Assessing and Exploiting RF Communications

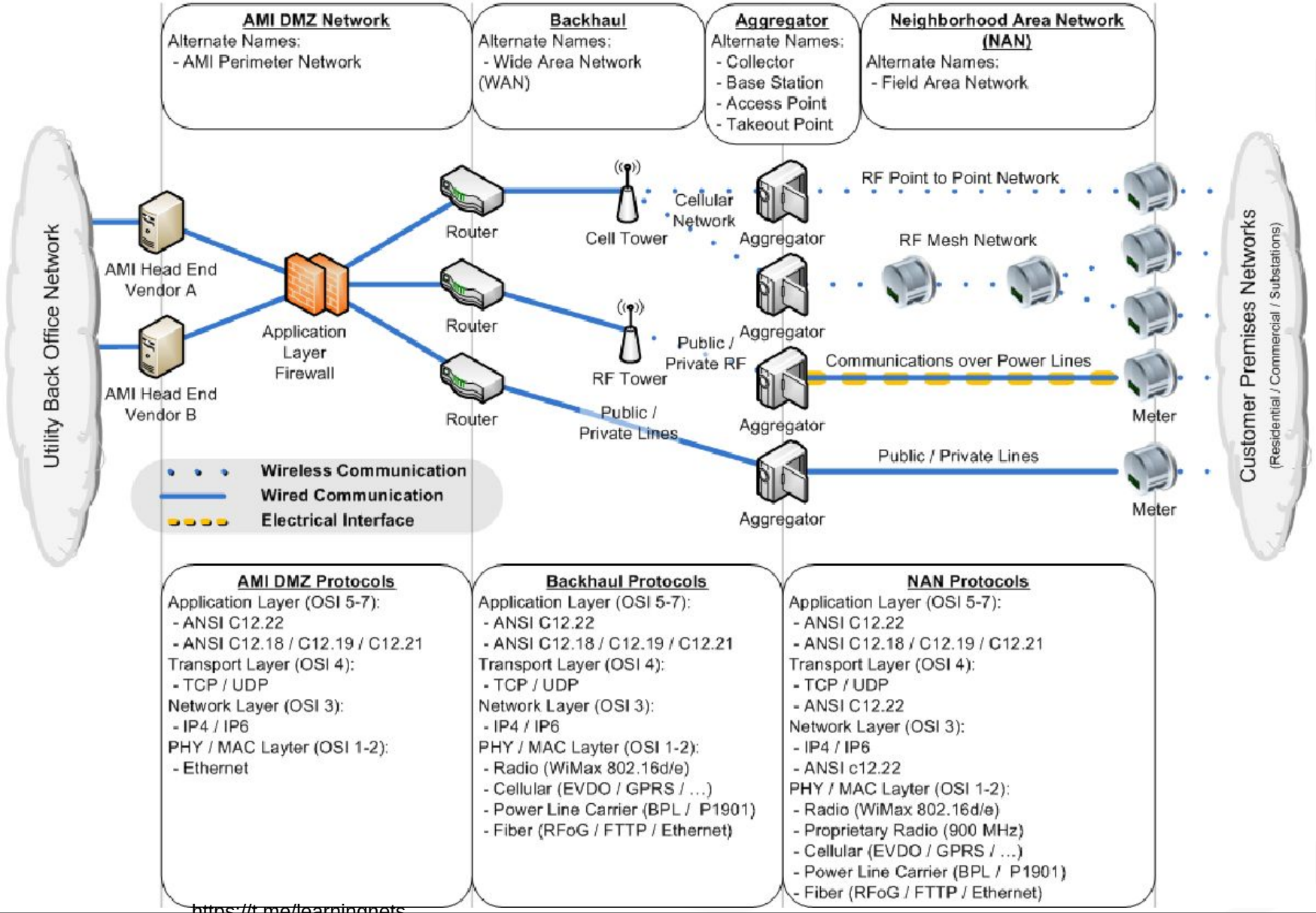
Version 38

801-784-2052 // justin@controlthings.io

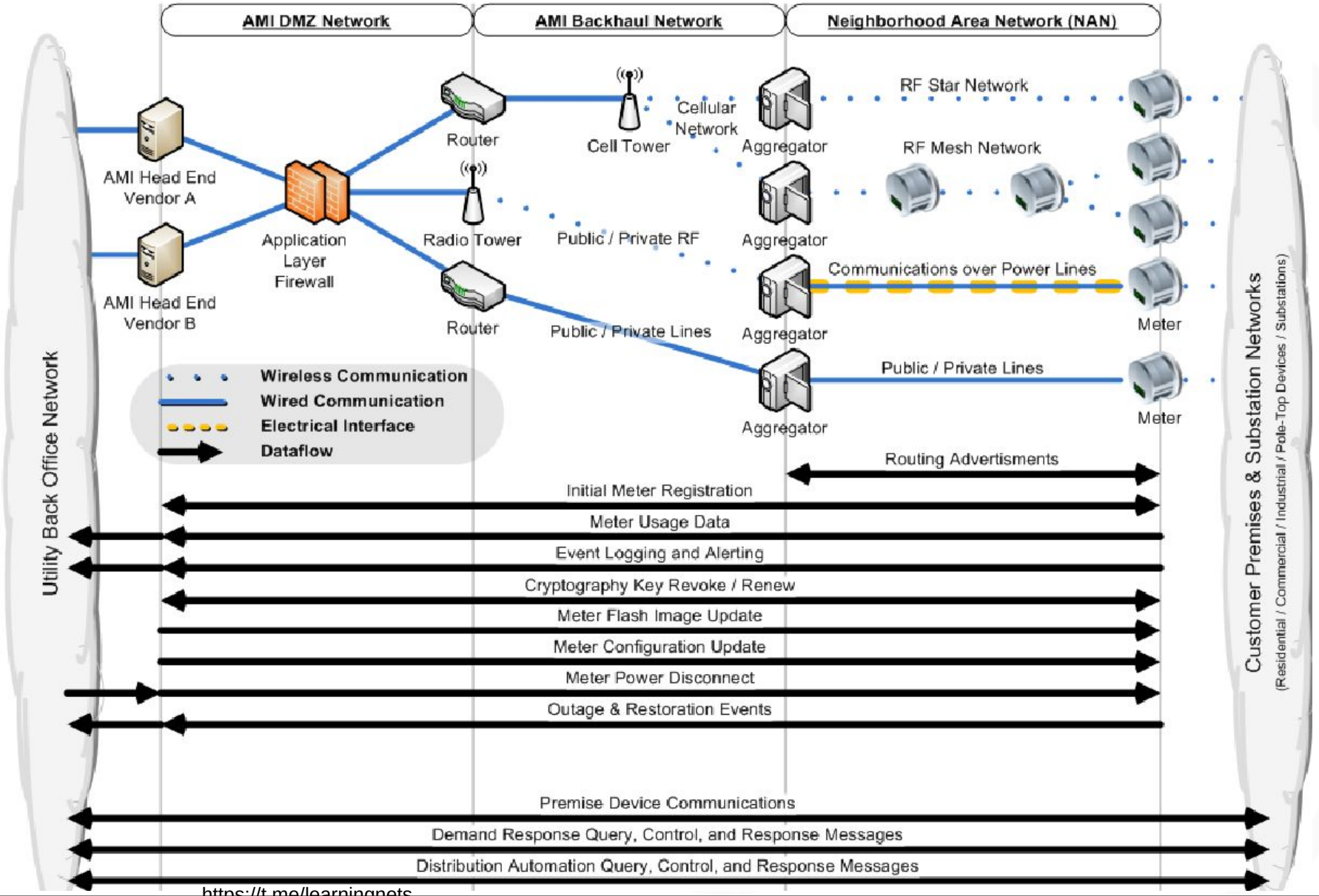
Where Wireless is Used in ICS

- Wi-Fi for OT staff laptops
- Bluetooth for calibration/maintenance/HMI
- Time synchronization (GPS, GLONASS, BeiDou2, Galileo)
- Satellite Uplink (VSAT, BGAN) for outstations
- Cellular (CDMA, GSM, LTE) for field devices
- Licensed Radio / Microwave for outstations & devices
- Mesh RF Wireless (WirelessHART, ISA100.11a, WiFi Mesh, ZigBee) for field devices, sensors, and meters

