

# NSA Playset: USB Tools

Dominic Spill

Michael Ossmann

Jared Boone

This presentation contains leaked, classified content.

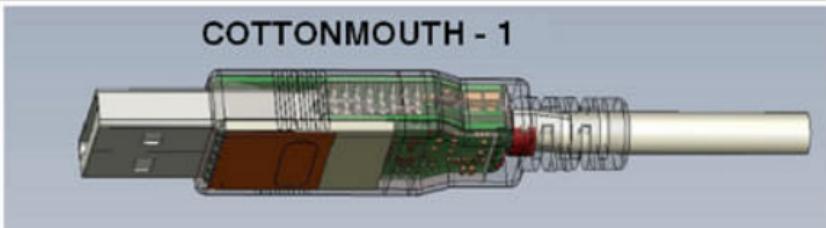


# COTTONMOUTH-I

## ANT Product Data

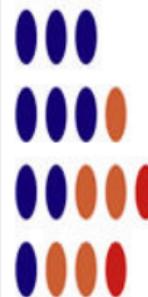
08/05/08

**(TS//SI//REL)** COTTONMOUTH-I (CM-I) is a Universal Serial Bus (USB) hardware implant which will provide a wireless bridge into a target network as well as the ability to load exploit software onto target PCs.



**(TS//SI//REL)** CM-I will provide air-gap bridging, software persistence capability, "in-field" re-programmability, and covert communications with a host software implant over the USB. The RF link will enable command and data infiltration and exfiltration. CM-I will also communicate with Data Network Technologies (DNT) software (STRAITBIZARRE) through a covert channel implemented on the USB, using this communication channel to pass commands and data between hardware and software implants. CM-I will be a GENIE-compliant implant based on CHIMNEYPOOL.

**(TS//SI//REL)** CM-I conceals digital components (TRINITY), USB 1.1 FS hub, switches, and HOWLERMONKEY (HM) RF Transceiver within the USB Series-A cable connector. MOCCASIN is the version permanently connected to a USB keyboard. Another version can be made with an unmodified USB connector at the other end. CM-I has the ability to communicate to other CM devices over the RF link using an over-the-air protocol called SPECULATION.





# COTTONMOUTH-II

## ANT Product Data

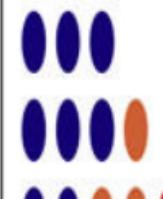
**(TS//SI//REL)** COTTONMOUTH-II (CM-II) is a Universal Serial Bus (USB) hardware Host Tap, which will provide a covert link over USB link into a targets network. CM-II is intended to be operate with a long haul relay subsystem, which is co-located within the target equipment. Further integration is needed to turn this capability into a deployable system.

08/05/08



**(TS//SI//REL)** CM-II will provide software persistence capability, "in-field" re-programmability, and covert communications with a host software implant over the USB. CM-II will also communicate with Data Network Technologies (DNT) software (STRAITBIZARRE) through a covert channel implemented on the USB, using this communication channel to pass commands and data between hardware and software implants. CM-II will be a GENIE-compliant implant based on CHIMNEYPOOL.

**(TS//SI//REL)** CM-II consists of the CM-I digital hardware and the long haul relay concealed somewhere within the target chassis. A USB 2.0 HS hub with switches is concealed in a dual stacked USB connector, and the two parts are hard-wired, providing a intra-chassis link. The long haul relay provides the wireless bridge into the target's network.



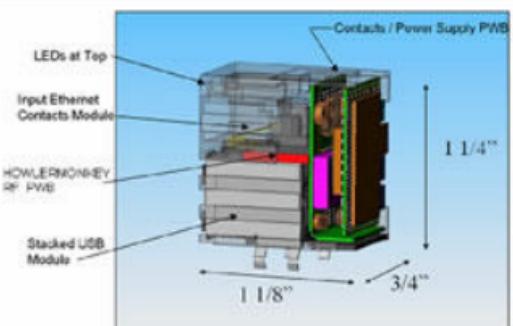


# COTTONMOUTH-III

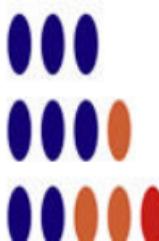
## ANT Product Data

08/05/08

**(TS//SI//REL)** COTTONMOUTH-I (CM-I) is a Universal Serial Bus (USB) hardware implant, which will provide a wireless bridge into a target network as well as the ability to load exploit software onto target PCs.



**(TS//SI//REL)** CM-III will provide air-gap bridging, software persistence capability, "in-field" re-programmability, and covert communications with a host software implant over the USB. The RF link will enable command and data infiltration and exfiltration. CM-III will also communicate with Data Network Technologies (DNT) software (STRAITBIZARRE) through a covert channel implemented on the USB, using this communication channel to pass commands and data between hardware and software implants. CM-III will be a GENIE-compliant implant based on CHIMNEYPOOL.



# **USBProxy**

Low cost USB 2.0

# USBProxy Overview

Open source USB 2.0 proxy library

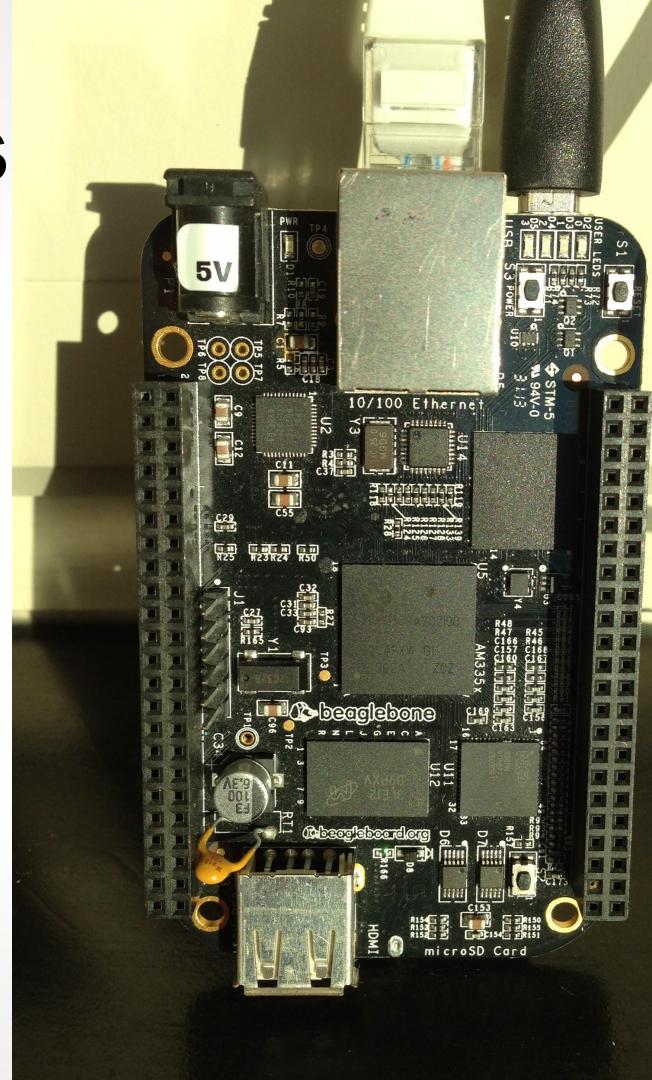
LibUSB and GadgetFS

BeagleBone Black (or similar)

