of the computer. The game makes use of both servos; the computer servo and the player servo. The computer moves its servo to a random position. The player controls their servo by rotating the discovery board.

Game play is as follows: The computer moves to a position and stops; The player then has a fixed amount of time to rotate the discovery board to the same position as the computer. The player servo is used as a live, real time, feedback interface to the player while the game is running. If the user makes it to the proper position before time elapses then it is a hit; if not it is a miss. Either way – the computer then moves its servo to a new random position and starts the game all over again. This repeats for 10 times; then a final score is computed.

The setup: Using the STM discovery board. Connect both servos to the board, these serve as the interface to the player. One servo is the computers; the other servo is the players. The demo board already as a gyroscope that is connected via the SPI bus.

Design Constraints:

- The gyroscopic sensor is used to determine the rotational position of the board; you may find that the Y axis is most appropriate to use.
- You may optionally use the red and green led's to show additional player feedback…
- Positions: are not discreet – they can be anywhere within the range of the servo.
- Successive computer random moves may create an overlap – make your game correct for this.
- The fixed amount of time to reach the destination is 5 seconds for the first level.
- There are libraries already available for interfacing to the gyroscope; however, you will need to add some signal processing to the raw returned information to get a final value. This is a rate sensor – not an accelerometer; you will need to process the output of this sensor in real time to get an angle.
- Player position counts as a hit if it is within +/-5% of the computer position.
- During play the "player" servo is continuously updated and shows the position of the demo board at all times; the player servo directly mirrors the boards position.

WARNING:
This project does involve moving the demo board while it is attached to the servos; use jumper wires carefully; "Strain relieve" the connections as necessary (tape, tie wraps, etc.) You need to maintain signal integrity at all times.

Report:
In addition to the demonstration of your project, a brief report with the required sections is required. List the trade-offs and assumptions of your implementation. Inclusion of proof of operation is not required for this project – the demo is sufficient. Your source code must be included in your electronic submission.

Grading Criteria:

- Program Operation and Demo – 50%
  - Hardware setup is orderly and well organized – 10%
  - Demo sheet functions all completed – 30%
  - Demo operates without faults or restarts – 10%
- Program Design --- 15%
  - Proper initialization
  - Correct use of functions (no copy/paste/edit slightly)
  - Separation of hardware related code from pure software (e.g. the results reporting code)
- Source Code Structure and Readability – 10%
  - Appropriate use of white space – 2%
  - Consistent and good indentation – 2%
  - Appropriate comments at the function and paragraph levels (such as a for loop) – 2%
  - Following C style guide (good names, etc.)
- Report Content – 25%
  - Report is at least 2 pages (not counting pictures, cover page, diagrams) – 5%
  - Demonstrates team understands the problem, solution, and technology (hardware and software) – 5%
  - For this project it must include a complete description of the communication system design – 5%
  - Report contains all required sections (except as noted above) per the report guidelines – 10%
- Bonus Opportunity – up to 20% at instructor discretion
  - Use the "recipe" concept from the other projects to define several progressively more difficult "levels" through the game;
  - Change the time intervals for each level to increase difficulty