Hacking the CampCopter

Dominic Spill & Michael Ossmann
Great Scott Gadgets

CX-10A

4 CHANNELS

3D ROTATION

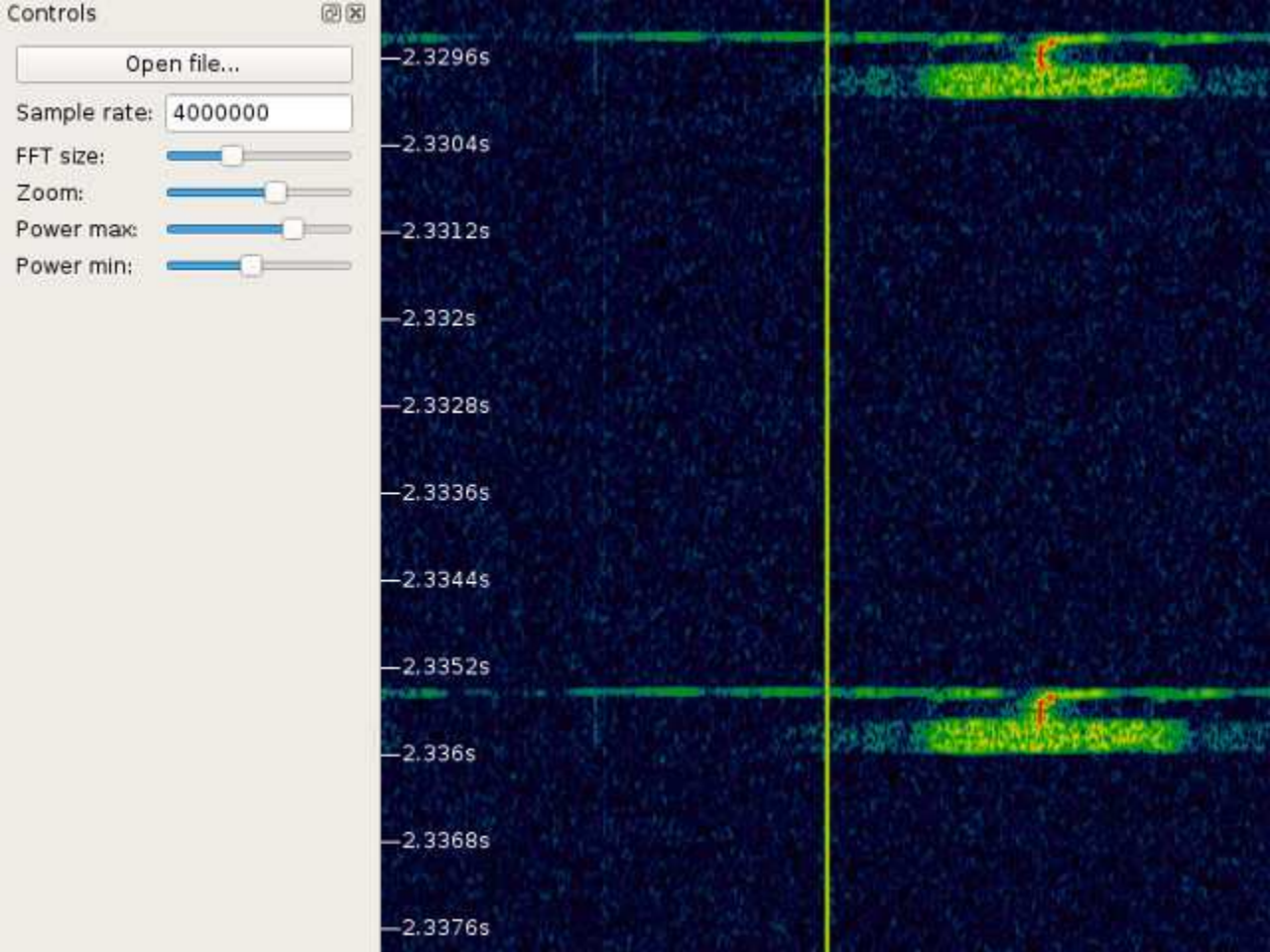
ITEM NO. CX-10A SIZE OF PRODUCT 4*4*2.2 CM
SIZE OF COLOR BOX 7.5*10.7*9.5
SIZE OF CARTON 64*24.5*84
N.W./G.W. 19/15 KGS
FEATURES 4-AXIS RC QUADCOPTER
WITH 6-AXIS GYRO ROLLING FUNTION