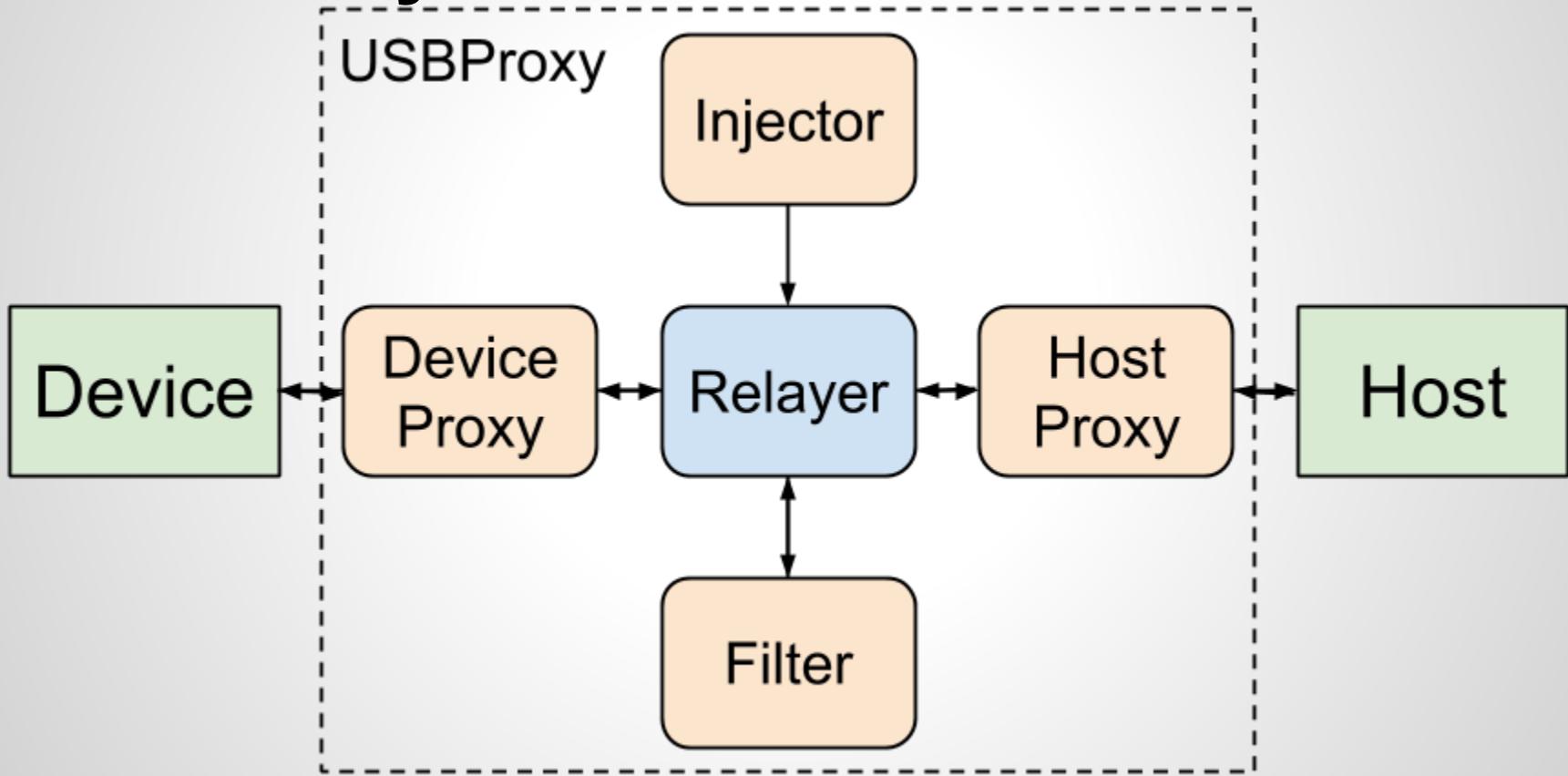
# Hardware requirements

USB device controller

Able to run Linux



# USBProxy Overview



# What's new in USBProxy

Bug fixes

There were a LOT of them

Write plugins in Python

Just like FaceDancer

```
def handle_get_string_descriptor_request(self, num):
    if num == 0:
        # HACK: hard-coding baaaaaad
        d = bytes([
            4,          # length of descriptor in bytes
            3,          # descriptor type 3 == string
            9,          # language code 0, byte 0
            4           # language code 0, byte 1
        ])
    else:
        if idx == 0:
            p_nbytes[0] = 4
            # FIXME: don't hard-code this
            dataptr[0] = c_ubyte(0x04)
            dataptr[1] = c_ubyte(0x03)
            dataptr[2] = c_ubyte(0x09)
            dataptr[3] = c_ubyte(0x04)
            return 0
```

# A word on demos

Keyboards are the calc.exe of USB devices

# FaceDancer support

## Demo

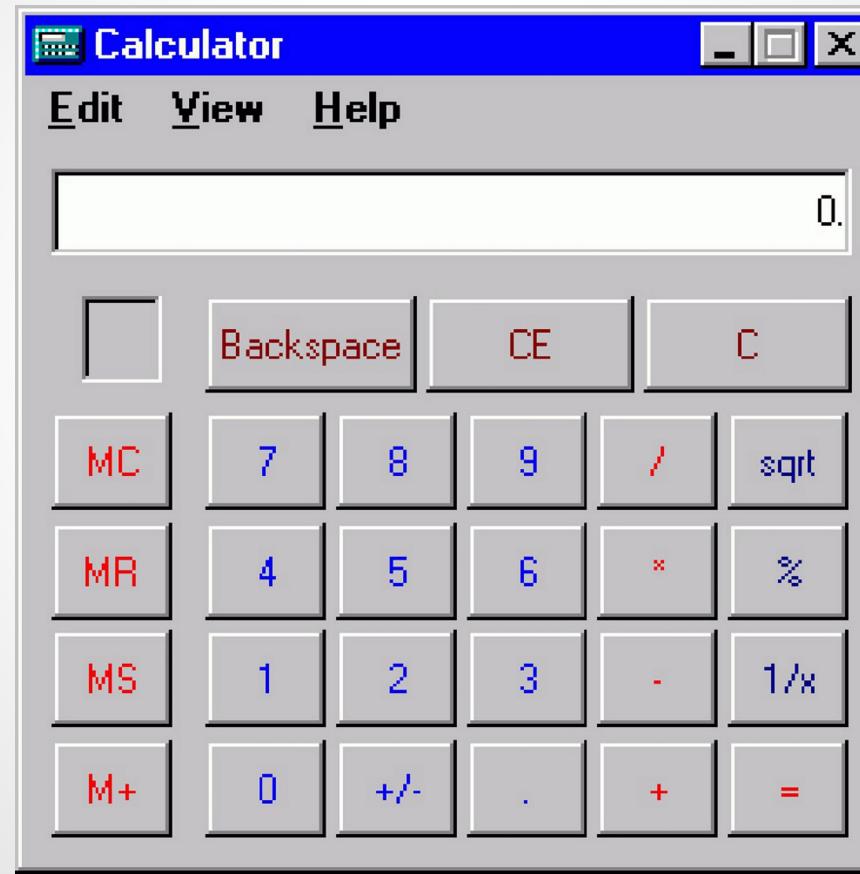
Feel free to pick on the Facedancer's limitations where appropriate.

-- Travis Goodspeed

# What you saw - it's a keyboard!!

```
[ 4726.243752] usb 1-1.2: new high-speed USB device number 12 using ehci-pci
[ 4726.367638] usb 1-1.2: New USB device found, idVendor=610b, idProduct=4653
[ 4726.367650] usb 1-1.2: Product: This is not the USB device you're looking for
[ 4726.367655] usb 1-1.2: Manufacturer: Move along
[ 4726.449866] input: Move along This is not the USB device you're looking for as
 /devices/pci0000:00/0000:00:1a.0/usb1/1-1/1-1.2/1-1.2:1.0/input/input13
```

# What you didn't see :(



# Whats next?

USB/IP implementation

Fixes for strange FaceDancer implementations

Fix all the bugs

# Daisho

USB 3.0 SuperSpeed

# USB 3: Targets

Exploits:

PC XHCI firmware

Operating system USB stack

Device firmware

Sniff, alter, inject data on bus -- MITM

# USB 3: Applications

- Mass storage
- Video capture
- Video display
- Ethernet
- Host-host transfer cable
- Docking stations

# USB 3: Background

Serial bus, full duplex.

500M bytes/second, two 5Gbps links.

Link establishment is complex.

Off-the-shelf PHYs exist, thankfully.

# USB 3: Challenges

Link is very high speed.

PCB design is tricky.

Link negotiation protocol complex.

Scrambling.

TX/RX clock offset.

# USB 3: Timing is Critical

Events at microsecond level.

Data rate is high.

Scrambling, clock management must be done  
in hardware.

# **USB 3: FPGA is the Answer**

Ideal for time-sensitive events.

Ideal for reliable, deterministic inspection and manipulation of high speed data streams.

# Daisho: Main Board

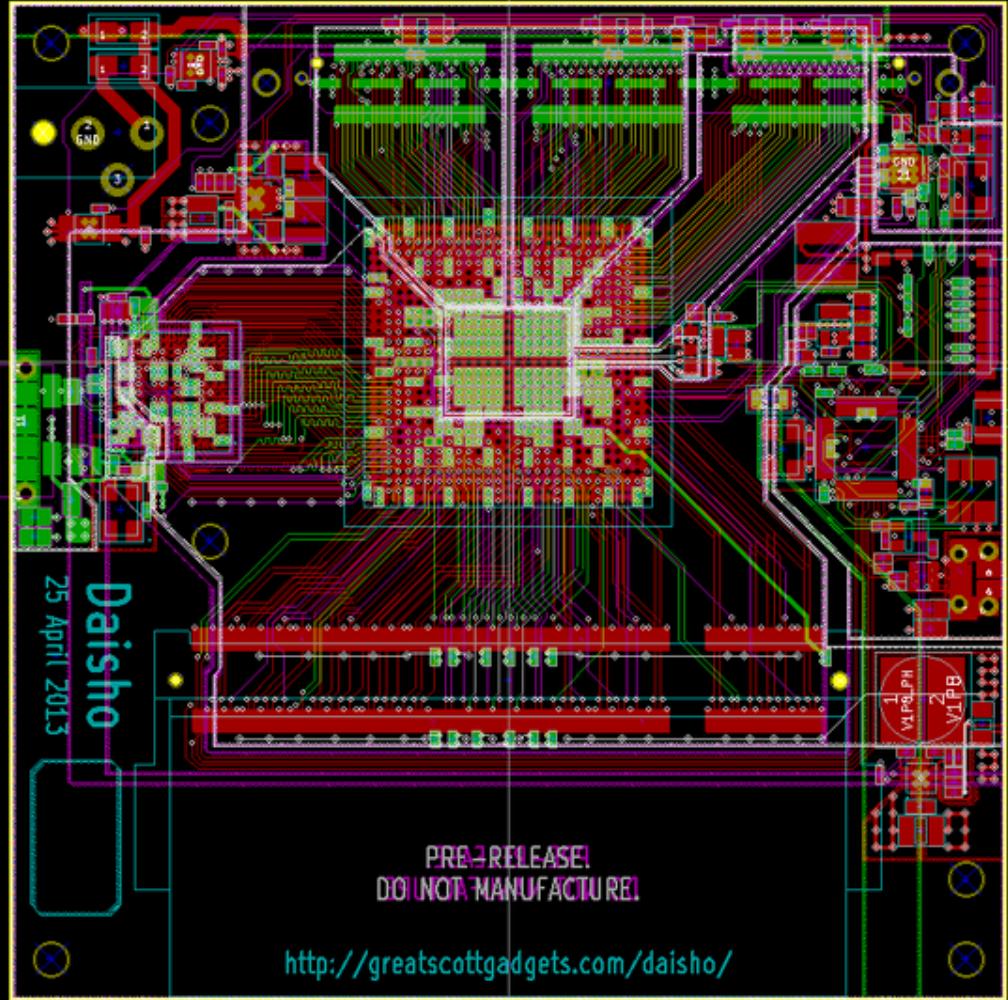
FPGA (Altera Cyclone IV)

