Final Project:

Proposal

Chad Huard :: ee6681@wayne.edu :: 4/4/10

Description:

This proposal describes a final project for ECE 2620 Intro to Microcomputers. The finished product of this project will be a simplified robotic arm built using two servo motors allowing rotation in 3 dimensions. The arm will be controlled by the user tilting a 3d accelerometer in space. This project will require the use of almost all of the skills learned in class including analog to digital conversion and use of the HC11 timing port. As the arm being built in this project serves no particular purpose the project is rather to be seen as an archetypal solution to using an accelerometer to control servo motors; other applications of which could include helicopter stabilization and terrain compensation in ground vehicles.

Project Goals:

- Accurately measure and calibrate input from accelerometer
- Smoothly and accurately control servo motors based on accelerometer input
- Produce clean and modular assembly code
- Create modular PWM 'library' for use with other projects using HC11

The main goal of this project is to ensure that the robotic arm operates smoothly and accurately based on the accelerometer input. Many such designs are not properly optimized and end up being jerky or slow to respond to stimulus. The HC11 is a competent processor and with careful programing should have no problem providing acceptable performance in this application.

Learning Objectives:

The project goals give a measure against which to judge the final result of this project, but it is also important to look at what skills are required in order to accomplish this task. These include:

- A/D Conversion: The accelerometer being used provides 3 analog outputs, one for each axis of motion. These must be read by the µP's built in analog to digital converter before they can be used.
- Timing Port: Servo motors require a very specific PWM signal to set their position. In order to accomplish this the HC11 timing port will be utilized.
- Interrupts: This project will not employ any external interrupts, but it will utilize the output compare interrupt to control the PWM signal used with the servos.
- Time Multiplexing: This project will use the four on-board 7 segment displays requiring time multiplexing.

Hardware:

1. 68HC11 EVBPlus:

The central component to this project is of course the 68HC11. The HC11 will be responsible for running the assembly program and acting as the brain of the entire system. Most of the input and output features of the HC11 will be employed in one way or another. The EVB provides a platform for the HC11, allowing us to program the chip easily and giving the chip access to a variety of peripherals. This project will also use the on-board pot, buttons and 7-segment displays.

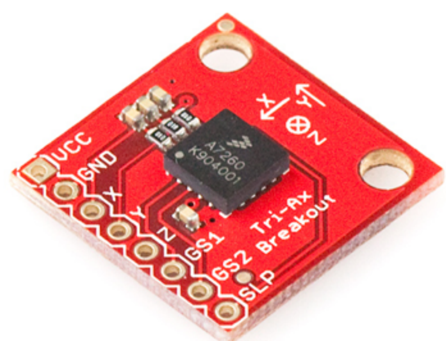
2. Servo Motors:

The servo chosen for this project is the TowerPro SG-5010:

Like most (non digital) servos this unit requires a PWM signal to control its position. I can not find a data-sheet, so its exact specifications will need to be determined experimentally. Initial trials show that it works well using fairly standard servo signal ranges. The PWM signal used in this case has a period of 20ms and a high time of 1-2ms with 1ms high time corresponding to left, 1.5ms corresponding to middle or zero position, and 2ms corresponding to right. These values will be verified experimentally.

3. Accelerometer:

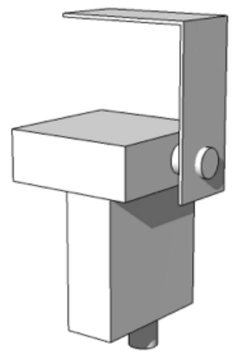
The accelerometer chosen is the Freescale MMA7260QT. This chip is mounted on a breakout board by Spark Fun Electronics to allow easy connection and implementation.



This board, once supplied appropriate power, provides analog output of the force exerted on the chip by gravity in each of the three axis. This analog signal will be converted to digital using the HC11's internal analog to digital converter. For more information on the output of this device see the data sheet at: http://www.sparkfun.com/datasheets/Accelerometers/MMA7260Q-Rev1.pdf.

Solution:

The arm will be a very simple construction of two servo motors mounted at 90° to one another. The bottom servo will have its shaft attached to a base, and the top servo will have its shaft attached to a yoke. This arrangement will allow the yoke to rotate through all 3 axis.



Each servo will be controlled by one of the output compare pins on the HC11. Since both units require the same frequency TOC1 will be used to establish the frequency of the PWM signal. Registers TOC2 and TOC3 will be used to control the pulse width since the this will be different for each servo.

Input from the accelerometer will be connected directly to Port E for A/D conversion. We will only need the X and Y axis data since the arm only has two degrees of freedom. This configuration will result in the arm rotating left or right when the user tips the accelerometer board left or right, and the arm tilting up or down when the user tips the board backward or forward. The accelerometer uses 3.3V and the output is ratiometric, so to increase resolution it may be necessary to use the Vrl and Vrh pins on the HC11.

In addition to using the accelerometer to control the arm I would like to implement a zero point adjustment routine to the system. This will be called when the user presses a button on the board and will give the user a way to adjust the middle or zero point of each servo, a kind of calibration mode. When the user presses the calibration button the arm will return to its zero point and display the current zero offset of the first servo on the 7-segment displays. The user will be able to use the on board pot to move the zero point left or right and see the offset change on the display. When the user presses the button again they will be able to adjust the zero point of the second servo. A third press of the button will return the arm to standard mode but with the new zero points.

Discussion:

The configuration above is intended to demonstrate my competence with assembly language and embedded programming, but in and of itself serves little purpose. If it is simply being used as in input device does the accelerometer have any qualities that a joystick would not? Probably not. The next logical step in the accelerometer project would be to mount the accelerometer to the yoke itself. In this configuration the system could be designed so as to keep the accelerometer level even when the base on which it is mounted is being tilted in any direction. This configuration of the device would be a more effective use of the accelerometers particular qualities, but is significantly more difficult to program. There being only three weeks before the final project is due I propose to complete the project as stated in the solution with a mind to expand the code to allow for the self-leveling functionality only if time allows.