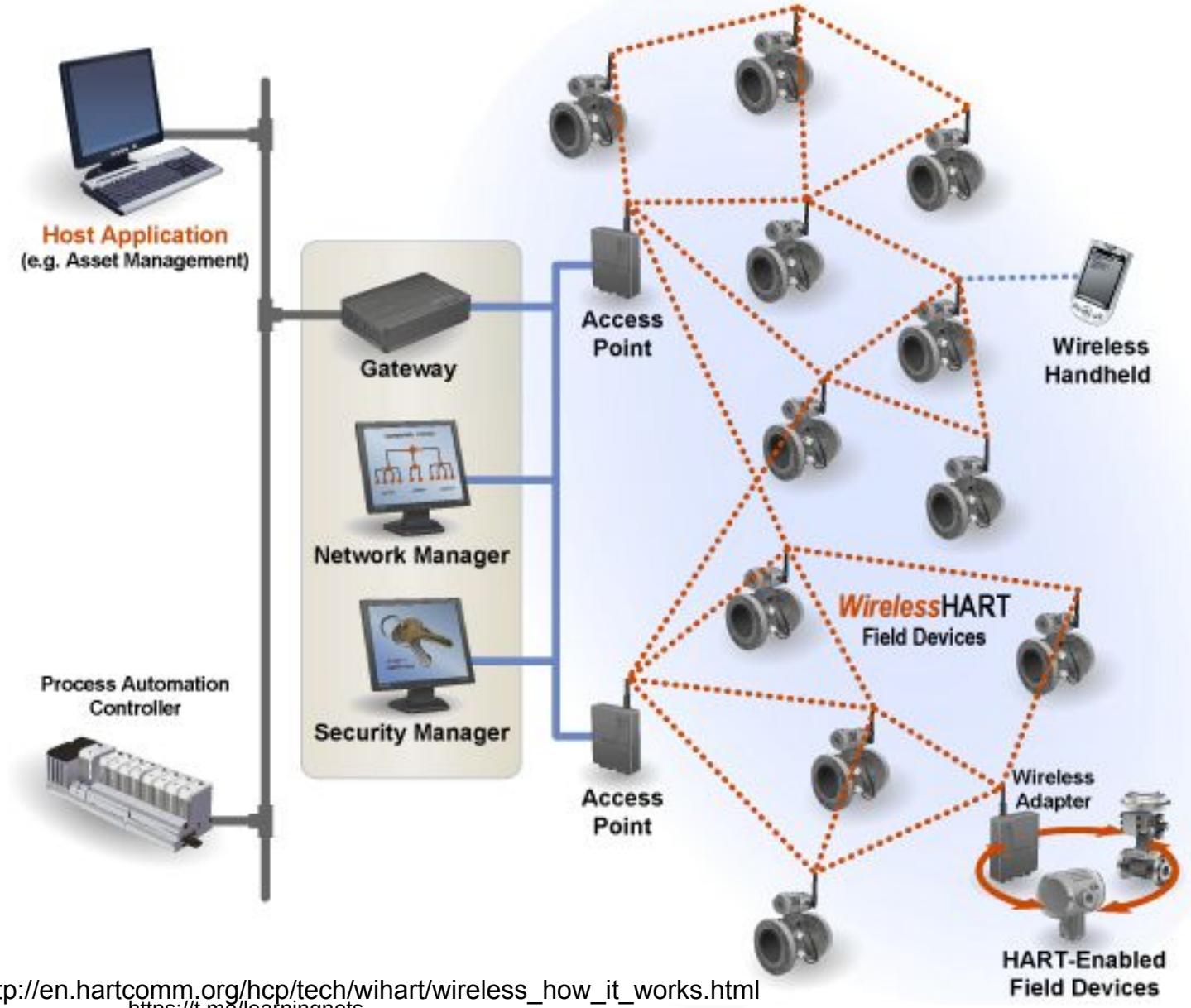
Example: Electric Grid's Smart Meter



Smart Meter Data Flows

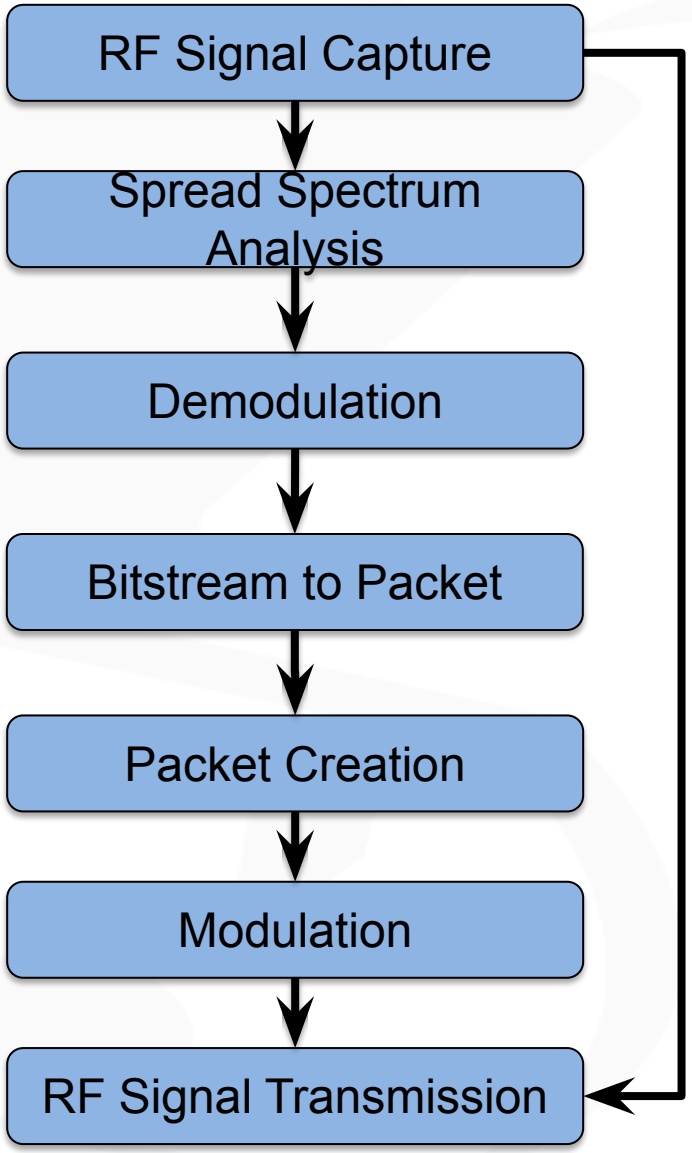


Example: WirelessHART



Source: http://en.hartcomm.org/hcp/tech/wihart/wireless_how_it_works.html

RF Communications Testing Tasks



Task 1: RF Signal Capture

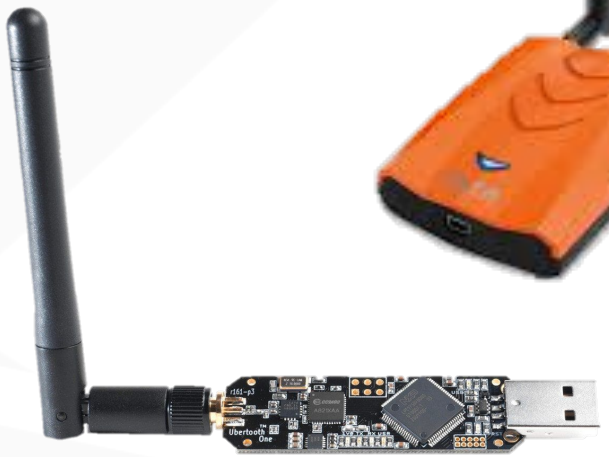
Level of Effort: Low to Medium

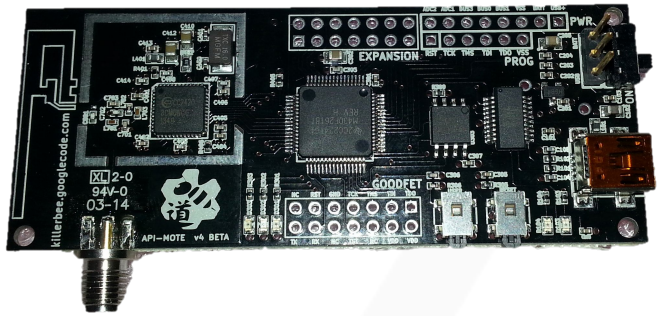
Task Description: Use a tool, such as a software defined radio (SDR) to find and capture the RF communications of the target field device.

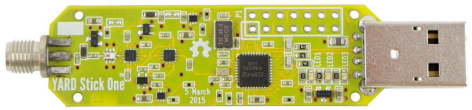
Task Goal: Obtain data for following tasks.

Specific RF Protocol Solutions

- Wi-Fi
  - Hardware: WiFi Pinapples, ALFA cards, and many others
  - Software: Kismet, Aircrack-NG
- Bluetooth
  - Hardware: Ubertooth
  - Software: Ubertooth
- ZigBee
  - Hardware: ApiMote, MoteIV Tmote Sky or TelosB, RZ RAVEN
  - Software: KillerBee
- Z-Wave
  - Hardware: Yardstick
  - Software: KillerZee







Generic RF Protocol Solutions

Software Defined Radio (SDR)

- Hardware Options
  - RTL-SDR: $20 (capture only)
  - HackRF: $300
  - BladeRF: $420 or $650
  - Ettus Research: $675-$4800
- Pros
  - Flexible because it acquires the signal via hardware and does all processing in software
  - GNURadio and GNURadio Companion are the tools of choice
- Cons
  - Easy to get lost in the complexity
  - Can be very computer resource intensive, making VMs difficult

RFCat + Hardware Radio

- Hardware Options
  - Yardstick: $100
  - IM-Me dongle: $35
  - CC1111EMK dongle: $49
- Pros
  - Faster, RFCat configures hardware instead of doing it in software
  - Simpler to use than GNURadio
  - Frequency hopping recovery (alpha)
- Cons
  - Frequencies: 281-361 MHz, 378-481 MHz, and 749-962 MHz
