



- [Home](#)
- [About](#)
- [Wiki](#)
- [Chat](#)
- [Forums](#)
- [Episodes](#)

- [Log in / create account](#)

[MP4](#) [WMV](#) [iTunes](#)

Episode 2x03

From Hak5 Wiki

Contents

- [1 Synopsis](#)
- [2 USB Hacksaw](#)
- [3 An App A Day](#)
- [4 Asterisk on NSLU2](#)
- [5 Adding MIDI capability to a Power Glove and a Speak & Spell](#)
 - [5.1 Background](#)
 - [5.2 Making musical instruments out of just about anything](#)
 - [5.3 A short overview of MIDI](#)
 - [5.4 Speak and Spell: Background](#)
 - [5.5 Speak & Spell: MIDI kit](#)
 - [5.6 Power Glove: Background](#)
 - [5.7 Power Glove: MIDI Interface](#)
 - [5.8 About the guest: who is this guy?](#)
 - [5.9 Further ideas and links](#)

Synopsis

In this episode of Hak5 Darren shows us a USB key, dubbed the USB Hacksaw, that instantly and silently installs on target Windows 2000 or higher computers with guest or higher priviledges which will monitor USB ports for flash drives and automatically download their contents for silent archival and email. Paul brings us an Asterisk setup for our home using a hacked Linksys NSLU2 and some inexpensive SIP phones. TheSoftwareJedi from AnAppADay.com joins us to talk about his project and software. And Wess heads to Toronto and meets with Andrew Barrow to see a Nintendo Power Glove, Speak & Spell, MIDI mashup. Plus this month's trivia, poll, & LAN party details from the lovely and talented Alli.

Special thanks to Ashley Witt for the awesome new Hak5 Theme Song, and Pronoboza for the bumper music and 3D animations, they really makes the episode even more spiffy!

USB Hacksaw

The USB Hacksaw is an evolution of the popular USB Switchblade that uses a modified version of USBDumper, Blat (<http://www.blat.net/>) , Stunnel (<http://www.stunnel.org/>) , and Gmail (<http://www.gmail.com>) to automatically infect Windows PCs with a payload that will retrieve documents from USB drives plugged into the target machine and securely transmit them to an email account. Proof of concept code shows how to deliver the payload instantly with a U3 autorun hack borrowed from the USB Switchblade on Windows 2000 or higher computers running as administrator or guest. Automatic propagation to other USB devices is possible however was not shown on **Episode 2x03**.

- Full article can be seen here: [USB Hacksaw](#)

An App A Day

- See <http://www.anappaday.com>

Asterisk on NSLU2

The idea behind this project is to be a tutorial on how to install Asterisk on an Unslung NSLU2, connect sip phones, enable voice mail notifications and add some simple demonstrations to the system. The point of this project will to be a launch pad of exploring IP Telephony projects based around Asterisk. So this guide will be limited to a simple installation scenario.

- Full article can be seen here: [Asterisk](#)

Adding MIDI capability to a Power Glove and a Speak & Spell

Background

so one night I'm at home and all chillin, listenin to some music and my friend Pronoboza calls me up and is like hey, you know **HAK.5**, right, and I'm like yeah bro, and he's like well they're comin up here to shoot some bits and we're meeting up, you and me should do a bit for their show and I'm like alright dude, and he's like August long weekend dude, and I'm like cool, so I go and bring some stuffs I made, and it's real early in the morning and Prono decides not to do the bit, so I'm like can I do it myself and the **HAK.5** cru is like we have to ask the producer, and they argue and I hear people yelling in another room, then I hear a loud crash, and they come back and everybody's tired and one guy's finger is busted up and they're like okay man, you can do the bit but here's real life, and it's like here, and then here's TV, and it's like an inch higher than real life right, and if you do this bit you

gotta be here at the TV level not all tired like you were partying all night and I'm like you mean alls I gotta do for the bit is be an inch higher than normal and they're like yeah then they look at each other and they're like oh and one more thing, if you do the bit you gotta write yer own show notes too, and I'm like sweet lets go, then they laugh and all hi-five each other and I'm like I don't get it and they're like sucker! and I'm like can we do the bit now and in the end it was FUN and we told a lotta jokes and stuff and after it all they're like have fun writing the show notes sucker! and they laugh hard in this mean way and I still don't get it and then a month goes by...

...and now I get it... show notes are hard work!

(Four-year-old, you say? Feel free to erase this when the real show notes arrive!)

