Accelerometer Mouse

Chris Wittmier

ET471 Project Description
Professor Todd Morton
Western Washington University
2004 – 2005
**Introduction**

Since revolutionizing the PC’s user interface two decades ago, the mouse has gradually evolved from a single-button box on wheels into the multitude of hi-tech, ergonomic pointing devices that we know today. In recent years, the affordability of low-g accelerometers has made possible a dramatic departure from conventional mouse operation. Using an accelerometer, I propose the development of a tilt-sensing wireless mouse. Operable in midair with no external motion sensors, the accelerometer mouse will offer freedom from the traditional limitations imposed by mousepads and cords.

**Operation and User Interface**

Physically, the accelerometer mouse will encompass two sections: a handheld control unit (referred to as the “transmitter” for clarity), and a base station (“receiver”). Illustrated in Figure 1, the transmitter will be housed within a modified joystick grip. Dimensioned to comfortably fit within the user’s palm, the transmitter will include a

---

**Figure 1: Transmitter Unit (Maximum Dimensions)**

---

2
detachable Velcro hand strap, allowing for keyboard use without setting the transmitter down. During keyboard use or momentary inactivity, the user may disable transmitter controls with a single push of the transmit-toggle button, located near the thumb. Pressing the transmit-toggle again will resume transmitter operation. As shown in Figure 2, the transmitter will be operated in an upright position, though optional modification may later allow the user to mechanically rotate the accelerometer for palm-down operation.

Also illustrated in Figure 2, the mouse will respond to tilt variance on a 2-dimensional plane oriented parallel with the floor. Tilt variance response specifies that the cursor will move only while the user’s hand is in motion, unlike a joystick which continues cursor movement if held at a fixed angle. Tilt vectors and button status will be broadcast to the
receiver unit, pictured in Figure 3. Plugged into a PC USB port, the receiver may be positioned up to 50 feet away from the transmitter. After processing the transmitted data, the receiver will report cursor movement and button status to the PC.

Typical installation on a PC running Windows XP will take no more than five minutes. Upon plugging the receiver into the PC’s USB port, the Windows Add New Hardware Wizard will prompt the user to select a device driver, which will be installed from disk. The Windows Device Manager will recognize and list the mouse as a Serial Intellimouse. The user will install two AAA batteries in the transmitter unit and switch the transmitter on, whereupon mouse operation will begin.

![Figure 3: Receiver Unit (Maximum Dimensions)](image)
**Functional Description**

Diagrammed in Figures 4 and 5 respectively, the functional transmitter and receiver systems will be discussed in terms of their major components.

**Inputs**

The transmitter will include two click-buttons and a scroll-wheel, as previously illustrated in Figure 1. Tilt will be tracked using Analog Devices’ ADXL202E dual-axis accelerometer. Confined to a tiny surface-mount IC, the ADXL202E is capable of sensing a 0-2g range of force, and provides pulse-width modulated outputs. Initially, I envisioned the mouse responding to both tilt and lateral movement. However, after basic testing of the accelerometer, I concluded that this would not reasonably fit within the scope of this project. The accelerometer is unable to distinguish between static...
gravitational force (which would be interpolated), and dynamic lateral force (which would be integrated). Isolating the two force origins would require a gravitational reference, possibly provided by a second accelerometer in alternate configuration. Besides the additional cost and mounting difficulty entailed, software complexity would be conservatively tripled. I chose to track only tilt, by far the easier force to track. The 0-1g force of gravity tends to dominate the accelerometer’s output response, and provides motional data that can be vastly oversampled even at modest sensor bandwidth.

Although the ADXL202E also provides analog outputs, the PWM outputs are favored for their relative noise immunity, and the ease with which they are decoded using the microcontroller’s input capture timing.

