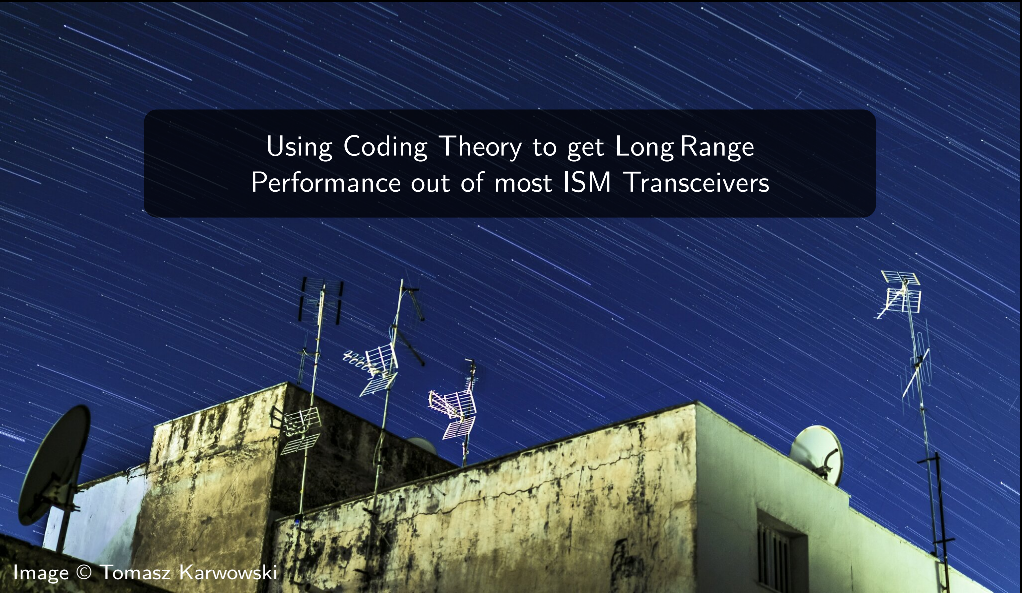
Reliable Radio Communications

37C3 UNLOCKED

Bernerd DO3RB



Using Coding Theory to get Long Range Performance out of most ISM Transceivers

Image © Tomasz Karwowski

Bernerd DO3RB

