1. Introduction

The present document shows step by step how to easily build an interesting pointing device: the wireless tilt mouse, that allows to control the mouse’s cursor on the PC screen through the tilt of the board itself. The analog data from a 3-D accelerometer and from two push-buttons are acquired, converted and radio transmitted by an XBee module using the standard ZigBee protocol. On the receiver side there is another XBee module connected to the PC through an EasyUSB module, that allows both file storing and communication on the same interface and it’s seen by the PC as a common mass storage device. In practice there is no need to install any application on the PC: the software that communicates with the XBee and emulates the mouse functions is resident inside the EasyUSB itself and it is automatically recognized by the PC as an USB drive. The accelerometer/transmitter board is powered by AA batteries.

This project aims to be an open source hardware/software platform, which can be used for learning about getting accelerometer’s data, configuring XBee modules and managing custom USB peripherals. All the hardware schematics and well commented software source codes are available on the attached project files.

In order to design a fast and easy-to-build platform, only breakout boards and modular components are used. Specific skills like SMD soldering and embedded software programming are not necessary.

2. Design

All the necessary materials can be purchased online from SparkFun Electronics. Following the links and in the attached documents there are plenty of useful information about each one of these components.

XBee Explorer Regulated
http://www.sparkfun.com/products/9132

XBee 1mW Chip Antenna
http://www.sparkfun.com/products/8664

Triple Axis Accelerometer Breakout - ADXL335
http://www.sparkfun.com/products/9269

5V DC to DC Step Up - 2xAA
http://www.sparkfun.com/products/8249

Breakout Board for USB Mini-B
http://www.sparkfun.com/products/9966

EasyUSB
http://www.sparkfun.com/products/10155

Mini Push Button Switch – SMD
http://www.sparkfun.com/products/8720
2.1. Building Steps

2.1.1. Hardware Connections
- Connect the boards: solder wires according to the hardware schematics and wiring diagrams. Note that XBee Chip Antenna modules should not be soldered: they fit into the XBee Explorer Regulated socket.

**Transmitter parts:**

<table>
<thead>
<tr>
<th>5V DC to DC Step Up</th>
<th>Triple Axis Accelerometer Breakout - ADXL335</th>
<th>XBee Explorer Regulated and XBee 1mW Chip Antenna</th>
<th>2 Mini Push Buttons</th>
</tr>
</thead>
</table>

**Receiver parts:**

<table>
<thead>
<tr>
<th>Breakout Board for USB Mini-B</th>
<th>EasyUSB</th>
<th>XBee Explorer Regulated and XBee 1mW Chip Antenna</th>
</tr>
</thead>
</table>

2.1.2. Transmitter Config
- Put an XBEE module on the RECEIVER socket and plug the USB cable into the PC. Wait for the automatic installation of the device as a common USB mass storage device.
- Copy the xbee_config.exe console into the USB device. Run xbee_config.exe console and press T to configure the XBEE module as TRANSMITTER. Press ESC to exit.
- Take off the XBEE module from the RECEIVER socket and place it into the TRANSMITTER socket.

2.1.3. Receiver Config
- Put the other XBEE module on the RECEIVER socket. Run xbee_config.exe console and press R to configure the XBEE module as RECEIVER. Press ESC to exit.
- Copy the tilmouse.exe program into the USB device.

The wireless tilt mouse transmitter and receiver are done! In order to use it, just run the tilmouse.exe console on the receiver and plug the batteries on the transmitter. The console, other than emulate the mouse’s functions, also prints on the screen the data acquired from each accelerometer’s axis and push buttons.
2.2. Schematics

TRANSMITTER

RECEIVER
2.2.1. Schematics references

<table>
<thead>
<tr>
<th>Comment</th>
<th>Description</th>
<th>Designator</th>
<th>LibRef</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V DC-DC</td>
<td>5V DC to DC Step Up - 2xAA</td>
<td>B1</td>
<td>5V DC-DC</td>
<td>1</td>
</tr>
<tr>
<td>USB MINI</td>
<td>USB mini connector</td>
<td>P1</td>
<td>USB MINI</td>
<td>1</td>
</tr>
<tr>
<td>SW-PB</td>
<td>Push button</td>
<td>S1, S2</td>
<td>SW-PB</td>
<td>2</td>
</tr>
<tr>
<td>XBee Regulated</td>
<td>XBee Explorer Regulated Board</td>
<td>U1, U3</td>
<td>XBee Regulated</td>
<td>2</td>
</tr>
<tr>
<td>ADXL335 Breakout</td>
<td>Accelerometer Breakout Board</td>
<td>U2</td>
<td>ADXL335 Breakout</td>
<td>1</td>
</tr>
<tr>
<td>EASY USB</td>
<td>USB interface module</td>
<td>U4</td>
<td>EASY USB</td>
<td>1</td>
</tr>
</tbody>
</table>

2.3. Wirings

2.3.1. Transmitter
3. Components

In this section the main components of the project will be described regarding their main features, functions and technical characteristics.

For further information about operation modes, addressing, supported commands and other characteristics please refer to the available reference datasheets, applications notes and software development kits.