2.4GHZ

INFRARED TECHNOLOGY


Swingline


HackRF One
GREAT SCOTT GADGETS
RESET
3V3
DFU
1V8
RF
USB
RX
TX
ANTENNA
CLKOUT
CLKIN


Controls
Open file...
Sample rate:
4000000
FFT size:
Zoom:
Power max:
Power min:
2.3296s
2.3304s
2.3312s
2.332s
2.3328s
2.3336s
2.3344s
2.3352s
2.336s
2.3368s
2.3376s

starts on 2402 MHz

1 Mbps GFSK

~500 µs Packets

every 6 ms

```
710f552f7d872649 e9 e290a704 7377ed96 d5b f fcc 2 801 5 30d 9 cacc 9649 50
^^^^^^^^^^^^^^^^ ^^ ^^^^^^^^ ^^^^^^^^ ^^^ ? ^^^ ? ^^^ ? ^^^ ? ???? ^^^^ ^^
   sync word     ph   CID      VID    rol   pit   thr   yaw        CRC  trailer
    ^^
    8 bits alternating

known whitening:
.............. .. ........ ........ ee1 . c76 . ... . 0b7 . .... .... ..
```

```
In [41]: a = BitStream('0xee1')

In [42]: b = BitStream('0xd5b')

In [43]: c = BitStream('0x6e1')

In [44]: d = BitStream('0xf9d')

In [45]: ab = a^b

In [46]: ab.reverse()

In [47]: ab.int
Out[47]: 1500

In [48]: ac = a^c

In [49]: ac.reverse()

In [50]: ac.int
Out[50]: 1

In [51]: ad = a^d

In [52]: ad.reverse()

In [53]: ad.int
Out[53]: 1000

In [54]:
```

replay
demo


Go


Google

Main Menu

Home
Forum
Wiki
Getting Started
FAQ
User Manual
User Manual for DEVO 6/8/12
User Manual for DEVO 7E/10
Supported Models
Downloads
Release Notes
Bug Tracker

TOPIC: JD 395 cx-10

JD 395 cx-10 17 Jul 2014 14:39

kamueone
OFFLINE
Posts: 2

The new green Board on the Cheerson cx-10 uses the same protocol as the JD 395.

Is there a way to use that protocol with a devo 7e?

