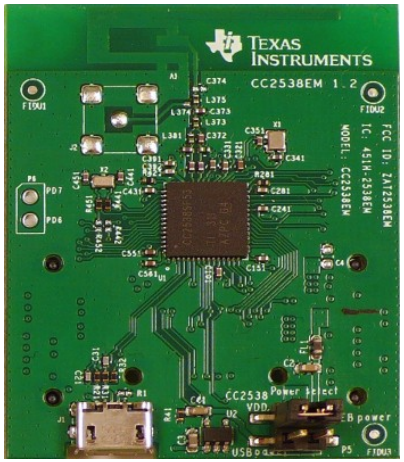
Cryptographic Randomness on a CC2538: A Case Study

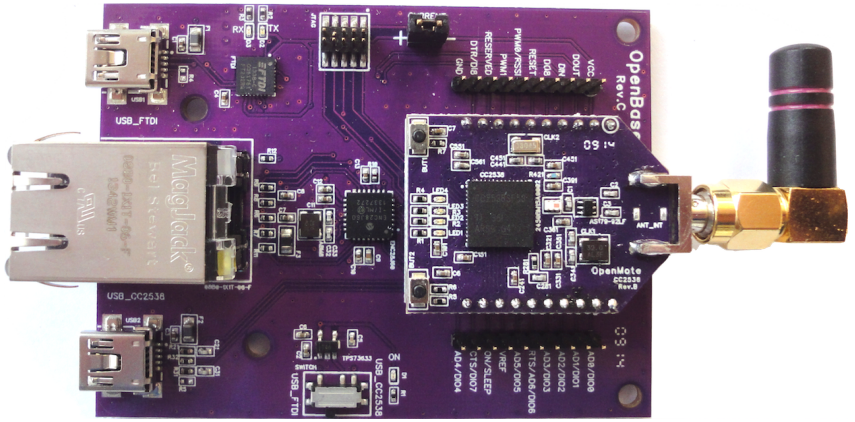
Yan YAN, Elisabeth OSWALD, Theo TRYFONAS

CC2538

(http://www.ti.com/product/CC2538)

- ARM® Cortex® -M3
- 512KB, 256KB or 128KB Programmable Flash
- 2.4-GHz IEEE 802.15.4 Compliant RF
- Security Hardware Acceleration
- Low Power
- Zigbee/6LoWPAN Support





Random Number Generators (RNGs)

- True RNG (TRNG)
  - Samples entropy in physical source
- Pseudo RNG (PRNG)
  - Seeded by TRNG
  - Efficient

- Cryptographic Usage:
  - Key Generation, etc

University of BRISTOL

Sad Stories...

SATURDAY, JANUARY 9, 2010

PRNG Vulnerability of Z-Stack ZigBee SEP ECC

by Travis Goodspeed <travis at radiantmachines.com>
with neighborly thanks to Nick DePetrillo,
concerning version 2.2.2-1.30 of TI Z-Stack
and a ZigBee Smart Energy Profile ECC vulnerability.

```
air% hexdump random.bin | grep --color "7c e1 e8 4e f4 87
0000000 02 01 00 60 e8 2e 7c e1 e8 4e f4 87 62 49 56 fe
0008000 01 00 60 e8 2e 7c e1 e8 4e f4 87 62 49 56 fe 80
0010000 00 60 e8 2e 7c e1 e8 4e f4 87 62 49 56 fe 80 00
0018000 60 e8 2e 7c e1 e8 4e f4 87 62 49 56 fe 80 00 60
air%
```

rdist

January 11, 2010

Smart meter crypto flaw worse than thought

Filed under: Crypto, Embedded, Hacking, Hardware, RFID, Security — Nate Lawson @ 1:08 pm

Travis Goodspeed has continued finding flaws in TI microcontrollers, branching out from the MS the random number generator. Why is this important? Because the MSP430 and ZigBee are fou

Travis describes two flaws: the PRNG is a 16-bit LFSR and it is not seeded with very much ent generator be used to create cryptographic keys. It's extremely scary to find such a poor unders off the power to your house.

University of BRISTOL

CC2538 PRNG

- Coding Issues in Contiki driver
- 16 bit LFSR (CRC16) as PRNG

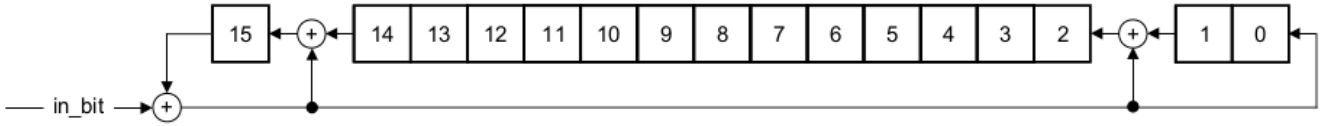


Figure 16-1. Basic Structure of the RNG

$2^{16} = 65536$

Far **NOT ENOUGH** for cryptographic security!

