Honeywell RedLINK reverse-engineering experiments

The project is to create hardware and software to understand the Honeywell RedLINK wireless HVAC communication, and then build compatible devices.

Honeywell uses (and licenses to Mitsubishi) a proprietary wireless network to communicate among thermostats, temperature sensors, and wireless receivers connected to heating and air condition equipment like fancoils and heat pumps. They do not publicly document the details.

The network is similar to WiFi, but not the same. It is frequency-hopping spread spectrum communication in the 902-926 Mhz band. Data is sent in small packets up to about 30 bytes.

This project is a work in progress, and I will post updates here from time to time. As of 7 May 2015, I've built a Sniffer to decode the traffic between the Honeywell microprocessor in a remote temperature sensor and the TI CC1101 RF transceiver. I see how the chip is configured (and reconfigured when the frequency changes), and I see some examples of packets sent by the sensor.

Here are possible next steps:
- capture more examples of packets from other devices and try to deduce the format and protocol
- build a board with an Arduino that controls an RF transceiver and can send and receive commands (I'm going to try using the Anaren A1101R09A-EM1 evaluation board)

This is all a leisure-time activity, so it's not clear how quickly I will make more progress.

-- Len Shustek

RedLINK CC1101 RF transceiver configuration

reg addr	name	value in hex	interpretation
00	GDO 2	06	output is "sync word sent/rcvd"
01	GDO 1	0D	output is "serial data out"
02	GDO 0	2F. 38	output is sometimes "hardwired to 0" and sometimes "CLK_XOSC/16"
03	FIFO threshold	47	ADC retention, threshold is 32 bytes TX, 32 bytes RX (of 64-byte FIFOs)
04/05	sync word	63xx,	where xx varies with the channel number; see table below
06	packet length	63	99 bytes(but not used in variable-packet mode)
07	packet control 1	44	preamble quality threshold 2 no autoflush of RX FIFO on bad CRC append 2 status bytes to packet RSSI (dBm signal power), CRC OK, LQI) no address check
08	packet control 0	45	data whitening on normal FIFO mode CRC enabled variable packet length, set by first byte after sync word
09	device address	0, or 06	(for packet filtering)
0A	channel number	varies	hops among 50 values; see table below
0B	frequency control 1	06	IF frequency: 152 Khz
0C	frequency control 0	00	frequency offset for base, in units of 1587 Hz (none)
0D/0E /0F	frequency control word (H/M/L, 24 bits)	22B333 22B330	2,274,099 * 396.7 = 902.1350 Mhz (only at init) 2,274,096 * 396.7 = 902.1338 Mhz (all other times)
10	modem config 4	CA	BW = 26MHz/(8*(4+00)*2**3) = 26Mhz/(32*8) = 101.5 Khz symbol rate exponent = 10
11	modem config 3	83	symbol rate mantissa = 0x83 = 131 symbol rate = 26 Mhz * (256+131)*2**10)/2**28 = 38.383 Kb
12	modem config 2	12	enable DC blocking, GFSK modulation, disable Manchester, 16/16 sync word bits
13	modem config 1	62	disable forward error correction, 16 preamble bytes, chan spacing exponent = 2
14	modem config 0	F8	chan spacing mantissa = 248. default spacing = 199.951 Khz
15	deviation	34	exp=3, man=4; deviation = 19.0 Khz
16	state machine 2	07	default (end-of-packet timeout for sync)
17	state machine 1	00	CCA always, go idle after packet sent or received

18	state machine 0	18	calibrate when going to RX or TX from idle; expire count 64 (149-155 usec)
19	freq offset config	16	gain 3K, K/2, sat BWchan/4
1A	bit sync config	6C	defaults
1B	AGC control 2	43	
1C	AGC control 1	40	
1D	AGC control 0	91	
1E/1F	event timeout (H/L)	876B	default = 34,667, or 1 second
20	wake on radio control	F8	default
21	RX config	56	default
22	TX config	10	default (select PATABLE entry 0)
23	freq cal 3	E9	
24	freq cal 2	2A	
25	freq cal 1	00	
26	freq cal 0	1F	
27	RC osc config 1	41	
28	RC osc config 0	00	
29	freq calib ctl	59	default
2A	prod test	7F	default
2B	AGC test	3F	default
2C	test2	81	
2D	test1	35	
2E	test0	09	
…			
3E	PATABLE (power amp)	C0	default, always

Other notes

JimmySwimmy says RedLINK uses "50 channels, 903 to 926.4 Mhz, 69 Khz each channel, 400 Khz spacing".