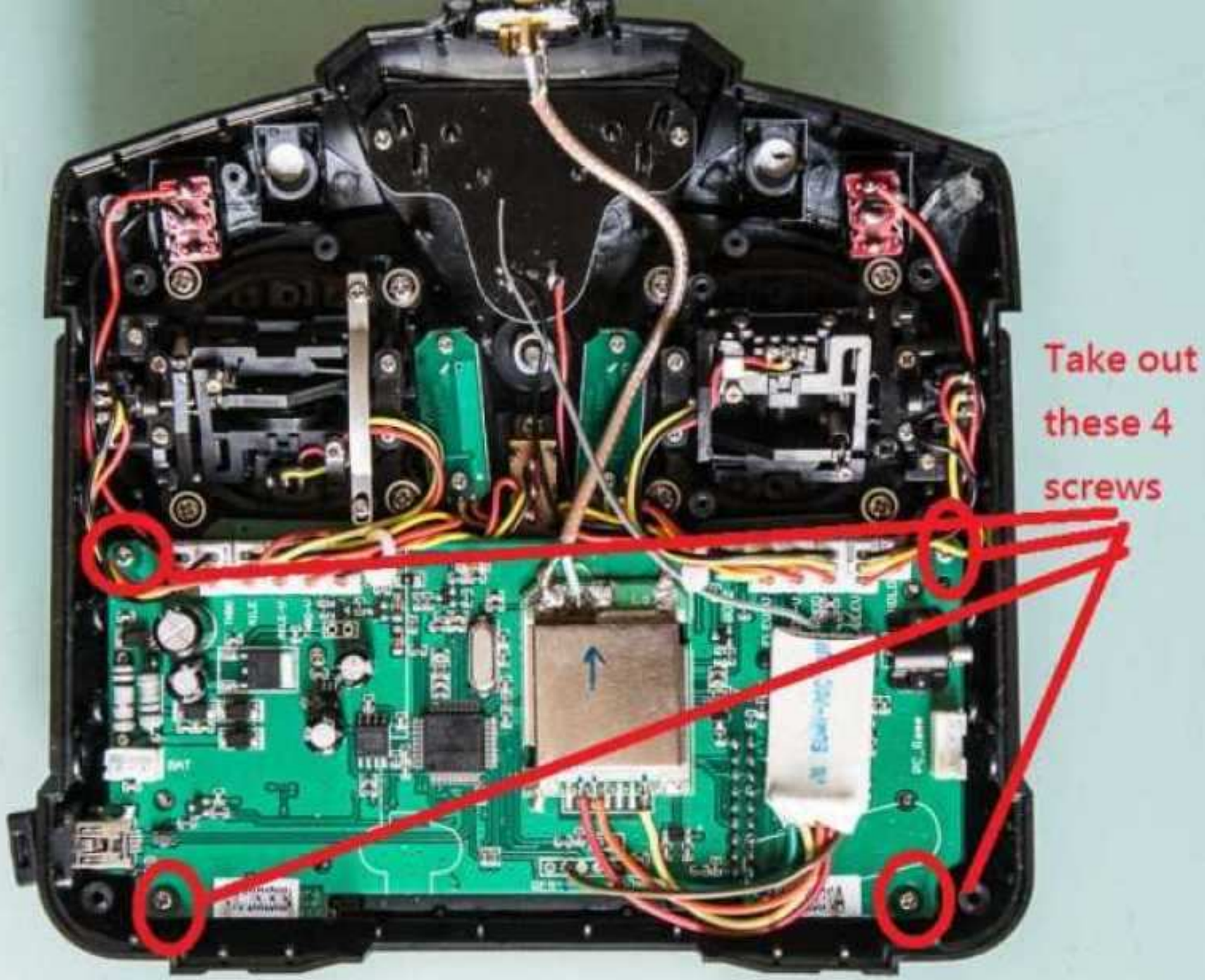
Thank you for your help,

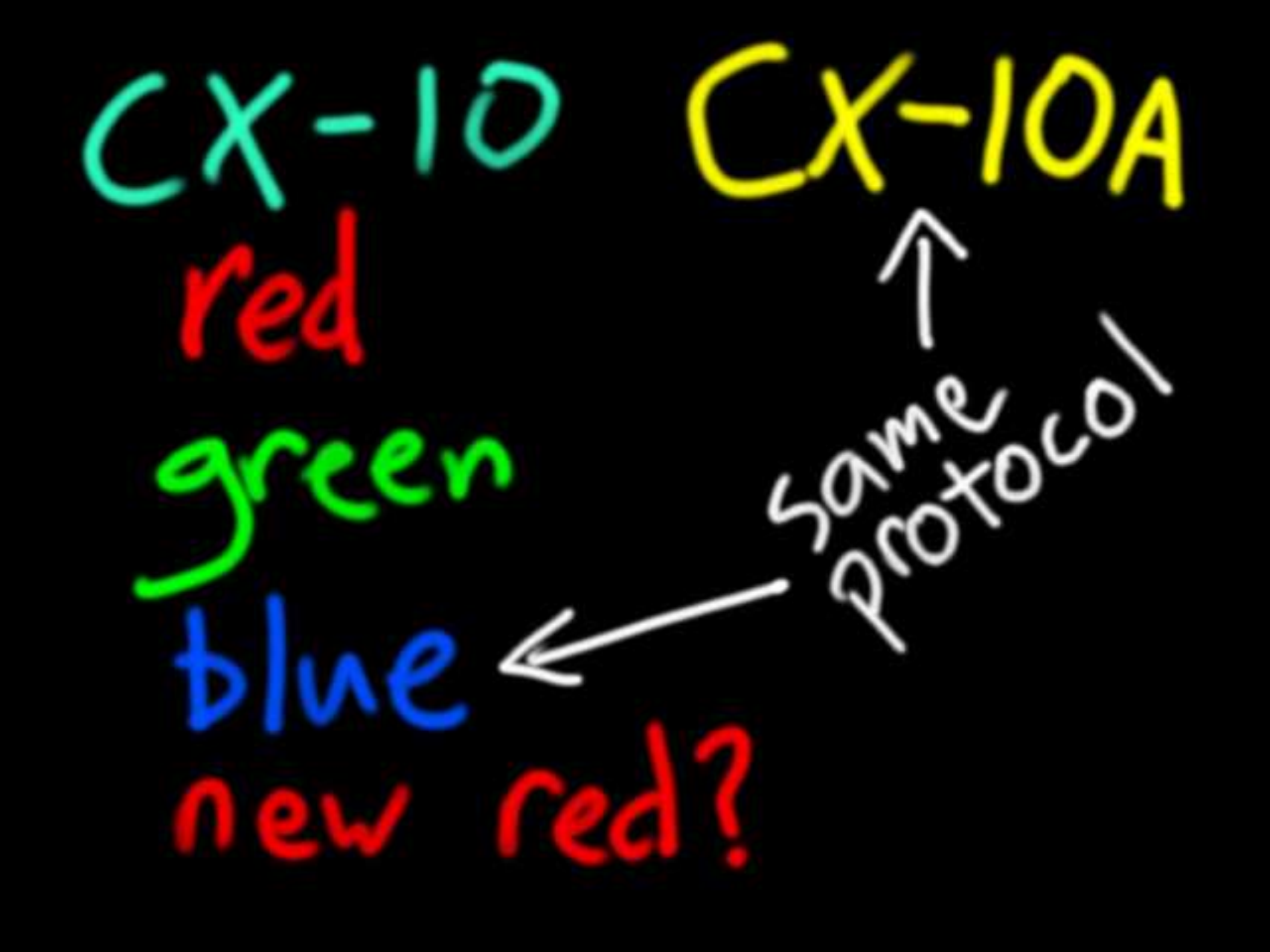
JD 395 cx-10 17 Jul 2014 15:59

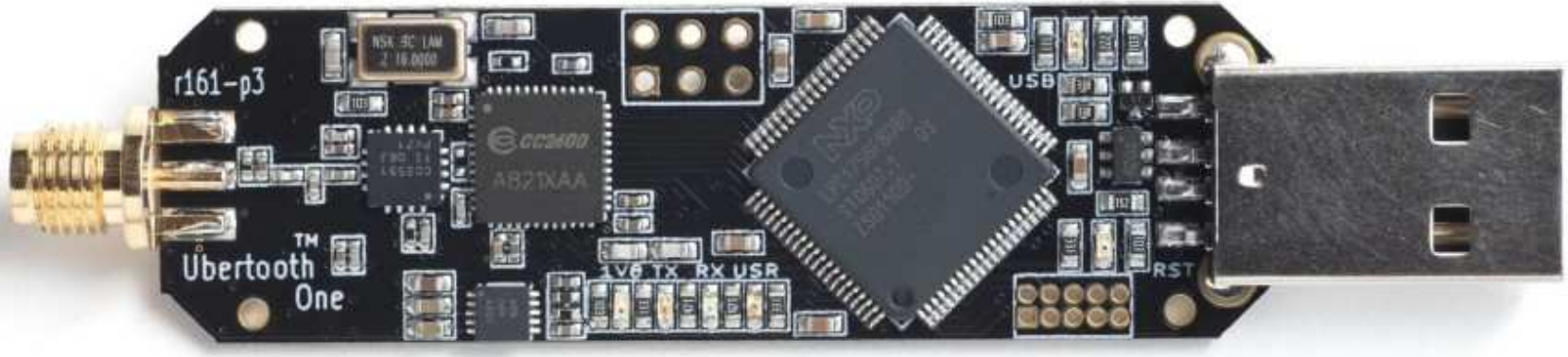
SeByDocKy
OFFLINE
Posts: 887

kamueone wrote:

> *The new green Board on the Cheerson cx-10 uses the same protocol as the JD 395.*
>
> *Is there a way to use that protocol with a devo 7e?*
>
> *Thank you for your help,*


Take out
these 4
screws


CX-10
red
green
blue
new red?
CX-10A
same protocol


r161-p3
Übertooth
One
™
CC2400
AB2TXAA
USB
RST
3V6 TX RX USR


sturdy palm tree

dominicgs / sturdy-palm-tree

No description or website provided.

7 commits 3 branches 0 releases 1 contributor

Branch: **master** New pull request Find file Clone or download

dominicgs Fix arbitrary channel/modem settings for sniffing Latest commit `cf10f59` 4 days ago

SturdyPalmTree	Fix arbitrary channel/modem settings for sniffing	4 days ago
.gitignore	Initial commit	a month ago
LICENSE	Initial commit	a month ago

```
    pitch = unwhitened[143:127:-1].read('uint:16')
    throttle = unwhitened[159:143:-1].read('uint:16')
    yaw = unwhitened[171:159:-1].read('uint:12')
    flip = unwhitened[175:171:-1].read('uint:4')
    special = unwhitened[191:175:-1].read('uint:16')