University of BRISTOL

e.g.: Broken ECDHE/ECDSA in DTLS
(tinydtls)

- Build Key Pair lookup table: 65536 entries

ECDHE

Require: A's secret r_A, B's secret r_B, Base Point G

1: A sends $Q_A = [r_A]G$
2: B sends $Q_B = [r_B]G$
3: A,B independently computes: $Q_{AB} = [r_A]Q_B = [r_B]Q_A$
4: **return** Shared Secret Q_{AB}

- Revert r_A from Q_A and r_B from Q_B.
- Compute Q_{AB}.

ECDSA

Require: Singer's secret key d, Message to be signed m, Base Point G

1: Select random k, computes: $kG = (r, y)$
2: Compute $e = SHA - 1(m)$
3: Compute $s = k^{-1}(e + dr)$
4: **return** (r, s) as signature of m

- Revert k from r.
- Extract d given (s,k,e,r).

University of BRISTOL

CC2538 TRNG: Sampling RF noise

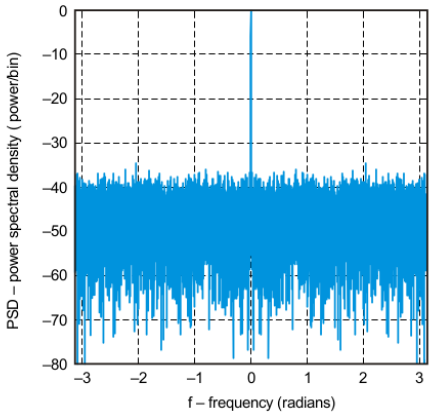


Figure 23-19. FFT of the Random Bytes

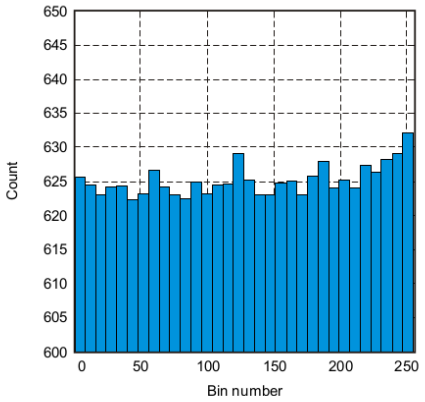


Figure 23-20. Histogram of 20 Million Bytes Generated With the RANDOM Instruction

- Passed NIST test tool(*), but…
- **Potentially tamperable**.

(*)http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html

University of BRISTOL

Hints of Circuit Design:

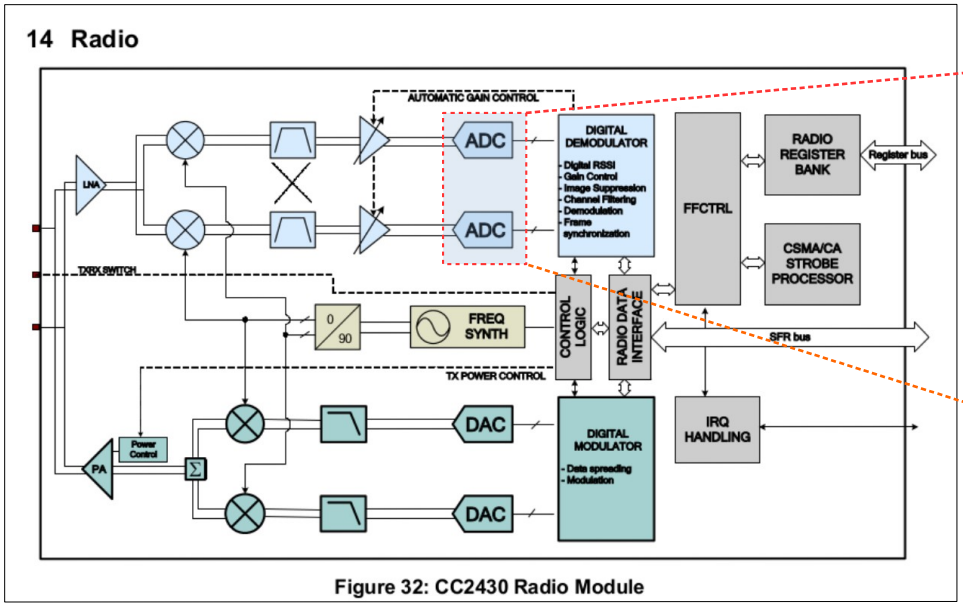


Figure 32: CC2430 Radio Module

CC2420 Manual

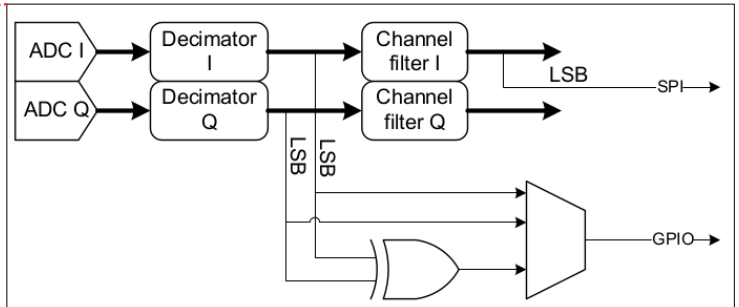
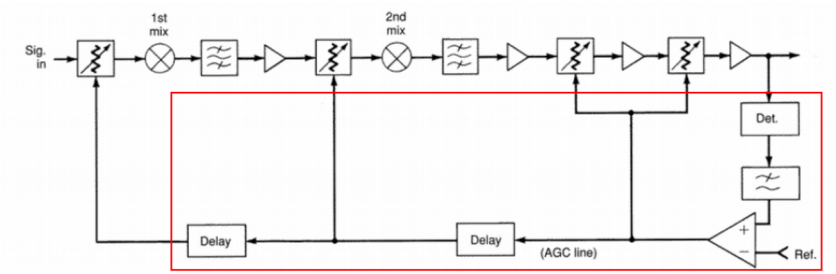


Figure 31: Random bit generation in the demodulator

CC2520 Manual

- Fixed ADC input – fixed ADC output (fixed LSB)
  - Use constant signal (i.e. carrier wave)
  - Challenge: Noises affect LSB

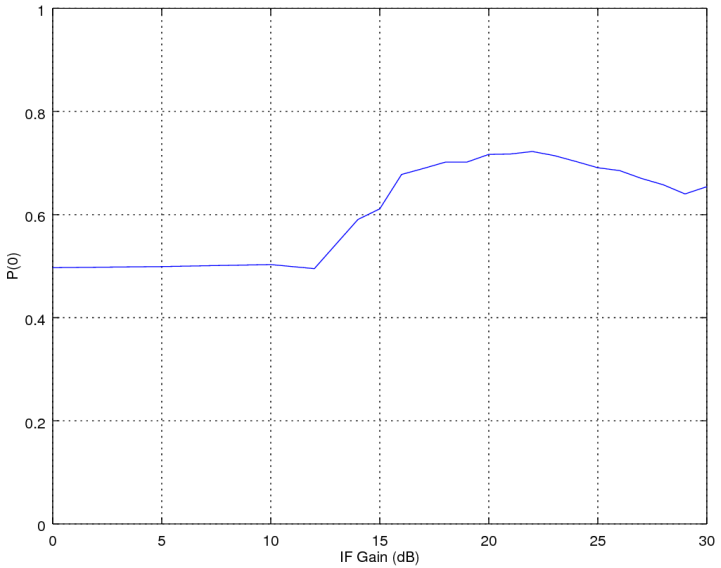


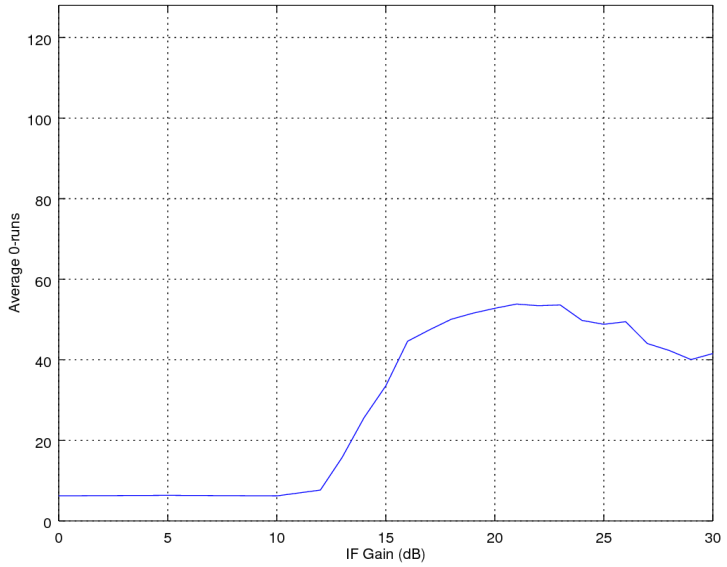
AGC Circuit

- Solution: Saturation (strong signal)
  - Noises became negligible.

University of BRISTOL

Result





Conclusion

- Problem: Really Not Good RNG:
  - Low entropy PRNG
    - Not recommended for any security usage.
  - Tamperable Entropy Source
    - Needs to be physically protected.
- Solution: Use Dedicated RNG:
  - Latest CC Series: CC26XX/CC13XX