# Designing RF Fuzzing Tools to Expose PHY Layer Vulnerabilities

Matt Knight, Ryan Speers

DEF CON 26

River Loop Security

# whois

## Matt Knight

- Senior Security Engineer at Cruise Automation
- RF Principal at River Loop Security
- BE in EE from Dartmouth College
- Software, hardware, and RF engineer
- RF, SDR, and embedded systems

## Ryan Speers

- Co-founder at River Loop Security
- Director of Research at Ionic Security
- Computer Science from Dartmouth College
- Cryptography, embedded systems, IEEE 802.15.4, automated firmware analysis

River Loop Security

# Background



"Making and Breaking a Wireless IDS", Troopers14

"Speaking the Local Dialect", ACM WiSec

- Ryan Speers, Sergey Bratus, Javier Vazquez, Ray Jenkins, bx, Travis Goodspeed, & David Dowd
- Idiosyncrasies in PHY implementations

Mechanisms for automating:

- RF fuzzing
- Bug discovery
- PHY FSM fingerprint generation

# Agenda



1. Overview of traditional fuzzing techniques (software and networks)
    > How these do and don't easily map to RF
2. RF fuzzing overview and state of the art
3. Ideal fuzzer design
4. TumbleRF introduction and overview
5. TumbleRF usage example
6. Introducing Orthrus

# Traditional Fuzzing Techniques

# What is fuzzing?



Measured application of pseudorandom input to a system

Why fuzz?

- Automates discovery of crashes, corner cases, bugs, etc.
- Unexpected input → unexpected state

# What can one fuzz?



Fuzzers generally attach to system interfaces, namely I/O:

- File format parsers
- Network interfaces
- Shared memory

# Software Fuzzing State of the Art



Abundant fully-featured software fuzzers

- AFL / AFL-Unicorn
- Peach
- Scapy

Software is easy to instrument and hook at every level

What else can one fuzz?

# Other Applications of Fuzzing

# Fuzzing Hardware



Challenges:

- H/W is often unique, less “standard interfaces” to measure on
- May not be able to simulate well in a test harness

Some Existing Techniques:

- AFL-Unicorn: simulate firmware in Unicorn to fuzz
- Bus Pirate: permutes pinouts and data rates to discover digital buses
- JTAGulator: permutes pinouts that could match unlocked JTAG
- …

# Fuzzing RF



WiFuzz

- MAC-focused 802.11 protocol fuzzer

Marc Newlin's Mousejack research

- Injected fuzzed RF packets at nRF24 HID dongles while looking for USB output

isotope:

- IEEE 802.15.4 PHY fuzzer

# Existing RF Fuzzing Limitations



RF fuzzing projects are siloed / protocol-specific

- COTS radio chipsets
- Generally limited to MAC layer and up

RF state is hard to instrument

- What constitutes a crash / bug / etc?

Implicit trust in chipset – one can only see what one's radio tells you is happening

# Trust and Physical Layer Vulnerabilities



Not all PHY state machines are created equal!

Radio chipsets implement RF state machines *differently*

- Differences can be fingerprinted and exploited
- Initial results on 802.15.4 were profound
- Specially-crafted PHYs can target certain chipsets while avoiding others

# RF PHYs: A Primer

# How Radios Work



Transmitter: digital data (bits) → analog RF energy

discrete → continuous

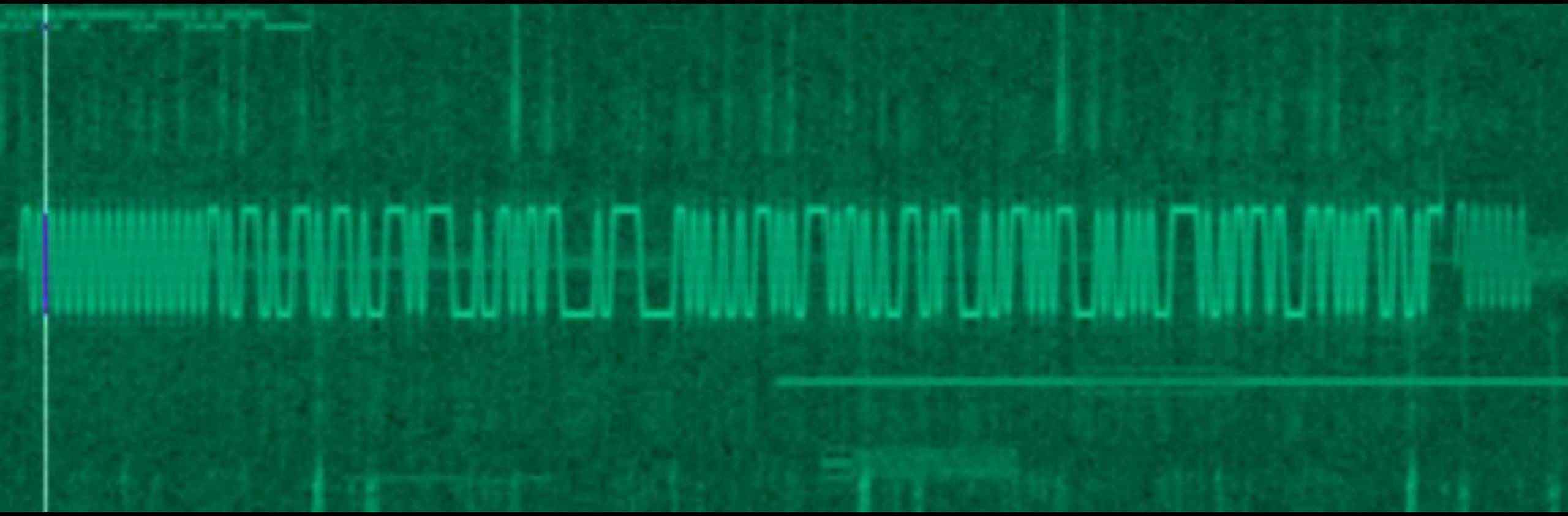
Receiver: analog RF energy → digital data (bits)

continuous → discrete

Receiving comes down to sampling and synchronization!