Making musical instruments out of just about anything

What we're looking at in this segment is actually two parts: first we modify a Speak & Spell to accept MIDI messages that will control its speech synthesis, then we check out a homemade MIDI interface for the Power Glove that will generate MIDI messages based on the 3D position of the glove. At the end of the segment we of course connect them both together, making a very cool performance instrument and fantastic modern sendup to some of our favourite childhood toys.

The first hack is actually built from a kit, while the second was designed and constructed from scratch. With inspiration from the segment and a little research the viewer can apply the same principles to interface all sorts of interesting stuff to MIDI; we present some links and further technical information below to start you on your way.

A short overview of MIDI

MIDI stands for Musical Instrument Digital Interface (http://en.wikipedia.org/wiki/Musical_Instrument_Digital_Interface) . It's a simple serial network originally designed in the early 1980s to send note information between electronic musical instruments. Nowadays it's commonly used to send note data from sequencing software running on a PC to outboard instruments and synthesizers, but it can be used to send any kind of data.

On the technical side, MIDI data is sent unidirectionally at 31.25kbps. MIDI messages consist of 2-3 bytes in total (think three-byte packet), with one status byte (kind of like the header) followed by 1-2 data bytes. Because it's simple and transmitted at a low data rate (remember, it's almost 25 years old), very simple modern hardware (ie PIC microcontroller) can decode or encode the MIDI messages, allowing us to make cheap easy interfaces between MIDI and other real-world devices not intended for musical applications.

Speak and Spell: Background

The Speak & Spell is a children's learning toy made by Texas Instruments from about 1978-1988. It was made famous in the movie *E.T. The Extraterrestrial* and became a symbol of the 1980s like the Atari 2600, Masters of the Universe, etc. Technically, It was the first low-cost speech synthesizer, and its main synthesis chip, the TMS0280, is widely regarded as the first ever DSP (digital signal processor). A derivative of the first ever microcontroller, the TMS1000 (originally designed for single-chip calculators) is the processor in the system, looking up words and word fragments from ROM, and driving the keypad and VFD display. The MCU is 4-bit and runs at 300 kHz.

A 1983 *Computers & Electronics* magazine article detailed the construction of a Speak & Spell interface for the Sinclair/Timex ZX-80 computer. Check it out here (<http://members.tripod.com/werdav/speakell.html>)

Speak & Spell: MIDI kit

The Speak & Spell MIDI interface is actually a DIY kit available from [highlyliquid.com/kits/midispeak](http://www.highlyliquid.com/kits/midispeak) (<http://www.highlyliquid.com/kits/midispeak>). We're not affiliated with highlyliquid.com; we just liked their interface. Their kit was cheap, complete, and easy to assemble. It's based around a PIC microcontroller that comes already programmed, so you don't have to know the low-level interfacing of either the MIDI messages or the Speak & Spell. In other words, a great first project.

Unfortunately during the segment, we didn't get a chance to document the assembly or installation of the kit. It's fairly straightforward; the kit gets soldered together, then wired to the Speak & Spell PCB as per the instructions on the site.

Before closing everything up, we set the interface mode and the MIDI channel on the DIP switches. (This is a great time to test the interface before sewing up the patient).

When it's working, we cut a hole for the DIN socket and attach the interface to the inside of the Speak & Spell case. Basic modding skills are required, but nothing too major.

One small thing we ran into here... for the segment we actually acquired three Speak & Spells and two kits. One of the units was the same as the install diagrams at the site; one was close enough to use (same part orientations.. slightly different placement) and the third was a 1978 model (probably first run) and had more memory chips and a different keypad connector. I had to take a look at the Speak & Spell schematics and use a meter to find some of the connection points for the old 1978 model (but ultimately it did work). You can identify these because they have plastic buttons instead of a membrane keypad--if you want the most straightforward installation, avoid the plastic-button models.

Check out some of the other kits at [highlyliquid.com](http://www.highlyliquid.com) (<http://www.highlyliquid.com>); the kits we got were top notch and their other stuff looks great.

Power Glove: Background

The Power Glove was a controller for the original 8-bit NES. Nintendo started selling these in 1989, and unfortunately (because they were hard to use and there were only two games designed to work natively with them) they didn't get extremely popular.

That said, for their time they were technically amazing. They work by transmitting ultrasonic 'pings' from the glove to a sensor array. The glove has two emitters and the array has three receivers; the unit measures the time the pings take to get from each of the emitters to all three of the receivers (six measurements total). From these measurements, the glove's X, Y, and Z position in front of the array can be calculated, as well as the roll (wrist twist). The glove additionally has bend sensors in four of the fingers. It also has a keypad and a complete NES control pad on the glove itself.