Figure 5: Receiver Unit Functional Block Diagram
**Microcontrollers**

Balancing ideal and practical merits, both microcontrollers will be members of Motorola’s Star12 family. The full-featured MC9S12DP256 will be employed in the receiver, making use of its dual SCI channels. The scaled-down MC9S12C32 will be used in the transmitter. Along with reduced physical dimensions, the ‘C32 incorporates several important power-conserving capabilities. Operating at 3.3V, the ‘C32 has a tunable PLL circuit for performance/power balancing, and is capable of wakeup from Stop and Wait modes using dedicated I/O port bits. Departmentally, both MCU’s are supported by the availability of development tools, as well as curricular material and faculty experience. These practical advantages factored heavily into selection of the two Star12 variants, which admittedly overshoot the design requirements of this project. Optimally, using a pair of 8-bit microcontrollers would provide a more cost-effective solution. Native support of USB communications would have made Motorola’s 68HC908JB16 my ideal choice for the receiver. The 68HC08QY4 would have been ideally suited for transmitter use, featuring diminutive size and power consumption, along with an internal crystal oscillator. Cypress and Atmel offer microcontrollers which are arguably better equipped for mouse applications than the aforementioned HC08’s. However, familiarity with Motorola’s products led me to strongly favor their use.

**Transmission**

Wireless transmission will be simplified with the use of matching RF SAW Tx/Rx modules operating at 433MHz. The transmitter’s Tx module will send asynchronous data packets from the microcontroller’s SCI, which will be received and relayed to the SCI of the receiver unit’s microcontroller. The RF module pair is intended as a ‘complete
wireless solution;’ both modules include internal crystal oscillators, signal amplification/mixing, and OOK modulation/demodulation. The mouse will transmit input data that has been processed as minimally as possible, leaving the majority of calculations to the receiver microcontroller. This will presumably reduce processor current draw in the transmitter unit, where battery supplied power comes at a premium.

PC Interface

Implementation of a USB interface presents a choice between two options. The more ‘elegant’ and commercially popular of the two would be selection of a microcontroller with USB communications support. This route requires researching the relatively complex workings of USB protocol, with which I have no prior experience. As mentioned in the discussion of microcontrollers, this would involve selection of an MCU lacking local development tools and technical support. As such, a standalone peripheral controller from FTDI will be used instead. This controller will handle all translation between the microcontroller’s SCI (RS-232 format) and the PC’s USB port. Besides the added size and cost of additional hardware, the major drawback of this adapter is that it requires installation of a virtual COM port (provided free by FTDI) on the host PC. On the other hand, use of FTDI’s translator adds a certain degree of versatility. The receiver will already be communicating under RS-232 serial protocol, allowing adaptation for use with older PC’s by simply adding an RS-232 line driver and connector.

The accelerometer mouse will emulate a Microsoft serial Intellimouse, sending data in 4-byte packets. The first three bytes will follow standard serial mouse protocol, containing button status, X-axis cursor speed, and Y-axis cursor speed respectively. The fourth byte contains scroll wheel motion.
**Power Supply and Management**

The receiver unit will be powered by the +5V USB port supply. With a maximum output of 500mA, the USB power bus will easily supply the 101mA drawn by the receiver unit under worst-case conditions. Transmitter power will be supplied by two AAA batteries, using a DC-DC step-up converter to provide a +3.3V output. During worst-case conditions, the transmitter will deplete a typical pair of 1150mAH alkaline cells in approximately 25 hours. This estimate assumes that every component of the transmitter remains on and is dissipating maximum power. To ensure that this scenario never actually persists for an extended period, aggressive power management will be implemented through software. A routine will constantly monitor transmitter activity or lack thereof, transitioning between three power modes as specified in Figure 6.

![Figure 6: Transmitter Power Management](image)
Software Description

Software will be written primarily in assembly, as I believe this will allow me to code as efficiently as possible. Depending on my experience in Winter Quarter’s Embedded Systems course, I may opt to save time by writing some of the longer modules in C. Software for both processors will be divided into modules as listed below.