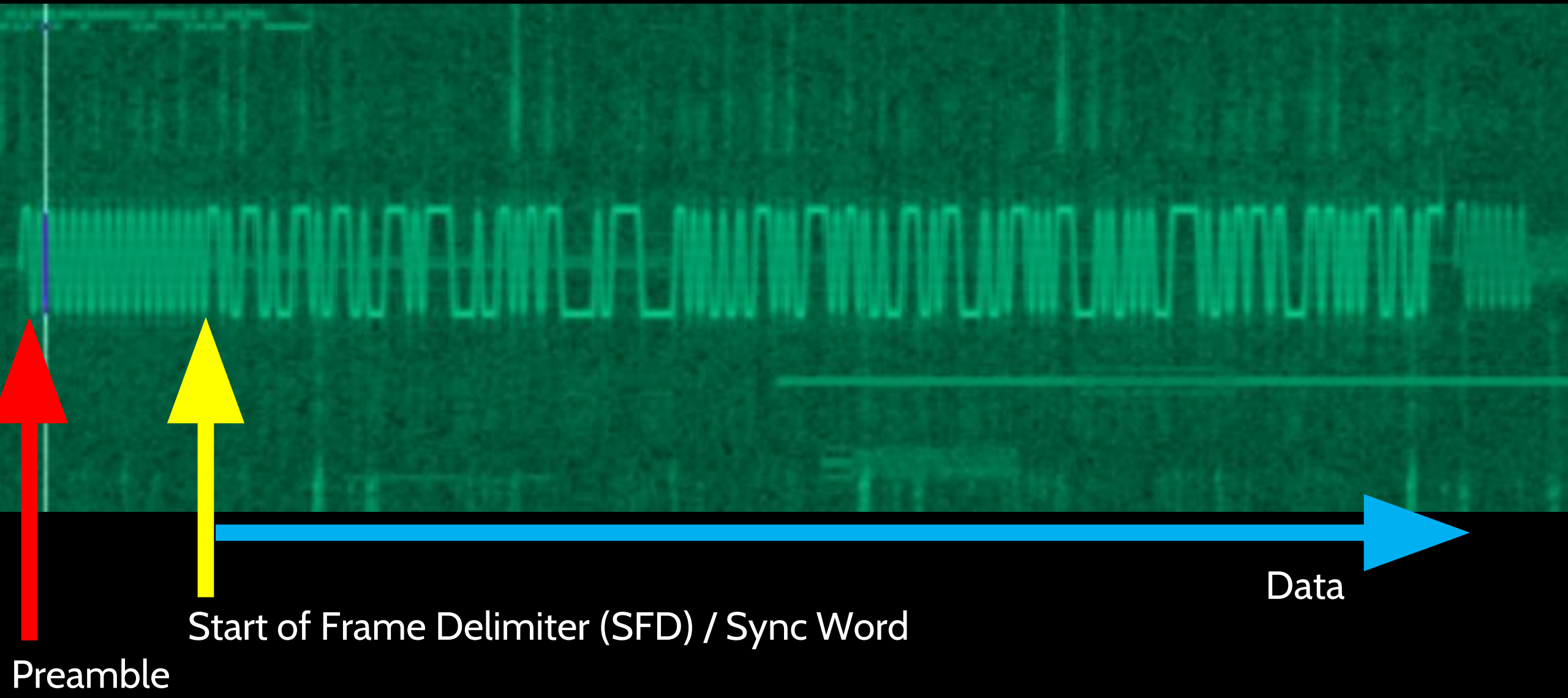
# Digitally Modulated Waveforms





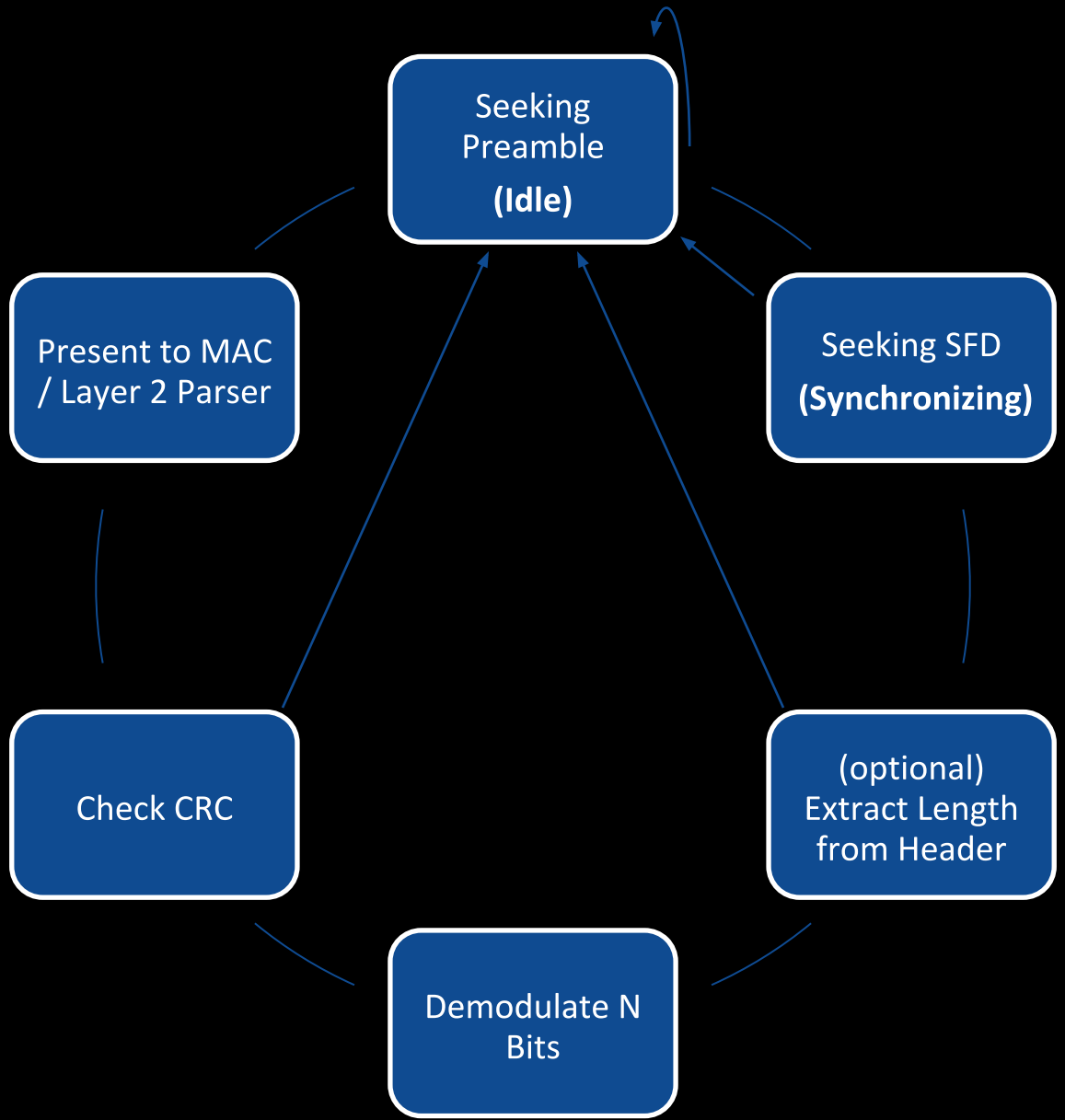
# Digitally Modulated Waveforms





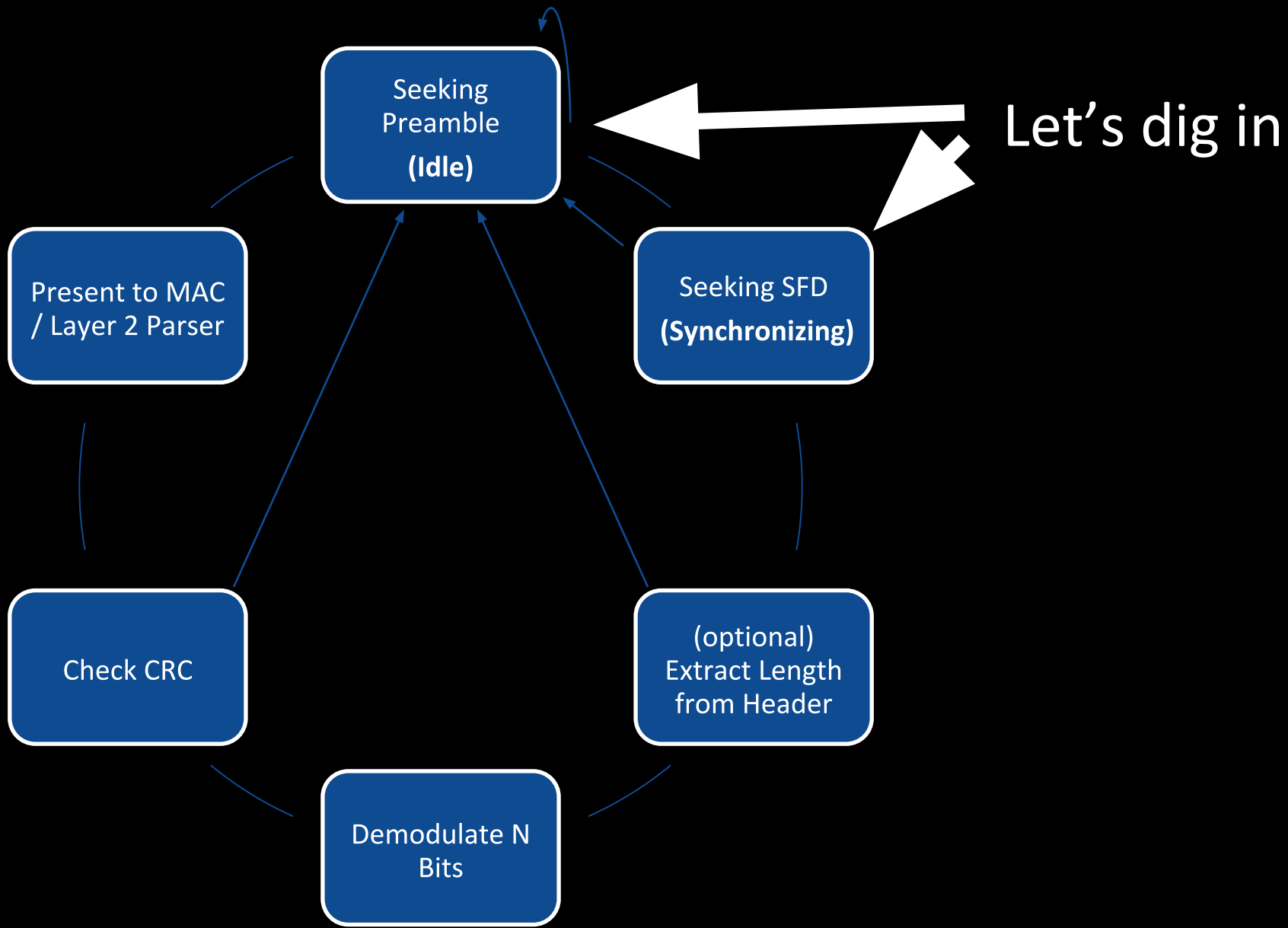
# RF PHY State Machines





# RF PHY State Machines

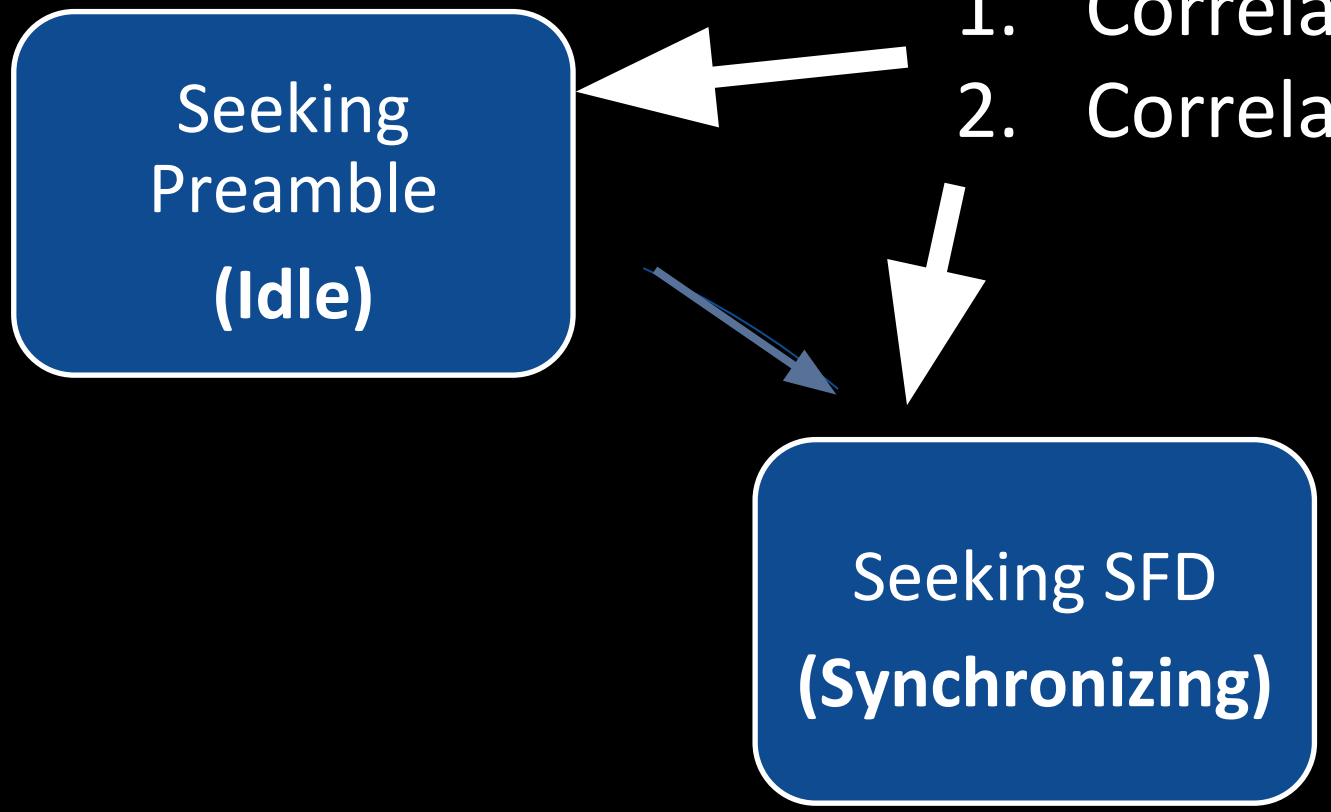




# RF PHY State Machines



**Correlation = shift register clocking bits through at symbol rate looking for a pattern**




1. Correlator looks for [1,0,1,0,…]
2. Correlator looks for [magic number]

If found, a packet is on-air

# RF PHY State Machines



| | → Shift Register → | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| → RF Symbol Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Preamble Correlation Value | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| XOR Result | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Hamming Distance | 4 | | | | | | | |

Seeking Preamble **(Idle)**

Hamming Distance
- # bits that are different between two values
- If 0, values are equal

When Hamming Distance <= some threshold, a preamble has been detected

# RF PHY State Machines



| | → Shift Register → | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| → RF Symbol Value | ***1*** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Preamble Correlation Value | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| XOR Result | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Hamming Distance | 5 | | | | | | | |

Seeking Preamble

**(Idle)**

Hamming Distance

- # bits that are different between two values
- If 0, values are equal

When Hamming Distance <= some threshold, a preamble has been detected

# RF PHY State Machines



| | → Shift Register → | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| → RF Symbol Value | ***0*** | ***1*** | 0 | 0 | 0 | 0 | 0 | 0 |
| Preamble Correlation Value | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| XOR Result | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| Hamming Distance | 3 | | | | | | | |

Seeking Preamble **(Idle)**

Hamming Distance
- # bits that are different between two values
- If 0, values are equal

When Hamming Distance <= some threshold, a preamble has been detected

# RF PHY State Machines



| | → Shift Register → | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| → RF Symbol Value | ***1*** | ***0*** | ***1*** | 0 | 0 | 0 | 0 | 0 |
| Preamble Correlation Value | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| XOR Result | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| Hamming Distance | 6 | | | | | | | |

Seeking Preamble

**(Idle)**

Hamming Distance

- # bits that are different between two values
- If 0, values are equal

When Hamming Distance <= some threshold, a preamble has been detected

# RF PHY State Machines



| | → Shift Register → | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| → RF Symbol Value | ***0*** | ***1*** | ***0*** | ***1*** | 0 | 0 | 0 | 0 |
| Preamble Correlation Value | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| XOR Result | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| Hamming Distance | 2 | | | | | | | |