DDR2 RAM socket

USB 3.0 interface (TUSB1310A PHY)

Microcontroller for power/config

HSMC connector



### Layer Stack:

Signal 1 (top): 35um Cu + plating (8um)  
Microstrip: 50 Ohms +/- 10% @ 0.15mm

100um prepreg, 370HR Dk=4.24 @ 100MHz

Top Ground: 35um Cu

200um core, 370HR Dk=4.24 @ 100MHz

Top Power: 35um Cu

100 um prepreg, 370HR Dk=4.24 @ 100MHz

Signal 2: 35um Cu

Asymmetric stripline: 50 Ohms +/- 10% @ 0.15mm

8 mil core, 370HR Dk=4.24 @ 100MHz

Signal 3: 35um Cu

Asymmetric stripline: 50 Ohms +/- 10% @ 0.15mm

100um prepreg, 370HR Dk=4.24 @ 100MHz

Bottom Power: 35um Cu

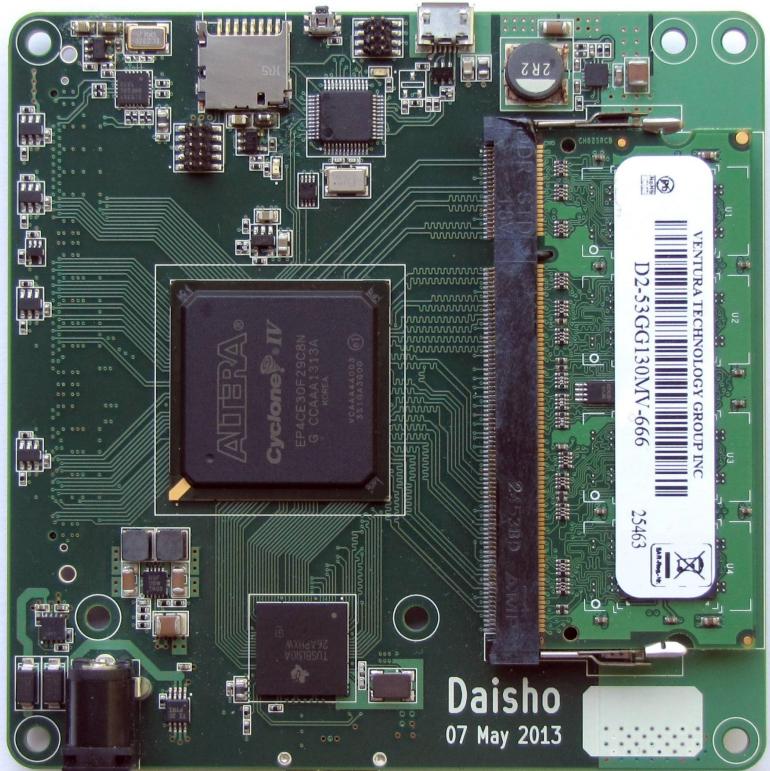
200um core, 370HR Dk=4.24 @ 100MHz

Bottom Ground: 35um Cu

100um prepreg, 370HR Dk=4.24 @ 100MHz

Signal 4 (bottom): 35um Cu + plating (8um)

