FatCat’s DIY Wheel Encoders

by Tim Foden,

MINOS 2012

Overview

- Aims
- Versions
  - Not suitable
  - Final
- Conclusions
- Questions

Aims

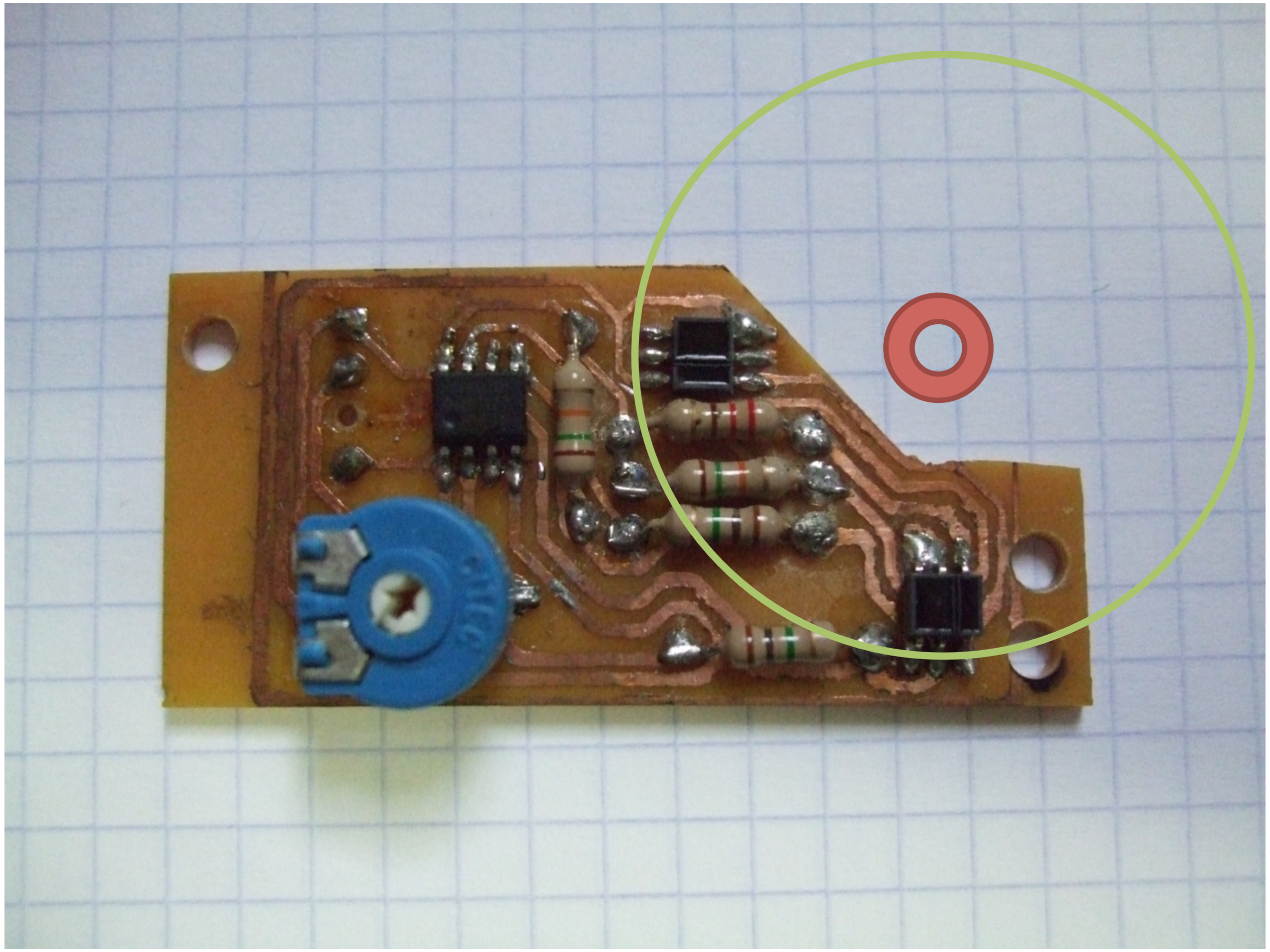
- Would like a resolution of < 1mm
- Would like quadrature output
- Be cheap to make
- Small and light
- Be buildable in my garage
- Easy to mount on the robot
- Get initial practice at making DIY PCBs

[Disclaimer: I don’t really know what I’m talking about]

