

Decoding RDS signal in German VHF radio

April 2012
Martin Kielhorn

Outline

- capture signal in 104MHz band
- FM demodulation (audio is understandable)
- BPSK demodulation (too much noise)

A/D converter

cheap USB device ([RTL2832](http://www.terryseaton.com/rtl2832/), originally to listen to digital radio, 17 EUR)

<http://sdr.osmocom.org/trac/wiki/rtl-sdr>

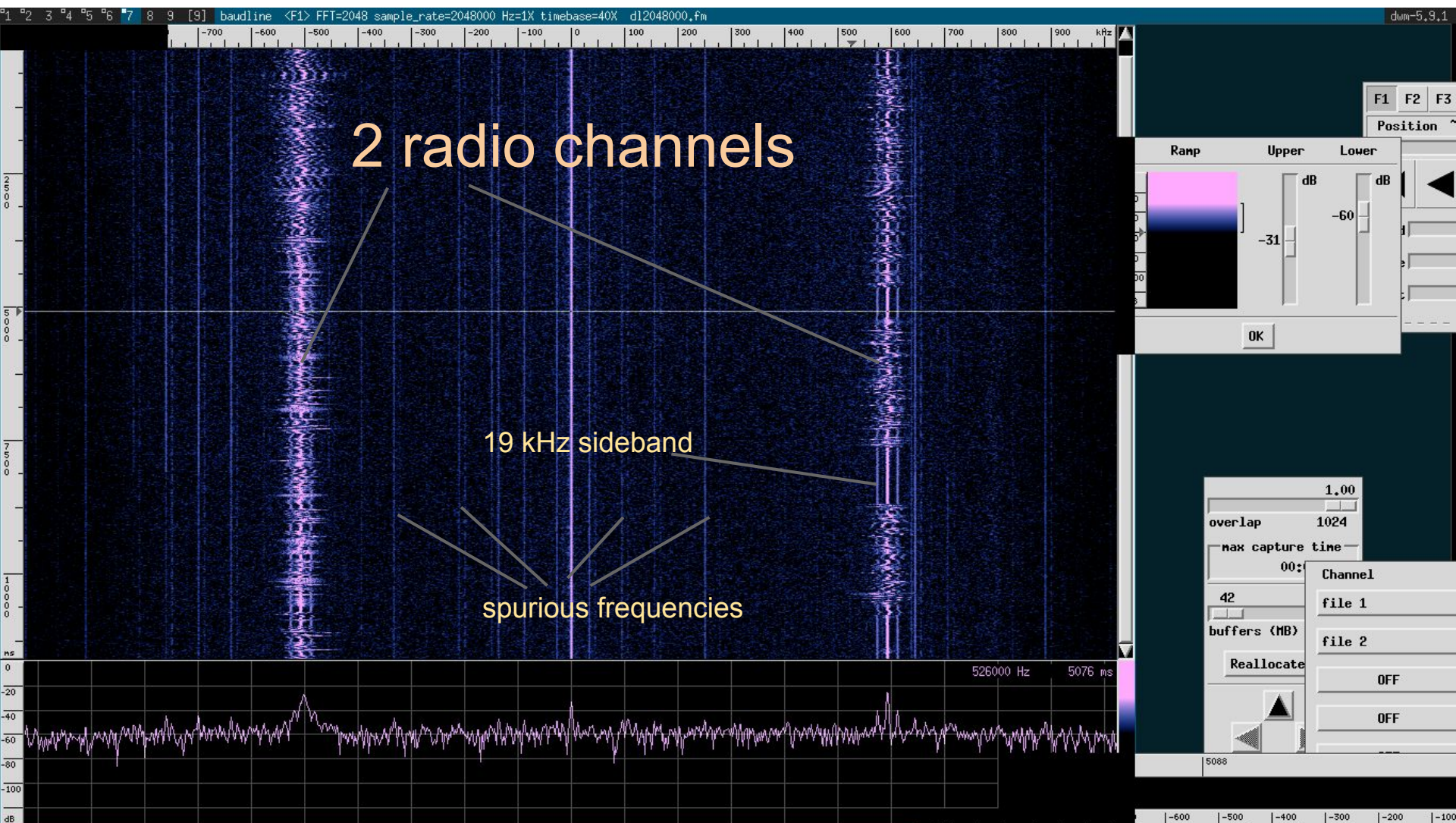
was reverse engineered

