

Reverse Engineering the Inkbird

- [What are we Reverse Engineering?](#)
- [Specs](#)
- [Our Signal](#)
- [Building RTL-433 on Mac](#)
- [Running RTL_433](#)
- [Recording Signal using URH](#)
- [Other Tests of RTL_433](#)
 - [RadioHead ASK code on ESP8266 Board \(Lolin/Wemos mini\)](#)
 - [RCSwitch code on ESP8266 Board \(Lolin/Wemos mini\)](#)
- [References](#)

What are we Reverse Engineering?

Inkbird ITH-20R provides direct and accurate display of humidity and temperature. You can choose Fahrenheit or Celsius according to your personal preference.

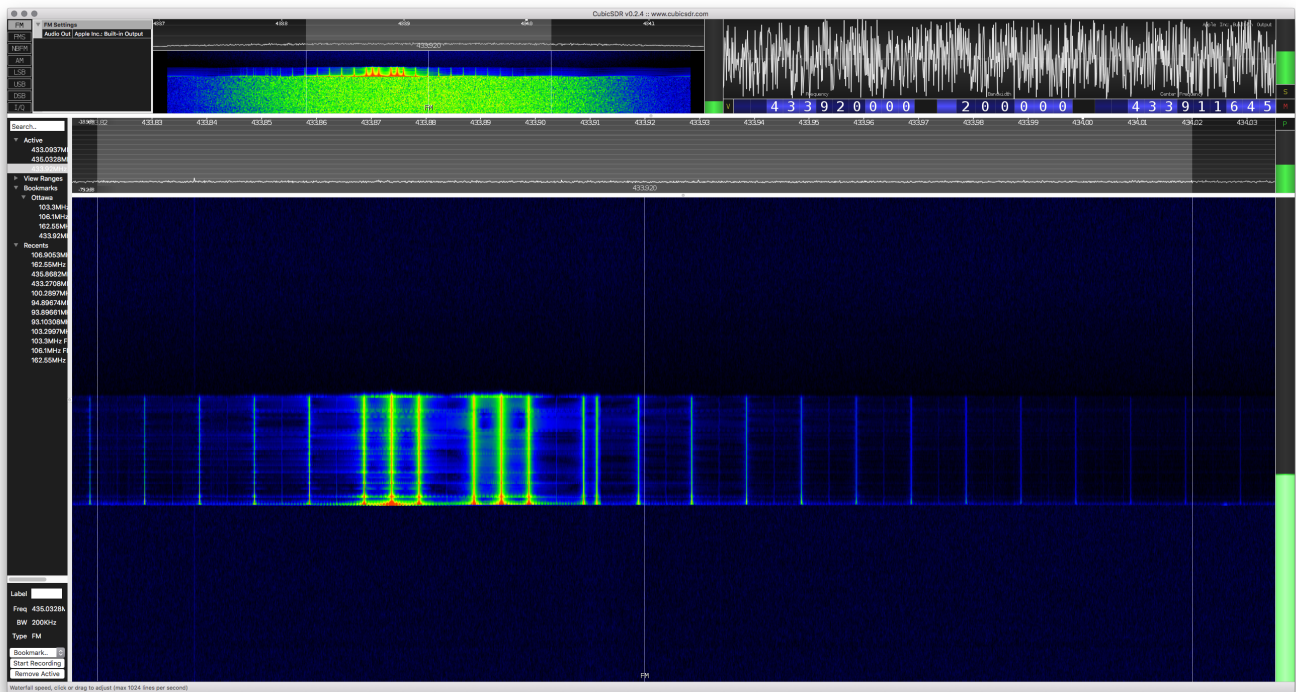
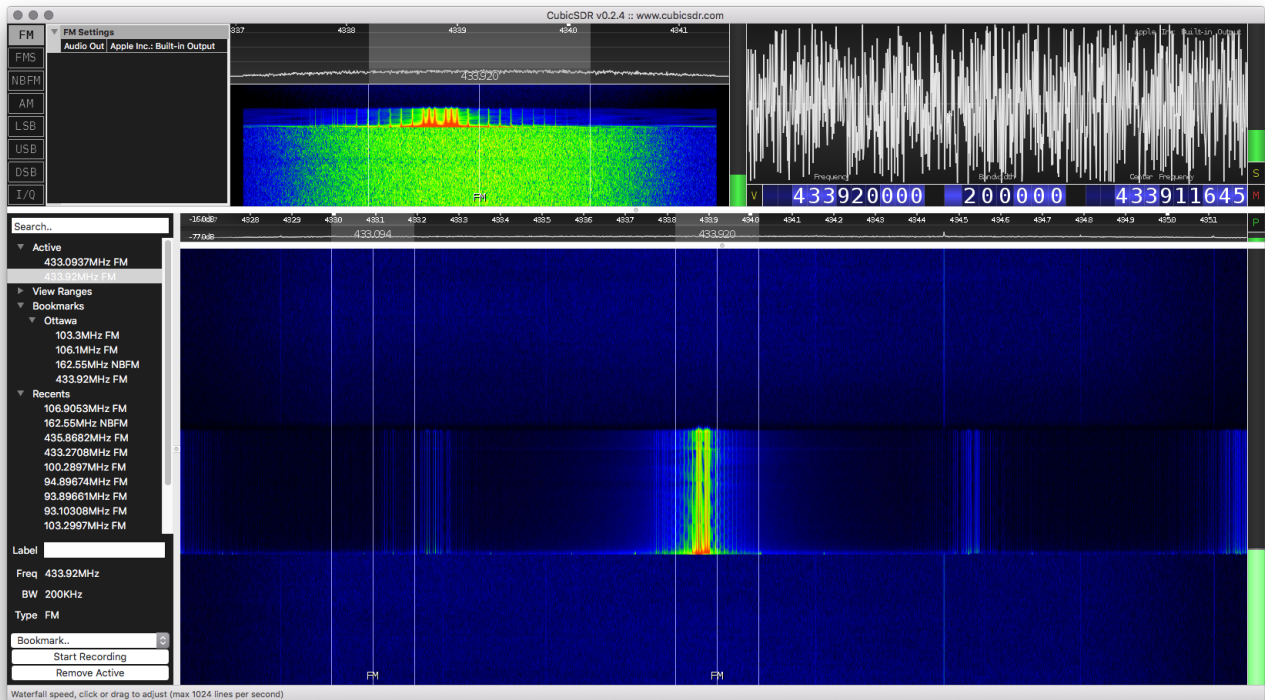
[blocked URL](#)[blocked URL](#)

