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, originally to listen to digital radio, 17 EUR)

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

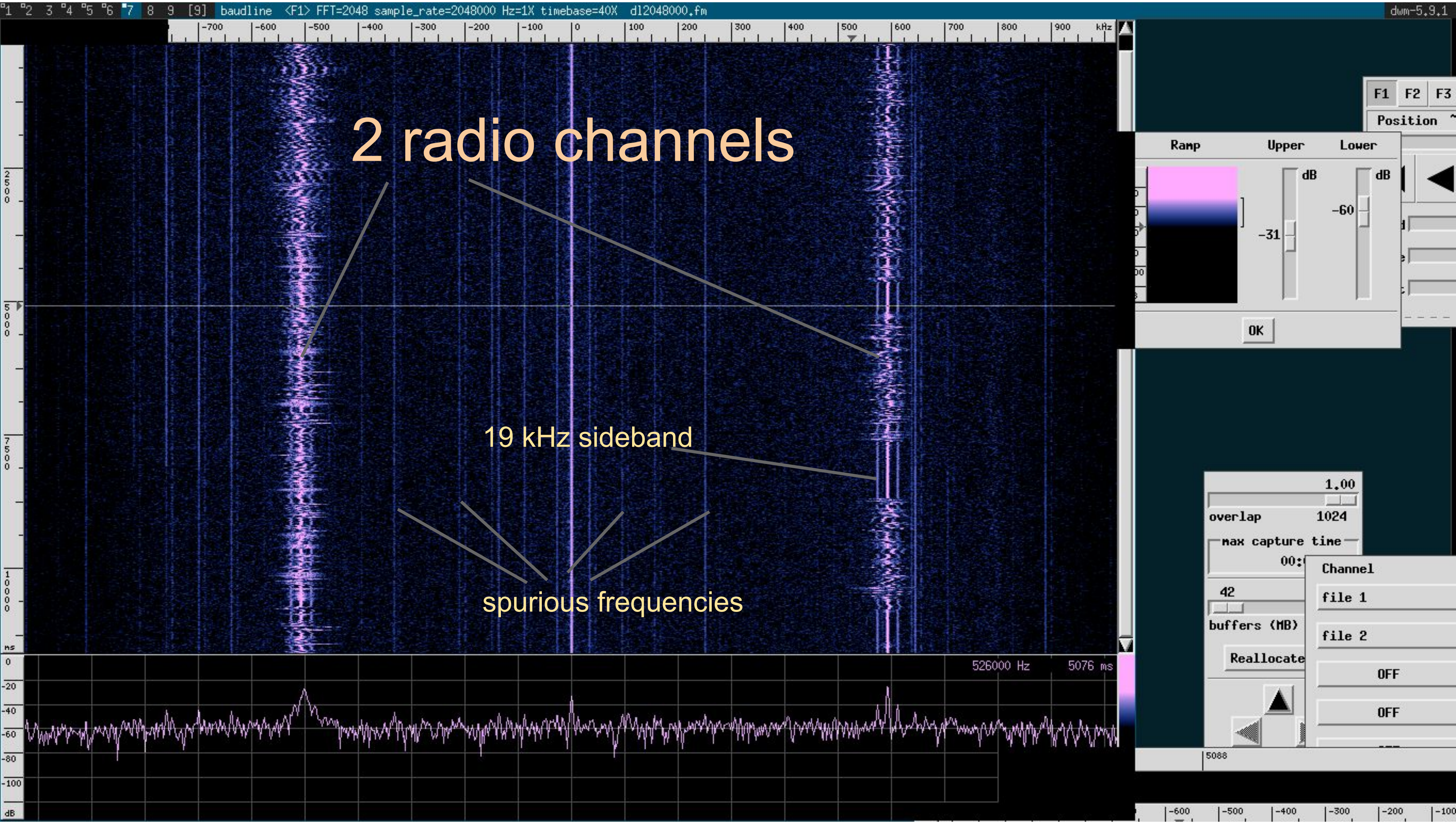