now there is code for Linux to read out with up to 3.2MS/s

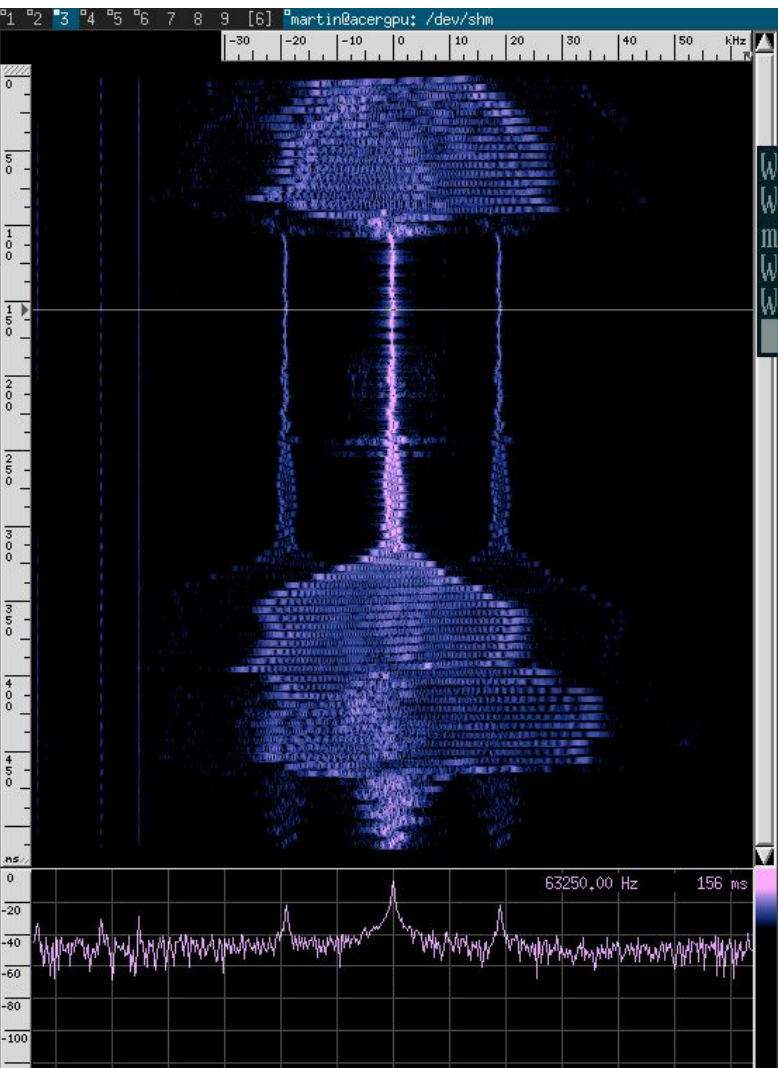
```
git clone git://git.osmocom.org/rtl-sdr.git
```

It seems to be 8 bit only, though.

10 seconds with 2048kS/s



Shift signal to base band



multiply each sample with complex exponential of center frequency:

```
(exp (complex 0d0
              (* (/ (* np2 -507250)
                    *rate*)
                 (/ (* 2d0 pi) *n-complex*)
                 i))))
```

