Uitnodiging 13 februari

- **Nieuws van het Egel project V2**

File Edit View History Bookmarks Tools Help

Egel (Hedgehog) for launchp

https://home.hccn 110% Search

Egel noForth Forth gg F1 joeblogsf1 Translate DL Translate AliExpress ING Tindie Catawiki Home WO

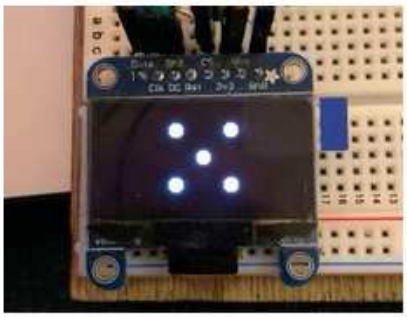
Four random generators

Generation of 16b random numbers with reduction to a range

Materials:
Just the Egel-kit or LaunchPad

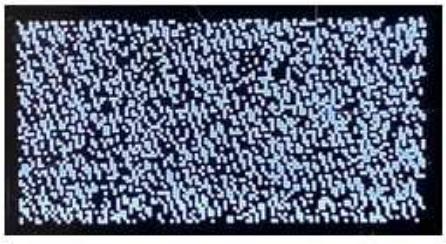
e70a noForth program

e70b noForth program

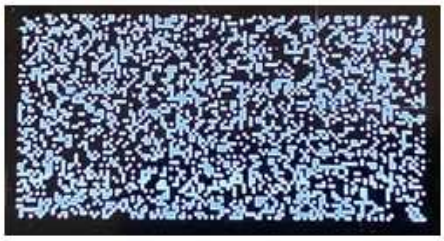


Generating a random number reliably is surprisingly difficult. In this chapter we demonstrate 4 different random number generators. Included are 3 generators well known in the Forth-world, and a modern routine. Two of the more useful routines are also coded in assembly, and comma-code for easy inclusion in your own code.

The random-generators:



The **Brodie**-generator was included in the legendary book Starting Forth by Leo Brodie. It is easy to implement and performs reasonably, but statistically two consecutive numbers are strongly related.