Seeking Preamble **(Idle)**

Hamming Distance
- # bits that are different between two values
- If 0, values are equal

When Hamming Distance <= some threshold, a preamble has been detected

# RF PHY State Machines



| | → Shift Register → | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| → RF Symbol Value | ***1*** | ***0*** | ***1*** | ***0*** | ***1*** | 0 | 0 | 0 |
| Preamble Correlation Value | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| XOR Result | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| Hamming Distance | 7 | | | | | | | |

Seeking Preamble **(Idle)**

Hamming Distance
- # bits that are different between two values
- If 0, values are equal

When Hamming Distance <= some threshold, a preamble has been detected

# RF PHY State Machines



| | → Shift Register → | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| → RF Symbol Value | ***0*** | ***1*** | ***0*** | ***1*** | ***0*** | ***1*** | 0 | 0 |
| Preamble Correlation Value | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| XOR Result | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Hamming Distance | 1 | | | | | | | |

Seeking Preamble **(Idle)**

Hamming Distance
- # bits that are different between two values
- If 0, values are equal

When Hamming Distance <= some threshold, a preamble has been detected

# RF PHY State Machines



| | → Shift Register → | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| → RF Symbol Value | ***1*** | ***0*** | ***1*** | ***0*** | ***1*** | ***0*** | ***1*** | 0 |
| Preamble Correlation Value | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| XOR Result | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Hamming Distance | 8 | | | | | | | |

Seeking Preamble **(Idle)**

Hamming Distance
- # bits that are different between two values
- If 0, values are equal

When Hamming Distance <= some threshold, a preamble has been detected

# RF PHY State Machines



| | → Shift Register → | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| → RF Symbol Value | ***0*** | ***1*** | ***0*** | ***1*** | ***0*** | ***1*** | ***0*** | ***1*** |
| Preamble Correlation Value | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| XOR Result | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Hamming Distance | **0** | | | | | | | |

Seeking Preamble

**(Idle)**

Hamming Distance

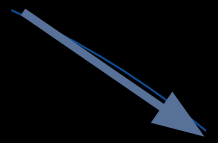
- # bits that are different between two values
- If 0, values are equal

When Hamming Distance <= some threshold, a preamble has been detected

# RF PHY State Machines



| | → Shift Register → | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| → RF Symbol Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Preamble Correlation Value | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| XOR Result | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Hamming Distance | 4 | | | | | | | |



Seeking SFD
**(Synchronizing)**

Repeat the process, correlating for the SFD value instead, to find the start of the PHY data unit

# Sync Words and Magic Numbers



Turns out not all sync words are created equally

- `0x00000000 == 802.15.4 Preamble`
- `0xA7 == 802.15.4 Sync Word`

The isotope research showed some chipsets correlated on “different” preambles / sync words than others

# Sync Words and Magic Numbers



Turns out not all sync words are created equally

- `0x00000000 == 802.15.4 Preamble`
- `0xA7 == 802.15.4 Sync Word`

**strategically malformed**

The isotope research showed some chipsets correlated on ~~"different"~~ preambles / sync words than others

# Sync Words and Magic Numbers



Turns out not all sync words are created equally

- `0x`**`XXXX`**`0000 == 802.15.4 Preamble`
- `0xA7 == 802.15.4 Sync Word`

**strategically malformed**

The isotope research showed some chipsets correlated on ~~"different"~~ preambles / sync words than others

**Short preamble?**

# Sync Words and Magic Numbers



Turns out not all sync words are created equally

- 0x**XXXX**0000 == 802.15.4 Preamble
- 0xAF == 802.15.4 Sync Word

**strategically malformed**

The isotope research showed some chipsets correlated on ~~"different"~~ preambles / sync words than others

**Short preamble? Flipped bits in SFD?**

# Fuzzing Makes Discovery Systematic

# Ideal RF Fuzzer Design

# Ideal Features



Extensible: easy to hook up new radios

Flexible: modular to enable plugging and playing different engines / interfaces / test cases

Reusable: re-use designs from one protocol on another

Comprehensive: exposes PHY in addition to MAC

# TumbleRF

# TumbleRF



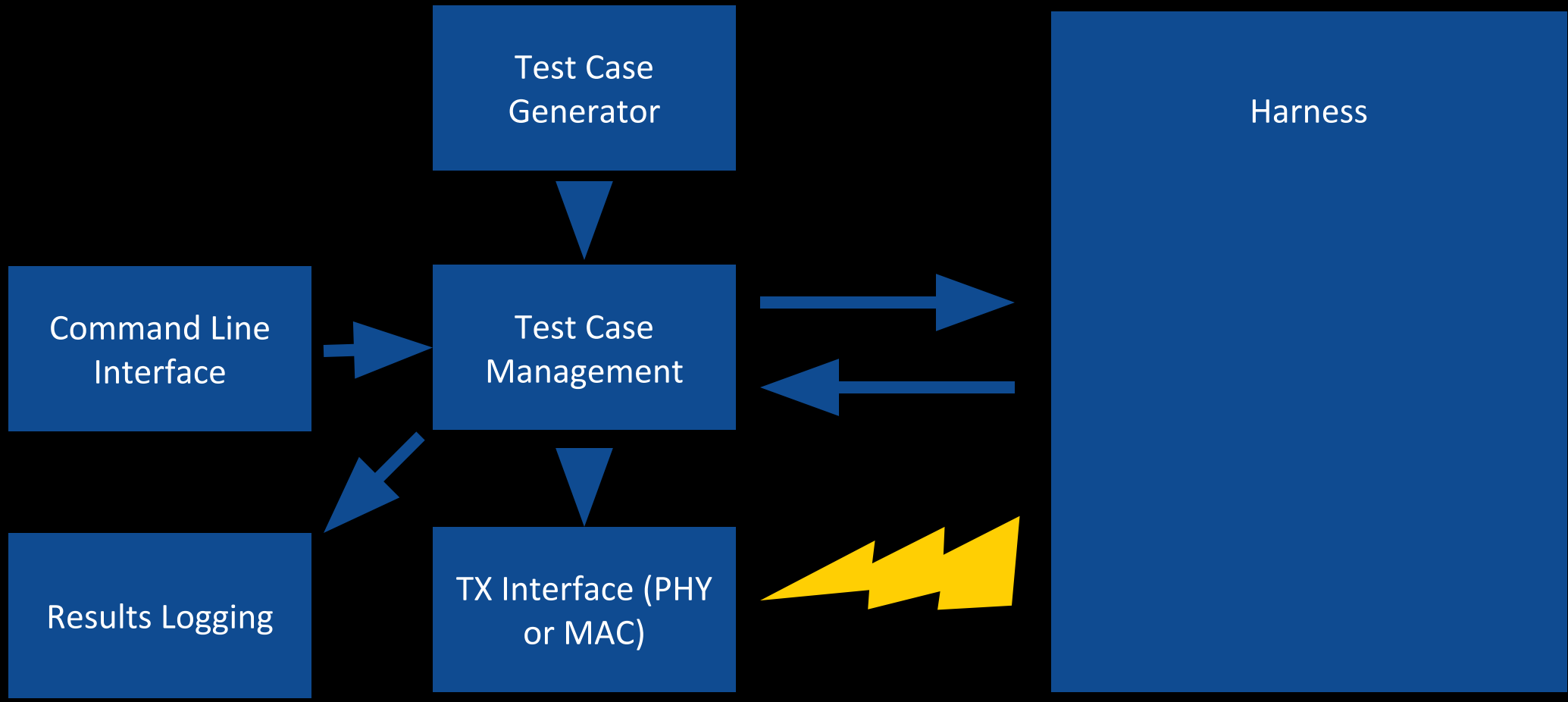
Software framework enabling fuzzing arbitrary RF protocols