    crc = packet[192:208].read('uint:16')

    if crc == cx10a_crc(packet):
        print "%02x %08x %08x %5d %5d %5d %5d %01x %04x %04x" \
            % (mode, cid, vid, roll, pitch, throttle, yaw, flip, special, crc)

def find_cx10a_packet(symbols):
    # search for whitened sync word
    if symbols.find('0x2f7d872649', 0, symbols.len - 216):
        decode_cx10a(symbols[symbols.pos:symbols.pos + 216])
        return True, symbols[symbols.pos + 216:]
    else:
        return False, symbols[symbols.len - 216:]

def ubertooth_rx():
    dev = Radio(Radio.UBERTOOTH)
    syncword = 0x2f7d8726
    dev.configure_radio(frequency=2402, freq_deviation=340, syncword=syncword)
    symbol_stream = bitstring.ConstBitStream()
    for metadata, pkt in dev.rx_pkts():
        print metadata
        print pkt.bin
        symbol_stream += pkt
        pkt_found, symbol_stream = find_cx10a_packet(symbol_stream)

if __name__ == "__main__":
    ubertooth_rx()
~
```

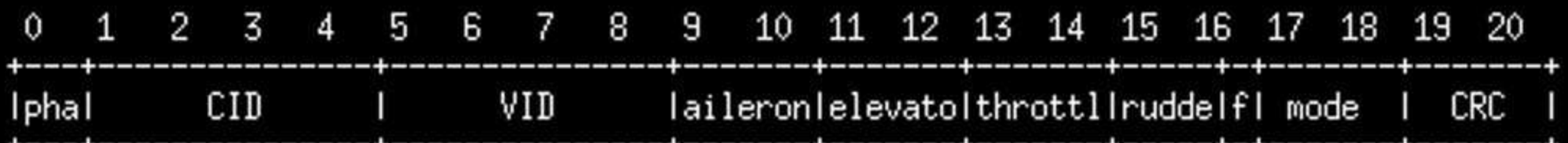
ubertooth sniffing demo

four-channel repeated frequency hopping sequence derived from the controller's unique identifier (CID). The controller transmits each packet on a new frequency in the hopping sequence, 5.25 ms after the start of the previous packet on the previous frequency.