was reverse engineered

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

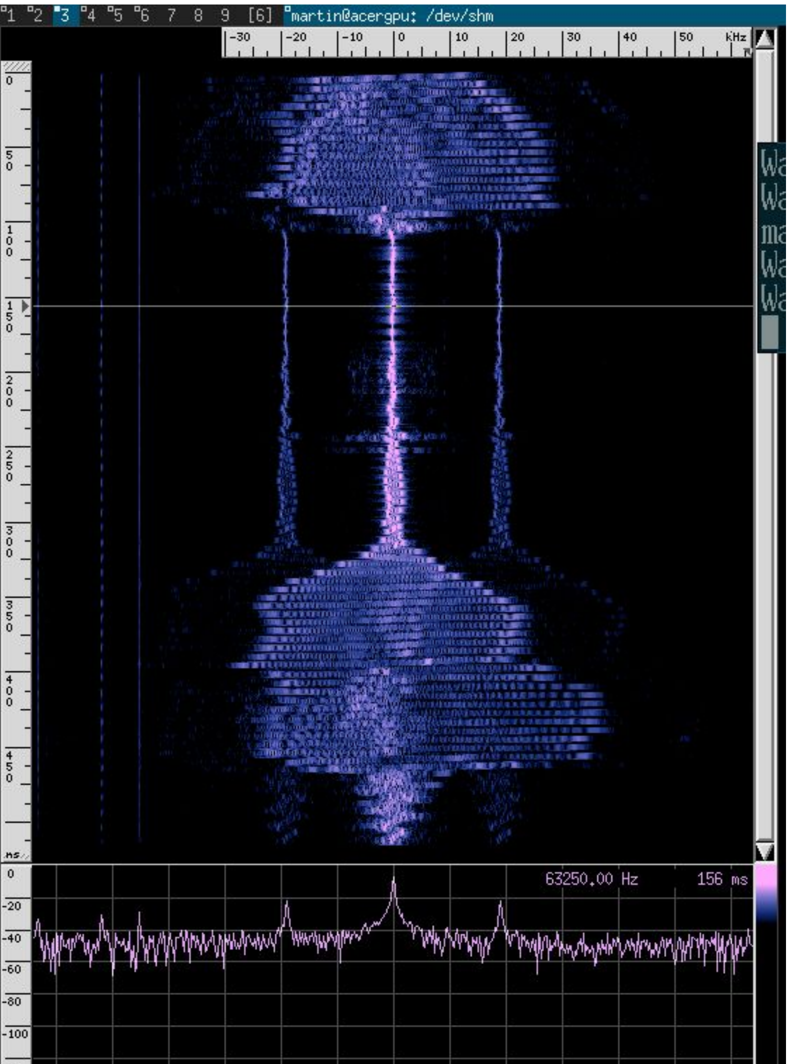
git clone git://git.osmocom.org/rtl-sdr.git

It seems to be 8 bit only, though.