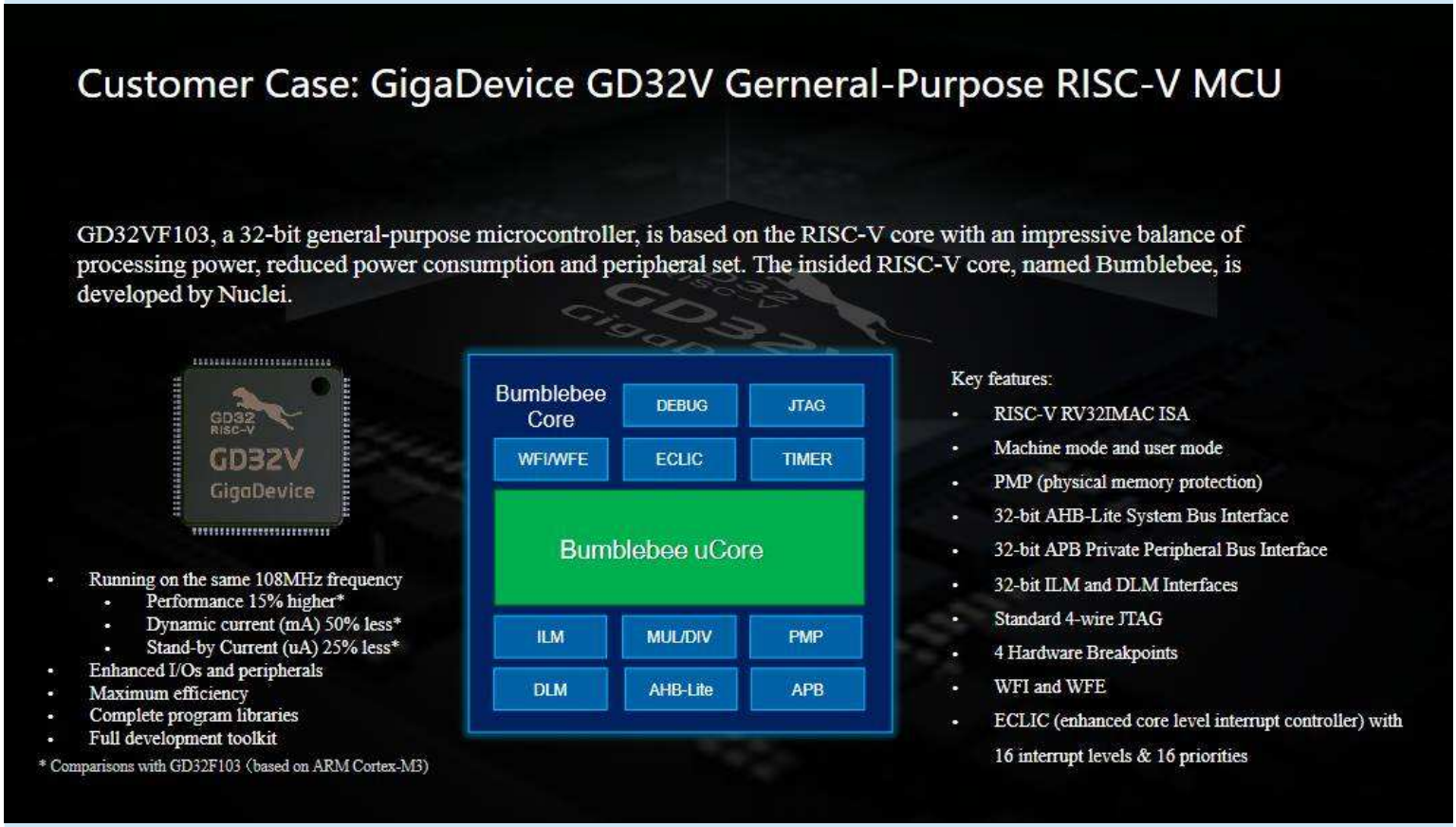


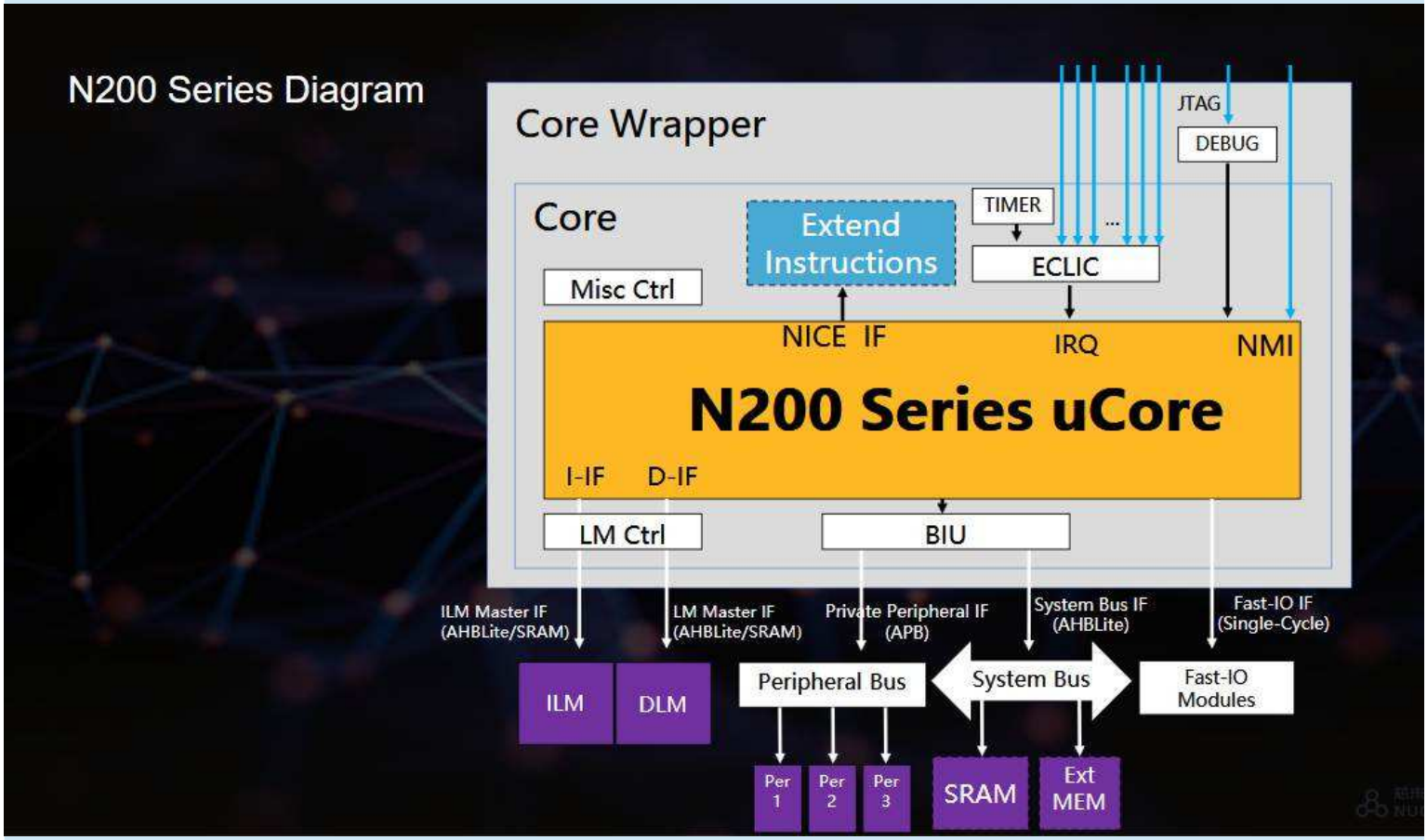
Donald **Knuth** solved the quality-problem of the Brodie-generator by using a 33 element table to shuffle the numbers generated by the Brodie-routine. The resulting generator produces random numbers with fair quality. The only disadvantages are the use of space for the table and the lower speed.