Microstrip: 50 Ohms +/- 10% @ 0.15mm

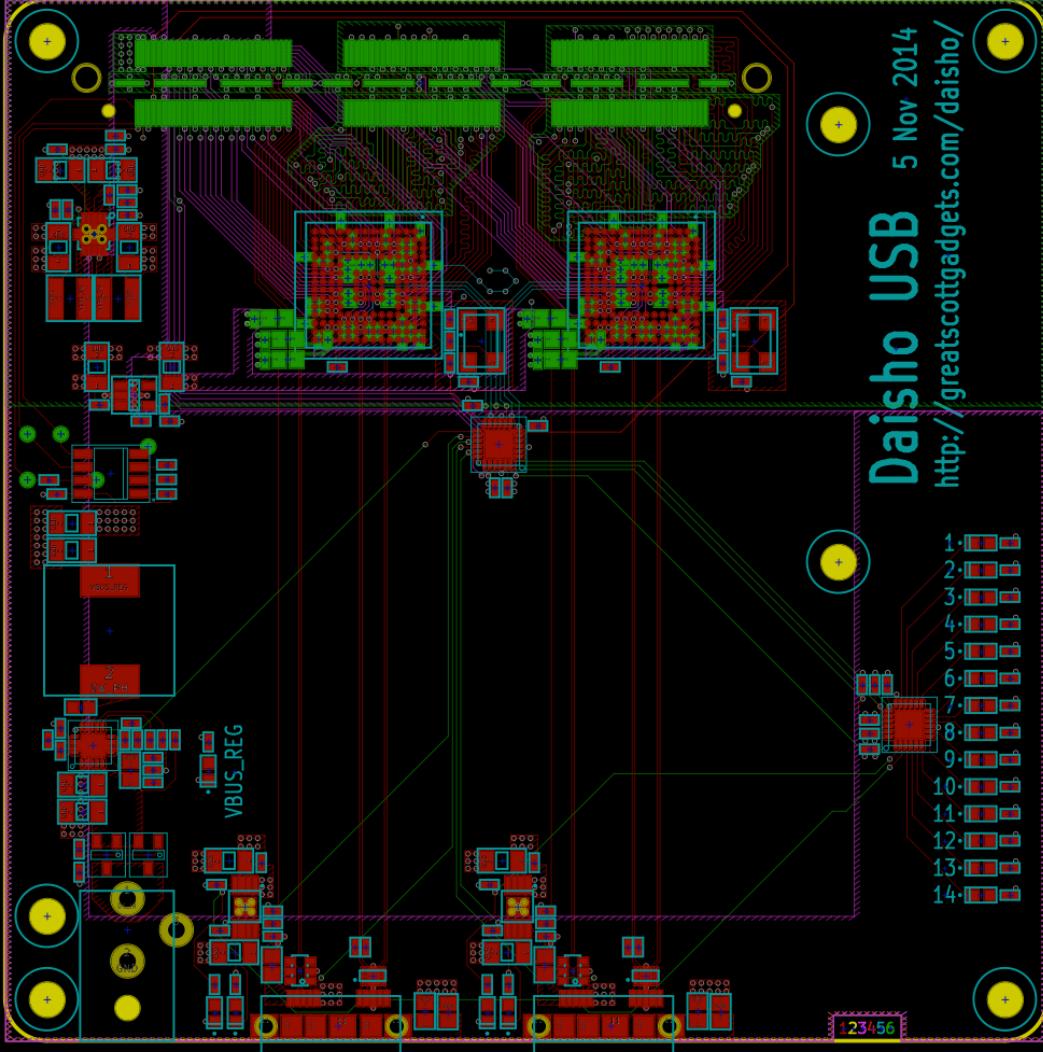


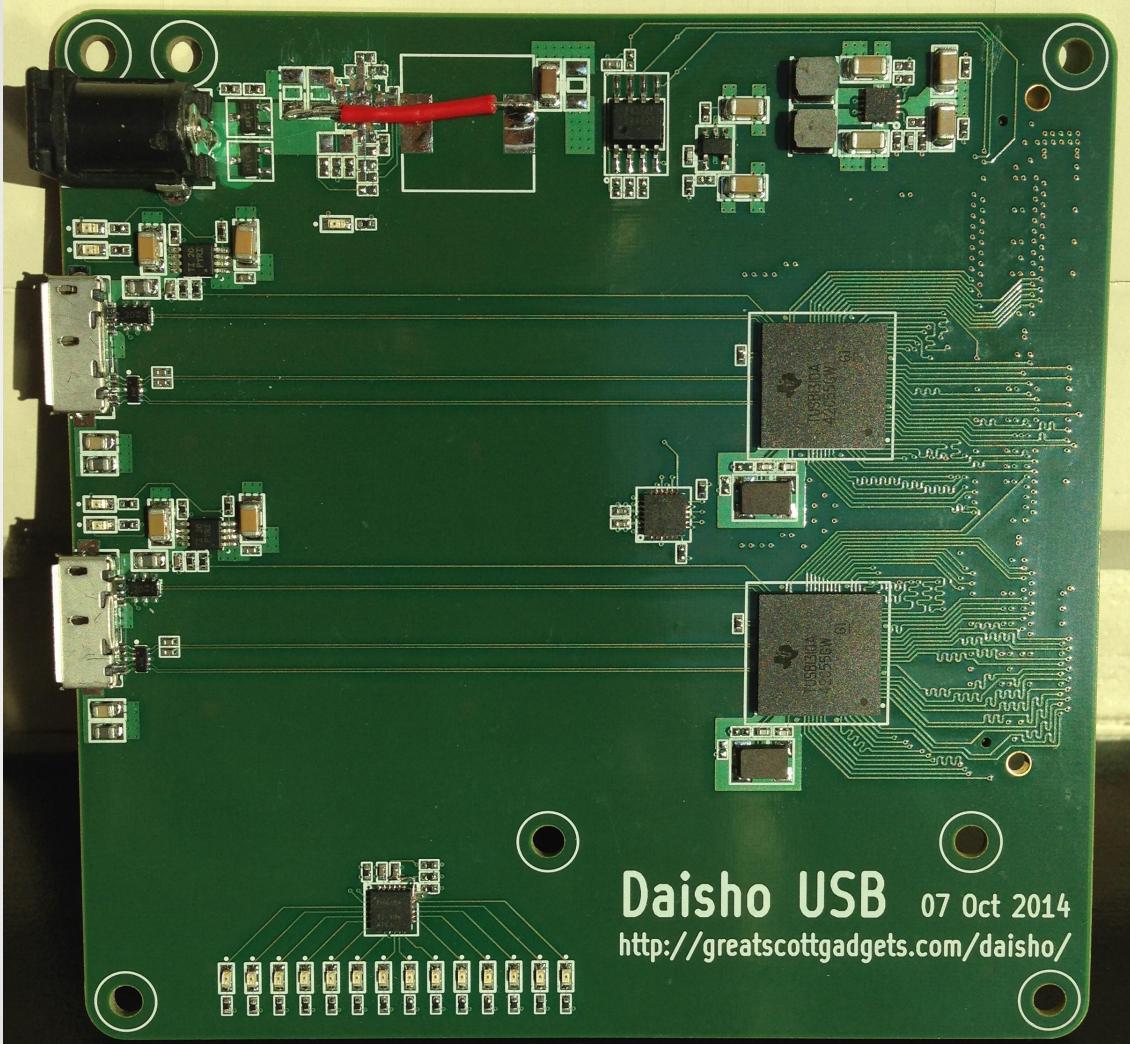
# Daisho: USB Front End

Two more USB 3.0 PHYs

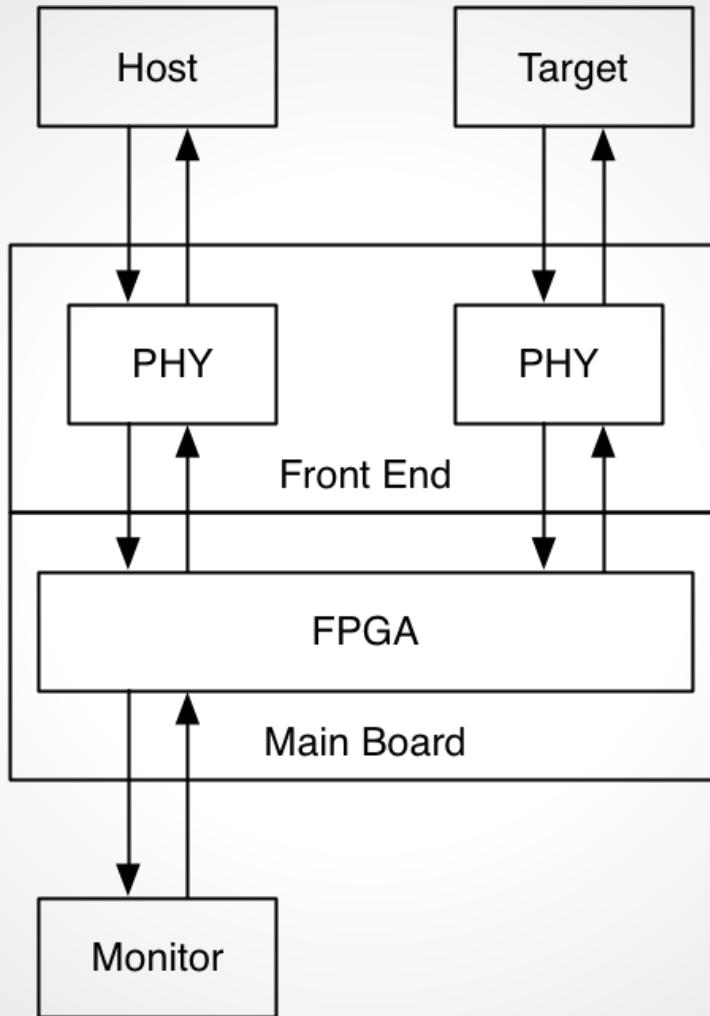
Regulator for supplying USB bus power

**Daiisho USB** 5 Nov 2014  
<http://greatscottgadgets.com/daiisho/>





Daisho USB 07 Oct 2014  
<http://greatscottgadgets.com/daisho/>



# USB 3: Handshaking

RxDetect

LFPS

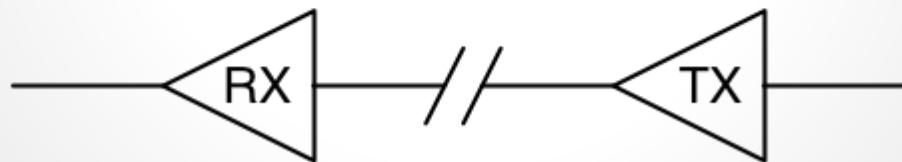
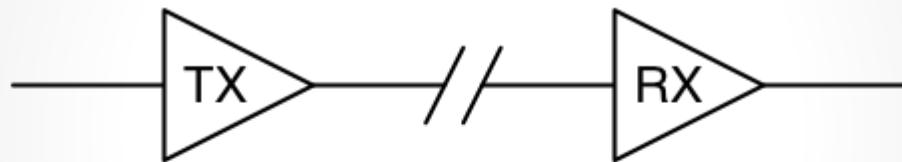
Training

Link Config

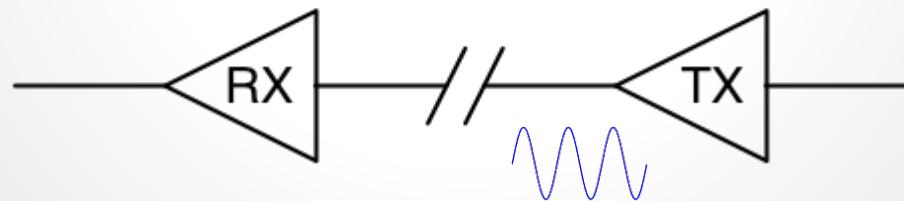
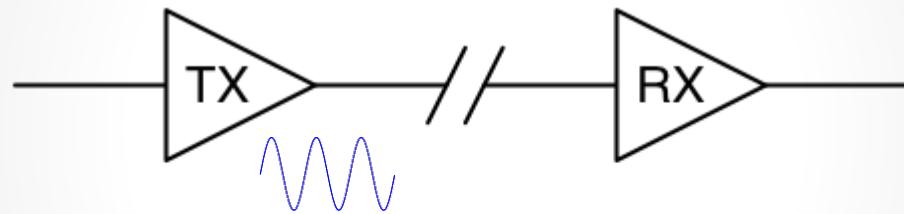
Enumeration

Control/Data transfer

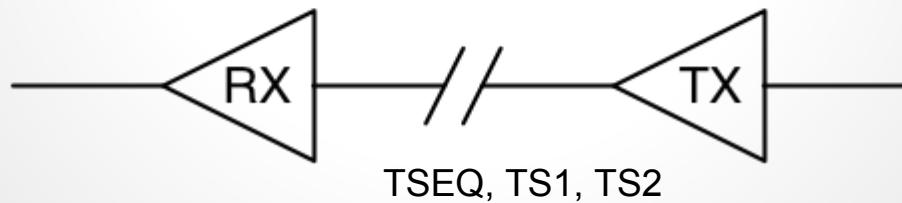
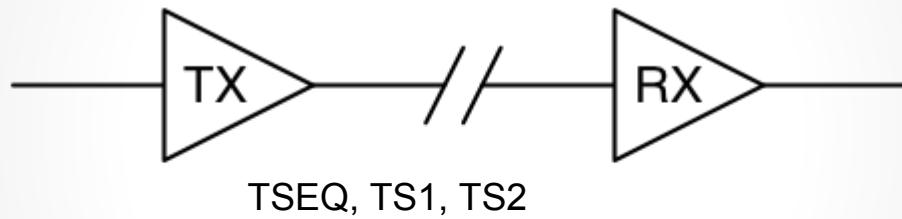
# USB 3: RxDetect



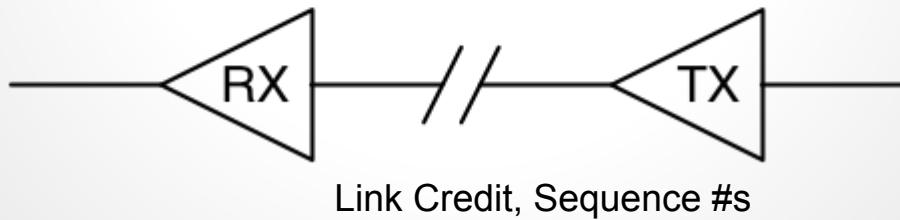
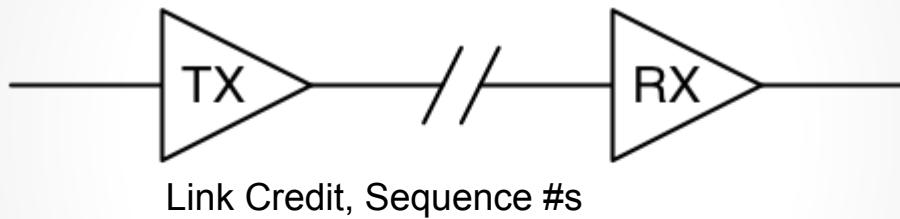
# USB 3: LFPS