Reliable Radio Communications

37C3 UNLOCKED

Using Coding Theory to get Long Range Performance out of most ISM Transceivers

and every self-synchronised serial link

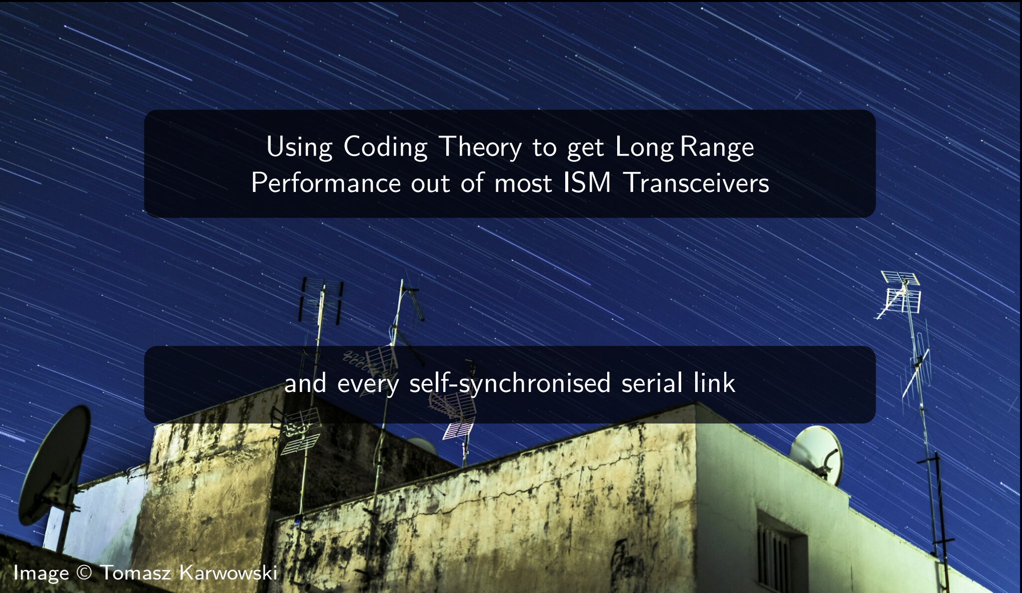
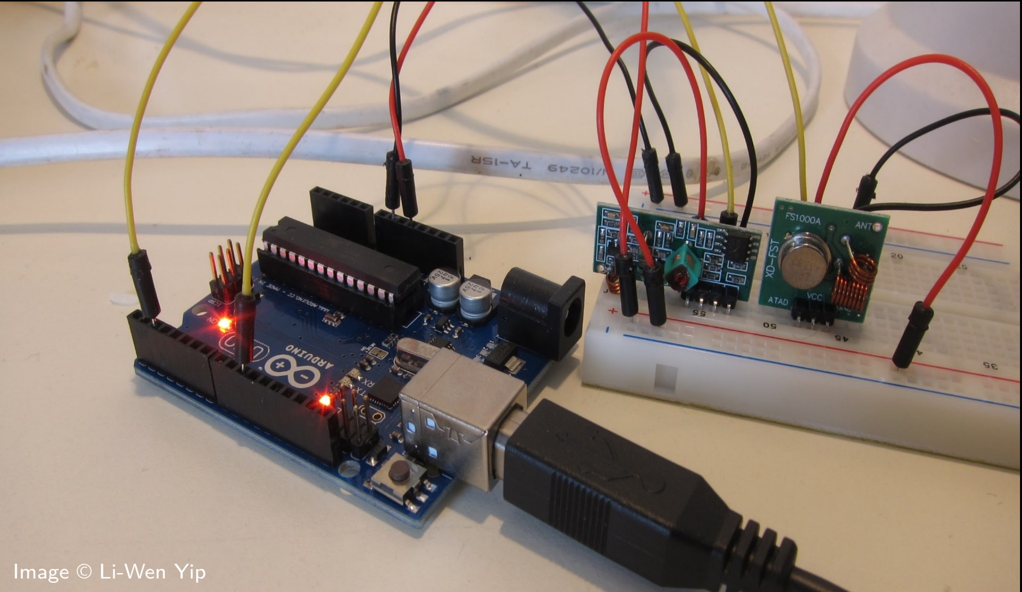