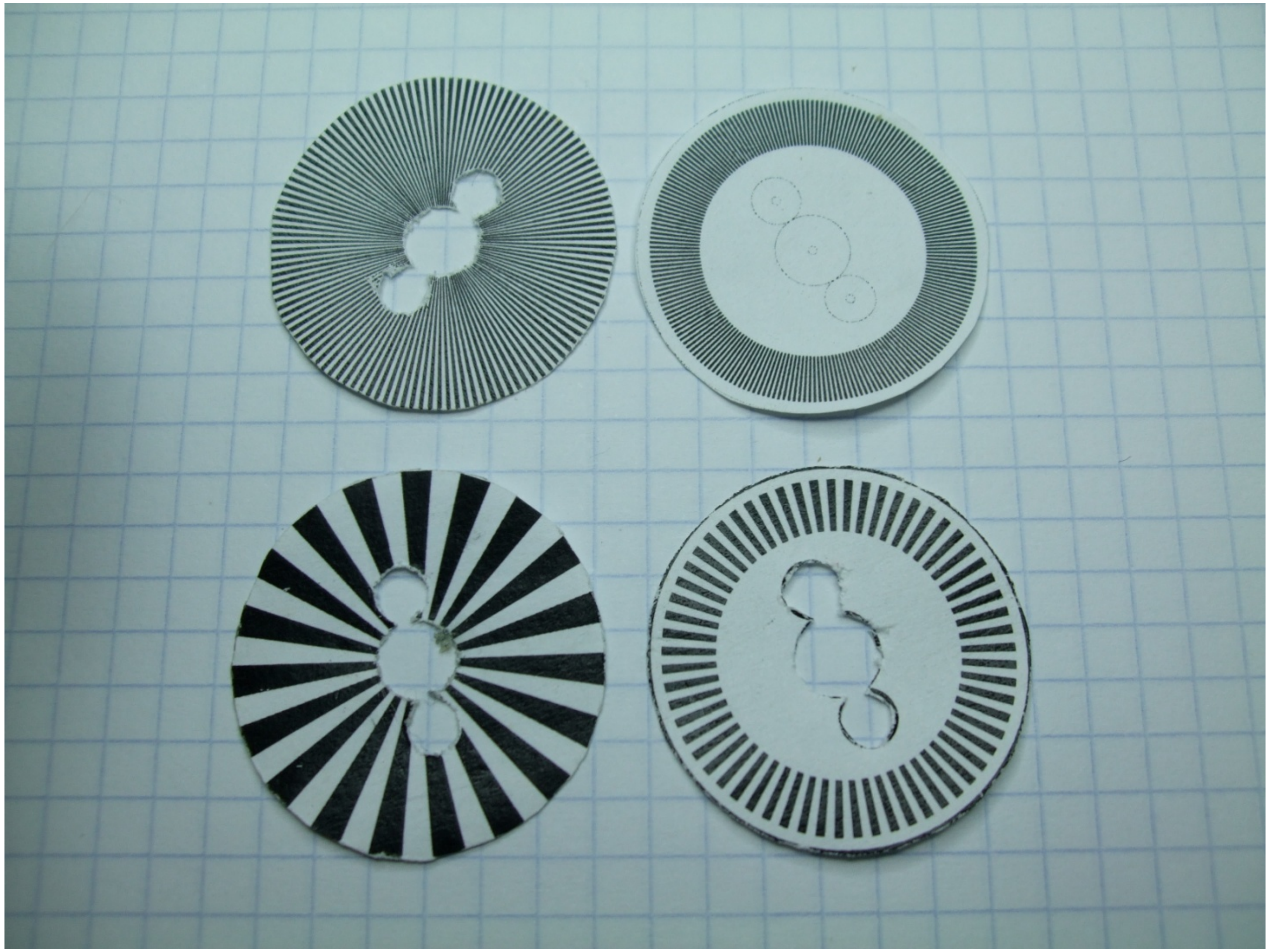
Initial Thoughts

- Use a laser printed encoder pattern disc
- Have two sensors, positioned 90° apart
  - If the encoder pattern has an odd number of cycles then 90° out of phase waveforms are guaranteed
- Use a comparator to convert sine waves to square waves
- Use a POT to set the mid-voltage level

Initial Thoughts - Board



Initial Thoughts – Early Encoder Discs

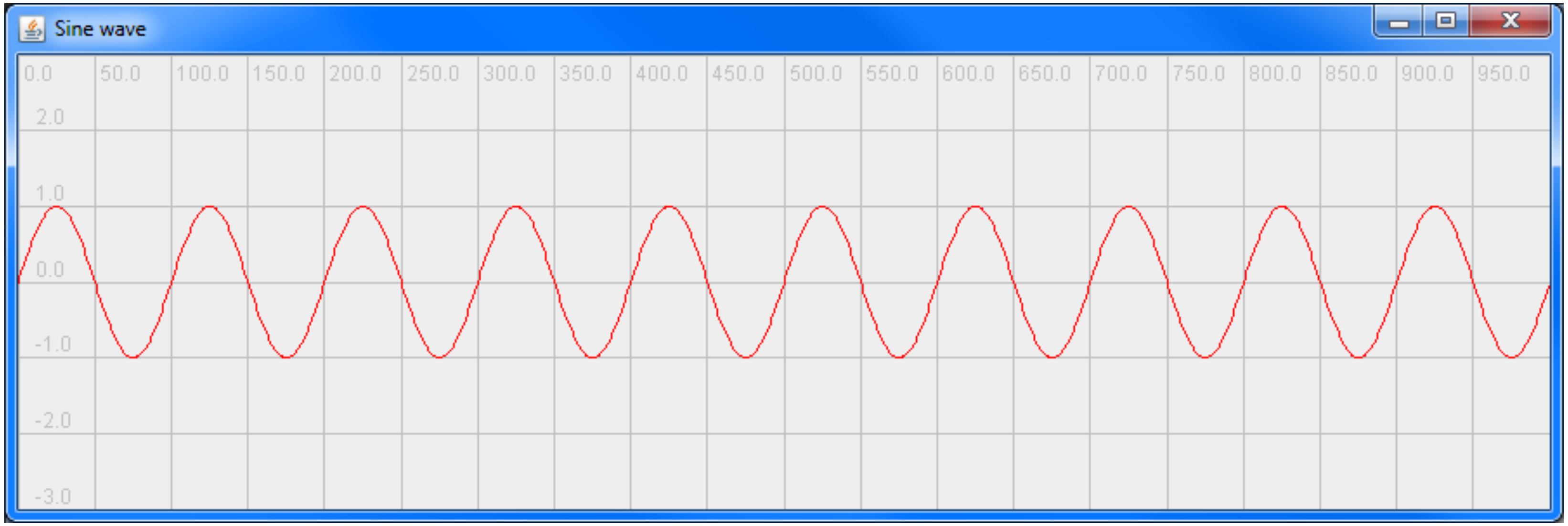


Initial Thoughts - Results

- PCB using toner transfer worked OK
- Laser printed encoder discs worked OK
- Quadrature separation difficult to achieve
  - Compounded by other ‘noisy’ signal effects

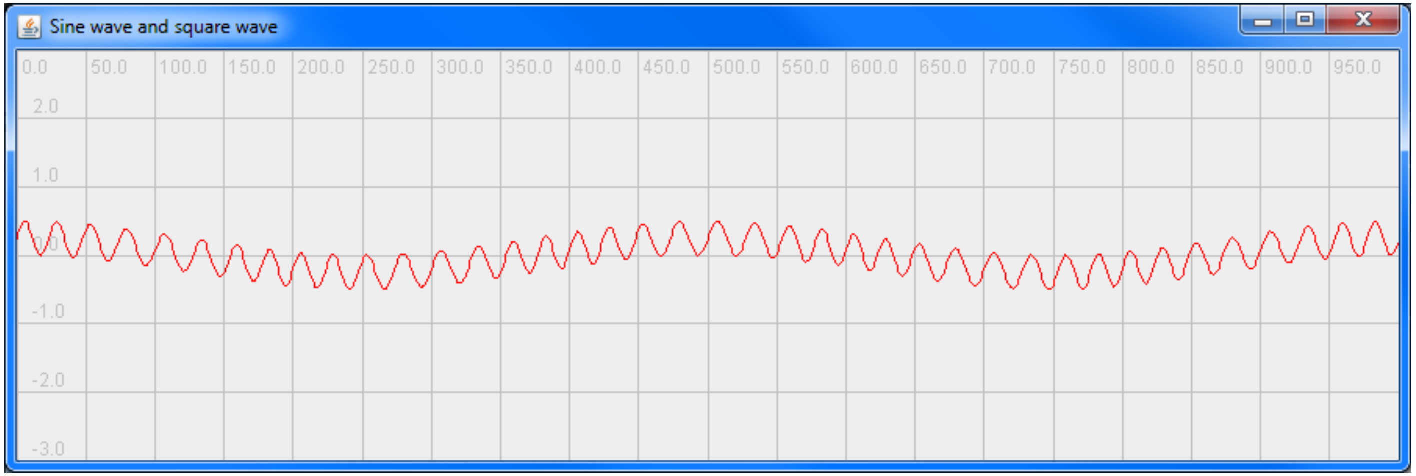
Initial Thoughts - Signals

- This is the kind of signal I expected to see:



Initial Thoughts - Signals

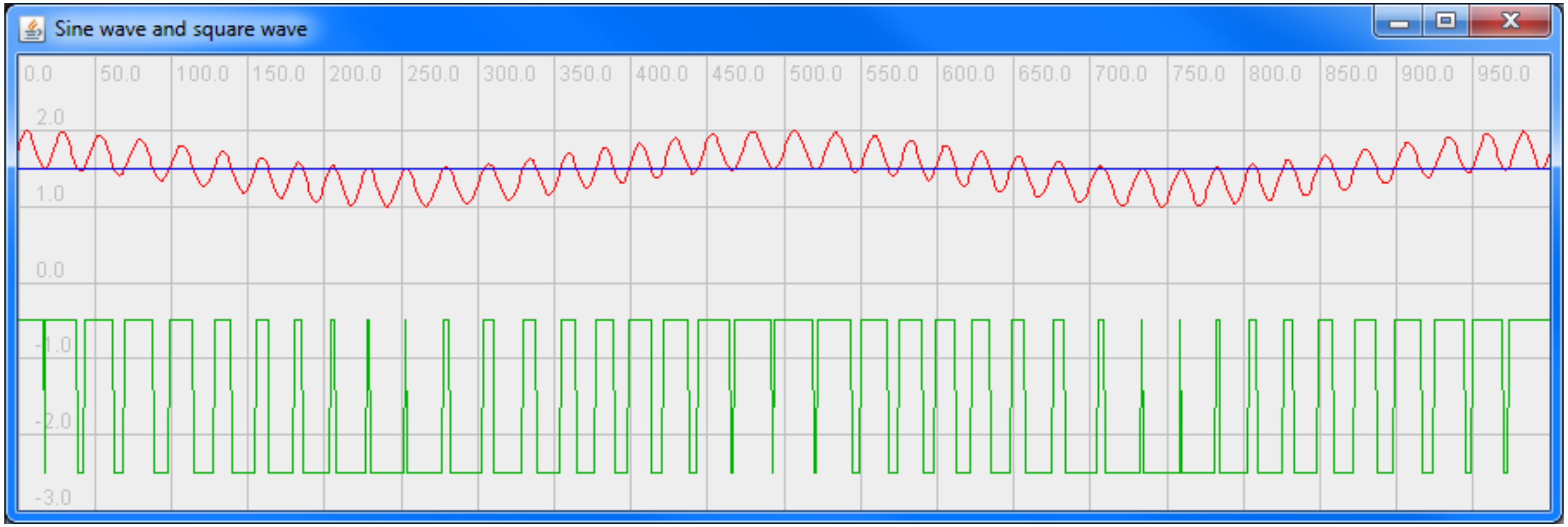
- This is the kind of signal I got:



  - Amplitude too small
  - Vertical 'wave'
  - Horizontally oscillating period

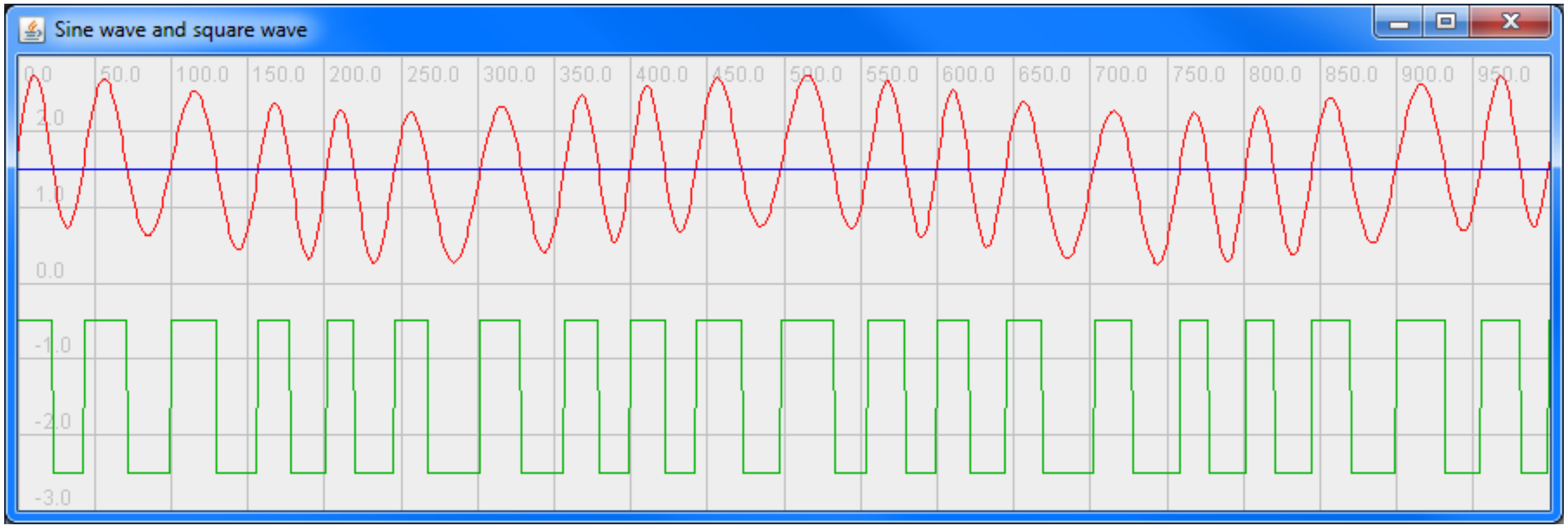
Initial Thoughts - Signals

- Useless for converting to a square wave:



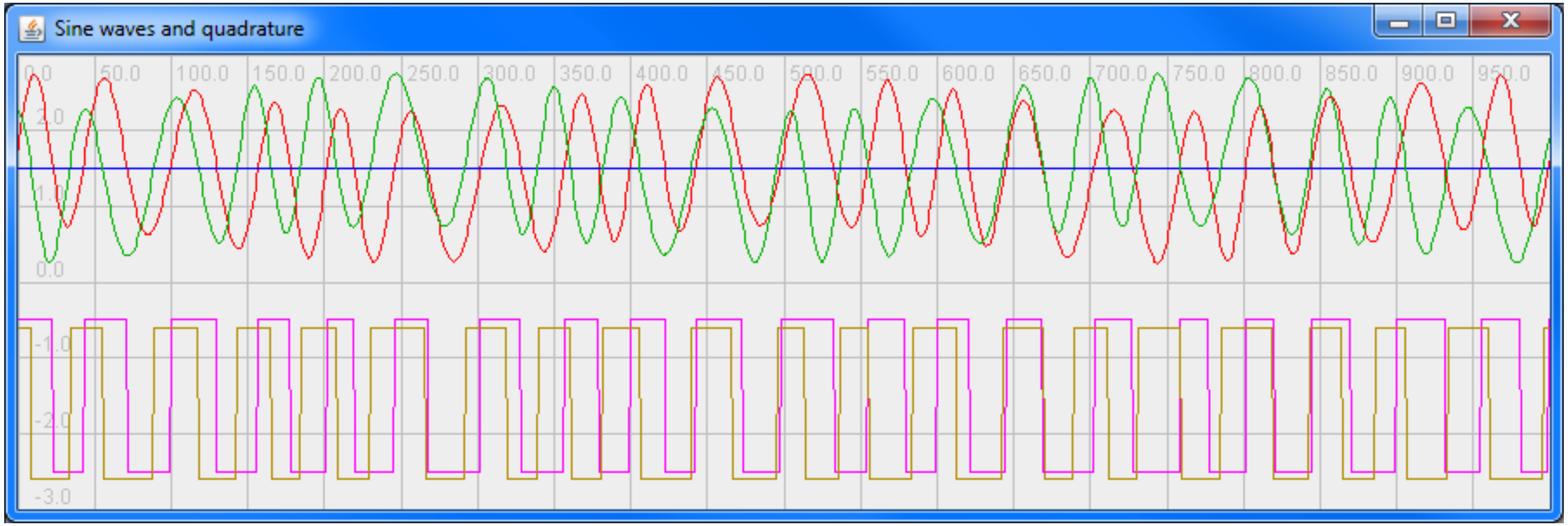
Initial Thoughts - Signals

- Use a lower resolution encoder pattern to get a larger amplitude signal:



Initial Thoughts - Signals

- This still produces a signal that isn't really good enough for reliable quadrature:



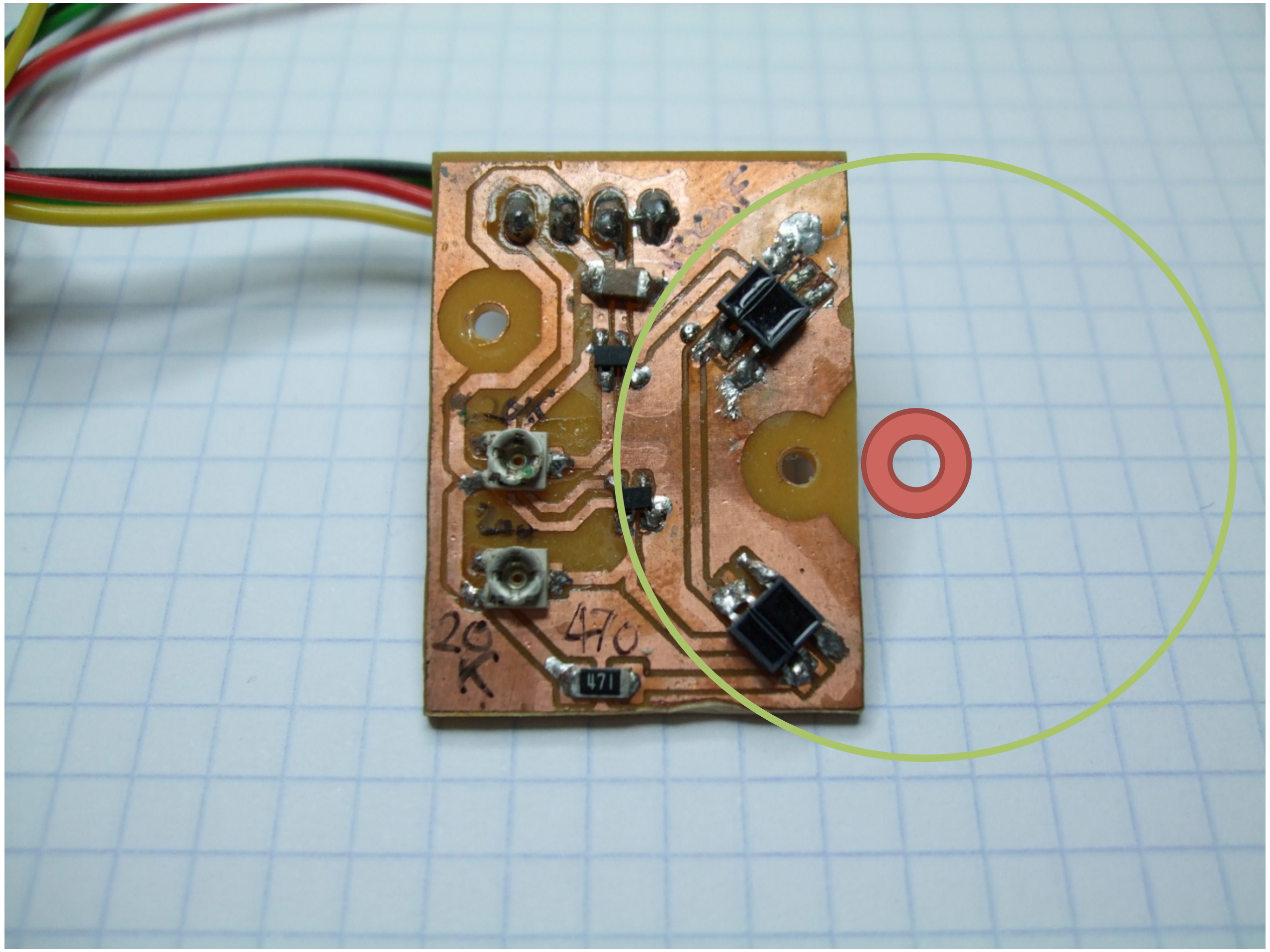
Initial Thoughts - Results

- PCB using toner transfer worked OK
- Laser printed encoder discs worked OK
- Quadrature separation difficult to achieve
  - Compounded by other ‘noisy’ signal effects
- Single pot for mid-range voltage not good enough
- Used op-amp (MCP6002) instead of comparator
  - Why?
  - I think this eventually failed

2nd Version

- Really use a comparator this time
- Allow slightly more flexibility in positioning to tweak quadrature separation
- Use 2 POTs for setting mid voltage
- Use SMT components throughout

