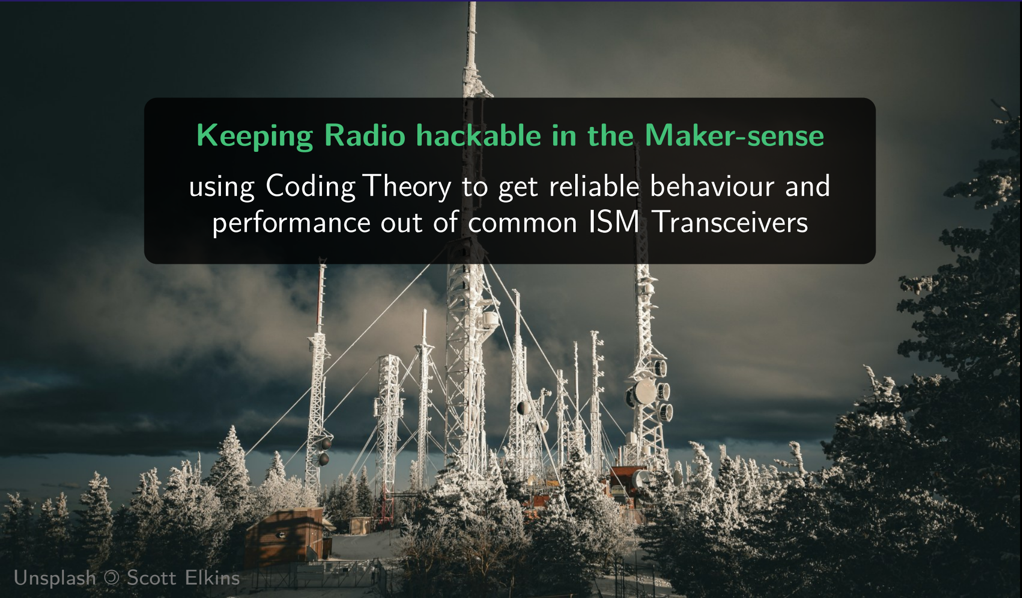
# Reliable Radio Communications

Unsplash © Scott Elkins



Bernerd DO3RB



## Keeping Radio hackable in the Maker-sense

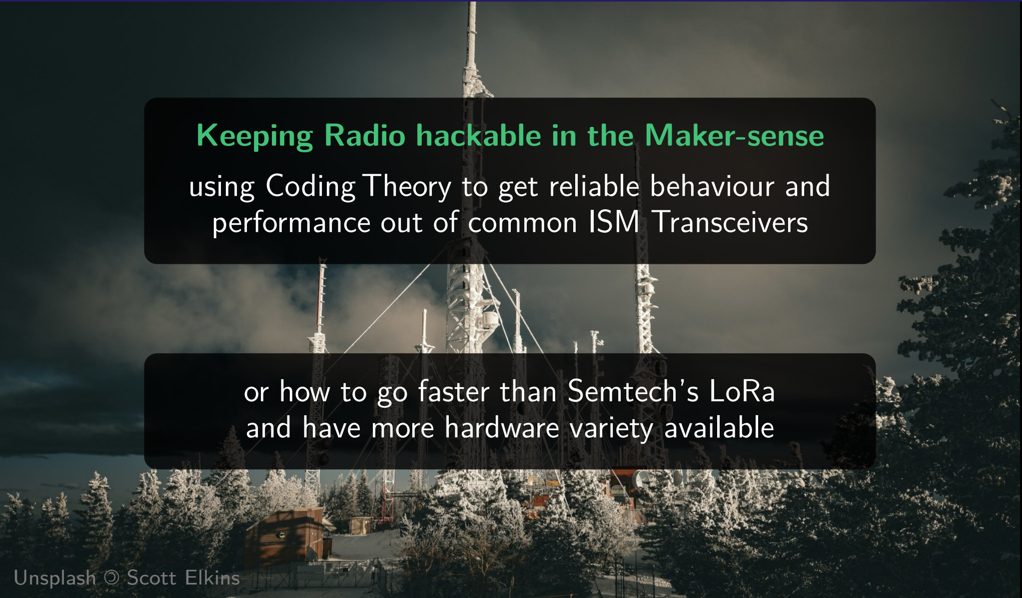
using Coding Theory to get reliable behaviour and performance out of common ISM Transceivers