10 seconds with 2048kS/s



Shift signal to base band



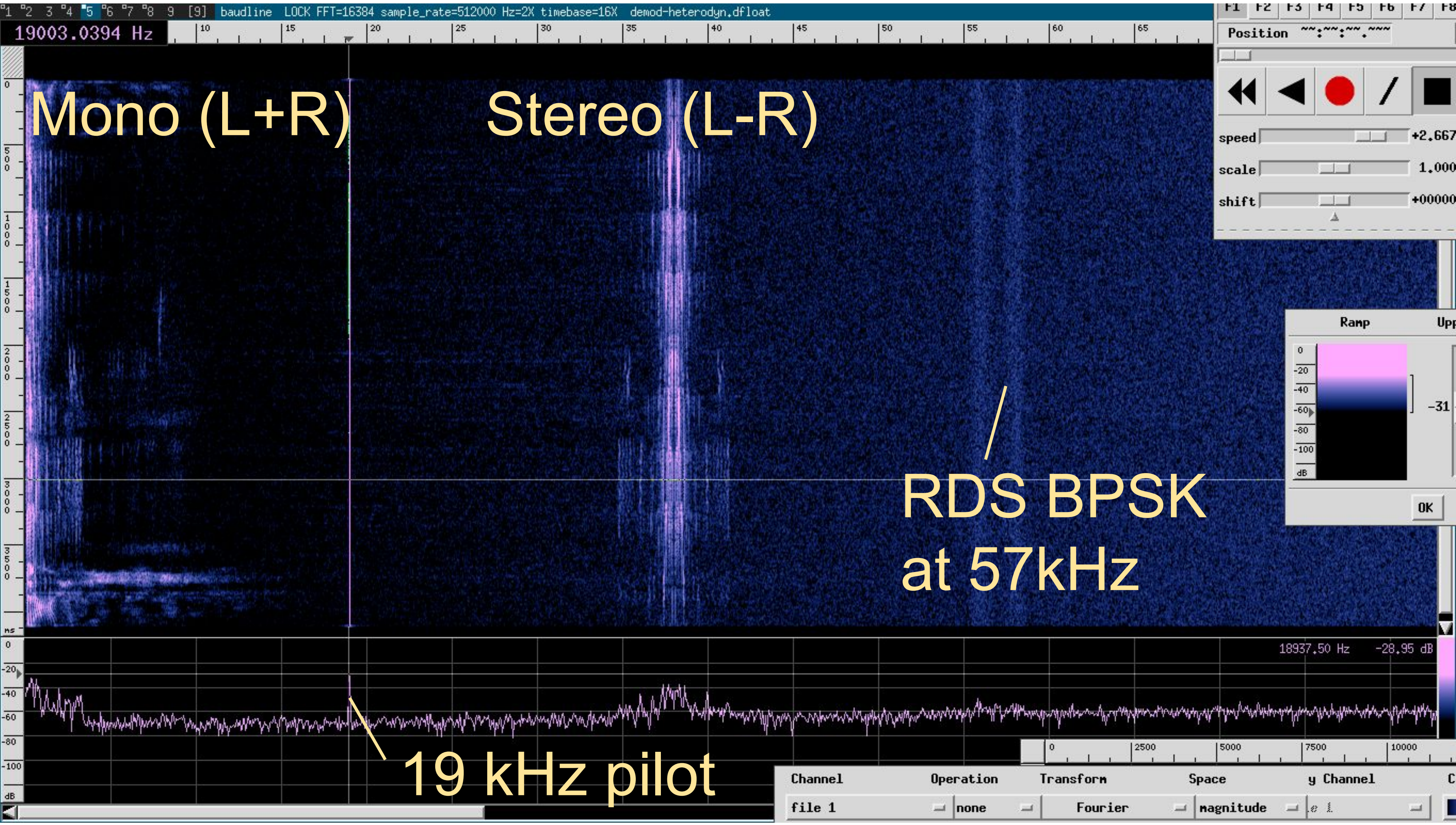
multiply each sample with complex exponential of center frequency:

```
(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' = 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

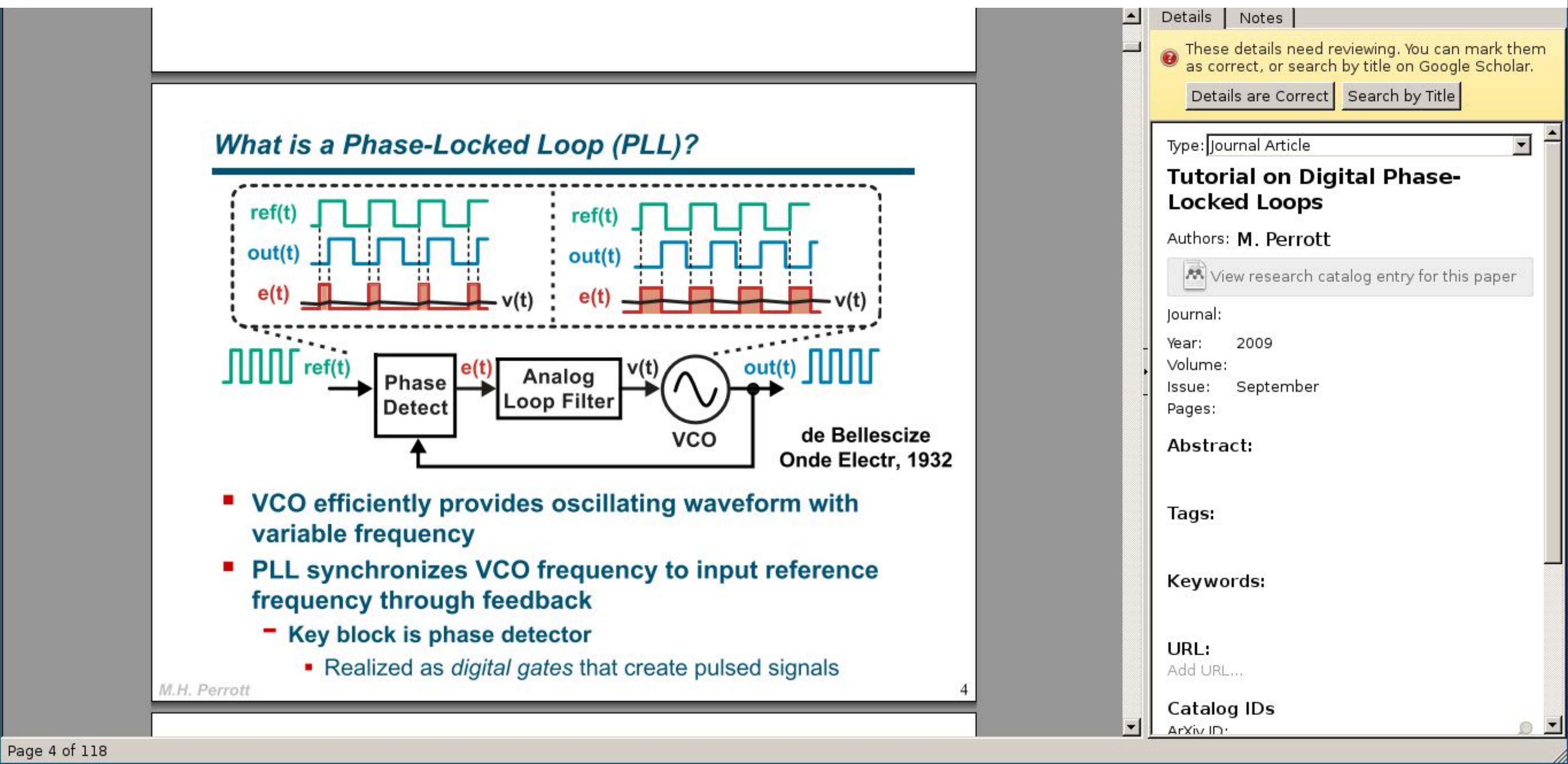
compare phase with BPSK signal

symbols are arriving with a rate of 57kHz/48

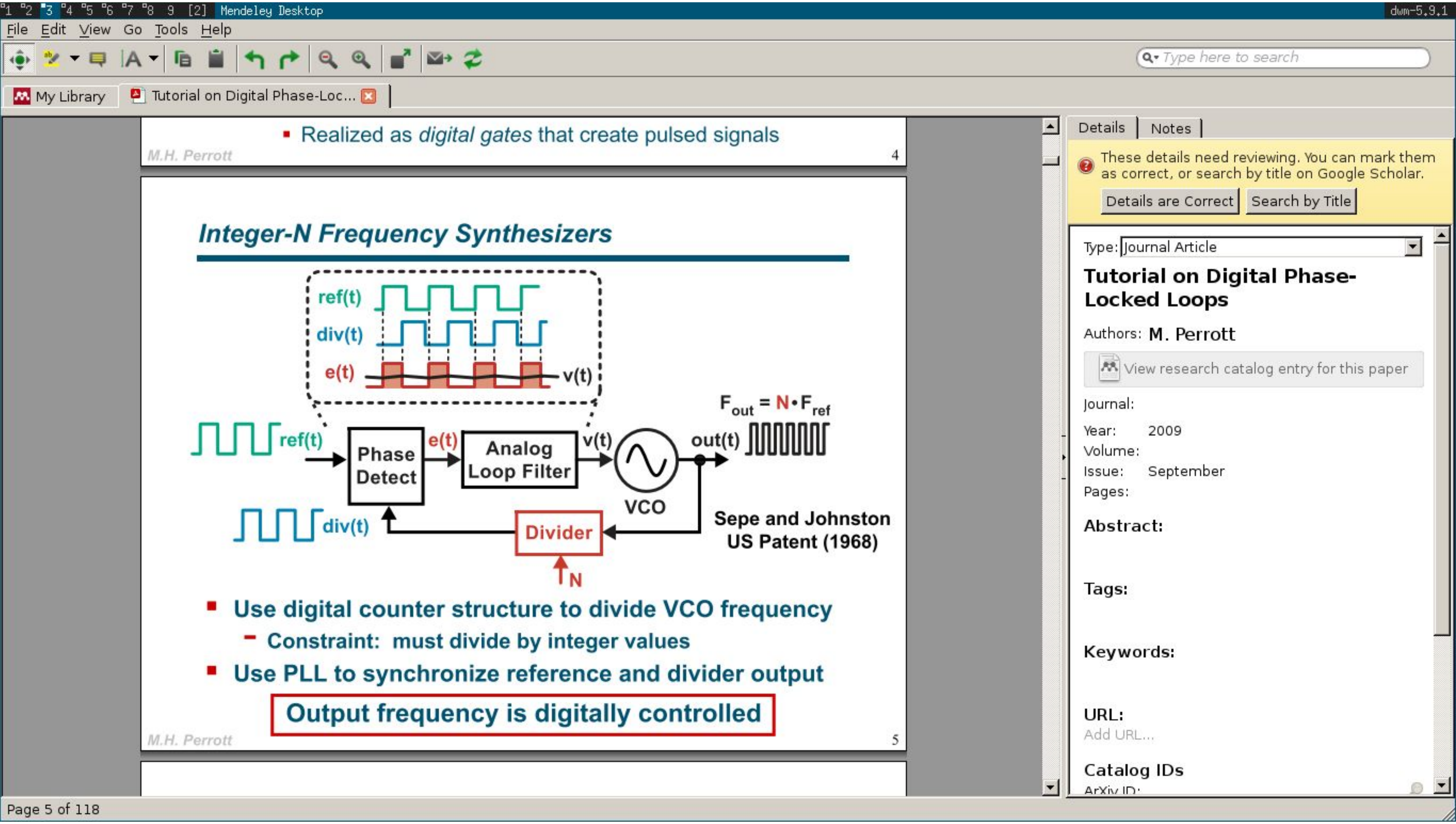
Phase locked loop