  - Modulations: ASK, OOK, GFSK, 2-FSK, 4-FSK, MSK
  - Max Data Rate: 500 kBaud

SDR Hardware Options

	RTL-SDR (Elonics E4000)	HackRF	bladeRF 2.0 xA4 / xA9	LimeSDR Mini / Original	Ettus USRP B200m / B210	Ettus USRP X310	Ettus USRP N310
Frequency Spectrum	52 MHz – 2.2 GHz	30 MHz - 6 GHz	47 MHz - 6 GHz	10 MHz - 3.5 GHz	70MHz - 6GHz	DC – 6GHz	10MHz – 6GHz
Duplex	Receive Only	Half	Full / 2x2 MIMO	Full / 2x2 MIMO	Full / 2x2 MIMO	2x2 MIMO or 4 x Rx	4 x 4 MIMO
Bandwidth	2.5 MHz Rx	20 MHz	56 MHz Filtered IBW	30.72 / 61.44 MHz	56 H / 30 F MHz	160 MHz 320 MHz Rx	153.6 MHz x 4 IBW
Sample Size	8bit	8 bit	12 bit	12 bit	12 bit	16 bit	16b Rx / 14b Tx
Sample Rate	2.5 MS/s	20 MS/s	61.44 MS/s	30.72 / 61.44 MS/s	45 MS/s Full 61 MS/s Half	2 x 200 MS/s 4 x 80 MS/s	153.6 MS/s x 4
Interface	USB 2	USB 2	USB 3	USB 3	USB 3	2 x 10GB Ethernet PCIe Express Dual 1GB Ethernet	2 x 10GB Ethernet PCIe Express Dual 1GB Ethernet
FPGA	None	Programmable CPLD	49 / 301 kLE (Cyclone V)	16k / 40k	75k / 150k	460k	460k
Cost	$20	$300	$480 / $720	$159 / $299	$733 / $1216	$5422 ++	$10,000

Other SDR devices are coming to market every few months, helping to progress this research

Finding the Right Frequencies

- Initial Architecture Review and Interviews
- Product Documentation
- RF Regulatory Registration Databases
  - FCC ID Search (U.S.)
    http://transition.fcc.gov/oet/ea/fccid/
- Patent Filings
  - Patent Number Search (U.S.)
    http://appft1.uspto.gov/netahtml/PTO/srchnum.html
  - Patent Keyword Search (U.S.)
    http://appft1.uspto.gov/netahtml/PTO/search-bool.html
  - Patent Keyword Search (Europe)
    http://worldwide.espacenet.com/advancedSearch?DB=EPODOC&submitted=false&locale=en_EP&AB=&ST=advanced&compact=false
- Patiently digging through available ISM bands