Unsplash © Scott Elkins

## Keeping Radio hackable in the Maker-sense

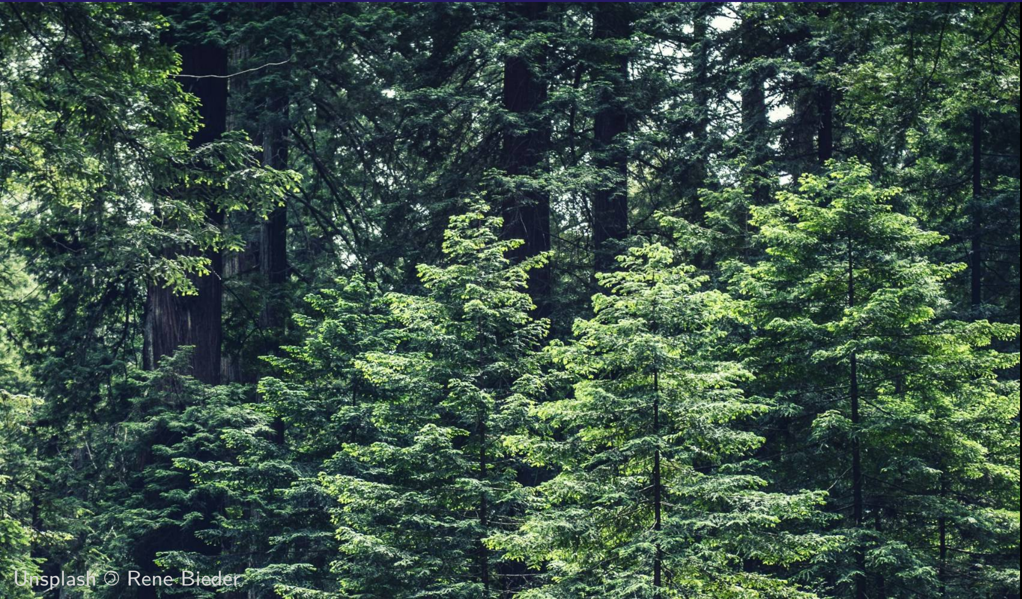
using Coding Theory to get reliable behaviour and performance out of common ISM Transceivers