Abstracts key components for easy extension:

- Radio API
- Test case generation API

# TumbleRF Architecture





# Interfaces



RF injection/sniffing functions abstracted to generic template

To add a new radio, inherit base Interface class and redefine its functions to map to the radio driver:

```
[set/get]_channel()
[set/get]_sfd()
[set/get]_preamble()
tx()
rx_start()
rx_stop()
rx_poll()
```

# Generators



Rulesets for generating fuzzed input (pythonically)
Extend to interface with software fuzzers of your choice

Implement 2 functions:

```
yield_control_case()
yield_test_case()
```

Three generators currently:

- Preamble length (isotope)
- Non-standard symbols in preamble (isotope)
- Random payloads in message

# Harnesses



Monitor the device under test to evaluate test case results

Manage device state in between tests

Three handlers currently:

- Received Frame Check: listen for given frames via an RF interface
- SSH Process Check: check whether processes on target crashed (beta)
- Serial Check: watch for specific output via Arduino (beta)

# Test Cases



Coordinate the generator, interface, and harness. Typically very lightweight.

Extend BaseCase to implement `run_test()`

*or build upon others, e.g.:*

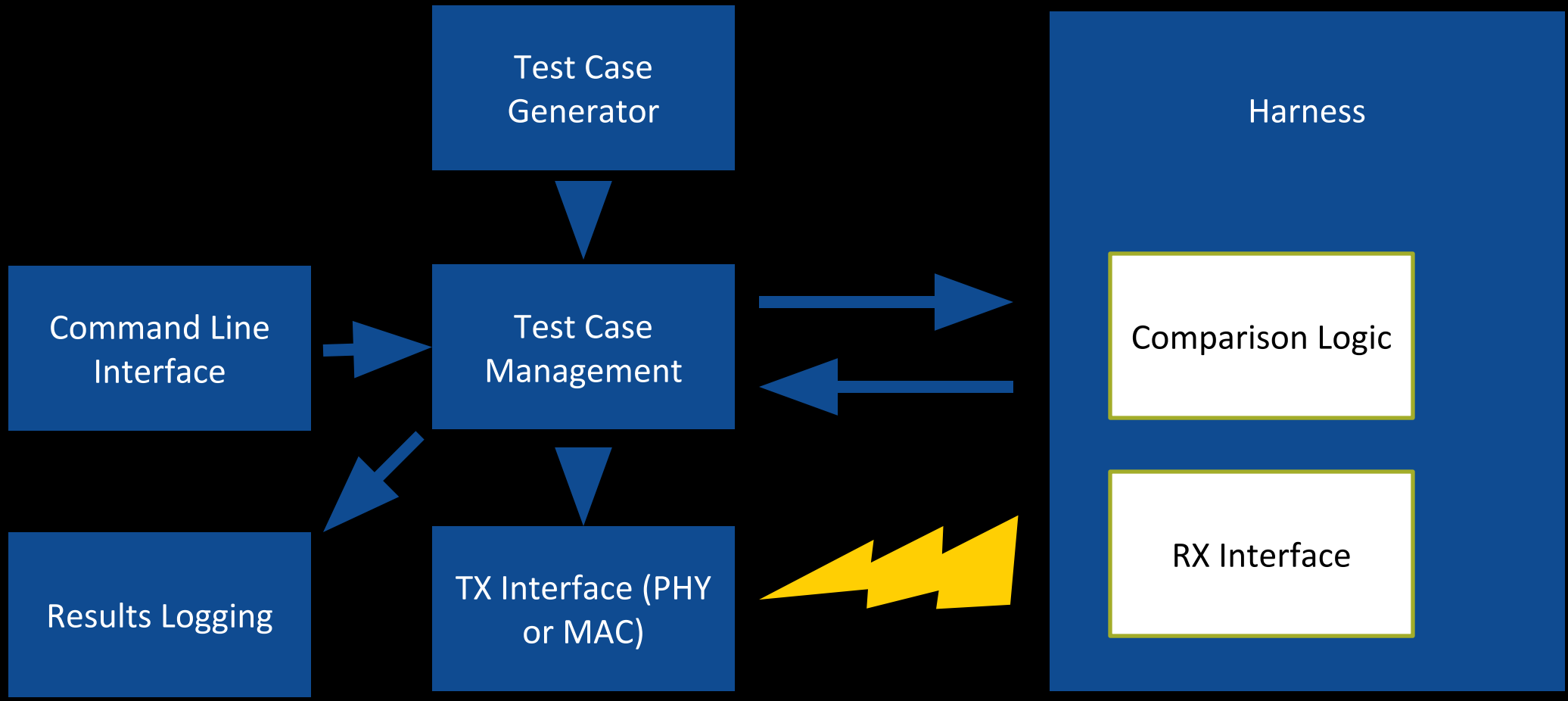
Extend AlternatorCase to implement:

```
does_control_case_pass()
throw_test_case()
```

Alternates test cases with known-good control case to check for crashes / ensure interface is still up

