

# IRQ & CRS0 Errors

Howard Budd, University of Rochester

Nov 3 2022

ArgonCube Meeting



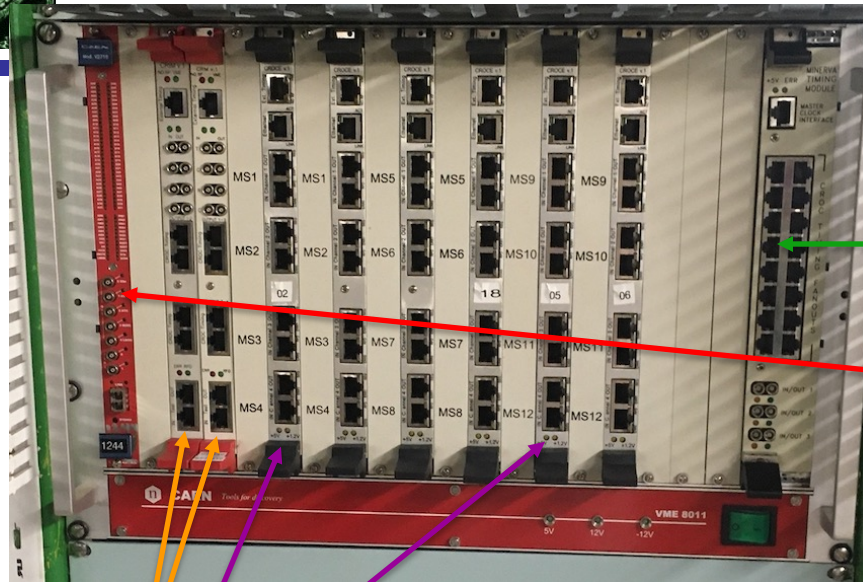
# IRQ & CRS0



- These are the 2 errors we have had during the assembly of MINERvA 2x2, IRQ (interrupt request), CRS0 (Continue Running Status = 0 )
  - Both of these problem ended the run in a “Event Builder” error
  - We did have occasional lock errors, but I expected those.
- As far as I know these problems did not exist in MINERvA, but the DAQ log file in MINERvA was not set to the level to show the problem.
- We will show what we did to try to understand these problems, so this is a bit detailed and is probably more directed to a small working group.
- It does give the details to show what we know.



# DAQ Crate



MTM

Crate Controller

- Crate
  - CRIM is the MINERvA timing module
  - CROCE reads out the FEB chain, this readout is called “The Sequencer”
  - 1 cable goes from CRIM -> CROCE, which gives has the clock
  - The clock has the encode “start gate” command which open the gate.
  - There is the Master CRIM which controls the clock and triggering, determined by a dip switch.
  - The Master CRIM tells crate controller it want attention. If the crate controller ignores the CRIM, the DAQ gives an IRQ error and which stops the run.



# IRQ Errors



- During the assembly almost all of these errors appeared during the PMT HV checkout.
  - Since the PMTs were so hard to replace, We ran the DAQ ~ 2-3 days after the assembly to be sure there were no HV problems. The HV problems would show up as HV spikes and wandering. We could easily replace the PMT if there was a HV problem. In MINERvA almost all PMTs were replace because of a HV problem.
- IRQ errors, did not appear during MINERvA running
- Start CRIM #6 S3, CRIM #12 S4 – none of below stopped the IRQ errors
  - Removed MTM from system– needed for beam running
  - CRIM #2 -> CRIM #12 S4 (S4 means slot 4)
  - CRIM #12 → CRIM #2 S4, CRIM #2 -> CRIM #6 S3,
    - CRIM #2 S3, CRIM #12 S4, address changed keep master CRIM S3
  - CRIM #2 <-> CRIM #12
    - CRIM #12 S3 CRIM #2 S4 – address not changed, master CRIM S4
  - Crate Controller 744 -> 1244



# IRQ errors



- CRIM #2 <-> CRIM #12
  - CRIM #2 S3, CRIM #12 S4
  - Light does not go on in CRIM #12 S4
  - CRIM #2 S3 is not recognized by slowcontrol
- CRIM #6 -> CRIM #12 S4
  - CRIM #6 light does not go on
  - CRIM 6 tested in Slot 5 – light does not go on
  - CRIM #2 S3, CRIM #12 S4, final configuration
- CRIMs in slot 2-3 not recognized by slowcontrol
- Crate replaced
  - Original hardware put back in original location
  - CRIM #6 S3, CRIM #12 S4
  - Slowcontrol now recognized both CRIMS
- IRQ error fixed, we have had 2 IRQ errors since then.
- Looks like the crate was origin of the IRQ errors



# However, Still not joy in DAQ land. CRS0



- Continue Running Status = 0 (CRS0)
- When the DAQ finishes a gate to goes to the next gate and checks
  - The variable: `continueRunning`
  - Initially set to 1. When it is set to 0, we get CRS0, and the run stops giving an event builder error.
  - Note that no error or warning is in the log file. Hence, CRS0 is not an error. Nobody takes responsibility for setting `continueRunning` to 0, everybody is silent, but the run ends.
  - It's not clear from the code who is setting `continueRunning = 0`. But I have some suspicions. I can smell the rat.