or how to go faster than Semtech's LoRa and have more hardware variety available

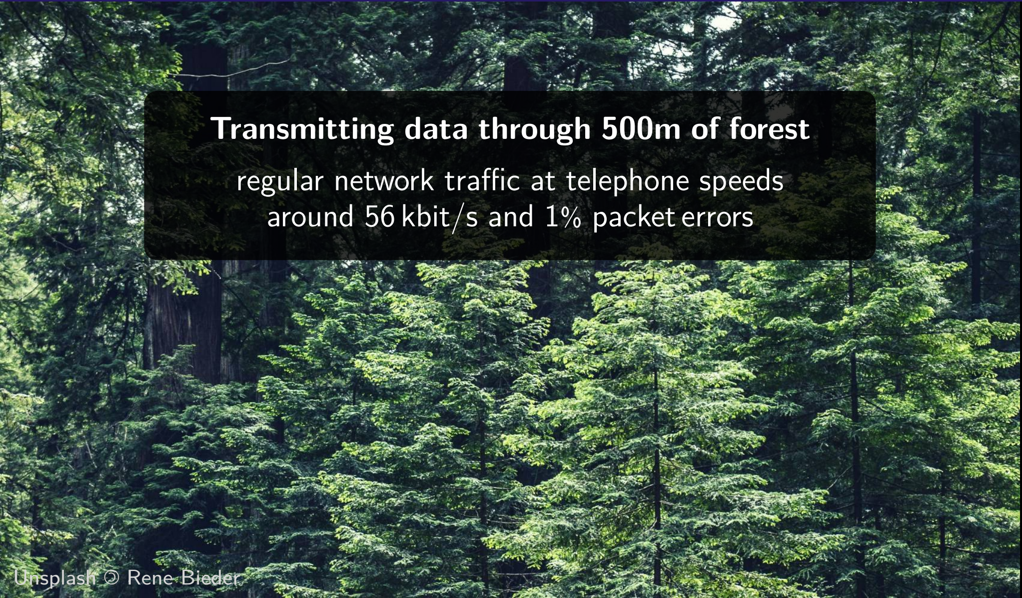


Unsplash © Scott Elkins

# dense Spruce Trees acting as vertical Absorbers



# dense Spruce Trees acting as vertical Absorbers



**Transmitting data through 500m of forest**

regular network traffic at telephone speeds
around 56 kbit/s and 1% packet errors

Unsplash © Rene Bieder

**Transmitting data through 500m of forest**

regular network traffic at telephone speeds
around 56 kbit/s and 1% packet errors

regular Wi-Fi drops out after 100m

cellular Phone drops out after 300m

Unsplash © Rene Bieder

# State of the Radio World

- *oldschool effort*

  PacketRadio using RaspberryPi with Direwolf connected to transceiver
  much hardware but would work, FX.25, kiss- or tncattach, limited to 9k6

- *being adventurous*

  simple subGHz ISM transceivers plus microcontroller as network device
  works over some range at 100 kbit/s up to 1 Mbit/s

- *proprietary commitment*

  Semtech's LoRa transceivers plus microcontroller as network device
  works over excellent range but limited to 22 kbit/s at SF7

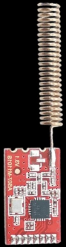
- *trusting the complexity*

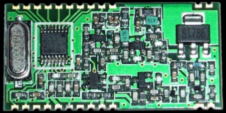
  IEEE 802.11ah HaLow 1-8 Mbit/s at 868 MHz announced in 2017
  2024 some boards got available with questionable driver support

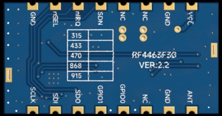
# Primer on the Intended Scope of the Codec

**Transceivers with raw access to the spectrum**

The mission is to
*make them work reliably*
especially at
maximum baudrate





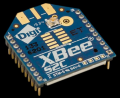



SiLabs 4421…4463
AnalogDevs ADF7021
TexasInst CC1101
and many more

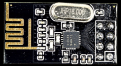
100 kb/s – 1 Mb/s
2FSK MSK 4(G)FSK

**inside scope**

---




IEEE 802.15.4
868M 20 kb/s
ZigBee stack
only checksum

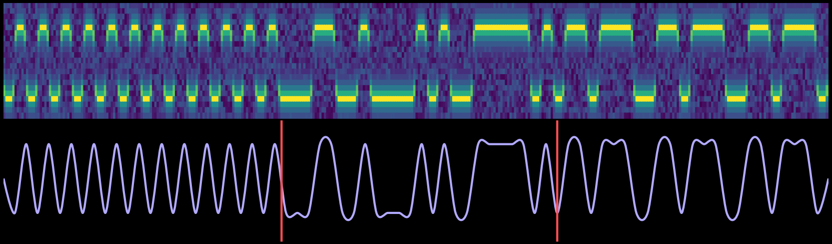


NordicRF24L
2.4G 2 Mb/s
MAC protocol
only checksum


Semtech LoRa
868M SF7 22k
chirp spectrum
plus Hamming

# on the Problem seen as Signal over Time

## Frame Acquisition Sequence



| Preamble | Syncword | Payload |
|---|---|---|
| *prime the receiver* | *notify the receiver* | *appease the receiver* |
| carrier tracking | detect start of frame | transmit datastream |
| clock recovery | spiky autocorrelation | signal conditioning |