Industrial/Science/Medical (ISM) Bands

Frequency range		Bandwidth	Center Frequency	Notes
6.765 MHz	6.795 MHz	30 kHz	6.780 MHz	Subject to local acceptance
13.553 MHz	13.567 MHz	14 kHz	13.560 MHz	RFID
26.957 MHz	27.283 MHz	326 kHz	27.120 MHz	
40.660 MHz	40.700 MHz	40 kHz	40.680 MHz	
433.050 MHz	434.790 MHz	1.84 MHz	433.920 MHz	Region 1: Europe, Africa, Middle East, Russia, Mongolia, but most accepted world wide
868.000 MHz	870.000 MHz	2 MHz	869.000 MHz	Not ISM but SRD band for Europe and India
902.000 MHz	928.000 MHz	26 MHz	915.000 MHz	Region 2: Americas, Greenland, and Pacific Islands
2.400 GHz	2.500 GHz	100 MHz	2.450 GHz	(used by Wi-Fi, Bluetooth, and ZigBee)
5.725 GHz	5.875 GHz	150 MHz	5.800 GHz	(used by Wi-Fi)
24.000 GHz	24.250 GHz	250 MHz	24.125 GHz	Automotive Radar
61.000 GHz	61.500 GHz	500 MHz	61.250 GHz	Subject to local acceptance
122.000 GHz	123.000 GHz	1 GHz	122.500 GHz	Subject to local acceptance
244.000 GHz	246.000 GHz	2 GHz	245.000 GHz	Subject to local acceptance

International Telecommunication Union:

- Part of the United Nations (UN)
- Coordinates world radio spectrum
- Each country can accept/reject
- ISM band set aside for commercial products

General rules of radio frequencies:

- At the same power, lower frequencies travel further than higher frequencies
- Lower frequencies have lower bitrates, so less data bandwidth
- Lower frequencies have larger antennas
- Compare these three points to what you already know about Wifi 2.4 GHz and Wifi 5 GHz.

Common ICS Frequencies

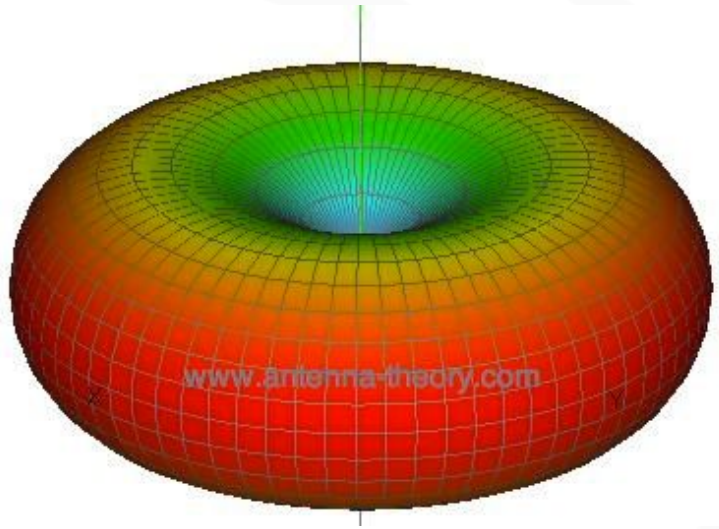
- **Wi-Fi 802.11b/g/n**: 2.4 - 2.5 GHz
  - **802.11y**: 3.655 - 3.695 GHz
  - **802.11a/h/j/n/ac**: 5.15 - 5.725 GHz
  - **802.11p**: 5.850 - 5.925 GHz
  - **802.11ad**: 60GHz
  - **802.11ah**: 900MHz
- **Bluetooth, ZigBee, WirelessHART, ISA100.11a**: 2.4 - 2.5 GHz
- **Microwave**: Anything above 1 - 2.4 GHz
- **Cellular**: 710 - 960 MHz, 1.71 - 1.99 GHz, 2.11 - 2.17 GHz, 2.5 - 2.7 GHz
- **GPS**: 1.57542 GHz (L1 signal) and 1.2276 GHz (L2 signal)
- **GLONASS**: 1.602 GHz (L1 signal) and 1.246 GHz (L2 signal)
- **VSAT**: C-Band: downlink 3.7 - 4.2 GHz, uplink 5.9 - 6.4 GHz
  - Ku-Band (Americas): downlink 11.7 - 12.2 GHz, uplink 14.0 - 14.5 GHz
    (Europe/Africa): downlink 11.45-11.7 and 12.5-12.75 GHz, uplink 14.0-14.5 GHz
  - Ka-Band: 18 to 40 GHz
- **BGAN**: Transmit: 1626.5 MHz - 1660.5 MHz, Receive: 1525 MHz - 1559 MHz

Wave and Antennae Length

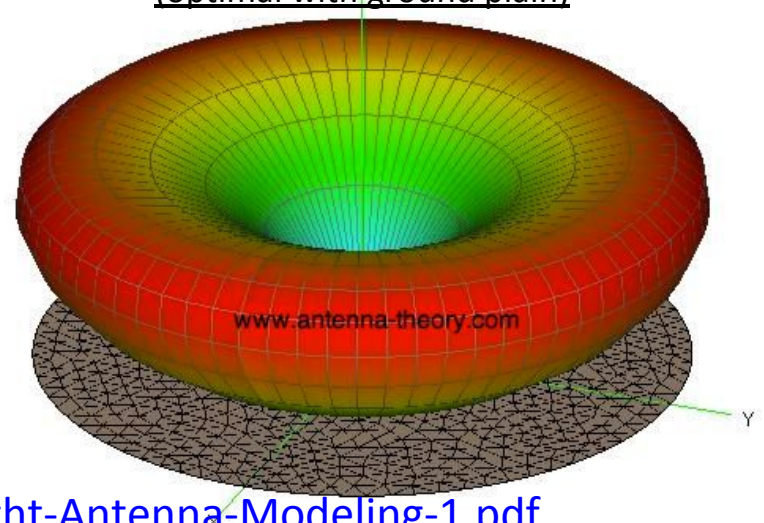
- **NEVER** power a radio without an antenna!!!
- Proper length most critical for transmitting not as critical for receiving
- Formulas for Wave Length:
  - Length in Meters = `300 / Frequency MHz`
  - Length in Feet = `984 / Frequency MHz`
- Divide by 2 for half length dipole antenna
  - Length in Meters = `150 / Frequency MHz`
  - Length in Feet = `492 / Frequency MHz`
- Divide by 4 for quarter length monopole antenna
  - Uses ground plane to reflect other half of wave to antenna
  - Length in Meters = `75 / Frequency MHz`
  - Length in Feet = `246 / Frequency MHz`
  - Rough length in Inches... = `3000 / Frequency MHz`

- Great resources for more antenna-theory:
  - http://www.antenna-theory.com
  - https://www.n1fd.org/wp-content/uploads/2018/03/2018-March-Tech-Night-Antenna-Modeling-1.pdf