Based on the CC1101 configuration, I see 101.5 Khz bandwidth channels with spacing of 199.9 Khz.

With a base frequency of 902.13 Mhz and a maximum observed channel number of 0x79 = 121, that implies that the highest frequency is 902.13 + 121*.1999 = 926.31, which just fits into the allowed band.

In the US the FCC requires a minimum of 50 channels for 1W operation in this band, and a channel change at least every 400 msec; see http://www.ti.com.cn/cn/lit/an/swra077/swra077.pdf.

Here is the observed frequency-hopping sequence of 50 channel numbers that Honeywell uses and changes every 2.685 msec. That's 372.44 times per second. It goes through all 50 channels in 134.25 msec.

```
 71, 38, 75, 49, 0C, 51, 14, 06, 42, 20, 5D, 04, 3E, 1A, 57, 1C, 59, 79, 3C, 1E, 5B, 5F, 22, 63, 2E,
 46, 0A, 61, 2C, 53, 16, 32, 6F, 3A, 77, 4F, 12, 44, 08, 6D, 30, 4D, 10, 18, 55, 36, 73, 0E, 4B, 34
```

Here is the low byte of the sync word corresponding to each channel; the high byte is always 63.

```
98, 81, 9A, 88, 6F, 8C, 73, 6C, 85, 79, 92, 6B, 84, 76, 8F, 77, 90, 9C, 83, 78, 91, 93, 7A, 95, 7C,
87, 6E, 94, 7B, 8D, 74, 7E, 97, 82, 9B, 8B, 72, 86, 6D, 96, 7D, 8A, 71, 75, 8E, 80, 99, 70, 89, 7F
```

I'm guessing these are just stored in a table, not generated by an algorithm in the microprocessor.

The CC1101 chip takes 712-724 usec to calibrate after each frequency hop, so the blanking interval is 787-799 usec.

Observations on receiver while thermostat is connecting to it

- Receiver enables receive every 2.685 msec in normal channel sequence, 134.25 msec around
- On channel 49, waits 90.326 msec, then skips to 3A and continues normally with 77
  - 90.326 msec is 33.64 channels
- On channel 36, waits 86.544 msec, then skips to 4F (40?? chans away) and continues normally with 12
  - 86.544 msec is 32.232 channels
- On channel 49, waits 88.620 msec, then skips to 12 (33 chans away) and continues normally with 44
  - 88.620 msec is 33 channels
- On channel 0C, waits, 88.620 msec, then skips to 44 (33 chans away) and continues normally with 08
  - 88.620 msec is 33 channels
- On channel 51, waits 88.619 msec, then skips to 08 (33 chans away) and continues normally with 6D
  - 88.619 msec is 33 channels
- With rcv enb on chan 04, sends 22 bytes on chans 3E, 1A, 57, and 1C (normal sequence) 59 msec apart
  - 59 msec is 22 channels
- Then wait 15 msec and rcv enb on chan 77 (19 chans away) then 4F, 12… (normal sequence)
  - 15 msec is 5.6 channels
- With rcv enb on chan 20 for 142 msec, sends 13 bytes on chan 5D (normal sequence), then enb rcv 5D
  - 142 msec is 52.88 channels
- Then waits 15 msec and rcv enb 1E (9 chans away) and then normal rcv sequence
  - 15 msec is 5.6 chans
- On channel 53, waits 88.620 msec, then skips to 3E (33 chans away) and continues normally with 1A
  - 88.620 msec is 33 chans
- With rcv enb on chan 77 for 142 msec, sends 13 bytes on chan 4F (normal sequence) then enb rcv 4F
- Then waits 11.7 msec and rcv enb 18 (8 chans away) and then normal rcv sequence
  - 11.7 msec is 4.3 chans
- With rcv enb on chan 4F for 142 msec, sends 13 bytes on chan 12 (normal sequence) then enb rcv 12
- Then waits 87.6 msec and rcv enb 22 (36 chans away) and then normal rcv sequence
  - 87.6 msec is 32.6 chans
- With rcv enb on chan 0E for 142 msec, sends 13 bytes on chan 4B (normal sequence) then enb rcv 4B
- Then waits 11.7 msec and rcv enb 14 (8 chans away) and then normal rcv sequence
- On channel 73, waits 86.2 msec. then skips to 53 (33 chans away) and continues normally with 16
- On channel 0E, waits 88.6 msec, then skips to 16 (33 chans away) and continues normally with 32
- On channel 4B, waits 88.6 msec, then skips to 32 (33 chans away) and continues normally with 6F
- On channel 34, waits 88.6 msec, then skips to 6F (33 chans away) and continues normally with 3A
- On channel 38, waits 86.2 msec, then skips to 77 (33 chans away) and continues normally with 4F
- On channel 75, waits 88.6 msec, then skips to 4F (33 chans away) and continues normally with 12
- On channel 49, waits 107.4 mse, then skips to 36 (42 chans away) and continues normally with 73

