USB Virtual Reality HID

Senior Project Proposal

Students:

Weston Taylor

&

Christopher Budzynski

Project Advisor:

Dr. Malinowski

December 11, 2008

Project Summary

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 a USB gamepad. The HID will translate user movements into on-screen in-game actions, to provide the user with a more lifelike interactive platform for PC games and other virtual environments.

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 absolute pointer locations. The second subsystem is the step pad; this subsystem will provide the commands that allow the user to move throughout the virtual environment. Finally, the USB communication board will receive movement and pointer location data from the other two subsystems; it will then translate this data into USB gamepad commands, which create in-game actions when sent to the host PC.

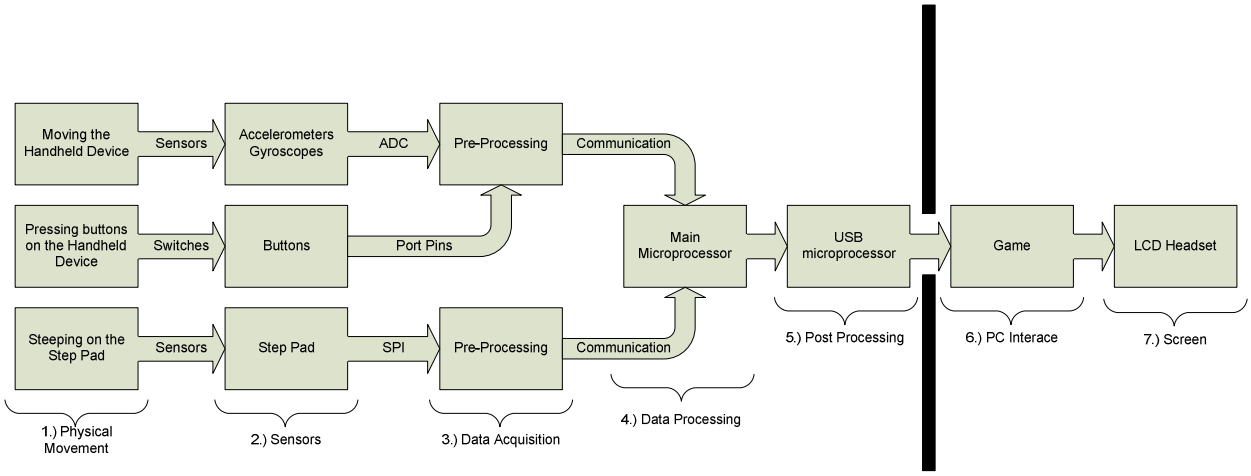
Goals

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 a USB gamepad
- Translate accelerometer and gyroscope readings into on-screen pointer locations
- Process data and USB communication using 8-bit embedded systems
- Interface PlayStation 2 DDR Step Pad for in-game movement
- 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 use the most appropriate form of wireless communication (ZigBee, Bluetooth, etc.)

Hardware Functional Description

Figure 1: Over-all System Block Diagram



1.) Physical movement always initiates interaction with the device. Maneuvering the handheld device will determine where the person is looking and pressing its buttons simulate in game actions. Pressure on the footpad sensors indicates movement within the virtual environment.

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

3.) The data acquisition algorithms will be determined by the sensors' output formats. However, the buttons on the handheld device itself will use port pins, and the step pad will use a modified serial interface. Pre-processing will prepare the data for processing by using software gain to insure that all calculations are 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 be needed for both the accelerometers and gyroscopes in order to get position data from rates of change. In addition, 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 video game, and the data passed over USB will translate into on-screen actions and movements within the virtual environment.

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

As previously stated, the primary objective of this project is to translate user motions into on-screen actions; therefore, the project will focus on the blocks to the left of the line in Figure 1 and existing PC software and LCD headset will be used.

Hardware System Requirements:

- According to preliminary testing, the average person can move a handheld device at a maximum of 350-500 degrees/second. The gyroscopes must be able to measure this movement without saturation.

- In regards to the footpad, most users will weigh less than 300 lbs; therefore, the footpad must function effortlessly up to this body weight threshold.

- USB 2.0 compatible devices will be used in Full Speed mode (12 Mbits/s) with frames being sent every 1ms (as specified by the USB protocol).