Bernerd DO3RB

# Related Developments using Coding Theory



m17project.org

Opensource replacement for **Digital Voice** modes

4 FSK raw 4.8 kBaud

½ convolutional coder

xor with PRBS 9

QPP bit interleaver

9.6 kb/s ½ **4.8** kb/s

OpenRTX for handheld radios e.g. MD380 RT3

cats.radio

Modern replacement for the **APRS** packet mode

2 FSK raw 9.6 kBaud

½ LDPC labrador-rs

xor with LFSR 10

32 bit matrix interleaver

9.6 kb/s ½ **4.8** kb/s

RasPi-DSP STM32-Buffer Si4463-Transceiver Bundle

github.com/do3rb

OSI **Network** over ISM Transceivers at fullspeed

2 FSK raw 115.2 kBaud

½ Golay FEC with RLL

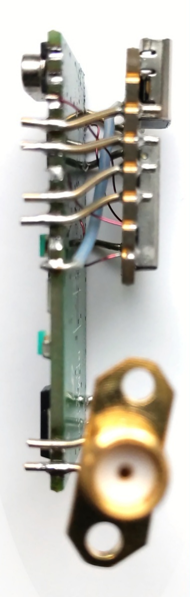
mod provides whitening

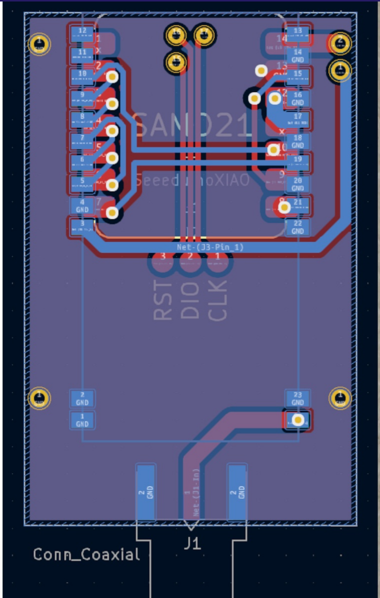
byte wise interleavable

115.2 kb/s ½ **57.6** kb/s

USB WLAN adapters for pushing through woods

# What does it look like?






# Last year's figures of merit

- composite USB CDC ( ACM Serial + ECM Ethernet )
  TCP/IP, HTTP, SSH, MTU 1500, all working
  behaves zeroconf plug'n'play under Linux
- TRX Si4421 434 MHz 100 mW raw 115.2 kBaud coded **57** kbit/s
  traverses the forest
- MCU SAMD21 Cortex-M0 Core @ 48MHz Flash @ 24MHz
  Decoder crunches **2.4** Mbit/s
- Benchmark 10k pings over two hours gives:
  **1.54** % packet loss 21.605/**22**.288/26.885/0.219 ms

# Thanks to Lorenz, Daniel, and Farkaš

DSP-MCOM 2005
Košice, Slovak republic
13-14 September 2004

## Extended Golay Code with Transcontrol Properties

Richard Chomist[1], Peter Farkaš[2]
[1]Author is with Air Traffic Services of the Slovak republic, M. R. Štefánik Airport, 823 07 Bratislava 21, Slovak Republic, e-mail: richard.chomist@lps.sk
[2]Author is with Slovak University of Technology, Ilkovicova 3, 812 19 Bratislava, Slovak Republic, e-mail: p.farkas@ieee.org

***Abstract*** **- There are many known approaches for improving the run length properties of block and convolutional codes without increasing overall redundancy [1]-[6]. In this paper we present a method for introducing Run-length limitation properties into binary Extended Golay Code with just a slight modification of output sequence and with no impact on original error-correction properties.**

***Index terms*** **– Extended Golay Code, Transcontrol, Run-length Limitation**

### 1 Introduction

Error control codes introduce redundancy in order to protect transmitted or stored information against errors [1]. Line codes also add redundancy, in order to meet the constraints of real discrete channels [1]. The construction methods of error control codes and line codes are presented separately in the traditional literature. Later it was shown in [4]-[6] that under certain conditions a combined code, i.e. a code that has error control capabilities as well as constraints performs better than a concatenated coding scheme. In [4] the name Transcontrol codes was proposed for these codes.