    - 107.4 msec is 40.00 chans
- On channel 14, waits 91.13 msec, then skips to 34 (43 chans away) and continues normally with 71
- On channel 4D, waits 135 msec and receives 10+5 bytes on 4D then sends 22 bytes on 4D
- Rcv enb on 4B (7 chans away) and then continues normally with 34
- On channel 79, waits 14.9 msec and receives 10+5 bytes on 79, then sends 14 bytes on 79

Packet exchange:

rcvr: 22 bytes,

Observations for the external temp sensor, during "link" phase:

It sends 19 or 22 byte packets every ½ second or so.

19 x 8 at 38.383 Khz would be 3.96 msec
22 x 8 at 38.383 Khz would be 4.58 msec

At 38.383 Khz symbol rate, each byte would take 208 usec. Each bit would take 26 usec.

But on the scope, small packets take 82 usec, large take 275 usec. What's going on? That's much faster than the symbol rate as computed above.

Packets start with 16 preamble bytes (or more if data isn't in the FIFO yet), then 2 bytes of sync, then length, then data, then 2-byte CRC. So a 22 byte data packet really takes 16+2+1+22+2 = 43 bytes.

For the 275 usec packets it looks on the scope like 48 bytes + 2*7 bytes + 1 byte = 63 bytes?
Each of the 48 bytes takes 2 usec with 1.5 usec between. Thus 3.5 usec/byte.
(Is that per bit with GFSK? Still doesn't comport with 19 or 22 byte packets.)

**** UPDATE: the scope traces are misleading; it's seeing communication with the CC1101 chip!!

The receiver changes listening channels every 2.685 msec, yet somehow it receives the packet! How does it know the frequency?

The temp sensor uses the same 50 channels, but in a different sequence that doesn't depend on the timing. It always uses two adjacent channels of the normal sequence, sometimes forward, sometimes reversed.

Example transmitted data packets from the Honeywell C7189R1004 indoor temperature sensor when not linked to any receiver:

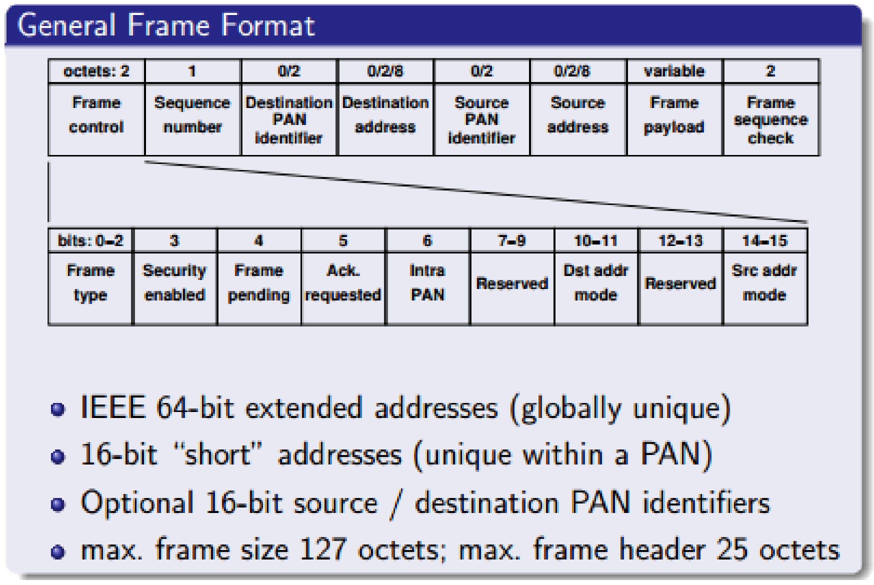
```
12 23 30 0B FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81
12 23 20 03 FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 1E, 44, 08)
12 23 F4 3F FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81

15 03 31 E4 E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 07 ED 7F FF 06
   ff ff ff  next freq?
15 03 9F 3D E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 BC 7F FF 06  (chan 63, 2E, 73)
15 03 56 65 E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 BC 7F FF 00  (chan 0E)
15 03 02 F5 E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 C4 7F FF 00  (chan 4B)
15 03 51 61 E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 C7 7F FF 00  (chan 06)
15 03 16 A9 E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 C7 7F FF 00  (chan 42)
15 03 31 24 E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 CC 7F FF 00  (chan 6D)
15 03 98 73 E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 D2 7F FF 00  (chan 30)
15 03 E9 7A E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 D7 7F FF 00  (chan 49)
15 03 35 9A E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 DC 7F FF 00  (chan 0C)
15 03 01 53 E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 DA 7F FF 00  (chan 1A)
15 03 A3 EE E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 E2 7F FF 00  (chan 57)
15 03 79 FB E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 E4 7F FF 00  (chan 4F)
15 03 03 E5 E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 EA 7F FF 00  (chan 12)
15 03 FB AF E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 E7 7F FF 00  (chan 14)
15 03 2A E2 E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 EA 7F FF 00  (chan 51)
15 03 F2 4F E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 EC 7F FF 00  (chan 4D)
15 03 61 EF E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 EF 7F FF 00  (chan 10)
15 03 FD BD E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F2 7F FF 00  (chan 5D)
15 03 EA DC E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F2 7F FF 00  (chan 20)
15 03 59 65 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F2 7F FF 00  (chan 32)
15 03 B5 4F E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 EF 7F FF 00  (chan 6F)
15 03 09 05 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F2 7F FF 00  (chan 71)
15 03 67 2D E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F2 7F FF 00  (chan 34)
15 03 C8 E9 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F5 7F FF 00  (chan 75)
15 03 A5 50 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F2 7F FF 00  (chan 38)
15 03 7D 6B E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 FA 7F FF 00  (chan 16)
15 03 83 21 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F7 7F FF 00  (chan 53)
15 03 D5 21 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 FD 7F FF 00  (chan 77)
… missing chan 3A
15 03 5A 5B E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 FD 7F FF 00  (chan 59)
15 03 E2 97 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 FA 7F FF 00  (chan 1C)
15 03 CB 4C E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 09 05 7F FF 00  (chan 18)
15 03 60 4D E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 09 07 7F FF 00  (chan 55)
15 03 B5 D1 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 09 02 7F FF 00  (chan 22)
```

```
reset

15 03 27 FF E8 1F E8 1F F0 01 00 0A 12 80 00 46 2C 08 FD 7F FF 06  (chan 73)
15 03 27 FF E8 1F E8 1F F0 01 00 0A 12 80 00 46 2C 08 FD 7F FF 06  (chan 36)
15 03 27 FF E8 1F E8 1F F0 01 00 0A 12 80 00 46 2C 08 FD 7F FF 06  (chan 2E)
15 03 27 FF E8 1F E8 1F F0 01 00 0A 12 80 00 46 2C 08 FD 7F FF 06  (chan 63)
15 03 51 CB E8 1F E8 1F F0 01 00 0A 12 80 00 46 2B 09 12 7F FF 00  (chan 0E)
12 23 08 6C FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 4B)
12 23 08 6C FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 06)
12 23 08 6C FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 42)
12 23 08 6C FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 6D)
15 03 26 DC E8 1F E8 1F F0 01 00 0A 12 80 00 46 2C 09 0D 7F FF 06  (chan 30)
missing  chans 49, 0C, 1A
12 23 AA CD FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 57)
12 23 AA CD FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 4F)
12 23 AA CD FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 12)
12 23 AA CD FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 14)
15 03 03 13 E8 1F E8 1F F0 01 00 0A 12 80 00 46 2B 09 07 7F FF 06  (chan 51)
15 03 03 13 E8 1F E8 1F F0 01 00 0A 12 80 00 46 2B 09 07 7F FF 06  (chan 4D)
15 03 03 13 E8 1F E8 1F F0 01 00 0A 12 80 00 46 2B 09 07 7F FF 06  (chan 10)

12 23 F4 3F FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 5B)
12 23 F4 3F FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 1E)
12 23 F4 3F FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 44)
12 23 F4 3F FF FE E8 1F F0 00 00 87 82 12 00 E8 1F FF 81  (chan 08)
15 03 9F 3D E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 BC 7F FF 06  (chan 73)
15 03 9F 3D E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 BC 7F FF 06  (chan 36)
15 03 9F 3D E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 BC 7F FF 06  (chan 2E)
15 03 9F 3D E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 BC 7F FF 06  (chan 63)
15 03 56 65 E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 BC 7F FF 00  (chan 0E)
15 03 02 F5 E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 C4 7F FF 00  (chan 4B)
15 03 51 61 E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 C7 7F FF 00  (chan 06)
15 03 16 A9 E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 C7 7F FF 00  (chan 42)
15 03 31 24 E8 1F E8 1F F0 01 00 0A 12 80 00 46 34 08 CC 7F FF 00  (chan 6D)
15 03 98 73 E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 D2 7F FF 00  (chan 30)
15 03 E9 7A E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 D7 7F FF 00  (chan 49)
15 03 35 9A E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 DC 7F FF 00  (chan 0C)
15 03 01 53 E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 DA 7F FF 00  (chan 1A)
15 03 A3 EE E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 E2 7F FF 00  (chan 57)
15 03 79 FB E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 E4 7F FF 00  (chan 4F)
15 03 03 E5 E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 EA 7F FF 00  (chan 12)
15 03 FB AF E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 E7 7F FF 00  (chan 14)
15 03 2A E2 E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 EA 7F FF 00  (chan 51)
15 03 F2 4F E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 EC 7F FF 00  (chan 4D)
```

```
15 03 61 EF E8 1F E8 1F F0 01 00 0A 12 80 00 46 33 08 EF 7F FF 00  (chan 10)
15 03 FD BD E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F2 7F FF 00  (chan 5D)
15 03 EA DC E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F2 7F FF 00  (chan 20)
15 03 59 65 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F2 7F FF 00  (chan 32)
15 03 B5 4F E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 EF 7F FF 00  (chan 6F)
15 03 09 05 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F2 7F FF 00  (chan 71)
15 03 67 2D E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F2 7F FF 00  (chan 34)
15 03 C8 E9 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F5 7F FF 00  (chan 75)
15 03 A5 50 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F2 7F FF 00  (chan 38)
15 03 7D 6B E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 FA 7F FF 00  (chan 16)
15 03 83 21 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 F7 7F FF 00  (chan 53)
15 03 D5 21 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 FD 7F FF 00  (chan 77)
15 03 1D BD E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 FD 7F FF 00  (chan 3A)
15 03 5A 5B E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 FD 7F FF 00  (chan 59)
15 03 E2 97 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 08 FA 7F FF 00  (chan 1C)
15 03 CB 4C E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 09 05 7F FF 00  (chan 18)
15 03 60 4D E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 09 07 7F FF 00  (chan 55)
15 03 B5 D1 E8 1F E8 1F F0 01 00 0A 12 80 00 46 32 09 02 7F FF 00  (chan 22)
```

