Real-Life Augmentation of the Xbox Controller

Justin Schmelzer, Christy Swartz

November 14, 2011

Overview

- Removes need for keyboard, mouse in PC games
- Augments the PC interface by implementing the communication protocols used by PS/2 keyboards and mice
- Accelerometer on the legs will detect movement in the x, y, and z directions (i.e., forward, backward, right and left, and up)
- Two gyros on the head will detect rotational movement of the head (i.e., left and right, up and down)

Gyro

Inputs and Outputs

Input Movement of the user's head.
Output Analog signal corresponding to the angular velocity of the user's head

Challenges

- Small movements picked up by the gyros

Testing

- View analog output of the gyro on an oscilloscope

Accelerometer

Inputs and Outputs

Input Movement of the user's legs in 3 directions
Output Three analog voltage signals, mapping to the forward, left and right, and vertical movement of the user

Challenges

- Small movements picked up by the accelerometer

Testing

- View analog output of the accelerometer on an oscilloscope

A/D - Head

Inputs and Outputs

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

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

Testing

- Give A/D varying input from function generator
- View digital output on the logic analyzer

A/D - Legs

Inputs and Outputs

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

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

Testing

- Give A/D varying input from function generator
- View digital output on the logic analyzer

Look module

Inputs

- 8-bit output from the *A/D - Head* = looking up, down, left, and right

Outputs

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

Look Module

Memory Requirements

- Two 8-bit buses represent the vertical and horizontal position of the head.
- Position = integral of the angular velocity
- Control line asserted for new information

Testing

- Modelsim
- Hex display
- Graphical

Walk module

Inputs

24 bits from the *A/D - Legs*, corresponding to jump, walk forward, and walk back.

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*

Walk module

Memory Requirements

- Three 8-bit buses = x,y, and z position data from the user
- Integral of acceleration from the *A/D - Legs* = 6 multiplications and 2 additions per clock cycle
- The control line to the *A/D - Legs* must be asserted to read information from the *A/D - Legs*.

Testing

- Modelsim
- Hex display
- Graphical

PS/2 Module

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

Output

- 6-bit mouse data to computer
- 6-bit keyboard data to computer
- 2-bit timing parameter to counter module

Testing

- Labkit hex display
- Connect PS/2 output to computer running text editor

Accessory

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

Accessory

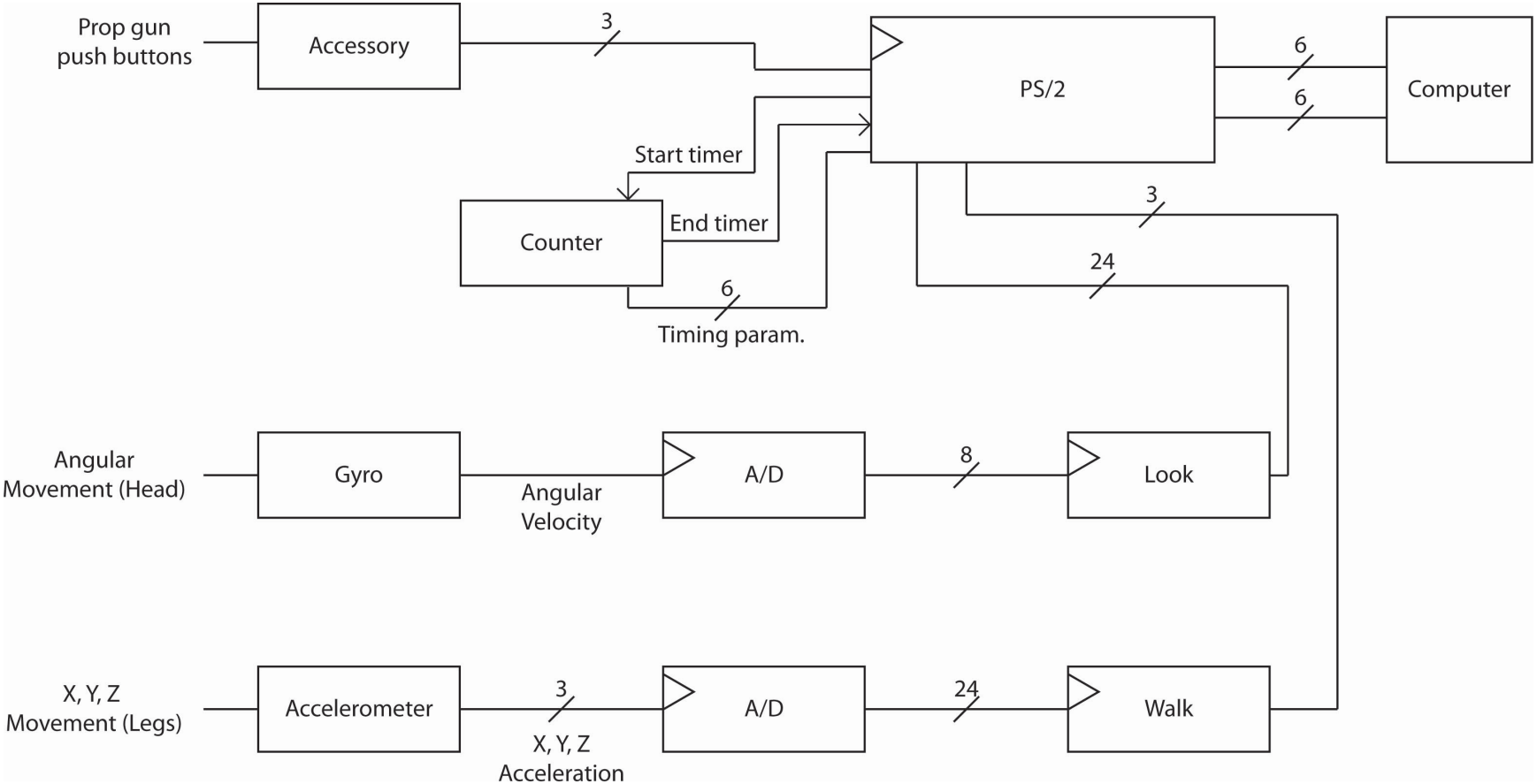
Outputs

4-bit bus representing button data

Testing

- View outputs on labkit LEDs

Block Diagram



Work Assignments

Justin

- Look module
- Walk module
- Gyro, A/D setup
- Accelerometer, A/D setup

Christy

- PS/2 module
- Graphical debugging (see Look, Walk modules)
- Assisting with other modules after assigned tasks are completed

Bill of Materials

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