2. Packet Payload Format

Packets in both binding phase and flying phase have the same format. The packet payload has a fixed length of 21 bytes divided into 9 fields. Each field is transmitted LSB first.

field name	byte offset	length in bytes	description
phase	0	1	operating phase
CID	1	4	controller identifier
VID	5	4	vehicle identifier
aileron	9	2	aileron (roll) control
elevator	11	2	elevator (pitch) control
throttle	13	2	throttle control
rudder	15	1.5	rudder (yaw) control
flip	16.5	0.5	flip control
mode	17	2	flight control mode
CRC	19	2	cyclic redundancy check

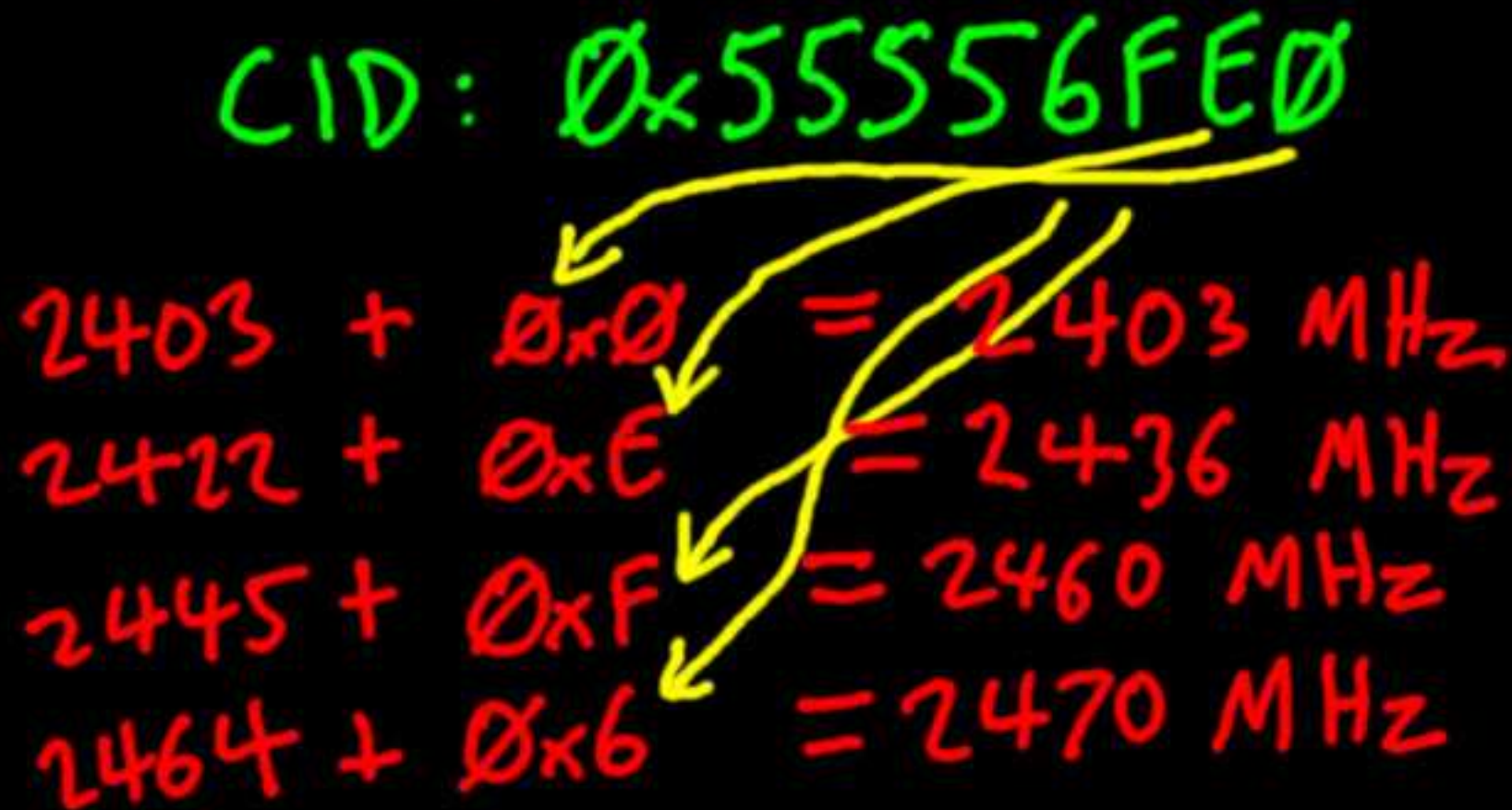


2.1 Phase Field


binding request
reply
confirmation
ack
2402 MHz

frequency hopping

CID: 0x5555 6FE0



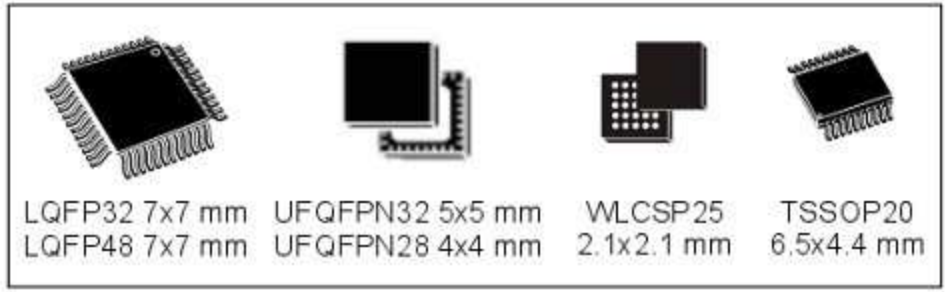
STM32F031x4 STM32F031x6

ARM®-based 32-bit MCU with up to 32 Kbyte Flash, 9 timers, ADC and communication interfaces, 2.0 - 3.6 V