# TumbleRF Architecture: Demo Setup





# Example Generated Data: Preamble Length

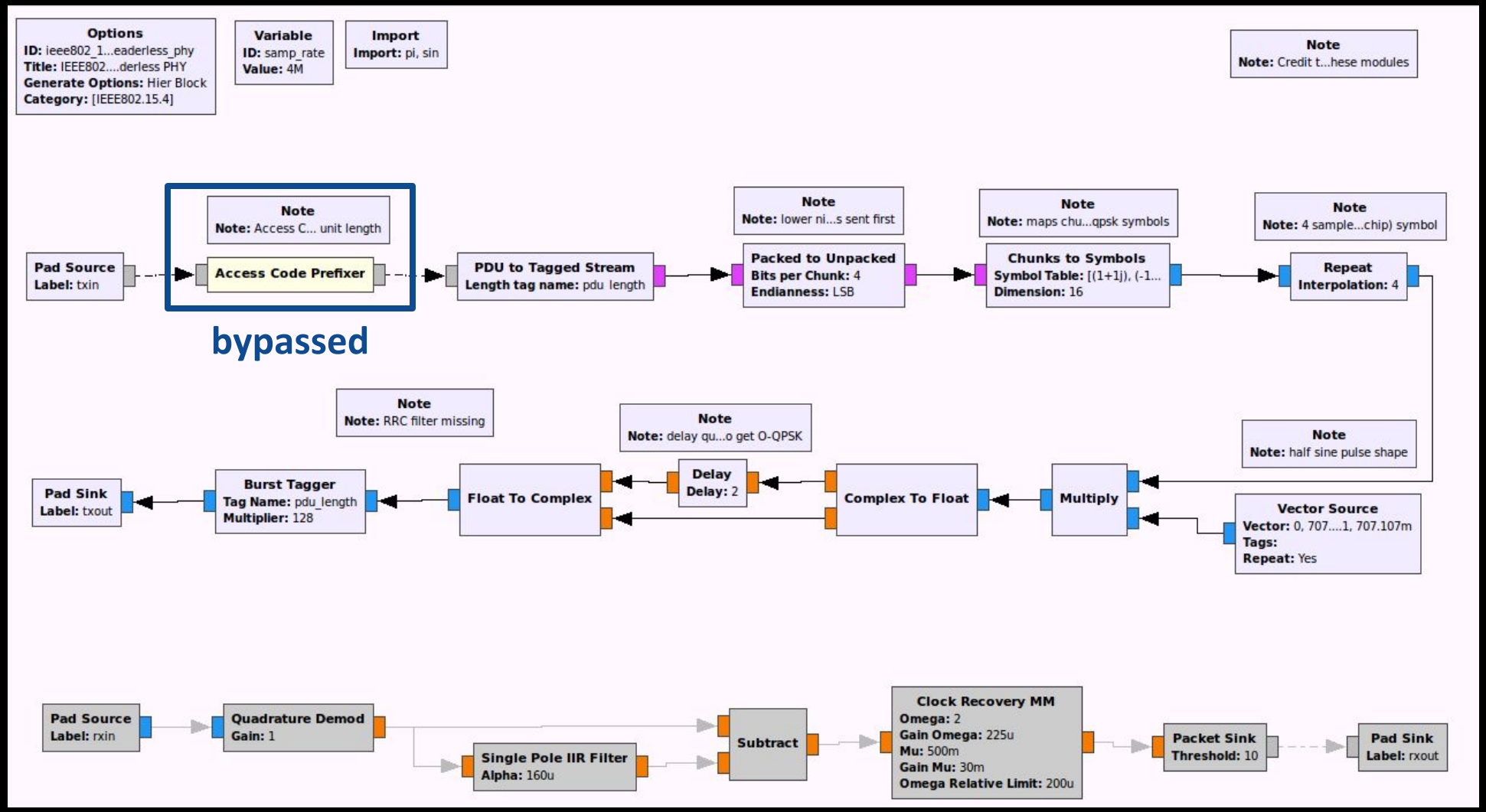


Standard 802.15.4 PHY Header == 0x00000000 + 0xA7 + 0xLL

| Preamble | | | | SFD | Length |
|---|---|---|---|---|---|
| | | | | 0xA7 | 0xLL |

# Example Generated Data: Preamble Length






Modify GNU Radio *gr-ieee802-15-4* to omit PHY header

Generate arbitrary PHY headers via TumbleRF test case generator

# Demo

# Results Dump



## TI CC2420

```
Test: preamble_length_apimote.json (using Dot15d4PreambleLengthGenerator)
        Case 0: 0 valid, 50 invalid     example case: a70a230800ffff000007fba6
        Case 1: 0 valid, 50 invalid     example case: 70aa308220f0ff0f0070d0eafa
        Case 2: 45 valid, 5 invalid     example case: 00a70a230804ffff00000757b6
        Case 3: 0 valid, 50 invalid     example case: 0070aa308260f0ff0f007010e0fb
        Case 4: 50 valid, 0 invalid     example case: 0000a70a230808ffff000007a387
        Case 5: 0 valid, 50 invalid     example case: 000070aa3082a0f0ff0f007050fff8
        Case 6: 50 valid, 0 invalid     example case: 000000a70a23080cffff0000070f97
        Case 7: 0 valid, 50 invalid     example case: 00000070aa3082e0f0ff0f007090f5f9
        Case 8: 48 valid, 2 invalid     example case: 00000000a70a230810ffff0000074be4
        Case 9: 0 valid, 50 invalid     example case: 0000000070aa308220f1ff0f0070d0c1fe
```

## Atmel AT86RF230

```
Test: preamble_length_rzusbstick.json (using Dot15d4PreambleLengthGenerator)
        Case 0: 0 valid, 50 invalid     example case: a70a230800ffff000007fb
        Case 1: 0 valid, 50 invalid     example case: 70aa308230f0ff0f007060
        Case 2: 0 valid, 50 invalid     example case: 00a70a230805ffff000007
        Case 3: 0 valid, 50 invalid     example case: 0070aa308270f0ff0f0070
        Case 4: 0 valid, 50 invalid     example case: 0000a70a230809ffff0000
        Case 5: 0 valid, 50 invalid     example case: 000070aa3082b0f0ff0f00
        Case 6: 37 valid, 13 invalid    example case: 000000a70a23080efff00
        Case 7: 0 valid, 50 invalid     example case: 00000070aa308200f1ff0f
        Case 8: 41 valid, 9 invalid     example case: 00000000a70a230813ffff
        Case 9: 0 valid, 50 invalid     example case: 0000000070aa308250f1ff
```

## TI CC2531

```
Test: preamble_length_cc2531.json (using Dot15d4PreambleLengthGenerator)
        Case 0: 0 valid, 50 invalid     example case: a70a230800ffff000007fba6
        Case 1: 0 valid, 50 invalid     example case: 70aa308220f0ff0f0070d0eafa
        Case 2: 13 valid, 37 invalid    example case: 00a70a230804ffff00000757b6
        Case 3: 0 valid, 50 invalid     example case: 0070aa308260f0ff0f007010e0fb
        Case 4: 48 valid, 2 invalid     example case: 0000a70a230808ffff000007a387
        Case 5: 0 valid, 50 invalid     example case: 000070aa3082a0f0ff0f007050fff8
        Case 6: 50 valid, 0 invalid     example case: 000000a70a23080cffff0000070f97
        Case 7: 0 valid, 50 invalid     example case: 00000070aa3082e0f0ff0f007090f5f9
        Case 8: 49 valid, 1 invalid     example case: 00000000a70a230810ffff0000074be4
        Case 9: 0 valid, 50 invalid     example case: 0000000070aa308220f1ff0f0070d0c1fe
```

3 transceivers
2 manufacturers
1 protocol

3 behaviors!

# Why Care?

# Those results can allow for WIDS evasion and selective targeting.

# Developing RF Interfaces

# RF Interfaces



Not all radios can generate arbitrary preambles, SFDs, modulations, packet formats, etc.

PHY manipulation requires:

- Software Defined Radio
- Transceiver chipset with lots of configurations

# Software Defined Radio



Prior example used Software Defined Radio:

- GNU Radio and a USRP
- *gr-ieee802-15-4* is flexible because it's well designed

SDR has some drawbacks:

- GNU Radio is complicated and hard to develop for
- SDRs are expensive
- High latency for host-based DSP
- Power hungry: hard to embed