Specs

Spec	Value
Make	Inkbird
Model	ITH-20R
User Manual	Inkbird ITH-20R.pdf
Sampling Period	10s
Transmission Frequency	433MHz
Transmission Distance	300 Feet

Our Signal

Using CubicSDR we can see our signal which happens every 80 seconds or so.



Building RTL-433 on Mac

Clone repository

> git clone https://github.com/merbanan/rtl_433.git

Install rtl-sdr and cmake using brew

```
> brew install rtl-sdr cmake
```

Build

```
cd rtl_433/  
mkdir build  
cd build  
cmake ..  
make  
make install
```

For more details see: https://github.com/merbanan/rtl_433/blob/master/docs/BUILDING.md

Running RTL_433

```
> rtl_433
```

```
$ rtl_433  
rtl_433 version 19.08-18-g8eecdbb branch master at 201909241811 inputs file rtl_tcp RTL-SDR  
Use -h for usage help and see https://triq.org/ for documentation.  
Trying conf file at "rtl_433.conf"..  
Trying conf file at "/Users/john.mehan/.config/rtl_433/rtl_433.conf"..  
Trying conf file at "/usr/local/etc/rtl_433/rtl_433.conf"..  
Trying conf file at "/etc/rtl_433/rtl_433.conf"..  
  
Consider using "-M newmodel" to transition to new model keys. This will become the default someday.  
A table of changes and discussion is at https://github.com/merbanan/rtl_433/pull/986.  
  
Registered 108 out of 138 device decoding protocols [ 1-4 8 11-12 15-17 19-21 23 25-26 29-36 38-60 63 67-71 73-  
100 102-103 108-116 119 121 124-128 131-138 ]  
Found Rafael Micro R820T tuner  
Exact sample rate is: 250000.000414 Hz  
[R82XX] PLL not locked!  
Sample rate set to 250000 S/s.  
Tuner gain set to Auto.  
Tuned to 433.920MHz.  
pulse_FSK_detect(): Maximum number of pulses reached!  
pulse_FSK_detect(): Maximum number of pulses reached!  
pulse_FSK_detect(): Maximum number of pulses reached!  
pulse_FSK_detect(): Maximum number of pulses reached!
```