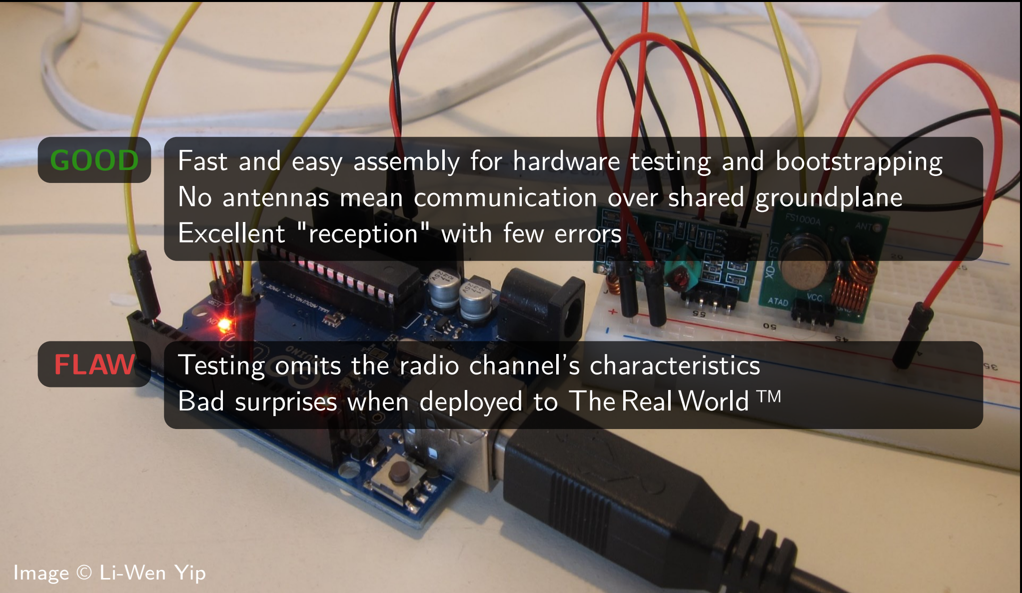
Image © Tomasz Karwowski

Bernerd DO3RB

Direct Coupling → Propagation along Wires ✗



Examples Direct Coupling ➔ Propagation along Wires ✗



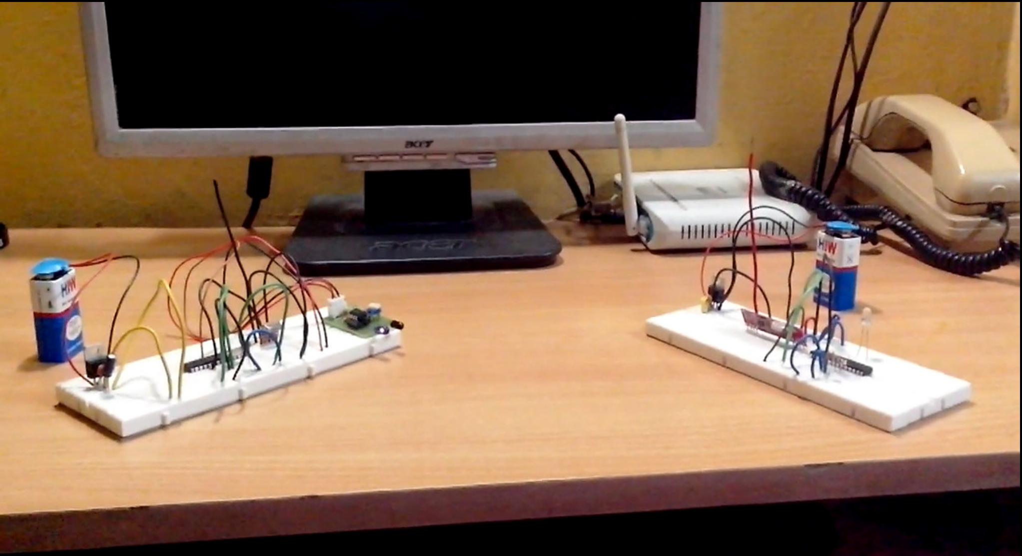
GOOD

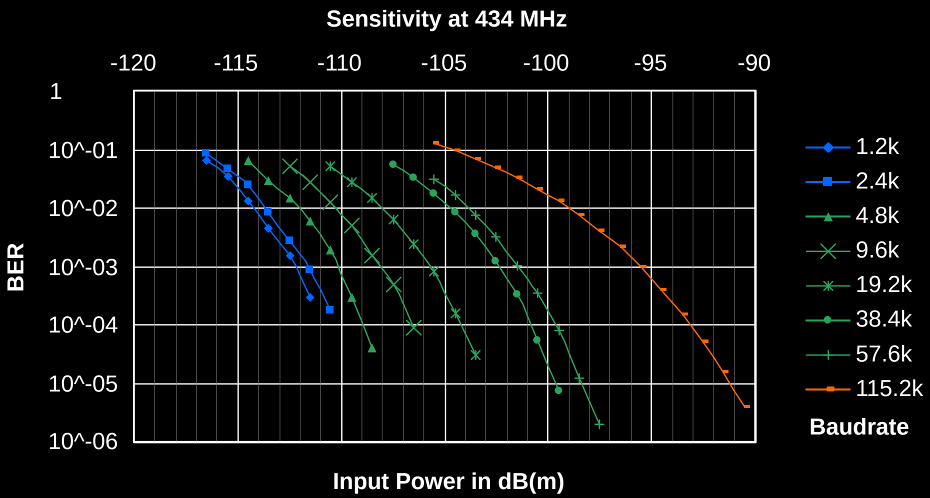
- Fast and easy assembly for hardware testing and bootstrapping
- No antennas mean communication over shared groundplane
- Excellent "reception" with few errors