Voorbeeld uit hoofdstuk 70

◆ Risc-V stand van zaken?

Er zijn ondertussen enkele eenvoudige voorbeelden die draaien op **noForth R(C)V:**

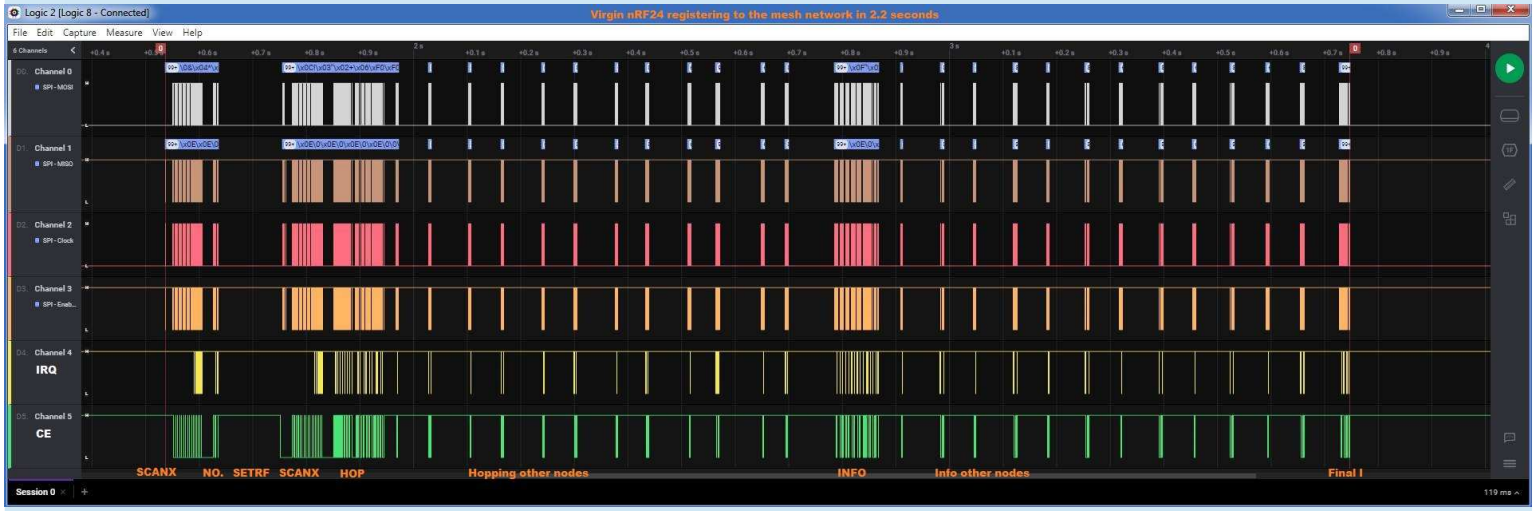




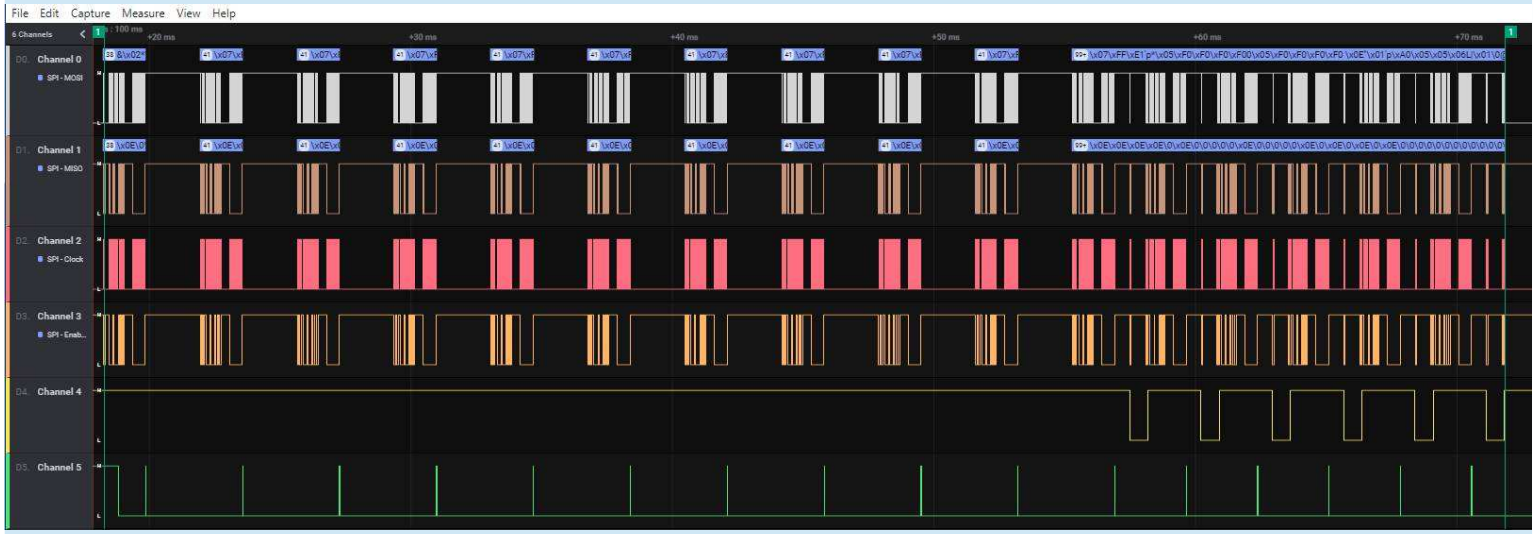
De GD32VF103 gebruikt de Nuclei N205 core