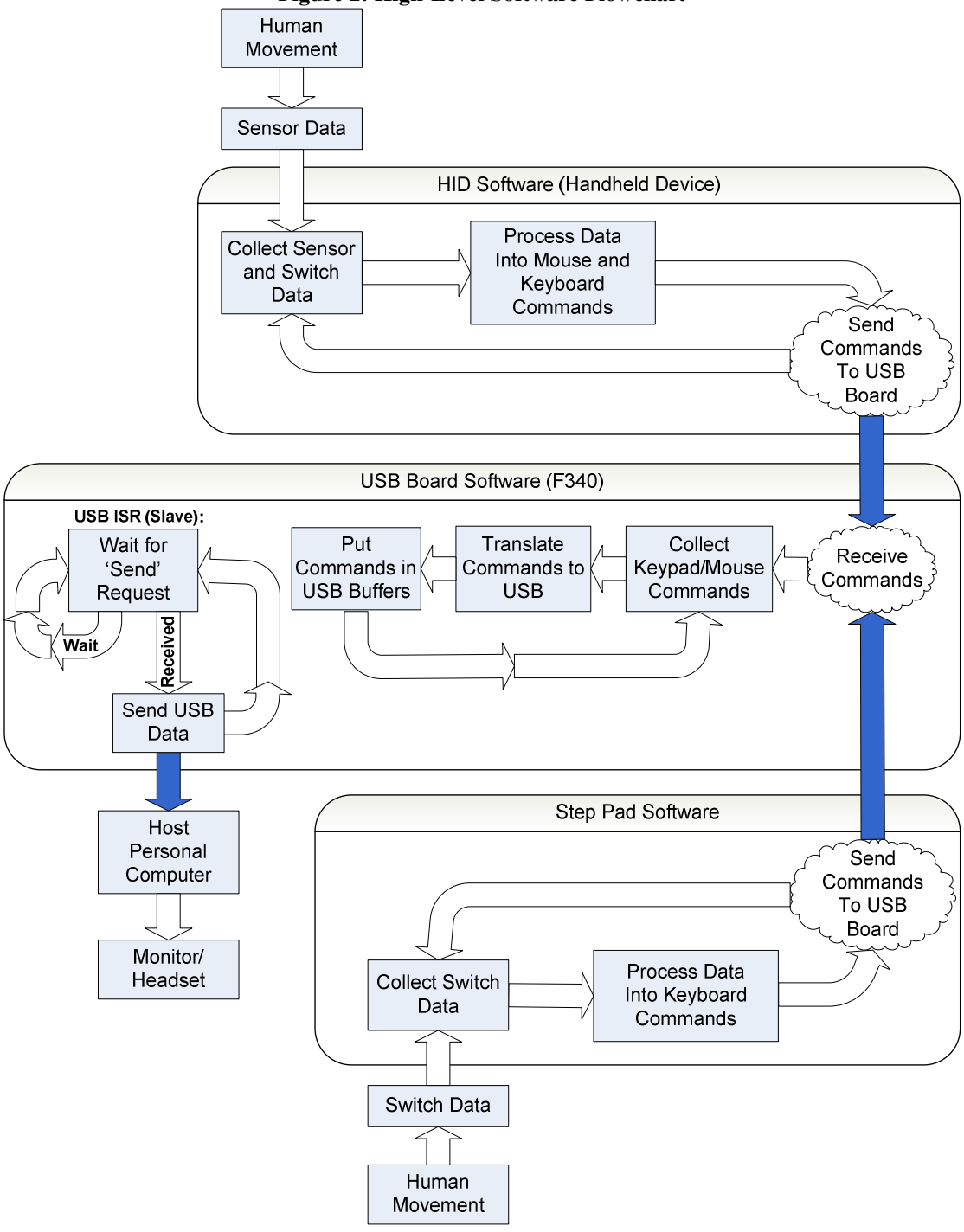
- All hardware devices must operate from 0°C to 40°C; this is not a large temperature range because this will be a consumer electronic product used inside at room temperature.

- Power: The USB board will be powered over USB (max 500mA) but the handheld device will have to be battery powered in order to remain portable.

Software Functional Description

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



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 USB gamepad absolute pointer position and movement commands, and send that information 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 USB gamepad movement information and absolute pointer position. The final block sends the data to the USB board, using a wireless or serial communication method, so that the information can be forwarded to the PC.

Step Pad Software (bottom of Figure 2)

This software block performs the same functionality as the Handheld Device Software except that the Step Pad will not influence the pointer location at all, so only gamepad movement 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 movement and pointer location information from the other devices and send it to the PC over 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 merged together so that all movements are recorded. The "Translate Commands to USB" block translates the collected information into USB bytes that can be sent to the PC. Finally, the last block puts the commands into the USB API buffers so that the USB ISR can send the data to the Host PC when requested.