then do FM demodulation using heterodyne division

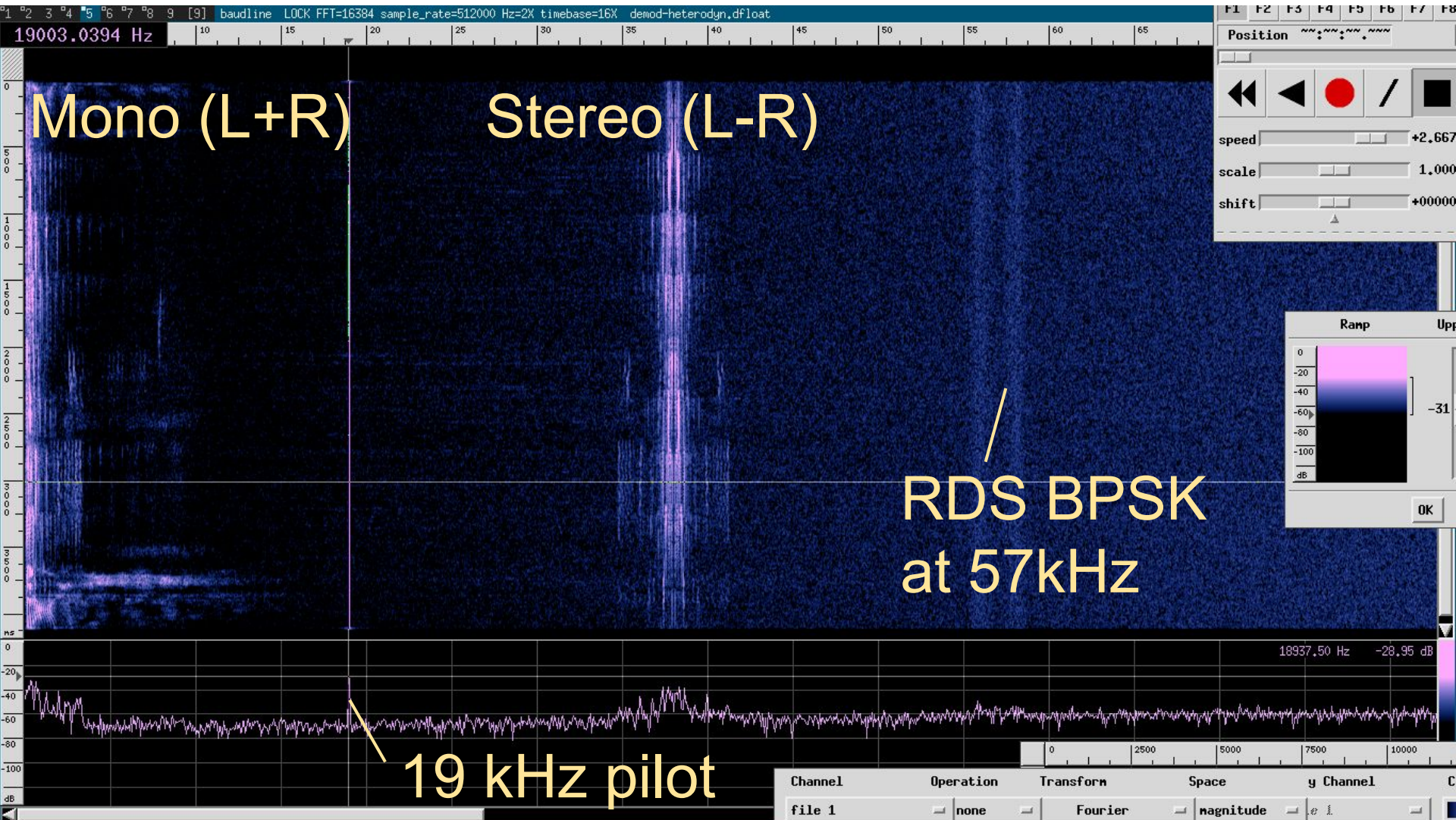
the modulated signal is: $s = A e^{ip}$

its time derivative is: $ds/dt = d/dt A e^{ip} = A ip' e^{ip}$

if you divide the time derivative by the signal you get a term with the phase derivative p' (the demodulated signal): $(ds/dt) / s = ip'$

$p' = \text{Im}[ds/dt / s]$

Radio signal after FM demodulation



How to get the RDS data?

