RASPDANCER

Redesigning
Facedancer11
for Raspberry Pi

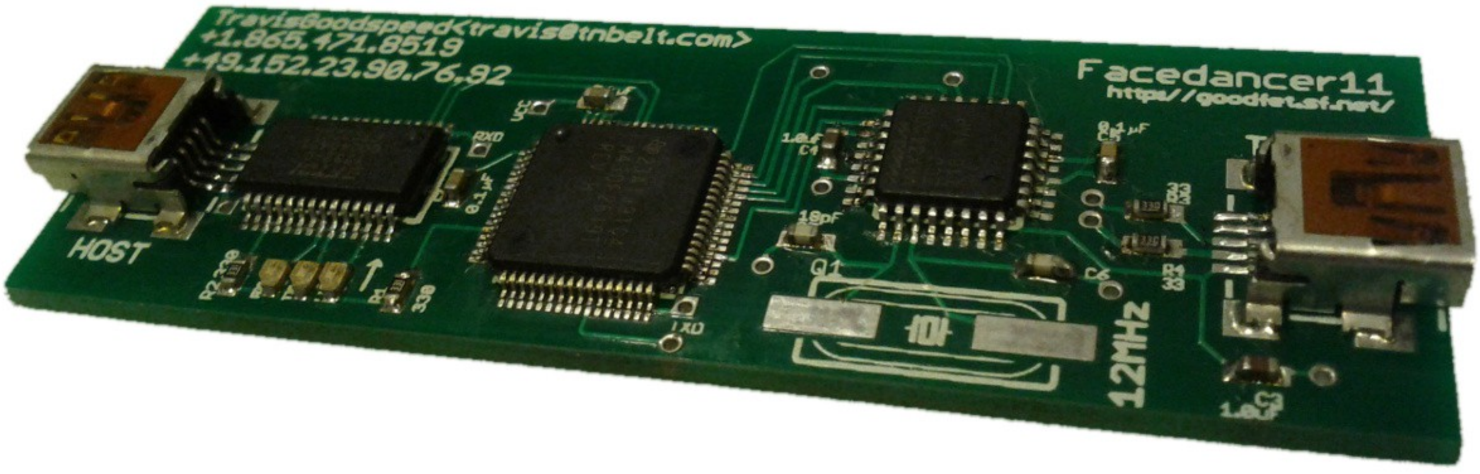
Philippe Teuwen
Hackito 2013

Facedancer

by Travis Goodspeed

Can pretend to be any USB peripheral

Allow fuzzing of USB device drivers of a target



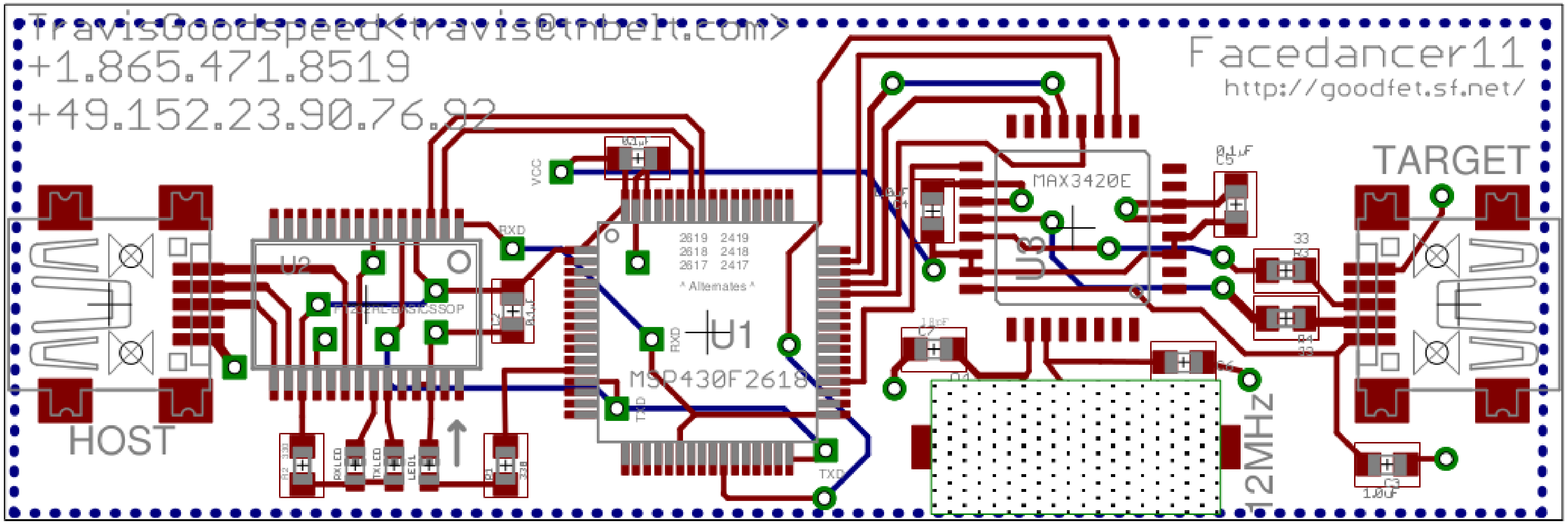
Just plug it in...