◆ Mesh netwerk met nRF24L01

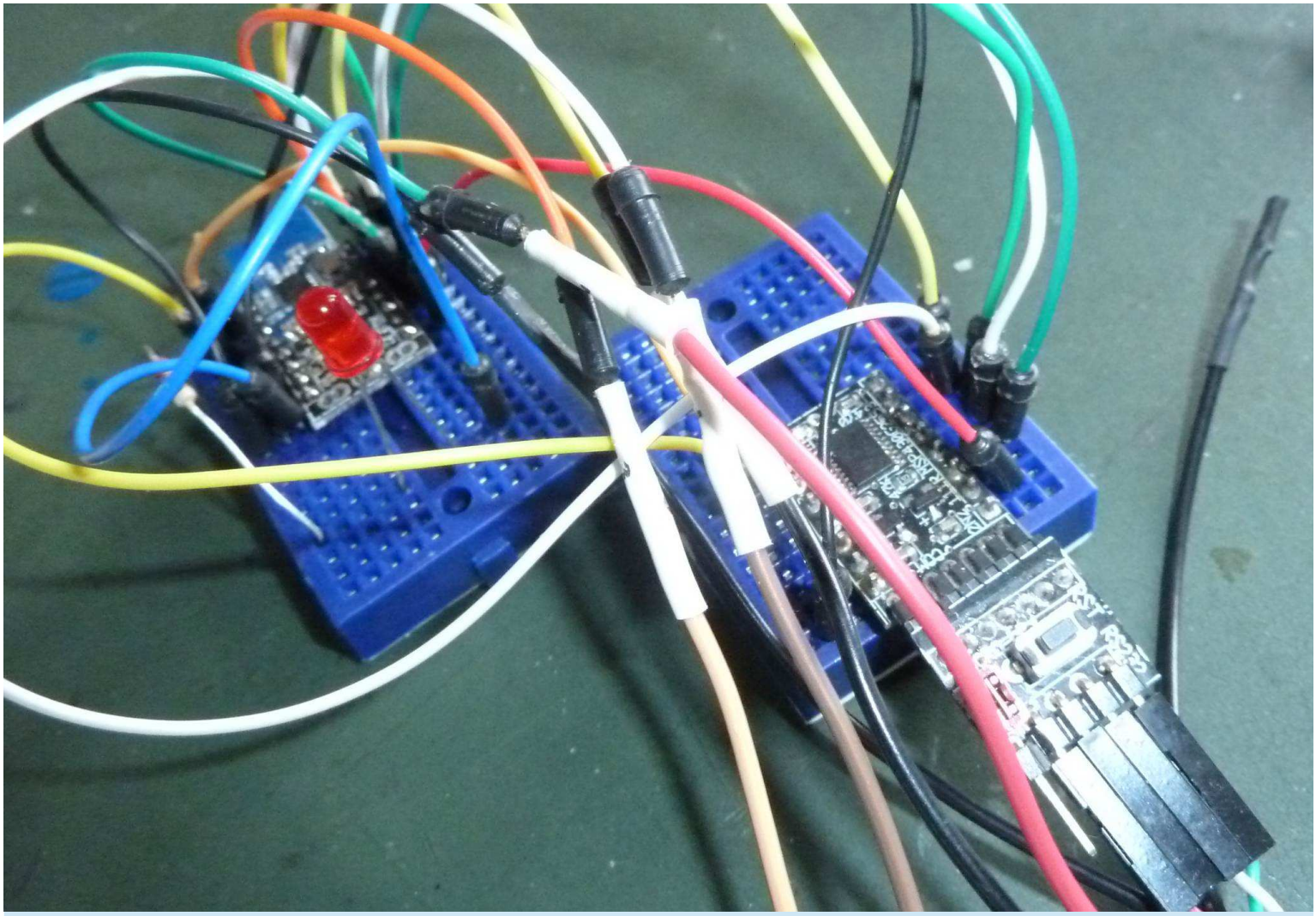
We zijn ondertussen aangekomen bij versie 3.9 van het netwerk. De code is verder opgeschoond en er zijn nieuwe opties toegevoegd.



Het registreren van een nieuwe node



SCANX binnen 70 millisec.



