Kristian Harder, Eckhard von Toerne, KSU

KSU readout status: 10 chip hybrid

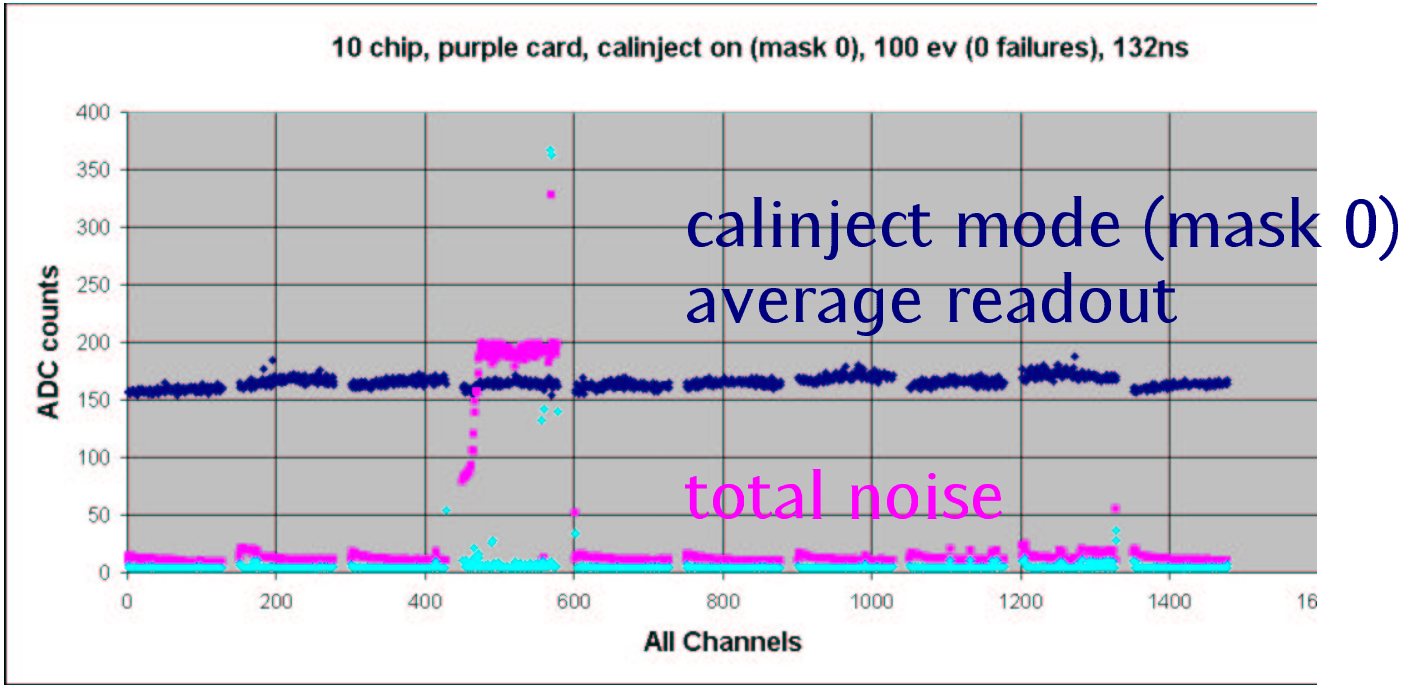
setup:

- SASEQ with line driver kludge card from Russell
  ($\longrightarrow$ same drivers as big sequencer)
- purple card
- 10 chip hybrid

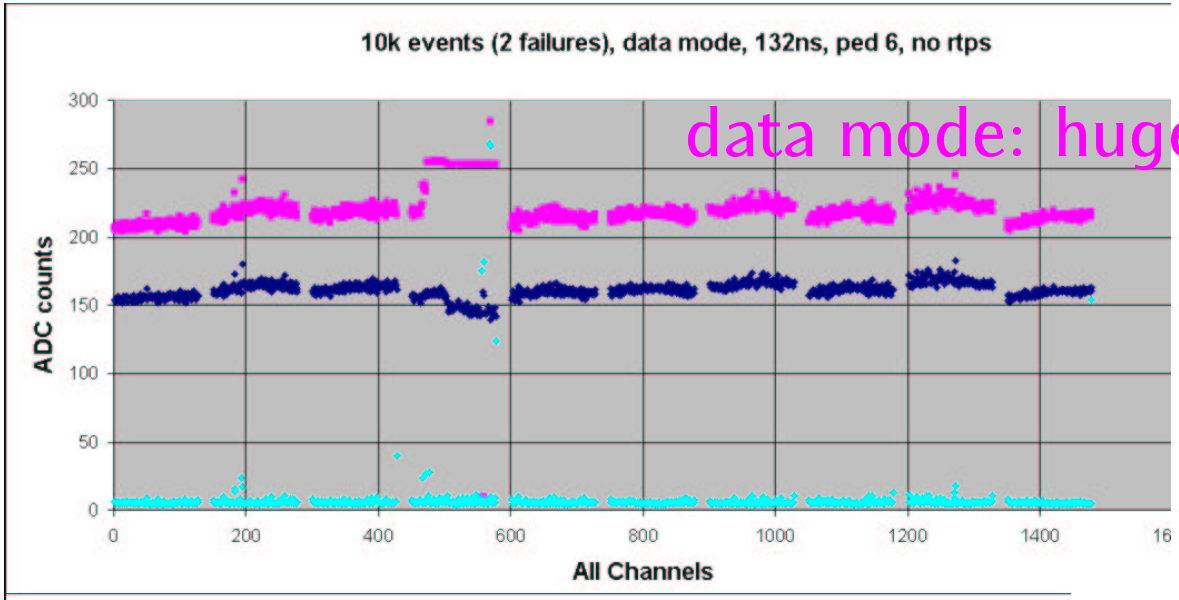
all tests done with (modified) spreadsheet version 2.0

Purple card readout: pedestal 6, no RTPS

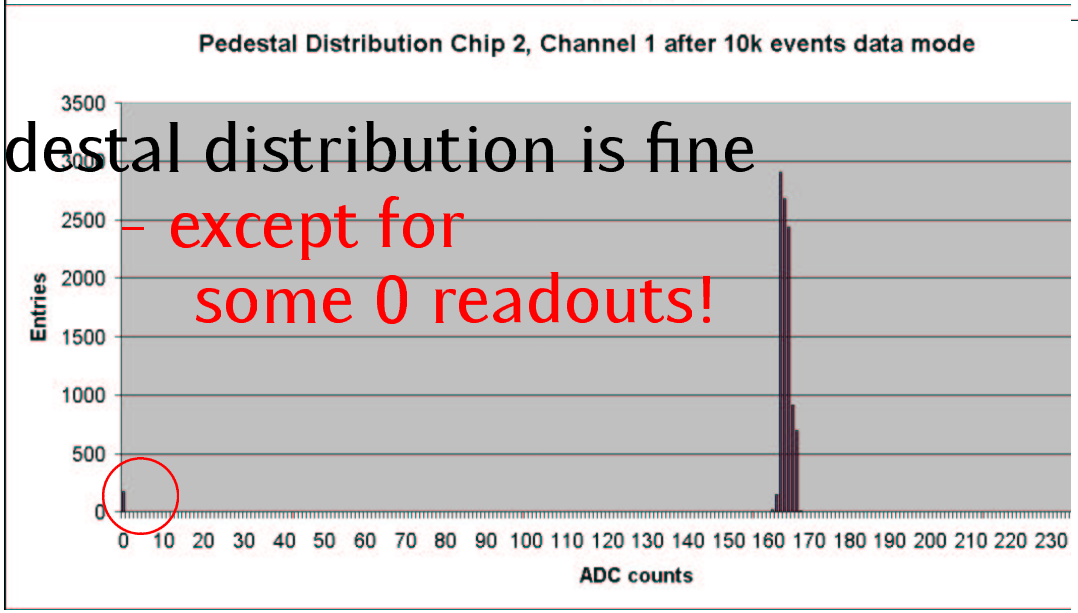
readout mode	readout error rate	noise level
CalInject mode	$< 0.01\%$	≈ 1
132ns data mode	$\approx 0.005\%$	≈ 20–30
396ns data mode	$\approx 0.005\%$	≈ 20–30