http://de.wikipedia.org/wiki/Radio_Data_System

use 19kHz pilot to generate 57kHz reference

compare phase with BPSK signal

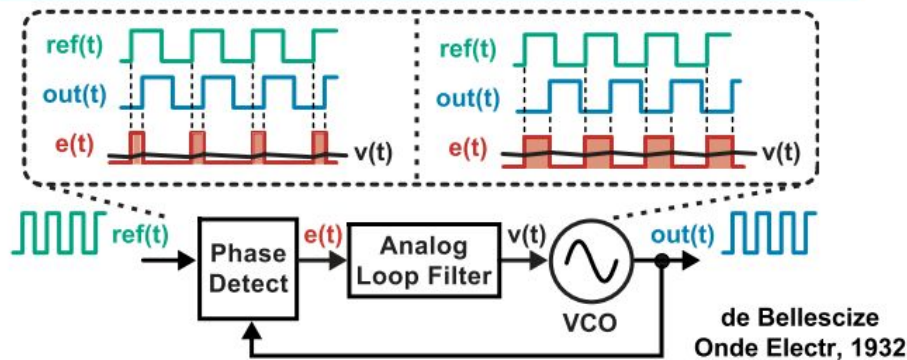
symbols are arriving with a rate of $57\text{kHz}/48$

Phase locked loop

The pilot is not exactly 19kHz during the whole 10 seconds.

We need to track the frequency very precisely.

What is a Phase-Locked Loop (PLL)?



- VCO efficiently provides oscillating waveform with variable frequency
- PLL synchronizes VCO frequency to input reference frequency through feedback
 - Key block is phase detector
 - Realized as *digital gates* that create pulsed signals

M.H. Perrott

4

Details | Notes

These details need reviewing. You can mark them as correct, or search by title on Google Scholar.

Details are Correct

Search by Title

Type: Journal Article

Tutorial on Digital Phase-Locked Loops

Authors: M. Perrott

View research catalog entry for this paper

Journal:

Year: 2009

Volume:

Issue: September

Pages:

Abstract:

Tags:

Keywords:

URL:

Add URL...

Catalog IDs

ArXiv ID:

Multiply frequency using a PLL

1 2 3 4 5 6 7 8 9 [2] Mendeley Desktop dwn-5.9.1

File Edit View Go Tools Help

My Library Tutorial on Digital Phase-Loc... x

Type here to search

Details Notes

These details need reviewing. You can mark them as correct, or search by title on Google Scholar.

Details are Correct Search by Title

Type: Journal Article

Tutorial on Digital Phase-Locked Loops

Authors: M. Perrott

View research catalog entry for this paper

Journal:

Year: 2009

Volume:

Issue: September

Pages:

Abstract:

Tags:

Keywords:

URL:
Add URL...

Catalog IDs
ArXiv ID:

Realized as *digital gates* that create pulsed signals

M.H. Perrott

4

Integer-N Frequency Synthesizers

$F_{out} = N \cdot F_{ref}$

Sepe and Johnston
US Patent (1968)

- Use digital counter structure to divide VCO frequency
 - Constraint: must divide by integer values
- Use PLL to synchronize reference and divider output