Transmitter Modules

MKERNAL: A straightforward time-slice scheduler
GETBUTTON: Debounces and reads button states on PORTP
GETPULSE: Obtains pulse widths from both input-capturing timer channels
TXER: Organizes serial bytes from inputs, writes to SCI Tx
PMANAGE: Monitors activity, sets power mode

Receiver Modules

RKERNEL: Also a time-slice scheduler, unless PC interface mandates otherwise
RMAIN: Initializes receiver, manages requests from PC
RXER: Reads input data packets from mouse received from SCI Rx
CALCSPEED: Interpolates and linearizes tilt data to calculate cursor speed
PCCOM: Writes formatted 4-byte packets to SCI

Development Plan

During fall quarter, I have been primarily concerned with researching the parts and tools required to develop the mouse. Additionally, I have conducted some basic testing of the ADXL202E to better understand the input signals that must be interpreted. Beginning winter quarter, a formal schedule of progress will be put into effect, which is expected to present a steady and considerable workload for the remainder of the year.
Although most testing and construction equipment will be available at home, the majority of development will be completed in the ET340 lab. Software will be written with CodeWright and loaded using the Noral flex debugger. Hardware debugging will be done primarily on the Hewlett Packard mixed signal scopes. A port monitoring utility will be used to analyze the content of packets received by the PC.

**Demonstration**

All circuits will be soldered point-to-point on generic pad-per-hole protoboards. Each board will be cut to size and mounted to the chassis using appropriate screws and spacers. Aside from the receiver unit’s USB cable, there will be no external wires or circuitry. The mouse will be ideal for hands-on exposition, allowing users to browse or use applications on any PC with no need for instruction. To highlight the project’s key features, I will design a small promo poster to accompany the mouse during demonstration. In preparation for the inevitable, the mouse will be designed and tested to easily withstand a five foot drop onto a hard surface.

**Development Schedule**

Development will be paced in accordance with the following tentative schedule. Listed tasks are to be completed by the end of the indicated week.

**Winter 2004**

Week 1: Finalize part models, order all parts and development tools

Week 2: Design and construct accelerometer PCB

Week 3: Begin transmitter physical construction

Week 4: Finish transmitter construction, less MCU PCB

Week 5: Begin transmitter software development
Week 6: Complete input acquisition code
Week 7: Complete transmitter communications
Week 8: Complete remaining mouse software
Week 9: Begin receiver software
Week 10: Complete input signal receive/processing

**Spring 2005**

Week 1: Design USB interface circuit, begin interface software
Week 2: Complete PC interface, mouse works hardwired. Hardware Review.
Week 3: Catch up, resolve outstanding issues
Week 4: Design and construct wireless stages, test
Week 5: Resolve bugs
Week 6: ‘Beta test’-refine code
Week 7: Continue beta
Week 8: Resolve anything yet unresolved
Week 9: Code Review
Week 10: Project Delivery
Specifications

System Specifications

Resolution:  1 degrees of tilt per axis

Wireless Transmission Freq:  433MHz License-free ISM band

Mouse Protocol:  Microsoft Serial Intellimouse compatible

USB Compatibility:  USB 1.1 & USB 2.0 Compatible

Transmission Range:  Up to 50 ft.

Operating Temperature Range:  32-110 °F

Transmitter Specifications

Power

Maximum Supply Current Drawn:  46.12mA

Estimated Worst-Case Battery Life:  25 hours

Enclosure

Maximum Physical Dimensions:  4” x 2” x 1.25”

Maximum Weight:  Not to exceed 1/2 lb.