Purple card readout: pedestal 6, no RTPS



data mode: huge noise???

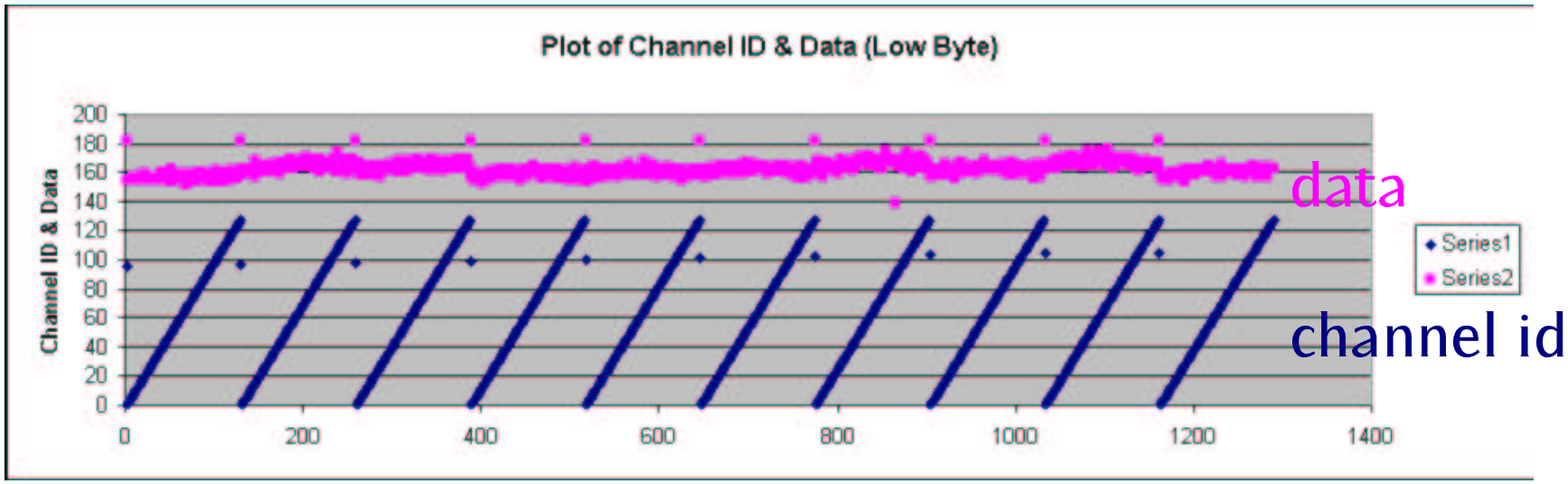


pedestal distribution is fine

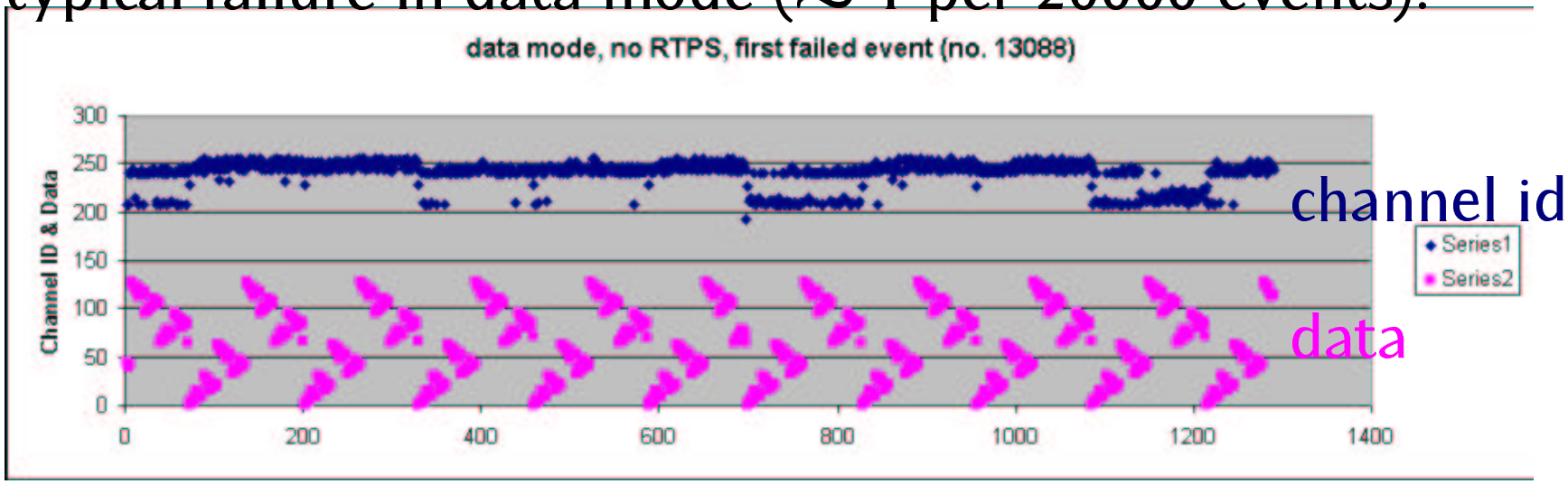
– except for
some 0 readouts!