Dipole Radiation Pattern



Monopole Radiation Pattern
(optimal with ground plain)



Listening to FM Radio with GQRX

- Adjust your antenna length for FM radio around 100 MHz (3000 / 100 inches...)
- Attach your SDR device to your USB port and use dmesg or lsusb to verify it is connected to ControlThings Platform
- From the activities menu, start gqrx inside ControlThings Platform
- In Gqrx, configure I/O Device to use Realtek RTL2838UHIDIR, no other settings there needed
- Tune your frequency to the 87.5 – 107.9 MHz range
- Start DSP (Digital Signal Processing)
- Find a strong radio signal and tune in to it
  - Change your Receiver Options mode to WFM (mono)
  - Verify your volume settings in the VM and on your host
  - Verify your squelch (between -25dB and -50dB) and gain (around 10)dB in GQRX Receiver tab
- Look for other FM broadcasts appropriate the country/region you are in
- Explore the interface and visualization settings

Task 2: Spread Spectrum Analysis

Level of Effort: Medium to High

Task Description: Determine if the RF communication uses one of the spread spectrum techniques (FHSS, DSSS, THSS, CSS). Analyze the RF capture to attempt to determine the spread spectrum algorithm used. Alternately, bus sniffing captures near the RF chip may also be used to determine this information.

Task Goal: Obtain data for following tasks.

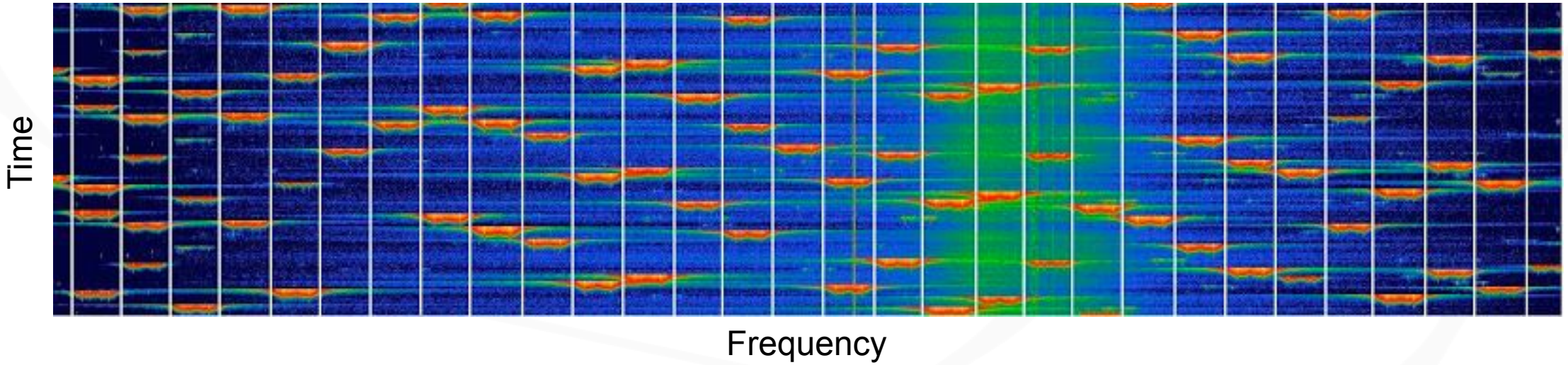
Recovering the FHSS Algorithm

- Initial architecture review and interviews
- RF regulatory registration databases
  - FCC ID Search (U.S.)
    http://transition.fcc.gov/oet/ea/fccid/
- Patent filings
  - Patent Number Search (U.S.)
    http://appft1.uspto.gov/netahtml/PTO/srchnum.html
  - Patent Keyword Search (U.S.)
    http://appft1.uspto.gov/netahtml/PTO/search-bool.html
  - Patent Keyword Search (Europe)
    http://worldwide.espacenet.com/advancedSearch?DB=EPODOC&submitted=false&locale=en_EP&AB=&ST=advanced&compact=false
- Recovering from the firmware (MCU/RF) or software
- Capturing using Bus Sniffing

http://www.digitalbond.com/blog/2013/02/11/s4x13-video-atlas-on-rf-comms-security-and-insecurity/

Or Don't Recover the Algorithm...

- With FHSS you can simply try to capture all the channels at the same time and reconstruct the data stream based on time
- But that does mean extracting each channel (usually 40-80 of them) one at a time then manually reassembling them...
- Easier to do if you can isolate one transmitter



Task 3: Demodulation

Level of Effort: Medium to High

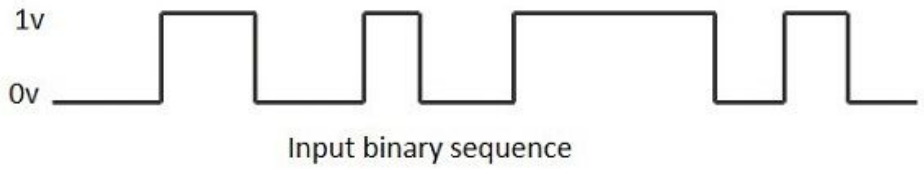
Task Description: Analyze the RF capture to determine modulation technique used.

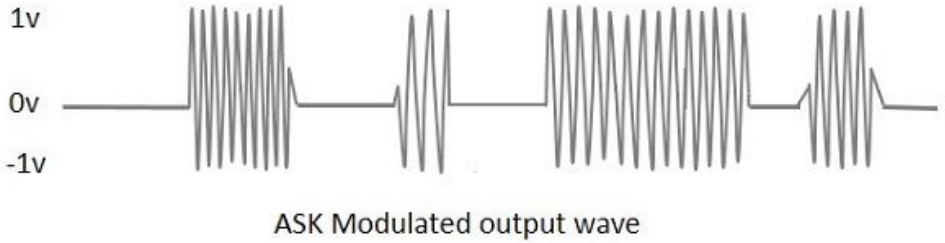
Task Goal: Obtain data for following tasks.