Googling Error

The error means the program is apparently receiving a very long FSK encoded signal with more bits/pulses than any currently known devices. However the Ambient sensors are not FSK encoded, but OOK encoded instead. A likely cause is that your sensors are now sending their signal at the same time, disturbing the transmission of each other. As the sensors send at slightly different frequencies it is received as something that might look like a FSK encoded signal. It will happen as the periodic transmission from the sensors drift slightly and becomes overlapping, and it will disappear again as the timing drift further.

Using Pulse Analyser with all protocols disabled:

```

rtl_433 -A -R 0
rtl_433 version 19.08-18-g8eecd8b branch master at 201909241811 inputs file rtl_tcp RTL-SDR
Use -h for usage help and see https://triq.org/ for documentation.
Trying conf file at "rtl_433.conf"...
Trying conf file at "/Users/john.mehan/.config/rtl_433/rtl_433.conf"...
Trying conf file at "/usr/local/etc/rtl_433/rtl_433.conf"...
Trying conf file at "/etc/rtl_433/rtl_433.conf"...
Disabling all device decoders.

    Consider using "-M newmodel" to transition to new model keys. This will become the default someday.
    A table of changes and discussion is at https://github.com/merbanan/rtl_433/pull/986.

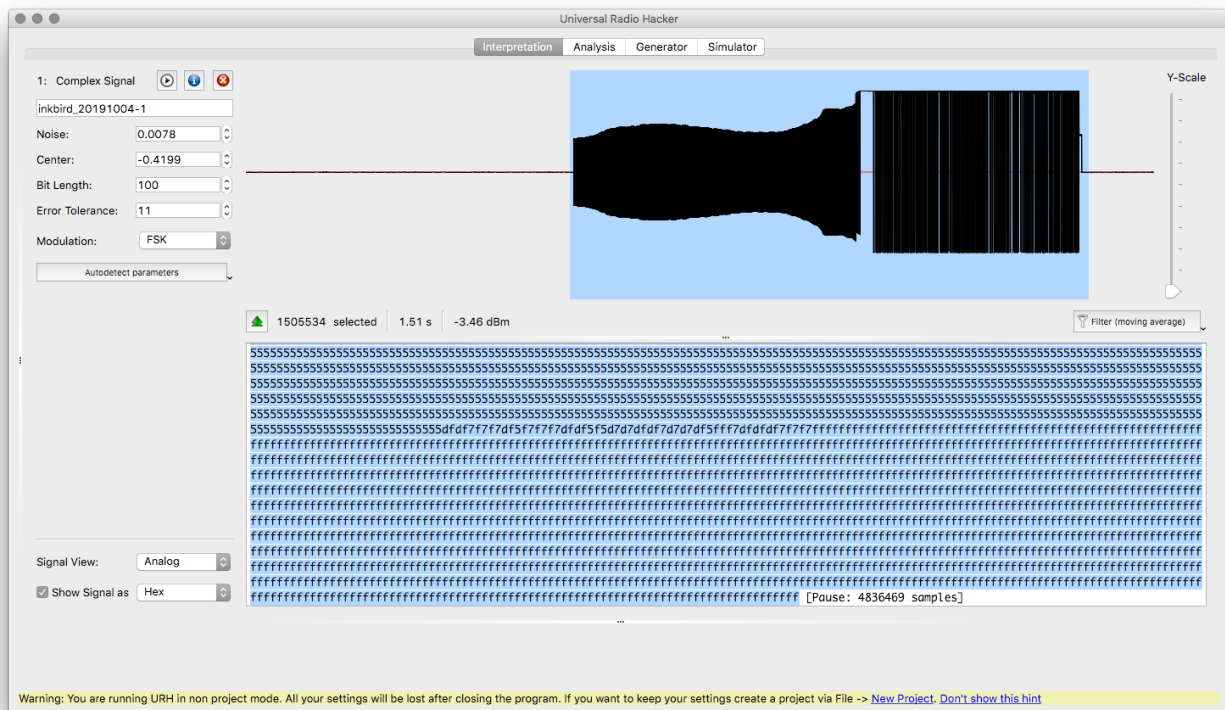
Registered 0 out of 138 device decoding protocols [ ]
Found Rafael Micro R820T tuner
Exact sample rate is: 250000.000414 Hz
[R82XX] PLL not locked!
Sample rate set to 250000 S/s.
Tuner gain set to Auto.
Tuned to 433.920MHz.

Detected OOK package          2019-10-02 12:21:20
Analyzing pulses...
Total count:    1, width: 1465.40 ms                (366351 S)
Pulse width distribution:
[ 0] count:    1, width: 1465404 us [1465404;1465404]      (366351 S)
Gap width distribution:
Pulse period distribution:
Level estimates [high, low]: 15892,    6
RSSI: -0.1 dB SNR: 33.6 dB Noise: -33.7 dB
Frequency offsets [F1, F2]: 19756,    0                (+75.4 kHz, +0.0 kHz)
Guessing modulation: Single pulse detected. Probably Frequency Shift Keying or just noise...

Detected OOK package          2019-10-02 12:22:41
Analyzing pulses...
Total count:    1, width: 1465.44 ms                (366359 S)
Pulse width distribution:
[ 0] count:    1, width: 1465436 us [1465436;1465436]      (366359 S)
Gap width distribution:
Pulse period distribution:
Level estimates [high, low]: 15921,    5
RSSI: -0.1 dB SNR: 34.2 dB Noise: -34.4 dB
Frequency offsets [F1, F2]: 20432,    0                (+77.9 kHz, +0.0 kHz)
Guessing modulation: Single pulse detected. Probably Frequency Shift Keying or just noise...