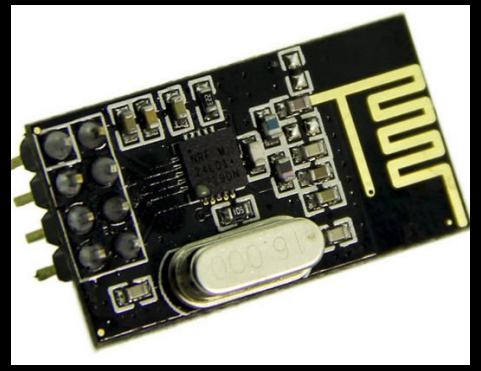
# Configurable Transceivers

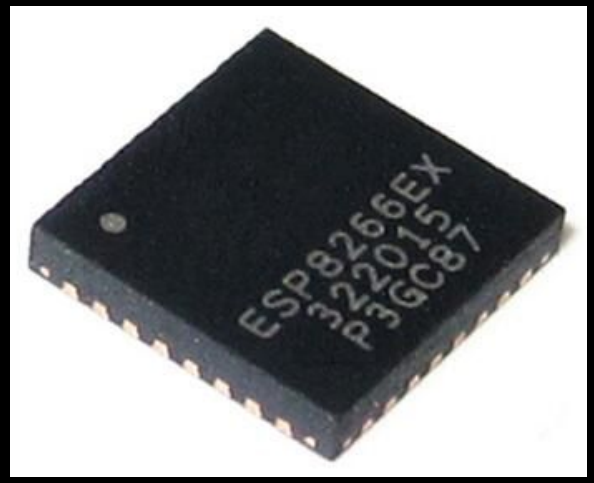


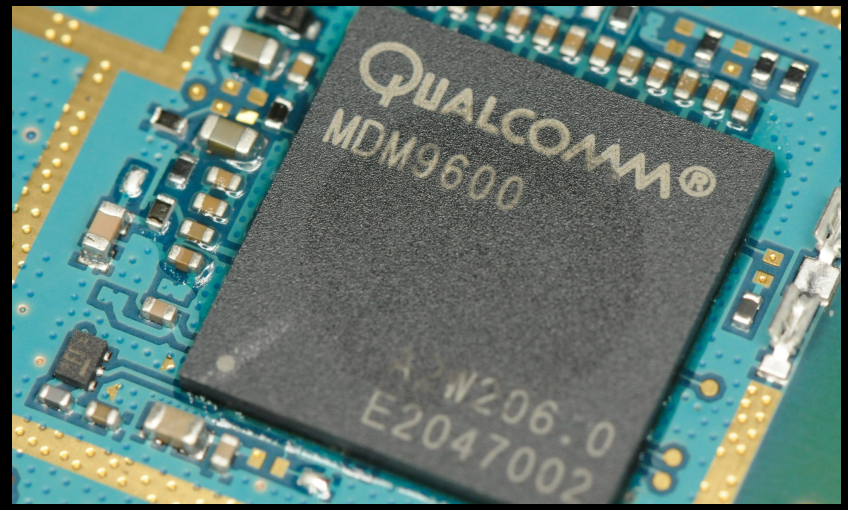
Discrete radio chipsets are purpose built:

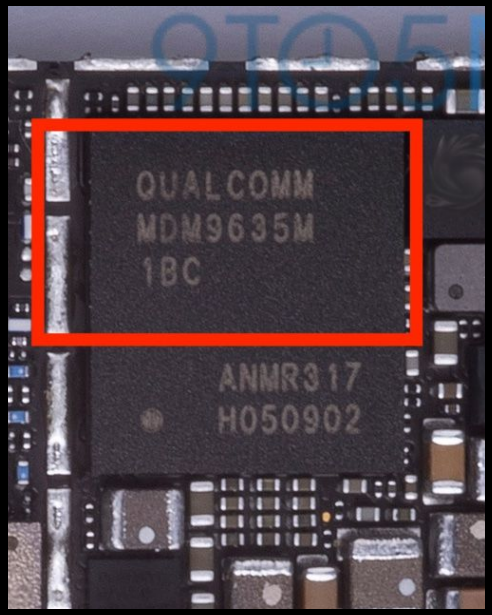
- Generally speak 1 protocol really well
- Band-limited
- Low power
- Some kind of API

Examples include:





# Flexible Transceivers



Certain discrete transceivers can be flexible, like SDR!

Some radios expose PHY configuration registers:

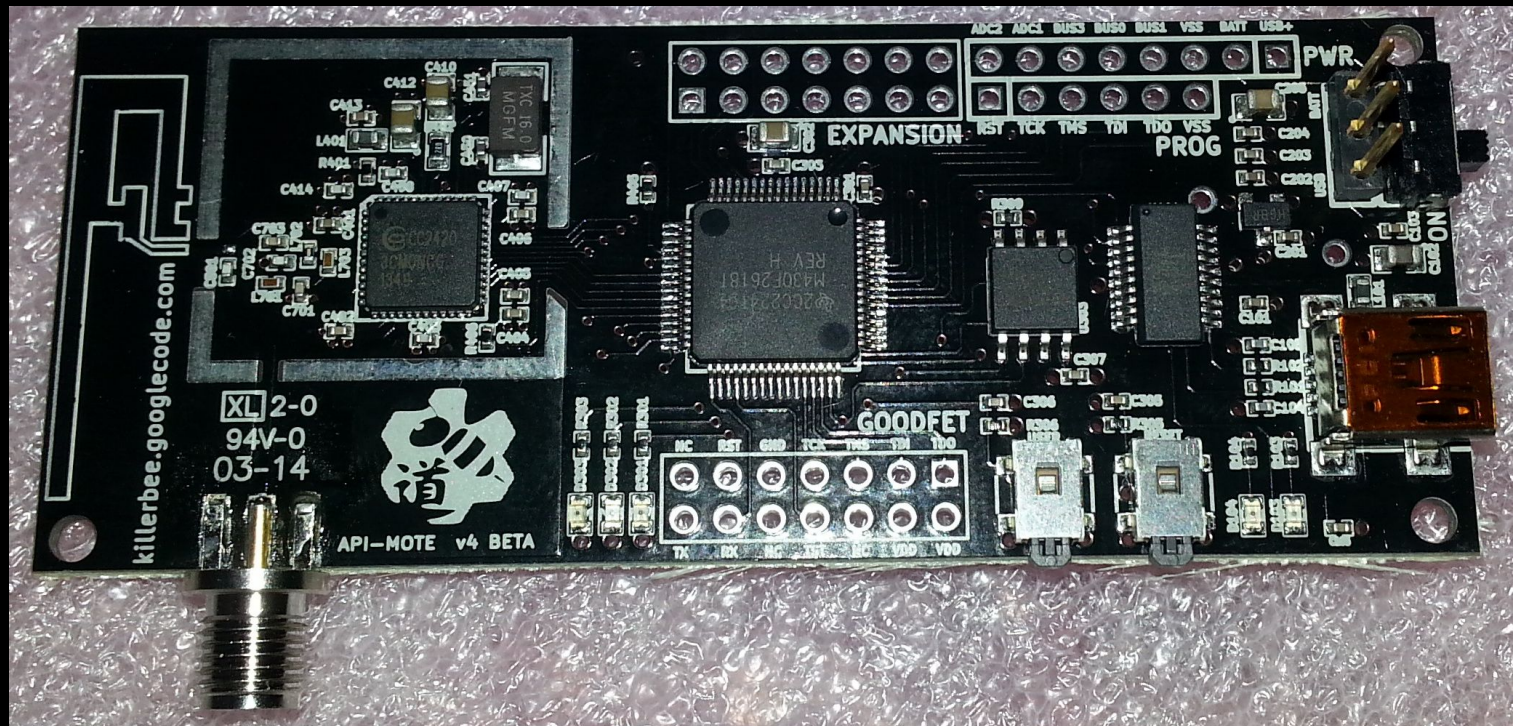
- Preamble length
- SFD magic number
- Header symbol error tolerance
- etc.

# ApiMote (1/2)



ApiMote, designed by Javier & Ryan, exposed PHY registers in TI CC2420:

- Preamble length
- SFD value
- Digital FSM state status pins for low latency injection




Pre-assembled/flashed are available via team@riverloopsecurity.com

# ApiMote (2/2)



However, the ApiMote needs an update:

- CC2420 is EOL
- Expensive BOM
- USB issues

CC2531 and others don’t have the same degree of PHY configuration, so started looking at other chipsets