Does RedLINK use the IEEE 802.15.4 standard, like ZigBee?

If so, then the packet format after the length byte would need to be like this:



General Frame Format

octets: 2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination PAN identifier	Destination address	Source PAN identifier	Source address	Frame payload	Frame sequence check

bits: 0–2	3	4	5	6	7–9	10–11	12–13	14–15
Frame type	Security enabled	Frame pending	Ack. requested	Intra PAN	Reserved	Dst addr mode	Reserved	Src addr mode

- IEEE 64-bit extended addresses (globally unique)
- 16-bit "short" addresses (unique within a PAN)
- Optional 16-bit source / destination PAN identifiers
- max. frame size 127 octets; max. frame header 25 octets

"All frame formats in this subclause are depicted in the order in which they are transmitted by the PHY, from left to right, where the leftmost bit is transmitted first in time. Bits within each field are numbered from 0 (leftmost and least significant) to k – 1 (rightmost and most significant), where the length of the field is k bits. Fields that are longer than a single octet are sent to the PHY in the order from the octet containing the lowest numbered bits to the octet containing the highest numbered bits."

I'm not sure how to interpret that because I don't know which way the TI CC1101 chip sends. But of the four interpretations of the 0x2330 frame control we see (23 30, 30 23, bits reversed or not), the only one that has no bits in reserved fields on is 001 1 0 0 0 000 10 00 11, which means "data packet, security enabled, 16-bit dst addr, 64-bit src addr". It seems unlikely that the address sizes would be different. And that only leaves 5 bytes of payload.

So I don't think RedLINK is using 802.15.4.