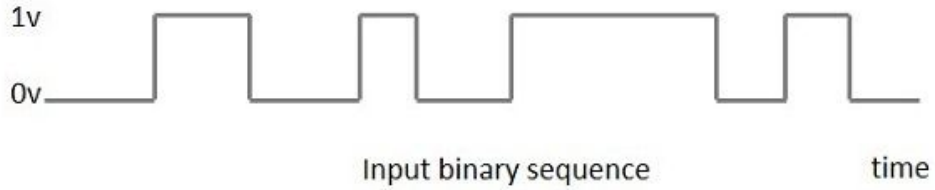
Analog Modulation Types

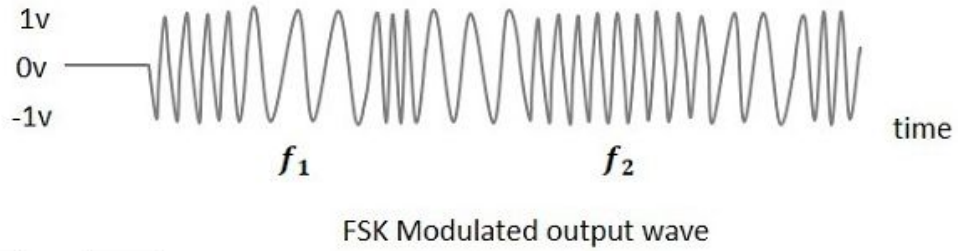
- Amplitude modulation (AM)
  - Double-sideband modulation (DSB)
    - Double-sideband modulation with carrier (DSB-WC)
    - Double-sideband suppressed-carrier transmission (DSB-SC)
    - Double-sideband reduced carrier transmission (DSB-RC)
  - Single-sideband modulation (SSB, or SSB-AM)
    - SSB with carrier (SSB-WC)
    - SSB suppressed carrier modulation (SSB-SC)
  - Vestigial sideband modulation (VSB, or VSB-AM)
  - Quadrature amplitude modulation (QAM)
- Angle modulation
  - Frequency modulation (FM)
  - Phase modulation (PM)

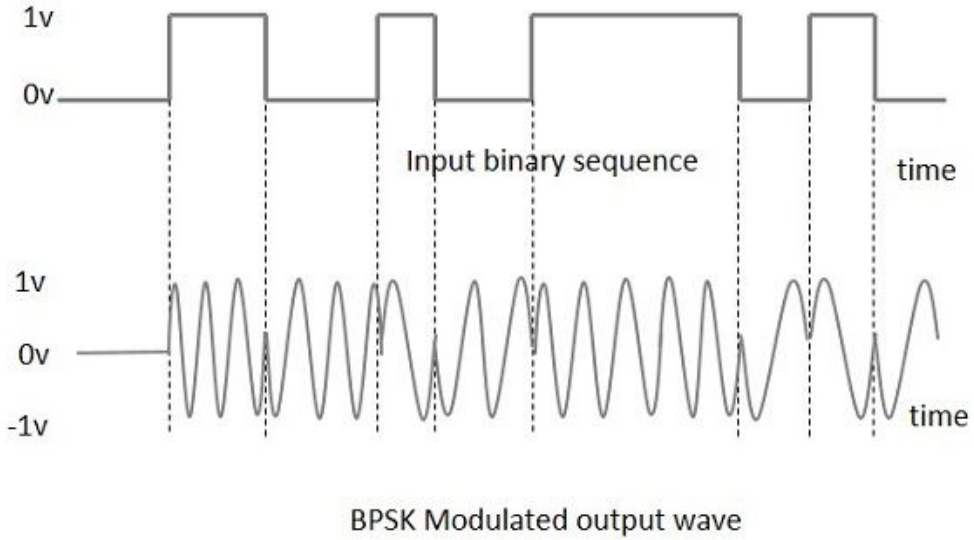
Digital Modulation Families











Source: https://www.tutorialspoint.com/digital_communication

Digital Modulation Types

- Phase-shift keying (PSK):
  - Binary PSK (BPSK); Quadrature PSK (QPSK); 8PSK; 16PSK; Differential PSK (DPSK); Differential QPSK (DQPSK); Offset QPSK (OQPSK); π/4–QPSK
- Frequency-shift keying (FSK):
  - Audio frequency-shift keying (AFSK); Multi-frequency shift keying (M-ary FSK or MFSK); Dual-tone multi-frequency (DTMF)
- Amplitude-shift keying (ASK)
  - On-off keying (OOK); M-ary vestigial sideband modulation, for example 8VSB
- Quadrature amplitude modulation (QAM) - a combination of PSK and ASK:
  - Polar modulation like QAM a combination of PSK and ASK.
- Continuous phase modulation (CPM) methods:
  - Minimum-shift keying (MSK); Gaussian minimum-shift keying (GMSK); Continuous-phase frequency-shift keying (CPFSK)
- Orthogonal frequency-division multiplexing (OFDM) modulation:
  - discrete multitone (DMT) - including adaptive modulation and bit-loading.
- Wavelet modulation
- Trellis coded modulation (TCM), also known as trellis modulation