2nd Version - Board



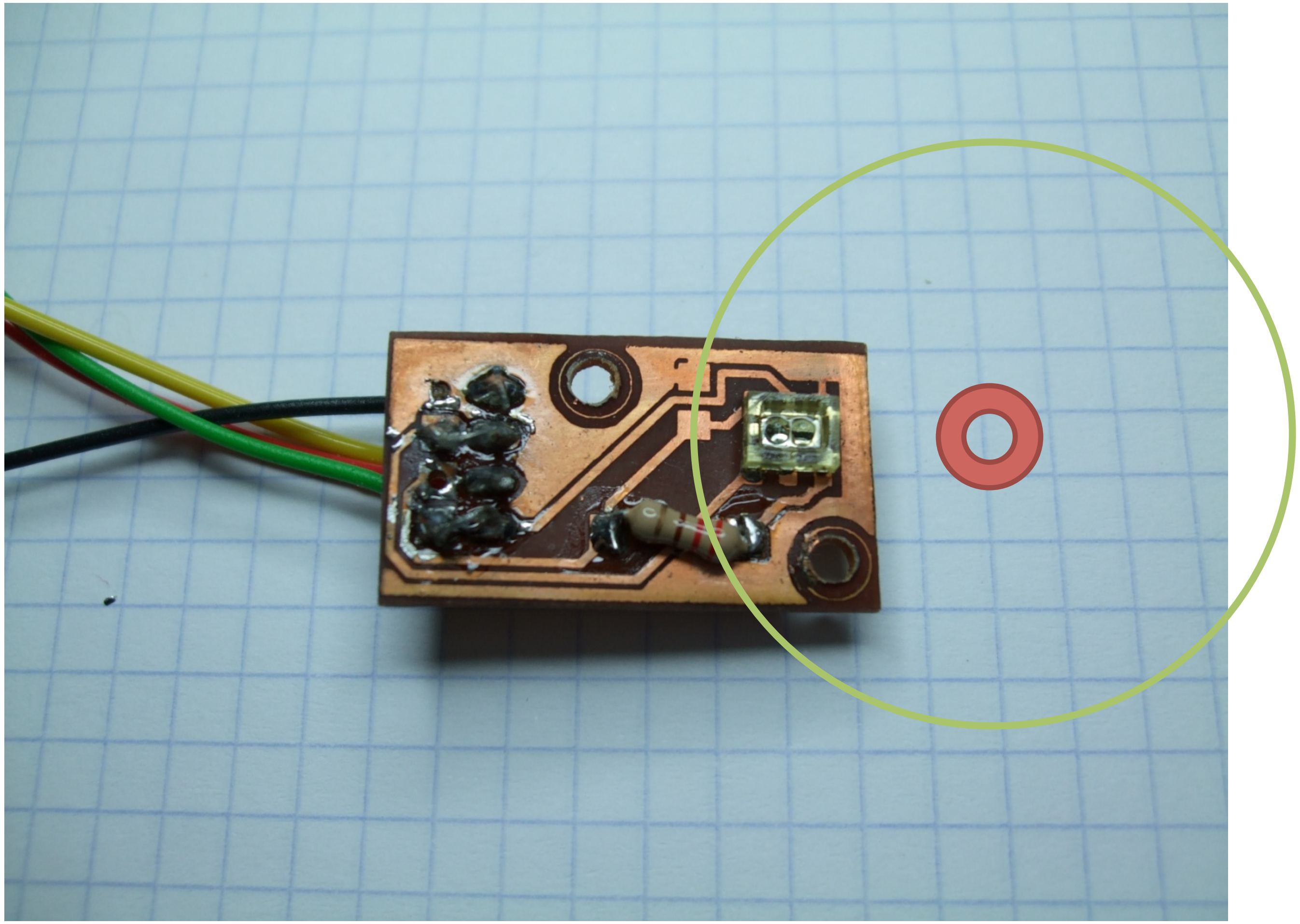
2nd Version - Results

- Still too difficult to align to get good quadrature separation
- Even if I could align the sensors to get the quadrature timing correct...
  - The signal ‘wobbles’ up and down
  - The signal still has horizontal ‘compression’
  - Difficult to get the resolution required

3rd Version

- Back to the drawing board
- Looked around once again for another way to achieve aims
  - Found a monolithic chip that appeared just the job
    - Avago Technologies
    - AEDR-8300-1P2
    - 2 Channel, 150 LPI

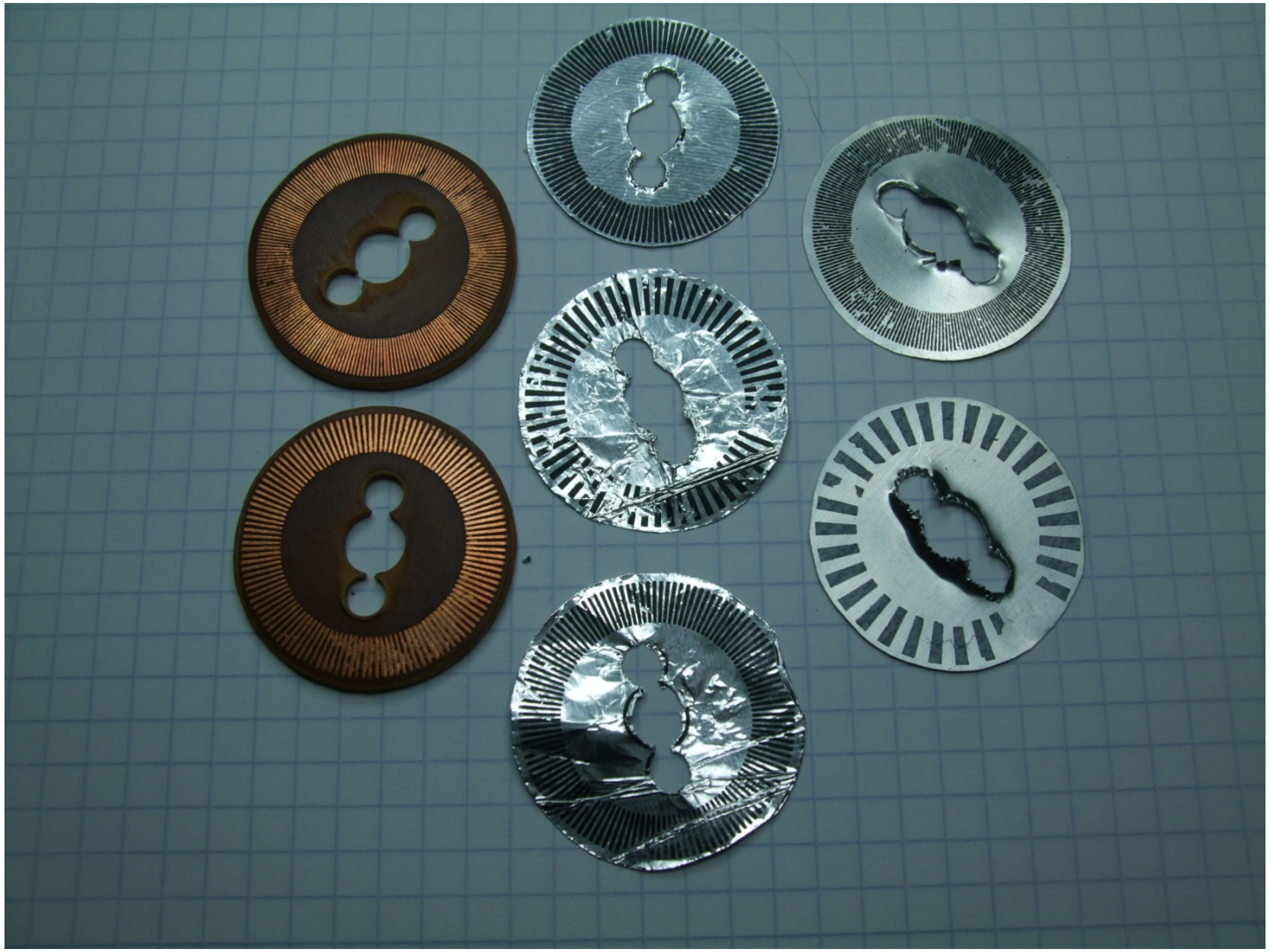
3rd Version - Board



3^{rd} Version - Results

- I failed to get this encoder to work
  - Needs to have a reflective disc
    - Laser printed encoder disk not suitable
    - Failed to make a suitable reflective encoder disc
  - Chip possibly ***requires*** the encoder pattern to be the correct resolution
    - And this is too high a resolution to be able to manufacture in my garage