FLAW

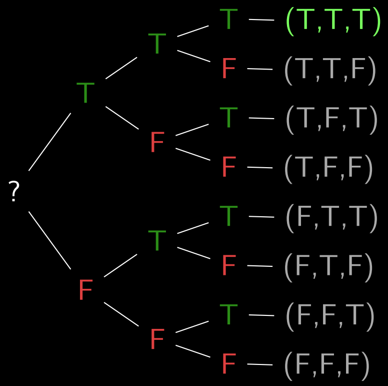
- Testing omits the radio channel's characteristics
- Bad surprises when deployed to The Real World ™

Loose Coupling ➔ Propagation through Space ✓




Sensitivity at 434 MHz
-120
-115
-110
-105
-100
-95
-90
1
10^-01
10^-02
10^-03
10^-04
10^-05
10^-06
BER
Input Power in dB(m)
1.2k
2.4k
4.8k
9.6k
19.2k
38.4k
57.6k
115.2k
Baudrate

Chained Probabilities



Probability that a stream of N bits is received correctly:

$$P_{\text{BITS}} = (1 - \text{BER})^{\text{NBITS}}$$

$$(1 - 0.1000)^{8 \cdot 64} = 3 \cdot 10^{-24}$$

$$(1 - 0.0100)^{8 \cdot 64} = 0.58\,\%$$

$$(1 - 0.0010)^{8 \cdot 64} = 59.91\,\%$$

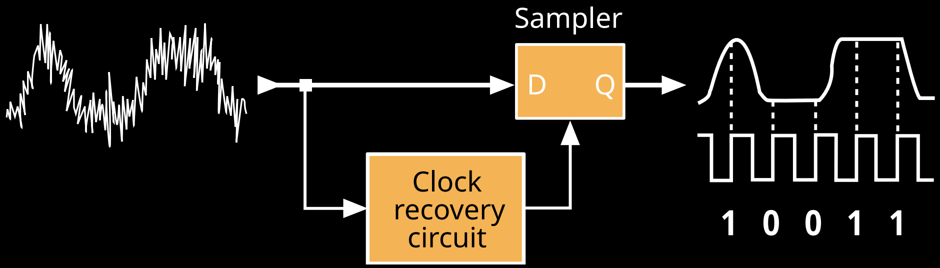
$$(1 - 0.0001)^{8 \cdot 64} = 95.01\,\%$$

One Generatormatrix of the Extended Golaycodes

$$
\text{Message}
\begin{pmatrix}
\circ \\ \bullet \\ \circ \\ \bullet \\ \circ \\ \bullet \\ \circ \\ \bullet \\ \circ \\ \bullet \\ \bullet \\ \circ
\end{pmatrix}
\otimes
\begin{pmatrix}
\bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \bullet & \bullet & \bullet & \bullet & \bullet & \circ & \circ & \circ & \bullet \\
\circ & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \bullet & \bullet & \bullet & \bullet & \bullet & \circ & \bullet & \circ \\
\circ & \circ & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \bullet & \bullet & \bullet & \bullet & \bullet & \circ & \bullet \\
\circ & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \bullet & \circ & \circ & \bullet & \bullet & \bullet & \bullet & \bullet & \circ \\
\circ & \circ & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \bullet & \circ & \circ & \bullet & \circ & \circ & \bullet & \bullet & \bullet & \circ & \bullet \\
\circ & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \bullet & \bullet & \circ & \circ & \bullet & \circ & \circ & \bullet & \bullet & \bullet & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & \circ & \bullet & \bullet & \bullet & \bullet & \circ & \circ & \bullet & \circ & \circ & \bullet & \circ & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & \bullet & \bullet & \bullet & \bullet & \bullet & \circ & \circ & \bullet & \circ & \circ & \bullet & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & \bullet & \bullet & \bullet & \bullet & \bullet & \circ & \circ & \bullet & \circ & \circ & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & \bullet & \bullet & \bullet & \bullet & \bullet & \circ & \circ & \bullet & \bullet & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \bullet & \circ & \bullet & \circ & \bullet & \circ & \bullet & \circ & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \bullet & \circ & \bullet & \circ & \bullet & \circ & \bullet & \circ & \bullet & \circ & \bullet & \bullet
\end{pmatrix}
\text{Generator-matrix}
$$