```

Recording Signal using URH



Data:

PAUSE

Data:

Iteration	After 5's	After a's	IN Temp	IN Humidity	OUT Temp	OUT Humidity	EXT Temp
1			66.7	47	67.1	56	65.6

Another Recording

Other Tests of RTL_433

RadioHead ASK code on ESP8266 Board (Lolin/Wemos mini)

Code

```
// ask_transmitter.pde
#include <RH_ASK.h>
#ifdef RH_HAVE_HARDWARE_SPI
#include <SPI.h> // Not actually used but needed to compile
#endif
RH_ASK driver(2000, D4, D2, D1); // (bps, RX, TX, PTT)

void setup()
{
  Serial.begin(115200);          // Debugging only
  Serial.println("\n\nask_transmitter");
}
void loop()
{
  const char *msg = "hello";
  driver.send((uint8_t *)msg, strlen(msg));
  driver.waitPacketSent();
  delay(200);
}
```

Results:

```

$ rtl_433
rtl_433 version 19.08-18-g8eecd8b branch master at 201909241811 inputs file rtl_tcp RTL-SDR
Use -h for usage help and see https://triiq.org/ for documentation.
Trying conf file at "rtl_433.conf"...
Trying conf file at "/Users/john.mehan/.config/rtl_433/rtl_433.conf"...
Trying conf file at "/usr/local/etc/rtl_433/rtl_433.conf"...
Trying conf file at "/etc/rtl_433/rtl_433.conf"...

    Consider using "-M newmodel" to transition to new model keys. This will become the default someday.
    A table of changes and discussion is at https://github.com/merbanan/rtl_433/pull/986.

Registered 108 out of 138 device decoding protocols [ 1-4 8 11-12 15-17 19-21 23 25-26 29-36 38-60 63 67-71 73-
100 102-103 108-116 119 121 124-128 131-138 ]
Found Rafael Micro R820T tuner
Exact sample rate is: 250000.000414 Hz
[R82XX] PLL not locked!
Sample rate set to 250000 S/s.
Tuner gain set to Auto.
Tuned to 433.920MHz.

-----
time       : 2019-10-02 08:36:26
model      : RadioHead ASK Data len   : 5           To           : 255           From           : 255
Id         : 0
Flags      : 0           Payload      : 104, 101, 108, 108, 111           Integrity : CRC
-----

```

RCSwitch code on ESP8266 Board (Lolin/Wemos mini)

