USB Virtual Reality HID

Functional Description and Complete System Block Diagram

Students:

Weston Taylor

&

Christopher Budzynski

Project Advisor:

Dr. Malinowski

November 11, 2008

Introduction

The purpose of this project is to create a USB (Universal Serial Bus) Virtual Reality HID (Human Interface Device). This USB HID will interface with personal computers and their programs by emulating both a mouse and a keyboard. The HID will translate user movements into on-screen in-game actions, to provide the user with a more lifelike interactive platform for personal computer software.

The USB HID will consist of three major subsystems all working together to provide the overall interactive environment. The first subsystem is the Handheld Pointing Device which will use accelerometer and gyroscope readings to translate the user's arm movements into on-screen mouse locations. The second subsystem is the Step Pad; this subsystem will provide the commands that allow the user to move throughout the on-screen virtual environment. The USB communication board is the final subsystem. This board will receive keypad and mouse location data from the other two subsystems; it will then translate this data into USB mouse and keyboard commands which create on-screen actions when sent to the host PC.

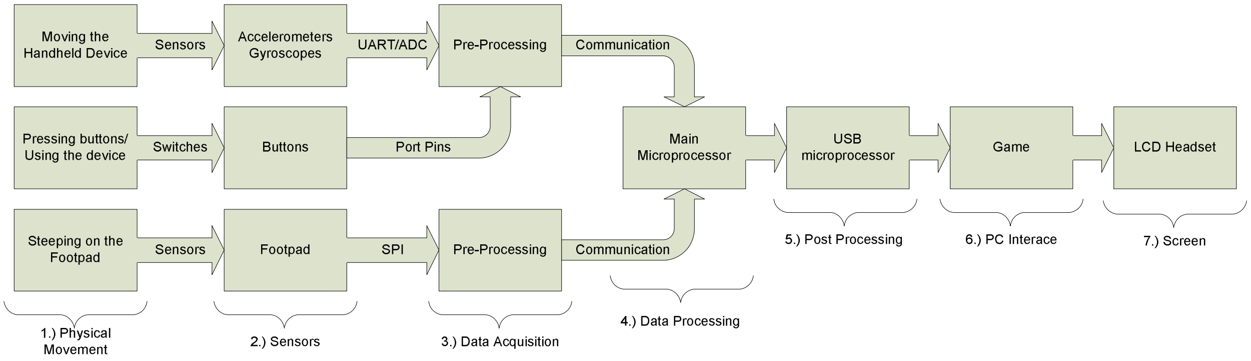
Goals

The overall goal is to translate user movements into on-screen actions to provide a more realistic in-game experience for the user. Here are the detailed objectives that will make this goal attainable:

- USB communication with a PC, emulating both a mouse and keyboard
- Translation of accelerometer and gyroscope readings into on-screen mouse locations
- Processing of data and USB communication using embedded systems
- Development of a Step Pad for in-game movement (or interfacing of an already produced step pad for the embedded application)
- Use of a Headset instead of a monitor to view in-game results
- All embedded programming done using the C programming language, to provide portability and reusability of code
- If time permits, communication between subsystems will be done using the most appropriate form of wireless communication

System Block Diagram

Figure 1: Over-all System Block Diagram



1.) Interaction with the device is always initiated by physical movement. Maneuvering the handheld device will determine where the person is looking and pressing its buttons simulate in game actions. Movement within the virtual environment is initiated by the user applying pressure to the footpad sensors by stepping on it.

2.) The sensors will be designed to translate a user's movement into a useable form, such as serial data or analog voltages. Accelerometers and gyroscopes will be used to calculate on-screen position, while buttons will be used to simulate functions in the game. Lastly, the step pad will allow a user to step in any direction to make the on-screen character move within the virtual environment.

3.) Data acquisition will be determined by the sensors; both analog and digital interfaces could be used. However the buttons on the handheld device itself will use port pins, and the step pad uses a modified serial interface. Pre-processing involves preparing the data for processing. Gain will be used to insure that all calculations will be done in the same frame of reference and units. Also, depending on the data, offsets may be needed to center the data in the proper area.

4.) Data processing involves taking the cleaned data from pre-processing and converting it to useful data that can be used by the USB communication module. Integrations will probably be needed for both the accelerometers and gyroscopes. Also, the buttons on the device and in the footpad need to be processed into their corresponding keyboard button presses.

5.) Post processing involves taking the data from data processing and forming packets to be sent across USB to the PC.