$$
= \begin{pmatrix}
\circ & \bullet & \circ & \bullet & \circ & \bullet & \circ & \bullet & \circ & \bullet & \bullet & \circ & \bullet & \circ & \bullet & \circ & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \circ
\end{pmatrix}
\text{Codeword}
$$

Reconstruction of Clock and Bitstream



Retrieving archived Hardcopies

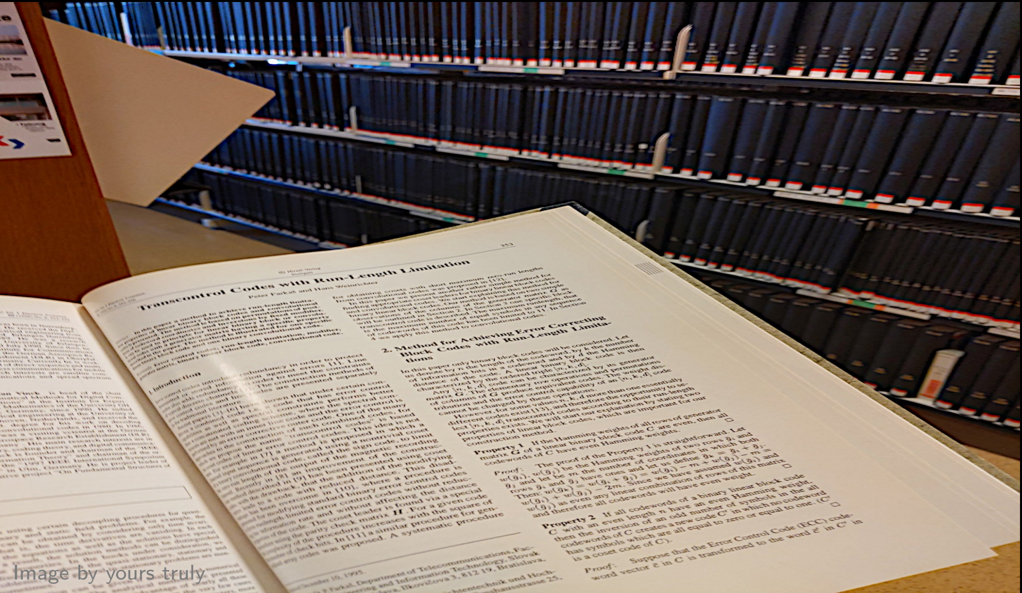


Image by yours truly

37C3 UNLOCKED

Constructive Lemma by Farkaš et al.