Datasheet - production data

Features

- Core: ARM® 32-bit Cortex®-M0 CPU, frequency up to 48 MHz
- Memories
  - 16 to 32 Kbytes of Flash memory
  - 4 Kbytes of SRAM with HW parity
- CRC calculation unit
- Reset and power management
  - Digital and I/Os supply: 2.0 to 3.6 V
  - Analog supply: V_{DDA} = from V_{DD} to 3.6 V
  - Power-on/Power-down reset (POR/PDR)
  - Programmable voltage detector (PVD)
  - Low power modes: Sleep, Stop and Standby
  - V_{BAT} supply for RTC and backup registers
- Clock management



LQFP32 7x7 mm
LQFP48 7x7 mm

UFQFPN32 5x5 mm
UFQFPN28 4x4 mm

WLCSP25
2.1x2.1 mm

TSSOP20
6.5x4.4 mm

  - 1 x 16-bit timer, with IC/OC and OCN, deadtime generation, emergency stop and modulator gate for IR control
  - 1 x 16-bit timer with 1 IC/OC
  - Independent and system watchdog timers
  - SysTick timer: 24-bit downcounter
- Calendar RTC with alarm and periodic wakeup from Stop/Standby
- Communication interfaces
  - 1 x I^2C interface, supporting Fast Mode Plus (1 Mbit/s) with 20 mA current sink, SMBus/PMBus, and wakeup from Stop mode

3.系统结构方框图

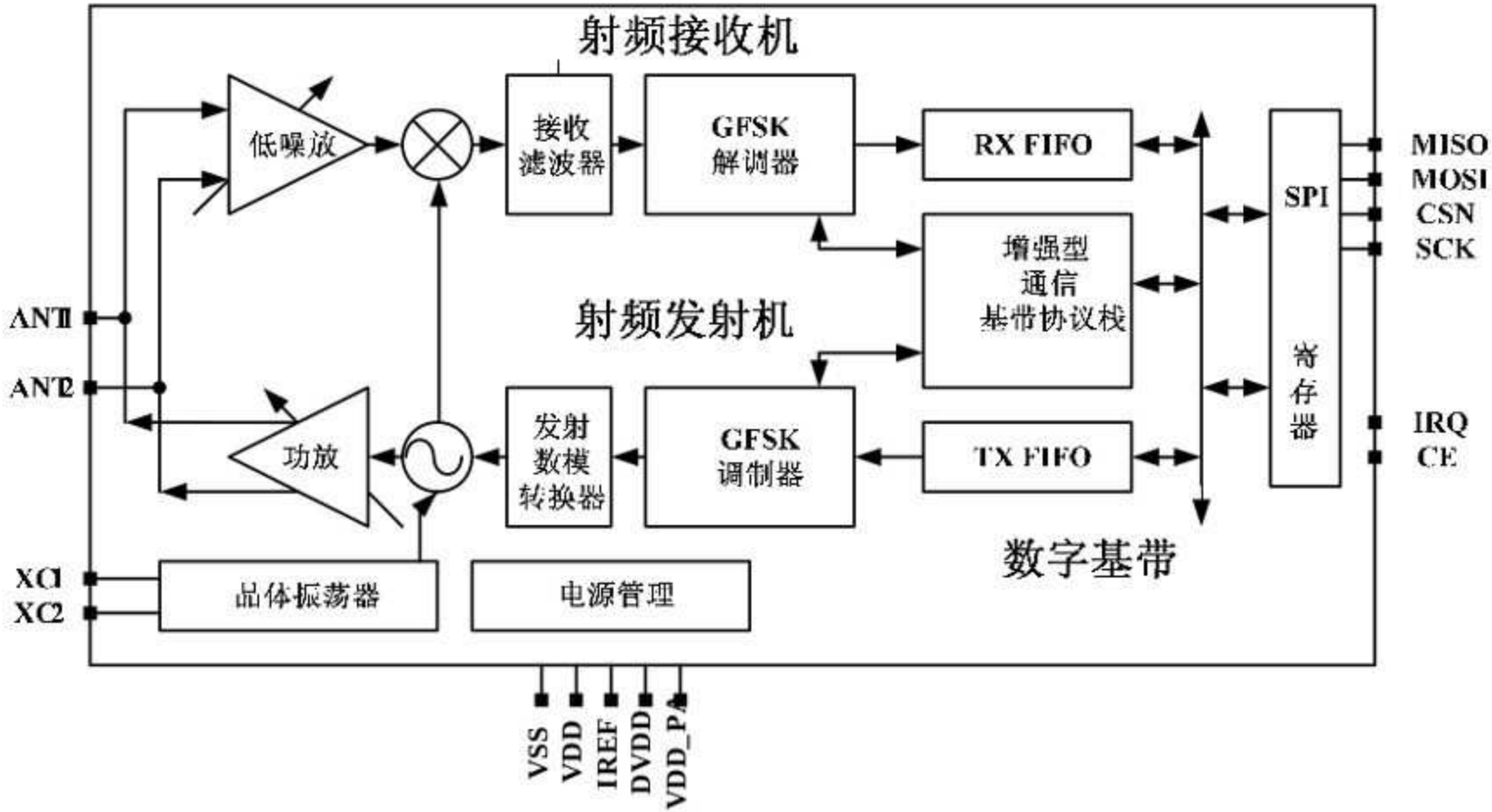
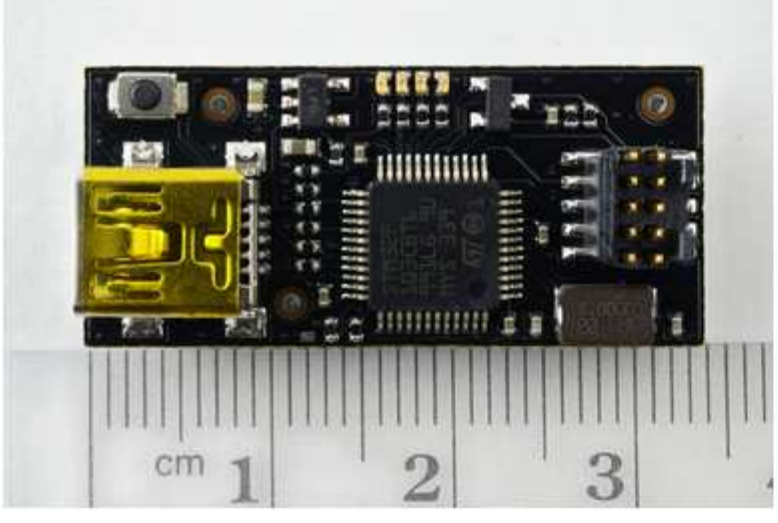