Code:

```

#include <RCSwitch.h>

RCSwitch mySwitch = RCSwitch();

void setup() {
  Serial.begin(115200);

  // Transmitter is connected to Arduino Pin #10
  mySwitch.enableTransmit(D2);
  Serial.println("\n\nSender Started");
  // Optional set protocol (default is 1, will work for most outlets)
  // mySwitch.setProtocol(2);

  // Optional set pulse length.
  // mySwitch.setPulseLength(320);

  // Optional set number of transmission repetitions.
  // mySwitch.setRepeatTransmit(15);
}

void loop() {
  Serial.print(".");
  mySwitch.switchOn("11111", "00010");
  delay(10000);
}

```

Results:

Tuned to 433.920MHz.

time : 2019-10-02 09:13:58
model : Waveman Switch Transmitter id : A
channel : 4 button : 3 state : off

time : 2019-10-02 09:13:58
model : Akhan 100F14 remote keyless entry ID (20bit): 0x151
Data (4bit): 0x1 (Lock)

time : 2019-10-02 09:13:58
model : Waveman Switch Transmitter id : A
channel : 4 button : 3 state : off

time : 2019-10-02 09:13:58
model : Akhan 100F14 remote keyless entry ID (20bit): 0x151
Data (4bit): 0x1 (Lock)

time : 2019-10-02 09:13:58
model : Waveman Switch Transmitter id : A
channel : 4 button : 3 state : off

time : 2019-10-02 09:13:58
model : Akhan 100F14 remote keyless entry ID (20bit): 0x151
Data (4bit): 0x1 (Lock)

time : 2019-10-02 09:13:58
model : Waveman Switch Transmitter id : A
channel : 4 button : 3 state : off

time : 2019-10-02 09:13:58
model : Akhan 100F14 remote keyless entry ID (20bit): 0x151
Data (4bit): 0x1 (Lock)

time : 2019-10-02 09:13:58
model : Waveman Switch Transmitter id : A
channel : 4 button : 3 state : off

time : 2019-10-02 09:13:58
model : Akhan 100F14 remote keyless entry ID (20bit): 0x151
Data (4bit): 0x1 (Lock)

time : 2019-10-02 09:13:58
model : Waveman Switch Transmitter id : A
channel : 4 button : 3 state : off

time : 2019-10-02 09:13:58
model : Akhan 100F14 remote keyless entry ID (20bit): 0x151
Data (4bit): 0x1 (Lock)

time : 2019-10-02 09:13:59
model : Waveman Switch Transmitter id : A
channel : 4 button : 3 state : off

time : 2019-10-02 09:13:59
model : Akhan 100F14 remote keyless entry ID (20bit): 0x151
Data (4bit): 0x1 (Lock)

time : 2019-10-02 09:13:59
model : Waveman Switch Transmitter id : A
channel : 4 button : 3 state : off

time : 2019-10-02 09:13:59
model : Akhan 100F14 remote keyless entry ID (20bit): 0x151
Data (4bit): 0x1 (Lock)

time : 2019-10-02 09:13:59
model : Waveman Switch Transmitter id : A
channel : 4 button : 3 state : off

time : 2019-10-02 09:13:59

```

model      : Akhan 100F14 remote keyless entry      ID (20bit): 0x151
Data (4bit): 0x1 (Lock)

-----
time       : 2019-10-02 09:13:59
model      : Waveman Switch Transmitter             id       : A
channel    : 4           button    : 3             state    : off
-----
time       : 2019-10-02 09:13:59
model      : Akhan 100F14 remote keyless entry      ID (20bit): 0x151
Data (4bit): 0x1 (Lock)

```

References

Reference	URL
User Manual	Inkbird ITH-20R.pdf
Inkbird ITH-20R	https://www.ink-bird.com/products-data-logger-ith20.html
RTL-SDR: 433.92Mhz ASK/OOK Decoding of Various Devices with rtl_433	https://goughlui.com/2013/12/20/rtl-sdr-433-92mhz-askook-decoding-of-various-devices-with-rtl_433/
RTL_433	https://github.com/merbanan/rtl_433
Building RTL_433	https://github.com/merbanan/rtl_433/blob/master/docs/BUILDING.md
Frequency Shift Keying	https://en.wikipedia.org/wiki/Frequency-shift_keying