Real-Life Augmentation of the Xbox Controller

Justin Schmelzer, Christy Swartz

November 4, 2011

Abstract

Our project is an augmentation of the Xbox controller. We will be using various sensors on the legs, head, and a prop gun to represent the inputs from the Xbox controller, and interface these inputs with the Xbox. For example, accelerometers on the legs will detect movement that maps to the left analog stick of the Xbox controller, and the trigger on the prop gun will map to the right trigger of the Xbox controller. A liquid crystal display will be mounted on the head and will allow the user to view the game while moving around.

Contents

1 Overview

The system augments the PC interface by implementing the communication protocols used by PS/2 keyboards and mice in the *PS/2* module, allowing the system to communicate with a PC. The *Walk* module will be responsible for producing the keyboard information used in the PC game Halo (e.g., the W, A, S, D keys correspond to walking forward, left, back, and right in the game), and the *Look* module will be responsible for producing the mouse information used in the game (e.g., forward movement of the mouse corresponds to looking up in the game). Detectors on the legs and head will remove the need for the use of keyboards and mice in the game.

An accelerometer on the legs will detect movement in the x, y, and z directions (i.e., forward, backward, right and left, and up). The analog voltage signals from the accelerometer, after being processed by 3 separate Analog to Digital converters, will be be converted to PS/2-compatible keyboard commands in the *Walk* module and sent to the *PS/2* module.

Two gyros on the head will act similarly as the accelerometers on the legs, except they will detect rotational movement of the head (i.e., left and right, up and down). Two Analog to Digital converters will be used to process the analog signals from the gyros, and the information will be converted to PS/2-compatible keyboard commands in the *Look* module and sent to the *PS/2* module.

2 PS/2 Module

2.1 Description

The PS/2 module controls the communication between the computer and our augmentation by implementing the PS/2 protocol for keyboards and mice.

2.2 Inputs

- 4-bit output from the Accessory module
- 24-bit output from Look module
- 3-bit output from Move module
- End timer pulse from Counter module
- Clock pulse

2.3 Output

- 6-bit mouse data to computer
- 6-bit keyboard data to computer

- 2-bit timing parameter to counter module

2.4 Testing

We plan to view the mouse and keyboard output on the labkit's hex display along with connecting a PS/2 keyboard and mouse cable from the labkit to the computer. To test the connection from the PS/2 cable, we will send inputs from the *PS/2* module into text file for keyboard testing. Display output values on hex display and see if mouse moves on screen for mouse testing.

3 Look module

3.1 Description

The Look module will be responsible for processing the information from the gyros on the head.

3.2 Inputs

8-bit output from the *A/D - Head*, corresponding to looking up, down, left, and right. The *A/D - Head* output is in the form of angular velocity of the head in the vertical and horizontal directions; in order to translate the data to position, we must integrate it.

3.3 Outputs

- 24-bit bus to PS/2 module
- Control line to the *A/D - Head*

3.4 Memory Requirements

Two 8-bit buses will represent the vertical and horizontal position of the head. Since PS/2 mice send an x and y position input to the computer, the vertical position bus will correspond to the y position input; the horizontal position bus, the x position input.

We must integrate the angular velocity over the clock cycle to obtain the position of the head. A Riemann sum integration will require 2 multiplications and 2 additions per clock cycle; the resulting x,y positions from the integrations will be added to the respective buses during the next clock cycle.

The control line to the *A/D - Head* must be asserted to read information from the *A/D - Head*.

3.5 Concerns

Since we are not sure what the output of this module will be with the gyro data as input, we may have to scale to accommodate for differences between a typical input on a mouse and our input.

3.6 Testing

We plan to simulate various inputs in Modelsim, and view outputs in both ModelSim and the hex display on the labkit.

4 Walk module

4.1 Description

The Walk module will be responsible for processing the information from the accelerometers on the legs.

4.2 Inputs

24 bits from the *A/D - Legs*, corresponding to jump, walk forward, and walk back. The *A/D - Legs* output is in the form of acceleration; in order to translate the data to position, we must integrate it twice.

4.3 Output

- 3-bit bus mapping to the W, A, S, D keys on a keyboard, plus the right and left buttons on a mouse
- Control line to *A/D - Legs*

4.4 Memory Requirements

Three 8-bit buses will represent the x,y, and z position data from the user, respectively. In order to calculate the integral of the acceleration from the *A/D - Legs*, we will need 6 multiplications and 2 additions per clock cycle, since we will be using a Riemann sum to calculate the integral per clock cycle by multiplying clock cycle time squared by the input ($\frac{1}{2}at^2$) and then add to x,y registers holding position information.

The control line to the *A/D - Legs* must be asserted to read information from the *A/D - Legs*.

4.5 Testing

We plan to simulate various inputs in Modelsim, and view outputs in both ModelSim and the hex display on the labkit.

5 Gyro

5.1 Inputs

Movement of the user's head.