It's gonna say:
“Hey I see you've plugged a new device”
And it's gonna load the appropriate drivers...

Quiz:
Does it ring a bell to anybody?

USB Plug&Play introduced in W98
Las Vegas, 1998

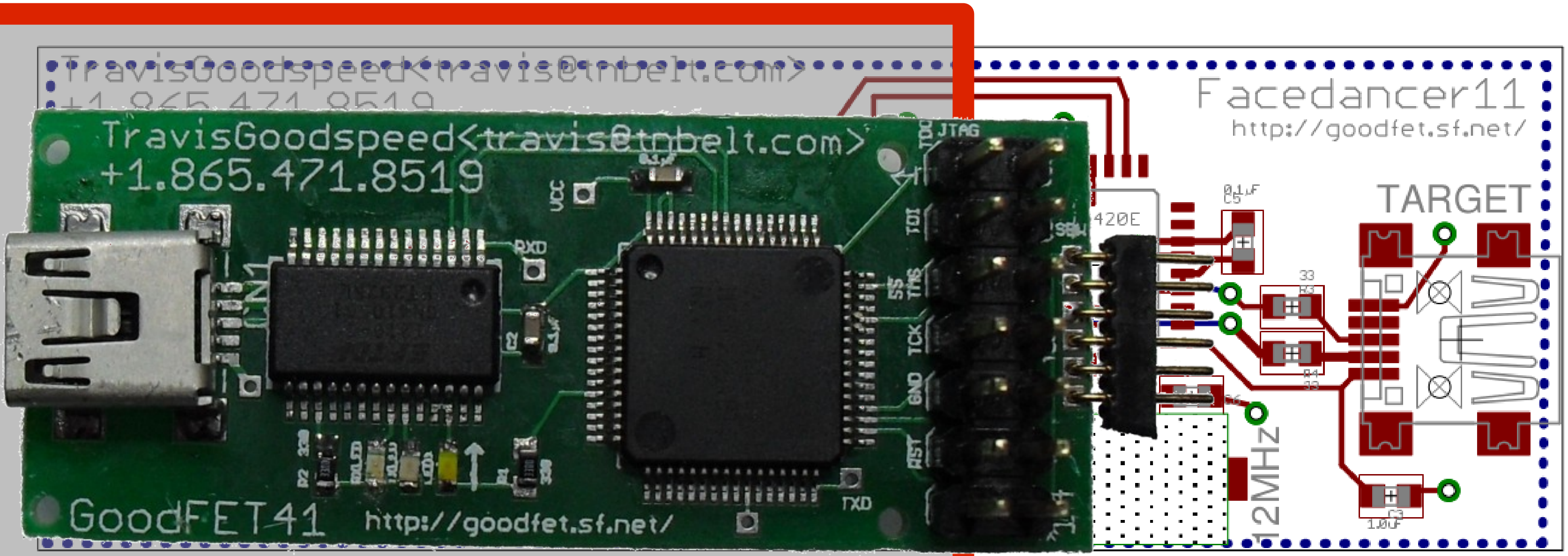
A closer look



USB Host	FT232RL	MSP430	MAX3420E	USB Target
USB	USB↔UART	UART↔SPI	SPI...	
	6.60€	15.80€	10.00€	

Bottleneck: UART @115200bauds

A closer look