# USB 3: Training



# USB 3: Link Config



Sp	Index	m.s.ms.us.ns	Len	Err	Dev	Ep	Record	Data	Summary
SS ↑	229478	0:18.323.789.020	13.1 us				[206 TSEQ]		
SS ↑	229685	0:18.323.802.204					LTSSM Transition		Polling.RxEq -> Polling.Active
SS ↑	229686	0:18.323.802.204	320 ns				[10 TS1]		
SS ↑	229697	0:18.323.802.524					LTSSM Transition		Polling.Active -> Polling.Config
SS ↓	229698	0:18.323.700.912	101 us				[3161 TS1]		
SS ↓	232860	0:18.323.802.652					LTSSM Transition		Polling.Active -> Polling.Config
SS ↓	232861	0:18.323.802.652	584 ns				[18 TS2]		
SS ↓	232880	0:18.323.803.236					LTSSM Transition		Polling.Config -> Polling.Idle
SS ↑	232881	0:18.323.802.524	1.57 us				[49 TS2]		
SS ↑	232931	0:18.323.804.100					LTSSM Transition		Polling.Config -> Polling.Idle
SS ↑	232932	0:18.323.804.148					LTSSM Transition		Polling.Idle -> U0
SS ↑	232933	0:18.323.804.148	8 B				Link Good 7	FE FE F7 07 68 07 68	
SS ↑	232934	0:18.323.804.164	8 B				Link Credit A	FE FE F7 80 A0 80 A0	
SS ↑	232935	0:18.323.804.180	8 B				Link Credit B	FE FE F7 81 58 81 58	
SS ↑	232936	0:18.323.804.196	8 B				Link Credit C	FE FE F7 82 18 82 18	
SS ↑	232937	0:18.323.804.212	8 B				Link Credit D	FE FE F7 83 E0 83 E0	
SS ↓	232938	0:18.323.804.308					LTSSM Transition		Polling.Idle -> U0
SS ↓	232939	0:18.323.804.308	8 B				Link Good 7	FE FE F7 07 68 07 68	
SS ↓	232940	0:18.323.804.324	8 B				Link Credit A	FE FE F7 80 A0 80 A0	
SS ↓	232941	0:18.323.804.340	8 B				Link Credit B	FE FE F7 81 58 81 58	
SS ↓	232942	0:18.323.804.356	8 B				Link Credit C	FE FE F7 82 18 82 18	
SS ↓	232943	0:18.323.804.372	8 B				Link Credit D	FE FE F7 83 E0 83 E0	
SS ↓	232944	0:18.323.804.388					Link Management		{PortCapability} [HdrSeq=0 DL]
SS ↑	232947	0:18.323.804.912	8 B				Link Credit A	FE FE F7 80 A0 80 A0	
SS ↑	232948	0:18.323.804.832					Link Management		{PortCapability} [HdrSeq=0]
SS ↓	232951	0:18.323.805.060	8 B				Link Credit A	FE FE F7 80 A0 80 A0	
SS ↓	232952	0:18.323.805.076					Link Management		{PortConfiguration} [HdrSeq=...
SS ↑	232955	0:18.323.805.564	8 B				Link Credit B	FE FE F7 81 58 81 58	
SS ↑	232956	0:18.323.805.596					Link Management		{PortConfigurationResponse} ...
SS ↓	232959	0:18.323.805.828	8 B				Link Credit B	FE FE F7 81 58 81 58	
SS ↓	232960	0:18.323.815.628	2.64 ms				[265 LUP & 258 LDN]		
SS ↑	233484	0:18.326.459.844					LTSSM Transition		U0 -> Recovery.Active
SS ↓	233485	0:18.326.460.048					LTSSM Transition		U0 -> Recovery.Active
SS ↓	233486	0:18.326.460.048	288 ns				[9 TS1]		

Text

LiveSearch



Ready

Disconnected

EN

# USB 3: What's Done

Hardware appears solid.

USB 3.0 stack for monitoring port.

Pass-through up to enumeration.

Accidental fuzzing...

# USB 3: Future Work

Fuzzing firmware, operating systems.

Altering data between host and target.

Spoofing devices not present.

Wholesale recording and replay.

Miniaturization.

# TURNIPSCHOOL

CABLES THAT BITE

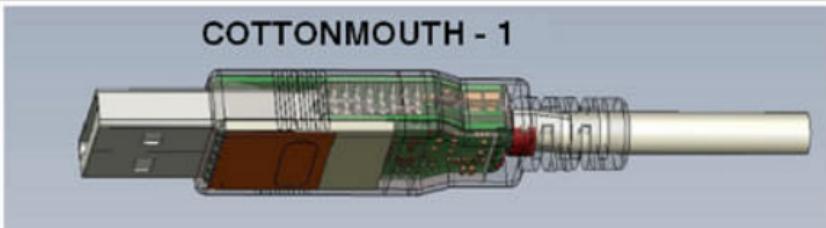


# COTTONMOUTH-I

## ANT Product Data

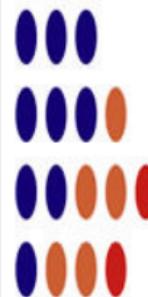
08/05/08

**(TS//SI//REL)** COTTONMOUTH-I (CM-I) is a Universal Serial Bus (USB) hardware implant which will provide a wireless bridge into a target network as well as the ability to load exploit software onto target PCs.