Software System Requirements

- Since the USB protocol sends out data packets every 1 ms, the major requirement for the software is that all data collection, processing, and translation into the USB protocol be completed within this 1ms period (including the intersystem communication). This will ensure that updated information is always transferred to the PC and that the worst case latency will be limited to 1 ms. However, since human reaction time is considerably greater than 1 ms and standard monitor refresh rates are approximately 60 Hz (16.7 ms), an update period of up to 10 ms will be acceptable for computationally intensive data.

- Assuming that all calculations can be completed within 0.5 ms, this leaves 0.5 ms for the data transmission between systems. The initial estimations are that at least 2 bytes per system will need to be sent every 0.5 ms meaning that both the Handheld Device and the Step Pad subsystems must transfer their data in 0.25 ms. Initially, assuming no protocol overhead, the data throughput would need to be at least 64 Kbits/s (2 bytes/.25 ms). In reality, protocol overhead will be present and the throughput will need to be much larger, but only the accelerometers and gyroscopes need to send their data at such a large bandwidth. The Step Pad's status will not change this quickly therefore a time-triggered or event-driven

communication method can be implemented where data is only set every 10 ms or just when new information is available.

- We will implement our system by emulating a USB gamepad which will interact well with existing PC games due to the gamepad's inherent in-game functionalities, such as dedicated crouch and jump commands.

Previous Work

Figure 3 lists the patents and standards that are applicable to the design of the final product. No patent was found that exactly matched this project's proposed system design but there are multiple patents on similar motion sensing controllers. However, no patent seemed to restrict the final design of our project if it were to become a consumer product after further development. The standards that are listed correspond to the proposed standardized communications protocols that will be utilized in the final design.

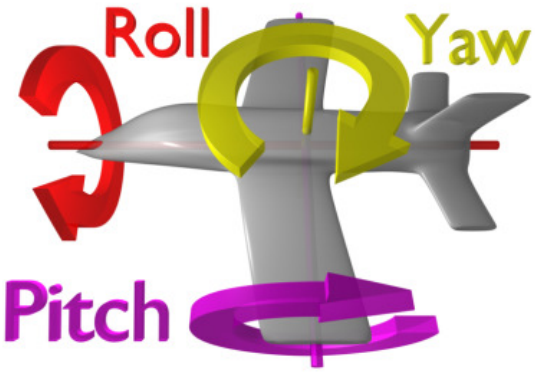
Figure 3: Related Patents and Standards

Patent/Standard Number	Patent/Standard Description
USB 2.0	Universal Serial Bus (www.usb.org)
IEEE 802.15.4	ZigBee Wireless Network Protocol
US Patent 5139261	Foot-actuated computer game controller serving as a joystick
US Patent 6545661	Video game system having a control unit with an accelerometer for controlling a video game
US Patent 4514600	Video game hand controller
US Patent 6902483	Handheld electronic game device having the shape of a gun
11/313,050 (application number)	Advanced video controller system

Analytical Evaluation

This project's motion sensing HID is essentially a stationary Inertial Navigation System (INS). An INS is a computer based platform that uses motion-sensing devices to continuously calculate velocity and position by integrating the data received from the motion sensors. For our stationary platform, no linear accelerations will need to be integrated because the user's position is assumed to be fixed on the step pad. However, angular accelerations such as pitch and yaw will be very important to the on-screen pointer location. Since gyroscopes measure angular acceleration, a 2-axis gyroscope will be used to measure the pitch and yaw of the HID system.

Figure 4: Roll / Pitch / Yaw Picture from Wikipedia [2]



Gyroscopes measure angular acceleration in degrees/second; so in order to find information about the system's position the data must be integrated once. This integration will introduce integration drift; this means that small errors in the measurement of angular acceleration will accumulate over time and eventually compound into a large error in position. This position measuring technique is open loop and the loop must be closed through some sort of augmentation to compensate for the accumulated error. Since our inertial system is operating around a fixed equilibrium position, the linear acceleration components are not being used. This means that true angular measurements can be made using a 2-axis accelerometer (when the system is relatively stable) which can be used to close the loop and compensate for drift on the pitch axis. This technique cannot be used for the yaw axis because it is parallel to the acceleration vector of gravity of the reference system. So in order to make the yaw axis' position measuring system closed loop, we will use an electronic compass that provides absolute heading information. If the compass is not tilt compensated, then yaw axis drift can only be corrected when the HID handheld device is relatively level in the reference system (parallel to the floor).

Finally, in order to measure angles relative to the reference system (the earth's surface), the gyroscope data must be mathematically transformed from HID co-ordinate system to the earth's three dimensional reference co-ordinate system. This will involve trigonometric functions and math intensive routines which could potentially bog down the processor if a mathematical co-processor is not utilized.