nRF24 node op MLP met bitbang SPI

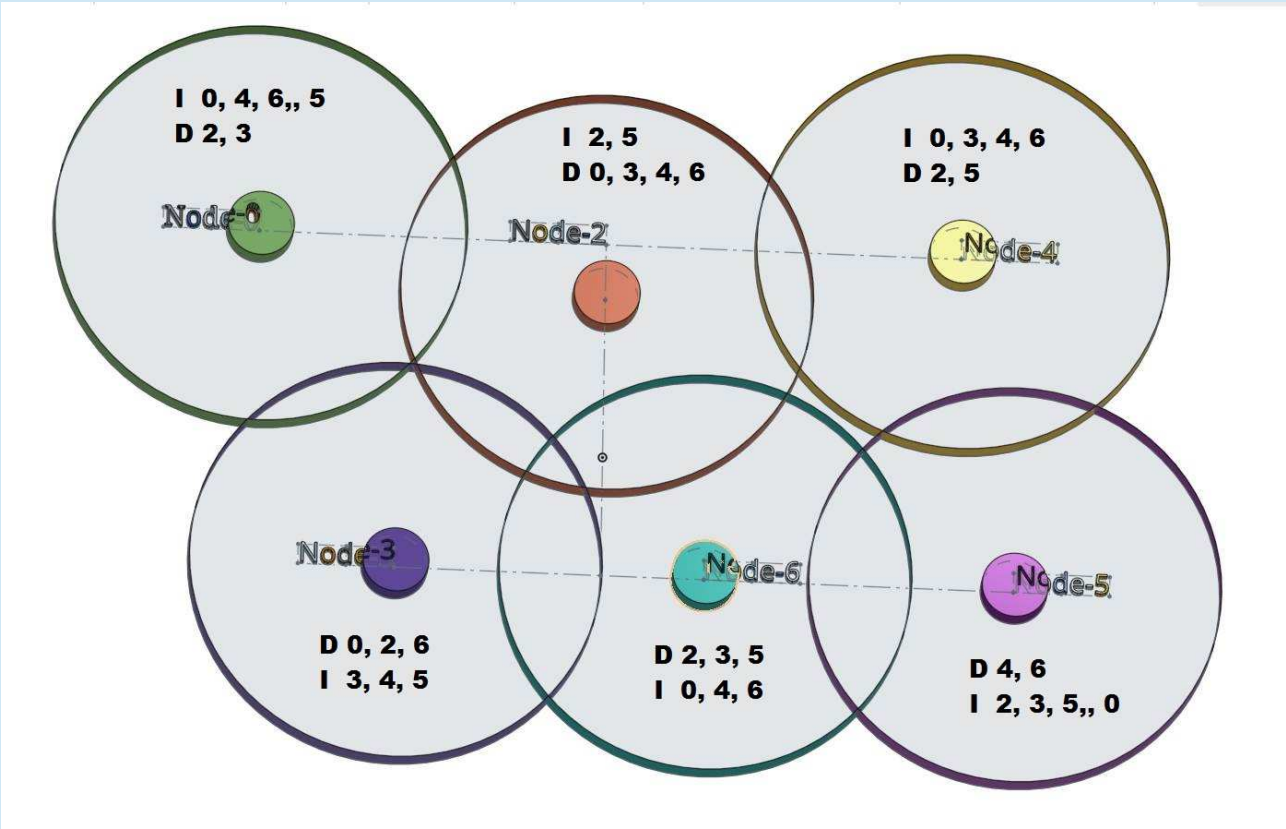
Druk nRF24L01+ status af

```
: .DB      ( +n -- ) 3 and  -3 + 6 * .  ." db" ;

: .BITRATE ( +n -- ) ?dup 0= if ." 250 kBit " exit then . ." Mbit, " ;

: .STATUS   ( -- )
    base @ >r  decimal
    ." Node " #me . ." nRF24 "    \ Which node with nRf24
    get-status ?dup if            \ nRF24 not connected?
        0E <> if  ." not "  then  \ nRF24 not ready?
        ." ok, " rf@ .bitrate     \ Show nRF24 RF settings
        .db  ." , Scan " pwr .db
        ." , RF channel = "  #ch .
    then  r> base ! ;
```

Voorbeeld:

```
@).status Node 0 nRF24 ok, 1 Mbit, 0 db, Scan -12 db, RF channel = 85  OK.0
@)
```

Vraag over de bereikbaarheid nodes



Om het netwerk zo betrouwbaar mogelijk te laten zijn is het handig als er voor een indirecte node een alternatieve route beschikbaar is. De basis routes worden vastgelegd, volgens het principe laagste nummertje eerst.

Van node-0 naar node-2 een alternatief pad maken is eenvoudig.
We vragen aan node-3 de bereikbare nodes op, en lezen daaruit dat hij node-2 rechtstreeks kan bereiken. Klaar er is één alternatief gevonden!

Van node-0 naar node-4 en/of node-6 geldt hetzelfde, dat is ook eenvoudig.

Nu is mijn vraag hoe los je dat voor b.v. node-0 en node-5 op?