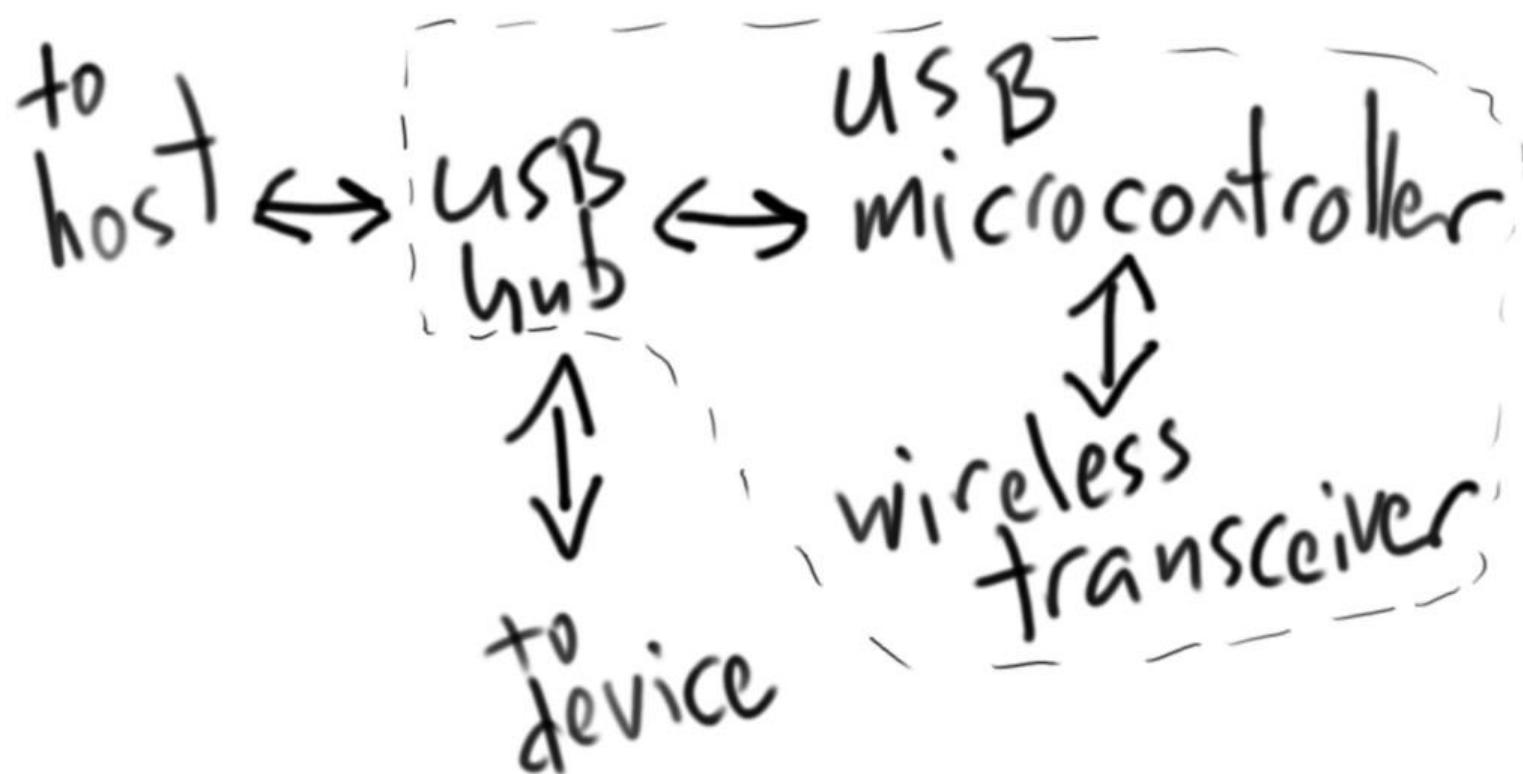


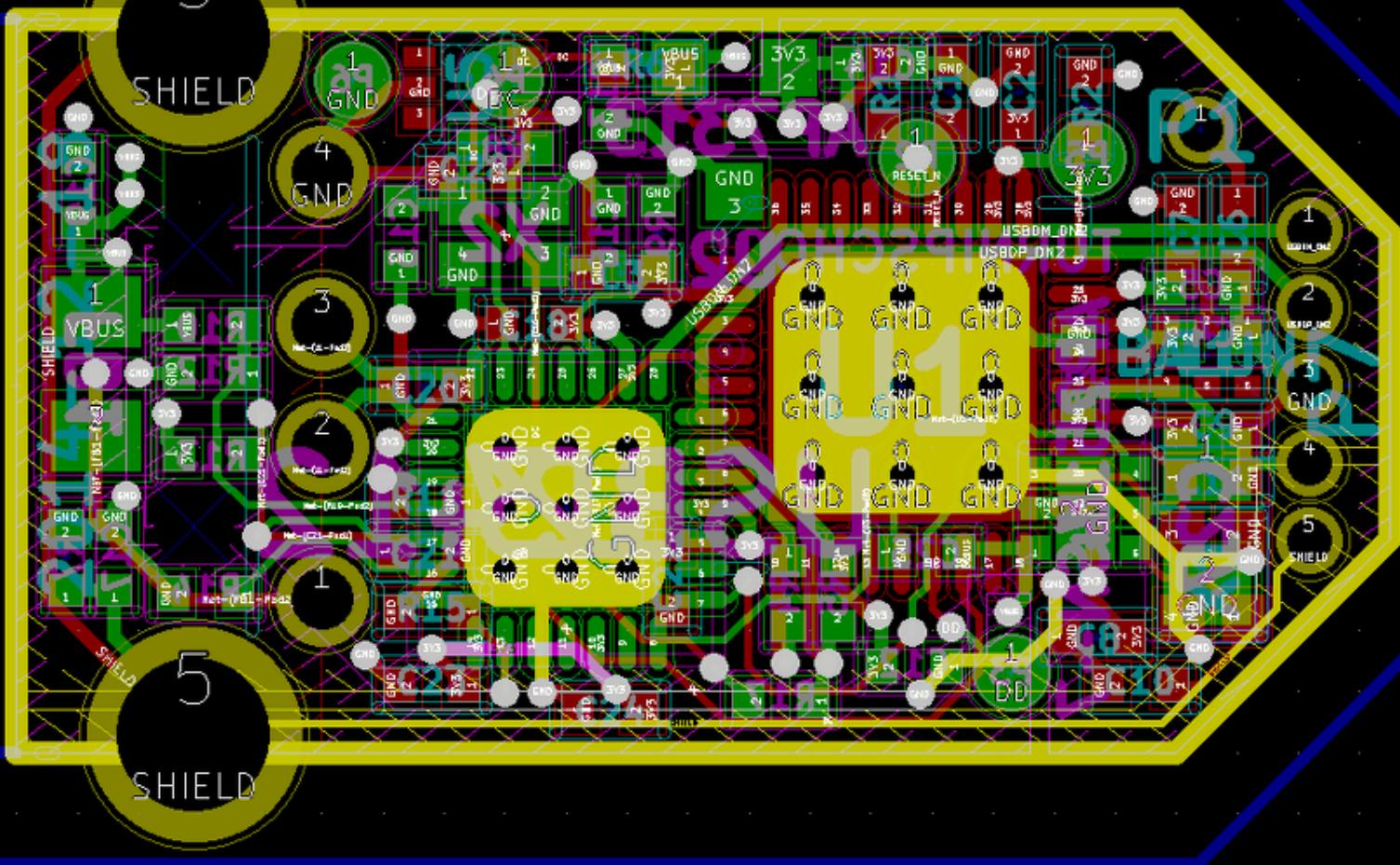
**(TS//SI//REL)** CM-I will provide air-gap bridging, software persistence capability, "in-field" re-programmability, and covert communications with a host software implant over the USB. The RF link will enable command and data infiltration and exfiltration. CM-I will also communicate with Data Network Technologies (DNT) software (STRAITBIZARRE) through a covert channel implemented on the USB, using this communication channel to pass commands and data between hardware and software implants. CM-I will be a GENIE-compliant implant based on CHIMNEYPOOL.

**(TS//SI//REL)** CM-I conceals digital components (TRINITY), USB 1.1 FS hub, switches, and HOWLERMONKEY (HM) RF Transceiver within the USB Series-A cable connector. MOCCASIN is the version permanently connected to a USB keyboard. Another version can be made with an unmodified USB connector at the other end. CM-I has the ability to communicate to other CM devices over the RF link using an over-the-air protocol called SPECULATION.