USB Host	FT232RL	MSP430	MAX3420E	USB Target
USB	USB↔UART	UART↔SP	SPI...	
	6.60€	15.80€	10.00€	

GoodFET

22.40€

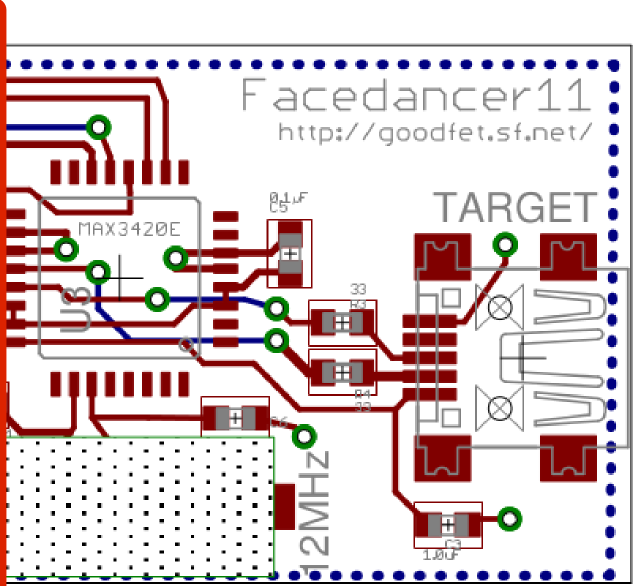
Bottleneck: UART @115200bauds

To summarize

- MAX3420E:
  USB Peripheral Controller with SPI Interface
- GoodFET hardwired to do USB↔SPI
- All intelligence moved to the host
  in a nice python library

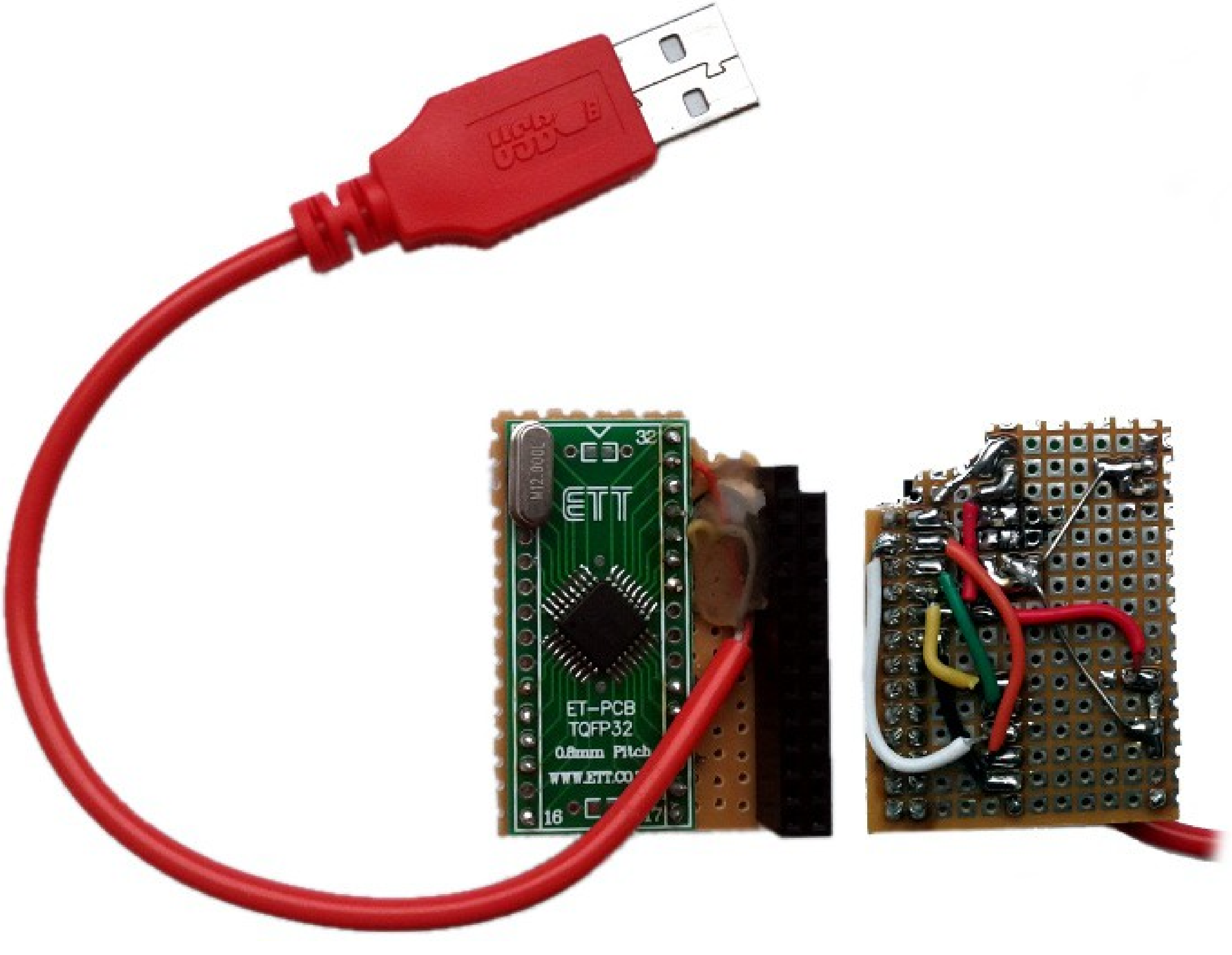
Can we do something like this?