Purple card readout: pedestal 6, no RTPS

good event in data mode:

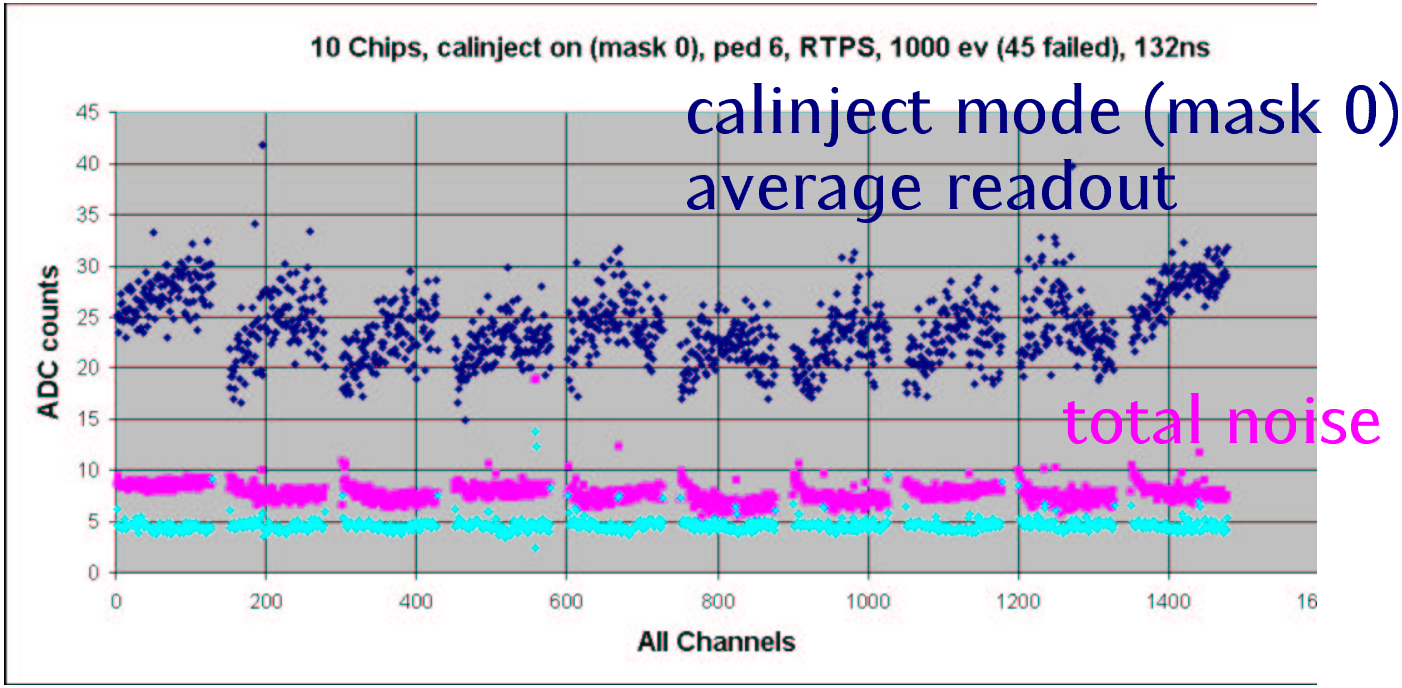


typical failure in data mode (≈ 1 per 20000 events):

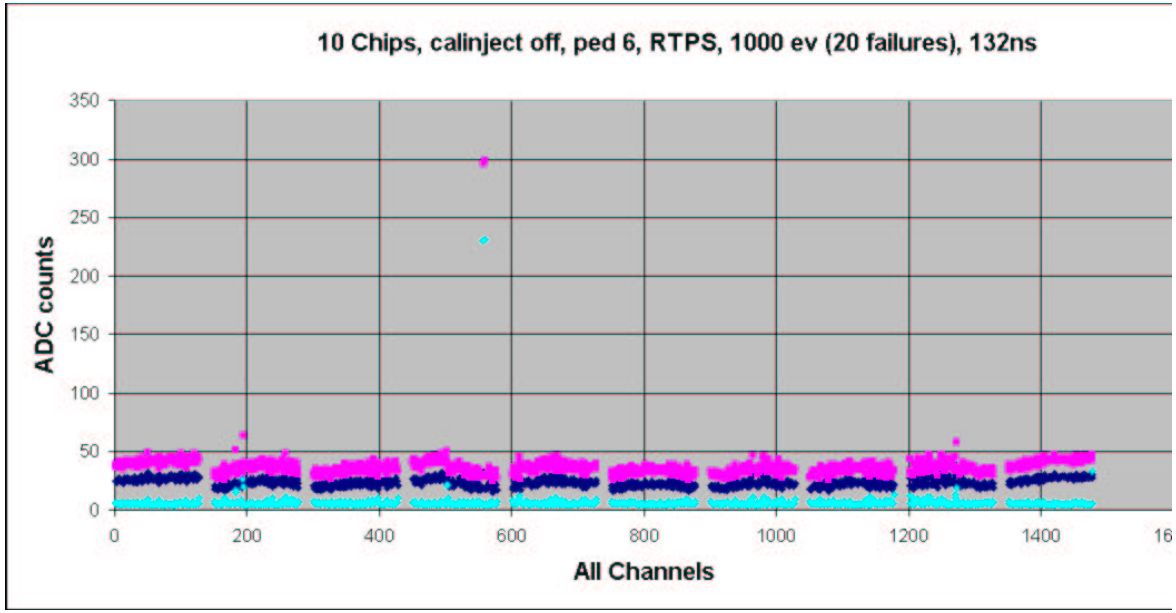


Purple card readout: pedestal 6, with RTPS

readout mode	readout error rate	noise level
CalInject mode	$\approx 5\%$	≈ 1
132ns data mode	$\approx 5\%$	≈ 5
396ns data mode	$\approx 5\%$	≈ 5



Purple card readout: pedestal 6, with RTPS

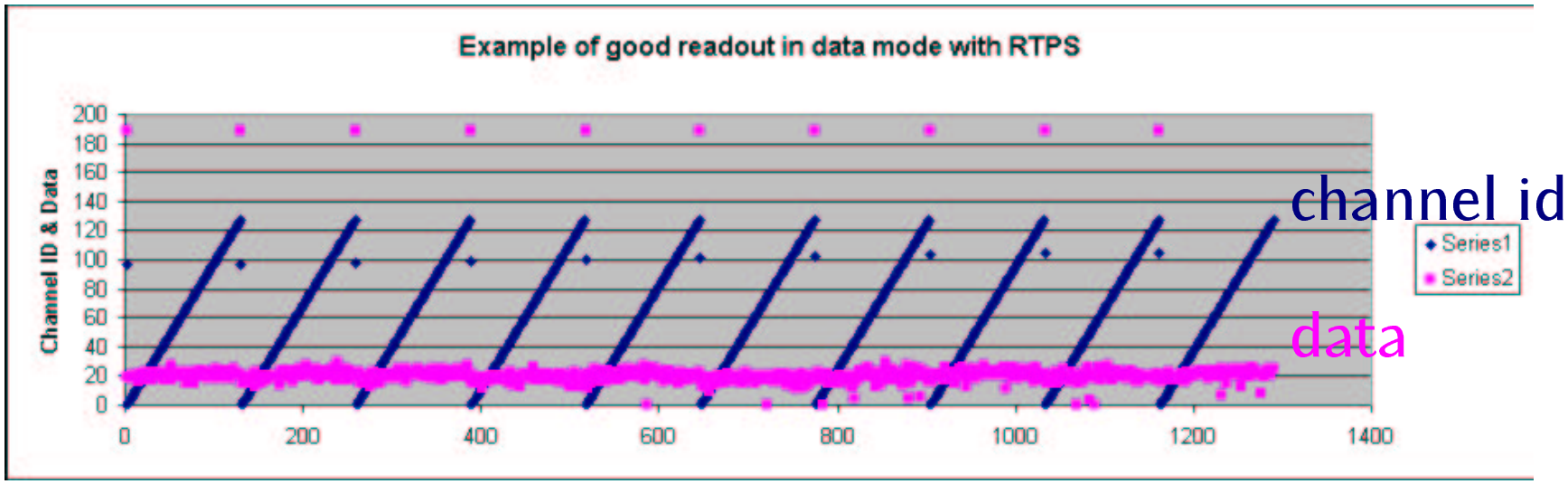


same problem as in data mode without RTPS:

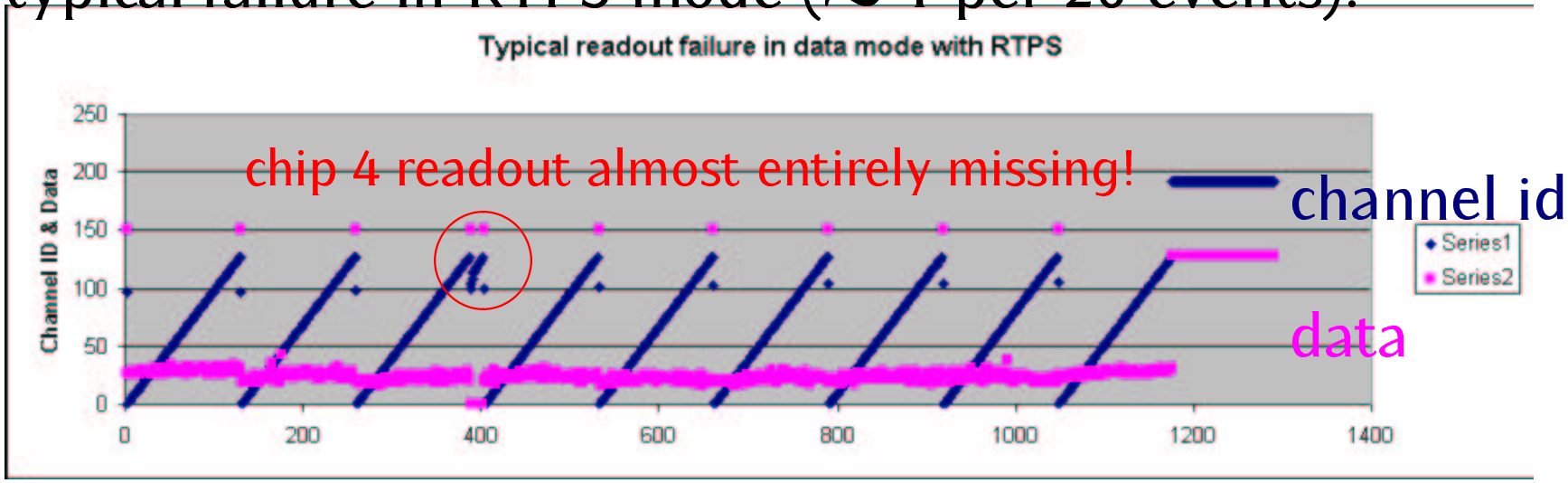
occasional 0 ADC readouts!

Purple card readout: pedestal 6, with RTPS

good event in RTPS mode:

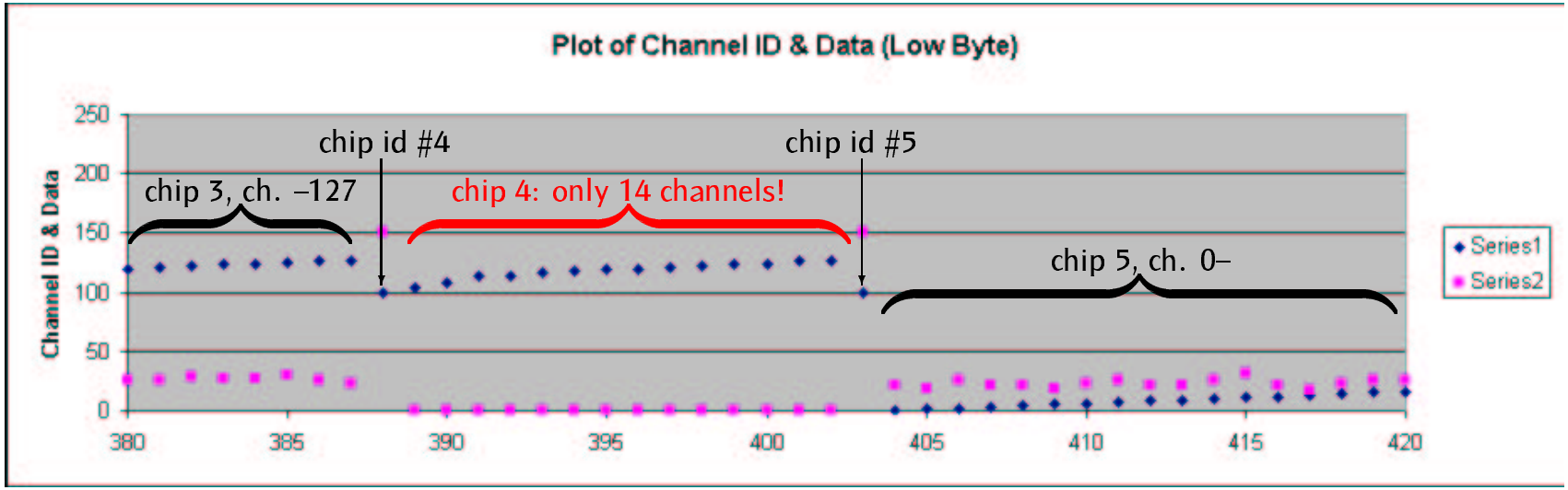


typical failure in RTPS mode (≈ 1 per 20 events):



Purple card readout: pedestal 6, with RTPS

zoom into chip 4:

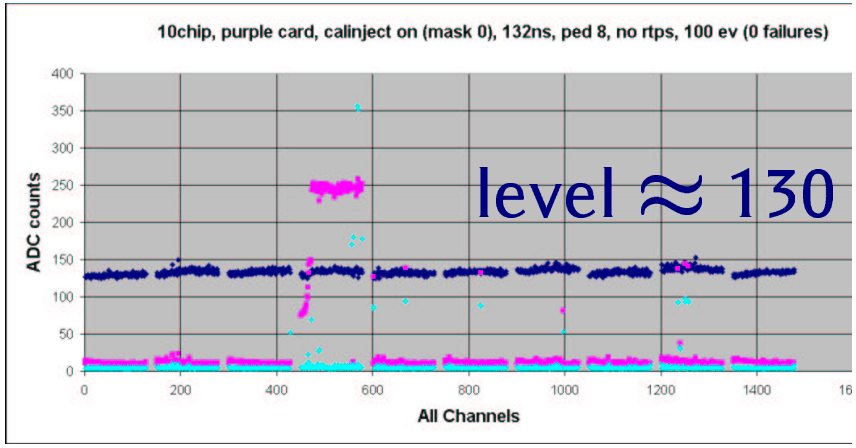


same channels missing in all failed readouts:

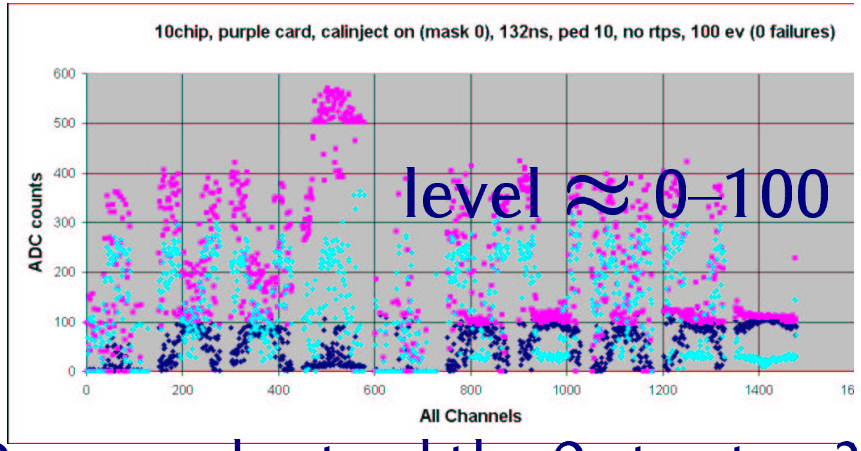
0–103, 105–107, 109–112, 115, 117, 125

Pedestal problems?

pedestal 8

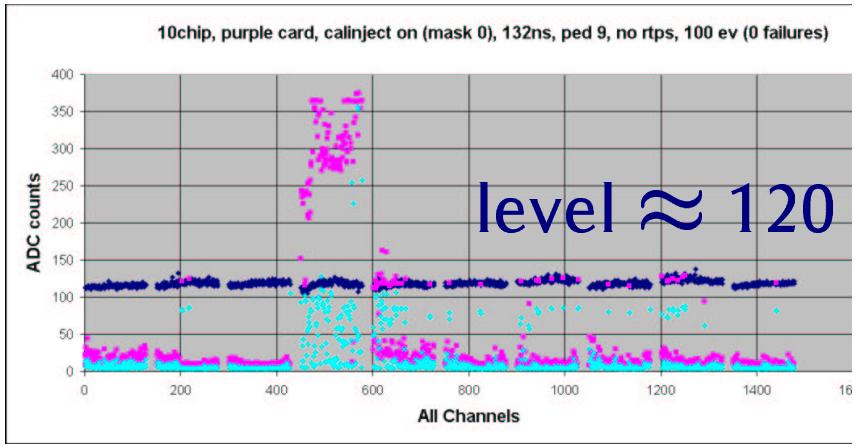


pedestal 10

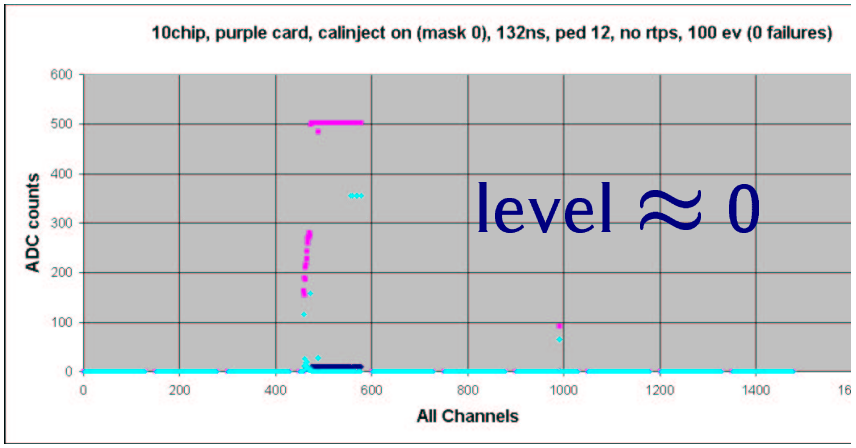


Do we understand the ∩ structure?

pedestal 9



pedestal 11



Overview of issues encountered so far

- first chip id was wrong

  $\longrightarrow$ resolved. typo in spreadsheet
- no CalInject signals when 396ns flag set

  $\longrightarrow$ resolved. spreadsheet feature
- readout errors:

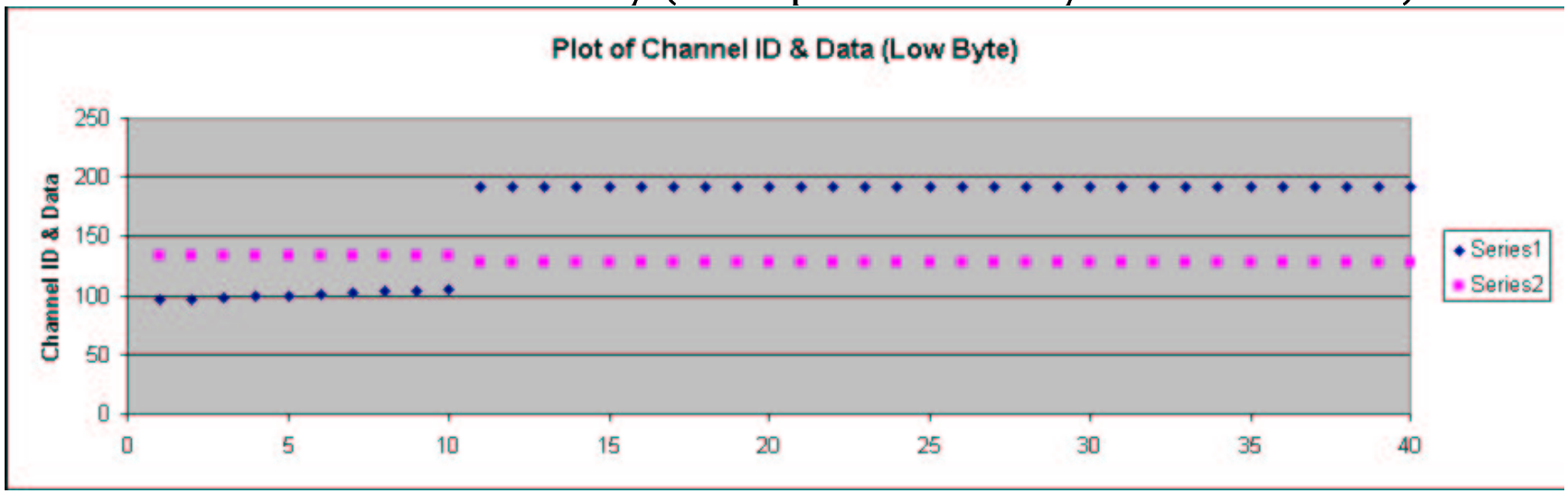
  5% with RTPS: problem with this hybrid (chip 4)?