# COTTONMOUTH - I





# CC1111

Sub-1 GHz RF transceiver

8051 microcontroller

32 kB flash memory

4 kB RAM

Full-Speed USB device controller

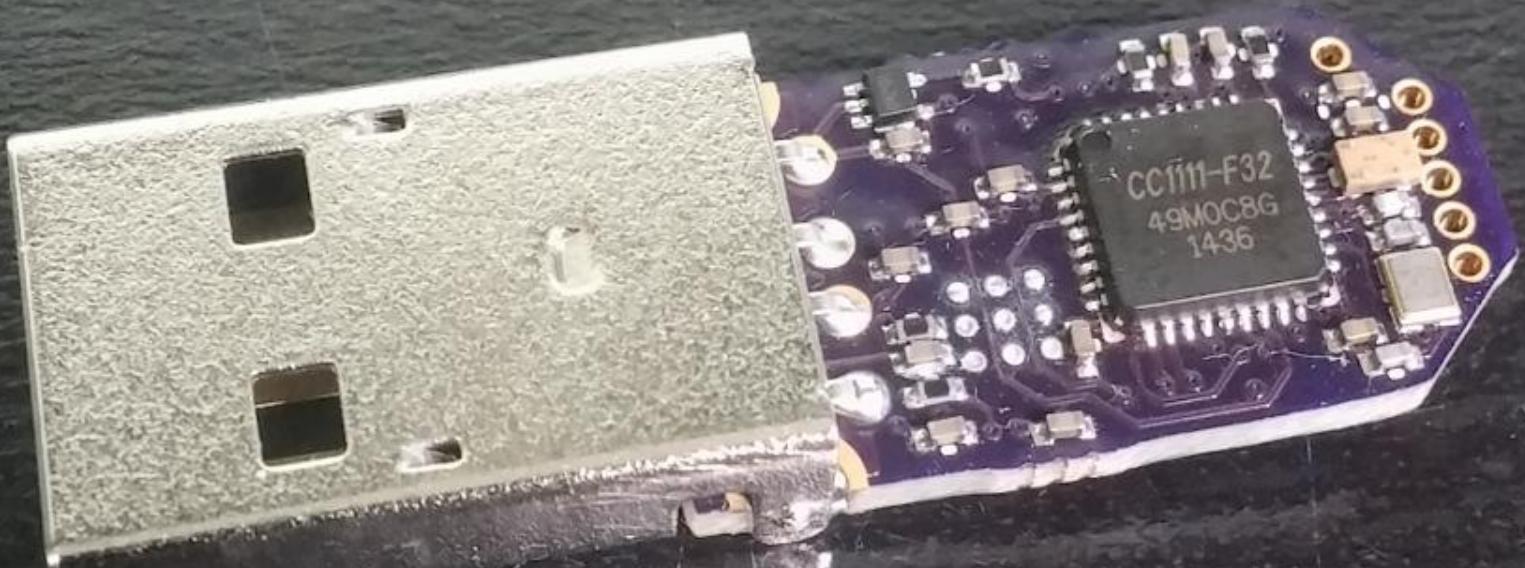
GoodFET-programmable

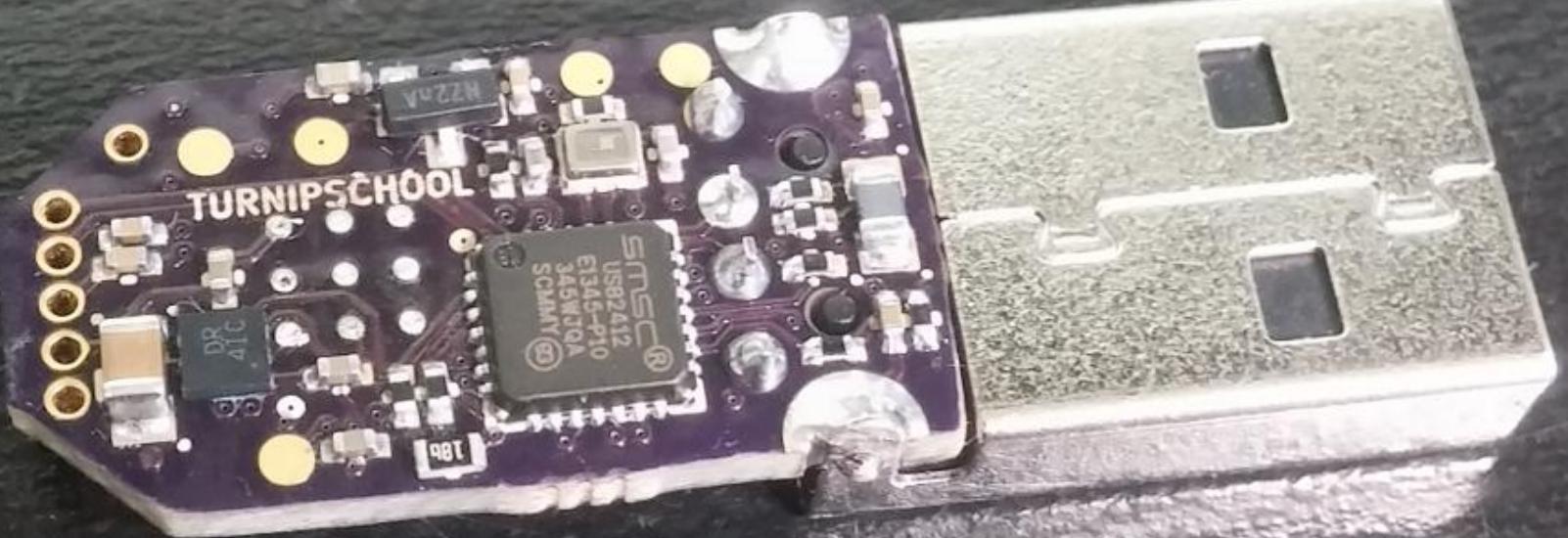
# IM-Me

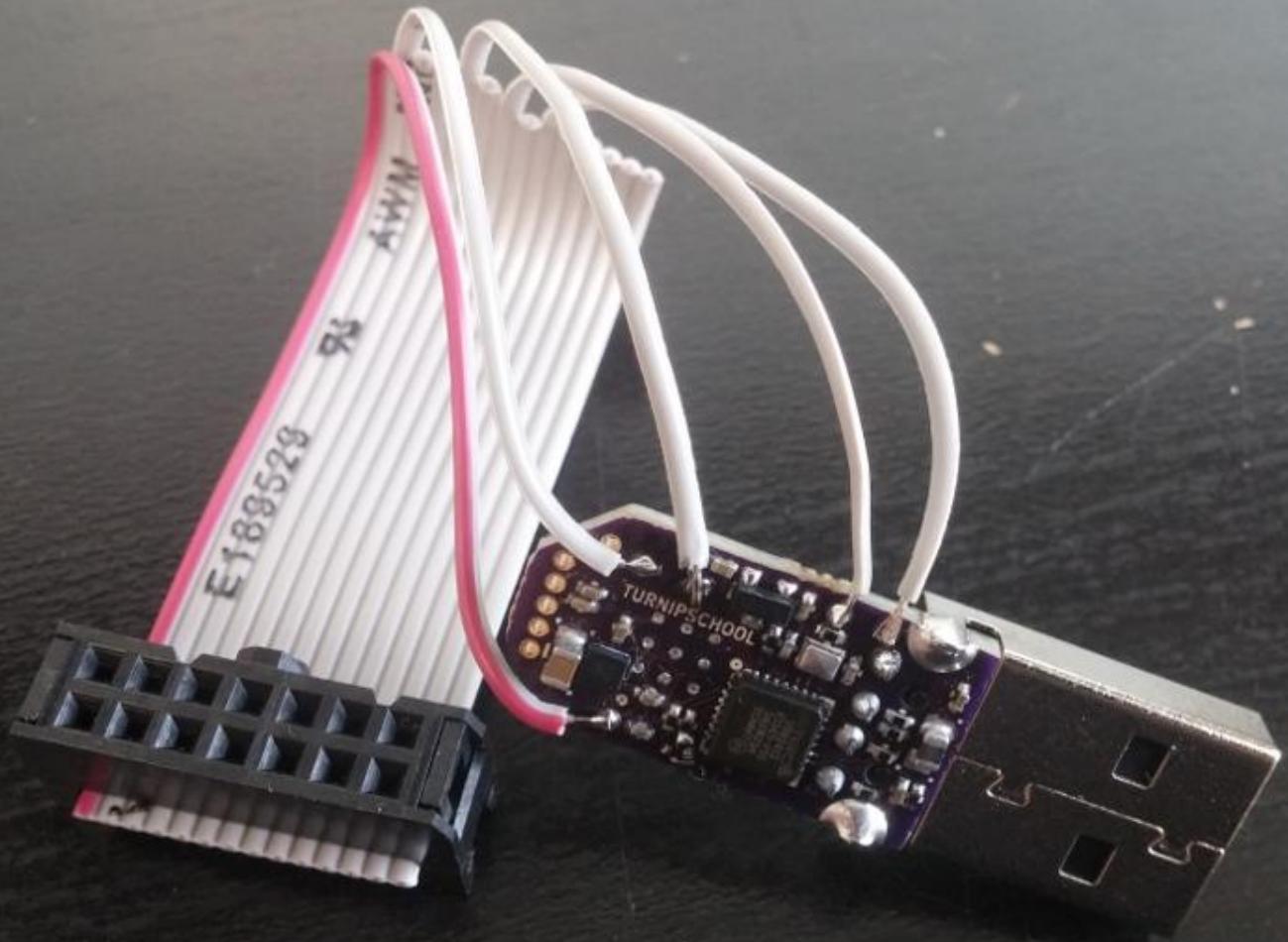


# YARD Stick One









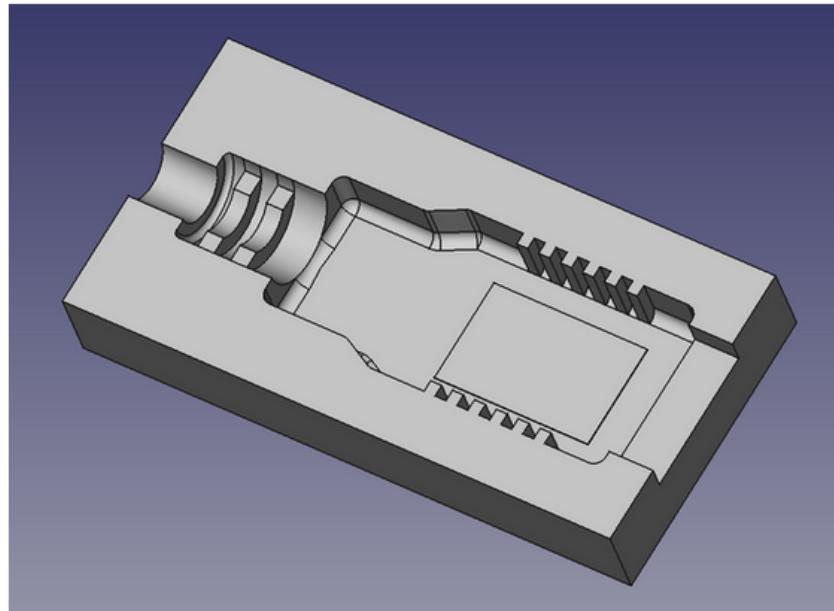
```
[ 1472.458043] usb 3-2: new high-speed USB device number 3 using xhci_hcd
[ 1472.469302] usb 3-2: New USB device found, idVendor=0424, idProduct=2412
[ 1472.470143] hub 3-2:1.0: USB hub found
[ 1472.470274] hub 3-2:1.0: 2 ports detected
[ 1472.470302] hub 3-2:1.0: individual port power switching
[ 1472.470313] hub 3-2:1.0: individual port over-current protection
[ 1472.470798] hub 3-2:1.0: enabling power on all ports
[ 1472.744448] usb 3-2.2: new full-speed USB device number 4 using xhci_hcd
[ 1472.758601] usb 3-2.2: New USB device found, idVendor=1d50, idProduct=6048
[ 1472.758612] usb 3-2.2: Product: Dons Dongle
[ 1472.758617] usb 3-2.2: Manufacturer: RfCat
[ 1473.573402] usb 3-2.1: new high-speed USB device number 5 using xhci_hcd
[ 1635.127787] usb 3-2.1: New USB device found, idVendor=1004, idProduct=6245
[ 1635.127799] usb 3-2.1: Product: LGE Android Phone
[ 1635.127804] usb 3-2.1: Manufacturer: LG Electronics Inc.
```