Raspberry Pi as host

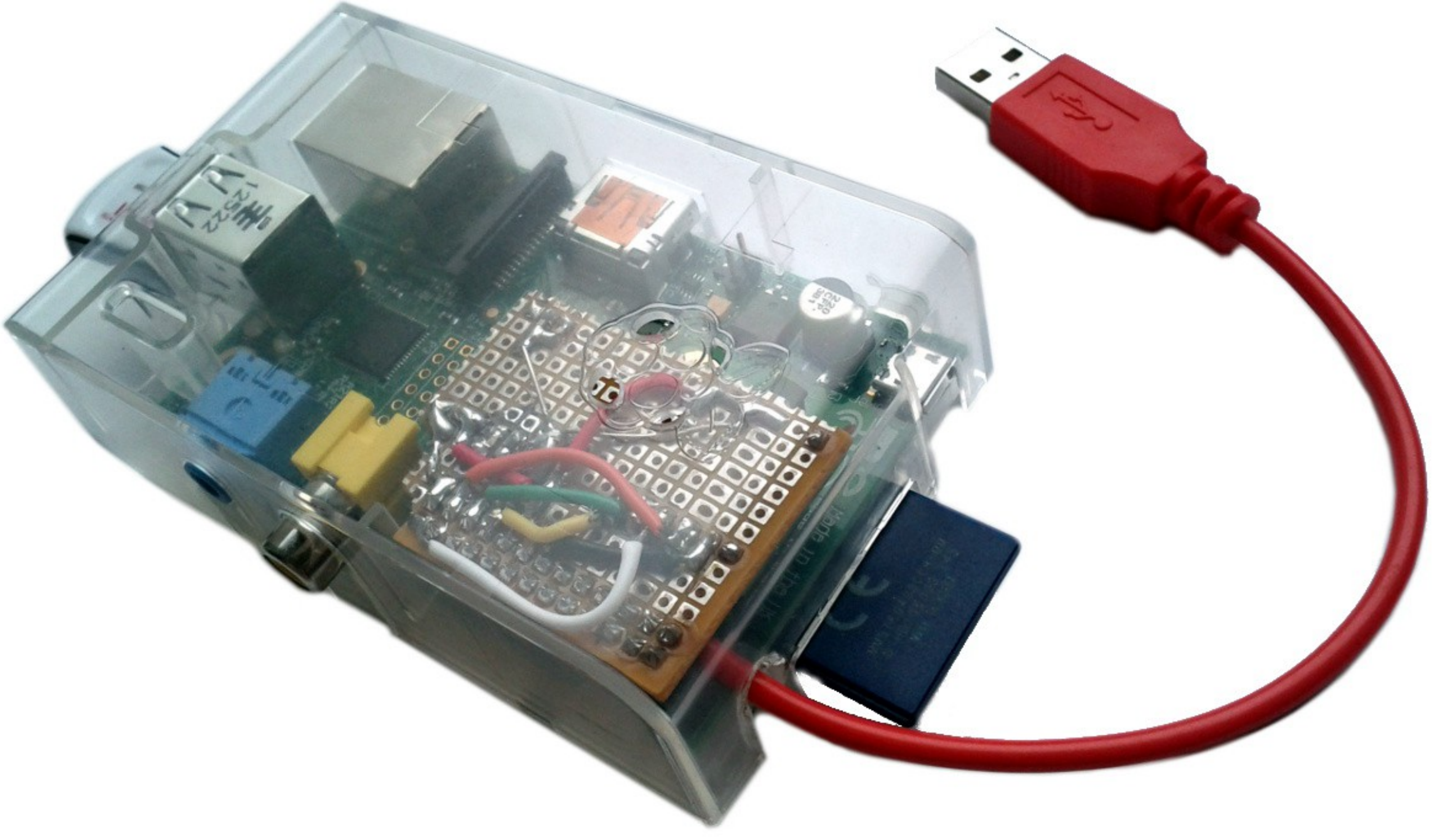


MAX3420E	USB Target
SPI...	
10.00€	

First mess^H^H^H^Hprototype



First prototype



Looks awesome... unless you use a crystal case... sigh.

Adapting the code

GoodFETMAXUSB.py with our raspdancer:

Drop-in replacement of GoodFET.py library

- no fork, no patch
- mutualize USB fuzzing efforts,
  no matter which hardware is used

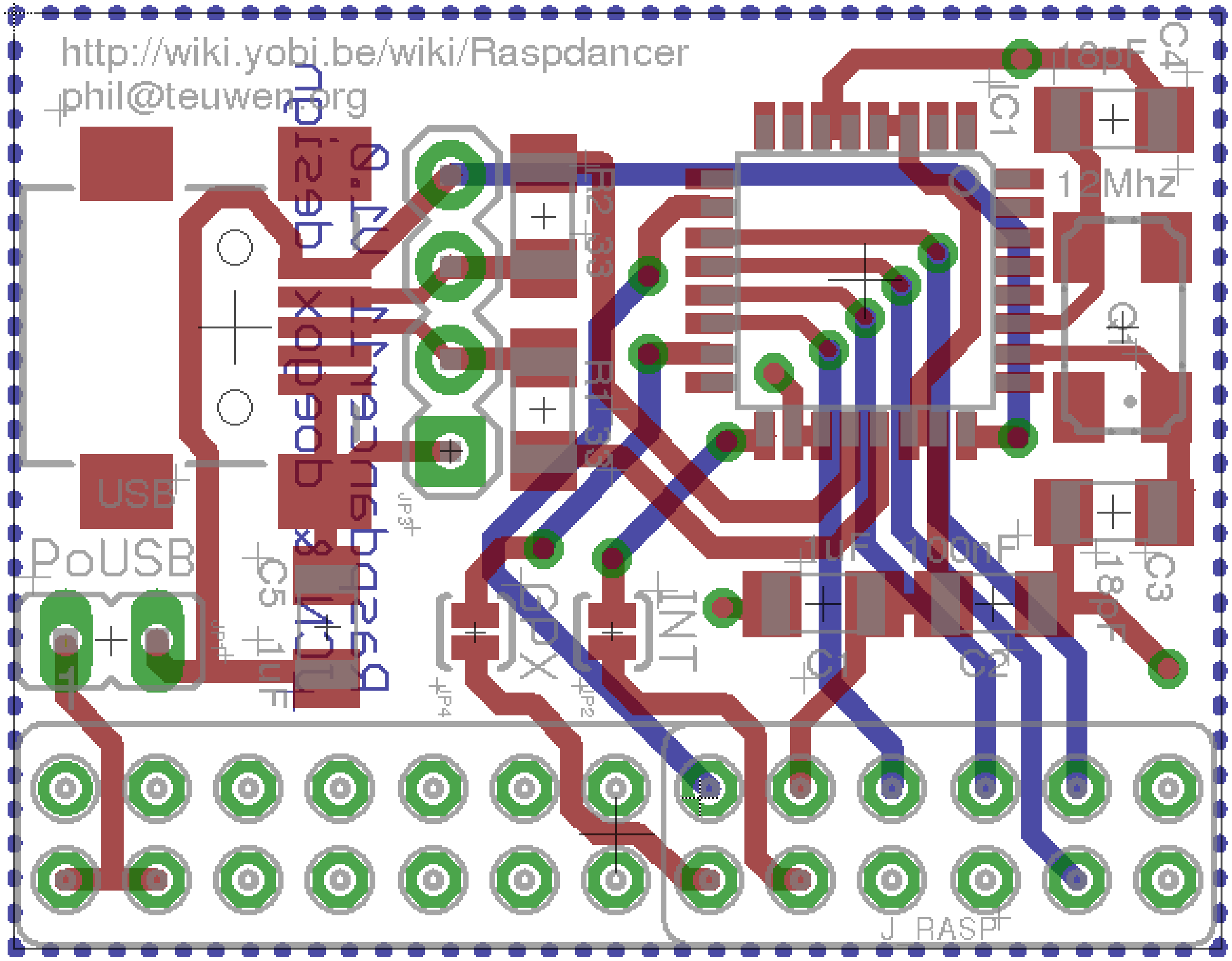
Our GoodFET.py

```python
import spi
import RPi.GPIO as GPIO

class GoodFET:
    data=""
    def __init__(self, *args, **kargs):
        GPIO.setmode(GPIO.BOARD)
        # pin15=GPIO22 is linked to MAX3420E -Reset
        GPIO.setup(15, GPIO.OUT, initial=GPIO.LOW)
        GPIO.output(15,GPIO.HIGH)
        spi.openSPI(speed=26000000)
    def __del__(self):
        spi.closeSPI()
        GPIO.cleanup()
    def writecmd(self, app, verb, count=0, data=[]):
        if verb: # ignore all but R/W cmd
            return
        if isinstance(data,str):
            data = [ord(x) for x in data]
        data = tuple(data)
        data = spi.transfer(data)
        self.data = "".join([chr(x) for x in data])
    def serInit(self):
        pass
```