Current Progress

Most of the lab time has been dedicated to hardware and motion sensing technology research. The initial hardware setup has been identified, and the fundamental theories for determining absolute cursor position have been documented along with their corresponding error correction methods.

In order to determine the gyroscope sensitivity that was required initial tests had to be run. Four tests were conducted with two variables being varied. For two of the tests, the individual's arms were moved through 180° and for the other two just 90°. Also, for two of the tests the individual used only one arm and for the other two they used both arms. The results can be seen in Figure 5 on the next page. These tests revealed that the average person can move a handheld device at a maximum of 350-500 degrees/sec (while holding a stopwatch). Since humans can move a handheld device up to 500 degrees/sec, the gyroscopes must be able to measure this movement without saturation. The tests were run with the individual holding an object that weighs less than a pound and only the arm was moving. However, when this system is used, the handheld device will weigh between 5-10 lbs and upper body movement will be required; both of which will slow the user's movement. Using this information, the sensors only need to be able to measure up to approximately 250-300 degrees/sec. Also, since the USB protocol sends packets every 1ms, the sensors' readings must update at least this fast or faster to reduce latency.

Figure 5: Gyroscope Maximum Sensitivity Test

1 Arm 180°	Time (s)		2 Arms 180°	Time (s)
	0.39			0.43
	0.35			0.39
	0.35			0.32
	0.39			0.35
	0.36			0.37
AVG °/s	489.1		*AVG °/s*	483.9
1 Arm 90°	**Time (s)**		**2 Arms 90°**	**Time (s)**
	0.28			0.29
	0.28			0.25
	0.21			0.32
	0.25			0.21
	0.27			0.26
AVG °/s	348.8		*AVG °/s*	338.3

The last lab period was dedicated to converting Silicon Lab's C8051F340 USB mouse example into a USB keyboard device. Currently, Windows recognizes the system as a USB keyboard, but it is unable to send any characters to the PC. Once the USB keyboard is working, converting it to a USB gamepad will be simple because the same conversion process will be implemented.

Schedule

Here is the proposed schedule for spring semester EE452. This schedule has strict deadlines that must be met in order for the project to be completed by the end of the semester.

Figure 6: Schedule

	Task Name	Duration	Start	Finish	Predecessors	Resource
1	USB Gamepad Conversion	5 days?	Thu 1/22/09	Wed 1/28/09		Both
2	Sensor Interface / Setup	5 days?	Thu 1/29/09	Wed 2/4/09	1	Both
3	Sensor Coding / Testing	5 days?	Thu 2/5/09	Wed 2/11/09	2	Chris
4	Step Pad Hardware	5 days?	Thu 2/5/09	Wed 2/11/09		Weston
5	Step Pad Software	5 days?	Thu 2/12/09	Wed 2/18/09	4	Both
6	Frame of Reference Equation/Algorithm	5 days?	Thu 2/19/09	Wed 2/25/09		Both
7	Frame of Reference Code	5 days?	Thu 2/26/09	Wed 3/4/09	6	Chris
8	FPU Co-processor 1	5 days?	Thu 2/26/09	Wed 3/4/09		Weston
9	FPU Co-processor 2	5 days	Thu 3/5/09	Wed 3/11/09	8	Both
10	Error / Drift Correction Algorithm / Code 1	2 days?	Thu 3/12/09	Fri 3/13/09		Both
11	Spring Break	8 days?	Mon 3/16/09	Wed 3/25/09		Both
12	Error / Drift Correction Algorithm / Code 2	5 days?	Thu 3/26/09	Wed 4/1/09		Both
13	Integration of Individual Subsystems	5 days?	Thu 4/2/09	Wed 4/8/09		Both
14	Catch-up / Implement Wireless	5 days?	Thu 4/9/09	Wed 4/15/09		Both
15	Testing	5 days?	Thu 4/16/09	Wed 4/22/09		Both
16	Prepare for Presentation and Final Report	5 days?	Thu 4/23/09	Wed 4/29/09		Both
17	Presentation / Report Due	5 days?	Thu 4/30/09	Wed 5/6/09	16	Both