There are different known approaches to the construction of such codes with combined translation and error control properties based on modifying existing block or convolutional codes [2]-[6]. Most of these techniques use modulo two addition of a special binary vector or sequence, so called modifier, to the code words. In Section 2 we will shortly describe the general ideas for Transcontrol code construction based on usage of modifiers. In Section 3 binary Extended Golay Code is introduced and consequently in Section 4 some applicability of transcontrol properties is discussed. In Conclusions possible future research directions concerning the new transcontrol block codes are proposed.

### 2 Transcontrol code principles

Modulo two addition of a pseudo random sequence (often called 'side stream scrambling') is a known technique, which is often applied for elimination of long runs of equal symbols in transmitted or stored information sequence. It is obvious that if an input sequence contains long runs of identical binary symbols (all ones or all zeros), than the modulo two addition of the pseudo random sequence inverts some positions and so it prevents long runs of identical symbols at the output of the modulo two adder.

However, there is a disadvantage of modulo two addition of a pseudo random sequence. Namely, the elimination of long sequences of equal symbols is not guaranteed for all cases. For example, if the input sequence is identical or complementary to the sequence delivered by the pseudo random sequence generator the resulting sequence will be composed only from zeros or ones, respectively. Therefore in order to omit such unwanted events it is much more appropriate to add a sequence which is adapted to a structure of codewords. We will call such adapted sequence modifier. The resulting output sequence could be interpreted as a sequence of a transcontrol codewords, which is illustrated in Fig. 1.

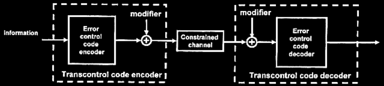



Fig.1 Illustration of the transcontrol code principle, where a synchronized modifier is applied

| | | n | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
|---|---|---|---|---|---|---|---|---|---|
| Order | RLL | | | | | | | | |
| 1 | 6 | | **4** | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 14 | | 7 | **11** | 16 | 22 | 29 | 37 | 46 |
| 3 | 30 | | | 15 | **26** | 42 | 64 | 93 | 130 |
| 4 | 62 | | | | 31 | **57** | 99 | 163 | 256 |
| 5 | 126 | | | | | 63 | **120** | 219 | 382 |
| 6 | 254 | | | | | | 127 | **247** | 466 |
| 7 | 510 | | | | | | | 255 | **502** |
| 8 | 1022 | | | | | | | | 511 |

**Table 1** Reed-Muller Codes - input vectors lengths ($k$)

We can see that for rates 11/16 and 16/32 a RLL=14 was achieved, therefore we can expect, that formerly presented RLL=22 for 12/24 Golay code will not be the maximum. In the following we will confirm that our expectations were correct.

$$G^* = \left[\begin{array}{cccccccc|cccccccc|cccccccc}
1&1&0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&1&1&0&0&0&0&0\\
1&0&0&1&0&0&0&0&1&1&0&0&0&0&0&0&0&1&1&1&1&0&0&0\\
1&1&1&0&1&0&0&0&0&1&1&0&0&0&0&0&0&0&1&1&0&0&0&0\\
1&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&1&1&0&1&0&1&0&0\\
1&0&1&0&0&0&0&0&0&1&0&1&0&0&0&0&1&1&1&0&0&0&1&0\\
1&0&0&1&0&0&0&0&0&0&1&1&0&0&0&0&1&0&1&1&0&0&0&1\\
1&1&1&1&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&1&0&0&0&0\\
1&1&1&1&0&0&0&0&1&0&0&0&0&0&1&0&1&0&1&0&0&0&0&0\\
1&1&0&0&0&0&0&0&1&1&0&1&0&0&0&1&1&0&0&1&0&0&0&0\\
1&0&1&1&0&1&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&1&0&0&1&0&0&1&1&0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&1&1&1&0&0&0&1&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0
\end{array}\right]$$