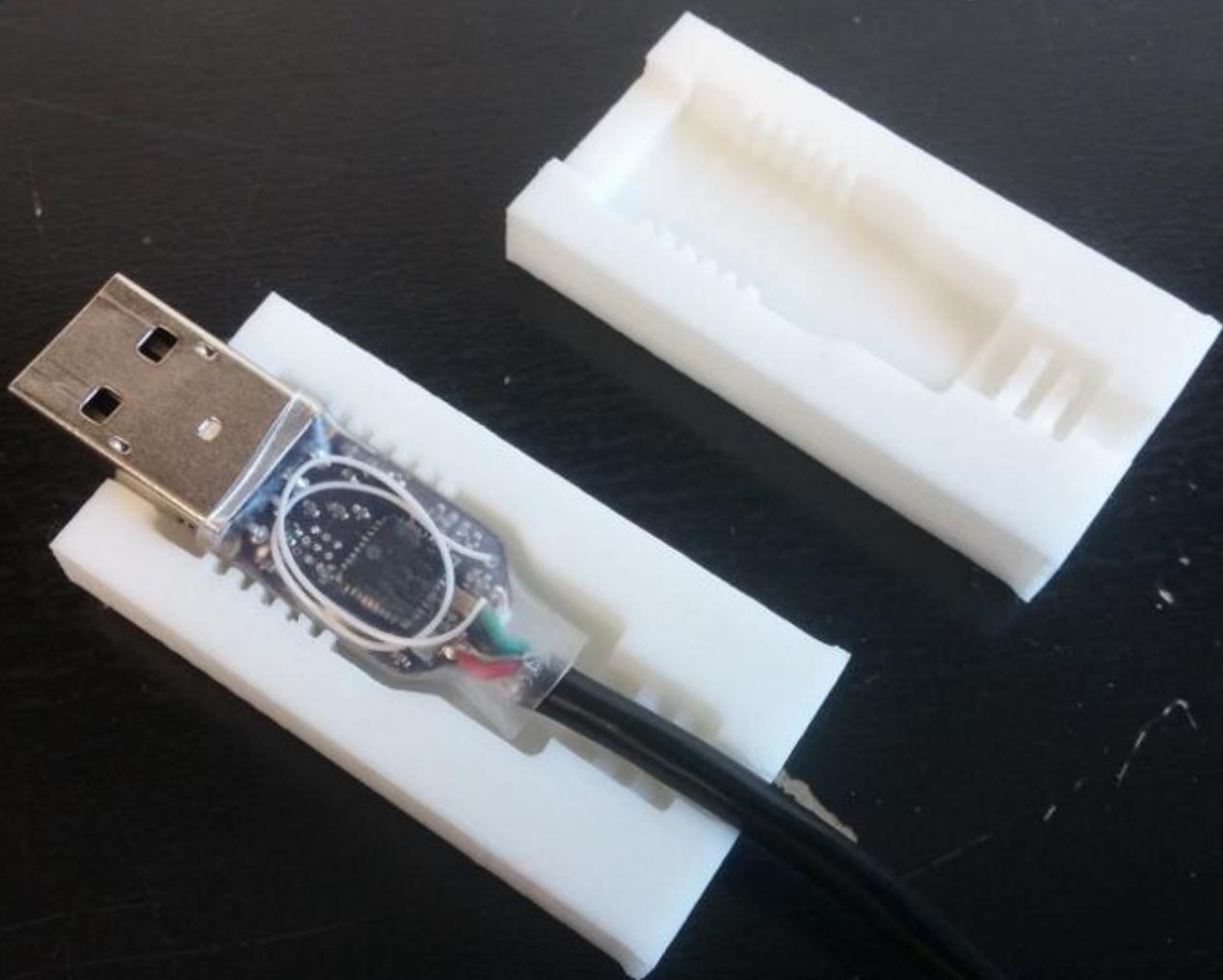
Kiki der Gecko  
@kikidergecko

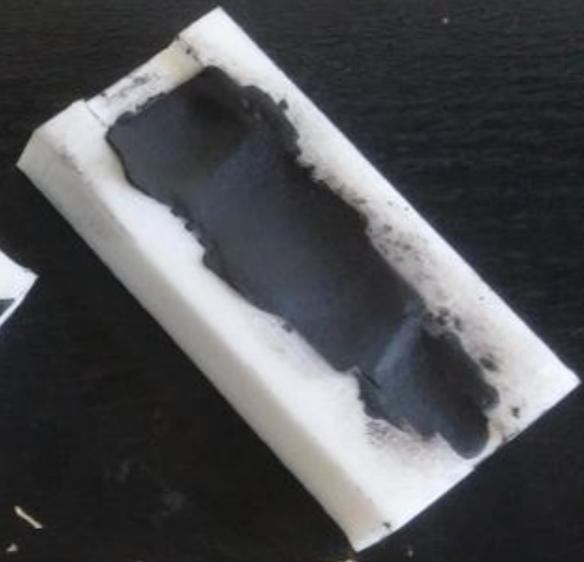
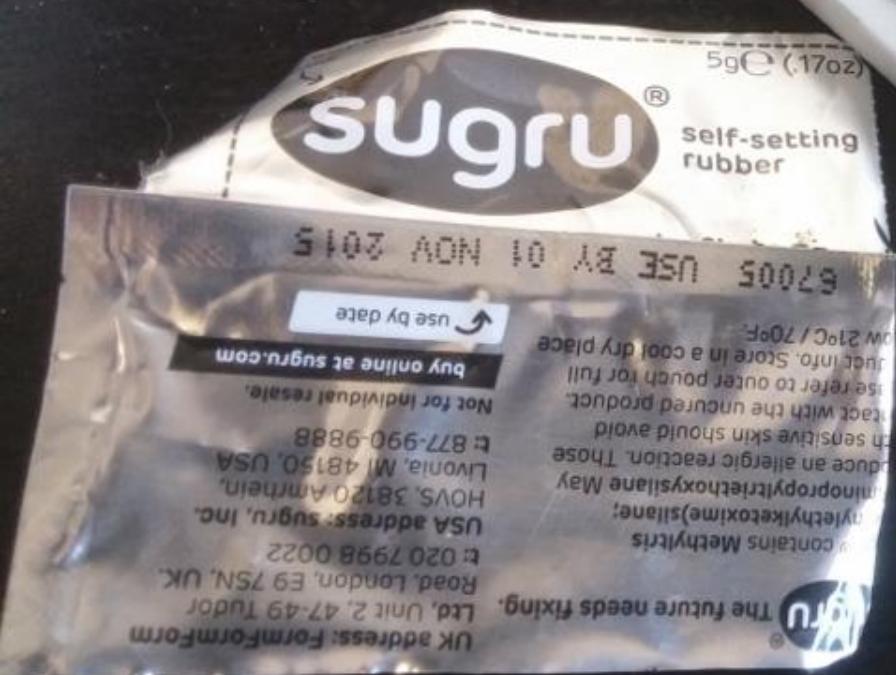
Follow

@michaelossmann Voilà! I have no clue about injection mold design though, so filling holes are still missing.



<https://github.com/kikithegecko/turnipschool-casing>















```
mossmann@falstaff ~/shmoo/2015 $ rfcat -r
'RfCat, the greatest thing since Frequency Hopping!'

Research Mode: enjoy the raw power of rflib

currently your environment has an object called "d" for dongle. this is how
you interact with the rfcat dongle:
>>> d.ping()
>>> d.setFreq(433000000)
>>> d.setMdmModulation(MOD_ASK_00K)
>>> d.makePktFLEN(250)
>>> d.RFxmit("HALLO")
>>> d.RFrecv()
>>> print d.reprRadioConfig()

In [1]: d.ping()
PING: 26 bytes transmitted, received: 'ABCDEFGHIJKLMNPQRSTUVWXYZ' (0.004013 seconds)
PING: 26 bytes transmitted, received: 'ABCDEFGHIJKLMNPQRSTUVWXYZ' (0.003001 seconds)
PING: 26 bytes transmitted, received: 'ABCDEFGHIJKLMNPQRSTUVWXYZ' (0.003221 seconds)
PING: 26 bytes transmitted, received: 'ABCDEFGHIJKLMNPQRSTUVWXYZ' (0.002548 seconds)
PING: 26 bytes transmitted, received: 'ABCDEFGHIJKLMNPQRSTUVWXYZ' (0.002948 seconds)
PING: 26 bytes transmitted, received: 'ABCDEFGHIJKLMNPQRSTUVWXYZ' (0.002693 seconds)
PING: 26 bytes transmitted, received: 'ABCDEFGHIJKLMNPQRSTUVWXYZ' (0.003444 seconds)
PING: 26 bytes transmitted, received: 'ABCDEFGHIJKLMNPQRSTUVWXYZ' (0.004563 seconds)
```

Bus 003 Device 003: ID 610b:4653  
Device Descriptor:  
    bLength                18  
    bDescriptorType       1  
    bcdUSB               1.10  
    bDeviceClass          0 (Defined at Interface level)  
    bDeviceSubClass      0  
    bDeviceProtocol      0  
    bMaxPacketSize0      32  
    idVendor             0x610b  
    idProduct            0x4653  
    bcdDevice            0.10  
    iManufacturer        1 Great Scott Gadgets  
    iProduct              2 TURNIPSCHOOL  
    iSerial              3 0000  
    bNumConfigurations  1  
Configuration Descriptor:  
    bLength              9  
    bDescriptorType      2  
    wTotalLength        34  
    bNumInterfaces      1  
    bConfigurationValue 1  
    iConfiguration      0  
    bmAttributes        0xc0  
        Self Powered  
    MaxPower            100mA  
Interface Descriptor:  
    bLength              9  
    bDescriptorType      4  
    bInterfaceNumber    0  
    bAlternateSetting   0  
    bNumEndpoints       1  
    bInterfaceClass     3 Human Interface Device  
    bInterfaceSubClass  0 No Subclass  
    bInterfaceProtocol  1 Keyboard  
    iInterface          0

# Build Your Own TURNIPSCHOOL

PCB: \$1.50 each

Solder paste stencil: \$15

Components: \$20

Sugru: \$10

3D printed mold: \$40

Solder paste, tweezers, hot plate, soldering iron

Open Source Hardware!

# Thank You

DARPA

BIT Systems

Marshall Hecht

Karoline “kikithegecko” Busse

Dean Pierce

Alexander Holler

Travis Goodspeed & Sergey Bratus

# Questions?

<http://www.nsaplayset.org/>

<http://greatscottgadgets.com/>