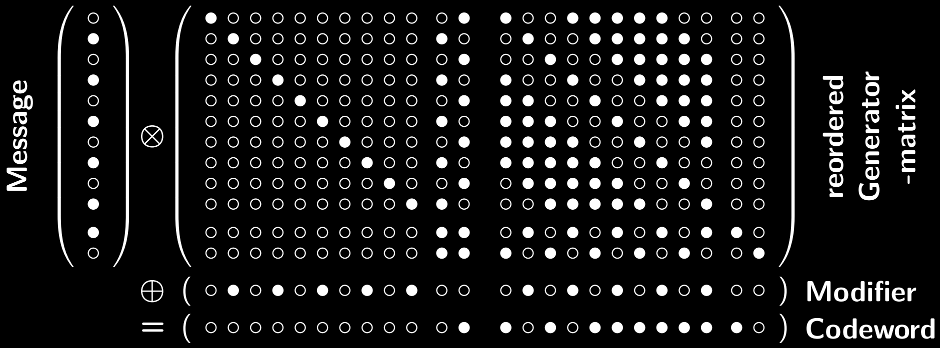
If all vectors in an even length matrix have even Hamming weight then the inversion of an odd number of symbols in each vector eliminates codewords being all-one or all-zero.

Construction of ECC RLL Codes Exploiting some Parity Matrix Properties, Farkašová K. Farkaš P.
Journal of Electrical Engineering, Volume 66, Number 3, **2015**, 182 – 184 ✓

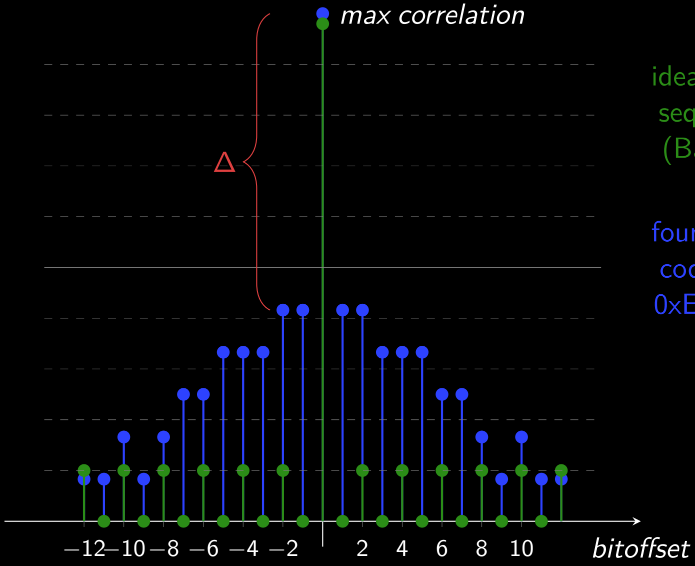
Transcontrol Codes with Run-Length Limitation, Farkaš P. Weinrichter H.
International Journal of Electronics and Communications (AEÜ), 50 (**1996**), 353 – 356 ✓

Extended Golay Code with Transcontrol Properties, Chomist R. Farkaš P.
International Conference on DSP and Multimedia Communications, 6th (**2005**), 102 – 105 ✗

Reordered Generatormatrix with Modifier



Frame Synchronisation Marker



ideal MLS sequence (Barker)

found FEC codeword 0xEB235E

Bernerd DO3RB

# of customised FEC RLL Encoder

```
uint_fast32_t do3rb_golay_encode(uint_fast16_t word)
{
    uint_fast32_t code = 0;
    #pragma GCC unroll 12 // branchless loop body
    for (uint_fast8_t n = 0; n < 12; n += 1) { // calc parity
        code ^= do3rb_golay_matrix[n] * (word&1); word >>= 1;
    } // concate parity | message
    code = (code << 12) | word; // shuffle two columns in halves
    code = (code&0xFFCFFC)|((code&0x3000)>>12)|((code&0x3)<<12);
    return code ^ do3rb_golay_modifier; // 0x554554
}

bool do3rb_synchron_detect(uint32_t dword) {
  return popcount(dword^0xEB235E) < 5; // tolerate four errors
}
```