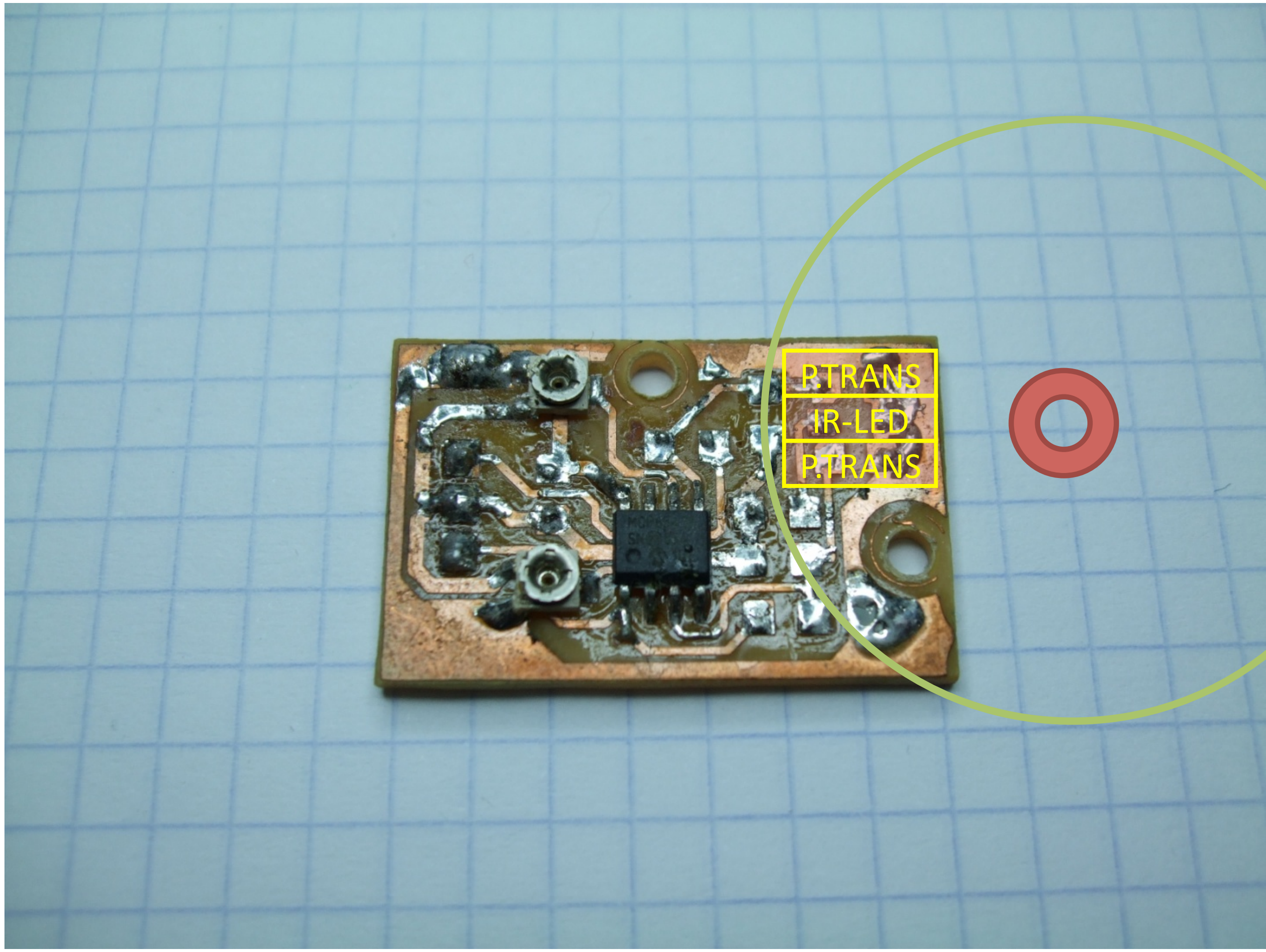
3rd Version - Results



4^{th} Version

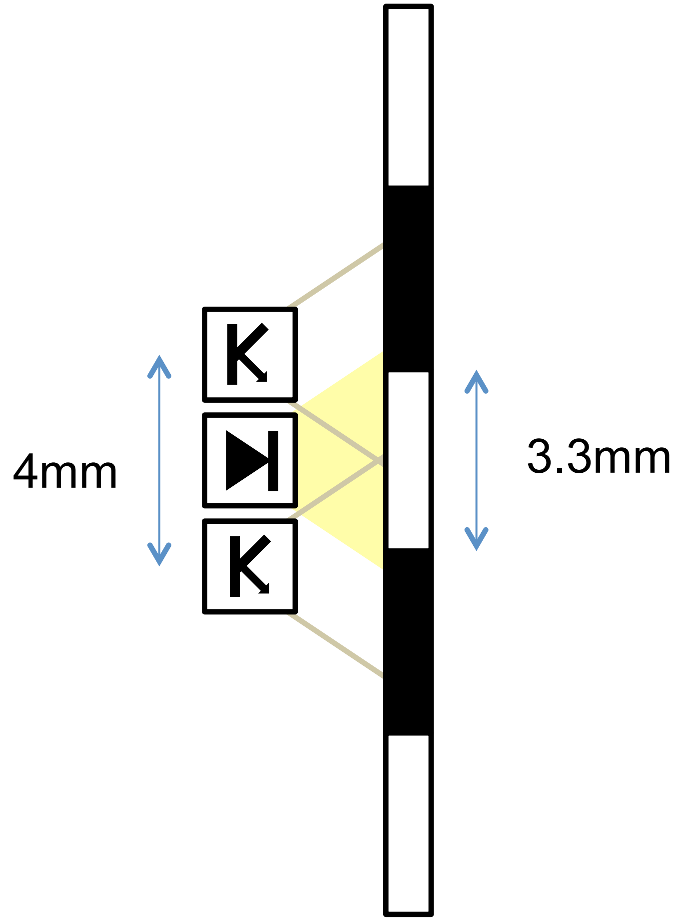
- The single chip encoders gave me an idea
  - It's possible to mount the detectors close together
    - No need for 90° separation of detectors
    - If distance between detectors matched with encoder pattern, quadrature separation should be guaranteed.

4th Version - Board



4th Version – Emitter/Detector arrangement

- 90° signal separation can be achieved if the distance between the centres of the phototransistors is matched to the width of the stripes on the encoder disc
  - To get 90° separation need 1½ stripes... so the stripe width should be 2/3 of the distance between the phototransistors



