

# KCU/H2GCROC2 decoding

---

You saw the mattermost chatter from the weekend - I have what I believe is a working decoder that

- Now gets all the samples
- And now also gets all the channels (288)

This was something that we appear to have just gotten lucky with, that the clocks on both cards were in sync. But I believe that the extern FMC cable makes that “same-time start” marginal.

In any case, we must be able to decode this.

I checked with old data, and there is no change in the data that are getting decoded from the BNL one-card setup.

# Here is the TL;DR

---

Channels 0..71 see no change

The old code (assuming one ROC board only, and chip numbers 0 and 1 only) went

```
int chip_offset =0;
if ( ichip) chip_offset = 72;

int group_offset =0;
if ( group) group_offset = 36;
```

So all ichip={1,2,3} would go into (and overwrite) channels 72++

What goes there and “survives” is determined by which chip info comes last

# Old code checks

---

I used KCU0's run 6 (/scratch/data/KCU0/junk/junk\_ROC\_0-00000006-0000.evt ) to this

I re-installed the original version from git (I did this on my machine not to disturb anyone one carbon).

Just for reference, here is the version before my changes:

```
commit 81f1221a623120e33d1af0979152a3eed541cfef (HEAD -> ePIC, origin/ePIC)
Author: Martin L Purschke <purschke@bnl.gov>
Date:   Sun Apr 5 19:14:14 2026 -0400
    mlp - committing the changes for the astropix decoder that I made a JLab on
the test beam floor
```

```
commit a7da5b11a9539655eaebfa4c73fcd87a172e3cc6
Author: Martin L Purschke <purschke@bnl.gov>
Date:   Sun Apr 5 19:08:24 2026 -0400
    mlp - a fix to deal with a new firmware that introduced a "jitter"
between same-sample timestamps
```

```
commit 9a635aa28d63bfec7313c898814d15be7ac38412 (github/ePIC)
Author: Martin L Purschke <purschke@bnl.gov>
Date:   Tue Mar 10 01:37:05 2026 -0400
```

# Old version

---

I checked out that version

```
$ git checkout 9a635aa28d63bfec7313c898814d15be7ac38412
```

And rebuilt with

```
$ make clean && make -j install
```

There is no need to rebuild the pmonitor project, it just uses the event libs even if they change

# New code

---

The new code does it correctly:

```
chip_offset = ichip * 72; // ichip is 0,1,2 or 3

int group_offset = 0;
if ( group) group_offset = 36; // group is 0 or 1
for ( int c = 0; c < 36; c++)
{
    _sample->ADC[chip_offset + group_offset + c] = (ch[c] >> 20) & 0x3ff;
    // and so on
}
```

So channel =  $ichip * 72 + group * 36$

# Some debug output

---

Color-Coded by group...

```
ichip = 0 group 0 trigger_in 0 trigger_out 1667 evt_ctr 0 timestamp 3abdd4d
ichip = 1 group 0 trigger_in 0 trigger_out 1667 evt_ctr 0 timestamp 3abdd4d
ichip = 2 group 0 trigger_in 0 trigger_out 1667 evt_ctr 0 timestamp 3abdd4d
ichip = 3 group 0 trigger_in 0 trigger_out 1667 evt_ctr 0 timestamp 3abdd4d
ichip = 0 group 1 trigger_in 0 trigger_out 1667 evt_ctr 0 timestamp 3abdd4d
ichip = 1 group 1 trigger_in 0 trigger_out 1667 evt_ctr 0 timestamp 3abdd4d
ichip = 2 group 1 trigger_in 0 trigger_out 1667 evt_ctr 0 timestamp 3abdd4d
ichip = 3 group 1 trigger_in 0 trigger_out 1667 evt_ctr 0 timestamp 3abdd4d

ichip = 0 group 0 trigger_in 0 trigger_out 1667 evt_ctr 1 timestamp 3abddf1
ichip = 1 group 0 trigger_in 0 trigger_out 1667 evt_ctr 1 timestamp 3abddf1
ichip = 2 group 0 trigger_in 0 trigger_out 1667 evt_ctr 1 timestamp 3abddf1
ichip = 3 group 0 trigger_in 0 trigger_out 1667 evt_ctr 1 timestamp 3abddf1
ichip = 0 group 1 trigger_in 0 trigger_out 1667 evt_ctr 1 timestamp 3abddf1
ichip = 1 group 1 trigger_in 0 trigger_out 1667 evt_ctr 1 timestamp 3abddf1
ichip = 2 group 1 trigger_in 0 trigger_out 1667 evt_ctr 1 timestamp 3abddf1
ichip = 3 group 1 trigger_in 0 trigger_out 1667 evt_ctr 1 timestamp 3abddf1
```

# Let's look at these chips/groups:

---

According to the new math, the channels from these ones

```
ichip = 3 group 0 trigger_in 0 trigger_out 1667 evt_ctr 0 timestamp 3abdd4d  
ichip = 3 group 1 trigger_in 0 trigger_out 1667 evt_ctr 0 timestamp 3abdd4d
```

go to channels  $3*72 = 216$  and

$3*72 + 36 = 252$  and up, respectively

The old, wrong calculation would have put them at channels 72++ and 108++

And the groups 1 and 2 would show up nowhere before

# ddump output, old vs new

---

Old: This would be the ichip=3, group=0 channel -> 72

72	ADC	53	57	54	50	54	54	53	52	53	53	52	52	51	54	52	55	54	54	56
72	TOT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72	TOA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The new code puts this to the right channel (216):

216	ADC	53	57	54	50	54	54	53	52	53	53	52	52	51	54	52	55	54	54	56
216	TOT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
216	TOA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Ditto for ichip=3, group=1 - ch 108 (wrong) and ch 252 (correct)

108	ADC	4d	50	51	50	51	52	50	50	50	52	51	51	50	50	51	51	52	51	51
108	TOT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
108	TOA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

252	ADC	4d	50	51	50	51	52	50	50	50	52	51	51	50	50	51	51	52	51	51
252	TOT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
252	TOA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

# Summary -

---

I think this is it

- We have all 288 channels
- The channels come out the way they should
- I'll commit these changes 2and then we are good.