3.1. XBee Module

The XBee modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices. The modules operate within the ISM 2.4 GHz frequency band. It was designed to mount into a receptacle (socket) and therefore does not require any soldering when mounting it to a board.

The XBee modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART. They support ADC (Analog-to-digital conversion) and digital I/O line passing and some of their pins support multiple functions, therefore for basic functionalities it is not needed any additional microprocessor.
Here are listed the AT commands (also implemented in the *xbee_config* console application software) that should be sent in order to config the two XBee modules of the project:

On the transmitter side, the XBee will also acquire the analog data from the accelerometer and push buttons and send them every 100ms; on the receiver side, the XBee will receive the data from the transmitter and send them to the EasyUSB module and then to the PC.

**Transmitter:**
Send the command sequence character to enter in XBee Command Mode: +++
Restore default parameters: ATRE
Set transmitter destination address: ATDL = 0x1234
Set transmitter source address: ATMY = 0x5678
Set D0 as analog input: ATD0 = 2
Set D1 as analog input: ATD1 = 2
Set D2 as analog input: ATD2 = 2
Set D3 as digital input: ATD3 = 3
Set D4 as digital input: ATD4 = 3
Enable pull up on digital inputs D3 and D4: ATPR = 0x03
Set sample rate to 100ms: ATIR = 0x64
Write parameters to non-volatile memory: ATWR

**Receiver:**
Send the command sequence character to enter in XBee Command Mode: +++
Restore default parameters: ATRE
Set receiver destination address: ATDL = 0x5678
Set receiver source address: ATMY = 0x1234
Set receiver input address: ATIA = 0x5678
Enable uart output: ATIU = 1
Set P0 in order to drive the RSSI LED according to signal strength: ATP0 = 1
Write parameters to non-volatile memory: ATWR

Once the modules are configured and powered, every 100ms the transmitter will send to the receiver a packet with the following information:

**RX (Receive) Packet: 16-bit Address**

API Identifier Value: 0x81
When the module receives an RF packet, it is sent out the UART using this message type.

Figure 3-14. RX Packet (16-bit address) Frames

<table>
<thead>
<tr>
<th>Source Address (Bytes 5-9)</th>
<th>RSSI (Byte 7)</th>
<th>Options (Byte 8)</th>
<th>RF Data (Bytes 9-n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSB (most significant byte) first, LSB (least significant last)</td>
<td>0x81</td>
<td>0 (reserved)</td>
<td>Up to 100 Bytes per packet</td>
</tr>
</tbody>
</table>
Where the RF Data bytes have the following structure:

3.2. Accelerometer

The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 g. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration. The user selects the bandwidth of the accelerometer using the CX, CY, and CZ capacitors at the XOUT, YOUT, and ZOUT pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

The ADXL335 uses a single structure for sensing the X, Y, and Z axes. As a result, the three axes’ sense directions are highly orthogonal and have little cross-axis sensitivity. Mechanical misalignment of the sensor die to the package is the chief source of cross-axis sensitivity.
3.2. EasyUSB

The EasyUSB is an embedded Universal Serial Bus interface with solid state memory and external asynchronous serial interface, designed to facilitate the file sharing and direct communication between host and device applications, therefore improving the end-user experience, reducing development costs and time to market.

On the host side, in almost any computer or portable device, no driver or further installation procedure is needed as the USB Mass Storage Class implemented is widely supported by almost any operating system.

On the device, the file system is accessible through a classical Universal Asynchronous Serial Interface. Eventually the software can be distributed inside the embedded memory, rendering it really plug-n-play and usable everywhere.

Furthermore host computer and embedded controller can directly communicate over a dedicated serial link, built using an innovative technique, without implementing the Communication Device Class. In practice it can benefit from both interfaces, mass storage and communication, using only the simple and widespread Mass Storage Class, eliminating the need of problematic composite peripherals or expensive custom driver development and maintenance.

**HOST:**

The host recognizes the EasyUSB interface as a standard mass storage device when it’s plugged in. No further user operation is needed. The UART interface is virtually direct accessed by the host through the library functions, detailed on the EasyUSB API documentation. External formatting is not supported. A software application that demonstrates all the library functions is available.

**DEVICE:**

The EasyUSB is easily controlled by embedded systems through the standard UART interface. When the VBUS is high (USB host is plugged) it outputs the message "Serial Bridge Mode" and acts as a bridge between the UART signals and the library read/write functions. When VBUS is low (USB host is unplugged), it outputs the message "File System Mode" and the embedded processor can access the file system.
4. Software

The application software was developed on Codegear C++ Builder Integrated Development Environment. For further information refer to the commented source code.

The algorithm that describes the working flow of the tiltmouse console application software is illustrated here:
5. References


Contacts

For technical questions, software, documentation and for information about producing, distributing, licensing, please contact the developer:

Email: info@inventionslab.com
Website: http://inventionslab.com

Disclaimer

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. The developer assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, the developer assumes no responsibility for the functioning of not described features or parameters. The developer reserves the right to make changes without further notice. The developer makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does the developer assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Unless specifically provided otherwise, the products are not suitable for, and shall not be used in, automotive applications and are not intended, authorized, or warranted for use as components in applications intended to support or sustain life. No freedom to use patents or other intellectual property rights is implied by the publication of this document.