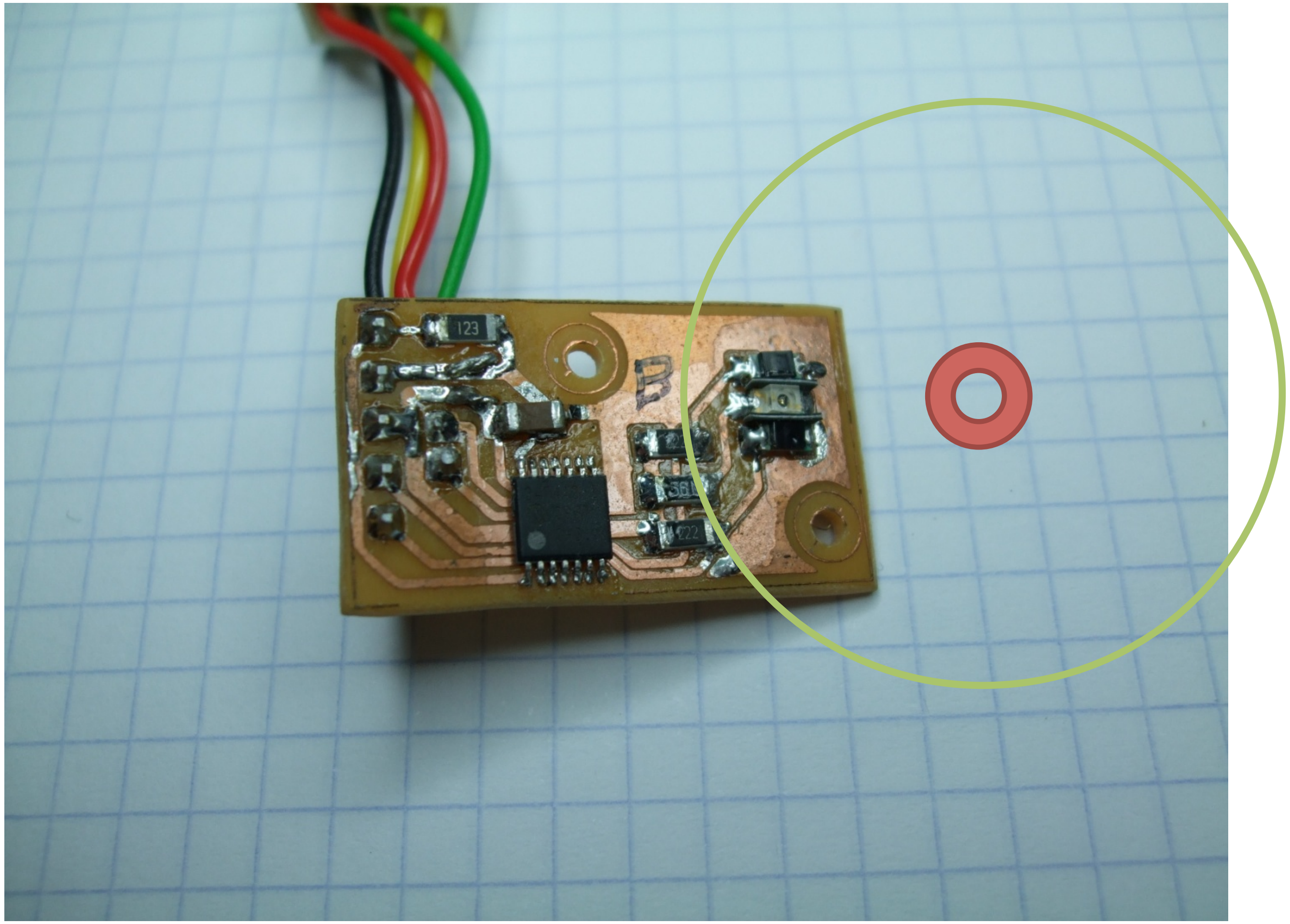
4^{th} Version - Results

- Easier to get quadrature alignment
- Still have various noisy elements
- Still not able to give fine enough resolution

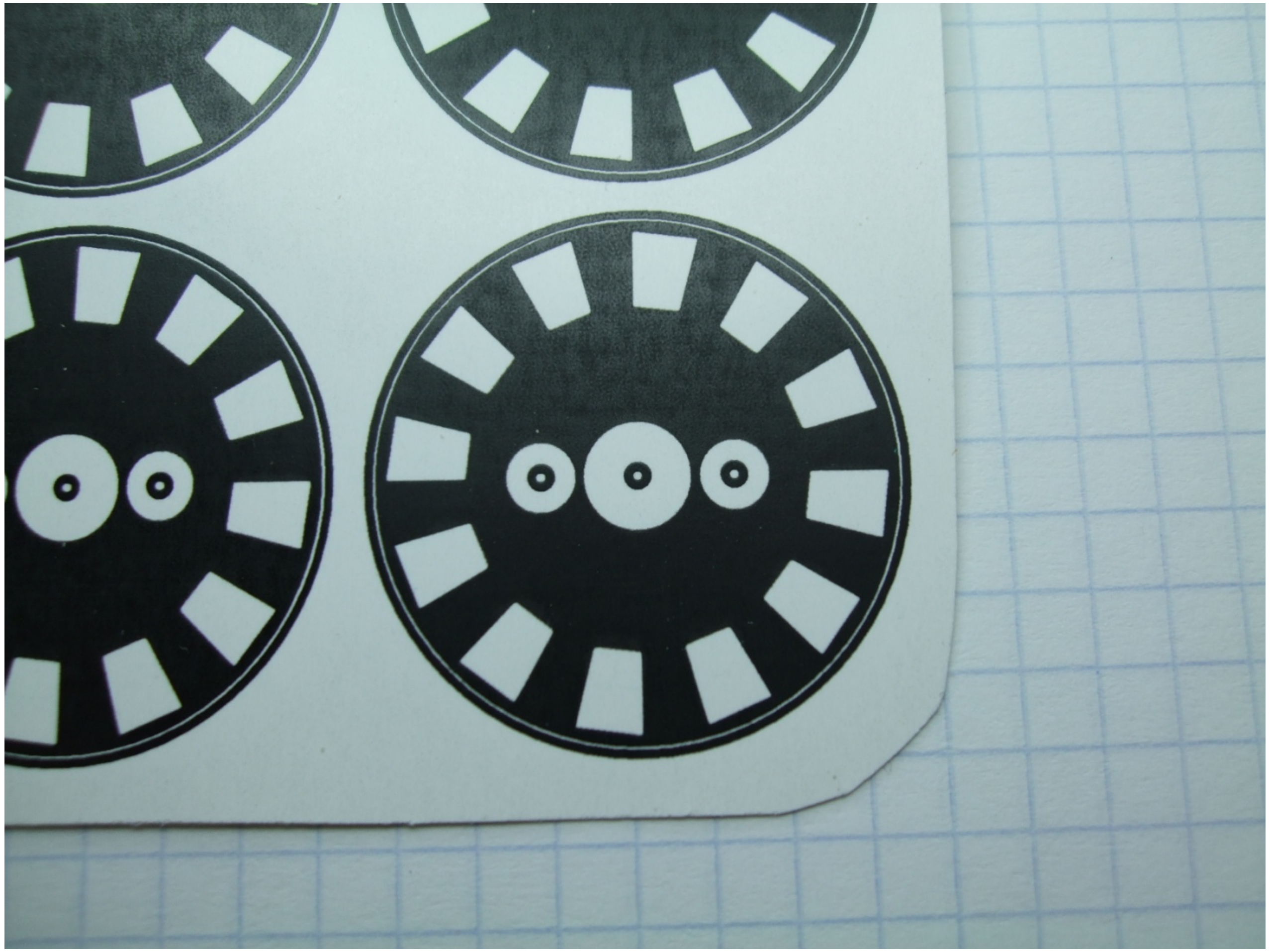
5^{th} (and Final) Version

- Back to drawing board (again!)
  - Looking at magnetic encoders that output quadrature give me a new idea – Use a PIC instead of a comparator
- Hopefully this will solve a number of problems:
  - Can use large encoder pattern to get large signal to noise ratio – reduces the effect of the 'wobble'
  - Can use trigonometry to work out actual angle between repetitions of the pattern, and thus still get a high resolution
  - Can auto-calibrate to reduce problems due to differing signal amplitudes and not quite accurate phase separation