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

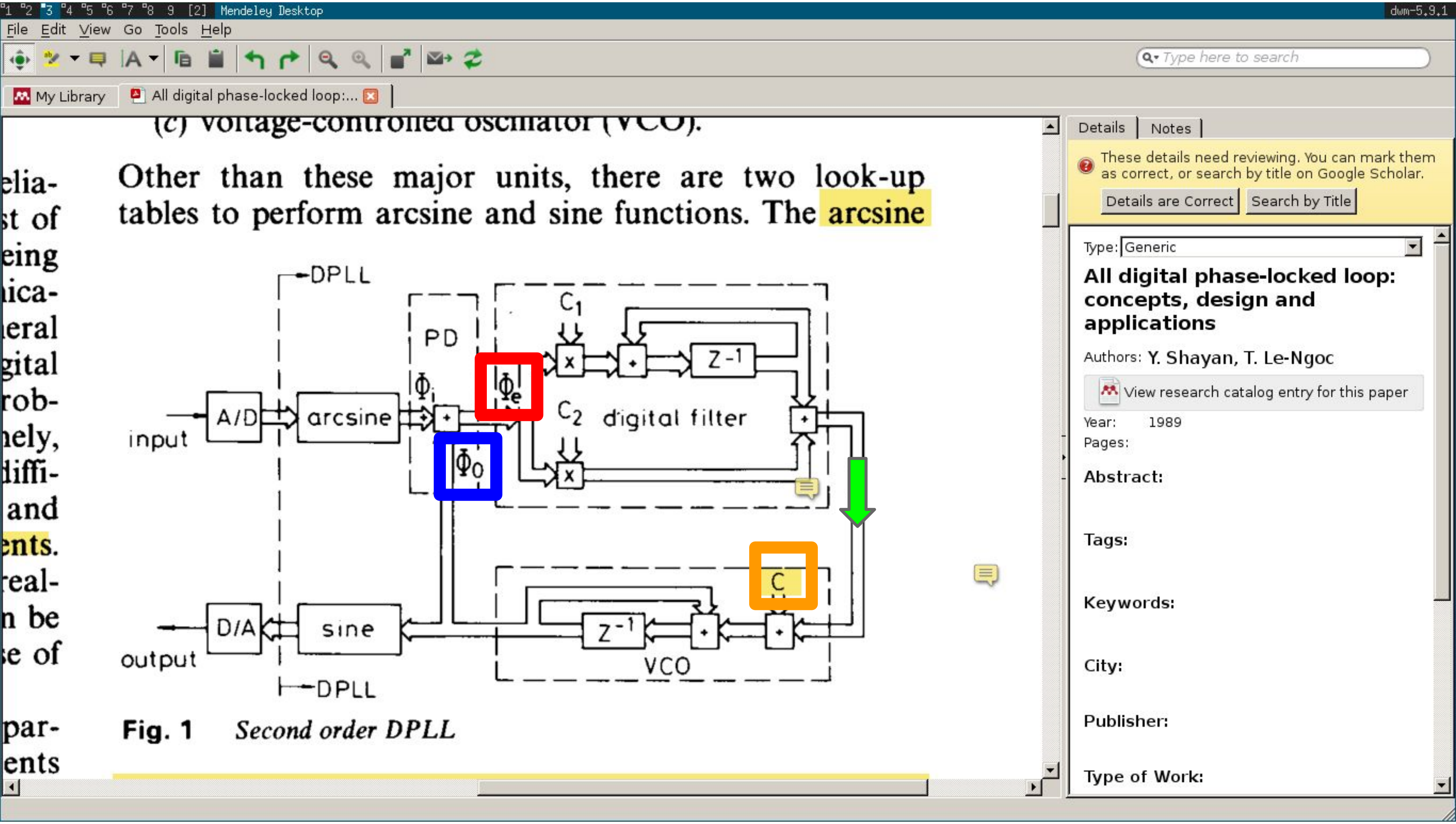
We need to track the frequency very precisely.