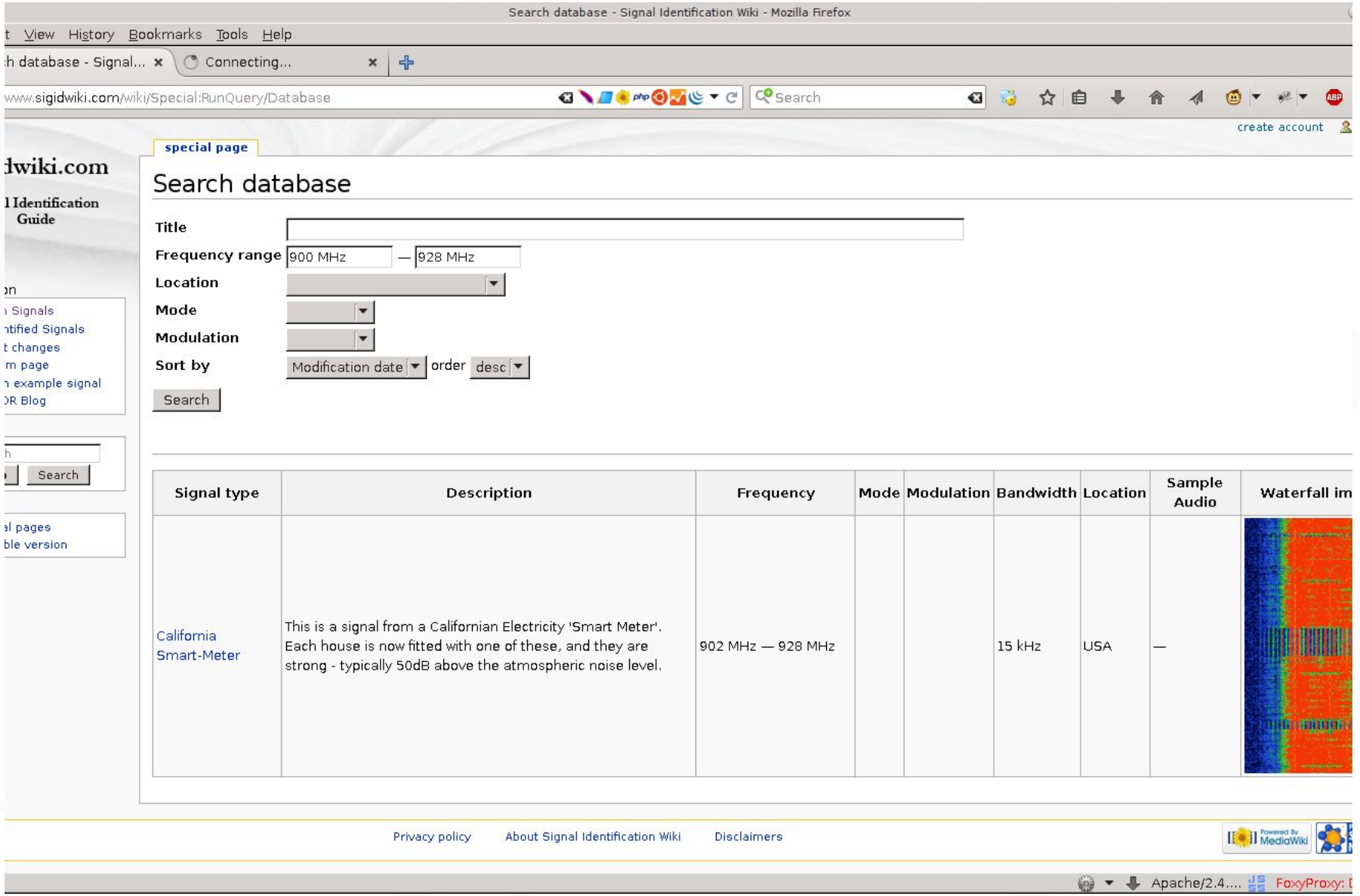
Recovering the Modulation

- Initial Architecture Review and Interviews
- Product Documentation
- RF Regulatory Registration Databases
  - FCC ID Search (U.S.)
    http://transition.fcc.gov/oet/ea/fccid/
- Patent Filings
  - Patent Number Search (U.S.)
    http://appft1.uspto.gov/netahtml/PTO/srchnum.html
  - Patent Keyword Search (U.S.)
    http://appft1.uspto.gov/netahtml/PTO/search-bool.html
  - Patent Keyword Search (Europe)
    http://worldwide.espacenet.com/advancedSearch?DB=EPODOC&submitted=false&locale=en_EP&AB=&ST=advanced&compact=false
- Recovering from the Firmware (MCU/RF) or Software

Signal Identification Wiki

- Sometimes visual examples can help you determine modulation type
- Signal Identification Wiki

http://www.sigidwiki.com



Example: TI Chronos Watch

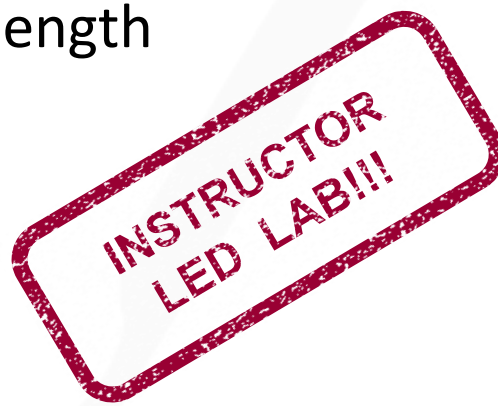
```
// ISM_US configuration
//
// Chipcon
// Product = CC1101
// Chip version = A   (VERSION = 0x04)
// Crystal accuracy = 10 ppm
// X-tal frequency = 26 MHz
// RF output power = 0 dBm
// RX filterbandwidth = 232.142857 kHz
// Deviation = 32 kHz
// Datarate = 76.766968 kBaud
// Modulation = (1) GFSK
// Manchester enable = (0) Manchester disabled
// RF Frequency = 905.998993 MHz
// Channel spacing = 199.951172 kHz
// Channel number = 0
// Optimization = -
// Sync mode = (3) 30/32 sync word bits detected
// Format of RX/TX data = (0) Normal mode, use F
IFOs for RX and TX
// CRC operation = (1) CRC calculation in TX and
 CRC check in RX enabled
// Forward Error Correction = (0) FEC disabled
// Length configuration = (1) Variable length pa
ckets, packet length configured by the first rec
eived byte after sync word.
// Packetlength = 255
// Preamble count = (2)  4 bytes
```

Pre-captured ASK-OOK Demodulation

- On your ControlThings VM, start the urh application (Universal Radio Hacker)
- From urh menu, open the following file:

~/RadioFrequencies/Basic-Modulation/RTL-SDR-433MHz-1MSps-1MHz-OOK-cropped-three.complex

- Adjust the Noise value so the highlighted red section is below signals but above noise
- Change your Modulation to ASK
- Change your Signal View to Demodulated to verify only valid signals demodulate
- Zoom into your signal to see individual waves
- Adjust the Center value by dragging the 1/0 boundary so 1s and 0s widths are even
- Find the smallest spike representing a bit and highlight it to measure its length
- Manually adjust the Bit Length to reflect the value you measured
- Change Show Signal as to ASCII



Pre-captured 2FSK Demodulation

- On your ControlThings VM, start the urh application (Universal Radio Hacker)
- From urh menu, open the following file:

~/RadioFrequencies/Basic-Modulation/RTL-SDR-433MHz-1MSps-1MHz-2FSK-cropped-three.complex

- Adjust the Noise value so the highlighted red section is below signals but above noise
- Change your Modulation to FSK
- Change your Signal View to Demodulated to verify only valid signals demodulate
- Zoom into your signal to see individual waves
- Adjust the Center value by dragging the 1/0 boundary so 1s and 0s widths are even
- Find the smallest spike representing a bit and highlight it to measure its length
- Manually adjust the Bit Length to reflect the value you measured
- Change Show Signal as to ASCII

Using Yardstick with rfcat

1. Attach/adjust your Yardstick antennae to work with 433 MHz if you have changed it
2. Plug in your yardstick to reset it to default settings and use lsusb or dmesg to verify the it is connected to ControlThings Platform. It is an an OpenMoko device.
3. Open a terminal and type `rfcat -r` for an interactive session, you will see messages in insert dongle
4. Insert your Yardstick and you should get a prompt
5. Attach Yardstick to your USB port and to ControlThings Platform VM
6. Check your current config with
   ```
   d.printRadioConfig()
   ```
7. Configure your yardstick with these settings and transmit:
   ```
   d.setFreq(433e6)
   d.setMdmModulation(MOD_ASK_OOK)
   d.RFxmit("Hello World")
   ```
8. If you you have a neighbor, try listening to their transmit with:
   ```
   d.RFlisten()
   ```
9. Pressing any key will stop the listen command

OOK Live Signal Capture

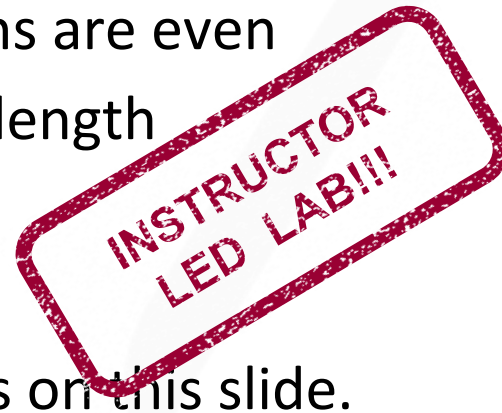
- Using `rfcat -r`, run the following commands to generate a signal:

```
import time
d.setFreq(433000000)
d.setMdmModulation(MOD_ASK_OOK)
while True:
    time.sleep(0.5)
    d.RFxmit("Hello World")
```