The glove has two major modes: controller emulation and high-resolution. Controller emulation allowed the user to enter a 'program' number on the glove keypad, and that would map 'moves' of the glove to plain control pad data (one-byte serial packets), for use with non-glove-specific games. High resolution mode outputs 12-byte packets as follows:

```
<header>,<X>,<Y>,<Z>,<rotation>,<fingers>,<control pad>,<junk>,<junk>,<junk>,<junk>,<junk>
```

Naturally, the high resolution packet is what we're interested in for the interface. In the show closeups with Wess

using the glove, we're only using the X data byte to generate notes; the interface can use any or all of the data in the packet as notes or controllers.

Power Glove: MIDI Interface

The interface basically performs two functions: it sets the glove into high resolution mode at startup and continuously polls it for packets, and it forms that glove data into MIDI messages. There are a several ways we could do this; in this case a PIC microcontroller was used (PIC16F877 at 4MHz). The firmware was written in PICBasic Pro because it's super fast and easy--it took under one day to go from scratch to generating notes with the glove... and it's easy for everyone to follow. The serial communications with the glove is bit-banged, while the MIDI transmission uses the hardware USART of the PIC (via the HSEROUT command). You can see in my source that I have two defines at the top for serial baud rate; this is because I have the debug serial port connected in parallel with the midi port (I think this is mentioned in the segment). I just unrem the appropriate define to choose whether I'm sending MIDI or RS232. The only trick here is that HSEROUT makes an inverted signal for use with an RS232 level shifter; we have to use a hardware inverter to drive the MIDI. I also use the inverter to drive the RS232 port at 5V (this is cheating but generally PC serial ports will receive 0-5V data correctly--an important cheat to remember).

Another small thing that happens with the X, Y and Z data is that out of the power glove, it's twos complement -127..127, so we need to convert it to make MIDI notes directly. The rotation data (I call it roll in the source) is a number from 1-12, representing hours on a clock of wrist roll. The finger data is packed in two-bit pairs for thumb, first, second, and third finger in the finger byte. We don't use finger or roll data in the source as it is, so no conversion is done.

This is not the first Power Glove to MIDI interface; in the early 1990s a commercial interface existed called the Gold Brick.

About the guest: who is this guy?

Andrew has been hacking stuff since an early age. He's built numerous synthesizers and robots, worked as a blacksmith, an auto mechanic, and an engineer. In university he built and raced solar cars across the US and Australia. As a kid he programmed on the Commodore 64, including interfacing a homemade speech synthesizer to the user port.

He got his first MIDI synthesizer in high school and has been experimenting with electronic music since then.

Now an electrical engineer, he works in R&D at a semiconductor company near Toronto. He collects vintage synthesizers and motorcycles.

The glove was built for Pronoboza, a Toronto electronic musician who has done music for [[hak5a}}. The segment was actually filmed at Pronoboza's studio.

Further ideas and links

With inspiration from the episode and understanding of the (very simple) code for the glove interface, we can interface lots of other sensors to use for note or controller input, and add MIDI to some other sound sources. Some ideas I want to try are accelerometers, ultrasonic rangefinders, IR rangefinders, the Spaceball 3dconnexion.com (<http://www.3dconnexion.com/products/3a2.php>) , etc.

A lot of other DIY MIDI projects are out there on the web. Here are some of my favourites; feel free to add any you know of:

Paia Electronics (<http://www.paia.com>) makes a DIY MIDI analog synthesizer called *Fatman*. It's pretty fun to put together and can sound very cool.

uCapps.de (<http://www.ucapps.de>) has designs for a number of MIDI input and output devices using a modular architecture they call *MIDIBox*. Their *MIDIBox SID* uses the Commodore 64 sound chip as a synthesizer under MIDI control--it sounds awesome!

The MIDI technical fanatic's brainwashing centre (<http://www.borg.com/~jglatt/>) has more technical info about the MIDI data format.

Audiomulch.com (<http://www.audiomulch.com/midipic/>) has various information about getting MIDI in and out of PIC microcontrollers and basic stamps.

Retrieved from "http://wiki.hak5.org/index.php?title=Episode_2x03"

Category: Show notes



- Page |
- Discussion |
- View source |
- History
- What links here |
- Related changes |
- Special pages
- | Permanent link

This page was last modified on 3 April 2011, at 10:20. This page has been accessed 28,392 times.

- About Hak5 Wiki |
- Disclaimers |
- Powered by MediaWiki |
- Darren hacked the frak out of Paul Gu's theme