$$r = \left[\begin{array}{cccccccc|cccccccc|cccccccc}
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1
\end{array}\right]$$

$$r = \left[\begin{array}{cccccccc|cccccccc|cccccccc}
0&0&0&0&0&1&1&1&0&0&0&1&1&1&1&1&0&0&0&0&0&1&1&1
\end{array}\right]$$

Fig.3 Modified Generation matrix for Extended Golay code with possible modifiers

of the original code. Furthermore, we can perform permutation of columns which is equivalent to reordering of information bits in output sequence on encoder site. Once this reordering is reversely done on the decoder site before decoding process takes place, we can consider this operation as transparent and with no influence to original error control properties of the given code.

By taking principles given in part 2 into account we have performed such permutation on G matrix of Extended Golay code in order to achieve as much as possible sub-matrixes with an even Hamming weight in each row. The result is in Fig. 3. In this case the s=8, therefore:

$$RLL = 2s\text{-}2 = 2\cdot 8\text{-}2 = 14 \qquad (9)$$

Therefore the Maximum run-length of 14 zeros or ones in output sequence respectively can be guaranteed.

### 4 Conclusions

In this paper, we have presented some new Transcontrol codes obtained by a slight modification of (24, 12, 8) binary Extended Golay Code with no impact on original error control capabilities.

The run-length limitations achieved in the last presented approach with reordering of columns are fully

The principle of final reordering of columns and the proof that this is the maximal achievable result will be subject of further research.

**References**

[1] Blahut R. E., *Principles and practice of information theory*, Addison Wesley, Reading/Mass., 1987.
[2] Popplewell, A.; O'Reilly, J.: "Runlength limited

# This year's figures of merit

- composite USB CDC ( ACM Serial + ECM Ethernet )
  TCP/IP, HTTP, SSH, MTU 1500, all working
  behaves zeroconf plug'n'play under Linux
- TRX Si4421 434 MHz 100 mW raw 115.2 kBaud coded **57** kbit/s
  traverses the forest
- MCU SAMD21 Cortex-M0 Core @ 48MHz Flash @ 24MHz
  Decoder crunched 2.4 now **3.1** Mbit/s
- Benchmark 10k pings over two hours gives:
  1.54 % now **0.15** % packet loss 21.635/**22**.178/654.398/6.638 ms

# Tour around the Firmware

- Wireless Network Transceiver
  - samd21g18
    - hardware — register definitions provided by Atmel now Microchip
    - startup.c — all the hw errata and tricks plus HardFault printing
    - fiber.c .h — stackful coroutines (what Arduino should have done)
  - code_rfm12bp
    - do3rb.c .h — copy-pastable coded modulation
    - conduit.c — hw triggered dma ringbuffer for SPI transport
    - tinyusb.c — composite serial + ethernet modified with backpressure
  - tinyusb — unmodified copy of upstream tinyusb/src subdir
  - Makefile — who needs more

```
# pacman -S arm-none-eabi-{binutils,gcc,newlib} make
$ git clone github.com/DO3RB/WirelessNetworkTransceiver.git
```

# Thanks for your attention!

## Future Work

- porting to Si4463 and reaping the megabit
- soft-decision decoding using sidechannel info
  in theory correcting up to 12 biterrors

## Contact

- Off-Grid Messaging Assembly
- Austrian Assembly and Affiliates AAAAAAAAAA
- github.com/DO3RB/WirelessNetworkTransceiver
- do3rb@elektronenpumpe.de