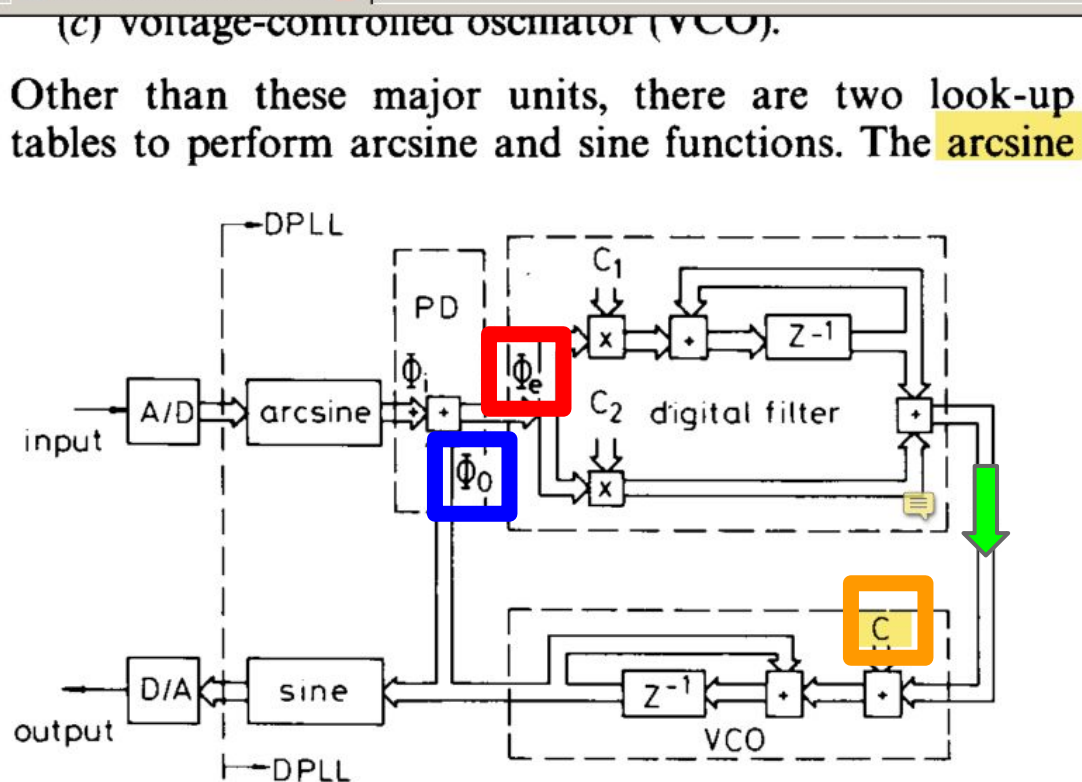
Output frequency is digitally controlled

M.H. Perrott

5

Page 5 of 118

How to implement a PLL using a computer



Details Notes

These details need reviewing. You can mark them as correct, or search by title on Google Scholar.

Details are Correct

Search by Title

Type: Generic

All digital phase-locked loop: concepts, design and applications

Authors: Y. Shayan, T. Le-Ngoc

[View research catalog entry for this paper](#)

Year: 1989

Pages:

Abstract:

Tags:

Keywords:

City:

Publisher:

Type of Work:

Actual implementation of a PLL

```
(defun dpll3 (z)
  (let* ((phi_i (phase z))                ;; PD
        (phi_e (- phi_i old-phi)))

    (if (< phi_e (* .9 -2d0 pi))           ;; phase unwrapping
        (incf phi_e (* 2 pi))
        (if (< (* .9 2d0 pi) phi_e)
            (decf phi_e (* 2 pi)))))

    (progn                                ;; digital filter
      (let* ((top (+ (* c1 phi_e) old-filt))
             (bottom (* c2 phi_e))
             (filt-out (+ top bottom)))
        (setf old-filt top)
        (let* ((znew ;; VCO
                (exp (complex 0 (+ (* divider c filt-out old-phi))))
                (znew3 (exp (complex 0
                               (+ c filt-out old-phi3))))))
          (setf old-phi (phase znew)
                old-phi3-cont (+ c filt-out old-phi3-cont)
                old-phi3 (phase znew3)))

        (values
         old-phi3-cont
         filt-out
         znew
         znew3))))))
```

allows retiming (by interpolation)