Receiver Specifications

Power

Maximum Supply Current Drawn:  101.04mA

Enclosure

Maximum Physical Dimensions:  5” x 4” x 1.5”

Maximum Weight:  Not to exceed 1.5 lbs.
Preliminary Parts List

Transmitter

<table>
<thead>
<tr>
<th>Item</th>
<th>Max Iq</th>
<th>Quantity</th>
<th>Price</th>
<th>Source</th>
<th>Lead Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller (MC9S12C32)</td>
<td>35mA</td>
<td>1</td>
<td>10.65</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>1 week</td>
</tr>
<tr>
<td>Dual Axis Low-g Accelerometer (ADXL202JE)</td>
<td>1mA</td>
<td>1</td>
<td>12.75</td>
<td><a href="http://www.analog.com">www.analog.com</a></td>
<td>In hand</td>
</tr>
<tr>
<td>433MHz RF SAW Tx (Linx)</td>
<td>6mA</td>
<td>1</td>
<td>6.90</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>1 week</td>
</tr>
<tr>
<td>3V-3.3V DC-DC Converter (TPS60123)</td>
<td>90µA</td>
<td>1</td>
<td>2.36</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>1 week</td>
</tr>
<tr>
<td>Battery Case – 2x AAA</td>
<td>N/A</td>
<td>1</td>
<td>.78</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>1 week</td>
</tr>
<tr>
<td>16MHz Crystal Oscillator</td>
<td>25µA</td>
<td>1</td>
<td>2.78</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>1 week</td>
</tr>
<tr>
<td>¼” Fixed Mount Antenna</td>
<td>N/A</td>
<td>1</td>
<td>2.03</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>1 week</td>
</tr>
<tr>
<td>Resistors, various</td>
<td>4mA</td>
<td>≈ 12</td>
<td>1.20</td>
<td>ET340</td>
<td>None</td>
</tr>
<tr>
<td>Capacitors, various</td>
<td>N/A</td>
<td>≈ 8</td>
<td>.80</td>
<td>ET340</td>
<td>None</td>
</tr>
<tr>
<td>N/O Pushbutton Switch</td>
<td>N/A</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>In Hand</td>
</tr>
<tr>
<td>SPST Slider Switch</td>
<td>N/A</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>In Hand</td>
</tr>
<tr>
<td>Mouse Wheel</td>
<td>N/A</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>In Hand, Salvage</td>
</tr>
<tr>
<td>Joystick Grip</td>
<td>N/A</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>In Hand, Salvage</td>
</tr>
<tr>
<td><strong>Transmitter Subtotal</strong></td>
<td>46.12mA</td>
<td></td>
<td>$40.25</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Receiver

<table>
<thead>
<tr>
<th>Item</th>
<th>Max Iq</th>
<th>Quantity</th>
<th>Price</th>
<th>Source</th>
<th>Lead Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller (MC9S12DP256B)</td>
<td>65mA</td>
<td>1</td>
<td>24.74</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>1 week</td>
</tr>
<tr>
<td>433MHz RF SAW Rx (Linx)</td>
<td>7mA</td>
<td>1</td>
<td>13.79</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>1 week</td>
</tr>
<tr>
<td>RS-232→USB Peripheral Controller (FTDI)</td>
<td>25mA</td>
<td>1</td>
<td>5.80</td>
<td><a href="http://www.saelig.com">www.saelig.com</a></td>
<td>1 week</td>
</tr>
<tr>
<td>USB Type-B Connector</td>
<td>N/A</td>
<td>1</td>
<td>1.02</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>1 week</td>
</tr>
<tr>
<td>16MHz Crystal Oscillator</td>
<td>25µA</td>
<td>1</td>
<td>2.78</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>1 week</td>
</tr>
<tr>
<td>6MHz Crystal Oscillator</td>
<td>12µA</td>
<td>1</td>
<td>2.78</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>1 week</td>
</tr>
<tr>
<td>Resistors, various</td>
<td>4mA</td>
<td>≈ 10</td>
<td>1.00</td>
<td>ET340</td>
<td>None</td>
</tr>
<tr>
<td>Capacitors, various</td>
<td>N/A</td>
<td>≈ 6</td>
<td>.60</td>
<td>ET340</td>
<td>None</td>
</tr>
<tr>
<td><strong>Receiver Subtotal</strong></td>
<td><strong>101.04mA</strong></td>
<td></td>
<td><strong>$52.51</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PROJECT TOTAL</strong></td>
<td></td>
<td></td>
<td><strong>$92.76</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>