Multiply frequency using a PLL



How to implement a PLL using a computer



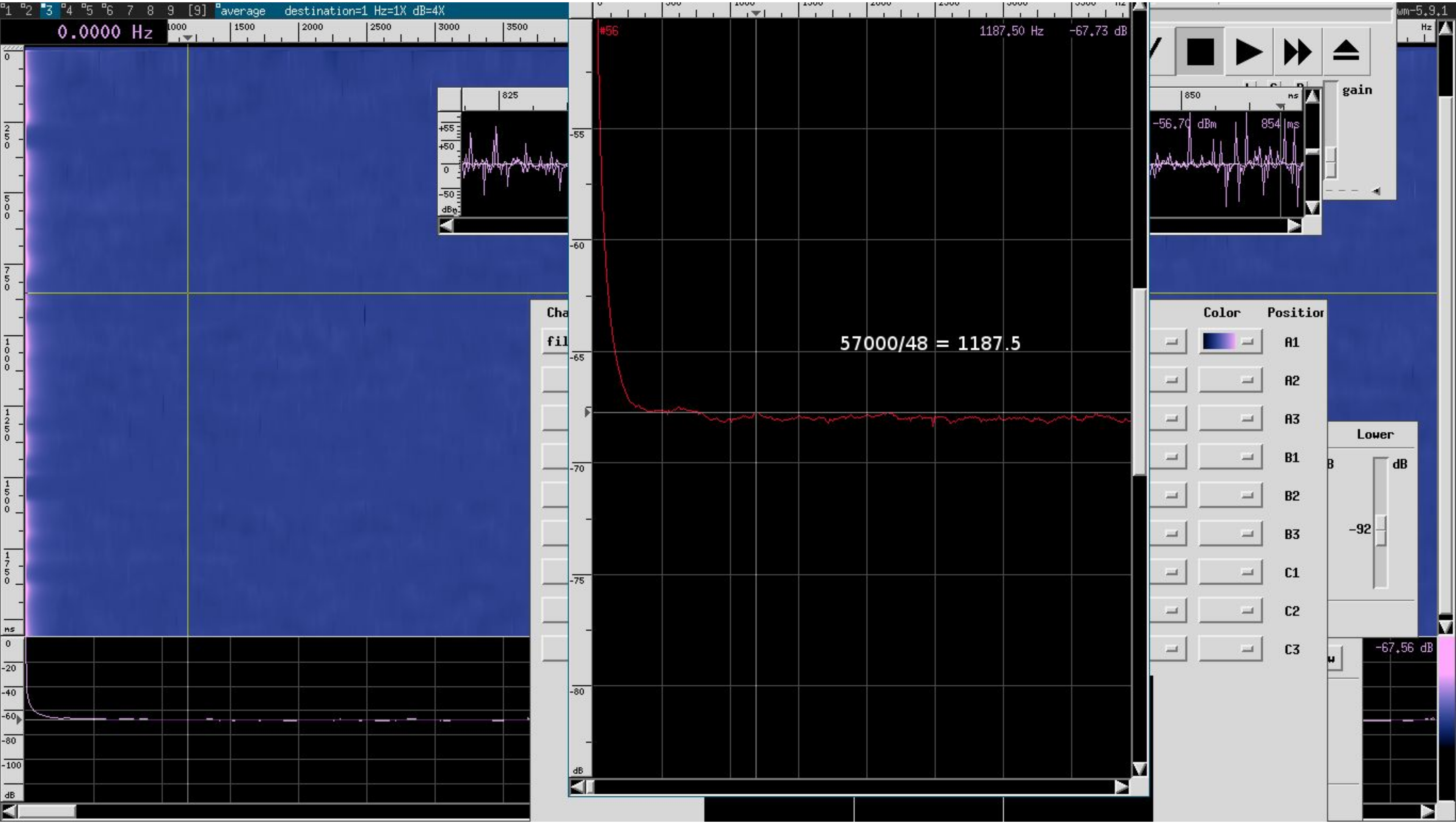
Actual implementation of a PLL

```
(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))))))
```