# of customised FEC RLL Decoder

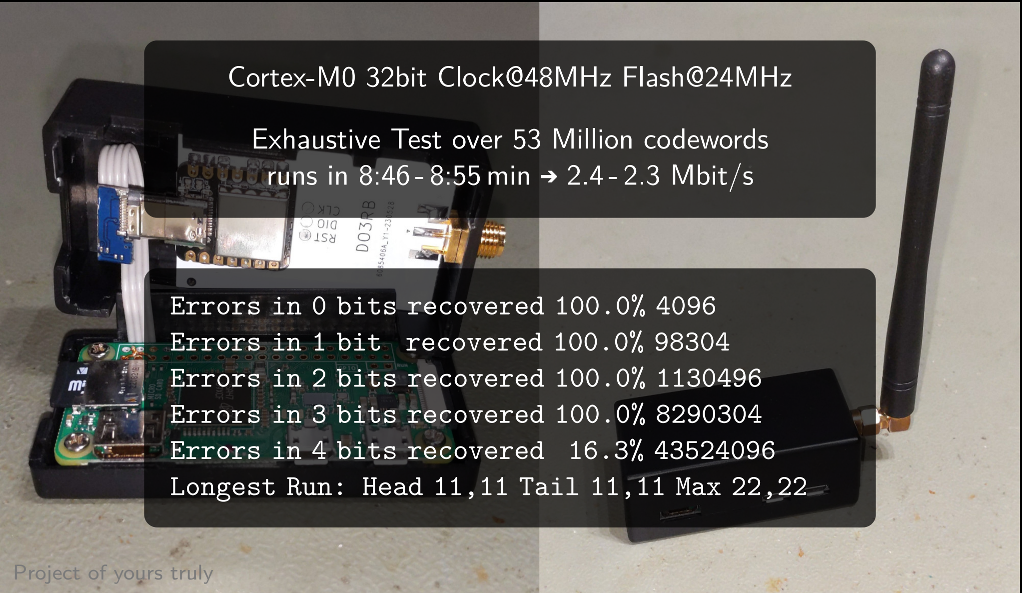

```
uint_fast16_t do3rb_golay_decode(uint_fast32_t code) {
  code ^= do3rb_golay_modifier; // undo modifier and shuffle
  uint_fast16_t P0 = ((code&0xFFC000)>>12) | (code&0x000003);
  uint_fast16_t D0 = ((code&0x003000)>>12) | (code&0x000FFC);
  uint_fast16_t P1 = do3rb_golay_parity(D0); // parity from data
  uint_fast16_t D1 = do3rb_golay_parity(P0); // data from parity
  if (P0 == P1) return D0; // no errors present
  if (popcount(P0^P1) < 4) return D0; // all in parity
  if (popcount(D0^D1) < 4) return D1; // all in data
  // one error in data, one or two errors in parity
  for (uint_fast8_t n = 0; n < 12; n++) {
    if (popcount(P0^P1^do3rb_golay_matrix[n]) < 4)
      return D0^(1<<n); }
  // one error in parity, one or two errors in data
  for (uint_fast8_t n = 0; n < 12; n++) {
    uint_fast16_t D2 = D1^do3rb_golay_matrix[n];
    if (popcount(D0^D2) < 4) return D2; }
  return D0; // four errors have 16% chance being correct
}            // more errors have none
```

Implementation using SAM D21

Project of yours truly

Implementation using SAM D21

37C3 UNLOCKED



Cortex-M0 32bit Clock@48MHz Flash@24MHz

Exhaustive Test over 53 Million codewords
runs in 8:46 - 8:55 min → 2.4 - 2.3 Mbit/s

```
Errors in 0 bits recovered 100.0% 4096
Errors in 1 bit  recovered 100.0% 98304
Errors in 2 bits recovered 100.0% 1130496
Errors in 3 bits recovered 100.0% 8290304
Errors in 4 bits recovered  16.3% 43524096
Longest Run: Head 11,11 Tail 11,11 Max 22,22
```

Thanks for your attention!

- Provide me with the missing Paper from 2005
- Teach me convolutional and soft-decision decoding

Contact

events.ccc.de/congress/2023/hub/en/user/bernerd
do3rb@elektronenpumpe.de

Project of yours truly