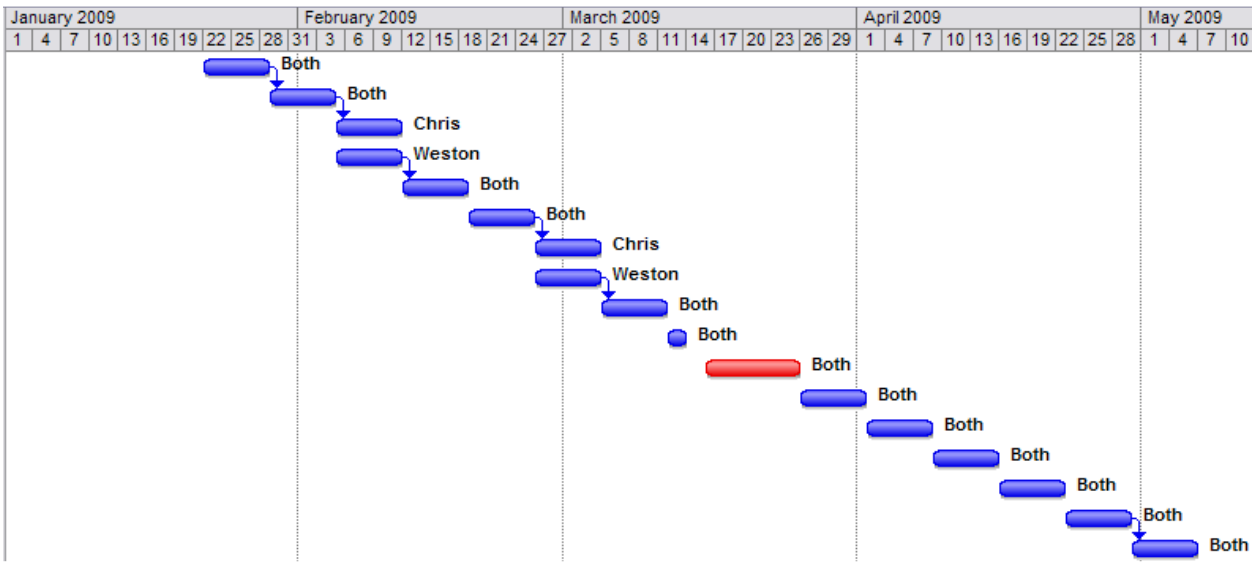


Figure 7: Division of Labor

Division of labor	
Task	Partner responsible
Hardware Research	Weston
USB Research	Christopher
Step Pad Interface / Code	Weston
Sensor Interface / Code	Christopher
Frame of Reference Code	Christopher
Error/Drift Checking Code	Weston
Conversion to USB Gamepad	Both
Testing	Both
Web-site	Weston

Equipment List

Figure 8 lists all the equipment that will be needed to continue for the fall semester.

Figure 8: Equipment List

Equipment	Part Description	Quantity	≈ Cost	Where
Computer	With USB and Game	1	$0.00	Personal Laptop
USB Board	SiLabs C8051F340	1	$99.00	In Lab*
Main Board	SiLabs C8051F120	1	$99.00	In Lab*
Step Pad	PlayStation 2 DDR Dance Pad	1	$15.00	www.amazon.com
FPU Co-Processor	Micromega uM-FPU v3.1 COM-08129	1	$20.00	www.sparkfun.com
Gyro + Accelerometer	IMU 5 Degrees of Freedom SEN-00741	1	$110.00	www.sparkfun.com
Electronic Compass	Compass Module - HMC6352 SEN-07915	1	$60.00	www.sparkfun.com
Level Converter	Logic Level Converter BOB-08745	1	$2.00	www.sparkfun.com
Mini Push Button	Mini Push Button Switch COM-00097	5	$1.75	www.sparkfun.com
		Total Price:	$406.75	

*items are already available in the lab and will not contribute to the cost of funding this project

If time permits, further equipment will be purchased to implement the wireless functionality. Also, higher resolution gyroscopes may be needed to detect slight changes in pitch and yaw, but this will not be known until some initial tests have been run.

Conclusion

The most difficult part of the project will be the translation of accelerometer and gyroscope readings into absolute on-screen position, while reducing the influence of noise, saturation, and drift due to accumulated error. Also, the device must emulate a USB gamepad and Silicon Labs only provides one HID example, a mouse example, for their C8051F340 USB board; therefore, a gamepad must be coded from the USB mouse example. 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] GeneSys Engineering Department, Inertial Sensors and Systems An Introduction,
http://www.genesys-offenburg.de/genesyse.htm

[4] D. Schertz, EE565 Fall 07 Lectures Notes 20 – 24, Bradley University

[5] Device Class Definition for Human Interface Device (HID) Version 1.11
http://www.usb.org/developers/hidpage/

[6] HID Usage Tables Version 1.12
http://www.usb.org/developers/hidpage/