- old-phi3-cont: allows retiming (by interpolation)
- filt-out: frequency changes
- znew: fits exactly on 19kHz signal
- znew3: corresponding 57kHz signal

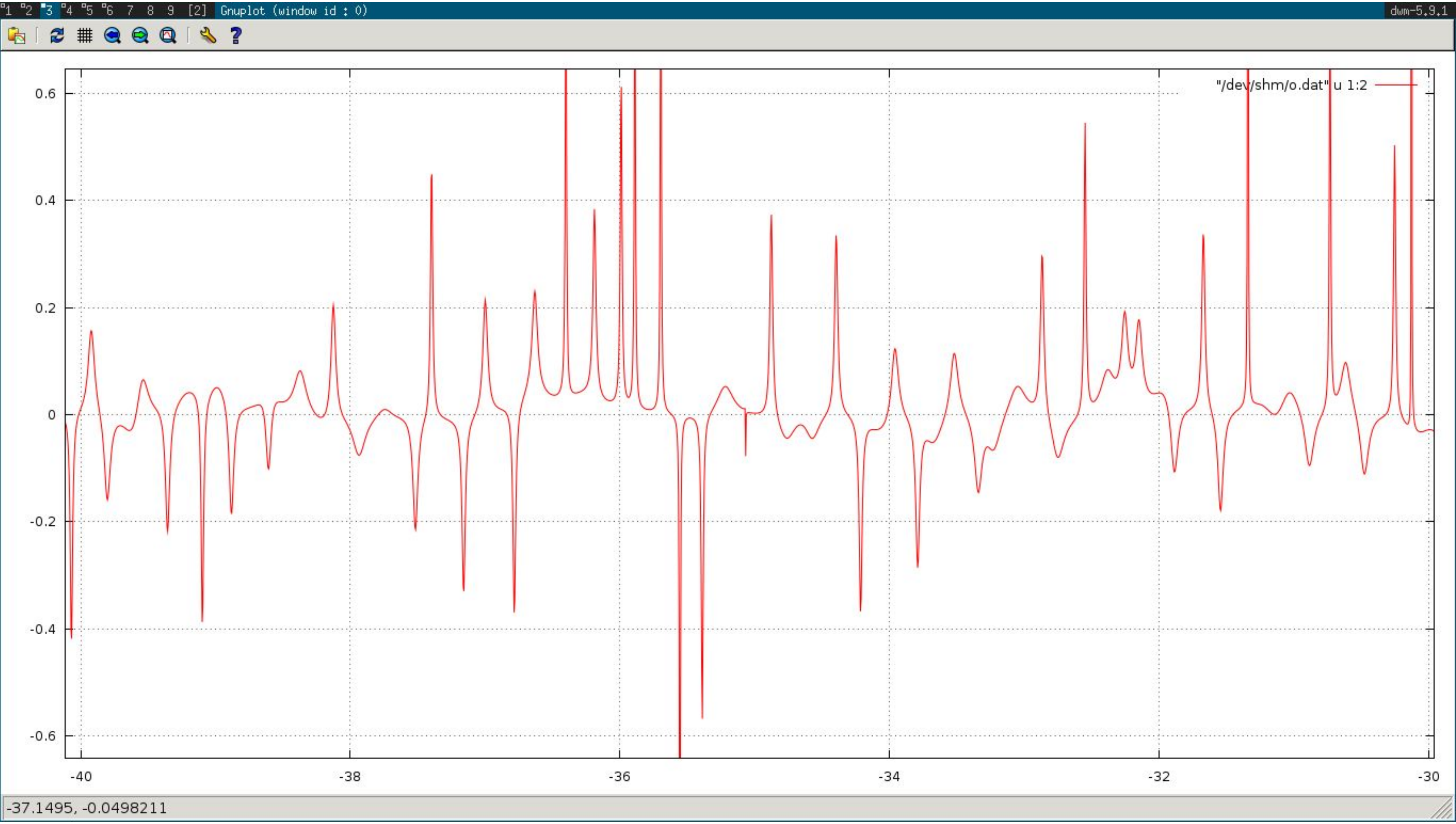
Results

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