# Running with CROCS



- Configuration
  - CROCE 1 - West MS 1-4
  - CROCE 2 - East MS 1-4
  - CROCE 3 – West MS 4-8
  - CROCE 4 - East MS 4-8
  - CROCE 5 - West MS 9-11
  - CROCE 6 – East MS 9-11
- We started with Just MS 11 & we were getting CRS0, CROCE 5,6
- Run 78, MS 11 removed from DAQ, 78/10 CRS0 problem
- Run 79-80, Just CROCE5 removed from DAQ, 80/3 CRS0 problem
- Run 81, CROCE 5 in, CROCE 6 out, 81/1 CRS0 problem
- Run 82, only read out one FEB, MS8W FEB 1, 82/6 CRS0 Problem



# Running with CROCEs



- Run 101 CROCE 5,6 removed from DAQ
  - Subruns 4-14, 16-28, 23, 27-29 CRS0
- The CRIMs were not manipulated during Run 78-101
- Conclusion, CRS0 cannot be caused by one bad CROCE, chain, or FEB.
- Just 1 FEB in the DAQ, we were still getting the CRS0 & not in MS11. This was with HV on and off. HV off & no discriminator hits. Low data rate
- What about the CRIMs?
  - CROCE 1,2,3,4 get clock From CRIM S3
  - CROCE 5,6 gets clock from CRIM S4.
  - We get CRS0 errors with just CROCE 5,6 & just CROCE 3,4
  - Hence, the CRIMs cannot be causing the CRS0 problem
- CRS0 problem before and after the Crate Controller replaced
- CRS0 problem before and after the Crate was replaced
- The hardware can't cause the problem, unless multiple pieces of hardware are causing the problem.





# CRS0 – mnvonline07



- mnvonline07 – Geoff was setting up the DAQ and Abbey was getting the nearline going:
  - The CRS0 happen 7 times in run 1
    - subruns 15,18, 19, 20, 21, 25, & 28
    - There were a total 1 run, 40 subruns
  - However, there is a difference between what happen on mnvonline07 and what is happening on srv03, the DAQ computer
  - readoutTime = 3851 for subrun 15
  - This will be shown as:
    - run. subrun. last event. error. ReadoutTime
    - 1. 15. 6. CRS0. 3861
    - I will keep this showing the information this way, time microseconds
  - Note that all the CRS0 readout times for mnvonline07 are short, same as for normal gates, continue running status = 1 gates.
- The MINERvA log files were not useful for finding this error, but I don't remember the CRS0 problem. All problems which stopped the run were hardware errors such as sequencer or lock errors.



# srv03, 2x2 DAQ computer



- 160 CRS0 for the 2x2 DAQ, Almost all during the HV PMT checkout.
- 1st one
  - 8 15 6 CRS0 1036362
- The first one started just as we were starting, MS11
- These continued during our saga of replacing hardware
  - They were so frequent that for MS 4 & 5, I did not put the individual crashed in ECL. I collected them in a list and put the list in ECL For MS4 and MS5 test we have:
    - <https://dbweb9.fnal.gov:8443/ECL/argoncube/E/show?e=423>
    - <https://dbweb9.fnal.gov:8443/ECL/argoncube/E/show?e=395>
- A recent example
  - 115 1 753 CRS0 138256352
- In almost all cases
  - The readout time is long, so it looks like something is timing out



# CRS0



- However, in a couple of run the readout time is normal
  - 88. 14 1073 CRS0. 77170
  - 108 22 74 CRS0. 70543
  - Looks like in these events there was no time out.
- et, event transfer is a JLAB software that writes out the data. It can write out the data to the data file and to the nearline area where nearline histograms can be formed. As a JLAB product, nobody on MINERvA understood it. It just worked and we never had any problems with it.
- Occasionally, the DAQ called et and there was nothing in the logs after that, it looks like et crashed the DAQ.
  - 108 20. 552. last statement in log “Putting Event into ET System”
  - 94. 2. 646 is another example
  - The DAQ seem badly beaten up by et, it couldn't cope with that call to et. I had to gently nurse the DAQ back to life.
- Putting all the CRS0 problems in a list for MS4 & 5 made it easier



et



- Possible problem
  - Some data is being passed to et which causes CRS0
    - As shown above, this does not seem likely.
    - However, sometimes it would crash after 10 event, sometimes 6500 events or more. It usually had crashed after 500 events. This variation looks more like a hardware problem than software problem.
  - et does not compile correctly
    - It ran for MINERvA.
    - We did not have this problem with MINERvA, but the MINERvA code had to be compiled with SLF7 instead of SLF6. et appears to use JAVA.
  - ?
- A clue is the long readout time, but there are occasional runs that have the CRS0 problem with short readout time. The DAQ for mnvonline07 had no long readout times when there were CRS0 failures.
- There are 2 versions of et in the DAQ directory. We could try to compile it with the other version.
  - Looks like they were written at the same time. I don't know the difference.



# et, JLab



- When the DAQ people wanted to switch from Windows DAQ to Linux DAQ one of the RA got et from JLAB.
- In looking in the et code, the author of one of the routines is given in the routine
  - Carl Zimmer
    - Looks like he is still at JLab
  - The site suggests sending issues to [coda@jlab.org](mailto:coda@jlab.org), the Jlab DATA Acquisition group.
    - <https://coda.jlab.org/drupal/content/about-us>
  - About et
    - <https://coda.jlab.org/drupal/content/event-transfer-et>
- This looks like one of the avenues to pursue.