26MHz!

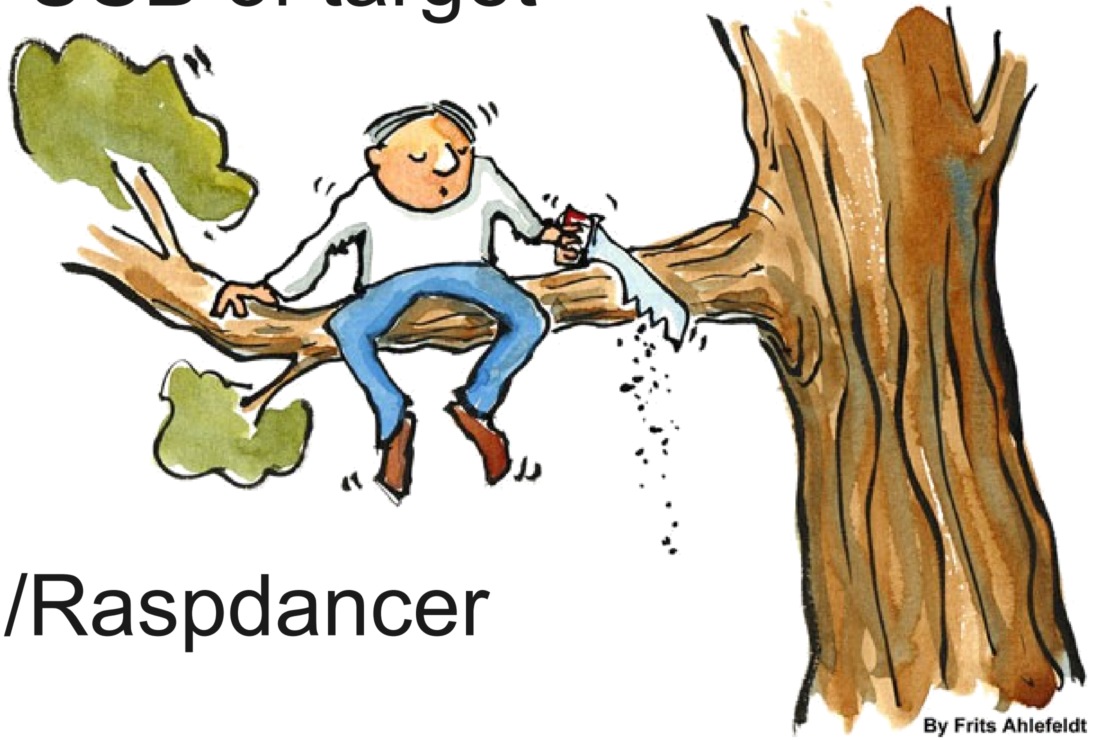
One step ahead



Thanks to Jean-Christophe Nicaise for his help!

Advantages

- Reuse of all the good GoodFETMAXUSB.py
- Speed & price
- Potentially autonomous or remote-controlled
- Can be powered over USB of target
  but beware...



http://wiki.yobi.be/wiki/Raspdancer