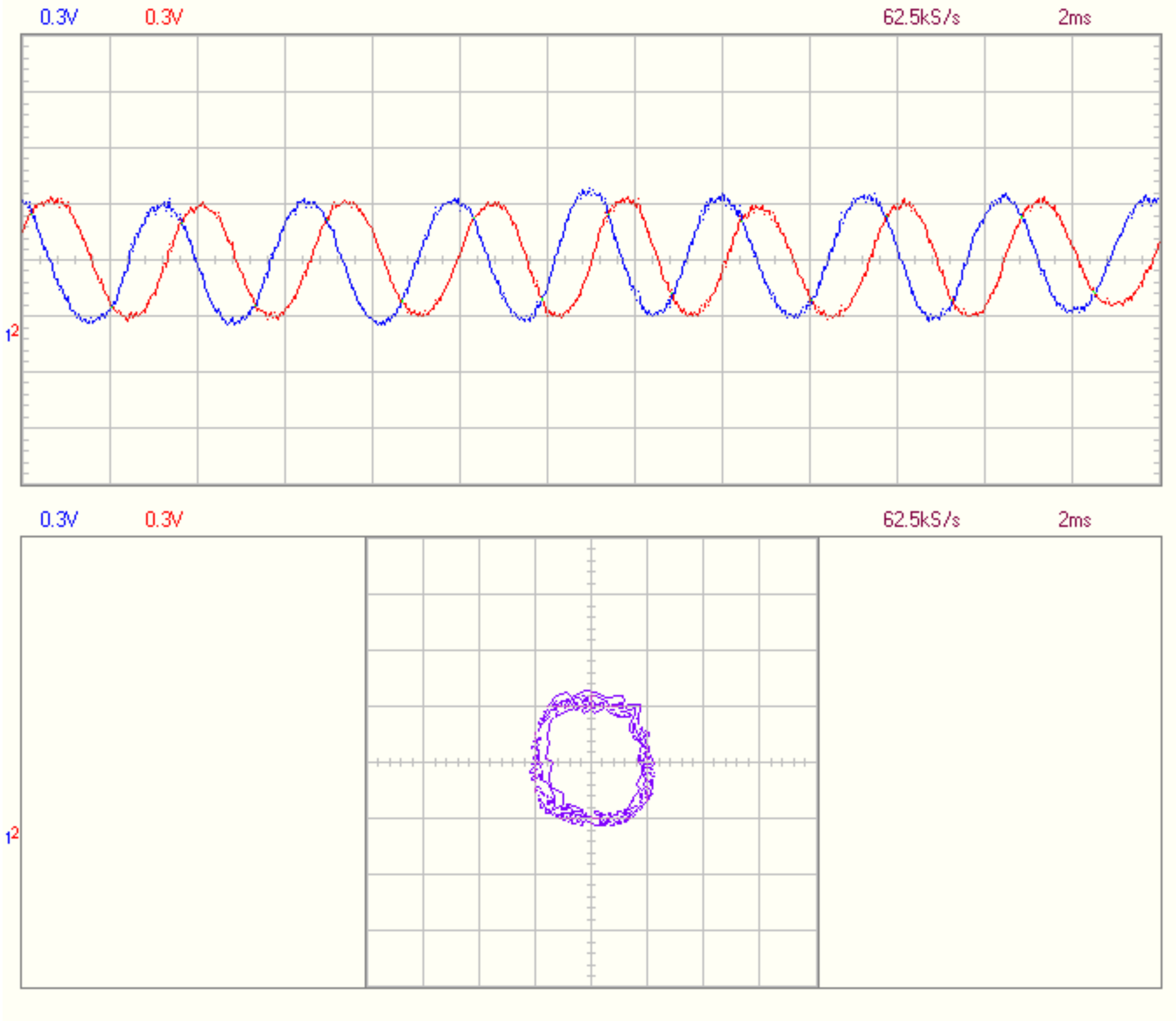
5^{th} Version - Board



5th Version – Encoder Pattern

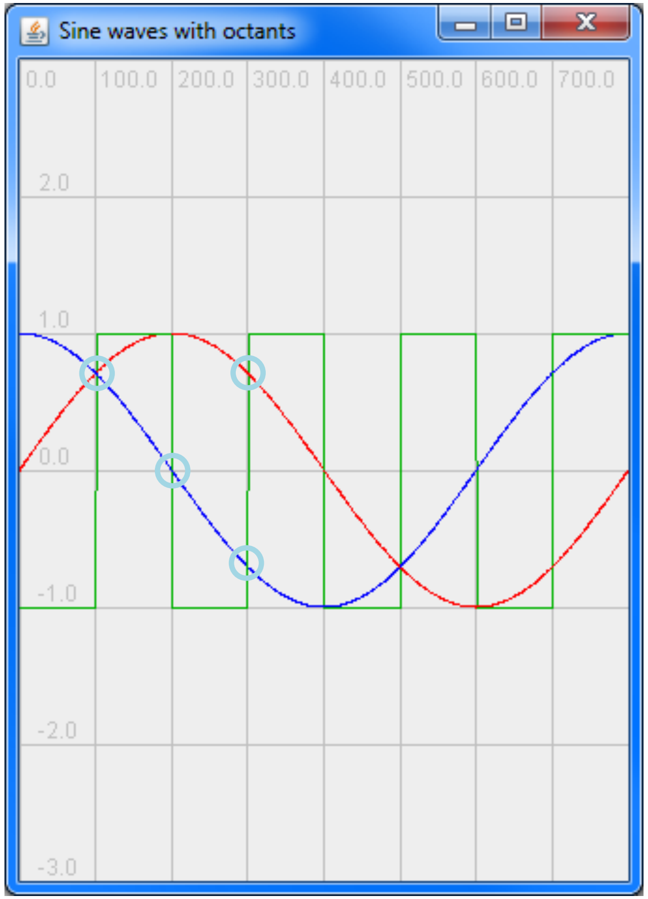


5th Version – Sine Wave Signal



5th Version - Calibration

- It is possible to identify 8 equidistant points in time and angle along a plot of sine waves in quadrature.
- These points are where the waves cross the 0 line, and where the absolute values cross each other.
- By running the wheels at a constant speed it is possible to get pairs of times and angles, and use these to linearise the output.



5th Version – Sensor Psuedo-code

```
complexQuadrature
{
        setupA2dForFreeRunning
        performCalibration
        loop-forever
        {
                waitForA2dCycleToComplete
                x, y = readA2dValues
                adjustMinAndMaxValues
                midX, midY = calculateMidPoints
                dx, dy = x – midX, y – midY
                angle = atan2(dy, dx)
                angle = perform-linearisation(angle)
                updateAccumulatedAngle
                performHysteresis
                sendAccumulatedAngle
        }
}
```

5^{th} Version – Results / Conclusion

- Worked satisfactorily – Hurrah!
  - Resolution < 1mm (actually about 1/3 mm)
  - Can output quadrature
  - Is relatively cheap to make
  - Small and light
  - I can build it in my garage
  - Easy to mount on FatCat
  - Lots of practice at making PCBs – more than I expected!

Questions