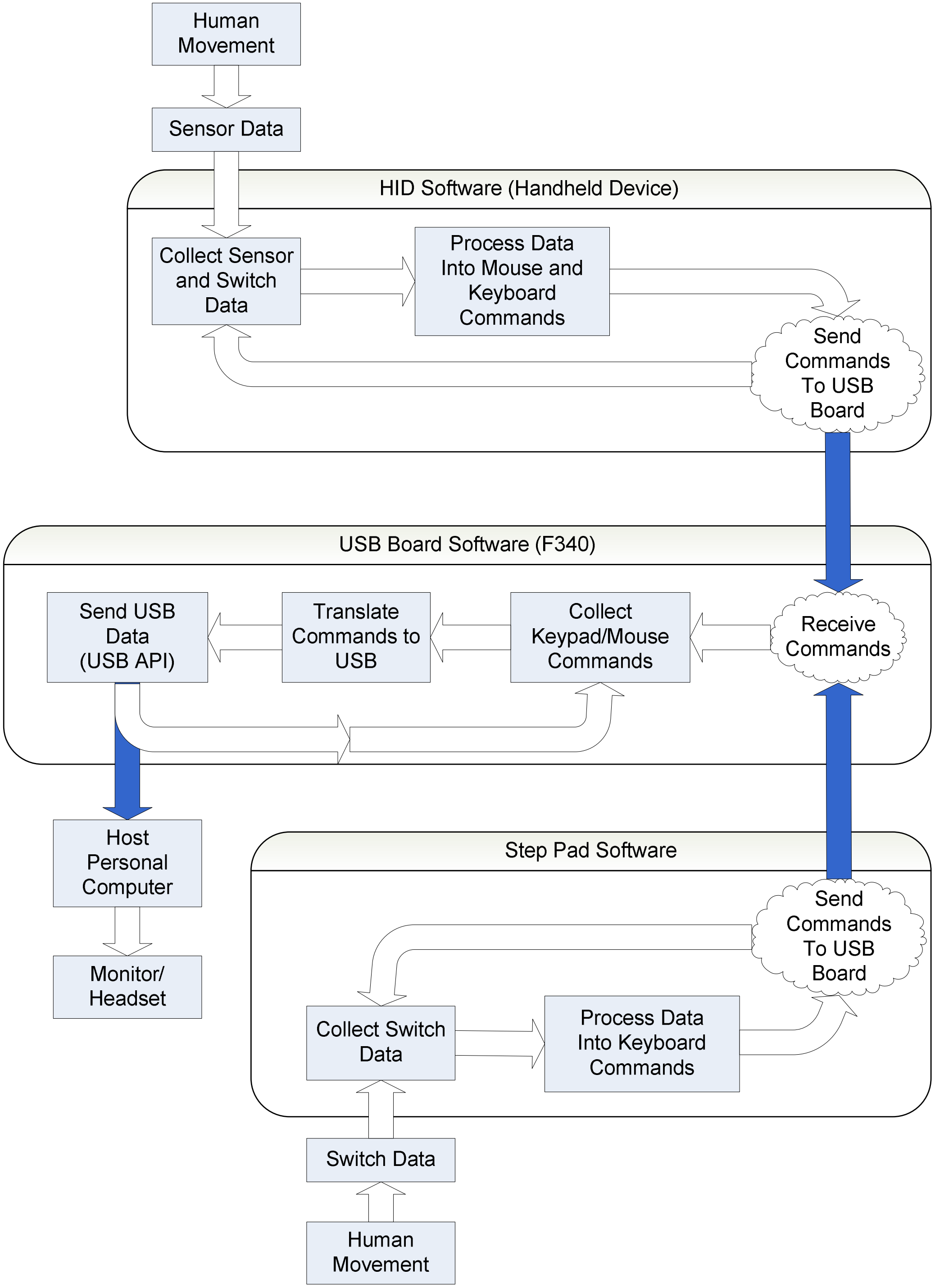
6.) The PC will be running a game; therefore, the data passed over USB will translate into in-game actions and movements within the virtual environment.

7.) The LCD headset will connect to the PC; this will provide the user with the freedom to turn round without worrying about a fixed screen.

High-Level Software Flowchart

This project will use multiple embedded systems to provide data acquisition, data processing, intersystem communication, and USB communication; this means that a major portion of the project will be software development. The overall high-level software flowchart can be seen in Figure 2 (below).

Figure 2: High-Level Software Flowchart


Human Movement
Sensor Data
HID Software (Handheld Device)
Collect Sensor and Switch Data
Process Data Into Mouse and Keyboard Commands
Send Commands To USB Board
USB Board Software (F340)
Send USB Data (USB API)
Translate Commands to USB
Collect Keypad/Mouse Commands
Receive Commands
Host Personal Computer
Monitor/ Headset
Step Pad Software
Send Commands To USB Board
Collect Switch Data
Process Data Into Keyboard Commands
Switch Data
Human Movement

Handheld Device Software (top of Figure 2)

This software block has a simple task; it must collect the sensor and button data, translate that data into keyboard and mouse movement commands, and send those commands to the USB communication board. The "Collect Sensor and Switch Data" block will use the Analog-to-Digital converter, the SPI, or the UART to collect sensor readings depending on the sensor hardware that is purchased. The next block will interpret that sensor data into keyboard button presses and mouse movement/position information. The final block sends the keyboard and mouse information to the USB board, using a wireless or serial communication method, so that it can route that information to the PC.

Step Pad Software (bottom of Figure 2)

This software block performs the same functionality as the Handheld Device Software except that only keyboard information produced. The Step Pad will not influence the mouse movement at all, so only keyboard button press information will be sent to the USB board.

USB Board Software (middle of Figure 2)

The USB board software's main function is to collect all keyboard and mouse movement information from the other devices and send it to the PC using USB. Its first software block, "Receive Commands", does just that; it receives the information from the other boards using either wireless or serial communication. Next, the two data streams are integrated together so that all keyboard presses and mouse movements are recorded. The "Translate Commands to USB" block translates the collected mouse and keyboard information into USB bytes that can be sent to the PC. Finally, the last block is the USB API that is in charge of the USB communication transfers with the PC.

Conclusion

The most difficult part of the project will be the translation of accelerometer and gyroscope readings into mouse movement and on-screen position, while reducing the influence of noise, saturation, and drift due to accumulated error. Also, the device must emulate both a mouse and a keyboard from the same device; this could potentially introduce some USB coding difficulties. Once these hurdles are overcome, this project will provide an exhilarating interactive platform for many personal computer software environments.

References

[1] Silicon Labs, C8051F34x Data Sheet,
https://www.silabs.com/products/mcu/usb/Pages/C8051F34x.aspx

[2] Wikipedia, Inertial Navigation System,
http://en.wikipedia.org/wiki/Inertial_navigation_system

[3] D. Schertz, EE565 Fall 07 Lectures Notes 20 - 24