5.2 Outputs

Analog signal corresponding to the angular velocity of the user's head

5.3 Challenges

There will be lots of small movements picked up by the gyros, and not all of them will be intentional. We must characterize the output of the gyros through testing so we can determine the threshold when the movement from the user is intentional, and use this threshold in the *Look* module.

5.4 Testing

We plan to view the analog output of the gyro on an oscilloscope while actively moving the gyro.

6 Accelerometer

6.1 Inputs

Movement of the user's legs in 3 directions.

6.2 Outputs

Three analog voltage signals, mapping to the forward, left and right, and vertical movement of the user.

6.3 Challenges

There will be lots of small movements picked up by the accelerometers, and not all of them will be intentional. We must characterize the output of the accelerometers through testing so we can determine the threshold when the movement from the user is intentional, and use this threshold in the *Walk* module.

6.4 Testing

We plan to view the analog output of the accelerometer on an oscilloscope while actively moving the gyro.

7 A/D - Legs

7.1 Inputs

Analog voltage output from the x,y,z axes of the accelerometer

7.2 Outputs

24-bit bus corresponding to an 8-bit value for each of the x,y,z directions of movement from the user

7.3 Testing

We plan to view the digital output on the logic analyzer while giving the A/D a varying input from an oscilloscope.

8 A/D - Head

8.1 Inputs

Analog voltage output from the two axes of rotation of the gyros

8.2 Outputs

8-bit value corresponding to anglar velocity of the user's head

8.3 Testing

We plan to view the digital output on the logic analyzer while giving the A/D a varying input from an oscilloscope.

9 Accessory

9.1 Description

The accessory module debounces and codes the inputs from the various buttons on the prop gun into a 4-bit bus to the *PS/2* module.

9.2 Inputs

Keyboard key → button mapping for the PC game Halo

- Switch Grenade = G
- Switch Weapon = Tab
- Reload = R
- Melee Attack = F
- Exchange Weapon = X
- Flashlight = Q
- Scope Zoom = Z
- Action = E
- Crouch = left Ctrl

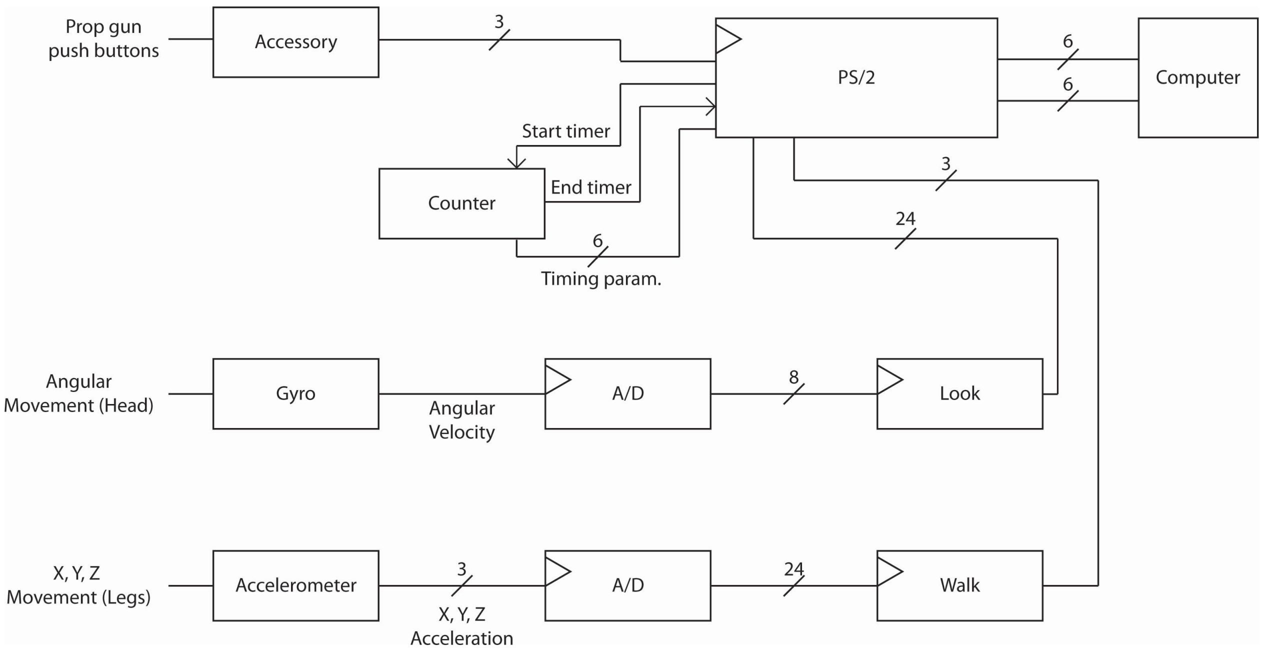
Mouse button → button mapping

- Fire weapon = left button
- Throw grenade = right button

9.3 Testing

We plan to view the outputs from the various buttons and the module itself on the LEDs on the labkit.

10 Block Diagram



11 Bill of Materials

- 1 3-axis accelerometer
- 2 1-axis gyro
- 7 push-buttons
- 1 prop gun