图1 XN297系统结构方框图



Black Magic Probe

$70.00

Black Magic Probe by Black Sphere Technologies is a JTAG and SWD Adapter used for programming and debugging ARM Cortex MCUs. Its the best friend of any ARM microcontroller developer.

Features:

- GDB server port without the need of special PC side software.
- TTL level serial interface
- SWD and JTAG support
- Semihosting support
- Works on Linux, Mac and Windows
- Works with Eclipse and other Integrated Development Environments
- Supports STM32, LPC11, LM3S - full support list
- DroneCode compatible

Dimensions:

- 33mm (1.3 in) x 15mm (0.6 in)
- 2.4g (0.85oz)

GND

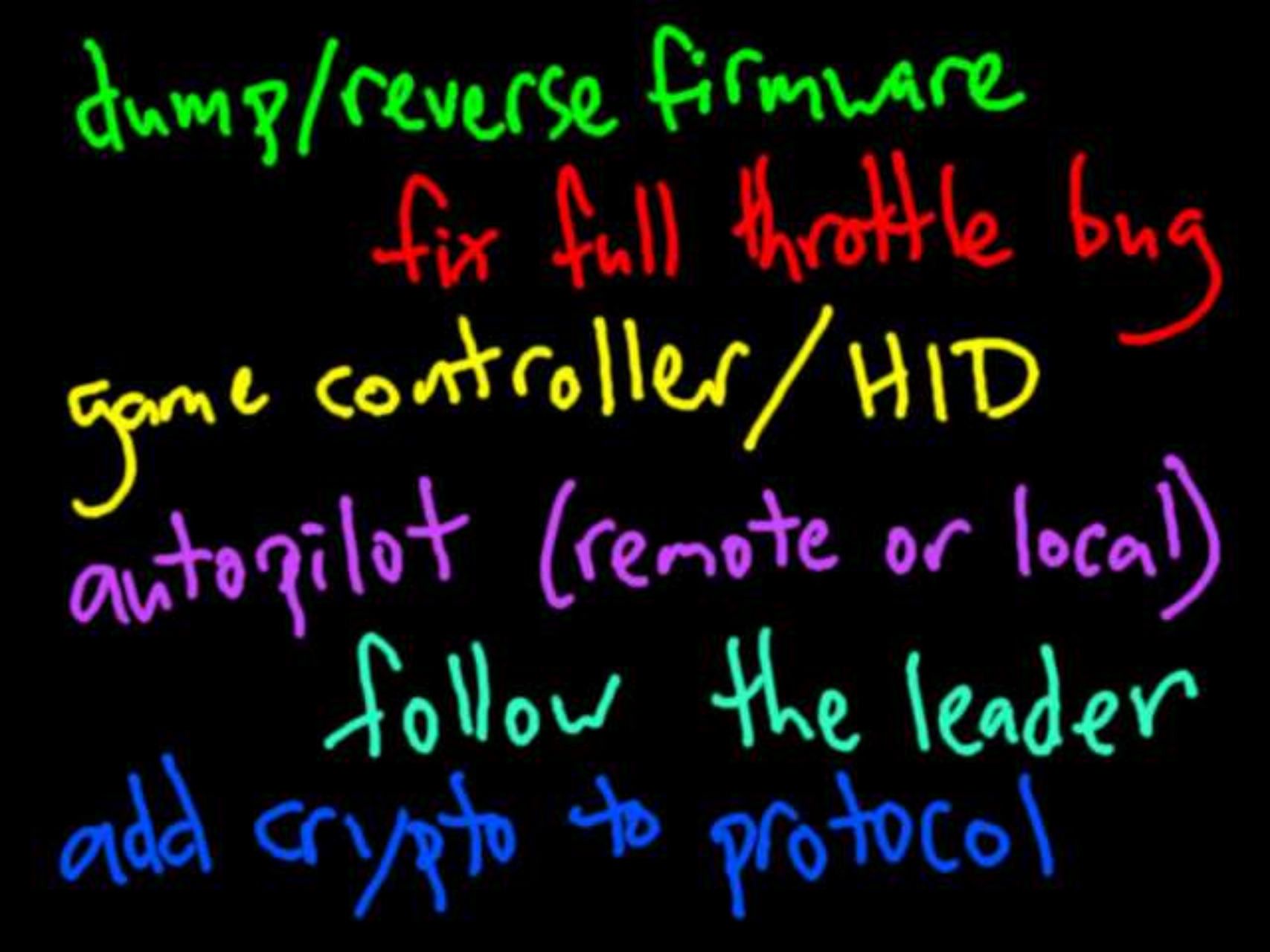
SWDIO

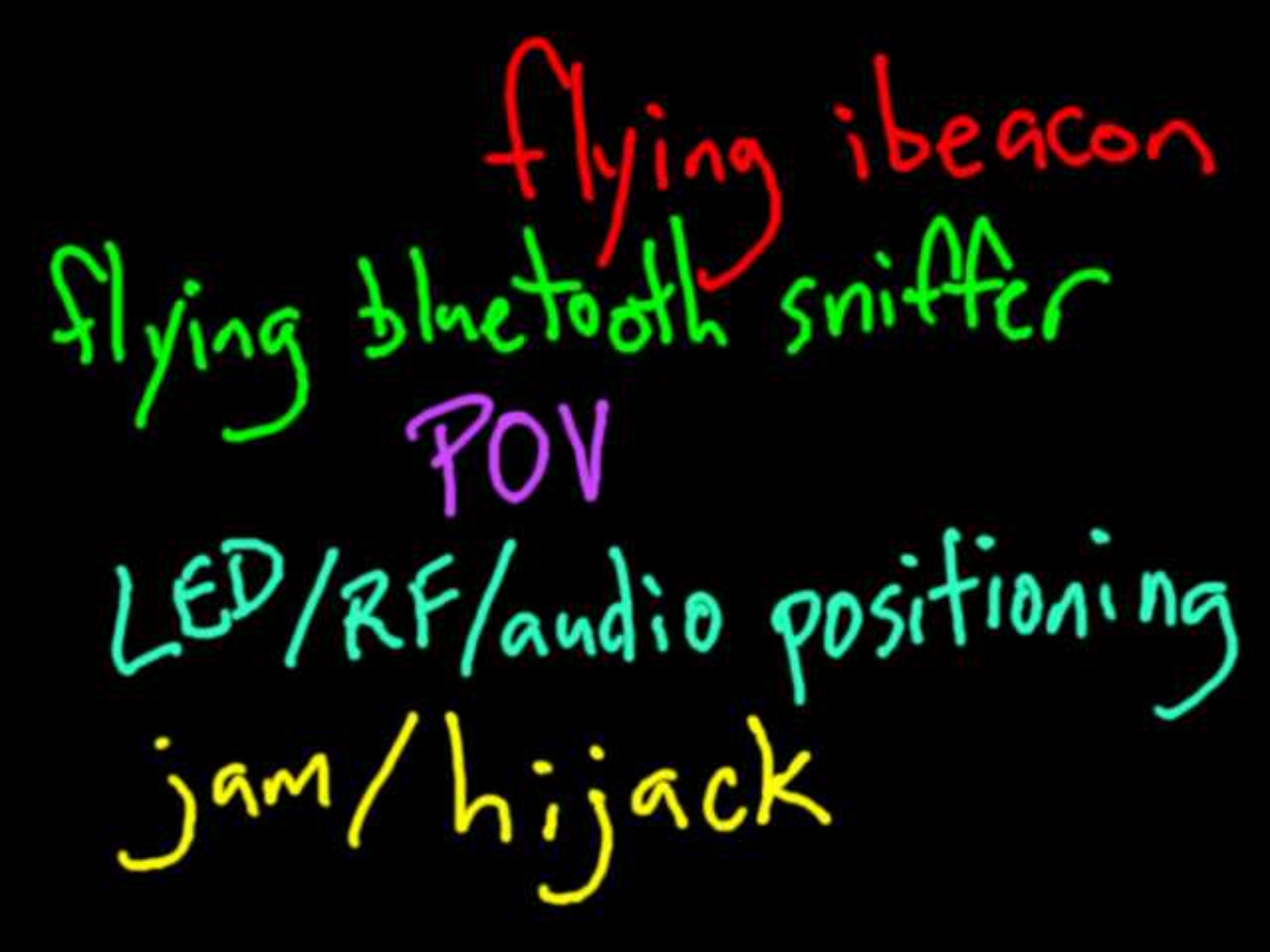
SWCLK

RST

3.3V

project ideas


dump/reverse firmware
fix full throttle bug
game controller/HID
autopilot (remote or local)
follow the leader
add crypto to protocol


flying ibeacon
flying bluetooth sniffer
POV
LED/RF/audio positioning
jam/hijack


sturdy palm tree
frequency hopping
binding
firmware functions
name

slide advance demo