  0.05% in data mode: general screw-ups
- zero ADC readouts in data mode (both 132ns/396ns)
- pedestals: cannot go up without nasty behaviour

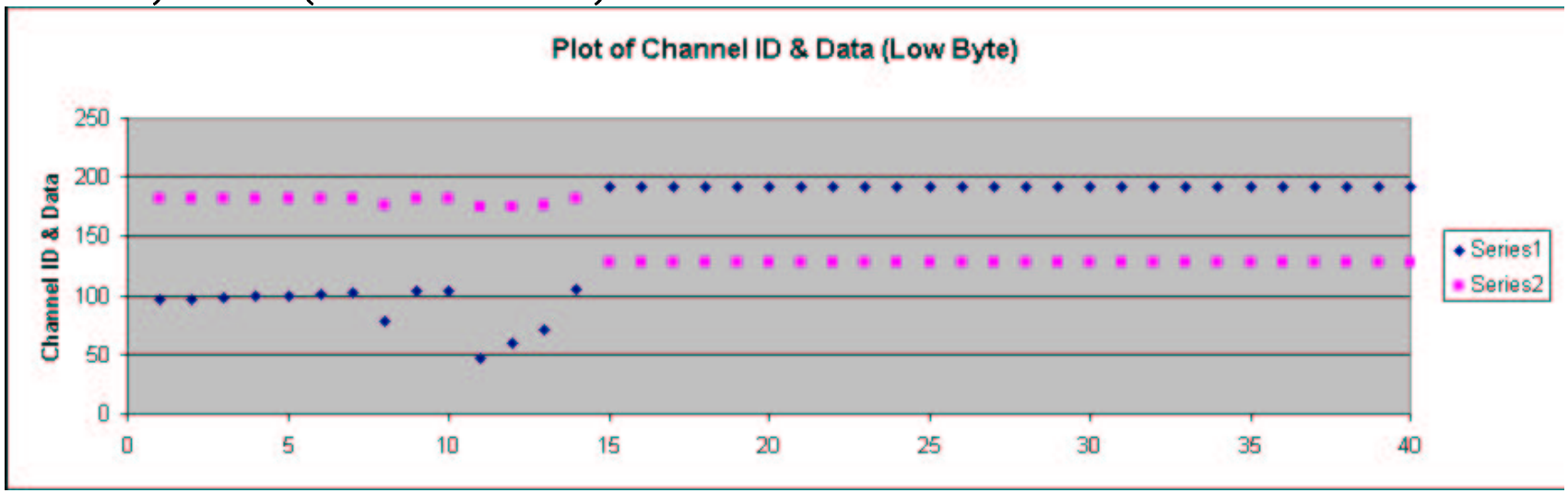
ADDENDUM:
readout in sparse/neighbor mode

Following the discussion at the meeting on Jan 6, we immediately implemented some small changes to the spreadsheet to allow us read and display events in sparse/neighbor mode properly. The plots on the following pages show some results we obtained.

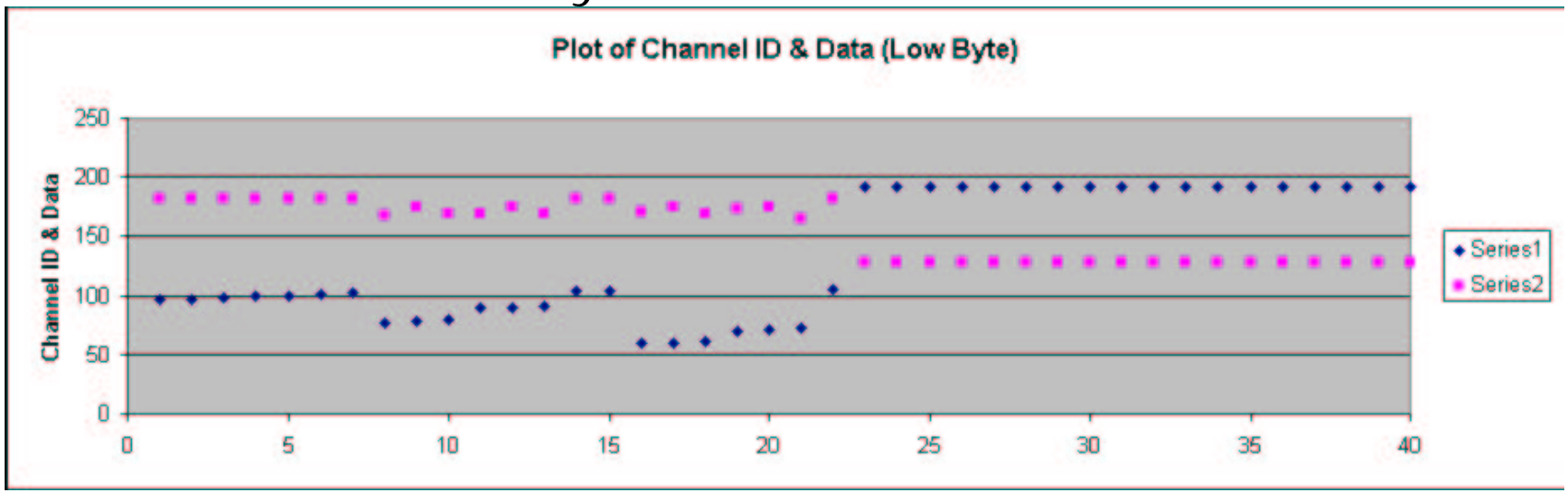
First we went to sparse mode with extremely high threshold. Only chip IDs are seen. Further entries are dummy (attempts to read beyond end of chain):



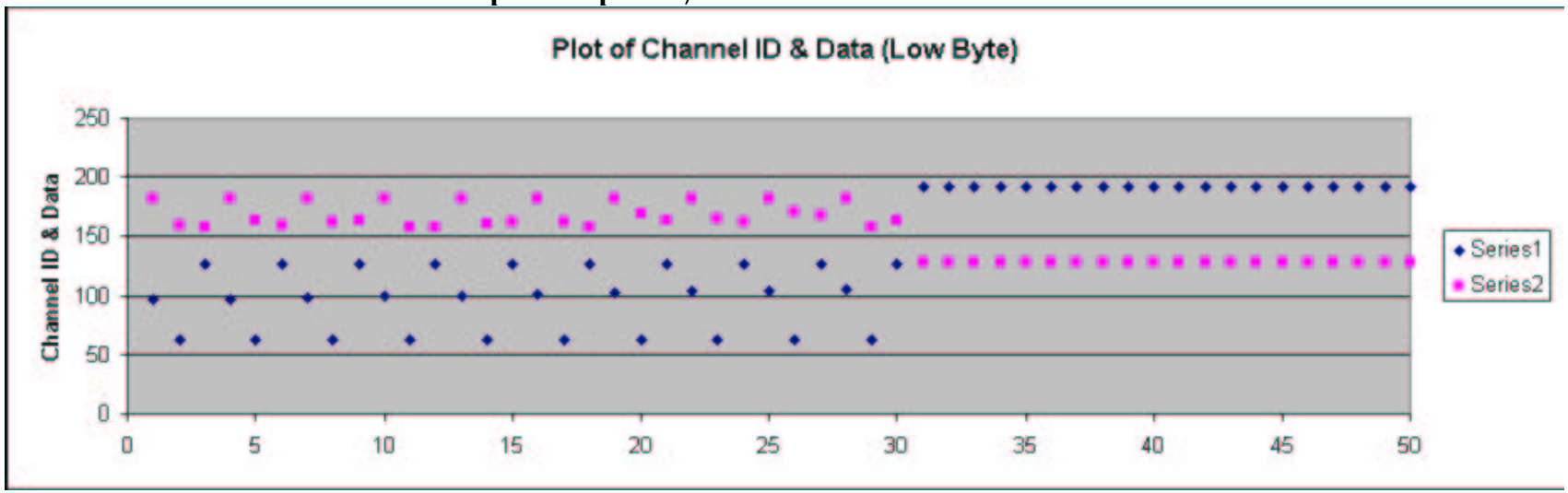
Occasionally, a few channels fluctuated above the threshold, here in chips 7 (one channel) and 9 (three channels):