- Adjust your antenna to 433MHz (3000/433 in inches) and attach your SDR to your VM
- Start urh, create a new project, and
  - Set the sample rate to 1M and the frequency to 433M
- Under File, start Spectrum Analyzer
  - Choose RTL-SDR as your capture device
  - Verify frequency and sample rate are set correctly
- Once you confirm you can see the traffic in the Spectrum Analyzer window, close it
- Under File, start Record Signal
- Verify the device, sample rate, and frequency settings and record that same signal
  - You should visually see a signal spike
  - Save the signal and close the capture signal window

OOK Live Signal Demodulation

- Crop your recording to just contain your signal by using the shift key to highlight the good signal and right clicking
- Save your new cropped signal
- Adjust the Noise value so the highlighted red section is below signals but above noise
- Change your Modulation to ASK
- Change your Signal View to Demodulated to verify only valid signals demodulate
- Zoom into your signal to see individual waves
- Adjust the Center value by dragging the 1/0 boundary so 1s and 0s widths are even
- Find the smallest spike representing a bit and highlight it to measure its length
- Manually adjust the Bit Length to reflect the value you measured
- Change Show Signal as to ASCII
- If you do not see the message the instructor is sending, review the steps on this slide.
- Still no? Try the pre-recorded OOK capture in ~/RadioFrequencies/Basic-Modulation/



2FSK Live Signal Capture and Demodulation

- Using `rfcat -r`, run the following commands to generate a signal:

```
import time
d.setFreq(433000000)
d.setMdmModulation(MOD_2FSK)
while True:
    time.sleep(0.5)
    d.RFxmit("Hello World")
```

- Zoom into your signal to see individual waves
- Change your Modulation to FSK
- Adjust the center value by dragging the red zone to visually center between the upper and lower spikes
- Change your Signal View to Demodulated
- Measure the length of the smallest double-spike and enter that value into Bit Length
- Manually adjust the Bit Length and Error Tolerance
- Change Show Signal as to ASCII

Task 4: Network Traffic Extraction

Level of Effort: Medium to High

Task Description: Use a tool to decode and extract communications payload from RF capture.

Task Goal: Obtain data for following tasks.

ToorChat with Yardstick

1. Connect a 433Mhz antenna to your Yardstick
2. Plug in your Yardstick and connect it to your ControlThings VM
3. Use lsusb to verify OpenMoko is connected, which is your Yardstick
4. Start toorchat from the main menu or the command prompt
5. In toorchat, set a username by typing u followed by your preferred username
   - **Note**: Frequency and channel settings do not currently work in ToorChat
6. Send a message to the group by pressing the Enter key followed by your desired message

Toorchat Capture

- Start urh and create a new project
  - Set the sample rate to 1M and the frequency to 433M
- Under File, start the Spectrum Analyzer
  - Choose RTL-SDR as your capture device
  - Verify frequency and sample rate are set correctly
- Once you confirm you can see the traffic in the Spectrum Analyzer window, close it and open the Record Signal window
- Verify the device, sample rate, and frequency settings
  - You should visually see a signal spike
  - Save the signal and close the capture signal window
  - Use the main urh window to interpret your signal..... (next slide)
- Record yourself sending some messages in Toorchat.
- Use URH to decode your messages...

Toorchat Packet Analysis

- In the project that you captured your Toorchat traffic, change to the URH Analysis tab
- Add labels to your your messages to define the Toorchat protocol
  - Preamble
  - Synchronization / Sync Word
  - then……?!?!?!
- Check out the Toorchat source code to find hints at their protocol fields: /usr/local/share/ToorChat/libtoorchat.py

Task 5: RF Signal Transmission

Level of Effort: Medium to High

Task Description: Use a tool to transmit RF signals at the appropriate frequencies and hopping patterns to either replay captured data, impersonate the target field device, or attempting to cause denial of service scenarios.

Task Goal: Identify vulnerabilities in the RF signaling.

Replaying Captures with URH and RFCAT

- Make sure the toorchat terminal application is closed (CTRL-C) so it frees up the Yardstick device for us.
- Now back in URH, with your toorchat capture open, go to the Generator tab so we can replay our capture
- Drag one of your captures over from the left hand side to the right hand side, and it should show all the packets
- Click the Send via RFCAT button on the bottom of the URH window and have one of your neighbors verify you are sending the packets

Homework

Capture, decode, and retransmit Toorchat with a TX enabled SDR like HackRF

Try the exercise in the Appendix of this slide deck

Additional RF/SDR Resources

- Additional video tutorials
  - Mike Ossman's video series on SDR and HackRF
    https://greatscottgadgets.com/sdr/
  - GNU Radio Tutorial Series
    https://www.youtube.com/playlist?list=PL618122BD66C8B3C4
  - GNU Radio with 802.11 (WiFi) and 802.15.4
    http://www.ccs-labs.org/software/
- Examples of signal reverse engineering process
  - Example: GRC Transmission Analysis: Getting To the Bytes
    https://www.inguardians.com/2016/02/07/grc-transmission-analysis-getting-to-the-bytes/
  - Example: Blind Reverse Engineering a Wireless Protocol at 433MHz
    https://github.com/r-ohare/Amateur-SIGINT
- Other SDR and RFCAT projects
  - rtl_433: decode traffic from Devices that are broadcasting on 433.9 MHz like temperature sensors
    https://github.com/merbanan/rtl_433
  - rfcrack: capture, replay, and jam sub-Ghz RF traffic with rfcat
    https://github.com/cclabsInc/RFCrack

Contact Information

Justin Searle
personal: justin@controlthings.io
work: justin@inguardians.com
cell: 801-784-2052

twitter: @meeas
Facebook: www.facebook.com/m33as
LinkedIn: www.linkedin.com/in/meeas
GitHub: github.com/meeas, github.com/ControlThingsTools

ControlThings Platform Releases:

https://goo.gl/wwqxqb

Appendix

If you purchase a TI EZ430-Chronos watch, you can try the following exercise

Exploring your EZ430-Chronos Watch Target

- Replace the battery in your watch if needed
- Visit the following page to learn more about the watch
  http://processors.wiki.ti.com/index.php/EZ430-Chronos
- On that page, you will find a download link to install the software on your host (note: the Linux version is statically compiled to 32 bit so will not work in ControlThings Platform)
- After installing, run the software to start interacting with your watch
  - Once you see the interface, start the access point
  - On the Chronos watch, push the lower left hand button to get it into ACC mode, then press the bottom right hand button once to start communications
  - Verify communication by using the 3-axis accelerometer

Capturing and Analyzing Watch Traffic

- Once you know how to communicate with the watch, try using your RTL-SDR and URH to capture that communication and decode it
  - Find the frequency by looking up the FCC ID on the watch or digging through the ISM bands (there are 433, 868, or 915 versions of the watch...)
  - Capture the traffic with URH
  - Check out the source code screenshot earlier in this slide deck for some other nuggets to help you with your decode
  - Try decoding the traffic (the protocol is from TI and named Simplicity, which might in your research and decode efforts)