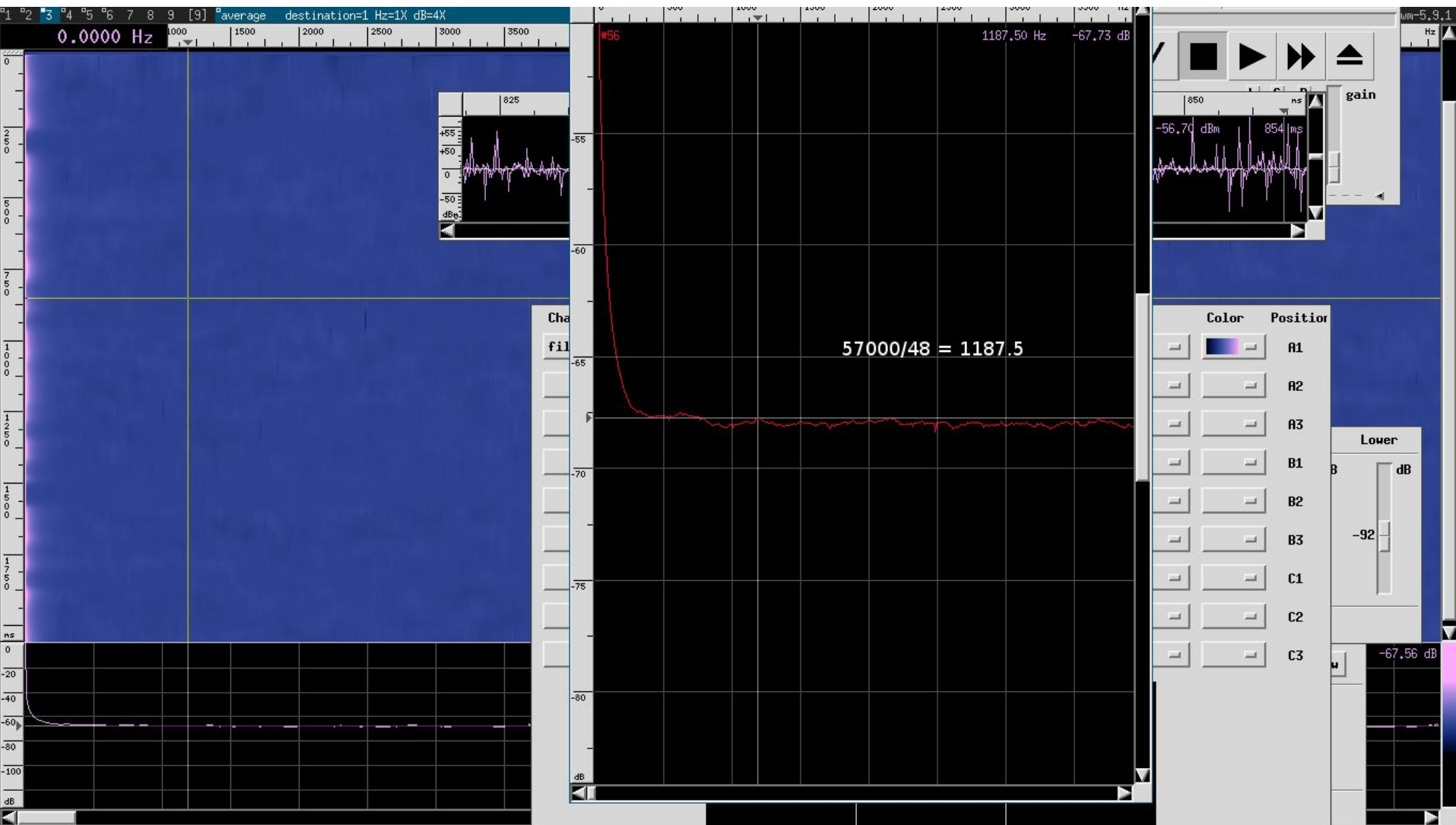
frequency changes

fits exactly on 19kHz signal

corresponding 57kHz signal

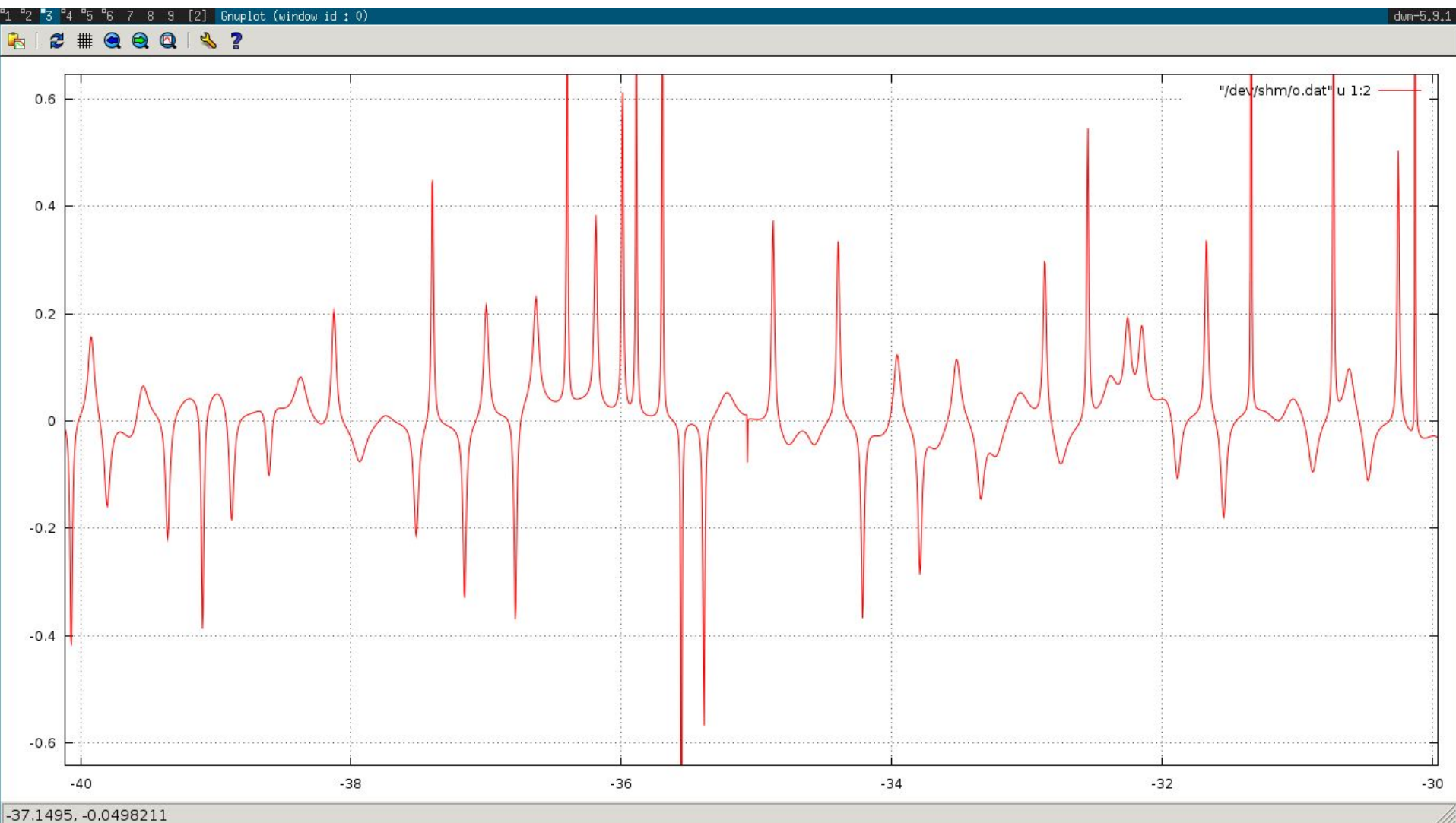
Results

autocorrelation shows only minute peak at symbol rate (data is too noisy)



Results

time derivative of phase changes between pilot and BPSK signal is shown
the x-axis shows symbols times (ideally phase changes should change for
nearly every integer x coordinate)



Discussion

- maybe a different FM modulation might help (right now I don't correct frequency drift of the FM carrier, actually the dongle should track this)
- perhaps searching for a better signal with the antenna will help
- maybe 8bit is just not enough to do FM modulation followed by BPSK
- how about [Pager](#), there data should be easier to receive

References

<https://www.cgran.org/wiki/RDS>

Acknowledgements

checkov for the hardware

some others for discussion