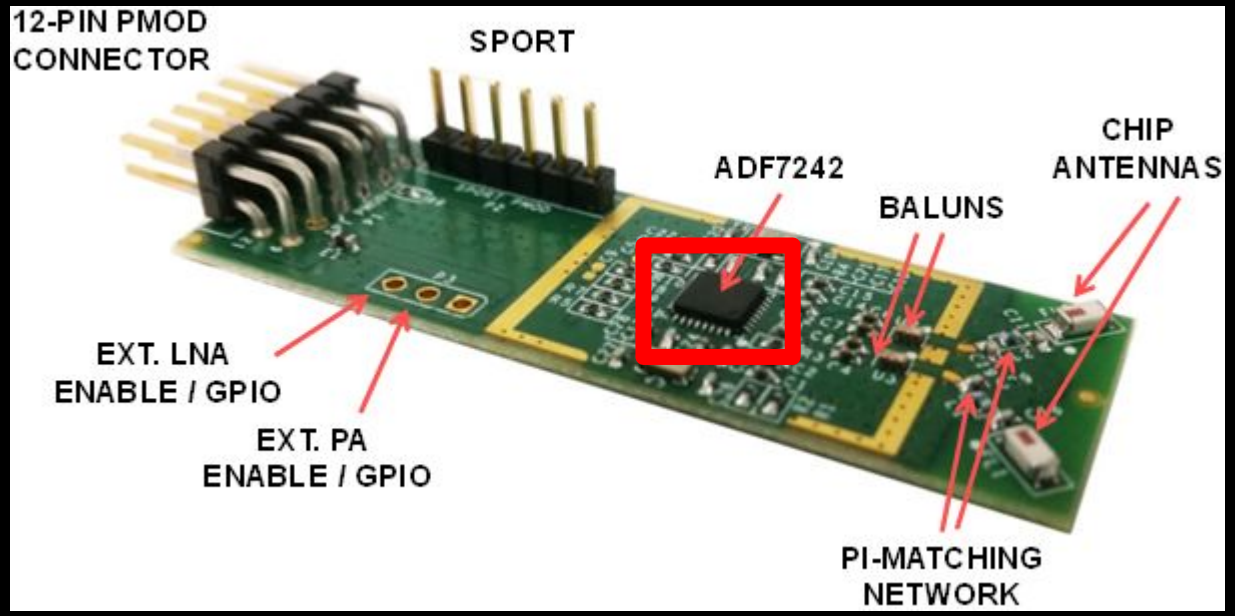
# Enter ADF7242



Most interesting option: Analog Devices ADF7242

2.4 GHz 802.15.4 radio with lots of features:

- Several modulations
- Lots of configurability
- SPORT mode



# SPORT Mode?



Streams demodulated symbols over serial, up to 2Msps

- Bypasses decoding and PHY header / packet framing
- We can implement these parts in software

Full control of PHY for most 2.4 GHz protocols!

ApiMote 2.0 >> 802.15.4

# Introducing Orthrus

# Orthrus (1/2)



Spiritual successor to the ApiMote

Named for 2-headed dog from Greek mythology

- Why? Because Orthrus has two heads!

# Orthrus (2/2)



NXP LPC4370 ARM MCU

- Host communication via USB
- Controlling radios
- RF state machine implementation and control

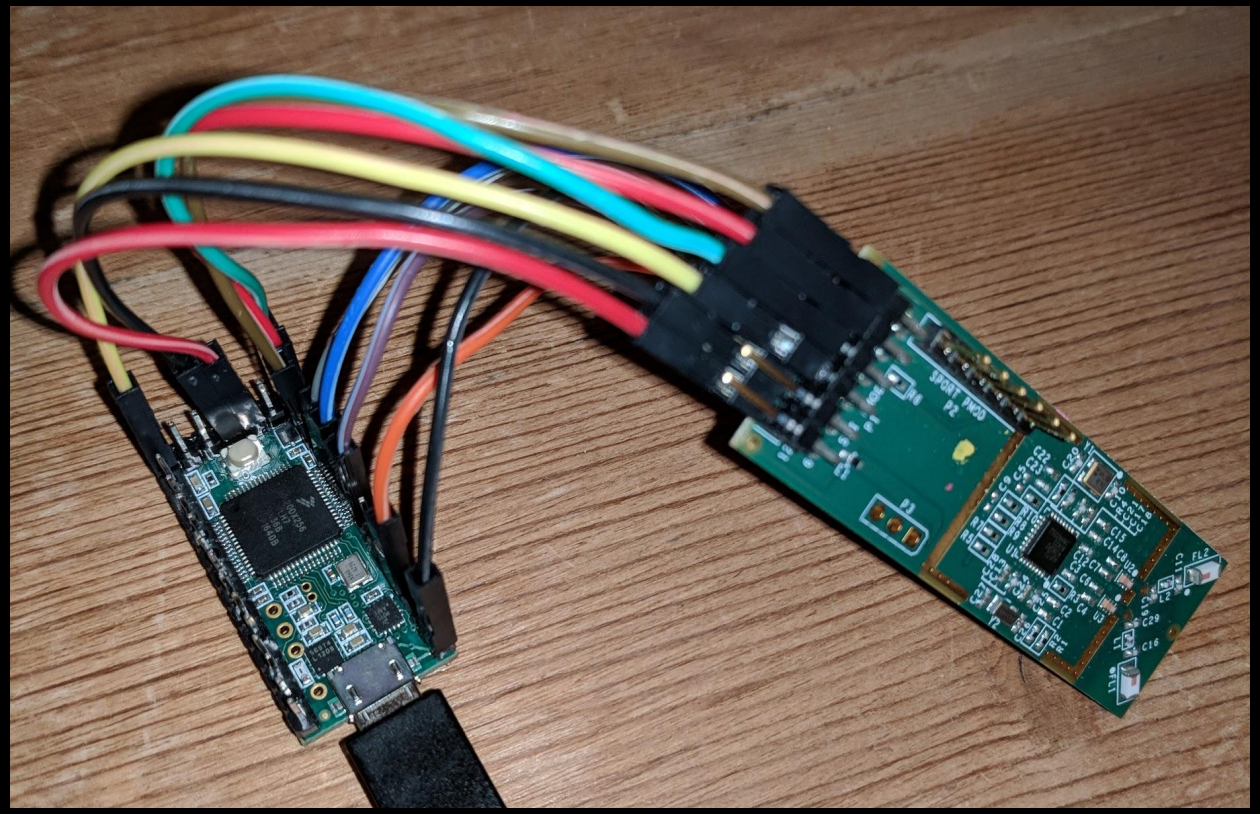
2x ADF7242 radios

- ADF7242 has a slow re-tune time
  - 2 allows for pre-emptive re-tune!
- 1 can listen while the other sits ready to transmit
  - High-speed responsive jamming

# Initial Prototype



ADF7242 dev board wired to Teensy:



Custom PCB is in progress

# Orthrus RF Design Flow



Implement event loop in firmware

Blue-Green frontends for fast retuning / channel hopping

TODO: State machine abstraction language?

- e.g. XASM / ASML / SCXML
- Implement PHYs via config definitions rather than code

# Packet-in-Packet Detection



## 802.15.4 Frame Structure:

| PHY Header | | | **PHY Data Unit** | |
|---|---|---|---|---|
| Preamble | SFD | Length | **Data** | CRC |

## Packet-in-Packet frame structure:

| PHY Header | | | **PHY Data Unit** | | | | |
|---|---|---|---|---|---|---|---|
| Preamble | SFD | Length | **Preamble** | **SFD** | **Length** | **Data** | CRC |

Traditional radio chipset would see the outer packet only

Software-defined decoder in Orthrus can be written to see both

# Interested? Get Involved!



Contribute something to TumbleRF:

- Generator for some cool new fuzzing idea you have
- Harness to check the state of a device you care about testing
- Interface to transmit with your favorite radio

Contribute to Orthrus:

- Firmware development
- State machine abstraction definitions

https://github.com/riverloopsec/tumblerf

# Thank You!

**DEF CON 26 Crew**

**River Loop Security**
**Cruise Automation**
**Ionic Security**

River Loop Security

https://github.com/riverloopsec/tumblerf

# Questions?

**@embeddedsec**
**@rmspeers**

**[matt | ryan]@riverloopsecurity.com**

River Loop Security