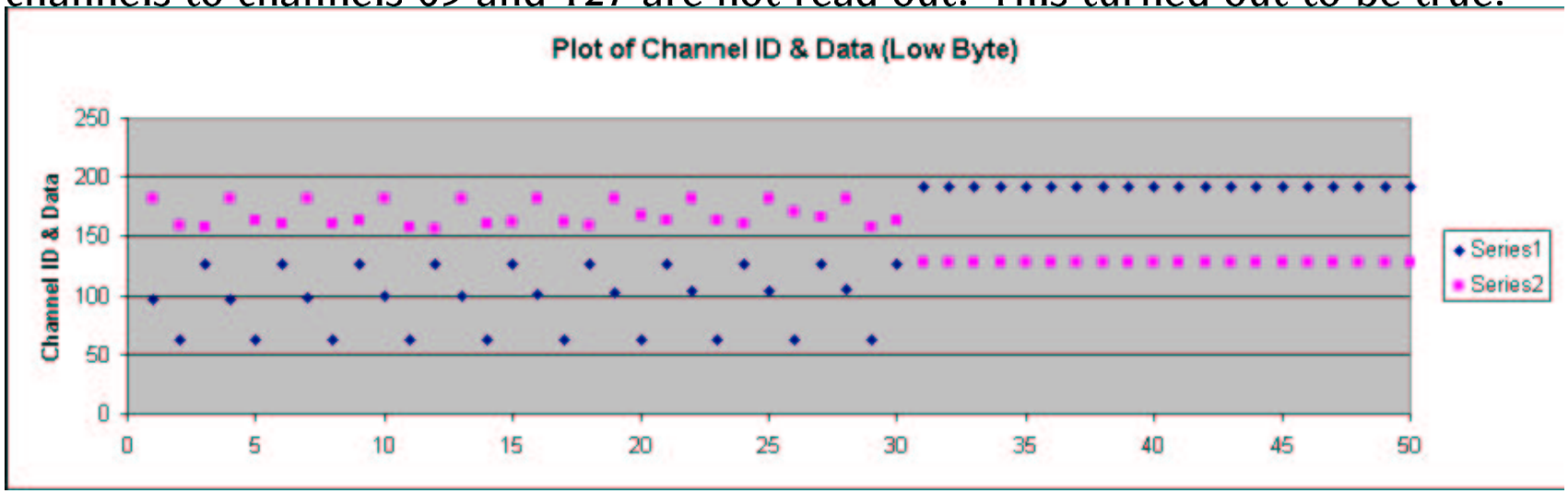
Similar fluctuations when neighbor mode is on:



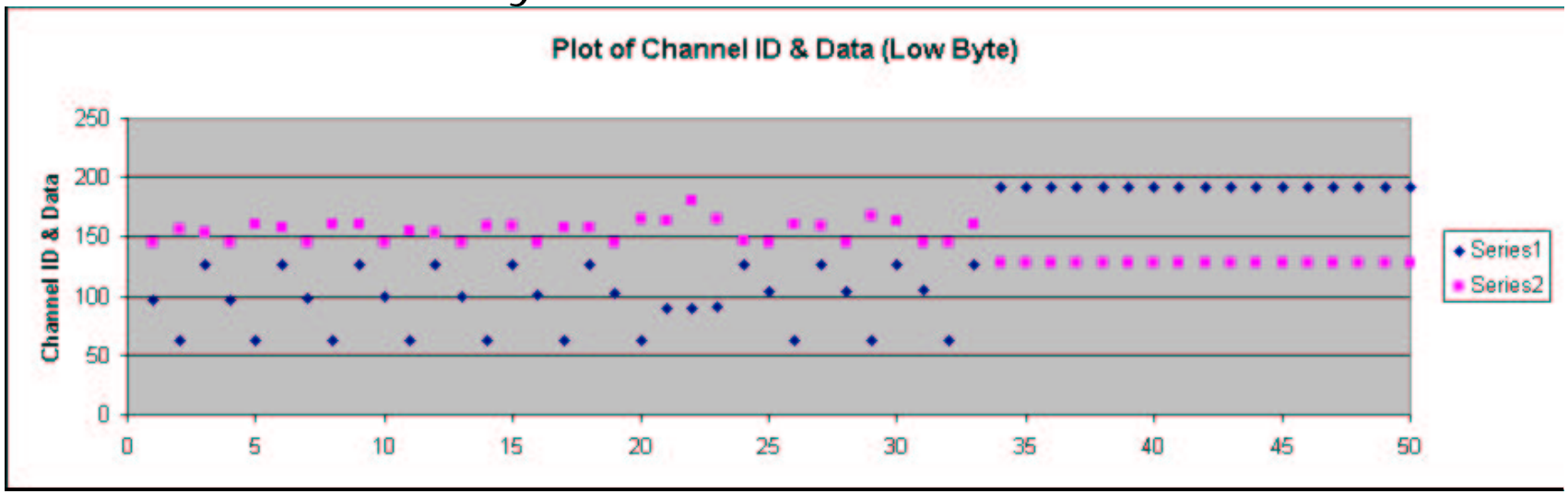
To see some more readouts, we switched on channels 63 and 127. Now we see three entries for each chip: chip ID, channel 63 and channel 127:



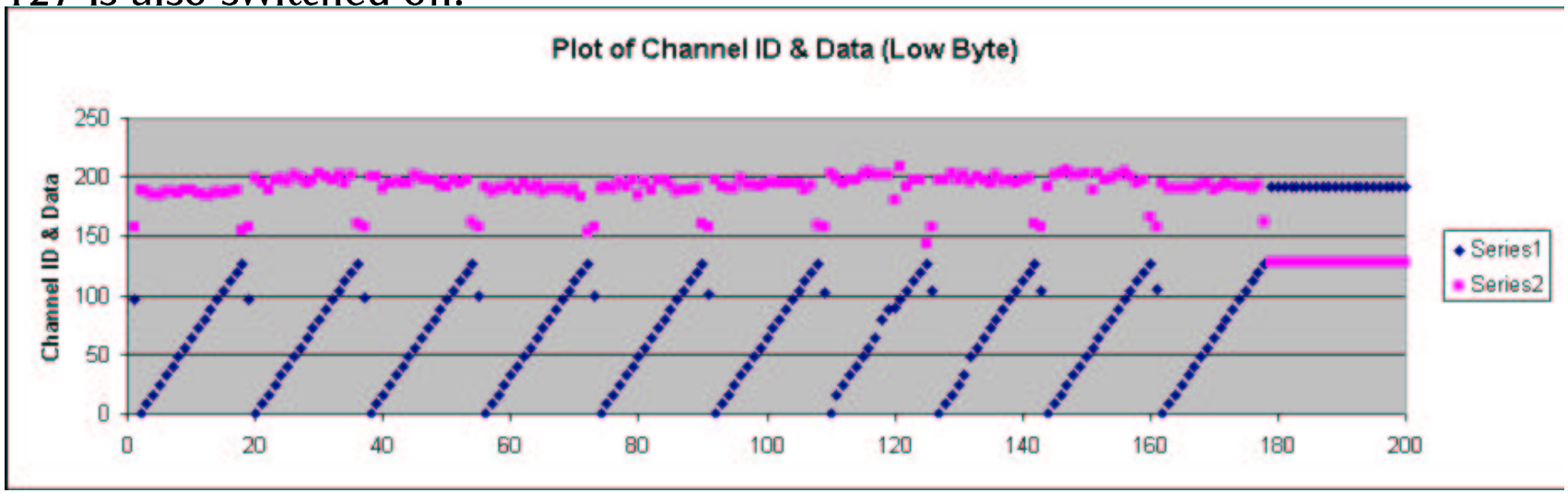
Next idea was to verify the statement that even in neighbor mode the neighboring channels to channels 63 and 127 are not read out. This turned out to be true:



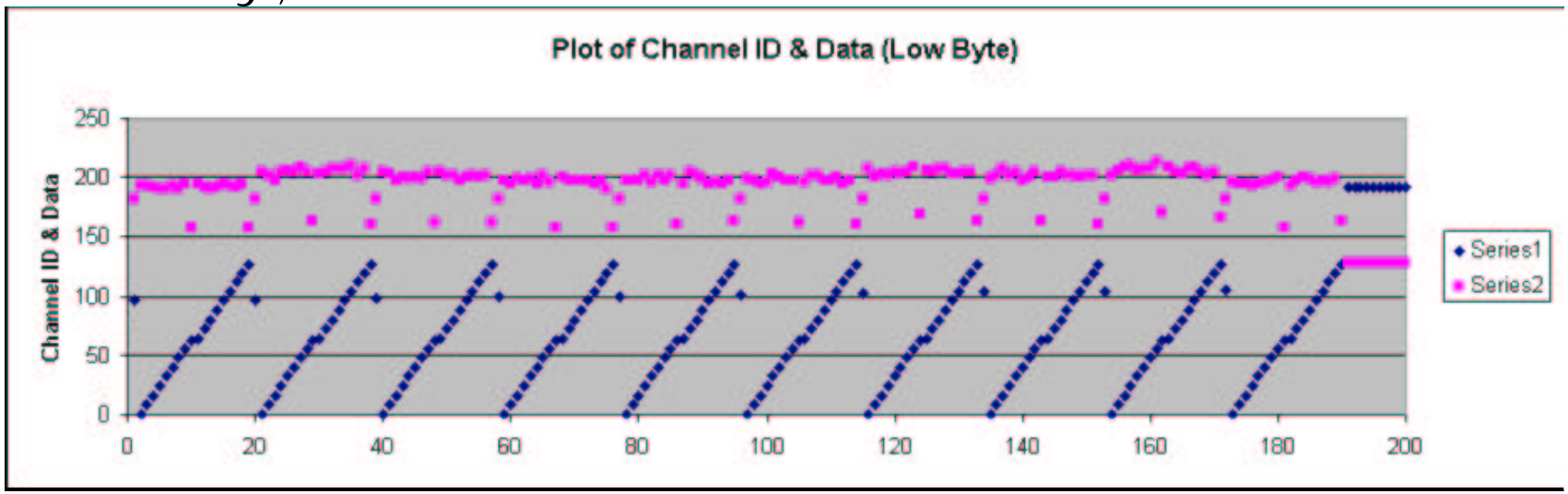
The following plot was done with the same setting, but shows one channel above threshold with its two neighbors:



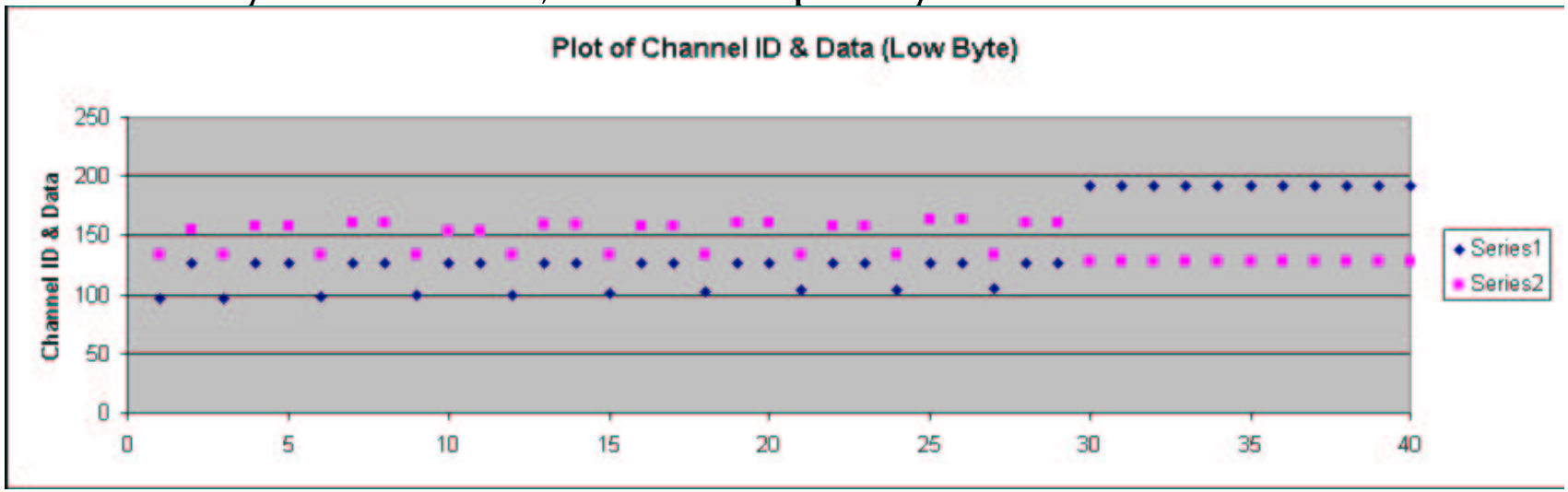
The following plot is done in sparse mode (no neighbors), calinjection and a threshold value of ee that lets calinjected channels pass, others not. Channel 127 is also switched on.



Same settings, but channels 127 and 63 on:



So far, everything worked out fine. Only possible problem we saw: The following plot shows readout in sparse mode with very high threshold (f8) and channel 127 switched on. Interestingly, 9 out of 10 chips give back TWO readouts of channel 127! We only saw this once, but will keep an eye on it.



Things to do immediately:
MCMUTIL2.DLL does not seem to support readout error analysis for sparse mode. We are currently ignoring the error flag assigned to each and every event in sparse/neighbor mode. A modification of either DLL or spreadsheet to fix this problem should not be too hard.
Noise analysis of channels 63 and 127 in sparse mode will presumably be done until next week.